



**PENGEMBANGAN
SISTEM INFORMASI REKAM MEDIS PASIEN KLINIK
HARAPAN BERSAMA AMBULU MENGGUNAKAN METODE
SDLC MODEL WATERFALL**

SKRIPSI

oleh

**Lukman Hadiyatulloh
NIM 102410101068**

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER
2018**



**PENGEMBANGAN
SISTEM INFORMASI REKAM MEDIS PASIEN KLINIK
HARAPAN BERSAMA AMBULU MENGGUNAKAN METODE
SDLC MODEL *WATERFALL***

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Sistem Informasi (S1)
dan mencapai gelar Sarjana Komputer

oleh

**Lukman Hadiyatulloh
NIM 102410101068**

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER
2018**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Alloh Subhaanau Wata'ala;
2. Keluargaku tercinta(Ibunda Riftakhul Janah, Ayahanda Sudarto, Adikku Tika Chusniatur Rohmah, Mbah Putri Siti Anisah);
3. Teman spesialku Cindy Septian Ayu Saputri;
4. Guru mengajiku yang telah banyak mengajariku ilmu agama;
5. Guru-guruku sejak sekolah dasar sampai dengan perguruan tinggi;
6. Semua temanku dan semua sahabatku;
7. Almamater Program Studi Sistem Informasi Universitas Jember.

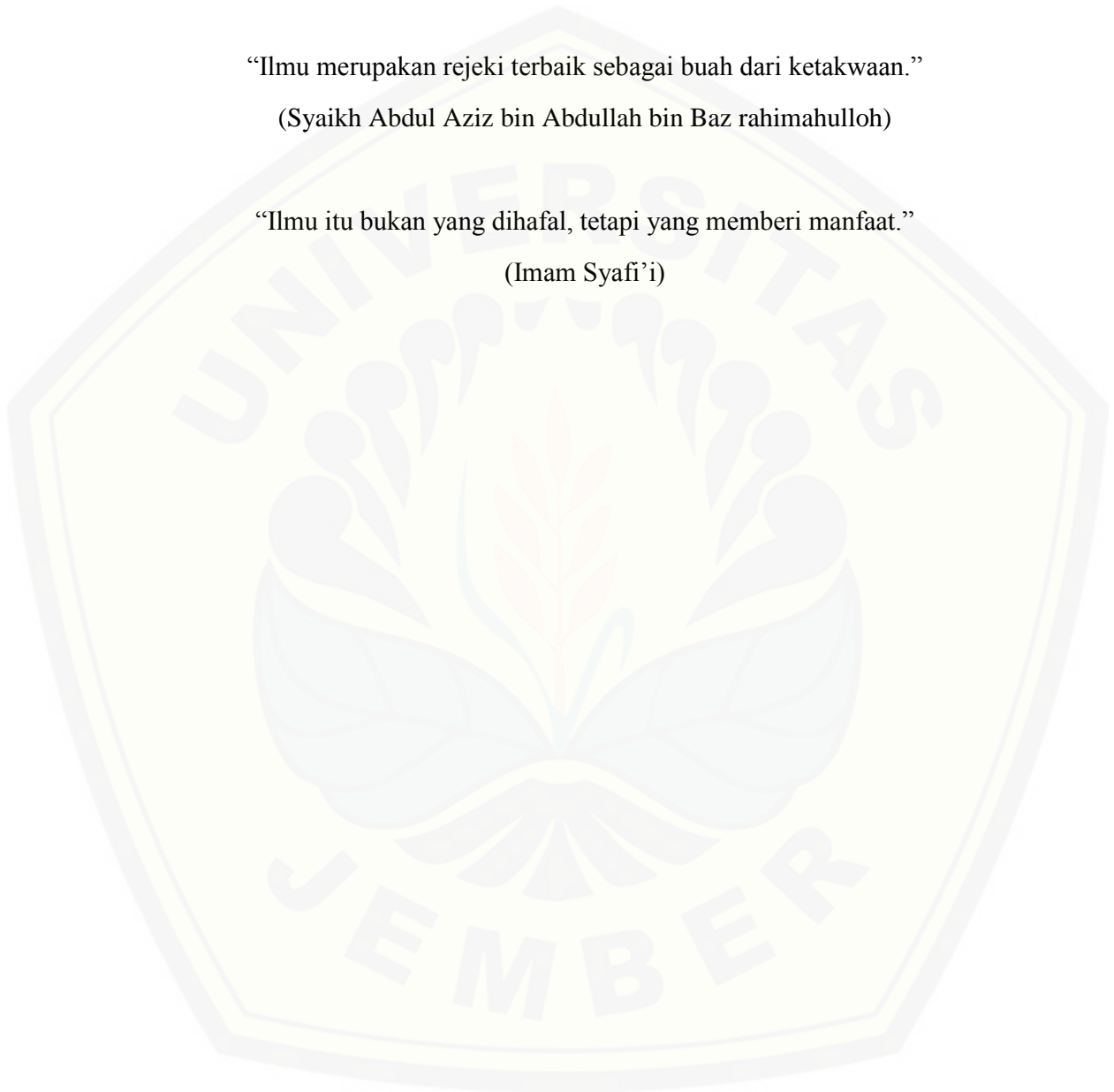
MOTO

“Ilmu merupakan rejeki terbaik sebagai buah dari ketakwaan.”

(Syaiikh Abdul Aziz bin Abdullah bin Baz rahimahulloh)

“Ilmu itu bukan yang dihafal, tetapi yang memberi manfaat.”

(Imam Syafi’i)



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Lukman Hadiyatulloh

NIM : 102410101068

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Pengembangan Sistem Informasi Rekam Medis Pasien Klinik Harapan Bersama Ambulu Menggunakan Metode SDLC Model *Waterfall*”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 17 Januari 2018

Yang menyatakan,

Lukman Hadiyatulloh
NIM 102410101068

SKRIPSI

**PENGEMBANGAN
SISTEM INFORMASI REKAM MEDIS PASIEN KLINIK
HARAPAN BERSAMA AMBULU MENGGUNAKAN METODE
SDLC MODEL *WATERFALL***

oleh

Lukman Hadiyatulloh

NIM 102410101068

Pembimbing

Dosen Pembimbing Utama : Anang Andrianto S.T., MT.

NIP. 19690615 199702 1 002

Dosen Pembimbing Anggota : Windi Eka Yulia Retnani, S.Kom., MT

NIP. 19840305 201012 2 002

PENGESAHAN PEMBIMBING

Pengembangan Sistem Informasi Rekam Medis Pasien Klinik Harapan Bersama Ambulu Menggunakan Metode SDLC Model *Waterfall* telah diuji dan disahkan pada:

Hari, tanggal :

Tempat : Program Studi Sistem Informasi Universitas Jember

Tim Pembimbing:

Pembimbing I,

Pembimbing II,

Anang Andrianto S.T., MT.

NIP. 19690615 199702 1 002

Windi Eka Yulia Retnani, S.Kom., MT.

NIP. 19840305 201012 2 002

PENGESAHAN

Skripsi berjudul “**Pengembangan Sistem Informasi Rekam Medis Pasien Klinik Harapan Bersama Ambulu Menggunakan Metode SDLC Model *Waterfall***”, telah diuji dan disahkan pada:

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim Penguji:

Penguji I,

Penguji II,

Nelly Oktavia Adiwijaya, S.Si., MT.

NIP. 19841024 200912 2 008

Fahrobby Adnan, S.Kom., M.MSI.

NIP. 19870619 201404 1 001

Mengesahkan

Ketua Program Studi,

Prof. Drs. Slamini, M.Comp.Sc.,Ph.D.

NIP. 19670420 199201 1 001

RINGKASAN

Pengembangan Sistem Informasi Rekam Medis Pasien Klinik Harapan Bersama Ambulu Menggunakan Metode SDLC Model *Waterfall*; Lukman Hadiyatulloh, 102410101068; 2018 : 352 halaman; Program Studi Sistem Informasi Universitas Jember.

Kesehatan merupakan hal yang terpenting untuk setiap orang sehingga menjadi suatu kewajiban untuk selalu tetap dijaga. Saat memeriksakan diri ke klinik atau rumah sakit terdekat, sering setiap orang yang secara langsung dapat dikatakan sebagai pasien biasanya ditanya tentang data diri dan riwayat penyakit. Proses pencatatan data riwayat penyakit pasien kebanyakan masih dilakukan secara manual sehingga tak jarang dari catatan tersebut sulit ditemukan pada saat mencarinya, atau bahkan hilang. Di sisi lain, pencatatan riwayat penyakit pasien tersebut sangat diperlukan apabila pasien tersebut berkunjung dan berobat kembali ke rumah sakit atau ke klinik yang sama.

Pencatatan data rekam medis Klinik Harapan Bersama Ambulu masih belum optimal sebab dilakukan secara manual sehingga masih kurang efektif dan efisien dalam pengolahan datanya. Maka perlu diadakan sebuah pengembangan sistem guna diterapkan pada Klinik Harapan Bersama. Dalam pengembangan sebuah sistem atau perangkat lunak, dibutuhkan suatu metode untuk mendefinisikan berbagai kebutuhan pada saat pengembangan perangkat lunak atau sistem tersebut. Peneliti menggunakan metode SDLC model *Waterfall* untuk pengembangan sistem yang ada di Klinik Harapan Bersama. Model *Waterfall* terdiri dari tahap analisis kebutuhan, desain sistem, pengkodean, pengujian, dan pemeliharaan. Model ini dipilih oleh peneliti sebab kebutuhan untuk pengembangan sistem pada Klinik Harapan Bersama telah terdefiniskan mulai pada tahap awal serta pengembangan sistem dikerjakan secara terurut berdasarkan tahap pada model *Waterfall*.

Penelitian ini menghasilkan beberapa kesimpulan yakni: (1) Sistem Informasi Rekam Medis Klinik Harapan Bersama dapat meningkatkan efektifitas, efisiensi, dan

relevan pada saat pencatatan data pasien di Klinik Harapan Bersama. (2) Sistem Informasi Rekam Medis Klinik Harapan Bersama merupakan sistem informasi berbasis *web* dibangun berdasarkan pada kebutuhan Klinik Harapan Besama Ambulu. (3) Sistem Informasi Rekam Medis Klinik Harapan Bersama terdiri dari fitur utama yaitu pencatatan data rekam medis pasien atau Manajemen data Rekam Medis serta terdapat fitur tambahan yaitu Autentifikasi (*Login*), Manajemen Data *User*, Manajemen Data Pegawai, Manajemen Data Pasien, Manajemen Data Kamar, Manajemen Data Fasilitas, Manajemen Data Dokter, Manajemen Data Obat, Manajemen Data Laborat, Manajemen Data Rujukan, Manajemen Transaksi Pembayaran, dan Melihat Laporan.

Saran untuk penelitian ini adalah pada pengembangan lebih lanjut dari dapat dilakukan dengan menambahkan ruang lingkup penelitian dan hasil dari penelitian yang lebih lengkap. Pengembangan lebih lanjut dapat dilakukan dengan penambahan fitur yang tidak terlihat terlalu kaku misal pada saat melihat dan membaca laporan. Penambahan fitur yang dimaksud misalnya dengan penambahan tampilan grafik untuk menampilkan banyaknya pengunjung klinik berdasarkan hari, minggu, dan bulan.

PRAKATA

Puji syukur ke hadirat Allah Sub hanahu Wata'ala atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Pengembangan Sistem Informasi Rekam Medis Pasien Klinik Harapan Bersama Ambulu Menggunakan Metode SDLC Model *Waterfall*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamir, M.CompSc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Anang Andrianto S.T.,MT., selaku Dosen Pembimbing Utama dan Windi Eka Yulia Retnani, S.Kom., MT., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Prof. Dr. Saiful Bukhori, ST., M.Kom., selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Pimpinan dan Seluruh Staff dan Karyawan Klinik Harapan Bersama Ambulu;
6. Semua pihak yang tidak dapat disebutkan satu-persatu.

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 17 Januari 2018

Penulis

DAFTAR ISI

	Halaman
HALAMAN SAMPUL.....	i
HALAMAN JUDUL	ii
PERSEMBAHAN.....	iii
MOTO.....	iv
PERNYATAAN	v
HALAMAN PEMBIMBINGAN.....	vi
PENGESAHAN PEMBIMBING	vii
PENGESAHAN PENGUJI.....	viii
RINGKASAN	ix
PRAKATA.....	xi
DAFTAR ISI.....	xii
DAFTAR TABEL.....	xv
DAFTAR GAMBAR	xvi
DAFTAR LAMPIRAN.....	xviii
BAB 1. PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	3
1.3 Tujuan dan Manfaat	3
1.3.1 Tujuan.....	4
1.3.2 Manfaat	4
1.4 Ruang Lingkup.....	4
1.5 Sistematika Penulisan.....	5
BAB 2. TINJAUAN PUSTAKA	6
2.1 Klinik.....	6
2.2 Klinik Harapan Bersama	6
2.3 Rekam Medis Pasien	7

2.4	Sistem Informasi Rekam Medis Pasien.....	9
2.5	Metode SDLC Model <i>Waterfall</i>	9
BAB 3. METODOLOGI PENELITIAN		12
3.1	Analisis Kebutuhan	12
3.1.1	Pengumpulan Data.....	12
3.1.2	Gambaran Umum Sistem.....	13
3.2	Desain Sistem.....	14
3.2.1	<i>Business Process</i>	14
3.2.2	<i>Use Case Diagram</i>	14
3.2.3	<i>Use Case Scenario</i>	14
3.2.4	<i>Activity Diagram</i>	14
3.2.5	<i>Sequence Diagram</i>	14
3.2.6	<i>Class Diagram</i>	14
3.2.7	<i>Entity Relationship Diagram (ERD)</i>	14
3.3	Implementasi	15
3.4	Pengujian.....	15
3.5	Pemeliharaan	15
BAB 4. ANALISIS DAN PENGEMBANGAN SISTEM.....		16
4.1	Analisis Kebutuhan	16
4.1.1	Pengumpulan Data.....	16
4.1.2	Kebutuhan Fungsional.....	18
4.1.3	Kebutuhan Non Fungsional	18
4.2	Desain Sistem.....	18
4.3.1	<i>Bussines Process</i>	19
4.3.2	<i>Use case Diagram</i>	19
4.3.3	<i>Use case Skenario</i>	23
4.3.4	<i>Activity Diagram</i>	34
4.3.5	<i>Sequence Diagram</i>	41

4.3.6	<i>Class Diagram</i>	53
4.3.7	<i>Entity Relationship Diagrams (ERD)</i>	55
4.4	Implementasi	55
4.5	Pengujian Program	56
4.5.1	Pengujian <i>White Box</i>	56
4.5.2	Pengujian <i>Black Box</i>	71
4.6	Pemeliharaan	71
BAB 5. HASIL DAN PEMBAHASAN		72
5.1	Implementasi Sistem	72
5.1.1	Autentifikasi (<i>Login</i>)	72
5.1.2	Halaman <i>Dashboard</i>	73
5.1.3	Halaman Pegawai	74
5.1.4	Halaman Pasien	75
5.1.5	Halaman Kamar	76
5.1.6	Halaman Fasilitas	76
5.1.7	Halaman Dokter	77
5.1.8	Halaman Obat	77
5.1.9	Halaman Laborat	78
5.1.10	Halaman Rekam Medis	79
5.1.11	Halaman Rujukan	79
5.1.12	Halaman Transaksi Pembayaran	80
5.1.13	Halaman Laporan	80
BAB 6. PENUTUP		81
6.1	Kesimpulan	81
6.2	Saran	81
DAFTAR PUSTAKA		82
LAMPIRAN		84

DAFTAR TABEL

	Halaman
Tabel 4. 1 Definisi Aktor	21
Tabel 4. 2 Definisi <i>Use case</i>	21
Tabel 4. 3 <i>Use case</i> Skenario Autentifikasi	23
Tabel 4. 4 Use Case Skenario Manajemen Data Rekam Medis	28
Tabel 4. 5 <i>Test Case</i> Autentifikasi (<i>Login</i>) jalur 1	58
Tabel 4. 6 <i>Test Case</i> Autentifikasi (<i>Login</i>) jalur 2.....	58
Tabel 4. 7 <i>Test Case</i> Autentifikasi (<i>Login</i>) jalur 3.....	58
Tabel 4. 8 <i>Test Case</i> Autentifikasi (<i>Login</i>) jalur 4.....	59
Tabel 4. 9 <i>Test Case</i> Autentifikasi (<i>Login</i>) jalur 5.....	59
Tabel 4. 10 <i>Test Case</i> Autentifikasi (<i>Login</i>) jalur 6.....	59
Tabel 4. 11 <i>Test Case</i> Fitur Manajemen Data Rekam Medis fungsi <i>construct</i> jalur 167	
Tabel 4. 12 <i>Test Case</i> Fitur Manajemen Data Rekam Medis fungsi <i>construct</i> jalur 268	
Tabel 4. 13 <i>Test Case</i> Fitur Manajemen Data Rekam Medis fungsi <i>Index</i> jalur 1	68
Tabel 4. 14 <i>Test Case</i> Fitur Manajemen Data Rekam Medis fungsi <i>Add</i> jalur 1.....	68
Tabel 4. 15 <i>Test Case</i> Fitur Manajemen Data Rekam Medis fungsi <i>Add</i> jalur 2.....	69
Tabel 4. 16 <i>Test Case</i> Fitur Manajemen Data Rekam Medis fungsi <i>Add</i> jalur 3.....	69
Tabel 4. 17 <i>Test Case</i> Fitur Manajemen Data Rekam Medis fungsi <i>Add</i> jalur 4.....	69
Tabel 4. 18 <i>Test Case</i> Fitur Manajemen Data Rekam Medis fungsi <i>Edit</i> jalur 1	70
Tabel 4. 19 <i>Test Case</i> Fitur Manajemen Data Rekam Medis fungsi <i>Edit</i> jalur 2	70
Tabel 4. 20 <i>Test Case</i> Fitur Manajemen Data Rekam Medis fungsi <i>Edit</i> jalur 3	70
Tabel 4. 21 <i>Test Case</i> Fitur Manajemen Data Rekam Medis fungsi <i>Delete</i> jalur 1....	70
Tabel 4. 22 <i>Test Case</i> Fitur Manajemen Data Rekam Medis fungsi <i>Pdf</i> jalur 1	71

DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Tahapan model <i>Waterfall</i>	10
Gambar 4. 1 <i>Bussines Process</i> Sistem Informasi Rekam Medis Pasien.....	19
Gambar 4. 2 <i>Use case</i> Diagram Sistem Informasi Rekam Medis.....	20
Gambar 4. 3 <i>Activity</i> Diagram <i>Login</i>	35
Gambar 4. 4 <i>Activity</i> Diagram Manajemen Data Rekam Medis.....	39
Gambar 4. 5 <i>Sequence</i> Diagram Autentifikasi (<i>Login</i>).....	41
Gambar 4. 6 <i>Sequence</i> Diagram Manajemen Data Rekam Medis.....	45
Gambar 4. 7 <i>Sequence</i> Diagram Transaksi Pembayaran Perawatan.....	46
Gambar 4. 8 <i>Sequence</i> Diagram Transaksi Pembayaran Laboratorium	47
Gambar 4. 9 <i>Sequence</i> Diagram Transaksi Pembayaran Obat.....	47
Gambar 4. 10 <i>Sequence</i> Diagram Laporan Pasien	48
Gambar 4. 11 <i>Sequence</i> Diagram Laporan Kamar	49
Gambar 4. 12 <i>Sequence</i> Diagram Laporan Dokter	49
Gambar 4. 13 <i>Sequence</i> Diagram Laporan Pegawai.....	50
Gambar 4. 14 <i>Sequence</i> Diagram Laporan Fasilitas	50
Gambar 4. 15 <i>Sequence</i> Diagram Laporan Obat.....	51
Gambar 4. 16 <i>Sequence</i> Diagram Laporan Laboratorium	51
Gambar 4. 17 <i>Sequence</i> Diagram Laporan Rekam Medis	52
Gambar 4. 18 <i>Sequence</i> Diagram Laporan Rujukan.....	52
Gambar 4. 19 <i>Class</i> Diagram Sistem Informasi Rekam Medis Pasien.....	54
Gambar 4. 20 ERD Sistem Informasi Rekam Medis Pasien	55
Gambar 4. 21 Diagram Alir Autentifikasi (<i>Login</i>)	57
Gambar 4. 22 Grafik Alir Manajemen Data Rekam Medis Fungsi <i>Construct</i>	63
Gambar 4. 23 Grafik Alir Manajemen Data Rekam Medis Fungsi <i>Index</i>	63
Gambar 4. 24 Grafik Alir Manajemen Data Rekam Medis Fungsi <i>Add</i>	64
Gambar 4. 25 Diagram Alir Manajemen Data Rekam Medis Fungsi <i>Edit</i>	64

Gambar 4. 26 Grafik Alir Manajemen Data Rekam Medis Fungsi <i>Delete</i>	65
Gambar 4. 27 Grafik Alir Manajemen Data Rekam Medis Fungsi <i>Pdf</i>	65
Gambar 5. 1 Halaman Autentifikasi (Login) Sistem Informasi Rekam Medis Pasien	72
Gambar 5. 2 Halaman <i>Dashboard</i> Untuk Aktor dengan <i>Role</i> Admin	73
Gambar 5. 3 Halaman <i>Dashboard</i> Untuk Aktor dengan <i>Role</i> Manajer.....	73
Gambar 5. 4 Halaman <i>Dashboard</i> Untuk Aktor dengan <i>Role</i> Apoteker	74
Gambar 5. 5 Halaman <i>Dashboard</i> Untuk Aktor dengan <i>Role</i> Laborat.....	74
Gambar 5. 6 Halaman Pegawai.....	75
Gambar 5. 7 Halaman Pasien.....	75
Gambar 5. 8 Halaman Kamar	76
Gambar 5. 9 Halaman Fasilitas.....	77
Gambar 5. 10 Halaman Dokter	77
Gambar 5. 11 Halaman Obat.....	78
Gambar 5. 12 Halaman Laborat.....	78
Gambar 5. 13 Halaman Rekam Medis	79
Gambar 5. 14 Halaman Rujukan.....	80

DAFTAR LAMPIRAN

	Halaman
LAMPIRAN A. DESAIN SISTEM	84
A.1 <i>Use case</i> Skenario	84
A.1.1 <i>Use case</i> Skenario Manajemen Data <i>User</i>	84
A.1.2 <i>Use case</i> Skenario Manajemen Data Pegawai	87
A.1.3 <i>Use case</i> Skenario Manajemen Data Pasien	91
A.1.4 <i>Use case</i> Skenario Manajemen Data Kamar	95
A.1.5 <i>Use case</i> Skenario Manajemen Data Fasilitas	100
A.1.6 <i>Use case</i> Skenario Manajemen Data Dokter.....	103
A.1.7 <i>Use case</i> Skenario Manajemen Data Obat	108
A.1.8 <i>Use case</i> Skenario Manajemen Data Laborat	112
A.1.9 <i>Use case</i> Skenario Manajemen Data Rujukan	115
A.1.10 <i>Use case</i> Skenario Manajemen Transaksi Pembayaran Pasien.....	118
A.1.11 <i>Use case</i> Skenario Melihat Laporan	124
A.2 <i>Activity Diagram</i>	136
A.2.1 <i>Activity Diagram</i> Manajemen Data <i>User</i>	136
A.2.2 <i>Activity Diagram</i> Manajemen Data Pegawai	137
A.2.3 <i>Activity Diagram</i> Manajemen Data Pasien.....	138
A.2.4 <i>Activity Diagram</i> Manajemen Data Kamar	139
A.2.5 <i>Activity Diagram</i> Manajemen Data Fasilitas.....	140
A.2.6 <i>Activity Diagram</i> Manajemen Data Dokter	141
A.2.7 <i>Activity Diagram</i> Manajemen Data Obat	142
A.2.8 <i>Activity Diagram</i> Manajemen Data Laborat.....	143
A.2.9 <i>Activity Diagram</i> Manajemen Data Rujukan	144
A.2.10 <i>Activity Diagram</i> Manajemen Transaksi Pembayaran	145
A.2.11 <i>Activity Diagram</i> Melihat Laporan.....	146

A.3 <i>Sequence</i> Diagram.....	147
A.3.1 <i>Sequence</i> Diagram Manajemen Data <i>User</i>	147
A.3.2 <i>Sequence</i> Diagram Manajemen Data Pegawai.....	148
A.3.3 <i>Sequence</i> Diagram Manajemen Data Pasien.....	149
A.3.4 <i>Sequence</i> Diagram Manajemen Data Kamar	150
A.3.5 <i>Sequence</i> Diagram Manajemen Data Fasilitas	151
A.3.6 <i>Sequence</i> Diagram Manajemen Data Dokter	152
A.3.7 <i>Sequence</i> Diagram Manajemen Data Obat	153
A.3.8 <i>Sequence</i> Diagram Manajemen Data Laborat.....	154
A.3.9 <i>Sequence</i> Diagram Manajemen Data Rujukan.....	155
LAMPIRAN B. KODE PROGRAM	156
B.1 Kode Program Autentifikasi (<i>Login</i>).....	156
B.2 Kode Program Manajemen Data <i>User</i>	156
B.3 Kode Program Manajemen Data Pegawai.....	158
B.4 Kode Program Manajemen Data Pasien.....	160
B.5 Kode Program Manajemen Data Kamar	163
B.6 Kode Program Manajemen Data Fasilitas.....	165
B.7 Kode Program Manajemen Data Dokter	167
B.8 Kode Program Manajemen Data Obat	169
B.9 Kode Program Manajemen Data Laborat.....	171
B.10 Kode Program Manajemen Data Rekam Medis.....	174
B.11 Kode Program Manajemen Data Rujukan.....	177
B.12 Kode Program Manajemen Data Transaksi Pembayaran Pasien	180
B.13 Kode Program Melihat Laporan.....	184
LAMPIRAN C. <i>WHITEBOX TESTING</i>	187
C.1 Manajemen Data <i>User</i>	187
C.2 Manajemen Data Pegawai	196
C. 3 Manajemen Data Pasien	206

C. 4 Manajemen Data Kamar.....	216
C. 5 Manajemen Data Fasilitas	226
C. 6 Manajemen Data Dokter	235
C. 7 Manajemen Data Obat.....	245
C. 8 Manajemen Data Laborat	254
C. 9 Manajemen Data Rujukan.....	265
C. 10 Manajemen Transaksi Pembayaran Pasien	279
C.11 Melihat Laporan	297
LAMPIRAN D. <i>BLACKBOX TESTING</i>	312
D. 1 AUTENTIFIKASI	312
D. 2 MANAJEMEN DATA <i>USER</i>	313
D. 3 MANAJEMEN DATA PEGAWAI.....	313
D. 4 MANAJEMEN DATA PASIEN	314
D. 5 MANAJEMEN DATA KAMAR.....	315
D. 6 MANAJEMEN DATA FASILITAS	316
D. 7 MANAJEMEN DATA DOKTER	317
D. 8 MANAJEMEN DATA OBAT.....	318
D. 9 MANAJEMEN DATA LABORAT.....	320
D. 10 MANAJEMEN DATA REKAM MEDIS.....	322
D. 11 MANAJEMEN DATA RUJUKAN.....	323
D. 12 MANAJEMEN TRANSAKSI PEMBAYARAN PASIEN	324
D. 13 MELIHAT LAPORAN.....	325
LAMPIRAN E. IMPLEMENTASI SISTEM	331
E. 1 HALAMAN TRANSAKSI PEMBAYARAN	331
E. 2 HALAMAN LAPORAN.....	331

BAB 1. PENDAHULUAN

Bab ini merupakan awal untuk penulisan skripsi. Pada bab ini akan dibahas tentang latar belakang, perumusan masalah, tujuan dan manfaat, ruang lingkup studi dan sistematika penulisan skripsi.

1.1 Latar Belakang

Kesehatan merupakan hal yang terpenting untuk setiap orang sehingga menjadi suatu kewajiban untuk selalu tetap dijaga. Namun pada kehidupan sehari-hari tak banyak orang yang memperhatikan bahkan sangat peduli terhadap kesehatannya sendiri. Hal tersebut menyebabkan banyak orang yang sakit dan berobat ke klinik ataupun rumah sakit terdekat dengan berbagai macam keluhan dan penyakit yang bermacam-macam. Seharusnya hal tersebut bisa dicegah dengan cara apabila merasakan kurang enak badan dapat memeriksakan diri ke klinik atau rumah sakit terdekat.

Pada saat memeriksakan diri ke klinik atau rumah sakit terdekat, setiap pasien pasti akan ditanya mengenai identitas dari pasien tersebut dan semua tentang keluhan yang dirasakan. Setelah itu, dokter akan melakukan diagnosa, tindakan medis, dan obat apa yang diberikan kepada pasien. Data mengenai identitas pasien sampai pada obat apa yang diberikan kepada pasien akan dicatat ke dalam berkas-berkas pasien yang sering disebut dengan berkas rekam medis pasien. Berkas ini dapat digunakan sebagai acuan apabila pasien tersebut berobat kembali. Salah satu hal penting yang terdapat dalam berkas rekam medis adalah kelengkapan pengisian data dan ketersediaan data saat dibutuhkan.

Khusus pada Klinik Harapan Bersama Ambulu, penyelenggaraan kesehatan juga tak lepas dari peran serta petugas dalam mencatat riwayat medis pasien atau sering disebut sebagai data rekam medis pasien. Adapun pelayanan kesehatan pada pasien yang dilakukan oleh Klinik Harapan Bersama Ambulu adalah pencatatan data pasien, pelayanan pemeriksaan, rawat inap, rawat jalan, dan transaksi pembelian obat.

Pencatatan data rekam medis oleh Klinik Harapan Bersama Ambulu masih belum optimal sebab masih dilakukan dengan cara pencatatan manual dan menggunakan pembukuan sehingga kurang efektif dan efisien dalam pengolahan datanya. Hal ini bisa mengakibatkan sulitnya mencari data pada saat dibutuhkan. Sama halnya dengan pencatatan data pasien, kegiatan administrasi Klinik Harapan Bersama Ambulu meliputi transaksi pembayaran dan beberapa transaksi lainnya masih menggunakan pencatatan manual sehingga membutuhkan waktu yang relatif lebih lama.

Dengan adanya acuan agar sistem yang dibutuhkan memiliki kriteria efektif dan efisien, perlu diadakan sebuah pengembangan sistem guna diterapkan pada Klinik Harapan Bersama. Dalam pengembangan sebuah sistem atau perangkat lunak, dibutuhkan suatu metode untuk mendefinisikan berbagai kebutuhan pada saat pengembangan perangkat lunak atau sistem tersebut. Terdapat beberapa model dari metode SDLC yang dapat dipakai untuk pengembangan perangkat lunak atau sistem diantaranya adalah model *Waterfall*, *Prototype*, dan lain-lain. Dalam hal ini, peneliti menggunakan metode SDLC model *Waterfall* untuk pengembangan sistem yang ada di Klinik Harapan Bersama. Model *Waterfall* terdiri dari tahap analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan. Model ini dipilih oleh peneliti sebab kebutuhan untuk pengembangan sistem pada Klinik Harapan Bersama telah terdefiniskan mulai pada tahap awal serta pengembangan sistem dikerjakan secara terurut berdasarkan tahap pada model *waterfall* (Bassil, 2012).

Penggunaan model *Waterfall* pernah diterapkan oleh Ginanjar Wiro Sasmito pada Sistem Informasi Geografis Industri Kabupaten Tegal. Masalah utama yang dirincikan oleh Ginanjar adalah kurangnya informasi industri yang disampaikan kepada masyarakat, menyebabkan industri-industri yang ada di Kabupaten Tegal kurang dikenal oleh masyarakat luas sehingga pangsa pasar industri Kabupaten Tegalpun tidak maksimal, disamping itu juga peluang untuk mendapatkan investor guna pengembangan usahapun menjadi terbatas. Sistem Informasi Geografis merupakan sistem komputer yang dapat merekam, menyimpan, menulis,

menganalisis dan menampilkan data geografis. Dengan menggunakan model *Waterfall* maka rancangan sistem informasi geografis dapat memberikan informasi mengenai profil industri, jenis produksi, nilai investasi, peta industri dan lokasi industri di setiap desa maupun kecamatan yang ada di Kabupaten Tegal (Sasmito, 2017).

Berdasarkan uraian singkat di atas, Klinik Harapan Bersama Ambulu membutuhkan sistem yang dapat mengolah data rekam medis pasien serta menyimpannya sehingga dapat menghasilkan informasi yang cepat, akurat, dan relevan. Diharapkan dengan sistem yang telah terkomputerisasi, pengelolaan data rekam medis pasien Klinik Harapan Bersama Ambulu dapat dilakukan secara efektif dan efisien. Untuk mengembangkan solusi tersebut, maka peneliti akan mengembangkan sistem dengan menggunakan model *waterfall* yang awalnya manual menjadi sistem yang terkomputerisasi yakni Sistem Informasi Rekam Medis.

1.2 Perumusan Masalah

Berdasarkan uraian yang telah disampaikan dalam latar belakang mendefinisikan beberapa permasalahan yang harus diselesaikan dalam penulisan ini, yaitu:

1. Bagaimana pencatatan data pasien di Klinik Harapan Bersama yang efektif, efisien, dan relevan agar proses pencatatan data pasien relatif menjadi lebih cepat?
2. Bagaimana cara membangun sistem informasi rekam medis pasien pada Klinik Harapan Bersama?

1.3 Tujuan dan Manfaat

Tujuan dan manfaat dalam penelitian ini merupakan jawaban dari perumusan masalah yang telah disebutkan.

1.3.1 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Membangun sistem informasi yang mampu menangani pencatatan data pasien Klinik Harapan Bersama agar lebih efektif dan efisien agar proses pencatatan data pasien relatif menjadi lebih cepat.
2. Membangun Sistem Informasi Rekam Medis Pasien Klinik Harapan Bersama sehingga proses pencatatan data pasien lebih mudah.

1.3.2 Manfaat

Manfaat yang ingin didapatkan dari penelitian ini adalah :

a. Bagi Akademisi

Memberikan informasi dan bahan literatur bagi dunia pendidikan, khususnya di bidang sistem informasi akademik yaitu membangun sistem informasi dengan model *waterfall*.

b. Bagi Instansi (Decision Maker)

Memberikan alternatif dalam solusi pengembangan perangkat lunak atau sistem informasi yang akan diimplementasikan.

c. Bagi Penulis

Sebagai salah satu syarat menyelesaikan program pendidikan Sistem Informasi S1 Program Studi Sistem Informasi Universitas Jember.

1.4 Ruang Lingkup

Ruang lingkup dalam penelitian ini merupakan batasan – batasan masalah dalam penelitian. Peneliti memberi batasan masalah untuk objek dan tema yang dibahas sehingga tidak terjadi penyimpangan dalam proses penelitian dan pembuatan sistem informasi. Berikut adalah batasan masalah yang dicantumkan:

1. Penelitian menggunakan model *waterfall* dalam pengembangan sistem.

2. Objek yang diteliti hanya data-data yang didapat dari hasil wawancara dan observasi yang dilakukan oleh peneliti kepada karyawan Klinik Harapan Bersama.
3. Sistem tidak untuk mengelola laporan keuangan.
4. Sistem informasi yang dibuat berbasis *web*.
5. Sistem informasi menggunakan web browser *Mozilla Firefox* dan digunakan secara *offline*.

1.5 Sistematika Penulisan

Sistematika penulisan dan keruntutan skripsi ini disusun sebagai berikut:

BAB 1. PENDAHULUAN

Bab ini menguraikan latar belakang, rumusan masalah, tujuan dan sistematika penulisan.

BAB 2. TINJAUAN PUSTAKA

Bab ini membahas mengenai tinjauan pustaka, definisi, pengertian, penjelasan teori dan metode yang berhubungan dengan permasalahan yang akan dibahas.

BAB 3. METODE PENELITIAN

Bab ini menjelaskan tentang penelitian yang akan dilakukan dan tahapan yang harus dilalui. Metode penelitian terdiri dari tiga tahapan yaitu pengumpulan data, analisis dan perancangan sistem.

BAB 4. ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi penerapan metodologi penelitian yang terdiri dari pengumpulan data, analisis dan perancangan sistem.

BAB 5. HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil dan pembahasan penelitian yang telah dilakukan.

BAB 6. PENUTUP

Bab ini berisi kesimpulan dan saran.

BAB 2. TINJAUAN PUSTAKA

Bab ini menjelaskan secara lengkap tentang profil objek penulisan, teori-teori yang ada dan informasi yang mendukung dalam penelitian.

2.1 Klinik

Berdasarkan Kamus Besar Bahasa Indonesia, klinik memiliki beberapa arti, yaitu: **1** (bagian) rumah sakit atau lembaga kesehatan tempat orang berobat dan memperoleh advis medis serta tempat mahasiswa kedokteran melakukan pengamatan thd kasus penyakit yg diderita para pasien; **2** balai pengobatan khusus: -- *keluarga berencana*; -- *penyakit paru-paru*; **3** organisasi kesehatan yg bergerak dalam penyediaan pelayanan kesehatan kuratif (diagnosis dan pengobatan), biasanya thd satu macam gangguan kesehatan. (Badan Pengembangan dan Pembinaan Bahasa, KEMDIKBUD, 2015)

Berdasarkan PERATURAN MENTERI KESEHATAN REPUBLIK INDONESIA NOMOR 9 TAHUN 2014 TENTANG KLINIK, klinik adalah fasilitas pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan yang menyediakan pelayanan medis dasar dan/atau spesialisik. Berdasarkan jenisnya, klinik dibagi menjadi klinik pratama dan klinik utama. Klinik pratama merupakan klinik yang menyelenggarakan pelayanan medik dasar baik umum maupun khusus. Sedangkan klinik utama merupakan klinik yang menyelenggarakan pelayanan medik spesialisik atau pelayanan medik dasar dan spesialisik. (PERMENKES RI, 2008)

2.2 Klinik Harapan Bersama

Klinik Harapan Bersama merupakan klinik pratama rawat iniap yang didirikan pada tahun 2013 oleh H. Ismail. Klinik ini beralamat di Jalan Kotta Blater No. 105 Langan Ambulu. Landasan hukum berdirinya klinik ini mengacu pada UU RI Nomor 36 Tahun 2009, Permenkes RI Nomor : 028/MENKES/PER/1/2011, dan Peraturan Bupati Jember Nomor 34 Tahun 2009. Klinik Harapan bersama memiliki

motto “*We Care with Cencerity*” yang artinya kami melayani dengan setulus hati.

Klinik Harapan Bersama karyawan yang terdiri:

1.1 Medis dan Paramedis

- a. Dokter Umum : 2 orang
- b. Perawatan : 12 orang
- c. Non Perawatan : 12 orang
- Total jumlah : 26 orang

2.1 Non Medis

- a. Administrasi : 6 orang
- b. Tenaga non medis lain : 4 orang
- Total jumlah : 10 orang
- Total Jumlah Karyawan : 36 orang

Klinik ini juga memiliki jenis pelayanan penunjang medis dan penunjang diagnostik yang terdiri dari: Laboratorium, Apotek, Medikal Chek-Up, dan Gizi. Selain itu, fasilitas atau sarana yang tersedia di klinik ini adalah Instalasi Rawat Darurat, Instalasi Rawat Jalan, Instalasi Rawat Inap, Instalasi Farmasi, dan Instalasi Rekam Medis.

2.3 Rekam Medis Pasien

Menurut Keputusan PERATURAN MENTERI KESEHATAN REPUBLIK INDONESIA NOMOR 269/MENKES/PER/III/2008 tentang rekam medis, Rekam Medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain yang telah diberikan kepada pasien. Rekam medis harus dibuat secara tertulis, lengkap, dan jelas atau secara elektronik. Penyelenggaraan rekam medis dengan menggunakan teknologi informasi elektronik diatur lebih lanjut dengan peraturan tersendiri (PERMENKES RI, 2008).

Definisi rekam medis menurut Edna K. Huffman, Rekam Medis adalah berkas yang menyatakan berbagai keterangan tentang pasien meliputi beberapa aspek seperti apa, siapa, kapan, mengapa, dimana, dan bagaimana pelayanan yang diperoleh

seorang pasien selama dirawat atau menjalani pengobatan (Reza, 2010). Sedangkan menurut Gemala Hatta, Rekam Medis merupakan kumpulan fakta tentang kehidupan seseorang dan data riwayat penyakitnya, termasuk yang didalamnya itu merupakan keadaan sakitnya, tentang pengobatan yang diterima saat ini dan masa lampau yang ditulis oleh praktisi kesehatan dalam upaya mereka memberikan pelayanan kesehatan kepada pasien (Reza, 2010).

Berdasarkan definisi di atas, dapat diambil kesimpulan bahwa rekam medis adalah semua berkas atau catatan yang berisi tentang apa saja mengenai data pasien yang sedang menjalani perawatan ataupun pengobatan baik itu pengobatan atau perawatan saat ini maupun masa lampau yang dicatat secara lengkap, jelas, dan jenis pencatatannya dilakukan secara manual dengan menggunakan pembukuan biasa maupun sudah terkomputerisasi guna memperoleh pelayanan kesehatan yang terbaik kepada pasien.

Penelitian tentang rakam medis pernah dilakukan oleh Endang Lestari, dkk. di Rumah Sakit Bersalin Graha RAP Tanjung Balai Karimun. Berdasarkan pengamatan seluruh kegiatan yang berhubungan dengan pasien masih dilakukan secara manual sehingga menghabiskan cukup banyak waktu untuk memproses seluruh data pasien serta meyebabkan media penyimpanan yang semakin lama semakin penuh. Ketika pasien kehilangan atau lupa membawa kartu pasien, maka petugas harus mencari kembali data pasien tersebut, jika data pasien yang dibutuhkan tidak ditemukan maka petugas akan mencatat ulang data pasien dengan nomor dokumen rekam medik yang baru. Mengingat pentingnya dokumen rekam medik, kejadian duplikasi nomor dokumen rekam medik tersebut mengakibatkan kerugian yang dialami oleh pasien, dokter maupun pihak manajemen rumah sakit. Hasil dari penelitian tersebut adalah peneliti membangun sistem rekam medis yang terkomputerisasi antar bagian-bagian maka pelayanan kepada pasien akan lebih cepat dan akurat, mempermudah dalam pencarian data pasien, dan dapat membuat laporan secara otomatis untuk beberapa pihak terkait (Lestari, 2011).

2.4 Sistem Informasi Rekam Medis Pasien

Sistem dapat didefinisikan sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen subsistem yang berinteraksi untuk mencapai tujuan tertentu. Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen, batas sistem, lingkungan luar sistem, penghubung, masukan, keluaran, pengolah, dan sasaran atau tujuan. Sedangkan informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk mengambil keputusan (Jogianto, 1989).

Sistem Informasi adalah kombinasi dari manusia, fasilitas atau alat teknologi, media, prosedur dan pengendalian yang bermaksud menata jaringan komunikasi yang penting, proses atas transaksi-transaksi tertentu secara rutin, membantu manajemen dan pemakai intern dan ekstern dan menyediakan dasar pengambilan keputusan yang tepat (John, 1995). Sedangkan pengertian dari Sistem informasi rekam medis merupakan perangkat lunak yang biasa digunakan untuk merekam riwayat kesehatan pasien dalam bentuk basis data (*database*) (Sukamto, 2012).

Berdasarkan uraian singkat diatas, maka dapat ditarik kesimpulan Sistem informasi rekam medis adalah sistem informasi yang menangani data riwayat pasien yang berupa data perawatan maupun data pengobatan pasien pada waktu saat ini ataupun masa lalu yang sudah tercatat secara terkomputerisasi dan tersistemasi sehingga dapat digunakan kembali sebagai acuan mendapat pelayanan kesehatan lebih baik bagi pasien untuk jangka waktu ke depan.

2.5 Metode SDLC Model *Waterfall*

System Development Life Cycle (SDLC) merupakan metodologi yang digunakan untuk merancang, membangun, memelihara, dan mengembangkan suatu sistem. Terdapat beberapa model dalam metode SDLC diantaranya model *Waterfall*, *Prototype*, dan lain-lain (Purwanto, 2017). Dalam penelitian ini, peneliti memilih model *waterfall* sebagai model pengembangan perangkat lunak atau sistem yang akan

dibangun.

Model *Waterfall* jika didefinisikan secara literature berarti air terjun. Namun bagi ilmu komputer dan juga teknologi informasi, *waterfall* merupakan salah satu jenis model yang digunakan untuk pengembangan sebuah perangkat lunak atau sistem. Model ini merupakan model pengembangan sistem yang setiap tahapan pengembangannya dilakukan secara berurutan. Terdapat beberapa tahap dalam model *waterfall*. Tahap-tahap tersebut adalah analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan (Nuryani, 2015). Tahapan model *Waterfall* dapat dilihat pada gambar 2.1.



Gambar 2. 1 Tahapan model *Waterfall*

Sumber: (Bassil, 2012)

Diawali dengan tahap analisis kebutuhan yaitu proses menganalisa data masukan penelitian yang telah dikumpulkan seperti proses bisnis dan permasalahan yang terjadi. Analisa ini ditujukan untuk mengidentifikasi masalah, menentukan kebutuhan sistem dan lainnya sehingga dapat merumuskan solusi atas permasalahan yang terjadi. Setelah proses analisis kebutuhan sistem ini selesai tahap selanjutnya adalah desain sistem. Desain sistem merupakan proses perencanaan dan pemecahan masalah untuk solusi perangkat lunak. Ini melibatkan pengembang perangkat lunak dan perancang untuk menentukan rencana untuk solusi yang mencakup perancangan algoritma, perancangan arsitektur perangkat lunak, skema konseptual basis data dan

desain diagram logis, desain konsep, desain antarmuka pengguna grafis, dan definisi struktur data (Bassil, 2012).

Tahap ketiga dari model *Waterfall* adalah implementasi yaitu tahap dimana kode sebenarnya ditulis dan disusun menjadi aplikasi operasional, dan dimana database dan file teks dibuat. Tahap ini mengacu pada realisasi persyaratan bisnis dan spesifikasi desain menjadi komponen program, database, situs web, atau perangkat lunak yang dapat dieksekusi melalui pemrograman dan penerapan. Dengan kata lain, proses pengubahan keseluruhan persyaratan dan cetak biru menjadi produk jadi (Bassil, 2012).

Tahap berikutnya adalah pengujian. Pengujian erat kaitannya dengan verifikasi dan validasi yang merupakan proses untuk memeriksa apakah perangkat lunak memenuhi persyaratan dan spesifikasi asli dan menyelesaikannya sesuai tujuannya. Pada dasarnya, verifikasi adalah proses evaluasi perangkat lunak untuk menentukan apakah produk dari tahap pengembangan tertentu memenuhi persyaratan yang diberlakukan pada awal fase tersebut. Sedangkan validasi adalah proses evaluasi perangkat lunak selama atau pada akhir proses pengembangan untuk menentukan apakah memenuhi persyaratan yang ditentukan. Selain itu, tahap pengujian adalah hasil keluaran untuk melakukan debugging sesuai dengan bug dan gangguan sistem yang ditemukan, lalu diperbaiki, dan setelah itu disempurnakan sesuai kebutuhan user (Bassil, 2012).

Tahap terakhir pada model *waterfall* adalah pemeliharaan. Apabila terdapat kesalahan yang terjadi pada sistem atau perangkat lunak, maka dilakukan proses pemeliharaan guna memperbaiki kesalahan. Perbaikan implementasi unit sistem akan dianggap sebagai kebutuhan baru. Kelebihan dan kekurangan metode *Waterfall* yaitu sifatnya yang kaku terhadap perubahan. Dokumen pengembangan sistem terorganisir, karena setiap tahapan harus terselesaikan dengan lengkap sebelum melangkah ke tahap berikutnya. Jadi setiap tahapan akan mempunyai dokumen tertentu (Nuryani, 2015).

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode-metode yang digunakan untuk mencapai tujuan penelitian. Penggunaan metode membuat alur dan arah penelitian menjadi terarah. Metode yang digunakan dalam Penelitian ini adalah metode Pengembangan sistem yaitu model *waterfall* yang merujuk pada gambar 2.1 yang memiliki beberapa tahapan yang akan dijadikan subbab dalam bab ini.

3.1 Analisis Kebutuhan

Analisis kebutuhan yang diperlukan untuk menyelesaikan tujuan penelitian ini didapat dari tahapan pengumpulan data yang akan dijelaskan pada subbab berikutnya. Setelah tahap pengumpulan data selesai dilakukan, maka akan didapat gambaran sistem yang dapat digunakan sebagai acuan dalam pembuatan sistem. Tempat penelitian dilakukan di Klinik Harapan Bersama Ambulu yang beralamat di Jalan Kotta Blater No. 105 Langon Ambulu dan waktu penelitian dilakukan selama 9 (sembilan) bulan, dimulai pada bulan Januari 2017 sampai dengan bulan September 2017.

3.1.1 Pengumpulan Data

Pengumpulan data dilakukan untuk memenuhi kebutuhan data yang diperlukan untuk menyelesaikan tujuan penelitian. Pengumpulan data tersebut dapat diperoleh dengan beberapa cara sebagai berikut:

a. Studi Literatur

Studi literatur digunakan untuk mendukung teori-teori yang berkaitan dengan masalah yang akan dibahas. Dengan studi literatur tersebut, peneliti mendapat referensi untuk menyelesaikan tujuan penelitian dalam pengumpulan data yang dibutuhkan untuk membangun sistem. Referensi yang dapat menjadi pendukung untuk pembuatan sistem didapat dari buku, *web pages*, dan jurnal *online*. Tujuan dari studi literatur adalah

untuk memperkuat landasan teori. Literatur yang pertama dicari oleh peneliti adalah literatur mengenai rekam medis sebagai acuan untuk membangun sistem informasi rekam medis. Literatur yang kedua adalah mengenai metode SDLC model *waterfall* sebagai acuan untuk pengembangan sistem informasi rekam medis. Literatur lainnya adalah literatur pendukung untuk memperkuat landasan teori.

b. Wawancara

Kegiatan ini dilakukan dengan mengumpulkan data kualitatif. Data kualitatif diperoleh dengan mengajukan pertanyaan langsung kepada narasumber objek penelitian. Penggunaan metode kualitatif dalam penulisan ini adalah dengan mencocokkan antara realita dengan teori yang berlaku dengan menggunakan metode deskriptif. Data yang dikumpulkan untuk penulisan ini berupa hasil wawancara pada pegawai Klinik Harapan Bersama Ambulu.

c. Pengamatan Terhadap Sistem yang Masih Berjalan

Kegiatan ini dilakukan dengan cara mengamati sistem yang masih berjalan pada objek penelitian. Pengamatan ini tujuan untuk memberikan gambaran ataupun acuan dalam pengembangan sistem yang akan dibangun.

3.1.2 Gambaran Umum Sistem

Sistem informasi rekam medis bertujuan mengelola pencatatan data pasien agar lebih efektif dan efisien sehingga data rekam medis pasien tidak ada yang menumpuk atau rangkap. Tujuan diadakan pencatatan data pasien yaitu memperlancar pengelolaan data pasien dan juga pengelolaan data rekam medis pasien, menyajikan informasi yang penting dalam bentuk laporan tertulis serta penyimpanan semua dokumen. Sistem ini mampu melakukan penyimpanan data berupa data pasien, kamar, dokter, pegawai, fasilitas, rekam medis dan rujukan. Sistem ini juga mampu mengelola transaksi pembayaran perawatan pasien.

3.2 Desain Sistem

Pembuatan desain sistem menggunakan *Unified Modeling Language* (UML) yang dirancang menggunakan konsep *Object-Oriented Programming* (OOP). Berikut Pemodelan UML yang digunakan antara lain:

3.2.1 *Business Process*

Digunakan untuk menggambarkan masukan data, keluaran dari sistem, dan tujuan dari pembuatan sistem. Dirancang sesuai dengan analisis kebutuhan sistem.

3.2.2 *Use Case Diagram*

Digunakan untuk mendeskripsikan hak akses antara aktor dengan sistem. Dirancang sesuai dengan hasil gambaran *business process* yang telah dibuat.

3.2.3 *Use Case Scenario*

Digunakan untuk menjabarkan alur kerja tiap step-step *use case* yang telah ditentukan.

3.2.4 *Activity Diagram*

Digunakan untuk menggambarkan *scenario* atau aktifitas dari sistem untuk dapat mengetahui alur yang dilakukan aktor serta respon sistem sesuai yang ditentukan.

3.2.5 *Sequence Diagram*

Digunakan untuk menggambarkan interaksi-interaksi antar *object* di dalam sistem. Dirancang sesuai dengan *activity diagram* sistem yang telah dibuat agar dapat mengetahui *method* yang berjalan ketika terjadi aksi.

3.2.6 *Class Diagram*

Digunakan untuk menggambarkan struktur dari segi pendefinisian kelas-kelas yang ada pada sistem sesuai dengan *sequence* yang telah dibuat.

3.2.7 *Entity Relationship Diagram* (ERD)

Digunakan untuk menggambarkan *database* sistem yang dibangun agar dapat mengetahui tabel *database* yang diperlukan sistem sesuai yang telah

ditentukan.

3.3 Implementasi

Tahap Implementasi merupakan tahap penerjemahan desain sistem dalam bahasa yang dapat dikenali oleh komputer. Desain sistem yang telah dimodelkan dengan notasi UML tersebut harus diubah menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Sistem ini dibangun dengan menggunakan bahasa pemrograman yang menerapkan konsep *Object Oriented Programming* (OOP) yaitu *Page Hyper Text Pre-Process* (PHP) dengan menggunakan *framework Codeigniter* (CI). CI adalah *framework* PHP yang menggunakan pola arsitektur *Model View Controller*(MVC). *Database* yang digunakan adalah MySQL dan *tool* XAMPP.

3.4 Pengujian

Tahap ini adalah tahap untuk menguji kode program yang telah dibuat. Tujuan pengujian adalah menilai apakah aplikasi yang dirancang telah sesuai dengan apa yang diharapkan, serta untuk mengevaluasi keunggulan dan kelemahan terhadap kualitas sistem. Teknik untuk pengujian sistem menggunakan *white box* dan *black box*. Pengujian *whitebox* menggunakan *Cyclomatic Complexity* yang dilakukan oleh penulis sendiri tanpa melibatkan *user*, sedangkan pengujian *blackbox* akan melibatkan *user*.

3.5 Pemeliharaan

Pemeliharaan dilakukan untuk mengatasi masalah pada sistem di lain waktu ketika sistem sudah dapat digunakan oleh *user*.

BAB 4. ANALISIS DAN PENGEMBANGAN SISTEM

Bab ini membahas tentang analisis dan pengembangan untuk mengembangkan Sistem Informasi Rekam Medis Klinik Harapan Bersama Ambulu. Tahapan-tahapan pengembangan dilaksanakan berdasarkan metode SDLC model *waterfall*.

4.1 Analisis Kebutuhan

Kebutuhan sistem adalah kemampuan yang harus dimiliki oleh sistem untuk memenuhi apa yang diinginkan oleh pengguna. Kebutuhan yang berkaitan dengan fungsi atau proses yang dikerjakan sistem dan kebutuhan yang menetapkan karakteristik yang dimiliki oleh sistem dapat dijelaskan pada kebutuhan fungsional sedangkan kebutuhan yang tidak terkait dengan fungsi yang dikerjakan sistem dijelaskan pada kebutuhan non fungsional.

4.1.1 Pengumpulan Data

Pengumpulan data dilakukan agar sistem yang dikembangkan sesuai dengan kebutuhan pengguna. Pengumpulan data dilakukan dengan berbagai cara, di antaranya adalah wawancara ke karyawan objek penelitian (Klinik Harapan Bersama) dan studi literatur. Data-data yang dibutuhkan untuk sistem informasi rekam medis pasien klinik arapan bersama yang akan dikembangkan adalah data pasien, data pegawai, data kamar, data fasilitas, data dokter, dan data obat di klinik Harapan Bersama Ambulu.

a. Data Pasien

Data pasien Klinik Harapan Bersama Ambulu dapat dengan cara wawancara ke pihak administrasi pencatatan data pasien Klinik Harapan Bersama. Dalam hal ini peneliti hanya diberikan sampel atau beberapa data riwayat pasien tanpa riwayat rekam medisnya.

b. Data Pegawai

Data pegawai Klinik Harapan Bersama Ambulu di dapat dengan cara wawancara ke pihak karyawan Klinik Harapan Bersama. Dalam hal ini peneliti diberikan detail karyawan yang bekerja di Klinik Harapan Bersama. Namun terkait data karyawan, peneliti hanya diberikan beberapa data saja yang terkait dengan data pegawai Klinik Harapan Bersama Ambulu.

c. Data Kamar

Data Kamar Klinik Harapan Bersama Ambulu di dapat dengan cara wawancara ke pihak Administrasi Klinik Harapan Bersama. Dalam hal ini peneliti diberikan detail kamar, jumlah kamar, kelas kamar, dan beberapa fasilitas yang ada didalamnya.

d. Data Fasilitas

Data Fasilitas Klinik Harapan Bersama Ambulu di dapat dengan cara wawancara ke pihak Administrasi Klinik Harapan Bersama. Dalam hal ini peneliti diberikan detail dari beberapa fasilitas yang ada di Klinik Harapan Bersama.

e. Data Dokter

Data Dokter Klinik Harapan Bersama Ambulu di dapat dengan cara wawancara ke pihak Administrasi Klinik Harapan Bersama. Dalam hal ini peneliti diberikan informasi mengenai data dari beberapa dokter yang praktek di Klinik Harapan Bersama.

f. Data Obat

Data Obat Klinik Harapan Bersama Ambulu di dapat dengan cara wawancara ke pihak Administrasi Klinik Harapan Bersama. Dalam hal ini peneliti diberikan sampel data dari seluruh data obat yang berada di Klinik Harapan Bersama.

4.1.2 Kebutuhan Fungsional

Kebutuhan fungsional dari Sistem Informasi Rekam Medis Pasien Klinik Harapan Bersama adalah sebagai berikut:

1. Sistem dapat memproses data *user*, pasien, pegawai, dokter, kamar, fasilitas, obat, laboratorium, rekam medis pasien, dan rujukan baik melihat, menambahkan maupun mengubah data baru sesuai dengan *role* yang dimiliki oleh *user* pada saat *Login*.
2. Sistem dapat mengelola sekaligus mencetak data transaksi pembayaran berupa: transaksi pembayaran perawatan pasien, transaksi pembayaran laboratorium, dan transaksi pembayaran obat dalam bentuk halaman pdf.
3. Sistem dapat melihat dan mencetak beberapa laporan seperti: laporan pasien, laporan rekam medis, laporan fasilitas, laporan kamar, laporan dokter, laporan pegawai, laporan laboratorium, laporan obat, laporan rujukan.

4.1.3 Kebutuhan Non Fungsional

Sedangkan kebutuhan nonfungsional dari Sistem Informasi Rekam Medis Pasien Klinik Harapan Bersama adalah sebagai berikut:

1. Sistem dapat diakses 24 jam sehari.
2. Tampilan sistem mudah dimengerti pengguna agar pengguna tidak kesulitan dalam mengoperasikannya (*user friendly*).
3. Sistem memberi *respon time* yang relatif cepat.
4. Keandalan(*Realibility*) sistem berjalan sesuai dengan kebutuhan *user*.
5. Ketepatan(*Correctness*) adalah sejauh mana sistem memenuhi spesifikasi dan tujuan dari yang dibutuhkan oleh pengguna.

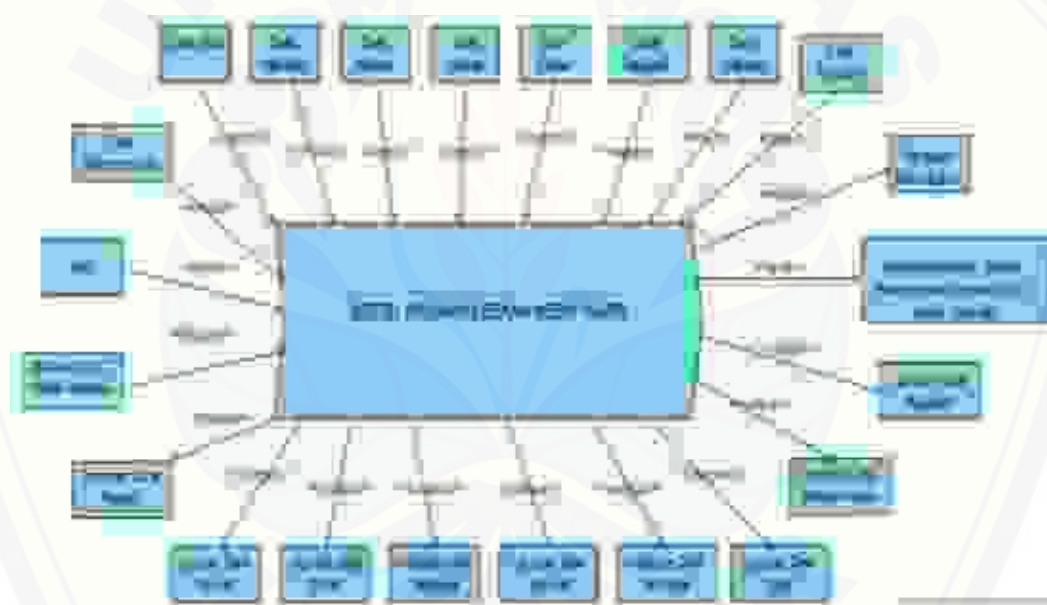
4.2 Desain Sistem

Desain sistem adalah tahapan untuk memodelkan sistem informasi rekam medis pasien klinik harapan bersama ambulu. Dokumen desain sistem yang akan dibuat meliputi *Bussines Process*, *Use case Diagram*, *Use case Skenario*, *Activity*

Diagram, *Sequence Diagram*, *Class Diagram* dan *Entity Relationship Diagrams* (ERD).

4.3.1 *Bussines Process*

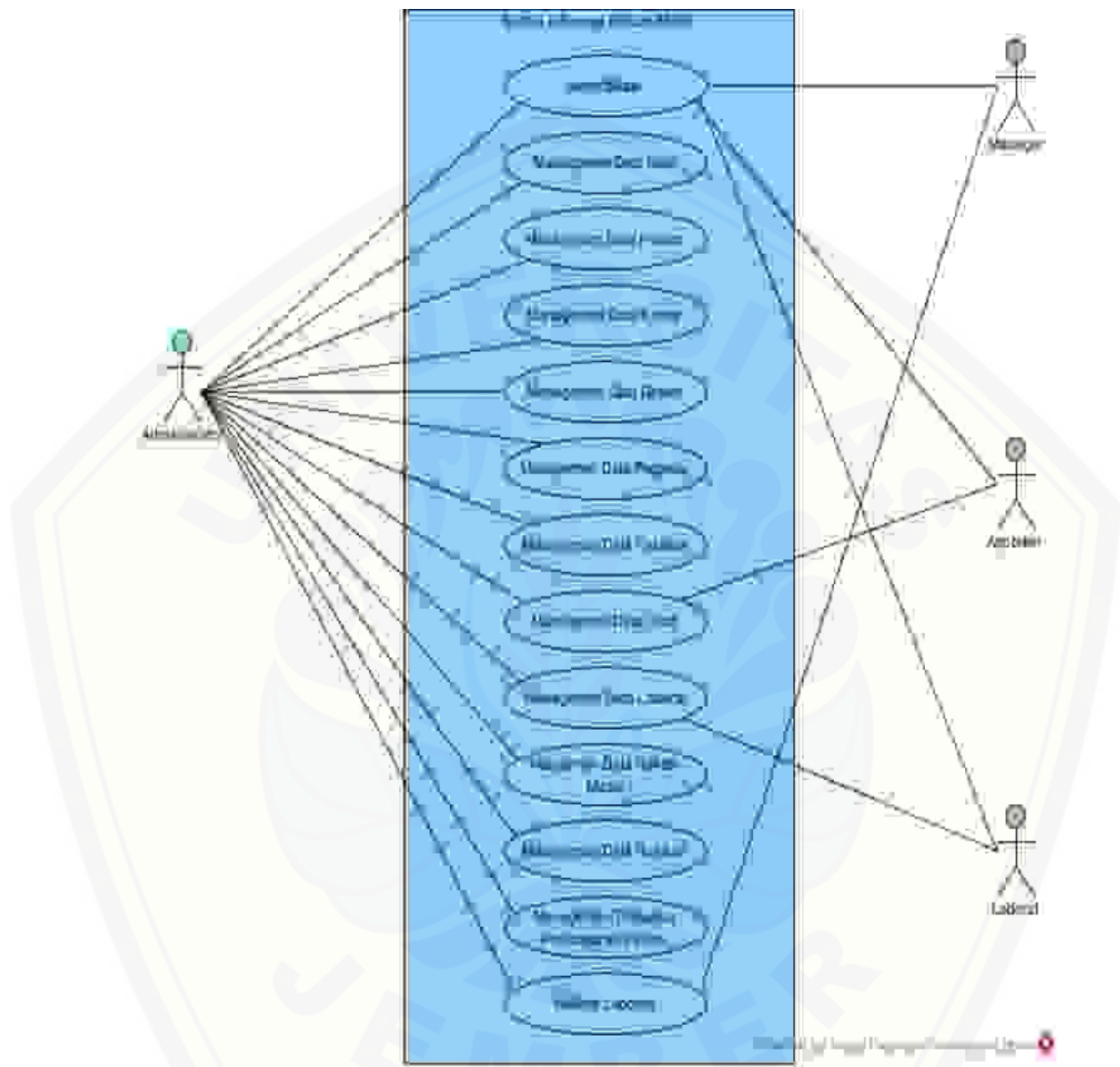
Bussines Process merupakan sekumpulan proses yang dilakukan untuk mencapai hasil yang diinginkan. Terdapat beberapa komponen untuk mencapai tujuan tersebut, yaitu data yang menjadi masukan (*input*) dan media yang digunakan sehingga akan dihasilkan data keluaran (*output*) beserta tujuan (*goal*) yang ingin dicapai. *Bussines Process* dari Sistem Informasi Rekam Medis Pasien Klinik Harapan Bersama Ambulu dapat dilihat pada gambar 4.1.



Gambar 4. 1 *Bussines Process* Sistem Informasi Rekam Medis Pasien
Sumber: (Hasil Analisis)

4.3.2 *Use case Diagram*

Use case Diagram adalah dokumentasi untuk menggambarkan fitur dan aktor yang terdapat pada sistem yang dibuat. *Use case diagram* Sistem Informasi Rekam Medis seperti yang dijelaskan pada gambar 4.2



Gambar 4. 2 Use case Diagram Sistem Informasi Rekam Medis
 Sumber: (Hasil Analisis)

a. Definisi aktor

Definisi aktor merupakan penjelasan tentang aktor-aktor sebagai pengguna dari sistem informasi rekam medis pasien klinik harapan bersama yang akan dibangun.

Terdapat 4 (empat) aktor dari hasil analisis seperti yang dijelaskan pada tabel 4.1.

Tabel 4. 1 Definisi Aktor
Sumber: (Hasil Analisis)

No.	Aktor	Deskripsi
1.	Administrator	Administrator merupakan aktor yang memiliki hak akses untuk mengelola (menambah, membaca, memperbarui dan menghapus) semua data di Sistem Informasi Rekam Medis Pasien.
2.	Manager	Manager merupakan aktor yang hanya memiliki hak akses untuk membaca Laporan pada Sistem Informasi Rekam Medis Pasien.
3.	Apoteker	Apoteker merupakan aktor yang hanya memiliki hak akses untuk mengelola (menambah, membaca, memperbarui dan menghapus) data Obat.
4.	Laborat	Laborat merupakan aktor yang hanya memiliki hak akses untuk mengelola (menambah, membaca, memperbarui dan menghapus) data Laboratorium.

b. Definisi *Use case*

Definisi *Use case* merupakan penjelasan dari masing – masing *Use case* atau fitur-fitur pada Sistem Informasi Rekam Medis Pasien. Penjelasan *use case* dapat di lihat pada Tabel 4.2.

Tabel 4. 2 Definisi *Use case*
Sumber: (Hasil Analisis)

ID	Use Case	Deskripsi
UC-01	Autentifikasi	Menggambarkan proses awal aktor mulai masuk atau <i>Login</i> ke dalam Sistem Informasi Rekam Medis Pasien. Autentifikasi dilakukan oleh semua aktor.
UC-02	Manajemen Data <i>User</i>	Menggambarkan proses mengelola (menambah, membaca, memperbarui dan menghapus) data <i>user</i> oleh administrator.
UC-03	Manajemen Data Pasien	Menggambarkan proses mengelola (menambah, membaca, memperbarui dan menghapus) data pasien oleh administrator.

UC-04	Manajemen Data Kamar		Menggambarkan proses mengelola (menambah, membaca, memperbarui dan menghapus) data kamar oleh administrator.
UC-05	Manajemen Data Dokter		Menggambarkan proses mengelola (menambah, membaca, memperbarui dan menghapus) data dokter oleh administrator.
UC-06	Manajemen Pegawai	Data	Menggambarkan proses mengelola (menambah, membaca, memperbarui dan menghapus) data pegawai oleh administrator.
UC-07	Manajemen Fasilitas	Data	Menggambarkan proses mengelola (menambah, membaca, memperbarui dan menghapus) data fasilitas oleh administrator.
UC-08	Manajemen Data Obat		Menggambarkan proses mengelola (menambah, membaca, memperbarui dan menghapus) data obat oleh administrator dan apoteker.
UC-09	Manajemen Laborat	Data	Menggambarkan proses mengelola (menambah, membaca, mencetak dan menghapus) data laborat oleh administrator dan laborat.
UC-10	Manajemen Data Rekam Medis		Menggambarkan proses mengelola (menambah, membaca, memperbarui, mencetak, dan menghapus) data rekam medis oleh administrator.
UC-11	Manajemen Rujukan	Data	Menggambarkan proses mengelola (menambah, membaca, memperbarui, mencetak, dan menghapus) data rujukan oleh administrator.
UC-12	Manajemen Transaksi Pembayaran Pasien	Data	Menggambarkan proses pembuatan data transaksi pembayaran berupa transaksi pembayaran perawatan, laboratorium, dan obat oleh administrator.
UC-13	Melihat Laporan		Menggambarkan proses untuk membaca laporan oleh administrator dan manager.

4.3.3 Use case Skenario

Use case skenario adalah dokumentasi terhadap kebutuhan fungsional sistem. Skenario terdiri dari nama usecase, aktor, pre condition, post condition, skenario normal dan skenario alternatif. Rincian penjelasan Use case skenario sistem informasi rekam medis pasien klinik harapan bersama adalah sebagai berikut.

a. Use case Skenario Autentifikasi (Login)

Use case skenario Autentifikasi (Login) menggambarkan proses awal aktor mulai masuk atau Login ke dalam Sistem Informasi Rekam Medis Pasien. Autentifikasi dapat dilakukan oleh semua aktor. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif Use case skenario Autentifikasi dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Use case Skenario Autentifikasi
Sumber: (Hasil Analisis)

ID Use Case	UC-01
Nama	Autentifikasi
Aktor	Administrator, Apoteker, Laborat, dan Manager.
Pre-Condition	Administrator, Apoteker, Laborat, dan Manager harus mempunyai <i>username</i> dan <i>password</i> dengan role masing-masing untuk masuk ke sistem
Post-Condition	Administrator, Apoteker, Laborat, dan Manager berhasil masuk ke sistem.
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
Aktor mulai menjalankan sistem dan melakukan Login ke sistem	
1. Menjalankan sistem	2. Menampilkan Form <i>Login</i> yang terdiri dari: - <i>Textfield username</i> - <i>Textfield password</i> - Menu pilihan <i>dropdown role</i>
3. Memasukkan <i>Username</i> , <i>Password</i> dan pilih <i>Role</i>	
4. Menekan tombol " <i>Login</i> "	

	5. Melakukan autentifikasi ke dalam <i>database</i>
	6. Menampilkan fitur sesuai <i>role</i> masing-masing aktor
SKENARIO ALTERNATIF	
Aktor tidak memasukkan <i>username</i>, <i>password</i> dan <i>role</i> atau <i>username</i>, <i>password</i>, dan <i>role</i> salah	
3a. <i>Username</i> (kosong/ salah), <i>Password</i> (kosong/ salah), dan pilih <i>role</i> (salah)	
4a. Menekan tombol “ <i>Login</i> ”	
	5a. Mengecek masukan <i>Username</i> , <i>password</i> , dan <i>role</i>
	6a. Menampilkan <i>alert</i> , “ <i>Wrong Username and Password</i> ”
	7a. Menampilkan Form <i>Login</i> kembali yang terdiri dari: <ul style="list-style-type: none"> - <i>Textfield username</i> - <i>Textfield password</i> - Menu pilihan <i>dropdown role</i> .

b. *Use case* Skenario Manajemen Data *User*

Use case Skenario Manajemen Data *User* menggambarkan proses mengelola data *user* yang hanya bisa dilakukan oleh administrator. Administrator berhak menambah, membaca, memperbarui, dan menghapus data *user* pada sistem. Skenario manajemen data *user* terdiri dari skenario normal dan skenario alternatif. Skenario normal menjabarkan tentang kegiatan administrator yang berkaitan tentang tambah, *edit*, *remove*, dan melihat atau membaca data *user* aktif yang dapat melakukan *login* ke sistem. Sedangkan skenario alternatif menjabarkan tentang pembatalan yang dilakukan oleh administrator pada saat mengelola data *user* aktif misal pembatalan saat salah klik hapus dan sebagainya. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *Use case* skenario manajemen data *user* dapat dilihat pada Lampiran A.

c. *Use case* Skenario Manajemen Data Pegawai

Use case Skenario Manajemen Data Pegawai menggambarkan proses mengelola data pegawai yang hanya bisa dilakukan oleh administrator. Administrator berhak menambah, membaca, memperbarui, dan menghapus data *user* pada sistem. Skenario manajemen data pegawai terdiri dari skenario normal dan skenario alternatif. Skenario normal menjabarkan tentang kegiatan administrator yang berkaitan tentang tambah, *edit*, *remove*, dan melihat atau membaca data pegawai yang ada di sistem. Sedangkan skenario alternatif menjabarkan tentang pembatalan yang dilakukan oleh administrator pada saat mengelola data pegawai misal pembatalan saat klik *edit* data pegawai dan sebagainya. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *Use case* skenario Manajemen Data Pegawai dapat dilihat pada Lampiran A.

d. *Use case* Skenario Manajemen Data Pasien

Use case Skenario Manajemen Data Pasien menggambarkan proses mengelola data pasien yang hanya bisa dilakukan oleh administrator. Administrator berhak menambah, membaca, memperbarui, dan menghapus data pasien pada sistem. Skenario manajemen data pasien terdiri dari skenario normal dan skenario alternatif. Skenario normal menjabarkan tentang kegiatan administrator yang berkaitan tentang tambah, *edit*, *remove*, dan melihat atau membaca data pasien yang ada di sistem. Sedangkan skenario alternatif menjabarkan tentang pembatalan yang dilakukan oleh administrator pada saat mengelola data pasien misal pembatalan saat klik *edit* data pasien dan sebagainya. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *Use case* skenario manajemen data pasien dapat dilihat pada Lampiran A.

e. *Use case* Skenario Manajemen Data Kamar

Use case Skenario Manajemen Data Kamar menggambarkan proses

mengelola data kamar yang hanya bisa dilakukan oleh administrator. Administrator berhak menambah, membaca, memperbarui, dan menghapus data kamar pada sistem. Skenario manajemen data kamar terdiri dari skenario normal dan skenario alternatif. Skenario normal menjabarkan tentang kegiatan administrator yang berkaitan tentang tambah, *edit*, *remove*, dan melihat atau membaca data kamar yang ada di sistem. Sedangkan skenario alternatif menjabarkan tentang pembatalan yang dilakukan oleh administrator pada saat mengelola data kamar misal pembatalan saat klik *edit* data pasien dan sebagainya. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *Use case* skenario manajemen data kamar dapat dilihat pada Lampiran A.

f. *Use case* Skenario Manajemen Data Fasilitas

Use case Skenario Manajemen Data Fasilitas menggambarkan proses mengelola data fasilitas yang hanya bisa dilakukan oleh administrator. Administrator berhak menambah, membaca, memperbarui, dan menghapus data fasilitas pada sistem. Skenario manajemen data fasilitas terdiri dari skenario normal dan skenario alternatif. Skenario normal menjabarkan tentang kegiatan administrator yang berkaitan tentang tambah, *edit*, *remove*, dan melihat atau membaca data fasilitas yang ada di sistem. Sedangkan skenario alternatif menjabarkan tentang pembatalan yang dilakukan oleh administrator pada saat mengelola data fasilitas misal pembatalan saat klik *edit* data fasilitas dan sebagainya. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *Use case* skenario Manajemen Data Fasilitas dapat dilihat pada Lampiran A.

g. *Use case* Skenario Manajemen Data Dokter

Use case Skenario Manajemen Data Dokter menggambarkan proses mengelola data dokter yang hanya bisa dilakukan oleh administrator. Administrator berhak menambah, membaca, memperbarui, dan menghapus data dokter pada sistem. Skenario manajemen data dokter terdiri dari skenario normal dan skenario alternatif.

Skenario normal menjabarkan tentang kegiatan administrator yang berkaitan tentang tambah, *edit*, *remove*, dan melihat atau membaca data dokter yang ada di sistem. Sedangkan skenario alternatif menjabarkan tentang pembatalan yang dilakukan oleh administrator pada saat mengelola data dokter misal pembatalan saat klik *edit* data dokter dan sebagainya. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *Use case* skenario Manajemen Data Dokter dapat dilihat pada Lampiran A.

h. *Use case* Skenario Manajemen Data Obat

Use case Skenario Manajemen Data Obat menggambarkan proses mengelola data obat yang bisa dilakukan oleh dua aktor yaitu administrator dan apoteker. Administrator dan apoteker berhak menambah, membaca, memperbarui, dan menghapus data obat pada sistem. Skenario manajemen data obat terdiri dari skenario normal dan skenario alternatif. Skenario normal menjabarkan tentang kegiatan administrator dan apoteker yang berkaitan tentang tambah, *edit*, *remove*, dan melihat atau membaca data obat yang ada di sistem. Sedangkan skenario alternatif menjabarkan tentang pembatalan yang dilakukan oleh aktor pada saat mengelola data obat misal pembatalan saat klik *edit* data obat dan sebagainya. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *Use case* skenario Manajemen Data Obat dapat dilihat pada Lampiran A.

i. *Use Case* Skenario Manajemen Data Laborat

Use case Skenario Manajemen Data Laborat menggambarkan proses mengelola data Laborat yang bisa dilakukan oleh dua aktor yaitu administrator dan Laborat. Aktor berhak menambah, membaca, mencetak, dan menghapus data obat pada sistem. Skenario manajemen data laborat terdiri dari skenario normal dan skenario alternatif. Skenario normal menjabarkan tentang kegiatan aktor yang berkaitan tentang tambah, cetak, *remove*, dan melihat atau membaca data laborat

yang ada di sistem. Sedangkan skenario alternatif menjabarkan tentang pembatalan yang dilakukan oleh aktor pada saat mengelola data laborat misal pembatalan saat klik tambah data laborat dan sebagainya. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *Use case* skenario Manajemen Data Laborat dapat dilihat pada Lampiran A.

j. *Use Case* Skenario Manajemen Data Rekam Medis

Use case Skenario Manajemen Data Rekam Medis menggambarkan proses mengelola data rekam medis yang bisa dilakukan oleh administrator. Administrator berhak menambah, membaca, memperbarui, mencetak, dan menghapus data rekam medis pada sistem. Skenario manajemen data rekam medis terdiri dari skenario normal dan skenario alternatif. Skenario normal menjabarkan tentang kegiatan aktor yang berkaitan tentang tambah, *edit*, *remove*, cetak, dan melihat atau membaca data rekam medis yang ada di sistem. Sedangkan skenario alternatif menjabarkan tentang pembatalan yang dilakukan oleh aktor pada saat mengelola data rekam medis misal pembatalan saat klik tambah data laborat dan sebagainya. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *Use case* skenario manajemen data rekam medis dapat dilihat pada Tabel 4.4.

Tabel 4. 4 Use Case Skenario Manajemen Data Rekam Medis
Sumber: (Hasil Analisis)

ID Use Case	UC-10
Nama	Manajemen Data Rekam Medis
Aktor	Administrator
Pre-Condition	Menampilkan Data Rekam Medis
Post-Condition	Melihat daftar informasi dan merubah isi informasi data rekam medis
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Klik menu Rekam Medis	2. Menampilkan halaman Rekam Medis yang berisi: - tombol Tambah Rekam Medis - <i>textfield Search</i>

-
- menu *dropdown Show entries* tabel, dan
 - tabel Data Rekam Medis yang berisi tentang ID Rekam Medis, Pasien, Dokter, Tanggal Masuk, Tanggal Pemeriksaan, Jenis Pemeriksaan, Pemeriksaan, dan *Action*. Kolom Pasien berisi tentang ID Pasien dan Nama Pasien. Kolom Pemeriksaan berisi tentang Anamnesis, Diagnosa, Hasil Pemeriksaan, dan Obat yang Dianjurkan. Sedangkan Kolom *Action* berisi tombol *Edit*, tombol *Remove*, dan tombol Cetak

SKENARIO NORMAL

Aktor menambah data rekam medis

-
3. Jika aktor ingin menambah data rekam medis, maka klik tombol Tambah Rekam Medis
-
4. Sistem menampilkan halaman Tambah Rekam Medis yang berisi:
 - *Textfield* Pasien,
 - *Textfield* Anamnesis,
 - *Textfield* Diagnosa,
 - *Textfield* Hasil Pemeriksaan,
 - Menu Pilihan *dropdown* Dokter yang Memeriksa,
 - *Textfield* Jenis Pemeriksaan,
 - *Textfield* Tanggal Masuk,
 - *Textfield* Tanggal Pemeriksaan,
 - *Textfield* Obat yang Dianjurkan.
-
5. Memasukkan data rekam medis
-
6. Klik tombol *Submit*
-
7. Melakukan validasi data
-
8. Apabila data yang diisi sudah valid, sistem menampilkan data rekam medis yang telah ditambah kedalam tabel Data Rekam Medis dan menampilkan kembali halaman Rekam Medis
-
9. Sistem menyimpan data di *database*
-

SKENARIO ALTERNATIF
Aktor menambah data rekam medis tetapi ingin mengosongkan semua *textfield*

6a. Ketika memasukkan data rekam medis pada halaman Tambah Rekam Medis, tetapi ingin mengosongkan semua *textfield*, klik tombol *Reset* untuk mengosongkan data yang telah diisi

7a. Sistem mengosongkan data yang telah dimasukkan

SKENARIO ALTERNATIF
Aktor menambah data rekam medis tetapi ingin membatalkannya

6b. Ketika memasukkan data rekam medis pada halaman Tambah Rekam Medis, tetapi ingin membatalkannya, klik tombol *Back*

7b. Sistem menampilkan kembali halaman Rekam Medis

SKENARIO ALTERNATIF
Aktor menambah data rekam medis tetapi *input* data tidak valid

6c. Ketika tombol *Submit* diklik pada halaman Tambah Rekam Medis, tetapi *input* data tidak valid

7c. Sistem menampilkan *message* “*Please fill out this field*”

SKENARIO NORMAL
Aktor mengubah data rekam medis

10. Jika aktor ingin mengubah data rekam medis, klik tombol *Edit* pada kolom *Action* di tabel Data Rekam Medis

11. Sistem menampilkan halaman *Edit* Rekam Medis yang berisi data rekam medis berupa:

- *Textfield* Pasien,
- *Textfield* Anamnesis,
- *Textfield* Diagnosa,
- *Textfield* Hasil Pemeriksaan,
- Menu Pilihan *dropdown* Dokter yang Memeriksa,
- *Textfield* Jenis Pemeriksaan,

	<ul style="list-style-type: none"> - <i>Textfield</i> Tanggal Masuk, - <i>Textfield</i> Tanggal Pemeriksaan, - <i>Textfield</i> Obat yang Dianjurkan.
12. Melakukan <i>edit</i> data	yang telah terisi berdasarkan ID RM.
13. Klik tombol <i>Submit</i>	
	14. Melakukan validasi data
	15. Apabila data yang diisi sudah valid, sistem menampilkan data rekam medis yang telah <i>diupdate</i> dan menampilkan kembali halaman Rekam Medis
	16. Sistem melakukan <i>update</i> data di <i>database</i>
SKENARIO ALTERNATIF	
Aktor mengubah data rekam medis tetapi aktor ingin membatalkannya	
13a. Ketika aktor ingin membatalkan proses <i>edit</i> , klik tombol <i>Cancel</i>	
	14a. Sistem menampilkan kembali halaman Rekam Medis
SKENARIO ALTERNATIF	
Aktor mengubah data rekam medis tetapi <i>input</i> data tidak valid	
13b. Ketika tombol <i>Submit</i> diklik pada halaman <i>Edit</i> Rekam Medis, tetapi <i>input</i> data tidak valid.	
	14b. Sistem menampilkan <i>message</i> “ <i>Please fill out this field</i> ”
SKENARIO NORMAL	
Aktor menghapus data rekam medis	
17. Jika aktor ingin menghapus data rekam medis, klik tombol <i>Remove</i> pada kolom <i>Action</i> di tabel Data Rekam Medis berdasarkan ID Rekam Medis yang akan dihapus	
	18. Sistem menampilkan <i>message dialog</i> “ <i>Are You Sure You Want to Remove This Data?</i> ”
19. Klik tombol OK	
	20. Sistem menghapus data yang dipilih kemudian menampilkan halaman Rekam Medis

SKENARIO ALTERNATIF
Aktor menghapus data rekam medis tetapi aktor ingin membatalkan proses hapus data setelah mengklik tombol *Remove*

19. Apabila aktor ingin membatalkan proses hapus data setelah mengklik tombol *Remove* pada kolom *Action* di tabel Data Rekam Medis, klik tombol *Cancel* pada *message dialog* konfirmasi penghapusan data

20. Sistem menampilkan kembali halaman Rekam Medis

SKENARIO NORMAL
Aktor mencetak data rekam medis

21. Jika aktor ingin mencetak data rekam medis, klik tombol Cetak pada kolom *Action* di tabel Data Rekam Medis berdasarkan ID Rekam Medis yang akan dicetak

22. Sistem mengubah data rekam medis yang akan dicetak dalam bentuk Portable Document Format (PDF) dan menampilkan data tersebut di tab baru

SKENARIO NORMAL
Aktor mencari data rekam medis

23. Jika aktor ingin mencari data rekam medis yang ada di tabel Data Rekam Medis, ketikkan data yang akan di cari pada *textfield Search*

24. Sistem mencari data rekam medis sesuai dengan data yang diketikkan pada *textfield Search*

25. Sistem menampilkan data tersebut pada tabel Data Rekam Medis

SKENARIO ALTERNATIF
Aktor mencari data rekam medis tetapi data yang dicari tidak ditemukan

23. Jika aktor ingin mencari data rekam medis yang ada di tabel Data Rekam Medis, tetapi data yang dicari tidak ditemukan

24. Sistem menampilkan *message* “No matching records found”

SKENARIO NORMAL

Aktor ingin menampilkan isi tabel Data Rekam Medis dengan jumlah tertentu

26. Jika aktor ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah *entries* data pada menu *dropdown Show entries*

27. Sistem menampilkan isi tabel berdasarkan jumlah *entries* yang dipilih

k. *User case* Skenario Manajemen Data Rujukan

Use case Skenario Manajemen Data rujukan menggambarkan proses mengelola data rekam medis yang bisa dilakukan oleh administrator. Administrator berhak menambah, membaca, memperbarui, mencetak, dan menghapus data rujukan pada sistem. Skenario manajemen data rujukan terdiri dari skenario normal dan skenario alternatif. Skenario normal menjabarkan tentang kegiatan aktor yang berkaitan tentang tambah, *edit*, *remove*, cetak, dan melihat atau membaca data rujukan yang ada di sistem. Sedangkan skenario alternatif menjabarkan tentang pembatalan yang dilakukan oleh aktor pada saat mengelola data rujukan misal pembatalan saat klik tambah data laborat dan sebagainya. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *Use case* skenario Manajemen Data Rujukan dapat dilihat pada Lampiran A.

l. *Use Case* Skenario Manajemen Transaksi Pembayaran Pasien

Manajemen Transaksi Pembayaran Pasien terdiri dari tiga(3) bagian transaksi yaitu Transaksi Pembayaran Perawatan, Transaksi Pembayaran Laboratorium, dan Transaksi Pembayaran Obat. Untuk *Use Case* skenario manajemen transaksi pembayaran pasien, aktor yang dapat mengaksesnya adalah administrator. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *Use case* skenario Manajemen Transaksi Pembayaran Pasien dapat dilihat pada Lampiran A.

m. *Use Case* Skenario Melihat Laporan

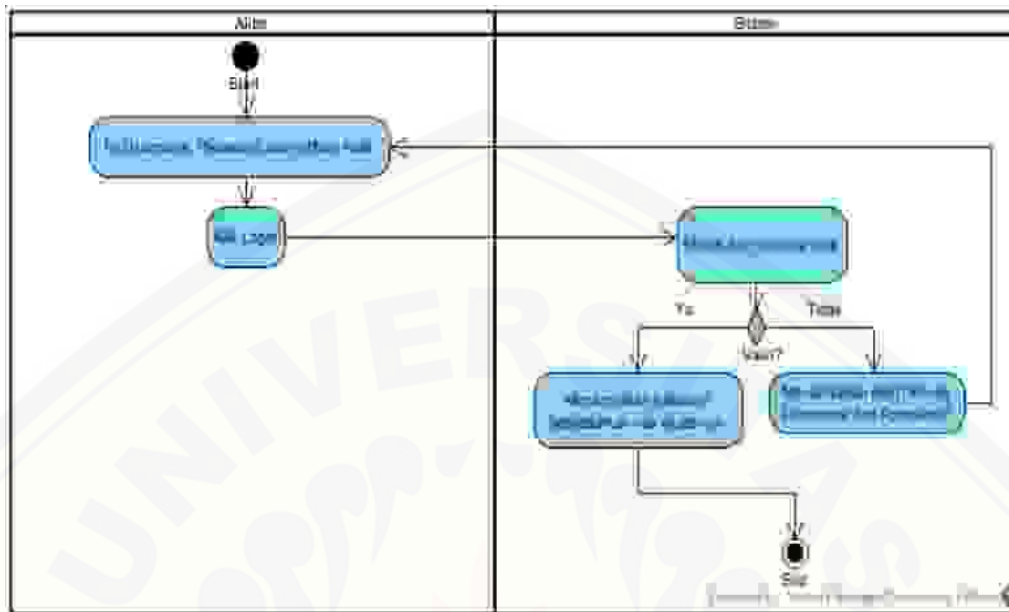
Fitur Melihat Laporan merupakan fitur yang dapat diakses oleh dua (2) aktor yaitu administrator dan manajer. Pada fitur ini, kedua aktor tersebut hanya dapat melihat dan mencetak Laporan. Terdapat 9 Laporan yang ada pada Fitur Manajemen Laporan. Laporan-laporan tersebut adalah Laporan Pasien, Kamar, Dokter, Pegawai, Fasilitas, Obat, Laboratorium, Rekam Medis, dan Rujukan. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *Use Case* Skenario Melihat Laporan dapat dilihat pada Lampiran A.

4.3.4 *Activity* Diagram

Activity Diagram menggambarkan aliran aktivitas dalam Sistem Informasi Rekam Medis Pasien yang akan dibangun. Sistem Informasi Rekam Medis Pasien ini memiliki tiga belas *Activity* Diagram, yaitu sebagai berikut:

a. *Activity* Diagram Autentifikasi (*Login*)

Aliran aktifitas pada *Activity* Diagram Autentifikasi(*login*) diawali dengan aktor mengisi *Username*, *Password*, dan memilih *role* yang sesuai berdasarkan hak akses yang dimiliki oleh setiap aktor. Setelah itu, aktor klik submit untuk masuk ke dalam sistem. Jika semua yang telah diisi tadi sesuai, maka aktor bisa masuk ke dalam sistem. Akan tetapi bila tidak sesuai, maka sistem akan menampilkan alert sesuai kesalahan pada saat mengisi *Username*, *Password*, dan memilih *role* tadi. Aliran aktifitas *Activity* Diagram Autentifikasi(*login*) dalam fitur Autentifikasi (*Login*) ditunjukkan pada gambar 4.3.



Gambar 4. 3 Activity Diagram Login
 Sumber: (Hasil Analisis)

b. Activity Diagram Manajemen Data User

Activity Diagram Manajemen Data User menggambarkan aliran aktifitas saat aktor menjalankan fitur Manajemen Data User. Aliran aktifitasnya bermula dari aktor setelah berhasil masuk ke sistem, maka sistem menampilkan halaman Dashboard yang berisi data user aktif atau bisa juga aktor mengklik menu Dashboard. Pada menu Dashboard, aktor dapat melakukan lihat user aktif, tambah, edit, dan hapus user aktif. Aliran aktifitas Activity Diagram Manajemen Data User dalam fitur Manajemen Data User dapat dilihat pada Lampiran A.

c. Activity Diagram Manajemen Data Pegawai

Activity Diagram Manajemen Data Pegawai merupakan gambaran aktifitas pada saat aktor berada pada fitur Manajemen Data Pegawai. Aliran aktifitasnya bermula saat aktor mengklik menu Pegawai pada sistem informasi rekam medis pasien. Setelah sistem menampilkan halaman Pegawai, aktor dapat melakukan

tambah, *edit*, *remove* data pegawai. Aktor juga dapat melakukan *searching* data pegawai dan juga menampilkan jumlah *entri* tabel data pegawai sesuai pilihan yang ada. Aliran aktifitas atau *Activity Diagram* Manajemen Data Pegawai dalam fitur Manajemen Data Pegawai dapat dilihat pada Lampiran A.

d. *Activity Diagram* Manajemen Data Pasien

Activity Diagram Manajemen Data Pasien merupakan gambaran aktifitas pada saat aktor berada pada fitur Manajemen Data Pasien. Aliran aktifitasnya bermula saat aktor mengklik menu Pasien pada sistem informasi rekam medis pasien. Setelah sistem menampilkan halaman Pasien, aktor dapat melakukan tambah, *edit*, *remove* data pasien. Aktor juga dapat melakukan *searching* data pasien dan juga menampilkan jumlah *entri* tabel data pasien sesuai pilihan yang ada. Aliran aktifitas *Activity Diagram* Manajemen Data Pegawai dalam fitur Manajemen Data Pasien dapat dilihat pada Lampiran A.

e. *Activity Diagram* Manajemen Data Kamar

Activity Diagram Manajemen Data Kamar merupakan gambaran aktifitas pada saat aktor berada pada fitur Manajemen Data Kamar. Aliran aktifitasnya bermula saat aktor mengklik menu Kamar pada sistem informasi rekam medis pasien. Setelah sistem menampilkan halaman Kamar, aktor dapat melakukan tambah, *edit*, *remove* data Kamar. Aktor juga dapat melakukan *searching* data kamar dan juga menampilkan jumlah *entri* tabel data kamar sesuai pilihan yang ada. Aliran aktifitas *Activity Diagram* Manajemen Data Kamar dalam fitur Manajemen Data Kamar dapat dilihat pada Lampiran A.

f. *Activity Diagram* Manajemen Data Fasilitas

Activity Diagram Manajemen Data Fasilitas merupakan gambaran aktifitas pada saat aktor berada pada fitur Manajemen Data Fasilitas. Aliran aktifitasnya

bermula saat aktor mengklik menu Fasilitas pada sistem informasi rekam medis pasien. Setelah sistem menampilkan halaman Fasilitas, aktor dapat melakukan tambah, *edit*, *remove* data Fasilitas. Aktor juga dapat melakukan *searching* data Fasilitas dan juga menampilkan jumlah *entri* tabel data Fasilitas sesuai pilihan yang ada. Aliran aktifitas *Activity Diagram* Manajemen Data Fasilitas dalam fitur Manajemen Data Fasilitas dapat dilihat pada Lampiran A.

g. *Activity Diagram* Manajemen Data Dokter

Activity Diagram Manajemen Data Dokter merupakan gambaran aktifitas pada saat aktor berada pada fitur Manajemen Data Dokter. Aliran aktifitasnya bermula saat aktor mengklik menu Dokter pada sistem informasi rekam medis pasien. Setelah sistem menampilkan halaman Dokter, aktor dapat melakukan tambah, *edit*, *remove* data Dokter. Aktor juga dapat melakukan *searching* data Dokter dan juga menampilkan jumlah *entri* tabel data Dokter sesuai pilihan yang ada. Aliran aktifitas *Activity Diagram* Manajemen Data Dokter dalam fitur Manajemen Data Dokter dapat dilihat pada Lampiran A.

h. *Activity Diagram* Manajemen Data Obat

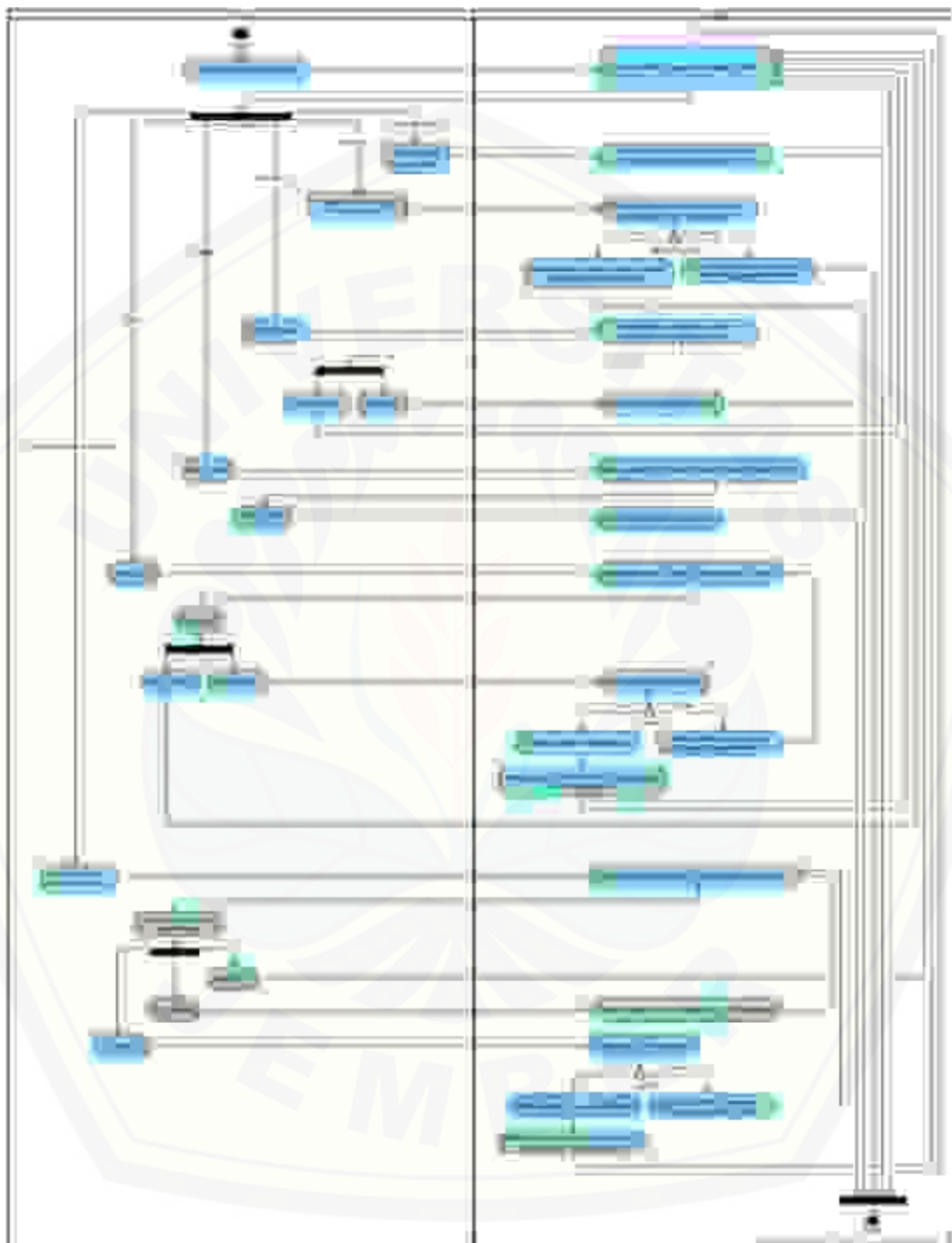
Activity Diagram Manajemen Data Obat merupakan gambaran aktifitas pada saat aktor berada pada fitur Manajemen Data Obat. Aliran aktifitasnya bermula saat aktor mengklik menu Obat pada sistem informasi rekam medis pasien. Aktor yang berperan dalam aliran aktifitas ini adalah administrator dan apoteker. Setelah sistem menampilkan halaman Obat, aktor dapat melakukan tambah, *edit*, *remove* data Obat. Aktor juga dapat melakukan *searching* data Obat dan juga menampilkan jumlah *entri* tabel data Obat sesuai pilihan yang ada. Aliran aktifitas *Activity Diagram* Manajemen Data Obat dalam fitur Manajemen Data Obat dapat dilihat pada Lampiran A.

i. *Activity Diagram* Manajemen Data Laborat

Activity Diagram Manajemen Data Laborat merupakan gambaran aktifitas pada saat aktor berada pada fitur Manajemen Data Laborat. Aliran aktifitasnya bermula saat aktor mengklik menu Laborat pada sistem informasi rekam medis pasien. Aktor yang berperan dalam aliran aktifitas ini adalah administrator dan laborat. Setelah sistem menampilkan halaman Laborat, aktor dapat melakukan tambah, *remove*, dan cetak data Laborat. Aktor juga dapat melakukan *searching* data Laborat dan juga menampilkan jumlah *entri* tabel data Laborat sesuai pilihan yang ada. Aliran aktifitas *Activity Diagram* Manajemen Data Laborat dalam fitur Manajemen Data Laborat dapat dilihat pada Lampiran A.

j. *Activity Diagram* Manajemen Data Rekam Medis

Activity Diagram Manajemen Data Rekam Medis merupakan gambaran aktifitas pada saat aktor berada pada fitur Manajemen Data Rekam Medis. Aliran aktifitasnya bermula saat aktor mengklik menu Rekam Medis pada sistem informasi rekam medis pasien. Setelah sistem menampilkan halaman Rekam Medis, aktor dapat melakukan tambah, *edit*, *remove*, dan cetak data Rekam Medis. Aktor juga dapat melakukan *searching* data Rekam Medis dan juga menampilkan jumlah *entri* tabel data Rekam Medis sesuai pilihan yang ada. Aliran aktifitas *Activity Diagram* Manajemen Data Rekam Medis dalam fitur Manajemen Data Rekam Medis ditunjukkan pada gambar 4.4.



Gambar 4. 4 *Activity* Diagram Manajemen Data Rekam Medis
Sumber: (Hasil analisis)

k. *Activity Diagram* Manajemen Data Rujukan

Activity Diagram Manajemen Data Rujukan merupakan gambaran aktifitas pada saat aktor berada pada fitur Manajemen Data Rujukan. Aliran aktifitasnya bermula saat aktor mengklik menu Rujukan pada sistem informasi rekam medis pasien. Setelah sistem menampilkan halaman Rujukan, aktor dapat melakukan tambah, *edit*, *remove*, dan cetak data Rujukan. Aktor juga dapat melakukan *searching* data Rujukan dan juga menampilkan jumlah *entri* tabel data Rujukan sesuai pilihan yang ada. Aliran aktifitas *Activity Diagram* Manajemen Data Rujukan dalam fitur Manajemen Data Rujukan dapat dilihat pada Lampiran A.

l. *Activity Diagram* Manajemen Transaksi Pembayaran Pasien

Activity Diagram Manajemen Data Transaksi Pembayaran Pasien merupakan gambaran aktifitas pada saat aktor berada pada fitur Manajemen Data Transaksi Pembayaran Pasien. Aliran aktifitasnya bermula saat aktor mengklik menu Transaksi Pembayaran pada sistem informasi rekam medis pasien. Setelah sistem menampilkan halaman Transaksi Pembayaran dan terdapat tiga(3) tombol, Transaksi Pembayaran Perawatan, Transaksi Pembayaran Laboratorium, dan Transaksi Pembayaran Obat. Aliran aktifitas *Activity Diagram* Manajemen Data Transaksi Pembayaran Pasien dalam fitur Manajemen Transaksi Pembayaran Pasien dapat dilihat pada Lampiran A.

m. *Activity Diagram* Melihat Laporan

Activity Diagram Melihat Laporan merupakan gambaran aktifitas pada saat aktor berada pada fitur Melihat Laporan. Aktor yang berperan pada aktifitas ini adalah administrator dan manajer serta kedua aktor ini hanya dapat melihat dan mencetak laporan saja. Aliran aktifitasnya bermula saat aktor mengklik menu Laporan pada sistem informasi rekam medis pasien. Penjelasan aliran aktifitas *Activity Diagram* Melihat Laporan dalam fitur Melihat Laporan dijelaskan pada Lampiran A.

4.3.5 Sequence Diagram

Sequence diagram adalah dokumentasi suatu diagram terurut yang menampilkan interaksi - interaksi antar objek di dalam sistem. *Sequence* diagram digunakan untuk menggambarkan skenario dan memodelkan aliran logika dalam sistem dengan cara *visual*. *Sequence* diagram dari Sistem Informasi Rekam Medis Pasien adalah sebagai berikut:

a. *Sequence* Diagram Autentifikasi (*Login*)

Pada *Sequence* Diagram Autentifikasi (*Login*) terdapat gambaran alur kegiatan yang menggambarkan interaksi pada antara *class* dan *method* yang digunakan yang berada pada fitur Autentifikasi (*Login*). Interaksi tersebut meliputi *view* (*login* dan *index*), *controller* (*welcome*, *laborat*, *admin*, *apoteker*, dan *manager*) dan *model* (*user_model*). *Sequence* Diagram Autentifikasi (*Login*) dalam fitur Autentifikasi (*Login*) ditunjukkan pada gambar 4.5.



Gambar 4. 5 *Sequence* Diagram Autentifikasi (*Login*)
Sumber: (Hasil analisis)

b. *Sequence* Diagram Manajemen Data *User*

Pada *Sequence* Diagram Manajemen Data *User* terdapat gambaran alur kegiatan yang menggambarkan interaksi pada antara *class* dan *method* yang digunakan yang berada pada fitur Manajemen Data *User*. Interaksi tersebut meliputi *view* (index, add-admin, dan edit-admin), *controller* (admin) dan *model* (user_model). *Sequence* Diagram Manajemen Data *User* dalam fitur Manajemen Data *User* dapat dilihat pada Lampiran A.

c. *Sequence* Diagram Manajemen Data Pegawai

Pada *Sequence* Diagram Manajemen Data Pegawai terdapat gambaran alur kegiatan yang menggambarkan interaksi pada antara *class* dan *method* yang digunakan yang berada pada fitur Manajemen Data Pegawai. Interaksi tersebut meliputi *view* (index, pegawai-data, add-pegawai, dan edit-pegawai), *controller* (pegawai) dan *model* (pegawai_model). *Sequence* Diagram Manajemen Data Pegawai dalam fitur Manajemen Data Pegawai dapat dilihat pada gambar Lampiran A.

d. *Sequence* Diagram Manajemen Data Pasien

Pada *Sequence* Diagram Manajemen Data Pasien terdapat gambaran alur kegiatan yang menggambarkan interaksi pada antara *class* dan *method* yang digunakan yang berada pada fitur Manajemen Data Pasien. Interaksi tersebut meliputi *view* (index, pasien-data, add-pasien, dan edit-pasien), *controller* (pasien) dan *model* (pasien_model). *Sequence* Diagram Manajemen Data Pasien dalam fitur Manajemen Data Pasien dapat dilihat pada Lampiran A.

e. *Sequence* Diagram Manajemen Data Kamar

Pada *Sequence* Diagram Manajemen Data Kamar terdapat gambaran alur kegiatan yang menggambarkan interaksi pada antara *class* dan *method* yang digunakan yang berada pada fitur Manajemen Data Kamar. Interaksi tersebut meliputi *view* (index, kamar-data, add-kamar, dan edit-kamar), *controller* (kamar)

dan *model* (*kamar_model*). *Sequence* Diagram Manajemen Data Kamar dalam fitur Manajemen Data Kamar dapat dilihat pada Lampiran A.

f. *Sequence* Diagram Manajemen Data Fasilitas

Pada *Sequence* Diagram Manajemen Data Fasilitas terdapat gambaran alur kegiatan yang menggambarkan interaksi pada antara *class* dan *method* yang digunakan yang berada pada fitur Manajemen Data Fasilitas. Interaksi tersebut meliputi *view* (*index*, *fasilitas-data*, *add-fasilitas*, dan *edit-fasilitas*), *controller* (*fasilitas*) dan *model* (*fasilitas_model*). *Sequence* Diagram Manajemen Data Fasilitas dalam fitur Manajemen Data Fasilitas dapat dilihat pada Lampiran A.

g. *Sequence* Diagram Manajemen Data Dokter

Pada *Sequence* Diagram Manajemen Data Dokter terdapat gambaran alur kegiatan yang menggambarkan interaksi pada antara *class* dan *method* yang digunakan yang berada pada fitur Manajemen Data Dokter. Interaksi tersebut meliputi *view* (*index*, *dokter-data*, *add-dokter*, dan *edit-dokter*), *controller* (*dokter*) dan *model* (*dokter_model*). *Sequence* Diagram Manajemen Data Dokter dalam fitur Manajemen Data Dokter dapat dilihat Lampiran A.

h. *Sequence* Diagram Manajemen Data Obat

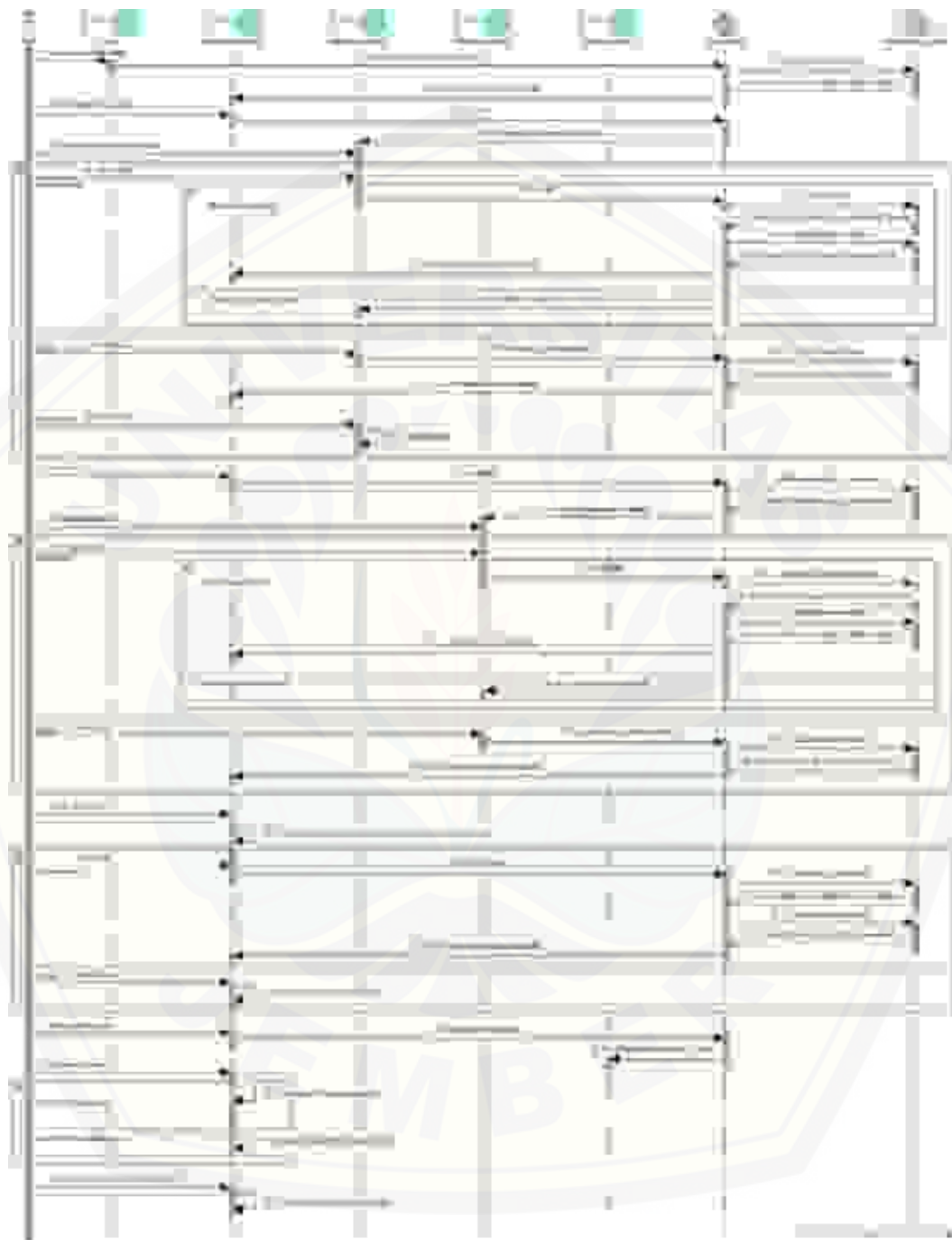
Pada *Sequence* Diagram Manajemen Data Obat terdapat gambaran alur kegiatan yang menggambarkan interaksi pada antara *class* dan *method* yang digunakan yang berada pada fitur Manajemen Data Obat. Interaksi tersebut meliputi *view* (*index*, *obat-data*, *add-obat*, dan *edit-obat*), *controller* (*obat*) dan *model* (*obat_model*). *Sequence* Diagram Manajemen Data Obat dalam fitur Manajemen Data Obat dapat dilihat Lampiran A.

i. *Sequence* Diagram Manajemen Data Laborat

Pada *Sequence* Diagram Manajemen Data Laborat terdapat gambaran alur kegiatan yang menggambarkan interaksi pada antara *class* dan *method* yang digunakan yang berada pada fitur Manajemen Data Laborat. Interaksi tersebut meliputi *view* (index, obat-laborat, add-laborat, dan edit-laborat), *controller* (laborat) dan *model* (laborat_model). *Sequence* Diagram Manajemen Data Laborat dalam fitur Manajemen Data Laborat dapat dilihat Lampiran A.

j. *Sequence* Diagram Manajemen Data Rekam Medis

Pada *Sequence* Diagram Manajemen Data Rekam Medis terdapat gambaran alur kegiatan yang menggambarkan interaksi pada antara *class* dan *method* yang digunakan yang berada pada fitur Manajemen Data Rekam Medis. Interaksi tersebut meliputi *view* (index, obat-rekammedis, add-rekammedis, dan edit-rekammedis), *controller* (rekammedis) dan *model* (rekammedis_model). *Sequence* Diagram Manajemen Data Rekam Medis dalam fitur Manajemen Data Rekam Medis ditunjukkan pada gambar 4.6.



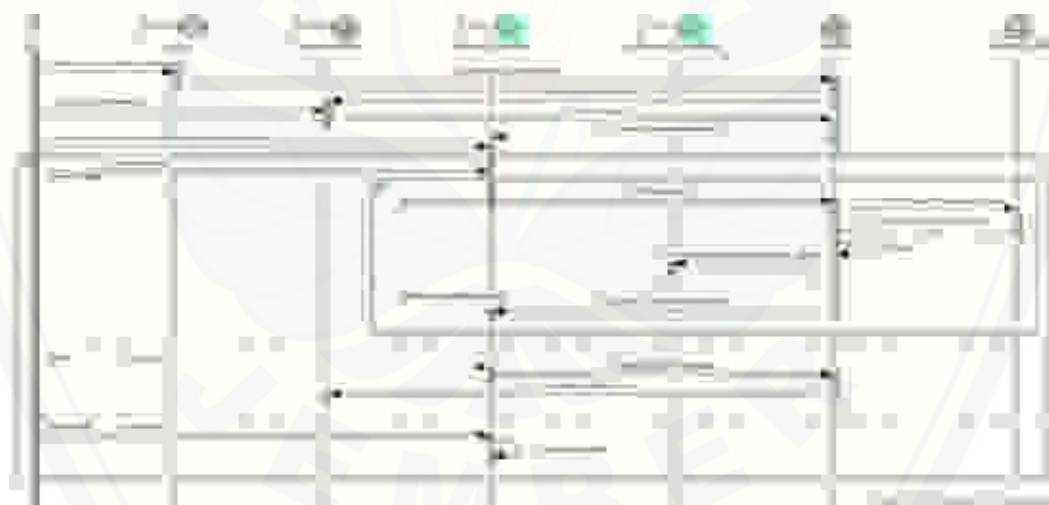
Gambar 4. 6 *Sequence* Diagram Manajemen Data Rekam Medis
Sumber: (Hasil analisis)

k. *Sequence* Diagram Manajemen Data Rujukan

Pada *Sequence* Diagram Manajemen Data Rujukan terdapat gambaran alur kegiatan yang menggambarkan interaksi pada antara *class* dan *method* yang digunakan yang berada pada fitur Manajemen Data Rujukan. Interaksi tersebut meliputi *view* (index, obat-rujukan, add-rujukan, dan edit-rujukan), *controller* (rujukan) dan *model* (rujukan_model). *Sequence* Diagram Manajemen Data Rujukan dalam fitur Manajemen Data Rujukan dapat dilihat Lampiran A.

l. *Sequence* Diagram Manajemen Transaksi Pembayaran Pasien

Fitur Manajemen Transaksi Pembayaran Pasien terbagi menjadi tiga transaksi pembayaran yaitu Transaksi Pembayaran Perawatan, Transaksi Pembayaran Laboratorium, dan Transaksi Pembayaran Obat. Penjelasan aliran aktifitas dalam Ketiga Transaksi Pembayaran pasien tersebut dijelaskan pada gambar 4.7, gambar 4.8, dan gambar 4.9.



Gambar 4. 7 *Sequence* Diagram Transaksi Pembayaran Perawatan
Sumber: (Hasil analisis)



Gambar 4. 8 *Sequence* Diagram Transaksi Pembayaran Laboratorium
Sumber: (Hasil analisis)



Gambar 4. 9 *Sequence* Diagram Transaksi Pembayaran Obat
Sumber: (Hasil analisis)

m. *Sequence* Diagram Melihat Laporan

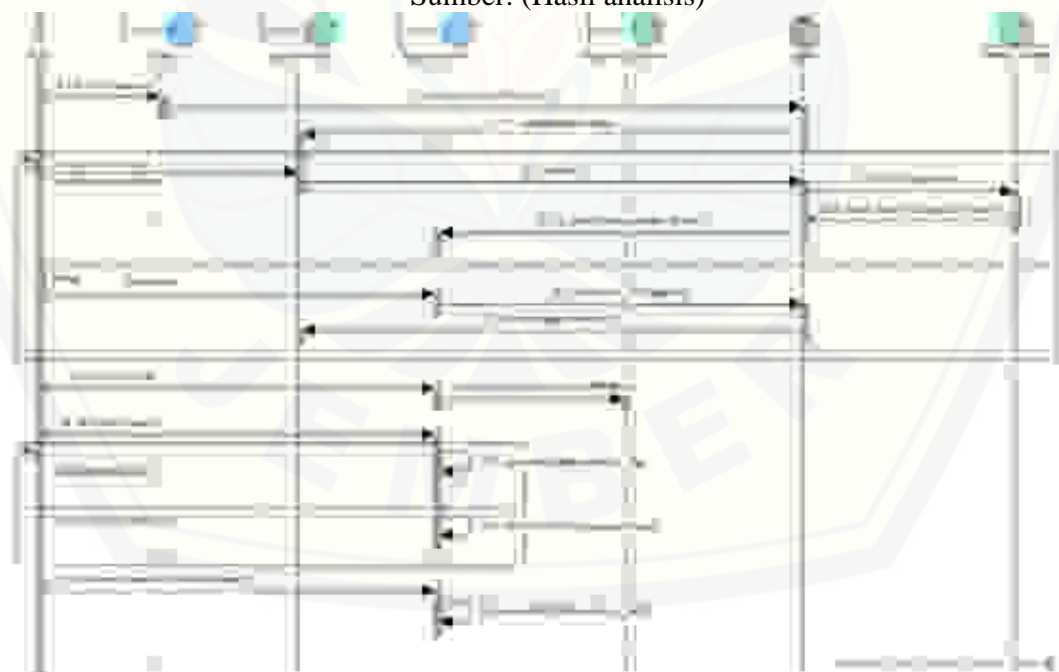
Fitur Melihat Laporan terbagi menjadi sembilan bagian yakni Laporan Pasien, Laporan Kamar, Laporan Dokter, Laporan Pegawai, Laporan Fasilitas, Laporan Obat, Laporan Laboratorium, Laporan Rekam Medis, dan Laporan Rujukan. Penjelasan aliran aktifitas dari sembilan Laporan tersebut berturut-turut dijelaskan pada gambar 4.10, gambar 4.11, gambar 4.12, gambar 4.13, gambar 4.14, gambar 4.15, gambar 4.16, gambar 4.17, dan gambar 4.18 .



Gambar 4. 10 *Sequence* Diagram Laporan Pasien
Sumber: (Hasil analisis)



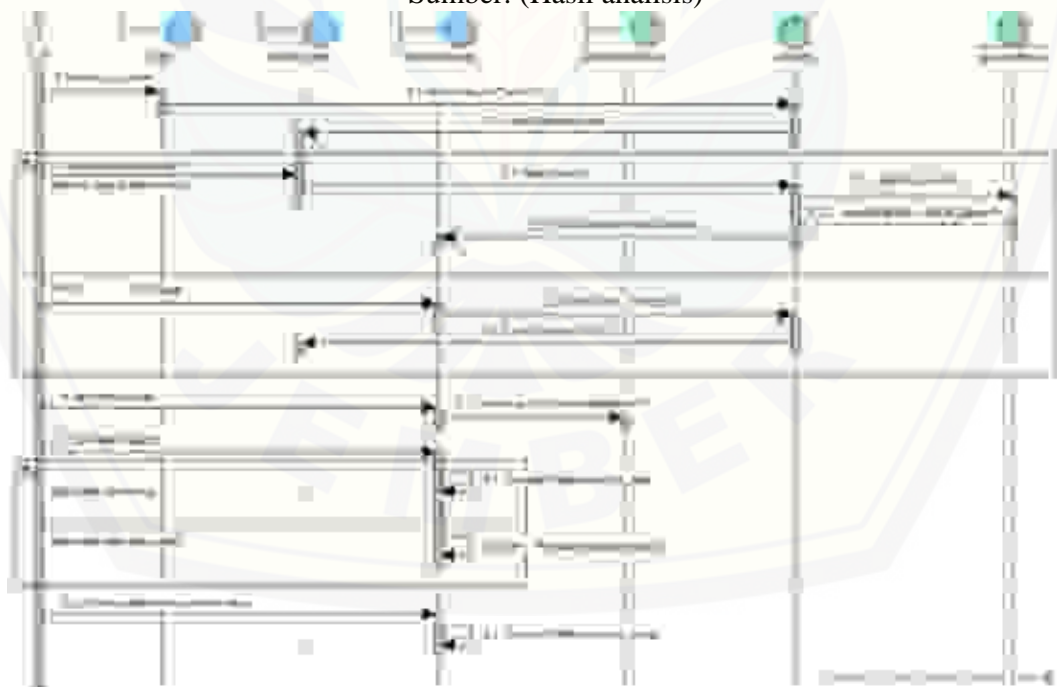
Gambar 4. 11 *Sequence* Diagram Laporan Kamar
Sumber: (Hasil analisis)



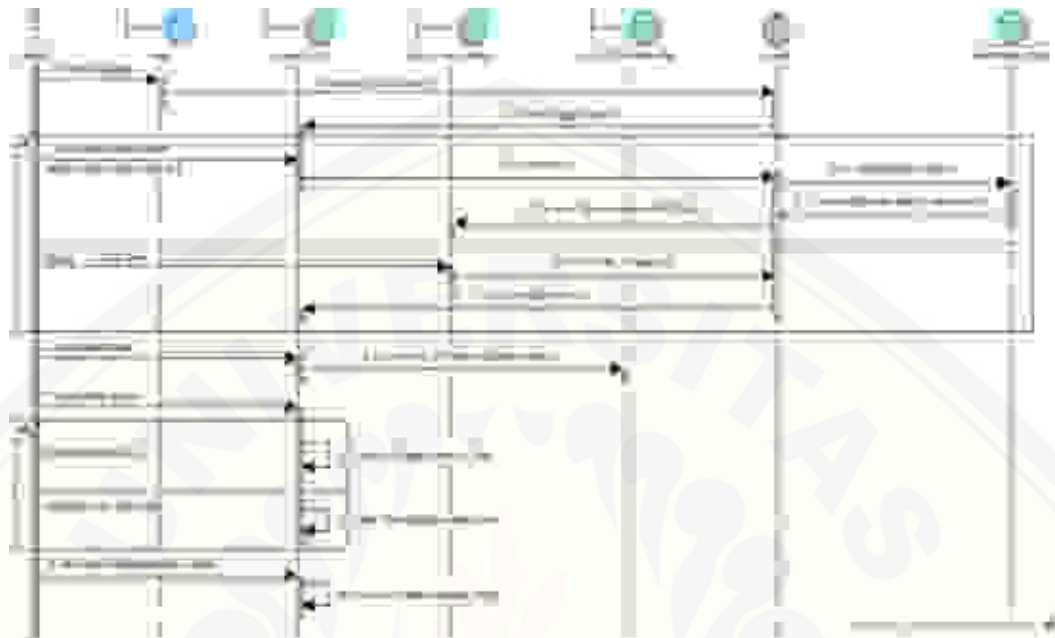
Gambar 4. 12 *Sequence* Diagram Laporan Dokter
Sumber: (Hasil analisis)



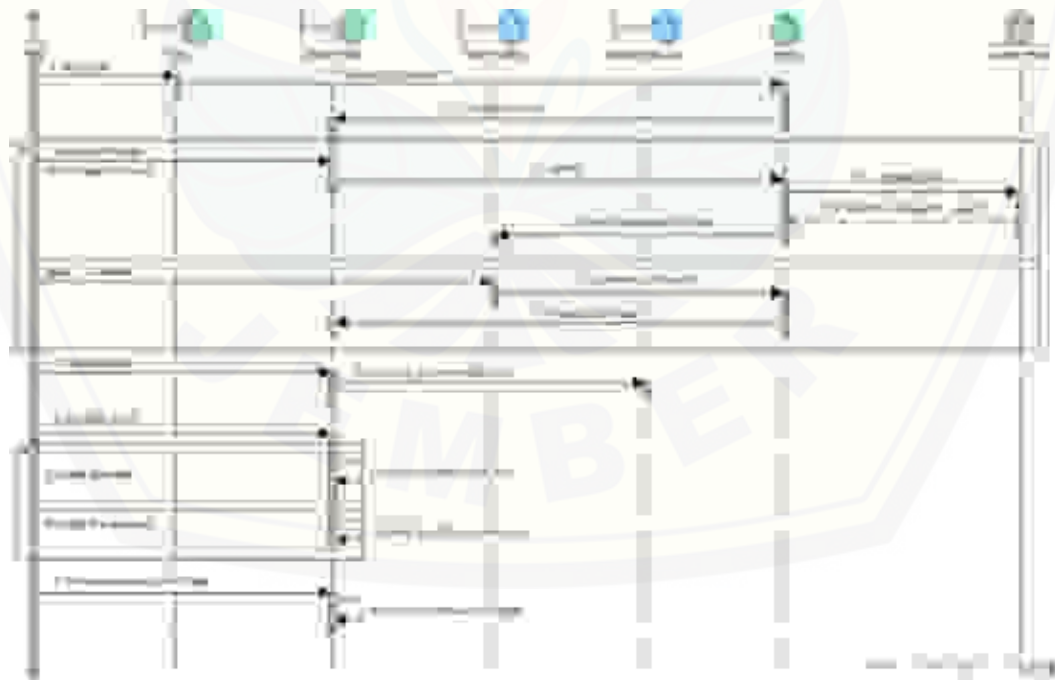
Gambar 4. 15 *Sequence Diagram Laporan Obat*
Sumber: (Hasil analisis)



Gambar 4. 16 *Sequence Diagram Laporan Laboratorium*
Sumber: (Hasil analisis)



Gambar 4. 17 *Sequence* Diagram Laporan Rekam Medis
Sumber: (Hasil analisis)

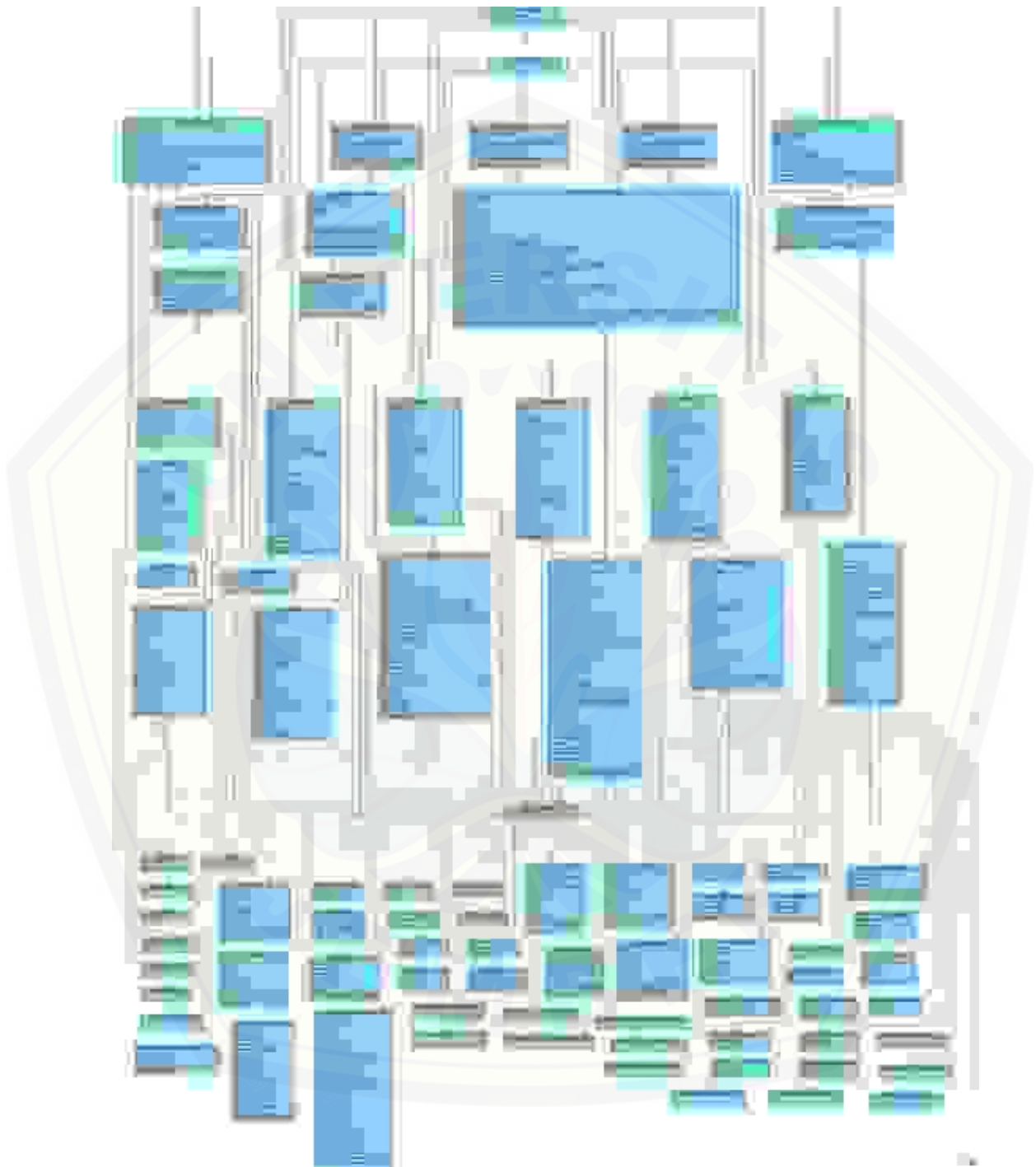


Gambar 4. 18 *Sequence* Diagram Laporan Rujukan
Sumber: (Hasil analisis)

4.3.6 Class Diagram

Class diagram adalah dokumentasi untuk menggambarkan struktur objek statis dari suatu sistem yang menunjukkan *class - class* objek yang menyusun sebuah sistem dan juga hubungan antara *class* objek tersebut. *Class* memiliki 3 area pokok yaitu: nama, atribut, dan operasi. Nama berfungsi untuk member identitas pada sebuah kelas, atribut berfungsi untuk member karakteristik pada data yang dimiliki suatu objek di dalam kelas, sedangkan operasi berfungsi memberikan sebuah fungsi ke sebuah objek. *Class* Diagram Sistem Informasi Rekam Medis Pasien dapat di lihat pada Gambar 4.19.





Gambar 4. 19 *Class Diagram* Sistem Informasi Rekam Medis Pasien
Sumber: (Hasil analisis)

4.5 Pengujian Program

4.5.1 Pengujian *White Box*

Pengujian *white box* merupakan pengujian dari kode-kode program yang meliputi *listing* program, grafik alir, kompleksitas siklomatis, *basis set* dan *test case*. *Whitebox Testing* dilakukan pada setiap fitur yang ada pada Sistem Informasi Rekam Medis. Fitur yang diuji menggunakan *whitebox testing* pada bab ini adalah Fitur Autentifikasi (*Login*) dan fitur Manajemen Data Rekam Medis. *Whitebox testing* untuk fitur-fitur lainnya dapat dilihat pada Lampiran C.

a. Autentifikasi (*Login*)

Pengujian kode-kode program pada fitur Autentifikasi (*Login*) adalah sebagai berikut:

1. *Listing* Program

```

1. public function Login() { 1
2.     $data = array();
3.
4.     if($this->input->post("Submit")) { 3
5.         $this->form_validation-
6.         >set_rules('username','Username','required');
7.         $this->form_validation-
8.         >set_rules('password','Password','required');
9.         5 if($this->form_validation->run()){
10.            $username = $this->input->post('username');
11.            $password = $this->input->post('password');
12.            6 $role = $this->input->post('role');
13.
14.            $this->load->model('user_model','user');
15.            $dataUser = $this->user-
16.            >getUserLogin($role,$username,$password);
17.            $dataLogin = array($username,$password);
18.            7 if($dataUser->num_rows()>0){
19.                if($this->authorization- 8
20.                >LoginVerification($dataUser,$dataLogin,$role){
21.                    9 if($role=="laborat"){
22.                        redirect($role.'/home'); 10
23.                    } 11
24.                    redirect($role.'/index'); 12
25.                }
26.            }
27.            else
28.            13 $data['error'] = '<div class="error_Login">Wrong
29.            Username and Password </div>';
30.        }
31.    }

```

```

26.         else {
27.             14 $data['error'] = '<div class="error_Login">Wrong Use
                rname and Password.</div>';
28.         }
29.     }
30.     else {
31.         15 $data['error'] = validation_errors();
32.     }
33. } 16
34. $this->load->view('Login', $data); 17
35. } 18
    
```

2. Grafik Alir

Grafik Alir fitur Autentifikasi (*Login*) berupa notasi – notasi yang menunjukkan alur kontrol. Grafik alir tersebut di gambarkan dengan simpul dan garis dapat di lihat pada Gambar 4.21.



Gambar 4. 21 Diagram Alir Autentifikasi (*Login*)
 Sumber: (Hasil analisis)

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur Autentifikasi (*Login*) dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 21 - 17 + 2 = 6$$

4. *Basis Set*

Basis set fitur Autentifikasi (*Login*) menghasilkan tiga (3) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

$$\text{Jalur 1} = 1 - 17 - 2 - 3 - 16 - 18 - 19$$

$$\text{Jalur 2} = 1 - 17 - 2 - 3 - 4 - 5 - 15 - 16 - 18 - 19$$

$$\text{Jalur 3} = 1 - 17 - 2 - 3 - 4 - 5 - 6 - 7 - 14 - 16 - 18 - 19$$

$$\text{Jalur 4} = 1 - 17 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 13 - 16 - 18 - 19$$

$$\text{Jalur 5} = 1 - 17 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 11 - 16 - 18 - 19$$

$$\text{Jalur 6} = 1 - 17 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 16 - 18 - 19$$

5. *Test Case*

Tabel 4. 5 *Test Case* Autentifikasi (*Login*) jalur 1
Sumber: (Hasil Analisis)

Test case	User Menjalankan sistem
Target yang diharapkan	Sistem menampilkan form <i>login</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	welcome.php

Tabel 4. 6 *Test Case* Autentifikasi (*Login*) jalur 2
Sumber: (Hasil Analisis)

Test case	User melakukan pengisian <i>username</i> , <i>password</i> , dan memilih <i>role</i> pada form <i>Login</i> . Lalu klik tombol <i>Login</i>
Target yang diharapkan	Sistem menampilkan <i>alert</i> , “ <i>Wrong Username and Password</i> ” di form <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	welcome.php

Tabel 4. 7 *Test Case* Autentifikasi (*Login*) jalur 3
Sumber: (Hasil Analisis)

Test case	User mengisi <i>username</i> , <i>password</i> dan memilih <i>role</i> pada form <i>Login</i> , lalu menekan tombol <i>Login</i>
Target yang diharapkan	Sistem menampilkan <i>alert</i> , “ <i>Wrong</i>

	<i>Username and Password</i> ' di form Login
Hasil pengujian	Benar
Jalur	Jalur 3
File	welcome.php

Tabel 4. 8 *Test Case* Autentifikasi (Login) jalur 4
Sumber: (Hasil Analisis)

Test case	User mengisi <i>username</i> , <i>password</i> dan memilih <i>role</i> pada form Login, lalu menekan tombol Login
Target yang diharapkan	Sistem menampilkan <i>alert</i> , "Wrong Username and Password" di form Login
Hasil pengujian	Benar
Jalur	Jalur 4
File	welcome.php

Tabel 4. 9 *Test Case* Autentifikasi (Login) jalur 5
Sumber: (Hasil Analisis)

Test case	User mengisi <i>username</i> , <i>password</i> dan memilih <i>role</i> pada form Login, lalu menekan tombol Login
Target yang diharapkan	Sistem menampilkan halaman awal sistem sesuai <i>username</i> , <i>password</i> dan <i>role</i> .
Hasil pengujian	Benar
Jalur	Jalur 5
File	welcome.php

Tabel 4. 10 *Test Case* Autentifikasi (Login) jalur 6
Sumber: (Hasil Analisis)

Test case	User mengisi <i>username</i> , <i>password</i> dan memilih <i>role</i> pada form Login, lalu menekan tombol Login
Target yang diharapkan	Sistem menampilkan halaman awal dengan <i>role</i> =laborat
Hasil pengujian	Benar
Jalur	Jalur 5
File	welcome.php

b. Manajemen Data Rekam Medis

Pengujian kode – kode program pada fitur Manajemen Data Rekam Medis adalah sebagai berikut:

1. Listing Program

```

1. 1 class Rekammedis extends CI_Controller {
2.
3.     public function __construct() { 2
4.         parent::__construct();
5.         3 $this->load->model("rekammedis_model", "rekammedis");
6.         // $this->authorization->isAdminLogin();
7.         4 if (!$this->session->userdata("rkmds-logged-in")) { 4
8.             5 redirect('/'); 5
9.         } 6
10.    } 7
11.
12.    public function index() { 8
13.        $data = array();
14.        9 $data['rekammedis'] = $this->rekammedis->getDataRekammedis();
15.        $this->load->view('rekammedis-data', $data);
16.    } 10
17.
18.    public function add() { 11
19.        12 if ($this->input->post("submit")) { 12
20.            $id_pasien = $this->input->post("id_pasien");
21.            $diagnosa = $this->input->post("diagnosa");
22.            $hasil_pemeriksaan = $this->input-
23.            >post("hasil_pemeriksaan");
24.            $obat_dianjurkan = $this->input->post("obat_dianjurkan");
25.            $id_dokter = $this->input->post("id_dokter");
26.            $jenis_pemeriksaan = $this->input-
27.            >post("jenis_pemeriksaan");
28.            $tanggal_masuk = $this->input->post("tanggal_masuk");
29.            $tanggal_pemeriksaan = $this->input-
30.            >post("tanggal_pemeriksaan");
31.            $anamnesis = $this->input->post("anamnesis");
32.            $arrtgl = explode("/", $tanggal_masuk);
33.            13 $arrtglpemeriksaan = explode("/", $tanggal_pemeriksaan);
34.            $kode_rekammedis = $this->input->post('kode_rekammedis');
35.
36.            $data = array(
37.                "id_pasien"=>$id_pasien,
38.                "diagnosa"=>$diagnosa,
39.                "hasil_pemeriksaan"=>$hasil_pemeriksaan,
40.                "obat_dianjurkan"=>$obat_dianjurkan,
41.                "id_dokter"=>$id_dokter,
42.                "jenis_pemeriksaan"=>$jenis_pemeriksaan,
43.                "tanggal_masuk"=>date("Y-m-d", strtotime($arrtgl[2]."-
44.                ".$arrtgl[1]."-".$arrtgl[0])),
45.                "tanggal_pemeriksaan"=>date("Y-m-
46.                d", strtotime($arrtglpemeriksaan[2]."-".$arrtglpemeriksaan[1]."-
47.                ".$arrtglpemeriksaan[0])),

```



```

42.         "anamnesis"=>$anamnesis,
43.         "kode_rekammedis"=>$kode_rekammedis,
44.     );
45.
46.     14 if($this->rekammedis->insert($data){
47.         $this->session-
>set_flashdata('message', 'Successfully add new data. '); 15
48.         redirect("rekammedis");
49.     }else{
50.         $this->session-
>set_flashdata('message', 'Failed add new data. '); 16
51.         redirect("rekammedis");
52.     }
53. }
54. 17 $id = "";
55. $sql = 'SELECT COUNT(*) AS total FROM tb_rekammedis ';
56. 18 $total = $this->db->query($sql)->row('total');
57.
58.     if($total==0){ //never made 19
59.         $autonum = "0001"; 20
60.     }
61.     else {
62.         21 $sql = 'SELECT * FROM tb_rekammedis ORDER BY id_rekammedis D
ESC LIMIT 1';
63.         $last = $this->db->query($sql)->row('kode_rekammedis');
64.         $lastnum = substr($last,4,4);
65.         $autonum = (int) $lastnum + 1;
66.         $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
67.     }
68.     $id = "RKM-".$autonum;
69.     $data['id'] = $id;
70.     22
71.     $this->load->view('add-rekammedis', $data);
72.     23 }
73.
74.     public function edit($id){ 24
75.         $data = array();
76.         25 $rekammedis = $this->rekammedis->getRekammedis($id);
77.         $data['rekammedis'] = $rekammedis;
78.         $this->load->view('edit-rekammedis', $data);
79.         26 if($this->input->post("submit")){
80.             $id_pasien = $this->input->post("id_pasien");
81.             $diagnosa = $this->input->post("diagnosa");
82.             $hasil_pemeriksaan = $this->input-
>post("hasil_pemeriksaan");
83.             $obat_dianjurkan = $this->input->post("obat_dianjurkan");
84.             $id_dokter = $this->input->post("id_dokter");
85.             $jenis_pemeriksaan = $this->input-
>post("jenis_pemeriksaan");
86.             $tanggal_masuk = $this->input->post("tanggal_masuk");
87.             $tanggal_pemeriksaan = $this->input-
>post("tanggal_pemeriksaan");
88.             $anamnesis = $this->input->post("anamnesis");
89.             $arrtgl = explode("/", $tanggal_masuk);
90.             27 $id_rekammedis = $this->input->post("id_rekammedis");
91.             $arrtgl = explode("/", $tanggal_masuk);

```

```

92.         $arrtglpemeriksaan = explode("/", $tanggal_pemeriksaan);
93.
94.         $data = array(
95.             "id_pasien"=>$id_pasien,
96.             "diagnosa"=>$diagnosa,
97.             "hasil_pemeriksaan"=>$hasil_pemeriksaan,
98.             "obat_dianjurkan"=>$obat_dianjurkan,
99.             "id_dokter"=>$id_dokter,
100.            "jenis_pemeriksaan"=>$jenis_pemeriksaan,
101.            "tanggal_masuk"=>date("Y-m-d", strtotime($arrtgl[2]."-
102.            ".$arrtgl[1]."-".$arrtgl[0])),
103.            "tanggal_pemeriksaan"=>date("Y-m-
104.            d", strtotime($arrtglpemeriksaan[2]."-".$arrtglpemeriksaan[1]."-
105.            ".$arrtglpemeriksaan[0])),
106.            "anamnesis"=>$anamnesis,
107.        );
108.
109.        28 if($this->rekammedis->update($id_rekammedis,$data)){
110.            $this->session-
111.            >set_flashdata('message', 'Successfully update data.'; 29
112.            redirect("rekammedis");
113.        }else{
114.            30 $this->session-
115.            >set_flashdata('message', 'Failed update data. ');
116.            redirect("rekammedis");
117.        }
118.    } 31
119. } 32
120.
121. public function delete($id){ 33
122.     $this->rekammedis->delete($id);
123.     34 $this->session-
124.     >set_flashdata('message', 'Successfully delete data. ');
125.     redirect("rekammedis");
126. } 35
127.
128. function pdf($id_rekammedis){ 36
129.     //$this->load->helper('pdf_helper');
130.     $data= array();
131.     $data['rekammedis'] = $this->db
132.     37 ->where("id_rekammedis",$id_rekammedis)
133.     -
134.     >select('tb_rekammedis.*, tb_pasien.kode_pasien,tb_pasien.nama as nama_p
135.     asien, tb_dokter.nama as nama_dokter', false)
136.     -
137.     >join('tb_pasien', 'tb_rekammedis.id_pasien = tb_pasien.id_pasien')
138.     -
139.     >join('tb_dokter', 'tb_rekammedis.id_dokter = tb_dokter.id_dokter')
140.     ->get("tb_rekammedis")->row_array();
141.     $this->load->view('pdf/rekammedis', $data);
142. } 38
143. } 39

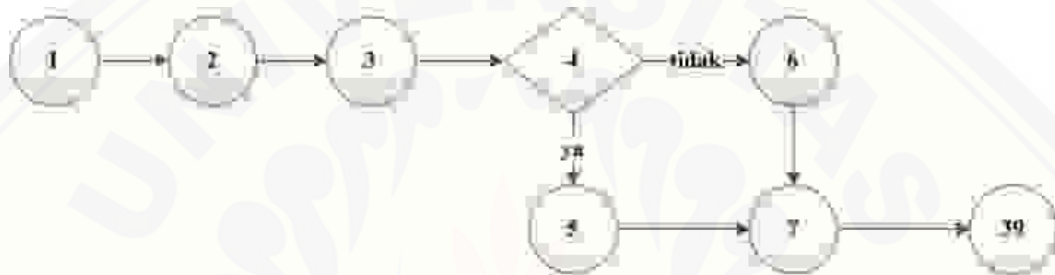
```

2. Grafik Alir

Grafik Alir fitur Manajemen Data Rekam Medis berupa notasi-notasi yang menunjukkan alur kontrol. Fitur Manajemen Data Rekam Medis terbagi menjadi beberapa fungsi, yaitu *construct*, *index*, *add*, *edit*, *delete*, dan *pdf*.

2.1. Grafik Alir Fitur Manajemen Data Rekam Medis Fungsi *Construct*

Grafik alir fitur Manajemen Data Rekam Medis Fungsi *Construct* digambarkan dengan simpul dan garis dapat di lihat pada Gambar 4.22.



Gambar 4. 22 Grafik Alir Manajemen Data Rekam Medis Fungsi *Construct*
 Sumber: (Hasil analisis)

2.2. Grafik Alir Fitur Manajemen Data Rekam Medis Fungsi *Index*

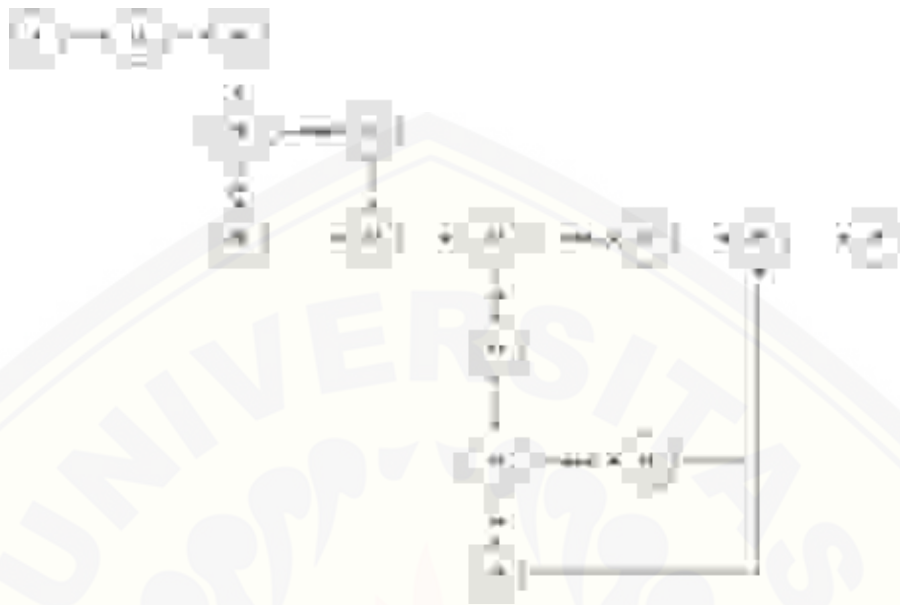
Grafik alir fitur Manajemen Data Rekam Medis Fungsi *Index* digambarkan dengan simpul dan garis dapat di lihat pada Gambar 4.23.



Gambar 4. 23 Grafik Alir Manajemen Data Rekam Medis Fungsi *Index*
 Sumber: (Hasil analisis)

2.3. Grafik Alir Fitur Manajemen Data Rekam Medis Fungsi *Add*

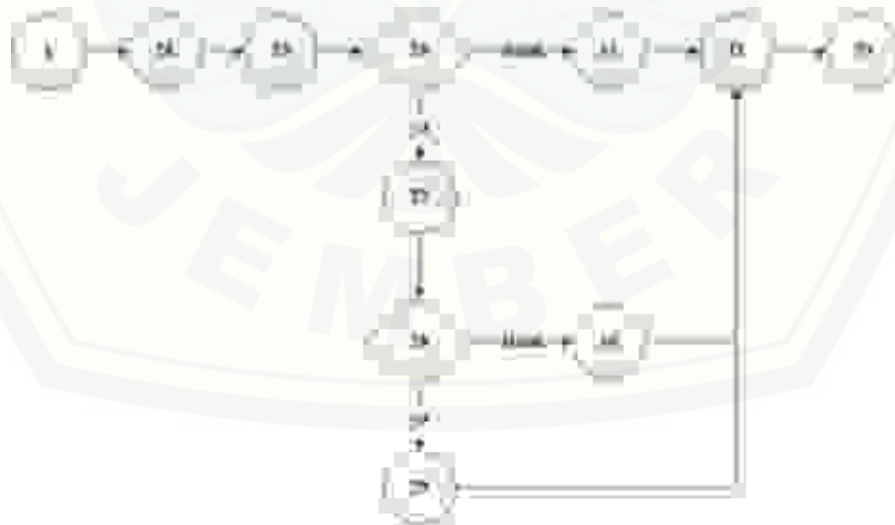
Grafik alir fitur Manajemen Data Rekam Medis Fungsi *Add* digambarkan dengan simpul dan garis dapat di lihat pada Gambar 4.24.



Gambar 4. 24 Grafik Alir Manajemen Data Rekam Medis Fungsi *Add*
Sumber: (Hasil analisis)

2.4. Grafik Alir Fitur Manajemen Data Rekam Medis Fungsi *Edit*

Grafik alir fitur Manajemen Data Rekam Medis Fungsi *Edit* digambarkan dengan simpul dan garis dapat di lihat pada Gambar 4.25.



Gambar 4. 25 Diagram Alir Manajemen Data Rekam Medis Fungsi *Edit*
Sumber: (Hasil analisis)

2.5. Grafik Alir Fitur Manajemen Data Rekam Medis Fungsi *Delete*

Grafik alir fitur Manajemen Data Rekam Medis Fungsi *Delete* digambarkan dengan simpul dan garis dapat di lihat pada Gambar 4.26.



Gambar 4. 26 Grafik Alir Manajemen Data Rekam Medis Fungsi *Delete*
 Sumber: (Hasil analisis)

2.6. Grafik Alir Fitur Manajemen Data Rekam Medis Fungsi *Pdf*

Grafik alir fitur Manajemen Data Rekam Medis Fungsi *Delete* digambarkan dengan simpul dan garis dapat di lihat pada Gambar 4.27.



Gambar 4. 27 Grafik Alir Manajemen Data Rekam Medis Fungsi *Pdf*
 Sumber: (Hasil analisis)

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur Manajemen Data Rekam Medis dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2$$

Dengan $V(G)$ adalah Kompleksitas siklomatis, E adalah *Edge* atau tepi atau garis dari sebuah grafik, dan N adalah *Node* atau titik dari sebuah grafik. Maka perhitungan dari setiap grafik dari Manajemen Data Rekam Medis adalah sebagai berikut.

3.1. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Rekam Medis

Fungsi *construct*

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 8 - 8 + 2 \\
 &= 2
 \end{aligned}$$

3.2. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Rekam Medis

Fungsi *Index*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

3.3. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Rekam Medis

Fungsi *Add*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 17 - 15 + 2 \\ &= 4 \end{aligned}$$

3.4. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Rekam Medis

Fungsi *Edit*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 12 - 11 + 2 \\ &= 3 \end{aligned}$$

3.5. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Rekam Medis

Fungsi *Delete*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

3.6. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Rekam Medis

Fungsi *Pdf*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

4. *Basis Set*

4.1. *Basis set* Fitur Manajemen Data Rekam Medis Fungsi *Construct*

Basis set Manajemen Data Rekam Medis fungsi *construct* menghasilkan dua

(2) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 - 2 - 3 - 4 - 6 - 7 - 39

Jalur 2 : 1 - 2 - 3 - 4 - 5 - 7 - 39

4.2. *Basis set* Fitur Manajemen Data Rekam Medis Fungsi *Index*

Basis set Manajemen Data Rekam Medis fungsi *index* menghasilkan satu (1)

jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 8 – 9 – 10 – 39

4.3. *Basis set* Fitur Manajemen Data Rekam Medis Fungsi *Add*

Basis set Manajemen Data Rekam Medis fungsi *Add* menghasilkan empat (4)

jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 11 – 18 – 19 – 20 – 22 – 12 – 13 – 14 – 15 – 23 – 39

Jalur 2 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 13 – 14 – 15 – 23 – 39

Jalur 3 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 13 – 14 – 16 – 23 – 39

Jalur 4 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 17 – 23 – 39

4.4. *Basis set* Fitur Manajemen Data Rekam Medis Fungsi *Edit*

Basis set Manajemen Data Rekam Medis fungsi *Edit* menghasilkan tiga (3)

jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 24 – 25 – 26 – 31 – 32 – 39

Jalur 2 : 1 – 24 – 25 – 26 – 27 – 28 – 30 – 32 – 39

Jalur 3 : 1 – 24 – 25 – 26 – 27 – 28 – 29 – 32 – 39

4.5. *Basis set* Fitur Manajemen Data Rekam Medis Fungsi *Delete*

Basis set Manajemen Data Rekam Medis fungsi *Delete* menghasilkan satu (1)

jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 33 – 34 – 35 – 39

4.6. *Basis set* Fitur Manajemen Data Rekam Medis Fungsi *Pdf*

Basis set Manajemen Data Rekam Medis fungsi *Delete* menghasilkan satu (1)

jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 33 – 34 – 35 – 39

5. *Test Case*

Tabel 4. 11 *Test Case* Fitur Manajemen Data Rekam Medis fungsi *construct* jalur 1

Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif kemudian klik menu Rekam Medis
-----------	--

Target yang diharapkan	Sistem menampilkan halaman Rekam Medis
Hasil pengujian	Benar
Jalur	Jalur 1
File	rekammedis.php

Tabel 4. 12 *Test Case* Fitur Manajemen Data Rekam Medis fungsi *construct* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rekam Medis, <i>User</i> melakukan <i>Logout</i> dan menuju halaman <i>Login</i> dan ketika di <i>back</i> tetap menampilkan halaman <i>Login</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	rekammedis.php

Tabel 4. 13 *Test Case* Fitur Manajemen Data Rekam Medis fungsi *Index* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif lalu klik menu Rekam Medis
Target yang diharapkan	Sistem menampilkan halaman Rekam Medis
Hasil pengujian	Benar
Jalur	Jalur 1
File	rekammedis.php

Tabel 4. 14 *Test Case* Fitur Manajemen Data Rekam Medis fungsi *Add* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rekam Medis, <i>User</i> mengklik Tombol Tambah Rekam Medis kemudian mengisi <i>form</i> Tambah Rekam Medis setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Rekam Medis serta menampilkan <i>alert</i> ,” <i>Successfully add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 1
File	rekammedis.php

Tabel 4. 15 *Test Case* Fitur Manajemen Data Rekam Medis fungsi *Add* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rekam Medis, <i>User</i> mengklik Tombol Tambah Rekam Medis kemudian mengisi <i>form</i> Tambah Rekam Medis setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Rekam Medis serta menampilkan <i>alert,</i> ” <i>Successfully add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 2
File	rekammedis.php

Tabel 4. 16 *Test Case* Fitur Manajemen Data Rekam Medis fungsi *Add* jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rekam Medis, <i>User</i> mengklik Tombol Tambah Rekam Medis kemudian mengisi <i>form</i> Tambah Rekam Medis setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Rekam Medis dan menampilkan <i>alert,</i> ” <i>Failed add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 3
File	rekammedis.php

Tabel 4. 17 *Test Case* Fitur Manajemen Data Rekam Medis fungsi *Add* jalur 4
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rekam Medis, <i>User</i> mengklik Tombol Tambah Rekam Medis
Target yang diharapkan	Sistem menampilkan halaman Tambah Rekam Medis
Hasil pengujian	Benar
Jalur	Jalur 4
File	rekammedis.php

Tabel 4. 18 *Test Case* Fitur Manajemen Data Rekam Medis fungsi *Edit* jalur 1
 Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rekam Medis, <i>User</i> mengklik Tombol <i>Edit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Edit</i> Rekam Medis
Hasil pengujian	Benar
Jalur	Jalur 1
File	rekammedis.php

Tabel 4. 19 *Test Case* Fitur Manajemen Data Rekam Medis fungsi *Edit* jalur 2
 Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Rekam Medis, <i>User</i> mengisi <i>form edit</i> Rekam Medis kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Rekam Medis dan menampilkan <i>alert,</i> ” <i>Failed update data..</i> ”
Hasil pengujian	Benar
Jalur	Jalur 2
File	rekammedis.php

Tabel 4. 20 *Test Case* Fitur Manajemen Data Rekam Medis fungsi *Edit* jalur 3
 Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Rekam Medis, <i>User</i> mengisi <i>form edit</i> Rekam Medis kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Rekam Medis dan menampilkan <i>alert,</i> ” <i>Successfully update data..</i> ”
Hasil pengujian	Benar
Jalur	Jalur 3
File	rekammedis.php

Tabel 4. 21 *Test Case* Fitur Manajemen Data Rekam Medis fungsi *Delete* jalur 1
 Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rekam Medis, <i>User</i> mengklik tombol <i>Remove</i>
Target yang diharapkan	Sistem menampilkan halaman Rekam Medis dan menampilkan <i>alert,</i> ”

	<i>Successfully delete data..”</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	rekammedis.php

Tabel 4. 22 *Test Case* Fitur Manajemen Data Rekam Medis fungsi Pdf jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rekam Medis, <i>User</i> mengklik tombol Cetak
Target yang diharapkan	Sistem menampilkan halaman pdf “DATA REKAM MEDIS PASIEN”
Hasil pengujian	Benar
Jalur	Jalur 1
File	rekammedis.php

4.5.2 Pengujian *Black Box*

Pengujian *Black box* dilakukan untuk mengetahui apakah *input* dan *output* dari sistem sesuai dengan kebutuhan fungsional atau tidak. Pengujian dilakukan pada *form* untuk setiap *usecase*. Pengujian ini dilakukan oleh calon pengguna Sistem Informasi Rekam Medis Klinik Harapan Bersama Ambulu. Dokumen hasil pengujian *black box* Sistem Informasi Rekam Medis Klinik Harapan Bersama Ambulu dapat dilihat pada Lampiran D.

4.6 Pemeliharaan

Pada tahap pemeliharaan sistem belum dilakukan karena dari pengujian sistem, sistem tidak mengalami kesalahan dan sudah memenuhi kebutuhan pengguna untuk saat ini. Tahapan ini akan terus dilakukan, karena tidak menutup kemungkinan ketika ada kebutuhan pengguna untuk menambahkan fitur – fitur baru. Pada pemeliharaan sistem yang dikembangkan ini akan mudah dilakukan karena pengembangan sistem menggunakan konsep sistem berorientasi objek jadi bila ada kesalahan ataupun penambahan fitur-fitur dalam kedepannya, tidak harus mengubah sistem secara keseluruhan.

BAB 5. HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil dan pembahasan dari Pengembangan Sistem Informasi Rekam Medis Pasien Klinik Harapan Bersama Ambulu Menggunakan Metode SDLC Model *Waterfall*. Pembahasan dimaksudkan untuk menjelaskan sistem atau perangkat lunak yang telah dikembangkan.

5.1 Implementasi Sistem

Antarmuka dari fitur-fitur sistem yang telah dibuat akan disajikan dalam subbab ini beserta keterangan yang menunjukkan fungsinya. Antarmuka yang tercakup di dalam sistem adalah:

5.1.1 Autentifikasi (*Login*)

Sistem Informasi Rekam Medis Pasien memiliki 4 aktor, yaitu: administrator, laborat, apoteker, dan manajer. Form *login* terdiri dari *Username*, *Password*, dan *Role*. Aktor memasukkan *Username*, *Password*, serta memilih *role* yang dimiliki. Jika *Username*, *Password*, dan *role* benar maka sistem menampilkan halaman *Dashboard* yang sesuai dengan hak akses Aktor tersebut. Halaman Autentifikasi (*Login*) dapat dilihat pada gambar 5.1.



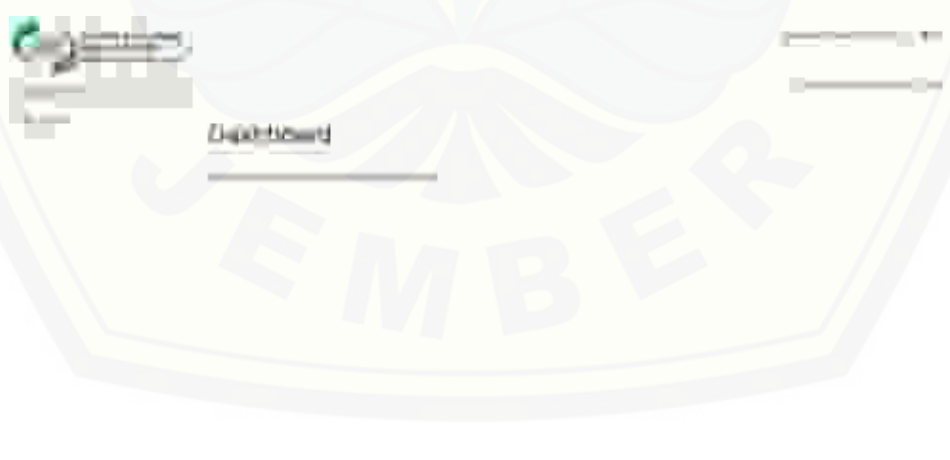
Gambar 5. 1 Halaman Autentifikasi (*Login*) Sistem Informasi Rekam Medis Pasien
Sumber: (Hasil Analisis)

5.1.2 Halaman *Dashboard*

Setelah aktor berhasil melakukan *Login*, maka halaman pertama yang muncul adalah halaman *Dashboard*. Tiap aktor memiliki halaman *dashoard* yang berbeda. Khusus pada aktor administrator, halaman *dashboard* berisi data user aktif. Artinya hanya administrator saja yang bisa melakukan tambah, ubah, dan hapus user atau aktor yang bisa *Login* sistem. Halaman *dashboard* untuk tiap aktor dapat dilihat berturut-turut pada gambar 5.2, gambar 5.3, gambar 5.4, dan gambar 5.5.



Gambar 5. 2 Halaman *Dashboard* Untuk Aktor dengan *Role* Admin
Sumber: (Hasil Analisis)



Gambar 5. 3 Halaman *Dashboard* Untuk Aktor dengan *Role* Manajer
Sumber: (Hasil Analisis)



Gambar 5. 4 Halaman *Dashboard* Untuk Aktor dengan *Role* Apoteker
Sumber: (Hasil Analisis)



Gambar 5. 5 Halaman *Dashboard* Untuk Aktor dengan *Role* Laborat
Sumber: (Hasil Analisis)

5.1.3 Halaman Pegawai

Halaman pegawai merupakan halaman yang hanya bisa diakses oleh aktor dengan role admin. Halaman pegawai berisi tabel Data Pegawai. aktor bisa melakukan tambah pegawai dengan cara klik tombol Tambah Pegawai yang berada pada sebelah kanan atas tabel Data Pegawai, edit pegawai berdasarkan ID Pegawai dengan cara menekan tombol *Edit* di kolom *Action* pada tabel Data Pegawai, dan aktor juga bisa melakukan *remove* data berdasarkan ID Pegawai dengan cara

menekan tombol *Remove* di kolom *Action* pada tabel Data Pegawai . Halaman Pegawai dapat dilihat pada gambar 5.6.



Gambar 5. 6 Halaman Pegawai
Sumber: (Hasil Analisis)

5.1.4 Halaman Pasien

Halaman Pasien berisi tabel Data Pasien. Pada halaman ini aktor yang memiliki role admin dapat melakukan tambah data pasien, edit data pasien berdasarkan ID Pasien, dan juga remove data pasien berdasarkan ID Pasien. Halaman pasien juga merupakan halaman yang signifikan sebab pertama kali pasien mendaftar untuk berobat di Klinik, aktor akan memasukkan data pasien ke dalam sistem melalui halaman pasien ini. Gambar 5.7 merupakan gambar yang menunjukkan halaman pasien.



Gambar 5. 7 Halaman Pasien
Sumber: (Hasil Analisis)

5.1.5 Halaman Kamar

Halaman Kamar merupakan halaman yang menunjukkan ketersediaan kamar di Klinik. Halaman Kamar juga menyediakan informasi tentang jumlah kamar baik itu yang terisi maupun yang kosong sehingga memudahkan admin untuk menemukan kamar untuk pasien yang akan menjalani rawat inap. Tidak hanya itu, pada halaman kamar terdapat pula informasi mengenai tarif, kelas, beserta fasilitas yang ada didalamnya. Halaman ini hanya bisa diakses oleh admin. Admin dapat menambah, mengubah, dan juga menghapus kamar pada sistem. Halaman kamar ditunjukkan pada gambar 5.8.



Gambar 5. 8 Halaman Kamar
Sumber: (Hasil Analisis)

5.1.6 Halaman Fasilitas

Halaman Fasilitas merupakan halaman yang berisi tentang beberapa fasilitas inventaris klinik yang kebanyakan digunakan di dalam kamar pasien. Dalam halaman fasilitas terdapat tabel Data Fasilitas yang berisi nama fasilitas beserta jumlahnya. Halaman fasilitas hanya dapat diakses oleh aktor dengan role admin. Admin dapat melakukan tambah data fasilitas, edit, maupun remove. Halaman ini juga berfungsi sebagai control bagi pegawai yang mengawasi bagian fasilitas agar selalu berkoordinasi dengan admin saat melakukan pengecekan jumlah fasilitas yang tersedia di klinik agar sesuai dengan yang ada di sistem. Berikut adalah gambar 5.9 yang menunjukkan Halaman Fasilitas.



Gambar 5. 9 Halaman Fasilitas
Sumber: (Hasil Analisis)

5.1.7 Halaman Dokter

Halaman dokter berisi tentang data dokter yang praktek di Klinik Harapan Bersama. Pada halaman ini, aktor yang memiliki role sebagai admin saja yang berhak melakukan tambah, edit, dan remove. Halaman dokter dapat dilihat pada gambar 5.10.



Gambar 5. 10 Halaman Dokter
Sumber: (Hasil Analisis)

5.1.8 Halaman Obat

Pada halaman obat, aktor yang berhak mengakses ada dua (2) yaitu aktor yang memiliki role sebagai admin dan apoteker. Admin dan apoteker berhak melakukan

tambah data obat, edit maupun remove. Halaman Obat berisi tentang data obat yang berada di klinik. Halaman obat dapat dilihat pada gambar 5.11.



Gambar 5. 11 Halaman Obat
Sumber: (Hasil Analisis)

5.1.9 Halaman Laborat

Halaman Laborat dapat diakses oleh aktor dengan role admin dan laborat. Admin dan laborat berhak melakukan tambah data laborat, cetak maupun remove. Untuk data laborat, Action hanya berisi remove dan cetak. Hal ini dilakukan agar menghindari data laborat yang kemungkinan diubahnya saat setelah melakukan cetak laborat. Halamman laborat ditunjukkan pada gambar 5.12.



Gambar 5. 12 Halaman Laborat
Sumber: (Hasil Analisis)

5.1.10 Halaman Rekam Medis

Halaman Rekam Medis merupakan halaman yang berhubungan pencatatan hasil pemeriksaan pasien oleh dokter yang menangani. Pada halaman rekam medis pasien memiliki kemungkinan mendapatkan satu (1), dua (2) atau lebih pencatatan riwayat rekam medis tergantung pada berapa kali si pasien dengan ID Pasien sama melakukan pemeriksaan atau berobat di Klinik. Dan memiliki kemungkinan juga satu pasien diperiksa oleh dokter yang berbeda saat melakukan pemeriksaan atau berobat lebih dari sekali tergantung pada dokter yang bertugas pada saat itu. Halaman rekam medis dapat dilihat pada gambar 5.13.



Gambar 5. 13 Halaman Rekam Medis
Sumber: (Hasil Analisis)

5.1.11 Halaman Rujukan

Halaman Rujukan hanya dapat diakses oleh aktor dengan role admin. Pada halaman rujukan, admin dapat melakukan tambah data rujukan berdasarkan kesepakatan antara keluarga pasien dan bisa juga atas saran dari dokter yang menangani pasien. Data Rujukan juga dapat di cetak, edit, dan remove berdasarkan ID Rujukan. Halaman Rujukan dapat dilihat pada gambar 5.14.



Gambar 5. 14 Halaman Rujukan
Sumber: (Hasil Analisis)

5.1.12 Halaman Transaksi Pembayaran

Halaman Transaksi Pembayaran memiliki tiga(3) tombol yang berfungsi untuk mencatat transaksi yang berbeda. Transaksi Pembayaran yang dimaksud adalah Transaksi Pembayaran Perawatan, Transaksi Pembayaran Laboratorium, dan Transaksi Pembayaran Obat. Halaman transaksi pembayaran hanya bisa diakses oleh aktor dengan role admin. Halaman transaksi pembayaran dapat ditunjukkan pada Lampiran E.

5.1.13 Halaman Laporan

Halaman Laporan dapat diakses oleh dua (2) aktor, yaitu aktor yang memiliki role admin dan manajer. Halaman Laporan berisi sembilan (9) tombol yaitu, tombol pasien, pegawai, laboratorium, kamar, fasilitas, rekam medis, dokter, obat, dan rujukan. Jika salah satu tombol tersebut di klik oleh aktor, maka sistem akan menampilkan halaman data sesuai dengan yang di klik. Misal aktor mengklik tombol pasien, maka sistem akan menampilkan halaman data pasien. Namun untuk laporan hanya bisa dilihat atau dicetak saja tanpa bisa melakukan tambah data, edit, ataupun remove. Halaman laporan beserta tampilan halaman setelah aktor mengklik salah satu tombol di halaman laporan ditunjukkan pada lampiran E.

BAB 6. PENUTUP

6.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil penelitian yang sudah dilakukan adalah sebagai berikut :

1. Sistem Informasi Rekam Medis Klinik Harapan Bersama dapat meningkatkan efektifitas, efisiensi, dan relevan pada saat pencatatan data pasien di Klinik Harapan Bersama.
2. Sistem Informasi Rekam Medis Klinik Harapan Bersama merupakan sistem informasi berbasis *web* dibangun berdasarkan pada kebutuhan Klinik Harapan Besama Ambulu.
3. Sistem Informasi Rekam Medis Klinik Harapan Bersama terdiri dari fitur utama yaitu pencatatan data rekam medis pasien atau Manajemen data Rekam Medis serta terdapat fitur tambahan yaitu Autentifikasi (*Login*), Manajemen Data *User*, Manajemen Data Pegawai, Manajemen Data Pasien, Manajemen Data Kamar, Manajemen Data Fasilitas, Manajemen Data Dokter, Manajemen Data Obat, Manajemen Data Laborat, Manajemen Data Rujukan, Manajemen Transaksi Pembayaran, dan Melihat Laporan.

6.2 Saran

Pengembangan lebih lanjut dari penelitian ini dapat dilakukan dengan menambahkan ruang lingkup penelitian dan hasil dari penelitian yang lebih lengkap. Pengembangan lebih lanjut dapat dilakukan dengan penambahan fitur yang tidak terlihat terlalu kaku misal pada saat melihat dan membaca laporan. Penambahan fitur yang dimaksud misalnya dengan penambahan tampilan grafik untuk menampilkan banyaknya pengunjung klinik berdasarkan hari, minggu, dan bulan.

DAFTAR PUSTAKA

- Badan Pengembangan dan Pembinaan Bahasa, KEMDIKBUD. (2015, Januari). *Kamus Besar Bahasa Indonesia (KBBI)*. Retrieved September 19, 2015, from Kamus Besar Bahasa Indonesia (KBBI): <http://kbbi.web.id/klinik>
- Bassil, Y. (2012, May). A Simulation Model for the Waterfall. *International Journal of Engineering & Technology (iJET)*, II, 1-7.
- Jogianto, H. (1989). *Pengenalan Komputer: Dasar Ilmu Komputer, Pemrograman, Sistem Informasi, dan Intelegensi Buatan*. Yogyakarta: Andi Yogyakarta.
- John, N. F. (1995). *Information System*.
- Lestari, E. d. (2011). SISTEM INFORMASI REKAM MEDIK PADA RUMAH SAKIT BERSALIN GRAHA RAP. *Jurnal Sistem Informasi (JSI)*, VOL. 3, NO. 2, Oktober 2011., 388-397.
- Nuryani, K. G. (2015). MEMBANGUN SISTEM INFORMASI PENGELOLAAN SURAT MENGGUNAKAN METODE WATERFALL STUDI KASUS DIREKTORAT PERENCANAAN DAN PENGEMBANGAN PENDANAAN PEMBANGUNAN. *e-Proceeding of Engineering : Vol.2, No.1 April 2015 / Page 1098*, 1098.
- PERMENKES RI. (2008). Retrieved September 19, 2015, from <http://www.apikes.com/files/permenkes-no-269-tahun-2008.pdf>
- Purwanto. (2017, Februari 7). *Metodologi System Development Life Cycle (SDLC)*. Retrieved Januari 22, 2018, from <https://medium.com:https://medium.com/@purwanto.dev/metodologi-system-development-life-cycle-sdlc-2f0349df1364>
- Reza, A. (2010, September 24). *Definisi Rekam Medis*. Retrieved September 20, 2015, from Blog.ugm.ac.id: <http://blog.ugm.ac.id/2010/09/24/definisi-rekam-medis/>
- Sasmito, G. W. (2017). Penerapan Metode Waterfall Pada Desain Sistem Informasi

Geografis Industri Kabupaten Tegal. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 1-7.

Sukanto, W. S. (2012). SISTEM TERPADU REKAM MEDIK RUMAH SAKIT. *JURNAL INFORMATIKA Vol 6, No. 1, Januari 2012*, 566-575.



LAMPIRAN

LAMPIRAN A. DESAIN SISTEM

A.1 Use case Skenario

A.1.1 Use case Skenario Manajemen Data User

Tabel A.1 Use Case Skenario Manajemen Data User

ID Use Case	UC-02
Nama	Manajemen Data User
Aktor	Administrator
Pre-Condition	Menampilkan Data User
Post-Condition	Data user yang telah di kelola (tambah, ubah, hapus, dan baca)
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Klik menu Dashboard. Menu dashboard biasanya akan muncul pertama kali setelah admin melakukan <i>Login</i>	2. Menampilkan halaman Dashboard atau halaman awal yang berisi tentang tombol Tambah User dan tabel Daftar User Aktif. Tabel Daftar User Aktif berisi tentang Nama, Alamat, Telepon, Username, dan Action. Kolom Action berisi tombol <i>Edit</i> dan tombol <i>Remove</i>
SKENARIO NORMAL Aktor menambah data user	
3. Jika admin ingin menambah user baru, maka klik tombol Tambah User	4. Menampilkan halaman Tambah User yang berisi: <ul style="list-style-type: none"> - <i>Textfield</i> Nama - <i>Textfield</i> Alamat - <i>Textfield</i> Telepon - <i>Textfield</i> Username - <i>Textfield</i> Password - Pilihan <i>Dropdown</i> Role - Tombol Submit dan tombol <i>Cancel</i>

5. Memasukkan data <i>user</i>	
6. Klik tombol <i>Submit</i>	
	7. Melakukan validasi data
	8. Apabila data yang diisi sudah valid, sistem menampilkan data <i>user</i> yang telah ditambah ke dalam tabel Daftar <i>User Aktif</i> dan menampilkan kembali halaman Dashboard
	9. Sistem menyimpan data di <i>database</i>

SKENARIO ALTERNATIF
Aktor menambah data *user* tetapi ingin membatalkannya

6a. Ketika memasukkan data <i>user</i> pada halaman Tambah <i>User</i> , tetapi ingin membatalkannya, klik tombol <i>Cancel</i>	
	7a. Sistem menampilkan kembali halaman Dashboard

SKENARIO ALTERNATIF
Aktor menambah data *user* tetapi *input* data tidak valid

6b. Ketika tombol <i>Submit</i> diklik pada halaman Tambah <i>User</i> , tetapi <i>input</i> data tidak valid	
	7b. Sistem menampilkan <i>message</i> , " <i>Please fill out this field</i> "

SKENARIO NORMAL
Aktor mengubah data *user*

10. Jika admin ingin mengubah data <i>user</i> , klik tombol <i>Edit</i> pada kolom <i>Action</i> di tabel Daftar <i>User Aktif</i>	
	11. Sistem menampilkan halaman <i>Edit User</i> yang berisi: <ul style="list-style-type: none"> - <i>Textfield</i> Nama - <i>Textfield</i> Alamat - <i>Textfield</i> Telepon - <i>Textfield</i> Username - <i>Textfield</i> Password - Pilihan <i>Dropdown Role</i> - Tombol <i>Submit</i> dan tombol <i>Cancel</i> Yang telah terisi berdasarkan ID <i>User</i> yang akan diedit
12. Melakukan <i>edit</i> data	
13. Klik tombol <i>Submit</i>	

	14. Melakukan validasi data
	15. Apabila data yang diisi sudah valid, sistem menampilkan data <i>user</i> yang telah di- <i>update</i> dan menampilkan kembali halaman Dashboard
	16. Sistem melakukan <i>update</i> data di <i>database</i>
SKENARIO ALTERNATIF Aktor mengubah data user	
13a. Ketika ingin membatalkan proses <i>edit</i> , klik tombol <i>Cancel</i>	
	14a. Sistem menampilkan kembali halaman Dashboard
SKENARIO NORMAL Aktor menghapus data user	
17. Jika admin ingin menghapus data <i>user</i> , klik tombol <i>Remove</i> pada kolom <i>Action</i> di tabel Daftar <i>User Aktif</i> berdasarkan Nama <i>User</i> yang akan dihapus	
	18. Sistem menampilkan <i>message dialog</i> "Are You Sure You Want to Remove This Data?"
19. Klik Tombol OK	
	20. Sistem menghapus data <i>user</i> yang dipilih kemudian menampilkan halaman Dashboard
SKENARIO ALTERNATIF Aktor menghapus data user	
19a. Apabila admin ingin membatalkan proses hapus data setelah mengklik tombol <i>Remove</i> pada kolom <i>Action</i> di tabel Daftar <i>User Aktif</i> , klik tombol <i>Cancel</i> pada <i>message dialog</i> konfirmasi penghapusan data	
	20a. Sistem menampilkan kembali halaman <i>Dashboard</i>

A.1.2 Use case Skenario Manajemen Data Pegawai

Tabel A.2 Use Case Skenario Manajemen Data Pegawai

ID Use Case	UC-03
Nama	Manajemen Data Pegawai
Aktor	Administrator
Pre-Condition	Menampilkan Data Pegawai
Post-Condition	Melihat daftar informasi dan merubah isi informasi data Pegawai
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Klik menu Pegawai	2. Menampilkan halaman Pegawai yang berisi tombol Tambah Pegawai, <i>textfield Search</i> , menu <i>dropdown Show entries</i> tabel, dan tabel Data Pegawai. Tabel Data Pegawai berisi tentang ID Pegawai, Nama Pegawai, Alamat, Tempat Tanggal Lahir, Usia, Jenis Kelamin, Jabatan, Telepon, dan <i>Action</i> . Kolom <i>Action</i> berisi tombol <i>Edit</i> dan tombol <i>Remove</i>
SKENARIO NORMAL	
Aktor menambah data pegawai	
3. Jika admin ingin menambah data pegawai, maka klik Tombol Tambah Pegawai	4. Sistem menampilkan halaman Tambah Pegawai yang berisi: <ul style="list-style-type: none"> - ID Pegawai yang telah terisi otomatis - <i>Textfield</i> Nama Pegawai, - <i>Textfield</i> Alamat, - <i>Textfield</i> Tempat Lahir, - <i>Textfield</i> Tanggal Lahir, - Pilihan radio button Jenis Kelamin, - <i>Textfield</i> Jabatan, - <i>Textfield</i> Telepon, - Tombol Back, tombol submit, dan tombol reset.
5. Memasukkan data pegawai	
6. Klik tombol <i>Submit</i>	

	7. Melakukan validasi data
	8. Apabila data yang diisi sudah valid, sistem menampilkan data pegawai yang telah ditambah kedalam tabel Data Pegawai dan menampilkan kembali halaman Pegawai
	9. Sistem menyimpan data di <i>database</i>
SKENARIO ALTERNATIF	
Aktor menambah data pegawai tetapi ingin mengosongkan semua <i>textfield</i>	
5a. Ketika memasukkan data pegawai pada halaman Tambah Pegawai, tetapi ingin mengosongkan semua <i>textfield</i> , klik tombol <i>Reset</i> untuk mengosongkan data yang telah diisi	
	6a. Sistem mengosongkan data yang telah dimasukkan
SKENARIO ALTERNATIF	
Aktor menambah data pegawai tetapi ingin membatalkannya	
5b. Ketika memasukkan data pegawai pada halaman Tambah Pegawai, tetapi ingin membatalkannya, klik tombol <i>Back</i>	
	6b. Sistem menampilkan kembali halaman Pegawai
SKENARIO ALTERNATIF	
Aktor menambah data pegawai tetapi <i>input</i> data tidak valid	
5c. Ketika tombol <i>Submit</i> diklik pada halaman Tambah Pegawai, tetapi <i>input</i> data tidak valid	
	6c. Sistem menampilkan <i>message</i> "Please fill out this field"
SKENARIO NORMAL	
Aktor mengubah data pegawai	
10. Jika admin ingin mengubah data pegawai, klik tombol <i>Edit</i> pada kolom <i>Action</i> di tabel Data Pegawai	
	11. Sistem menampilkan halaman <i>Edit</i> Pegawai yang berisi: <ul style="list-style-type: none"> - ID Pegawai yang telah terisi dan tak bisa diedit, - <i>Textfield</i> Nama Pegawai, - <i>Textfield</i> Alamat,

	<ul style="list-style-type: none"> - <i>Textfield</i> Tempat Lahir, - <i>Textfield</i> Tanggal Lahir, - Pilihan radio button Jenis Kelamin, - <i>Textfield</i> Jabatan, - <i>Textfield</i> Telepon - Tombol submit dan tombol cancel. <p>Semua <i>textfield</i> telah terisi berdasarkan ID Pegawai yang akan diedit</p>
--	--

12. Melakukan <i>edit</i> data	
--------------------------------	--

13. Klik tombol <i>Submit</i>	
-------------------------------	--

	14. Melakukan validasi data.
--	------------------------------

	15. Apabila data yang diisi sudah valid, sistem menampilkan data pegawai yang telah <i>diupdate</i> dan menampilkan kembali halaman Pegawai
--	---

	16. Sistem melakukan <i>update</i> data di <i>database</i>
--	--

SKENARIO ALTERNATIF
Aktor mengubah data pegawai tetapi ingin membatalkan proses mengubah data

13a. Ketika ingin membatalkan proses mengubah data, klik tombol <i>Cancel</i>	
---	--

	14a. Sistem menampilkan kembali halaman Pegawai
--	---

SKENARIO ALTERNATIF
Aktor mengubah data pegawai tetapi *input* data tidak valid

13b. Ketika tombol <i>Submit</i> diklik pada halaman <i>Edit</i> Pegawai, tetapi <i>input</i> data tidak valid.	
---	--

	14b. Sistem menampilkan <i>message</i> “ <i>Please fill out this field</i> ”
--	--

SKENARIO NORMAL
Aktor menghapus data pegawai

17. Jika admin ingin menghapus data pegawai, klik tombol <i>Remove</i> pada kolom <i>Action</i> di tabel Data Pegawai berdasarkan ID Pegawai yang akan dihapus	
--	--

	18. Sistem menampilkan <i>message dialog</i> “ <i>Are You Sure You Want to Remove This Data?</i> ”
--	--

19. Klik tombol OK	20. Sistem menghapus data yang dipilih kemudian menampilkan halaman Pegawai
SKENARIO ALTERNATIF	
Aktor menghapus data pegawai tetapi ingin membatalkan proses hapus data setelah mengklik tombol <i>Remove</i>	
19a. Apabila admin ingin membatalkan proses hapus data setelah mengklik tombol <i>Remove</i> pada kolom <i>Action</i> di tabel Data Pegawai, klik tombol <i>Cancel</i> pada <i>message dialog</i> konfirmasi penghapusan data	20a. Sistem menampilkan kembali halaman Pegawai
SKENARIO NORMAL	
Aktor mencari data pegawai	
21. Jika admin ingin mencari data pegawai yang ada di tabel Data Pegawai, ketikkan data yang akan di cari pada <i>textfield Search</i>	22. Sistem mencari data pegawai sesuai dengan data yang diketikkan pada <i>textfield Search</i>
	23. Sistem menampilkan data yang dicari pada tabel Data Pegawai
SKENARIO ALTERNATIF	
Aktor mencari data pegawai tetapi data yang dicari tidak ditemukan	
21a. Jika admin ingin mencari data pegawai yang ada di tabel Data Pegawai, tetapi data yang dicari tidak ditemukan	22a. Sistem menampilkan <i>message "No matching records found"</i>
SKENARIO NORMAL	
Aktor ingin menampilkan isi tabel Data Pegawai dengan jumlah tertentu	
24. Jika admin ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah <i>entries</i> data pada menu <i>dropdown Show entries</i>	25. Sistem menampilkan isi tabel berdasarkan jumlah <i>entries</i> yang

dipilih

A.1.3 Use case Skenario Manajemen Data Pasien

Tabel A.3 Use case Skenario Manajemen Data Pasien

ID Use Case	UC-04
Nama	Manajemen Data Pasien
Aktor	Administrator
Pre-Condition	Menampilkan Data Pasien
Post-Condition	Melihat daftar informasi dan merubah isi informasi data pasien
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Klik menu Pasien	2. Menampilkan halaman Pasien yang berisi tombol Tambah Pasien, <i>textfield Search</i> , menu <i>dropdown Show entries</i> tabel, dan tabel Data Pasien. Tabel Data Pasien berisi tentang ID Pasien, Pasien, Umur, Jenis Kelamin, Golongan Darah, Tanggal Daftar, Penanggung Jawab, Dokter, dan <i>Action</i> . Kolom Pasien berisi tentang Nama Pasien, Alamat, Pekerjaan, dan Jenis Pasien. Kolom Penanggung Jawab berisi tentang Nama, Alamat, Telepon, Pekerjaan, dan Hubungan dengan Pasien. Sedangkan Kolom <i>Action</i> berisi tombol <i>Edit</i> dan tombol <i>Remove</i>
SKENARIO NORMAL	
Aktor menambah data pasien	
3. Jika admin ingin menambah data pasien, maka klik tombol Tambah Pasien	4. Sistem menampilkan halaman Tambah Pasien yang berisi: <ul style="list-style-type: none"> - ID Pasien yang telah terisi otomatis - <i>Textfield</i> nama, - <i>Textfield</i> telepon, - <i>Textfield</i> usia,

- *Textfield* pekerjaan,
 - *Textfield* alamat,
 - Pilihan radio button jenis kelamin,
 - Pilihan dropdown golongan darah,
 - Pilihan dropdown jenis pasien,
- Dan penanggungjawab pasien yang berisi:
- *Textfield* nama penanggungjawab,
 - *Textfield* alamat penanggungjawab,
 - *Textfield* telepon penanggungjawab,
 - *Textfield* pekerjaan penanggungjawab,
 - *Textfield* hubungan dengan pasien,
 - pilihan dropdown dokter,
 - tombol Back, tombol submit, dan tombol reset.

5. Memasukkan data pasien	
6. Klik tombol <i>Submit</i>	
	7. Melakukan validasi data
	8. Apabila data yang diisi sudah valid, sistem menampilkan data pasien yang telah ditambah kedalam tabel Data Pasien dan menampilkan kembali halaman Pasien
	9. Sistem menyimpan data di <i>database</i>

SKENARIO ALTERNATIF

Aktor menambah data pasien tetapi ingin mengosongkan semua *textfield*

6a. Ketika memasukkan data pasien pada halaman Tambah Pasien, tetapi ingin mengosongkan semua <i>textfield</i> , klik tombol <i>Reset</i> untuk mengosongkan data yang telah diisi	
	7a. Sistem mengosongkan data yang telah dimasukkan

SKENARIO ALTERNATIF

Aktor menambah data pasien tetapi ingin membatalkannya

6b. Ketika memasukkan data pasien pada halaman Tambah Pasien, tetapi ingin membatalkannya, klik tombol <i>Back</i>	
--	--

7b. Sistem menampilkan kembali halaman Pasien

SKENARIO ALTERNATIF

Aktor menambah data pasien tetapi *input* data tidak valid

6c. Ketika tombol *Submit* diklik pada halaman Tambah Pasien, tetapi *input* data tidak valid

7c. Sistem menampilkan *message* "Please fill out this field"

SKENARIO NORMAL

Aktor mengubah data pasien

10. Jika admin ingin mengubah data pasien, klik tombol *Edit* pada kolom *Action* di tabel Data Pasien

11. Sistem menampilkan halaman *Edit* Pasien yang berisi:

- ID Pasien yang telah terisi dan tak bisa diedit,
- *Textfield* nama,
- *Textfield* telepon,
- *Textfield* usia,
- *Textfield* pekerjaan,
- *Textfield* alamat,
- Pilihan radio button jenis kelamin,
- Pilihan dropdown golongan darah,
- Pilihan dropdown jenis pasien,

Dan penanggungjawab pasien ang berisi:

- *Textfield* nama penanggungjawab,
- *Textfield* alamat penanggungjawab,
- *Textfield* telepon penanggungjawab,
- *Textfield* pekerjaan penanggungjawab,
- *Textfield* hubungan dengan pasien,

Serta pilihan dropdown dokter, dan juga tombol subbmit dan tombol cancel. Isi data pasien tersebut berdasarkan ID pasien yang akan diedit.

12. Melakukan *edit* data

13. Klik tombol *Submit*

	14. Melakukan validasi data
	15. Apabila data yang diisi sudah valid, sistem menampilkan data pasien yang telah <i>diupdate</i> dan menampilkan kembali halaman Pasien
	16. Sistem melakukan <i>update</i> data di <i>database</i>
SKENARIO ALTERNATIF	
Aktor mengubah data pasien tetapi tidak jadi mengubah data	
13a. Ketika ingin membatalkan proses <i>edit</i> , klik tombol <i>Cancel</i>	
	14a. Sistem menampilkan kembali halaman Pasien
SKENARIO ALTERNATIF	
Aktor mengubah data pasien tetapi tetapi <i>input</i> data tidak valid	
13b. Ketika tombol <i>Submit</i> diklik pada halaman <i>Edit</i> Pasien, tetapi <i>input</i> data tidak valid.	
	14b. Sistem menampilkan <i>message</i> " <i>Please fill out this field</i> "
SKENARIO NORMAL	
Aktor menghapus data pasien	
17. Jika admin ingin menghapus data pasien, klik tombol <i>Remove</i> pada kolom <i>Action</i> di tabel Data Pasien berdasarkan ID Pasien yang akan dihapus	
	18. Sistem menampilkan <i>message dialog</i> " <i>Are You Sure You Want to Remove This Data?</i> "
19. Klik tombol OK	
	20. Sistem menghapus data yang dipilih kemudian menampilkan halaman Pasien
SKENARIO ALTERNATIF	
Aktor menghapus data pasien tetapi ingin membatalkan proses hapus data setelah mengklik tombol <i>Remove</i>	
19a. Apabila admin ingin membatalkan proses hapus data setelah mengklik tombol <i>Remove</i> pada kolom <i>Action</i> di tabel Data Pasien, klik tombol	

<i>Cancel</i> pada <i>message dialog</i> konfirmasi penghapusan data	
	20a. Sistem menampilkan kembali halaman Pasien
SKENARIO NORMAL Aktor mencari data pasien	
21. Jika admin ingin mencari data pasien yang ada di tabel Data Pasien, ketikkan data yang akan dicari pada <i>textfield Search</i>	
	22. Sistem mencari data pasien sesuai dengan data yang diketikkan pada <i>textfield Search</i>
	23. Sistem menampilkan data tersebut pada tabel Data Pasien
SKENARIO ALTERNATIF Aktor mencari data pasien tetapi data yang dicari tidak ditemukan	
21a. Jika admin ingin mencari data pasien yang ada di tabel Data Pasien, tetapi data yang dicari tidak ditemukan	
	22a. Sistem menampilkan <i>message "No matching records found"</i>
SKENARIO NORMAL Aktor ingin menampilkan isi tabel Data Pasien dengan jumlah tertentu	
24. Jika admin ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah <i>entries</i> data pada menu dropdown <i>Show entries</i>	
	25. Sistem menampilkan isi tabel berdasarkan jumlah <i>entries</i> yang dipilih

A.1.4 Use case Skenario Manajemen Data Kamar

Tabel A. 4 Use case Skenario Manajemen Data Kamar

ID Use Case	UC-05
Nama	Manajemen Data Kamar
Aktor	Administrator
Pre-Condition	Menampilkan Data Kamar

Post-Condition	Melihat daftar informasi dan merubah isi informasi data kamar
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Klik menu Kamar	2. Menampilkan halaman Kamar yang berisi keterangan tentang Jumlah Kamar terisi, Jumlah Kamar kosong, Jumlah Kamar Kelas 1, Jumlah Kamar Kelas 2, Jumlah Kamar Kelas 3, total jumlah kamar, tombol Tambah Kamar, <i>textfield Search</i> , menu <i>dropdown Show entries</i> tabel, dan tabel Data Kamar. Tabel Data Kamar berisi tentang ID Kamar, Nama Kamar, Kelas, Fasilitas, Biaya Kamar, Status Kamar, dan <i>Action</i> . Kolom <i>Action</i> berisi tombol <i>Edit</i> dan tombol <i>Remove</i>
SKENARIO NORMAL	
Aktor menambah data kamar	
3. Jika admin ingin menambah data kamar, maka klik tombol Tambah Kamar	4. Sistem menampilkan halaman Tambah Kamar yang berisi: <ul style="list-style-type: none"> - ID Kamar yang telah terisi otomatis - <i>Textfield</i> Nama Kamar, - Pilihan <i>dropdown</i> Kelas, - Pilihan <i>dropdown</i> Fasilitas, - Tombol <i>add more data</i> untuk fasilitas - <i>Textfield</i> Biaya Kamar - Tombol Back, Tombol submit, dan Tombol reset
5. Memasukkan data kamar	7. Melakukan validasi data
6. Klik tombol <i>Submit</i>	8. Apabila data yang diisi sudah valid, sistem menampilkan data kamar yang telah ditambah kedalam tabel Data

	Kamar dan menampilkan kembali halaman Kamar
	9. Sistem menyimpan data di <i>database</i>
SKENARIO ALTERNATIF	
Aktor menambah data kamar tetapi ingin mengosongkan semua <i>textfield</i>	
5a. Klik tombol <i>Reset</i> untuk mengosongkan data yang telah diisi	
	6a. Sistem mengosongkan data yang telah dimasukkan
SKENARIO ALTERNATIF	
Aktor menambah data kamar tetapi ingin membatalkannya	
5b. Ketika memasukkan data kamar pada halaman Tambah Kamar, tetapi ingin membatalkannya, klik tombol <i>Back</i>	
	6b. Sistem menampilkan kembali halaman Kamar
SKENARIO ALTERNATIF	
Aktor menambah data kamar tetapi <i>input</i> data tidak valid	
7a. Ketika tombol <i>Submit</i> diklik pada halaman Tambah Kamar, tetapi <i>input</i> data tidak valid	
	8a. Sistem menampilkan <i>message</i> "Please fill out this field"
SKENARIO NORMAL	
Aktor mengubah data kamar	
10. Jika admin ingin mengubah data kamar, klik tombol <i>Edit</i> pada kolom <i>Action</i> di tabel Data Kamar	
	11. Sistem menampilkan halaman <i>Edit</i> Kamar yang berisi: <ul style="list-style-type: none"> - ID Kamar yang telah terisi dan tak bisa diedit, - <i>Textfield</i> Nama Kamar, - Pilihan dropdown Kelas, - Pilihan dropdown Fasilitas, - Tombol <i>add more data</i> untuk fasilitas - <i>Textfield</i> Biaya Kamar, - Tombol Back, Tombol submit, dan Tombol reset.

data Kamar yang diedit berdasarkan ID kamar yang akan diedit.

12. Melakukan *edit* data

13. Klik tombol *Submit*

14. Melakukan validasi data

15. Apabila data yang diisi sudah valid, sistem menampilkan data kamar yang telah *diupdate* dan menampilkan kembali halaman Kamar

16. Sistem melakukan *update* data di *database*

SKENARIO ALTERNATIF

Aktor mengubah data kamar tetapi ingin membatalkan proses *edit*

13a. Ketika ingin membatalkan proses *edit*, klik tombol *Cancel*

14a. Sistem menampilkan kembali halaman Kamar

SKENARIO ALTERNATIF

Aktor mengubah data kamar tetapi *input* data tidak valid

12a. Ketika tombol *Submit* diklik pada halaman *Edit* Kamar, tetapi *input* data tidak valid.

13a. Sistem menampilkan *message* "Please fill out this field"

SKENARIO ALTERNATIF

Aktor mengubah data kamar tetapi ingin membatalkan proses *edit*

13b. Ketika ingin membatalkan proses *edit*, klik tombol *Cancel*

14b. Sistem menampilkan kembali halaman Kamar

SKENARIO NORMAL

Aktor menghapus data kamar

17. Jika admin ingin menghapus data kamar, klik tombol *Remove* pada kolom *Action* di tabel Data Kamar berdasarkan ID Kamar yang akan dihapus

18. Sistem menampilkan *message dialog* "Are You Sure You Want to Remove"

<i>This Data?"</i>	
19. Klik tombol OK	20. Sistem menghapus data yang dipilih kemudian menampilkan halaman Kamar
SKENARIO ALTERNATIF Aktor menghapus data kamar tetapi ingin membatalkan proses hapus data setelah mengklik tombol <i>Remove</i>	
19a. Apabila admin ingin membatalkan proses hapus data setelah mengklik tombol <i>Remove</i> pada kolom <i>Action</i> di tabel Data Kamar, klik tombol <i>Cancel</i> pada <i>message dialog</i> konfirmasi penghapusan data	20a. Sistem menampilkan kembali halaman Kamar
SKENARIO NORMAL Aktor mencari data kamar	
21. Jika admin ingin mencari data Kamar yang ada di tabel Data Kamar, ketikkan data yang akan di cari pada <i>textfield Search</i>	22. Sistem mencari data kamar sesuai dengan data yang diketikkan pada <i>textfield Search</i>
	23. Sistem menampilkan data tersebut pada tabel Data Kamar
SKENARIO ALTERNATIF Aktor mencari data kamar tetapi data yang dicari tidak ditemukan	
21a. Jika admin ingin mencari data kamar yang ada di tabel Data Kamar, tetapi data yang dicari tidak ditemukan	22a. Sistem menampilkan <i>message "No matching records found"</i>
SKENARIO NORMAL Aktor ingin menampilkan isi tabel Data Kamar dengan jumlah tertentu	
24. Jika admin ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah <i>entries</i> data pada menu <i>dropdown Show entries</i>	25. Sistem menampilkan isi tabel

berdasarkan jumlah *entries* yang dipilih

A.1.5 Use case Skenario Manajemen Data Fasilitas

Tabel A.5 Use case Skenario Manajemen Data Fasilitas

ID Use Case	UC-06
Nama	Manajemen Data Fasilitas
Aktor	Administrator
Pre-Condition	Menampilkan Data Fasilitas
Post-Condition	Melihat daftar informasi dan merubah isi informasi data Fasilitas
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Klik menu Fasilitas	2. Menampilkan halaman Fasilitas yang berisi tombol Tambah Fasilitas, <i>textfield Search</i> , menu <i>dropdown Show entries</i> tabel, dan tabel Data Fasilitas. Tabel Data Fasilitas berisi tentang ID Fasilitas, Nama Fasilitas, Jumlah, dan <i>Action</i> . Kolom <i>Action</i> berisi tombol <i>Edit</i> dan tombol <i>Remove</i>
SKENARIO NORMAL	
Aktor menambah data fasilitas	
3. Jika admin ingin menambah data fasilitas, maka klik tombol Tambah Fasilitas	4. Sistem menampilkan halaman tambah Fasilitas: <ul style="list-style-type: none"> - ID fasilitas yang telah terisi otomatis, - <i>Textfield</i> Nama Fasilitas, - <i>Textfield</i> Jumlah, - Tombol Back, Tombol submit, dan Tombol reset
5. Memasukkan data fasilitas	
6. Klik tombol <i>Submit</i>	7. Melakukan validasi data
	8. Apabila data yang diisi sudah valid, sistem menampilkan data fasilitas

yang telah ditambah kedalam tabel Data Fasilitas dan menampilkan kembali halaman Fasilitas

9. Sistem menyimpan data di *database*

SKENARIO ALTERNATIF

Aktor memasukkan data fasilitas tetapi ingin mengosongkan semua *textfield*

5a. Ketika memasukkan data fasilitas pada halaman Tambah Fasilitas, tetapi ingin mengosongkan semua *textfield*, klik tombol *Reset* untuk mengosongkan data yang telah diisi

6a. Sistem mengosongkan data yang telah dimasukkan

SKENARIO ALTERNATIF

Aktor memasukkan data fasilitas tetapi ingin membatalkannya

6a. Ketika memasukkan data fasilitas pada halaman Tambah Fasilitas, tetapi ingin membatalkannya, klik tombol *Back*

7a. Sistem menampilkan kembali halaman Fasilitas

SKENARIO ALTERNATIF

Aktor memasukkan data fasilitas tetapi *input* data tidak valid

6b. Ketika tombol *Submit* diklik pada halaman Tambah Fasilitas, tetapi *input* data tidak valid

7b. Sistem menampilkan *message* "Please fill out this field"

SKENARIO NORMAL

Aktor mengubah data fasilitas

10. Jika admin ingin mengubah data fasilitas, klik tombol *Edit* pada kolom *Action* di tabel Data Fasilitas

11. Sistem menampilkan halaman *Edit* Fasilitas yang berisi:

- ID fasilitas yang telah terisi dan tak bisa diedit,
- *Textfield* Nama Fasilitas,
- *Textfield* Jumlah,
- Tombol submit dan Tombol cancel

Data Fasilitas yang diedit berdasarkan ID Fasilitas yang akan diedit.

12. Melakukan <i>edit</i> data	
13. Klik tombol <i>Submit</i>	
	14. Melakukan validasi data
	15. Apabila data yang diisi sudah valid, sistem menampilkan data fasilitas yang telah <i>diupdate</i> dan menampilkan kembali halaman Fasilitas
	16. Sistem melakukan <i>update</i> data di <i>database</i>

SKENARIO ALTERNATIF
Aktor mengubah data fasilitas tetapi ingin membatalkan

13a. Ketika ingin membatalkan proses <i>edit</i> , klik tombol <i>Cancel</i>	
	14a. Sistem menampilkan kembali halaman Fasilitas

SKENARIO ALTERNATIF
Aktor mengubah data fasilitas tetapi *input* data tidak valid

13b. Ketika tombol <i>Submit</i> diklik pada halaman <i>Edit</i> Fasilitas, tetapi <i>input</i> data tidak valid.	
	14b. Sistem menampilkan <i>message</i> “ <i>Please fill out this field</i> ”

SKENARIO NORMAL
Aktor menghapus data fasilitas

17. Jika admin ingin menghapus data fasilitas, klik tombol <i>Remove</i> pada kolom <i>Action</i> di tabel Data Fasilitas berdasarkan ID Fasilitas yang akan dihapus	
	18. Sistem menampilkan <i>message dialog</i> “ <i>Are You Sure You Want to Remove This Data?</i> ”
19. Klik tombol OK	
	20. Sistem menghapus data yang dipilih kemudian menampilkan halaman Fasilitas

SKENARIO ALTERNATIF
Aktor menghapus data fasilitas tetapi ingin membatalkan proses hapus data setelah mengklik tombol *Remove*

19a. Apabila admin ingin membatalkan proses hapus data setelah mengklik tombol *Remove* pada kolom *Action* di tabel Data Fasilitas, klik tombol *Cancel* pada *message dialog* konfirmasi penghapusan data

20a. Sistem menampilkan kembali halaman Fasilitas

SKENARIO NORMAL
Aktor mencari data fasilitas

21. Jika admin ingin mencari data fasilitas yang ada di tabel Data Fasilitas, ketikkan data yang akan di cari pada *textfield Search*

22. Sistem mencari data fasilitas sesuai dengan data yang diketikkan pada *textfield Search*

23. Sistem menampilkan data tersebut pada tabel Data Fasilitas

SKENARIO ALTERNATIF
Aktor mencari data fasilitas tetapi data yang dicari tidak ditemukan

21a. Jika admin ingin mencari data fasilitas yang ada di tabel Data Fasilitas, tetapi data yang dicari tidak ditemukan

22a. Sistem menampilkan *message* "No matching records found"

SKENARIO NORMAL
Aktor ingin menampilkan isi tabel Data Fasilitas dengan jumlah tertentu

24. Jika admin ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah *entries* data pada menu *dropdown Show entries*

25. Sistem menampilkan isi tabel berdasarkan jumlah *entries* yang dipilih

A.1.6 Use case Skenario Manajemen Data Dokter

Tabel A.6 Use case Skenario Manajemen Data Dokter

ID Use Case	UC-07
--------------------	-------

Nama	Manajemen Data Dokter
Aktor	Administrator
Pre-Condition	Menampilkan Data Dokter
Post-Condition	Melihat daftar informasi dan merubah isi informasi data dokter

Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	

<p>1. Klik menu Dokter</p>	<p>2. Menampilkan halaman Dokter yang berisi tombol Tambah Dokter, <i>textfield Search</i>, menu <i>dropdown Show entries</i> tabel, dan tabel Data Dokter. Tabel Data Dokter berisi tentang ID Dokter, Nama Dokter, Tempat Tanggal Lahir, Usia, Alamat, Jenis Kelamin, Telepon, Spesialis, dan <i>Action</i>. Kolom <i>Action</i> berisi tombol <i>Edit</i> dan tombol <i>Remove</i></p>
----------------------------	---

SKENARIO NORMAL	
Aktor menambah data dokter	

<p>3. Jika admin ingin menambah data dokter, maka klik tombol Tambah Dokter</p>	<p>4. Sistem menampilkan halaman Tambah Dokter yang berisi:</p> <ul style="list-style-type: none"> - ID Dokter yang telah terisi otomatis, - <i>Textfield</i> Nama Dokter, - <i>Textfield</i> Tempat Lahir, - <i>Textfield</i> Tanggal Lahir, - <i>Textfield</i> Usia yang terisi otomatis setelah mengisi tanggal lahir, - <i>Textfield</i> Alamat, - Pilihan radio button Jenis Kelamin, - <i>Textfield</i> Telepon, - Pilihan <i>dropdown</i> Spesialis, - Tombol Back, Tombol Submit, dan Tombol Reset.
---	---

<p>5. Memasukkan data dokter</p>	<p>7. Melakukan validasi data</p>
<p>6. Klik tombol <i>Submit</i></p>	<p>8. Apabila data yang diisi sudah valid,</p>

	sistem menampilkan data dokter yang telah ditambah kedalam tabel Data Dokter dan menampilkan kembali halaman Dokter
	9. Sistem menyimpan data di <i>database</i>
SKENARIO ALTERNATIF	
Aktor menambah data dokter tetapi ingin mengosongkan semua <i>textfield</i>	
6a. Ketika memasukkan data dokter pada halaman Tambah Dokter, tetapi ingin mengosongkan semua <i>textfield</i> , klik tombol Reset	
	7a. Sistem mengosongkan data yang telah dimasukkan
SKENARIO ALTERNATIF	
Aktor menambah data dokter tetapi ingin membatalkannya	
6b. Ketika memasukkan data dokter pada halaman Tambah Dokter, tetapi ingin membatalkannya, klik tombol <i>Back</i>	
	7b. Sistem menampilkan kembali halaman Dokter
SKENARIO ALTERNATIF	
Aktor menambah data dokter tetapi <i>input</i> data tidak valid	
6c. Ketika tombol <i>Submit</i> diklik pada halaman Tambah Dokter, tetapi <i>input</i> data tidak valid	
	7c. Sistem menampilkan <i>message</i> "Please fill out this field"
SKENARIO NORMAL	
Aktor mengubah data dokter	
10. Jika admin ingin mengubah data dokter, klik tombol <i>Edit</i> pada kolom <i>Action</i> di tabel Data Dokter	
	11. Sistem menampilkan halaman <i>Edit</i> Dokter yang berisi: <ul style="list-style-type: none"> - ID Dokter yang telah terisi dan tak bisa diedit, - <i>Textfield</i> Nama Dokter, - <i>Textfield</i> Tempat Lahir, - <i>Textfield</i> Tanggal Lahir, - <i>Textfield</i> Usia yang terisi otomatis setelah mengisi tanggal lahir,

	<ul style="list-style-type: none"> - <i>Textfield</i> Alamat, - Pilihan radio button Jenis Kelamin, - <i>Textfield</i> Telepon, - Pilihan <i>dropdown</i> Spesialis, - Tombol Submit dan Tombol Cancel. <p>Data dokter yang diedit terisi berdasarkan ID dokter yang akan diedit.</p>
12. Melakukan <i>edit</i> data	
13. Klik tombol <i>Submit</i>	
	14. Melakukan validasi data
	15. Apabila data yang diisi sudah valid, sistem menampilkan data dokter yang telah <i>diupdate</i> dan menampilkan kembali halaman Dokter
	16. Sistem melakukan <i>update</i> data di <i>database</i>
SKENARIO ALTERNATIF	
Aktor mengubah data dokter ingin membatalkan proses <i>edit</i>	
13a. Ketika ingin membatalkan proses <i>edit</i> , klik tombol <i>Cancel</i>	
	14a. Sistem menampilkan kembali halaman Dokter
SKENARIO ALTERNATIF	
Aktor mengubah data dokter tetapi <i>input</i> data tidak valid	
13b. Ketika tombol <i>Submit</i> diklik pada halaman <i>Edit</i> Dokter, tetapi <i>input</i> data tidak valid.	
	14b. Sistem menampilkan <i>message</i> " <i>Please fill out this field</i> "
SKENARIO NORMAL	
Aktor menghapus data dokter	
17. Jika admin ingin menghapus data dokter, klik tombol <i>Remove</i> pada kolom <i>Action</i> di tabel Data Dokter berdasarkan ID Dokter yang akan dihapus	
	18. Sistem menampilkan <i>message dialog</i> " <i>Are You Sure You Want to Remove This Data?</i> "

19. Klik tombol OK	20. Sistem menghapus data yang dipilih kemudian menampilkan halaman Dokter
SKENARIO ALTERNATIF	
Aktor menghapus data dokter tetapi ingin membatalkan proses hapus data setelah mengklik tombol <i>Remove</i>	
19a. Apabila admin ingin membatalkan proses hapus data setelah mengklik tombol <i>Remove</i> pada kolom <i>Action</i> di tabel Data Dokter, klik tombol <i>Cancel</i> pada <i>message dialog</i> konfirmasi penghapusan data	20a. Sistem menampilkan kembali halaman Dokter
SKENARIO NORMAL	
Aktor mencari data dokter	
21. Jika admin ingin mencari data dokter yang ada di tabel Data Dokter, ketikkan data yang akan di cari pada <i>textfield Search</i>	22. Sistem mencari data dokter sesuai dengan data yang diketikkan pada <i>textfield Search</i>
	23. Sistem menampilkan data tersebut pada tabel Data Dokter
SKENARIO ALTERNATIF	
Aktor mencari data dokter tetapi data yang dicari tidak ditemukan	
21a. Jika admin ingin mencari data dokter yang ada di tabel Data Dokter, tetapi data yang dicari tidak ditemukan	22a. Sistem menampilkan <i>message "No matching records found"</i>
SKENARIO NORMAL	
Aktor ingin menampilkan isi tabel Data Dokter dengan jumlah tertentu	
24. Jika admin ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah <i>entries</i> data pada menu <i>dropdown Show entries</i>	25. Sistem menampilkan isi tabel berdasarkan jumlah <i>entries</i> yang

dipilih

A.1.7 Use case Skenario Manajemen Data Obat

Tabel A.7 Use case Skenario Manajemen Data Obat

ID Use Case	UC-08
Nama	Manajemen Data Obat
Aktor	Administrator dan Apoteker
Pre-Condition	Menampilkan Data Obat
Post-Condition	Melihat daftar informasi dan merubah isi informasi data obat
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Klik menu Obat	2. Menampilkan halaman Obat yang berisi tombol Tambah Obat, <i>textfield Search</i> , menu <i>dropdown Show entries</i> tabel, dan tabel Data Obat. Tabel Data Obat berisi tentang ID Obat, Nama Obat, Jenis Obat, Satuan, Stok, Harga Pokok, Harga Jual, dan <i>Action</i> . Kolom <i>Action</i> berisi tombol <i>Edit</i> dan tombol <i>Remove</i>
SKENARIO NORMAL Aktor menambah data obat	
3. Jika aktor ingin menambah data obat, maka klik tombol Tambah Obat	4. Sistem menampilkan halaman Tambah Obat yang berisi: <ul style="list-style-type: none"> - ID Obat yang telah terisi otomatis, - <i>Textfield</i> Nama Obat, - Pilihan <i>dropdown</i> Jenis Obat, - Pilihan <i>dropdown</i> Satuan, - <i>Textfield</i> Stok, - <i>Textfield</i> Harga Pokok, - <i>Textfield</i> Harga Jual, - Tombol Back, Tombol Submit, dan Tombol Reset.
5. Memasukkan data obat	
6. Klik tombol <i>Submit</i>	

	7. Melakukan validasi data
	8. Apabila data yang diisi sudah valid, sistem menampilkan data obat yang telah ditambah kedalam tabel Data Obat dan menampilkan kembali halaman Obat
	9. Sistem menyimpan data di <i>database</i>
SKENARIO ALTERNATIF	
Aktor menambah data obat tetapi ingin mengosongkan semua <i>textfield</i>	
6a. Ketika memasukkan data obat pada halaman Tambah Obat, tetapi ingin mengosongkan semua <i>textfield</i> , klik tombol <i>Reset</i> untuk mengosongkan data yang telah diisi	
	7a. Sistem mengosongkan data yang telah dimasukkan
SKENARIO ALTERNATIF	
Aktor menambah data obat tetapi ingin membatalkannya	
6b. Ketika memasukkan data obat pada halaman Tambah Obat, tetapi ingin membatalkannya, klik tombol <i>Back</i>	
	7b. Sistem menampilkan kembali halaman Obat
SKENARIO ALTERNATIF	
Aktor menambah data obat tetapi <i>input</i> data tidak valid	
6c. Ketika tombol <i>Submit</i> diklik pada halaman Tambah Obat, tetapi <i>input</i> data tidak valid	
	7c. Sistem menampilkan <i>message</i> "Please fill out this field"
SKENARIO NORMAL	
Aktor mengubah data obat	
10. Jika aktor ingin mengubah data obat, klik tombol <i>Edit</i> pada kolom <i>Action</i> di tabel Data Obat	
	11. Sistem menampilkan halaman <i>Edit</i> Obat yang berisi: <ul style="list-style-type: none"> - ID Obat yang telah terisi otomatis, - <i>Textfield</i> Nama Obat, - Pilihan <i>dropdown</i> Jenis Obat, - Pilihan <i>dropdown</i> Satuan, - <i>Textfield</i> Stok,

	<ul style="list-style-type: none"> - <i>Textfield</i> Harga Pokok, - <i>Textfield</i> Harga Jual, - Tombol <i>Submit</i> dan Tombol <i>Cancel</i>. <p>Data Obat yang diedit telah terisi berdasarkan ID Obat yang akan diedit.</p>
12. Melakukan <i>edit</i> data	
13. Klik tombol <i>Submit</i>	
	14. Melakukan validasi data
	15. Apabila data yang diisi sudah valid, sistem menampilkan data obat yang telah <i>diupdate</i> dan menampilkan kembali halaman Obat
	16. Sistem melakukan <i>update</i> data di <i>database</i>
SKENARIO ALTERNATIF	
Aktor mengubah data obat tetapi membatalkan proses <i>edit</i>	
13a. Ketika ingin membatalkan proses <i>edit</i> , klik tombol <i>Cancel</i>	
	14a. Sistem menampilkan kembali halaman Obat
SKENARIO ALTERNATIF	
Aktor mengubah data obat tetapi <i>input</i> data tidak valid	
13b. Ketika tombol <i>Submit</i> diklik pada halaman <i>Edit</i> Obat, tetapi <i>input</i> data tidak valid.	
	14b. Sistem menampilkan <i>message</i> " <i>Please fill out this field</i> "
SKENARIO NORMAL	
Aktor menghapus data obat	
21. Jika aktor ingin menghapus data obat, klik tombol <i>Remove</i> pada kolom <i>Action</i> di tabel Data Obat berdasarkan Kode Obat yang akan dihapus	
	22. Sistem menampilkan <i>message dialog</i> " <i>Are You Sure You Want to Remove This Data?</i> "
23. Klik tombol OK	
	24. Sistem menghapus data yang dipilih kemudian menampilkan halaman

Obat

SKENARIO ALTERNATIF

Aktor menghapus data obat tetapi ingin membatalkan proses hapus data setelah mengklik tombol *Remove*

23a. Apabila aktor ingin membatalkan proses hapus data setelah mengklik tombol *Remove* pada kolom *Action* di tabel Data Obat, klik tombol *Cancel* pada *message dialog* konfirmasi penghapusan data

24a. Sistem menampilkan kembali halaman Obat

SKENARIO NORMAL

Aktor mencari data obat

25. Jika aktor ingin mencari data obat yang ada di tabel Data Obat, ketikkan data yang akan di cari pada *textfield Search*

26. Sistem mencari data obat sesuai dengan data yang diketikkan pada *textfield Search*

27. Sistem menampilkan data tersebut pada tabel Data Obat

SKENARIO ALTERNATIF

Aktor mencari data obat tetapi data yang dicari tidak ditemukan

25a. Jika aktor ingin mencari data obat yang ada di tabel Data Obat, tetapi data yang dicari tidak ditemukan

26a. Sistem menampilkan *message* “*No matching records found*”

SKENARIO NORMAL

Aktor ingin menampilkan isi tabel Data Obat dengan jumlah tertentu

28. Jika aktor ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah *entries* data pada menu *dropdown Show entries*

29. Sistem menampilkan isi tabel berdasarkan jumlah *entries* yang dipilih

A.1.8 Use case Skenario Manajemen Data Laborat

Tabel A.8 Use case Skenario Manajemen Data Laborat

ID Use Case	UC-09
Nama	Manajemen Data Laborat
Aktor	Administrator dan Laborat
Pre-Condition	Menampilkan data laborat
Post-Condition	Melihat daftar informasi dan merubah isi informasi data laborat
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Klik menu Laborat	2. Menampilkan halaman data laborat yang berisi tombol tambah laborat, <i>textfield Search</i> , menu <i>dropdown Show entries</i> tabel, dan tabel daftar laborat. Tabel daftar laborat berisi tentang ID_Laborat, Nama, Jenis, Biaya, Keperluan, ID_Pasien, Anjuran Dari, Hasil Pemeriksaan, dan <i>Action</i> . Kolom <i>Action</i> berisi tombol <i>edit</i> , tombol <i>Remove</i> , dan tombol cetak
SKENARIO NORMAL Aktor menambah data laborat	
3. Jika aktor ingin menambah data laborat, maka klik tombol tambah laborat	4. Sistem menampilkan form tambah laborat
5. Memasukkan data laborat	
6. Klik tombol <i>Submit</i>	7. Melakukan validasi data
	8. Apabila data yang diisi sudah valid, sistem menampilkan data laborat yang telah ditambah kedalam tabel daftar laborat dan menampilkan kembali halaman data laborat
	9. Sistem menyimpan data di <i>database</i>
SKENARIO ALTERNATIF Aktor menambah data laborat tetapi ingin mengosongkan semua <i>textfield</i>	

6a. Ketika memasukkan data laborat pada form tambah laborat, tetapi ingin mengosongkan semua *textfield*, klik tombol *Reset* untuk mengosongkan data yang telah diisi

7a. Sistem mengosongkan data yang telah dimasukkan

SKENARIO ALTERNATIF

Aktor menambah data laborat tetapi ingin membatalkannya

6b. Ketika memasukkan data laborat pada form tambah laborat, tetapi ingin membatalkannya, klik tombol *Back*

7b. Sistem menampilkan kembali halaman data laborat

SKENARIO ALTERNATIF

Aktor menambah data laborat tetapi *input* data tidak valid

6c. Ketika tombol *Submit* diklik pada form tambah laborat, tetapi *input* data tidak valid

7c. Sistem menampilkan *message* “*Please fill out this field*”

SKENARIO NORMAL

Aktor menghapus data laborat

10. Jika aktor ingin menghapus data laborat, klik tombol *Remove* pada kolom *Action* di tabel daftar laborat berdasarkan *ID_Laborat* yang akan dihapus

11. Sistem menampilkan *message dialog* “*Are You Sure You Want to Remove This Data?*”

12. Klik tombol OK

13. Sistem menghapus data yang dipilih kemudian menampilkan halaman data laborat

SKENARIO ALTERNATIF

Aktor menghapus data laborat tetapi ingin membatalkan proses hapus data setelah mengklik tombol *Remove*

12a. Apabila aktor ingin membatalkan proses hapus data setelah mengklik tombol *Remove* pada kolom *Action*

di tabel daftar laborat, klik tombol *cancel* pada *message dialog* konfirmasi penghapusan data

13a. Sistem menampilkan kembali halaman data laborat

SKENARIO NORMAL
Aktor mencetak data laborat

14. Jika aktor ingin mencetak data laborat, klik tombol cetak pada kolom *Action* di tabel daftar laborat berdasarkan ID_Laborat yang akan dicetak

15. Sistem mengubah data laborat yang akan dicetak dalam bentuk *Portable Document Format* (PDF) dan menampilkan data tersebut di tab baru

SKENARIO NORMAL
Aktor mencari data laborat

16. Jika aktor ingin mencari data laborat yang ada di tabel daftar laborat, ketikkan data yang akan di cari pada kolom search

17. Sistem mencari data laborat sesuai dengan data yang diketikkan pada kolom search

18. Sistem menampilkan data tersebut pada tabel daftar laborat

SKENARIO ALTERNATIF
Aktor mencari data laborat tetapi data yang dicari tidak ditemukan

16a. Jika aktor ingin mencari data laborat yang ada di tabel daftar laborat, tetapi data yang dicari tidak ditemukan

17a. Sistem menampilkan *message* “*No matching records found*”

SKENARIO NORMAL
Aktor ingin menampilkan isi tabel Data Laborat dengan jumlah tertentu

19. Jika aktor ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah *entries* data pada menu dropdown *Show entries*

20. Sistem menampilkan isi tabel

berdasarkan jumlah *entries* yang dipilih

A.1.9 Use case Skenario Manajemen Data Rujukan

Tabel A.9 Use case Skenario Manajemen Data Rujukan

ID Use Case	UC-11
Nama	Manajemen Data Rujukan
Aktor	Administrator
Pre-Condition	Menampilkan Data Rujukan
Post-Condition	Melihat daftar informasi dan merubah isi informasi data rujukan
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Klik menu Rujukan	2. Menampilkan halaman Rujukan yang berisi tombol Tambah Rujukan, <i>textfield Search</i> , menu <i>dropdown Show entries</i> tabel, dan tabel Data Rujukan. Tabel Data Rujukan berisi tentang No. Transaksi, Tanggal Transaksi, Pasien, Dokter, Tujuan RS Rujukan, Keterangan, dan <i>Action</i> . Kolom <i>Action</i> berisi tombol <i>Edit</i> , tombol <i>Remove</i> , dan tombol Cetak
SKENARIO NORMAL Aktor menambah data rujukan	
3. Jika aktor ingin menambah data rujukan, maka klik tombol Tambah Rujukan	4. Sistem menampilkan halaman Tambah Rujukan
5. Memasukkan data rujukan	7. Melakukan validasi data
6. Klik tombol <i>Submit</i>	8. Apabila data yang diisi sudah valid, sistem menampilkan data rujukan yang telah ditambah kedalam tabel Data Rujukan dan menampilkan kembali halaman Rujukan
	9. Sistem menyimpan data di <i>database</i>

SKENARIO ALTERNATIF	
Aktor menambah data rujukan tetapi ingin mengosongkan semua <i>textfield</i>	
6a. Ketika memasukkan data rujukan pada halaman Tambah Rujukan, tetapi ingin mengosongkan semua <i>textfield</i> , klik tombol <i>Reset</i> untuk mengosongkan data yang telah diisi	7a. Sistem mengosongkan data yang telah dimasukkan
SKENARIO ALTERNATIF	
Aktor menambah data rujukan tetapi ingin membatalkannya	
6b. Ketika memasukkan data rujukan pada halaman Tambah Rujukan, tetapi ingin membatalkannya, klik tombol <i>Back</i>	7b. Sistem menampilkan kembali halaman Rujukan
SKENARIO ALTERNATIF	
Aktor menambah data rujukan tetapi <i>input</i> data tidak valid	
6c. Ketika tombol <i>Submit</i> diklik pada halaman Tambah Rujukan, tetapi <i>input</i> data tidak valid	7c. Sistem menampilkan <i>message</i> "Please fill out this field"
SKENARIO NORMAL	
Aktor mengubah data rujukan	
10. Jika aktor ingin mengubah data rujukan, klik tombol <i>Edit</i> pada kolom <i>Action</i> di tabel Data Rujukan	11. Sistem menampilkan halaman <i>Edit</i> Rujukan yang berisi data rujukan
12. Melakukan <i>edit</i> data	
13. Klik tombol <i>Submit</i>	14. Melakukan validasi data
	15. Apabila data yang diisi sudah valid, sistem menampilkan data rujukan yang telah diupdate dan menampilkan kembali halaman Rujukan
	16. Sistem melakukan <i>update</i> data di <i>database</i>
SKENARIO ALTERNATIF	
Aktor mengubah data rujukan tetapi ingin membatalkannya	

13a. Ketika ingin membatalkan proses <i>edit</i> , klik tombol <i>Cancel</i>	
	14a. Sistem menampilkan kembali halaman Rujukan
SKENARIO ALTERNATIF Aktor mengubah data rujukan tetapi <i>input</i> data tidak valid	
13b. Ketika tombol <i>Submit</i> diklik pada halaman <i>Edit</i> Rujukan, tetapi <i>input</i> data tidak valid.	
	14b. Sistem menampilkan <i>message</i> “ <i>Please fill out this field</i> ”
SKENARIO NORMAL Aktor menghapus data rujukan	
17. Jika aktor ingin menghapus data rujukan, klik tombol <i>Remove</i> pada kolom <i>Action</i> di tabel Data Rujukan berdasarkan No. Transaksi yang akan dihapus	
	18. Sistem menampilkan <i>message dialog</i> “Delete?”
19. Klik tombol OK	
	20. Sistem menghapus data yang dipilih kemudian menampilkan halaman Rujukan
SKENARIO ALTERNATIF Aktor menghapus data rujukan tetapi ingin membatalkan proses hapus data setelah mengklik tombol <i>Remove</i>	
19a. Apabila aktor ingin membatalkan proses hapus data setelah mengklik tombol <i>Remove</i> pada kolom <i>Action</i> di tabel Data Rujukan, klik tombol <i>Cancel</i> pada <i>message dialog</i> konfirmasi penghapusan data	
	20a. Sistem menampilkan kembali halaman Rujukan
SKENARIO NORMAL Aktor mencetak data rujukan	
21. Jika aktor ingin mencetak data rujukan, klik tombol Cetak pada kolom <i>Action</i> di tabel Data Rujukan berdasarkan No. Transaksi yang	

akan dicetak	22. Sistem mengubah data rujukan yang akan dicetak dalam bentuk Portable Document Format (PDF) dan menampilkan data tersebut di tab baru
SKENARIO NORMAL	
Aktor mencari data rujukan	
23. Jika aktor ingin mencari data rujukan yang ada di tabel Data Rujukan, ketikkan data yang akan di cari pada <i>textfield Search</i>	24. Sistem mencari data rujukan sesuai dengan data yang diketikkan pada <i>textfield Search</i>
	25. Sistem menampilkan data tersebut pada tabel Data Rujukan
SKENARIO ALTERNATIF	
Aktor mencari data rujukan tetapi data yang dicari tidak ditemukan	
25a. Jika aktor ingin mencari data rujukan yang ada di tabel Data Rujukan, tetapi data yang dicari tidak ditemukan	26a. Sistem menampilkan <i>message "No matching records found"</i>
SKENARIO NORMAL	
Aktor ingin menampilkan isi tabel Data Rujukan dengan jumlah tertentu	
26. Jika aktor ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah <i>entries</i> data pada menu <i>dropdown Show entries</i>	27. Sistem menampilkan isi tabel berdasarkan jumlah <i>entries</i> yang dipilih

A.1.10 Use case Skenario Manajemen Transaksi Pembayaran Pasien

Tabel A.10 Use case Skenario Manajemen Transaksi Pembayaran Pasien

ID Use Case	UC-12
Nama	Manajemen Transaksi Pembayaran Pasien
Aktor	Administrator

Pre-Condition	Menampilkan Data Transaksi Pembayaran Pasien
Post-Condition	Mengelola Transaksi Pembayaran Pasien

Aksi Aktor	Reaksi Sistem
------------	---------------

SKENARIO NORMAL

1. Klik menu Transaksi Pembayaran	2. Menampilkan halaman Transaksi Pembayaran yang berisi tentang tombol-tombol Transaksi Pembayaran yang terdiri dari Perawatan, Laboratorium, dan Obat
-----------------------------------	--

SKENARIO NORMAL

Aktor mengelola Transaksi Pembayaran Perawatan

3. Jika aktor ingin mengelola Transaksi Pembayaran Perawatan, maka klik tombol Perawatan	4. Menampilkan halaman Transaksi Pembayaran Perawatan yang berisi ID Transaksi, Tanggal Transaksi, menu dropdown Pasien, menu dropdown Kamar, Kelas Kamar, Harga Kamar, Tanggal Masuk, Tanggal Keluar, Lama Inap, Keterangan yang berisi menu dropdown (Pulang, Rawat Jalan, Rujukan, Meninggal), dan <i>textfield</i> Biaya-biaya yaitu Biaya Dokter, Biaya Visit Dokter, Biaya Perawatan, Biaya Kamar, Biaya Obat, Biaya Laboratorium, Biaya Fasilitas Lain, dan Total
5. Memasukkan data transaksi pembayaran perawatan sesuai kebutuhan pasien dengan cara memilih salah satu menu <i>dropdown</i> yang berada pada keterangan	
6. Klik tombol <i>Submit</i>	7. Melakukan validasi data
	8. Apabila data yang diisi sudah valid, sistem akan menampilkan data transaksi pembayaran perawatan dalam bentuk pdf dan secara otomatis menyimpan di <i>database</i>

9. Klik tombol cetak untuk mencetak transaksi pembayaran perawatan dalam bentuk pdf

10. Sistem mencetak transaksi pembayaran perawatan dalam bentuk Pdf

SKENARIO ALTERNATIF

Aktor mengelola Transaksi Pembayaran Perawatan tetapi tidak jadi atau salah klik tombol Transaksi Pembayaran Perawatan

5a. Ketika aktor tidak jadi atau salah klik tombol Transaksi Pembayaran Perawatan, maka klik tombol *Back*

6a. Sistem menampilkan kembali halaman Transaksi Pembayaran

SKENARIO ALTERNATIF

Aktor mengelola Transaksi Pembayaran Perawatan tetapi ingin mengosongkan semua *textfield*

6a. Ketika memasukkan data transaksi pembayaran perawatan, tetapi ingin mengosongkan semua *textfield*, klik tombol *Reset* untuk mengosongkan data yang telah diisi

7a. Sistem mengosongkan data yang telah dimasukkan

SKENARIO ALTERNATIF

Aktor mengelola Transaksi Pembayaran Perawatan tetapi *input* data tidak valid

6b. Ketika tombol *Submit* diklik pada halaman Transaksi Pembayaran Perawatan, tetapi *input* data tidak valid

7b. Sistem menampilkan *message* "Please fill out this field"

SKENARIO NORMAL

Aktor mengelola Transaksi Pembayaran Laboratorium

11. Jika aktor ingin mengelola Transaksi Pembayaran Laboratorium, maka klik tombol Laboratorium

12. Menampilkan halaman Transaksi Pembayaran Laboratorium yang berisi ID Transaksi, Tanggal Transaksi,

	menu <i>dropdown</i> Pasien, <i>textfield</i> Biaya-biaya Jenis Pemeriksaan, dan Total Biaya
13. Memasukkan data transaksi pembayaran laboratorium	
14. Klik tombol <i>Submit</i>	
	15. Melakukan validasi data
	16. Apabila data yang diisi sudah valid, sistem akan menampilkan data transaksi pembayaran laboratorium dalam bentuk pdf dan secara otomatis menyimpan di <i>database</i>
17. Klik tombol cetak untuk mencetak transaksi pembayaran laboratorim dalam bentuk Pdf	
	18. Sistem mencetak transaksi pembayaran laboratorim dalam bentuk Pdf

SKENARIO ALTERNATIF

Aktor mengelola Transaksi Pembayaran Laboratorium tetapi tidak jadi atau salah klik tombol Transaksi Pembayaran Laboratorium

13a. Ketika aktor tidak jadi atau salah klik tombol Transaksi Pembayaran Laboratorium, maka klik tombol <i>Back</i>	
	14a. Sistem menampilkan kembali halaman Transaksi Pembayaran

SKENARIO ALTERNATIF

Aktor mengelola Transaksi Pembayaran Laboratorium tetapi ingin mengosongkan semua *textfield*

14a. Ketika memasukkan data transaksi pembayaran laboratorium, tetapi ingin mengosongkan semua <i>textfield</i> , klik tombol <i>Reset</i> untuk mengosongkan data yang telah diisi	
	15a. Sistem mengosongkan data yang telah dimasukkan

SKENARIO ALTERNATIF

Aktor mengelola Transaksi Pembayaran Laboratorium tetapi *input* data tidak valid

14b. Ketika tombol *Submit* diklik pada halaman Transaksi Pembayaran Laboratorium, tetapi *input* data tidak valid

15b. Sistem menampilkan *message* "Please fill out this field"

SKENARIO NORMAL

Aktor mengelola Transaksi Pembayaran Obat

19. Jika aktor ingin mengelola Transaksi Pembayaran Obat, maka klik tombol Obat

20. Menampilkan halaman Transaksi Pembayaran Obat yang berisi ID Transaksi, Tanggal Transaksi, menu *dropdown* Pasien, menu *dropdown* Obat, *textfield* Qty, tombol Add, Tabel Sementara Detail Obat, dan Grand Total. Tabel Sementara Detail Obat berisi tentang Nama Obat, Harga, Qty, dan Total

21. Memasukkan data transaksi pembayaran obat yang terdiri dari data pasien dan data obat

22. Klik tombol Add

23. Sistem memasukkan dan menampilkan transaksi biaya obat dan total biaya obat kedalam tabel Sementara Detail Obat berdasarkan data yang telah dimasukkan dan secara otomatis kotak isian Data Obat menjadi kosong kembali

24. Jika obat yang dibeli lebih dari satu, maka admin memasukkan Data Obat kembali kemudian klik tombol Add

25. Sistem kembali memasukkan dan menampilkan transaksi biaya obat dan total biaya obat kedalam tabel Sementara Detail Obat berdasarkan data yang telah dimasukkan dan secara otomatis kotak isian Data Obat menjadi kosong kembali

SKENARIO NORMAL
Aktor menghapus Data Obat yang telah dimasukkan kedalam tabel Sementara Detail Obat

<p>26. Jika aktor ingin menghapus Data Obat yang telah dimasukkan kedalam tabel Sementara Detail Obat, maka klik tombol <i>Remove</i> yang terletak di sebelah kolom Total</p>	
	<p>27. Menampilkan <i>message dialog</i> “Are You Sure You Want to Remove This Data?”</p>
<p>28. Klik tombol OK</p>	
	<p>29. Sistem akan menghapus data yang dipilih kemudian menampilkan kembali halaman Transaksi Pembayaran Obat</p>
<p>30. Klik tombol <i>Submit</i></p>	
	<p>31. Melakukan validasi data</p>
	<p>32. Apabila data yang diisi sudah valid, sistem menampilkan data transaksi pembayaran Obat dalam bentuk pdf dan secara otomatis menyimpan di <i>database</i></p>
<p>33. Klik tombol cetak untuk mencetak transaksi pembayaran obat dalam bentuk Pdf</p>	
	<p>34. Sistem mencetak transaksi pembayaran obat dalam bentuk Pdf</p>

SKENARIO ALTERNATIF
Aktor mengelola Transaksi Pembayaran Obat tetapi tidak jadi atau salah klik tombol Transaksi Pembayaran Obat

<p>21a. Ketika aktor tidak jadi atau salah klik tombol Transaksi Pembayaran Obat, maka klik tombol <i>Back</i></p>	
	<p>22a. Sistem menampilkan kembali halaman Transaksi Pembayaran</p>

SKENARIO ALTERNATIF
Aktor menghapus Data Obat yang telah dimasukkan kedalam tabel Sementara Detail Obat ingin membatalkan proses hapus data setelah mengklik tombol *Remove*

28a. Apabila aktor ingin membatalkan proses hapus data setelah mengklik tombol <i>Remove</i> pada tabel Sementara Detail Obat, klik tombol <i>Cancel</i> pada <i>message dialog</i> konfirmasi penghapusan data	29a. Sistem menampilkan kembali halaman Transaksi Pembayaran Obat
SKENARIO ALTERNATIF	
Aktor mengelola Transaksi Pembayaran Obat tetapi <i>input</i> data tidak valid	
30a. Ketika tombol <i>Submit</i> diklik pada halaman Transaksi Pembayaran Obat, tetapi <i>input</i> data tidak valid	31a. Sistem menampilkan <i>message</i> “ <i>Please fill out this field</i> ”

A.1.11 Use case Skenario Melihat Laporan

Tabel A.11 Use case Skenario Melihat Laporan

ID Use Case	UC-13
Nama	Manajemen Laporan
Aktor	Administrator dan Manajer
Pre-Condition	Menampilkan Data Laporan
Post-Condition	Melihat daftar dan isi Laporan
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Klik menu Laporan	2. Menampilkan halaman Laporan yang berisi tentang tombol-tombol jenis Laporan yaitu terdiri dari Laporan Pasien, Laporan Kamar, Laporan Dokter, Laporan Pegawai, Laporan Fasilitas, Laporan Obat, Laporan Laboratorium, Laporan Rekam Medis, dan Laporan Rujukan
SKENARIO NORMAL	
Aktor membaca Laporan Pasien	
3. Jika aktor ingin membaca Laporan Pasien, maka klik tombol Pasien	4. Menampilkan halaman Laporan Pasien yang berisi tombol <i>Back</i> ,

tombol Cetak, *textfield Search*, menu dropdown *Show entries* tabel, dan tabel Data Pasien

5. Jika aktor ingin mencetak Laporan Pasien, maka klik tombol Cetak

6. Menampilkan Laporan Pasien dalam bentuk Pdf di tab baru

SKENARIO ALTERNATIF
Aktor membaca Laporan Pasien tetapi tidak jadi atau salah klik tombol Laporan Pasien

5a. Ketika aktor tidak jadi atau salah klik tombol Laporan Pasien, maka klik tombol *Back*

6a. Sistem menampilkan kembali halaman Laporan

SKENARIO NORMAL
Aktor mencari data pasien yang ada di tabel Data Pasien

7. Jika aktor ingin mencari data pasien yang ada di tabel Data Pasien, ketikkan data yang akan di cari pada *textfield Search*

8. Sistem mencari data pasien sesuai dengan data yang diketikkan pada *textfield Search*

9. Sistem menampilkan data tersebut pada tabel Data Pasien

SKENARIO ALTERNATIF
Aktor mencari data pasien yang ada di tabel Data Pasien tetapi data yang dicari tidak ditemukan

7a. Jika aktor ingin mencari data pasien yang ada di tabel Data Pasien, tetapi data yang dicari tidak ditemukan

8a. Sistem menampilkan *message "No matching records found"*

SKENARIO NORMAL
Aktor ingin menampilkan isi tabel dengan jumlah tertentu

10. Jika aktor ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah *entries* data pada menu dropdown *Show entries*

11. Sistem menampilkan isi tabel

berdasarkan jumlah *entries* yang dipilih

SKENARIO NORMAL
Aktor membaca Laporan Kamar

12. Jika aktor ingin membaca Laporan Kamar, maka klik tombol Kamar

13. Menampilkan halaman Laporan Kamar yang berisi tombol *Back*, tombol Cetak, *textfield Search*, menu dropdown *Show entries* tabel, dan tabel Data Kamar

14. Jika aktor ingin mencetak Laporan Kamar, maka klik tombol Cetak

15. Menampilkan Laporan Kamar dalam bentuk Pdf di tab baru

SKENARIO ALTERNATIF
Aktor membaca Laporan Kamar tetapi tidak jadi atau salah klik tombol Laporan Kamar

14a. Ketika aktor tidak jadi atau salah klik tombol Laporan Kamar, maka klik tombol *Back*

15a. Sistem menampilkan kembali halaman Laporan

SKENARIO NORMAL
Aktor mencari data kamar yang ada di tabel Data Kamar

16. Jika aktor ingin mencari data kamar yang ada di tabel Data Kamar, ketikkan data yang akan dicari pada *textfield Search*

17. Sistem mencari data kamar sesuai dengan data yang diketikkan pada *textfield Search*

18. Sistem menampilkan data tersebut pada tabel Data Kamar

SKENARIO ALTERNATIF
Aktor mencari data kamar yang ada di tabel Data Kamar tetapi data yang dicari tidak ditemukan

16a. Jika aktor ingin mencari data kamar yang ada di tabel Data Kamar, tetapi data yang dicari tidak ditemukan

17a. Sistem menampilkan *message* “No matching records found”

SKENARIO NORMAL

Aktor ingin menampilkan isi tabel dengan jumlah tertentu

19. Jika aktor ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah *entries* data pada menu dropdown *Show entries*

20. Sistem menampilkan isi tabel berdasarkan jumlah *entries* yang dipilih

SKENARIO NORMAL

Aktor membaca Laporan Dokter

21. Jika aktor ingin membaca Laporan Dokter, maka klik tombol Dokter

22. Menampilkan halaman Laporan Dokter yang berisi tombol *Back*, tombol Cetak, *textfield Search*, menu dropdown *Show entries* tabel, dan tabel Data Dokter

23. Jika aktor ingin mencetak Laporan Dokter, maka klik tombol Cetak

24. Menampilkan Laporan Dokter dalam bentuk Pdf di tab baru

SKENARIO ALTERNATIF

Aktor membaca Laporan Dokter tetapi tidak jadi atau salah klik tombol Laporan Dokter

23a. Ketika aktor tidak jadi atau salah klik tombol Laporan Dokter, maka klik tombol *Back*

24a. Sistem menampilkan kembali halaman Laporan

SKENARIO NORMAL

Aktor mencari data dokter yang ada di tabel Data Dokter

25. Jika aktor ingin mencari data dokter yang ada di tabel Data Dokter, ketikkan data yang akan di cari pada *textfield Search*

26. Sistem mencari data dokter sesuai dengan data yang diketikkan pada *textfield Search*

27. Sistem menampilkan data tersebut

pada tabel Data Dokter

SKENARIO ALTERNATIF

Aktor mencari data dokter yang ada di tabel Data Dokter tetapi data yang dicari tidak ditemukan

25a. Jika aktor ingin mencari data dokter yang ada di tabel Data Dokter, tetapi data yang dicari tidak ditemukan

26a. Sistem menampilkan *message* “*No matching records found*”

SKENARIO NORMAL

Aktor ingin menampilkan isi tabel dengan jumlah tertentu

28. Jika aktor ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah *entries* data pada menu dropdown *Show entries*

29. Sistem menampilkan isi tabel berdasarkan jumlah *entries* yang dipilih

SKENARIO NORMAL

Aktor membaca Laporan Pegawai

30. Jika aktor ingin membaca Laporan Pegawai, maka klik tombol Pegawai

31. Menampilkan halaman Laporan Pegawai yang berisi tombol *Back*, tombol Cetak, *textfield Search*, menu dropdown *Show entries* tabel, dan tabel Data Pegawai

32. Jika aktor ingin mencetak Laporan Pegawai, maka klik tombol Cetak

33. Menampilkan Laporan Pegawai dalam bentuk Pdf di tab baru

SKENARIO ALTERNATIF

Aktor membaca Laporan Pegawai tetapi tidak jadi atau salah klik tombol Laporan Pegawai

32a. Ketika aktor tidak jadi atau salah klik tombol Laporan Pegawai, maka klik tombol *Back*

33a. Sistem menampilkan kembali halaman Laporan

SKENARIO NORMAL

Aktor mencari data pegawai yang ada di tabel Data Pegawai

34. Jika aktor ingin mencari data pegawai yang ada di tabel Data Pegawai, ketikkan data yang akan di cari pada *textfield Search*

35. Sistem mencari data pegawai sesuai dengan data yang diketikkan pada *textfield Search*

36. Sistem menampilkan data tersebut pada tabel Data Pegawai

SKENARIO ALTERNATIF**Aktor mencari data pegawai yang ada di tabel Data Pegawai tetapi data yang dicari tidak ditemukan**

34a. Jika aktor ingin mencari data pegawai yang ada di tabel Data Pegawai, tetapi data yang dicari tidak ditemukan

35a. Sistem menampilkan *message "No matching records found"*

SKENARIO NORMAL**Aktor ingin menampilkan isi tabel dengan jumlah tertentu**

37. Jika aktor ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah *entries* data pada menu dropdown *Show entries*

38. Sistem menampilkan isi tabel berdasarkan jumlah *entries* yang dipilih

SKENARIO NORMAL**Aktor membaca Laporan Fasilitas**

39. Jika aktor ingin membaca Laporan Fasilitas, maka klik tombol Fasilitas

40. Menampilkan halaman Laporan Fasilitas yang berisi tombol *Back*, tombol *Cetak*, *textfield Search*, menu dropdown *Show entries* tabel, dan tabel Data Fasilitas

41. Jika aktor ingin mencetak Laporan Fasilitas, maka klik tombol *Cetak*

42. Menampilkan Laporan Fasilitas dalam bentuk Pdf di tab baru

SKENARIO ALTERNATIF

Aktor membaca Laporan Fasilitas tetapi tidak jadi atau salah klik tombol Laporan Fasilitas

41a. Ketika aktor tidak jadi atau salah klik tombol Laporan Fasilitas, maka klik tombol *Back*

42a. Sistem menampilkan kembali halaman Laporan

SKENARIO NORMAL

Aktor mencari data fasilitas yang ada di tabel Data Fasilitas

43. Jika aktor ingin mencari data fasilitas yang ada di tabel Data Fasilitas, ketikkan data yang akan di cari pada *textfield Search*

44. Sistem mencari data fasilitas sesuai dengan data yang diketikkan pada *textfield Search*

45. Sistem menampilkan data tersebut pada tabel Data Fasilitas

SKENARIO ALTERNATIF

Aktor mencari data fasilitas yang ada di tabel Data Fasilitas tetapi data yang dicari tidak ditemukan

43a. Jika aktor ingin mencari data fasilitas yang ada di tabel Data Fasilitas, tetapi data yang dicari tidak ditemukan

44a. Sistem menampilkan *message "No matching records found"*

SKENARIO NORMAL

Aktor ingin menampilkan isi tabel dengan jumlah tertentu

46. Jika aktor ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah *entries* data pada menu dropdown *Show entries*

47. Sistem menampilkan isi tabel berdasarkan jumlah *entries* yang dipilih

SKENARIO NORMAL

Aktor membaca Laporan Obat

48. Jika aktor ingin membaca Laporan Obat, maka klik tombol Obat

49. Menampilkan halaman Laporan Obat yang berisi tombol *Back*, tombol

Cetak, *textfield Search*, menu dropdown *Show entries* tabel, dan tabel Data Obat

50. Jika aktor ingin mencetak Laporan Obat, maka klik tombol Cetak

51. Menampilkan Laporan Obat dalam bentuk Pdf di tab baru

SKENARIO ALTERNATIF
Aktor membaca Laporan Obat tetapi tidak jadi atau salah klik tombol Laporan Obat

50a. Ketika aktor tidak jadi atau salah klik tombol Laporan Obat, maka klik tombol *Back*

51a. Sistem menampilkan kembali halaman Laporan

SKENARIO NORMAL
Aktor mencari data obat yang ada di tabel Data Obat

52. Jika aktor ingin mencari data obat yang ada di tabel Data Obat, ketikkan data yang akan di cari pada *textfield Search*

53. Sistem mencari data obat sesuai dengan data yang diketikkan pada *textfield Search*

54. Sistem menampilkan data tersebut pada tabel Data Obat

SKENARIO ALTERNATIF
Aktor mencari data obat yang ada di tabel Data Obat tetapi data yang dicari tidak ditemukan

52a. Jika aktor ingin mencari data obat yang ada di tabel Data Obat, tetapi data yang dicari tidak ditemukan

53a. Sistem menampilkan *message "No matching records found"*

SKENARIO NORMAL
Aktor ingin menampilkan isi tabel dengan jumlah tertentu

55. Jika aktor ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah *entries* data pada menu dropdown *Show entries*

56. Sistem menampilkan isi tabel berdasarkan jumlah *entries* yang

dipilih

SKENARIO NORMAL

Aktor membaca Laporan Laboratorium

57. Jika aktor ingin membaca Laporan Laboratorium, maka klik tombol Laboratorim

58. Menampilkan halaman Laporan Laboratorium yang berisi tombol *Back*, tombol Cetak, *textfield Search*, menu dropdown *Show entries* tabel, dan tabel Data Laboratorium

59. Jika aktor ingin mencetak Laporan Laboratorium, maka klik tombol Cetak

60. Menampilkan Laporan Laboratorium dalam bentuk Pdf di tab baru

SKENARIO ALTERNATIF

Aktor membaca Laporan Laboratorium tetapi tidak jadi atau salah klik tombol Laporan Laboratorium

59a. Ketika aktor tidak jadi atau salah klik tombol Laporan Laboratorium, maka klik tombol *Back*

60a. Sistem menampilkan kembali halaman Laporan

SKENARIO NORMAL

Aktor mencari data laboratorim yang ada di tabel Data Laboratorium

61. Jika aktor ingin mencari data laboratorim yang ada di tabel Data Laboratorium, ketikkan data yang akan di cari pada *textfield Search*

62. Sistem mencari data laboratorium sesuai dengan data yang diketikkan pada *textfield Search*

63. Sistem menampilkan data tersebut pada tabel Data Laboratorium

SKENARIO ALTERNATIF

Aktor mencari data laboratorim yang ada di tabel Data Laboratorium tetapi data yang dicari tidak ditemukan

61a. Jika aktor ingin mencari data laboratorium yang ada di tabel Data Laboratorium, tetapi data yang dicari tidak ditemukan

62a. Sistem menampilkan *message* “*No matching records found*”

SKENARIO NORMAL

Aktor ingin menampilkan isi tabel dengan jumlah tertentu

64. Jika aktor ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah *entries* data pada menu dropdown *Show entries*

65. Sistem menampilkan isi tabel berdasarkan jumlah *entries* yang dipilih

SKENARIO NORMAL

Aktor membaca Laporan Rekam Medis

66. Jika aktor ingin membaca Laporan Rekam Medis, maka klik tombol Rekam Medis

67. Menampilkan halaman Laporan Rekam Medis yang berisi tombol *Back*, tombol Cetak, *textfield Search*, menu dropdown *Show entries* tabel, dan tabel Data Rekam Medis

68. Jika aktor ingin mencetak Laporan Rekam Medis, maka klik tombol Cetak

69. Menampilkan Laporan Rekam Medis dalam bentuk Pdf di tab baru

SKENARIO ALTERNATIF

Aktor membaca Laporan Rekam Medis tetapi tidak jadi atau salah klik tombol Laporan Rekam Medis

68a. Ketika aktor tidak jadi atau salah klik tombol Laporan Rekam Medis, maka klik tombol *Back*

69a. Sistem menampilkan kembali halaman Laporan

SKENARIO NORMAL

Aktor mencari data rekam medis yang ada di tabel Data Rekam Medis

70. Jika aktor ingin mencari data rekam medis yang ada di tabel Data Rekam Medis, ketikkan data yang akan di cari pada *textfield Search*

71. Sistem mencari data rekam medis sesuai dengan data yang diketikkan

pada *textfield Search*

72. Sistem menampilkan data tersebut pada tabel Data Rekam Medis

SKENARIO ALTERNATIF

Aktor mencari data rekam medis yang ada di tabel Data Rekam Medis tetapi data yang dicari tidak ditemukan

70a. Jika aktor ingin mencari data rekam medis yang ada di tabel Data Rekam Medis, tetapi data yang dicari tidak ditemukan

71a. Sistem menampilkan *message "No matching records found"*

SKENARIO NORMAL

Aktor ingin menampilkan isi tabel dengan jumlah tertentu

73. Jika aktor ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah *entries* data pada menu dropdown *Show entries*

74. Sistem menampilkan isi tabel berdasarkan jumlah *entries* yang dipilih

SKENARIO NORMAL

Aktor membaca Laporan Rujukan

75. Jika aktor ingin membaca Laporan Rujukan, maka klik tombol Rujukan

76. Menampilkan halaman Laporan Rujukan yang berisi tombol *Back*, tombol Cetak, *textfield Search*, menu dropdown *Show entries* tabel, dan tabel Data Rujukan

77. Jika aktor ingin mencetak Laporan Rujukan, maka klik tombol Cetak

78. Menampilkan Laporan Rujukan dalam bentuk Pdf di tab baru

SKENARIO ALTERNATIF

Aktor membaca Laporan Rujukan tetapi tidak jadi atau salah klik tombol Laporan Rujukan

77a. Ketika aktor tidak jadi atau salah klik tombol Laporan Rujukan, maka klik tombol *Back*

78a. Sistem menampilkan kembali halaman Laporan

SKENARIO NORMAL

Aktor mencari data rujukan yang ada di tabel Data Rujukan

79. Jika aktor ingin mencari data rujukan yang ada di tabel Data Rujukan, ketikkan data yang akan di cari pada *textfield Search*

80. Sistem mencari data rujukan sesuai dengan data yang diketikkan pada *textfield Search*

81. Sistem menampilkan data tersebut pada tabel Data Rujukan

SKENARIO ALTERNATIF

Aktor mencari data rujukan yang ada di tabel Data Rujukan tetapi data yang dicari tidak ditemukan

79a. Jika aktor ingin mencari data rujukan yang ada di tabel Data Rujukan, tetapi data yang dicari tidak ditemukan

80a. Sistem menampilkan *message "No matching records found"*

SKENARIO NORMAL

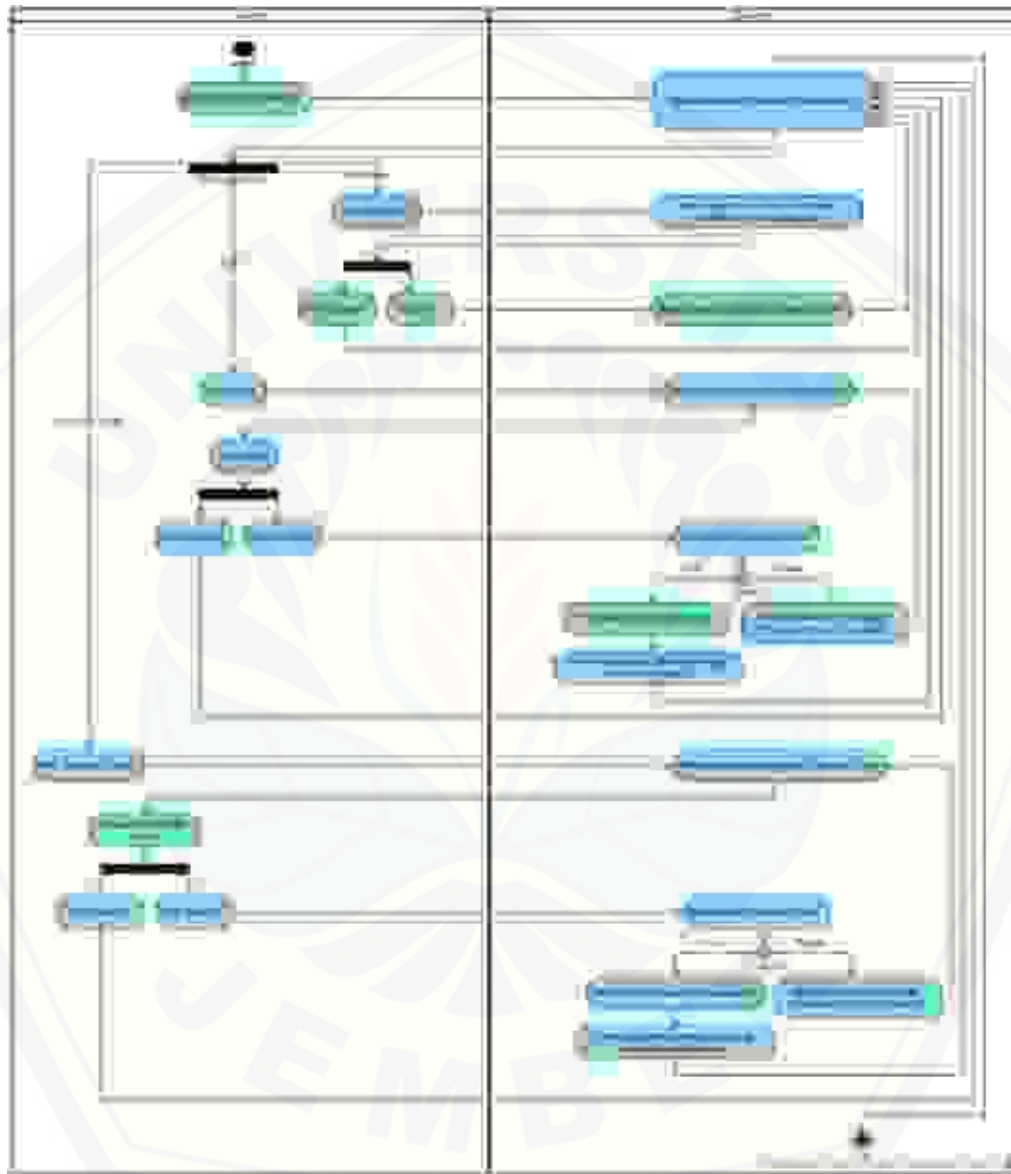
Aktor ingin menampilkan isi tabel masing-masing Laporan dengan jumlah tertentu

82. Jika aktor ingin menampilkan isi tabel dengan jumlah tertentu, maka pilih jumlah *entries* data pada menu dropdown *Show entries*

83. Sistem menampilkan isi tabel berdasarkan jumlah *entries* yang dipilih

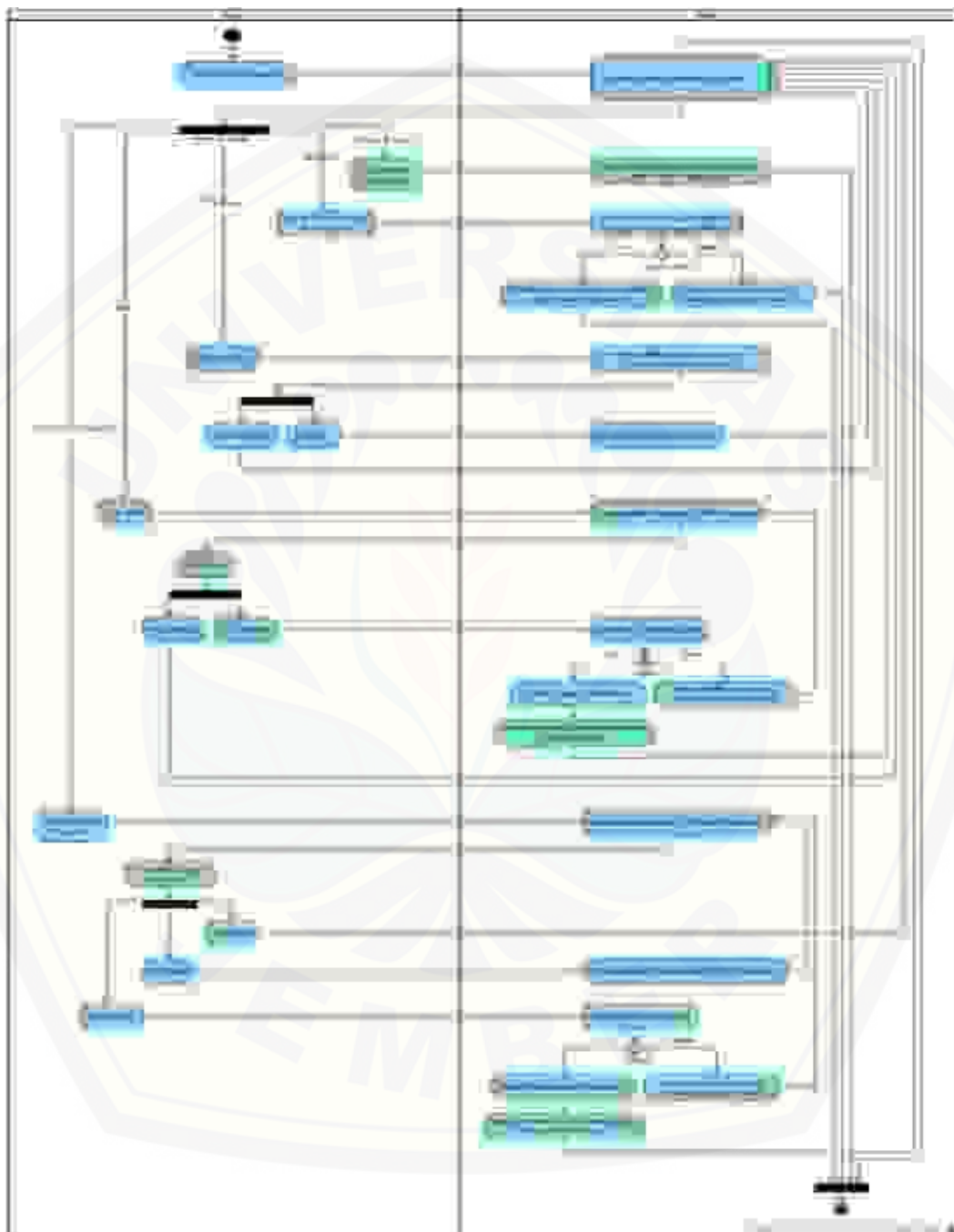
A.2 Activity Diagram

A.2.1 Activity Diagram Manajemen Data User



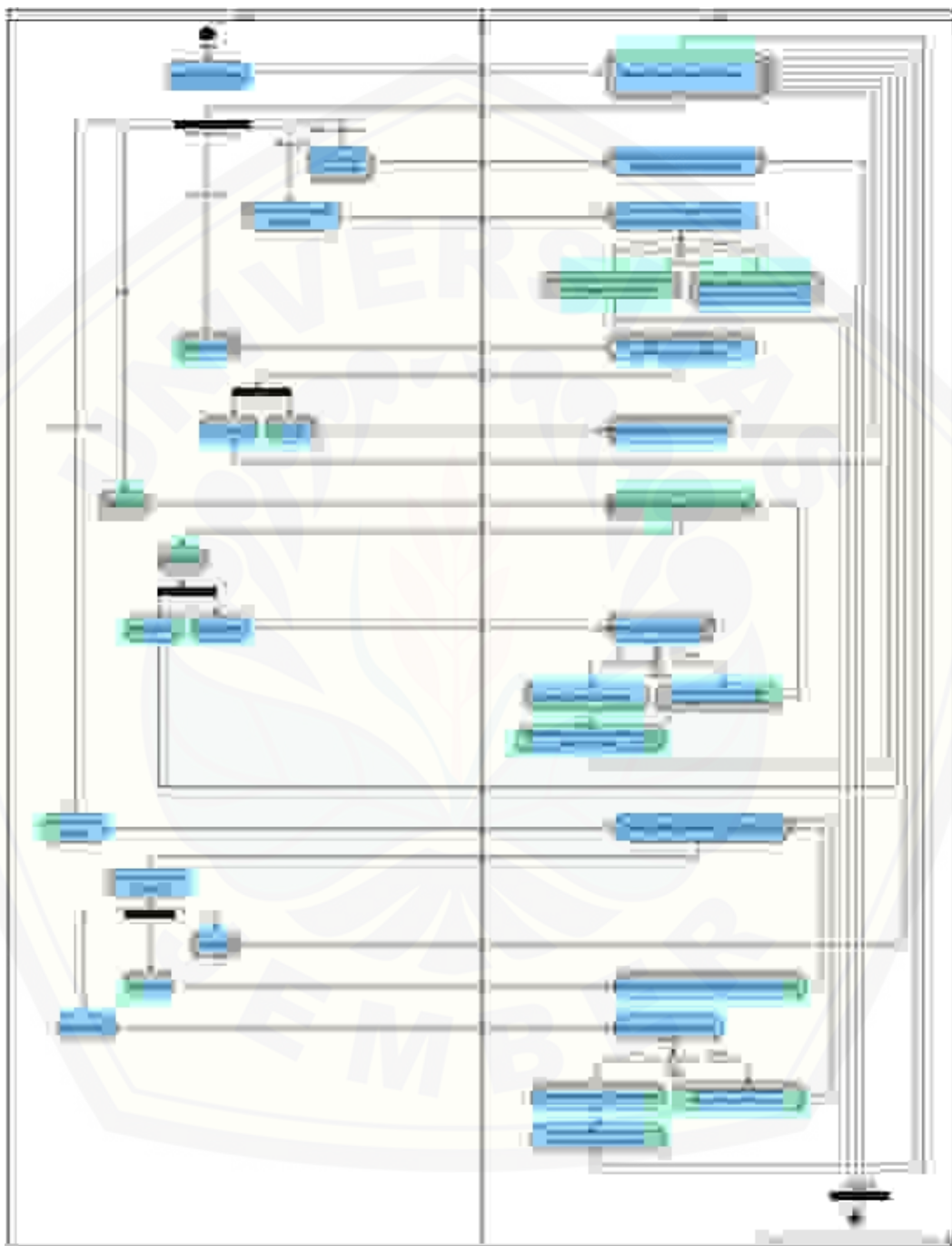
Gambar A. 1. Activity Diagram Manajemen Data User

A.2.2 Activity Diagram Manajemen Data Pegawai



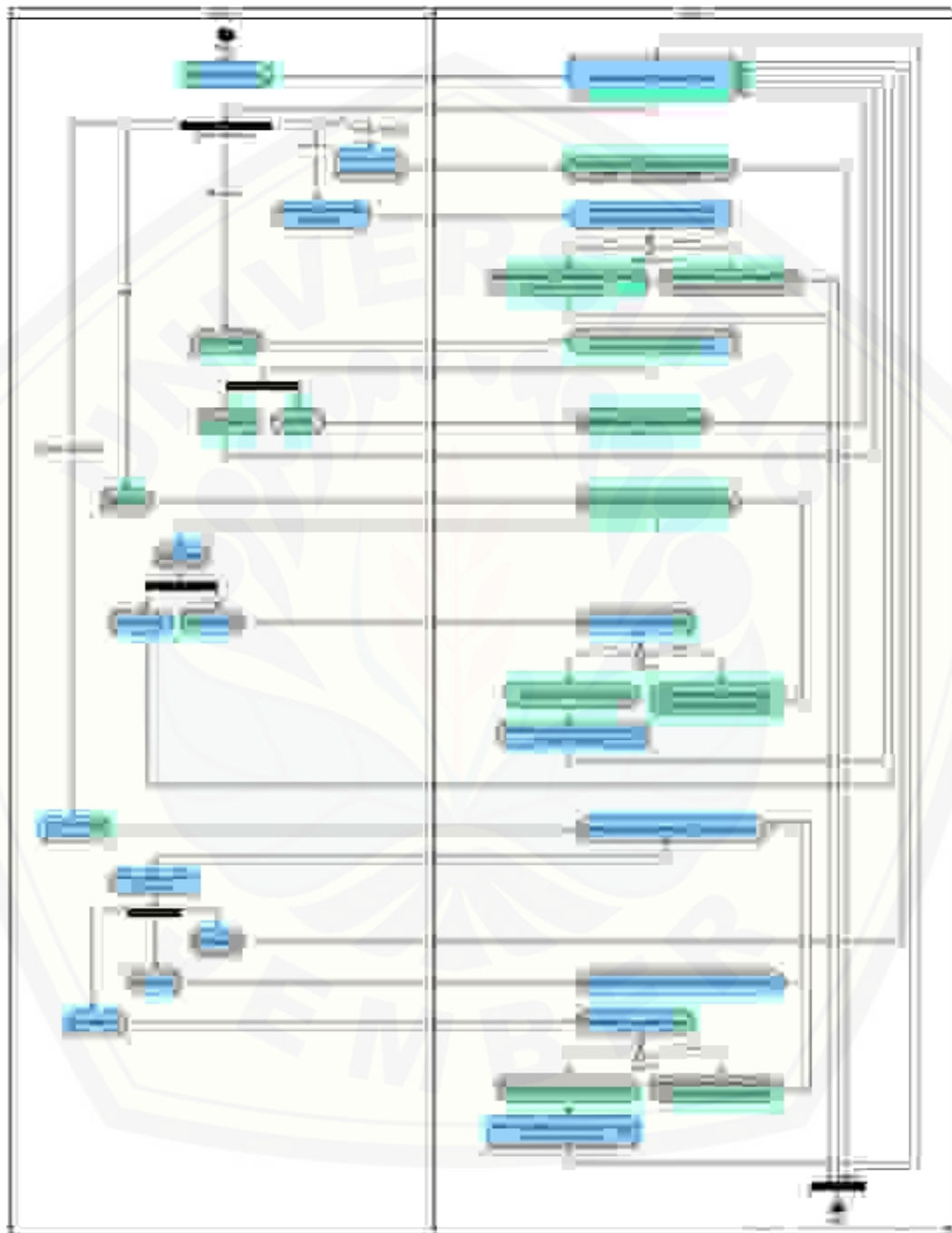
Gambar A.2 Activity Diagram Manajemen Data Pegawai

A.2.3 Activity Diagram Manajemen Data Pasien



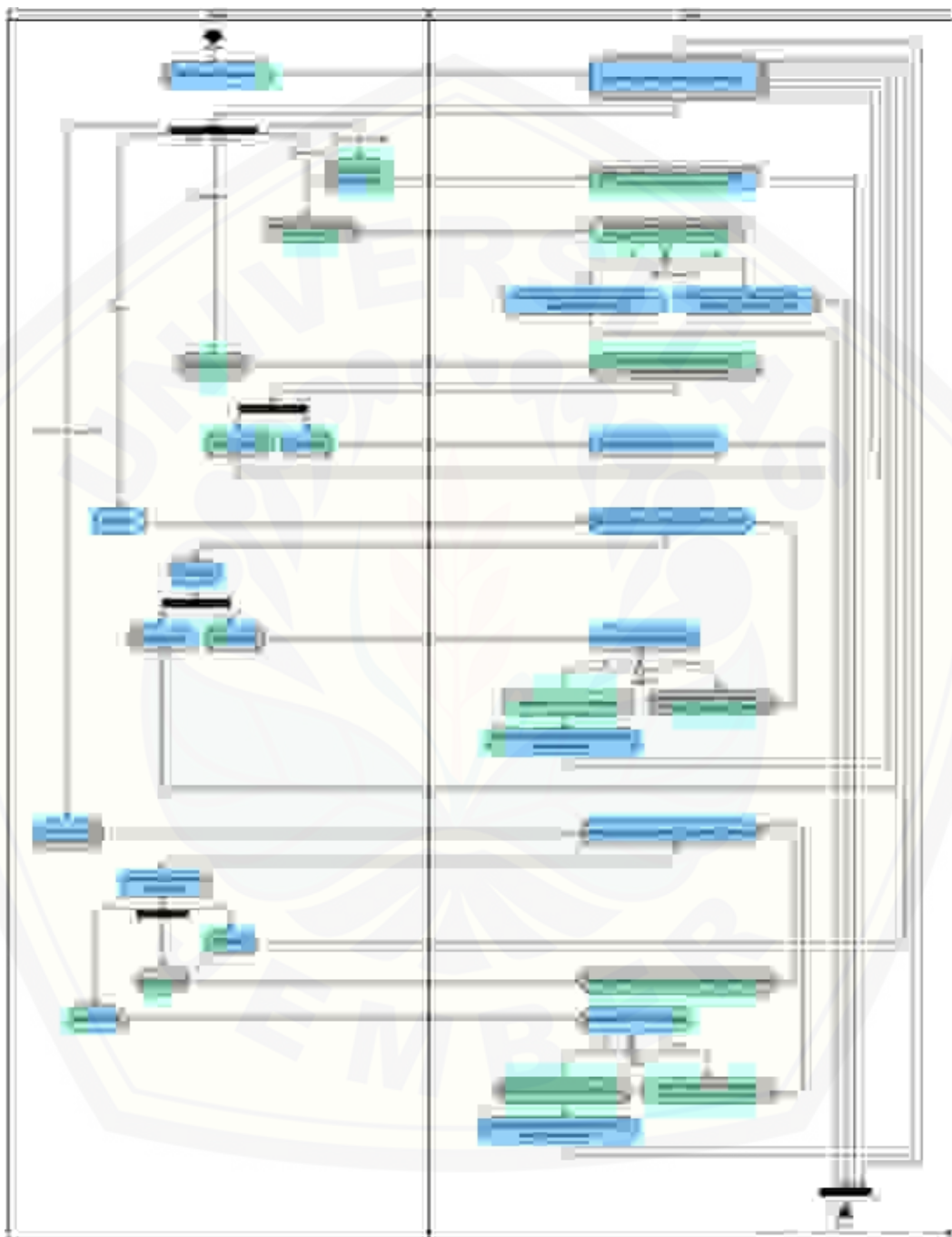
Gambar A.3 Activity Diagram Manajemen Data Pasien

A.2.4 Activity Diagram Manajemen Data Kamar



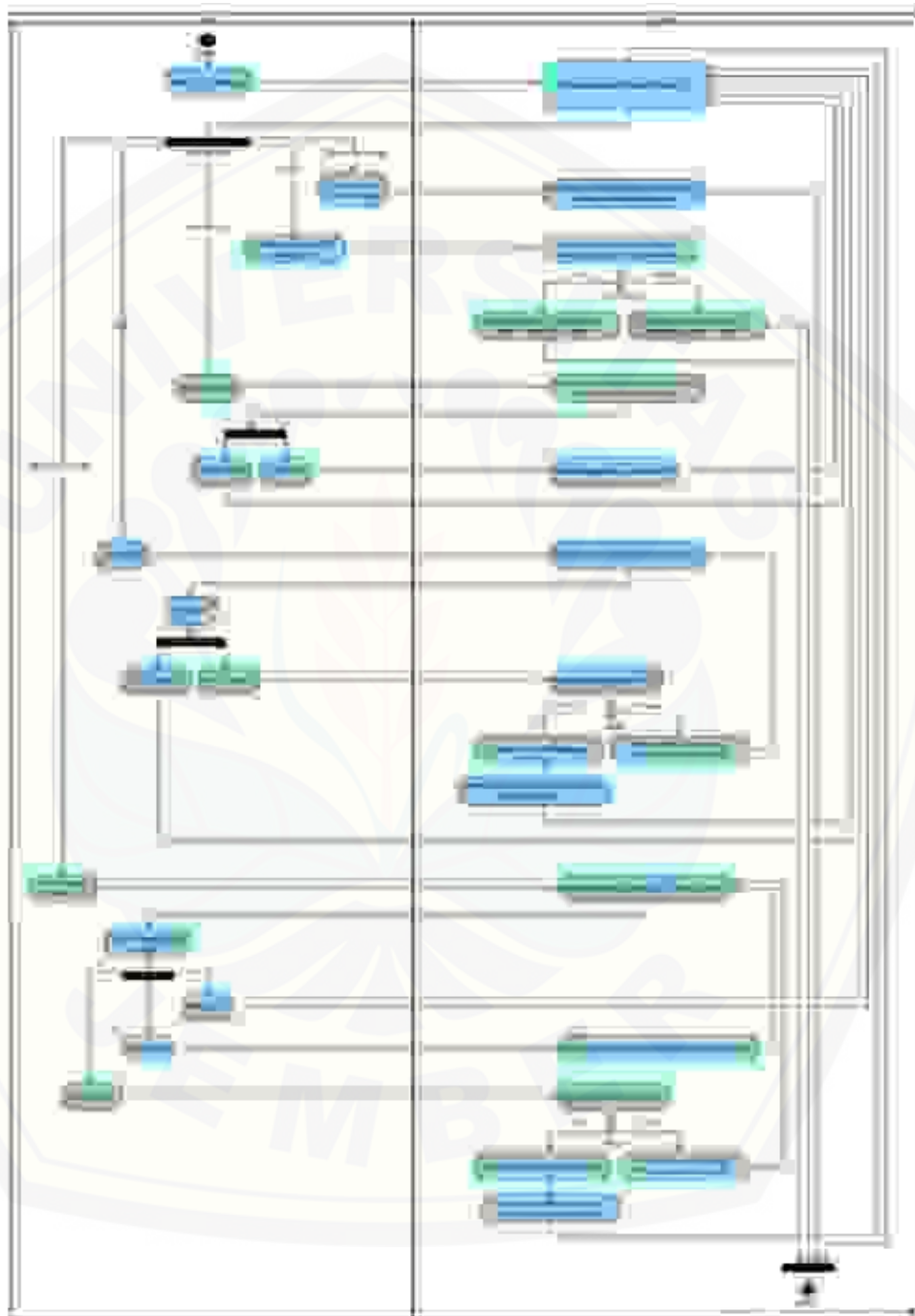
Gambar A.4 Activity Diagram Manajemen Data Kamar

A.2.5 Activity Diagram Manajemen Data Fasilitas



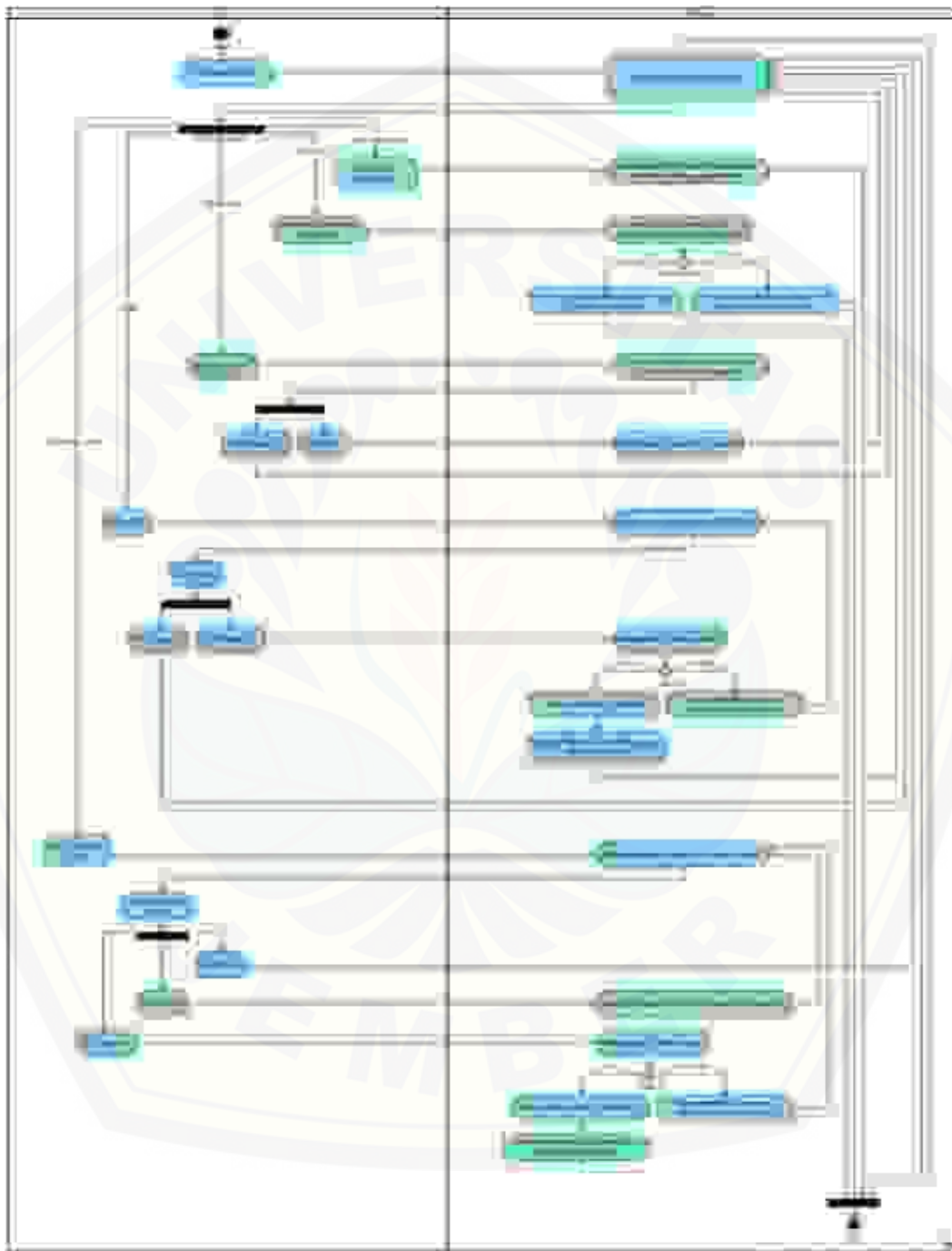
Gambar A.5 Activity Diagram Manajemen Data Fasilitas

A.2.6 Activity Diagram Manajemen Data Dokter



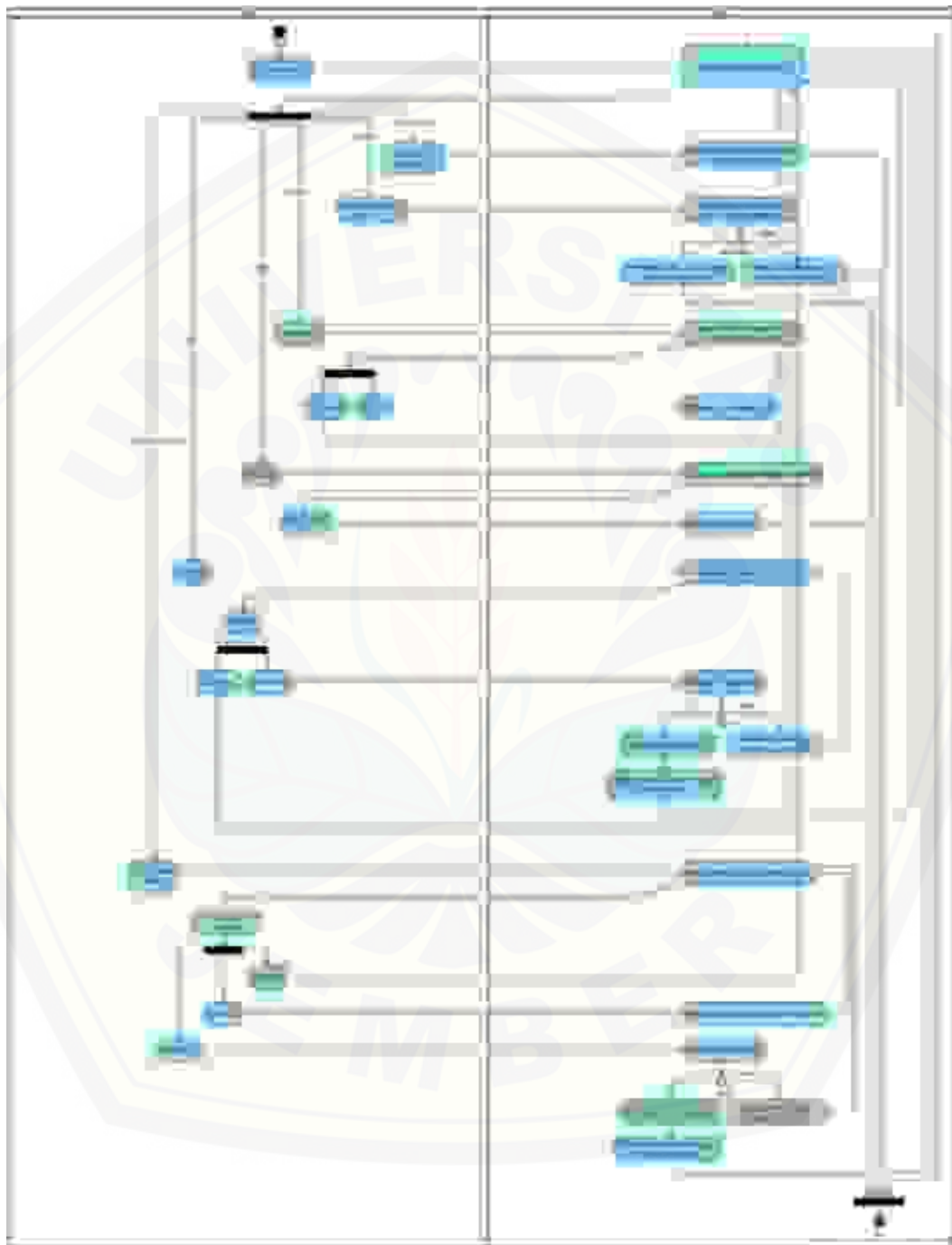
Gambar A.6 Activity Diagram Manajemen Data Dokter

A.2.7 Activity Diagram Manajemen Data Obat



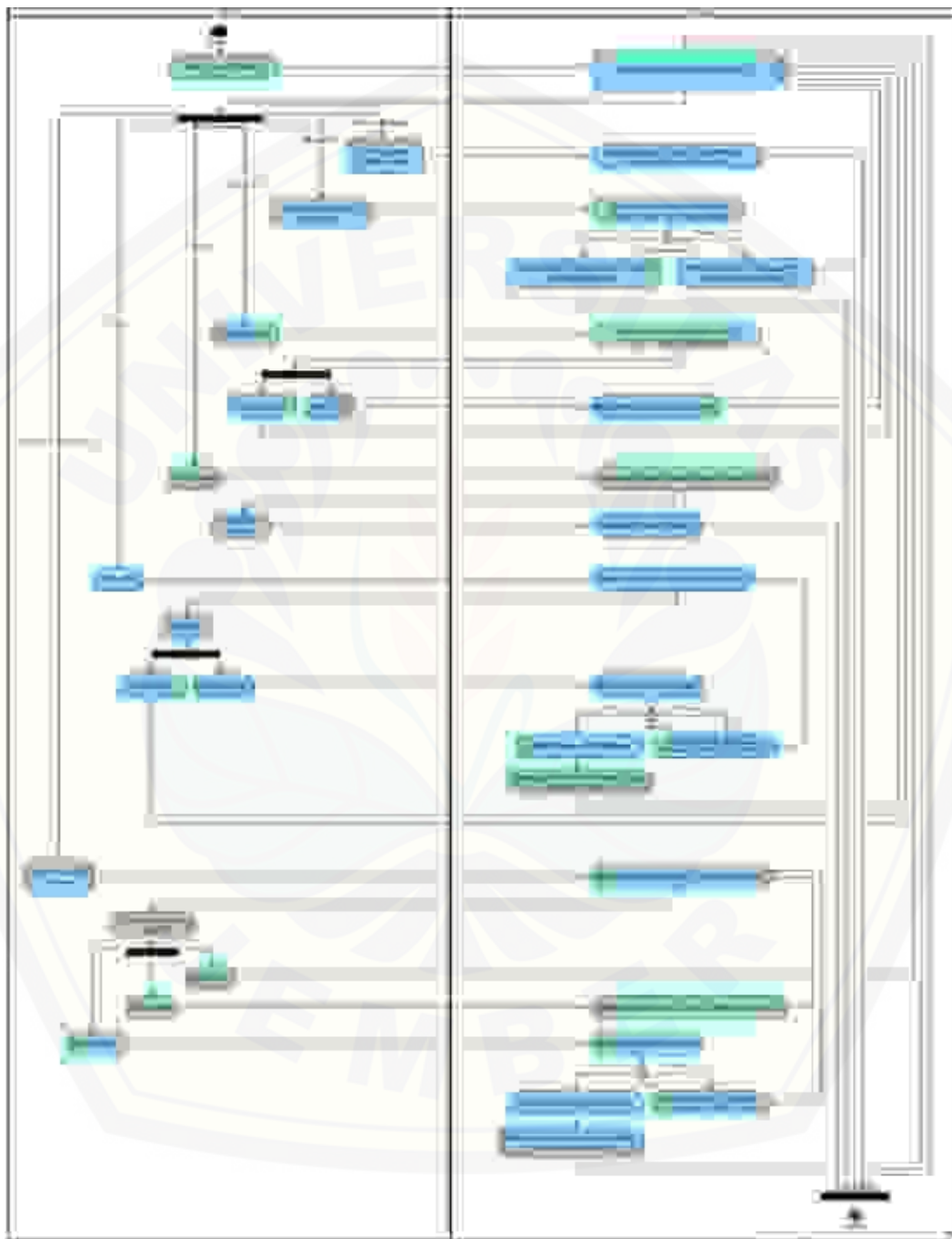
Gambar A.7 Activity Diagram Manajemen Data Obat

A.2.8 Activity Diagram Manajemen Data Laborat



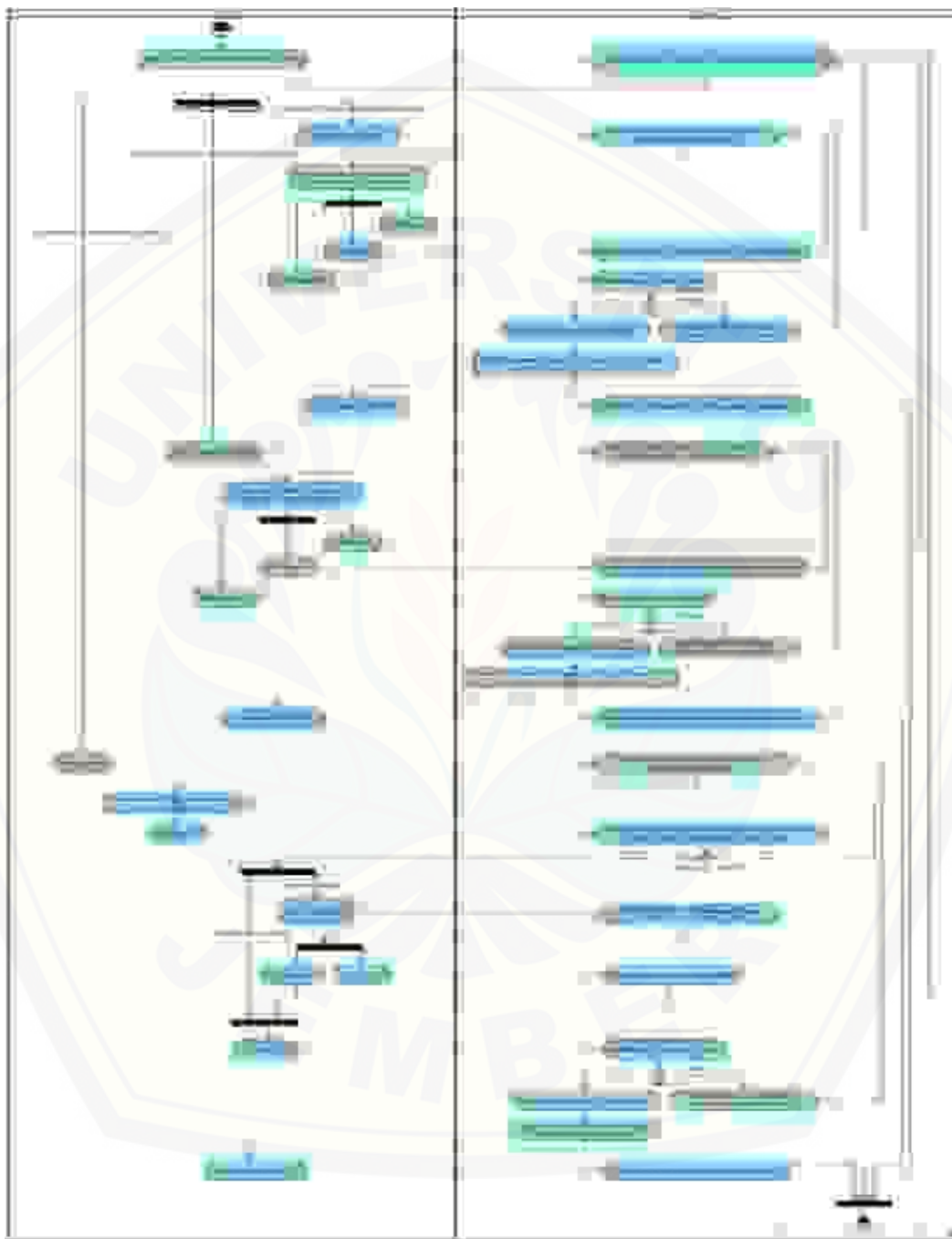
Gambar A.8 Activity Diagram Manajemen Data Laborat

A.2.9 Activity Diagram Manajemen Data Rujukan



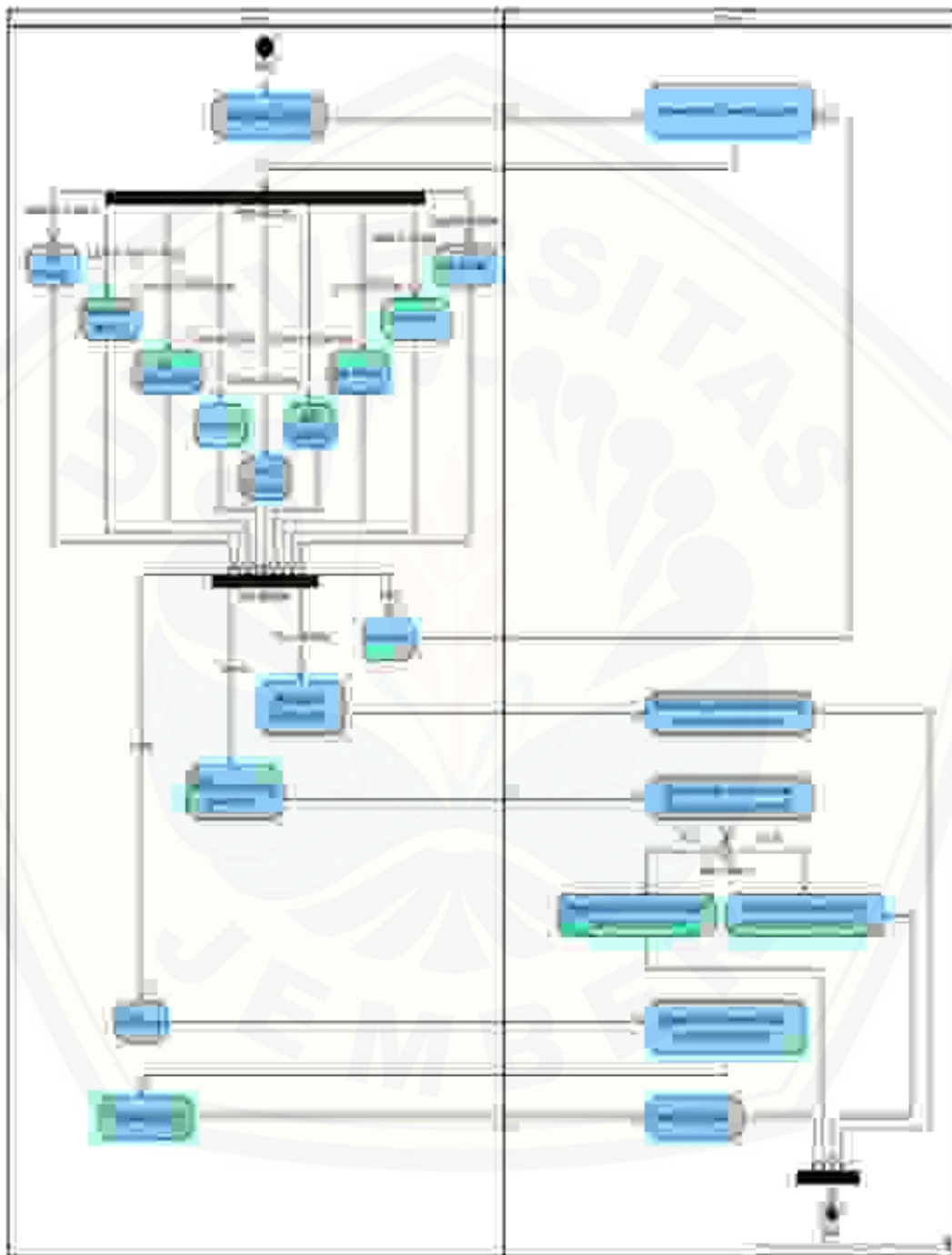
Gambar A.9 Activity Diagram Manajemen Data Rujukan

A.2.10 Activity Diagram Manajemen Transaksi Pembayaran



Gambar A.10 Activity Diagram Manajemen Transaksi Pembayaran

A.2.11 Activity Diagram Melihat Laporan



Gambar A.11 Activity Diagram Melihat Laporan

A.3 Sequence Diagram

A.3.1 Sequence Diagram Manajemen Data User



Gambar A.12 Sequence Diagram Manajemen Data User

A.3.2 Sequence Diagram Manajemen Data Pegawai



Gambar A.13 Sequence Diagram Manajemen Data Pegawai

A.3.3 Sequence Diagram Manajemen Data Pasien



Gambar A.14 Sequence Diagram Manajemen Data Pasien

A.3.4 Sequence Diagram Manajemen Data Kamar



Gambar A.15 Sequence Diagram Manajemen Data Kamar

A.3.5 Sequence Diagram Manajemen Data Fasilitas



Gambar A.16 Sequence Diagram Manajemen Data Fasilitas

A.3.6 Sequence Diagram Manajemen Data Dokter



Gambar A.17 Sequence Diagram Manajemen Data Dokter

A.3.7 Sequence Diagram Manajemen Data Obat



Gambar A.18 *Sequence* Diagram Manajemen Data Obat

A.3.8 Sequence Diagram Manajemen Data Laborat



Gambar A.19 Sequence Diagram Manajemen Data Laborat

A.3.9 Sequence Diagram Manajemen Data Rujukan



Gambar A.20 Sequence Diagram Manajemen Data Rujukan

LAMPIRAN B. KODE PROGRAM

B.1 Kode Program Autentifikasi (*Login*)

```

1. public function Login() {
2.     $data = array();
3.     //$this->load->view('welcome_message');
4.     if($this->input->post("Submit")) {
5.         $this->form_validation-
6.         >set_rules('username','Username','required');
7.         $this->form_validation-
8.         >set_rules('password','Password','required');
9.         if($this->form_validation->run()) {
10.            $username = $this->input->post('username');
11.            $password = $this->input->post('password');
12.            $role = $this->input->post('role');
13.
14.            $this->load->model('user_model','user');
15.            $dataUser = $this->user-
16.            >getUserLogin($role,$username,$password);
17.            $dataLogin = array($username,$password);
18.            if($dataUser->num_rows()>0) {
19.                if($this->authorization-
20.                >LoginVerification($dataUser,$dataLogin,$role)) {
21.                    if($role=="laborat") {
22.                        redirect($role.'/home');
23.                    }
24.                    else{
25.                        $data['error'] = '<div class="error_Login">Wrong
26.                        Username and Password.</div>';
27.                    }
28.                }
29.                else{
30.                    $data['error'] = '<div class="error_Login">Wrong Use
31.                    rname and Password.</div>';
32.                }
33.            }
34.            $data['error'] = validation_errors();
35.        }
36.    }
37. }

```

B.2 Kode Program Manajemen Data User

```

1. class Admin extends CI_Controller {
2.     public function __construct() {
3.         parent::__construct();
4.         $this->load->model("user_model","user");
5.         $this->authorization->isAdminLogin();
6.         if(!$this->session->userdata("rkmds-logged-in")) {

```



```
7.         redirect('/');
8.     }
9. }
10.
11. public function index(){
12.     $data['admins'] = $this->user->getDataAdmin();
13.     $this->load->view('index',$data);
14. }
15.
16. public function add(){
17.     $this->load->view('add-admin');
18.     if($this->input->post('submit')){
19.         //die('x');
20.         $name = $this->input->post("name");
21.         $alamat = $this->input->post("alamat");
22.         $telpon = $this->input->post("telpon");
23.         $role = $this->input->post("role");
24.         $username = $this->input->post("username");
25.         $password = $this->input->post("password");
26.         $data = array(
27.             "name"=>$name,
28.             "alamat"=>$alamat,
29.             "telpon"=>$telpon,
30.             "username"=>$username,
31.             "password"=>md5($password),
32.             "role"=>$role,
33.         );
34.         if($this->user->insert($data)){
35.             $this->session-
>set_flashdata('message', 'Successfully add new data. ');
36.             redirect("admin");
37.         }else{
38.             $this->session-
>set_flashdata('message', 'Failed add new data. ');
39.             redirect("admin");
40.         }
41.     }
42. }
43.
44. public function edit($id){
45.     $data = array();
46.     $user = $this->user->getUser($id);
47.     $data['user'] = $user;
48.     $this->load->view('edit-admin',$data);
49.     if($this->input->post('submit')){
50.         //die('x');
51.         $name = $this->input->post("name");
52.         $alamat = $this->input->post("alamat");
53.         $telpon = $this->input->post("telpon");
54.         $password = $this->input->post("password");
55.         $role = $this->input->post("role");
56.         $id_user = $this->input->post("id_user");
57.         $data = array(
58.             "name"=>$name,
59.             "alamat"=>$alamat,
60.             "telpon"=>$telpon,
```

```

61.         "role"=>$role,
62.     );
63.
64.     if($password!=""){
65.         $data["password"]=md5($password);
66.     }
67.     if($this->user->update($id_user,$data){
68.         $this->session-
>set_flashdata('message', 'Successfully update data. ');
69.         redirect("admin");
70.     }else{
71.         $this->session-
>set_flashdata('message', 'Failed update new data. ');
72.         redirect("admin");
73.     }
74.     }
75. }
76.
77. public function delete($id){
78.     $this->user->delete($id);
79.     $this->session-
>set_flashdata('message', 'Successfully delete data. ');
80.     redirect("admin");
81. }
82.
83. public function logout(){
84.     $user_data = array(
85.         'rkmds-id'=>',
86.         'rkmds-username'=>',
87.         'rkmds-name'=>',
88.         'rkmds-logged-in'=>FALSE,
89.         'rkmds-role'=>'
90.     );
91.     $this->session->unset_userdata($user_data);
92.     redirect("admin");
93. }
94. }

```

B.3 Kode Program Manajemen Data Pegawai

```

1. class Pegawai extends CI_Controller {
2.
3.     public function __construct(){
4.         parent::__construct();
5.         $this->load->model("pegawai_model","pegawai");
6.         // $this->authorization->isAdminLogin();
7.         if(!$this->session->userdata("rkmds-logged-in")){
8.             redirect('/');
9.         }
10.    }
11.
12.    public function index(){
13.        $data = array();
14.        $data['pegawais'] = $this->pegawai->getDatapegawai();
15.        $this->load->view('pegawai-data',$data);

```

```

16.     }
17.
18.     public function add(){
19.         if($this->input->post("submit")){
20.             $kode_pegawai = $this->input->post("kode_pegawai");
21.             $nama = $this->input->post("nama");
22.             $tempat_lahir = $this->input->post("tempat_lahir");
23.             $tanggal_lahir = $this->input->post("tanggal_lahir");
24.             $usia = $this->input->post("usia");
25.             $jenis_kelamin = $this->input->post("jenis_kelamin");
26.             $telpon = $this->input->post("telpon");
27.             $jabatan = $this->input->post("jabatan");
28.             $alamat = $this->input->post("alamat");
29.             $arrtgl = explode("/", $tanggal_lahir);
30.             $data = array(
31.                 "kode_pegawai"=>$kode_pegawai,
32.                 "nama"=>$nama,
33.                 "tempat_lahir"=>$tempat_lahir,
34.                 "tanggal_lahir"=>date("Y-m-d",strtotime($arrtgl[2]."-
". $arrtgl[1]."-". $arrtgl[0])),
35.                 "usia"=>$usia,
36.                 "jenis_kelamin"=>$jenis_kelamin,
37.                 "telpon"=>$telpon,
38.                 "jabatan"=>$jabatan,
39.                 "alamat"=>$alamat,
40.             );
41.
42.             if($this->pegawai->insert($data)){
43.                 $this->session-
>set_flashdata('message', 'Successfully add new data. ');
44.                 redirect("pegawai");
45.             }else{
46.                 $this->session-
>set_flashdata('message', 'Failed add new data. ');
47.                 redirect("pegawai");
48.             }
49.         }
50.         $id = "";
51.         $sql = 'SELECT COUNT(*) AS total FROM tb_pegawai ';
52.         $total = $this->db->query($sql)->row('total');
53.
54.         if($total==0){ //never made
55.             $autonum = "0001";
56.         }
57.         else{
58.             $sql = 'SELECT * FROM tb_pegawai ORDER BY id_pegawai DESC LI
MIT 1';
59.             $last = $this->db->query($sql)->row('kode_pegawai');
60.             $lastnum = substr($last,4,4);
61.             $autonum = (int) $lastnum + 1;
62.             $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
63.         }
64.         $id = "PGW-".$autonum;
65.         $data['id'] = $id;
66.
67.         $this->load->view('add-pegawai', $data);

```

```

68.     }
69.
70.     public function edit($id){
71.         $data = array();
72.         $pegawai = $this->pegawai->getpegawai($id);
73.         $data['pegawai'] = $pegawai;
74.         $this->load->view('edit-pegawai',$data);
75.         if($this->input->post("submit")){
76.             $nama = $this->input->post("nama");
77.             $tempat_lahir = $this->input->post("tempat_lahir");
78.             $tanggal_lahir = $this->input->post("tanggal_lahir");
79.             $usia = $this->input->post("usia");
80.             $jenis_kelamin = $this->input->post("jenis_kelamin");
81.             $telpon = $this->input->post("telpon");
82.             $jabatan = $this->input->post("jabatan");
83.             $alamat = $this->input->post("alamat");
84.             $arrtgl = explode("/", $tanggal_lahir);
85.             $id_pegawai = $this->input->post("id_pegawai");
86.             $data = array(
87.                 "nama"=>$nama,
88.                 "tempat_lahir"=>$tempat_lahir,
89.                 "tanggal_lahir"=>date("Y-m-d",strtotime($arrtgl[2]."-
". $arrtgl[1]."-". $arrtgl[0])),
90.                 "usia"=>$usia,
91.                 "jenis_kelamin"=>$jenis_kelamin,
92.                 "telpon"=>$telpon,
93.                 "jabatan"=>$jabatan,
94.                 "alamat"=>$alamat,
95.             );
96.             if($this->pegawai->update($id_pegawai,$data){
97.                 $this->session-
>set_flashdata('message', 'Successfully update data. ');
98.                 redirect("pegawai");
99.             }else{
100.                 $this->session-
>set_flashdata('message', 'Failed update new data. ');
101.                 redirect("pegawai");
102.             }
103.         }
104.     }
105.
106.     public function delete($id){
107.         $this->pegawai->delete($id);
108.         $this->session-
>set_flashdata('message', 'Successfully delete data. ');
109.         redirect("pegawai");
110.     }
111. }

```

B.4 Kode Program Manajemen Data Pasien

```

1. class Pasien extends CI_Controller{
2.     public function __construct(){
3.         parent::__construct();
4.         $this->load->model("pasien_model","pasien");

```

```

5.         // $this->authorization->isAdminLogin();
6.         if(!$this->session->userdata("rkmds-logged-in")){
7.             redirect('/');
8.         }
9.     }
10.
11.     public function index(){
12.         $data = array();
13.         $data['pasiens'] = $this->pasien->getDataPasien();
14.         $this->load->view('pasien-data',$data);
15.     }
16.
17.     public function add(){
18.         if($this->input->post("submit")){
19.             $nama = $this->input->post("nama");
20.             $telpon = $this->input->post("telpon");
21.             $umur = $this->input->post("umur");
22.             $pekerjaan = $this->input->post("pekerjaan");
23.             $alamat = $this->input->post("alamat");
24.             $jenis_kelamin = $this->input->post("jenis_kelamin");
25.             $golongan_darah = $this->input->post("golongan_darah");
26.
27.             $nama_penanggung_jawab = $this->input-
28. >post("nama_penanggung_jawab");
29.             $alamat_penanggung_jawab = $this->input-
30. >post("alamat_penanggung_jawab");
31.             $telpon_penanggung_jawab = $this->input-
32. >post("telpon_penanggung_jawab");
33.             $pekerjaan_penanggung_jawab = $this->input-
34. >post("pekerjaan_penanggung_jawab");
35.             $hubungan_penanggung_jawab = $this->input-
36. >post("hubungan_penanggung_jawab");
37.             $id_dokter = $this->input->post("id_dokter");
38.             $kode_pasien = $this->input->post('kode_pasien');
39.             $jenis_pasien = $this->input->post('jenis_pasien');
40.
41.             $data = array(
42.                 "nama"=>$nama,
43.                 "telpon"=>$telpon,
44.                 "umur"=>$umur,
45.                 "pekerjaan"=>$pekerjaan,
46.                 "alamat"=>$alamat,
47.                 "jenis_kelamin"=>$jenis_kelamin,
48.                 "golongan_darah"=>$golongan_darah,
49.                 "tanggal_daftar"=>date("Y-m-d H:i:s"),
50.                 "nama_penanggung_jawab"=>$nama_penanggung_jawab,
51.                 "alamat_penanggung_jawab"=>$alamat_penanggung_jawab,
52.                 "telpon_penanggung_jawab"=>$telpon_penanggung_jawab,
53.                 "pekerjaan_penanggung_jawab"=>$pekerjaan_penanggung_jawab,
54.                 "hubungan_penanggung_jawab"=>$hubungan_penanggung_jawab,
55.                 "id_dokter"=>$id_dokter,
56.                 "kode_pasien"=>$kode_pasien,
57.                 "jenis_pasien"=>$jenis_pasien,

```

```

54.         );
55.
56.         if($this->pasien->insert($data){
57.             $this->session-
>set_flashdata('message', 'Successfully add new data. ');
58.             redirect("pasien");
59.         }else{
60.             $this->session-
>set_flashdata('message', 'Failed add new data. ');
61.             redirect("pasien");
62.         }
63.     }
64.     $id = "";
65.     $sql = 'SELECT COUNT(*) AS total FROM tb_pasien ';
66.     $total = $this->db->query($sql)->row('total');
67.     if($total==0){ //never made
68.         $autonum = "0001";
69.     }
70.     else{
71.         $sql = 'SELECT * FROM tb_pasien ORDER BY id_pasien DESC LIMIT
T 1';
72.         $last = $this->db->query($sql)->row('kode_pasien');
73.         $lastnum = substr($last,4,4);
74.         $autonum = (int) $lastnum + 1;
75.         $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
76.     }
77.     $id = "PAS-".$autonum;
78.     $data['id'] = $id;
79.
80.     $this->load->view('add-pasien',$data);
81. }
82.
83. public function edit($id){
84.     $data = array();
85.     $pasien = $this->pasien->getPasien($id);
86.     $data['pasien'] = $pasien;
87.     $this->load->view('edit-pasien',$data);
88.     if($this->input->post("submit")){
89.         $nama = $this->input->post("nama");
90.         $telpon = $this->input->post("telpon");
91.         $umur = $this->input->post("umur");
92.         $pekerjaan = $this->input->post("pekerjaan");
93.         $alamat = $this->input->post("alamat");
94.         $jenis_kelamin = $this->input->post("jenis_kelamin");
95.         $golongan_darah = $this->input->post("golongan_darah");
96.         $id_pasien = $this->input->post("id_pasien");
97.
98.         $nama_penanggung_jawab = $this->input-
>post("nama_penanggung_jawab");
99.         $alamat_penanggung_jawab = $this->input-
>post("alamat_penanggung_jawab");
100.         $telpon_penanggung_jawab = $this->input-
>post("telpon_penanggung_jawab");
101.         $pekerjaan_penanggung_jawab = $this->input-
>post("pekerjaan_penanggung_jawab");

```

```

102.         $hubungan_penanggung_jawab = $this->input-
>post("hubungan_penanggung_jawab");
103.         $id_dokter = $this->input->post("id_dokter");
104.         $jenis_pasien = $this->input->post('jenis_pasien');
105.
106.         $data = array(
107.             "nama"=>$nama,
108.             "telpon"=>$telpon,
109.             "umur"=>$umur,
110.             "pekerjaan"=>$pekerjaan,
111.             "alamat"=>$alamat,
112.             "jenis_kelamin"=>$jenis_kelamin,
113.             "golongan_darah"=>$golongan_darah,
114.             "tanggal_daftar"=>date("Y-m-d H:i:s"),
115.
116.             "nama_penanggung_jawab"=>$nama_penanggung_jawab,
117.             "alamat_penanggung_jawab"=>$alamat_penanggung_jawab,
118.
119.             "telpon_penanggung_jawab"=>$telpon_penanggung_jawab,
120.             "pekerjaan_penanggung_jawab"=>$pekerjaan_penanggung_jawab,
121.             "hubungan_penanggung_jawab"=>$hubungan_penanggung_jawab,
122.             "id_dokter"=>$id_dokter,
123.             "jenis_pasien"=>$jenis_pasien,
124.         );
125.         if($this->pasien->update($id_pasien,$data)){
126.             $this->session-
>set_flashdata('message', 'Successfully update data. ');
127.             redirect("pasien");
128.         }else{
129.             $this->session-
>set_flashdata('message', 'Failed update new data. ');
130.             redirect("pasien");
131.         }
132.     }
133. }
134.
135.     public function delete($id){
136.         $this->pasien->delete($id);
137.         $this->session-
>set_flashdata('message', 'Successfully delete data. ');
138.         redirect("pasien");
139.     }
140. }

```

B.5 Kode Program Manajemen Data Kamar

```

1. class Kamar extends CI_Controller {
2.
3.     public function __construct(){
4.         parent::__construct();
5.         $this->load->model("kamar_model","kamar");
6.         // $this->authorization->isAdminLogin();

```

```

7.         if(!$this->session->userdata("rkmds-logged-in")){
8.             redirect('/');
9.         }
10.    }
11.
12.    public function index(){
13.        $data = array();
14.        $data['kamars'] = $this->kamar->getDataKamar();
15.        $data['count_regular'] = $this->db->where('kelas','regular')->
>get('tb_kamar')->num_rows();
16.        $data['count_vip'] = $this->db->where('kelas','vip')->
>get('tb_kamar')->num_rows();
17.        $data['count_kosong'] = $this->db->where('status','kosong')->
>get('tb_kamar')->num_rows();
18.        $data['count_terisi'] = $this->db->where('status','terisi')->
>get('tb_kamar')->num_rows();
19.        $data['count_jumlah'] = $this->db->get('tb_kamar')->
>num_rows();
20.        $this->load->view('kamar-data',$data);
21.    }
22.
23.    public function add(){
24.        if($this->input->post("submit")){
25.            $nama = $this->input->post("nama");
26.            $kelas = $this->input->post("kelas");
27.            $biaya = $this->input->post("biaya");
28.            $id_fasilitas = $this->input->post("id_fasilitas");
29.            $kode_kamar = $this->input->post("kode_kamar");
30.            $data = array(
31.                "nama"=>$nama,
32.                "kelas"=>$kelas,
33.                "biaya"=>$biaya,
34.                "kode_kamar"=>$kode_kamar,
35.            );
36.            if($this->kamar->insert($data,$id_fasilitas){
37.                $this->session->
>set_flashdata('message','Successfully add new data. ');
38.                redirect("kamar");
39.            }else{
40.                $this->session->
>set_flashdata('message','Failed add new data. ');
41.                redirect("kamar");
42.            }
43.        }
44.
45.        $id = "";
46.        $sql = 'SELECT COUNT(*) AS total FROM tb_kamar ';
47.        $total = $this->db->query($sql)->row('total');
48.
49.        if($total==0){ //never made
50.            $autonum = "0001";
51.        }
52.        else{
53.            $sql = 'SELECT * FROM tb_kamar ORDER BY id_kamar DESC LIMIT
1';
54.            $last = $this->db->query($sql)->row('kode_kamar');

```



```

55.         $lastnum = substr($last,4,4);
56.         $autonum = (int) $lastnum + 1;
57.         $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
58.     }
59.
60.     $id = "KMR-".$autonum;
61.     $data['id'] = $id;
62.
63.     $this->load->view('add-kamar',$data);
64. }
65.
66. public function edit($id){
67.     $data = array();
68.     $kamar = $this->kamar->getkamar($id);
69.     $data['kamar'] = $kamar;
70.     $this->load->view('edit-kamar',$data);
71.     if($this->input->post("submit")){
72.         $nama = $this->input->post("nama");
73.         $kelas = $this->input->post("kelas");
74.         $biaya = $this->input->post("biaya");
75.         $status = $this->input->post("status");
76.         $id_fasilitas = $this->input->post("id_fasilitas");
77.         $id_kamar = $this->input->post("id_kamar");
78.         $data = array(
79.             "nama"=>$nama,
80.             "kelas"=>$kelas,
81.             "biaya"=>$biaya,
82.             "status"=>$status,
83.         );
84.         if($this->kamar->update($id_kamar,$data,$id_fasilitas){
85.             $this->session-
86. >set_flashdata('message', 'Successfully update data. ');
87.             redirect("kamar");
88.         }else{
89.             $this->session-
90. >set_flashdata('message', 'Failed update new data. ');
91.             redirect("kamar");
92.         }
93.     }
94. public function delete($id){
95.     $this->kamar->delete($id);
96.     $this->session-
97. >set_flashdata('message', 'Successfully delete data. ');
98.     redirect("kamar");
99. }

```

B.6 Kode Program Manajemen Data Fasilitas

```

1. class Fasilitas extends CI_Controller {
2.
3.     public function __construct(){
4.         parent::__construct();

```

```
5.         $this->load->model("fasilitas_model","fasilitas");
6.         // $this->authorization->isAdminLogin();
7.         if(!$this->session->userdata("rkmds-logged-in")){
8.             redirect('/');
9.         }
10.    }
11.
12.    public function index(){
13.        $data = array();
14.        $data['fasilitass'] = $this->fasilitas->getDatafasilitas();
15.        $this->load->view('fasilitas-data',$data);
16.    }
17.
18.    public function add(){
19.        if($this->input->post("submit")){
20.            $nama = $this->input->post("nama");
21.            $jumlah = $this->input->post("jumlah");
22.            $biaya = $this->input->post("biaya");
23.            $kode_fasilitas = $this->input->post("kode_fasilitas");
24.            $data = array(
25.                "nama"=>$nama,
26.                "jumlah"=>$jumlah,
27.                "biaya"=>$biaya,
28.                "kode_fasilitas"=>$kode_fasilitas,
29.            );
30.            if($this->fasilitas->insert($data)){
31.                $this->session-
32.                >set_flashdata('message', 'Successfully add new data. ');
33.                redirect("fasilitas");
34.            }else{
35.                $this->session-
36.                >set_flashdata('message', 'Failed add new data. ');
37.                redirect("fasilitas");
38.            }
39.
40.            $id = "";
41.            $sql = 'SELECT COUNT(*) AS total FROM tb_fasilitas ';
42.            $total = $this->db->query($sql)->row('total');
43.
44.            if($total==0){ //never made
45.                $autonum = "0001";
46.            }
47.            else{
48.                $sql = 'SELECT * FROM tb_fasilitas ORDER BY id_fasilitas DES
49.                C LIMIT 1';
50.                $last = $this->db->query($sql)->row('kode_fasilitas');
51.                $lastnum = substr($last,4,4);
52.                $autonum = (int) $lastnum + 1;
53.                $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
54.            }
55.            $id = "FAS-".$autonum;
56.            $data['id'] = $id;
57.            $this->load->view('add-fasilitas',$data);
58.        }
59.    }
```

```

58.
59.     public function edit($id){
60.         $data = array();
61.         $fasilitas = $this->fasilitas->getfasilitas($id);
62.         $data['fasilitas'] = $fasilitas;
63.         $this->load->view('edit-fasilitas',$data);
64.         if($this->input->post("submit")){
65.             $nama = $this->input->post("nama");
66.             $jumlah = $this->input->post("jumlah");
67.             $biaya = $this->input->post("biaya");
68.             $id_fasilitas = $this->input->post("id_fasilitas");
69.             $data = array(
70.                 "nama"=>$nama,
71.                 "jumlah"=>$jumlah,
72.                 "biaya"=>$biaya,
73.             );
74.
75.             if($this->fasilitas->update($id_fasilitas,$data)){
76.                 $this->session-
77. >set_flashdata('message', 'Successfully update data. ');
78.                 redirect("fasilitas");
79.             }else{
80.                 $this->session-
81. >set_flashdata('message', 'Failed update new data. ');
82.                 redirect("fasilitas");
83.             }
84.         }
85.     public function delete($id){
86.         $this->fasilitas->delete($id);
87.         $this->session-
88. >set_flashdata('message', 'Successfully delete data. ');
89.         redirect("fasilitas");
90.     }

```

B.7 Kode Program Manajemen Data Dokter

```

1. class Dokter extends CI_Controller {
2.
3.     public function __construct(){
4.         parent::__construct();
5.         $this->load->model("dokter_model","dokter");
6.         // $this->authorization->isAdminLogin();
7.         if(!$this->session->userdata("rkmds-logged-in")){
8.             redirect('/');
9.         }
10.    }
11.
12.    public function index(){
13.        $data = array();
14.        $data['dokter'] = $this->dokter->getDataDokter();
15.        $this->load->view('dokter-data',$data);
16.    }

```

```

17.
18.     public function add(){
19.         if($this->input->post("submit")){
20.             $nama = $this->input->post("nama");
21.             $tempat_lahir = $this->input->post("tempat_lahir");
22.             $tanggal_lahir = $this->input->post("tanggal_lahir");
23.             $usia = $this->input->post("usia");
24.             $jenis_kelamin = $this->input->post("jenis_kelamin");
25.             $telpon = $this->input->post("telpon");
26.             $spesialis = $this->input->post("spesialis");
27.             $alamat = $this->input->post("alamat");
28.             $arrtgl = explode("/", $tanggal_lahir);
29.             $kode_dokter = $this->input->post('kode_dokter');
30.             $data = array(
31.                 "nama"=>$nama,
32.                 "tempat_lahir"=>$tempat_lahir,
33.                 "tanggal_lahir"=>date("Y-m-d",strtotime($arrtgl[2]."-
". $arrtgl[1]."-". $arrtgl[0])),
34.                 "usia"=>$usia,
35.                 "jenis_kelamin"=>$jenis_kelamin,
36.                 "telpon"=>$telpon,
37.                 "spesialis"=>$spesialis,
38.                 "alamat"=>$alamat,
39.                 "kode_dokter"=>$kode_dokter,
40.             );
41.
42.             if($this->dokter->insert($data)){
43.                 $this->session-
>set_flashdata('message', 'Successfully add new data. ');
44.                 redirect("dokter");
45.             }else{
46.                 $this->session-
>set_flashdata('message', 'Failed add new data. ');
47.                 redirect("dokter");
48.             }
49.         }
50.         $id = "";
51.         $sql = 'SELECT COUNT(*) AS total FROM tb_dokter ';
52.         $total = $this->db->query($sql)->row('total');
53.
54.         if($total==0){ //never made
55.             $autonum = "0001";
56.         }
57.         else{
58.             $sql = 'SELECT * FROM tb_dokter ORDER BY id_dokter DESC LIM
T 1';
59.             $last = $this->db->query($sql)->row('kode_dokter');
60.             $lastnum = substr($last,4,4);
61.             $autonum = (int) $lastnum + 1;
62.             $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
63.         }
64.         $id = "DOK-".$autonum;
65.         $data['id'] = $id;
66.
67.         $this->load->view('add-dokter',$data);
68.     }

```

```

69.
70.     public function edit($id){
71.         $data = array();
72.         $dokter = $this->dokter->getDokter($id);
73.         $data['dokter'] = $dokter;
74.         $this->load->view('edit-dokter', $data);
75.         if($this->input->post("submit")){
76.             $nama = $this->input->post("nama");
77.             $tempat_lahir = $this->input->post("tempat_lahir");
78.             $tanggal_lahir = $this->input->post("tanggal_lahir");
79.             $usia = $this->input->post("usia");
80.             $jenis_kelamin = $this->input->post("jenis_kelamin");
81.             $telpon = $this->input->post("telpon");
82.             $spesialis = $this->input->post("spesialis");
83.             $alamat = $this->input->post("alamat");
84.             $arrtgl = explode("/", $tanggal_lahir);
85.             $id_dokter = $this->input->post("id_dokter");
86.             $data = array(
87.                 "nama"=>$nama,
88.                 "tempat_lahir"=>$tempat_lahir,
89.                 "tanggal_lahir"=>date("Y-m-d", strtotime($arrtgl[2]."-
". $arrtgl[1]."-". $arrtgl[0])),
90.                 "usia"=>$usia,
91.                 "jenis_kelamin"=>$jenis_kelamin,
92.                 "telpon"=>$telpon,
93.                 "spesialis"=>$spesialis,
94.                 "alamat"=>$alamat,
95.             );
96.
97.             if($this->dokter->update($id_dokter, $data)){
98.                 $this->session-
99.                 >set_flashdata('message', 'Successfully update data. ');
100.                 redirect("dokter");
101.             }else{
102.                 $this->session-
103.                 >set_flashdata('message', 'Failed update new data. ');
104.                 redirect("dokter");
105.             }
106.
107.     public function delete($id){
108.         $this->dokter->delete($id);
109.         $this->session-
110.         >set_flashdata('message', 'Successfully delete data. ');
111.         redirect("dokter");
112.     }

```

B.8 Kode Program Manajemen Data Obat

```

1. class Obat extends CI_Controller {
2.
3.     public function __construct(){
4.         parent::__construct();

```

```

5.         $this->load->model("obat_model","obat");
6.         // $this->authorization->isAdminLogin();
7.         if(!$this->session->userdata("rkmds-logged-in")){
8.             redirect('/');
9.         }
10.    }
11.
12.    public function index(){
13.        $data = array();
14.        $data['obats'] = $this->obat->getDataobat();
15.        $this->load->view('obat-data',$data);
16.    }
17.
18.    public function add(){
19.        if($this->input->post("submit")){
20.            $nama = $this->input->post("nama");
21.            $jenis = $this->input->post("jenis");
22.            $satuan = $this->input->post("satuan");
23.            $stok = $this->input->post("stok");
24.            $harga_pokok = $this->input->post("harga_pokok");
25.            $harga_jual = $this->input->post("harga_jual");
26.            $kode_obat = $this->input->post("kode_obat");
27.            $data = array(
28.                "nama"=>$nama,
29.                "jenis"=>$jenis,
30.                "satuan"=>$satuan,
31.                "stok"=>$stok,
32.                "harga_pokok"=>$harga_pokok,
33.                "harga_jual"=>$harga_jual,
34.                "kode_obat"=>$kode_obat,
35.            );
36.
37.            if($this->obat->insert($data)){
38.                $this->session-
39.                >set_flashdata('message', 'Successfully add new data. ');
40.                redirect("obat");
41.            }else{
42.                $this->session-
43.                >set_flashdata('message', 'Failed add new data. ');
44.                redirect("obat");
45.            }
46.
47.            $id = "";
48.            $sql = 'SELECT COUNT(*) AS total FROM tb_obat ';
49.            $total = $this->db->query($sql)->row('total');
50.
51.            if($total==0){ //never made
52.                $autonum = "0001";
53.            }
54.            else{
55.                $sql = 'SELECT * FROM tb_obat ORDER BY id_obat DESC LIMIT 1'
56.                ;
57.                $last = $this->db->query($sql)->row('kode_obat');
58.                $lastnum = substr($last,4,4);
59.                $autonum = (int) $lastnum + 1;

```

```

58.         $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
59.     }
60.     $sid = "OBT-".$autonum;
61.     $data['id'] = $sid;
62.
63.     $this->load->view('add-obat',$data);
64. }
65.
66.     public function edit($id){
67.         $data = array();
68.         $obat = $this->obat->getobat($id);
69.         $data['obat'] = $obat;
70.         $this->load->view('edit-obat',$data);
71.         if($this->input->post("submit")){
72.             $nama = $this->input->post("nama");
73.             $jenis = $this->input->post("jenis");
74.             $satuan = $this->input->post("satuan");
75.             $stok = $this->input->post("stok");
76.             $harga_pokok = $this->input->post("harga_pokok");
77.             $harga_jual = $this->input->post("harga_jual");
78.             $id_obat = $this->input->post("id_obat");
79.             $data = array(
80.                 "nama"=>$nama,
81.                 "jenis"=>$jenis,
82.                 "satuan"=>$satuan,
83.                 "stok"=>$stok,
84.                 "harga_pokok"=>$harga_pokok,
85.                 "harga_jual"=>$harga_jual,
86.             );
87.
88.             if($this->obat->update($id_obat,$data)){
89.                 $this->session-
90. >set_flashdata('message', 'Successfully update data. ');
91.                 redirect("obat");
92.             }else{
93.                 $this->session-
94. >set_flashdata('message', 'Failed update new data. ');
95.                 redirect("obat");
96.             }
97.         }
98.     public function delete($id){
99.         $this->obat->delete($id);
100.         $this->session-
101. >set_flashdata('message', 'Successfully delete data. ');
102.         redirect("obat");
103.     }

```

B.9 Kode Program Manajemen Data Laborat

```

1. class Laborat extends CI_Controller {
2.
3.     public function __construct(){

```

```

4.     parent::__construct();
5.     $this->load->model("laborat_model","laborat");
6.     // $this->authorization->isAdminLogin();
7.     if(!$this->session->userdata("rkmds-logged-in")){
8.         redirect('/');
9.     }
10.    }
11.
12.    public function home(){
13.        $this->load->view('index');
14.    }
15.
16.    public function index(){
17.        $data = array();
18.        $data['laborats'] = $this->laborat->getDataLaborat();
19.        $this->load->view('laborat-data',$data);
20.    }
21.
22.    public function add(){
23.        if($this->input->post("submit")){
24.            $nama = $this->input->post("nama");
25.            $jenis = $this->input->post("jenis");
26.            $biaya = $this->input->post("biaya");
27.            $keperluan = $this->input->post("keperluan");
28.            $id_pasien = $this->input->post("id_pasien");
29.            $anjuran_dari = $this->input->post("anjuran_dari");
30.            $hasil_pemeriksaan = $this->input-
>post("hasil_pemeriksaan");
31.            $kode_laborat = $this->input->post("kode_laborat");
32.            //$hasil_pemeriksaan = "";
33.            $hasil = $this->input->post("hasil");
34.            //$hasil = $_POST["hasil"];
35.
36.            $data = array(
37.                "nama"=>$nama,
38.                "jenis"=>$jenis,
39.                "biaya"=>$biaya,
40.                "keperluan"=>$keperluan,
41.                "id_pasien"=>$id_pasien,
42.                "hasil_pemeriksaan"=>$hasil_pemeriksaan,
43.                "kode_laborat"=>$kode_laborat,
44.                "anjuran_dari"=>$anjuran_dari,
45.            );
46.
47.            if($this->laborat->insert($data,$hasil)){
48.                $this->session-
>set_flashdata('message', 'Successfully add new data. ');
49.                redirect("laborat");
50.            }else{
51.                $this->session-
>set_flashdata('message', 'Failed add new data. ');
52.                redirect("laborat");
53.            }
54.        }
55.
56.        $id = "";

```



```

57.     $sql = 'SELECT COUNT(*) AS total FROM tb_laborat ';
58.     $total = $this->db->query($sql)->row('total');
59.
60.     if($total==0){ //never made
61.         $autonum = "0001";
62.     }
63.     else{
64.         $sql = 'SELECT * FROM tb_laborat ORDER BY id_laborat DESC LI
MIT 1';
65.         $last = $this->db->query($sql)->row('kode_laborat');
66.         $lastnum = substr($last,4,4);
67.         $autonum = (int) $lastnum + 1;
68.         $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
69.     }
70.     $sid = "LBR-".$autonum;
71.     $data['id'] = $sid;
72.     $data['pemeriksaan']=$this->db->get('tb_pemeriksaan')-
>result_array();
73.
74.     $this->load->view('add-laborat',$data);
75. }
76.
77. public function delete($id){
78.     $this->laborat->delete($id);
79.     $this->session-
>set_flashdata('message', 'Successfully delete data. ');
80.     redirect("laborat");
81. }
82.
83. public function logout(){
84.     $user_data = array(
85.         'rkmds-id'=>',
86.         'rkmds-username'=>',
87.         'rkmds-name'=>',
88.         'rkmds-logged-in'=>FALSE,
89.         'rkmds-role'=>'
90.     );
91.     $this->session->unset_userdata($user_data);
92.     redirect('admin');
93. }
94.
95. function pdf($id_laborat){
96.     //$this->load->helper('pdf_helper');
97.     //$this->load->library('pdfku');
98.     $data= array();
99.     $data['laborat'] = $this->db
100.         ->where("id_laborat",$id_laborat)
101.         -
>select('tb_laborat.*, tb_pasien.nama as nama_pasien, tb_pasien.kode_pas
ien', false)
102.         -
>join('tb_pasien', 'tb_laborat.id_pasien = tb_pasien.id_pasien')
103.         ->get("tb_laborat");
104.
105.     $data['laborat_detail'] = $this->db
106.         ->where("tb_laborat_detail.id_laborat",$id_laborat)

```

```

107.         -
>select('tb_pemeriksaan.*,tb_laborat_detail.hasil', false)
108.         -
>join('tb_pemeriksaan', 'tb_laborat_detail.id_pemeriksaan = tb_pemeriksaan.id_pemeriksaan')
109.         ->get("tb_laborat_detail");
110.         $this->load->view('pdf/laborat', $data);
111.     }
112. }

```

B.10 Kode Program Manajemen Data Rekam Medis

```

1. class Rekammedis extends CI_Controller {
2.
3.     public function __construct(){
4.         parent::__construct();
5.         $this->load->model("rekammedis_model","rekammedis");
6.         // $this->authorization->isAdminLogin();
7.         if(!$this->session->userdata("rkmds-logged-in")){
8.             redirect('/');
9.         }
10.    }
11.
12.    public function index(){
13.        $data = array();
14.        $data['rekammedis'] = $this->rekammedis->getDataRekammedis();
15.        $this->load->view('rekammedis-data',$data);
16.    }
17.
18.    public function add(){
19.        if($this->input->post("submit")){
20.            $id_pasien = $this->input->post("id_pasien");
21.            $diagnosa = $this->input->post("diagnosa");
22.            $hasil_pemeriksaan = $this->input->post("hasil_pemeriksaan");
23.            $obat_dianjurkan = $this->input->post("obat_dianjurkan");
24.            $id_dokter = $this->input->post("id_dokter");
25.            $jenis_pemeriksaan = $this->input->post("jenis_pemeriksaan");
26.            $tanggal_masuk = $this->input->post("tanggal_masuk");
27.            $tanggal_pemeriksaan = $this->input->post("tanggal_pemeriksaan");
28.            $anamnesis = $this->input->post("anamnesis");
29.            $sarrtgl = explode("/", $tanggal_masuk);
30.            $sarrtglpemeriksaan = explode("/", $tanggal_pemeriksaan);
31.            $kode_rekammedis = $this->input->post('kode_rekammedis');
32.
33.            $data = array(
34.                "id_pasien"=>$id_pasien,
35.                "diagnosa"=>$diagnosa,
36.                "hasil_pemeriksaan"=>$hasil_pemeriksaan,
37.                "obat_dianjurkan"=>$obat_dianjurkan,
38.                "id_dokter"=>$id_dokter,
39.                "jenis_pemeriksaan"=>$jenis_pemeriksaan,

```

```

40.         "tanggal_masuk"=>date("Y-m-d",strtotime($arrtgl[2]."-
". $arrtgl[1]."-". $arrtgl[0])),
41.         "tanggal_pemeriksaan"=>date("Y-m-
d",strtotime($arrtgpemeriksaan[2]."-". $arrtgpemeriksaan[1]."-
". $arrtgpemeriksaan[0])),
42.         "anamnesis"=>$anamnesis,
43.         "kode_rekammedis"=>$kode_rekammedis,
44.     );
45.
46.     if($this->rekammedis->insert($data){
47.         $this->session-
>set_flashdata('message', 'Successfully add new data. ');
48.         redirect("rekammedis");
49.     }else{
50.         $this->session-
>set_flashdata('message', 'Failed add new data. ');
51.         redirect("rekammedis");
52.     }
53. }
54. $id = "";
55. $sql = 'SELECT COUNT(*) AS total FROM tb_rekammedis ';
56. $total = $this->db->query($sql)->row('total');
57.
58. if($total==0){ //never made
59.     $autonum = "0001";
60. }
61. else{
62.     $sql = 'SELECT * FROM tb_rekammedis ORDER BY id_rekammedis D
ESC LIMIT 1';
63.     $last = $this->db->query($sql)->row('kode_rekammedis');
64.     $lastnum = substr($last,4,4);
65.     $autonum = (int) $lastnum + 1;
66.     $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
67. }
68. $id = "RKM-".$autonum;
69. $data['id'] = $id;
70.
71. $this->load->view('add-rekammedis', $data);
72. }
73.
74. public function edit($id){
75.     $data = array();
76.     $rekammedis = $this->rekammedis->getRekammedis($id);
77.     $data['rekammedis'] = $rekammedis;
78.     $this->load->view('edit-rekammedis', $data);
79.     if($this->input->post("submit")){
80.         $id_pasien = $this->input->post("id_pasien");
81.         $diagnosa = $this->input->post("diagnosa");
82.         $hasil_pemeriksaan = $this->input-
>post("hasil_pemeriksaan");
83.         $obat_dianjurkan = $this->input->post("obat_dianjurkan");
84.         $id_dokter = $this->input->post("id_dokter");
85.         $jenis_pemeriksaan = $this->input-
>post("jenis_pemeriksaan");
86.         $tanggal_masuk = $this->input->post("tanggal_masuk");

```

```

87.         $tanggal_pemeriksaan = $this->input-
>post("tanggal_pemeriksaan");
88.         $anamnesis = $this->input->post("anamnesis");
89.         $arrtgl = explode("/", $tanggal_masuk);
90.         $id_rekammedis = $this->input->post("id_rekammedis");
91.         $arrtgl = explode("/", $tanggal_masuk);
92.         $arrtglpemeriksaan = explode("/", $tanggal_pemeriksaan);
93.
94.         $data = array(
95.             "id_pasien"=>$id_pasien,
96.             "diagnosa"=>$diagnosa,
97.             "hasil_pemeriksaan"=>$hasil_pemeriksaan,
98.             "obat_dianjurkan"=>$obat_dianjurkan,
99.             "id_dokter"=>$id_dokter,
100.            "jenis_pemeriksaan"=>$jenis_pemeriksaan,
101.            "tanggal_masuk"=>date("Y-m-d", strtotime($arrtgl[2]."-
".$arrtgl[1]."-".$arrtgl[0])),
102.            "tanggal_pemeriksaan"=>date("Y-m-
d", strtotime($arrtglpemeriksaan[2]."-".$arrtglpemeriksaan[1]."-
".$arrtglpemeriksaan[0])),
103.            "anamnesis"=>$anamnesis,
104.        );
105.
106.        if($this->rekammedis->update($id_rekammedis,$data)){
107.            $this->session-
>set_flashdata('message', 'Successfully update data. ');
108.            redirect("rekammedis");
109.        }else{
110.            $this->session-
>set_flashdata('message', 'Failed update data. ');
111.            redirect("rekammedis");
112.        }
113.    }
114. }
115.
116.     public function delete($id){
117.         $this->rekammedis->delete($id);
118.         $this->session-
>set_flashdata('message', 'Successfully delete data. ');
119.         redirect("rekammedis");
120.     }
121.
122.     function pdf($id_rekammedis){
123.         //$this->load->helper('pdf_helper');
124.         $data= array();
125.         $data['rekammedis'] = $this->db
->where("id_rekammedis",$id_rekammedis)
126.         -
127.         >select('tb_rekammedis.*, tb_pasien.kode_pasien,tb_pasien.nama as nama_p
asien, tb_dokter.nama as nama_dokter', false)
128.         -
>join('tb_pasien', 'tb_rekammedis.id_pasien = tb_pasien.id_pasien')
129.         -
>join('tb_dokter', 'tb_rekammedis.id_dokter = tb_dokter.id_dokter')
130.         ->get("tb_rekammedis")->row_array();
131.         $this->load->view('pdf/rekammedis', $data);

```

```

132.     }
133. }

```

B.11 Kode Program Manajemen Data Rujukan

```

1.  class Rujukan extends CI_Controller {
2.
3.      public function __construct() {
4.          parent::__construct();
5.          $this->load->model("rujukan_model", "rujukan");
6.          if (!$this->session->userdata("rkmds-logged-in")) {
7.              redirect('/');
8.          }
9.      }
10.
11.     public function index() {
12.         $data = array();
13.         $data['rujukans'] = $this->rujukan->getDatarujukan();
14.         $this->load->view('rujukan-data', $data);
15.     }
16.
17.     public function edit($id_rujukan) {
18.         $data = array();
19.         $rujukan = $this->rujukan->getrujukan($id_rujukan);
20.         $data['rujukan'] = $rujukan;
21.         $this->load->view('edit-rujukan', $data);
22.     }
23.
24.     public function add() {
25.         if ($this->input->post("submit")) {
26.             $no_transaksi = $this->input->post("no_transaksi");
27.             $tanggal_transaksi = $this->input->
>post("tanggal_transaksi");
28.             $kode_pasien = $this->input->post("kode_pasien");
29.             $nama = $this->input->post("nama");
30.             $id_dokter = $this->input->post("id_dokter");
31.             $tujuan = $this->input->post("tujuan");
32.             $keterangan = $this->input->post("keterangan");
33.
34.             $data = array(
35.                 "no_transaksi"=>$no_transaksi,
36.                 "tanggal_transaksi"=>date("Y-m-d H:i:s"),
37.                 "kode_pasien"=>$kode_pasien,
38.                 "nama"=>$nama,
39.                 "id_dokter"=>$id_dokter,
40.                 "tujuan"=>$tujuan,
41.                 "keterangan"=>$keterangan,
42.             );
43.
44.             if ($this->rujukan->insert($data)) {
45.                 $this->session->
>set_flashdata('message', 'Successfully add new data. ');
46.                 redirect("rujukan");
47.             } else {

```

```

48.         $this->session-
>set_flashdata('message', 'Failed add new data.');
```

```

49.         redirect("rujukan");
50.     }
51. }
52.     $id = "";
53.     $sql = 'SELECT COUNT(*) AS total FROM tb_rujukan ';
54.     $total = $this->db->query($sql)->row('total');
```

```

55.
56.     if($total==0){ //never made
57.         $autonum = "0001";
58.     }
59.     else{
60.         $sql = 'SELECT * FROM tb_rujukan ORDER BY id_rujukan DES
C LIMIT 1';
61.         $last = $this->db->query($sql)->row('no_transaksi');
```

```

62.         $lastnum = substr($last,4,4);
63.         $autonum = (int) $lastnum + 1;
64.         $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
65.     }
66.     $id = "RJK-".$autonum;
67.     $data['id'] = $id;
68.     $this->load->view('add-rujukan',$data);
69. }
70.
71. public function insert(){
72.     if($this->input->post("submit")){
73.         $no_transaksi = $this->input->post("no_transaksi");
74.         $id_pasien = $this->input->post("id_pasien");
75.         $id_dokter = $this->input->post("id_dokter");
76.         $tujuan = $this->input->post("tujuan");
77.         $keterangan = $this->input->post("keterangan");
78.
79.         $data = array(
80.             "no_transaksi"=>$no_transaksi,
81.             "tanggal_transaksi"=>date("Y-m-d H:i:s"),
82.             "id_pasien"=>$id_pasien,
83.             "id_dokter"=>$id_dokter,
84.             "tujuan"=>$tujuan,
85.             "keterangan"=>$keterangan,
86.         );
87.
88.         if($this->rujukan->insert($data)){
89.             $this->session-
>set_flashdata('message', 'Successfully add new data.');
```

```

90.             redirect("rujukan");
91.         }else{
92.             $this->session-
>set_flashdata('message', 'Failed add new data.');
```

```

93.             redirect("rujukan");
94.         }
95.     }
96. }
97.
98. public function update(){
99.     if($this->input->post("submit")){
```

```

100.         $no_transaksi = $this->input->post("no_transaksi");
101.         $id_pasien = $this->input->post("id_pasien");
102.         $id_dokter = $this->input->post("id_dokter");
103.         $tujuan = $this->input->post("tujuan");
104.         $keterangan = $this->input->post("keterangan");
105.         $id_rujukan = $this->input->post("id_rujukan");
106.
107.         $data = array(
108.             "no_transaksi"=>$no_transaksi,
109.             "id_pasien"=>$id_pasien,
110.             "id_dokter"=>$id_dokter,
111.             "tujuan"=>$tujuan,
112.             "keterangan"=>$keterangan,
113.         );
114.         if($this->rujukan->update($id_rujukan, $data)){
115.             $this->session-
116.             >set_flashdata('message', 'Successfully update data. ');
117.             redirect("rujukan");
118.         }else{
119.             $this->session-
120.             >set_flashdata('message', 'Failed update new data. ');
121.             redirect("rujukan");
122.         }
123.
124.         public function delete($id_rujukan){
125.             $this->rujukan->delete($id_rujukan);
126.             $this->session-
127.             >set_flashdata('message', 'Successfully delete data. ');
128.             redirect("rujukan");
129.         }
130.         private function generateNomorTransaksi(){
131.             return time()."";
132.         }
133.
134.         function pdf($id_rujukan){
135.             $data= array();
136.             $data['rujukan'] = $this->db
137.             ->where("id_rujukan",$id_rujukan)
138.             -
139.             >select('tb_rujukan.no_transaksi, tb_rujukan.tanggal_transaksi, tb_pasie
140.             n.id_pasien,tb_pasien.kode_pasien, tb_pasien.nama as nama_pasien, tb_dok
141.             ter.nama as nama_dokter, tb_rujukan.tujuan, tb_rujukan.keterangan', fals
142.             e)
143.             -
144.             >join('tb_pasien', 'tb_rujukan.id_pasien = tb_pasien.id_pasien')
145.             -
146.             >join('tb_dokter', 'tb_rujukan.id_dokter = tb_dokter.id_dokter')
147.             ->get("tb_rujukan");
148.             $data['rekammedis'] = $this->db
149.             ->where("id_pasien",$data['rujukan']->row()->id_pasien)
150.             //-
151.             >select('amamnesis, diagnosa, hasil_pemeriksaan, jenis_pemeriksaan')
152.             ->get("tb_rekammedis");

```

```

146.         $this->load->view('pdf/rujukan', $data);
147.     }
148. }

```

B.12 Kode Program Manajemen Data Transaksi Pembayaran Pasien

```

1. class Transaksi extends CI_Controller {
2.
3.     public function __construct(){
4.         parent::__construct();
5.         $this->load->model("transaksi_model","transaksi");
6.         if(!$this->session->userdata("rkmds-logged-in")){
7.             redirect('/');
8.         }
9.     }
10.
11.    public function index(){
12.        $this->load->view('transaksi-index');
13.    }
14.
15.    public function rawatinap(){
16.        if($this->input->post("submit")){
17.            $no_transaksi = $this->generateNomorTransaksi();
18.            $tanggal_transaksi = $this->input-
>post("tanggal_transaksi");
19.            $id_pasien = $this->input->post("id_pasien");
20.            $id_kamar = $this->input->post("id_kamar");
21.            $tanggal_masuk = $this->input->post("tanggal_masuk");
22.            $tanggal_keluar = $this->input->post("tanggal_keluar");
23.            $lama_inap = $this->input->post("lama_inap");
24.            $keterangan = $this->input->post("keterangan");
25.            $biaya_dokter = $this->input->post("biaya_dokter");
26.            $biaya_visit_dokter = $this->input-
>post("biaya_visit_dokter");
27.            $biaya_perawatan = $this->input->post("biaya_perawatan");
28.            $biaya_kamar = $this->input->post("biaya_kamar");
29.            $biaya_obat = $this->input->post("biaya_obat");
30.            $biaya_laboratorium = $this->input-
>post("biaya_laboratorium");
31.            $nama_biaya_fasilitas_lain = $this->input-
>post("nama_biaya_fasilitas_lain");
32.            $biaya_fasilitas_lain = $this->input-
>post("biaya_fasilitas_lain");
33.            $arr_tgl_transaksi = explode("/", $tanggal_transaksi);
34.            $arr_tgl_masuk = explode("/", $tanggal_masuk);
35.            $arr_tgl_keluar = explode("/", $tanggal_keluar);
36.
37.            $data = array(
38.                "no_transaksi"=>$no_transaksi,
39.                "tanggal_transaksi"=>date("Y-m-
d",strtotime($arr_tgl_transaksi[2]."-".$arr_tgl_transaksi[1]."-
".$arr_tgl_transaksi[0])),
40.                "id_kamar"=>$id_kamar,
41.                "id_pasien"=>$id_pasien,

```



```

42.         "tanggal_masuk"=>date("Y-m-
d",strtotime($arr_tgl_masuk[2]."-".$arr_tgl_masuk[1]."-
".$arr_tgl_masuk[0])),
43.         "tanggal_keluar"=>date("Y-m-
d",strtotime($arr_tgl_keluar[2]."-".$arr_tgl_keluar[1]."-
".$arr_tgl_keluar[0])),
44.         "lama_inap"=>$lama_inap,
45.         "keterangan"=>$keterangan,
46.         "biaya_dokter"=>$biaya_dokter,
47.         "biaya_visit_dokter"=>$biaya_visit_dokter,
48.         "biaya_perawatan"=>$biaya_perawatan,
49.         "biaya_kamar"=>$biaya_kamar,
50.         "biaya_obat"=>$biaya_obat,
51.         "biaya_laboratorium"=>$biaya_laboratorium,
52.         "nama_biaya_fasilitas_lain"=>$nama_biaya_fasilitas_lain,
53.         "biaya_fasilitas_lain"=>$biaya_fasilitas_lain,
54.     );
55.
56.     if($id_transaksi = $this->transaksi-
>insertTransaksiRawatInap($data)){
57.         redirect('transaksi/pdf/rawatinap/'.$id_transaksi);
58.     }
59. }
60. $this->load->view('transaksi-rawatinap');
61. }
62.
63. public function pdf(){
64.     $type = $this->uri->segment(3);
65.     $id = $this->uri->segment(4);
66.
67.     if($type=="rawatinap"){
68.         $data= array();
69.         $data['transaksi'] = $this->db
70.             ->where("id_transaksi",$id)
71.             -
>select('tb_transaksi_rawatinap.*, tb_pasien.nama as nama_pasien, tb_kam
ar.nama as nama_kamar', false)
72.             -
>join('tb_pasien', 'tb_transaksi_rawatinap.id_pasien = tb_pasien.id_pasi
en')
73.             -
>join('tb_kamar', 'tb_transaksi_rawatinap.id_kamar = tb_kamar.id_kamar')
74.             ->get("tb_transaksi_rawatinap");
75.         $this->load->view('pdf/transaksi/rawatinap', $data);
76.     }
77.     else if($type=="laboratorium"){
78.         $data= array();
79.         $data['transaksi'] = $this->db
80.             ->where("id_transaksi",$id)
81.             -
>select('tb_transaksi_laboratorium.*, tb_pasien.nama as nama_pasien', fa
lse)

```

```

82.         -
83.     >join('tb_pasien', 'tb_transaksi_laboratorium.id_pasien = tb_pasien.id_p
      pasien')
84.         ->get("tb_transaksi_laboratorium");
85.         $data['transaction_details'] = $this->db
86.         ->where("id_transaksi",$id)
87.         ->get("tb_transaksi_laboratorium_detail");
88.         $this->load->view('pdf/transaksi/laboratorium', $data);
89.     }
90.     else if($type=="obat"){
91.         $data= array();
92.         $data['transaksi'] = $this->db
93.         ->where("id_transaksi",$id)
94.         -
95.         >select('tb_transaksi_obat.*, tb_pasien.nama as nama_pasien', false)
96.         -
97.         >join('tb_pasien', 'tb_transaksi_obat.id_pasien = tb_pasien.id_pasien')
98.         ->get("tb_transaksi_obat");
99.         $data['transaction_details'] = $this->db
100.        ->where("id_transaksi",$id)
101.        -
102.        >select('tb_transaksi_obat_detail.*, tb_obat.nama as nama_obat, tb_obat.
      harga_jual', false)
103.        -
104.        >join('tb_obat', 'tb_transaksi_obat_detail.id_obat = tb_obat.id_obat')
105.        ->get("tb_transaksi_obat_detail");
106.        $this->load->view('pdf/transaksi/obat', $data);
107.    }
108.    }
109.    public function obat(){
110.        if($this->input->post("submit")){
111.            $no_transaksi = $this->generateNomorTransaksi();
112.            $tanggal_transaksi = $this->input-
113.            >post("tanggal_transaksi");
114.            $id_pasien = $this->input->post("id_pasien");
115.            $cart_id_obat = $this->input->post("cart_id_obat");
116.            $cart_qty = $this->input->post("cart_qty");
117.            $arr_tgl_transaksi = explode("/", $tanggal_transaksi);
118.            $data = array(
119.                "no_transaksi"=>$no_transaksi,
120.                "tanggal_transaksi"=>date("Y-m-
121.                d",strtotime($arr_tgl_transaksi[2]."-". $arr_tgl_transaksi[1]."-
122.                ".$arr_tgl_transaksi[0])),
123.                "id_pasien"=>$id_pasien,
124.            );
125.            if($id_transaksi = $this->transaksi-
126.            >insertTransaksiObat($data,$cart_id_obat,$cart_qty)){
127.                redirect('transaksi/pdf/obat/'.$id_transaksi);
128.            }
129.        }
130.        $this->load->view('transaksi-obat');
131.    }
132.    }

```

```

126.     public function laboratorium(){
127.         if($this->input->post("submit")){
128.             $no_transaksi = $this->generateNomorTransaksi();
129.             $tanggal_transaksi = $this->input-
>post("tanggal_transaksi");
130.             $id_pasien = $this->input->post("id_pasien");
131.             $cart_jenis_pemeriksaan = $this->input-
>post("jenis_pemeriksaan"); //array
132.             $cart_biaya_pemeriksaan = $this->input-
>post("biaya_pemeriksaan"); //array
133.             $arr_tgl_transaksi = explode("/", $tanggal_transaksi);
134.
135.             $data = array(
136.                 "no_transaksi"=>$no_transaksi,
137.                 "tanggal_transaksi"=>date("Y-m-
d", strtotime($arr_tgl_transaksi[2]."-". $arr_tgl_transaksi[1]."-
". $arr_tgl_transaksi[0])),
138.                 "id_pasien"=>$id_pasien,
139.             );
140.             if($id_transaksi = $this->transaksi-
>insertTransaksiLaboratorium($data,$cart_jenis_pemeriksaan,$cart_biaya_p
emeriksaan)){
141.                 //pdf laborat
142.                 redirect('transaksi/pdf/laboratorium/'.$id_transaksi)
;
143.             }
144.
145.         }
146.         $this->load->view('transaksi-laboratorium');
147.     }
148.
149.     public function getkamar(){
150.         $id_kamar = $this->input->post("id_kamar");
151.         $kamar = $this->db->where("id_kamar",$id_kamar)-
>get("tb_kamar")->row();
152.         $kamar = array(
153.             "id_kamar"=>$kamar->id_kamar,
154.             "nama"=>$kamar->nama,
155.             "kelas"=>$kamar->kelas,
156.             "biaya"=>($kamar->biaya),
157.             "status"=>$kamar->status,
158.         );
159.         echo json_encode($kamar);
160.     }
161.
162.     public function getobat(){
163.         $id_obat = $this->input->post("id_obat");
164.         $obat = $this->db->where("id_obat",$id_obat)->get("tb_obat")-
>row();
165.         $obat = array(
166.             "id_obat"=>$obat->id_obat,
167.             "nama"=>$obat->nama,
168.             "jenis"=>$obat->jenis,
169.             "satuan"=>$obat->satuan,
170.             "stok"=>$obat->stok,
171.             "harga_pokok"=>$obat->harga_pokok,

```

```

172.         "harga_jual"=>$obat->harga_jual,
173.         );
174.         echo json_encode($obat);
175.     }
176.
177.     public function gettanggalmasuk(){
178.         $id_pasien = $this->input->post("id_pasien");
179.         $pasien = $this->db->where("id_pasien",$id_pasien)-
>get("tb_pasien")->row();
180.         $pasien = array(
181.             "tanggal_daftar"=>date('d/m/Y',strtotime($pasien-
>tanggal_daftar)),
182.         );
183.         echo json_encode($pasien);
184.     }
185.
186.     private function generateNomorTransaksi(){
187.         return time()."";
188.     }
189. }

```

B.13 Kode Program Melihat Laporan

```

1. class Laporan extends CI_Controller {
2.
3.     public function __construct(){
4.         parent::__construct();
5.         // $this->authorization->isAdminLogin();
6.         if(!$this->session->userdata("rkmds-logged-in")){
7.             redirect('/');
8.         }
9.     }
10.
11.     public function index(){
12.         $this->load->view('laporan-index');
13.     }
14.
15.     public function pasien(){
16.         $this->load->model("pasien_model","pasien");
17.         $data['pasiens'] = $this->pasien->getDataPasien();
18.         $this->load->view("laporan-pasien",$data);
19.     }
20.
21.     public function kamar(){
22.         $this->load->model("kamar_model","kamar");
23.         $data['kamars'] = $this->kamar->getDataKamar();
24.         $this->load->view("laporan-kamar",$data);
25.     }
26.
27.     public function dokter(){
28.         $this->load->model("dokter_model","dokter");
29.         $data['dokters'] = $this->dokter->getDataDokter();
30.         $this->load->view("laporan-dokter",$data);
31.     }
32.

```

```
33.     public function pegawai(){
34.         $this->load->model("pegawai_model","pegawai");
35.         $data['pegawais'] = $this->pegawai->getDataPegawai();
36.         $this->load->view("laporan-pegawai",$data);
37.     }
38.
39.     public function fasilitas(){
40.         $this->load->model("fasilitas_model","fasilitas");
41.         $data['fasilitass'] = $this->fasilitas->
>getDataFasilitas();
42.         $this->load->view("laporan-fasilitas",$data);
43.     }
44.
45.     public function obat(){
46.         $this->load->model("obat_model","obat");
47.         $data['obats'] = $this->obat->getDataObat();
48.         $this->load->view("laporan-obat",$data);
49.     }
50.
51.     public function laboratorium(){
52.         $this->load->model("laborat_model","laborat");
53.         $data['laborats'] = $this->laborat->getDataLaborat();
54.         $this->load->view("laporan-laborat",$data);
55.     }
56.
57.     public function rekammedis(){
58.         $this->load->model("rekammedis_model","rekammedis");
59.         $data['rekammediss'] = $this->rekammedis->getDataRekammedis();
60.         $this->load->view("laporan-rekammedis",$data);
61.     }
62.
63.     public function rujukan(){
64.         $this->load->model("rujukan_model","rujukan");
65.         $data['rujukans'] = $this->rujukan->getDataRujukan();
66.         $this->load->view("laporan-rujukan",$data);
67.     }
68.
69.     public function getkamar(){
70.         $id_kamar = $this->input->post("id_kamar");
71.         $kamar = $this->db->where("id_kamar",$id_kamar)->
>get("tb_kamar")->row();
72.         $kamar = array(
73.             "id_kamar"=>$kamar->id_kamar,
74.             "nama"=>$kamar->nama,
75.             "kelas"=>$kamar->kelas,
76.             "biaya"=>number_format($kamar->biaya),
77.             "status"=>$kamar->status,
78.         );
79.         echo json_encode($kamar);
80.     }
81.
82.     public function getobat(){
83.         $id_obat = $this->input->post("id_obat");
84.         $obat = $this->db->where("id_obat",$id_obat)->get("tb_obat")->
>row();
85.         $obat = array(
```

```
86.         "id_obat"=>$obat->id_obat,
87.         "nama"=>$obat->nama,
88.         "jenis"=>$obat->jenis,
89.         "satuan"=>$obat->satuan,
90.         "stok"=>$obat->stok,
91.         "harga_pokok"=>$obat->harga_pokok,
92.         "harga_jual"=>$obat->harga_jual,
93.     );
94.     echo json_encode($obat);
95. }
96.
97. private function generateNomorTransaksi(){
98.     return time()."";
99. }
100.
101. function pdf($table){
102.     //$this->load->helper('pdf_helper');
103.     //$this->load->library('mypdf');
104.
105.     $_table = $this->getTable()[$table];
106.     $data = array();
107.     if($_table=="tb_pasien"){
108.         $sql = "select a.*,b.nama as nama_dokter from tb_pasien a
109.
110.         left join tb_dokter b on a.id_dokter = b.id_dokter";
111.         $data['data'] = $this->db->query($sql);
112.     }
113.     else{
114.         $data['data'] = $this->db->get($_table);
115.     }
116.     //$this->mypdf->subtitle = $_table;
117.
118.     //var_dump($data);exit;
119.     $this->load->view('pdf/laporan/'.$table, $data);
120. }
121. private function getTable(){
122.     return array(
123.         "pasien" => "tb_pasien",
124.         "kamar" => "tb_kamar",
125.         "dokter" => "tb_dokter",
126.         "pegawai" => "tb_pegawai",
127.         "fasilitas" => "tb_fasilitas",
128.         "obat" => "tb_obat",
129.         "laboratorium" => "tb_laborat",
130.         "rekammedis" => "tb_rekammedis",
131.         "rujukan" => "tb_rujukan",
132.         "laborat" => array("tb_laborat"),
133.     );
134. }
135. }
```

LAMPIRAN C. WHITEBOX TESTING

C.1 Manajemen Data User

Pengujian kode-kode program pada fitur Manajemen Data *User* adalah sebagai berikut:

1. Listing Program

```

1. 1 class Admin extends CI_Controller {
2.     public function __construct() { 2
3.         parent::__construct();
4.         3 $this->load->model("user_model", "user");
5.         $this->authorization->isAdminLogin();
6.         if(!$this->session->userdata("rkmds-logged-in")) { 4
7.             redirect('/'); 5
8.         } 6
9.     } 7
10.
11.     public function index() {
12.         $data['admins'] = $this->user->getDataAdmin(); 10
13.         $this->load->view('index', $data);
14.     } 11
15.
16.     public function add() {
17.         12 $this->load->view('add-admin'); 13
18.         14 if($this->input->post('submit')) {
19.             //die('x');
20.             $name = $this->input->post("name");
21.             $alamat = $this->input->post("alamat");
22.             $telpon = $this->input->post("telpon");
23.             $role = $this->input->post("role");
24.             15 $username = $this->input->post("username");
25.             $password = $this->input->post("password");
26.             $data = array(
27.                 "name"=>$name,
28.                 "alamat"=>$alamat,
29.                 "telpon"=>$telpon,
30.                 "username"=>$username,
31.                 "password"=>md5($password),
32.                 "role"=>$role,
33.             );
34.             16 if($this->user->insert($data)) {
35.                 $this->session-
36.                 >set_flashdata('message', 'Successfully add new data. '); 17
37.                 redirect("admin");
38.             } 18 else {
39.                 18 $this->session-
40.                 >set_flashdata('message', 'Failed add new data. ');
41.                 redirect("admin");
42.             } 19
43.         } 20
44.     } 21

```

```

45. $data = array();
46. 22 $user = $this->user->getUser($id);
47. $data['user'] = $user;
48. $this->load->view('edit-admin', $data);
49. 23 if ($this->input->post('submit')) {
50.     //die('x');
51.     $name = $this->input->post("name");
52.     $alamat = $this->input->post("alamat");
53.     $telpon = $this->input->post("telpon");
54.     $password = $this->input->post("password");
55. 24 $role = $this->input->post("role");
56.     $id_user = $this->input->post("id_user");
57.     $data = array(
58.         "name"=>$name,
59.         "alamat"=>$alamat,
60.         "telpon"=>$telpon,
61.         "role"=>$role,
62.     );
63.
64. 25 if ($password!="") {
65.     $data["password"]=md5($password);
66. } 27
67.     if ($this->user->update($id_user, $data)) { 28
68.         $this->session-
>set_flashdata('message', 'Successfully update data.'). 29
        redirect("admin");
69.     } else {
70. 30 $this->session-
>set_flashdata('message', 'Failed update new data.').
        redirect("admin");
71.     }
72. }
73. } 31
74.
75. 32}
76.
77. public function delete($id) { 33
78.     $this->user->delete($id);
79.     $this->session-
>set_flashdata('message', 'Successfully delete data.').
80.     redirect("admin");
81. } 35
82.
83. public function logout() { 36
84.     $user_data = array(
85. 37 'rkmds-id'=>',
86.     'rkmds-username'=>',
87.     'rkmds-name'=>',
88.     'rkmds-logged-in'=>FALSE,
89.     'rkmds-role'=>'
90.     );
91.     $this->session->unset_userdata($user_data);
92.     redirect("admin");
93. } 38
94. } 39

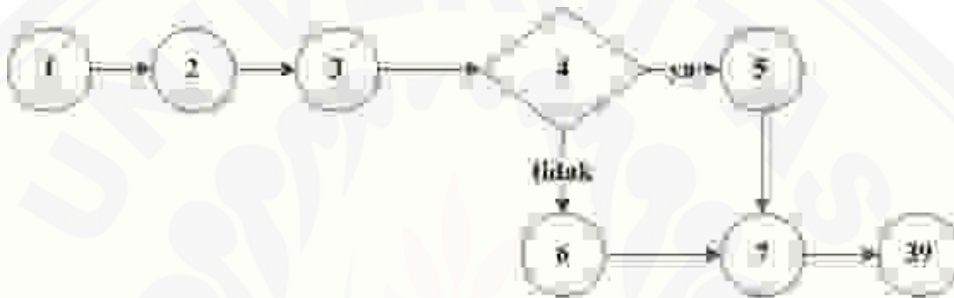
```


2. Grafik Alir

Grafik Alir fitur Manajemen Data *User* berupa notasi – notasi yang menunjukkan alur kontrol. Fitur Manajemen Data *User* terbagi menjadi beberapa fungsi, yaitu *construct*, *index*, *add*, *edit*, *delete*, dan *logout*.

2.1. Grafik Alir Fitur Manajemen Data *User* Fungsi *Construct*

Grafik alir fitur Manajemen Data *User* Fungsi *Construct* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.1.



Gambar C.1 Grafik Alir Manajemen Data *User* Fungsi *Construct*

Sumber: (Hasil analisis)

2.2. Grafik Alir Fitur Manajemen Data *User* Fungsi *Index*

Grafik alir fitur Manajemen Data *User* Fungsi *Index* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.2.

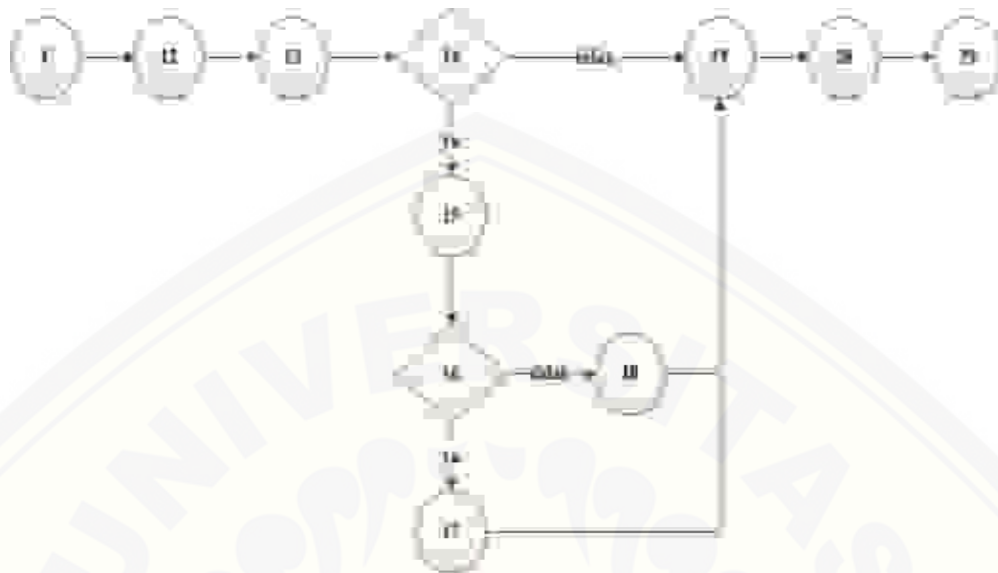


Gambar C.2 Grafik Alir Manajemen Data *User* Fungsi *Index*

Sumber: (Hasil analisis)

2.3. Grafik Alir Fitur Manajemen Data *User* Fungsi *Add*

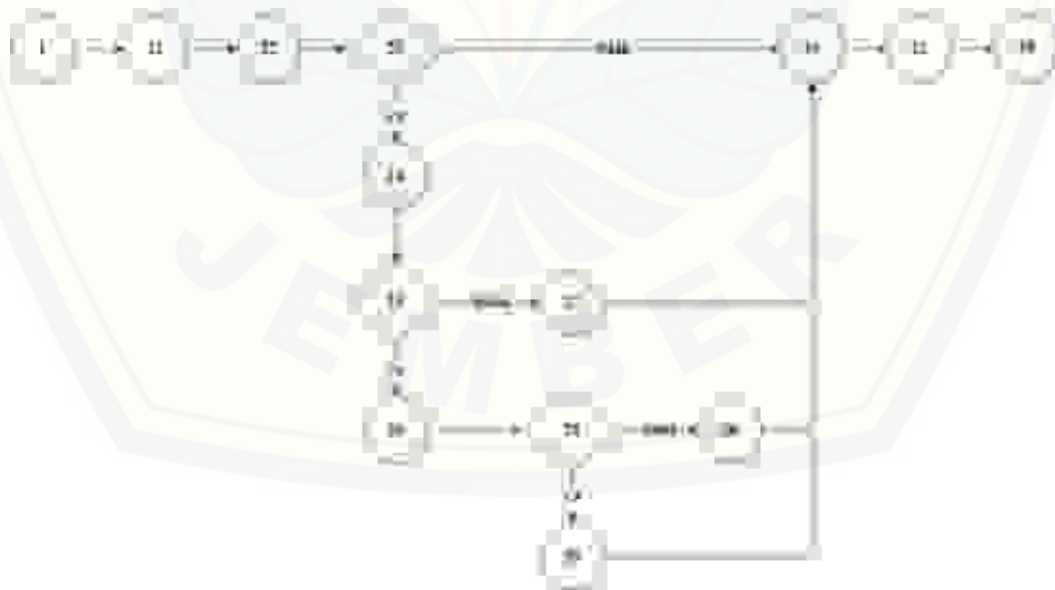
Grafik alir fitur Manajemen Data *User* Fungsi *Add* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.3.



Gambar C.3 Grafik Alir Manajemen Data *User Fungsi Add*
 Sumber: (Hasil analisis)

2.4. Grafik Alir Fitur Manajemen Data *User Fungsi Edit*

Grafik alir fitur Manajemen Data *User Fungsi Edit* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.4.



Gambar C. 4 Diagram Alir Manajemen Data *User Fungsi Edit*
 Sumber: (Hasil analisis)

2.5. Grafik Alir Fitur Manajemen Data *User* Fungsi *Delete*

Grafik alir fitur Manajemen Data *User* Fungsi *Delete* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.5.



Gambar C. 5 Grafik Alir Manajemen Data *User* Fungsi *Delete*
 Sumber: (Hasil analisis)

2.6. Grafik Alir Fitur Manajemen Data *User* Fungsi *Logout*

Grafik alir fitur Manajemen Data *User* Fungsi *Logout* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.6.



Gambar C. 6 Grafik Alir Manajemen Data *User* Fungsi *Logout*
 Sumber: (Hasil analisis)

3. Kompleksitas Siklomatis

Kompleksitas siklomatis dari fitur Manajemen Data *User* dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2$$

Dengan $V(G)$ adalah Kompleksitas siklomatis, E adalah *Edge* atau tepi atau garis dari sebuah grafik, dan N adalah *Node* atau titik dari sebuah grafik. Maka perhitungan dari setiap grafik dari Manajemen Data *User* adalah sebagai berikut.

3.1. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data *User* Fungsi

Construct

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 8 - 8 + 2 \\
 &= 2
 \end{aligned}$$

3.2. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data *User* Fungsi *Index*

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 4 - 5 + 2 \\
 &= 1
 \end{aligned}$$

3.3. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data *User Fungsi Add*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 12 - 11 + 2 \\ &= 3 \end{aligned}$$

3.4. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data *User Fungsi Edit*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 16 - 14 + 2 \\ &= 4 \end{aligned}$$

3.5. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data *User Fungsi**Delete*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

3.6. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data *User fungsi logout*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

4. *Basis Set*4.1. *Basis set* Fitur Manajemen Data *User Fungsi Construct*

Basis set Manajemen Data *User* fungsi *construct* menghasilkan dua (2) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 5 – 7 – 39

Jalur 2 : 1 – 2 – 3 – 4 – 6 – 7 – 39

4.2. *Basis set* Fitur Manajemen Data *User Fungsi Index*

Basis set Manajemen Data *User* fungsi *index* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 8 – 9 – 10 – 39

4.3. *Basis set* Fitur Manajemen Data *User Fungsi Add*

Basis set Manajemen Data *User* fungsi *Add* menghasilkan tiga (3) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 12 – 13 – 14 – 19 – 20 – 39

Jalur 2 : 1 – 12 – 13 – 14 – 15 – 16 – 18 – 19 – 20 – 39

Jalur 3 : 1 – 12 – 13 – 14 – 15 – 16 – 17 – 19 – 20 – 39

4.4. *Basis set* Fitur Manajemen Data User Fungsi *Edit*

Basis set Manajemen Data User fungsi *Edit* menghasilkan empat (4) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 21 – 22 – 23 – 31 – 32 – 39

Jalur 2 : 1 – 21 – 22 – 23 – 24 – 25 – 27 – 31 – 32 – 39

Jalur 3 : 1 – 21 – 22 – 23 – 24 – 25 – 26 – 28 – 30 – 31 – 32 – 39

Jalur 4 : 1 – 21 – 22 – 23 – 24 – 25 – 26 – 28 – 29 – 31 – 32 – 39

4.5. *Basis set* Fitur Manajemen Data User Fungsi *Delete*

Basis set Manajemen Data User fungsi *Delete* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 33 – 34 – 35 – 39

4.6. *Basis set* Fitur Manajemen Data User Fungsi *Logout*

Basis set Manajemen Data User fungsi *Logout* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 36 – 37 – 38 – 39

5. *Test Case*

Tabel C. 1 Test Case Fitur Manajemen Data User fungsi construct jalur 1
Sumber: (Hasil analisis)

Test case	User berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data User aktif
Target yang diharapkan	Sistem menampilkan halaman <i>Dashboard</i> atau Data User aktif
Hasil pengujian	Benar
Jalur	Jalur 1
File	admin.php

Tabel C. 2 *Test Case* Fitur Manajemen Data *User* fungsi *construct* jalur 2
Sumber: (Hasil analisis)

Test case	<i>User</i> melakukan <i>Logout</i> dan menuju halaman <i>Login</i> dan ketika di <i>back</i> tetap menampilkan halaman <i>Login</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	admin.php

Tabel C. 3 *Test Case* Fitur Manajemen Data *User* fungsi *Index* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif atau klik menu <i>Dashboard</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Dashboard</i> atau Data <i>User</i> aktif
Hasil pengujian	Benar
Jalur	Jalur 1
File	admin.php

Tabel C. 4 *Test Case* Fitur Manajemen Data *User* fungsi *Add* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> mengklik <i>Tambah User</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Tambah User</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	admin.php

Tabel C. 5 *Test Case* Fitur Manajemen Data *User* fungsi *Add* jalur 2
Sumber: (Hasil analisis)

Test case	<i>User</i> mengisi <i>form</i> <i>tambah user</i> kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Dashboard</i> dan menampilkan <i>alert,</i> ” <i>Failed add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 2
File	admin.php

Tabel C. 6 *Test Case* Fitur Manajemen Data *User* fungsi *Add* jalur 3
Sumber: (Hasil analisis)

Test case	<i>User</i> mengisi <i>form</i> tambah <i>user</i> kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Dashboard</i> dan menampilkan <i>alert</i> ,” <i>Successfully add new data..</i> ”
Hasil pengujian	Benar
Jalur	Jalur 3
File	admin.php

Tabel C. 7 *Test Case* Fitur Manajemen Data *User* fungsi *Edit* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> mengklik tombol <i>edit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Edit User</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	admin.php

Tabel C. 8 *Test Case* Fitur Manajemen Data *User* fungsi *Edit* jalur 2
Sumber: (Hasil analisis)

Test case	<i>User</i> mengisi <i>form edit user</i> kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Dashboard</i> dan menampilkan <i>alert</i> ,” <i>Successfully update data..</i> ”
Hasil pengujian	Benar
Jalur	Jalur 2
File	admin.php

Tabel C. 9 *Test Case* Fitur Manajemen Data *User* fungsi *Edit* jalur 3
Sumber: (Hasil analisis)

Test case	<i>User</i> mengisi <i>form edit user</i> kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Dashboard</i> dan menampilkan <i>alert</i> ,” <i>Failed update data..</i> ”
Hasil pengujian	Benar
Jalur	Jalur 3
File	admin.php

Tabel C. 10 *Test Case* Fitur Manajemen Data *User* fungsi *Edit* jalur 4
Sumber: (Hasil analisis)

Test case	<i>User</i> mengisi <i>form edit user</i> kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Dashboard</i> dan menampilkan <i>alert,</i> ” <i>Successfully update data..</i> ”
Hasil pengujian	Benar
Jalur	Jalur 4
File	admin.php

Tabel C. 11 *Test Case* Fitur Manajemen Data *User* fungsi *Delete* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> mengklik tombol <i>Remove</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Dashboard</i> dan menampilkan <i>alert,</i> ” <i>Successfully delete data..</i> ”
Hasil pengujian	Benar
Jalur	Jalur 1
File	admin.php

Tabel C. 12 *Test Case* Fitur Manajemen Data *User* fungsi *Logout* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> mengklik tombol <i>Logout</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	admin.php

C.2 Manajemen Data Pegawai

Pengujian kode-kode program pada fitur Manajemen Data Pegawai adalah sebagai berikut:

1. Listing Program

```

1. 1 class Pegawai extends CI_Controller {
2.
3.   2 public function __construct() {
4.     parent::__construct();
5.     $this->load->model("pegawai_model", "pegawai"); 3
6.     // $this->authorization->isAdminLogin();
7.     if (!$this->session->userdata("rkmds-logged-in")) { 4
8.       redirect('/', 5); 5

```



```

9.
10.     } 7 } 6
11.
12.     public function index() { 8
13.         $data = array();
14.         9 $data['pegawais'] = $this->pegawai->getDatapegawai();
15.         $this->load->view('pegawai-data', $data);
16.     } 10
17.
18.     public function add() { 11
19.         12 if ($this->input->post("submit")) {
20.             $kode_pegawai = $this->input->post("kode_pegawai");
21.             $nama = $this->input->post("nama");
22.             $tempat_lahir = $this->input->post("tempat_lahir");
23.             $tanggal_lahir = $this->input->post("tanggal_lahir");
24.             $usia = $this->input->post("usia");
25.             $jenis_kelamin = $this->input->post("jenis_kelamin");
26.             $telpon = $this->input->post("telpon");
27.             $jabatan = $this->input->post("jabatan");
28.             $alamat = $this->input->post("alamat");
29.             $arrtgl = explode("/", $tanggal_lahir);
30.             13 $data = array(
31.                 "kode_pegawai"=>$kode_pegawai,
32.                 "nama"=>$nama,
33.                 "tempat_lahir"=>$tempat_lahir,
34.                 "tanggal_lahir"=>date("Y-m-d", strtotime($arrtgl[2]."-
35.                 ".$arrtgl[1]."-".$arrtgl[0])),
36.                 "usia"=>$usia,
37.                 "jenis_kelamin"=>$jenis_kelamin,
38.                 "telpon"=>$telpon,
39.                 "jabatan"=>$jabatan,
40.                 "alamat"=>$alamat,
41.             );
42.             14 if ($this->pegawai->insert($data)) {
43.                 $this->session-
44.                 >set_flashdata('message', 'Successfully add new data.'; 15
45.                 redirect("pegawai");
46.             } else {
47.                 16 $this->session-
48.                 >set_flashdata('message', 'Failed add new data.');
```

```

61.         $autonum = (int) $lastnum + 1;
62.         $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
63.     }
64.     $sid = "PGW-".$autonum;
65.     $data['id'] = $sid;
66.
67.     $this->load->view('add-pegawai', $data);
68. }
69.
70. public function edit($id) {
71.     $data = array();
72.     $pegawai = $this->pegawai->getpegawai($id);
73.     $data['pegawai'] = $pegawai;
74.     $this->load->view('edit-pegawai', $data);
75.     if($this->input->post("submit")) {
76.         $nama = $this->input->post("nama");
77.         $tempat_lahir = $this->input->post("tempat_lahir");
78.         $tanggal_lahir = $this->input->post("tanggal_lahir");
79.         $usia = $this->input->post("usia");
80.         $jenis_kelamin = $this->input->post("jenis_kelamin");
81.         $telpon = $this->input->post("telpon");
82.         $jabatan = $this->input->post("jabatan");
83.         $alamat = $this->input->post("alamat");
84.         $arrtgl = explode("/", $tanggal_lahir);
85.         $id_pegawai = $this->input->post("id_pegawai");
86.         $data = array(
87.             "nama"=>$nama,
88.             "tempat_lahir"=>$tempat_lahir,
89.             "tanggal_lahir"=>date("Y-m-d", strtotime($arrtgl[2]."-
90.             ".$arrtgl[1]."-".$arrtgl[0])),
91.             "usia"=>$usia,
92.             "jenis_kelamin"=>$jenis_kelamin,
93.             "telpon"=>$telpon,
94.             "jabatan"=>$jabatan,
95.             "alamat"=>$alamat,
96.         );
97.         if($this->pegawai->update($id_pegawai, $data) {
98.             $this->session-
99.             >set_flashdata('message', 'Successfully update data. ');
100.            redirect("pegawai");
101.        } else {
102.            $this->session-
103.            >set_flashdata('message', 'Failed update new data. ');
104.            redirect("pegawai");
105.        }
106.    }
107.    public function delete($id) {
108.        $this->pegawai->delete($id);
109.        $this->session-
110.        >set_flashdata('message', 'Successfully delete data. ');
111.        redirect("pegawai");
112.    }
113. }

```

2. Grafik Alir

Grafik Alir fitur Manajemen Data Pegawai berupa notasi-notasi yang menunjukkan alur kontrol. Fitur Manajemen Data Pegawai terbagi menjadi beberapa fungsi, yaitu *construct*, *index*, *add*, *edit*, dan *delete*.

2.1. Grafik Alir Fitur Manajemen Data Pegawai Fungsi *Construct*

Grafik alir fitur Manajemen Data Pegawai Fungsi *Construct* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.7.



Gambar C. 7 Grafik Alir Manajemen Data Pegawai Fungsi *Construct*

Sumber: (Hasil analisis)

2.2. Grafik Alir Fitur Manajemen Data Pegawai Fungsi *Index*

Grafik alir fitur Manajemen Data Pegawai Fungsi *Index* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.8.



Gambar C. 8 Grafik Alir Manajemen Data Pegawai Fungsi *Index*

Sumber: (Hasil analisis)

2.3. Grafik Alir Fitur Manajemen Data Pegawai Fungsi *Add*

Grafik alir fitur Manajemen Data Pegawai Fungsi *Add* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.9.

2.5. Grafik Alir Fitur Manajemen Data Pegawai Fungsi *Delete*

Grafik alir fitur Manajemen Data Pegawai Fungsi *Delete* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.11.



Gambar C. 11 Grafik Alir Manajemen Data Pegawai Fungsi *Delete*
Sumber: (Hasil analisis)

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur Manajemen Data Pegawai dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2$$

Dengan $V(G)$ adalah Kompleksitas siklomatis, E adalah *Edge* atau tepi atau garis dari sebuah grafik, dan N adalah *Node* atau titik dari sebuah grafik. Maka perhitungan dari setiap grafik dari Manajemen Data Pegawai adalah sebagai berikut.

3.1. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Pegawai Fungsi *construct*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 8 - 8 + 2 \\ &= 2 \end{aligned}$$

3.2. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Pegawai Fungsi *Index*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

3.3. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Pegawai Fungsi *Add*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 17 - 15 + 2 \\ &= 4 \end{aligned}$$

3.4. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Pegawai Fungsi *Edit*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 12 - 11 + 2 \\ &= 3 \end{aligned}$$

3.5. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Pegawai Fungsi

Delete

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

4. Basis Set

4.1. Basis set Fitur Manajemen Data Pegawai Fungsi *Construct*

Basis set Manajemen Data Pegawai fungsi *construct* menghasilkan dua (2) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 6 – 7 – 36

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 7 – 36

4.2. Basis set Fitur Manajemen Data Pegawai Fungsi *Index*

Basis set Manajemen Data Pegawai fungsi *index* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 8 – 9 – 10 – 36

4.3. Basis set Fitur Manajemen Data Pegawai Fungsi *Add*

Basis set Manajemen Data Pegawai fungsi *Add* menghasilkan empat (4) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 11 – 18 – 19 – 20 – 22 – 12 – 13 – 14 – 15 – 23 – 36

Jalur 2 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 13 – 14 – 15 – 23 – 36

Jalur 3 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 13 – 14 – 16 – 23 – 36

Jalur 4 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 17 – 23 – 36

4.4. Basis set Fitur Manajemen Data Pegawai Fungsi *Edit*

Basis set Manajemen Data Pegawai fungsi *Edit* menghasilkan tiga (3) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 24 – 25 – 26 – 31 – 32 – 36

Jalur 2 : 1 – 24 – 25 – 26 – 27 – 28 – 30 – 32 – 36

Jalur 3 : 1 – 24 – 25 – 26 – 27 – 28 – 29 – 32 – 36

4.5. Basis set Fitur Manajemen Data Pegawai Fungsi *Delete*

Basis set Manajemen Data Pegawai fungsi *Delete* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 33 – 34 – 35 – 36

5. Test Case

Tabel C. 13 *Test Case* Fitur Manajemen Data Pegawai fungsi *construct* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif kemudian klik menu Pegawai
Target yang diharapkan	Sistem menampilkan halaman Pegawai
Hasil pengujian	Benar
Jalur	Jalur 1
File	pegawai.php

Tabel C. 14 *Test Case* Fitur Manajemen Data Pegawai fungsi *construct* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Pegawai, <i>User</i> melakukan <i>Logout</i> dan menuju halaman <i>Login</i> dan ketika di <i>back</i> tetap menampilkan halaman <i>Login</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	pegawai.php

Tabel C. 15 *Test Case* Fitur Manajemen Data Pegawai fungsi *Index* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif atau klik menu Pegawai
Target yang diharapkan	Sistem menampilkan halaman Pegawai
Hasil pengujian	Benar

Jalur	Jalur 1
File	pegawai.php

Tabel C. 16 *Test Case* Fitur Manajemen Data Pegawai fungsi Add jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Pegawai, <i>User</i> mengklik Tombol Tambah Pegawai kemudian mengisi <i>form</i> Tambah Pegawai setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Pegawai serta menampilkan <i>alert, "Successfully add new data."</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	pegawai.php

Tabel C. 17 *Test Case* Fitur Manajemen Data Pegawai fungsi Add jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Pegawai, <i>User</i> mengklik Tombol Tambah Pegawai kemudian mengisi <i>form</i> Tambah Pegawai setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Pegawai serta menampilkan <i>alert, "Successfully add new data."</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	pegawai.php

Tabel C. 18 *Test Case* Fitur Manajemen Data Pegawai fungsi Add jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Pegawai, <i>User</i> mengklik Tombol Tambah Pegawai kemudian mengisi <i>form</i> Tambah Pegawai setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Pegawai dan menampilkan <i>alert, "Failed add new data."</i>

Hasil pengujian	Benar
Jalur	Jalur 3
File	pegawai.php

Tabel C. 19 *Test Case* Fitur Manajemen Data Pegawai fungsi *Add* jalur 4
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Pegawai, <i>User</i> mengklik Tombol Tambah Pegawai
Target yang diharapkan	Sistem menampilkan halaman Tambah Pegawai
Hasil pengujian	Benar
Jalur	Jalur 4
File	pegawai.php

Tabel C. 20 *Test Case* Fitur Manajemen Data Pegawai fungsi *Edit* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Pegawai, <i>User</i> mengklik Tombol <i>Edit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Edit</i> Pegawai
Hasil pengujian	Benar
Jalur	Jalur 1
File	pegawai.php

Tabel C. 21 *Test Case* Fitur Manajemen Data Pegawai fungsi *Edit* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Pegawai, <i>User</i> mengisi <i>form edit</i> Pegawai kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Pegawai dan menampilkan <i>alert, "Failed update data.."</i>
Hasil pengujian	Benar
Jalur	Jalur
File	pegawai.php

Tabel C. 22 *Test Case* Fitur Manajemen Data Pegawai fungsi *Edit* jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Pegawai,
-----------	---

	<i>User</i> mengisi <i>form edit</i> Pegawai kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Pegawai dan menampilkan <i>alert, "Successfully update data.."</i>
Hasil pengujian	Benar
Jalur	Jalur 3
File	pegawai.php

Tabel C. 23 Test Case Fitur Manajemen Data Pegawai fungsi *Delete* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Pegawai, <i>User</i> mengklik tombol <i>Remove</i>
Target yang diharapkan	Sistem menampilkan halaman Pegawai dan menampilkan <i>alert, "Successfully delete data.."</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	pegawai.php

C. 3 Manajemen Data Pasien

Pengujian kode – kode program pada fitur Manajemen Data Pasien adalah sebagai berikut:

1. Listing Program

```

1. 1 class Pasien extends CI_Controller {
2.     public function __construct() { 2
3.         parent::__construct();
4.         3 $this->load->model("pasien_model", "pasien");
5.         // $this->authorization->isAdminLogin();
6.         if (!$this->session->userdata("rkmds-logged-in")) { 4
7.             redirect('/'); 5
8.         } 6
9.     } 7
10.
11.     public function index() { 8
12.         $data = array();
13.         $data['pasien'] = $this->pasien->getDataPasien(); 9
14.         $this->load->view('pasien-data', $data);
15.     } 10
16.
17.     public function add() { 11
18.         if ($this->input->post("submit")) { 12
19.             $nama = $this->input->post("nama");

```

```

20.         $telpon = $this->input->post("telpon");
21.         $umur = $this->input->post("umur");
22.         $pekerjaan = $this->input->post("pekerjaan");
23.         $alamat = $this->input->post("alamat");
24.         $jenis_kelamin = $this->input->post("jenis_kelamin");
25.         $golongan_darah = $this->input->post("golongan_darah");
26.
27.         $nama_penanggung_jawab = $this->input-
>post("nama_penanggung_jawab");
28.         $alamat_penanggung_jawab = $this->input-
>post("alamat_penanggung_jawab");
29.         $telpon_penanggung_jawab = $this->input-
>post("telpon_penanggung_jawab");
30.         $pekerjaan_penanggung_jawab = $this->input-
>post("pekerjaan_penanggung_jawab");
31.         $hubungan_penanggung_jawab = $this->input-
>post("hubungan_penanggung_jawab");
32.         $id_dokter = $this->input->post("id_dokter");
33.         $kode_pasien = $this->input->post('kode_pasien');
34.         $jenis_pasien = $this->input->post('jenis_pasien');
35.
36.         $data = array(
37.             "nama"=>$nama,
38.             "telpon"=>$telpon,
39.             "umur"=>$umur,
40.             "pekerjaan"=>$pekerjaan,
41.             "alamat"=>$alamat,
42.             "jenis_kelamin"=>$jenis_kelamin,
43.             "golongan_darah"=>$golongan_darah,
44.             "tanggal_daftar"=>date("Y-m-d H:i:s"),
45.
46.             "nama_penanggung_jawab"=>$nama_penanggung_jawab,
47.             "alamat_penanggung_jawab"=>$alamat_penanggung_jawab,
48.             "telpon_penanggung_jawab"=>$telpon_penanggung_jawab,
49.             "pekerjaan_penanggung_jawab"=>$pekerjaan_penanggung_jawa
b,
50.             "hubungan_penanggung_jawab"=>$hubungan_penanggung_jawab,
51.
52.             "id_dokter"=>$id_dokter,
53.             "kode_pasien"=>$kode_pasien,
54.             "jenis_pasien"=>$jenis_pasien,
55.         );
56.         14 if($this->pasien->insert($data)){
57.             $this->session-
>set_flashdata('message', 'Successfully add new data. '); 15
58.             redirect("pasien");
59.         } 16 else{
60.             $this->session-
>set_flashdata('message', 'Failed add new data. ');
61.             redirect("pasien");
62.         }
63.     } 17
64.     $id = "";
65.     $sql = 'SELECT COUNT(*) AS total FROM tb_pasien ' 18
66.     $total = $this->db->query($sql)->row('total'); 19

```

```

67.         if($total==0){ //never made
68.             $autonum = "0001";20
69.         }
70.         else{
71.             21 $sql = 'SELECT * FROM tb_pasien ORDER BY id_pasien DESC LIMIT
T 1';
72.             $last = $this->db->query($sql)->row('kode_pasien');
73.             $lastnum = substr($last,4,4);
74.             $autonum = (int) $lastnum + 1;
75.             $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
76.         }
77.         $id = "PAS-".$autonum;
78.         22 $data['id'] = $id;
79.
80.         $this->load->view('add-pasien',$data);
81.     23 }
82.
83.     public function edit($id){24
84.         $data = array();
85.         25 $pasien = $this->pasien->getPasien($id);
86.         $data['pasien'] = $pasien;
87.         $this->load->view('edit-pasien',$data);
88.         26 if($this->input->post("submit")){
89.             $nama = $this->input->post("nama");
90.             $telpon = $this->input->post("telpon");
91.             $umur = $this->input->post("umur");
92.             $pekerjaan = $this->input->post("pekerjaan");
93.             $alamat = $this->input->post("alamat");
94.             $jenis_kelamin = $this->input->post("jenis_kelamin");
95.             $golongan_darah = $this->input->post("golongan_darah");
96.             $id_pasien = $this->input->post("id_pasien");
97.
98.             $nama_penanggung_jawab = $this->input-
>post("nama_penanggung_jawab");
99.             $alamat_penanggung_jawab = $this->input-
>post("alamat_penanggung_jawab");
100.             $telpon_penanggung_jawab = $this->input-
>post("telpon_penanggung_jawab");
101.             $pekerjaan_penanggung_jawab = $this->input-
>post("pekerjaan_penanggung_jawab");
102.             $hubungan_penanggung_jawab = $this->input-
>post("hubungan_penanggung_jawab");
103.             $id_dokter = $this->input->post("id_dokter");
104.             $jenis_pasien = $this->input->post('jenis_pasien');
105.
106.             27 $data = array(
107.                 "nama"=>$nama,
108.                 "telpon"=>$telpon,
109.                 "umur"=>$umur,
110.                 "pekerjaan"=>$pekerjaan,
111.                 "alamat"=>$alamat,
112.                 "jenis_kelamin"=>$jenis_kelamin,
113.                 "golongan_darah"=>$golongan_darah,
114.                 "tanggal_daftar"=>date("Y-m-d H:i:s"),
115.
116.                 "nama_penanggung_jawab"=>$nama_penanggung_jawab,

```

```

117.         "alamat_penanggung_jawab"=>$alamat_penanggung_jawab,
118.         "telpon_penanggung_jawab"=>$telpon_penanggung_jawab,
119.         "pekerjaan_penanggung_jawab"=>$pekerjaan_penanggung_j
120.         awab,
121.         "hubungan_penanggung_jawab"=>$hubungan_penanggung_jaw
122.         ab,
123.         "id_dokter"=>$id_dokter,
124.         "jenis_pasien"=>$jenis_pasien,
125.     );
126.     28 if($this->pasien->update($id_pasien,$data)){
127.         $this->session-
128.         >set_flashdata('message', 'Successfully update data. '); 29
129.         redirect("pasien");
130.     }else{
131.         30 $this->session-
132.         >set_flashdata('message', 'Failed update new data. ');
133.         redirect("pasien");
134.     }
135. } 31
136. } 32
137. public function delete($id){ 33
138.     $this->pasien->delete($id);
139.     $this->session-
140.     >set_flashdata('message', 'Successfully delete data. ');
141.     redirect("pasien");
142. } 34
143. } 35
144. } 36
    
```

2. Grafik Alir

Grafik Alir fitur Manajemen Data Pasien berupa notasi-notasi yang menunjukkan alur kontrol. Fitur Manajemen Data Pasien terbagi menjadi beberapa fungsi, yaitu *construct*, *index*, *add*, *edit*, dan *delete*.

2.1. Grafik Alir Fitur Manajemen Data Pasien Fungsi *Construct*

Grafik alir fitur Manajemen Data Pasien Fungsi *Construct* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.12.



Gambar C. 12 Grafik Alir Manajemen Data Pasien Fungsi *Construct*
 Sumber: (Hasil analisis)

2.2. Grafik Alir Fitur Manajemen Data Pasien Fungsi *Index*

Grafik alir fitur Manajemen Data Pasien Fungsi *Index* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.13.



Gambar C. 13 Grafik Alir Manajemen Data Pasien Fungsi *Index*
 Sumber: (Hasil analisis)

2.3. Grafik Alir Fitur Manajemen Data Pasien Fungsi *Add*

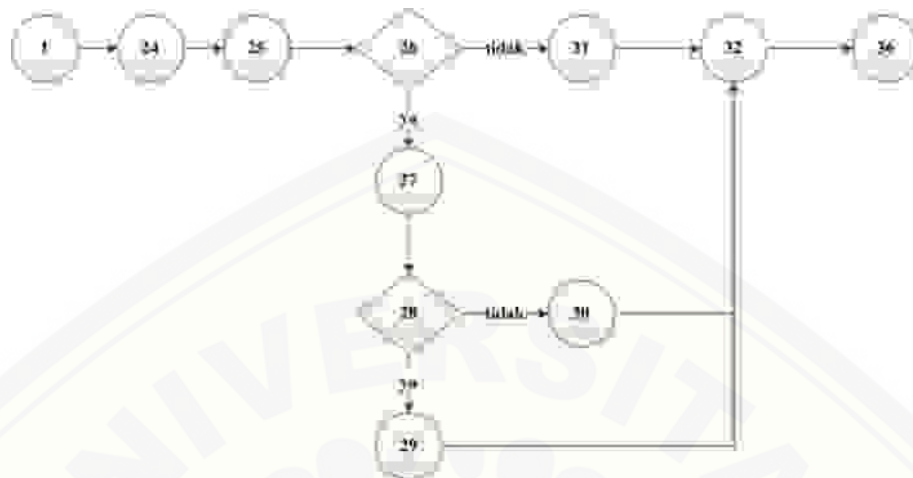
Grafik alir fitur Manajemen Data Pasien Fungsi *Add* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.14.



Gambar C. 14 Grafik Alir Manajemen Data Pasien Fungsi *Add*
 Sumber: (Hasil analisis)

2.4. Grafik Alir Fitur Manajemen Data Pasien Fungsi *Edit*

Grafik alir fitur Manajemen Data Pasien Fungsi *Edit* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.15.



Gambar C. 15 Diagram Alir Manajemen Data Pasien Fungsi *Edit*
 Sumber: (Hasil analisis)

2.5. Grafik Alir Fitur Manajemen Data Pasien Fungsi *Delete*

Grafik alir fitur Manajemen Data Pasien Fungsi *Delete* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.16.



Gambar C. 16 Grafik Alir Manajemen Data Pasien Fungsi *Delete*
 Sumber: (Hasil analisis)

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur Manajemen Data Pasien dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2$$

Dengan $V(G)$ adalah Kompleksitas siklomatis, E adalah *Edge* atau tepi atau garis dari sebuah grafik, dan N adalah *Node* atau titik dari sebuah grafik. Maka perhitungan dari setiap grafik dari Manajemen Data Pasien adalah sebagai berikut.

3.1. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Pasien Fungsi *construct*

$$V(G) = E - N + 2$$

$$= 8 - 8 + 2$$

$$= 2$$

- 3.2. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Pasien Fungsi *Index*

$$V(G) = E - N + 2$$

$$= 4 - 5 + 2$$

$$= 1$$

- 3.3. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Pasien Fungsi *Add*

$$V(G) = E - N + 2$$

$$= 17 - 15 + 2$$

$$= 4$$

- 3.4. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Pasien Fungsi *Edit*

$$V(G) = E - N + 2$$

$$= 12 - 11 + 2$$

$$= 3$$

- 3.5. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Pasien Fungsi *Delete*

$$V(G) = E - N + 2$$

$$= 4 - 5 + 2$$

$$= 1$$

4. *Basis Set*

- 4.1. *Basis set* Fitur Manajemen Data Pasien Fungsi *Construct*

Basis set Manajemen Data Pasien fungsi *construct* menghasilkan dua (2) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 - 2 - 3 - 4 - 6 - 7 - 36

Jalur 2 : 1 - 2 - 3 - 4 - 5 - 7 - 36

- 4.2. *Basis set* Fitur Manajemen Data Pasien Fungsi *Index*

Basis set Manajemen Data Pasien fungsi *index* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 - 8 - 9 - 10 - 36

- 4.3. *Basis set* Fitur Manajemen Data Pasien Fungsi *Add*

Basis set Manajemen Data Pasien fungsi *Add* menghasilkan empat (4) jalur

independent dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 11 – 18 – 19 – 20 – 22 – 12 – 13 – 14 – 15 – 23 – 36

Jalur 2 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 13 – 14 – 15 – 23 – 36

Jalur 3 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 13 – 14 – 16 – 23 – 36

Jalur 4 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 17 – 23 – 36

4.4. *Basis set* Fitur Manajemen Data Pasien Fungsi *Edit*

Basis set Manajemen Data Pasien fungsi *Edit* menghasilkan tiga (3) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 24 – 25 – 26 – 31 – 32 – 36

Jalur 2 : 1 – 24 – 25 – 26 – 27 – 28 – 30 – 32 – 36

Jalur 3 : 1 – 24 – 25 – 26 – 27 – 28 – 29 – 32 – 36

4.5. *Basis set* Fitur Manajemen Data Pasien Fungsi *Delete*

Basis set Manajemen Data Pasien fungsi *Delete* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 33 – 34 – 35 – 36

5. *Test Case*

Tabel C. 24 *Test Case* Fitur Manajemen Data Pasien fungsi *construct* jalur 1
Sumber: (Hasil analisis)

Test case	User berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif kemudian klik menu Pasien
Target yang diharapkan	Sistem menampilkan halaman Pasien
Hasil pengujian	Benar
Jalur	Jalur 1
File	pasien.php

Tabel C. 25 *Test Case* Fitur Manajemen Data Pasien fungsi *construct* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Pasien, <i>User</i> melakukan <i>Logout</i> dan menuju halaman <i>Login</i> dan ketika di <i>back</i> tetap menampilkan halaman <i>Login</i>
-----------	---

Target yang diharapkan	Sistem menampilkan halaman <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	pasien.php

Tabel C. 26 *Test Case* Fitur Manajemen Data Pasien fungsi *Index* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif lalu klik menu Pasien
Target yang diharapkan	Sistem menampilkan halaman Pasien
Hasil pengujian	Benar
Jalur	Jalur 1
File	pasien.php

Tabel C. 27 *Test Case* Fitur Manajemen Data Pasien fungsi *Add* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Pasien, <i>User</i> mengklik Tombol Tambah Pasien kemudian mengisi <i>form</i> Tambah Pasien setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Pasien serta menampilkan <i>alert</i> ,” <i>Successfully add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 1
File	pasien.php

Tabel C. 28 *Test Case* Fitur Manajemen Data Pasien fungsi *Add* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Pasien, <i>User</i> mengklik Tombol Tambah Pasien kemudian mengisi <i>form</i> Tambah Pasien setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Pasien serta menampilkan <i>alert</i> ,” <i>Successfully add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 2
File	pasien.php

Tabel C. 29 *Test Case* Fitur Manajemen Data Pasien fungsi *Add* jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Pasien, <i>User</i> mengklik Tombol Tambah Pasien kemudian mengisi <i>form</i> Tambah Pasien setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Pasien dan menampilkan <i>alert, "Failed add new data."</i>
Hasil pengujian	Benar
Jalur	Jalur 3
File	pasien.php

Tabel C. 30 *Test Case* Fitur Manajemen Data Pasien fungsi *Add* jalur 4
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Pasien, <i>User</i> mengklik Tombol Tambah Pasien
Target yang diharapkan	Sistem menampilkan halaman Tambah Pasien
Hasil pengujian	Benar
Jalur	Jalur 4
File	pasien.php

Tabel C. 31 *Test Case* Fitur Manajemen Data Pasien fungsi *Edit* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Pasien, <i>User</i> mengklik Tombol <i>Edit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Edit</i> Pasien
Hasil pengujian	Benar
Jalur	Jalur 1
File	pasien.php

Tabel C. 32 *Test Case* Fitur Manajemen Data Pasien fungsi *Edit* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Pasien, <i>User</i> mengisi <i>form edit</i> Pasien kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Pasien dan menampilkan <i>alert, "Failed update data."</i>

Hasil pengujian	Benar
Jalur	Jalur
File	pasien.php

Tabel C. 33 *Test Case* Fitur Manajemen Data Pasien fungsi *Edit* jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Pasien, <i>User</i> mengisi <i>form edit</i> Pasien kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Pasien dan menampilkan <i>alert, "Successfully update data.."</i>
Hasil pengujian	Benar
Jalur	Jalur 3
File	pasien.php

Tabel C. 34 *Test Case* Fitur Manajemen Data Pasien fungsi *Delete* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Pasien, <i>User</i> mengklik tombol <i>Remove</i>
Target yang diharapkan	Sistem menampilkan halaman Pasien dan menampilkan <i>alert, "Successfully delete data.."</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	pasien.php

C. 4 Manajemen Data Kamar

Pengujian kode-kode program pada fitur Manajemen Data Kamar adalah sebagai berikut:

1. Listing Program

```

1. 1 class Kamar extends CI_Controller {
2.
3.     2public function __construct() {
4.         parent::__construct();
5.         $this->load->model("kamar_model", "kamar"); 3
6.         // $this->authorization->isAdminLogin();
7.         4 if (!$this->session->userdata("rkmds-logged-in")) {
8.             redirect('/', 5);
9.         }
7     } 6

```

```

10.     }
11.
12.     public function index() { 8
13.         $data = array();
14.         $data['kamars'] = $this->kamar->getDataKamar();
15.         $data['count_regular'] = $this->db->where('kelas','regular')->
>get('tb_kamar')->num_rows();
16.         $data['count_vip'] = $this->db->where('kelas','vip')->
>get('tb_kamar')->num_rows();
17.         9 $data['count_kosong'] = $this->db->where('status','kosong')->
>get('tb_kamar')->num_rows();
18.         $data['count_terisi'] = $this->db->where('status','terisi')->
>get('tb_kamar')->num_rows();
19.         $data['count_jumlah'] = $this->db->get('tb_kamar')->
>num_rows();
20.         $this->load->view('kamar-data',$data);
21.     } 10
22.
23.     11 public function add() {
24.         if($this->input->post("submit")) { 12
25.             $nama = $this->input->post("nama");
26.             $kelas = $this->input->post("kelas");
27.             $biaya = $this->input->post("biaya");
28.             $id_fasilitas = $this->input->post("id_fasilitas");
29.             $kode_kamar = $this->input->post("kode_kamar");
30.             13 $data = array(
31.                 "nama"=>$nama,
32.                 "kelas"=>$kelas,
33.                 "biaya"=>$biaya,
34.                 "kode_kamar"=>$kode_kamar,
35.             );
36.             14 if($this->kamar->insert($data,$id_fasilitas){
37.                 $this->session->
>set_flashdata('message', 'Successfully add new data. '); 15
38.                 redirect("kamar");
39.             } else {
40.                 16 $this->session->
>set_flashdata('message', 'Failed add new data. ');
41.                 redirect("kamar");
42.             }
43.         } 17
44.         $id = "";
45.         18 $sql = 'SELECT COUNT(*) AS total FROM tb_kamar ';
46.         $total = $this->db->query($sql)->row('total');
47.         19 if($total==0) { //never made
48.             $autonum = "0001"; 20
49.         }
50.         else {
51.             21 $sql = 'SELECT * FROM tb_kamar ORDER BY id_kamar DESC LIMIT
1';
52.             $last = $this->db->query($sql)->row('kode_kamar');
53.             $lastnum = substr($last,4,4);
54.             $autonum = (int) $lastnum + 1;
55.             $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
56.
57.

```

```

58.     }
59.
60.     $id = "KMR-".$autonum;
61.     $data['id'] = $id;
62.
63.     $this->load->view('add-kamar',$data);
64. }
65.
66. public function edit($id){
67.     $data = array();
68.     $kamar = $this->kamar->getkamar($id);
69.     $data['kamar'] = $kamar;
70.     $this->load->view('edit-kamar',$data);
71.     if($this->input->post("submit")){
72.         $nama = $this->input->post("nama");
73.         $kelas = $this->input->post("kelas");
74.         $biaya = $this->input->post("biaya");
75.         $status = $this->input->post("status");
76.         $id_fasilitas = $this->input->post("id_fasilitas");
77.         $id_kamar = $this->input->post("id_kamar");
78.         $data = array(
79.             "nama"=>$nama,
80.             "kelas"=>$kelas,
81.             "biaya"=>$biaya,
82.             "status"=>$status,
83.         );
84.         if($this->kamar->update($id_kamar,$data,$id_fasilitas)){
85.             $this->session-
86.             >set_flashdata('message', 'Successfully update data. ');
87.             redirect("kamar");
88.         }else{
89.             $this->session-
90.             >set_flashdata('message', 'Failed update new data. ');
91.             redirect("kamar");
92.         }
93.     }
94.     public function delete($id){
95.         $this->kamar->delete($id);
96.         $this->session-
97.         >set_flashdata('message', 'Successfully delete data. ');
98.         redirect("kamar");
99.     }

```

2. Grafik Alir

Grafik Alir fitur Manajemen Data Kamar berupa notasi-notasi yang menunjukkan alur kontrol. Fitur Manajemen Data Kamar terbagi menjadi beberapa fungsi, yaitu *construct*, *index*, *add*, *edit*, dan *delete*.

2.5. Grafik Alir Fitur Manajemen Data Kamar Fungsi *Delete*

Grafik alir fitur Manajemen Data Kamar Fungsi *Delete* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.21.



Gambar C. 21 Grafik Alir Manajemen Data Kamar Fungsi *Delete*

Sumber: (Hasil analisis)

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur Manajemen Data Kamar dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2$$

Dengan $V(G)$ adalah Kompleksitas siklomatis, E adalah *Edge* atau tepi atau garis dari sebuah grafik, dan N adalah *Node* atau titik dari sebuah grafik. Maka perhitungan dari setiap grafik dari Manajemen Data Kamar adalah sebagai berikut.

3.1. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Kamar Fungsi *construct*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 8 - 8 + 2 \\ &= 2 \end{aligned}$$

3.2. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Kamar Fungsi *Index*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

3.3. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Kamar Fungsi *Add*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 17 - 15 + 2 \\ &= 4 \end{aligned}$$

3.4. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Kamar Fungsi *Edit*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 12 - 11 + 2 \\ &= 3 \end{aligned}$$

3.5. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Kamar Fungsi

Delete

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

4. *Basis Set*

4.1. *Basis set* Fitur Manajemen Data Kamar Fungsi *Construct*

Basis set Manajemen Data Kamar fungsi *construct* menghasilkan dua (2) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 6 – 7 – 36

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 7 – 36

4.2. *Basis set* Fitur Manajemen Data Kamar Fungsi *Index*

Basis set Manajemen Data Kamar fungsi *index* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 8 – 9 – 10 – 36

4.3. *Basis set* Fitur Manajemen Data Kamar Fungsi *Add*

Basis set Manajemen Data Kamar fungsi *Add* menghasilkan empat (4) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 11 – 18 – 19 – 20 – 22 – 12 – 13 – 14 – 15 – 23 – 36

Jalur 2 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 13 – 14 – 15 – 23 – 36

Jalur 3 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 13 – 14 – 16 – 23 – 36

Jalur 4 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 17 – 23 – 36

4.4. *Basis set* Fitur Manajemen Data Kamar Fungsi *Edit*

Basis set Manajemen Data Kamar fungsi *Edit* menghasilkan tiga (3) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 24 – 25 – 26 – 31 – 32 – 36

Jalur 2 : 1 – 24 – 25 – 26 – 27 – 28 – 30 – 32 – 36

Jalur 3 : 1 – 24 – 25 – 26 – 27 – 28 – 29 – 32 – 36

4.5. *Basis set* Fitur Manajemen Data Kamar Fungsi *Delete*

Basis set Manajemen Data Kamar fungsi *Delete* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 33 – 34 – 35 – 36

5. *Test Case*

Tabel C. 35 *Test Case* Fitur Manajemen Data Kamar fungsi *construct* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif kemudian klik menu Kamar
Target yang diharapkan	Sistem menampilkan halaman Kamar
Hasil pengujian	Benar
Jalur	Jalur 1
File	kamar.php

Tabel C. 36 *Test Case* Fitur Manajemen Data Kamar fungsi *construct* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Kamar, <i>User</i> melakukan <i>Logout</i> dan menuju halaman <i>Login</i> dan ketika di <i>back</i> tetap menampilkan halaman <i>Login</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	kamar.php

Tabel C. 37 *Test Case* Fitur Manajemen Data Kamar fungsi *Index* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif lalu klik menu Kamar
Target yang diharapkan	Sistem menampilkan halaman Kamar
Hasil pengujian	Benar
Jalur	Jalur 1
File	kamar.php

Tabel C. 38 *Test Case* Fitur Manajemen Data Kamar fungsi *Add* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Kamar, <i>User</i> mengklik Tombol Tambah Kamar kemudian mengisi <i>form</i> Tambah Kamar setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Kamar serta menampilkan <i>alert</i> ,” <i>Successfully add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 1
File	kamar.php

Tabel C. 39 *Test Case* Fitur Manajemen Data Kamar fungsi *Add* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Kamar, <i>User</i> mengklik Tombol Tambah Kamar kemudian mengisi <i>form</i> Tambah Kamar setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Kamar serta menampilkan <i>alert</i> ,” <i>Successfully add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 2
File	kamar.php

Tabel C. 40 *Test Case* Fitur Manajemen Data Kamar fungsi *Add* jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Kamar, <i>User</i> mengklik Tombol Tambah Kamar kemudian mengisi <i>form</i> Tambah Kamar setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Kamar dan menampilkan <i>alert</i> ,” <i>Failed add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 3
File	kamar.php

Tabel C. 41 *Test Case* Fitur Manajemen Data Kamar fungsi *Add* jalur 4
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Kamar, <i>User</i> mengklik Tombol Tambah Kamar
Target yang diharapkan	Sistem menampilkan halaman Tambah Kamar
Hasil pengujian	Benar
Jalur	Jalur 4
File	kamar.php

Tabel C. 42 *Test Case* Fitur Manajemen Data Kamar fungsi *Edit* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Kamar, <i>User</i> mengklik Tombol <i>Edit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Edit</i> Kamar
Hasil pengujian	Benar
Jalur	Jalur 1
File	kamar.php

Tabel C. 43 *Test Case* Fitur Manajemen Data Kamar fungsi *Edit* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Kamar, <i>User</i> mengisi <i>form edit</i> Kamar kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Kamar dan menampilkan <i>alert," Failed update data.."</i>
Hasil pengujian	Benar
Jalur	Jalur
File	kamar.php

Tabel C. 44 *Test Case* Fitur Manajemen Data Kamar fungsi *Edit* jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Kamar, <i>User</i> mengisi <i>form edit</i> Kamar kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Kamar dan menampilkan <i>alert," Successfully update data.."</i>

Hasil pengujian	Benar
Jalur	Jalur 3
File	kamar.php

Tabel C. 45 Test Case Fitur Manajemen Data Kamar fungsi *Delete* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Kamar, <i>User</i> mengklik tombol <i>Remove</i>
Target yang diharapkan	Sistem menampilkan halaman Kamar dan menampilkan <i>alert, "Successfully delete data.."</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	kamar.php

C. 5 Manajemen Data Fasilitas

Pengujian kode – kode program pada fitur Manajemen Data Fasilitas adalah sebagai berikut:

1. Listing Program

```

1. 1 class Fasilitas extends CI_Controller {
2.
3.     2 public function __construct() {
4.         parent::__construct();
5.         $this->load->model("fasilitas_model", "fasilitas" ; 3
6.         // $this->authorization->isAdminLogin();
7.         4 if (!$this->session->userdata("rkmds-logged-in")) {
8.             redirect('/'); 5
9.         } 6
10.    } 7
11.
12.    8 public function index() {
13.        $data = array();
14.        $data['fasilitass'] = $this->fasilitas->getDatafasilitas( ; 9
15.        $this->load->view('fasilitas-data', $data);
16.    } 10
17.
18.    public function add() { 11
19.        if ($this->input->post("submit") ) { 12
20.            $nama = $this->input->post("nama");
21.            $jumlah = $this->input->post("jumlah");
22.            $biaya = $this->input->post("biaya");
23.            $kode_fasilitas = $this->input->post("kode_fasilitas");
24.            $data = array(
25.                "nama"=>$nama,
26.                "jumlah"=>$jumlah,

```

```

27.         "biaya"=>$biaya,
28.         "kode_fasilitas"=>$kode_fasilitas,
29.     );
30.     14 if($this->fasilitas->insert($data)){
31.         $this->session-
>set_flashdata('message', 'Successfully add new data. '); 15
32.         redirect("fasilitas");
33.     }else{
34.         16 $this->session-
>set_flashdata('message', 'Failed add new data. ');
35.         redirect("fasilitas");
36.     }
37. }
38.
39.     $id = "";
40.     18 $sql = 'SELECT COUNT(*) AS total FROM tb_fasilitas ';
41.     $total = $this->db->query($sql)->row('total');
42.
43.     19 if($total==0){ //never made
44.         $autonum = "0001"; 20
45.     }
46.     else{
47.         21 $sql = 'SELECT * FROM tb_fasilitas ORDER BY id_fasilitas DES
C LIMIT 1';
48.         $last = $this->db->query($sql)->row('kode_fasilitas');
49.         $lastnum = substr($last,4,4);
50.         $autonum = (int) $lastnum + 1;
51.         $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
52.     }
53.
54.     $id = "FAS-".$autonum;
55.     22 $data['id'] = $id;
56.     $this->load->view('add-fasilitas', $data);
57. } 23
58.
59.     public function edit($id){ 24
60.         $data = array();
61.         $fasilitas = $this->fasilitas->getfasilitas($id);
62.         25 $data['fasilitas'] = $fasilitas;
63.         $this->load->view('edit-fasilitas', $data);
64.         if($this->input->post("submit")){ 26
65.             $nama = $this->input->post("nama");
66.             $jumlah = $this->input->post("jumlah");
67.             $biaya = $this->input->post("biaya");
68.             $id_fasilitas = $this->input->post("id_fasilitas");
69.             27 $data = array(
70.                 "nama"=>$nama,
71.                 "jumlah"=>$jumlah,
72.                 "biaya"=>$biaya,
73.             );
74.
75.             28 if($this->fasilitas->update($id_fasilitas, $data){
76.                 $this->session-
>set_flashdata('message', 'Successfully update data. '); 29
77.                 redirect("fasilitas");
78.             }else{
30

```

```

79.         $this->session-
>set_flashdata('message', 'Failed update new data. ');
80.         redirect("fasilitas");
81.     }
82. } 31
83. } 32
84.
85.     public function delete($id){ 33
86.         $this->fasilitas->delete($id);
87.         $this->session-
>set_flashdata('message', 'Successfully delete data. '); 34
88.         redirect("fasilitas");
89.     } 35
90. } 36
    
```

2. Grafik Alir

Grafik Alir fitur Manajemen Data Fasilitas berupa notasi-notasi yang menunjukkan alur kontrol. Fitur Manajemen Data Fasilitas terbagi menjadi beberapa fungsi, yaitu *construct*, *index*, *add*, *edit*, dan *delete*.

2.1. Grafik Alir Fitur Manajemen Data Fasilitas Fungsi *Construct*

Grafik alir fitur Manajemen Data Fasilitas Fungsi *Construct* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.22.



Gambar C. 22 Grafik Alir Manajemen Data Fasilitas Fungsi *Construct*
 Sumber: (Hasil analisis)

2.2. Grafik Alir Fitur Manajemen Data Fasilitas Fungsi *Index*

Grafik alir fitur Manajemen Data Fasilitas Fungsi *Index* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.23.



Gambar C. 23 Grafik Alir Manajemen Data Fasilitas Fungsi *Index*
 Sumber: (Hasil analisis)

2.3. Grafik Alir Fitur Manajemen Data Fasilitas Fungsi *Add*

Grafik alir fitur Manajemen Data Fasilitas Fungsi *Add* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.24.

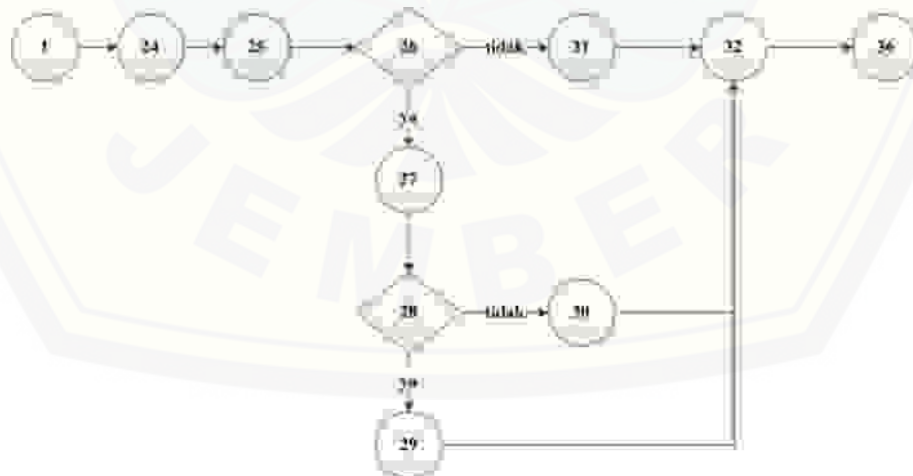


Gambar C. 24 Grafik Alir Manajemen Data Fasilitas Fungsi *Add*

Sumber: (Hasil analisis)

2.4. Grafik Alir Fitur Manajemen Data Fasilitas Fungsi *Edit*

Grafik alir fitur Manajemen Data Fasilitas Fungsi *Edit* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.25.



Gambar C. 25 Diagram Alir Manajemen Data Fasilitas Fungsi *Edit*

Sumber: (Hasil analisis)

2.5. Grafik Alir Fitur Manajemen Data Fasilitas Fungsi *Delete*

Grafik alir fitur Manajemen Data Fasilitas Fungsi *Delete* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.26.



Gambar C. 26 Grafik Alir Manajemen Data Fasilitas Fungsi *Delete*
Sumber: (Hasil analisis)

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur Manajemen Data Fasilitas dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2$$

Dengan $V(G)$ adalah Kompleksitas siklomatis, E adalah *Edge* atau tepi atau garis dari sebuah grafik, dan N adalah *Node* atau titik dari sebuah grafik. Maka perhitungan dari setiap grafik dari Manajemen Data Fasilitas adalah sebagai berikut.

3.1. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Fasilitas Fungsi *construct*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 8 - 8 + 2 \\ &= 2 \end{aligned}$$

3.2. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Fasilitas Fungsi *Index*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

3.3. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Fasilitas Fungsi *Add*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 17 - 15 + 2 \\ &= 4 \end{aligned}$$

3.4. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Fasilitas Fungsi *Edit*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 12 - 11 + 2 \\ &= 3 \end{aligned}$$

3.5. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Fasilitas Fungsi

Delete

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

4. Basis Set

4.1. Basis set Fitur Manajemen Data Fasilitas Fungsi *Construct*

Basis set Manajemen Data Fasilitas fungsi *construct* menghasilkan dua (2) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 6 – 7 – 36

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 7 – 36

4.2. Basis set Fitur Manajemen Data Fasilitas Fungsi *Index*

Basis set Manajemen Data Fasilitas fungsi *index* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 8 – 9 – 10 – 36

4.3. Basis set Fitur Manajemen Data Fasilitas Fungsi *Add*

Basis set Manajemen Data Fasilitas fungsi *Add* menghasilkan empat (4) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 11 – 18 – 19 – 20 – 22 – 12 – 13 – 14 – 15 – 23 – 36

Jalur 2 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 13 – 14 – 15 – 23 – 36

Jalur 3 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 13 – 14 – 16 – 23 – 36

Jalur 4 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 17 – 23 – 36

4.4. Basis set Fitur Manajemen Data Fasilitas Fungsi *Edit*

Basis set Manajemen Data Fasilitas fungsi *Edit* menghasilkan tiga (3) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 24 – 25 – 26 – 31 – 32 – 36

Jalur 2 : 1 – 24 – 25 – 26 – 27 – 28 – 30 – 32 – 36

Jalur 3 : 1 – 24 – 25 – 26 – 27 – 28 – 29 – 32 – 36

4.5. *Basis set* Fitur Manajemen Data Fasilitas Fungsi *Delete*

Basis set Manajemen Data Fasilitas fungsi *Delete* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 33 – 34 – 35 – 36

5. *Test Case*

Tabel C. 46 *Test Case* Fitur Manajemen Data Fasilitas fungsi *construct* jalur 1

Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif kemudian klik menu Fasilitas
Target yang diharapkan	Sistem menampilkan halaman Fasilitas
Hasil pengujian	Benar
Jalur	Jalur 1
File	fasilitas.php

Tabel C. 47 *Test Case* Fitur Manajemen Data Fasilitas fungsi *construct* jalur 2

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Fasilitas, <i>User</i> melakukan <i>Logout</i> dan menuju halaman <i>Login</i> dan ketika di <i>back</i> tetap menampilkan halaman <i>Login</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	fasilitas.php

Tabel C. 48 *Test Case* Fitur Manajemen Data Fasilitas fungsi *Index* jalur 1

Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif lalu klik menu Fasilitas
Target yang diharapkan	Sistem menampilkan halaman Fasilitas
Hasil pengujian	Benar

Jalur	Jalur 1
File	fasilitas.php

Tabel C. 49 *Test Case* Fitur Manajemen Data Fasilitas fungsi *Add* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Fasilitas, <i>User</i> mengklik Tombol Tambah Fasilitas kemudian mengisi <i>form</i> Tambah Fasilitas setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Fasilitas serta menampilkan <i>alert, "Successfully add new data."</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	fasilitas.php

Tabel C. 50 *Test Case* Fitur Manajemen Data Fasilitas fungsi *Add* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Fasilitas, <i>User</i> mengklik Tombol Tambah Fasilitas kemudian mengisi <i>form</i> Tambah Fasilitas setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Fasilitas serta menampilkan <i>alert, "Successfully add new data."</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	fasilitas.php

Tabel C. 51 *Test Case* Fitur Manajemen Data Fasilitas fungsi *Add* jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Fasilitas, <i>User</i> mengklik Tombol Tambah Fasilitas kemudian mengisi <i>form</i> Tambah Fasilitas setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Fasilitas dan menampilkan <i>alert, "Failed add new data."</i>

Hasil pengujian	Benar
Jalur	Jalur 3
File	fasilitas.php

Tabel C. 52 *Test Case* Fitur Manajemen Data Fasilitas fungsi *Add* jalur 4
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Fasilitas, <i>User</i> mengklik Tombol Tambah Fasilitas
Target yang diharapkan	Sistem menampilkan halaman Tambah Fasilitas
Hasil pengujian	Benar
Jalur	Jalur 4
File	fasilitas.php

Tabel C. 53 *Test Case* Fitur Manajemen Data Fasilitas fungsi *Edit* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Fasilitas, <i>User</i> mengklik Tombol <i>Edit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Edit</i> Fasilitas
Hasil pengujian	Benar
Jalur	Jalur 1
File	fasilitas.php

Tabel C. 54 *Test Case* Fitur Manajemen Data Fasilitas fungsi *Edit* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Fasilitas, <i>User</i> mengisi <i>form edit</i> Fasilitas kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Fasilitas dan menampilkan <i>alert, "Failed update data.."</i>
Hasil pengujian	Benar
Jalur	Jalur
File	fasilitas.php

Tabel C. 55 *Test Case* Fitur Manajemen Data Fasilitas fungsi *Edit* jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Fasilitas,
-----------	---

	<i>User</i> mengisi <i>form edit</i> Fasilitas kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Fasilitas dan menampilkan <i>alert, "Successfully update data.."</i>
Hasil pengujian	Benar
Jalur	Jalur 3
File	fasilitas.php

Tabel C. 56 Test Case Fitur Manajemen Data Fasilitas fungsi *Delete* jalur 1
 Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Fasilitas, <i>User</i> mengklik tombol <i>Remove</i>
Target yang diharapkan	Sistem menampilkan halaman Fasilitas dan menampilkan <i>alert, "Successfully delete data.."</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	fasilitas.php

C. 6 Manajemen Data Dokter

Pengujian kode – kode program pada fitur Manajemen Data Dokter adalah sebagai berikut:

1. Listing Program

```

1. 1 class Dokter extends CI_Controller {
2.
3.     public function __construct() { 2
4.         parent::__construct();
5.         $this->load->model("dokter_model", "dokter");
6.         // $this->authorization->isAdminLogin();
7.         4 if (!$this->session->userdata("rkmds-logged-in")) {
8.             redirect('/'); 5
9.         } 6
10.    } 7
11.
12.    public function index() { 8
13.        $data = array();
14.        9 $data['dokters'] = $this->dokter->getDataDokter();
15.        $this->load->view('dokter-data', $data);
16.    } 10
17.
18.    public function add() { 11
19.        if ($this->input->post("submit")) { 12
    
```

```

20.     $nama = $this->input->post("nama");
21.     $tempat_lahir = $this->input->post("tempat_lahir");
22.     $tanggal_lahir = $this->input->post("tanggal_lahir");
23.     $usia = $this->input->post("usia");
24.     $jenis_kelamin = $this->input->post("jenis_kelamin");
25.     $telpon = $this->input->post("telpon");
26.     $spesialis = $this->input->post("spesialis");
27.     $alamat = $this->input->post("alamat");
28.     $arrtgl = explode("/", $tanggal_lahir);
29.     $kode_dokter = $this->input->post('kode_dokter');
30.     $data = array(
31.         13         "nama"=>$nama,
32.         "tempat_lahir"=>$tempat_lahir,
33.         "tanggal_lahir"=>date("Y-m-d",strtotime($arrtgl[2]."-
". $arrtgl[1]."-". $arrtgl[0])),
34.         "usia"=>$usia,
35.         "jenis_kelamin"=>$jenis_kelamin,
36.         "telpon"=>$telpon,
37.         "spesialis"=>$spesialis,
38.         "alamat"=>$alamat,
39.         "kode_dokter"=>$kode_dokter,
40.     );
41.
42.     14 if($this->dokter->insert($data)){
43.         $this->session-
44.         >set_flashdata('message', 'Successfully add new data. 15
         redirect("dokter");
45.     }else{
46.         16 $this->session-
47.         >set_flashdata('message', 'Failed add new data.
         redirect("dokter");
48.     }
49.     17}
50.     $id = "";
51.     $sql = 'SELECT COUNT(*) AS total FROM tb_dokter';
52.     $total = $this->db->query($sql)->row('total'); 18
53.
54.     19 if($total==0){ //never made
55.         $autonum = "000"; 20
56.     }
57.     else{
58.         21 $sql = 'SELECT * FROM tb_dokter ORDER BY id_dokter DESC LIM
         T 1';
59.         $last = $this->db->query($sql)->row('kode_dokter');
60.         $lastnum = substr($last,4,4);
61.         $autonum = (int) $lastnum + 1;
62.         $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
63.     }
64.     $id = "DOK-".$autonum;
65.     22 $data['id'] = $id;
66.
67.     $this->load->view('add-dokter',$data);
68. }
69. 23
70. public function edit($id){ 24
71.     $data = array();

```



```

72.     $dokter = $this->dokter->getDokter($id);
73.     $data['dokter'] = $dokter;
74.     $this->load->view('edit-dokter', $data);
75.     26 if($this->input->post("submit")){
76.         $nama = $this->input->post("nama");
77.         $tempat_lahir = $this->input->post("tempat_lahir");
78.         $tanggal_lahir = $this->input->post("tanggal_lahir");
79.         $usia = $this->input->post("usia");
80.         $jenis_kelamin = $this->input->post("jenis_kelamin");
81.         $telpon = $this->input->post("telpon");
82.         $spesialis = $this->input->post("spesialis");
83.         $alamat = $this->input->post("alamat");
84.         $arrtgl = explode("/", $tanggal_lahir);
85.         $id_dokter = $this->input->post("id_dokter");
86.         27 $data = array(
87.             "nama"=>$nama,
88.             "tempat_lahir"=>$tempat_lahir,
89.             "tanggal_lahir"=>date("Y-m-d", strtotime($arrtgl[2]."-
90.             ".$arrtgl[1]."-".$arrtgl[0])),
91.             "usia"=>$usia,
92.             "jenis_kelamin"=>$jenis_kelamin,
93.             "telpon"=>$telpon,
94.             "spesialis"=>$spesialis,
95.             "alamat"=>$alamat,
96.         );
97.         28 if($this->dokter->update($id_dokter, $data)){
98.             $this->session-
99.             >set_flashdata('message', 'Successfully update data. '); 29
100.             redirect("dokter");
101.         }else{
102.             30 $this->session-
103.             >set_flashdata('message', 'Failed update new data. ');
104.             redirect("dokter");
105.         }
106.     }
107.     32 } 31
108.     public function delete($id){ 33
109.         $this->dokter->delete($id);
110.         34 $this->session-
111.         >set_flashdata('message', 'Successfully delete data. ');
112.         redirect("dokter");
113.     }
114.     35 }
115. } 36 }

```

2. Grafik Alir

Grafik Alir fitur Manajemen Data Dokter berupa notasi-notasi yang menunjukkan alur kontrol. Fitur Manajemen Data Dokter terbagi menjadi beberapa fungsi, yaitu *construct*, *index*, *add*, *edit*, dan *delete*.

2.1. Grafik Alir Fitur Manajemen Data Dokter Fungsi *Construct*

Grafik alir fitur Manajemen Data Dokter Fungsi *Construct* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.27.



Gambar C. 27 Grafik Alir Manajemen Data Dokter Fungsi *Construct*
 Sumber: (Hasil analisis)

2.2. Grafik Alir Fitur Manajemen Data Dokter Fungsi *Index*

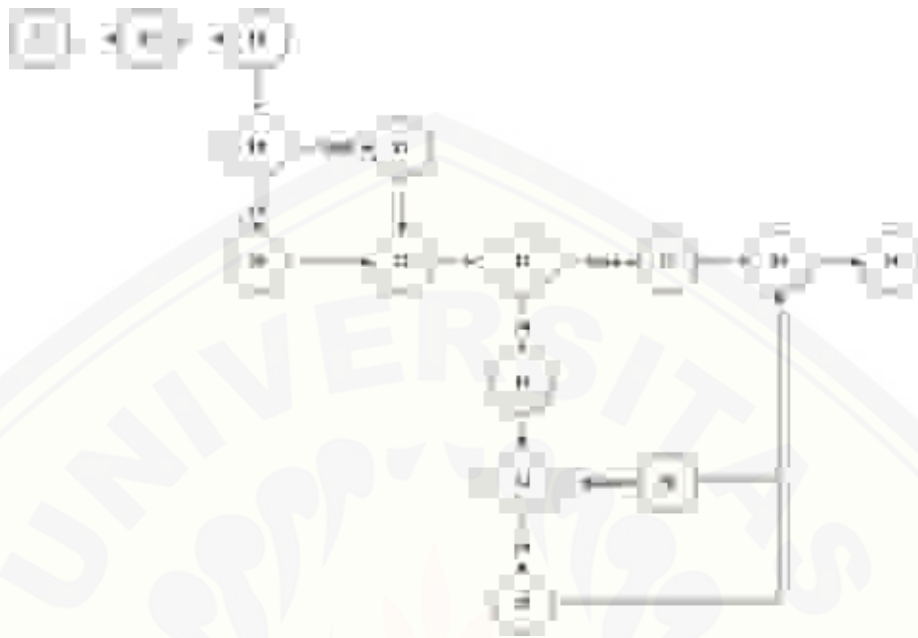
Grafik alir fitur Manajemen Data Dokter Fungsi *Index* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.28.



Gambar C. 28 Grafik Alir Manajemen Data Dokter Fungsi *Index*
 Sumber: (Hasil analisis)

2.3. Grafik Alir Fitur Manajemen Data Dokter Fungsi *Add*

Grafik alir fitur Manajemen Data Dokter Fungsi *Add* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.29.



Gambar C. 29 Grafik Alir Manajemen Data Dokter Fungsi *Add*
 Sumber: (Hasil analisis)

2.4. Grafik Alir Fitur Manajemen Data Dokter Fungsi *Edit*

Grafik alir fitur Manajemen Data Dokter Fungsi *Edit* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.30.



Gambar C. 30 Diagram Alir Manajemen Data Dokter Fungsi *Edit*
 Sumber: (Hasil analisis)

2.5. Grafik Alir Fitur Manajemen Data Dokter Fungsi *Delete*

Grafik alir fitur Manajemen Data Dokter Fungsi *Delete* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.31.



Gambar C. 31 Grafik Alir Manajemen Data Dokter Fungsi *Delete*
Sumber: (Hasil analisis)

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur Manajemen Data Dokter dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2$$

Dengan $V(G)$ adalah Kompleksitas siklomatis, E adalah *Edge* atau tepi atau garis dari sebuah grafik, dan N adalah *Node* atau titik dari sebuah grafik. Maka perhitungan dari setiap grafik dari Manajemen Data Dokter adalah sebagai berikut.

3.1. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Dokter Fungsi *construct*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 8 - 8 + 2 \\ &= 2 \end{aligned}$$

3.2. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Dokter Fungsi *Index*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

3.3. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Dokter Fungsi *Add*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 17 - 15 + 2 \\ &= 4 \end{aligned}$$

3.4. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Dokter Fungsi *Edit*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 12 - 11 + 2 \\ &= 3 \end{aligned}$$

3.5. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Dokter Fungsi

Delete

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

4. *Basis Set*

4.1. *Basis set* Fitur Manajemen Data Dokter Fungsi *Construct*

Basis set Manajemen Data Dokter fungsi *construct* menghasilkan dua (2) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 6 – 7 – 36

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 7 – 36

4.2. *Basis set* Fitur Manajemen Data Dokter Fungsi *Index*

Basis set Manajemen Data Dokter fungsi *index* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 8 – 9 – 10 – 36

4.3. *Basis set* Fitur Manajemen Data Dokter Fungsi *Add*

Basis set Manajemen Data Dokter fungsi *Add* menghasilkan empat (4) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 11 – 18 – 19 – 20 – 22 – 12 – 13 – 14 – 15 – 23 – 36

Jalur 2 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 13 – 14 – 15 – 23 – 36

Jalur 3 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 13 – 14 – 16 – 23 – 36

Jalur 4 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 17 – 23 – 36

4.4. *Basis set* Fitur Manajemen Data Dokter Fungsi *Edit*

Basis set Manajemen Data Dokter fungsi *Edit* menghasilkan tiga (3) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 24 – 25 – 26 – 31 – 32 – 36

Jalur 2 : 1 – 24 – 25 – 26 – 27 – 28 – 30 – 32 – 36

Jalur 3 : 1 – 24 – 25 – 26 – 27 – 28 – 29 – 32 – 36

4.5. *Basis set* Fitur Manajemen Data Dokter Fungsi *Delete*

Basis set Manajemen Data Dokter fungsi *Delete* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 33 – 34 – 35 – 36

5. *Test Case*

Tabel C. 57 *Test Case* Fitur Manajemen Data Dokter fungsi *construct* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif kemudian klik menu Dokter
Target yang diharapkan	Sistem menampilkan halaman Dokter
Hasil pengujian	Benar
Jalur	Jalur 1
File	dokter.php

Tabel C. 58 *Test Case* Fitur Manajemen Data Dokter fungsi *construct* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Dokter, <i>User</i> melakukan <i>Logout</i> dan menuju halaman <i>Login</i> dan ketika di <i>back</i> tetap menampilkan halaman <i>Login</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	dokter.php

Tabel C. 59 *Test Case* Fitur Manajemen Data Dokter fungsi *Index* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif lalu klik menu Dokter
Target yang diharapkan	Sistem menampilkan halaman Dokter
Hasil pengujian	Benar
Jalur	Jalur 1
File	dokter.php

Tabel C. 60 *Test Case* Fitur Manajemen Data Dokter fungsi *Add* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Dokter, <i>User</i> mengklik Tombol Tambah Dokter kemudian mengisi <i>form</i> Tambah Dokter setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Dokter serta menampilkan <i>alert</i> ,” <i>Successfully add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 1
File	dokter.php

Tabel C. 61 *Test Case* Fitur Manajemen Data Dokter fungsi *Add* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Dokter, <i>User</i> mengklik Tombol Tambah Dokter kemudian mengisi <i>form</i> Tambah Dokter setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Dokter serta menampilkan <i>alert</i> ,” <i>Successfully add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 2
File	dokter.php

Tabel C. 62 *Test Case* Fitur Manajemen Data Dokter fungsi *Add* jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Dokter, <i>User</i> mengklik Tombol Tambah Dokter kemudian mengisi <i>form</i> Tambah Dokter setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Dokter dan menampilkan <i>alert</i> ,” <i>Failed add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 3
File	dokter.php

Tabel C. 63 *Test Case* Fitur Manajemen Data Dokter fungsi *Add* jalur 4
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Dokter, <i>User</i> mengklik Tombol Tambah Dokter
Target yang diharapkan	Sistem menampilkan halaman Tambah Dokter
Hasil pengujian	Benar
Jalur	Jalur 4
File	dokter.php

Tabel C. 64 *Test Case* Fitur Manajemen Data Dokter fungsi *Edit* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Dokter, <i>User</i> mengklik Tombol <i>Edit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Edit</i> Dokter
Hasil pengujian	Benar
Jalur	Jalur 1
File	dokter.php

Tabel C. 65 *Test Case* Fitur Manajemen Data Dokter fungsi *Edit* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Dokter, <i>User</i> mengisi <i>form edit</i> Dokter kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Dokter dan menampilkan <i>alert," Failed update data.."</i>
Hasil pengujian	Benar
Jalur	Jalur
File	dokter.php

Tabel C. 66 *Test Case* Fitur Manajemen Data Dokter fungsi *Edit* jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Dokter, <i>User</i> mengisi <i>form edit</i> Dokter kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Dokter dan menampilkan <i>alert," Successfully update data.."</i>

Hasil pengujian	Benar
Jalur	Jalur 3
File	dokter.php

Tabel C. 67 Test Case Fitur Manajemen Data Dokter fungsi Delete jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Dokter, User mengklik tombol <i>Remove</i>
Target yang diharapkan	Sistem menampilkan halaman Dokter dan menampilkan <i>alert, "Successfully delete data.."</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	dokter.php

C. 7 Manajemen Data Obat

Pengujian kode – kode program pada fitur Manajemen Data Obat adalah sebagai berikut:

1. Listing Program

```

1. 1 class Obat extends CI_Controller {
2.
3.   2 public function __construct() {
4.     parent::__construct();
5.     $this->load->model("obat_model", "obat"); 3
6.     // $this->authorization->isAdminLogin();
7.     4 if (!$this->session->userdata("rkmds-logged-in")) {
8.       redirect('/', 5);
9.     }
10.  } 6
11.
12.   public function index() { 8
13.     $data = array();
14.     9 $data['obats'] = $this->obat->getDataobat();
15.     $this->load->view('obat-data', $data);
16.  } 10
17.
18.   public function add() { 11
19.     12 if ($this->input->post("submit")) {
20.       $nama = $this->input->post("nama");
21.       $jenis = $this->input->post("jenis");
22.       $satuan = $this->input->post("satuan");
23.       $stok = $this->input->post("stok");
24.       $harga_pokok = $this->input->post("harga_pokok");
25.       13 $harga_jual = $this->input->post("harga_jual");
26.       $kode_obat = $this->input->post("kode_obat");

```

```

27.         $data = array(
28.             "nama"=>$nama,
29.             "jenis"=>$jenis,
30.             "satuan"=>$satuan,
31.             "stok"=>$stok,
32.             "harga_pokok"=>$harga_pokok,
33.             "harga_jual"=>$harga_jual,
34.             "kode_obat"=>$kode_obat,
35.         );
36.
37.         14 if($this->obat->insert($data){
38.             $this->session-
39. >set_flashdata('message', 'Successfully add new data. '); 15
40.             redirect("obat");
41.         }else{
42.             $this->session-
43. >set_flashdata('message', 'Failed add new data. '); 16
44.             redirect("obat");
45.         }
46.     }
47.     $id = "";
48.     18 $sql = 'SELECT COUNT(*) AS total FROM tb_obat ';
49.     $total = $this->db->query($sql)->row('total');
50.     19 if($total==0){ //never made
51.         $autonum = "000 "; 20
52.     }
53.     else{
54.         21 $sql = 'SELECT * FROM tb_obat ORDER BY id_obat DESC LIMIT 1 '
55. ;
56.         $last = $this->db->query($sql)->row('kode_obat');
57.         $lastnum = substr($last,4,4);
58.         $autonum = (int) $lastnum + 1;
59.         $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
60.     }
61.     22 $id = "OBT-".$autonum;
62.     $data['id'] = $id;
63.     $this->load->view('add-obat', $data);
64. }
65.
66. public function edit($id){ 24
67.     $data = array();
68.     25 $obat = $this->obat->getobat($id);
69.     $data['obat'] = $obat;
70.     $this->load->view('edit-obat', $data);
71.     26 if($this->input->post("submit")){
72.         $nama = $this->input->post("nama");
73.         $jenis = $this->input->post("jenis");
74.         $satuan = $this->input->post("satuan");
75.         $stok = $this->input->post("stok");
76.         $harga_pokok = $this->input->post("harga_pokok");
77.         $harga_jual = $this->input->post("harga_jual");
78.         27 $id_obat = $this->input->post("id_obat");
79.         $data = array(

```

```

80.         "nama"=>$nama,
81.         "jenis"=>$jenis,
82.         "satuan"=>$satuan,
83.         "stok"=>$stok,
84.         "harga_pokok"=>$harga_pokok,
85.         "harga_jual"=>$harga_jual,
86.     );
87.
88.     28 if($this->obat->update($id_obat,$data)){
89.         $this->session-
90.     >set_flashdata('message', 'Successfully update data. '); 29
91.         redirect("obat");
92.     }else{
93.         30 $this->session-
94.     >set_flashdata('message', 'Failed update new data. ');
95.         redirect("obat");
96.     }
97. }
98. } 31
99. } 32
100. public function delete($id){ 33
101.     $this->obat->delete($id);
102.     $this->session-
103. >set_flashdata('message', 'Successfully delete data. '); 34
104.     redirect("obat");
105. } 35
106. } 36
    
```

2. Grafik Alir

Grafik Alir fitur Manajemen Data Obat berupa notasi-notasi yang menunjukkan alur kontrol. Fitur Manajemen Data Obat terbagi menjadi beberapa fungsi, yaitu *construct*, *index*, *add*, *edit*, dan *delete*.

2.1. Grafik Alir Fitur Manajemen Data Obat Fungsi *Construct*

Grafik alir fitur Manajemen Data Obat Fungsi *Construct* digambarkan dengan simbol dan garis dapat di lihat pada Gambar C.32.



Gambar C. 32 Grafik Alir Manajemen Data Obat Fungsi *Construct*
 Sumber: (Hasil analisis)

2.2. Grafik Alir Fitur Manajemen Data Obat Fungsi *Index*

Grafik alir fitur Manajemen Data Obat Fungsi *Index* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.33.



Gambar C. 33 Grafik Alir Manajemen Data Obat Fungsi *Index*
 Sumber: (Hasil analisis)

2.3. Grafik Alir Fitur Manajemen Data Obat Fungsi *Add*

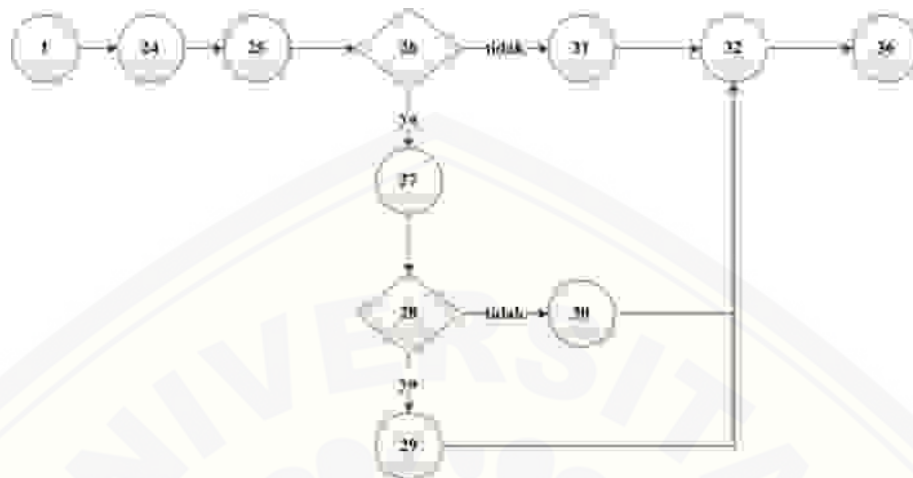
Grafik alir fitur Manajemen Data Obat Fungsi *Add* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.34.



Gambar C. 34 Grafik Alir Manajemen Data Obat Fungsi *Add*
 Sumber: (Hasil analisis)

2.4. Grafik Alir Fitur Manajemen Data Obat Fungsi *Edit*

Grafik alir fitur Manajemen Data Obat Fungsi *Edit* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.35.



Gambar C. 35 Diagram Alir Manajemen Data Obat Fungsi *Edit*
 Sumber: (Hasil analisis)

2.5. Grafik Alir Fitur Manajemen Data Obat Fungsi *Delete*

Grafik alir fitur Manajemen Data Obat Fungsi *Delete* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.36.



Gambar C. 36 Grafik Alir Manajemen Data Obat Fungsi *Delete*
 Sumber: (Hasil analisis)

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur Manajemen Data Obat dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2$$

Dengan $V(G)$ adalah Kompleksitas siklomatis, E adalah *Edge* atau tepi atau garis dari sebuah grafik, dan N adalah *Node* atau titik dari sebuah grafik. Maka perhitungan dari setiap grafik dari Manajemen Data Obat adalah sebagai berikut.

3.1. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Obat Fungsi *construct*

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 8 - 8 + 2 \\ &= 2\end{aligned}$$

3.2. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Obat Fungsi *Index*

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1\end{aligned}$$

3.3. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Obat Fungsi *Add*

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 17 - 15 + 2 \\ &= 4\end{aligned}$$

3.4. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Obat Fungsi *Edit*

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 12 - 11 + 2 \\ &= 3\end{aligned}$$

3.5. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Obat Fungsi *Delete*

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1\end{aligned}$$

4. *Basis Set*

4.1. *Basis set* Fitur Manajemen Data Obat Fungsi *Construct*

Basis set Manajemen Data Obat fungsi *construct* menghasilkan dua (2) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 6 – 7 – 36

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 7 – 36

4.2. *Basis set* Fitur Manajemen Data Obat Fungsi *Index*

Basis set Manajemen Data Obat fungsi *index* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 8 – 9 – 10 – 36

4.3. *Basis set* Fitur Manajemen Data Obat Fungsi *Add*

Basis set Manajemen Data Obat fungsi *Add* menghasilkan empat (4) jalur

independent dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 11 – 18 – 19 – 20 – 22 – 12 – 13 – 14 – 15 – 23 – 36

Jalur 2 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 13 – 14 – 15 – 23 – 36

Jalur 3 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 13 – 14 – 16 – 23 – 36

Jalur 4 : 1 – 11 – 18 – 19 – 21 – 22 – 12 – 17 – 23 – 36

4.4. *Basis set* Fitur Manajemen Data Obat Fungsi *Edit*

Basis set Manajemen Data Obat fungsi *Edit* menghasilkan tiga (3) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 24 – 25 – 26 – 31 – 32 – 36

Jalur 2 : 1 – 24 – 25 – 26 – 27 – 28 – 30 – 32 – 36

Jalur 3 : 1 – 24 – 25 – 26 – 27 – 28 – 29 – 32 – 36

4.5. *Basis set* Fitur Manajemen Data Obat Fungsi *Delete*

Basis set Manajemen Data Obat fungsi *Delete* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 33 – 34 – 35 – 36

5. *Test Case*

Tabel C. 68 *Test Case* Fitur Manajemen Data Obat fungsi *construct* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif kemudian klik menu Obat
Target yang diharapkan	Sistem menampilkan halaman Obat
Hasil pengujian	Benar
Jalur	Jalur 1
File	obat.php

Tabel C. 69 *Test Case* Fitur Manajemen Data Obat fungsi *construct* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Obat, <i>User</i> melakukan <i>Logout</i> dan menuju halaman <i>Login</i> dan ketika di <i>back</i> tetap menampilkan halaman <i>Login</i>
-----------	---

Target yang diharapkan	Sistem menampilkan halaman <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	obat.php

Tabel C. 70 *Test Case* Fitur Manajemen Data Obat fungsi *Index* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif lalu klik menu Obat
Target yang diharapkan	Sistem menampilkan halaman Obat
Hasil pengujian	Benar
Jalur	Jalur 1
File	obat.php

Tabel C. 71 *Test Case* Fitur Manajemen Data Obat fungsi *Add* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Obat, <i>User</i> mengklik Tombol Tambah Obat kemudian mengisi <i>form</i> Tambah Obat setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Obat serta menampilkan <i>alert</i> ,” <i>Successfully add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 1
File	obat.php

Tabel C. 72 *Test Case* Fitur Manajemen Data Obat fungsi *Add* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Obat, <i>User</i> mengklik Tombol Tambah Obat kemudian mengisi <i>form</i> Tambah Obat setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Obat serta menampilkan <i>alert</i> ,” <i>Successfully add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 2
File	obat.php

Tabel C. 73 *Test Case* Fitur Manajemen Data Obat fungsi *Add* jalur 3

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Obat, <i>User</i> mengklik Tombol Tambah Obat kemudian mengisi <i>form</i> Tambah Obat setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Obat dan menampilkan <i>alert, "Failed add new data.."</i>
Hasil pengujian	Benar
Jalur	Jalur 3
File	obat.php

Tabel C. 74 *Test Case* Fitur Manajemen Data Obat fungsi *Add* jalur 4

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Obat, <i>User</i> mengklik Tombol Tambah Obat
Target yang diharapkan	Sistem menampilkan halaman Tambah Obat
Hasil pengujian	Benar
Jalur	Jalur 4
File	obat.php

Tabel C. 75 *Test Case* Fitur Manajemen Data Obat fungsi *Edit* jalur 1

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Obat, <i>User</i> mengklik Tombol <i>Edit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Edit</i> Obat
Hasil pengujian	Benar
Jalur	Jalur 1
File	obat.php

Tabel C. 76 *Test Case* Fitur Manajemen Data Obat fungsi *Edit* jalur 2

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Obat, <i>User</i> mengisi <i>form edit</i> Obat kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Obat dan menampilkan <i>alert, "Failed update"</i>

	<i>data..”</i>
Hasil pengujian	Benar
Jalur	Jalur
File	obat.php

Tabel C. 77 Test Case Fitur Manajemen Data Obat fungsi *Edit* jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Obat, <i>User</i> mengisi <i>form edit</i> Obat kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Obat dan menampilkan <i>alert,” Successfully update data..”</i>
Hasil pengujian	Benar
Jalur	Jalur 3
File	obat.php

Tabel C. 78 Test Case Fitur Manajemen Data Obat fungsi *Delete* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Obat, <i>User</i> mengklik tombol <i>Remove</i>
Target yang diharapkan	Sistem menampilkan halaman Obat dan menampilkan <i>alert,” Successfully delete data..”</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	obat.php

C. 8 Manajemen Data Laborat

Pengujian kode-kode program pada fitur Manajemen Data Laborat adalah sebagai berikut:

1. Listing Program

```

1. 1 class Laborat extends CI_Controller {
2.
3.     public function __construct() { 2
4.         parent::__construct();
5.         $this->load->model("laborat_model", "laborat");
6.         // $this->authorization->isAdminLogin();
7.         if (!$this->session->userdata("rkmds-logged-in")) { 4
8.             redirect('/', 'refresh', true); 5

```

```

9.
10.     }7} }6
11.
12.     public function home() {8
13.         $this->load->view('index');9
14.     }10}
15.
16.     public function index() {11
17.         $data = array();
18.         12 $data['laborats'] = $this->laborat->getDataLaborat();
19.         $this->load->view('laborat-data', $data);
20.     }13}
21.
22.     public function add() {14
23.         if($this->input->post("submit") {15
24.             $nama = $this->input->post("nama");
25.             $jenis = $this->input->post("jenis");
26.             $biaya = $this->input->post("biaya");
27.             $keperluan = $this->input->post("keperluan");
28.             $id_pasien = $this->input->post("id_pasien");
29.             $anjuran_dari = $this->input->post("anjuran_dari");
30.             $hasil_pemeriksaan = $this->input-
>post("hasil_pemeriksaan");
31.             16 $kode_laborat = $this->input->post("kode_laborat");
32.             //$hasil_pemeriksaan = "";
33.             $hasil = $this->input->post("hasil");
34.             //$hasil = $_POST["hasil"];
35.
36.             $data = array(
37.                 "nama"=>$nama,
38.                 "jenis"=>$jenis,
39.                 "biaya"=>$biaya,
40.                 "keperluan"=>$keperluan,
41.                 "id_pasien"=>$id_pasien,
42.                 "hasil_pemeriksaan"=>$hasil_pemeriksaan,
43.                 "kode_laborat"=>$kode_laborat,
44.                 "anjuran_dari"=>$anjuran_dari,
45.             );
46.
47.             17 if($this->laborat->insert($data, $hasil)) {
48.                 $this->session-
>set_flashdata('message', 'Successfully add new data. '); 18
49.                 redirect("laborat");
50.             }else{
51.                 19 $this->session-
>set_flashdata('message', 'Failed add new data. ');
52.                 redirect("laborat");
53.             }
54.         }
55.         20 $id = "";
56.         21 $sql = 'SELECT COUNT(*) AS total FROM tb_laborat ';
57.         $total = $this->db->query($sql)->row('total');
58.
59.
60.         22 if($total==0) { //never made
61.             $autonum = "0001"; 23

```

```

62.     }
63.     else{
64.         MIT 1'. 24 $sql = 'SELECT * FROM tb_laborat ORDER BY id_laborat DESC LI
65.             $last = $this->db->query($sql)->row('kode_laborat');
66.             $lastnum = substr($last,4,4);
67.             $autonum = (int) $lastnum + 1;
68.             $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
69.         }
70.         $sid = "LBR-".$autonum;
71.         $data['id'] = $sid;
72.         $data['pemeriksaan']=$this->db->get('tb_pemeriksaan')-
73.         >result_array();
74.         $this->load->view('add-laborat',$data);
75.     26 }
76.
77.     public function delete($id){ 27
78.         $this->laborat->delete($id);
79.         28 $this->session-
80.         >set_flashdata('message', 'Successfully delete data. ');
81.         redirect("laborat");
82.     29 }
83.     public function logout(){ 30
84.         $user_data = array(
85.             'rkmds-id'=>',
86.             'rkmds-username'=>',
87.             31 'rkmds-name'=>',
88.             'rkmds-logged-in'=>FALSE,
89.             'rkmds-role'=>'
90.         );
91.         $this->session->unset_userdata($user_data);
92.         redirect('admin');
93.     32 }
94.
95.     function pdf($id_laborat){ 33
96.         //$this->load->helper('pdf_helper');
97.         //$this->load->library('pdfku');
98.         $data= array();
99.         $data['laborat'] = $this->db
100.             ->where("id_laborat",$id_laborat)
101.             -
102.             >select('tb_laborat.*, tb_pasien.nama as nama_pasien, tb_pasien.kode_pas
103.                 ien', false)
104.             -
105.             >join('tb_pasien', 'tb_laborat.id_pasien = tb_pasien.id_pasien')
106.             ->get("tb_laborat");
107.
108.         34 $data['laborat_detail'] = $this->db
109.             ->where("tb_laborat_detail.id_laborat",$id_laborat)
110.             -
111.             >select('tb_pemeriksaan.*,tb_laborat_detail.hasil', false)
112.             -
113.             >join('tb_pemeriksaan', 'tb_laborat_detail.id_pemeriksaan = tb_pemeriksa
114.                 an.id_pemeriksaan')

```

```

109.         ->get("tb_laborat_detail");
110.         $this->load->view('pdf/laborat', $data);
111.     } 35
112. } 36

```

2. Grafik Alir

Grafik Alir fitur Manajemen Data Laborat berupa notasi-notasi yang menunjukkan alur kontrol. Fitur Manajemen Data Laborat terbagi menjadi beberapa fungsi, yaitu *construct*, *home*, *index*, *add*, *delete*, *logout*, dan *pdf*.

2.1. Grafik Alir Fitur Manajemen Data Laborat Fungsi *Construct*

Grafik alir fitur Manajemen Data Laborat Fungsi *Construct* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.37.



Gambar C. 37 Grafik Alir Manajemen Data Laborat Fungsi *Construct*
Sumber: (Hasil analisis)

2.2. Grafik Alir Fitur Manajemen Data Laborat Fungsi *home*

Grafik alir fitur Manajemen Data Laborat Fungsi *home* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.38.



Gambar C. 38 Grafik Alir Manajemen Data Laborat Fungsi *home*
Sumber: (Hasil analisis)

2.3. Grafik Alir Fitur Manajemen Data Laborat Fungsi *Index*

Grafik alir fitur Manajemen Data Laborat Fungsi *Index* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.39.

2.6. Grafik Alir Fitur Manajemen Data Laborat Fungsi *Logout*

Grafik alir fitur Manajemen Data Laborat Fungsi *Logout* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.42.



Gambar C. 42 Grafik Alir Manajemen Data Laborat Fungsi Logout
 Sumber: (Hasil analisis)

2.7. Grafik Alir Fitur Manajemen Data Laborat Fungsi *pdf*

Grafik alir fitur Manajemen Data Laborat Fungsi *pdf* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.43.



Gambar C. 43 Grafik Alir Manajemen Data Laborat Fungsi pdf
 Sumber: (Hasil analisis)

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur Manajemen Data Laborat dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2$$

Dengan $V(G)$ adalah Kompleksitas siklomatis, E adalah *Edge* atau tepi atau garis dari sebuah grafik, dan N adalah *Node* atau titik dari sebuah grafik. Maka perhitungan dari setiap grafik dari Manajemen Data Laborat adalah sebagai berikut.

3.1. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Laborat Fungsi *construct*

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 8 - 8 + 2 \\
 &= 2
 \end{aligned}$$

3.2. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Laborat Fungsi *home*.

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 4 - 5 + 2
 \end{aligned}$$

$$= 1$$

- 3.3. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Laborat Fungsi *index*.

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

- 3.4. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Laborat Fungsi *add*.

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 17 - 15 + 2 \\ &= 4 \end{aligned}$$

- 3.5. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Laborat Fungsi *delete*.

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

- 3.6. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Laborat Fungsi *logout*.

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

- 3.7. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Laborat Fungsi *pdf*.

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

4. Basis Set

- 4.1. *Basis set* Fitur Manajemen Data Laborat Fungsi *Construct*

Basis set Manajemen Data Laborat fungsi *construct* menghasilkan dua (2) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

$$\text{Jalur 1 : } 1 - 2 - 3 - 4 - 6 - 7 - 36$$

$$\text{Jalur 2 : } 1 - 2 - 3 - 4 - 5 - 7 - 36$$

- 4.2. *Basis set* Fitur Manajemen Data Laborat Fungsi *Home*

Basis set Manajemen Data Laborat fungsi *index* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 8 – 9 – 10 – 36

4.3. *Basis set* Fitur Manajemen Data Laborat Fungsi *index*

Basis set Manajemen Data Laborat fungsi *index* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 11 – 12 – 13 – 36

4.4. *Basis set* Fitur Manajemen Data Laborat Fungsi *Add*

Basis set Manajemen Data Laborat fungsi *Add* menghasilkan empat (4) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 14 – 21 – 22 – 23 – 25 – 15 – 16 – 17 – 18 – 23 – 36

Jalur 2 : 1 – 14 – 21 – 22 – 24 – 25 – 15 – 16 – 17 – 18 – 23 – 36

Jalur 3 : 1 – 14 – 21 – 22 – 24 – 25 – 15 – 16 – 17 – 19 – 23 – 36

Jalur 4 : 1 – 14 – 21 – 22 – 24 – 25 – 15 – 20 – 23 – 36

4.5. *Basis set* Fitur Manajemen Data Laborat Fungsi *Delete*

Basis set Manajemen Data Laborat fungsi *Delete* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 27 – 28 – 29 – 36

4.6. *Basis set* Fitur Manajemen Data Laborat Fungsi *Logout*

Basis set Manajemen Data Laborat fungsi *Delete* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 30 – 31 – 32 – 36

4.7. *Basis set* Fitur Manajemen Data Laborat Fungsi *Pdf*

Basis set Manajemen Data Laborat fungsi *Delete* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 33 – 34 – 35 – 36

5. Test Case

Tabel C. 79 Test Case Fitur Manajemen Data Laborat fungsi *construct* jalur 1

Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif kemudian klik menu Laborat
Target yang diharapkan	Sistem menampilkan halaman Laborat
Hasil pengujian	Benar
Jalur	Jalur 1
File	laborat.php

Tabel C. 80 Test Case Fitur Manajemen Data Laborat fungsi *construct* jalur 2

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laborat, <i>User</i> melakukan <i>Logout</i> dan menuju halaman <i>Login</i> dan ketika di <i>back</i> tetap menampilkan halaman <i>Login</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	laborat.php

Tabel C. 81 Test Case Fitur Manajemen Data Laborat fungsi *home* jalur 1

Sumber: (Hasil analisis)

Test case	<i>User</i> melakukan <i>Login</i> ke sistem dengan <i>role=laborat</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif lalu klik menu Laborat
Target yang diharapkan	Sistem menampilkan halaman Laborat
Hasil pengujian	Benar
Jalur	Jalur 1
File	laborat.php

Tabel C. 82 Test Case Fitur Manajemen Data Laborat fungsi *Index* jalur 1

Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif lalu klik menu Laborat
Target yang diharapkan	Sistem menampilkan halaman Laborat
Hasil pengujian	Benar

Jalur	Jalur 1
File	laborat.php

Tabel C. 83 *Test Case* Fitur Manajemen Data Laborat fungsi *Add* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laborat, <i>User</i> mengklik Tombol Tambah Laborat kemudian mengisi <i>form</i> Tambah Laborat setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Laborat serta menampilkan <i>alert</i> ,” <i>Successfully add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 1
File	laborat.php

Tabel C. 84 *Test Case* Fitur Manajemen Data Laborat fungsi *Add* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laborat, <i>User</i> mengklik Tombol Tambah Laborat kemudian mengisi <i>form</i> Tambah Laborat setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Laborat serta menampilkan <i>alert</i> ,” <i>Successfully add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 2
File	laborat.php

Tabel C. 85 *Test Case* Fitur Manajemen Data Laborat fungsi *Add* jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laborat, <i>User</i> mengklik Tombol Tambah Laborat kemudian mengisi <i>form</i> Tambah Laborat setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Laborat dan menampilkan <i>alert</i> ,” <i>Failed add new data.</i> ”
Hasil pengujian	Benar
Jalur	Jalur 3
File	laborat.php

Tabel C. 86 *Test Case* Fitur Manajemen Data Laborat fungsi *Add* jalur 4
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laborat, <i>User</i> mengklik Tombol Tambah Laborat
Target yang diharapkan	Sistem menampilkan halaman Tambah Laborat
Hasil pengujian	Benar
Jalur	Jalur 4
File	laborat.php

Tabel C. 87 *Test Case* Fitur Manajemen Data Laborat fungsi *Delete* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laborat, <i>User</i> mengklik tombol <i>Remove</i>
Target yang diharapkan	Sistem menampilkan halaman Laborat dan menampilkan <i>alert, "Successfully delete data.."</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	laborat.php

Tabel C. 88 *Test Case* Fitur Manajemen Data Laborat fungsi *Logout* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> mengklik tombol <i>Logout</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	laborat.php

Tabel C. 89 *Test Case* Fitur Manajemen Data Laborat fungsi *Pdf* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laborat, <i>User</i> mengklik tombol Cetak
Target yang diharapkan	Sistem menampilkan halaman "PEMERIKSAAN LABORATORIUM"
Hasil pengujian	Benar
Jalur	Jalur 1
File	laborat.php

C. 9 Manajemen Data Rujukan

Pengujian kode-kode program pada fitur Manajemen Data Rujukan adalah sebagai berikut:

1. Listing Program

```

1. 1 class Rujukan extends CI_Controller {
2.
3.     3 public function __construct() { 2
4.         parent::__construct();
5.         $this->load->model("rujukan_model", "rujukan");
6.         4 if (!$this->session->userdata("rkmds-logged-in")) { 4
7.             5 redirect('/', 'index');
8.         } 6
9.     } 7
10.
11.     public function index() { 8
12.         $data = array();
13.         9 $data['rujukans'] = $this->rujukan->getDatarujukan();
14.         $this->load->view('rujukan-data', $data);
15.     } 10
16.
17.     public function edit($id_rujukan) { 11
18.         $data = array();
19.         12 $rujukan = $this->rujukan->getrujukan($id_rujukan);
20.         $data['rujukan'] = $rujukan;
21.         $this->load->view('edit-rujukan', $data);
22.     } 13
23.
24.     14 public function add() {
25.         15 if ($this->input->post("submit")) { 15
26.             $no_transaksi = $this->input->post("no_transaksi");
27.             $tanggal_transaksi = $this->input->
28.                 >post("tanggal_transaksi");
29.             $kode_pasien = $this->input->post("kode_pasien");
30.             $nama = $this->input->post("nama");
31.             $id_dokter = $this->input->post("id_dokter");
32.             $tujuan = $this->input->post("tujuan");
33.             $keterangan = $this->input->post("keterangan");
34.
35.             16 $data = array(
36.                 "no_transaksi"=>$no_transaksi,
37.                 "tanggal_transaksi"=>date("Y-m-d H:i:s"),
38.                 "kode_pasien"=>$kode_pasien,
39.                 "nama"=>$nama,
40.                 "id_dokter"=>$id_dokter,
41.                 "tujuan"=>$tujuan,
42.                 "keterangan"=>$keterangan,
43.             );
44.             17 if ($this->rujukan->insert($data)) {
45.                 $this->session->
46.                 >set_flashdata('message', 'Successfully add new data. '); 18

```

```

46.         redirect("rujukan");
47.     }else{
48.         19 $this->session-
         >set_flashdata('message', 'Failed add new data. ');
49.         redirect("rujukan");
50.     }
51.     20 }
52.     $id = "";
53.     21 $sql = 'SELECT COUNT(*) AS total FROM tb_rujukan ';
54.     $total = $this->db->query($sql)->row('total');
55.
56.     22 if($total==0){ //never made
57.         $autonum = "000"; 23
58.     }
59.     else{
60.         24 $sql = 'SELECT * FROM tb_rujukan ORDER BY id_rujukan DES
        C LIMIT 1';
61.         $last = $this->db->query($sql)->row('no_transaksi');
62.         $lastnum = substr($last,4,4);
63.         $autonum = (int) $lastnum + 1;
64.         $autonum = str_pad($autonum, 4, "0", STR_PAD_LEFT);
65.     }
66.     $id = "RJK-".$autonum;
67.     25 $data['id'] = $id;
68.     $this->load->view('add-rujukan', $data);
69.     26 }
70.
71.     public function insert(){ 27
72.         if($this->input->post("submit")){28
73.             $no_transaksi = $this->input->post("no_transaksi");
74.             $id_pasien = $this->input->post("id_pasien");
75.             $id_dokter = $this->input->post("id_dokter");
76.             $tujuan = $this->input->post("tujuan");
77.             29 $keterangan = $this->input->post("keterangan");
78.
79.             $data = array(
80.                 "no_transaksi"=>$no_transaksi,
81.                 "tanggal_transaksi"=>date("Y-m-d H:i:s"),
82.                 "id_pasien"=>$id_pasien,
83.                 "id_dokter"=>$id_dokter,
84.                 "tujuan"=>$tujuan,
85.                 "keterangan"=>$keterangan,
86.             );
87.
88.             30 if($this->rujukan->insert($data)){
89.                 $this->session-
                 >set_flashdata('message', 'Successfully add new data. ');31
90.                 redirect("rujukan");
91.             }else{
92.                 32 $this->session-
                 >set_flashdata('message', 'Failed add new data. ');
93.                 redirect("rujukan");
94.             }
95.         } 33
96.     } 34
97.

```

```

98. public function update(){
99. 35 if($this->input->post("submit")){36
100.     $no_transaksi = $this->input->post("no_transaksi");
101.     $id_pasien = $this->input->post("id_pasien");
102.     $id_dokter = $this->input->post("id_dokter");
103.     $tujuan = $this->input->post("tujuan");
104.     $keterangan = $this->input->post("keterangan");
105.     $id_rujukan = $this->input->post("id_rujukan");
106.
107.     37 $data = array(
108.         "no_transaksi"=>$no_transaksi,
109.         "id_pasien"=>$id_pasien,
110.         "id_dokter"=>$id_dokter,
111.         "tujuan"=>$tujuan,
112.         "keterangan"=>$keterangan,
113.     );
114.     38 if($this->rujukan->update($id_rujukan, $data){
115.         $this->session-
116.         >set_flashdata('message', 'Successfully update data. '); 39
117.         redirect("rujukan");
118.     }else{
119.         40 $this->session-
120.         >set_flashdata('message', 'Failed update new data. ');
121.         redirect("rujukan");
122.     }
123. }
124. 42 } 41
125. public function delete($id_rujukan){43
126.     $this->rujukan->delete($id_rujukan);
127.     $this->session-
128.     >set_flashdata('message', 'Successfully delete data. '); 44
129.     redirect("rujukan");
130. }
131. 45 }
132. private function generateNomorTransaksi(){
133.     return time()."";
134. }
135. 46function pdf($id_rujukan){
136.     $data= array();
137.     $data['rujukan'] = $this->db
138.     ->where("id_rujukan",$id_rujukan)
139.     -
140.     >select('tb_rujukan.no_transaksi, tb_rujukan.tanggal_transaksi, tb_pasie
141.     n.id_pasien,tb_pasien.kode_pasien, tb_pasien.nama as nama_pasien, tb_dok
142.     ter.nama as nama_dokter, tb_rujukan.tujuan, tb_rujukan.keterangan', fals
143.     e)
144.     -
145.     >join('tb_pasien', 'tb_rujukan.id_pasien = tb_pasien.id_pasien')
146.     -
147.     >join('tb_dokter', 'tb_rujukan.id_dokter = tb_dokter.id_dokter')
148.     -
149.     >get("tb_rujukan");
150.     $data['rekammedis'] = $this->db
151.     >where("id_pasien",$data['rujukan']->row()->id_pasien)

```

```

144.         //-
145.         >select('amamnesis, diagnosa, hasil_pemeriksaan, jenis_pemeriksaan')
146.         ->get("tb_rekammedis");
147.         $this->load->view('pdf/rujukan', $data);
148.     } 48
149. }
    
```

2. Grafik Alir

Grafik Alir fitur Manajemen Data Rujukan berupa notasi-notasi yang menunjukkan alur kontrol. Fitur Manajemen Data Rujukan terbagi menjadi beberapa fungsi, yaitu *construct*, *index*, *edit*, *add*, *insert*, *update*, *delete*, dan *pdf*.

2.1. Grafik Alir Fitur Manajemen Data Rujukan Fungsi *Construct*

Grafik alir fitur Manajemen Data Rujukan Fungsi *Construct* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.44.



Gambar C. 44 Grafik Alir Manajemen Data Rujukan Fungsi *Construct*
 Sumber: (Hasil analisis)

2.2. Grafik Alir Fitur Manajemen Data Rujukan Fungsi *Index*

Grafik alir fitur Manajemen Data Rujukan Fungsi *Index* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.45.



Gambar C. 45 Grafik Alir Manajemen Data Rujukan Fungsi *Index*
 Sumber: (Hasil analisis)

2.3. Grafik Alir Fitur Manajemen Data Rujukan Fungsi *Edit*

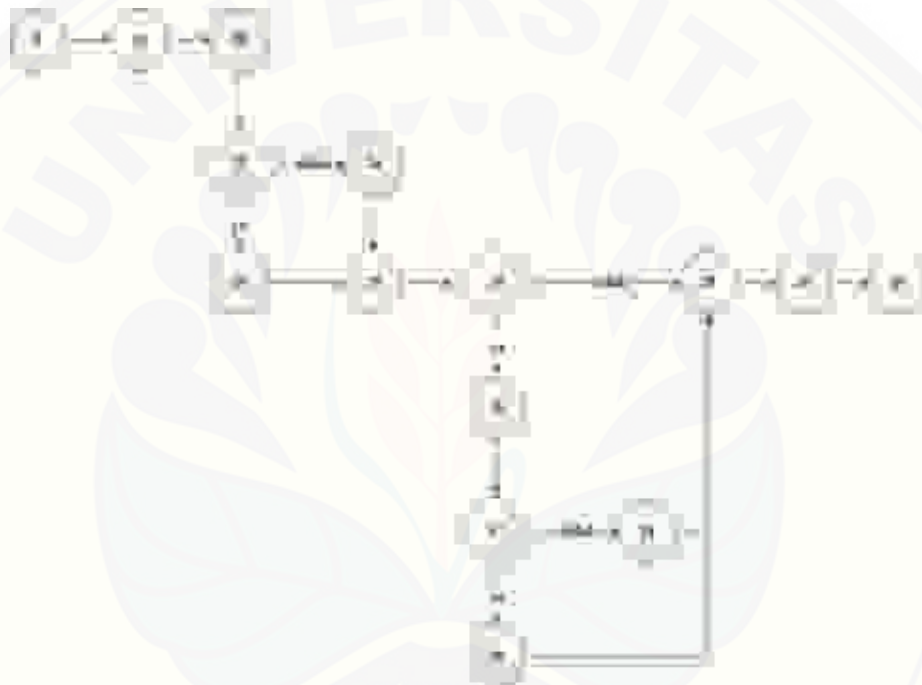
Grafik alir fitur Manajemen Data Rujukan Fungsi *Edit* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.46.



Gambar C. 46 Grafik Alir Manajemen Data Rujukan Fungsi *Edit*
Sumber: (Hasil analisis)

2.4. Grafik Alir Fitur Manajemen Data Rujukan Fungsi *Add*

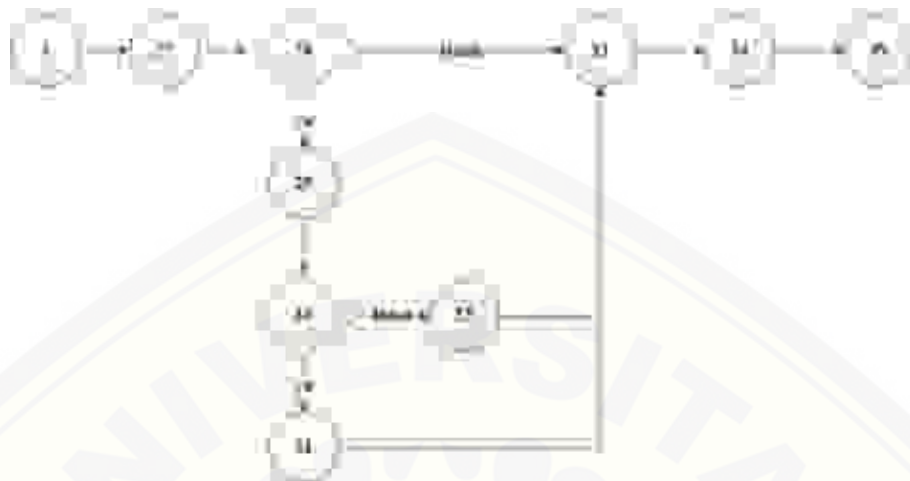
Grafik alir fitur Manajemen Data Rujukan Fungsi *Add* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.47.



Gambar C. 47 Grafik Alir Manajemen Data Rujukan Fungsi *Add*
Sumber: (Hasil analisis)

2.5. Grafik Alir Fitur Manajemen Data Rujukan Fungsi *Insert*

Grafik alir fitur Manajemen Data Rujukan Fungsi *Insert* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.48.



Gambar C. 48 Diagram Alir Manajemen Data Rujukan Fungsi *Insert*
 Sumber: (Hasil analisis)

2.6. Grafik Alir Fitur Manajemen Data Rujukan Fungsi *Update*

Grafik alir fitur Manajemen Data Rujukan Fungsi *Update* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.49.



Gambar C. 49 Diagram Alir Manajemen Data Rujukan Fungsi *Update*
 Sumber: (Hasil analisis)

2.7. Grafik Alir Fitur Manajemen Data Rujukan Fungsi *Delete*

Grafik alir fitur Manajemen Data Rujukan Fungsi *Delete* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.50.



Gambar C. 50 Grafik Alir Manajemen Data Rujukan Fungsi *Delete*
 Sumber: (Hasil analisis)

2.8. Grafik Alir Fitur Manajemen Data Rujukan Fungsi *Pdf*

Grafik alir fitur Manajemen Data Rujukan Fungsi *Delete* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.51.



Gambar C. 51 Grafik Alir Manajemen Data Rujukan Fungsi *Pdf*
 Sumber: (Hasil analisis)

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur Manajemen Data Rujukan dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2$$

Dengan $V(G)$ adalah Kompleksitas siklomatis, E adalah *Edge* atau tepi atau garis dari sebuah grafik, dan N adalah *Node* atau titik dari sebuah grafik. Maka perhitungan dari setiap grafik dari Manajemen Data Rujukan adalah sebagai berikut.

3.1. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Rujukan Fungsi *construct*

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 8 - 8 + 2 \\
 &= 2
 \end{aligned}$$

3.2. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Rujukan Fungsi *Index*

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 4 - 5 + 2 \\
 &= 1
 \end{aligned}$$

3.3. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Rujukan Fungsi *Edit*

$$V(G) = E - N + 2$$

$$= 4 - 5 + 2$$

$$= 1$$

- 3.4. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Rujukan Fungsi

Add

$$V(G) = E - N + 2$$

$$= 17 - 15 + 2$$

$$= 4$$

- 3.5. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Rujukan Fungsi

Insert

$$V(G) = E - N + 2$$

$$= 11 - 10 + 2$$

$$= 3$$

- 3.6. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Rujukan Fungsi

Update

$$V(G) = E - N + 2$$

$$= 11 - 10 + 2$$

$$= 3$$

- 3.7. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Rujukan Fungsi

Delete

$$V(G) = E - N + 2$$

$$= 4 - 5 + 2$$

$$= 1$$

- 3.8. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Rujukan Fungsi

Pdf

$$V(G) = E - N + 2$$

$$= 4 - 5 + 2$$

$$= 1$$

4. Basis Set

- 4.1. *Basis set* Fitur Manajemen Data Rujukan Fungsi *Construct*

Basis set Manajemen Data Rujukan fungsi *construct* menghasilkan dua (2) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 - 2 - 3 - 4 - 6 - 7 - 49

Jalur 2 : 1 - 2 - 3 - 4 - 5 - 7 - 49

4.2. *Basis set* Fitur Manajemen Data Rujukan Fungsi *Index*

Basis set Manajemen Data Rujukan fungsi *index* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 8 – 9 – 10 – 49

4.3. *Basis set* Fitur Manajemen Data Rujukan Fungsi *Edit*

Basis set Manajemen Data Rujukan fungsi *Edit* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 11 – 12 – 13 – 49

4.4. *Basis set* Fitur Manajemen Data Rujukan Fungsi *Add*

Basis set Manajemen Data Rujukan fungsi *Add* menghasilkan empat (4) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 14 – 21 – 22 – 23 – 25 – 15 – 16 – 17 – 18 – 20 – 26 – 49

Jalur 2 : 1 – 14 – 21 – 22 – 24 – 25 – 15 – 16 – 17 – 18 – 20 – 26 – 49

Jalur 3 : 1 – 14 – 21 – 22 – 23 – 25 – 15 – 16 – 17 – 19 – 20 – 26 – 49

Jalur 4 : 1 – 14 – 21 – 22 – 23 – 25 – 15 – 20 – 26 – 49

4.5. *Basis set* Fitur Manajemen Data Rujukan Fungsi *Insert*

Basis set Manajemen Data Rujukan fungsi *Insert* menghasilkan tiga (3) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 27 – 28 – 33 – 34 – 49

Jalur 2 : 1 – 27 – 28 – 29 – 30 – 32 – 34 – 49

Jalur 3 : 1 – 27 – 28 – 29 – 30 – 31 – 34 – 49

4.6. *Basis set* Fitur Manajemen Data Rujukan Fungsi *Update*

Basis set Manajemen Data Rujukan fungsi *Update* menghasilkan tiga (3) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 35 – 36 – 41 – 42 – 49

Jalur 2 : 1 – 35 – 36 – 37 – 38 – 40 – 41 – 42 – 49

Jalur 3 : 1 – 35 – 36 – 37 – 38 – 39 – 41 – 42 – 49

4.7. *Basis set* Fitur Manajemen Data Rujukan Fungsi *Delete*

Basis set Manajemen Data Rujukan fungsi *Delete* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 43 – 44 – 45 – 49

4.8. *Basis set* Fitur Manajemen Data Rujukan Fungsi Pdf

Basis set Manajemen Data Rujukan fungsi *Delete* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 46 – 47 – 48 – 49

5. *Test Case*

Tabel C. 90 *Test Case* Fitur Manajemen Data Rujukan fungsi *construct* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif kemudian klik menu Rujukan
Target yang diharapkan	Sistem menampilkan halaman Rujukan
Hasil pengujian	Benar
Jalur	Jalur 1
File	rujukan.php

Tabel C. 91 *Test Case* Fitur Manajemen Data Rujukan fungsi *construct* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rujukan, <i>User</i> melakukan <i>Logout</i> dan menuju halaman <i>Login</i> dan ketika di <i>back</i> tetap menampilkan halaman <i>Login</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	rujukan.php

Tabel C. 92 *Test Case* Fitur Manajemen Data Rujukan fungsi *Index* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif lalu klik menu Rujukan
Target yang diharapkan	Sistem menampilkan halaman Rujukan

Hasil pengujian	Benar
Jalur	Jalur 1
File	rujukan.php

Tabel C. 93 *Test Case* Fitur Manajemen Data Rujukan fungsi *Edit* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rujukan, <i>User</i> mengklik Tombol <i>Edit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Edit</i> Rujukan
Hasil pengujian	Benar
Jalur	Jalur 1
File	rujukan.php

Tabel C. 94 *Test Case* Fitur Manajemen Data Rujukan fungsi *Add* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rujukan, <i>User</i> mengklik Tombol Tambah Rujukan kemudian mengisi <i>form</i> Tambah Rujukan setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Rujukan serta menampilkan <i>alert, "Successfully add new data."</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	rujukan.php

Tabel C. 95 *Test Case* Fitur Manajemen Data Rujukan fungsi *Add* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rujukan, <i>User</i> mengklik Tombol Tambah Rujukan kemudian mengisi <i>form</i> Tambah Rujukan setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Rujukan serta menampilkan <i>alert, "Successfully add new data."</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	rujukan.php

Tabel C. 96 *Test Case* Fitur Manajemen Data Rujukan fungsi *Add* jalur 3

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rujukan, <i>User</i> mengklik Tombol Tambah Rujukan kemudian mengisi <i>form</i> Tambah Rujukan setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Rujukan dan menampilkan <i>alert," Failed add new data."</i>
Hasil pengujian	Benar
Jalur	Jalur 3
File	rujukan.php

Tabel C. 97 *Test Case* Fitur Manajemen Data Rujukan fungsi *Add* jalur 4

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rujukan, <i>User</i> mengklik Tombol Tambah Rujukan
Target yang diharapkan	Sistem menampilkan halaman Tambah Rujukan
Hasil pengujian	Benar
Jalur	Jalur 4
File	rujukan.php

Tabel C. 98 *Test Case* Fitur Manajemen Data Rujukan fungsi *Insert* jalur 1

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rujukan, <i>User</i> mengklik Tombol Tambah Rujukan.
Target yang diharapkan	Sistem menampilkan halaman Tambah Rujukan
Hasil pengujian	Benar
Jalur	Jalur 1
File	rujukan.php

Tabel C. 99 *Test Case* Fitur Manajemen Data Rujukan fungsi *Insert* jalur 2

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Tambah Rujukan, <i>User</i> mengisi <i>form</i> Tambah Rujukan kemudian klik tombol <i>Submit</i>
-----------	--

Target yang diharapkan	Sistem menampilkan halaman Rujukan dan menampilkan <i>alert,</i> ” <i>Failed add new data..</i> ”
Hasil pengujian	Benar
Jalur	Jalur 2
File	rujukan.php

Tabel C. 100 *Test Case* Fitur Manajemen Data Rujukan fungsi *Insert* jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Tambah Rujukan, <i>User</i> mengisi <i>form</i> Tambah Rujukan kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Rujukan dan menampilkan <i>alert,</i> ” <i>Successfully add new data..</i> ”
Hasil pengujian	Benar
Jalur	Jalur 3
File	rujukan.php

Tabel C. 101 *Test Case* Fitur Manajemen Data Rujukan fungsi *Update* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rujukan, <i>User</i> mengklik Tombol <i>Edit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Edit</i> Rujukan
Hasil pengujian	Benar
Jalur	Jalur 1
File	rujukan.php

Tabel C. 102 *Test Case* Fitur Manajemen Data Rujukan fungsi *Insert* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Rujukan, <i>User</i> mengisi <i>form edit</i> Rujukan kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Rujukan dan menampilkan <i>alert,</i> ” <i>Failed update data..</i> ”
Hasil pengujian	Benar
Jalur	Jalur 2
File	rujukan.php

Tabel C. 103 *Test Case* Fitur Manajemen Data Rujukan fungsi *Insert* jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu <i>Edit</i> Rujukan, <i>User</i> mengisi <i>form edit</i> Rujukan kemudian klik tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Rujukan dan menampilkan <i>alert, "Successfully update data.."</i>
Hasil pengujian	Benar
Jalur	Jalur 3
File	rujukan.php

Tabel C. 104 *Test Case* Fitur Manajemen Data Rujukan fungsi *Delete* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rujukan, <i>User</i> mengklik tombol <i>Remove</i>
Target yang diharapkan	Sistem menampilkan halaman Rujukan Rujukan dan menampilkan <i>alert, "Successfully delete data.."</i>
Hasil pengujian	Benar
Jalur	Jalur 1
File	rujukan.php

Tabel C. 105 *Test Case* Fitur Manajemen Data Rujukan fungsi *Pdf* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Rujukan, <i>User</i> mengklik tombol Cetak
Target yang diharapkan	Sistem menampilkan halaman <i>pdf</i> "DATA RUJUKAN"
Hasil pengujian	Benar
Jalur	Jalur 1
File	rujukan.php

C. 10 Manajemen Transaksi Pembayaran Pasien

Pengujian kode – kode program pada fitur Manajemen Transaksi Pembayaran Pasien adalah sebagai berikut:

1. Listing Program

```

1. 1class Transaksi extends CI_Controller {
2.
3.     public function __construct( 2
4.         parent::__construct();
5.         $this->load->model("transaksi_model","transaksi");
6.         if(!$this->session->userdata("rkmds-logged-in")) { 4
7.             redirect('/');5
8.         } 6
9.     } 7
10.
11.     public function index() { 8
12.         $this->load->view("transaksi-index"); 9
13.     } 10
14.
15.     public function rawatinap() { 11
16.         if($this->input->post("submit")) { 12
17.             $no_transaksi = $this->generateNomorTransaksi();
18.             $tanggal_transaksi = $this->input-
19. >post("tanggal_transaksi");
20.             $id_pasien = $this->input->post("id_pasien");
21.             $id_kamar = $this->input->post("id_kamar");
22.             $tanggal_masuk = $this->input->post("tanggal_masuk");
23.             $tanggal_keluar = $this->input->post("tanggal_keluar");
24.             $lama_inap = $this->input->post("lama_inap");
25.             $keterangan = $this->input->post("keterangan");
26.             $biaya_dokter = $this->input->post("biaya_dokter");
27.             $biaya_visit_dokter = $this->input-
28. >post("biaya_visit_dokter");
29.             $biaya_perawatan = $this->input->post("biaya_perawatan");
30.             $biaya_kamar = $this->input->post("biaya_kamar");
31.             $biaya_obat = $this->input->post("biaya_obat");
32.             $biaya_laboratorium = $this->input-
33. >post("biaya_laboratorium");
34.             $nama_biaya_fasilitas_lain = $this->input-
35. >post("nama_biaya_fasilitas_lain");
36.             $biaya_fasilitas_lain = $this->input-
37. >post("biaya_fasilitas_lain");
38.             $arr_tgl_transaksi = explode("/", $tanggal_transaksi);
39.             $arr_tgl_masuk = explode("/", $tanggal_masuk);
40.             $arr_tgl_keluar = explode("/", $tanggal_keluar);
41.
42.             $data = array(
43.                 "no_transaksi"=>$no_transaksi,
44.                 "tanggal_transaksi"=>date("Y-m-
45. d",strtotime($arr_tgl_transaksi[2]."-". $arr_tgl_transaksi[1]."-
46. ". $arr_tgl_transaksi[0])),
47.                 "id_kamar"=>$id_kamar,

```

```

41.         "id_pasien"=>$id_pasien,
42.         "tanggal_masuk"=>date("Y-m-
d",strtotime($arr_tgl_masuk[2]."-". $arr_tgl_masuk[1]."-
". $arr_tgl_masuk[0])),
43.         "tanggal_keluar"=>date("Y-m-
d",strtotime($arr_tgl_keluar[2]."-". $arr_tgl_keluar[1]."-
". $arr_tgl_keluar[0])),
44.         "lama_inap"=>$lama_inap,
45.         "keterangan"=>$keterangan,
46.         "biaya_dokter"=>$biaya_dokter,
47.         "biaya_visit_dokter"=>$biaya_visit_dokter,
48.         "biaya_perawatan"=>$biaya_perawatan,
49.         "biaya_kamar"=>$biaya_kamar,
50.         "biaya_obat"=>$biaya_obat,
51.         "biaya_laboratorium"=>$biaya_laboratorium,
52.         "nama_biaya_fasilitas_lain"=>$nama_biaya_fasilitas_lain,
53.         "biaya_fasilitas_lain"=>$biaya_fasilitas_lain,
54.     );
55.
56.     14 if($id_transaksi = $this->transaksi-
>insert_transaksiRawatInap($data)) {
57.         redirect('transaksi/pdf/rawatinap/'.$id_transaksi); 15
58.     } 16
59. } 17
60. $this->load->view('transaksi-rawatinap'); 18
61. } 19
62.
63. public function pdf() { 20
64.     $type = $this->uri->segment(3);
65.     21 $id = $this->uri->segment(4);
66.
67.     if($type=="rawatinap") { 22
68.         $data= array();
69.         23 $data['transaksi'] = $this->db
->where("id_transaksi",$id)
70.         -
71.         >select('tb_transaksi_rawatinap.*, tb_pasien.nama as nama_pasien, tb_kam
ar.nama as nama_kamar', false)
72.         -
>join('tb_pasien', 'tb_transaksi_rawatinap.id_pasien = tb_pasien.id_pasi
en')
73.         -
>join('tb_kamar', 'tb_transaksi_rawatinap.id_kamar = tb_kamar.id_kamar')
74.         ->get("tb_transaksi_rawatinap");
75.         $this->load->view('pdf/transaksi/rawatinap', $data);
76.     } 24
77.     else if($type=="laboratorium") { 25
78.         $data= array();
79.         $data['transaksi'] = $this->db
->where("id_transaksi",$id)
80.         -
81.         >select('tb_transaksi_laboratorium.*, tb_pasien.nama as nama_pasien', fa
lse)
82.     } 26

```

```

82.         -
83.     >join('tb_pasien', 'tb_transaksi_laboratorium.id_pasien = tb_pasien.id_p
      pasien')
84.         ->get("tb_transaksi_laboratorium");
85.         $data['transaction_details'] = $this->db
86.         ->where("id_transaksi",$id)
87.         ->get("tb_transaksi_laboratorium_detail");
88.     $this->load->view('pdf/transaksi/laboratorium', $data);
89. } 27
90.     else if($type=="obat") { 28
91.         $data= array();
92.         $data['transaksi'] = $this->db
93.         ->where("id_transaksi",$id)
94.         -
95.     >select('tb_transaksi_obat.*, tb_pasien.nama as nama_pasien', false)
96.     >join('tb_pasien', 'tb_transaksi_obat.id_pasien = tb_pasien.id_pasien')
97.     } 29
98.     ->get("tb_transaksi_obat");
99.     $data['transaction_details'] = $this->db
100.    ->where("id_transaksi",$id)
101.    -
102.    >select('tb_transaksi_obat_detail.*, tb_obat.nama as nama_obat, tb_obat.
      harga_jual', false)
103.    -
104.    >join('tb_obat', 'tb_transaksi_obat_detail.id_obat = tb_obat.id_obat')
105.    ->get("tb_transaksi_obat_detail");
106.    $this->load->view('pdf/transaksi/obat', $data);
107.    } 30
108. } 31
109.     public function obat() { 32
110.         if($this->input->post("submit")) { 33
111.             $no_transaksi = $this->generateNomorTransaksi();
112.             $tanggal_transaksi = $this->input-
113.             >post("tanggal_transaksi");
114.             $id_pasien = $this->input->post("id_pasien");
115.             $cart_id_obat = $this->input->post("cart_id_obat");
116.             $cart_qty = $this->input->post("cart_qty");
117.             $arr_tgl_transaksi = explode("/", $tanggal_transaksi);
118.             $data = array(
119.                 "no_transaksi"=>$no_transaksi,
120.                 "tanggal_transaksi"=>date("Y-m-
121.                 d",strtotime($arr_tgl_transaksi[2]."-".
122.                 $arr_tgl_transaksi[1]."-
123.                 ".$arr_tgl_transaksi[0])),
124.                 "id_pasien"=>$id_pasien,
125.             );
126.             35 if($id_transaksi = $this->transaksi-
127.             >insertTransaksiObat($data,$cart_id_obat,$cart_qty)) {
128.                 redirect('transaksi/pdf/obat/'.$id_transaksi); 36
129.             } 37
130.         } 38
131.     } 39
132.     $this->load->view('transaksi-obat'); 39
133. } 40
134. }
135. }

```

```

126. public function laboratorium(){
127. 41 if($this->input->post("submit")){ 42
128.     $no_transaksi = $this->generateNomorTransaksi();
129.     $tanggal_transaksi = $this->input-
>post("tanggal_transaksi");
130.     $id_pasien = $this->input->post("id_pasien");
131.     $cart_jenis_pemeriksaan = $this->input-
>post("jenis_pemeriksaan"); //array
132.     $cart_biaya_pemeriksaan = $this->input-
>post("biaya_pemeriksaan"); //array
133.     $arr_tgl_transaksi = explode("/", $tanggal_transaksi);
134.
135.     43 $data = array(
136.         "no_transaksi"=>$no_transaksi,
137.         "tanggal_transaksi"=>date("Y-m-
d",strtotime($arr_tgl_transaksi[2]."-".
$arr_tgl_transaksi[1]."-
".$arr_tgl_transaksi[0])),
138.         "id_pasien"=>$id_pasien,
139.     );
140.     44 if($id_transaksi = $this->transaksi-
>insertTransaksiLaboratorium($data,$cart_jenis_pemeriksaan,$cart_biaya_p
emeriksaan)){
141.         //pdf laborat
142.         45 redirect('transaksi/pdf/laboratorium/'.$id_transaksi)
;
143.     } 46
144.
145.     47 }
146.     $this->load->view('transaksi-laboratorium'); 48
147.     49 }
148.
149.     public function getkamar() { 50
150.         $id_kamar = $this->input->post("id_kamar");
151.         $kamar = $this->db->where("id_kamar",$id_kamar)-
>get("tb_kamar")->row();
152.         51 $kamar = array(
153.             "id_kamar"=>$kamar->id_kamar,
154.             "nama"=>$kamar->nama,
155.             "kelas"=>$kamar->kelas,
156.             "biaya"=>($kamar->biaya),
157.             "status"=>$kamar->status,
158.         );
159.         52 echo json_encode($kamar);
160.     } 53
161.
162.     public function getobat() { 54
163.         $id_obat = $this->input->post("id_obat");
164.         $obat = $this->db->where("id_obat",$id_obat)->get("tb_obat")-
>row(); 55
165.         $obat = array(
166.             "id_obat"=>$obat->id_obat,
167.             "nama"=>$obat->nama,
168.             "jenis"=>$obat->jenis,
169.             "satuan"=>$obat->satuan,
170.             "stok"=>$obat->stok,
171.             "harga_pokok"=>$obat->harga_pokok,

```

```

172.         "harga_jual"=>$obat->harga_jual,
173.     );
174.     echo json_encode($obat); 56
175. } 57
176.
177.     public function gettanggalmasuk() { 58
178.         $id_pasien = $this->input->post("id_pasien");
179.         $pasien = $this->db->where("id_pasien",$id_pasien)-
>get("tb_pasien")->row();
180.         $pasien = array(
181.             "tanggal_daftar"=>date('d/m/Y',strtotime($pasien-
>tanggal_daftar)),
182.         );
183.         echo json_encode($pasien); 60
184.     } 61
185.
186.     private function generateNomorTransaksi(){
187.         return time()."";
188.     }
189. } 62

```

2. Grafik Alir

Grafik Alir fitur Manajemen Data Transaksi Pembayaran Pasien berupa notasi-notasi yang menunjukkan alur kontrol. Fitur Manajemen Data Transaksi Pembayaran Pasien terbagi menjadi beberapa fungsi, yaitu *construct*, *index*, *rawatinap*, *pdf*, *obat*, *laboratorium*, *getkamar*, *getobat*, dan *gettanggalmasuk*.

2.1. Grafik Alir Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *Construct*

Grafik alir fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *Construct* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.52.



Gambar C. 52 Grafik Alir Manajemen Data Transaksi Pembayaran Pasien Fungsi *Construct*

Sumber: (Hasil analisis)

2.2. Grafik Alir Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *Index*

Grafik alir fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *Index* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.53.



Gambar C. 53 Grafik Alir Manajemen Data Transaksi Pembayaran Pasien jukan Fungsi *Index*

Sumber: (Hasil analisis)

2.3. Grafik Alir Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi rawatinap

Grafik alir fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi rawatinap digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.54.



Gambar C. 54 Grafik Alir Manajemen Data Transaksi Pembayaran Pasien Fungsi rawatinap

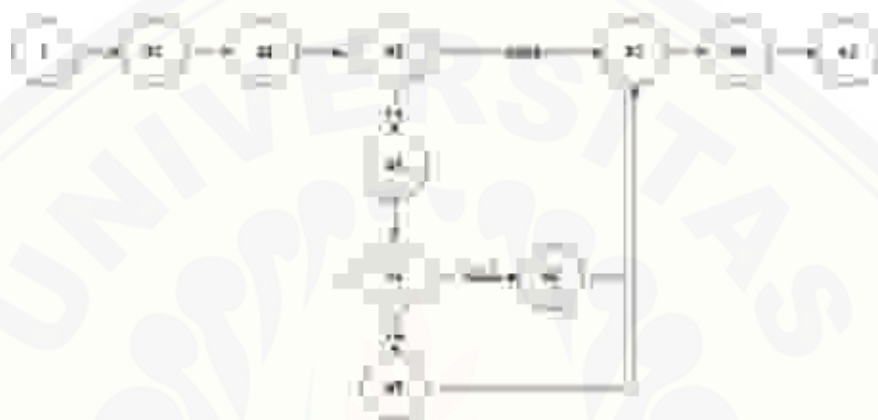
Sumber: (Hasil analisis)

2.4. Grafik Alir Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *Pdf*

Grafik alir fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *Pdf* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.55.

2.6. Grafik Alir Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi Laboratorium

Grafik alir fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi Laboratorium digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.57.



Gambar C. 57 Diagram Alir Manajemen Data Transaksi Pembayaran Pasien Fungsi Laboratorium

Sumber: (Hasil analisis)

2.7. Grafik Alir Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *getkamar*

Grafik alir fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *getkamar* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.58.



Gambar C. 58 Grafik Alir Manajemen Data Transaksi Pembayaran Pasien Fungsi *getkamar*

Sumber: (Hasil analisis)

2.8. Grafik Alir Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *getobat*

Grafik alir fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *getobat* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.59.



Gambar C. 59 Grafik Alir Manajemen Data Transaksi Pembayaran Pasien Fungsi *getobat*
 Sumber: (Hasil analisis)

2.9. Grafik Alir Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *gettanggalmasuk*

Grafik alir fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *gettanggalmasuk* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.60.



Gambar C. 60 Grafik Alir Manajemen Data Transaksi Pembayaran Pasien Fungsi *gettanggalmasuk*
 Sumber: (Hasil analisis)

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur Manajemen Data Transaksi Pembayaran Pasien dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2$$

Dengan $V(G)$ adalah Kompleksitas siklomatis, E adalah *Edge* atau tepi atau garis dari sebuah grafik, dan N adalah *Node* atau titik dari sebuah grafik. Maka perhitungan dari setiap grafik dari Manajemen Data Transaksi Pembayaran Pasien adalah sebagai berikut.

3.1. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *construct*

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 8 - 8 + 2 \\
 &= 2
 \end{aligned}$$

- 3.2. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Transaksi
Pembayaran Pasien Fungsi *Index*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

- 3.3. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Transaksi
Pembayaran Pasien Fungsi *rawatnap*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 12 - 11 + 2 \\ &= 3 \end{aligned}$$

- 3.4. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Transaksi
Pembayaran Pasien Fungsi *Pdf*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 16 - 14 + 2 \\ &= 4 \end{aligned}$$

- 3.5. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Transaksi
Pembayaran Pasien Fungsi *Obat*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 12 - 11 + 2 \\ &= 3 \end{aligned}$$

- 3.6. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Transaksi
Pembayaran Pasien Fungsi *Laboratorium*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 12 - 11 + 2 \\ &= 3 \end{aligned}$$

- 3.7. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Transaksi
Pembayaran Pasien Fungsi *getkamar*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 5 - 6 + 2 \\ &= 1 \end{aligned}$$

- 3.8. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Transaksi
Pembayaran Pasien Fungsi *getobat*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 5 - 6 + 2 \\ &= 1 \end{aligned}$$

3.9. Perhitungan Kompleksitas Siklomatis fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *gettangalmasuk*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 5 - 6 + 2 \\ &= 1 \end{aligned}$$

4. *Basis Set*

4.1. *Basis set* Fitur Manajemen Data Transaksi Pembayaran Pasien Transaksi Pembayaran Pasien Fungsi *Construct*

Basis set Manajemen Data Transaksi Pembayaran Pasien fungsi *construct* menghasilkan dua (2) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 6 – 7 – 62

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 7 – 62

4.2. *Basis set* Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *Index*

Basis set Manajemen Data Transaksi Pembayaran Pasien fungsi *index* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 8 – 9 – 10 – 62

4.3. *Basis set* Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *rawatinap*

Basis set Manajemen Data Transaksi Pembayaran Pasien fungsi *rawatinap* menghasilkan tiga (3) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 11 – 18 – 12 – 17 – 19 – 62

Jalur 2 : 1 – 11 – 18 – 12 – 13 – 14 – 16 – 17 – 19 – 62

Jalur 3 : 1 – 11 – 18 – 12 – 13 – 14 – 15 – 17 – 19 – 62

4.4. *Basis set* Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *Pdf*

Basis set Manajemen Data Transaksi Pembayaran Pasien fungsi *Pdf* menghasilkan empat (4) jalur *independent* dari perhitungan kompleksitas siklomatik,

yaitu:

Jalur 1 : 1 – 20 – 21 – 22 – 23 – 25 – 31 – 62

Jalur 2 : 1 – 20 – 21 – 22 – 24 – 25 – 26 – 31 – 62

Jalur 3 : 1 – 20 – 21 – 22 – 24 – 27 – 28 – 29 – 31 – 62

Jalur 4 : 1 – 20 – 21 – 22 – 24 – 27 – 28 – 30 – 31 – 62

4.5. *Basis set* Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi Obat

Basis set Manajemen Data Transaksi Pembayaran Pasien fungsi Obat menghasilkan tiga (3) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 32 – 39 – 33 – 17 – 40 – 62

Jalur 2 : 1 – 32 – 39 – 33 – 34 – 35 – 36 – 17 – 40 – 62

Jalur 3 : 1 – 32 – 39 – 33 – 34 – 35 – 37 – 17 – 40 – 62

4.6. *Basis set* Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi Laboratorium

Basis set Manajemen Data Transaksi Pembayaran Pasien fungsi Laboratorium menghasilkan tiga (3) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 41 – 48 – 42 – 47 – 49 – 62

Jalur 2 : 1 – 41 – 48 – 42 – 43 – 44 – 46 – 47 – 49 – 62

Jalur 3 : 1 – 41 – 48 – 42 – 43 – 44 – 45 – 47 – 49 – 62

4.7. *Basis set* Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *getkamar*

Basis set Manajemen Data Transaksi Pembayaran Pasien fungsi *getkamar* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 50 – 51 – 52 – 53 – 62

4.8. *Basis set* Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *getobat*

Basis set Manajemen Data Transaksi Pembayaran Pasien fungsi *getobat* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik,

yaitu:

Jalur 1 : 1 – 54 – 55 – 56 – 57 – 62

4.9. Basis set Fitur Manajemen Data Transaksi Pembayaran Pasien Fungsi *gettangalmasuk*

Basis set Manajemen Data Transaksi Pembayaran Pasien fungsi *gettangalmasuk* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 58 – 59 – 60 – 61 – 62

5. Test Case

Tabel C. 106 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi *construct* jalur 1
Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif kemudian klik menu Transaksi Pembayaran
Target yang diharapkan	Sistem menampilkan halaman Transaksi Pembayaran
Hasil pengujian	Benar
Jalur	Jalur 1
File	transaksi.php

Tabel C. 107 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi *construct* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> melakukan <i>Logout</i> dan menuju halaman <i>Login</i> dan ketika di <i>back</i> tetap menampilkan halaman <i>Login</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	transaksi.php

Tabel C. 108 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi *Index* jalur 1

Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif lalu klik menu Transaksi Pembayaran
Target yang diharapkan	Sistem menampilkan halaman Transaksi Pembayaran
Hasil pengujian	Benar
Jalur	Jalur 1
File	transaksi.php

Tabel C. 109 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi rawatinap jalur 1

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol Perawatan
Target yang diharapkan	Sistem menampilkan halaman Transaksi Pembayaran Perawatan
Hasil pengujian	Benar
Jalur	Jalur 1
File	transaksi.php

Tabel C. 110 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi rawatinap jalur 2

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol Perawatan kemudian mengisi <i>form</i> Transaksi Pembayaran Perawatan.
Target yang diharapkan	Sistem menampilkan halaman Transaksi Pembayaran Perawatan
Hasil pengujian	Benar
Jalur	Jalur 2
File	transaksi.php

Tabel C. 111 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi rawatinap jalur 3

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi
-----------	------------------------------------

	Pembayaran, <i>User</i> mengklik Tombol Perawatan kemudian mengisi <i>form</i> Transaksi Pembayaran Perawatan setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>pdf</i> Data Transaksi Pembayaran Perawatan Pasien
Hasil pengujian	Benar
Jalur	Jalur 3
File	transaksi.php

Tabel C. 112 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi Pdf jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol Perawatan kemudian mengisi <i>form</i> Transaksi Pembayaran Perawatan setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Pdf Data Transaksi Pembayaran Perawatan
Hasil pengujian	Benar
Jalur	Jalur 1
File	transaksi.php

Tabel C. 113 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi Pdf jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol Laboratorium kemudian mengisi <i>form</i> Transaksi Pembayaran Laboratorium setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Pdf Data Transaksi Pembayaran Laboratorium
Hasil pengujian	Benar
Jalur	Jalur 2
File	transaksi.php

Tabel C. 114 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi Pdf jalur 3

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol Obat kemudian mengisi <i>form</i> Transaksi Pembayaran Obat setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Pdf Data Transaksi Pembayaran Obat
Hasil pengujian	Benar
Jalur	Jalur 3
File	transaksi.php

Tabel C. 115 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi Pdf jalur 4

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol salah satu Tombol Transaksi Pembayaran kemudian mengisi <i>form</i> yang tersedia setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman Pdf yang sesuai dari jenis transaksi yang telah diisi sebelumnya
Hasil pengujian	Benar
Jalur	Jalur 4
File	transaksi.php

Tabel C. 116 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi Obat jalur 1

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol Obat
Target yang diharapkan	Sistem menampilkan halaman Transaksi Pembayaran Obat
Hasil pengujian	Benar
Jalur	Jalur 1
File	transaksi.php

Tabel C. 117 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi Obat jalur 2

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol Obat kemudian mengisi <i>form</i> Transaksi Pembayaran Obat.
Target yang diharapkan	Sistem menampilkan halaman Transaksi Pembayaran Obat
Hasil pengujian	Benar
Jalur	Jalur 2
File	transaksi.php

Tabel C. 118 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi Obat jalur 3

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol Obat kemudian mengisi <i>form</i> Transaksi Pembayaran Obat setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>pdf</i> Data Transaksi Pembayaran Obat
Hasil pengujian	Benar
Jalur	Jalur 3
File	transaksi.php

Tabel C. 119 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi Laboratorium jalur 1

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol Laboratorium
Target yang diharapkan	Sistem menampilkan halaman Transaksi Pembayaran Laboratorium
Hasil pengujian	Benar
Jalur	Jalur 1
File	transaksi.php

Tabel C. 120 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi Laboratorium jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol Laboratorium kemudian mengisi <i>form</i> Transaksi Pembayaran Laboratorium.
Target yang diharapkan	Sistem menampilkan halaman Transaksi Pembayaran Laboratorium
Hasil pengujian	Benar
Jalur	Jalur 2
File	transaksi.php

Tabel C. 121 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi Laboratorium jalur 3
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol Laboratorium kemudian mengisi <i>form</i> Transaksi Pembayaran Laboratorium setelah itu klik Tombol <i>Submit</i>
Target yang diharapkan	Sistem menampilkan halaman <i>pdf</i> Data Transaksi Pembayaran Laboratorium
Hasil pengujian	Benar
Jalur	Jalur 3
File	transaksi.php

Tabel C. 122 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi *getkamar* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol Perawatan kemudian memilih dropdown kamar
Target yang diharapkan	Sistem menampilkan halaman Transaksi Pembayaran Perawatan
Hasil pengujian	Benar
Jalur	Jalur 1
File	transaksi.php

Tabel C. 123 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi *getkamar* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol Perawatan kemudian memilih kamar dengan cara klik dropdown kamar kemudian klik salah satu pilihan kamar.
Target yang diharapkan	Sistem menampilkan halaman Transaksi Pembayaran Perawatan
Hasil pengujian	Benar
Jalur	Jalur 1
File	transaksi.php

Tabel C. 124 *Test Case* Fitur Manajemen Data Transaksi Pembayaran Pasien fungsi *getobat* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Transaksi Pembayaran, <i>User</i> mengklik Tombol Obat kemudian memilih dengan cara klik dropdown Obat kemudian klik salah satu pilihan obat
Target yang diharapkan	Sistem menampilkan halaman Transaksi Pembayaran Obat
Hasil pengujian	Benar
Jalur	Jalur 1
File	transaksi.php

C.11 Melihat Laporan

Pengujian kode-kode program pada fitur Melihat Laporan adalah sebagai berikut:

1. Listing Program

```

1 class Laporan extends CI_Controller {
2.
3.     public function __construct() { 2
4.         3 parent::__construct();
5.         // $this->authorization->isAdminLogin();
6.         if (!$this->session->userdata("rkmds-logged-in")) { 4
7.             redirect('/', ); 5
8.         } 6
9.     } 7

```

```

10.
11.     public function index() { 8
12.         $this->load->view("laporan-index"); 9
13.     } 10
14.
15.     public function pasien() { 11
16.         $this->load->model("pasien_model", "pasien");
17.         $data['pasiens'] = $this->pasien->getDataPasien(); 12
18.         $this->load->view("laporan-pasien", $data);
19.     } 13
20.
21.     public function kamar() { 14
22.         $this->load->model("kamar_model", "kamar");
23.         $data['kamars'] = $this->kamar->getDataKamar();
24.         $this->load->view("laporan-kamar", $data);
25.     } 15
26.
27.     public function dokter() { 17
28.         $this->load->model("dokter_model", "dokter");
29.         $data['dokters'] = $this->dokter->getDataDokter(); 18
30.         $this->load->view("laporan-dokter", $data);
31.     } 19
32.
33.     public function pegawai() { 20
34.         $this->load->model("pegawai_model", "pegawai");
35.         $data['pegawais'] = $this->pegawai->getDataPegawai(); 21
36.         $this->load->view("laporan-pegawai", $data);
37.     } 22
38.
39.     public function fasilitas() { 23
40.         $this->load->model("fasilitas_model", "fasilitas");
41.         $data['fasilitass'] = $this->fasilitas->
42.             >getDataFasilitas(); 24
43.         $this->load->view("laporan-fasilitas", $data);
44.     } 25
45.
46.     public function c() { 26
47.         $this->load->model("obat_model", "obat");
48.         $data['obats'] = $this->obat->getDataObat(); 27
49.         $this->load->view("laporan-obat", $data);
50.     } 28
51.
52.     public function laboratorium() { 29
53.         $this->load->model("laborat_model", "laborat");
54.         $data['laborats'] = $this->laborat->getDataLaborat(); 30
55.         $this->load->view("laporan-laborat", $data);
56.     } 31
57.
58.     public function rekammedis() { 32
59.         $this->load->model("rekammedis_model", "rekammedis");
60.         $data['rekammediss'] = $this->rekammedis->getDataRekammedis();
61.         $this->load->view("laporan-rekammedis", $data);
62.     } 33
63.
64.     public function rujukan() { 35
65.         $this->load->model("rujukan_model", "rujukan");
66.     } 34
67.
68.     public function rujukan() { 35
69.         $this->load->model("rujukan_model", "rujukan");
70.     } 36

```

```

65.         $data['rujukans'] = $this->rujukan->getDataRujukan();
66.         $this->load->view("laporan-rujukan",$data);
67.     } 37
68.
69.     private function generateNomorTransaksi(){
70.         return time()."";
71.     }
72.
73.     38 function pdf($table){
74.         //$this->load->helper('pdf_helper');
75.         //$this->load->library('mypdf');
76.
77.         39 $_table = $this->getTable()[$table];
78.         $data = array();
79.         if($_table=="tb_pasien"){ 40
80.             $sql = "select a.*,b.nama as nama_dokter from tb_pasien a
81.             41 left join tb_dokter b on a.id_dokter = b.id_dokter";
82.             $data['data'] = $this->db->query($sql);
83.         }
84.         else{
85.             42 $data['data'] = $this->db->get($_table);
86.         }
87.         //$this->mypdf->subtitle = $_table;
88.
89.         //var_dump($data);exit;
90.         $this->load->view('pdf/laporan/'.$table, $data); 43
91.     } 44
92.
93.     private function getTable(){ 45
94.         return array(
95.             "pasien" => "tb_pasien",
96.             46 "kamar" => "tb_kamar",
97.             "dokter" => "tb_dokter",
98.             "pegawai" => "tb_pegawai",
99.             "fasilitas" => "tb_fasilitas",
100.            "obat" => "tb_obat",
101.            "laboratorium" => "tb_laborat",
102.            "rekammedis" => "tb_rekammedis",
103.            "rujukan" => "tb_rujukan",
104.            "laborat" => array("tb_laborat"),
105.        );
106.    } 47
107. } 48

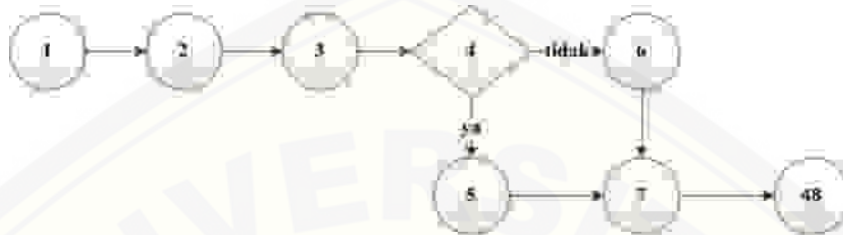
```

2. Grafik Alir

Grafik Alir fitur Melihat Laporan berupa notasi-notasi yang menunjukkan alur kontrol. Fitur Melihat Laporan terbagi menjadi beberapa fungsi, yaitu *construct*, *index*, pasien, kamar, dokter, pegawai, fasilitas, obat, laboratorium, rekammedis, rujukan, *pdf*, dan *getTable*.

2.1. Grafik Alir Fitur Melihat Laporan Fungsi *Construct*

Grafik alir fitur Melihat Laporan Fungsi *Construct* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.61.



Gambar C. 61 Grafik Alir Melihat Laporan Fungsi *Construct*
 Sumber: (Hasil analisis)

2.2. Grafik Alir Fitur Melihat Laporan Fungsi *Index*

Grafik alir fitur Melihat Laporan Fungsi *Index* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.62.



Gambar C. 62 Grafik Alir Melihat Laporan jukan Fungsi *Index*
 Sumber: (Hasil analisis)

2.3. Grafik Alir Fitur Melihat Laporan Fungsi pasien

Grafik alir fitur Melihat Laporan Fungsi pasien digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.63.



Gambar C. 63 Grafik Alir Melihat Laporan Fungsi pasien
 Sumber: (Hasil analisis)

2.4. Grafik Alir Fitur Melihat Laporan Fungsi kamar

Grafik alir fitur Melihat Laporan Fungsi kamar digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.64.



Gambar C. 64 Grafik Alir Melihat Laporan Fungsi kamar
Sumber: (Hasil analisis)

2.5. Grafik Alir Fitur Melihat Laporan Fungsi dokter

Grafik alir fitur Melihat Laporan Fungsi dokter digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.65.



Gambar C. 65 Diagram Alir Melihat Laporan Fungsi dokter
Sumber: (Hasil analisis)

2.6. Grafik Alir Fitur Melihat Laporan Fungsi pegawai

Grafik alir fitur Melihat Laporan Fungsi pegawai digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.66.



Gambar C. 66 Diagram Alir Melihat Laporan Fungsi pegawai
Sumber: (Hasil analisis)

2.7. Grafik Alir Fitur Melihat Laporan Fungsi fasilitas

Grafik alir fitur Melihat Laporan Fungsi fasilitas digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.67.



Gambar C. 67 Grafik Alir Melihat Laporan Fungsi fasilitas
Sumber: (Hasil analisis)

2.8. Grafik Alir Fitur Melihat Laporan Fungsi obat

Grafik alir fitur Melihat Laporan Fungsi obat digambarkan dengan simpul dan

garis dapat di lihat pada Gambar C.68.



Gambar C. 68 Grafik Alir Melihat Laporan Fungsi obat
Sumber: (Hasil analisis)

2.9. Grafik Alir Fitur Melihat Laporan Fungsi laboratorium

Grafik alir fitur Melihat Laporan Fungsi laboratorium digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.69.



Gambar C. 69 Grafik Alir Melihat Laporan Fungsi laboratorium
Sumber: (Hasil analisis)

2.10. Grafik Alir Fitur Melihat Laporan Fungsi rekam medis

Grafik alir fitur Melihat Laporan Fungsi rekam medis digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.70.



Gambar C. 70 Grafik Alir Melihat Laporan Fungsi rekam medis
Sumber: (Hasil analisis)

2.11. Grafik Alir Fitur Melihat Laporan Fungsi rujukan

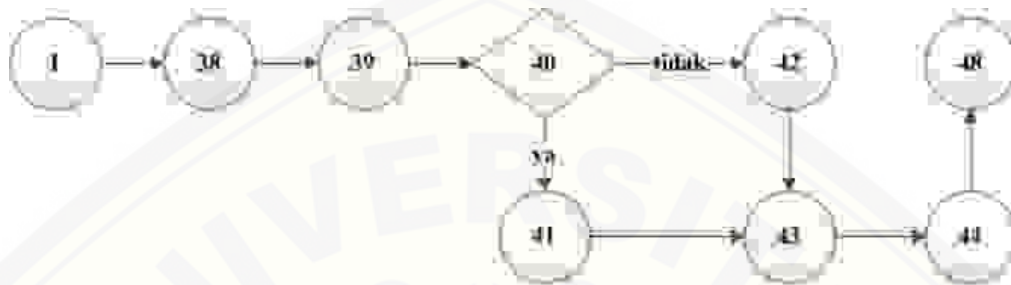
Grafik alir fitur Melihat Laporan Fungsi rujukan digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.71.



Gambar C. 71 Grafik Alir Melihat Laporan Fungsi rujukan
Sumber: (Hasil analisis)

2.12. Grafik Alir Fitur Melihat Laporan Fungsi *pdf*

Grafik alir fitur Melihat Laporan Fungsi *pdf* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.72.



Gambar C. 72 Grafik Alir Melihat Laporan Fungsi *pdf*
 Sumber: (Hasil analisis)

2.13. Grafik Alir Fitur Melihat Laporan Fungsi *getTable*

Grafik alir fitur Melihat Laporan Fungsi *getTable* digambarkan dengan simpul dan garis dapat di lihat pada Gambar C.73.



Gambar C. 73 Grafik Alir Melihat Laporan Fungsi *getTable*
 Sumber: (Hasil analisis)

3. Kompleksitas siklomatis

Kompleksitas siklomatis dari fitur Melihat Laporan dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2$$

Dengan $V(G)$ adalah Kompleksitas siklomatis, E adalah *Edge* atau tepi atau garis dari sebuah grafik, dan N adalah *Node* atau titik dari sebuah grafik. Maka perhitungan dari setiap grafik dari Fitur Melihat Laporan adalah sebagai berikut.

3.1. Perhitungan Kompleksitas Siklomatis fitur Melihat Laporan Fungsi *construct*

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 8 - 8 + 2 \\
 &= 2
 \end{aligned}$$

3.2. Perhitungan Kompleksitas Siklomatis fitur Melihat Laporan Fungsi *Index*

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1\end{aligned}$$

3.3. Perhitungan Kompleksitas Siklomatis fitur Melihat Laporan Fungsi pasien

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1\end{aligned}$$

3.4. Perhitungan Kompleksitas Siklomatis fitur Melihat Laporan Fungsi kamar

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1\end{aligned}$$

3.5. Perhitungan Kompleksitas Siklomatis fitur Melihat Laporan Fungsi dokter

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1\end{aligned}$$

3.6. Perhitungan Kompleksitas Siklomatis fitur Melihat Laporan Fungsi pegawai

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1\end{aligned}$$

3.7. Perhitungan Kompleksitas Siklomatis fitur Melihat Laporan Fungsi fasilitas

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1\end{aligned}$$

3.8. Perhitungan Kompleksitas Siklomatis fitur Melihat Laporan Fungsi obat

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1\end{aligned}$$

3.9. Perhitungan Kompleksitas Siklomatis fitur Melihat Laporan Fungsi laboratorium

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1\end{aligned}$$

3.10. Perhitungan Kompleksitas Siklomatis fitur Melihat Laporan Fungsi rekammedis

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1\end{aligned}$$

3.11. Perhitungan Kompleksitas Siklomatis fitur Melihat Laporan Fungsi rujukan

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

3.12. Perhitungan Kompleksitas Siklomatis fitur Melihat Laporan Fungsi *pdf*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 9 - 9 + 2 \\ &= 2 \end{aligned}$$

3.13. Perhitungan Kompleksitas Siklomatis fitur Melihat Laporan Fungsi *getTable*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 5 + 2 \\ &= 1 \end{aligned}$$

4. *Basis Set*4.1. *Basis set* Fitur Melihat Laporan Fungsi *Construct*

Basis set Melihat Laporan fungsi *construct* menghasilkan dua (2) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 6 – 7 – 48

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 7 – 48

4.2. *Basis set* Fitur Melihat Laporan Fungsi *Index*

Basis set Melihat Laporan fungsi *Index* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 8 – 9 – 10 – 48

4.3. *Basis set* Fitur Melihat Laporan Fungsi pasien

Basis set Melihat Laporan fungsi pasien menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 11 – 12 – 13 – 48

4.4. *Basis set* Fitur Melihat Laporan Fungsi kamar

Basis set Melihat Laporan fungsi kamar menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 14 – 15 – 16 – 48

4.5. *Basis set* Fitur Melihat Laporan Fungsi dokter

Basis set Melihat Laporan fungsi dokter menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 17 – 18 – 19 – 48

4.6. *Basis set* Fitur Melihat Laporan Fungsi pegawai

Basis set Melihat Laporan fungsi pegawai menghasilkan tiga (3) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 20 – 21 – 22 – 48

4.7. *Basis set* Fitur Melihat Laporan Fungsi fasilitas

Basis set Melihat Laporan fungsi fasilitas menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 23 – 24 – 25 – 48

4.8. *Basis set* Fitur Melihat Laporan Fungsi obat

Basis set Melihat Laporan fungsi obat menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 26 – 27 – 28 – 48

4.9. *Basis set* Fitur Melihat Laporan Fungsi laboratorium

Basis set Melihat Laporan fungsi laboratorium menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 29 – 30 – 31 – 48

4.10. *Basis set* Fitur Melihat Laporan Fungsi rekammedis

Basis set Melihat Laporan fungsi rekammedis menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 32 – 33 – 34 – 48

4.11. *Basis set* Fitur Melihat Laporan Fungsi rujukan

Basis set Melihat Laporan fungsi rujukan menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 35 – 36 – 37 – 48

4.12. *Basis set* Fitur Melihat Laporan Fungsi *pdf*

Basis set Melihat Laporan fungsi *pdf* menghasilkan dua (2) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 38 – 39 – 40 – 41 – 43 – 44 – 48

Jalur 1 : 1 – 38 – 39 – 40 – 42 – 43 – 44 – 48

4.13. *Basis set* Fitur Melihat Laporan Fungsi *getTable*

Basis set Melihat Laporan fungsi *getTable* menghasilkan satu (1) jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 : 1 – 45 – 46 – 47 – 48

5. *Test Case*

Tabel C. 125 *Test Case* Fitur Melihat Laporan fungsi *construct* jalur 1

Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif kemudian klik menu Laporan
Target yang diharapkan	Sistem menampilkan halaman Laporan
Hasil pengujian	Benar
Jalur	Jalur 1
File	laporan.php

Tabel C. 126 *Test Case* Fitur Melihat Laporan fungsi *construct* jalur 2

Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laporan, <i>User</i> melakukan <i>Logout</i> dan menuju halaman <i>Login</i> dan ketika di <i>back</i> tetap menampilkan halaman <i>Login</i>
Target yang diharapkan	Sistem menampilkan halaman <i>Login</i>
Hasil pengujian	Benar
Jalur	Jalur 2
File	laporan.php

Tabel C. 127 *Test Case* Fitur Melihat Laporan fungsi *Index* jalur 1

Sumber: (Hasil analisis)

Test case	<i>User</i> berhasil melakukan <i>Login</i> dan
-----------	---

	menuju halaman <i>Dashboard</i> atau Data <i>User</i> aktif lalu klik menu Laporan
Target yang diharapkan	Sistem menampilkan halaman Laporan
Hasil pengujian	Benar
Jalur	Jalur 1
File	laporan.php

Tabel C. 128 *Test Case* Fitur Melihat Laporan fungsi Pasien jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laporan, <i>User</i> mengklik Tombol Pasien
Target yang diharapkan	Sistem menampilkan halaman Laporan Pasien
Hasil pengujian	Benar
Jalur	Jalur 1
File	laporan.php

Tabel C. 129 *Test Case* Fitur Melihat Laporan fungsi kamar jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laporan, <i>User</i> mengklik Tombol Kamar
Target yang diharapkan	Sistem menampilkan halaman Laporan Kamar
Hasil pengujian	Benar
Jalur	Jalur 1
File	laporan.php

Tabel C. 130 *Test Case* Fitur Melihat Laporan fungsi dokter jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laporan, <i>User</i> mengklik Tombol Dokter
Target yang diharapkan	Sistem menampilkan halaman Laporan Dokter
Hasil pengujian	Benar
Jalur	Jalur 1
File	laporan.php

Tabel C. 131 *Test Case* Fitur Melihat Laporan fungsi pegawai jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laporan, <i>User</i> mengklik Tombol Pegawai
-----------	---

Target yang diharapkan	Sistem menampilkan halaman Laporan Pegawai
Hasil pengujian	Benar
Jalur	Jalur 1
File	laporan.php

Tabel C. 132 *Test Case* Fitur Melihat Laporan fungsi fasilitas jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laporan, <i>User</i> mengklik Tombol Fasilitas
Target yang diharapkan	Sistem menampilkan halaman Laporan Fasilitas
Hasil pengujian	Benar
Jalur	Jalur 1
File	laporan.php

Tabel C. 133 *Test Case* Fitur Melihat Laporan fungsi obat jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laporan, <i>User</i> mengklik Tombol Obat
Target yang diharapkan	Sistem menampilkan halaman Laporan Obat
Hasil pengujian	Benar
Jalur	Jalur 1
File	laporan.php

Tabel C. 134 *Test Case* Fitur Melihat Laporan fungsi laboratorium jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laporan, <i>User</i> mengklik Tombol Laboratorium
Target yang diharapkan	Sistem menampilkan halaman Laporan Laboratorium
Hasil pengujian	Benar
Jalur	Jalur 1
File	laporan.php

Tabel C. 135 *Test Case* Fitur Melihat Laporan fungsi rekam medis jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laporan, <i>User</i> mengklik Tombol Rekam Medis
-----------	---

Target yang diharapkan	Sistem menampilkan halaman Laporan Rekam Medis
Hasil pengujian	Benar
Jalur	Jalur 1
File	laporan.php

Tabel C. 136 *Test Case* Fitur Melihat Laporan fungsi rujukan jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laporan, <i>User</i> mengklik Tombol Rujukan
Target yang diharapkan	Sistem menampilkan halaman Laporan Rujukan
Hasil pengujian	Benar
Jalur	Jalur 1
File	laporan.php

Tabel C. 137 *Test Case* Fitur Melihat Laporan fungsi *pdf* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laporan, <i>User</i> mengklik Tombol Pasien lalu klik Tombol Cetak
Target yang diharapkan	Sistem menampilkan halaman <i>pdf</i> Laporan Pasien
Hasil pengujian	Benar
Jalur	Jalur 1
File	laporan.php

Tabel C. 138 *Test Case* Fitur Melihat Laporan fungsi *pdf* jalur 2
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laporan, <i>User</i> mengklik salah satu Tombol Laporan lalu klik Tombol Cetak
Target yang diharapkan	Sistem menampilkan halaman <i>pdf</i> Laporan sesuai dengan Laporan yang di klik sebelumnya
Hasil pengujian	Benar
Jalur	Jalur 2
File	laporan.php

Tabel C. 139 *Test Case* Fitur Melihat Laporan fungsi *getTable* jalur 1
Sumber: (Hasil analisis)

Test case	Pada saat berada di menu Laporan, <i>User</i> mengklik salah satu Tombol Laporan lalu klik Tombol Cetak
Target yang diharapkan	Sistem menampilkan halaman <i>pdf</i> Laporan sesuai dengan Laporan ang di klik sebelumnya
Hasil pengujian	Benar
Jalur	Jalur 1
File	laporan.php

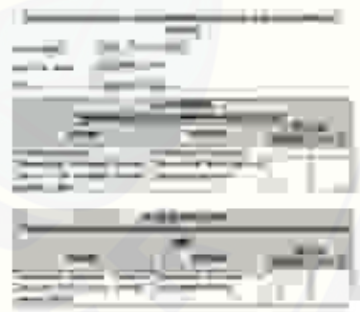
LAMPIRAN D. BLACKBOX TESTING

D. 1 AUTENTIFIKASI



Pengujian Fitur Autentifikasi(Login) Role = admin

Pengujian Fitur Autentifikasi(Login) Role = manajer



Pengujian Fitur Autentifikasi(Login) Role = apoteker

Pengujian Fitur Autentifikasi(Login) Role = laborat

D. 2 MANAJEMEN DATA *USER*



Pengujian Fitur Manajemen Data *User Role* = admin

Pengujian Fitur Manajemen Data *User Role* = admin

D. 3 MANAJEMEN DATA PEGAWAI



Pengujian Fitur Manajemen Data Pegawai *Role* = admin

Pengujian Fitur Manajemen Data Pegawai *Role* = admin



Pengujian Fitur Manajemen Data Pegawai *Role* = admin

D. 4 MANAJEMEN DATA PASIEN



Pengujian Fitur Manajemen Data Pasien *Role* = admin



Pengujian Fitur Manajemen Data Pasien *Role* = admin



Pengujian Fitur Manajemen Data Pasien *Role* = admin

D. 5 MANAJEMEN DATA KAMAR



Pengujian Fitur Manajemen Data Kamar *Role* = admin

Pengujian Fitur Manajemen Data Kamar *Role* = admin



Pengujian Fitur Manajemen Data Kamar *Role* = admin

D. 6 MANAJEMEN DATA FASILITAS



Pengujian Fitur Manajemen Data Fasilitas *Role* = admin

Pengujian Fitur Manajemen Data Fasilitas *Role* = admin



Pengujian Fitur Manajemen Data Fasilitas *Role* = admin

D. 7 MANAJEMEN DATA DOKTER



Pengujian Fitur Manajemen Data Dokter *Role* = admin



Pengujian Fitur Manajemen Data Dokter *Role* = admin



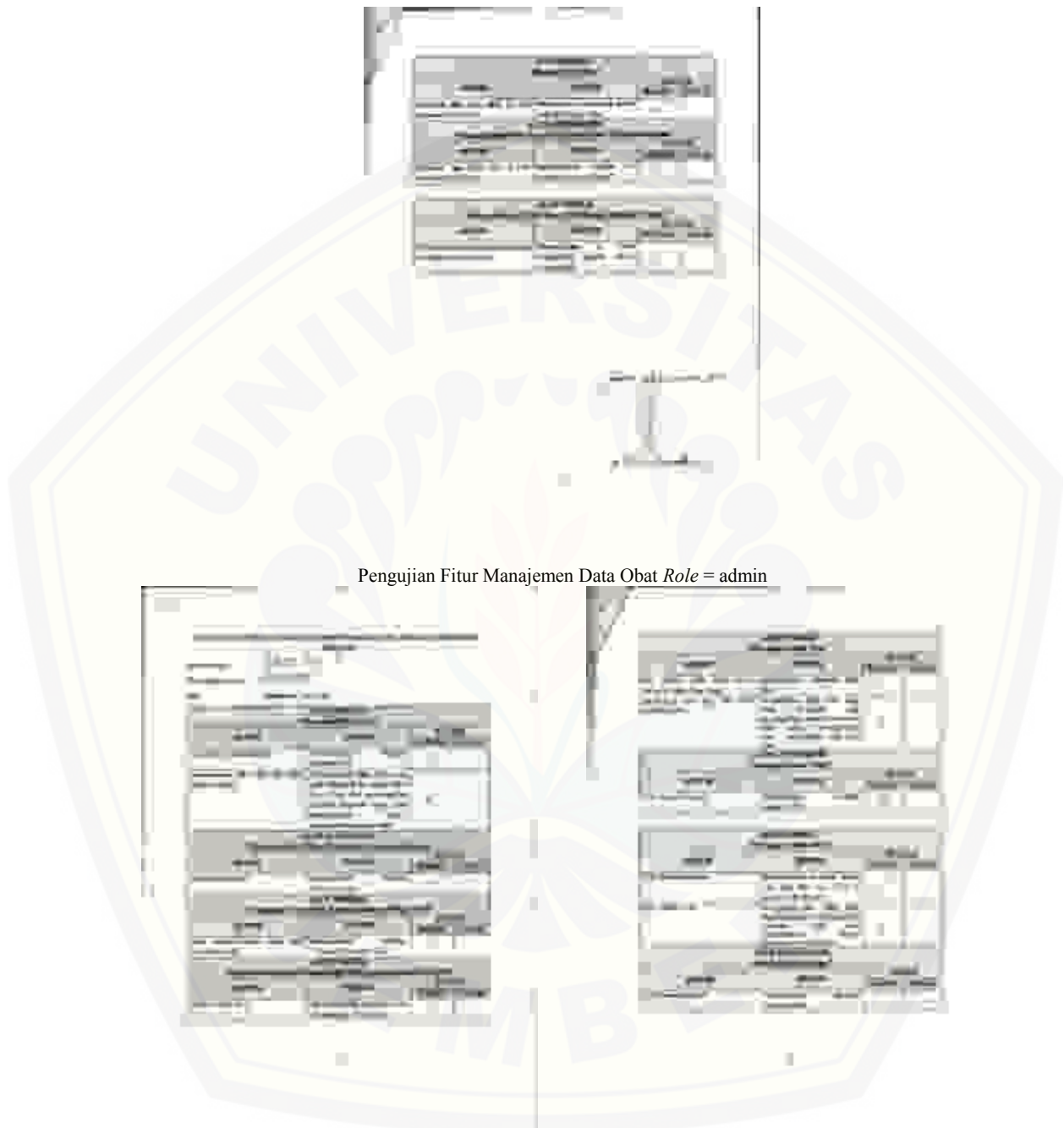
Pengujian Fitur Manajemen Data Dokter *Role* = admin

D. 8 MANAJEMEN DATA OBAT



Pengujian Fitur Manajemen Data Obat *Role* = admin

Pengujian Fitur Manajemen Data Obat *Role* = admin



Pengujian Fitur Manajemen Data Obat *Role* = admin

Pengujian Fitur Manajemen Data Obat *Role* = apoteker

Pengujian Fitur Manajemen Data Obat *Role* = apoteker



Pengujian Fitur Manajemen Data Obat *Role* = apoteker

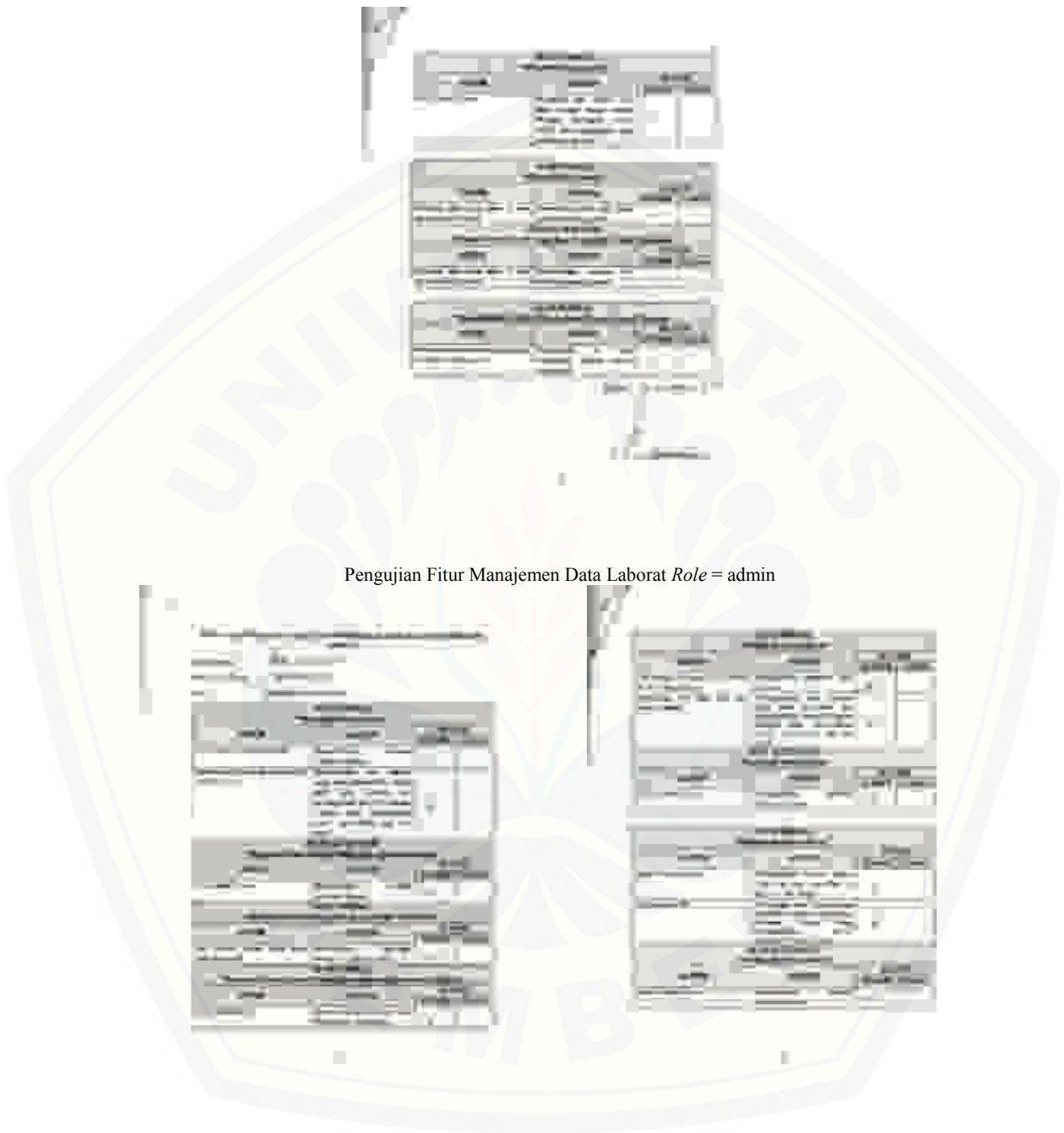
D. 9 MANAJEMEN DATA LABORAT



Pengujian Fitur Manajemen Data Laborat *Role* = admin



Pengujian Fitur Manajemen Data Laborat *Role* = admin



Pengujian Fitur Manajemen Data Laborat *Role* = admin

Pengujian Fitur Manajemen Data Laborat *Role* =
laborat

Pengujian Fitur Manajemen Data Laborat *Role* =
laborat



Pengujian Fitur Manajemen Data Laborat *Role* = laborat

D. 10 MANAJEMEN DATA REKAM MEDIS



Pengujian Fitur Manajemen Data Rekam Medis *Role* = admin



Pengujian Fitur Manajemen Data Rekam Medis *Role* = admin



Pengujian Fitur Manajemen Data Rekam Medis *Role* = admin

D. 11 MANAJEMEN DATA RUJUKAN



Pengujian Fitur Manajemen Data Rujukan *Role* = admin

Pengujian Fitur Manajemen Data Rujukan *Role* = admin



Pengujian Fitur Manajemen Data Rujukan *Role* = admin

D. 12 MANAJEMEN TRANSAKSI PEMBAYARAN PASIEN



Pengujian Fitur Manajemen Transaksi Pembayaran Pasien *Role* = admin



Pengujian Fitur Manajemen Transaksi Pembayaran Pasien *Role* = manajer



Pengujian Fitur Manajemen Transaksi Pembayaran Pasien *Role* = apoteker



Pengujian Fitur Manajemen Transaksi Pembayaran Pasien *Role* = laborat

D. 13 MELIHAT LAPORAN



Pengujian Fitur Melihat Laporan *Role* = admin



Pengujian Fitur Melihat Laporan *Role* = admin



Pengujian Fitur Melihat Laporan *Role* = admin



Pengujian Fitur Melihat Laporan *Role* = admin



Pengujian Fitur Melihat Laporan *Role* = admin



Pengujian Fitur Melihat Laporan *Role* = admin



Pengujian Fitur Melihat Laporan *Role* = admin



Pengujian Fitur Melihat Laporan *Role* = admin



Pengujian Fitur Melihat Laporan *Role* = admin



Pengujian Fitur Melihat Laporan *Role* = admin



Pengujian Fitur Melihat Laporan *Role* = manajer

Pengujian Fitur Melihat Laporan *Role* = manajer



Pengujian Fitur Melihat Laporan *Role* = manajer

Pengujian Fitur Melihat Laporan *Role* = manajer



Pengujian Fitur Melihat Laporan *Role* = manajer



Pengujian Fitur Melihat Laporan *Role* = manajer



Pengujian Fitur Melihat Laporan *Role* = manajer



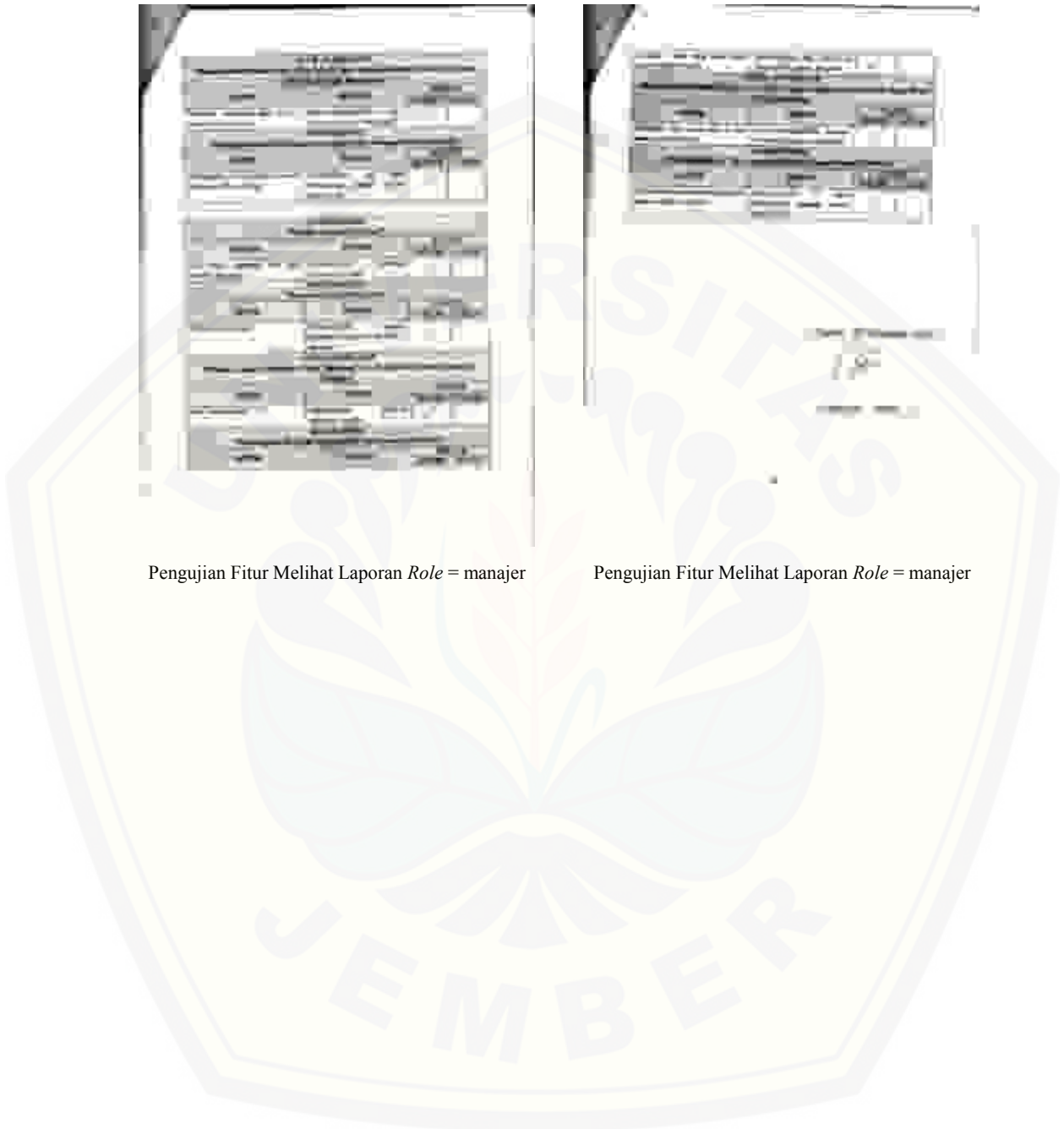
Pengujian Fitur Melihat Laporan *Role* = manajer



Pengujian Fitur Melihat Laporan *Role* = manajer



Pengujian Fitur Melihat Laporan *Role* = manajer



LAMPIRAN E. IMPLEMENTASI SISTEM

E. 1 HALAMAN TRANSAKSI PEMBAYARAN



Halaman awal transaksi pembayaran



Halaman transaksi pembayaran perawatan

Halaman transaksi pembayaran laboratorium

Halaman transaksi pembayaran obat

E. 2 HALAMAN LAPORAN



Halaman awal halaman laporan



Halaman Laporan Pasien



Halaman Laporan Kamar



Halaman Laporan Dokter



Halaman Laporan Pegawai



Halaman Laporan Fasilitas



Halaman Laporan Obat



Halaman Laporan Laborat

Halaman Laporan Rekam Medis



Halaman Laporan Rujukan

