



**SISTEM INFORMASI KLINIK MATA DENGAN PREDIKSI WAKTU
PEMERIKSAAN MENGGUNAKAN ALGORITMA C4.5 PADA
FITUR REGISTRASI PASIEN**

SKRIPSI

Oleh

Zakiyah Qurrotul Aini

NIM 102410101063

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER
2015**



**SISTEM INFORMASI KLINIK MATA DENGAN PREDIKSI WAKTU
PEMERIKSAAN MENGGUNAKAN ALGORITMA C4.5 PADA
FITUR REGISTRASI PASIEN**

SKRIPSI

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

oleh

Zakiyah Qurrotul Aini

NIM 102410101063

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER
2015**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT;
2. Yang tercinta Ibunda Afri Asiatin dan ayahanda Jamaluddin yang senantiasa memberi kasih sayang, do'a, dan dukungan;
3. Kakak serta adik-adikku tersayang yang selalu memberi semangat dan motivasi;
4. Dosen Pembimbing; Bapak Anang Andrianto, ST., MT dan Bapak M. Arief Hidayat, S. Kom. M. Kom yang selalu memberikan bimbingan dalam mengerjakan hingga menyelesaikan skripsi ini;
5. Fikra Taris yang selalu memberikan support dan semangat kepada penulis.
6. Teman – teman seperjuangan Fitriyana D, Eka Adji S, Ragilliyandi Erick, Yusa Dwi, Desi Wulandari, Risha Prasetyo, Angga Ari W, Iwan Bagus S, Nayirotul F, Syafiq, Bitu Diflia, Ratih P, Rizky Arif, Gayatri Dwi S, Hawwin Mardhiana, Rahmadinata, Margaretha, Ruroh, Anggi Septiana, Ayu P, Nisa, Mala, Sayid, Apyu Nila, dan Anggota Zerone yang lainnya.
7. Keluarga besar Program Studi Sistem Informasi Universitas Jember.

MOTTO

اللَّهُ سَبِيلِ فِي فَهُوَ الْعِلْمِ طَلَبِ فِي جَ حَرَ مَنْ

“Barang siapa keluar untuk mencari ilmu maka dia berada di jalan Allah “

(HR.Turmudzi)

“Man Jadda Wa Jadda”

“Barang siapa yang bersungguh - sungguh akan mendapatkannya.

“Innallaha yuhibbul mutawakkiliin”

“Sesungguhnya Allah mencintai orang-orang yang bertawakkal.”

(QS. 3: 159)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : ZAKIYAH QURROTUL AINI

NIM : 102410101063

menyatakan dengan sesungguhnya bahwa karya tulis ilmiah dengan judul “Sistem Informasi Klinik Mata Dengan Prediksi Waktu Pemeriksaan Menggunakan Algoritma C4.5 Pada Fitur Registrasi Pasien” adalah benar - benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan 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 sebenar - benarnya, tanpa ada tekanan dan paksaan dari pihak manapun, serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 12 Mei 2015

Yang menyatakan,

Zakiyah Qurrotul Aini

NIM.102410101063

PENGESAHAN PEMBIMBING

Skripsi berjudul “**Sistem Informasi Klinik Mata Dengan Prediksi Waktu Pemeriksaan Menggunakan Algoritma C4.5 Pada Fitur Registrasi Pasien**” telah diuji dan disahkan pada:

Hari, tanggal : Selasa, 12 Mei 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Anang Andrianto, ST., MT
NIP. 196906151997021002

M.Arief Hidayat, S. Kom. M. Kom
NIP. 198101232010121003

SKRIPSI

**SISTEM INFORMASI KLINIK MATA DENGAN PREDIKSI WAKTU
PEMERIKSAAN MENGGUNAKAN ALGORITMA C4.5 PADA
FITUR REGISTRASI PASIEN**

Oleh

**Zakiah Qurrotul Aini
NIM 102410101063**

Pembimbing

Pembimbing Utama : Anang Andrianto, ST., MT

Pembimbing Pendamping : M.Arief Hidayat, S. Kom. M. Kom

PENGESAHAN

Skripsi berjudul “**Sistem Informasi Klinik Mata Dengan Prediksi Waktu Pemeriksaan Menggunakan Algoritma C4.5 Pada Fitur Registrasi Pasien**” telah diuji dan disahkan pada :

Hari : Selasa

Tanggal : 12 Mei 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji

Penguji I

Penguji II

Prof. Drs. Slamin, M.Comp.Sc.,Ph.D
NIP. 196704201992011001

Yanuar Nurdiansyah S.T.,M.Cs
NIP. 198201012010121004

Mengesahkan
Ketua Program Studi,

Prof. Drs. Slamin, M.Comp.Sc.,Ph.D
NIP. 196704201992011001

RINGKASAN

Sistem Informasi Klinik Mata Dengan Prediksi Waktu Pemeriksaan Menggunakan Algoritma C4.5 Pada Fitur Registrasi Pasien, Zakiyah Qurrotul Aini, 102410101063; 2015:114 halaman ; Program Studi Sistem Informasi Universitas Jember.

Klinik Mata yang bertempat di Bangkalan, Madura merupakan sebuah klinik yang bekerja dalam bidang kesehatan mata yang bertugas memberikan pelayanan kesehatan mata kepada masyarakat. Klinik mata tersebut melayani pasien dalam bentuk pemeriksaan pasien, tindakan operasi, pemesanan kacamata, dan lain-lain. Selama ini segala kegiatan klinik maupun pengelolaan data pasien dan obat dilakukan secara manual. Pasien yang ingin melakukan tindakan pemeriksaan dan operasi harus melakukan beberapa proses yang rumit yang menimbulkan beberapa permasalahan antara sistem informasi dan pasien ataupun sistem informasi itu sendiri.

Tujuan dilakukan penelitian ini adalah merancang dan membangun sistem yang diharapkan dapat digunakan untuk membantu pihak klinik mata dalam mengelola data dan transaksi penjualan serta dapat menentukan waktu pemeriksaan pasien sehingga dapat mengurangi keresahan pasien dalam menunggu antrian di klinik mata tersebut.

Sistem Informasi Klinik Mata menerapkan algoritma c4.5 untuk memprediksi waktu pemeriksaan pada fitur registrasi pasien karena algoritma ini merupakan algoritma yang dapat digunakan untuk memprediksi suatu kejadian.

PRAKATA

Puji syukur kepada Allah SWT, atas segala limpahan rahmat, hidayat dan karunia-Nya maka penulis dapat menyelesaikan skripsi yang berjudul “Sistem Informasi Klinik Mata Dengan Prediksi Waktu Pemeriksaan Menggunakan Algoritma C4.5 Pada Fitur Registrasi Pasien”. Skripsi ini disusun guna melengkapi tugas akhir dan sebagai salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Dalam proses penyusunan skripsi ini, penulis tidak lepas dari bantuan berbagai pihak. Maka dari itu, dalam kesempatan ini penulis ingin mengucapkan banyak terima kasih kepada:

1. Bapak Anang Andrianto, ST., MT dan Bapak M.Arief Hidayat, S.Kom,.M.Kom sebagai Dosen Pembimbing utama dan Pendamping;
2. Bapak Prof. Drs. Slamin, M.Comp.Sc.,Ph.D selaku ketua Program Studi Sistem Informasi yang menjadi tempat mahasiswa menuntut ilmu;
3. Abi dan Ibu yang telah memberikan doa, dukungan, dan semangat kepada penulis sehingga skripsi ini dapat diselesaikan;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan Program Studi Sistem Informasi.

Demi kesempurnaan skripsi ini, penulis akan menerima segala masukan baik berupa kritik atau saran dari semua pihak. Harapan penulis dengan terselesaikannya skripsi ini adalah semoga skripsi ini dapat bermanfaat bagi pihak - pihak yang membutuhkan.

Jember, 12 Mei 2015

Penulis

DAFTAR ISI

	Halaman
PERSEMBAHAN	iii
MOTTO	iv
PERNYATAAN.....	v
PENGESAHAN PEMBIMBING.....	vi
HALAMAN PEMBIMBINGAN	vii
PENGESAHAN	viii
RINGKASAN	ix
PRAKATA	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN.....	xviii
BAB 1. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan dan Manfaat.....	3
1.3.1. Tujuan	3
1.3.2. Manfaat	3
1.4. Batasan Masalah.....	4
1.5. Sistematika Penulisan.....	4
BAB 2. TINJAUAN PUSTAKA	6
2.1. Sistem Informasi.....	6
2.2. Sistem Registrasi Pasien.....	6

2.3.	Rekam Medis.....	7
2.4.	Pasien.....	8
2.5.	Algoritma C4.5.....	8
2.6.	Metode Evaluasi Algoritma Klasifikasi	14
2.7.	<i>Software Development Life Cycle (SDLC)</i>	16
2.7.1	Analisis.....	18
2.7.2	Desain Sistem.....	18
2.7.3	<i>Coding</i>	18
2.7.4	<i>Testing</i>	18
BAB 3.	METODOLOGI PENELITIAN	20
3.1.	Pendekatan Penelitian.....	20
3.2.	Tempat dan Waktu Penelitian	20
3.3.	Alur Penelitian.....	20
3.3.1.	Studi Literatur	22
3.3.2.	Tahap Pengumpulan Data	22
3.3.3.	Penerapan Algoritma C4.5 untuk Prediksi Waktu Pemeriksaan Pasien.....	24
3.3.4.	Tahap Pengembangan Sistem	40
BAB 4.	ANALISIS DAN PENGEMBANGAN SISTEM.....	42
4.1.	Pengumpulan Data	42
4.1.1	Data Set.....	42
4.1.2	<i>Workflow</i> atau Alur Kerja pada Objek Penelitian.....	43
4.2.	Algoritma C4.5 untuk Prediksi Waktu Pemeriksaan Pasien	46
4.3.	Kebutuhan Sistem.....	48
4.3.1	Kebutuhan Fungsional	48
4.3.2	Kebutuhan Non-fungsional	49
4.4.	Desain Sistem.....	49
4.4.1	Business Process	50
4.4.2	Pembagian Fitur	50

Sumber : (Hasil Analisis, 2014).....	51
4.4.3 <i>Usecase Diagram</i>	51
4.4.4 Desain Sistem Tahap <i>Increment 1</i>	56
4.4.5 Desain Sistem Tahap <i>Increment 2</i>	81
4.4.6 Desain Sistem Tahap <i>Increment 3</i>	82
4.4.7 Desain Sistem Tahap <i>Increment 4</i>	82
4.5. Pengkodean	82
BAB 5. HASIL DAN PEMBAHASAN	87
5.1. Implementasi Algoritma C4.5 untuk Prediksi Waktu Pemeriksaan Pasien .	87
5.2. Implementasi Sistem	94
5.2.1 <i>Login</i>	94
5.2.2 Implementasi Tahap <i>Increment 1</i>	96
5.2.3 Implementasi Tahap <i>Increment 2</i>	105
5.2.4 Implementasi Tahap <i>Increment 3</i>	105
5.2.5 Implementasi Tahap <i>Increment 4</i>	105
5.3. Pengujian Sistem	105
5.3.1 <i>Blackbox Testing</i>	105
5.3.2 <i>Whitebox Testing</i>	106
BAB 6. KESIMPULAN	115
6.1. Kesimpulan.....	115
6.2. Saran	115
DAFTAR PUSTAKA	117
LAMPIRAN	119

DAFTAR TABEL

Tabel 2.1 Contoh Data <i>Training</i>	11
Tabel 2.2 Model <i>Confussion Matrix</i>	14
Tabel 3.1 Atribut dan Nilai Atribut.....	22
Tabel 3.2 Contoh Data <i>Training</i>	30
Tabel 3.3 Probabilitas Kelas	31
Tabel 3.4 tabel distribusi kelas atribut mata merah.....	31
Tabel 3.5 Tabel Distribusi Kelas Atribut Umur.....	32
Tabel 3.6 Tabel hasil perhitungan <i>information-gain</i> tiap nilai batas	33
Tabel 3.7 Hasil Perhitungan <i>Information-gain</i> dan <i>Gain-ratio</i> Iterasi 1	33
Tabel 3.8 Probabilitas kelas iterasi 2.....	35
Tabel 3.9 Distribusi kelas atribut mata merah iterasi 2.....	35
Tabel 3.10 Hasil perhitungan <i>information-gain</i> dan <i>gain-ratio</i> tiap atribut pada iterasi 2.....	36
Tabel 4.1 Pembagian Fitur	51
Tabel 4.2 Definisi Aktor dan Hak Akses	54
Tabel 4.3 Definisi <i>Usecase</i>	55
Tabel 4.4 <i>Usecase scenario</i> data user	58
Tabel 4.5 <i>Usecase Scenario</i> Buat Akun.....	60
Tabel 4.6 <i>Usecase scenario</i> aktivasi akun lewat <i>email</i>	61
Tabel 4.7 <i>Usecase scenario</i> data pasien.....	61
Tabel 4.8 <i>Usecase scenario</i> registrasi pasien baru.....	63
Tabel 4.9 <i>Usecase Scenario</i> registrasi pasien lama	65
Tabel 4.10 potongan kode program untuk fitur registrasi pasien lama (<i>controller</i>) ...	83
Tabel 4.11 potongan kode program <i>function</i> prediksi().....	83
Tabel 5.1 <i>Confussion Matrix</i> Hasil Klasifikasi Waktu Penanganan Pasien.....	93

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Contoh Struktur <i>Decision tree</i>	9
Gambar 2.2 <i>Incremental Model</i> (Original: Mills).....	17
Gambar 3.1 Alur Penelitian.....	21
Gambar 3.2 <i>Flowchart</i> Penerapan Algoritma C4.5 pada Fitur Registrasi Pasien.....	25
Gambar 3.3 <i>Flowchart</i> proses <i>request</i> data.....	26
Gambar 3.4 <i>Flowchart</i> proses <i>learning</i>	27
Gambar 3.5 <i>flowchart</i> proses pembangunan <i>tree</i>	28
Gambar 3.6 <i>flowchart</i> proses membuat <i>node</i>	29
Gambar 3.7 <i>Decision tree</i> iterasi 1.....	34
Gambar 3.8 <i>Decision tree</i> iterasi 2.....	37
Gambar 3.9 <i>Decision tree</i> (hasil akhir).....	38
Gambar 4.1 <i>Workflow</i> Pemeriksaan.....	43
Gambar 4.2 <i>Workflow</i> Penjualan Obat.....	44
Gambar 4.3 <i>Workflow</i> Pembelian Obat.....	44
Gambar 4.4 <i>Workflow</i> Penjualan Kacamata.....	45
Gambar 4.5 <i>Workflow</i> Pengambilan Kacamata	45
Gambar 4.6 <i>Business Process</i> Sistem Informasi Klinik Mata	50
Gambar 4.7 <i>Usecase Diagram</i> SIKM.....	53
Gambar 4.8 <i>usecase</i> diagram tahap <i>increment</i> 1.....	57
Gambar 4.9 <i>Activity</i> diagram data user	67
Gambar 4.10 <i>Activity Diagram</i> Buat Akun.....	68
Gambar 4.11 <i>Activity Diagram</i> Aktivasi Akun Lewat Email	69
Gambar 4.12 <i>Activity</i> diagram data pasien.....	70
Gambar 4.13 <i>Activity Diagram</i> Registrasi Pasien Baru	71
Gambar 4.14 <i>Activity</i> diagram registrasi pasien.....	72

Gambar 4.15 <i>Sequence</i> diagram data user	74
Gambar 4.16 <i>Sequence Diagram</i> Buat Akun.....	75
Gambar 4.17 <i>Sequence Diagram</i> Aktivasi Akun Lewat Email.....	76
Gambar 4.18 <i>Sequence</i> diagram data pasien.....	77
Gambar 4.19 <i>Sequence Diagram</i> Registrasi Pasien Baru.....	78
Gambar 4.20 <i>Sequence Diagram</i> Registrasi Pasien.....	79
Gambar 4.21 <i>Class Diagram</i> Tahap <i>Increment 1</i>	80
Gambar 4.22 ERD Tahap <i>Increment 1</i>	81
Gambar 5.1 Tampilan Menu data <i>training</i> dan data <i>testing</i>	87
Gambar 5.2 tampilan menu proses <i>mining</i>	88
Gambar 5.3 Potongan kode program perhitungan entropi.....	88
Gambar 5.4 Potongan kode program perhitungan <i>information gain</i>	89
Gambar 5.5 Potongan kode program perhitungan <i>split-info</i>	89
Gambar 5.6 Potongan kode program perhitungan <i>gain-ratio</i>	89
Gambar 5.7 Potongan kode program pencarian <i>gain-ratio</i> tertinggi.....	89
Gambar 5.8 Hasil perhitungan proses <i>mining</i> dengan algoritma c4.5.....	90
Gambar 5.9 hasil pohon keputusan	90
Gambar 5.10 <i>Rule</i> yang dihasilkan sistem	91
Gambar 5.11 tampilan menu perbandingan	92
Gambar 5.12 Tampilan menu <i>login</i> karyawan.....	95
Gambar 5.13 Tampilan menu <i>login</i> pasien	95
Gambar 5.14 Tampilan menu data <i>user</i>	96
Gambar 5.15 tampilan <i>form</i> tambah data <i>user</i>	97
Gambar 5.16 tampilan <i>form</i> edit data <i>user</i>	97
Gambar 5.17 Tampilan <i>form</i> buat akun	98
Gambar 5.18 Contoh pesan aktivasi <i>email</i>	99
Gambar 5.19 Tampilan menu data pasien.....	100
Gambar 5.20 Tampilan <i>form</i> tambah data pasien	100
Gambar 5.21 Tampilan <i>form</i> edit data pasien	101

Gambar 5.22 Tampilan <i>view detail</i> data pasien	102
Gambar 5.23 Tampilan <i>form</i> registrasi pasien baru	103
Gambar 5.24 tampilan <i>form</i> registrasi pasien lama untuk resepsionis.....	103
Gambar 5.25 tampilan <i>form</i> registrasi pasien lama untuk pasien	104
Gambar 5.26 Potongan Listing Program Fitur Registrasi Pasien Lama Bagian <i>Controller</i>	106
Gambar 5.27 Grafik Alir <i>Controller</i> Registrasi Pasien Lama.....	107
Gambar 5.28 Potongan <i>Listing</i> Program Model Fitur Registrasi Pasien Lama	109
Gambar 5.29 Grafik Alir <i>Controller</i> Registrasi Pasien Lama.....	110

DAFTAR LAMPIRAN

LAMPIRAN A. DESAIN SISTEM.....	119
A.1. Desain Sistem Tahap <i>Increment 2</i>	119
A.2. Desain Sistem Tahap <i>Increment 3</i>	138
A.3. Desain Sistem Tahap <i>Increment 4</i>	151
LAMPIRAN B. IMPLEMENTASI SISTEM.....	163
B.1. Implementasi Sistem Tahap <i>Increment Dua</i>	163
B.2. Implementasi Sistem Tahap <i>Increment Tiga</i>	167
B.3. Implementasi Sistem Tahap <i>Increment Empat</i>	168
LAMPIRAN C. PENULISAN KODE PROGRAM.....	170
C.1. Penulisan Kode Program Tahap <i>Increment 1</i>	170
C.2. Penulisan Kode Program Tahap <i>Increment 2</i>	177
C.3. Penulisan Kode Program Tahap <i>Increment 3</i>	186
C.4. Penulisan Kode Program Tahap <i>Increment 4</i>	193
LAMPIRAN D. <i>BLACKBOX TESTING</i>	199
D.1 <i>Blackbox Testing</i> Tahap <i>Increment 1</i>	199
D.2 <i>Blackbox Testing</i> Tahap <i>Increment 2</i>	204
D.3 <i>Blackbox Testing</i> Tahap <i>Increment 3</i>	209
D.4 <i>Blackbox Testing</i> Tahap <i>Increment 4</i>	212
LAMPIRAN E. <i>WHITEBOX TESTING</i>	216
E.1. <i>Whitebox Testing</i> Tahap <i>Increment Dua</i>	216
E.2. <i>Whitebox Testing</i> Tahap <i>Increment Dua</i>	230
E.3. <i>Whitebox Testing</i> Tahap <i>Increment Tiga</i>	250
E.4. <i>Whitebox Testing</i> Tahap <i>Increment Empat</i>	260
LAMPIRAN F. DATA SET	268

BAB 1. PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi yang sangat pesat seperti sekarang ini menuntut para ahli pengembang teknologi informasi bekerja dengan cepat dan kreatif. Mereka harus mampu menyediakan pelayanan berbagai macam aplikasi yang dibutuhkan *user* dalam berbagai bidang dan kebutuhan yang beragam sebagai salah satu manfaat dari perkembangan teknologi yaitu membantu kinerja manusia.

Penyajian informasi yang masih bersifat manual, yaitu dengan pencatatan langsung di kertas akan membuat kinerja *user* menjadi tidak efisien. Pencatatan manual dapat dikatakan lebih mengedepankan suatu subyek utama yaitu manusia sebagai tumpuan dalam proses pengerjaannya. Apabila *user* melakukan suatu kesalahan maka kesalahan tersebut akan mempengaruhi proses pencatatan yang akan berakibat buruk pada *record*-nya. Selain itu, pencatatan manual yang masih menggunakan media kertas menjadi masalah baru jika kertas-kertas tersebut hilang. Maka dari itu dibutuhkan sebuah sistem informasi yang dapat membantu kinerja manusia, yaitu sebuah sistem informasi yang dapat menangani segala transaksi yang ada.

Klinik Mata yang bertempat di Bangkalan, Madura merupakan sebuah klinik yang bekerja dalam bidang kesehatan mata yang bertugas memberikan pelayanan kesehatan mata kepada masyarakat. Klinik mata tersebut melayani pasien dalam bentuk pemeriksaan pasien, tindakan operasi, pemesanan kacamata, dan lain-lain. Selama ini segala kegiatan klinik maupun pengelolaan data pasien dan obat dilakukan secara manual. Pasien yang melakukan tindakan pemeriksaan dan operasi harus melakukan beberapa proses yang rumit yang menimbulkan beberapa permasalahan antara sistem informasi dan pasien ataupun sistem informasi itu sendiri.

Proses tersebut diantaranya adalah proses registrasi yang masih dicatat secara manual menggunakan prinsip *First in First out* (FIFO) dengan cara pasien harus datang sendiri ke klinik kemudian mendaftar dan menunggu antrian untuk diperiksa. Hal tersebut tidak akan menimbulkan masalah jika antrian pasien tidak panjang, namun jika antrian pasien pada saat itu cukup panjang maka pasti akan terdapat masalah yang akan timbul, yaitu ketidaknyamanan pasien, karena pasien dengan keluhan penyakit serius harus mendapatkan penanganan khusus yang pastinya memakan waktu pemeriksaan yang cukup lama sehingga akan berpengaruh kepada waktu tunggu pasien lain yang melakukan antri.

Sedangkan untuk tindakan operasi, selain melalui proses registrasi pasien harus melakukan proses penjadwalan operasi. Penjadwalan operasi harus menyesuaikan waktu pasien dan dokter yang bersangkutan. Selain hal-hal tersebut data pasien, rekam medis, data obat, data kaca mata, serta data-data transaksi masih dikelola secara manual. Semakin banyaknya pasien dan banyaknya permintaan pelayanan, maka pengelolaan data secara manual sudah tidak efisien dan beresiko. Dapat dikatakan tidak efisien karena data yang dikelola secara manual akan memakan berkarton kertas, dan jika salah satu kertas yang berisi data penting tersebut hilang maka akan mempengaruhi data-data lainnya, itulah mengapa dikatakan bahwa pengelolaan data secara manual juga beresiko.

Berdasarkan masalah-masalah di atas, maka diperlukan sebuah pengelolaan data dalam bentuk sistem terkomputerisasi yaitu sistem informasi klinik dokter mata. Algoritma C4.5 merupakan salah satu algoritma *decision tree* yang dapat digunakan untuk memprediksi suatu kejadian, algoritma ini merupakan salah satu algoritma induksi pohon keputusan yang dikembangkan oleh J. Ross Quinlan. Pemanfaatan Algoritma C4.5 pada sistem digunakan pada fitur registrasi pasien untuk proses penentuan atau prediksi waktu penanganan pasien, metode ini menentukan waktu penanganan dengan melihat keluhan yang dialami oleh pasien. Data keluhan tersebut

nantinya akan diproses menggunakan *decision tree* sehingga dapat menghasilkan suatu keputusan waktu penanganan pasien yang diterjemahkan ke dalam bahasa linguistik yaitu, cepat, sedang, dan lama. Pasien dapat melakukan pendaftaran secara *online* dengan mengisi data pendaftaran dan keluhan yang dialami, kemudian pasien dapat melihat nomer antrian dan waktu pemeriksaan pada *website*, sehingga pasien tersebut dapat menunggu di rumah sampai dengan waktu periksa yang telah ditentukan.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka penulis dapat mengidentifikasi masalah sebagai berikut:

1. Bagaimana membangun fitur registrasi pasien dengan memprediksi waktu pemeriksaan pasien menggunakan algoritma c4.5?
2. Bagaimana membangun sistem informasi klinik mata sesuai dengan kebutuhan?

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 fitur registrasi pasien dengan memprediksi waktu pemeriksaan pasien dengan menggunakan algoritma c4.5.
2. Membangun sistem informasi klinik mata yang sesuai dengan kebutuhan.

1.3.2. Manfaat

Adapun manfaat dari penelitian ini adalah :

1. **Bagi Klinik**

Sistem informasi yang dibangun ini diharapkan bermanfaat dalam mempermudah aktifitas manajemen yang ada pada instansi tersebut serta dapat membantu pasien untuk memperkirakan waktu pemeriksaan sehingga tidak menunggu lama di klinik untuk mendapat giliran periksa.

2. **Bagi Akademisi**

Hasil penelitian ini diharapkan dapat bermanfaat dalam proses pembelajaran serta dapat menjadi acuan dalam penelitian-penelitian selanjutnya.

3. **Bagi Penulis**

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

1.4. Batasan Masalah

Penulis memberi batasan masalah untuk objek dan tema yang dibahas sehingga tidak terjadi penyimpangan dalam proses penulisan dan pembuatan sistem informasi. Berikut adalah masalah yang dicantumkan:

1. Sistem yang dibangun merupakan Sistem Informasi Klinik Mata yang meliputi registrasi pasien, pengelolaan data pasien, rekam medis, obat, serta kacamata, melayani penjualan dan pembelian kacamata serta obat, dan penjadwalan operasi.
2. Sistem yang dibuat tidak mampu mengelola laporan keuangan.
3. Sistem informasi yang dibuat berbasis web.

1.5. Sistematika Penulisan

Sistematika penulisan dan keruntunan skripsi ini disusun sebagai berikut:

1. **Pendahuluan**

Bab ini menjelaskan latar belakang, perumusan masalah, tujuan dan manfaat, ruang lingkup studi dan sistematika penulisan.

2. Tinjauan Pustaka

Bab ini berisi materi, informasi, dan kajian teori yang digunakan dalam penulisan.

3. Metodologi Penelitian

Bab ini menjelaskan tentang metodogi penelitian yang digunakan selama proses penulisan berlangsung.

4. Analisis dan Pengembangan Sistem

Bab ini menjelaskan tentang analisis dan pengembangan sistem yang dibangun.

5. Hasil dan Pembahasan

Bab ini berisi tentang hasil dan pembahasan dari system yang sudah dibangun.

6. Penutup

Bab ini berisi tentang kesimpulan dari penulis dan saran untuk penulis selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Bab 2 ini akan menguraikan lebih jauh tentang teori serta konsep yang diperlukan dan dijadikan sebagai kerangka pemikiran dalam penelitian.

2.1. Sistem Informasi

Sistem adalah suatu kumpulan dari beberapa elemen yang saling berkaitan dan berinteraksi guna mencapai tujuan tertentu. Informasi merupakan kumpulan dari data-data yang memiliki nilai. Informasi adalah data yang mempunyai makna, artinya ketika sesuatu hal (data) tidak mempunyai makna maka belum dapat dikatakan sebagai sebuah informasi (Surachman 2008).

Sistem informasi adalah *entity* (kesatuan) formal yang terdiri dari berbagai sumber daya fisik maupun logika yang mempunyai tujuan untuk menyediakan dan mensistematikan informasi yang merefleksikan seluruh kejadian atau kegiatan yang diperlukan untuk mengendalikan operasi-operasi organisasi dari mengambil mengolah, menyimpan informasi yang diperlukan dan untuk terjadinya komunikasi yang diperlukan untuk mengoperasikan seluruh aktifitas di dalam organisasi (Prahasta 2001).

2.2. Sistem Registrasi Pasien

Menurut kamus besar bahasa Indonesia kata registrasi yang berasal kata register dapat diartikan menjadi “pendaftaran” dan “pendataan”. Registrasi merupakan suatu kegiatan pencatatan atau pendaftaran. Registrasi pasien merupakan suatu kegiatan pencatatan atau pendaftaran data pasien pada tempat praktek, klinik, ataupun rumah sakit sebagai langkah awal pasien sebelum melakukan pemeriksaan.

Sistem registrasi pasien adalah sebuah sistem terkomputerisasi untuk mengolah atau menangani data pendaftaran pasien yang akan melakukan pemeriksaan.

2.3. Rekam Medis

Menurut PERMENKES No: 269/MENKES/PER/III/2008, rekam medis adalah berkas yang berisi catatan dan dokumen antara lain identitas pasien, hasil pemeriksaan, pengobatan yang telah diberikan, serta tindakan dan pelayanan lain yang telah diberikan kepada pasien.

Pencatatan rekam medis dapat dilakukan dengan dua cara, yaitu secara tertulis maupun elektronik. Rekam medis yang digunakan sebagai data dalam penelitian ini meliputi data identitas pasien, tanggal dan waktu pemeriksaan, hasil anamnesis (keluhan dan riwayat penyakit pasien), hasil pemeriksaan, diagnosa, tindakan yang dilakukan, pelayanan lain yang telah dilakukan terhadap pasien, serta persetujuan bila perlu dilakukan tindakan-tindakan tertentu.

Kegiatan pencatatan rekam medis memiliki beberapa manfaat, yaitu :

1. Sebagai sumber informasi kepada pihak lain
2. Sebagai pemeliharaan kesehatan dan pengobatan pasien
3. Sebagai dasar ilmu dalam pembelajaran di dunia pendidikan dan dalam penelitian
4. Sebagai bahan penelitian
5. Sebagai alat bukti dalam proses hukum, disiplin kedokteran, disiplin kedokteran gigi, etika kedokteran, dan etika kedokteran gigi
6. Sebagai dasar pembiayaan pelayanan kesehatan.

2.4. Pasien

Menurut PERMENKES No: 269/MENKES/PER/III/2008, Pasien adalah setiap orang yang melakukan konsultasi masalah kesehatannya untuk memperoleh pelayanan kesehatan yang diperlukan baik secara langsung maupun tidak langsung. Pasien merupakan aset yang berharga bagi instansi yang memberikan pelayanan kesehatan seperti rumah sakit, klinik, puskesmas, praktek dokter, dan lain-lain.

2.5. Algoritma C4.5

Data Mining adalah suatu proses pencarian pola data dari sekumpulan data yang besar sehingga menghasilkan suatu *knowledge* atau informasi yang baru yang tidak dapat diperkirakan sebelumnya. *Data Mining* didefinisikan sebagai proses penemuan pola dalam data (Witten dan Frank 2005). Tugas utama *data mining* dikelompokkan menjadi *description, estimation, prediction, classification, clustering,* dan *assosiation* (Larose 2005).

Klasifikasi merupakan proses penempatan objek atau konsep tertentu ke dalam satu set kategori berdasarkan objek yang digunakan. Salah satu teknik klasifikasi yang paling populer digunakan adalah *decision tree* (Han and Kamber 2006). *Decision tree* adalah salah satu metode klasifikasi yang menggunakan representasi struktur pohon. *Node* paling atas pada *decision tree* disebut *root*. Setiap *node* merepresentasikan atribut, sedangkan cabangnya merepresentasikan nilai dari atribut, dan daunnya merepresentasikan kelas. Terdapat 3 jenis *node* pada metode *decision tree*, yaitu:

1. *Root Node*

Root node merupakan bagian paling atas dari *decision tree*. *Root node* tidak memiliki *input* dan memiliki *output* lebih dari satu.

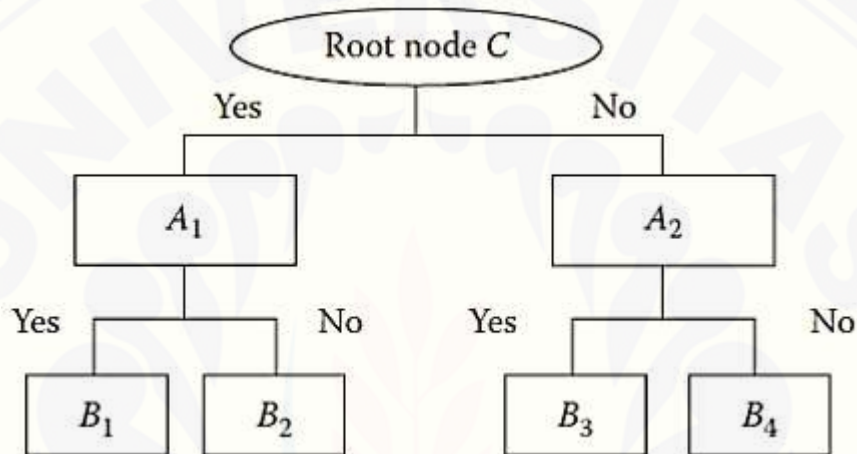
2. *Internal Node*

Internal node merupakan *node* percabangan. *Internal node* hanya terdapat satu *input* dan memiliki dua atau lebih *output*.

3. *Leaf Node*

Leaf node merupakan *node* akhir dari *decision tree*. *Node* ini memiliki satu *input* dan tidak memiliki *output*.

Contoh struktur *decision tree* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Contoh Struktur *Decision tree*
Sumber: (Dua and Du 2011)

Seperti ditunjukkan dalam Gambar 2.1, *decision tree* tergantung pada aturan *if-then*, tetapi tidak membutuhkan parameter dan metrik. Strukturnya yang sederhana dan dapat ditafsirkan memungkinkan *decision tree* untuk memecahkan masalah atribut *multi-type*. *Decision tree* juga dapat mengelola nilai-nilai yang hilang atau data *noise* (Dua and Du 2011).

Decision tree memiliki banyak algoritma yang dapat digunakan dalam pembentukan pohon keputusan, salah satunya adalah algoritma C4.5. Algoritma C4.5 merupakan algoritma yang dapat digunakan untuk memprediksi suatu kejadian, algoritma ini merupakan salah satu algoritma induksi pohon keputusan yang dikembangkan oleh J. Ross Quinlan. Prediksi merupakan salah satu tugas data *mining* yang mencari sebuah model atau *rule* yang mampu melakukan prediksi pada suatu

data yang belum pernah ada. Algoritma C4.5 merupakan algoritma hasil pengembangan dari algoritma ID. Hasil dari pengembangan yaitu, algoritma C4.5 dapat menghasilkan pohon yang lebih akurat, algoritma ini dapat menangani data *training* dengan nilai atribut yang hilang, serta dapat menangani atribut kontinyu atau numerik.

Proses membuat sebuah pohon keputusan dengan algoritma c4.5 dibagi menjadi beberapa tahap, yaitu :

1. Mempersiapkan data *training* yang dapat diambil dari data yang pernah terjadi sebelumnya dan telah dikelompokkan dalam beberapa kelas.
2. Menentukan *root* dari pohon dengan ukuran pemilihan atribut (*Attribut selection measure*) yaitu ukuran statistika yang digunakan untuk memilih atribut mana yang terbaik yang digunakan sebagai *root*. Algoritma c4.5 memiliki dua cara perhitungan *attribut selection measure*, yaitu *information gain* dan *gain-ratio*. Sebelum menghitung *information gain* perlu dilakukan perhitungan entropi kelas terlebih dahulu dengan menggunakan rumus:

$$H(X) = - \sum_i p_i \log_2 p_i \dots\dots\dots (2.1)$$

Keterangan:

X = Himpunan kasus

i = Jumlah partisi X

p_i = Jumlah proporsi X_i terhadap X

Kemudian menghitung nilai entropi masing-masing atribut dengan menggunakan rumus:

$$H_S(X) = \sum_{i=1}^k P_i H_S(X_i) \dots\dots\dots (2.2)$$

Hitung *information gain* dengan rumus:

$$gain(S) = H(X) - H_S(T).....(2.3)$$

Kemudian menghitung *gain-ratio*. Menghitung *gain-ratio* diperlukan untuk menghitung *split information* dengan menggunakan rumus:

$$Split - info(X) = - \sum_{i=1}^n ((|T_i|/|T|) \log_2(|T_i|/|T|)).... (2.4)$$

Rumus untuk menghitung *gain-ratio*:

$$Gain - ratio(X) = gain(X)/split - info(X)....(2.5)$$

3. Ulangi langkah kedua hingga semua *record* terpartisi.
 Proses partisi pohon keputusan akan berhenti saat:
 - a. Semua *record* dalam simpul N mendapat kelas yang sama.
 - b. Tidak ada atribut dalam *record* yang dapat dipartisi lagi.
 - c. Tidak ada *record* didalam cabang yang kosong

Contoh perhitungan *information gain* dan *gain-ratio* untuk contoh data *training* pada tabel 2.1.

Tabel 2.1 Contoh Data *Training*

Attr1	Attr2	Attr3	Class
A	70	True	1
A	90	True	2
A	85	False	2
A	95	False	2
A	70	False	1
B	90	True	1
B	78	False	1
B	65	True	1
B	75	False	1
C	80	True	2
C	70	True	2
C	80	False	1
C	80	False	1

C	96	False	1
---	----	-------	---

Terdapat 2 macam kelas, yaitu : kelas 1 dan 2.

Jumlah record yang memiliki kelas 1 = 9

Jumlah record yang memiliki kelas 2 = 5

Jumlah record keseluruhan = 14

Maka, entropi sebelum splitting yang dihitung dengan persamaan 2.1 adalah :

$$H(T) = - 9/14 \log_2 (9/14) - 5/14 \log_2 (5/14) = 0,94$$

Tes Attr1 (x1) :

Ada 3 macam nilai atribut (outcomes), yaitu : A, B, dan C

- Untuk nilai atribut A

Jumlah record yang memiliki nilai attr1 A = 5

Jumlah record yang memiliki nilai attr1 A dan kelas 1 = 2

Jumlah record yang memiliki nilai attr1 A dan kelas 2 = 3

- Untuk nilai atribut B

Jumlah record yang memiliki nilai attr1 B = 4

Jumlah record yang memiliki nilai attr1 B dan kelas 1 = 4

Jumlah record yang memiliki nilai attr1 B dan kelas 2 = 0

- Untuk nilai atribut C

Jumlah record yang memiliki nilai attr1 C = 5

Jumlah record yang memiliki nilai attr1 C dan kelas 1 = 3

Jumlah record yang memiliki nilai attr1 C dan kelas 2 = 2

Maka Entropi dari attr1, yang dihitung dengan menggunakan persamaan 2.2 adalah :

$$H_{x_1}(T) = 5/14(-2/5 \log_2 (2/5) - 3/5 \log_2 (3/5)) + 4/14 (-4/4 \log_2 (4/4) - 0/4 \log_2 (0/4)) + 5/14 (-3/5 \log_2 (3/5) - 2/5 \log_2 (2/5)) = 0,694$$

Maka, *gain* yang dihitung dengan menggunakan persamaan 2.3 adalah :

$$\text{Gain}(x_1) = 0,94 - 0,694 = 0,246$$

Kemudian *split-information* dihitung menggunakan persamaan 2.4.

$$\begin{aligned}\text{Split-info} &= -5/14 \log_2 (5/14) - 4/14 \log_2 (4/14) - 5/14 \log_2 (5/14) \\ &= 1,557\end{aligned}$$

Maka, *gain-ratio* yang dihitung menggunakan persamaan 2.5.

$$\text{gain-ratio}(x_1) = 0,246/1,557 = 0,15$$

Catatan : untuk perhitungan yang memiliki nilai pembilang 0 (misal $0/y = x$), maka $\log_2 (x) = 0$. (Larose 2005)

Perhitungan *information gain* dan *gain-ratio* dilakukan juga pada atribut yang lain. Apabila ada atribut yang bertipe numeris (kontinyu), diperlukan cara untuk menentukan nilai batas (*threshold*) yang dijadikan nilai pembagi (*split-point*) untuk membagi data selanjutnya. Nilai *information gain* dari kandidat nilai pembagi tersebut dipilih untuk dijadikan *information gain* atribut. Dalam menangani atribut numeris (kontinyu), algoritma c4.5 tidak menggunakan nilai tengah dari atribut seperti yang dilakukan pada algoritma ID3. Algoritma c4.5 memiliki cara tersendiri untuk menghitung nilai batas yang optimal (Kantardzic 2003). Adapun langkah-langkah untuk menentukan nilai batas yaitu:

1. Urutkan nilai atribut dari kecil ke besar.
2. Ambil nilai Z yang mungkin untuk selanjutnya digunakan untuk menentukan apakah sampel termasuk dalam $Y \leq Z$ atau $Y > Z$. Jumlah banyaknya Z yang mungkin yaitu $n-1$ (n =banyaknya nilai yang berbeda pada *record* atribut).
3. Hitung *information gain* untuk tiap-tiap Z.
4. Z dengan *information gain* tertinggi menjadi nilai batas yang terpilih dan nilai *gain*-nya dibandingkan dengan *gain* atribut lain.

Contoh perhitungan untuk menangani atribut bertipe numeris (kontinyu):

Misalkan untuk menghitung information gain atribut 2 dari tabel data *training* yang diberikan sebelumnya. Setelah nilai atribut diurutkan, maka didapat nilai-nilai yang terdapat pada atribut 2 yaitu {65, 70, 75, 78, 80, 85, 90, 95, 96} dan nilai-nilai yang mungkin dijadikan nilai nilai batas Z, yaitu : 65, 70, 75, 78, 80, 85, 90, 95. Kemudian, dari ke delapan buah nilai tersebut, dihitung information gain-nya dan nilai nilai batas dengan information gain tertinggi dipilih menjadi nilai nilai batas Z. Misalkan sebagai contoh, nilai Z yang terpilih yaitu Z = 80 (Atribut 2 ≤ 80 or Atribut 2 > 80). Perhitungan information gain-nya yaitu sebagai berikut :

$$H_{x2}(T) = 9/14(-7/9 \log_2 (7/9) - 2/9 \log_2 (2/9)) + 5/14 (-2/5 \log_2 (2/5) - 3/5 \log_2 (3/5)) = 0,837$$

$$\text{Gain}(x_2) = 0,94 - 0,837 = 0,103$$

2.6. Metode Evaluasi Algoritma Klasifikasi

Mengevaluasi algoritma pengklasifikasian dalam data *mining* yaitu dengan mengukur tingkat akurasi algoritma tersebut, salah satu metode yang dapat digunakan adalah *confussion matrix*. *Confussion matrix* merupakan salah satu metode yang digunakan pada data *mining* untuk melakukan perhitungan akurasi. *Confussion matrix* merupakan sebuah metode untuk evaluasi yang menggunakan tabel matrix (Andriani 2013). Contoh tabel *confussion matrix* dapat dilihat pada tabel 2.2.

Tabel 2.2 Model *Confussion Matrix*

<i>Correct Classification</i>	<i>Classified As</i>	
	+	-
+	<i>True Positives</i>	<i>False Negatives</i>
-	<i>False Positives</i>	<i>True Negatives</i>

Sumber : Han & Kamber (dalam Andriani, 2013)

Pada tabel 2.2, data set yang digunakan menggunakan dua kategori kelas yaitu positif dan negatif. *True* positif merupakan jumlah *record* positif yang diklasifikasikan sebagai positif, *false* positif merupakan jumlah *record* negatif yang diklasifikasikan sebagai positif, *false* negatif adalah jumlah *record* positif yang di klasifikasikan sebagai *record* negatif, sedangkan *true* negatif merupakan jumlah *record* negatif yang diklasifikasikan sebagai negatif.

Langkah untuk mengevaluasi algoritma pengklasifikasian yang pertama yaitu melakukan pengujian dengan menggunakan data uji atau data *testing*, kemudian masukkan ke dalam *confussion matrix*. Dari tabel *confussion matrix*, dapat dihitung nilai *accuracy* dan *error rate*. Menurut Han & Kamber (dalam Andriani, 2013) nilai *accuracy* merupakan presentase jumlah *record* data yang diklasifikasikan secara benar oleh sebuah algoritma dapat membuat klasifikasi tersebut. *Error rate* merupakan kebalikan dari *accuracy* yaitu presentase jumlah *record* data yang tidak diklasifikasikan secara benar. Menghitung nilai *accuracy* dan *error rate* menggunakan persamaan dibawah ini.

$$Accuracy = \frac{t_pos+t_neg}{pos+neg} \times 100\% \dots\dots\dots(2.6)$$

$$Error\ rate = \frac{f_pos+f_neg}{pos+neg} \times 100\% \dots\dots\dots(2.7)$$

Keterangan :

t_pos = jumlah *true positives*

t_neg = jumlah *true negatives*

pos = jumlah *record positives*

neg = jumlah *record negatives*

f_pos = jumlah *false negatives*

f_neg = jumlah *false negatives*

2.7. *Software Development Life Cycle (SDLC)*

SDLC merupakan kepanjangan dari *Software Development Life Cycle*. *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat sebelumnya (A.S. dan Shalahuddin 2011). Terdapat banyak model SDLC yang dapat digunakan dalam pengembangan sistem, salah satunya adalah model *incremental*.

Model *incremental* merupakan penyempurnaan dari model *waterfall*. Keuntungan dari model ini adalah ketika terjadi penambahan kebutuhan fungsional akan lebih mudah diuji, diverifikasi, dan divalidasi. Dalam *incremental* pembangunan sistem dilakukan bertahap sehingga *user* dapat menggunakan terlebih dahulu bagian yang telah selesai dibangun tanpa harus menunggu sistem selesai dibangun sepenuhnya. Selain itu model *incremental* akan mengurangi beban pekerjajanya dikarenakan setiap modul akan dilakukan pengujian, verifikasi, serta validasi sehingga perubahan yang akan terjadi pada sistem tidak terlalu besar (Proboyekti 2008).

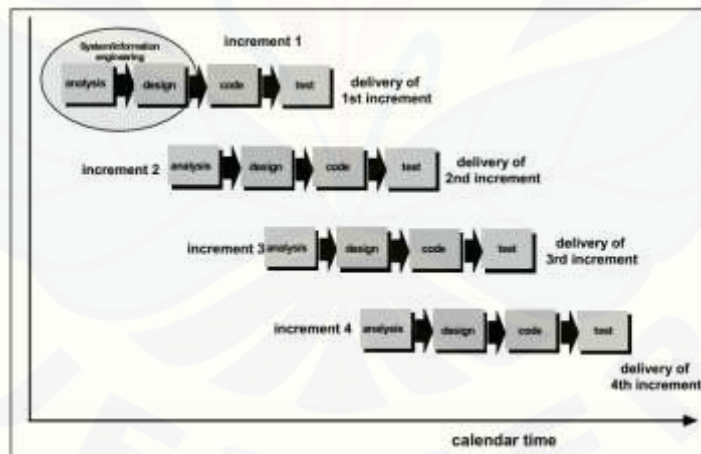
Model *Incremental* memiliki beberapa kelebihan dan kekurangan (Mega 2011) yaitu:

1. Merupakan model dengan manajemen yang sederhana
2. Pelanggan tidak perlu menunggu sampai seluruh sistem dikirim untuk mengambil keuntungan dari sistem tersebut. *Increment* yang pertama sudah memenuhi persyaratan mereka yang paling kritis, sehingga perangkat lunak dapat segera digunakan.
3. Pelanggan dapat memakai *increment* yang pertama sebagai bentuk *prototype* dan mendapatkan pengalaman yang dapat menginformasikan persyaratan untuk *increment* sistem berikutnya.

4. Resiko kegagalan proyek secara keseluruhan lebih rendah. Walaupun masalah dapat ditemukan pada beberapa *increment*, biasanya beberapa *increment* diserahkan dengan sukses kepada pelanggan.
5. Karena layanan dengan prioritas tertinggi diserahkan pertama dan *increment* berikutnya diintegrasikan dengannya, sangatlah penting bahwa layanan sistem yang paling penting mengalami pengujian yang paling ketat. Ini berarti bahwa pelanggan akan memiliki kemungkinan kecil untuk memenuhi kegagalan perangkat lunak pada *increment* sistem yang paling kecil.

Sedangkan kekurangan penggunaan *Incremental* model adalah:

1. *Increment* harus relatif lebih kecil (tidak lebih dari 20.000 baris kode) dan setiap *increment* harus menyediakan sebagian dari fungsional sistem.
2. Adanya kesulitan untuk memetakan persyaratan pelanggan pada *increment* dengan ukuran yang benar dan penjelasan model *incremental* terdapat pada Gambar 3.1



Gambar 2.2 *Incremental* Model (Original: Mills)

Model ini cocok digunakan dalam pengembangan sistem ini karena terbatasnya sumber daya manusia (SDM).

2.7.1 Analisis

Analisis merupakan tahap awal yang dilakukan dalam model *incremental*. Tahap analisis merupakan tahap pencarian dan analisa kebutuhan. Pada tahap ini penulis melakukan analisis kebutuhan berdasarkan hasil observasi dan wawancara yang dilakukan ditempat penelitian.

2.7.2 Desain Sistem

Desain sistem adalah proses menentukan bagaimana sebuah sistem dapat menyelesaikan apa yang harus diselesaikan, yang dilakukan berdasarkan hasil analisis. Desain sistem digambarkan dengan menggunakan diagram UML yaitu *Usecase Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*. Sedangkan untuk pemodelan *database* menggunakan *Entity Relationship Diagram*.

2.7.3 Coding

Coding atau penulisan kode program merupakan tahap dimana sistem dibuat. Sistem dibuat berdasarkan desain sistem yang telah dirancang sebelumnya.

2.7.4 Testing

Testing atau Pengujian sistem dilakukan dengan *black-box testing* dan *white-box testing*. *Black-box testing* merupakan pengujian program yang melihat dari segi fungsional tanpa melihat desain dan kode program. Tujuan dari *black-box testing* adalah untuk mengetahui apakah *input*, *output*, dan fitur-fitur pada program yang telah dibangun sesuai dengan kebutuhan *user*. *Black-box testing* dilakukan pihak dari tempat penelitian.

White-box testing merupakan pengujian program dengan melihat design dan kode program. Desain dan kode program di uji untuk melihat apakah dapat menghasilkan *input*, *output*, serta fitur-fitur yang diinginkan. Pada pengujian *white-box* juga dilakukan pengukuran program untuk melihat tingkat kompleksitas program dengan menggunakan metode *cyclomatic*.



BAB 3. METODOLOGI PENELITIAN

Bab ini menguraikan tentang pendekatan penelitian, tempat dan waktu penelitian, alur penelitian, dan metode yang digunakan dalam penelitian untuk menjawab rumusan masalah yang telah dijelaskan.

3.1. Pendekatan Penelitian

Pendekatan penelitian yang dipilih peneliti untuk menjawab perumusan masalah penelitian yang telah dijelaskan adalah pendekatan penelitian kualitatif. Peneliti menggunakan pendekatan penelitian kualitatif karena hasil dari penelitian ini tidak didapat dari prosedur statistik melainkan dari data-data yang dikumpulkan melalui beberapa cara yaitu wawancara dan survei atau observasi.

3.2. Tempat dan Waktu Penelitian

Penelitian ini dilakukan di klinik mata dr. Jamaluddin Sp.M. yang beralamat di jl. Bhayangkara M. Hosen 29 Bangkalan, Madura. Waktu penelitian dilaksanakan selama tujuh bulan dimulai bulan April 2014 hingga bulan Oktober 2014.

3.3. Alur Penelitian

Guna menjawab rumusan masalah serta mencapai tujuan yang telah dijelaskan peneliti menggunakan beberapa metode yaitu, metode pengumpulan data, metode yang diimplementasikan dalam membangun sistem yaitu algoritma c4.5 dalam menentukan waktu pemeriksaan pasien, dan metode pengembangan sistem yaitu dengan model *incremental*. Alur penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Penelitian
Sumber : (Hasil Analisis, 2014)

Gambar 3.1 menjelaskan beberapa tahap penelitian yaitu studi literatur, pengumpulan data, Perancangan sistem prediksi waktu pemeriksaan menggunakan algoritma c4.5 dan evaluasi algoritma, serta pengembangan sistem informasi klinik mata. Berikut penjelasan dari masing-masing tahapan.

3.3.1. Studi Literatur

Alur penelitian yang pertama adalah studi literatur, studi dilakukan untuk mendapatkan gambaran, pemahaman, dan pengetahuan tentang teori dan konsep sebagai dasar dalam melakukan penelitian ini. Studi literatur yang digunakan meliputi jurnal *online*, buku, skripsi, buku *online* yang terkait dengan penelitian.

3.3.2. Tahap Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam penelitian. Data-data yang dibutuhkan meliputi :

a. Data Primer

Data primer adalah data yang diperoleh langsung dari sumbernya dengan melakukan wawancara dan survei langsung di tempat penelitian. Data primer yang dibutuhkan pada penelitian ini seperti data pasien, data rekam medis pasien untuk menentukan atribut dalam pembuatan sistem, alur kerja pada klinik mata untuk menentukan fitur- fitur yang akan dibangun pada sistem, serta data set yang akan digunakan sebagai data *training* dan data *testing* dalam penerapan algoritma c4.5 untuk menentukan waktu pemeriksaan pasien.

Data set yang dibutuhkan dalam penelitian ini memiliki beberapa atribut dan kelas. Atribut yang digunakan terdiri usia, jenis kelamin, dan tiga belas keluhan pasien yang sering terjadi pada klinik mata tempat penelitian. Berikut merupakan atribut dan nilai dari atribut yang digunakan sebagai data set pada penelitian ini.

Tabel 3.1 Atribut dan Nilai Atribut

No	Atribut	Nilai
1	Usia	Numeris
2	Jenis Kelamin	Perempuan atau Laki-laki
3	Mata Merah	Ya atau Tidak

4	Pandangan Kabur	Ya atau Tidak
5	Mata Berair	Ya atau Tidak
6	Silau	Ya atau Tidak
7	Gatal	Ya atau Tidak
8	Nyeri	Ya atau Tidak
9	Kelopak Mata Bengkak	Ya atau Tidak
10	Pandangan Berkabut	Ya atau Tidak
11	Keluar Kotoran	Ya atau Tidak
12	Tidak Dapat Membedakan Warna	Ya atau Tidak
13	Pandangan Sempit	Ya atau Tidak
14	Kabur Lihat Jauh	Ya atau Tidak
15	Kabur Lihat Dekat	Ya atau Tidak

Sumber : (Hasil Analisis, 2014)

Kelas pada data set yang digunakan pada penelitian ini berupa kelas waktu yang merupakan lama waktu dari pemeriksaan pasien terdiri dari tiga kategori yaitu, cepat, sedang, dan lama. Penentuan kategori tersebut merupakan hasil konversi dari bentuk waktu dalam satuan menit. Jika pemeriksaan pasien kurang dari sepuluh menit maka waktu cepat, jika pemeriksaan antara sepuluh dan dua puluh menit maka waktu sedang, dan jika pemeriksaan lebih dari dua puluh menit maka waktu lama.

b. Data Sekunder

Data sekunder adalah data yang diperoleh dari sumber lain selain tempat penelitian, tetapi mempunyai keterkaitan yang sangat erat dengan tema penelitian contohnya literatur yang berhubungan dengan penelitian.

Metode yang digunakan pada tahap pengumpulan data dalam penelitian ini adalah sebagai berikut :

a. Observasi

Pengumpulan data dengan metode observasi yaitu dengan melakukan pengamatan langsung pada objek yang diteliti. Metode ini bertujuan untuk dapat mengetahui langsung bagaimana alur kerja yang terjadi pada objek yang diteliti, serta

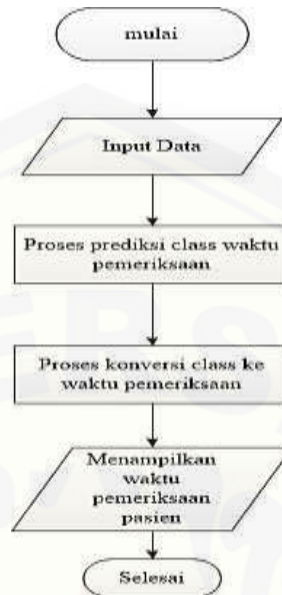
data set yang akan digunakan dalam penelitian ini. Setelah melakukan pengamatan, dilakukan pencatatan secara sistematis dari hasil pengamatan tersebut. Observasi dilakukan oleh peneliti dengan mendatangi klinik mata dr Jamaluddin Sp.M yang berada di kota Bangkalan, Madura untuk mengamati serta mencatat data pasien, keluhan pasien, serta lama penanganan pasien yang akan digunakan sebagai data set dalam sistem prediksi waktu pemeriksaan pasien menggunakan algoritma c4.5.

b. Wawancara

Pengumpulan data dengan melakukan tanya jawab langsung dengan narasumber dari objek yang diteliti untuk memperoleh data yang diinginkan. Wawancara dilakukan guna mendapatkan alur kerja pada objek yang diteliti yang akan digunakan dalam menentukan fitur-fitur yang akan dibangun.

3.3.3. Penerapan Algoritma C4.5 untuk Prediksi Waktu Pemeriksaan Pasien

Sistem informasi klinik mata memiliki beberapa fitur yang diharapkan dapat membantu pekerjaan pada klinik mata. Salah satu fitur yaitu fitur registrasi pasien yang dapat memprediksi waktu pemeriksaan pasien. Memprediksi waktu pemeriksaan tersebut menggunakan *rule* yang dihasilkan dari proses *data mining* dengan menggunakan salah satu algoritma *decision tree* yaitu algoritma c4.5. *Flowchart* penerapan algoritma c4.5 pada fitur registrasi pasien terdapat pada Gambar 3.1.

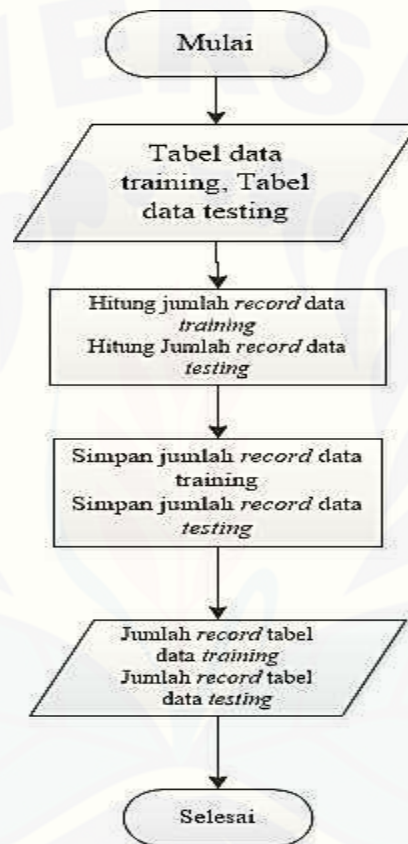


Gambar 3.2 *Flowchart* Penerapan Algoritma C4.5 pada Fitur Registrasi Pasien

Penjelasan dari Gambar 3.2 penerapan algoritma c4.5 pada fitur registrasi pasien sesuai dengan *flowchart* di atas. Terdapat beberapa tahapan, yang pertama adalah *input* data yang berupa data keluhan yang dialami pasien, usia, serta jenis kelamin pasien. Kemudian data tersebut digunakan dalam proses prediksi *class* waktu periksa. Ada tiga kategori *class* waktu periksa yaitu cepat, sedang, dan lama. Setelah ditemukan kategori *class* yang sesuai, *class* akan dikonversi kedalam waktu pemeriksaan dalam bentuk menit (cepat = 10 menit, sedang = 20, dan lama = 30), lalu ditentukan waktu pemeriksaan dalam bentuk jam.

Proses prediksi *class* didasarkan pada *rule* yang telah dikonversi sebelumnya dari hasil membangun *decision tree*. Dalam hal ini, untuk membangun *decision tree* peneliti menggunakan algoritma c4.5. Untuk melakukan proses membangun *decision tree*, peneliti membuat sistem yaitu sistem prediksi waktu pemeriksaan pasien. Proses pembangunan *decision tree* pada sistem tersebut terdapat beberapa proses utama yaitu: proses *request* data, proses pembangunan model (proses *learning*), dan pengklasifikasian data (proses *testing*).

Proses yang pertama dilakukan yaitu proses *request* data, dimana sistem mendapatkan data untuk diolah, yaitu data *training* dan data *testing* beserta informasinya. Informasi yang didapat dari data *training* yang didapatkan yaitu berupa jumlah *record* tabel data *training*, sedangkan informasi yang didapat dari data *testing* yaitu jumlah *record* tabel data *testing*. Proses tersebut dapat dilihat pada Gambar 3.3.



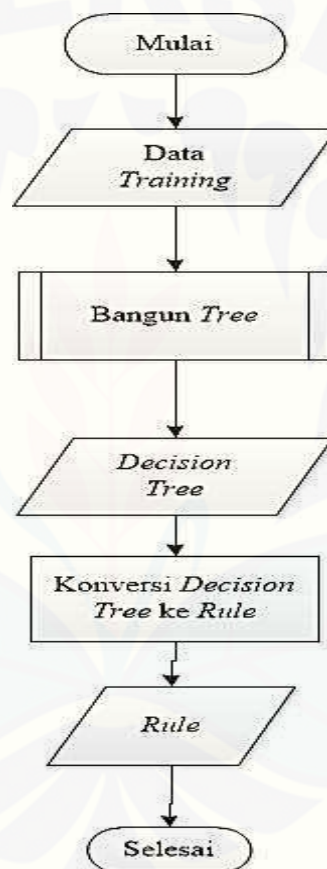
Gambar 3.3 Flowchart Proses Request Data

Penjelasan dari gambar 3.3 proses *request* data yang pertama mendapatkan masukan yaitu tabel data *training* dan data *testing* kemudian jumlah *record* data *training* dan *testing* dihitung lalu disimpan.

Setelah sistem mendapatkan data *training* dan data *testing*, proses selanjutnya yaitu proses *learning*. Adapun langkah-langkah pada proses *learning*, yaitu:

1. Sistem mendapatkan data *training*.
2. Sistem membangun *decision tree* dari data *training* dengan algoritma c4.5
3. Sistem menghasilkan *decision tree*.
4. Sistem mengkonversi *decision tree* ke dalam bentuk *rule* dan menyimpan *rule* tersebut.

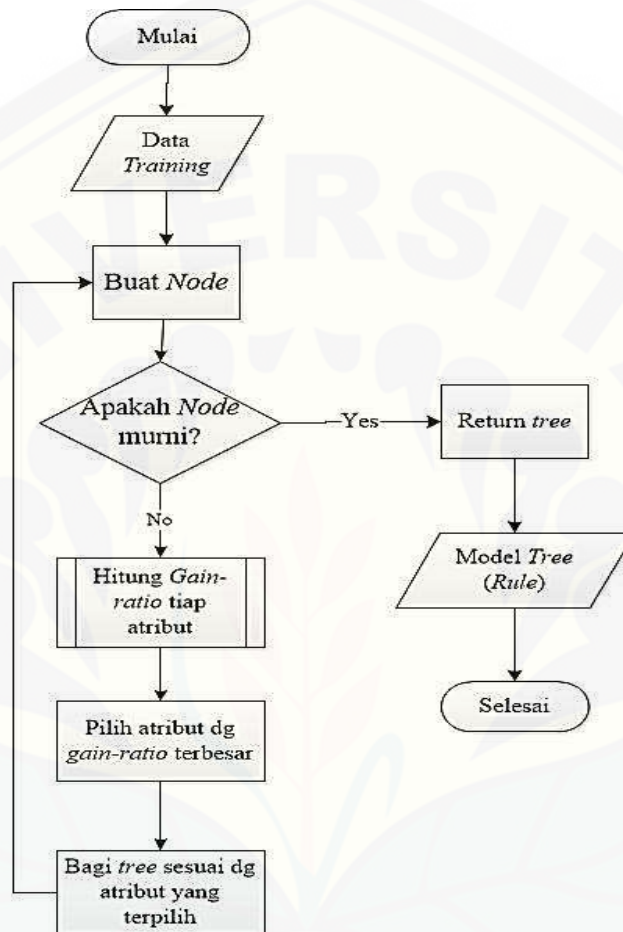
Alur proses *learning* dapat dilihat pada Gambar 3.4.



Gambar 3.4 Flowchart Proses *Learning*

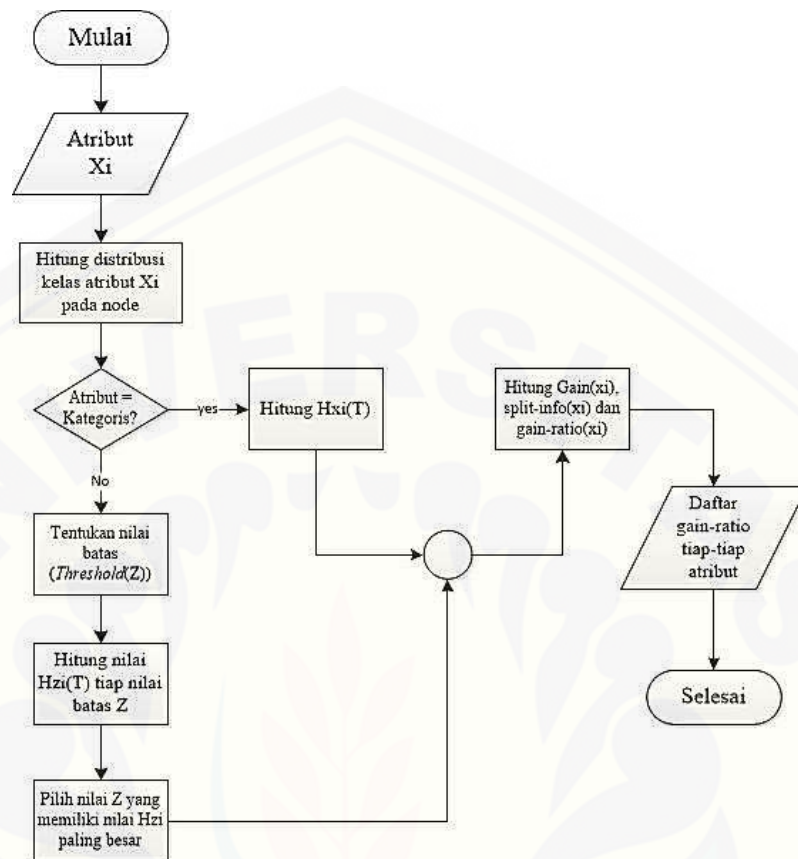
Penjelasan gambar 3.4 proses *learning* terdapat beberapa tahapan, yang pertama adalah mendapatkan data *training* kemudian proses pembangunan *tree* sehingga menghasilkan *decision tree*. Setelah didapatkan *decision tree*, *decision tree*

di konversi ke dalam bentuk *rule*. Proses pembangunan *tree* ditunjukkan pada Gambar 3.5.



Gambar 3.5 flowchart proses pembangunan *tree*

Penjelasan gambar 3.5 proses pembangunan *tree* terdapat beberapa langkah. Langkah yang pertama setelah mendapatkan data *training* yaitu membuat *node*. setelah *node* dibuat *node* tersebut diverifikasi kemurniannya. Jika *node* sudah murni maka proses pembangunan *tree* selesai. *Node* dikatakan murni apabila semua data didalamnya sudah memiliki kelas yang sama. Jika *node* belum murni, maka melanjutkan ke proses membagi *node*. Untuk membuat *node*, proses dapat dilihat pada Gambar 3.6.



Gambar 3.6 *flowchart* proses membuat *node*

Penjelasan gambar 3.6 proses membuat *node* diawali dengan mendapatkan data atribut x_i , lalu hitung distribusi kelas atribut x_i pada *node*. setelah itu, cek apakah atribut merupakan kategori, jika ya maka hitung nilai entropi menggunakan persamaan 2.2, jika tidak maka tentukan nilai batas (*Threshold*) Z kemudian hitung entropi tiap nilai batas Z . Pilih nilai batas yang memiliki nilai H_{z_i} paling tinggi untuk menjadi nilai batas atribut yang akan dibandingkan dengan atribut lainnya. Setelah mendapat nilai entropi tiap atribut, proses selanjutnya yaitu menghitung $gain(x_i)$ menggunakan persamaan 2.3, $split-info(x_i)$ menggunakan persamaan 2.4, serta $gain-ratio(x_i)$ menggunakan Persamaan 2.5.

Contoh proses pembangunan *tree* dengan menggunakan tabel data *training* yang dapat dilihat pada Tabel 3.1.

Tabel 3.2 Contoh Data *Training*

No	Usia	J.Kelamin	Mata Merah	Pandangan Kabur	Mata Berair	Silau	Gatal	Nyeri	Kelopak Mata Bengkak	Pandangan Berkabut	Keluar Kotoran (Belek)	Tidak dapat membedakan warna	Pandangan Sempit	Kabur lihat jauh	Kabur lihat dekat	Waktu
1	50	Laki-laki	Ya	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Cepat
2	49	Perempuan	Ya	Ya	Ya	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Sedang
3	47	Perempuan	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
4	60	Laki-laki	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Cepat
5	13	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Sedang
6	24	Laki-laki	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
7	60	Perempuan	Tidak	ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
8	13	Perempuan	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
9	14	Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
10	16	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
26	85	Laki-laki	Tidak	ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Lama
124	15	Perempuan	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Lama
131	30	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Lama
142	50	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
145	18	Perempuan	Tidak	Tidak	Ya	Ya	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Sedang

Contoh perhitungan iterasi 1:

- Jumlah record keseluruhan = 15
- Distribusi kelas:
 - Kelas cepat = 6
 - Kelas sedang = 6
 - Kelas lama = 3

Tabel 3.3 Probabilitas Kelas

Kelas	Jumlah	Probabilitas kelas
Cepat	6	6/15
Sedang	6	6/15
Lama	3	3/15

Perhitungan entropi kelas:

$$H(X) = -6/15 \log_2 6/15 - 6/15 \log_2 6/15 - 3/15 \log_2 3/15 = \mathbf{1,521928095}$$

- Menghitung informasi tiap atribut:
 - Distribusi kelas atribut mata merah (x1)

Tabel 3.4 tabel distribusi kelas atribut mata merah

<i>Outcomes</i>	Cepat	Sedang	Lama	Jumlah <i>records</i>
Ya	1	2	2	5
Tidak	5	4	1	10
Jumlah	6	6	3	15

Perhitungan entropi dan *information-gain* untuk atribut x1, sebagai berikut:

$$H_{x_1}(X) = 6/15(-3/6 \log_2 3/6) - 2/6 \log_2 2/6 - 1/6 \log_2 1/6 + 9/15 (-3/9 \log_2 3/9) - 4/9 \log_2 4/9 - 2/9 \log_2 2/9 = 1,501955001$$

$$Gain(x_1) = 1,521928095 - 1,501955001 = 0,019973094$$

Sedangkan perhitungan *split-info* dan *gain-ratio* untuk atribut x1 sebagai berikut:

$$Split-info(x_1) = -6/15*(\log_2 6/15) - 9/15*(\log_2 9/15) = 0,970950594$$

$$Gain-ratio(x_1) = 0,970950594/0,019973094 = 0,020570659$$

Lakukan perhitungan seperti pada perhitungan atribut mata merah pada setiap atribut yang memiliki tipe kategori. Sedangkan atribut yang memiliki tipe numeris, memiliki langkah yang berbeda. Langkah pertama yang dilakukan yaitu mencari nilai batas, dan mencari informasi gain dari tiap-tiap nilai batas.

Contoh : Perhitungan atribut numeris (umur).

Unique value = 13, 14, 15, 16, 18, 24, 30, 47, 49, 50, 60, 85

Nilai batas = 13, 14, 15, 16, 18, 24, 30, 47, 49, 50, 60

- Distribusi kelas atribut umur

Tabel 3.5 Tabel Distribusi Kelas Atribut Umur

Nilai batas	Cepat	Sedang	Lama
<=13	1	1	0
>13	5	5	3
<=14	2	1	0
>14	4	5	3
<=15	2	1	1
>15	4	5	2
<=16	2	2	1
>16	4	4	2
<=18	2	3	1
>18	4	3	2
<=24	3	3	1
>24	3	3	2
<=30	3	3	2
>30	3	3	1
<=47	4	3	2
>47	2	3	1
<=49	4	4	2
>49	2	2	1
<=50	5	5	2
>50	1	1	1
<=60	6	6	2
>60	0	0	1

Contoh perhitungan *information-gain* nilai batas 13 :

$$H_{y \leq 13 \text{ or } y > 13}(X) = 2/15(-1/2 \log_2 1/2 - 1/2 \log_2 1/2) + 13/15 (-5/13 \log_2 5/13 - 5/13 \log_2 5/13 - 3/13 \log_2 3/13) = 1,475436526$$

$$\text{Gain}_{y \leq 13 \text{ or } y > 13} = 1,521928095 - 1,475436526 = 0,046491569$$

Perhitungan *gain* nilai batas lainnya dihitung sama seperti perhitungan *gain* nilai batas 13. Tabel 3.6 menyajikan hasil perhitungan *information-gain* tiap nilai batas.

Tabel 3.6 Tabel hasil perhitungan *information-gain* tiap nilai batas

batas	Hx(X)	Gain
y≤13 or y>13	1,475436526	0,046491569
y≤14 or y>14	1,427327302	0,094600793
y≤15 or y>15	1,496273822	0,025654273
y≤16 or y>16	1,521928095	0
y≤18 or y>18	1,501955001	0,019973094
y≤24 or y>24	1,50879563	0,013132465
y≤30 or y>30	1,50879563	0,013132465
y≤47 or y>47	1,618947501	-0,097019406
y≤49 or y>49	1,521928095	0
y≤50 or y>50	1,503677104	0,018250991
y≤60 or y>60	1,352227927	0,169700168

Dari perhitungan di atas, didapat nilai batas dengan nilai *information-gain* tertinggi **0,169700168** yaitu nilai batas 13. Maka, *information-gain* dari nilai batas 13 digunakan untuk dibandingkan dengan atribut lain untuk mencari *node*.

Hasil perhitungan *information-gain* dan *gain-ratio* dapat dilihat pada Tabel 3.7.

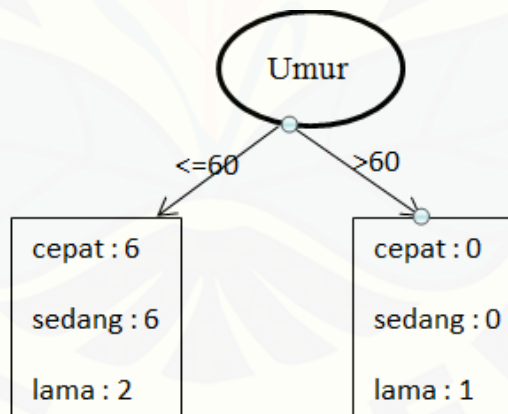
Tabel 3.7 Hasil Perhitungan *Information-gain* dan *Gain-ratio* Iterasi 1

No	Class	Hs(x)	Gain(x)	Split-info(x)	Gain-ratio(x)
1	Mata Merah	1,501955001	0,019973094	0,970950594	0,020570659
2	Pandangan Kabur	1,41461873	0,107309365	0,918295834	0,116857075
3	Mata Berair	1,370950594	0,1509775	0,918295834	0,164410525

4	Silau	1,427327302	0,094600793	0,721928095	0,131039079
5	Gatal	1,428577729	0,093350366	0,353359335	0,264179709
6	Nyeri	1,398118963	0,123809132	0,996791632	0,124207635
7	Kelopak Mata Bengkak	1,475436526	0,046491569	0,566509507	0,082066706
8	Pandangan Berkabut	1,428577729	0,093350366	0,353359335	0,264179709
9	Keluar Kotoran	1,503677104	0,018250991	0,721928095	0,025280898
10	tdk dpt mmbeda warna	1,217542476	0,304385619	0	0
11	pandangan sempit	1,217542476	0,284412525	0	0
12	kabur lihat jauh	1,41992402	0,102004075	0,836640742	0,121920999
13	kabur lihat dekat	1,475436526	0,046491569	0,566509507	0,082066706
14	Umur	1,352227927	0,169700168	0,353359335	0,480248153
15	Jenis Kelamin	1,36196397	0,159964125	0,970950594	0,164750015

Setelah didapatkan nilai *gain-ratio* tiap-tiap atribut, pilih atribut dengan nilai paling besar. Dari perhitungan diatas, didapat *gain-ratio* terbesar pada atribut umur yaitu **0,480248153**. Maka atribut umur yang di jadikan *node*.

Kemudian bagi node berdasarkan atribut yang terpilih. *Decision tree* yang terbentuk pada iterasi 1 ditunjukkan pada Gambar 3.7.



Gambar 3.7 *Decision tree* iterasi 1

Langkah selanjutnya adalah mengecek apakah *node* sudah murni atau belum. Jika *node* sudah murni, maka proses berhenti dan *decision tree* terbentuk. Jika belum,

proses membagi *node* dilakukan kembali secara berulang hingga semua *node* yang terbentuk menjadi murni dan sudah tidak ada lagi *node* yang dapat dibagi.

Untuk iterasi dua, hasil pengecekan $node > 60$ menunjukkan *node* sudah murni (*node* memiliki satu kelas yaitu, lama). Untuk $node \leq 60$, *node* belum murni, karena didalamnya masih terdapat beberapa kelas yang berbeda. Maka, $node \leq 60$ di-split. Untuk mencari atribut yang menjadi *node*, dilakukan kembali menghitung informasi tiap-tiap atribut yang belum menjadi *node*.

Contoh perhitungan iterasi 2 yaitu untuk pembagian *node* umur ≤ 60 :

- Jumlah *record* keseluruhan = 14
- Distribusi kelas:
 - Kelas cepat = 6
 - Kelas sedang = 6
 - Kelas lama = 2

Tabel 3.8 Probabilitas kelas iterasi 2

<i>Class</i>	jml	probabilitas
Cepat	6	6/14
Sedang	6	6/14
Lama	2	2/14

Perhitungan entropi kelas iterasi 2:

$$H(X) = -6/14 \log_2 6/14 - 6/14 \log_2 6/14 - 2/14 \log_2 2/14 = 1,448815636$$

- Menghitung informasi tiap atribut
 - Distribusi kelas atribut mata merah (x1)

Tabel 3.9 Distribusi kelas atribut mata merah iterasi 2

<i>Outcomes</i>	Cepat	Sedang	Lama	Jumlah records
Ya	3	2	1	6
Tidak	3	4	1	8
Jumlah	6	6	2	14

Perhitungan entropi dan *information-gain* untuk atribut x1, sebagai berikut:

$$H_{x_2}(X) = 5/14(-1/4 \log_2 1/4) - 2/4 \log_2 2/4 - 1/4 \log_2 1/4 + 10/14 (-5/10 \log_2 5/10) - 4/10 \log_2 4/10 - 1/10 \log_2 1/10 = 1,335880794$$

$$Gain(x_1) = 1,448815636 - 1,335880794 = 0,112934842$$

$$Split-info(x_1) = -6/14*(\log_2 6/14) - 8/14*(\log_2 8/14) = 0,985228136$$

$$Gain-ratio(x_1) = 0,985228136 / 0,112934842 = 0,020547736$$

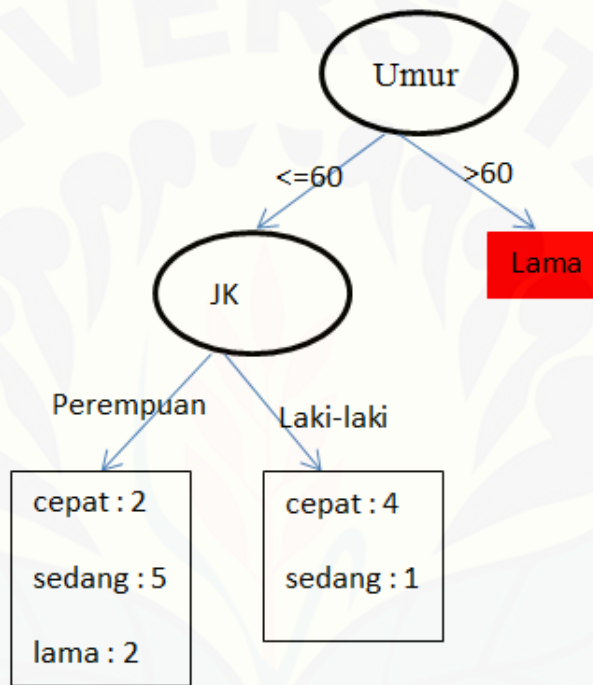
Ulangi perhitungan tersebut pada masing-masing atribut yang belum menjadi *node*. Hasil perhitungan *information-gain* dan *gain-ratio* tiap atribut pada iterasi kedua dapat dilihat pada tabel 3.10.

Tabel 3.10 Hasil perhitungan *information-gain* dan *gain-ratio* tiap atribut pada iterasi 2

NO	Class	Hs(x)	Gain(x)	Split-info(x)	Gain-ratio(x)
1	Mata Merah	1,428571429	0,020244207	0,985228136	0,020547736
2	Pandangan Kabur	1,400688605	0,04812703	0,863120569	0,055759337
3	Mata Berair	1,265063904	0,183751732	0,940285959	0,195421117
4	Silau	1,371356774	0,077458862	0,749595257	0,103334248
5	Gatal	1,356164347	0,092651289	0,371232327	0,249577642
6	Nyeri	1,345005357	0,103810279	0,985228136	0,105366742
7	Kelopak Mata Bengkak	1,414304933	0,034510703	0,591672779	0,058327346
8	Pandangan Berkabut	1,356164347	0,092651289	0,371232327	0,249577642
9	Keluar Kotoran	1,444072055	0,004743581	0,749595257	0,006328189
10	tdk dpt bedakan warna	1,448815636	0	0	0
11	pandangan sempit	1,448815636	0	0	0
12	kabur lihat jauh	1,372805782	0,076009854	0,863120569	0,088064005
13	kabur lihat dekat	1,414304933	0,034510703	0,591672779	0,058327346
14	Jenis Kelamin	1,180666072	0,268149563	0,940285959	0,285178738

Setelah didapatkan nilai *gain-ratio* tiap-tiap atribut, pilih atribut dengan nilai paling besar. Dari perhitungan diatas, didapat *gain-ratio* terbesar pada atribut jenis kelamin yaitu **0,285178738**. Maka atribut jenis kelamin yang di jadikan *node*.

Kemudian bagi *node* berdasarkan atribut yang terpilih. *Decision tree* yang terbentuk pada iterasi 2 ditunjukkan pada Gambar 3.8.



Gambar 3.8 *Decision Tree* iterasi 2

Seperti pada iterasi 1, *node* dicek apakah *node* sudah murni atau belum. Jika *node* sudah murni maka *decision tree* terbentuk. Jika *node* belum murni maka proses diulangi sampai *node* murni dan tidak ada lagi *node* yang dapat dibagi. *Decision tree* yang terbentuk dari beberapa iterasi setelahnya dapat dilihat pada Gambar 3.9.

Gambar 3.9 *Decision tree*(hasil akhir)

Setelah model *tree* dibangun, kemudian dikonversi ke dalam bentuk *rule*. Pembentukan *rule* ini menggunakan aturan IF-THEN. Maka, *rule* yang terbentuk adalah sebagai berikut.


```

IF Umur > 60
  THEN WAKTU = Lama

ELSE IF Umur <= 60 AND Jenis_Kelamin = 'laki-laki' AND Kelopak_mata_bengkak = 'Tidak'
  THEN WAKTU = Cepat

ELSE IF Umur <= 60 AND Jenis_Kelamin = 'laki-laki' AND Kelopak_mata_bengkak = 'Ya'
  THEN WAKTU = Sedang

ELSE IF Umur <= 60 AND Jenis_Kelamin = 'Perempuan' AND Pandangan_berkabut = 'Ya'
  THEN WAKTU = Sedang

ELSE IF Umur <= 60 AND Jenis_Kelamin = 'Perempuan' AND Pandangan_berkabut = 'Tidak'
AND Kelopak_mata_bengkak = 'Ya'
  THEN WAKTU = Cepat

ELSE IF Umur <= 60 AND Jenis_Kelamin = 'Perempuan' AND Pandangan_berkabut = 'Tidak'
AND Kelopak_mata_bengkak = 'Tidak' AND Nyeri = 'Tidak' AND Pandangan_sempit = 'Tidak'
  THEN WAKTU = Lama

ELSE IF Umur <= 60 AND Jenis_Kelamin = 'Perempuan' AND Pandangan_berkabut = 'Tidak'
AND Kelopak_mata_bengkak = 'Tidak' AND Nyeri = 'Tidak' AND Pandangan_sempit = 'Ya'
  THEN WAKTU = Sedang

ELSE IF Umur <= 60 AND Jenis_Kelamin = 'Perempuan' AND Pandangan_berkabut = 'Tidak'
AND Kelopak_mata_bengkak = 'Tidak' AND Nyeri = 'Ya' AND Mata_berair = 'Ya' AND Silau =
'Ya'
  THEN WAKTU = Sedang

ELSE IF Umur <= 60 AND Jenis_Kelamin = 'Perempuan' AND Pandangan_berkabut = 'Tidak'
AND Kelopak_mata_bengkak = 'Tidak' AND Nyeri = 'Ya' AND Mata_berair = 'Ya' AND Silau =
'Tidak'
  THEN WAKTU = Cepat

ELSE IF Umur <= 60 AND Jenis_Kelamin = 'Perempuan' AND Pandangan_berkabut = 'Tidak'
AND Kelopak_mata_bengkak = 'Tidak' AND Nyeri = 'Ya' AND Mata_berair = 'Tidak'
  THEN WAKTU = Sedang Or Waktu = Lama

```

Rule yang didapat digunakan untuk memprediksi jika ada data baru yang belum diketahui kelasnya. Dalam sistem informasi klinik mata, *rule* digunakan pada fitur registrasi pasien untuk menebak atau memprediksi lama penanganan pasien yang kemudian akan di *convert* menjadi waktu pemeriksaan pasien.

3.3.4. Tahap Pengembangan Sistem

Pengembangan sistem registrasi pasien klinik mata ini menggunakan model *incremental* yang akan di bagi menjadi empat tahap *increment* dimana di setiap *increment*-nya dibagi beberapa fitur yang akan diselesaikan. Pada setiap *increment* terdapat beberapa tahapan, yaitu tahap analisis, desain sistem, *coding*, dan *testing*.

1. Analisis

Analisis merupakan tahap awal yang dilakukan dalam model *incremental*. Tahap analisis merupakan tahap pencarian dan analisa kebutuhan. Pada tahap ini penulis melakukan analisis kebutuhan berdasarkan hasil observasi dan wawancara yang dilakukan ditempat penelitian.

2. Desain Sistem

Desain atau perancangan dalam membangun sistem informasi klinik mata menggunakan *Unified Modeling Language* (UML) yang dirancang menggunakan konsep *Object-Oriented Programming* (OOP). Pemodelan UML yang digunakan antara lain :

- a. *Business Process*
- b. *Use Case Diagram*
- c. *Use Case Scenario*
- d. *Sequence Diagram*
- e. *Activity Diagram*
- f. *Class Diagram*

Pemodelan basis data menggunakan *Entity Relationship Diagram* (ERD).

3. Coding

Coding atau penulisan kode program dilakukan menggunakan bahasa pemrograman *Hyper Text Pre-Processor* (PHP), *Hyper Text Markup Language* (HTML), *Cascading Style Sheet* (CSS), dan *Javascript*. DBMS (*Database Management System*) yang digunakan adalah *MySQL* dengan *tool* XAMPP.

4. *Testing*

Testing dilakukan *black-box testing* dan *whitebox testing*. *Black-box testing* dilakukan oleh *user*, *user* yang melakukan pengujian adalah karyawan dari tempat penelitian. Sedangkan *whitebox testing* dilakukan langsung oleh penulis.



BAB 4. ANALISIS DAN PENGEMBANGAN SISTEM

Bab ini membahas tentang analisis dan pengembangan untuk membangun Sistem Informasi Klinik Mata Dengan Prediksi Waktu Pemeriksaan Menggunakan Algoritma C4.5 Pada Fitur Registrasi Pasien. Tahapan yang akan dilakukan dalam perancangan sistem ini menggunakan model *incremental* yaitu analisis, desain sistem, *coding*, dan *testing* yang dilakukan per modul.

4.1. Pengumpulan Data

Pengumpulan data dilakukan untuk mendapatkan data dari objek penelitian secara lengkap dan sesuai dengan dibutuhkan dalam proses penelitian. Cara yang dilakukan untuk mendapatkan data tersebut yaitu dengan observasi dan wawancara. Data-data yang dapat dikumpulkan adalah sebagai berikut:

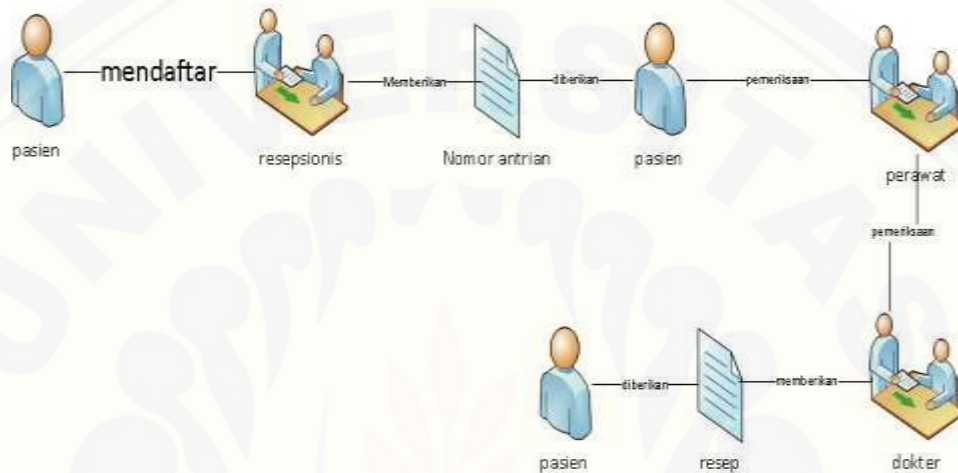
4.1.1 Data Set

Data set merupakan kumpulan atribut beserta kelasnya, yang akan dibagi menjadi data *training* dan data *testing*. Data *training* digunakan untuk proses *learning* dalam *decision tree*, sedangkan data *testing* digunakan dalam proses pengujian. Dalam pembuatan sistem prediksi waktu pemeriksaan yang diterapkan pada fitur registrasi pasien menggunakan seluruh data set sebagai data training dan data testing, sedangkan atribut yang digunakan meliputi jenis_kelamin, umur, mata_merah, pandangan_kabur, mata_berair, silau, gatal, nyeri, kelopak_mata_bengkak, pandangan_berkabut, keluar_kotoran, tidak_dapat_membedakan_warna, pandangan_sempit, kabur_lihat_jauh, dan kabur_lihat_dekat. Sedangkan kelas yang digunakan adalah waktu. Data set dapat dilihat pada lampiran F.

4.1.2 *Workflow* atau Alur Kerja pada Objek Penelitian

A. *Workflow* Pemeriksaan

Alur kerja pemeriksaan secara manual yang dapat dilihat pada Gambar 4.1.



Gambar 4.1 *Workflow* Pemeriksaan
Sumber: (Hasil Analisis, 2014)

Pada gambar 4.1 digambarkan alur pemeriksaan pasien yang di mulai dari registrasi pasien, pemeriksaan pasien hingga pasien mendapatkan resep yang nantinya akan diberikan kepada bagian obat untuk mendapatkan obat.

B. *Workflow* Bagian Obat

Workflow bagian obat menggambarkan proses penjualan dan pembelian obat yang terjadi di bagian obat.

B.1 Penjualan Obat

Alur kerja penjualan obat pada bagian obat secara manual yang dapat dilihat pada Gambar 4.2.



Gambar 4.2 *Workflow* Penjualan Obat
Sumber: (Hasil Analisis, 2014)

Pada gambar 4.2 digambarkan alur penjualan obat di mulai dengan pasien menyerahkan resep kepada yang bertugas di bagian obat hingga pasien mendapatkan obat.

B.2 Pembelian Obat

Alur kerja pembelian obat pada bagian obat secara manual dapat dilihat pada Gambar 4.3.



Gambar 4.3 *Workflow* Pembelian Obat
Sumber: (Hasil Analisis, 2014)

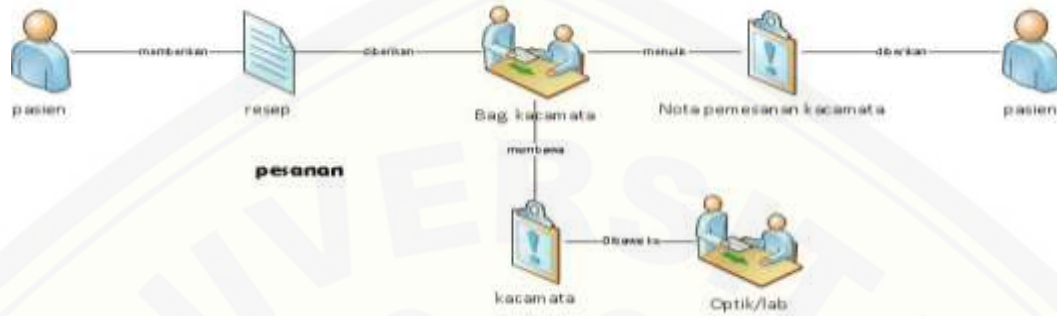
Pada gambar 4.3 digambarkan alur pembelian obat di mulai dari melihat kartu stok kemudian membuat daftar obat habis hingga bagian obat memperbaharui kartu stok obat.

C. *Workflow* Bag. Kacamata

Workflow Bagian Kacamata menggambarkan proses penjualan dan pengambilan kacamata yang terjadi pada bagian kacamata.

C.1 Penjualan kacamata

Alur kerja penjualan kacamata pada bagian kacamata secara manual yang dapat dilihat pada Gambar 4.4.



Gambar 4.4 *Workflow* Penjualan Kacamata
Sumber: (Hasil Analisis, 2014)

Pada gambar 4.4 digambarkan alur penjualan kacamata di mulai dari pasien menyerahkan resep kepada yang bertugas di bagian kacamata hingga petugas memesan kacamata ke optik atau laboratorium.

C.2 Pengambilan Kacamata

Alur kerja pengambilan kacamata pada bagian kacamata secara manual yang dapat dilihat pada Gambar 4.5.



Gambar 4.5 *Workflow* Pengambilan Kacamata
Sumber: (Hasil Analisis, 2014)

Pada gambar 4.3 digambarkan alur pengambilan kacamata di mulai dari pasien menyerahkan nota pemesanan kepada petugas hingga kacamata diberikan kepada pasien.

4.2. Algoritma C4.5 untuk Prediksi Waktu Pemeriksaan Pasien

Bagian ini menguraikan tentang bagaimana algoritma c4.5 digunakan untuk memprediksi waktu pemeriksaan pasien yang akan digunakan dalam registrasi pasien. Seperti yang telah dijelaskan pada subbab 3.4, proses membangun pohon keputusan atau *decision tree* dengan menggunakan algoritma c4.5 dilakukan sebelum proses prediksi untuk mendapatkan *rule* yang nantinya akan digunakan pada saat melakukan prediksi. Dalam hal ini, fitur registrasi pada sistem informasi klinik mata hanya menggunakan *rule* dari hasil proses pembangunan *tree* dan sistem informasi klinik mata tidak memiliki fitur untuk melakukan proses pembangunan *tree*. Maka dari itu, untuk mendapatkan *rule*, penulis membuat sistem prediksi waktu pemeriksaan pasien untuk melakukan proses pembangunan *tree*. Sistem tersebut memiliki tiga proses utama yaitu : *request* data, proses *learning*, dan pengklasifikasian atau pengujian. Pada subbab ini, akan membahas proses *request* data dan proses *learning* yang diterapkan pada sistem.

4.2.1 Request Data

Proses *request* data merupakan proses yang pertama dilakukan sistem dimana sistem mendapatkan data untuk diolah, berupa data *training* dan data *testing*. Seperti yang dijelaskan pada subbab 4.1.2, jumlah data *training* yang digunakan sama dengan jumlah data set yaitu 164 data. Data set dalam kasus ini memiliki lima belas atribut dan satu kelas bernama waktu, satu atribut bertipe numeris yaitu atribut umur, sedangkan yang lainnya bertipe kategori. Kelas waktu bertipe kategori, terdapat tiga kategori yaitu cepat, lama, dan sedang. ketiga kategori tersebut didapat dari proses merubah data dalam bentuk numeris menjadi kategori. Data dalam bentuk numeris dalam kasus ini berupa waktu dengan satuan menit. Waktu 0 sampai 10 menit menjadi kategori cepat, 11 sampai 20 menit menjadi kategori sedang, dan diatas 21 menit menjadi kategori lama. Untuk melihat data *training* dan data *testing*, sistem yang dibuat menyediakan menu data *training* dan data *testing*.

4.2.2 Proses *Learning*

Setelah sistem mendapatkan data untuk diolah (data *training*), proses selanjutnya yaitu proses *learning*. Adapun langkah-langkah pada proses *learning*, yaitu :

1. Sistem mendapatkan data *training*
2. Sistem membangun *decision tree* (model *tree*) dari data *training* dengan menggunakan algoritma c4.5 yang merupakan proses utama dari proses *learning*.
3. Sistem menghasilkan *decision tree*.
4. Sistem mengkonversi *decision tree* ke dalam bentuk *rule* dan menyimpan *rule* tersebut.

Pada proses membangun *tree* menggunakan algoritma c4.5, sistem melakukan pembangunan *tree* sesuai langkah-langkah dan perhitungan dari algoritma c4.5 yang telah dijelaskan pada subbab 2.7.

Untuk melakukan proses pembangunan *tree* tersebut, sistem yang dibuat menyediakan menu proses *mining*. Hasil *tree* dan hasil konversi *tree* menjadi *rule* ditampilkan pada menu pohon keputusan.

4.2.3 Pengklasifikasian (Proses *Testing*)

Proses *testing* yaitu proses pengklasifikasian data *testing* yang terbentuk. Hasil dari proses *testing* yaitu kelas baru untuk tiap data *testing* yang kemudian akan dibandingkan dengan kelas lama dari data *testing*, untuk mendapatkan akurasi *rule* yang terbentuk. Proses pengklasifikasian akan dijelaskan pada Bab 5.

4.3. Kebutuhan Sistem

Data-data yang didapat dari tahap pengumpulan data kemudian di analisis sehingga peneliti mendapatkan kebutuhan-kebutuhan apa saja yang akan dibangun pada sistem informasi klinik mata. Kebutuhan sistem menjelaskan hal-hal apa saja yang mampu dilakukan oleh sistem. Kebutuhan sistem terdiri dari kebutuhan fungsional dan nonfungsional.

4.3.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan jenis kebutuhan yang menggambarkan proses-proses atau fitur-fitur apa saja yang diberikan oleh sistem. Kebutuhan fungsional dari sistem informasi klinik mata ini antara lain:

1. Sistem mampu mengelola data *user*.
2. Sistem mampu mengelola data pasien.
3. Sistem mampu melakukan registrasi pasien baru
4. Sistem mampu melakukan registrasi pasien lama
5. Sistem mampu mencetak kartu antrian pasien
6. Sistem mampu memprediksi waktu pemeriksaan pasien
7. Sistem mampu mengelola data obat.
8. Sistem mampu mengelola data supplier obat.
9. Sistem mampu menampilkan daftar obat habis.
10. Sistem mampu mencetak daftar obat habis.
11. Sistem mampu melakukan transaksi pembelian obat.
12. Sistem mampu melakukan transaksi penjualan obat.
13. Sistem mampu mencetak nota pembayaran.
14. Sistem mampu mengelola data frame.
15. Sistem mampu melakukan transaksi penjualan kacamata.
16. Sistem mampu melakukan transaksi pelunasan kacamata.

17. Sistem mampu meng-*input* data pemeriksaan pasien.
18. Sistem mampu menampilkan data rekam medis pasien
19. Sistem mampu melakukan pendaftaran operasi pasien.
20. Sistem mampu menampilkan jadwal operasi pasien

4.3.2 Kebutuhan Non-fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang tidak berkaitan dengan fungsi sistem tetapi berkaitan dengan layanan yang diberikan sistem. Kebutuhan fungsional dari sistem informasi klinik mata ini antara lain:

1. *Security*
Sistem memiliki keamanan sistem yang dapat digunakan untuk mengenali *user* yang mempunyai hak untuk mengakses sistem.
2. *Usability*
User dapat menggunakan sistem dengan mudah karena sistem yang dibuat *user friendly*.

4.4. Desain Sistem

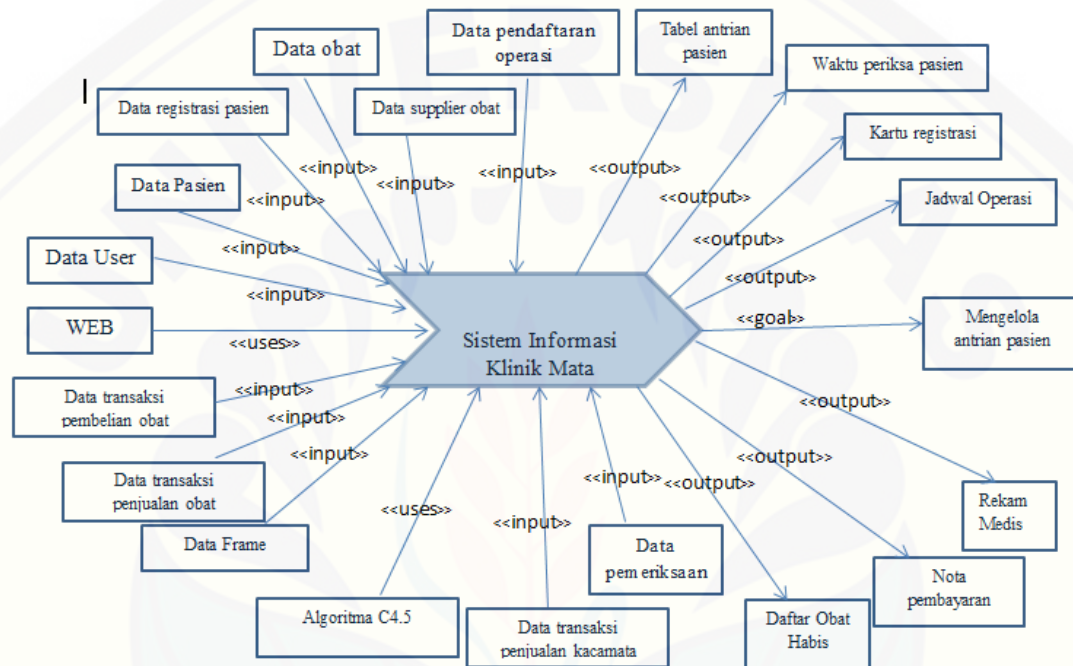
Desain perancangan sistem yang akan dibuat menggunakan *Unified Modeling Language* (UML) yang dirancang menggunakan konsep *Object-Oriented Programming* (OOP). Pemodelan UML yang digunakan antara lain :

- a. *Business Process*
- b. *Use Case Diagram*
- c. *Use Case Scenario*
- d. *Sequence Diagram*
- e. *Activity Diagram*
- f. *Class Diagram*

Pemodelan basis data menggunakan *Entity Relationship Diagram* (ERD).

4.4.1 Business Process

Gambar 4.4 merupakan business process dari sistem informasi klinik mata, didalamnya terdapat *input*, *output*, dan tujuan(*goal*) dari sistem informasi tersebut.



Gambar 4.6 *Business Process* Sistem Informasi Klinik Mata
Sumber : (Hasil Analisis, 2014)

Gambar 4.6 menggambarkan apa saja input yang diterima oleh sistem, output yang dikeluarkan sistem, serta tujuan dari sistem.

4.4.2 Pembagian Fitur

Empat tahap *increment* yang dibangun terdapat beberapa fitur yang akan diselesaikan. Pembagian fitur dapat dilihat pada tabel 4.1.

Tabel 4.1 Pembagian Fitur

No	Increment	Fitur
1	<i>Increment 1</i>	Data user
		Data pasien
		Buat Akun
		Aktivasi Akun Lewat <i>Email</i>
		Registrasi pasien lama
		Registrasi pasien baru
2	<i>Increment 2</i>	Data Obat
		Data Supplier
		Daftar obat habis
		Transaksi penjualan obat
		Transaksi pembelian obat
3	<i>Increment 3</i>	Data frame
		Transaksi penjualan kacamata
		Transaksi pelunasan kacamata
4	<i>Increment 4</i>	Input data pemeriksaan
		Lihat data rekam medis
		Pendaftaran operasi
		Lihat jadwal operasi

Sumber : (Hasil Analisis, 2014)

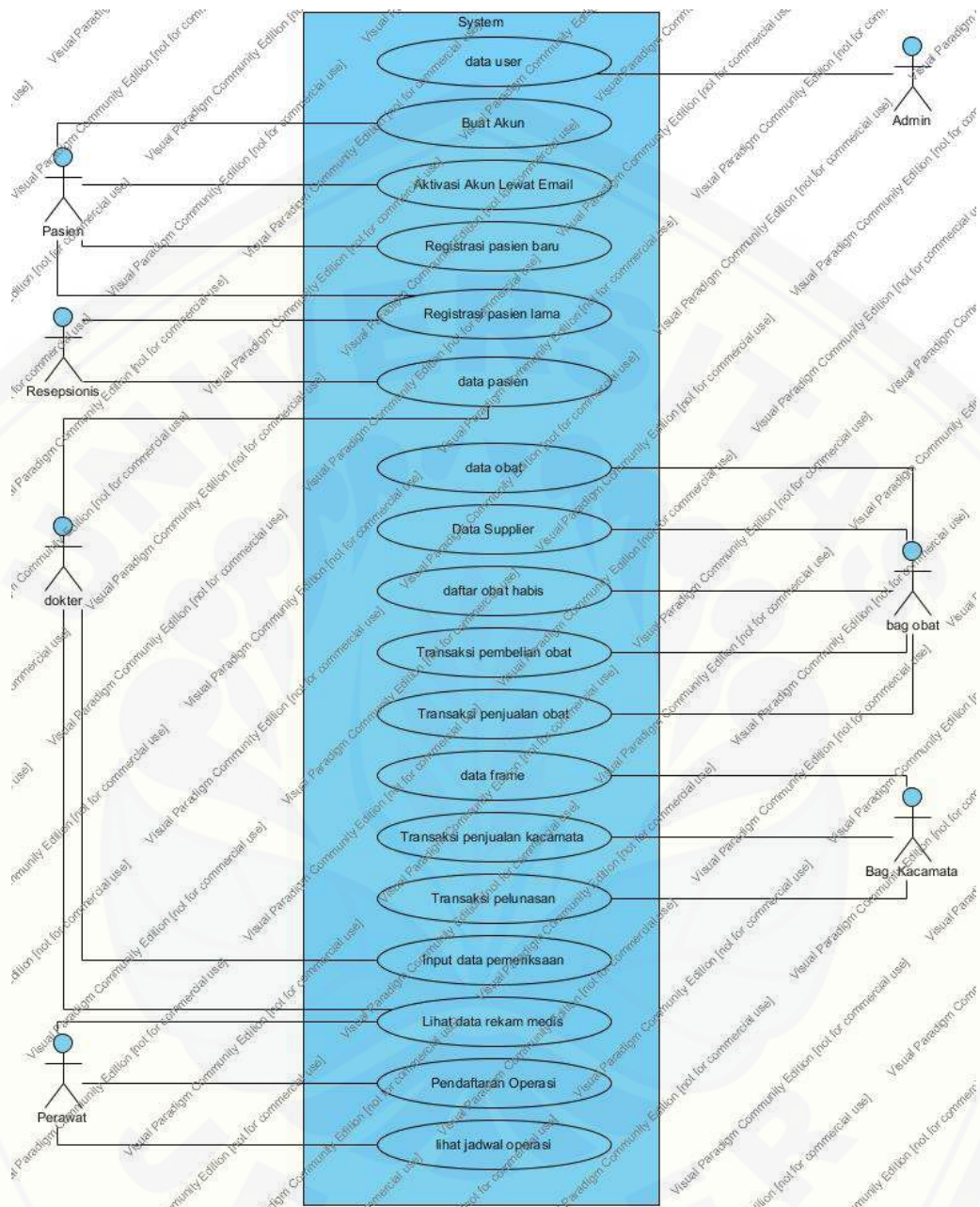
4.4.3 Usecase Diagram

Usecase diagram menggambarkan sebuah interaksi antara satu atau lebih aktor dengan fitur-fitur sistem informasi yang akan dibuat. Gunanya untuk mengetahui fitur apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fitur – fitur tersebut. Pada sub bab ini berisi *usecase diagram*, definisi aktor, dan definisi *usecase*.

1. *Usecase Diagram*

Usecase diagram digunakan untuk mendefinisikan fitur-fitur yang terdapat pada sistem informasi klinik mata, serta menggambarkan fitur-fitur yang dapat diakses oleh *actor*. *Usecase diagram* sistem informasi klinik mata dapat dilihat pada Gambar 4.5.





Gambar 4.7 Usecase Diagram SIKM
 Sumber : (Hasil Analisis, 2014)

Gambar 4.7 merupakan *usecase diagram* Sistem Informasi Klinik Mata yang menggambarkan beberapa fitur seperti data pasien, registrasi pasien, data obat, dan

lain sebagainya. Selain itu, pada gambar tersebut juga di gambarkan siapa saja yang berhak mengakses fitur-fitur tersebut.

2. Definisi Aktor

Definisi aktor digunakan untuk mendeskripsikan aktor atau *user* yang memiliki hak akses terhadap sistem informasi klinik mata. Definisi aktor dapat dilihat pada tabel 4.1.

Tabel 4.2 Definisi Aktor dan Hak Akses

No	Aktor	Hak Akses
1	Admin	- Data user
2	Resepsionis	- Registrasi pasien lama - Data pasien
3	Pasien	- Buat Akun - Aktivasi Akun Lewat <i>Email</i> - Registrasi Pasien Baru - Registrasi Pasien Lama
4	Dokter	- Data pasien - Input data pemeriksaan - Lihat data rekam medis
5	Perawat	- Lihat data rekam medis - Pendaftaran operasi - Lihat jadwal operasi
6	Bag. Obat	- Data obat - Data supplier - Daftar obat habis - Transaksi pembelian obat - Transaksi penjualan obat
7	Bag. Kacamata	- Data frame - Transaksi penjualan kacamata - Transaksi pelunasan kacamata

3. Definisi *Usecase*

Definisi *usecase* menjelaskan definisi dari setiap *usecase* sistem informasi klinik mata. Definisi usecase dapat dilihat pada tabel 4.2.

Tabel 4.3 Definisi *Usecase*

No	Usecase	Deskripsi
1	Data user	Merupakan <i>usecase</i> data user menggambarkan proses pengelolaan data user
2	Buat Akun	Merupakan <i>usecase</i> buat akun menggambarkan proses membuat akun pada sistem
3	Aktivasi Akun Lewat <i>Email</i>	Merupakan <i>usecase</i> aktivasi akun lewat <i>email</i> menggambarkan proses aktivasi akun lewat <i>email</i>
4	Registrasi pasien lama	Merupakan <i>usecase</i> registrasi pasien menggambarkan proses registrasi pasien yang telah memiliki id pasien sebelumnya pada sistem
5	Registrasi pasien baru	Merupakan <i>usecase</i> registrasi pasien via sms gateway menggambarkan proses registrasi pasien bagi pasien yang belum memiliki id pasien
6	Data pasien	Merupakan <i>usecase</i> data pasien menggambarkan proses pengelolaan data pasien
7	Data obat	Merupakan <i>usecase</i> data obat menggambarkan proses pengelolaan data obat
8	Data supplier	Merupakan <i>usecase</i> data supplier menggambarkan proses pengelolaan data supplier
9	Daftar obat habis	Merupakan <i>usecase</i> daftar obat habis menggambarkan proses melihat dan mencetak daftar obat habis
10	Transaksi pembelian obat	Merupakan <i>usecase</i> transaksi pembelian obat menggambarkan proses transaksi pembelian obat
11	Transaksi penjualan obat	Merupakan <i>usecase</i> transaksi penjualan obat menggambarkan proses transaksi penjualan obat
12	Data frame	Merupakan <i>usecase</i> data frame menggambarkan

		proses pengelolaan data frame
13	Transaksi penjualan kacamata	Merupakan <i>usecase</i> transaksi penjualan kacamata menggambarkan proses transaksi penjualan kacamata
14	Transaksi pelunasan kacamata	Merupakan <i>usecase</i> transaksi pelunasan kacamata menggambarkan proses transaksi pelunasan kacamata
15	Input data pemeriksaan	Merupakan <i>usecase</i> input data pemeriksaan menggambarkan proses input data pemeriksaan
16	Lihat data rekam medis	Merupakan <i>usecase</i> lihat data rekam medis menggambarkan proses melihat data rekam medis
17	Pendaftaran operasi	Merupakan <i>usecase</i> pendaftaran operasi menggambarkan proses pendaftaran operasi pasien
18	Lihat jadwal operasi	Merupakan <i>usecase</i> lihat jadwal operasi menggambarkan proses melihat jadwal operasi

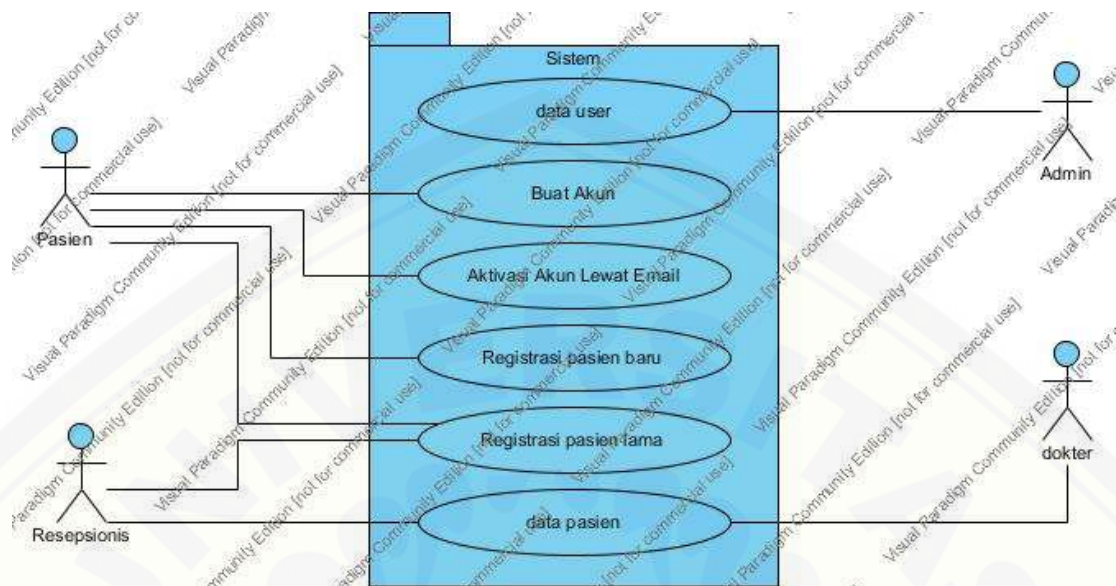
4.4.4 Desain Sistem Tahap *Increment 1*

Pada subbab ini, menjelaskan tentang desain sistem pada tahap *incement* satu. Seperti yang dijelaskan pada subbab 4.3, desain sistem dibuat dengan pemodelan UML (*unified modeling language*).

Telah dijelaskan pada bab 3 subbab 3.4, Fitur yang akan diselesaikan pada tahap *increment* satu meliputi fitur data user, data pasien, buat akun, aktivasi akun lewat *email*, registrasi pasien baru dan registrasi pasien lama.

4.4.4.1 Usecase *Diagram* Tahap *Increment 1*

Usecase diagram pada subbab ini hanya menggambarkan sebuah interaksi antara satu atau lebih aktor dengan fitur-fitur yang dibangun pada tahap *increment 1*. *Usecase* diagram tahap *increment* satu dapat dilihat pada Gambar 4.8.



Gambar 4.8 usecase diagram tahap *increment 1*
 Sumber : (Hasil Analisis, 2014)

Pada gambar diatas, dijelaskan bahwa fitur *data user* dapat diakses oleh admin, fitur *buat akun* dapat di akses oleh pasien, fitur *aktivasi akun lewat email* dapat di akses oleh pasien, fitur *registrasi pasien baru* dapat di akses oleh pasien, fitur *registrasi pasien lama* dapat diakses oleh resepsionis dan pasien, serta fitur *data pasien* dapat diakses oleh resepsionis dan dokter.

4.4.4.2 Usecase Scenario Tahap Increment 1

Usecase scenario menjelaskan urutan interaksi antara aktor dengan sistem. Pada subbab ini, *usecase scenario* menjelaskan urutan interaksi antara aktor dengan sistem pada fitur yang dibangun pada tahap *increment 1*.

a. *Usecase Scenario Data User*

Usecase scenario data user terdiri dari penjelasan urutan interaksi antara aktor dengan sistem fitur *data user* pada skenario normal maupun skenario alternatif. Penjelasan *usecase scenario data user* dapat dilihat pada tabel 4.3.

Tabel 4.4 *Usecase scenario data user*

<i>Name</i>	<i>Data User</i>
<i>Description</i>	Admin dapat menambah, meng- <i>edit</i> , dan menghapus data <i>user</i>
<i>Actor</i>	Admin
<i>Precondition</i>	Admin memilih menu data user
<i>Post condition</i>	Data <i>user</i> telah ditambahkan, diperbarui, atau dihapus.
Normal flow “Tambah data user”	
<i>Actor</i>	<i>System</i>
1. Pilih menu data <i>user</i>	
	2. Menampilkan halaman data <i>user</i> berupa data <i>user</i> . Pada setiap baris data <i>user</i> terdapat dua pilihan yaitu <i>edit</i> dan hapus serta pilihan tambah diatas tabel yang digunakan untuk menambahkan data <i>user</i> baru.
3. Klik Tambah	
	4. Menampilkan form data <i>user</i>
5. Mengisi form data <i>user</i>	
6. Klik ‘ <i>Submit</i> ’	
	7. Memeriksa kelengkapan data
	8. Menyimpan data <i>user</i> baru
	9. Mengembalikan pada halaman data <i>user</i> .
Normal flow “Edit Data User”	
10. Klik <i>edit</i> pada baris data <i>user</i> yang ingin di- <i>edit</i> .	
	11. Menampilkan data <i>user</i> yang akan di <i>edit</i> pada form data <i>user</i> .
12. Meng- <i>edit</i> data <i>user</i> pada form data <i>user</i> .	
13. Klik “ <i>Submit</i> ”	
	14. Memeriksa kelengkapan data
	15. Menyimpan data <i>user</i>

16. Mengembalikan ke halaman Data *user*.

Normal flow “Hapus Data User”

17. Klik hapus pada baris data *user* yang ingin dihapus.

18. Menampilkan *alert* “anda yakin ingin menghapus data ini?”

19. Klik “Ok”

20. Menghapus data *user*

21. Mengembalikan ke halaman data *user*

Alternative flow “Kelengkapan data pada tambah pasien”

- 7.a Apabila *user* memasukkan data tidak lengkap.

- 7.b Klik “Submit”

- 7.c Menampilkan *alert* “Maaf, data belum lengkap”

- 7.d Klik “Ok”

- 7.e Mengembalikan ke form data *user*

Alternative flow “Kelengkapan data pada edit pasien”

- 14.a Apabila *user* memasukkan data tidak lengkap.

- 14.b Klik “Submit”

- 14.c Menampilkan *alert* “Maaf, data belum lengkap”

- 14.d Klik “Ok”

- 14.e Mengembalikan ke form data *user*

Alternative flow “Batal menghapus data user”

- 18.a Apabila *user* batal menghapus data *user*.

- 18.b Klik “Batal”

- 18.c Mengembalikan ke halaman data *user*”

b. *Usecase Scenario* Buat Akun

Usecase scenario Buat Akun terdiri dari penjelasan urutan interaksi antara aktor dengan sistem fitur data pasien pada skenario normal maupun skenario alternatif. Penjelasan *usecase scenario* buat akun dapat dilihat pada tabel 4.4.

Tabel 4.5 *Usecase Scenario* Buat Akun

<i>Name</i>		Buat Akun
<i>Description</i>	Pasien membuat akun untuk mengakses sistem	
<i>Actor</i>	Pasien	
<i>Precondition</i>	Pasien memilih menu buat akun	
<i>Postcondition</i>	Pasien telah memiliki akun	
Normal flow “Buat Akun”		
<i>Actor</i>	<i>System</i>	
1. Pilih menu buat akun		
	2.	Menampilkan form buat akun
3. Mengisi data akun		
4. Klik “ <i>Submit</i> ”		
	5.	Memeriksa kelengkapan data
	6.	Mengirim pesan aktivasi ke email pasien
	7.	Menyimpan data akun pasien
	8.	Menampilkan halaman login
Alternative flow “kelengkapan data”		
5.a	Apabila data yang di masukkan tidak lengkap	
5.b	Klik “ <i>Submit</i> ”	
	5.c	Menampilkan <i>alert</i> “Maaf, data tidak lengkap”
5.d	Klik “Ok”	
	5.e	Mengembalikan ke form buat akun

c. *Usecase Scenario* Aktivasi Akun Lewat *Email*

Usecase scenario aktivasi akun lewat *email* terdiri dari penjelasan urutan interaksi antara aktor dengan sistem fitur data pasien pada skenario normal maupun skenario alternatif. Penjelasan *usecase scenario* aktivasi akun lewat *email* dapat dilihat pada tabel 4.5.

Tabel 4.6 *Usecase scenario* aktivasi akun lewat *email*

<i>Name</i>	Aktivasi Akun Lewat <i>Email</i>
<i>Description</i>	Pasien melakukan aktivasi akun melalui <i>email</i> agar akun menjadi aktif
<i>Actor</i>	Pasien
<i>Precondition</i>	Pasien telah membuat akun
<i>Postcondition</i>	Akun pasien telah aktif
Normal flow “Aktivasi Akun Lewat <i>Email</i>”	
<i>Actor</i>	<i>System</i>
1. Klik <i>link</i> aktivasi yang telah dikirim pada <i>email</i>	
	2. Mengupdate status akun pada database
	3. Menampilkan halaman login sistem

d. *Usecase Scenario* Data Pasien

Usecase scenario data pasien terdiri dari penjelasan urutan interaksi antara aktor dengan sistem fitur data pasien pada skenario normal maupun skenario alternatif. Penjelasan *usecase scenario* data pasien dapat dilihat pada tabel 4.6.

Tabel 4.7 *Usecase scenario* data pasien

<i>Name</i>	Data Pasien
<i>Description</i>	Resepsionis dapat menambah, melihat, dan meng- <i>edit</i> data pasien
<i>Actor</i>	Resepsionis
<i>Precondition</i>	Resepsionis memilih menu data pasien
<i>Postcondition</i>	Data pasien telah ditambahkan atau diperbarui.
Normal flow “Tambah data pasien”	

<i>Actor</i>	<i>System</i>
1. Pilih menu data pasien	2. Menampilkan halaman Data pasien berupa data pasien. Pada setiap baris data pasien terdapat pilihan yaitu <i>edit</i> serta pilihan tambah diatas tabel yang digunakan untuk menambahkan data pasien baru.
3. Klik "Tambah"	4. Menampilkan form data pasien
5. Mengisi form data pasien	7. Memeriksa kelengkapan data
6. Klik "Submit"	8. Menyimpan data pasien baru
	9. Menampilkan <i>alert</i> "Data berhasil ditambahkan"
10. Klik "Ok"	11. Menampilkan halaman data pasien yang telah diperbaharui
Normal flow "Edit Data Pasien"	
12. Klik " <i>edit</i> " pada baris data pasien yang ingin di- <i>edit</i> .	13. Menampilkan data pasien yang akan di- <i>edit</i> pada form data pasien.
14. Meng- <i>edit</i> data pasien pada form data pasien.	
15. Klik "Submit"	16. Memeriksa kelengkapan data
	17. Meng- <i>update</i> data pasien
	18. Menampilkan <i>alert</i> "data pasien berhasil diperbarui"
19. Klik "Ok"	20. Menampilkan halaman data pasien yang telah diperbaharui.
Normal flow "View Data Pasien"	
21. Klik " <i>view</i> " pada baris data pasien yang ingin dilihat	22. Menampilkan detail data pasien sesuai yang dipilih
23. Klik " <i>close</i> "	

	24. Menampilkan halaman data pasien
Alternative flow “kelengkapan data pada tambah pasien”	
7.a	Apabila <i>user</i> memasukkan data tidak lengkap.
7.b	Klik “ <i>Submit</i> ”
	7.c Menampilkan <i>alert</i> “Maaf, data belum lengkap”
7.d	Klik “Ok”
	7.e Mengembalikan ke form tambah data pasien
Alternative flow “kelengkapan data pada edit pasien”	
16.a	Apabila <i>user</i> memasukkan data tidak lengkap.
16.b	Klik “ <i>Submit</i> ”
	16.c Menampilkan <i>alert</i> “Maaf, data belum lengkap”
16.d	Klik “Ok”
	16.e Mengembalikan ke form tambah data pasien
Alternative flow “Batal menambah data pasien”	
6.a	Klik “ <i>cancel</i> ”
	6.b Mengembalikan ke halaman data pasien
Alternative flow “Batal mengedit data pasien”	
15.a	Klik “ <i>cancel</i> ”
	15.b Mengembalikan ke halaman data pasien

e. *Usecase Scenario* Registrasi Pasien Baru

Usecase scenario registrasi pasien baru terdiri dari penjelasan urutan interaksi antara aktor dengan sistem fitur registrasi pasien pada skenario normal maupun skenario alternatif. Penjelasan *usecase scenario* registrasi pasien baru dapat dilihat pada tabel 4.7.

Tabel 4.8 *Usecase scenario* registrasi pasien baru

Name	Registrasi Pasien Baru
Description	pasien dapat melakukan registrasi pasien

	melalui web
Actor	pasien
Precondition	Pasien belum memiliki id pasien
Postcondition	Pasien telah terdaftar dan mendapatkan nomor antrian
Normal flow “Registrasi Pasien Baru”	
Actor	System
1. Pilih menu registrasi pasien (baru)	
	2. Menampilkan form registrasi pasien
3. Mengisi form registrasi pasien	
4. Klik “Submit”	
	5. Memeriksa kelengkapan data
	6. Menyimpan data registrasi pasien
	7. Menampilkan halaman lihat antrian
Alternative flow “kelengkapan data”	
5.a Apabila data yang di masukkan tidak lengkap	
5.b Klik “Submit”	
	5.c Menampilkan <i>alert</i> “Maaf, data tidak lengkap”
5.d Klik “Ok”	
	5.e Mengembalikan ke form registrasi pasien
Alternative flow “batal registrasi”	
4.a Apabila <i>user</i> batal melakukan registrasi	
4.b Klik ‘cancel’	
	4.c Menampilkan halaman utama

f. *Usecase Scenario* Registrasi Pasien Lama

Usecase scenario registrasi pasien terdiri dari penjelasan urutan interaksi antara aktor dengan sistem fitur registrasi pasien pada skenario normal maupun

skenario alternatif. Penjelasan *usecase scenario* registrasi pasien dapat dilihat pada tabel 4.8.

Tabel 4.9 *Usecase Scenario* registrasi pasien lama

Name	Registrasi Pasien
Description	Resepsionis dan pasien dapat melakukan registrasi pasien melalui web
Actor	Resepsionis dan pasien
Precondition	Pasien memiliki id pasien
Postcondition	Pasien telah terdaftar dan mendapatkan nomor antrian
Normal flow “Registrasi Pasien Lama”	
Actor	System
1. Pilih menu registrasi pasien (lama)	
	2. Menampilkan form registrasi pasien beserta daftar antrian pasien
3. Mengisi form registrasi pasien	
4. Klik “Submit”	
	5. Memeriksa kelengkapan data
	6. Menyimpan data registrasi pasien
	7. Menampilkan alert berisi No Antrian dan Waktu Pemeriksaan
8. Klik “Cetak Kartu”	
	9. Mencetak kartu antrian
	10. Menampilkan form registrasi pasien beserta daftar antrian pasien
Alternative flow “kelengkapan data”	
5.a Apabila data yang di masukkan tidak lengkap	
5.b Klik “Submit”	
	5.c Menampilkan <i>alert</i> “Maaf, data tidak lengkap”
5.d Klik “Ok”	

	5.e Mengembalikan ke form registrasi pasien
Alternative flow “batal registrasi”	
4.a Apabila <i>user</i> batal melakukan registrasi	
4.b Klik ‘cancel’	
	4.c Mengosongkan <i>form</i> registrasi pasien

4.4.4.3 Activity Diagram Tahap Increment 1

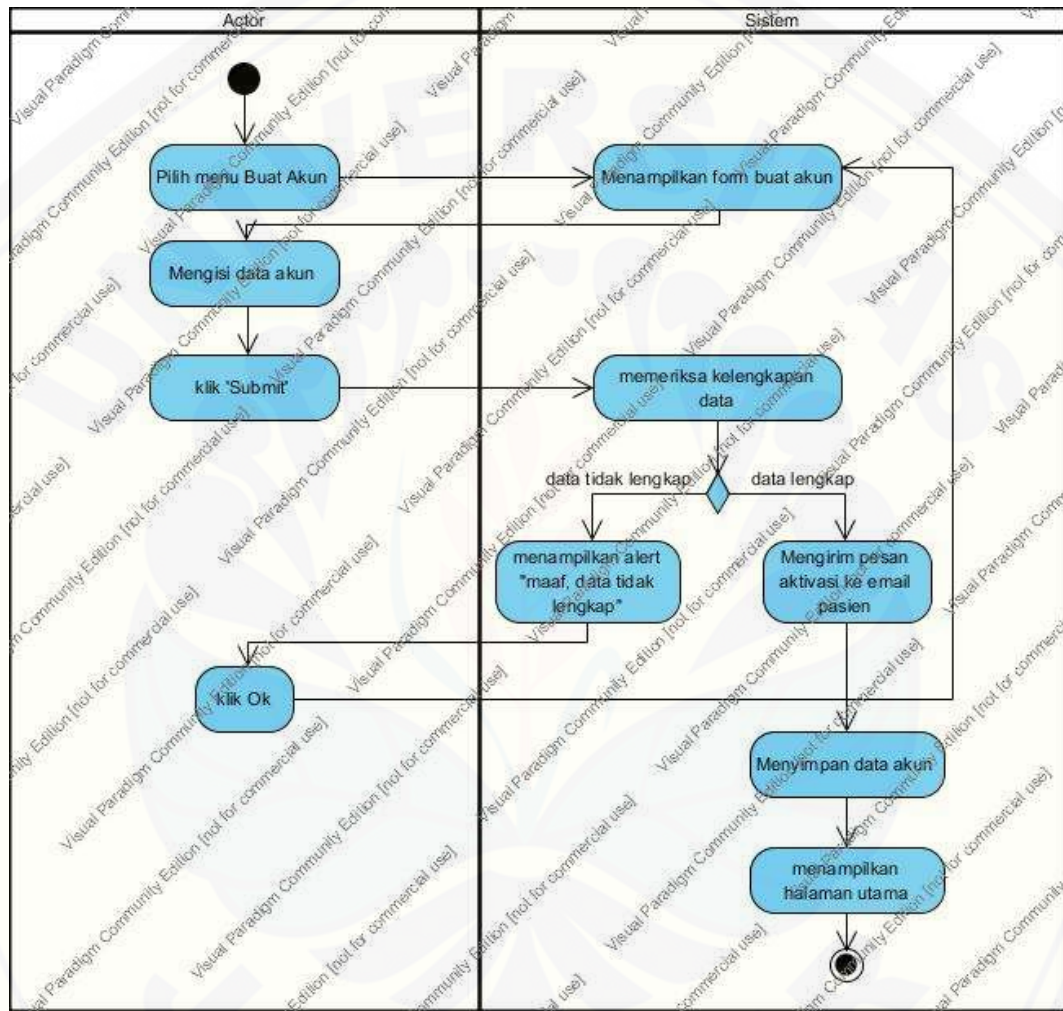
Activity diagram menggambarkan alur aktivitas antara aktor dan sistem. Pada subbab ini, *activity* diagram menggambarkan alur aktivitas pada fitur yang dibangun pada tahap *increment* 1.

a. Activity Diagram Data User

Activity diagram data user merupakan penggambaran alur aktivitas antara aktor dan sistem. Penggambaran *activity* diagram data *user* dapat dilihat pada Gambar 4.9.

b. *Activity Diagram* Buat Akun

Activity diagram buat akun merupakan penggambaran alur aktivitas antara aktor dan sistem. Penggambaran *activity* diagram buat akun dapat dilihat pada Gambar 4.10.



Gambar 4.10 *Activity Diagram* Buat Akun

Sumber : (Hasil Analisis, 2014)

Gambar 4.10 menggambarkan alur aktifitas pada fitur buat akun yang dimulai dari memilih menu buat akun, mengisi data akun, klik 'submit', hingga menampilkan halaman login.

c. *Activity Diagram* Aktivasi Akun Melalui *Email*

Activity diagram aktivasi akun melalui *email* merupakan penggambaran alur aktivitas antara aktor dan sistem. Penggambaran *activity diagram* aktivasi akun melalui *email* dapat dilihat pada Gambar 4.11.

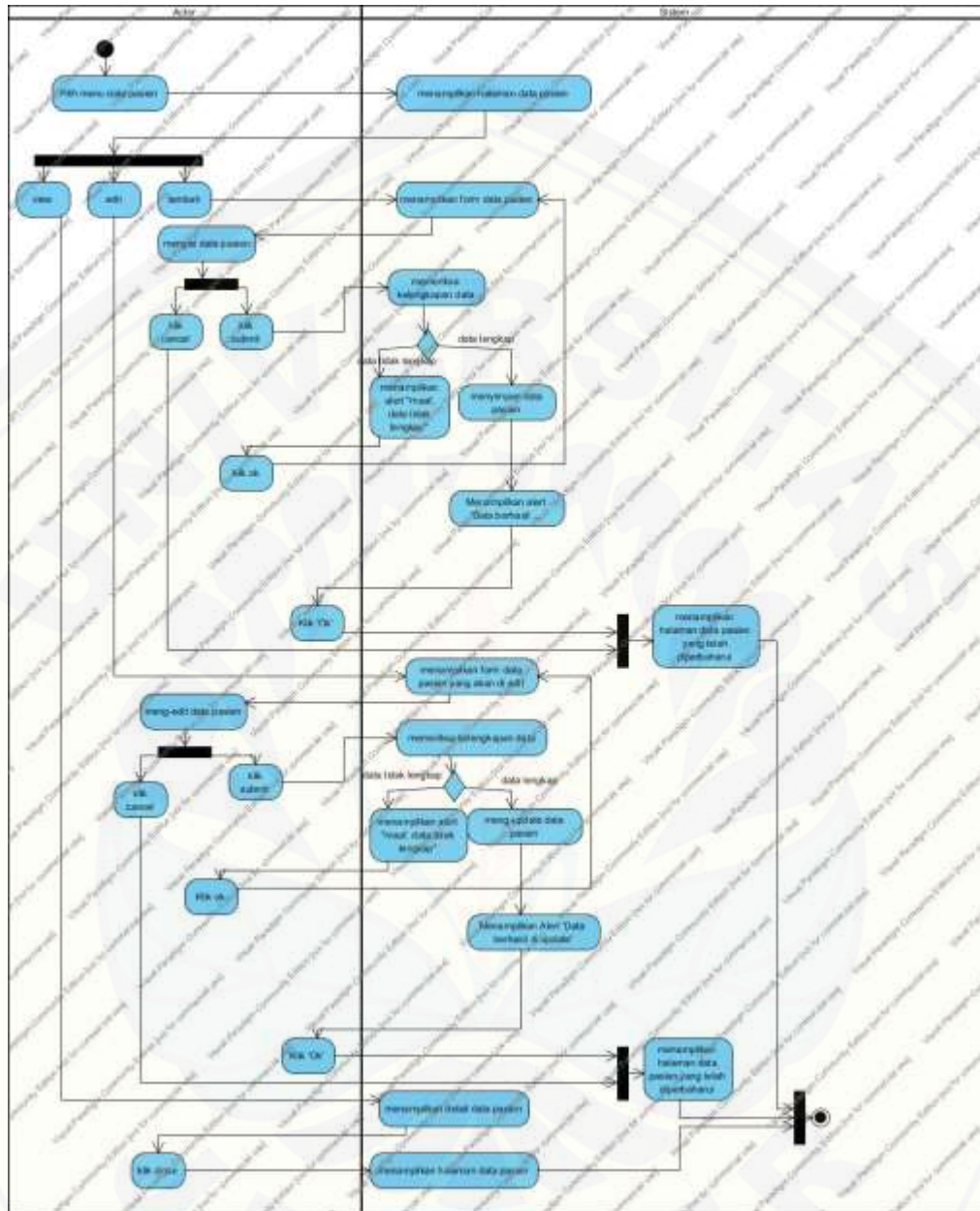


Gambar 4.11 *Activity Diagram* Aktivasi Akun Lewat Email
Sumber : (Hasil Analisis, 2015)

Gambar 4.11 menggambarkan alur aktifitas pada fitur aktivasi akun melalui *email* yang dimulai dari menekan link aktivasi pada *email* hingga menampilkan halaman login.

d. *Activity Diagram* Data Pasien

Activity diagram data pasien merupakan penggambaran alur aktivitas antara aktor dan sistem. Penggambaran *activity diagram* data pasien dapat dilihat pada Gambar 4.12.

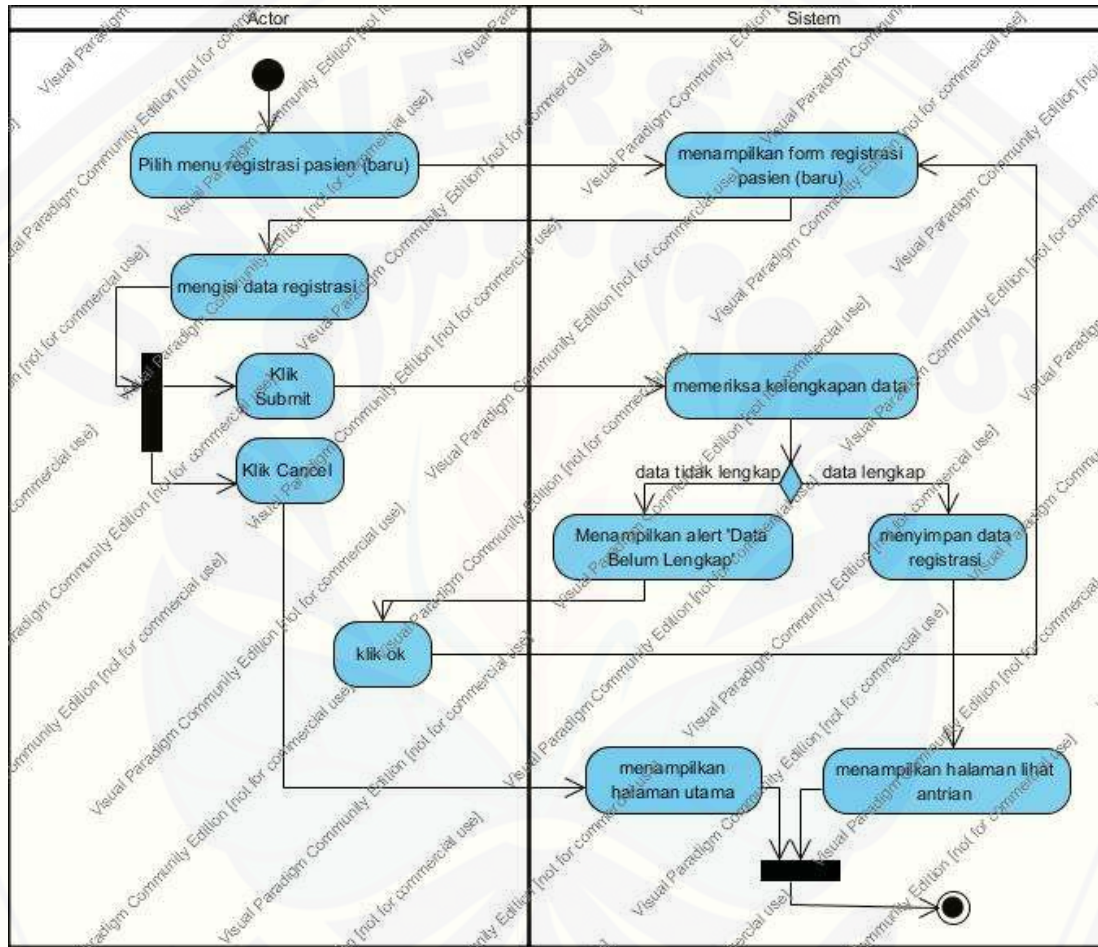


Gambar 4.12 Activity diagram data pasien
 Sumber : (Hasil Analisis, 2014)

Gambar 4.12 menggambarkan alur aktifitas pada fitur data pasien yang dimulai dari memilih menu data pasien. Halaman menu data user terdapat tabel data user dan tiga tombol yaitu, tambah, edit, dan *view*. Pada gambar diatas digambarkan alur aktifitas hingga akhir aktifitas yang terjadi pada fitur data pasien.

e. *Activity Diagram* Registrasi Pasien Baru

Activity diagram registrasi pasien baru merupakan penggambaran alur aktivitas antara aktor dan sistem. Penggambaran *activity* diagram registrasi pasien baru dapat dilihat pada Gambar 4.13.



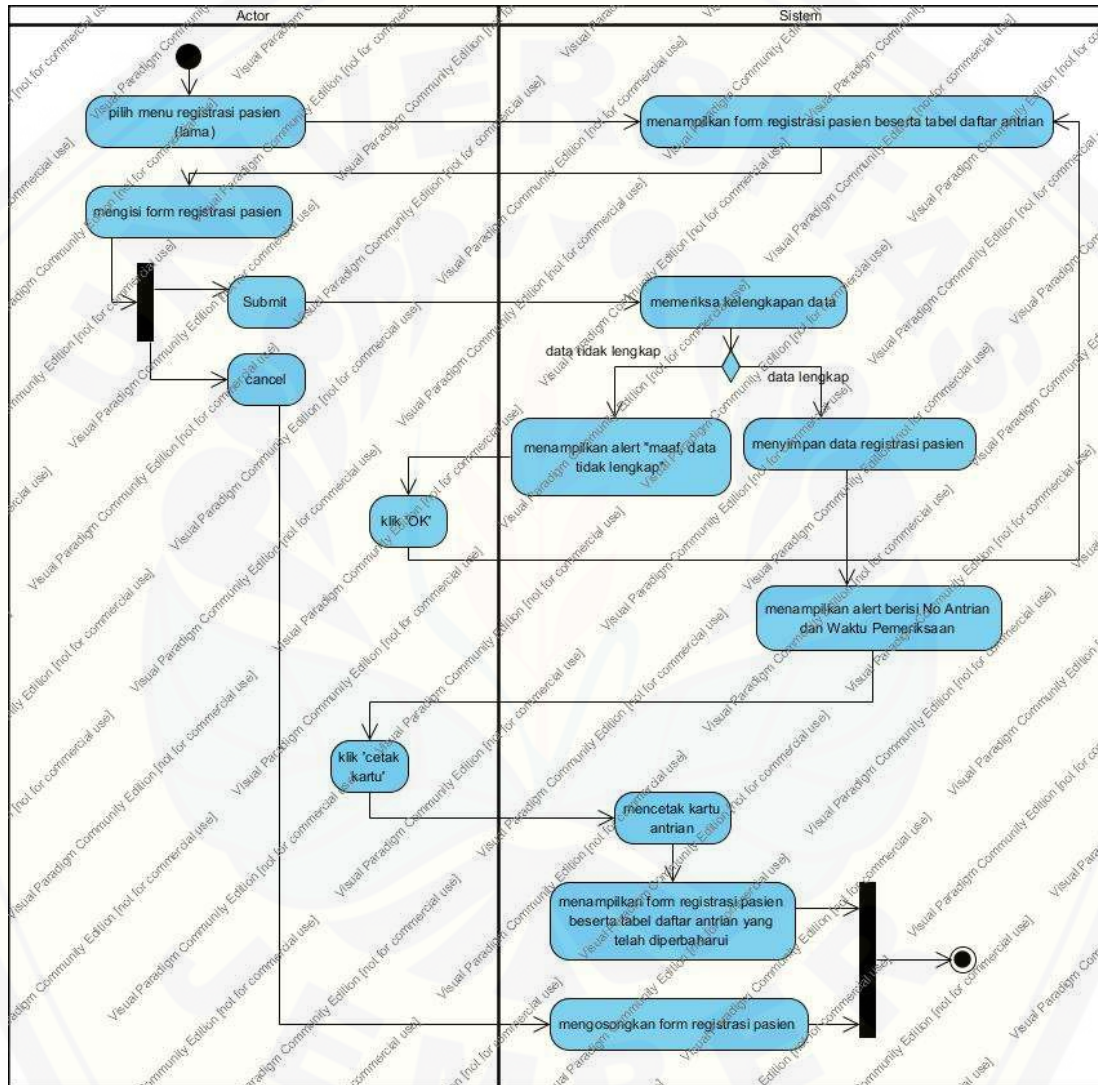
Gambar 4.13 *Activity Diagram* Registrasi Pasien Baru

Sumber : (Hasil Analisis, 2015)

Gambar 4.13 menggambarkan alur aktifitas pada fitur registrasi pasien baru yang dimulai dari memilih menu registrasi pasien. Pada gambar diatas digambarkan alur aktifitas hingga akhir aktifitas yang terjadi pada fitur registrasi pasien.

f. *Activity* Diagram Registrasi Pasien Lama

Activity diagram registrasi pasien lama merupakan penggambaran alur aktivitas antara aktor dan sistem. Penggambaran *activity* diagram registrasi pasien lama dapat dilihat pada Gambar 4.14.



Gambar 4.14 *Activity* diagram registrasi pasien
 Sumber : (Hasil Analisis, 2014)

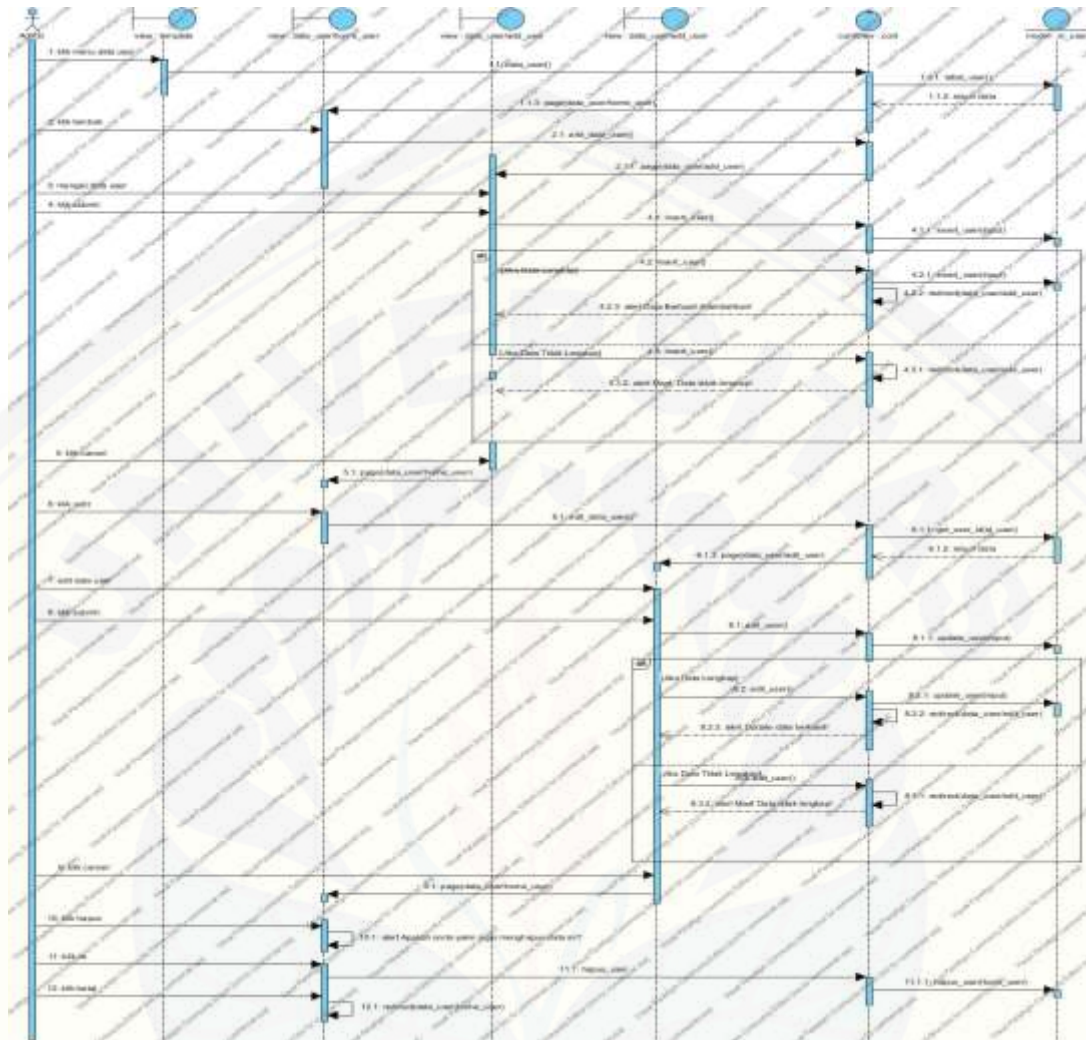
Gambar 4.14 menggambarkan alur aktifitas pada fitur registrasi pasien yang dimulai dari memilih menu registrasi pasien. Pada gambar diatas digambarkan alur aktifitas hingga akhir aktifitas yang terjadi pada fitur registrasi pasien.

4.4.4.4 Sequence Diagram Tahap Increment 1

Sequence diagram menggambarkan skenario dan memodelkan aliran logika sistem. Pada subbab ini, *sequence* diagram menggambarkan skenario dan memodelkan aliran logika pada fitur yang dibangun pada tahap *increment 1*.

a. *Sequence Diagram Data User*

Sequence diagram data *user* merupakan penggambaran skenario dan pemodelan aliran logika sistem pada fitur data *user*. Penggambaran *sequence* diagram data *user* dapat dilihat pada Gambar 4.15.

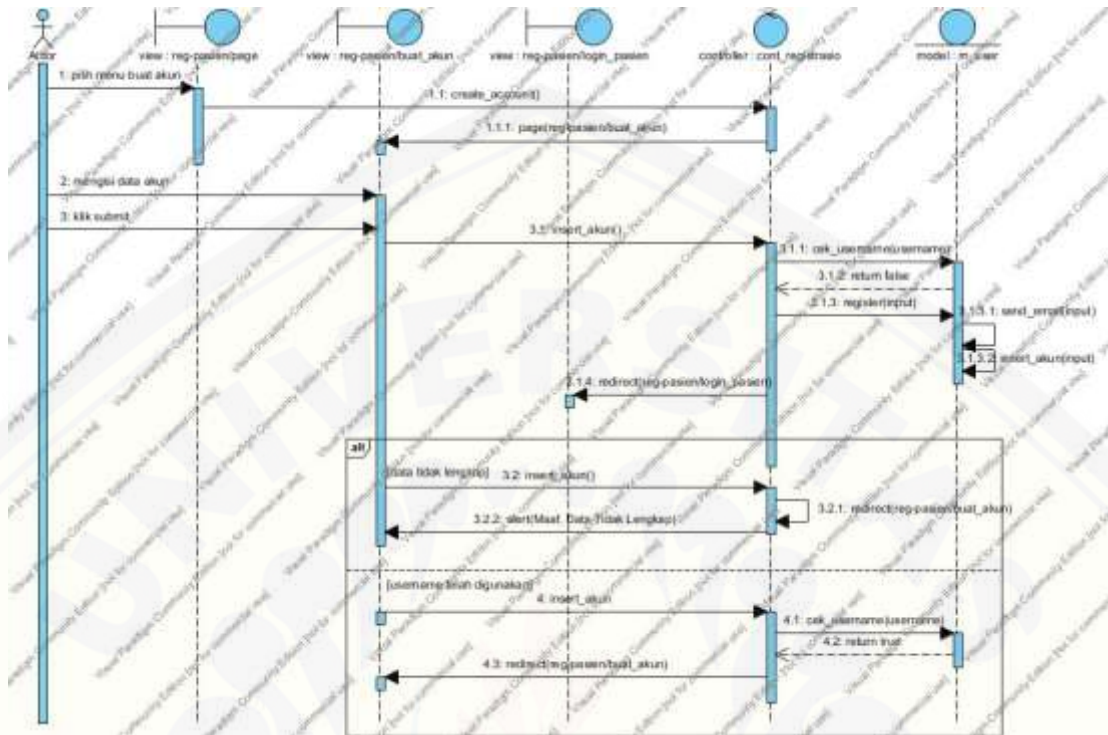


Gambar 4.15 *Sequence* diagram data user
Sumber : (Hasil Analisis, 2014)

Gambar 4.15 menggambarkan skenario dan memodelkan aliran logika pada fitur data user. Terdapat empat *view*, satu *controller*, dan satu model yang digunakan pada fitur data user.

b. *Sequence* Diagram Buat Akun

Sequence diagram Buat Akun merupakan penggambaran skenario dan pemodelan aliran logika sistem pada fitur Buat Akun. Penggambaran *sequence* diagram Buat Akun dapat dilihat pada Gambar 4.16.

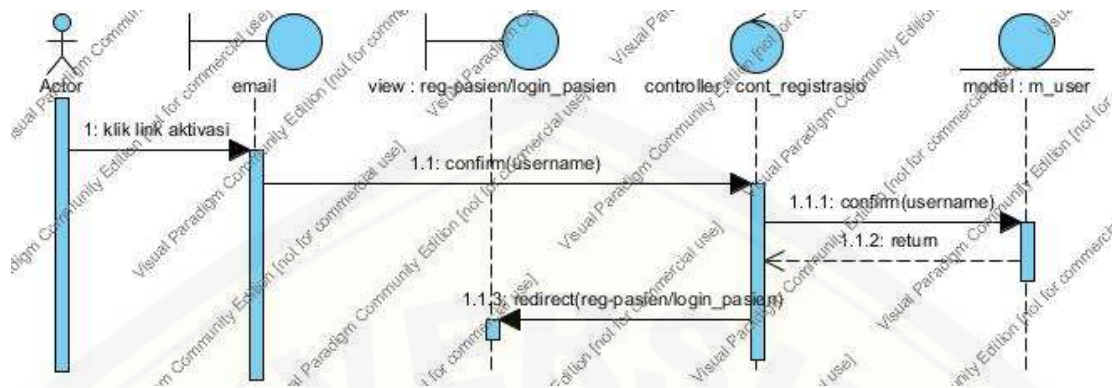


Gambar 4.16 *Sequence Diagram* Buat Akun
Sumber : (Hasil Analisis, 2015)

Gambar 4.16 menggambarkan skenario dan memodelkan aliran logika pada fitur buat akun. Terdapat tiga *view*, satu *controller*, dan satu model yang digunakan pada fitur buat akun.

c. *Sequence Diagram* Aktivasi Akun Lewat Email

Sequence diagram Aktivasi Akun Lewat Email merupakan penggambaran skenario dan pemodelan aliran logika sistem pada fitur Aktivasi Akun Lewat Email. Penggambaran *sequence diagram* Aktivasi Akun Lewat Email dapat dilihat pada Gambar 4.17.



Gambar 4.17 *Sequence Diagram* Aktivasi Akun Lewat Email
Sumber : (Hasil Analisis, 2015)

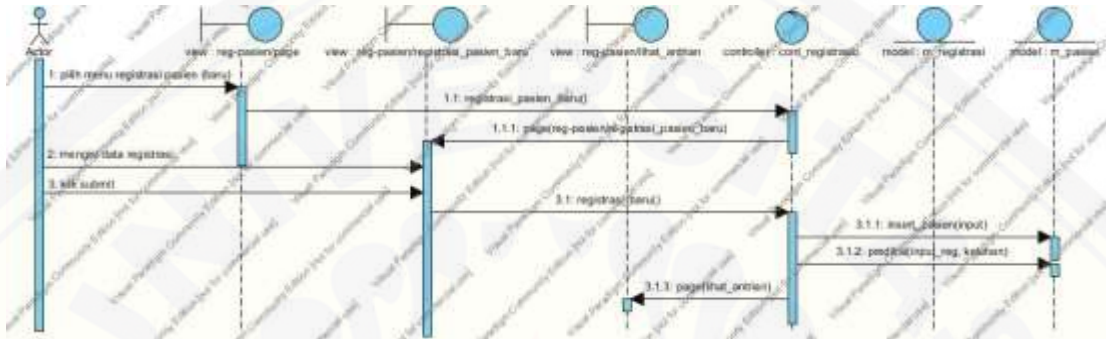
Gambar 4.17 menggambarkan skenario dan memodelkan aliran logika pada fitur Aktivasi Akun Lewat Email. Terdapat satu *view*, satu *controller*, dan satu model yang digunakan pada fitur Aktivasi Akun Lewat Email.

d. *Sequence Diagram* Data Pasien

Sequence diagram data pasien merupakan penggambaran skenario dan pemodelan aliran logika sistem pada fitur data pasien. Penggambaran *sequence diagram* data pasien dapat dilihat pada Gambar 4.18.

e. *Sequence Diagram* Registrasi Pasien Baru

Sequence diagram data pasien merupakan penggambaran skenario dan pemodelan aliran logika sistem pada fitur data pasien. Penggambaran *sequence* diagram data pasien dapat dilihat pada Gambar 4.19.

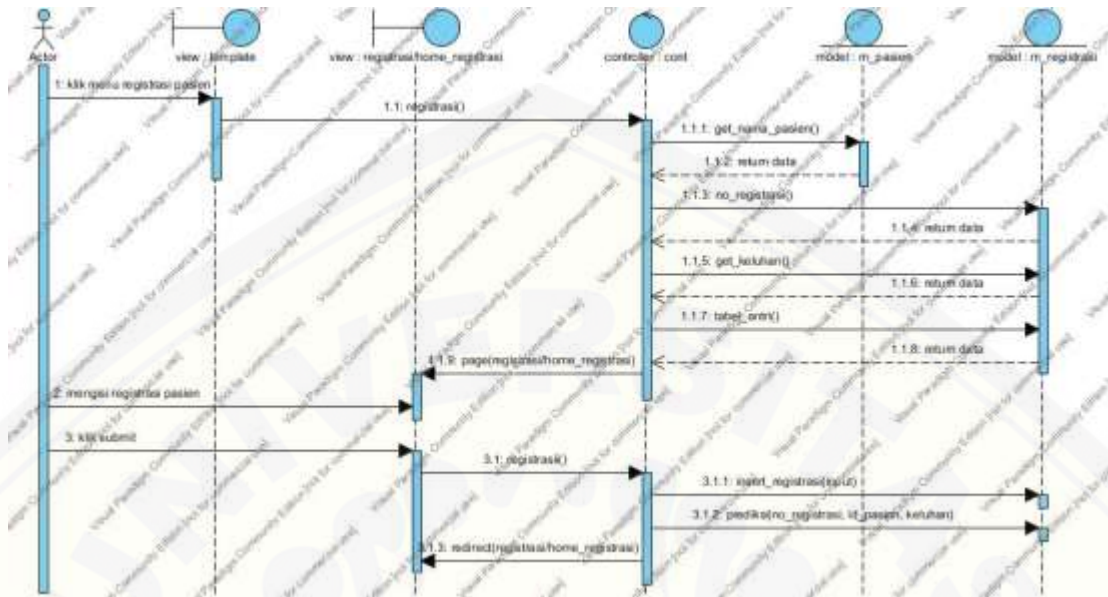


Gambar 4.19 *Sequence Diagram* Registrasi Pasien Baru

Gambar 4.19 menggambarkan skenario dan memodelkan aliran logika pada fitur registrasi pasien baru. Terdapat empat *view*, satu *controller*, dan satu model yang digunakan pada fitur registrasi pasien baru.

f. *Sequence Diagram* Registrasi Pasien Lama

Sequence diagram registrasi pasien merupakan penggambaran skenario dan pemodelan aliran logika sistem pada fitur registrasi pasien. Penggambaran *sequence* diagram registrasi pasien dapat dilihat pada Gambar 4.20.

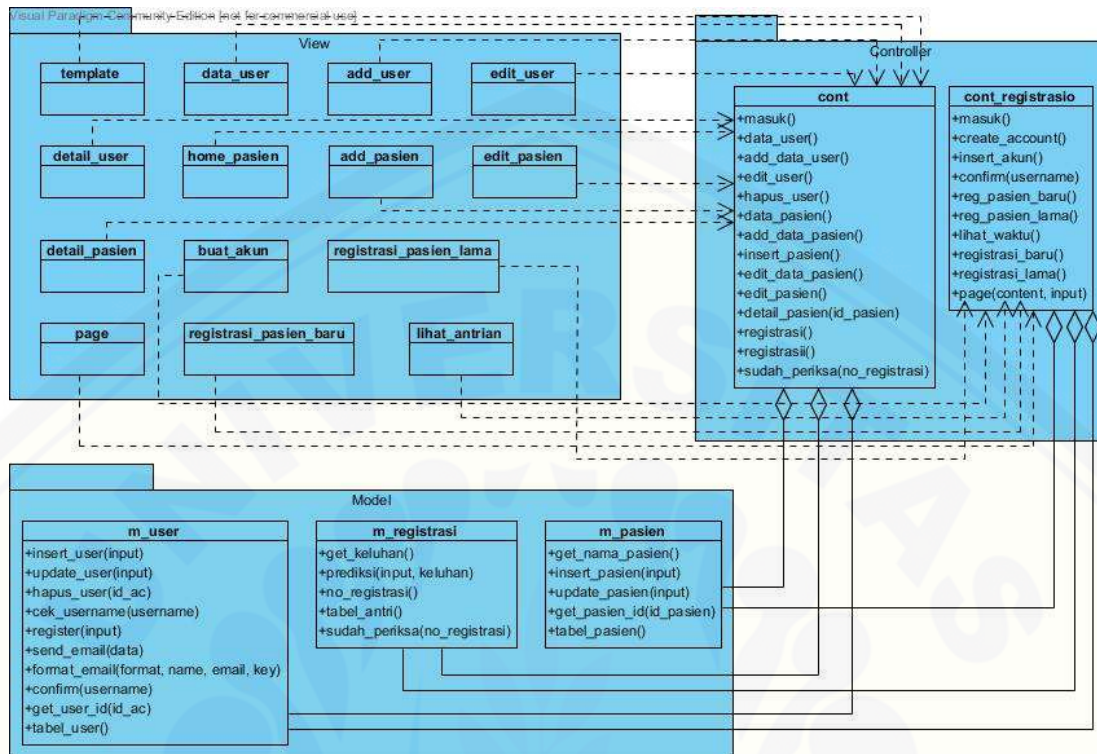


Gambar 4.20 *Sequence Diagram* Registrasi Pasien
Sumber : (Hasil Analisis, 2014)

Gambar 4.15 menggambarkan skenario dan memodelkan aliran logika pada fitur registrasi pasien lama. Terdapat dua *view*, satu *controller*, dan dua model yang digunakan pada fitur data user registrasi pasien lama.

4.4.4.5 *Class Diagram Tahap Increment 1*

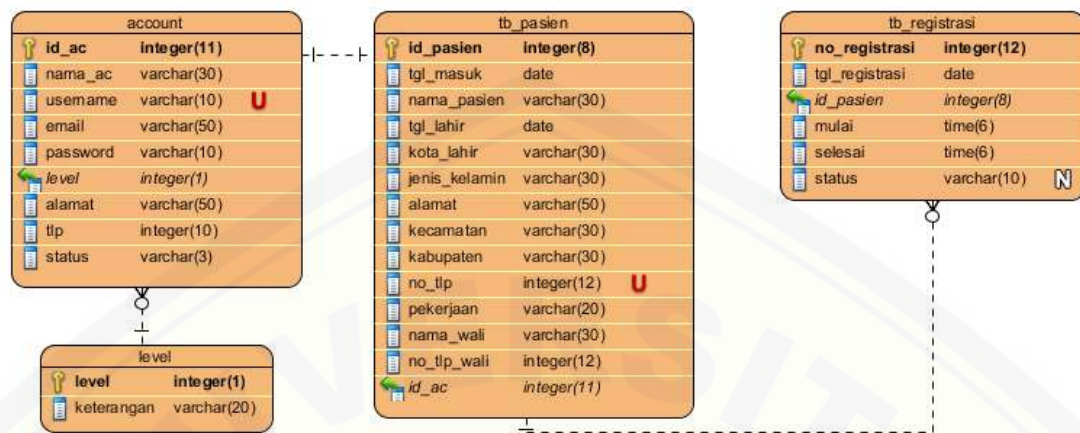
Class diagram menggambarkan kelas-kelas pada sistem yang dibangun dan hubungan antara kelas satu dan lainnya serta berisi atribut dan *method* apa saja yang ada didalamnya. Pada subbab ini, *class* diagram menggambarkan kelas-kelas yang digunakan pada fitur yang dibangun pada tahap *increment 1*.

Gambar 4.21 *Class Diagram Tahap Increment 1*

Gambar 4.21 merupakan *class diagram* tahap *increment* satu. Terdapat empat belas view, dua controller, dan tiga model. Delapan view memiliki hubungan dependency dengan controller-cont dan enam view memiliki hubungan dependency dengan controller cont_registrasio. Sedangkan masing-masing model memiliki hubungan agregat terhap masing-masing controller.

4.4.4.6 *Entity Relationship Diagram Tahap Increment 1*

Entity relationship diagram menggambarkan hubungan antar entitas dalam sebuah basis data. Pada subbab ini, *entity relationship* diagram menggambarkan hubungan antar entitas pada fitur yang dibangun pada tahap *increment 1*.

Gambar 4.22 ERD Tahap *Increment 1*

Sumber : (Hasil Analisis, 2014)

Gambar 4.22 merupakan entity relationship diagram tahap *increment* satu. Sesuai dengan fitur-fitur yang dibuat pada tahap *increment* satu, entitas yang dibutuhkan adalah *account*, *level*, *tb_pasien*, dan *tb_registrasi*. Entitas *account* dan *level* memiliki relasi *one-to-many* dengan atribut *level* yang merupakan *primary key* (PK) pada entitas *level* menjadi *foreign key* (FK) pada entitas *account*. Sedangkan entitas *tb_pasien* dan *tb_registrasi* memiliki relasi *one-to-many* dengan atribut *id_pasien* yang merupakan *primary key* (PK) pada entitas *tb_registrasi* menjadi *foreign key* (FK) pada entitas *tb_registrasi*.

4.4.5 Desain Sistem Tahap *Increment 2*

Desain sistem tahap *increment* dua akan menjelaskan tentang desain sistem yang akan dibangun pada tahap *increment* dua. Seperti yang telah di jelaskan pada bab 3 subbab 3.4, fitur-fitur yang akan diselesaikan pada tahap *increment* dua meliputi fitur data obat, data *supplier*, daftar obat habis, transaksi pembelian obat, dan transaksi penjualan obat. Desain sistem tahap *increment* ini akan dijelaskan pada Lampiran A.1.

4.4.6 Desain Sistem Tahap *Increment* 3

Desain sistem tahap *increment* tiga akan menjelaskan tentang desain sistem yang akan dibangun pada tahap *increment* tiga. Seperti yang telah di jelaskan pada bab 3 subbab 3.4, fitur-fitur yang akan diselesaikan pada tahap *increment* tiga meliputi fitur data *frame*, transaksi pembelian kacamata, transaksi penjualan kacamata, dan transaksi pelunasan kacamata. Desain sistem tahap *increment* ini akan dijelaskan pada Lampiran A.2.

4.4.7 Desain Sistem Tahap *Increment* 4

Desain sistem tahap *increment* empat akan menjelaskan tentang desain sistem yang akan dibangun pada tahap *increment* empat. Seperti yang telah di jelaskan pada bab 3 subbab 3.4, fitur-fitur yang akan diselesaikan pada tahap *increment* empat meliputi fitur input data pemeriksaan, lihat data rekam medis, pendaftaran operasi, dan lihat jadwal operasi. Desain sistem tahap *increment* ini akan dijelaskan pada lampiran A.3.

4.5. Pengkodean

Pengkodean atau *coding* merupakan proses menulis kode dalam membangun sebuah program. Dalam proses pengkodean sistem informasi klinik mata, penulis menggunakan bahasa pemrograman *Page Hyper Text Pre-processor* (PHP), *Hyper Text Markup Language* (HTML), *Cascading Style Sheet* (CSS), dan *Javascript*. MySQL sebagai sistem manajemen *database*. Pengkodean sistem menggunakan konsep *object oriented programming* (OOP) menggunakan *framework codeigniter*. Berikut potongan kode program untuk fitur registrasi pasien lama yang terdapat pada kelas *controller*.

Tabel 4.10 potongan kode program untuk fitur registrasi pasien lama (*controller*)

```
function registrasii(){
    $input = array(
        "no_registrasi" => $this->input->post('no_registrasi'),
        "tgl_registrasi" => date('Y-m-d', strtotime($this->input-
>post('tgl_registrasi'))),
        "id_pasien" => $this->input->post('id_pasien')
    );

    $keluhan = $this->input->post('keluhan');

    $this->m_registrasi->prediksi($input, $keluhan);
    redirect(base_url() . "index.php/cont/registrasi?status=sukses");
}
```

Tabel 4.9 merupakan potongan kode program fitur registrasi pasien lama pada kelas *controller*. Nama *function* yang digunakan adalah *registrasii()*, didalamnya terdapat beberapa variabel dan beberapa kode program untuk memanggil *function* yang berada pada kelas model *m_registrasi* yaitu *function* *prediksi()*. Potongan kode program pada *function* *prediksi()* dapat dilihat pada Tabel 4.10.

Tabel 4.11 potongan kode program *function* *prediksi()*

```
function prediksi($input, $keluhan) {
    $nilai = array();
    $arr_keluhan = array();
    $atr = $this->db->query('select nama_atribut from tb_jenisatribut '
        . 'where nama_atribut <> "usia" AND nama_atribut <> "jenis_kelamin" ');
    $result = $atr->result_array();
    foreach ($result as $atribut) {
        $sama = false;
        foreach ($keluhan as $k) {
            if ($atribut["nama_atribut"] == $k) {
                $sama = true;
            }
        }
    }
    if ($sama) {
        $arr_keluhan[$atribut["nama_atribut"]] = "Ya";
    } else {
        $arr_keluhan[$atribut["nama_atribut"]] = "Tidak";
    }
}
```



```
}
}
//ambil usia
$umurjk = $this->db->query('SELECT id_pasien,
    YEAR(CURDATE()) - YEAR(tgl_lahir) as umur, jenis_kelamin
    FROM tb_pasien where id_pasien=' . $input['id_pasien']);
$result1 = $umurjk->row_array();

$arr_keluhan["usia_temp"] = $result1['umur'];
$arr_keluhan["jenis_kelamin"] = $result1['jenis_kelamin'];

$rule = $this->db->get("rule_c45");
$resut_rule = $rule->result_array();

$hasil = "";
foreach ($resut_rule as $rrule) {
    $id_rule = $rrule["id"];
    $group_rule = $this->db->get_where("rule_penentu_keputusan",
array("id_rule" => $id_rule));
    $result_group = $group_rule->result_array();
    $cocok = true;
    foreach ($result_group as $group) {
        foreach ($arr_keluhan as $keluh => $nilai) {
            if ($group["atribut"] == $keluh) {
                if ($group["atribut"] == "usia_temp") {
                    $rule_usia = explode(" ", $group["nilai_atribut"]);
                    $tanda = $rule_usia[0];
                    $usia = $rule_usia[1];

                    if ($tanda == ">") {
                        if ($nilai <= $usia) {
                            $cocok = false;
                            break;
                        }
                    }
                    } else if ($tanda == "<=") {
                        if ($nilai > $usia) {
                            $cocok = false;
                            break;
                        }
                    }
                }
            }
        }
    }
}
```

```
        } else {
            if ($group["nilai_atribut"] != $nilai) {
                $cocok = false;
                break;
            }
        }
    }
    if ($cocok) {
        $hasil = $rrule["keputusan"];
        break;
    }
    if ($hasil == "") {
        $hasil = "Cepat";
    }

    //select dari antrian
    $selectAntrian = $this->db->query("select selesai as mulai from "
        . "tb_registrasi where tgl_registrasi = " . $input['tgl_registrasi']
        . " order by no_registrasi desc limit 1");
    $mulai = $selectAntrian->row_array();

    if ($mulai['mulai'] == NULL) {
        $waktuMulai = "17:00:00";
    } else {
        $waktuMulai = $mulai['mulai'];
    }
    //hitung selesainya
    $waktuSelesai = explode(":", $waktuMulai);
    $tambahMenit = 0;
    if ($hasil == "Cepat") {
        $tambahMenit = 10;
    } elseif ($hasil == "Sedang") {
        $tambahMenit = 20;
    } else { //ini kalo lama
        $tambahMenit = 30;
    }
    $waktuSelesai[1] += $tambahMenit;
    if (($waktuSelesai[1]) >= 60) {
        $waktuSelesai[0] ++;
        $waktuSelesai[1] -= 60;
    }
    $Selesai = implode(":", $waktuSelesai);
```

```
$this->db->insert("tb_registrasi", array("no_registrasi" =>
$input['no_registrasi'],
    "tgl_registrasi" => $input['tgl_registrasi'], "id_pasien" => $input['id_pasien'],
    "mulai" => $waktuMulai, "selesai" => $Selesai,
    "status" => "Waiting"));
}
```

Tabel 4.10 merupakan potongan kode program model registrasi pasien lama yaitu *function* prediksi(). Terdapat beberapa logika dalam menentukan waktu pemeriksaan yaitu untuk menebak lama penanganan pasien kemudian di konversi menjadi waktu pemeriksaan pasien dan penyimpanan data registrasi ke dalam database. Kode program fitur-fitur lain yang bangun pada sistem informasi klinik mata dapat dilihat pada Lampiran C.

BAB 5. HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil dan pembahasan dari Sistem Informasi Klinik Mata Dengan Prediksi Waktu Pemeriksaan Menggunakan Algoritma C4.5 Pada Fitur Registrasi Pasien

5.1. Implementasi Algoritma C4.5 untuk Prediksi Waktu Pemeriksaan Pasien

Implementasi algoritma c4.5 pada fitur registrasi pasien merupakan algoritma c4.5 yang diterapkan pada fitur registrasi pasien untuk menentukan waktu penanganan pasien. Seperti yang dijelaskan pada subbab 4.2 algoritma c4.5 digunakan dalam proses *decision tree* yaitu pembangunan *tree*.

1. Request Data

Dijelaskan pada subbab 4.2.1 bahwa proses *request* data yang diterapkan pada ditunjukkan pada menu data *training* dan data *testing*. Tampilan menu data *training* dan data *testing* dapat dilihat pada Gambar 5.1.

No	Nama Pasien	Umur	Jenis Kelamin	Warna Rambut	Penggunaan Kacamata	Warna Kulit	ALAM	SUKSES	PROSES	REKOMENDASI (Sifat Kerja)	REKOMENDASI (Sifat Kerja)	REKOMENDASI (Sifat Kerja)	REKOMENDASI (Sifat Kerja)	REKOMENDASI (Sifat Kerja)	REKOMENDASI (Sifat Kerja)
1	Berli	32	Laki-laki	Ya	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Ya
2	Selwyn	39	Pemempuan	Ya	Ya	Ya	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Ya
3		47	Pemempuan	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak
4	Riz	33	Laki-laki	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
5	Wend	13	Pemempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
6	Sam	24	Laki-laki	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak
7	Kubun	38	Pemempuan	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak
8	Anast	43	Pemempuan	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak
9	Kusari	14	Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak
10	Hassan	48	Pemempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak
11	Safit	23	Laki-laki	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Ya
12	Nad Ward	25	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak
13	Arifa	1	Pemempuan	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak
14	Ri Hasan	38	Laki-laki	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak
15	Shi M	33	Pemempuan	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak
16	Ri Yudi	47	Laki-laki	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak
17	Mahdi	48	Laki-laki	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak
18	Fala	14	Pemempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Tidak	Ya	Tidak	Tidak	Tidak
19	Nari	14	Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak
20	Nad Hafid	14	Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
21	Serani	32	Pemempuan	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
22	Wafa	28	Laki-laki	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
23	Shi	23	Pemempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak
24	Ri Rendi	22	Laki-laki	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak
25	Sofia	48	Pemempuan	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak
26	Muhann	28	Laki-laki	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak
27	Nir	48	Pemempuan	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak

Gambar 5.1 Tampilan Menu data *training* dan data *testing*

2. Proses *learning*

Tahap pertama pada proses *learning* yaitu mendapat data *training*. Data *training* telah dijelaskan pada subbab sebelumnya. Tahap kedua proses ini adalah proses pembangunan *tree*. Pada sistem, proses pembangunan *tree* dilakukan dengan menekan menu proses *mining*. Kemudian sistem akan melakukan proses pembangunan *tree*. Tampilan menu proses *mining* dapat dilihat pada Gambar 5.16.



Gambar 5.2 tampilan menu proses *mining*

Setelah menekan menu proses *mining*, akan muncul *alert*, untuk melanjutkan proses *mining* maka dapat menekan tombol ok. Kemudian sistem akan menjalankan proses *mining* atau proses pembangunan *tree* dengan menggunakan algoritma c4.5. langkah-langkah pada proses tersebut didasarkan pada *flowchart* Gambar 3.5. Langkah pertama adalah menyiapkan data training, kemudian langkah kedua menentukan *node* atau *root*. Untuk menentukan node, proses yang pertama adalah menghitung nilai entropi, menghitung entropi kelas menggunakan persamaan 2.1. Kemudian menghitung entropi per atribut. Berikut potongan kode program yang menunjukkan perhitungan entropi per atribut.

```
// Jika nilai kelas = 0 maka entropi = 0
if ($getJumlahKelasTotalEntropy == 0) {
    $getEntropy = 0
}
// Jika nilai kelas sama, maka entropi = 0
else if ($getJumlahKelasLamaEntropy == $getJumlahKelasSedangEntropy AND $getJumlahKelasLamaEntropy == $getJumlahKelasCepatEntropy) {
    $getEntropy = 0
}
else {
    // Jika nilai kelas != 0, maka hitung rumus entropi:
    $perbandingan_lama = $getJumlahKelasLamaEntropy / $getJumlahKelasTotalEntropy;
    $perbandingan_sedang = $getJumlahKelasSedangEntropy / $getJumlahKelasTotalEntropy;
    $perbandingan_cepat = $getJumlahKelasCepatEntropy / $getJumlahKelasTotalEntropy;

    $rumusEntropy = (-(($perbandingan_lama) * log($perbandingan_lama,2)) + (-(($perbandingan_sedang) * log($perbandingan_sedang,2)) + (-(($perbandingan_cepat) * log($perbandingan_cepat,2))) / 2)
    $getEntropy = round($rumusEntropy, 4) // + angka di belakang koma
}
```

Gambar 5.3 Potongan kode program perhitungan entropi

Setelah menghitung entropi, selanjutnya adalah menghitung *information gain* tiap atribut. Seperti persamaan 2.3. berikut potongan kode program perhitungan *information gain*.


```

// jumlahkan masing2 inf_gain temp @@@@@@
$eq1AtributInfGain = mysql_query("SELECT SUM(inf_gain_temp) as inf_gain FROM gain_uuar WHERE nilai_atribut = '$user'");
while ($rowAtributInfGain = mysql_fetch_array($eq1AtributInfGain)) {
    $getAtributInfGain = $rowAtributInfGain['inf_gain'];
    // echo "atribut = $getAtribut | inf_gain = $getAtributInfGain <BR>";

    if ($getAtributInfGain == 0) {
        $getInfGainFix = 0;
    } else {
        // hitung inf gain
        $getInfGainFix = round(($getEntropy + $getAtributInfGain),4);
    }
}

```

Gambar 5.4 Potongan kode program perhitungan *information gain*

Langkah berikutnya adalah menghitung nilai *split-info* dan *gain-ratio* tiap atribut dengan menggunakan persamaan 2.4 dan 2.5.

```

// rumus split info
$getSplitInfo = (($getJumlahKasusTotalEntropy / $getJumlahKasusTotalInfGain) * (log(($getJumlahKasusTotalEntropy / $getJumlahKasusTotalInfGain),2)));

```

Gambar 5.5 Potongan kode program perhitungan *split-info*

```

// rumus gain ratio
$getGainRatio = round(($rowGainRatio['inf_gain'] / $rowGainRatio['split_info']),4);

```

Gambar 5.6 Potongan kode program perhitungan *gain-ratio*

Langkah selanjutnya adalah mencari atribut yang memiliki nilai *gain-ratio* tertinggi. Berikut potongan kode program yang menunjukkan pencarian *gain-ratio* tertinggi.

```

function getInfGainMax($atribut, $nilai_atribut)
{
    // select inf gain max
    $eqInfGainMaxAtribut = mysql_query("SELECT MAX(inf_gain) FROM mining_c45 WHERE gain_ratio in (SELECT max(gain_ratio) FROM 'mining_c45' WHERE inf_gain = 0) LIMIT 1");
    while ($rowInfGainMaxAtribut = mysql_fetch_array($eqInfGainMaxAtribut)) {
        $inf_gain_max_atribut = $rowInfGainMaxAtribut['inf_gain'];
        if (empty($atribut) AND empty($nilai_atribut)) {
            // jika atribut kosong, browses atribut dan inf gain max pada fungsi loopingMiningC45()
            loopingMiningC45($inf_gain_max_atribut);
        } else if (empty($atribut) AND empty($nilai_atribut)) {
            // proses rekursi dan inf gain max pada fungsi loopingMiningC45()
            loopingMiningC45($inf_gain_max_atribut);
        }
    }
}

```

Gambar 5.7 Potongan kode program pencarian *gain-ratio* tertinggi

Setelah didapatkan atribut dengan *gain-ratio* tertinggi maka atribut tersebut menjadi *node*. Kemudian *node* tersebut di *split* per kelas dan di cek apakah sudah murni atau belum. Jika *node* sudah murni maka proses berhenti dan membentuk pohon keputusan, sedangkan jika *node* belum murni maka sistem akan melakukan *looping* proses diatas hingga semua node murni atau semua atribut telah digunakan

berdasar pada aturan *if-then*. *Rule* yang dihasilkan sistem dapat dilihat pada Gambar 5.18.

```

Rule:
1. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Tidak AND kabur_lihat_jauh == Tidak AND usia_temp == > 48) then Sedang (id = 6)
2. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Tidak AND kabur_lihat_jauh == Tidak AND usia_temp == <= 48) then Cepat (id = 7)
3. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Tidak AND kabur_lihat_jauh == Ya AND usia_temp == > 15) then Lama (id = 8)
4. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Tidak AND kabur_lihat_jauh == Ya AND usia_temp == <= 15) then Sedang (id = 10)
5. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Ya AND mata_berair == Tidak) then Cepat (id = 12)
6. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Ya AND mata_berair == Ya) then Cepat (id = 13)
7. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Ya AND mata_merah == Tidak AND jenis_kelamin == Laki-laki) then Sedang (id = 16)
8. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Ya AND mata_merah == Tidak AND jenis_kelamin == Perempuan) then Sedang (id = 17)
9. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Ya AND mata_merah == Ya) then Sedang (id = 18)
10. if (pandangan_sempit == Tidak AND silau == Ya AND jenis_kelamin == Laki-laki) then Cepat (id = 20)
11. if (pandangan_sempit == Tidak AND silau == Ya AND jenis_kelamin == Perempuan) then Cepat (id = 21)
12. if (pandangan_sempit == Ya AND jenis_kelamin == Laki-laki) then Sedang (id = 23)
13. if (pandangan_sempit == Ya AND jenis_kelamin == Perempuan) then Sedang (id = 24)

```

Gambar 5.10 *Rule* yang dihasilkan sistem

Terdapat dua belas *rule* yang dihasilkan oleh proses pembangunan *tree*. Berikut *rule* yang didapat, yang nantinya akan digunakan pada proses prediksi pada fitur registrasi pasien.

if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Tidak AND kabur_lihat_jauh == Tidak AND usia_temp == >

48) then **Sedang**

if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Tidak AND kabur_lihat_jauh == Tidak AND usia_temp == <=

48) then **Cepat**

if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Tidak AND kabur_lihat_jauh == Ya AND usia_temp == > 15) then **Lama**

if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Tidak AND kabur_lihat_jauh == Ya AND usia_temp == <=

15) then **Sedang**

if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Ya AND mata_berair == Tidak) then **Cepat**

if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Ya AND mata_berair == Ya) then **Cepat**

if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Ya AND mata_merah == Tidak AND jenis_kelamin == Laki-laki) then **Sedang**

if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Ya AND

Gambar 5.20 merupakan tampilan menu perbandingan yang merupakan hasil dari proses *testing*. Pada tabel tersebut terdapat data *testing*, keputusan asli dari data *testing* serta keputusan atau kelas baru yang didapat dari *rule* yang terbentuk.

Proses pengklasifikasian atau *testing* dilakukan untuk mengukur kinerja atau mengevaluasi dari proses pembentukan *tree* dengan menggunakan algoritma c4.5. Untuk mengevaluasi hasil *testing*, penulis menggunakan *confussion matrix*. Data set yang digunakan dalam pembangunan *tree* untuk menentukan waktu penanganan pasien memiliki tiga kelas yaitu cepat, sedang, dan lama. Maka *confussion matrix* hasil klasifikasi waktu penanganan pasien dapat dilihat pada tabel 5.2.

Tabel 5.1 *Confussion Matrix* Hasil Klasifikasi Waktu Penanganan Pasien

Correct Classified / Keputusan Asli / Kelas Asli	Classied As / Keputusan C4.5 / Kelas Baru		
	Cepat	Sedang	Lama
Cepat	100	3	0
Sedang	40	6	2
Lama	7	3	0

Sumber : (Hasil Analisis, 2015)

Tabel 5.2 merupakan *confussion matrix* hasil klasifikasi waktu penanganan pasien. Jumlah *record* yang memiliki keputusan asli dan keputusan c4.5 benar ada 106 data, sedangkan *record* yang memiliki keputusan asli dan keputusan c4.5 berbeda ada 55 data. Dengan mengevaluasi *decision tree* menggunakan *confussion matrix* akan menghasilkan nilai *accuracy* dan *error rate*.

Berdasarkan persamaan 2.6 dan persamaan 2.7, berikut adalah hasil perhitungan *accuracy* dan *error rate* untuk evaluasi hasil klasifikasi waktu penanganan pasien.

$$Accuracy = \frac{100+6+0}{100+6+0+40+7+3+3+0+2} \times 100\% = \frac{106}{161} \times 100\% = 65.8\%$$

$$Error\ rate = \frac{3+0+2+40+7+3}{100+6+0+40+7+3+3+0+2} \times 100\% = \frac{55}{161} \times 100\% = 34.2\%$$

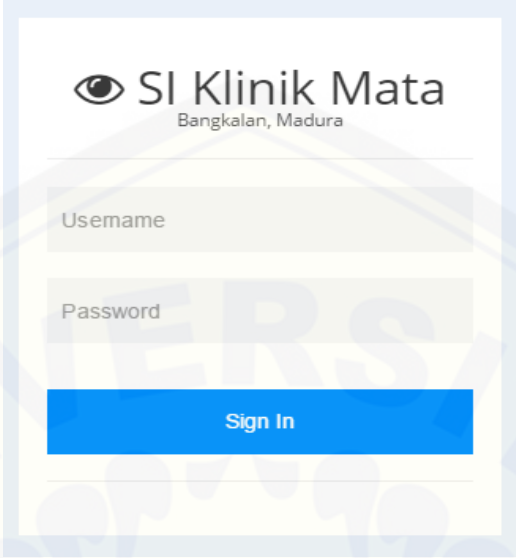
Berdasarkan hasil perhitungan diatas, nilai *accuracy* adalah 65.8% yang berarti bahwa jumlah *record* data yang diklasifikasikan dengan benar adalah 65.8 %. Sedangkan nilai error adalah 34.2%. Kesalahan prediksi tertinggi berada pada keputusan asli ‘sedang’ dan keputusan c4.5 cepat yaitu mencapai 72,7 % dari 34.2% kesalahan. Hal tersebut bukan kesalahan yang fatal karena pasien akan mendapatkan waktu pemeriksaan sepuluh menit lebih cepat sehingga tidak akan merugikan proses antrian pasien. Sedangkan apabila terjadi kesalahan sebaliknya atau seperti keputusan asli ‘cepat’ dan keputusan c4.5 ‘lama’ maka akan merugikan proses antrian pasien karena waktu yang didapatkan pasien menjadi sepuluh menit atau tiga puluh menit lebih lama.

5.2. Implementasi Sistem

Implementasi sistem merupakan tahap pengkodean sistem berdasarkan pada perancangan yang telah dibuat ke dalam bahasa pemrograman. Penulis melakukan pengkodean menggunakan bahasa pemrograman PHP, HTML, CSS, dan *Javascript*. Tahap pengkodean menghasilkan beberapa tampilan atau *interface*. Berikut beberapa implementasi yang dibuat oleh penulis.

5.2.1 Login

Tampilan menu login ini digunakan untuk setiap user agar dapat mengakses fitur yang telah disediakan sesuai dengan hak akses masing-masing user. Sistem informasi klinik mata dapat diakses oleh karyawan dan pasien. Tampilan menu login karyawan berbeda dengan tampilan menu login pasien. Tampilan login karyawan dapat dilihat pada Gambar 5.1.



SI Klinik Mata
Bangkalan, Madura

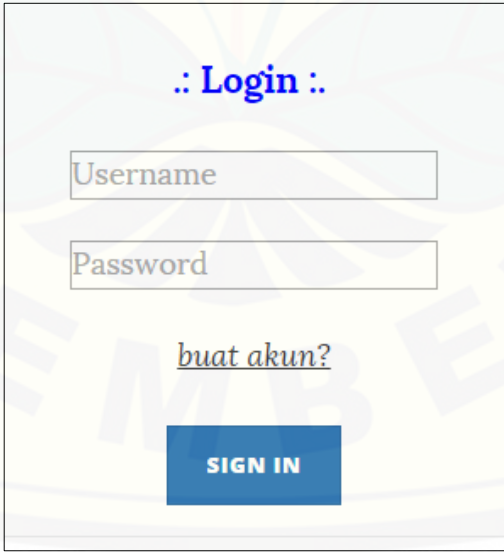
Username

Password

Sign In

Gambar 5.12 Tampilan menu *login* karyawan

Gambar 5.7 merupakan tampilan menu *login* pada sistem informasi klinik mata untuk karyawan. Terdapat *form login* berisi *username* dan *password* yang dapat diisi dengan *username* dan *password* yang telah dimiliki *user* serta tombol *sign in* untuk melakukan proses *login*. Tampilan menu *login* karyawan berbeda dengan tampilan menu *login* pasien. Tampilan *login* pasien dapat dilihat pada Gambar 5.13.



:: Login ::

Username

Password

[buat akun?](#)

SIGN IN

Gambar 5.13 Tampilan menu *login* pasien

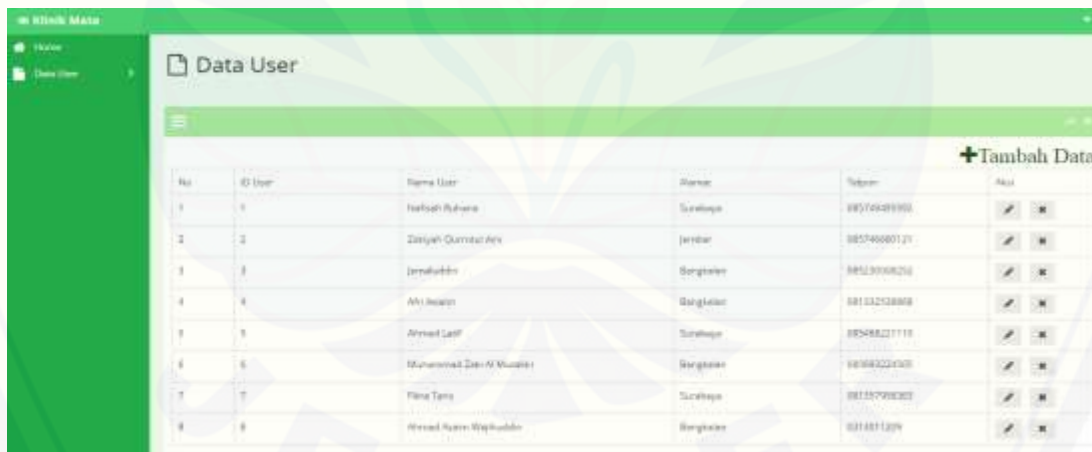
Gambar 5.13 merupakan tampilan menu *login* pada sistem informasi klinik mata untuk pasien. Terdapat form *login* berisi *username* dan *password* yang dapat diisi dengan *username* dan *password* yang telah dimiliki *user* serta tombol *sign in* untuk melakukan proses *login*.

5.2.2 Implementasi Tahap *Increment* 1

Implementasi tahap *increment* satu merupakan tahap pengkodean fitur-fitur yang dirancang pada tahap *increment* satu yaitu fitur data user, buat akun, aktivasi akun lewat email, data pasien, registrasi pasien baru, dan registrasi pasien lama. Berikut penjelasan implementasi tahap *increment* satu.

A. Data User

Tampilan menu data user ini hanya dapat diakses oleh admin untuk menambah, mengedit, serta menghapus data user. Tampilan menu data user dapat dilihat pada Gambar 5.14.



No	ID User	Nama User	Alamat	Nomor	Aksi
1	1	Herfandi Rahana	Surabaya	88576489900	[Edit] [Hapus]
2	2	Zakiyah Qurmatul Aini	Jember	88574680121	[Edit] [Hapus]
3	3	Jemakudin	Bangkalan	88573986200	[Edit] [Hapus]
4	4	Ahmad Asyraf	Bangkalan	88132108888	[Edit] [Hapus]
5	5	Ahmad Lutfi	Surabaya	88548822110	[Edit] [Hapus]
6	6	Muhammad Zam-Ah Nuzairi	Bangkalan	88583221000	[Edit] [Hapus]
7	7	Rina Tama	Surabaya	88155720888	[Edit] [Hapus]
8	8	Muhammad Nuzairi	Bangkalan	8818811000	[Edit] [Hapus]

Gambar 5.14 Tampilan menu data *user*

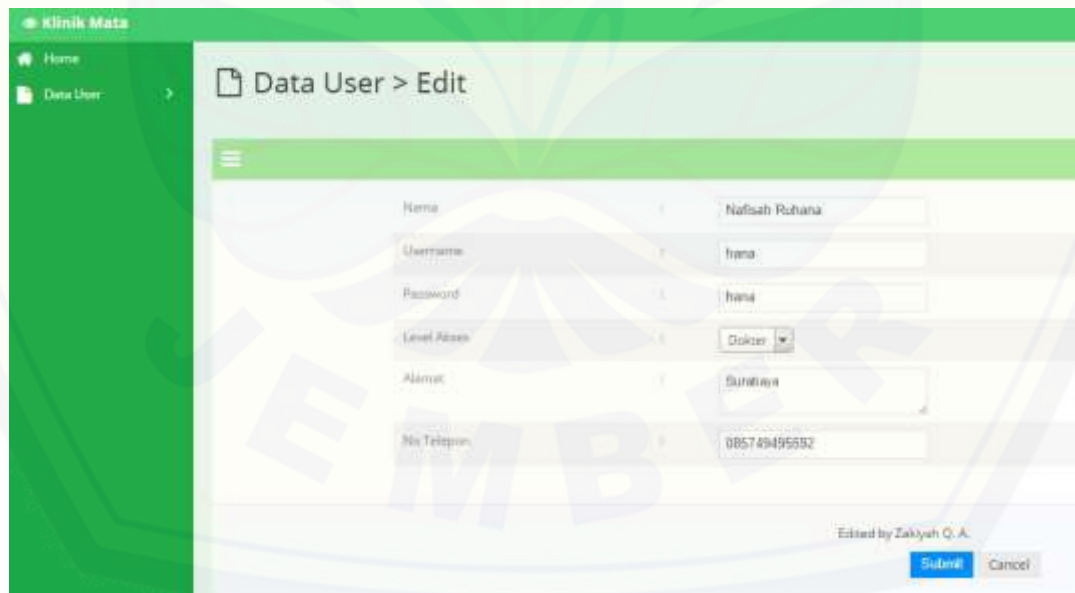
Gambar 5.14 merupakan tampilan menu data user yang berisi tabel data user, tombol tambah data, edit, dan hapus. Untuk menambah data *user*, admin dapat menekan tombol tambah data dan kemudian sistem akan menampilkan *form* tambah data *user*. Tampilan *form* tambah data *user* dapat dilihat pada gambar 5.4.



The screenshot shows a web application interface for 'Klinik Mata'. The main content area is titled 'Data User > Tambah'. It contains a form with the following fields: 'Nama' (Name), 'Username', 'Password', 'Level Akses' (a dropdown menu currently showing 'Dokter'), 'Alamat', and 'No Telepon'. At the bottom right of the form, there are two buttons: 'Submit' and 'Cancel'.

Gambar 5.15 tampilan *form* tambah data *user*

Gambar 5.15 merupakan tampilan *form* tambah data *user* yang berisi nama, username, password, level user, alamat, no telepon, dan tombol *submit* untuk melakukan proses menambahkan data ke dalam *database*. *User* admin juga dapat mengedit data *user* dengan menekan tombol edit pada tabel data *user* yang ingin di *edit* datanya. Tampilan *form edit* data *user* dapat dilihat pada Gambar 5.16.



The screenshot shows the 'Data User > Edit' form. The fields are pre-filled with the following information: 'Nama' (Nafisah Rohana), 'Username' (nana), 'Password' (nana), 'Level Akses' (Dokter), 'Alamat' (Surabaya), and 'No Telepon' (085749495552). The 'Submit' button is highlighted in blue. At the bottom right, there is a small text 'Edited by Zakyah Q. A.' and a 'Cancel' button.

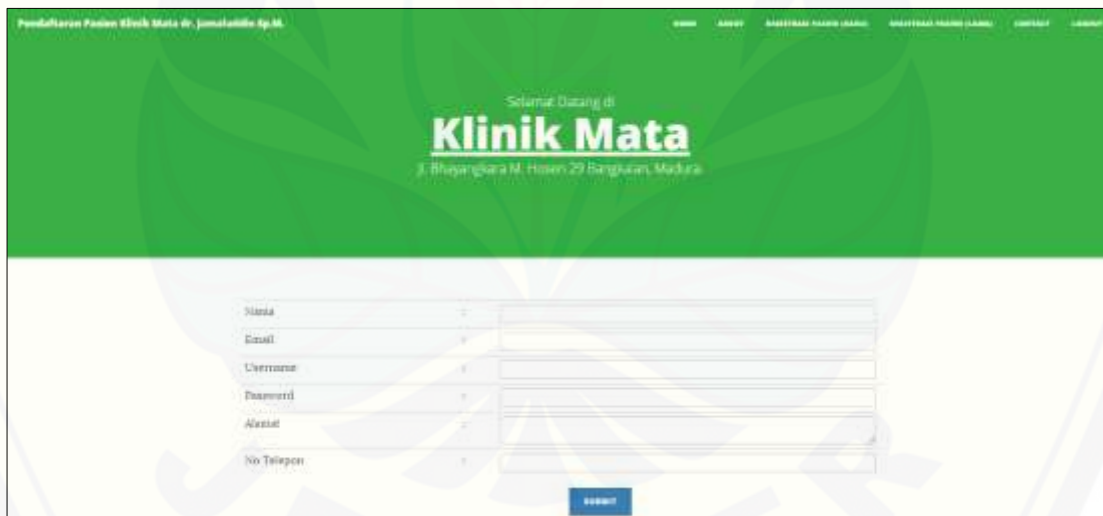
Gambar 5.16 tampilan *form edit* data *user*

Gambar 5.11 merupakan tampilan *form* edit data *user*. *User* admin hanya mengedit bagian yang ingin di *edit* kemudian klik *submit* agar sistem melakukan proses *update* pada *database*.

Tombol hapus pada menu data *user* adalah tombol untuk menghapus data *user* yang ingin dihapus. Dengan menekan tombol hapus maka sistem akan menampilkan *alert* berisi “Apakah anda yakin ingin menghapus data ini?”, jika admin ingin menghapus maka menekan tombol *ok* pada *alert* dan sistem akan menghapus data pada *database* dan mengembalikan ke tampilan menu data *user*.

B. Buat Akun

Tampilan *form* buat akun merupakan menu dari sistem informasi klinik mata yang hanya dapat di akses oleh pasien. Tampilan *form* buat akun dapat dilihat pada Gambar 5.12.



The image shows a web form for creating an account on the 'Klinik Mata' website. The form is titled 'Buat Akun' and is located on a page with a green header. The header contains the text 'Selamat Datang di Klinik Mata' and the address 'J. Bhayangkara M. Hosen 29 Bangunan, Madura'. The form fields are: Nama, Email, Username, Password, Alamat, and No Telepon. A blue 'submit' button is located at the bottom right of the form.

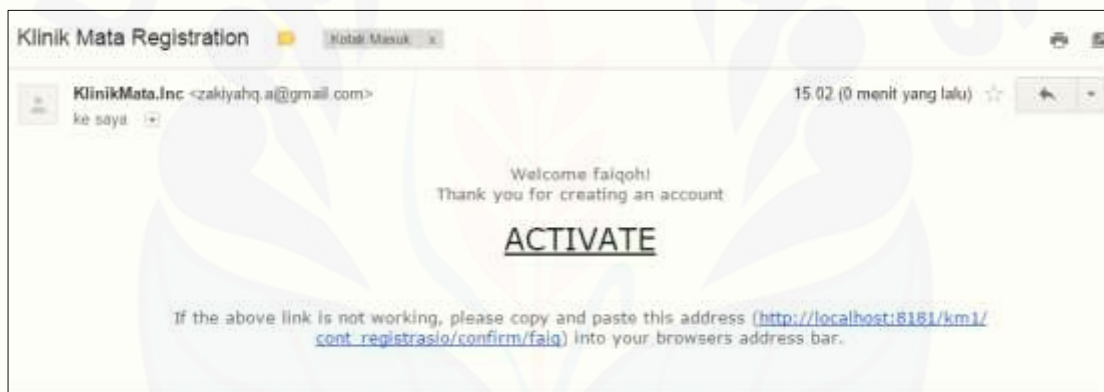
Gambar 5.17 Tampilan *form* buat akun

Gambar 5.12 merupakan tampilan *form* buat akun yang berisi nama, *email*, *username*, *password*, alamat, no telepon dan tombol *submit* untuk membuat akun. Dengan menekan tombol *submit* maka sistem akan mengirim *email* sebagai aktivasi akun agar dapat sukses melakukan *login* ke sistem. *Username* pada *form* buat akun

merupakan atribut yang unik, setiap *user* tidak boleh memiliki *username* yang sama. Jika *username* telah digunakan oleh *user* lain maka sistem tidak akan menyimpan data maupun mengirim pesan aktivasi dan akan mengembalikan halaman ke tampilan buat akun.

C. Aktivasi Akun Lewat *Email*

Aktivasi akun lewat email merupakan salah satu fitur pada sistem informasi klinik mata yang dapat diakses oleh pasien. Fitur ini berguna untuk mengaktifkan akun pasien yang telah didaftarkan sehingga pasien dapat login ke sistem. Pesan aktivasi dikirimkan pada saat user menekan tombol submit pada fitur buat akun. Contoh pesan aktivasi yang dikirimkan ke email user dapat dilihat pada Gambar 5.13.




Gambar 5.18 Contoh pesan aktivasi *email*

Gambar 5.13 merupakan contoh pesan aktivasi yang dikirimkan oleh sistem ke *email user*. Agar dapat mengaktifasi akun, *user* dapat menekan *link* 'Aktivasi' seperti contoh pada gambar. Lalu sistem akan menampilkan halaman *login* yang berisi pesan bahwa *username user* telah aktif.

D. Data Pasien

Tampilan menu data pasien ini hanya dapat diakses oleh resepsionis dan dokter untuk menambah, mengedit, serta melihat data pasien. Tampilan menu data pasien dapat dilihat pada Gambar 5.14.



No	ID Pasien	Nama Pasien	Jenis Kelamin	Alamat	No Telp
1	001	Andi	A	Jember	081 234 567
2	002	Budi	P	Jember	081 234 567
3	003	Cici	P	Jember	081 234 567
4	004	Dina	P	Jember	081 234 567
5	005	Eko	P	Jember	081 234 567
6	006	Fani	P	Jember	081 234 567
7	007	Gani	P	Jember	081 234 567
8	008	Hani	P	Jember	081 234 567
9	009	Ibu	P	Jember	081 234 567
10	010	Joni	P	Jember	081 234 567
11	011	Kiki	P	Jember	081 234 567
12	012	Lili	P	Jember	081 234 567
13	013	Mami	P	Jember	081 234 567
14	014	Nani	P	Jember	081 234 567
15	015	Oti	P	Jember	081 234 567

Gambar 5.19 Tampilan menu data pasien

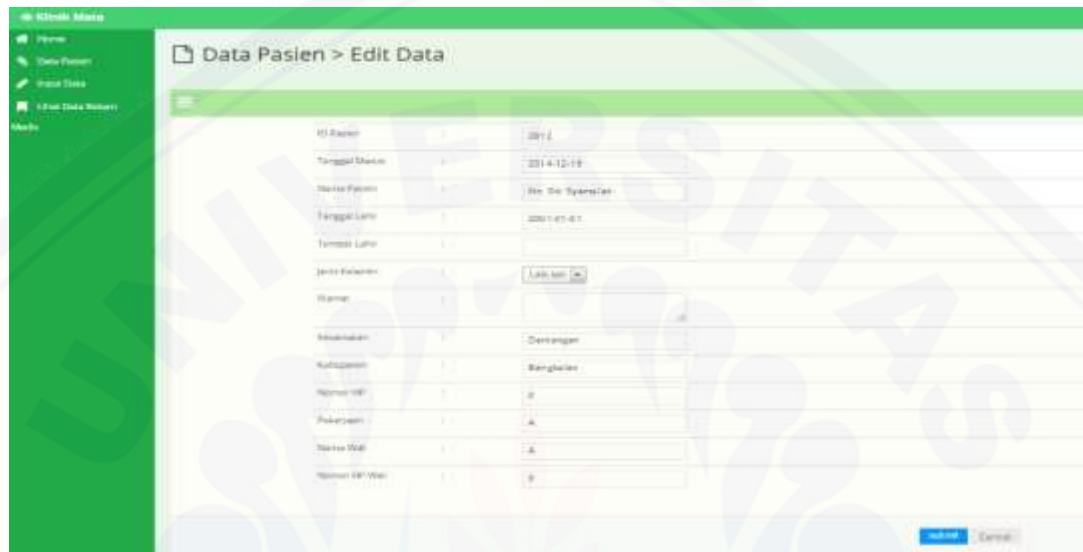
Gambar 5.14 merupakan tampilan menu data pasien yang berisi tabel data pasien, tombol tambah data, edit, dan *view*. Untuk menambah data pasien, *user* dapat menekan tombol tambah data dan kemudian sistem akan menampilkan *form* tambah data pasien. Tampilan *form* tambah data pasien dapat dilihat pada Gambar 5.15.



Gambar 5.20 Tampilan *form* tambah data pasien

Gambar 5.15 merupakan tampilan *form* tambah data pasien yang berisi Id pasien, tanggal masuk, nama pasien, tanggal lahir, tempat lahir, jenis kelamin, alamat, kecamatan, kabupaten, nomor telepon, pekerjaan, nama wali, nomor telepon wali, dan tombol *submit* untuk melakukan proses menambahkan data ke dalam *database*. *User*

juga dapat mengedit data pasien dengan menekan tombol edit pada tabel data pasien yang ingin di *edit* datanya. Tampilan *form edit* data pasien dapat dilihat pada Gambar 5.16.



ID Pasien	2014
Tanggal Masuk	2014-12-18
Nama Pasien	Mr. Dr. Syamsul
Tanggal Lahir	2001-01-01
Tempat Lahir	
Jenis Kelamin	Laki-laki
Nama	
Alamat	Delem
Pekerjaan	Bengkel
Nomor HP	0
Nomor ID	A
Nomor ID	A
Nomor ID	B

Gambar 5.21 Tampilan *form edit* data pasien

Gambar 5.17 merupakan tampilan *form edit* data pasien. *User admin* hanya mengedit bagian yang ingin di *edit* kemudian klik *submit* agar sistem melakukan proses *update* pada *database*.

Tombol *view* pada menu data pasien adalah tombol untuk melihat detail data pasien. Tampilan *view detail* pasien dapat dilihat pada Gambar 5.18.

:: DATA PASIEN ::	
ID Pasien	: 8705
Tanggal Masuk	: 2014-12-15
Nama Pasien	: Ny. Asmi'ah
Tanggal Lahir	: 1968-01-01
Tempat Lahir	: Bangkalan
Jenis Kelamin	: Perempuan
Alamat	: A
Kecamatan	: Geger
Kabupaten	: Bangkalan
Nomor HP	: 0
Pekerjaan	: A
Nama Wali	: A
Nomor HP Wali	: 0

Close

Edited by Zakiyah Q. A.

Gambar 5.22 Tampilan *view detail* data pasien

E. Registrasi Pasien Baru

Tampilan menu registrasi pasien baru berisi *form* registrasi bagi pasien yang belum memiliki id pasien. Menu ini hanya dapat di akses oleh pasien. Tampilan form registrasi pasien baru dapat dilihat pada Gambar 5.19.

Pendaftaran Pasien Klinik Mata di: jember@klinikmata.com

Selamat Datang di
Klinik Mata
Jl. Dhanyangjara M. Jombang 29 Bangkalan, Madura

Data Pasien *input user*

ID Pasien	<input type="text"/>
Tanggal Masuk	<input type="text" value="01/01/2014"/>
Nama Pasien	<input type="text"/>
Tanggal Lahir	<input type="text"/>
Tempat Lahir	<input type="text"/>
Jenis Kelamin	<input type="text" value="Laki-laki"/>

Gambar 5.23 Tampilan *form* registrasi pasien baru

Gambar 5.18 merupakan tampilan form registrasi pasien baru yang berisi data pasien beserta data registrasi untuk mendapatkan antrian.

F. Registrasi Pasien Lama

Tampilan menu registrasi pasien lama berisi *form* registrasi bagi pasien yang telah memiliki id pasien. Menu ini dapat di akses oleh resepsionis dan pasien. Tampilan form registrasi pasien lama untuk user resepsionis berbeda dengan tampilan untuk *user* pasien. Tampilan *form* registrasi pasien lama untuk *user* resepsionis dapat dilihat pada Gambar 5.19.

Gambar 5.24 tampilan *form* registrasi pasien lama untuk resepsionis

Gambar 5.19 merupakan tampilan *form* registrasi pasien lama untuk resepsionis yang berisi *form* registrasi dan tabel daftar antrian hari ini. Pada tabel terdapat kolom sudah yang berfungsi pada saat pasien telah masuk untuk diperiksa maka resepsionis menekan tombol *check* untuk menghilangkan data dari daftar antrian. Sedangkan tampilan registrasi pasien lama untuk pasien dapat dilihat pada Gambar 5.20.



Gambar 5.25 tampilan *form* registrasi pasien lama untuk pasien

Gambar 5.20 merupakan tampilan *form* registrasi pasien untuk pasien yang telah memiliki id. Pada *form* registrasi pasien terdapat *field* keluhan yang berguna untuk menginputkan keluhan apa saja yang diderita oleh pasien pada saat registrasi yang kemudian akan digunakan untuk proses prediksi waktu penanganan pasien menggunakan *rule* yang telah dihasilkan oleh proses *learning* algoritma c4.5. Proses prediksi yaitu dengan mencocokkan data yang dimasukkan pada form registrasi dengan dua belas *rule* yang dihasilkan dari algoritma c4.5, dicocokkan satu persatu dari *rule* yang pertama hingga *rule* terakhir, jika data yang di masukkan cocok dengan *rule* yang pertama maka kelas waktu dari data yang dimasukkan sama dengan kelas dari *rule* pertama, jika tidak, maka data akan dicocokkan dengan *rule* selanjutnya. Proses implementasi algoritma c4.5 akan dijelaskan pada subbab 5.1.

5.2.3 Implementasi Tahap *Increment 2*

Implementasi tahap *increment* dua merupakan tahap pengkodean fitur-fitur yang dirancang pada tahap *increment* dua yaitu fitur data obat, data supplier, transaksi penjualan obat, transaksi pembelian obat, dan daftar obat habis. Tampilan fitur-fitur tahap *increment* dua dapat dilihat pada lampiran B.1.

5.2.4 Implementasi Tahap *Increment 3*

Implementasi tahap *increment* tiga merupakan tahap pengkodean fitur-fitur yang dirancang pada tahap *increment* dua yaitu fitur data *frame*, transaksi penjualan kacamata, dan transaksi pelunasan kacamata. Tampilan fitur-fitur tahap *increment* tiga dapat dilihat pada lampiran B.2.

5.2.5 Implementasi Tahap *Increment 4*

Implementasi tahap *increment* empat merupakan tahap pengkodean fitur-fitur yang dirancang pada tahap *increment* empat yaitu fitur input data pemeriksaan, lihat data rekam medis, pendaftaran operasi, dan lihat jadwal operasi. Tampilan fitur-fitur tahap *increment* empat dapat dilihat pada lampiran B.3.

5.3. Pengujian Sistem

Pengujian atau *testing* sistem dilakukan dengan *blackbox testing* dan *whitebox testing*. Pengujian dilakukan di setiap tahap *increment*.

5.3.1 *Blackbox Testing*

Blackbox testing merupakan pengujian program yang melihat dari segi fungsional tanpa melihat desain dan kode program. Pengujian ini dilakukan oleh *user*

yang merupakan karyawan dari klinik mata yang menjadi tempat penelitian. Blackbox Testing dapat dilihat pada lampiran D.

5.3.2 Whitebox Testing

White-box testing merupakan pengujian program dengan melihat kode program. Pengujian *whitebox* dilakukan dengan metode *cyclomatic complexity* untuk mengetahui tingkat kompleksitas program. Berikut pengujian program tahap *increment* satu yaitu fitur registrasi pasien lama.

A. Controller Registrasi()

1. Listing Program

Gambar 5.21 merupakan potongan *listing* program *controller* registrasi pasien lama.

```

function registrasii(){
457     $input = array(
458     1     "no_registrasi" => $this->input->post('no_registrasi'),
459         "tgl_registrasi" => date('Y-m-d', strtotime($this->input->post('tgl_registrasi'))),
460         "id_pasien" => $this->input->post('id_pasien')
461     );
462
463     2     $keluhan = $this->input->post('keluhan');
464
465     3     $this->m_registrasi->prediksi($input, $keluhan);
466
467
468     4     redirect(base_url() . "index.php/cont/registrasi?status=sukses");
469 }

```

Gambar 5.26 Potongan Listing Program Fitur Registrasi Pasien Lama Bagian *Controller*

2. Grafik Alir

Menggambaran grafik alir *controller* fitur registrasi pasien lama berdasarkan *listing program* di atas. Grafik alir bisa dilihat pada Gambar 5.22 di bawah ini.



Gambar 5.27 Grafik Alir *Controller* Registrasi Pasien Lama
Sumber : (Hasil Analisis, 2015)

3. Kompleksitas Siklomatis dari grafik alir dapat diperoleh dengan perhitungan :
$$V(G) = E - N + 2 = 3 - 4 + 2 = 1$$
4. *Basis set* yang dihasilkan dari jalur independen secara linier adalah satu jalur, yaitu :
Jalur 1 : 1-2-3-4

Test case, user mengisi data registrasi kemudian menekan tombol submit. Hasil pengujian *path* jalur 1 (1-2-3-4), dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan *software*, sistem ini telah memenuhi syarat.

B. Model Prediksi()

1. Listing Program

Gambar 5.23 merupakan potongan *listing* program model registrasi pasien lama.

```

23 function prediksi($input, $keluhan) {
24     $nilai = array();
25     $arr_keluhan = array();
26     $atr = $this->db->query('select nama_atribut from tb_jenisatribut '
27         . 'where nama_atribut <> "usia" AND nama_atribut <> "jenis_kelamin" ');
28     $result = $atr->result_array();
29     foreach ($result as $atribut) {
30         3 $sama = false;
31         foreach ($keluhan as $k) {
32             5 if ($atribut["nama_atribut"] == $k) {
33                 $sama = true;
34             }
35         }
36         8 if ($sama) {
37             9 $arr_keluhan[$atribut["nama_atribut"]] = "Ya";
38         } else {
39             10 $arr_keluhan[$atribut["nama_atribut"]] = "Tidak";
40         }
41     }
42     //ambil usia
43     $umurjk = $this->db->query('SELECT id_pasien,
44         YEAR(CURDATE()) - YEAR(tbl_lahir) as umur, jenis_kelamin
45         FROM tb_pasien where id_pasien=' . $input['id_pasien']);
46     $result1 = $umurjk->row_array();
47     13 $arr_keluhan["usia_temp"] = $result1['umur'];
48     $arr_keluhan["jenis_kelamin"] = $result1['jenis_kelamin'];
49     $rule = $this->db->get("rule_c45");
50     $result_rule = $rule->result_array();
51     $hasil = "";
52     14 foreach ($result_rule as $rrule) {
53         $id_rule = $rrule["id"];
54         15 $group_rule = $this->db->get_where("rule_penentu_keputusan", array("id_rule" => $id_rule));
55         $result_group = $group_rule->result_array();
56         $cocok = true;
57         16 foreach ($result_group as $group) {
58             17 foreach ($arr_keluhan as $keluh => $nilai) {
59                 18 if ($group["atribut"] == $keluh) {
60                     19 if ($group["atribut"] == "usia_temp") {
61                         $rule_usia = explode(" ", $group["nilai_atribut"]);
62                         20 $tanda = $rule_usia[0];
63                         $usia = $rule_usia[1];
64                     }
65                     21 if ($tanda == ">") {
66                         22 if ($nilai <= $usia) {
67                             $cocok = false;
68                             break;
69                         }
70                     }
71                     25 else if ($tanda == "<=") {
72                         26 if ($nilai > $usia) {
73                             $cocok = false;
74                             break;
75                         }
76                     }
77                     29 } else {
78                         30 if ($group["nilai_atribut"] != $nilai) {
79                             $cocok = false;
80                             break;
81                         }
82                     }
83                 }
84             }
85             34 if ($cocok) {
86                 $hasil = $rrule["keputusan"];
87                 break;
88             }
89         }
90     }
91     37 if ($hasil == "") {
92         $hasil = "Cepat";
93     }
94     39 }

```



```

90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120

```

```

$selectAntrian = $this->db->query("select selesai as mulai from "
40 . "tb_registrasi where tgl_registrasi = '". $input['tgl_registrasi']
. "' order by no_registrasi desc limit 1");
$mulai = $selectAntrian->row_array();

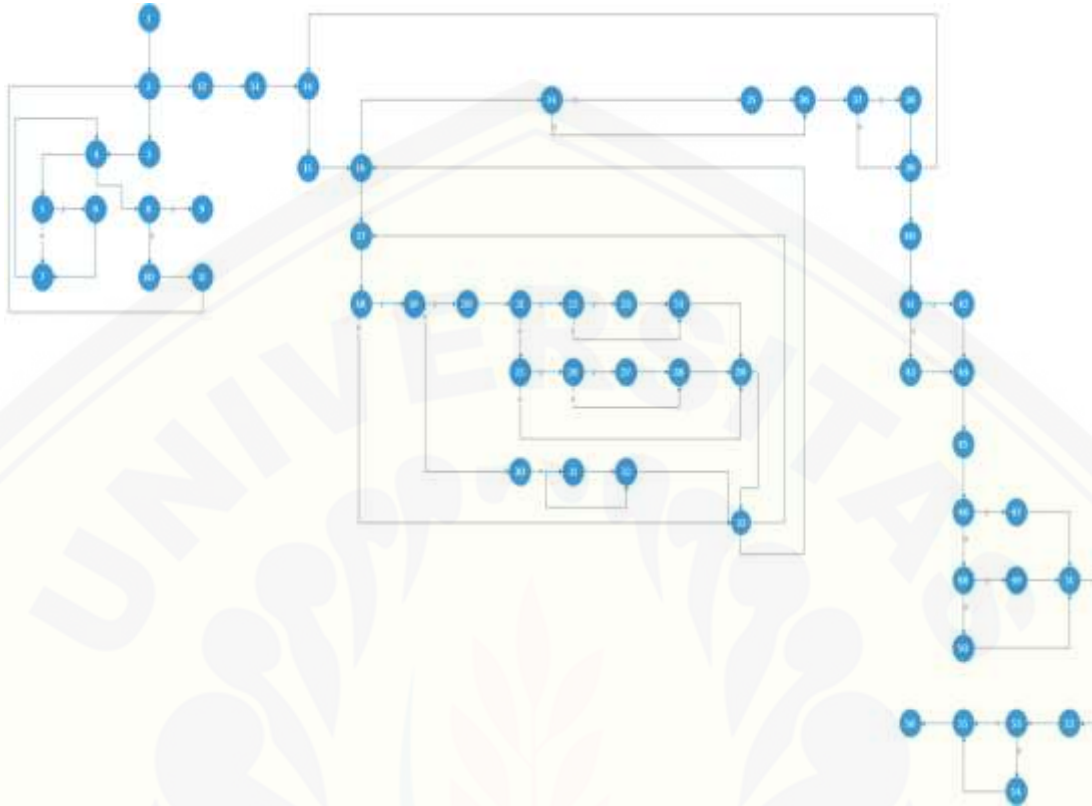
41 if ($mulai['mulai'] == NULL) {
    $waktuMulai = "17:00:00";42
} else {
43 $waktuMulai = $mulai['mulai'];44
}
45 $waktuSelesai = explode(":", $waktuMulai);
$tambahMenit = 0;
46 if ($hasil == "Cepat") {
47 $tambahMenit = 10;
48 } elseif ($hasil == "Sedang") {
49 $tambahMenit = 20;
} else { //ini kalo lama
50 $tambahMenit = 30;51
}
52 $waktuSelesai[1] += $tambahMenit;
53 if (($waktuSelesai[1] >= 60) {
54 $waktuSelesai[0] ++;
    $waktuSelesai[1] -= 60;
55 }
$Selesai = implode(":", $waktuSelesai);
56 $this->db->insert("tb_registrasi", array("no_registrasi" => $input['no_registrasi'],
"tgl_registrasi" => $input['tgl_registrasi'], "id_pasien" => $input['id_pasien'],
"mulai" => $waktuMulai , "selesai" => $Selesai,
"status" => "Waiting" ));

```

Gambar 5.28 Potongan *Listing* Program Model Fitur Registrasi Pasien Lama

2. Grafik Alir

Menggambarkan grafik alir model fitur registrasi pasien lama berdasarkan *listing program* di atas. Grafik alir bisa dilihat pada Gambar 5.24 di bawah ini.



Gambar 5.29 Grafik Alir *Controller* Registrasi Pasien Lama
Sumber : (Hasil Analisis, 2015)

3. Kompleksitas Siklomatis dari grafik alir dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 = 74 - 56 + 2 = 22$$
4. Basis set yang dihasilkan dari jalur independen secara linier adalah satu jalur, yaitu :
 Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 – 16
 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 29 – 33 – 17 – 34 – 35 – 36
 – 37 – 38 – 39 – 40 – 41 – 42 – 44 – 45 – 46 – 47 – 51 – 52 – 53 – 54
 – 55 – 56
 Jalur 2 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 – 16
 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 29 – 33 – 17 – 34 – 35 – 36

– 37 – 38 – 39 – 40 – 41 – 42 – 44 – 45 – 46 – 47 – 51 – 52 – 53 – 55
– 56

Jalur 3 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 – 16
– 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 29 – 33 – 17 – 34 – 35 – 36
– 37 – 38 – 39 – 40 – 41 – 42 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53
– 54 – 55 – 56

Jalur 4 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 – 16
– 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 29 – 33 – 17 – 34 – 35 – 36
– 37 – 38 – 39 – 40 – 41 – 42 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53
– 55 – 56

Jalur 5 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 – 16
– 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 29 – 33 – 17 – 34 – 35 – 36
– 37 – 38 – 39 – 40 – 41 – 42 – 44 – 45 – 46 – 48 – 50 – 51 – 52 – 53
– 54 – 55 – 56

Jalur 6 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 – 16
– 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 29 – 33 – 17 – 34 – 35 – 36
– 37 – 38 – 39 – 40 – 41 – 42 – 44 – 45 – 46 – 48 – 50 – 51 – 52 – 53
– 55 – 56

Jalur 7 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 – 16
– 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 29 – 33 – 17 – 34 – 35 – 36
– 37 – 38 – 39 – 40 – 41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53
– 54 – 55 – 56

Jalur 8 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 – 16
– 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 29 – 33 – 17 – 34 – 35 – 36
– 37 – 38 – 39 – 40 – 41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53
– 55 – 56

Jalur 9 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 – 16
– 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 29 – 33 – 17 – 34 – 36 – 37

– 38 – 39 – 40 – 41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53 – 54
– 55 – 56

Jalur 10 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 –
16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 29 – 33 – 17 – 34 – 36 –
37 – 38 – 39 – 40 – 41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53 –
55 – 56

Jalur 11 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 –
16 – 17 – 18 – 19 – 20 – 21 – 22 – 24 – 29 – 33 – 17 – 34 – 36 – 37 –
38 – 39 – 40 – 41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53 – 54 –
55 – 56

Jalur 12 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 –
16 – 17 – 18 – 19 – 20 – 21 – 22 – 24 – 29 – 33 – 17 – 34 – 36 – 37 –
38 – 39 – 40 – 41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53 – 55 –
56

Jalur 13 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 –
16 – 17 – 18 – 19 – 20 – 21 – 25 – 26 – 27 – 28 – 29 – 33 – 17 – 34 –
36 – 37 – 38 – 39 – 40 – 41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 –
53 – 54 – 55 – 56

Jalur 14 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 –
16 – 17 – 18 – 19 – 20 – 21 – 25 – 26 – 27 – 28 – 29 – 33 – 17 – 34 –
36 – 37 – 38 – 39 – 40 – 41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 –
53 – 55 – 56

Jalur 15 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 –
16 – 17 – 18 – 19 – 20 – 21 – 25 – 26 – 28 – 29 – 33 – 17 – 34 – 36 –
37 – 38 – 39 – 40 – 41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53 –
54 – 55 – 56

Jalur 16 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 –
16 – 17 – 18 – 19 – 20 – 21 – 25 – 26 – 28 – 29 – 33 – 17 – 34 – 36 –

37 – 38 – 39 – 40 – 41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53 –
55 – 56

Jalur 17 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 –
16 – 17 – 18 – 19 – 20 – 21 – 25 – 29 – 33 – 17 – 34 – 36 – 37 – 38 –
39 – 40 – 41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53 – 54 – 55 –
56

Jalur 18 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 –
16 – 17 – 18 – 19 – 30 – 31 – 32 – 33 – 17 – 34 – 36 – 37 – 38 – 39 –
40 – 41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53 – 55 – 56

Jalur 19 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 –
16 – 17 – 18 – 19 – 30 – 32 – 33 – 17 – 34 – 36 – 37 – 38 – 39 – 40 –
41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53 – 55 – 56

Jalur 20 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 –
16 – 17 – 18 – 33 – 17 – 34 – 36 – 37 – 38 – 39 – 40 – 41 – 43 – 44 –
45 – 46 – 48 – 49 – 51 – 52 – 53 – 54 – 55 – 56

Jalur 21 : 1 – 2 – 3 – 4 – 5 – 7 – 4 – 8 – 9 – 11 – 2 – 12 – 13 – 14 – 15 – 16 –
17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 29 – 33 – 34 – 36 – 37 – 38 –
39 – 40 – 41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53 – 54 – 55 –
56

Jalur 22 : 1 – 2 – 3 – 4 – 5 – 7 – 4 – 8 – 10 – 11 – 2 – 12 – 13 – 14 – 15 – 16
– 17 – 18 – 19 – 20 – 21 – 25 – 26 – 27 – 28 – 29 – 33 – 17 – 34 – 36
– 37 – 38 – 39 – 40 – 41 – 43 – 44 – 45 – 46 – 48 – 49 – 51 – 52 – 53
– 55 – 56

Test case, user mengisi data registrasi kemudian menekan tombol *submit*.

Hasil pengujian path jalur 1 (1 – 2 – 3 – 4 – 5 – 6 – 7 – 4 – 8 – 9 – 11 – 2 – 12
– 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 29 – 33 – 17 –
34 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 42 – 44 – 45 – 46 – 47 – 51 – 52 – 53
– 54 – 55 – 56), dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi
kelayakan *software*, sistem ini telah memenuhi syarat.

Whitebox Testing dilakukan pada setiap fitur tiap tahap *increment*. Penjelasan diatas merupakan *whitebox testing* untuk fitur registrasi pasien lama, sedangkan fitur-fitur lainnya dapat dilihat pada lampiran E.



BAB 6. KESIMPULAN

6.1. Kesimpulan

Kesimpulan dari sistem yang dibangun pada penelitian ini adalah sebagai berikut:

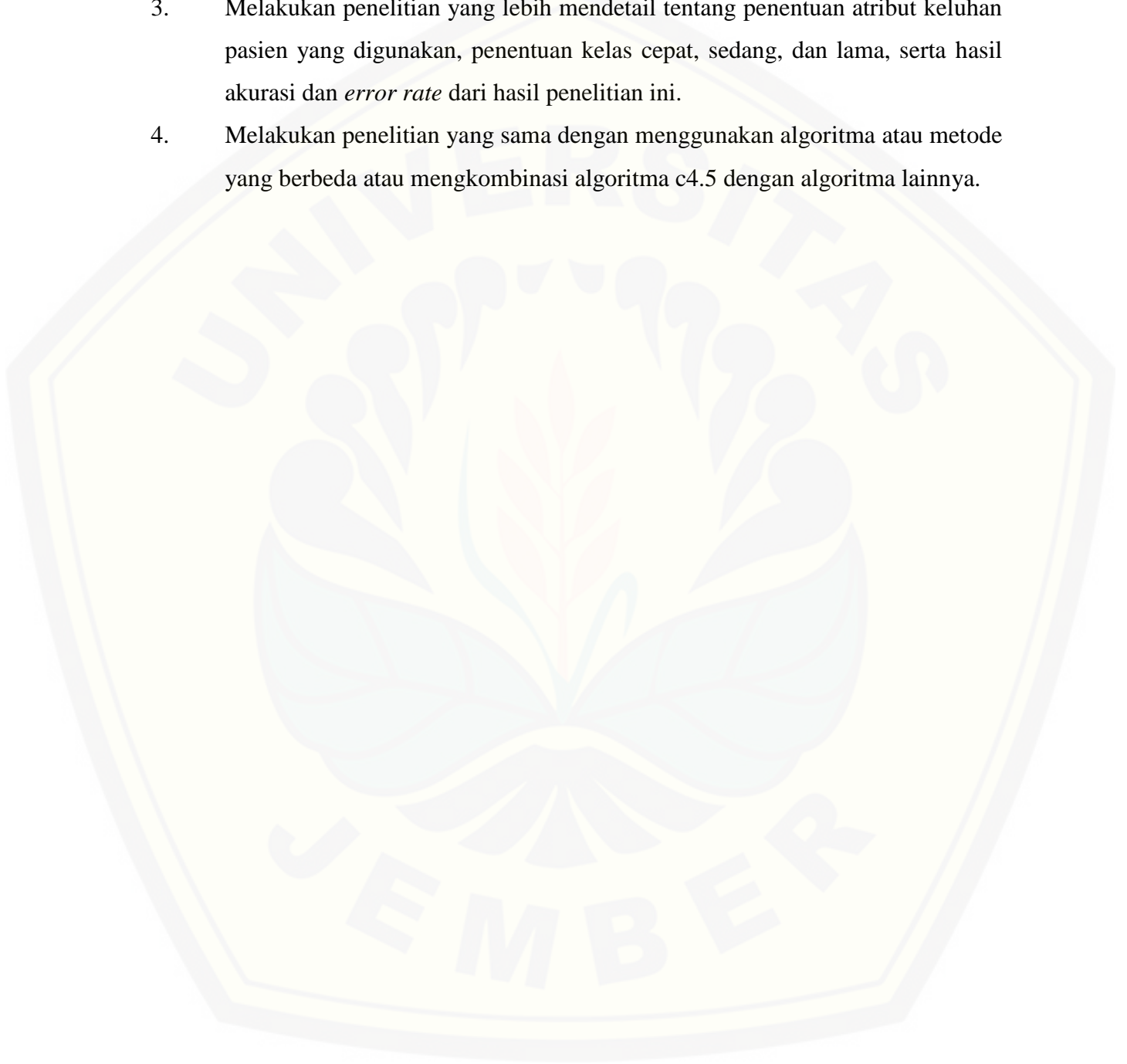
1. *Rule* yang dihasilkan oleh algoritma c4.5 memiliki nilai *accuracy* sebesar 65,8% dan *error rate* 34,2% sehingga algoritma ini dapat digunakan dalam fitur registrasi pasien. Kesalahan prediksi tertinggi berada pada keputusan asli 'sedang' dan keputusan c4.5 cepat yaitu mencapai 72,7 % dari 34.2% kesalahan. Hal tersebut bukan kesalahan yang fatal karena pasien akan mendapatkan waktu pemeriksaan sepuluh menit lebih cepat sehingga tidak akan merugikan proses antrian pasien.
2. Fitur registrasi pasien yang dibangun dengan menerapkan algoritma c4.5 menggunakan lima belas atribut yaitu usia, jenis kelamin, dan tiga belas jenis keluhan yang sering terjadi. Penerapan algoritma c4.5 dapat memprediksi waktu pemeriksaan pasien sehingga pasien tidak menunggu lama di klinik mata tersebut untuk mendapatkan giliran periksa.
3. Sistem informasi klinik mata dibangun sesuai kebutuhan dari objek penelitian dan dapat membantu kegiatan pengelolaan data yang terdapat pada klinik mata tersebut.

6.2. Saran

Hasil yang dicapai dari penelitian ini belum sempurna, oleh karena itu untuk meningkatkan hasil yang dicapai, beberapa saran untuk pengembangan penelitian ini diantaranya :

1. Menambahkan fitur pengelolaan laporan keuangan pada sistem informasi klinik mata.

2. Melakukan percobaan dengan menggunakan jumlah data *training* yang lebih banyak sehingga hasil yang dicapai lebih akurat.
3. Melakukan penelitian yang lebih mendetail tentang penentuan atribut keluhan pasien yang digunakan, penentuan kelas cepat, sedang, dan lama, serta hasil akurasi dan *error rate* dari hasil penelitian ini.
4. Melakukan penelitian yang sama dengan menggunakan algoritma atau metode yang berbeda atau mengkombinasi algoritma c4.5 dengan algoritma lainnya.



DAFTAR PUSTAKA

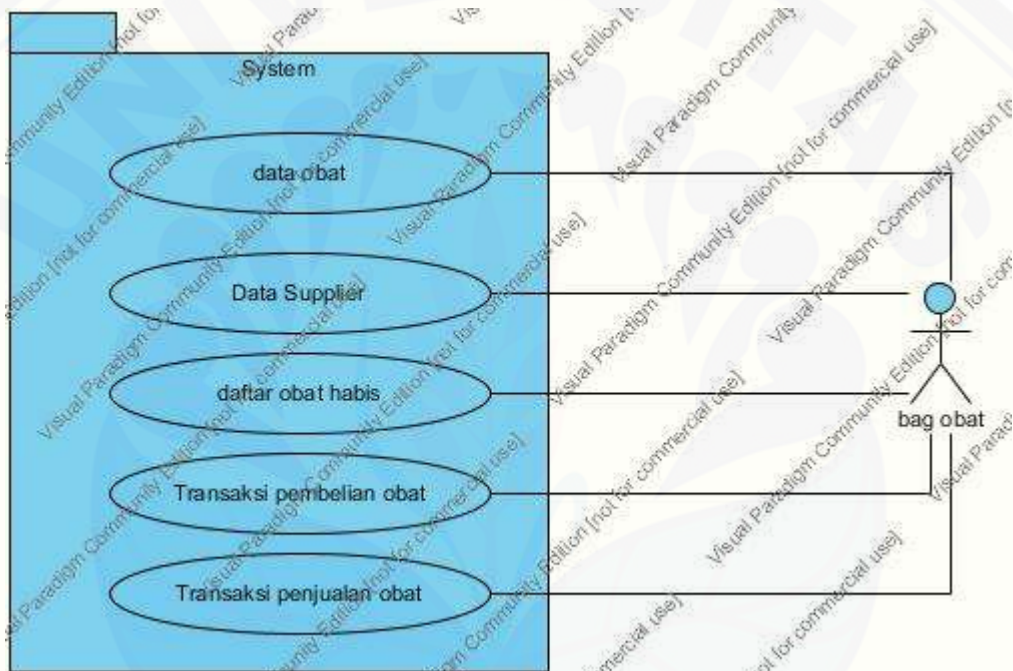
- A.S., Rosa, dan M. Shalahuddin. *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula, 2011.
- Andriani, Anik. "Sistem Pendukung Keputusan Berbasis Decision Tree Dalam Pemberian Beasiswa Studi Kasus: AMIK "BSI YOGYAKARTA"." *SENTIKA 2013*, 2013: 165.
- Dua, S, dan Xian Du. *Data Mining and Machine Learning in Cybersecurity*. Taylor & Francis Group, 2011.
- Han, J, dan M Kamber. *Data Mining Concept and Tehniques*. San Francisco: Morgan Kauffman, 2006.
- Kantardzic, Mehmed. *Data Mining: Concepts, Models, Methods, and Algorithms*. New York: John Wiley & Sons, INC., 2003.
- Khang, Bustam. *Trik Pemrograman Aplikasi Berbasis SMS*. Jakarta: Elex Media Komputindo, 2002.
- Larose, Daniel T. *Discovering Knowledge In Data: An Introduction to Data Mining*. Canada: John Wiley & Sons, Inc, 2005.
- Mega, Christanti. *megaboy16.blogspot.com*. 12 Maret 2011. <http://megaboy16.blogspot.com/2011/03/model-incremental.html> (diakses Agustus 18, 2014).
- Prahasta, Eddy. *Konsep-konsep Dasar Sistem Informasi Geografi*. Bandung: Informatika, 2001.
- Proboyekti, S. M. *Software Process Model I*. 1 Agustus 2008. <http://www.scribd.com/doc/192850316/Software-Process> (diakses Oktober 18, 2014).
- Surachman, Arif. *ANALISIS PENERIMAAN SISTEM INFORMASI PERPUSTAKAAN (SIPUS) TERPADU VERSI 3 DI LINGKUNGAN UNIVERSITAS GADJAH MADA (UGM)*. Yogyakarta: Perpustakaan Digital UIN Sunan Kalijaga Yogyakarta, 2008.

Witten, Ian H., dan Eibe Frank. *Data Mining : Practical Machine Learning Tools and Techniques Second Edition*. San Fransisco: Morgan Kaufmann, 2005.



LAMPIRAN**LAMPIRAN A. DESAIN SISTEM****A.1. Desain Sistem Tahap *Increment 2*****A.1.1 *Usecase Diagram Tahap Increment 2***

Usecase diagram tahap *increment 2* dapat dilihat pada Gambar 1.

**A.1.2 *Usecase Scenario Tahap Increment 2***

Usecase scenario menjelaskan urutan interaksi antara aktor dengan sistem. Pada subbab ini, *usecase scenario* menjelaskan urutan interaksi antara aktor dengan sistem pada fitur yang dibangun pada tahap *increment* dua.

a. *Usecase Scenario Data Obat*

Usecase scenario data obat dapat dilihat pada tabel 1.

Tabel 0.1 *Usecase scenario* data obat

<i>Name</i>	Data Obat
<i>Description</i>	Bag. Obat dapat menambah, melihat, meng- <i>edit</i> , dan menghapus data obat
<i>Actor</i>	Bag. Obat
<i>Precondition</i>	Bag. Obat memilih menu data obat
<i>Postcondition</i>	Data obat telah ditambahkan, diperbarui atau dihapus
Normal flow “Tambah data obat”	
<i>Actor</i>	<i>System</i>
1. Pilih menu data obat	
	2. Menampilkan halaman Data obat berupa data obat. Pada setiap baris data obat terdapat pilihan yaitu <i>edit</i> , <i>view</i> , hapus serta pilihan tambah diatas tabel yang digunakan untuk menambahkan data obat baru.
3. Klik “Tambah”	
	4. Menampilkan form data obat
5. Mengisi form data obat	
6. Klik “ <i>Submit</i> ”	
	7. Memeriksa kelengkapan data
	8. Menyimpan data obat baru
	9. Menampilkan <i>alert</i> “Data berhasil ditambahkan”
10. Klik “Ok”	
	11. Menampilkan halaman data obat yang telah diperbaharui
Normal flow “Edit Data Obat”	
12. Klik “ <i>edit</i> ” pada baris data obat yang ingin di- <i>edit</i> .	
	13. Menampilkan data obat yang akan di- <i>edit</i> pada form data obat.
14. Meng- <i>edit</i> data obat pada form data obat.	

15. Klik “ <i>Submit</i> ”	
	16. Memeriksa kelengkapan data
	17. Meng- <i>update</i> data obat
	18. Menampilkan <i>alert</i> “data obat berhasil diperbarui”
19. Klik “Ok”	
	20. Menampilkan halaman data obat yang telah diperbaharui.
<i>Normal flow “View Data Obat”</i>	
21. Klik “ <i>view</i> ” pada baris data obat yang ingin dilihat	
	22. Menampilkan detail data obat sesuai yang dipilih
23. Klik “ <i>close</i> ”	
	24. Menampilkan halaman data obat
<i>Normal flow “Hapus Data Obat”</i>	
25. Klik hapus pada baris data obat yang ingin dihapus.	
	26. Menampilkan <i>alert</i> “anda yakin ingin menghapus data ini?”
27. Klik “Ok”	
	28. Menghapus data obat
	29. Mengembalikan ke halaman data obat
<i>Alternative flow “kelengkapan data pada tambah data obat”</i>	
7.a Apabila <i>user</i> memasukkan data tidak lengkap.	
7.b Klik “ <i>Submit</i> ”	
	7.c Menampilkan <i>alert</i> “Maaf, data belum lengkap”
7.d Klik “Ok”	
	7.e Mengembalikan ke form tambah data obat
<i>Alternative flow “kelengkapan data pada edit data obat”</i>	
16.a Apabila <i>user</i> memasukkan data tidak lengkap	

16.b	Klik “ <i>Submit</i> ”	
		16.c Menampilkan <i>alert</i> “Maaf, data belum lengkap”
16.d	Klik “Ok”	
		16.e Mengembalikan ke form edit tambah data obat
Alternative flow “Batal menambah data obat”		
6.a	Apabila <i>user</i> batal menambah data obat.	
6.b	Klik “ <i>cancel</i> ”	
		6.c Mengembalikan ke halaman data obat
Alternative flow “Batal mengedit data obat”		
15.a	Apabila <i>user</i> batal mengedit data obat.	
15.b	Klik “ <i>Cancel</i> ”	
		15.c Mengembalikan ke halaman data obat
Alternative flow “Batal menghapus data obat”		
26.a	Apabila <i>user</i> batal menghapus data obat.	
26.b	Klik “Batal”	
		26.c Mengembalikan ke halaman data obat

b. *Usecase Scenario Data Supplier*

Usecase scenario data supplier dapat dilihat pada tabel 2.

Tabel 0.2 *Usecase Scenario data supplier*

Name	Data Supplier
Description	Bag. Obat dapat menambah, melihat, meng- <i>edit</i> , dan menghapus data <i>supplier</i>
Actor	Bag. Obat
Precondition	Bag. Obat memilih menu data <i>supplier</i>
Postcondition	Data <i>supplier</i> telah ditambahkan atau diperbarui.

Normal flow “Tambah data <i>supplier</i>”	
Actor	System
1. Pilih menu data <i>supplier</i>	
	2. Menampilkan halaman Data <i>supplier</i> berupa data <i>supplier</i> . Pada setiap baris data <i>supplier</i> terdapat pilihan yaitu <i>edit</i> , <i>view</i> , hapus serta pilihan tambah diatas tabel yang digunakan untuk menambahkan data <i>supplier</i> baru.
3. Klik “Tambah”	
	4. Menampilkan form data <i>supplier</i>
5. Mengisi form data <i>supplier</i>	
6. Klik “Submit”	
	7. Memeriksa kelengkapan data
	8. Menyimpan data <i>supplier</i> baru
	9. Menampilkan <i>alert</i> “Data berhasil ditambahkan”
10. Klik “Ok”	
	11. Menampilkan halaman data <i>supplier</i> yang telah diperbaharui
Normal flow “Edit Data <i>Supplier</i>”	
12. Klik “ <i>edit</i> ” pada baris data <i>supplier</i> yang ingin di- <i>edit</i> .	
	13. Menampilkan data <i>supplier</i> yang akan di- <i>edit</i> pada form data <i>supplier</i> .
14. Meng- <i>edit</i> data <i>supplier</i> pada form data <i>supplier</i> .	
15. Klik “Submit”	
	16. Memeriksa kelengkapan data
	17. Meng- <i>update</i> data <i>supplier</i>
	18. Menampilkan <i>alert</i> “data <i>supplier</i> berhasil diperbarui”
19. Klik “Ok”	
	20. Menampilkan halaman data

supplier yang telah diperbaharui.

Normal flow “View Data Supplier”

- | | |
|---|---|
| 21. Klik “ <i>view</i> ” pada baris data <i>supplier</i> yang ingin dilihat | 22. Menampilkan detail data <i>supplier</i> sesuai yang dipilih |
| 23. Klik “ <i>close</i> ” | 24. Menampilkan halaman data <i>supplier</i> |

Normal flow “Hapus Data Supplier”

- | | |
|--|---|
| 25. Klik hapus pada baris data <i>supplier</i> yang ingin dihapus. | 26. Menampilkan <i>alert</i> “anda yakin ingin menghapus data ini?” |
| 27. Klik “Ok” | 28. Menghapus data <i>supplier</i> |
| | 29. Mengembalikan ke halaman data <i>supplier</i> |

Alternative flow “kelengkapan data pada tambah data supplier”

- | | |
|--|---|
| 7.a Apabila <i>user</i> memasukkan data tidak lengkap. | 7.c Menampilkan <i>alert</i> “Maaf, data belum lengkap” |
| 7.b Klik “ <i>Submit</i> ” | 7.e Mengembalikan ke form tambah data <i>supplier</i> |
| 7.d Klik “Ok” | |

Alternative flow “kelengkapan data pada edit data supplier”

- | | |
|---|--|
| 16.a Apabila <i>user</i> memasukkan data tidak lengkap. | 16.c Menampilkan <i>alert</i> “Maaf, data belum lengkap” |
| 16.b Klik “ <i>Submit</i> ” | 16.e Mengembalikan ke form data <i>supplier</i> |
| 16.d Klik “Ok” | |

<i>Alternative flow “Batal menambah data supplier”</i>	
6.a	Apabila <i>user</i> batal menambah/mengedit data <i>supplier</i> .
6.b	Klik “cancel”
6.c	Mengembalikan ke halaman data <i>supplier</i>

<i>Alternative flow “Batal mengedit data supplier”</i>	
15.a	Apabila <i>user</i> batal menambah/mengedit data <i>supplier</i> .
15.b	Klik “cancel”
15.c	Mengembalikan ke halaman data <i>supplier</i>

<i>Alternative flow “Batal menghapus data supplier”</i>	
26.a	Apabila <i>user</i> batal menghapus data <i>supplier</i> .
26.b	Klik “Batal”
26.c	Mengembalikan ke halaman data <i>supplier</i>

c. *Usecase Scenario* Daftar Obat Habis

Usecase scenario daftar obat habis dapat dilihat pada tabel 3.

Tabel 0.3 *Usecase scenario* daftar obat habis

Name	
Daftar Obat Habis	
Description	Bag. Obat dapat melihat dan mencetak daftar obat yang habis
Actor	Bag. Obat
Precondition	Bag. Obat telah login
Postcondition	Daftar obat yang habis telah dicetak
Normal flow “Daftar Obat Habis”	
Actor	System
1. Pilih menu daftar obat habis	2. Menampilkan halaman Daftar obat habis yang berisi tabel daftar obat yang habis beserta tombol

	cetak untuk mencetak daftar obat habis (Stok obat kurang dari 50)
3. Klik “cetak”	
	4. Menampilkan alert, “Apakah anda ingin mencetak?”
5. Klik “Ok”	
	6. Mencetak daftar obat yang habis
	7. Menampilkan halaman Daftar obat habis
Alternative flow “Batal mencetak”	
4.a	Apabila <i>user</i> batal mencetak daftar obat habis
4.b	Klik “Batal”
	4.c Mengembalikan ke halaman Daftar obat habis

d. *Usecase Scenario* Transaksi Pembelian Obat

Usecase scenario transaksi pembelian obat dapat dilihat pada tabel 4.

Tabel 0.4 Usecase scenario transaksi pembelian obat

Transaksi Pembelian Obat	
Name	Transaksi Pembelian Obat
Description	Bag. Obat dapat mencatat transaksi pembelian obat
Actor	Bag. Obat
Precondition	Bag. Obat telah login ke sistem
Postcondition	Transaksi pembelian obat telah simpan
Normal flow “Transaksi Pembelian Obat”	
Actor	System
1. Pilih menu Transaksi Pembelian	
	2. Menampilkan halaman transaksi pembelian yang berisi form transaksi pembelian obat. terdapat dua jenis form. Form yang pertama berisi data transaksi pembelian yang berisi tanggal transaksi dan data supplier, sedangkan form yang

	kedua berisi data obat yang dibeli, dimana user harus menekan 'Add' setiap satu jenis obat diinputkan
3. Mengisi data obat yang dibeli	
4. Klik "Add"	
	5. Menampilkan tabel yang berisi data obat yang diinputkan dan tombol hapus disetiap baris data
6. Mengisi data transaksi pembelian	
7. Klik "Submit"	
	8. Memeriksa kelengkapan data
	9. Menyimpan data transaksi pembelian obat
	10. Menampilkan <i>alert</i> "Transaksi Berhasil"
11. Klik "Ok"	
	12. Menampilkan halaman transaksi pembelian obat
Normal flow "Hapus Baris Data"	
13. Klik " <i>hapus</i> " pada baris data obat yang ingin dihapus.	
	14. Menghapus data obat sesuai baris yang dipilih
Alternative flow "kelengkapan data"	
8.a Apabila <i>user</i> memasukkan data tidak lengkap.	
8.b Klik "Submit"	
	8.c Menampilkan <i>alert</i> "Maaf, data belum lengkap"
8.d Klik "Ok"	
	8.e Mengembalikan ke form transaksi pembelian obat

e. *Usecase Scenario* Transaksi Penjualan Obat

Usecase scenario transaksi penjualan obat dapat dilihat pada tabel 5.

Tabel 0.5 Usecase scenario transaksi penjualan obat

Name		Transaksi Penjualan Obat
Description	Bag. Obat dapat mencatat transaksi penjualan obat	
Actor	Bag. Obat	
Precondition	Bag. Obat telah login ke sistem	
Postcondition	Transaksi penjualan obat telah disimpan	
Normal flow "Transaksi Penjualan Obat"		
<i>Actor</i>	<i>System</i>	
1. Pilih menu Transaksi Penjualan	2. Menampilkan halaman transaksi penjualan yang berisi form transaksi penjualan obat. terdapat dua jenis form. Form yang pertama adalah form data transaksi pembelian yang berisi tanggal transaksi dan data pasien, sedangkan form yang kedua berisi data obat yang dibeli, dimana user harus menekan 'Add' setiap satu jenis obat diinputkan	
3. Mengisi data obat yang dibeli		
4. Klik "Add"	5. Menampilkan tabel yang berisi data obat yang diinputkan dan tombol hapus disetiap baris data	
6. Mengisi data transaksi penjualan		
7. Klik "Cetak"	8. Memeriksa kelengkapan data	
	9. Menyimpan data transaksi	
	10. Mencetak nota pembayaran	
	11. Menampilkan halaman transaksi penjualan obat	

Normal flow “Hapus Baris Data”

- | | |
|---|---|
| 12. Klik “hapus” pada baris data obat yang ingin dihapus. | |
| | 13. Menghapus data obat sesuai baris yang dipilih |

Alternative flow “kelengkapan data”

- | | |
|--|---|
| 8.a Apabila <i>user</i> memasukkan data tidak lengkap. | |
| 8.b Klik “cetak” | |
| | 8.c Menampilkan <i>alert</i> “Maaf, data belum lengkap” |
| 8.d Klik “Ok” | |
| | 8.e Mengembalikan ke form transaksi penjualan obat |

A.1.3 Activity Diagram Tahap Increment 2

Activity diagram menggambarkan alur aktivitas antara aktor dan sistem. Pada subbab ini, *activity diagram* menggambarkan alur aktivitas pada fitur yang dibangun pada tahap *increment* dua.

a. *Activity Diagram* Data Obat

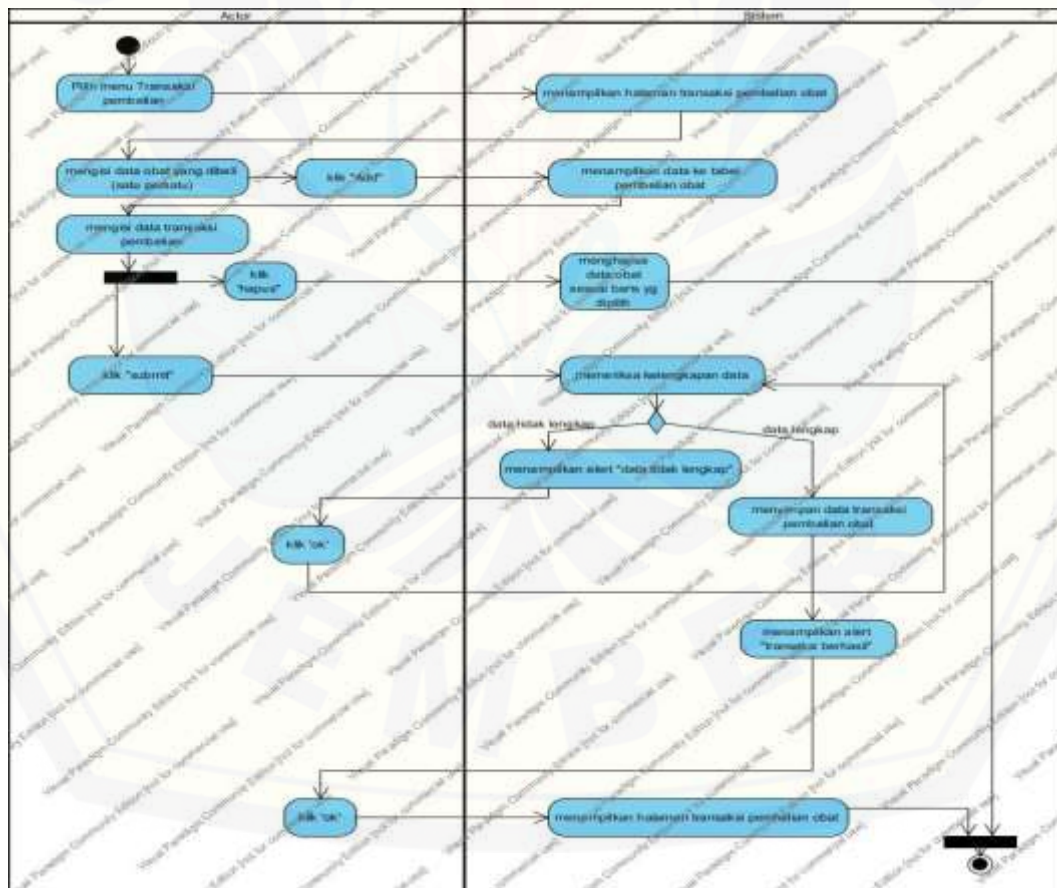
Activity diagram data obat dapat dilihat pada gambar 2.



Gambar 0.3 Activity diagram daftar obat habis

d. Activity Diagram Transaksi Pembelian Obat

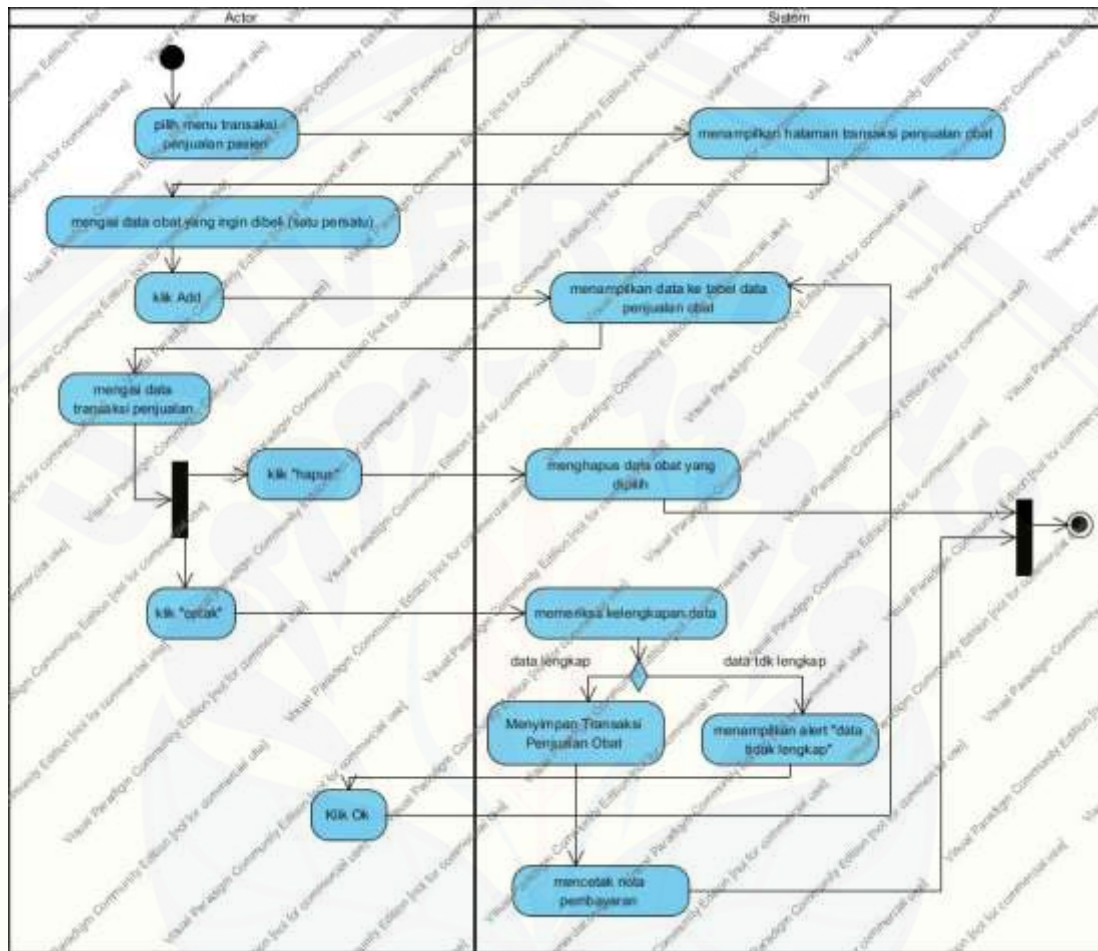
Activity diagram data obat dapat dilihat pada gambar 5.



Gambar 0.4 Activity diagram transaksi penjualan obat

e. *Activity Diagram* Transaksi Penjualan Obat

Activity diagram data obat dapat dilihat pada gambar 6.

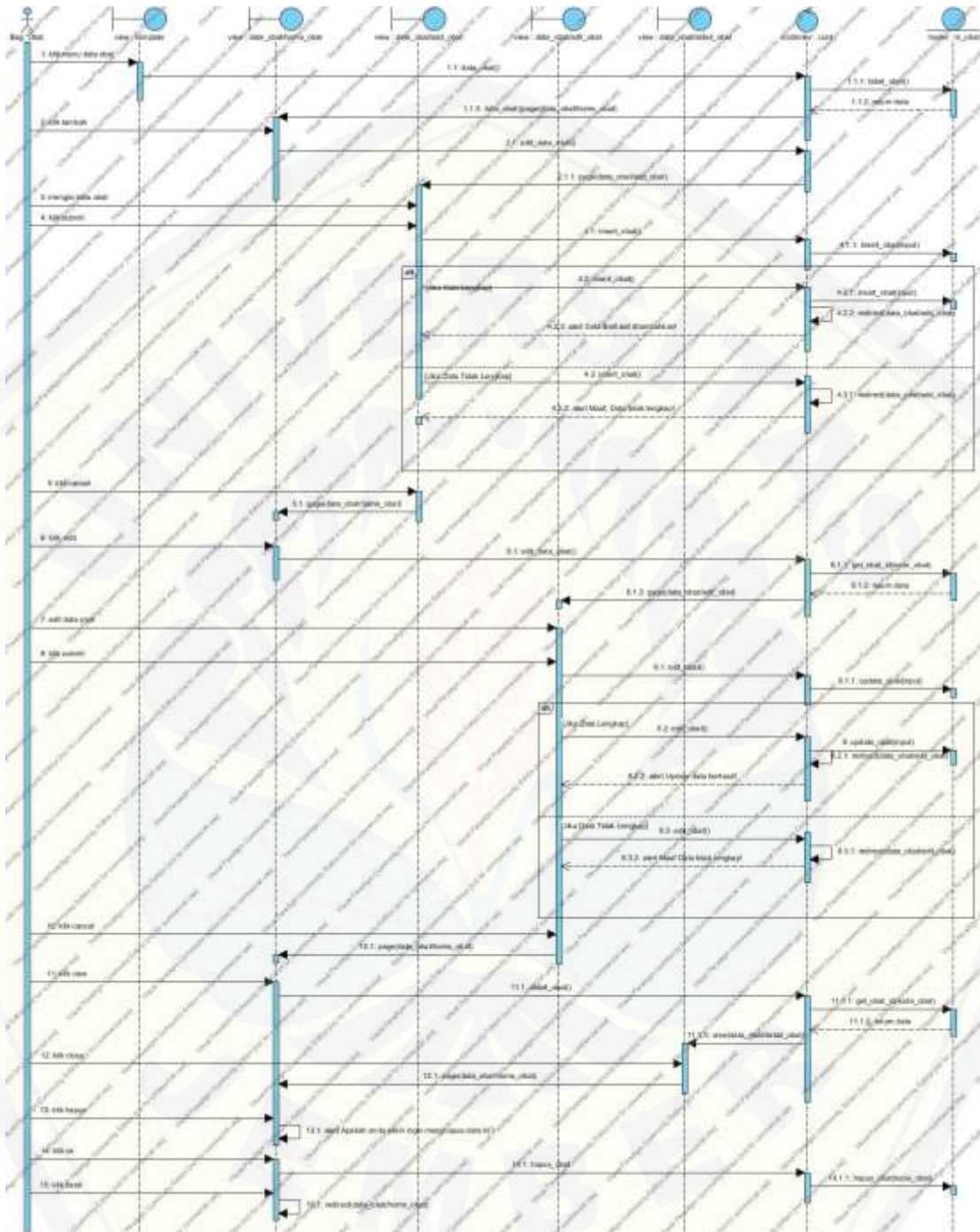


Gambar 0.5 Activity diagram transaksi penjualan obat

A.1.4 Sequence Diagram Tahap Increment 2

a. *Sequence Diagram* Data Obat

Sequence diagram data obat dapat dilihat pada gambar 7.



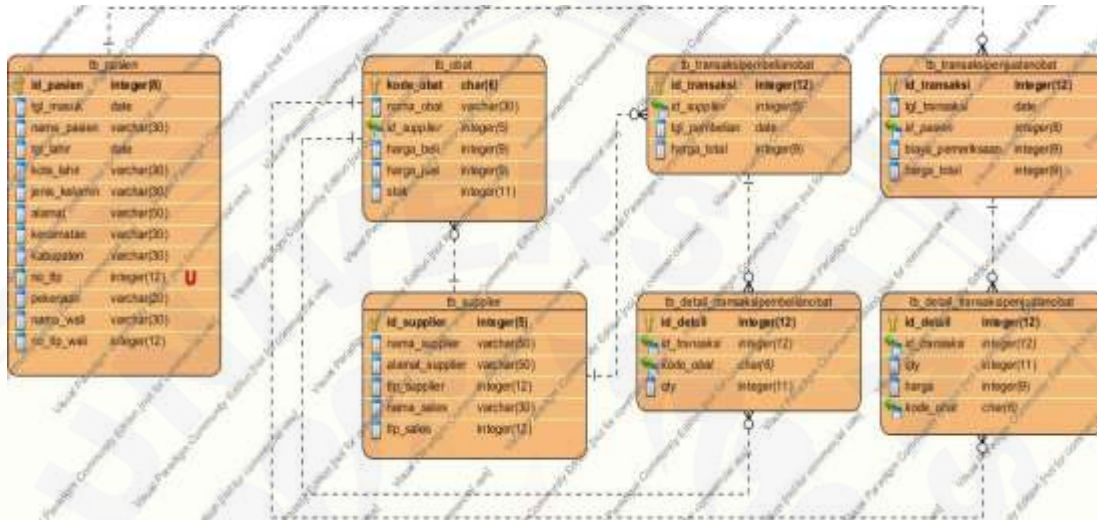
Gambar 0.6 *Sequence diagram* data obat

b. *Sequence Diagram* Data Supplier

Sequence diagram data supplier dapat dilihat pada gambar 8.

A.1.6 Entity Relationship Diagram Tahap Increment 2

Entity relationship diagram tahap increment dua dapat dilihat pada gambar 13.

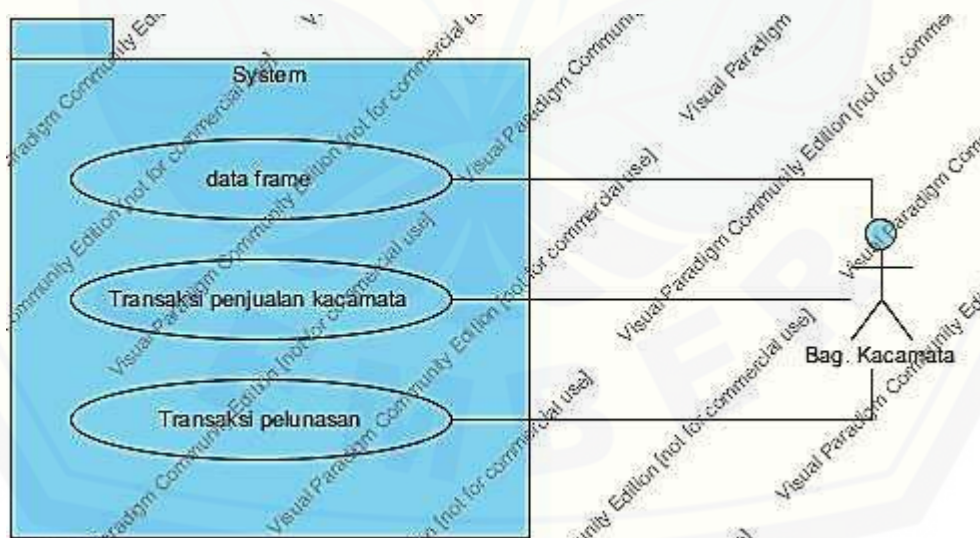


Gambar 0.11 ERD tahap increment dua

A.2. Desain Sistem Tahap Increment 3

A.2.1 Usecase Diagram Tahap Increment 3

Usecase diagram tahap increment tiga dapat dilihat pada gambar .



Gambar 0.12 Usecase diagram tahap increment tiga

A.2.2 Usecase Scenario Tahap Increment 3

a. Usecase Scenario Data Frame

Usecase scenario data frame dapat dilihat pada tabel 6.

Tabel 0.6 usecase scenario data frame

Name	Data Frame
Description	Bag. Kacamata dapat menambah, melihat, meng- <i>edit</i> , dan menghapus data <i>frame</i>
Actor	Bag. Kacamata
Precondition	Bag. Kacamata memilih menu data <i>frame</i>
Postcondition	Data <i>frame</i> telah ditambahkan, atau diperbarui.
Normal flow “Tambah Data Frame”	
Actor	System
1. Pilih menu data <i>frame</i>	
	2. Menampilkan halaman Data kacamata berupa data <i>frame</i> . Pada setiap baris data <i>frame</i> terdapat pilihan yaitu <i>edit</i> dan hapus serta pilihan tambah diatas tabel yang digunakan untuk menambahkan data kacamata baru.
3. Klik “Tambah”	
	4. Menampilkan form data <i>frame</i>
5. Mengisi form data <i>frame</i>	
6. Klik “ <i>Submit</i> ”	
	7. Memeriksa kelengkapan data
	8. Menyimpan data <i>frame</i> baru
	9. Menampilkan <i>alert</i> “Data berhasil ditambahkan”
10. Klik “Ok”	
	11. Menampilkan halaman data <i>frame</i> yang telah diperbaharui
Normal flow “Edit Data Frame”	

12. Klik “ <i>edit</i> ” pada baris data <i>frame</i> yang ingin di- <i>edit</i> .	
	13. Menampilkan data <i>frame</i> yang akan di- <i>edit</i> pada form data <i>frame</i> .
14. Meng- <i>edit</i> data <i>frame</i> pada form data <i>frame</i> .	
15. Klik “ <i>Submit</i> ”	
	16. Memeriksa kelengkapan data
	17. Meng- <i>update</i> data <i>frame</i>
	18. Menampilkan <i>alert</i> “data <i>frame</i> berhasil diperbarui”
19. Klik “Ok”	
	20. Menampilkan halaman data <i>frame</i> yang telah diperbaharui.
<i>Normal flow “Hapus Data Frame”</i>	
21. Klik hapus pada baris data <i>frame</i> yang ingin dihapus.	
	22. Menampilkan <i>alert</i> “anda yakin ingin menghapus data ini?”
23. Klik “Ok”	
	24. Menghapus data <i>frame</i>
	25. Mengembalikan ke halaman data <i>frame</i>
<i>Alternative flow “kelengkapan data pada tambah data frame”</i>	
7.a Apabila <i>user</i> memasukkan data tidak lengkap.	
7.b Klik “ <i>Submit</i> ”	
	7.c Menampilkan <i>alert</i> “Maaf, data belum lengkap”
7.a Klik “Ok”	
	7.b Mengembalikan ke form tambah data <i>frame</i>
<i>Alternative flow “kelengkapan data pada edit data frame”</i>	
16.a Apabila <i>user</i> memasukkan data tidak lengkap.	

16.b	Klik “ <i>Submit</i> ”	
		16.c Menampilkan <i>alert</i> “Maaf, data belum lengkap”
16.d	Klik “Ok”	
		16.e Mengembalikan ke form edit data <i>frame</i>
Alternative flow “Batal menambah data frame”		
6.a	Apabila <i>user</i> batal menambah data <i>frame</i> .	
6.b	Klik “ <i>cancel</i> ”	
		6.c Mengembalikan ke halaman data <i>frame</i>
Alternative flow “Batal mengedit data frame”		
15.a	Apabila <i>user</i> batal mengedit data <i>frame</i> .	
15.b	Klik “ <i>cancel</i> ”	
		15.c Mengembalikan ke halaman data <i>frame</i>
Alternative flow “Batal menghapus data frame”		
22.a	Apabila <i>user</i> batal menghapus data <i>frame</i> .	
22.b	Klik “Batal”	
		22.c Mengembalikan ke halaman data <i>frame</i>

b. *Usecase Scenario* Transaksi Penjualan Kacamata

Usecase scenario transaksi penjualan kacamata dapat dilihat pada tabel 8.

Tabel 0.7 *Usecase scenario* transaksi penjualan kacamata

Name	Transaksi Penjualan Kacamata
Description	Bag. Kacamata dapat mencatat transaksi penjualan kacamata
Actor	Bag. Kacamata
Precondition	Bag. Kacamata memilih menu transaksi

	penjualan
Postcondition	Transaksi penjualan kacamata telah simpan
Normal flow “Transaksi Penjualan Kacamata”	
Actor	System
1. Pilih menu Transaksi Penjualan	
	2. Menampilkan halaman transaksi penjualan yang berisi form transaksi penjualan kacamata. terdapat dua jenis form. Form yang pertama berisi data transaksi penjualan yang berisi tanggal transaksi dan data pasien, sedangkan form yang kedua berisi data kacamata yang dibeli, dimana user harus menekan ‘Add’ setiap satu jenis kacamata diinputkan
3. Mengisi data kacamata yang dibeli	
4. Klik “Add”	
	5. Menampilkan tabel yang berisi data kacamata yang diinputkan dan tombol hapus disetiap baris data
6. Mengisi data transaksi penjualan	
7. Klik “cetak”	
	8. Memeriksa kelengkapan data
	9. Menyimpan data transaksi
	10. Mencetak nota pembayaran
	11. Menampilkan halaman transaksi penjualan kacamata
Normal flow “Hapus Baris Data”	
12. Klik “hapus” pada baris data	

kacamata yang ingin dihapus.	
	13. Menghapus data kacamata sesuai baris yang dipilih
Alternative flow “Kelengkapan data”	
8.a	Apabila <i>user</i> memasukkan data tidak lengkap.
8.b	Klik “cetak”
	8.c Menampilkan <i>alert</i> “Maaf, data belum lengkap”
8.d	Klik “Ok”
	8.e Mengembalikan ke form transaksi penjualan kacamata

c. *Usecase Scenario* Transaksi Pelunasan Kacamata

Usecase scenario transaksi pelunasan kacamata dapat dilihat pada tabel 9.

Tabel 0.8 *Usecase scenario* transaksi pelunasan kacamata

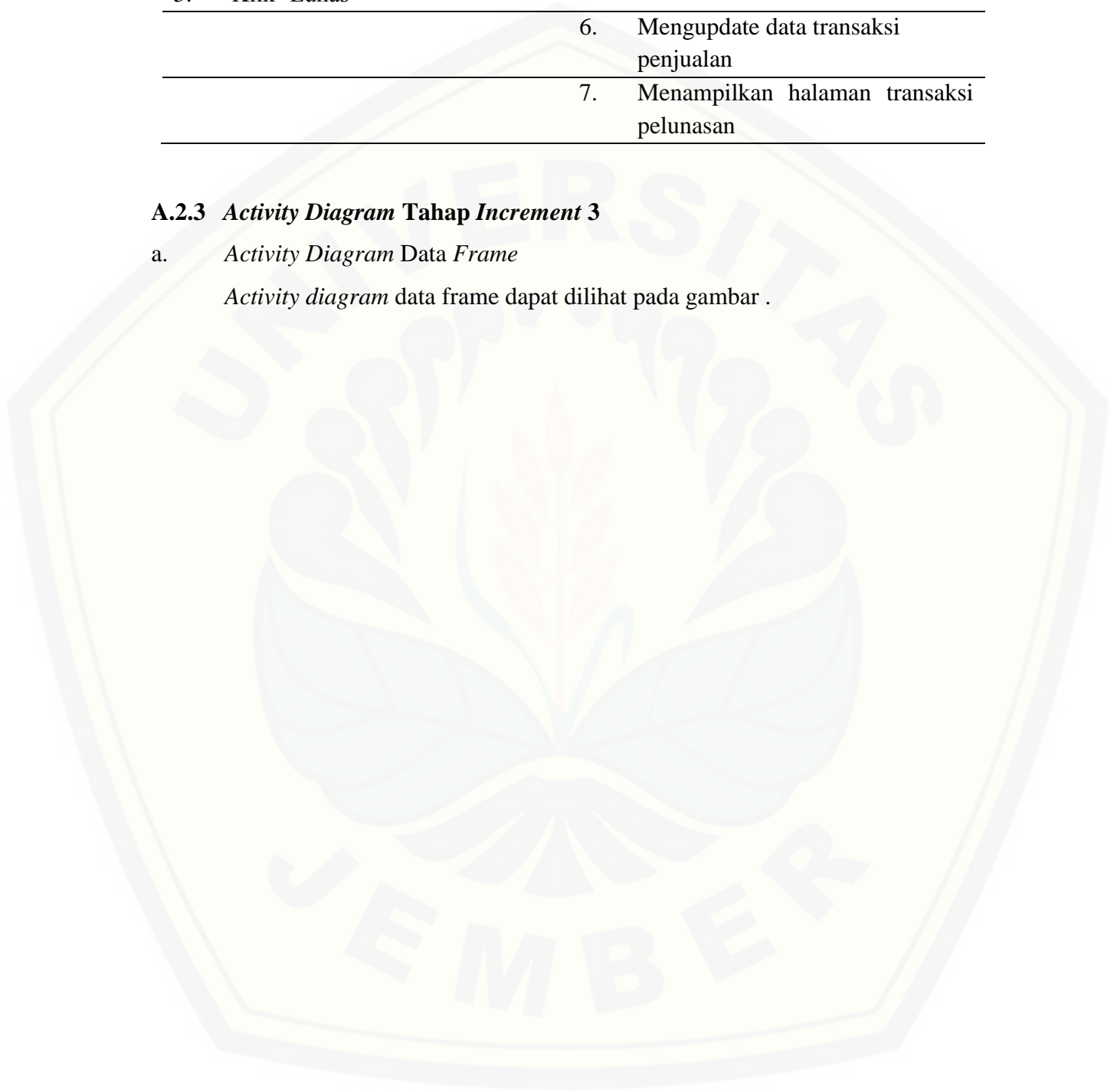
Name	Transaksi Pelunasan
Description	Bag. Kacamata dapat memasukkan pelunasan pembayaran pembelian kacamata pasien
Actor	Bag. Kacamata
Precondition	Pasien memesan kacamata dan belum melunasi pembayaran
Postcondition	Status pembayaran kacamata pasien lunas
Normal flow “Transaksi Pelunasan Kacamata”	
Actor	System
1. Pilih menu Transaksi Pelunasan	
	2. Menampilkan halaman transaksi pelunasan berupa tabel (berisi data transaksi penjualan kacamata dengan status BELUM LUNAS)
3. Mengisi id pasien pada kolom pencarian	

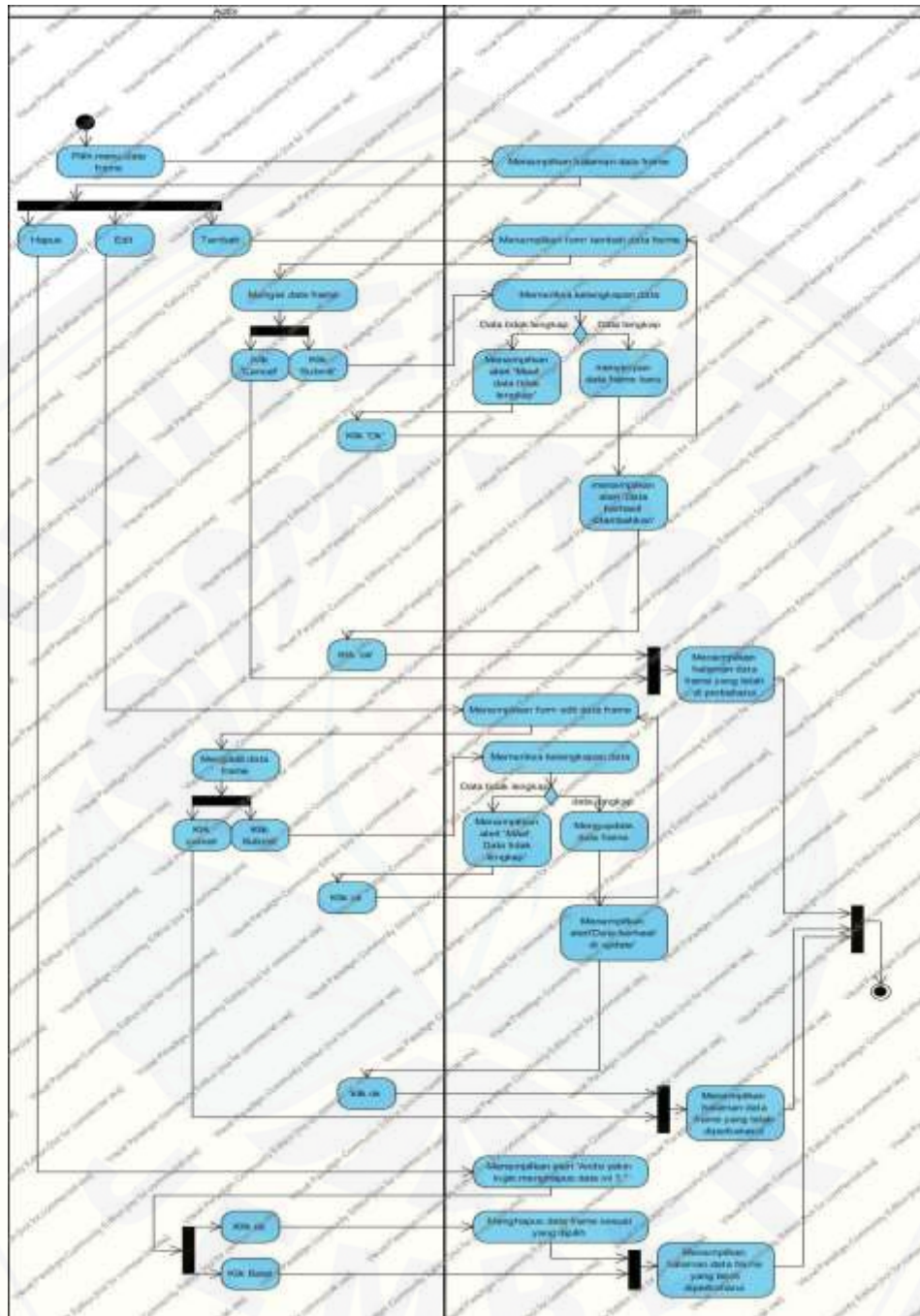
	4. Menampilkan data yang dicari
5. Klik 'Lunas'	
	6. Mengupdate data transaksi penjualan
	7. Menampilkan halaman transaksi pelunasan

A.2.3 Activity Diagram Tahap Increment 3

a. Activity Diagram Data Frame

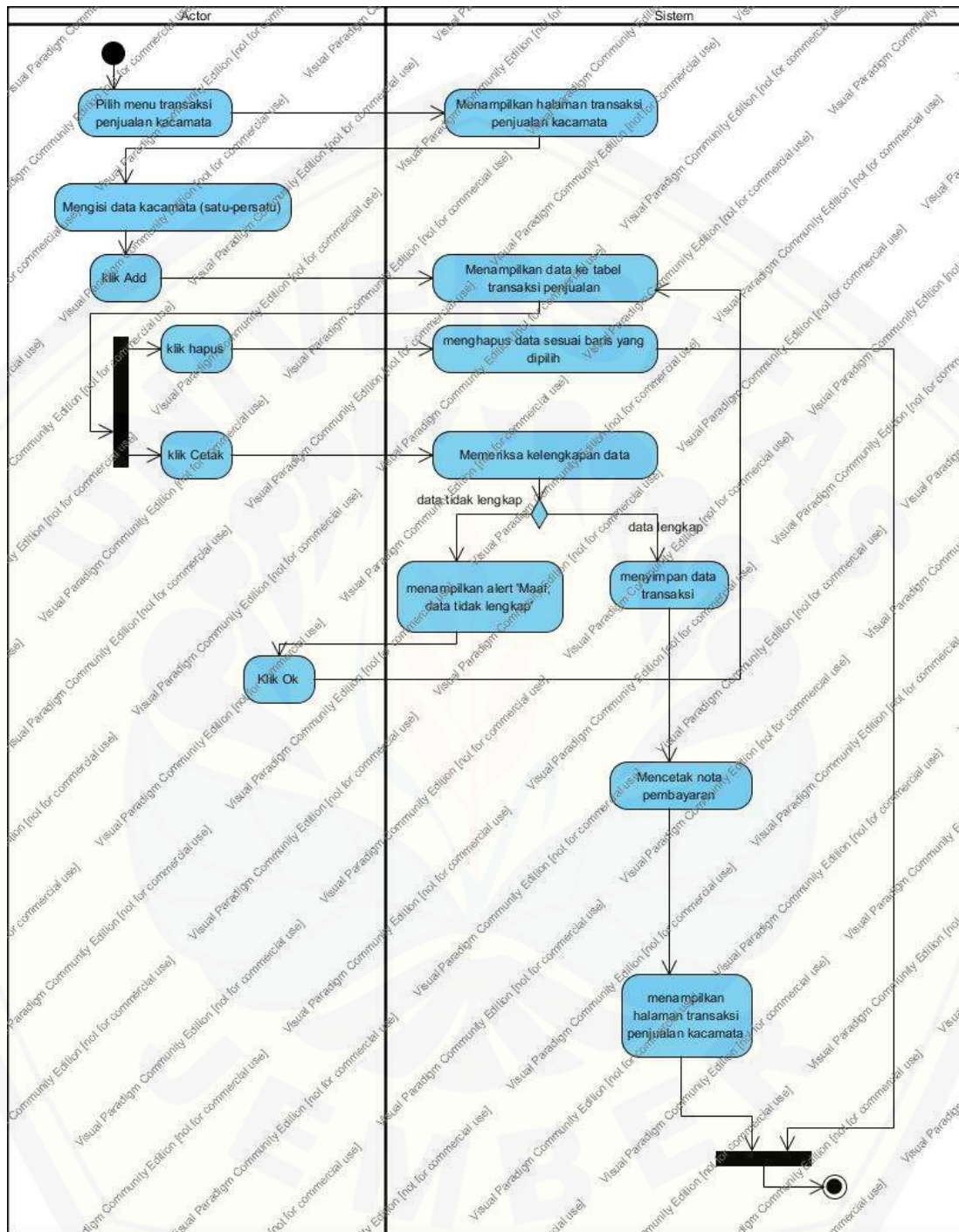
Activity diagram data frame dapat dilihat pada gambar .





Gambar 0.13 Activity diagram data frame

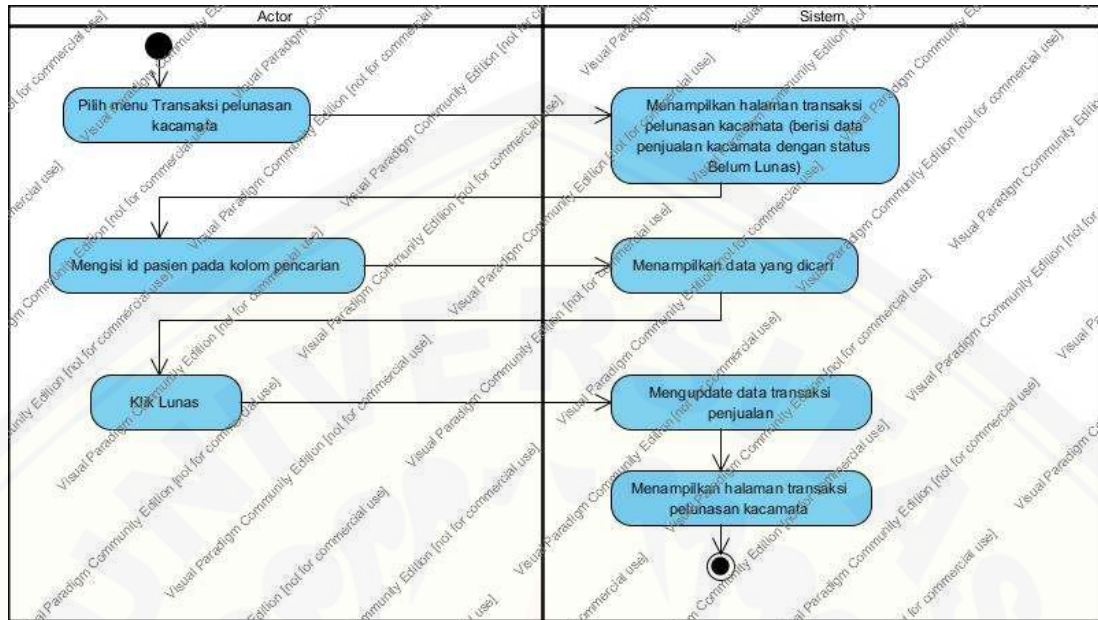
- b. Activity Diagram Transaksi Penjualan Kacamata
 Activity diagram transaksi penjualan kacamata dapat dilihat pada gambar .



Gambar 0.14 Activity diagram transaksi penjualan kacamata

c. Activity Diagram Transaksi Pelunasan Kacamata

Activity diagram transaksi pelunasan kacamata dapat dilihat pada gambar .

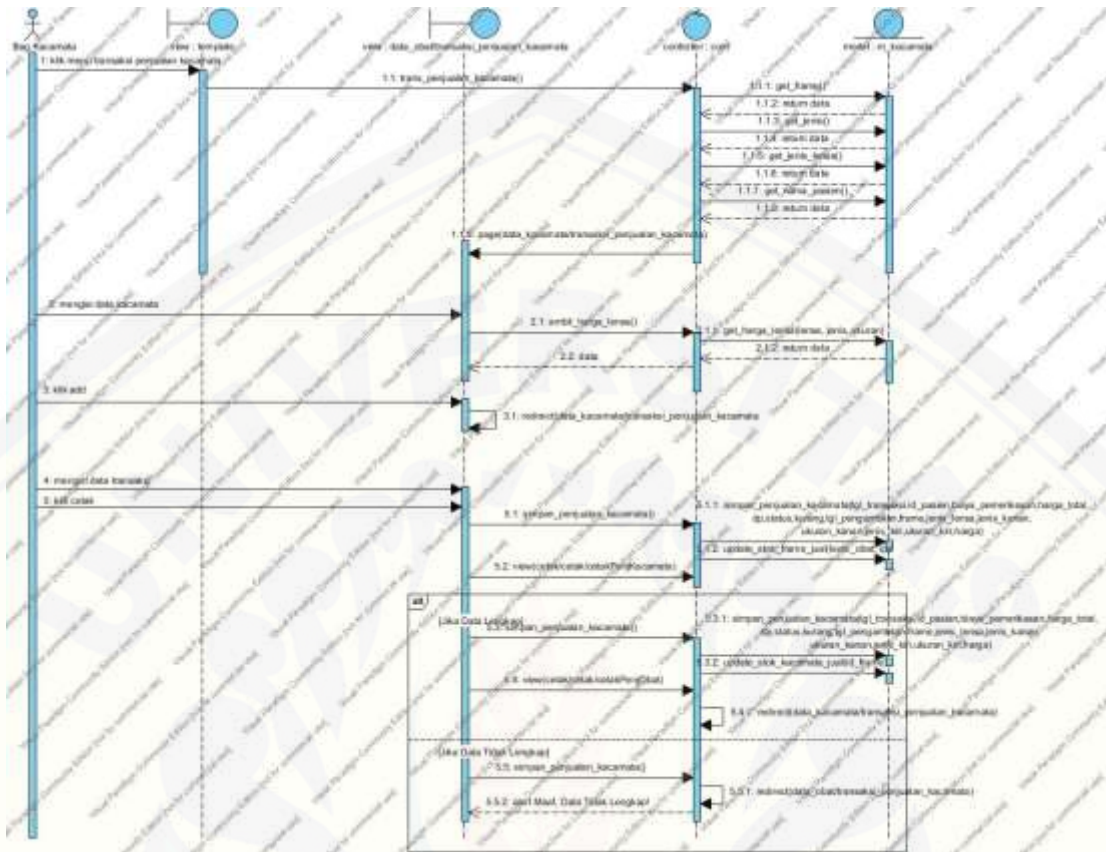


Gambar 0.15 Activity diagram transaksi pelunasan kacamata

A.2.4 Sequence Diagram Tahap Increment 3

a. Sequence Diagram Data Frame

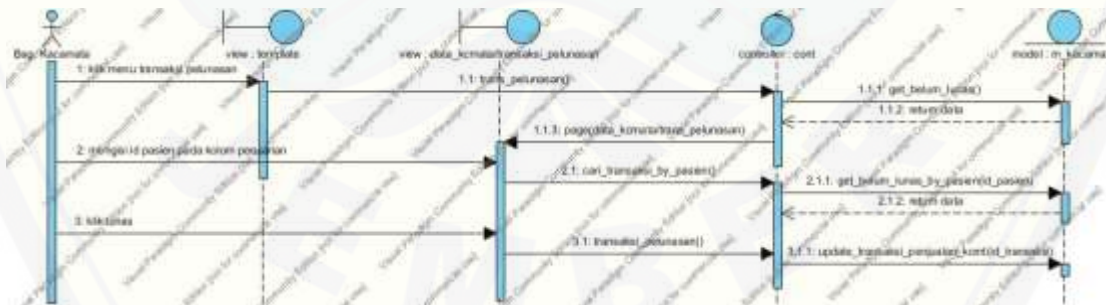
Sequence diagram data frame dapat dilihat pada gambar .



Gambar 0.17 Sequence diagram transaksi penjualan kacamata

c. Sequence Diagram Transaksi Pelunasan Kacamata

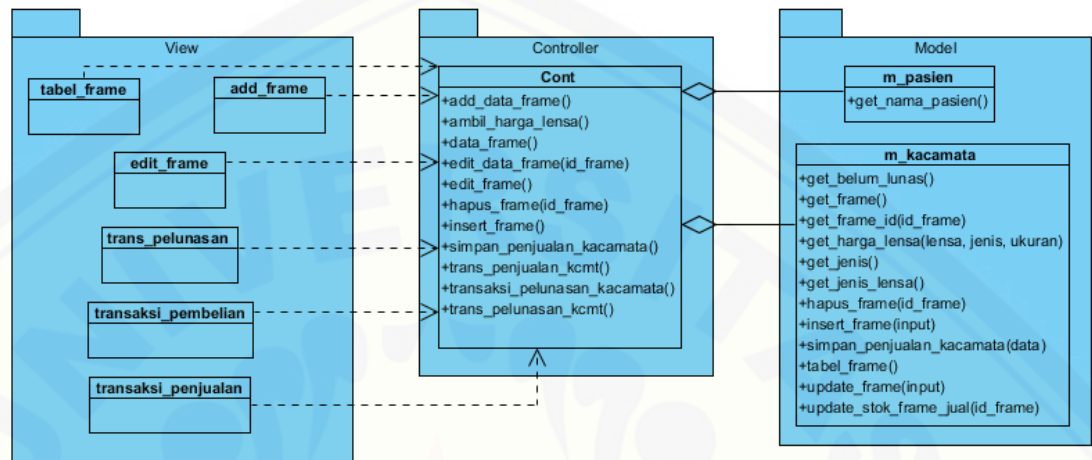
Sequence diagram transaksi pelunasan kacamata dapat dilihat pada gambar .



Gambar 0.18 sequence diagram transaksi pelunasan kacamata

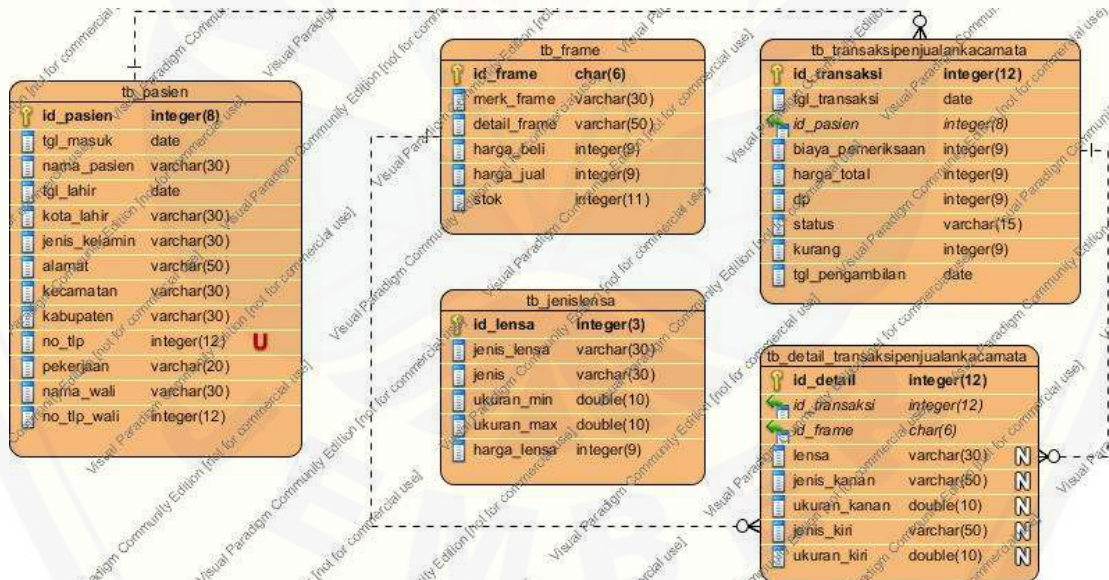
A.2.5 Class Diagram Tahap Increment 3

Class diagram tahap increment tiga dapat dilihat pada gambar .



A.2.6 Entity Relationship Diagram Tahap Increment 3

Entity relationship diagram tahap increment tiga dapat dilihat pada gambar .

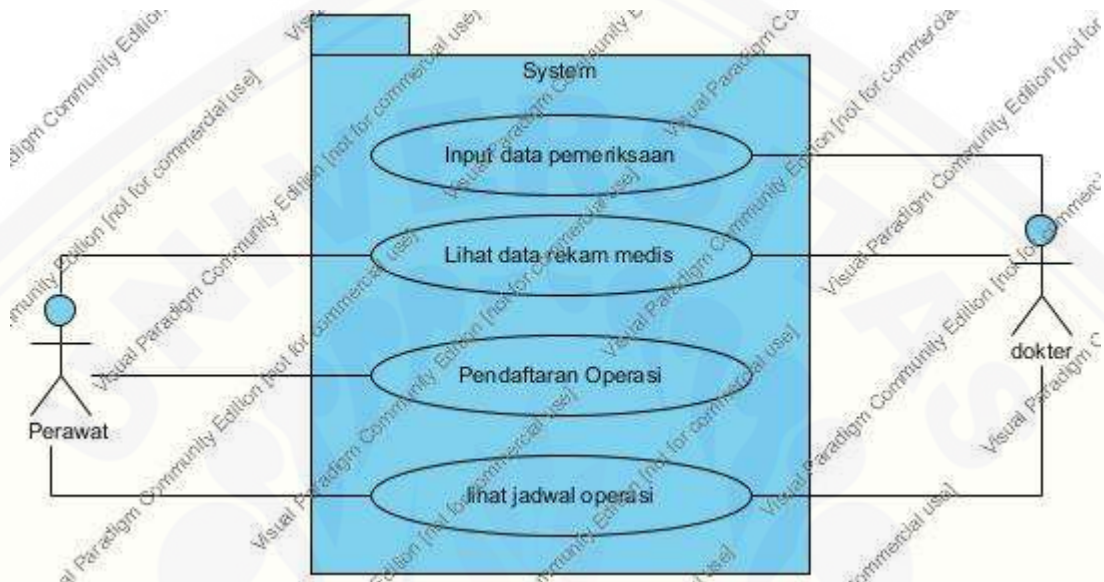


Gambar 0.19 ERD tahap increment tiga

A.3. Desain Sistem Tahap Increment 4

A.3.1 Usecase Diagram Tahap Increment 4

Usecase diagram tahap increment empat dapat dilihat pada gambar .



Gambar 0.20 Usecase diagram tahap increment empat

A.3.2 Usecase Scenario Tahap Increment 4

a. *Usecase Scenario* Input Data Pemeriksaan

Usecase scenario input data pemeriksaan dapat dilihat pada tabel .

Tabel 0.9 *Usecase scenario* input data pemeriksaan

Name		Input Data Pemeriksaan	
Description		Dokter dapat meng- <i>input</i> data pemeriksaan pasien	
Actor		Dokter	
Precondition		-	
Post condition		Data pemeriksaan telah ditambahkan	
Normal flow “Input Data Pemeriksaan”			
Actor		System	
1.	Pilih menu <i>input</i> data pemeriksaan	2.	Menampilkan halaman input data pemeriksaan

3. Mengisi data pemeriksaan	
4. Klik “Submit”	
	5. Memeriksa kelengkapan data
	6. Menyimpan data pemeriksaan
	7. Mengembalikan pada halaman input data pemeriksaan
Alternative flow “kelengkapan data”	
5.a Apabila <i>user</i> memasukkan data tidak lengkap.	
5.b Klik “Submit”	
	5.c Menampilkan <i>alert</i> “Maaf, data belum lengkap”
5.d Klik “Ok”	
	5.e Mengembalikan ke halaman <i>input</i> data pemeriksaan

b. *Usecase Scenario* Lihat Data Rekam Medis

Usecase scenario lihat data rekam medis dapat dilihat pada tabel .

Tabel 0.10 *Usecase scenario* lihat data rekam medis

Name	Data Pasien
Description	Dokter atau Perawat dapat melihat data rekam medis pasien
Actor	Dokter dan Perawat
Precondition	User memilih menu lihat data rekam medis
Postcondition	Sistem menampilkan data rekam medis
Normal flow “Lihat Data Rekam Medis ”	
Actor	System
1. Pilih menu lihat data rekam medis	
	2. Menampilkan halaman lihat data rekam medis
3. Pilih id pasien	
	4. Menampilkan tabel data rekam medis pasien sesuai dengan id

pasien yang dipilih	
Alternative flow “user tidak memilih id pasien”	
3.a	Apabila <i>user</i> tidak memilih id pasien
3.b	Pilih id pasien
	3.c Menampilkan <i>alert</i> , “Pilih ID Pasien!”
3.d	Klik “Ok”
	3.e Mengembalikan ke halaman lihat data rekam medis
Alternative flow “ID pasien tidak ditemukan”	
3.a	Apabila ID pasien yang di masukkan oleh <i>user</i> tidak ditemukan datanya
3.b	Pilih id pasien
	3.c Menampilkan alert , “ Data tidak ditemukan!”
3.d	Klik “Ok”
	3.e Menampilkan halaman lihat data rekam medis

c. *Usecase Scenario* Pendaftaran Operasi

Usecase scenario pendaftaran operasi dapat dilihat pada tabel .

Tabel 0.11 *Usecase scenario* pendaftaran operasi

Name	Pendaftaran Operasi
Description	Perawat mendaftarkan operasi pasien
Actor	Perawat
Precondition	User memilih menu pendaftaran operasi
Postcondition	Pasien telah terdaftar untuk operasi
Normal flow “Pendaftaran Operasi”	
Actor	System
11. Pilih menu pendaftaran operasi	12. Menampilkan halam pendaftaran operasi berisi form pendaftaran operasi pasien
13. Mengisi form pendaftaran operasi	

14. Klik “Submit”	
	15. Menyimpan data pendaftaran operasi
	16. Menampilkan halaman pendaftaran operasi
Normal flow “Cek Ketersediaan Tanggal”	
17. Klik ‘Cek kuota’	
	18. Jika pada tanggal yang dipilih kuota telah terisi 4 pasien maka, Menampilkan <i>alert</i> “Kuota tidak mencukupi.(Jumlah kuota terpenuhi : 4)”
	19. Jika pada tanggal yang dipilih kuota kurang dari 4 pasien maka, Menampilkan <i>alert</i> “Kuota mencukupi.(Kuota saat ini : jumlah pasien terdaftar)”
20. Klik “Ok”	
	21. Mengembalikan ke halaman pendaftaran operasi
Alternative flow “Memeriksa kelengkapan data”	
4.a Apabila data yang dimasukkan tidak lengkap	
4.b Klik “Submit”	
	4.c Menampilkan <i>alert</i> “Maaf, data tidak lengkap”
4.d Klik “Ok”	
	4.e Mengembalikan ke halaman pendaftaran operasi

d. *Usecase Scenario* Lihat Jadwal Operasi

Usecase scenario lihat jadwal operasi dapat dilihat pada tabel .

Tabel 0.12 *Usecase scenario* lihat jadwal operasi

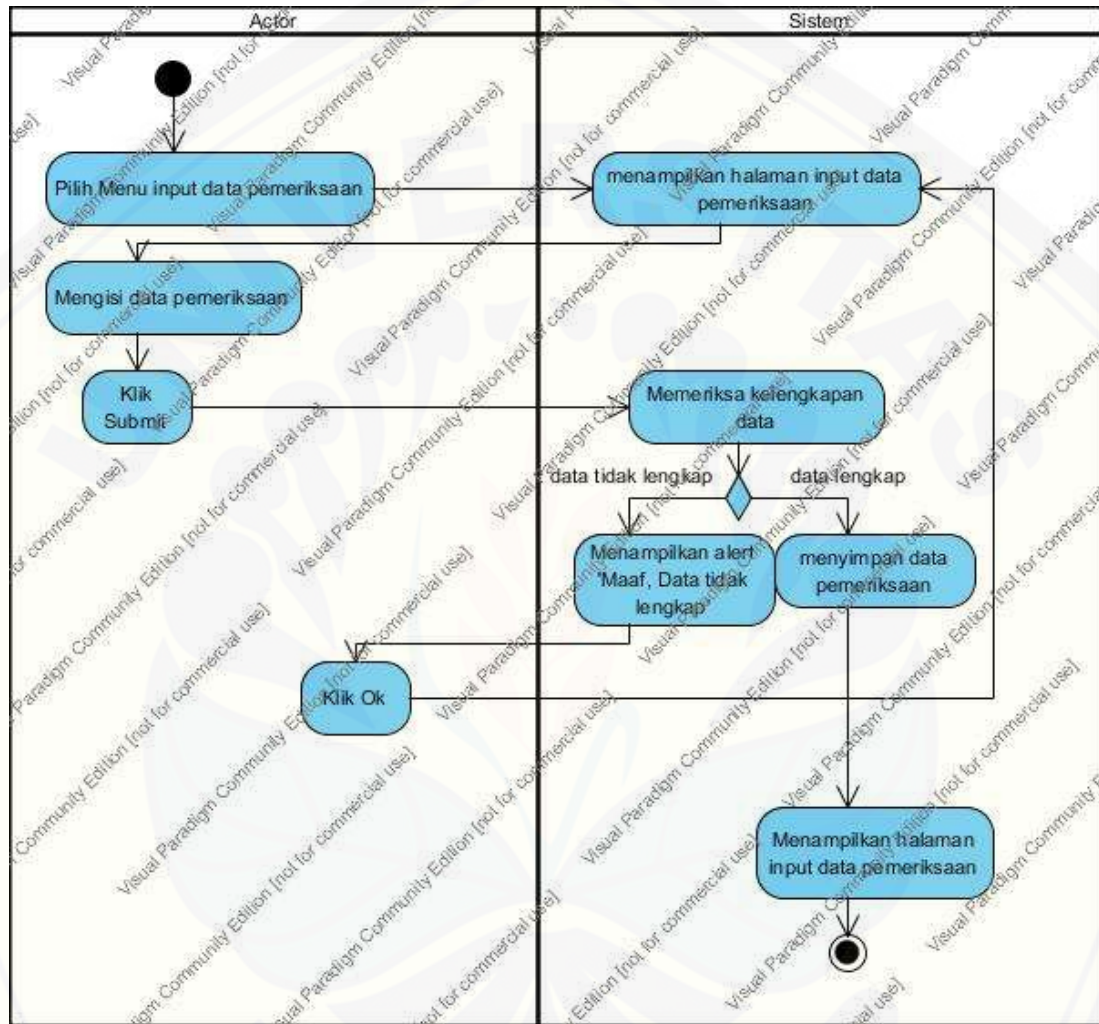
Name	Lihat Jadwal Operasi
Description	Dokter atau Perawat dapat melihat

Actor	Dokter dan Perawat
Precondition	User memilih menu lihat jadwal operasi
Postcondition	Dokter atau pasien melihat jadwal operasi
Normal flow “Lihat Jadwal Operasi”	
<i>Actor</i>	<i>System</i>
1. Pilih menu lihat jadwal operasi	
	2. Menampilkan halaman lihat jadwal operasi
3. Pilih tanggal, bulan dan tahun	
4. Klik “Ok”	
	5. Menampilkan tabel berisi data operasi sesuai tanggal yang dipilih
Alternative flow “user tidak mengisi pilihan tanggal”	
4.a Apabila <i>user</i> tidak memilih tanggal	
4.b Klik “Ok”	
	4.c Menampilkan alert “Pilih Tanggal!”
4.d Klik “Ok”	
	4.e Mengembalikan ke halaman lihat jadwal operasi
Alternative flow “Tanggal yang dicari tidak di ada”	
4.a Apabila tanggal yang di masukkan oleh <i>user</i> tidak ditemukan datanya	
4.b Klik “Ok”	
	4.c Menampilkan alert , “ Data tidak ditemukan!”
4.d Klik “Ok”	
	4.e Menampilkan halaman lihat jadwal operasi

A.3.3 Activity Diagram Tahap Increment 4

a. *Activity Diagram* Input Data Pemeriksaan

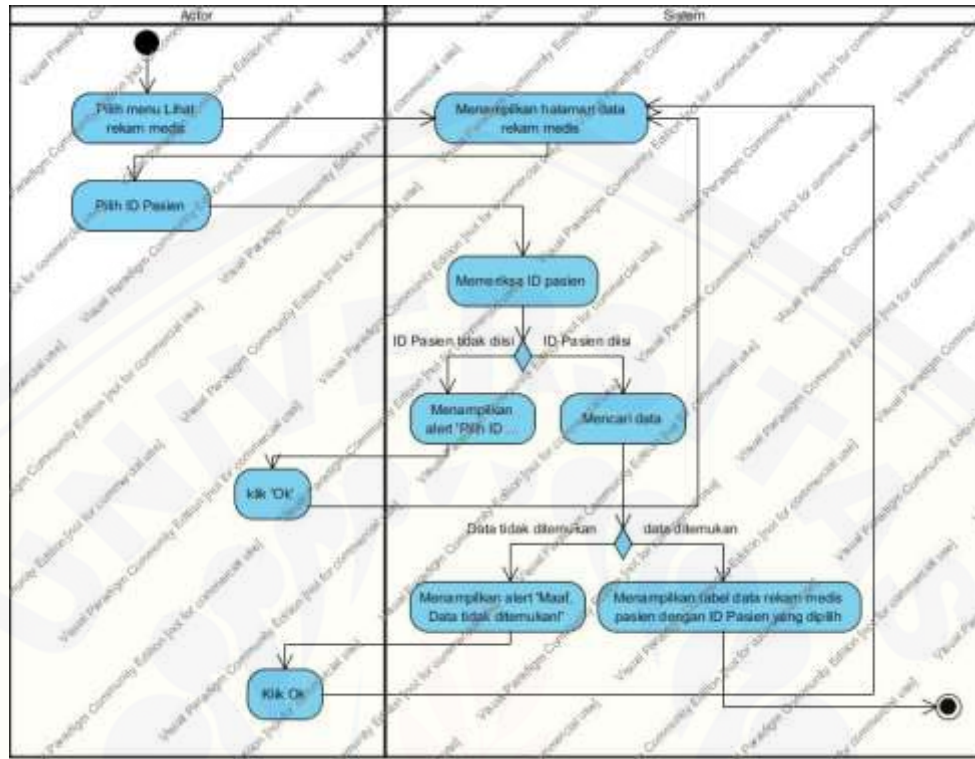
Activity diagram input data pemeriksaan dapat dilihat pada gambar .



Gambar 0.21 *Activity diagram* input data pemeriksaan

b. *Activity Diagram* Lihat Data Rekam Medis

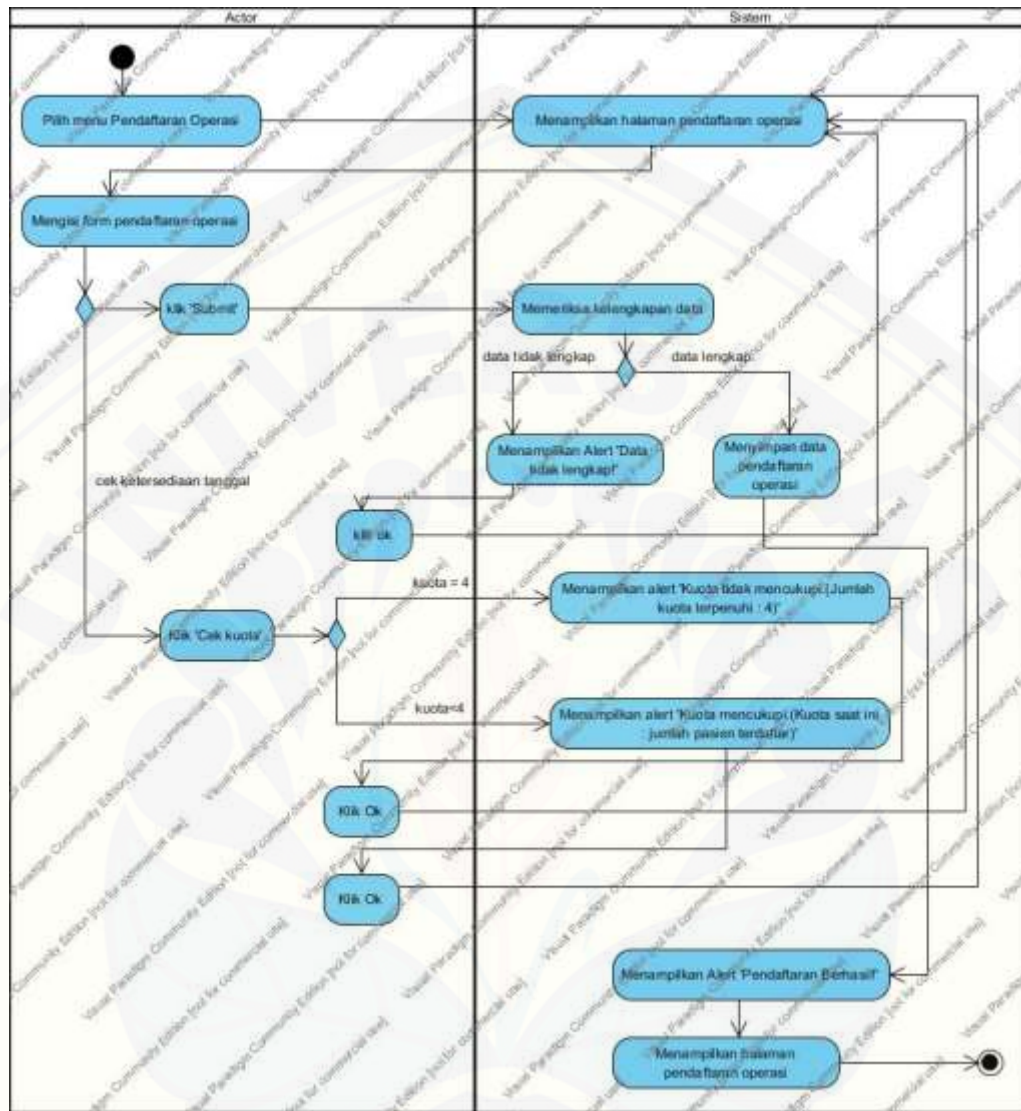
Activity diagram lihat data rekam medis dapat dilihat pada gambar .



Gambar 0.22 Activity diagram lihat data rekam medis

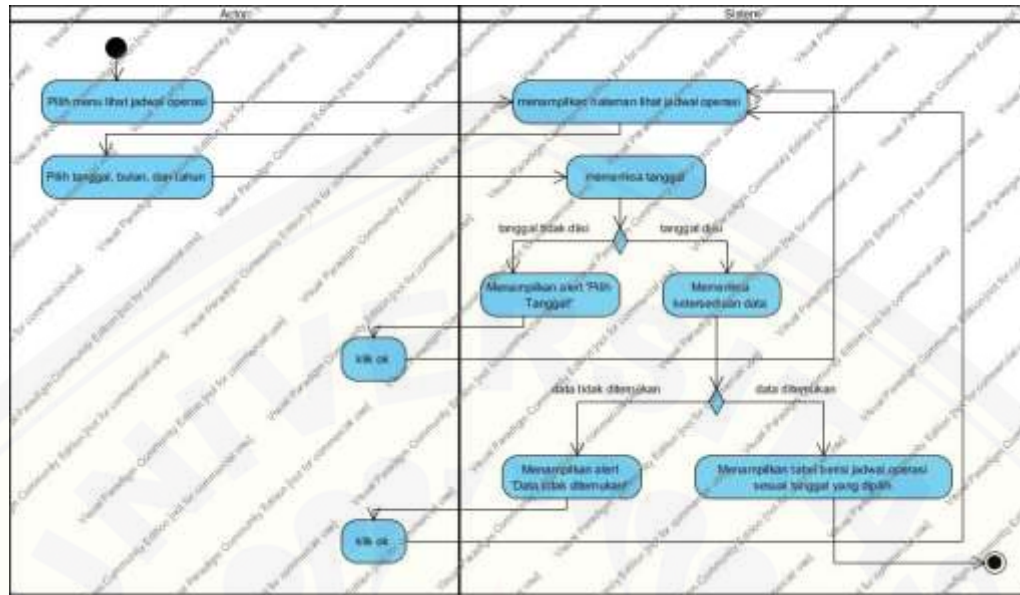
c. *Activity Diagram* Pendaftaran Operasi

Activity diagram pendaftaran operasi dapat dilihat pada gambar .



Gambar 0.23 Activity diagram pendaftaran operasi

- d. Activity Diagram Lihat Jadwal Operasi
Activity diagram lihat jadwal operasi dapat dilihat pada gambar .

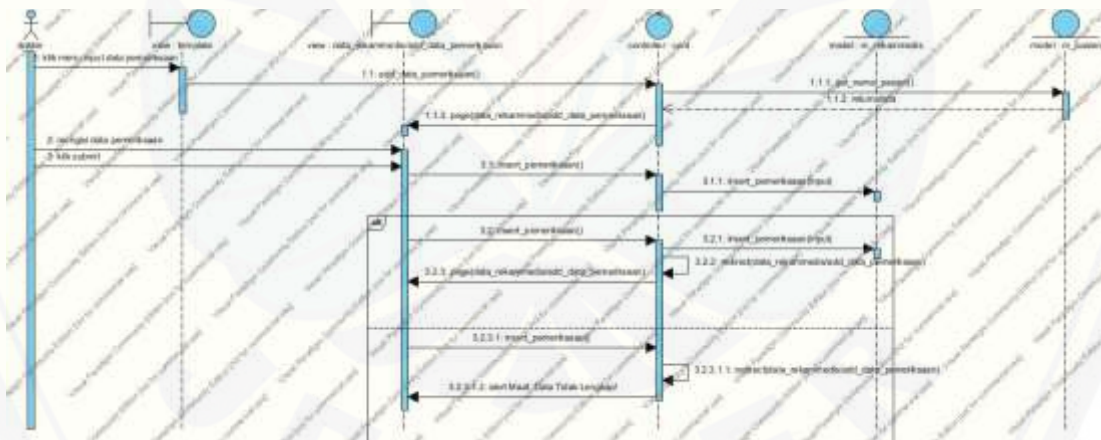


Gambar 0.24 Activity diagram lihat jadwal operasi

A.3.4 Sequence Diagram Tahap Increment 4

a. *Sequence Diagram* Input Data Pemeriksaan

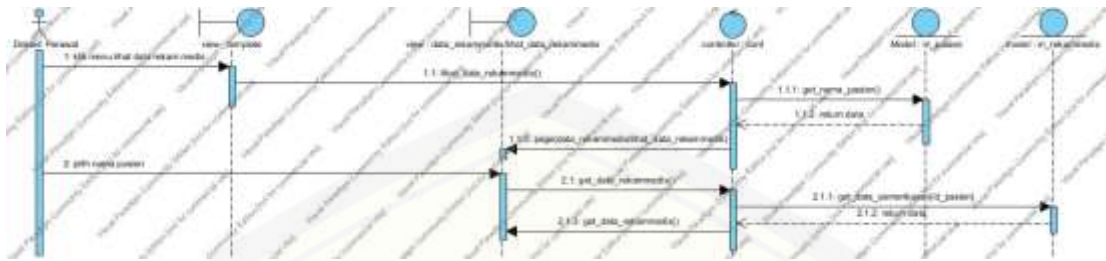
Sequence diagram input data pemeriksaan dapat dilihat pada gambar .



Gambar 0.25 Sequence diagram input data pemeriksaan

b. *Sequence Diagram* Lihat Data Rekam Medis

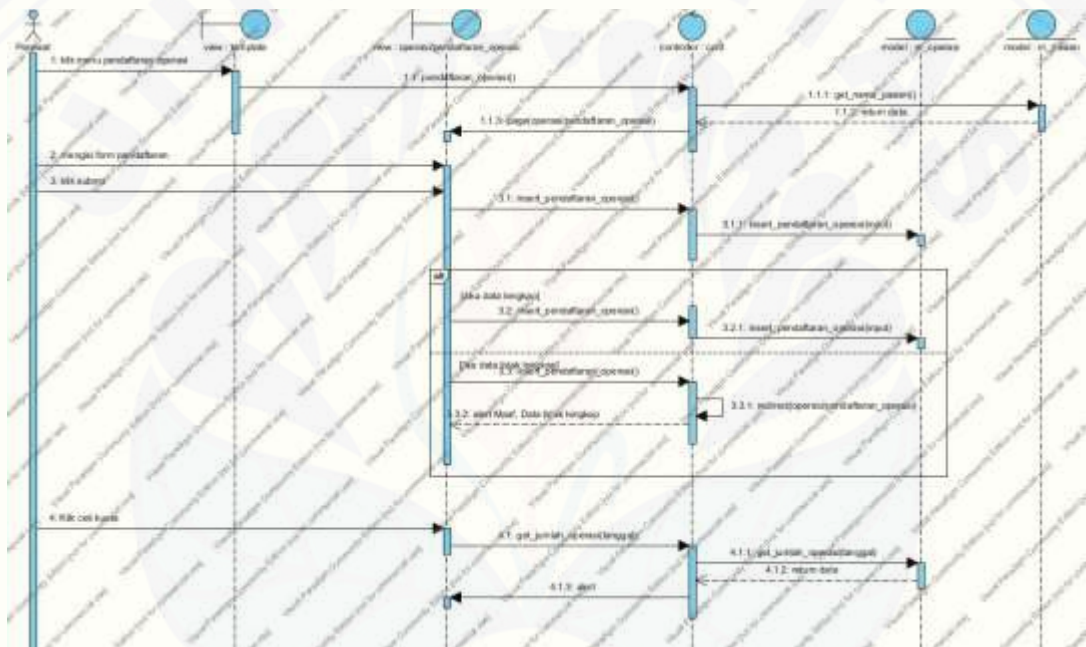
Sequence diagram lihat data rekam medis dapat dilihat pada gambar .



Gambar 0.26 Sequence diagram lihat data rekam medis

c. Sequence Diagram Pendaftaran Operasi

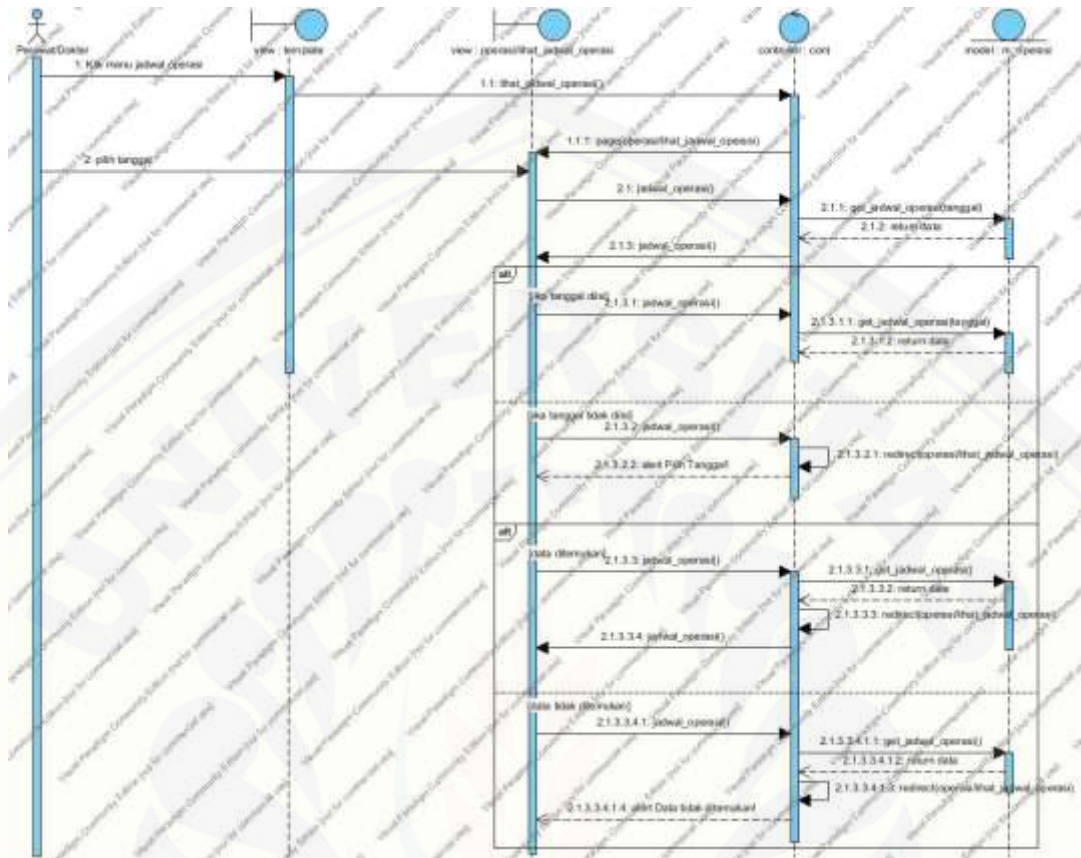
Sequence diagram pendaftaran operasi dapat dilihat pada gambar .



Gambar 0.27 Sequence diagram pendaftaran operasi

d. Sequence Diagram Lihat Jadwal Operasi

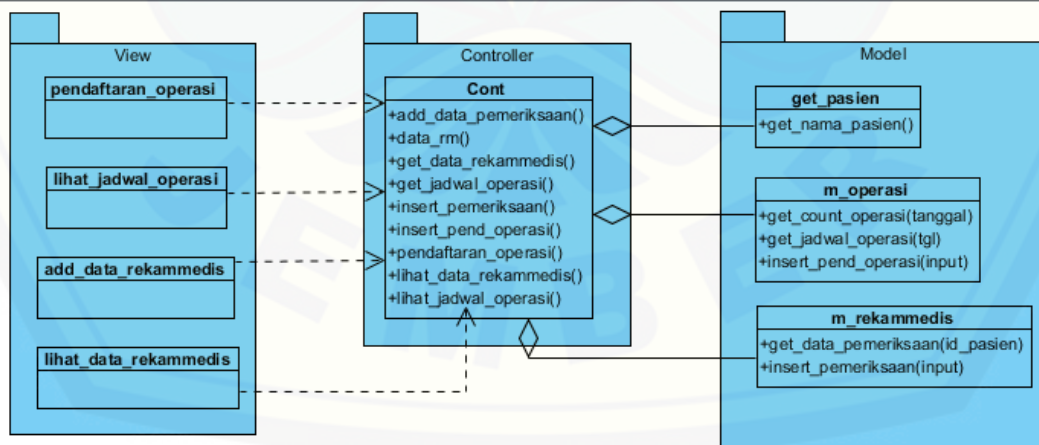
Sequence diagram lihat jadwal operasi dapat dilihat pada gambar .



Gambar 0.28 Sequence diagram lihat jadwal operasi

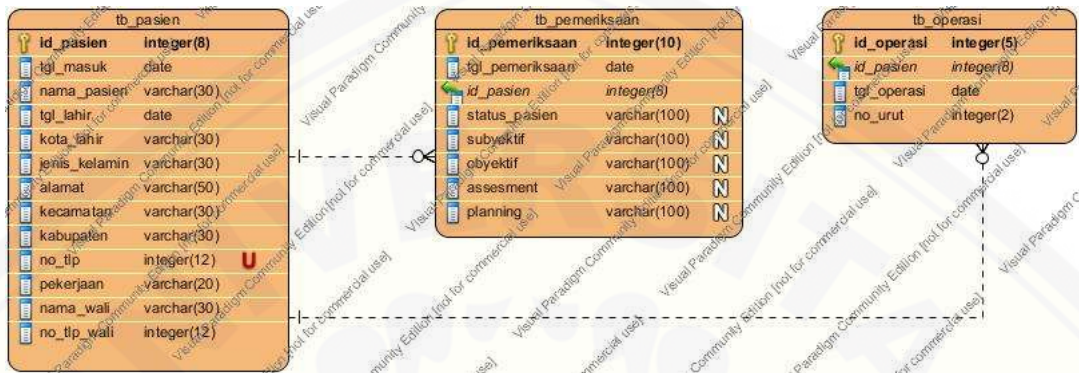
A.3.5 Class Diagram Tahap Increment 4

Class diagram tahap increment empat dapat dilihat pada gambar .



A.3.6 Entity Relationship Diagram Tahap Increment 4

Entity relationship diagram tahap increment empat dapat dilihat pada gambar



Gambar 0.29 ERD tahap increment empat

LAMPIRAN B. IMPLEMENTASI SISTEM**B.1. Implementasi Sistem Tahap *Increment* Dua****a. Implementasi Sistem Data Obat****1. Tampilan menu data obat**

No.	KODE OBAT	NAMA OBAT	STOK	Aksi
1	00001	Gando ketanurung	2000	[edit] [delete]
2	00002	GENOCAPRIS 150 MG	500	[edit] [delete]
3	00003	GENOCAPRIS 150 MG	400	[edit] [delete]
4	00004	GENOCAPRIS 150 MG	450	[edit] [delete]
5	00005	GENOCAPRIS 150 MG	300	[edit] [delete]
6	00006	GENOCAPRIS 150 MG	200	[edit] [delete]
7	00007	GENOCAPRIS 150 MG	100	[edit] [delete]
8	00008	GENOCAPRIS 150 MG	50	[edit] [delete]
9	00009	GENOCAPRIS 150 MG	10	[edit] [delete]

2. Tampilan tambah data obat

Form fields:

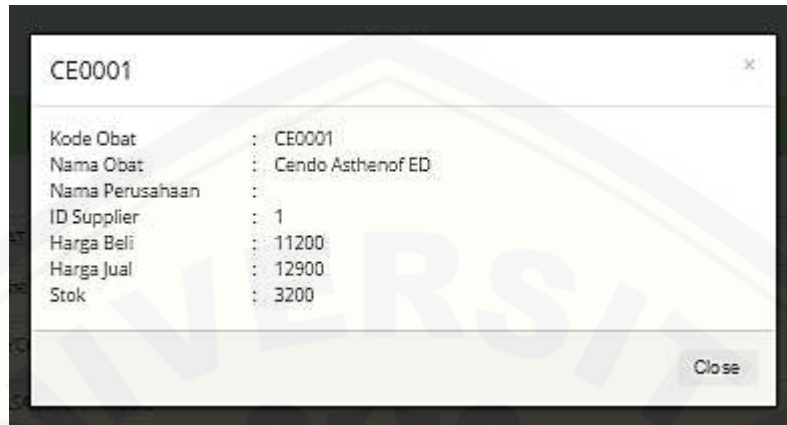
- Nama Obat:
- Nama Obat:
- Supplier:
- Harga Beli:
- Harga Jual:
- Stock:

3. Tampilan edit data obat

Form fields:

- Nama Obat:
- Nama Obat:
- Supplier:
- Harga Beli:
- Harga Jual:
- Stock:

4. Tampilan view data obat



b. Implementasi Sistem Data Supplier

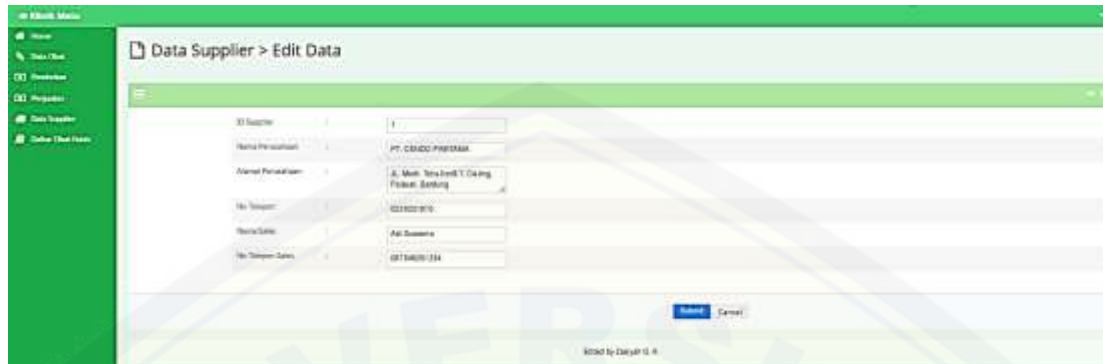
1. Tampilan menu data supplier



2. Tampilan tambah data supplier



3. Tampilan edit data supplier



4. Tampilan view data supplier



c. Implementasi Sistem Daftar Obat Habis



PRAKTEK DOKTER MATA dr. Jamaluddin Sp.M
Jl. Bhayangkara M. Husein 29 Bangkalan, Telp. (031) - XXXXXX

KARTU STOCK OBAT

No.	Kode Obat	Nama Obat	Stok	Nama Sales	Telp Sales	Keterangan
1	RD0001	RD010 10 ML	50	Syaifuldin	089836541121	

d. Implementasi Sistem Transaksi Pembelian Obat

Transaksi Pembelian

Nama Obat: RD0001 - GandaApnea/10

Qty:

Harga:

Add

Penger Transaksi: RD 10.000

Total : 0

Prinsip	Nama Obat	Qty	Harga	Subtotal
---------	-----------	-----	-------	----------

Simpan

e. Implementasi Sistem Transaksi Penjualan Obat

Transaksi Penjualan

Nama Obat: [Dropdown: 00001 - Geroch Acromed 02]

Dpt: [Text]

Harga: [Text]

[Simpan]

Tanggal Transaksi: [Date: 03/10/2018]

ID Pasien: [Dropdown]

Nama-Pemilik: [Text]

Total : 0

No	Kode Obat	Nama Obat	Dpt	Subtotal	Rapun
----	-----------	-----------	-----	----------	-------

[Simpan]

B.2. Implementasi Sistem Tahap *Increment* Tiga

- Implementasi Sistem Data Frame
- Implementasi Sistem Transaksi Penjualan Kacamata

Transaksi Penjualan

Nama: [Dropdown: Kacamata Hitam 00001]

Kode: [Dropdown]

Harga: [Text]

[Simpan]

Tanggal Transaksi: [Date: 03/10/2018]

Nama Pasien: [Text]

Nama-Pemilik: [Text]

Total : 0

No	Nama	Kode	Harga	Rapun
----	------	------	-------	-------

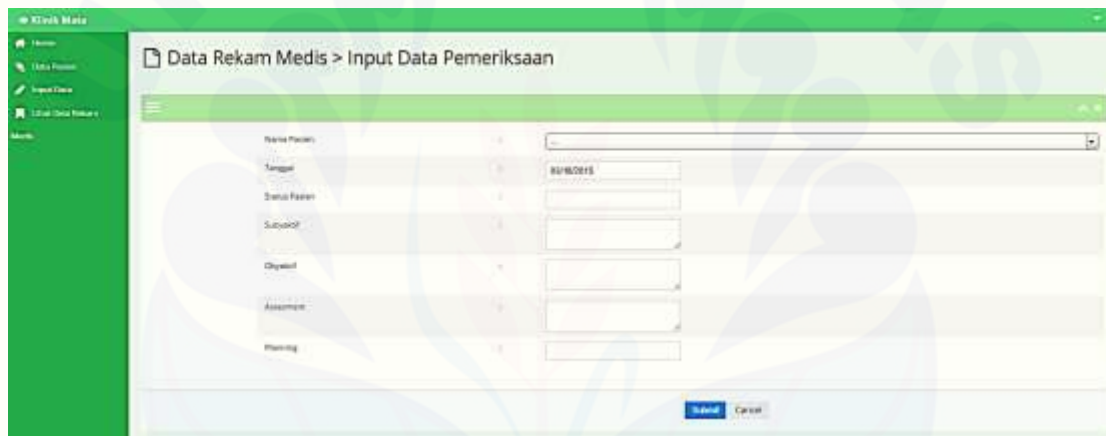
[Simpan]

- Implementasi Sistem Transaksi Pelunasan Kacamata



B.3. Implementasi Sistem Tahap *Increment* Empat

a. Implementasi Sistem Input Data Pemeriksaan



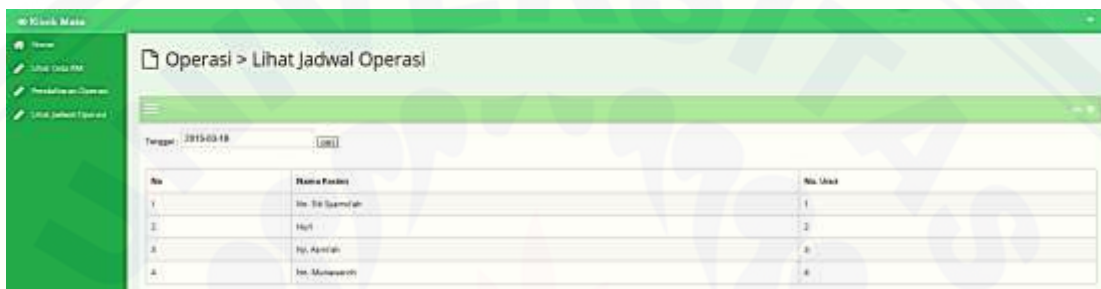
b. Implementasi Sistem Lihat Data Rekam Medis



c. Implementasi Sistem Pendaftaran Operasi



d. Implementasi Sistem Lihat Jadwal Operasi



LAMPIRAN C. PENULISAN KODE PROGRAM

C.1. Penulisan Kode Program Tahap *Increment 1*

a. Data User

A. Tambah Data User
- Controller Tambah Data User (cont.php/insert_user)
<pre>public function insert_user(){ \$input = array("nama_ac" => \$this->input->post('nama_ac'), "username" => \$this->input->post('username'), "password" => \$this->input->post('password'), "level" => \$this->input->post('level'), "alamat" => \$this->input->post('alamat'), "tlp" => \$this->input->post('tlp')); if(\$this->m_user->cek_username(\$input['username']) == TRUE){ redirect(base_url() . "index.php/cont/add_data_user?status=gagal"); }else{ \$logic = \$this->m_user->insert_user(\$input); redirect(base_url() . "index.php/cont/data_user?status=\$logic"); } }</pre>
- Model Tambah Data User (m_user /insert_user)
<pre>function insert_user(\$input) { \$result = \$this->db->insert('account', \$input); return \$result; }</pre>
B. Edit Data Obat
- Controller Edit Data User (cont.php/edit_user)
<pre>function edit_user() { \$input = array("id_ac" => \$this->input->post('id_ac'),</pre>

<pre> "nama_ac" => \$this->input->post('nama_ac'), "username" => \$this->input->post('username'), "password" => \$this->input->post('password'), "level" => \$this->input->post('level'), "alamat" => \$this->input->post('alamat'), "tlp" => \$this->input->post('tlp')); \$logic = \$this->m_user->update_user(\$input); redirect(base_url() . "index.php/cont/data_user?status=\$logic"); } </pre>
- Model Edit Data User (m_user /edit_user)
<pre> function update_user(\$input) { \$this->db->where('id_ac', \$input['id_ac']); return \$this->db->update('account', \$input); } </pre>
C. Hapus Data User
- Controller Hapus Data User (cont.php/hapus_user)
<pre> function hapus_user(\$id_ac){ \$this->m_user->hapus_user(\$id_ac); redirect(base_url() . "index.php/cont/data_user?status=\$id_ac"); } </pre>
- Model Hapus Data User (m_user /hapus_user)
<pre> function hapus_user(\$id_ac) { \$query = \$this->db->query("DELETE FROM account WHERE id_ac = '" . \$id_ac . "'"); if (\$query) { RETURN TRUE; } else { RETURN FALSE; } } } </pre>

b. Buat Akun

A. Tambah Buat Akun
- Controller Buat Akun (cont_registrasio.php/insert_akun)
<pre> public function insert_akun(){ \$input = array("nama_ac" => \$this->input->post('nama_ac'), "email" => \$this->input->post('email'), "username" => \$this->input->post('username'), "password" => \$this->input->post('password'), "level" => 7, "alamat" => \$this->input->post('alamat'), "tlp" => \$this->input->post('tlp'), "status" => "No"); if(\$this->m_user->cek_username(\$input['username']) == TRUE){ \$this->session->set_flashdata("message","Username Telah Digunakan!"); redirect("cont_registrasio/create_account"); }else{ \$logic = \$this->m_user->register(\$input); if(\$logic){ \$this->session->set_flashdata("message",\$input['username'].", Silahkan aktivasi melalui email anda"); redirect("cont_registrasio/login"); } else{ \$this->session->set_flashdata("message","Registrasi gagal"); redirect("cont_registrasio/create_account"); } } } </pre>
- Model Buat Akun (m_user/register)
<pre> function register(\$input){ </pre>


```

// email
if($this->send_email($input)){
    // insert database
    return $this->insert_user($input);
}
else{
    return false;
}
}

```

c. **Aktivasi Akun Lewat *Email***

A. Aktivasi Akun Lewat *Email*

- **Controller Aktivasi Akun Lewat *Email* (cont_registrasio.php/confirm)**

```

public function confirm($username){
    $result = $this->m_user->confirm($username);
    if($result){
        $this->session->set_flashdata("message",$username." telah aktif");
    }
    else{
        $this->session->set_flashdata("message",$username." gagal di aktivasi");
    }
    redirect("cont_registrasio/login");
}

```

- **Model Aktivasi Akun Lewat *Email* (m_user/confirm)**

```

public function confirm($username){
    $this->db->where('username', $username);
    return $this->db->update("account",array("status"=>"Yes"));
}

```

d. **Registrasi Pasien Baru**

A. Registrasi Pasien Baru

- Controller Registrasi Pasien Baru (cont_registrasio.php/registrasi_pasien)

```
function registrasi_baru(){
    $input_pasien = array("id_pasien" => $this->input->post('id_pasien'),
        "tgl_masuk" => date('Y-m-d', strtotime($this->input->post('tgl_masuk'))),
        "nama_pasien" => $this->input->post('nama_pasien'),
        "tgl_lahir" => date('Y-m-d', strtotime($this->input->post('tgl_lahir'))),
        "kota_lahir" => $this->input->post('kota_lahir'),
        "jenis_kelamin" => $this->input->post('jk_pasien'),
        "alamat" => $this->input->post('alamat_pasien'),
        "kecamatan" => $this->input->post('kecamatan'),
        "kabupaten" => $this->input->post('kabupaten'),
        "no_tlp" => $this->input->post('no_tlp'),
        "pekerjaan" => $this->input->post('pekerjaan'),
        "nama_wali" => $this->input->post('nama_wali'),
        "no_tlp_wali" => $this->input->post('no_tlp_wali'));
    $insert_pasien = $this->m_pasien->insert_pasien($input_pasien);
    if($insert_pasien){
        $input_reg = array(
            "no_registrasi" => $this->input->post('no_registrasi'),
            "tgl_registrasi" => date('Y-m-d', strtotime($this->input->post('tgl_registrasi'))),
            "id_pasien" => $this->input->post('id_pasien') );
        $keluhan = $this->input->post('keluhan');
        $reg = $this->m_registrasi->prediksi($input_reg, $keluhan);
        if($reg){
            $this->session->set_flashdata("message",$input_pasien['nama_pasien'].", Telah
Terdaftar");
            redirect(base_url() . "index.php/cont_registrasio/lihat_waktu?status=sukses");
        }else{
            $this->session->set_flashdata("message","Pendaftaran Gagal");
            redirect(base_url() . "index.php/cont_registrasio/reg_pasien_baru?status=gagal");
        }
        $this->session->set_flashdata("message",$input_pasien['nama_pasien'].", Telah
Terdaftar");
    }
}
```

```

        redirect(base_url() . "index.php/cont_registrasio/lihat_waktu?status=sukses");
    }else{
        $this->session->set_flashdata("message","Pendaftaran Gagal");
        redirect(base_url() . "index.php/cont_registrasio/reg_pasien_baru?status=gagal");
    } }

```

e. Data Pasien

A. Tambah Data Pasien

- Controller Tambah Data Pasien (cont.php/insert_pasien)

```

function insert_pasien() {
    $input = array(
        "id_pasien" => $this->input->post('id_pasien'),
        "tgl_masuk" => date('Y-m-d', strtotime($this->input->post('tgl_masuk'))),
        "nama_pasien" => $this->input->post('nama_pasien'),
        "tgl_lahir" => date('Y-m-d', strtotime($this->input->post('tgl_lahir'))),
        "kota_lahir" => $this->input->post('kota_lahir'),
        "jenis_kelamin" => $this->input->post('jk_pasien'),
        "alamat" => $this->input->post('alamat_pasien'),
        "kecamatan" => $this->input->post('kecamatan'),
        "kabupaten" => $this->input->post('kabupaten'),
        "no_tlp" => $this->input->post('no_tlp'),
        "pekerjaan" => $this->input->post('pekerjaan'),
        "nama_wali" => $this->input->post('nama_wali'),
        "no_tlp_wali" => $this->input->post('no_tlp_wali')
    );
    $logic = $this->m_pasien->insert_pasien($input);

    redirect(base_url() . "index.php/cont/data_pasien?status=$logic");
}

```

- Model Tambah Data Pasien (m_pasien /insert_pasien)

```

function insert_pasien($input){
    $result = $this->db->insert('tb_pasien', $input);
}

```

<pre> return \$result; } </pre>
B. Edit Data Pasien
- Controller Edit Data Pasien (cont.php/edit_pasien)
<pre> function edit_pasien() { \$input = array("id_pasien" => \$this->input->post('id_pasien'), "tgl_masuk" => \$this->input->post('tgl_masuk'), "nama_pasien" => \$this->input->post('nama_pasien'), "tgl_lahir" => \$this->input->post('tgl_lahir'), "kota_lahir" => \$this->input->post('kota_lahir'), "jenis_kelamin" => \$this->input->post('jk_pasien'), "alamat" => \$this->input->post('alamat_pasien'), "kecamatan" => \$this->input->post('kecamatan'), "kabupaten" => \$this->input->post('kabupaten'), "no_tlp" => \$this->input->post('no_tlp'), "pekerjaan" => \$this->input->post('pekerjaan'), "nama_wali" => \$this->input->post('nama_wali'), "no_tlp_wali" => \$this->input->post('no_tlp_wali')); \$logic = \$this->m_pasien->update_pasien(\$input); redirect(base_url() . "index.php/cont/data_pasien?status=\$logic"); } </pre>
- Model Edit Data Pasien (m_pasien /update_pasien)
<pre> function update_pasien(\$input) { \$this->db->where('id_pasien', \$input['id_pasien']); return \$this->db->update('tb_pasien', \$input); } </pre>
C. View Data Pasien
- Controller View Data Pasien (cont.php/detail_pasien)
<pre> function detail_pasien(\$id_pasien) { \$data = \$this->m_pasien->get_pasien_id(\$id_pasien)->row_array(); \$this->load->view('include/datapasien/detail_pasien', \$data); } </pre>

}
- Model View Data Pasien (m_pasien /get_pasien_id)
<pre>function get_pasien_id(\$id_pasien) { \$query = 'select * from tb_pasien where id_pasien = ' . \$id_pasien; \$data = \$this->db->query(\$query); return \$data; }</pre>

C.2. Penulisan Kode Program Tahap *Increment 2*

a. Data Obat

A. Tambah Data Obat
- Controller Tambah Data Obat (cont.php/insert_obat)
<pre>function insert_obat() { \$input = array("kode_obat" => \$this->input->post('kode_obat'), "nama_obat" => \$this->input->post('nama_obat'), "id_supplier" => \$this->input->post('id_supplier'), "harga_jual" => \$this->input->post('harga_jual'), "harga_beli" => \$this->input->post('harga_beli'), "stok" => \$this->input->post('stok')); \$logic = \$this->m_obat->insert_obat(\$input); redirect(base_url() . "index.php/cont/data_obat?status=\$logic"); }</pre>
- Model Tambah Data Obat (m_obat/insert_obat)
<pre>function insert_obat(\$input) { \$result = \$this->db->insert('tb_obat', \$input);</pre>

<pre> return \$result; } </pre>
B. Edit Data Obat
- Controller Edit Data Obat (cont.php/edit_obat)
<pre> function edit_obat() { \$input = array("kode_obat" => \$this->input->post('kode_obat'), "nama_obat" => \$this->input->post('nama_obat'), "id_supplier" => \$this->input->post('id_supplier'), "harga_jual" => \$this->input->post('harga_jual'), "harga_beli" => \$this->input->post('harga_beli'), "stok" => \$this->input->post('stok')); \$logic = \$this->m_obat->update_obat(\$input); redirect(base_url() . "index.php/cont/data_obat?status=\$logic"); } </pre>
Model Edit Data Obat (m_obat/edit_obat)
<pre> function update_obat(\$input) { \$this->db->where('kode_obat', \$input['kode_obat']); return \$this->db->update('tb_obat', \$input); } </pre>
C. Hapus Data Obat
- Controller Hapus Data Obat (cont.php/hapus_obat)
<pre> function hapus_obat(\$kode_obat){ \$this->m_obat->hapus_obat(\$kode_obat); redirect(base_url() . "index.php/cont/data_obat?status=\$kode_obat"); } </pre>
- Model Hapus Data Obat (m_obat/hapus_obat)
<pre> function hapus_obat(\$kode_obat) { \$query = \$this->db->query("DELETE FROM tb_obat WHERE kode_obat = " . \$kode_obat </pre>

```
. "");
```

```
if ($query) {
    RETURN TRUE;
} else {
    RETURN FALSE;
}
}
```

D. View Data Obat

- Controller View Data Obat (cont.php/detail_obat)

```
function detail_obat($kode_obat) {
    $data = $this->m_obat->get_obat_id($kode_obat)->row_array();
    $this->load->view('include/data_obat/detail', $data);
}
```

- Model View Data Obat (m_obat/get_obat_id)

```
function get_obat_id($kode_obat) {
    $query = "select * from tb_obat
            where kode_obat =" . $kode_obat . """;
    $result = $this->db->query($query);
    return $result;
}
```

b. Data Supplier

A. Tambah Data Supplier

- Controller Tambah Data Supplier (cont.php/insert_supplier)

```
function insert_supplier() {

    $input = array(
        "id_supplier" => $this->input->post('id_supplier'),
        "nama_supplier" => $this->input->post('nama_supplier'),
        "alamat_supplier" => $this->input->post('alamat_supplier'),
```

<pre> "tlp_supplier" => \$this->input->post('tlp_supplier'), "nama_sales" => \$this->input->post('nama_sales'), "tlp_sales" => \$this->input->post('tlp_sales')); \$insert = \$this->m_obat->insert_supplier(\$input); redirect(base_url() . "index.php/cont/data_supplier?status=\$insert"); } </pre>
- Model Tambah Data Supplier (m_obat /insert_ supplier)
<pre> function insert_supplier(\$input) { \$query = \$this->db->insert('tb_supplier', \$input); return \$query; } </pre>
B. Edit Data Supplier
- Controller Edit Data Supplier (cont.php/edit_ supplier)
<pre> function edit_supplier() { \$input = array("id_supplier" => \$this->input->post('id_supplier'), "nama_supplier" => \$this->input->post('nama_supplier'), "alamat_supplier" => \$this->input->post('alamat_supplier'), "tlp_supplier" => \$this->input->post('tlp_supplier'), "nama_sales" => \$this->input->post('nama_sales'), "tlp_sales" => \$this->input->post('tlp_sales')); \$logic = \$this->model->update_supplier(\$input); redirect(base_url() . "index.php/cont/data_obat?status=\$logic"); } </pre>
- Model Edit Data Supplier (m_obat/edit_ supplier)
<pre> function update_supplier(\$input) { \$this->db->where('id_supplier', \$input['id_supplier']); } </pre>

<pre> return \$this->db->update('tb_supplier', \$input); } </pre>
C. Hapus Data Supplier
- Controller Hapus Data Supplier (cont.php/hapus_ supplier)
<pre> function hapus_supplier(\$id_supplier){ \$this->m_obat->hapus_supplier(\$id_supplier); redirect(base_url() . "index.php/cont/data_supplier?status=\$id_supplier"); } </pre>
- Model Hapus Data Supplier (m_obat /hapus_ supplier)
<pre> function hapus_supplier(\$id_supplier) { \$query = \$this->db->query("DELETE FROM tb_supplier WHERE id_supplier = '" . \$id_supplier . "'"); if (\$query) { RETURN TRUE; } else { RETURN FALSE; } } </pre>
D. View Data Supplier
- Controller View Data Supplier (cont.php/detail_ supplier)
<pre> function detail_supplier(\$id_supplier) { \$data = \$this->m_obat->get_supplier_id(\$id_supplier)->row_array(); \$this->load->view('include/data_supplier/detail_supplier', \$data); } </pre>
- Model View Data Supplier (m_obat/get_ supplier _id)
<pre> function get_supplier_id(\$id_supplier) { \$query = 'select * from tb_supplier where id_supplier =' . \$id_supplier; \$result = \$this->db->query(\$query); return \$result; } </pre>

c. Daftar Obat Habis

A. Daftar Obat Habis
- Controller Daftar Obat Habis (cont.php/daftar_obat_habis)
<pre>public function daftar_obat_habis() { \$data['obat_habis'] = \$this->m_obat->get_obat_habis(); \$this->page('include/data_obat/daftar_obat_habis', \$data); }</pre>
- Model Daftar Obat Habis (m_obat/get_obat_habis)
<pre>function get_obat_habis() { \$query = 'select a.kode_obat, a.nama_obat, a.stok, b.nama_supplier, b.nama_sales, b.tlp_sales from tb_obat a join tb_supplier b on a.id_supplier = b.id_supplier where stok <= 50'; \$result = \$this->db->query(\$query); return \$result->result_array(); }</pre>

d. Transaksi Pembelian Obat

A. Transaksi Pembelian Obat
- Controller Transaksi Pembelian Obat (cont.php/simpan_pembelian_obat)
<pre>public function simpan_pembelian_obat() { \$tgl_transaksi = date('Y-m-d', strtotime(\$this->input->post('tgl_transaksi'))); \$harga_total = \$this->input->post('total'); //detailnya</pre>

```

$kode_obat = $this->input->post('kode_obat');
$qty = $this->input->post('qty');
$harga = $this->input->post('harga');

$this->m_obat->simpan_pembelian_obat(
    $tgl_transaksi, $harga_total,
    $kode_obat, $qty, $harga);

$this->m_obat->update_stok_obat_beli($kode_obat, $qty);

redirect(base_url()."index.php/cont/trans_pembelian_obat");
}

```

- Model Transaksi Pembelian Obat (m_obat/simpan_pembelian_obat)

```

public function simpan_pembelian_obat($tgl_transaksi, $harga_total, $kode_obat, $qty, $harga)
{
    $data = array(
        "tgl_pembelian" => $tgl_transaksi,
        "harga_total" => $harga_total);

    $insert_1 = $this->db->insert('tb_transaksi_pembelian_obat', $data);

    $id_transaksi = "";
    $selectID = $this->db->query("select id_transaksi from tb_transaksi_pembelian_obat order
by id_transaksi desc limit 1");
    foreach ($selectID->result() as $row) {
        $id_transaksi = $row->id_transaksi;
    }
    $arr_kode_obat = array();
    $arr_qty = array();
    $arr_harga = array();
    $i = 0;
    foreach ($kode_obat as $kode_obat => $a) {
        $arr_kode_obat[$i] = $a;
    }
}

```

```

        $i++;    }
    $i = 0;
    foreach ($qty as $qty => $b) {
        $arr_qty[$i] = $b;
        $i++;    }
    $i = 0;
    foreach ($harga as $harga => $c) {
        $arr_harga[$i] = $c;
        $i++;    }
    for ($k = 0; $k < count($arr_harga); $k++) {
        $insert_detail = $this->db->query("INSERT INTO tb_detail_transaksi_pembelian_obat(
            id_transaksi, kode_obat, qty ) values (
                '" . $id_transaksi . "',
                '" . $arr_kode_obat[$k] . "',
                '" . $arr_qty[$k] . "
            )");
    }
}

```

e. Transaksi Penjualan Obat

A. Transaksi Penjualan Obat

- Controller Transaksi Penjualan Obat (cont.php/ simpan_penjualan_obat)

```

public function simpan_penjualan_obat() {
    $tgl_transaksi = date('Y-m-d', strtotime($this->input->post('tgl_transaksi')));
    $id_pasien = $this->input->post('id_pasien');
    $biaya_pemeriksaan = $this->input->post('biaya_pemeriksaan');
    $harga_total = $this->input->post('total');

    //detailnya
    $kode_obat = $this->input->post('kode_obat');
    $qty = $this->input->post('qty');
    $harga = $this->input->post('harga');
}

```



```

$this->m_obat->simpan_penjualan_obat(
    $tgl_transaksi, $id_pasien, $biaya_pemeriksaan, $harga_total,
    $kode_obat, $qty, $harga);

$this->m_obat->update_stok_obat_jual($kode_obat, $qty);

$data['struuk'] = $this->m_obat->struuk();

$this->load->view('include/cetak/cetak/cetakNotaPenjObat', $data);
//redirect(base_url()."index.php/cont/trans_penjualan_obat");
}

```

- Model Transaksi Penjualan Obat (m_obat/ simpan_penjualan_obat)

```

public function simpan_penjualan_obat($tgl_transaksi, $id_pasien, $biaya_pemeriksaan,
    $harga_total, $kode_obat, $qty, $harga) {
    $data = array(
        "tgl_transaksi" => $tgl_transaksi,
        "id_pasien" => $id_pasien,
        "biaya_pemeriksaan" => $biaya_pemeriksaan,
        "harga_total" => $harga_total);

    $insert_1 = $this->db->insert('tb_transaksipenjualanobat', $data);

    $id_transaksi = "";
    $selectID = $this->db->query("select id_transaksi from "
        . "tb_transaksipenjualanobat order by id_transaksi desc limit 1");
    foreach ($selectID->result() as $row) {
        $id_transaksi = $row->id_transaksi;
    }
    $arr_kode_obat = array();
    $arr_qty = array();
    $arr_harga = array();
}

```

```

$i = 0;
foreach ($kode_obat as $kode_obat => $a) {
    $arr_kode_obat[$i] = $a;
    $i++;
}
$i = 0;
foreach ($qty as $qty => $b) {
    $arr_qty[$i] = $b;
    $i++;
}
$i = 0;
foreach ($harga as $harga => $c) {
    $arr_harga[$i] = $c;
    $i++;
}
for ($k = 0; $k < count($arr_harga); $k++) {
    $insert_detail = $this->db->query("INSERT INTO tb_detail_transaksipenjualanobat(
        id_transaksi,kode_obat, qty,harga) values (
            " . $id_transaksi . ",
            " . $arr_kode_obat[$k] . ",
            " . $arr_qty[$k] . ",
            " . $arr_harga[$k] . "
        )");
}
}
}

```

C.3. Penulisan Kode Program Tahap *Increment 3*

a. Data Frame

A. Tambah Data Frame

- Controller Tambah Data Frame (cont.php/insert_frame)

```
function insert_frame() {
```

```

$input = array(
    "id_frame" => $this->input->post('id_frame'),
    "merk_frame" => $this->input->post('merk_frame'),
    "detail_frame" => $this->input->post('detail_frame'),
    "harga_beli" => $this->input->post('harga_beli'),
    "harga_jual" => $this->input->post('harga_jual'),
    "stok" => $this->input->post('stok')
);
$insert = $this->m_kacamata->insert_frame($input);

redirect(base_url() . "index.php/cont/data_frame?status=$insert");
}

```

- Model Tambah Data Frame (m_kacamata /insert_frame)

```

function insert_frame($input){
    $query = $this->db->insert('tb_frame', $input);
    return $query;
}

```

B. Edit Data Frame

- Controller Edit Data Frame (cont.php/edit_frame)

```

function edit_frame() {

    $input = array(
        "id_frame" => $this->input->post('id_frame'),
        "merk_frame" => $this->input->post('merk_frame'),
        "detail_frame" => $this->input->post('detail_frame'),
        "harga_beli" => $this->input->post('harga_beli'),
        "harga_jual" => $this->input->post('harga_jual'),
        "stok" => $this->input->post('stok')
    );
    $logic = $this->m_kacamata->update_frame($input);
    redirect(base_url() . "index.php/cont/data_frame?status=$logic");
}

```

- Model Edit Data Frame (m_ kacangata /edit_ frame)
<pre>function update_frame(\$input) { \$this->db->where('id_frame', \$input['id_frame']); return \$this->db->update('tb_frame', \$input); }</pre>
C. Hapus Data Frame
- Controller Hapus Data Frame (cont.php/hapus_ frame)
<pre>function hapus_frame(\$id_frame){ \$this->m_kacamatata->hapus_frame(\$id_frame); redirect(base_url() . "index.php/cont/data_frame?status=\$id_frame"); }</pre>
- Model Hapus Data Frame (m_ kacangata /hapus_ frame)
<pre>function hapus_frame(\$id_frame) { \$query = \$this->db->query("DELETE FROM tb_frame WHERE id_frame = ".\$id_frame.""); if(\$query){ RETURN TRUE; } else { RETURN FALSE; } }</pre>

b. Transaksi Penjualan Kacamata

A. Transaksi Penjualan Kacamata
- Controller Transaksi Penjualan Kacamata (cont.php/simpan_penjualan_kacamata)
<pre>public function simpan_penjualan_kacamata() { \$id_transaksi = \$this->input->post('id_transaksi'); \$tgl_transaksi = date('Y-m-d', strtotime(\$this->input->post('tgl_transaksi'))); \$biaya_pemeriksaan = \$this->input->post('biaya_pemeriksaan');</pre>

```

$id_pasien = $this->input->post('id_pasien');
$harga_total = $this->input->post('total');
$dpp = $this->input->post('dp');
$status = $this->input->post('status');

$skurang = $this->input->post('kurang');

$ttl_pengambilan = date('Y-m-d', strtotime($this->input->post('ttl_pengambilan')));

//detailnya
$id_frame = $this->input->post('id_frame');
$jenis_lensa = $this->input->post('jenis_lensa');
$jenis_kanan = $this->input->post('jenis_kanan');
$ukuran_kanan = $this->input->post('ukuran_kanan');
$jenis_kiri = $this->input->post('jenis_kiri');
$ukuran_kiri = $this->input->post('ukuran_kiri');
$harga = $this->input->post('harga');

$this->m_kacamata->simpan_penjualan_kacamata(
    $ttl_transaksi, $id_pasien, $biaya_pemeriksaan, $harga_total,
    $dpp, $status, $skurang, $ttl_pengambilan, $id_frame, $jenis_lensa,
    $jenis_kanan, $ukuran_kanan, $jenis_kiri, $ukuran_kiri, $harga);
$this->m_kacamata->update_stok_frame_jual($id_frame);
redirect(base_url() . "index.php/cont/trans_penjualan_kcmt");
}

```

- Model	Transaksi	Penjualan
Kacamata(m_kacamata/simpan_penjualan_kacamata)		

```

public function simpan_penjualan_kacamata(
    $ttl_transaksi,
    $id_pasien,
    $biaya_pemeriksaan,
    $harga_total,
    $dpp,

```



```
        $status,  
        $kurang,  
        $tgl_pengambilan,  
        $frame,  
        $jenis_lensa,  
        $jenis_kanan,  
        $ukuran_kanan,  
        $jenis_kiri,  
        $ukuran_kiri,  
        $harga  
    ){  
  
    $data = array(  
        "tgl_transaksi" => $tgl_transaksi,  
        "id_pasien" => $id_pasien,  
        "biaya_pemeriksaan" => $biaya_pemeriksaan,  
        "harga_total" => $harga_total,  
        "dp" => $dp,  
        "status" => $status,  
        "kurang" => $kurang,  
        "tgl_pengambilan" => $tgl_pengambilan);  
  
    $insert_1 = $this->db->insert('tb_transaksipenjualankacamata', $data);  
  
    $id_transaksi = "";  
    $selectID = $this->db->query("select id_transaksi from tb_transaksipenjualankacamata  
order by id_transaksi desc limit 1");  
    foreach($selectID -> result() as $row)  
    {  
        $id_transaksi = $row -> id_transaksi;  
    }  
  
    $arr_frame = array();
```

```
$arr_jenis_lensa = array();
$arr_jenis_kanan = array();
$arr_ukuran_kanan = array();
$arr_jenis_kiri = array();
$arr_ukuran_kiri = array();
$arr_harga = array();

$i = 0;
foreach($frame as $frame => $a){
    $arr_frame[$i] = $a;
    $i++;
}
$i = 0;
foreach($jenis_lensa as $jenis_lensa => $b){
    $arr_jenis_lensa[$i] = $b;
    $i++;
}
$i = 0;
foreach($jenis_kanan as $jenis_kanan => $c){
    $arr_jenis_kanan[$i] = $c;
    $i++;
}
$i = 0;
foreach($ukuran_kanan as $ukuran_kanan => $d){
    $arr_ukuran_kanan[$i] = $d;
    $i++;
}
$i = 0;
foreach($jenis_kiri as $jenis_kiri => $e){
    $arr_jenis_kiri[$i] = $e;
    $i++;
}
$i = 0;
```

```
foreach($ukuran_kiri as $ukuran_kiri => $f){
    $arr_ukuran_kiri[$i] = $f;
    $i++;
}
$i = 0;
foreach($harga as $harga => $g){
    $arr_harga[$i] = $g;
    $i++;
}

for($k = 0;$k<count($arr_harga);$k++){
    $insert_detail = $this->db->query("INSERT INTO
tb_detail_transaksipenjualankacamata(
        id_transaksi,
        id_frame,
        lensa,
        jenis_kanan,
        ukuran_kanan,
        jenis_kiri,
        ukuran_kiri,
        harga
    ) values (
        ".$id_transaksi.",
        ".$arr_frame[$k].",
        ".$arr_jenis_lensa[$k].",
        ".$arr_jenis_kanan[$k].",
        ".$arr_ukuran_kanan[$k].",
        ".$arr_jenis_kiri[$k].",
        ".$arr_ukuran_kiri[$k].",
        ".$arr_harga[$k]."
    )");
} }
```

c. Transaksi Pelunasan Kacamata

A. Transaksi Pelunasan Kacamata			
-	Controller	Transaksi	Pelunasan Kacamata
	(cont.php/transaksi_pelunasan_kacamata)		
	<pre>public function transaksi_pelunasan_kacamata(\$id_transaksi){ \$this->m_kacamata->update_status_lunas(\$id_transaksi); redirect("cont/trans_pelunasan_kcmt"); }</pre>		
-	Model Transaksi Pelunasan Kacamata (m_kacamata/update_status_lunas)		
	<pre>public function update_status_lunas(\$id_transaksi){ \$this->db->where('id_transaksi', \$id_transaksi); return \$this->db->update('tb_transaksipenjualankacamata', array("status" => "LUNAS", "kurang" => 0)); }</pre>		

C.4. Penulisan Kode Program Tahap Increment 4

a. Input Data Pemeriksaan

A. Input Data Pemeriksaan	
-	Controller Input Data Pemeriksaan (cont.php/insert_pemeriksaan)
	<pre>function insert_pemeriksaan() { \$input = array(//AUTO INCREMENT GIMANA "id_pasien" => \$this->input->post('id_pasien'), "tgl_pemeriksaan" => date('Y-m-d', strtotime(\$this->input->post('tgl_periksa'))), "status_pasien" => \$this->input->post('status_pasien'), "subyektif" => \$this->input->post('subyektif'), "obyektif" => \$this->input->post('obyektif'), "assesment" => \$this->input->post('assesment'), "planning" => \$this->input->post('planning'));</pre>

```

$insert = $this->m_rekammedis->insert_pemeriksaan($input);

redirect(base_url() . "index.php/cont/add_data_pemeriksaan?status=$insert");
}

```

- Model Input Data Pemeriksaan (m_rekammedis/insert_pemeriksaan)

```

function insert_pemeriksaan($input){
    $query = $this->db->insert('tb_pemeriksaan', $input);

    return $query;
}

```

b. Lihat Data Rekam Medis

A. Lihat Data Rekam Medis

- Controller Lihat Data Rekam Medis (cont.php/lihat_data_rekammedis)

```

public function get_data_rekammedis() {
    $id = $this->input->post('nama_pasien');
    $dataa = $this->m_rekammedis->get_data_pemeriksaan($id);
    echo "Nama : ".$dataa[0]['nama_pasien']."<br>";
    echo "TTL : ".$dataa[0]['ttl']."<br>";
    echo "Umur : ".$dataa[0]['umur']."<br>";
    echo "Alamat : ".$dataa[0]['alamat']."<br>";
    $string = "
        <table class='table table-striped table-bordered table-hover'>
            <thead>
                <tr>
                    <th style=''>No</th>
                    <th style='width: 20%'>Tanggal</th>
                    <th style='width: 10%'>Status</th>
                    <th style='width: 15%'>Subyektif</th>
                    <th style='width: 15%'>Obyektif</th>
                    <th style='width: 15%'>Assesment</th>
                    <th style='width: 15%'>Planning</th>

```



```

        </tr>
    </thead>
    <tbody id='listdatai'>

        ";

    $i = 0;

    foreach($dataa as $d) :

        $string .= '
        <tr>
            <td>' . ++$i . '</td>
            <td>' . $d['tgl_pemeriksaan'] . '</td>
            <td>' . $d['status_pasien'] . '</td>
            <td>' . $d['subyektif'] . '</td>
            <td>' . $d['obyektif'] . '</td>
            <td>' . $d['assesment'] . '</td>
            <td>' . $d['planning'] . '</td>
        </tr>';

    endforeach;
    $string.="</tbody>
    </table>";

    echo $string;
//    return $string;
}

```

- Model Lihat Data Rekam Medis (m_rekammedis/get_data_pemeriksaan)

```

function get_data_pemeriksaan($id_pasien){
    $query = $this->db->query('SELECT a.id_pasien, a.nama_pasien,
        concat(a.kota_lahir, " ", a.tgl_lahir) as ttl,
        YEAR(CURDATE()) - YEAR(a.tgl_lahir) as umur,
        concat(a.alamat, " ", a.kecamatan, " ", a.kabupaten) as alamat,
        a.no_tlp,
        a.pekerjaan,

```

```

        a.nama_wali,
        a.no_tlp_wali,
        b.tgl_pemeriksaan,
        b.status_pasien,
        b.subyektif,
        b.obyektif,
        b.assesment,
        b.planning
    FROM tb_pasien a join tb_pemeriksaan b on a.id_pasien = b.id_pasien where
a.id_pasien='.$id_pasien);

    return $query->result_array();
}

```

c. Pendaftaran Operasi

A. Pendaftaran Operasi

- Controller Pendaftaran Operasi (cont.php/insert_pend_operasi)

```

function insert_pend_operasi() {
    $input = array(
        "id_pasien" => $this->input->post('id_pasien'),
        "tgl_operasi" => $this->input->post('tgl_operasi'),
        "no_urut" => $this->input->post('kuota')
    );
    $insert = $this->m_operasi->insert_pend_operasi($input);

    redirect(base_url() . "index.php/cont/pendaftaran_operasi?status=$insert");
}

```

- Model Pendaftaran Operasi (m_operasi/insert_pend_operasi)

```

function insert_pend_operasi($input) {
    $query = $this->db->insert('tb_operasi', $input);
}

```

```

return $query;
}

```

d. Lihat Jadwal Operasi

A. Lihat Jadwal Operasi

- Controller Lihat Jadwal Operasi (cont.php/lihat_jadwal_operasi)

```

public function get_jadwal_operasi() {
    $tgl = $this->input->post('tgl_operasi');

    $dataa = $this->m_operasi->get_jadwal_operasi($tgl);
    $string = "";
    $i = 0;

    foreach($dataa as $d) :
        $string .= '
            <tr>
                <td>' . ++$i . '</td>
                <td>' . $d['nama_pasien'] . '</td>
                <td>' . $d['no_urut'] . '</td>

            </tr>';
    endforeach;
    echo $string; }

```

- Model Lihat Jadwal Operasi (m_operasi/get_jadwal_operasi)

```

function get_jadwal_operasi($tgl) {
    $query = $this->db->query("SELECT a.nama_pasien, b.no_urut
        FROM tb_pasien a join tb_operasi b on a.id_pasien = b.id_pasien where b.tgl_operasi =
        '" . $tgl . "'");

    return $query->result_array();
}

```



LAMPIRAN D. BLACKBOX TESTING

D.1 Blackbox Testing Tahap Increment 1

D.1.1 Admin

A. Data User

Pengujian User

Nama Penguji : Dr. Endangwati

Level Hak Akses : Admin


Fitur : Data User

Data User		Status	
Aksi	Skenario	Berhasil	Gagal
1. Pilih menu data user	2. Menampilkan halaman data user berupa data user. Pada setiap baris data user terdapat dua pilihan yaitu edit dan hapus serta pilihan tambah data user yang digunakan untuk menambahkan data user baru.	✓	
3. Klik Tambah	4. Menampilkan form data user	✓	
5. Mengisi form data user	6. Klik "Simpan"		
	7. Memeriksa kelengkapan data	-	
	8. Menyimpan data user baru		
	9. Mengembalikan pada halaman data user	✓	
Normal flow "Edit Data User"		Berhasil	Gagal
10. Klik edit pada baris data user yang ingin di edit.	11. Menampilkan data user yang akan di edit pada form data user	✓	
12. Mengedit data user pada form data user	14. Memeriksa kelengkapan data		
13. Klik "Simpan"	15. Menyimpan data user		
	16. Mengembalikan ke halaman Data user	✓	
Normal flow "Hapus Data User"		Berhasil	Gagal
17. Klik hapus pada baris data user yang ingin dihapus.	18. Menampilkan alert "anda yakin ingin menghapus data ini?"	✓	
19. Klik "Ya"	20. Menghapus data user		
	21. Mengembalikan ke halaman data user	✓	
Alternative flow "Kelengkapan data pada tambah user"		Berhasil	Gagal

7a. Apabila user memasukkan data tidak lengkap.			
7b. Klik "Simpan"	7a. Menampilkan alert "Maaf, data belum lengkap"	✓	
7c. Klik "OK"	7a. Mengembalikan ke form data user	✓	
Alternative flow "Kelengkapan data pada edit user"		Berhasil	Gagal
14a. Apabila user memasukkan data tidak lengkap.			
14b. Klik "Simpan"	14a. Menampilkan alert "Maaf, data belum lengkap"	✓	
14c. Klik "OK"	14a. Mengembalikan ke form data user	✓	
Alternative flow "Batal menghapus data user"		Berhasil	Gagal
18a. Apabila user batal menghapus data user.			
18b. Klik "Batal"	18a. Mengembalikan ke halaman data user	✓	

Pengujian 01 Juni 2017

TTD:


 (Dr. Endangwati)

Pengujian Fitur Data User Hal. 2

Pengujian Fitur Data User Hal. 1

D.1.2 Resepsionis

A. Data Pasien

Pengujian User																																																																																																																			
Nama Penguj: <u>Fitriyana D.</u>																																																																																																																			
Level Hak Akses: <u>Resepsionis</u>																																																																																																																			
Fitur: <u>Data Pasien</u>																																																																																																																			
Data Pasien																																																																																																																			
Normal flow "Tambah data pasien"		Status																																																																																																																	
Aksi	Sistem	Berhasil	Gagal																																																																																																																
1. Klik menu data pasien																																																																																																																			
2. Menampilkan halaman Data pasien berupa data pasien. Pada setiap baris data pasien terdapat pilihan yaitu edit serta pilihan tambah data yang digunakan untuk menambahkan data pasien baru.		✓																																																																																																																	
3. Klik "Tambah"																																																																																																																			
4. Menampilkan form data pasien		✓																																																																																																																	
5. Mengisi form data pasien		✓																																																																																																																	
6. Klik "Submit"																																																																																																																			
7. Menyerkan kelengkapan data		✓																																																																																																																	
8. Menyimpan data pasien baru		✓																																																																																																																	
9. Menampilkan alert "Data berhasil ditambahkan"		✓																																																																																																																	
10. Klik "Ok"		✓																																																																																																																	
11. Menampilkan halaman data pasien yang telah diperbaharui		✓																																																																																																																	
Normal flow "Edit Data Pasien"		Berhasil	Gagal																																																																																																																
<table border="1"> <thead> <tr> <th colspan="4">Normal flow "View Data Pasien"</th> </tr> <tr> <th>Aksi</th> <th>Sistem</th> <th>Berhasil</th> <th>Gagal</th> </tr> </thead> <tbody> <tr> <td>12. Klik "edit" pada baris data pasien yang ingin di edit</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>13. Menampilkan data pasien yang akan di edit pada form data pasien.</td> <td></td> <td>✓</td> <td>layar kelengkapan data yang tidak lengkap / tidak lengkap</td> </tr> <tr> <td>14. Meng edit data pasien pada form data pasien</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>15. Klik "Submit"</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>16. Memeriksa kelengkapan data</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>17. Menyimpan data pasien</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>18. Menampilkan alert "data pasien berhasil diperbaharui"</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>19. Klik "Ok"</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>20. Menampilkan halaman data pasien yang telah diperbaharui.</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td colspan="4">Normal flow "View Data Pasien"</td> </tr> <tr> <td>21. Klik "view" pada baris data pasien yang ingin dilihat</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>22. Menampilkan detail data pasien sesuai yang dipilih</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>23. Klik "close"</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>24. Menampilkan halaman data pasien</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td colspan="2">Alternative flow "kelengkapan data pada tambah pasien"</td> <td>Berhasil</td> <td>Gagal</td> </tr> <tr> <td>25.a Apabila user memasukkan data tidak lengkap.</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>25.b Klik "Submit"</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>25.c Menampilkan alert "Maaf, data belum lengkap"</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>25.d Klik "Ok"</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>25.e Mengembalikan ke form tambah data pasien</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td colspan="2">Alternative flow "Batal menambah data pasien"</td> <td>Berhasil</td> <td>Gagal</td> </tr> <tr> <td>6.a Klik "cancel"</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>6.b Mengembalikan ke halaman data pasien</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td colspan="2">Alternative flow "Batal mengedit data pasien"</td> <td>Berhasil</td> <td>Gagal</td> </tr> <tr> <td>15.a Klik "cancel"</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>15.b Mengembalikan ke halaman data pasien</td> <td></td> <td>✓</td> <td></td> </tr> </tbody> </table>				Normal flow "View Data Pasien"				Aksi	Sistem	Berhasil	Gagal	12. Klik "edit" pada baris data pasien yang ingin di edit		✓		13. Menampilkan data pasien yang akan di edit pada form data pasien.		✓	layar kelengkapan data yang tidak lengkap / tidak lengkap	14. Meng edit data pasien pada form data pasien		✓		15. Klik "Submit"		✓		16. Memeriksa kelengkapan data		✓		17. Menyimpan data pasien		✓		18. Menampilkan alert "data pasien berhasil diperbaharui"		✓		19. Klik "Ok"		✓		20. Menampilkan halaman data pasien yang telah diperbaharui.		✓		Normal flow "View Data Pasien"				21. Klik "view" pada baris data pasien yang ingin dilihat		✓		22. Menampilkan detail data pasien sesuai yang dipilih		✓		23. Klik "close"		✓		24. Menampilkan halaman data pasien		✓		Alternative flow "kelengkapan data pada tambah pasien"		Berhasil	Gagal	25.a Apabila user memasukkan data tidak lengkap.		✓		25.b Klik "Submit"		✓		25.c Menampilkan alert "Maaf, data belum lengkap"		✓		25.d Klik "Ok"		✓		25.e Mengembalikan ke form tambah data pasien		✓		Alternative flow "Batal menambah data pasien"		Berhasil	Gagal	6.a Klik "cancel"		✓		6.b Mengembalikan ke halaman data pasien		✓		Alternative flow "Batal mengedit data pasien"		Berhasil	Gagal	15.a Klik "cancel"		✓		15.b Mengembalikan ke halaman data pasien		✓	
Normal flow "View Data Pasien"																																																																																																																			
Aksi	Sistem	Berhasil	Gagal																																																																																																																
12. Klik "edit" pada baris data pasien yang ingin di edit		✓																																																																																																																	
13. Menampilkan data pasien yang akan di edit pada form data pasien.		✓	layar kelengkapan data yang tidak lengkap / tidak lengkap																																																																																																																
14. Meng edit data pasien pada form data pasien		✓																																																																																																																	
15. Klik "Submit"		✓																																																																																																																	
16. Memeriksa kelengkapan data		✓																																																																																																																	
17. Menyimpan data pasien		✓																																																																																																																	
18. Menampilkan alert "data pasien berhasil diperbaharui"		✓																																																																																																																	
19. Klik "Ok"		✓																																																																																																																	
20. Menampilkan halaman data pasien yang telah diperbaharui.		✓																																																																																																																	
Normal flow "View Data Pasien"																																																																																																																			
21. Klik "view" pada baris data pasien yang ingin dilihat		✓																																																																																																																	
22. Menampilkan detail data pasien sesuai yang dipilih		✓																																																																																																																	
23. Klik "close"		✓																																																																																																																	
24. Menampilkan halaman data pasien		✓																																																																																																																	
Alternative flow "kelengkapan data pada tambah pasien"		Berhasil	Gagal																																																																																																																
25.a Apabila user memasukkan data tidak lengkap.		✓																																																																																																																	
25.b Klik "Submit"		✓																																																																																																																	
25.c Menampilkan alert "Maaf, data belum lengkap"		✓																																																																																																																	
25.d Klik "Ok"		✓																																																																																																																	
25.e Mengembalikan ke form tambah data pasien		✓																																																																																																																	
Alternative flow "Batal menambah data pasien"		Berhasil	Gagal																																																																																																																
6.a Klik "cancel"		✓																																																																																																																	
6.b Mengembalikan ke halaman data pasien		✓																																																																																																																	
Alternative flow "Batal mengedit data pasien"		Berhasil	Gagal																																																																																																																
15.a Klik "cancel"		✓																																																																																																																	
15.b Mengembalikan ke halaman data pasien		✓																																																																																																																	

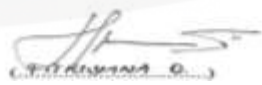
Pengujian Fitur Data Pasien Hal. 1

Pengujian Fitur Data User Hal. 2

	7.a Mengembalikan ke form tambah data pasien	✓	
Alternative flow "kelengkapan data pada edit pasien"			
16.a Apabila user memasukkan data tidak lengkap.		✓	
16.b Klik "Submit"		✓	
16.c Menampilkan alert "Maaf, data belum lengkap"		✓	
16.d Klik "Ok"		✓	
16.e Mengembalikan ke form tambah data pasien		✓	
Alternative flow "Batal menambah data pasien"		Berhasil	Gagal
6.a Klik "cancel"		✓	
6.b Mengembalikan ke halaman data pasien		✓	
Alternative flow "Batal mengedit data pasien"		Berhasil	Gagal
15.a Klik "cancel"		✓	
15.b Mengembalikan ke halaman data pasien		✓	

Bongkaran 20 Maret 2020

TTD,


 (FITRIYANA D.)

Pengujian Fitur Data User Hal. 3

D.1.3 Pasien

A. Buat Akun

Pengujian User

Nama Penguji : Zaki Alwazidi


Level Hak Akses : Pasien

Fitur : Buat Akun

Buat Akun			
Normal flow "Buat Akun"		Status	
Actor	System	Berhasil	Gagal
1. Pilih menu buat akun			
	2. Menampilkan form buat akun	✓	
3. Mengisi data akun			
4. Klik "Submit"			
	5. Memeriksa kelengkapan data		
	6. Mengirim pesan aktivasi ke email pasien	✓	
	7. Menyimpan data akun pasien	✓	
	8. Menampilkan halaman login	✓	
Alternative flow "kelengkapan data"		Berhasil	Gagal
5.a Apabila data yang di masukkan tidak lengkap			
5.b Klik "Submit"			
	5.c Menampilkan alert "Maaf, data tidak lengkap"	✓	
5.d Klik "Ok"			
	5.e Mengembalikan ke form buat akun	✓	

Bngalem, 30 Maret 2015

TID.


 (Zaki Alwazidi)

Pengujian Fitur Buat Akun

B. Aktivasi Akun Lewat Email

Pengujian User

Nama Penguji : Drs. Alimatus
 Level Hak Akses : Pasien
 Fitur : Aktivasi Akun Lewat Email

Aktivasi Akun Lewat Email			
Normal flow "Aktivasi Akun Lewat Email"		Status	
Actor	System	Berhasil	Gagal
1. Klik link aktivasi yang telah dikirim pada email			
	2. Mengupdate status akun pada database	✓	
	3. Menampilkan halaman login sistem	✓	

Jember, 30 Maret 2019
 TTD.

 (...Drs. Alimatus...)

Pengujian Fitur Aktivasi Akun Lewat Email


C. Registrasi Pasien Baru

Pengujian User

Nama Penguji : Drs. Alimatus
 Level Hak Akses : Pasien
 Fitur : Registrasi Pasien Baru

Registrasi Pasien Baru			
Normal flow "Registrasi Pasien Baru"		Status	
Actor	System	Berhasil	Gagal
1. Klik menu registrasi pasien (baru)			
	1. Menampilkan form registrasi pasien	✓	
3. Mengisi form registrasi pasien		✓	
4. Klik "Simpan"		✓	
	2. Memeriksa kelengkapan data	✓	
	4. Menyimpan data registrasi pasien	✓	
	3. Menampilkan informasi dan status	✓	
Alternative flow "Kelengkapan data"		Berhasil	Gagal
3a. Apabila data yang di masukkan tidak lengkap		✓	
3b. Klik "Simpan"		✓	
	3a. Menampilkan alert "Masuk data tidak lengkap"	✓	
3c. Klik "OK"		✓	
	3a. Mengembalikan ke form registrasi pasien	✓	
Alternative flow "Status registrasi"		Berhasil	Gagal
4a. Apabila user tidak melakukan registrasi		✓	
4b. Klik "cancel"		✓	

4a. Menampilkan informasi status	✓	
----------------------------------	---	--

Jember, 30 Maret 2019
 TTD.

 (...Drs. Alimatus...)

Pengujian Fitur Registrasi Pasien Baru Hal. 2

Pengujian Fitur Registrasi Pasien Baru Hal. 1

D. Registrasi Pasien Lama

Pengujian User

Nama Penguji : ...
 Level Hak Akses : ...
 Fitur : Registrasi Pasien Lama

Registrasi Pasien		Status	
Normal flow "Registrasi Pasien Lama"		Berhasil	Gagal
Actor	System		
1. Pilih menu registrasi pasien (lama)		✓	
	2. Menampilkan form registrasi pasien beserta daftar antrian pasien	✓	
3. Mengisi form registrasi pasien		✓	
4. Klik "Submit"		✓	
	5. Memeriksa kelengkapan data	✓	
	6. Menyimpan data registrasi pasien	✓	
	7. Menampilkan alert berisi No Antrian dan Waktu Pemeriksaan	✓	
8. Klik "Cetak Kartu"		✓	
	9. Mencetak kartu antrian	✓	
	10. Menampilkan form registrasi pasien beserta daftar antrian pasien	✓	
Alternative flow "kelengkapan data"		Berhasil	Gagal
5.a Apabila data yang di masukkan tidak lengkap		✓	
5.b Klik "Submit"		✓	
	5.c Menampilkan alert "Maaf, data tidak lengkap"	✓	
5.d Klik "OK"		✓	
	5.e Mengembalikan ke form registrasi pasien	✓	
Alternative flow "batal registrasi"		Berhasil	Gagal
4.a Apabila user batal melakukan registrasi		✓	
4.b Klik "cancel"		✓	
	4.c Mengosongkan form registrasi pasien	✓	

Jember, 21 Maret 2015
 TTD

 (...

Pengujian Fitur Registrasi Pasien Lama

D.2 Blackbox Testing Tahap Increment 2

D.2.1 Bagian Obat


A. Data Obat

Penguji User			
Nama Penguj: <u>ALBA...ELATIN...</u>			
Level Hak Akses: _____			
Fitur: <u>Data Obat</u>			
Data Obat			
Normal flow "Tambah data obat"		Status	
Aksi	System	Berkas	Gagal
1. Klik menu data obat	2. Menampilkan halaman Data obat berupa data obat. Pada setiap baris data obat terdapat pilihan yaitu edit, view, hapus serta pilihan tombol dan tabel yang digunakan untuk menambahkan data obat baru.	✓	
3. Klik "tambah"	4. Menampilkan form data obat	✓	
5. Mengisi form data obat	6. Klik "tambah"	✓	
	7. Menampilkan kelengkapan data	✓	
	8. Menyimpan data obat baru	✓	
	9. Menampilkan alert "Data berhasil ditambahkan"	✓	
10. Klik "OK"	11. Menampilkan halaman data obat yang telah diperbaharui	✓	
Normal flow "Edit Data Obat"			
12. Klik "edit" pada baris data obat yang ingin di edit.	13. Menampilkan data obat yang akan di edit pada form data obat.	✓	
14. Mengedit data obat pada form data obat.	15. Klik "tambah"	✓	
	16. Menampilkan kelengkapan data	✓	
	17. Mengupdate data obat	✓	
	18. Menampilkan alert "data obat berhasil diperbaharui"	✓	
19. Klik "OK"	20. Menampilkan halaman data obat yang telah diperbaharui.	✓	
Normal flow "View Data Obat"			
21. Klik "view" pada baris data obat yang ingin dilihat	22. Menampilkan detail data obat sesuai yang dipilih	✓	
23. Klik "tutup"	24. Menampilkan halaman data obat	✓	
Normal flow "Hapus Data Obat"			
25. Klik hapus pada baris data obat yang ingin dihapus.	26. Menampilkan alert "anda yakin ingin menghapus data ini?"	✓	
27. Klik "YA"	28. Menhapus data obat	✓	
	29. Mengembalikan ke halaman data obat	✓	
Alternative flow "Kelengkapan data pada tombol data obat"			
7.a Apabila user memasukkan data tidak lengkap	7.b Klik "tambah"	7.c Menampilkan alert "Maaf, data belum lengkap"	✓
7.d Klik "OK"	7.e Mengembalikan ke form setelah data obat	7.f Mengembalikan ke form setelah data obat	✓
Alternative flow "Kelengkapan data pada edit data obat"			
10.a Apabila user memasukkan data tidak lengkap	10.b Klik "tambah"	10.c Menampilkan alert "Maaf, data belum lengkap"	✓
10.d Klik "OK"	10.e Mengembalikan ke form setelah data obat	10.f Mengembalikan ke form setelah data obat	✓
Alternative flow "Batal menambahkan data obat"			
6.a Apabila user batal menambahkan data obat.	6.b Klik "cancel"	6.c Mengembalikan ke halaman data obat	✓
Alternative flow "Batal mengedit data obat"			
13.a Apabila user batal mengedit data obat.			✓

Penguji User			
Nama Penguj: <u>ALBA...ELATIN...</u>			
Level Hak Akses: _____			
Fitur: <u>Data Obat</u>			
15.b Klik "Cancel"	15.c Mengembalikan ke halaman data obat	✓	
Alternative flow "Batal menghapus data obat"			
26.a Apabila user batal menghapus data obat.		✓	
26.b Klik "Batal"	26.c Mengembalikan ke halaman data obat	✓	

BANDARAJAR, 30 MARET 2019

TTD.


 (ALBA...ELATIN...)

B. Data Supplier

Penguujian User			
Nama Penguji : <u>Amel Rizki Nur</u>			
Level Hak Akses : _____			
Fitur : <u>Data Supplier</u>			
Data Supplier			
Normal flow "Tambah data supplier"		Status	
Aksi	System	Berhasil	Gagal
1. Klik menu data supplier	2. Menampilkan halaman Data supplier berupa data supplier. Pada setiap baris data supplier terdapat pilihan yaitu edit, view, hapus serta pilihan tambah data tabel yang digunakan untuk menambahkan data supplier baru.	✓	
3. Klik "Tambah"	4. Menampilkan form data supplier	✓	
5. Mengisi form data supplier		✓	
6. Klik "Simpan"	7. Menertikan kelengkapan data	✓	
	8. Menyimpan data supplier baru	✓	
	9. Menampilkan alert "Data berhasil ditambahkan"	✓	
10. Klik "Ok"	11. Menampilkan halaman data supplier yang telah diperbaharui	✓	
Normal flow "Edit Data Supplier"		Berhasil	Gagal
12. Klik "edit" pada baris data supplier yang ingin di-edit	13. Menampilkan data supplier yang akan di-edit pada form data supplier.	✓	
14. Meng-edit data supplier pada form data supplier		✓	
15. Klik "Simpan"		✓	
	16. Menertikan kelengkapan data	✓	
	17. Meng-upload data supplier	✓	
	18. Menampilkan alert "data supplier berhasil diperbaharui"	✓	
19. Klik "Ok"	20. Menampilkan halaman data supplier yang telah diperbaharui	✓	
Normal flow "View Data Supplier"		Berhasil	Gagal
21. Klik "view" pada baris data supplier yang ingin dilihat	22. Menampilkan detail data supplier sesuai yang dipilih	✓	
23. Klik "Kembali"	24. Menampilkan halaman data supplier	✓	
Normal flow "Hapus Data Supplier"		Berhasil	Gagal
25. Klik hapus pada baris data supplier yang ingin dihapus	26. Menampilkan alert "anda yakin ingin menghapus data ini?"	✓	
27. Klik "Ok"	28. Menghapus data supplier	✓	
	29. Mengembalikan ke halaman data supplier	✓	
Alternative flow "kelengkapan data pada tambah data supplier"		Berhasil	Gagal
7.a Apabila user memasukkan data tidak lengkap		✓	
7.b Klik "Simpan"	7.c Menampilkan alert "Maaf, data belum lengkap"	✓	
7.d Klik "Ok"	7.e Mengembalikan ke form tambah data supplier	✓	
Alternative flow "kelengkapan data pada edit data supplier"		Berhasil	Gagal
16.a Apabila user memasukkan data tidak lengkap		✓	
16.b Klik "Simpan"	16.c Menampilkan alert "Maaf, data belum lengkap"	✓	

Penguujian Fitur Data Supplier Hal. 1

Penguujian Fitur Data Supplier Hal. 2

16.d Klik "Ok"	16.e Mengembalikan ke form data supplier	✓	
Alternative flow "Batal menambah data supplier"		Berhasil	Gagal
6.a Apabila user batal menambah/mengedit data supplier.		✓	
6.b Klik "cancel"	6.c Mengembalikan ke halaman data supplier	✓	
Alternative flow "Batal mengedit data supplier"		Berhasil	Gagal
15.a Apabila user batal menambah/mengedit data supplier.		✓	
15.b Klik "cancel"	15.c Mengembalikan ke halaman data supplier	✓	
Alternative flow "Batal menghapus data supplier"		Berhasil	Gagal
26.a Apabila user batal menghapus data supplier.		✓	
26.b Klik "Batal"	26.c Mengembalikan ke halaman data supplier	✓	

BANDILAN, 20 MARET 2019

TTD.



AGUS ARIATNI

Penguujian Fitur Data Supplier Hal. 3

C. Daftar Obat Habis

Pengujian User


Nama Penguji : AFRI ASIATIN

Level Hak Akses :

Fitur : Daftar Obat Habis

Daftar Obat Habis			
Normal flow "Daftar Obat Habis"		Status	
Actor	System	Berhasil	Gagal
1. Pilih menu daftar obat habis		✓	
	2. Menampilkan halaman Daftar obat habis yang berisi tabel daftar obat yang habis beserta tombol cetak untuk mencetak daftar obat habis (Stok obat kurang dari 50)	✓	
3. Klik "cetak"		✓	
	4. Menampilkan alert, "Apakah anda ingin mencetak?"	✓	
5. Klik "Ok"		✓	
	6. Mencetak daftar obat yang habis	✓	
	7. Menampilkan halaman Daftar obat habis	✓	
Alternative flow "Batal mencetak"		Berhasil	Gagal
4.a Apabila user batal mencetak daftar obat habis		✓	
4.b Klik "Batal"		✓	
	4.c Mengembalikan ke halaman Daftar obat habis	✓	

BANGKALAN, 30 MARET 2019

TTD.

 (...AFRI ASIATIN...)

Pengujian Fitur Daftar Obat Habis

D. Transaksi Pembelian Obat

Pengujian User

Nama Penguji : AERI ACIATIN
 Level Hak Akses :
 Fitur : Transaksi Pembelian Obat

Transaksi Pembelian Obat			
Normalflow "Transaksi Pembelian Obat"		Status	
Actor	System	Berhasil	Gagal
1. Pilih menu Transaksi Pembelian		✓	
	2. Menampilkan halaman transaksi pembelian yang berisi form transaksi pembelian obat. terdapat dua jenis form. Form yang pertama berisi data transaksi pembelian yang berisi tanggal transaksi dan data supplier, sedangkan form yang kedua berisi data obat yang dibeli, dimana user harus menekan 'Add' setiap satu jenis obat diinputkan	✓	
3. Mengisi data obat yang dibeli		✓	
4. Klik "Add"		✓	
	5. Menampilkan tabel yang berisi data obat yang diinputkan dan tombol hapus disetiap baris data	✓	
6. Mengisi data transaksi pembelian		✓	
7. Klik "Submit"		✓	
	8. Memeriksa kelengkapan data	✓	
	9. Menyimpan data transaksi pembelian obat	✓	
	10. Menampilkan alert "Transaksi Berhasil"	✓	
11. Klik "Ok"		✓	
	12. Menampilkan halaman transaksi	✓	

Pengujian Fitur Transaksi Pembelian Obat Hal. 1

Normal flow "Hapus Baris Data"		Berhasil	Gagal
13. Klik "hapus" pada baris data obat yang ingin dihapus.		✓	
	14. Menghapus data obat sesuai baris yang dipilih	✓	
Alternative flow "kelengkapan data"		Berhasil	Gagal
8.a Apabila user memasukkan data tidak lengkap.		✓	
8.b Klik "Submit"		✓	
	8.c Menampilkan alert "Maaf, data belum lengkap"	✓	
8.d Klik "Ok"		✓	
	8.e Mengembalikan ke form transaksi pembelian obat	✓	

BANDARAN, 20 MARET 2019
 TTD.

 (Aeri Aciatin.....)

Pengujian Fitur Transaksi Pembelian Obat Hal. 2

E. Transaksi Penjualan Obat

Pengujian User

Nama Penguji : Desi Harnidoh
 Level Hak Akses :
 Fitur : Transaksi Penjualan Kacamata

Transaksi Penjualan Kacamata			
Normal flow "Transaksi Penjualan Kacamata"		Status	
Actor	System	Berhasil	Gagal
1. Pilih menu Transaksi Penjualan		✓	
	2. Menampilkan halaman transaksi penjualan yang berisi form transaksi penjualan kacamata, terdapat dua jenis form. Form yang pertama berisi data transaksi penjualan yang berisi tanggal transaksi dan data pasien, sedangkan form yang kedua berisi data kacamata yang dibeli, dimana user harus menekan "Add" setiap satu jenis kacamata diinputkan	✓	
3. Mengisi data kacamata yang dibeli		✓	
4. Klik "Add"		✓	
	5. Menampilkan tabel yang berisi data kacamata yang diinputkan dan tombol hapus disetiap baris data	✓	
6. Mengisi data transaksi penjualan		✓	
7. Klik "cetak"		✓	
	8. Memeriksa kelengkapan data	✓	
	9. Menyimpan data transaksi	✓	
	10. Mencetak nota pembayaran	✓	
	11. Menampilkan halaman transaksi penjualan kacamata	✓	
<i>Normal flow "Hapus Baris Data"</i>			
12. Klik "hapus" pada baris data kacamata yang ingin dihapus.		✓	
	13. Menghapus data kacamata sesuai baris yang dipilih	✓	
<i>Alternative flow "Kelengkapan data"</i>			
R.a Apabila user memasukkan data tidak lengkap.		✓	
R.b Klik "cetak"		✓	
	R.c Menampilkan alert "Maaf, data belum lengkap"	✓	
R.d Klik "OK"		✓	
	R.e Mengembalikan ke form transaksi penjualan kacamata	✓	

Pengujian Fitur Transaksi penjualan Obat Hal. 1

Bengkalua, 20 Maret 2020
 TTD,

 (Desi Harnidoh)

Pengujian Fitur Transaksi penjualan Obat Hal. 2

D.3 Blackbox Testing Tahap Increment 3

D.3.1 Bagian Kacamata

A. Data Frame

Pengujian User

Nama Penguj: Desi Hanichah
 Level UAT: _____
 Fitur: Data Frame


Normal flow "Tambah Data Frame"		Status	
Aksi	Hasil	Berhasil	Gagal
1. Klik menu data frame		✓	
2. Menampilkan halaman Data kacamata berupa data frame. Pada setiap baris data frame terdapat pilihan yaitu edit dan hapus serta digunakan untuk menambahkan data kacamata baru.		✓	
3. Klik "Tambah"		✓	
4. Menginput form data frame		✓	
5. Klik "Simpan"		✓	
6. Klik "OK"		✓	
7. Menampilkan kelengkapan data		✓	
8. Menampilkan data frame baru		✓	
9. Menampilkan alert "Data berhasil ditambahkan"		✓	
10. Klik "OK"		✓	
11. Menampilkan halaman data frame yang telah diperbaharui		✓	
Normal flow "Edit Data Frame"		Status	
Aksi	Hasil	Berhasil	Gagal
12. Klik "edit" pada baris data frame yang ingin di edit		✓	
13. Menampilkan data frame yang akan di edit pada form data frame		✓	
14. Meng edit data frame pada form data frame		✓	
15. Klik "Simpan"		✓	
16. Menampilkan kelengkapan data		✓	
17. Meng update data frame		✓	
18. Menampilkan alert "data frame berhasil diperbaharui"		✓	
19. Klik "OK"		✓	
20. Menampilkan halaman data frame		✓	

Pengujian Fitur Data Frame Hal. 1

yang telah diperbaharui		Status	
Normal flow "Hapus Data Frame"	Hasil	Berhasil	Gagal
21. Klik hapus pada baris data frame yang ingin di hapus		✓	
22. Menampilkan alert "anda yakin ingin menghapus data ini?"		✓	
23. Klik "OK"		✓	
24. Menghapus data frame		✓	
25. Mengembalikan ke halaman data frame		✓	
Alternative flow "kelengkapan data pada baris data frame"		Status	
Aksi	Hasil	Berhasil	Gagal
26. Apakah user menambahkan data tidak lengkap		✓	
27. Klik "Simpan"		✓	
28. Menampilkan alert "Maaf, data belum lengkap"		✓	
29. Klik "OK"		✓	
30. Mengembalikan ke form tambah data frame		✓	
Alternative flow "kelengkapan data pada edit data frame"		Status	
Aksi	Hasil	Berhasil	Gagal
31. Apakah user menambahkan data tidak lengkap		✓	
32. Klik "Simpan"		✓	
33. Menampilkan alert "Maaf, data belum lengkap"		✓	
34. Klik "OK"		✓	
35. Mengembalikan ke form edit data frame		✓	
Alternative flow "Batal menambah data frame"		Status	
Aksi	Hasil	Berhasil	Gagal
36. Apakah user batal menambah data frame		✓	
37. Klik "cancel"		✓	
38. Mengembalikan ke halaman data frame		✓	
Alternative flow "Batal mengedit data frame"		Status	
Aksi	Hasil	Berhasil	Gagal
39. Apakah user batal mengedit data frame		✓	
40. Klik "cancel"		✓	
41. Mengembalikan ke halaman data frame		✓	
Alternative flow "Batal menghapus data frame"		Status	
Aksi	Hasil	Berhasil	Gagal
42. Apakah user batal menghapus data		✓	

Pengujian Fitur Data Frame Hal. 2

frame.			
22.bKlik "Batal"		✓	
	22.c Mengembalikan ke halaman data frame	✓	

Bengkalon, 30 Maret 2015.
 TTD.

 (Desi Hanichah...)

Pengujian Fitur Data Frame Hal. 3

B. Transaksi Penjualan Kacamata

Pengujian User

Nama Penguji : *Desni Harwidi*

Level Hak Akses :

Fitur : Transaksi Penjualan Kacamata

Transaksi Penjualan Kacamata			
Normal flow "Transaksi Penjualan Kacamata"		Status	
Actor	System	Berhasil	Gagal
1. Pilih menu Transaksi Penjualan		✓	
	2. Menampilkan halaman transaksi penjualan yang berisi form transaksi penjualan kacamata, terdapat dua jenis form. Form yang pertama berisi data transaksi penjualan yang berisi tanggal transaksi dan data pasien, sedangkan form yang kedua berisi data kacamata yang dibeli, dimana user harus menekan 'Add' setiap satu jenis kacamata diinputkan	✓	
3. Mengisi data kacamata yang dibeli		✓	
4. Klik "Add"		✓	
	5. Menampilkan tabel yang berisi data kacamata yang diinputkan dan tombol hapus disetiap baris data	✓	
6. Mengisi data transaksi penjualan		✓	
7. Klik "cetak"		✓	
	8. Memeriksa kelengkapan data	✓	
	9. Menyimpan data transaksi	✓	
	10. Mencetak nota pembayaran	✓	
	11. Menampilkan halaman transaksi penjualan kacamata	✓	
Normal flow "Hapus Baris Data"			
	12. Klik "hapus" pada baris data kacamata yang ingin dihapus.	✓	
	13. Menghapus data kacamata sesuai baris yang dipilih	✓	
Alternative flow "Kelengkapan data"			
8.a Apabila user memasukkan data tidak lengkap.		✓	
8.b Klik "cetak"		✓	
	8.c Menampilkan alert "Maaf, data belum lengkap"	✓	
8.d Klik "Ok"		✓	
	8.e Mengembalikan ke form transaksi penjualan kacamata	✓	

Pengujian Fitur Transaksi penjualan Kacamata Hal. 1

12. Klik "hapus" pada baris data kacamata yang ingin dihapus.		✓	
	13. Menghapus data kacamata sesuai baris yang dipilih	✓	
Alternative flow "Kelengkapan data"			
8.a Apabila user memasukkan data tidak lengkap.		✓	
8.b Klik "cetak"		✓	
	8.c Menampilkan alert "Maaf, data belum lengkap"	✓	
8.d Klik "Ok"		✓	
	8.e Mengembalikan ke form transaksi penjualan kacamata	✓	

Bengkalis, ... Maret 2015
TID.
(Desni Harwidi)

Pengujian Fitur Transaksi penjualan Kacamata Hal. 2

C. Transaksi Pelunasan Kacamata

Pengujian User

Nama Penguji : Sigit P.


Level Hak Akses : kacamata

Fitur : Transaksi Pelunasan Kacamata

Transaksi Pelunasan			
Normal flow "Transaksi Pelunasan Kacamata"		Status	
Actor	System	Berhasil	Gagal
1. Pilih menu Transaksi Pelunasan			
	2. Menampilkan halaman transaksi pelunasan berupa tabel (berisi data transaksi penjualan kacamata dengan status BELUM LUNAS)	✓	
3. Mengisi id pasien pada kolom pencarian			
	4. Menampilkan data yang dicari	✓	
5. Klik 'Lunas'			
	6. Mengupdate data transaksi penjualan	✓	
	7. Menampilkan halaman transaksi pelunasan	✓	

Lengkap 30 Maret '15.

TTD.


(Sigit P.)

Pengujian Fitur Transaksi Pelunasan Kacamata

D.4 Blackbox Testing Tahap Increment 4

D.4.1 Dokter

A. Input Data Pemeriksaan

Pengujian User

Nama Penguji : *Jamaluudin*

Level Hak Akses : *Dokter*

Fitur : Input Data Pemeriksaan

<i>Input Data Pemeriksaan</i>			
Normal flow " <i>Input Data Pemeriksaan</i> "		Status	
<i>Actor</i>	<i>System</i>	Berhasil	Gagal
1. Pilih menu <i>input data pemeriksaan</i>		✓	
	2. Menampilkan halaman <i>input data pemeriksaan</i>	✓	
3. Mengisi <i>data pemeriksaan</i>		✓	
4. Klik " <i>Submit</i> "		✓	
	5. Memeriksa kelengkapan <i>data</i>	✓	
	6. Menyimpan <i>data pemeriksaan</i>	✓	
	7. Mengembalikan pada halaman <i>input data pemeriksaan</i>	✓	
<i>Alternative flow "kelengkapan data"</i>			
5.a Apabila <i>user</i> memasukkan <i>data</i> tidak lengkap.		✓	
5.b Klik " <i>Submit</i> "		✓	
	5.c Menampilkan <i>alert</i> "Maaf, <i>data</i> belum lengkap"	✓	
5.d Klik "Ok"		✓	
	5.e Mengembalikan ke halaman <i>input data pemeriksaan</i>	✓	

Bangkalan 20 Maret 2020

TTD.

[Signature]

(*Jamaluudin*)

Pengujian Fitur Input Data Pemeriksaan

B. Lihat Data Rekam Medis

Pengujian User

Nama Penguji : Jamalusidin


Level Hak Akses : Dokter

Fitur : Lihat Data Rekam Medis

Data Pasien			
Normal flow "Lihat Data Rekam Medis"		Status	
Actor	System	Berhasil	Gagal
1. Pilih menu lihat data rekam medis		✓	
	2. Menampilkan halaman lihat data rekam medis	✓	
3. Pilih id pasien		✓	
	4. Menampilkan tabel data rekam medis pasien sesuai dengan id pasien yang dipilih	✓	
Alternative flow "user tidak memilih id pasien"		Berhasil	Gagal
3.a Apabila user tidak memilih id pasien	3.b	✓	
3.c Pilih id pasien	3.d	✓	
	3.e Menampilkan alert, "Pilih ID Pasien!"	✓	
3.f Klik "Ok"		✓	
	3.g Mengembalikan ke halaman lihat data rekam medis	✓	
Alternative flow "ID pasien tidak ditemukan"		Berhasil	Gagal
3.a Apabila ID pasien yang di masukkan oleh user tidak ditemukan datanya		✓	
3.b Pilih id pasien		✓	
	3.c Menampilkan alert, "Data tidak ditemukan!"	✓	
3.d Klik "Ok"		✓	
	3.e Menampilkan halaman lihat data rekam medis	✓	

Bengkulu, 30 Mei 2019

TTD.



(Jamalusidin)

Pengujian Fitur Input Data Pemeriksaan

D.4.2 Perawat

A. Pendaftaran Operasi


Pengujian User

Nama Penguji : Hanna M.
 Level Hak Akses :
 Fitur : Pendaftaran Operasi

Pendaftaran Operasi			
Normal flow "Pendaftaran Operasi"		Status	
Actor	System	Berhasil	Gagal
1. Pilih menu pendaftaran operasi		✓	
	2. Menampilkan halaman pendaftaran operasi berisi form pendaftaran operasi pasien	✓	
3. Mengisi form pendaftaran operasi		✓	
4. Klik "Submit"		✓	
	5. Menyimpan data pendaftaran operasi	✓	
	6. Menampilkan halaman pendaftaran operasi	✓	
Normal flow "Cek Ketersediaan Tanggal"			
7. Klik "Cek kuota"			
	8. Jika pada tanggal yang dipilih kuota telah terisi 4 pasien maka, Menampilkan <i>alert</i> "Kuota tidak mencukupi.(Jumlah kuota terpenuhi : 4)"	✓	
	9. Jika pada tanggal yang dipilih kuota kurang dari 4 pasien maka, Menampilkan <i>alert</i> "Kuota mencukupi.(Kuota saat ini : jumlah pasien terdaftar)"	✓	
10. Klik "Ok"		✓	
	11. Mengembalikan ke halaman pendaftaran operasi	✓	
Alternative flow "Memeriksa kelengkapan data"			

Pengujian Fitur Pendaftaran Operan Hal 1

4.a	Apabila data yang dimasukkan tidak lengkap	✓	
4.b	Klik "Submit"	✓	
	4.c Menampilkan <i>alert</i> "Maaf, data tidak lengkap"	✓	
4.d	Klik "Ok"	✓	
	4.e Mengembalikan ke halaman pendaftaran operasi	✓	

Jember, 21 Maret 2021
 TTD.

 (Hanna M.)

Pengujian Fitur Pendaftaran Operan Hal 2

B. Lihat Jadwal Operasi

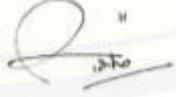
Pengujian User

Nama Penguji : Richo Resa
 Level Hak Akses : Perawat
 Fitur : Lihat Jadwal Operasi

Lihat Jadwal Operasi			
Normal flow "Lihat Jadwal Operasi"		Status	
Actor	System	Berhasil	Gagal
1. Pilih menu lihat jadwal operasi		✓	
	2. Menampilkan halaman lihat jadwal operasi	✓	
3. Pilih tanggal, bulan dan tahun		✓	
4. Klik "Ok"		✓	
	5. Menampilkan tabel berisi data operasi sesuai tanggal yang dipilih	✓	
Alternative flow "user tidak mengisi pilihan tanggal"		Berhasil	Gagal
4.a Apabila user tidak memilih tanggal		✓	
4.b Klik "Ok"		✓	
	4.c Menampilkan alert "Pilih Tanggal!"	✓	
4.d Klik "Ok"		✓	
	4.e Mengembalikan ke halaman lihat jadwal operasi	✓	
Alternative flow "Tanggal yang dicari tidak di ada"		Berhasil	Gagal
4.a Apabila tanggal yang di masukkan oleh user tidak ditemukan datanya		✓	
4.b Klik "Ok"		✓	
	4.c Menampilkan alert , " Data tidak ditemukan!"	✓	

Pengujian Fitur Lihat Jadwal Operasi Hal 1

4.d Klik "Ok"		✓	
	4.e Menampilkan halaman lihat jadwal operasi	✓	

Bongkolan 30 Maret 2018
 TTD.

 (Richo Resa.....)


Pengujian Fitur Lihat Jadwal Operasi Hal 2



LAMPIRAN E. WHITEBOX TESTING

E.1. Whitebox Testing Tahap Increment Dua

a. Whitebox Testing Data User

A. Tambah Data User	
✓ Controller Tambah Data User [cont.php/insert_user]	
<pre> 36 public function insert_user(){ 37 1 \$input = array(38 "nama_ac" => \$this->input->post('nama_ac'), 39 "username" => \$this->input->post('username'), 40 "password" => \$this->input->post('password'), 41 "level" => \$this->input->post('level'), 42 "alamat" => \$this->input->post('alamat'), 43 "tlp" => \$this->input->post('tlp') 44); 45 2 \$logic = \$this->m_user->insert_user(\$input); 46 47 3 redirect(base_url() . "index.php/cont/data_user?status=\$logic"); 48 } </pre>	
<pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) </pre>	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 2 - 3 + 2$ $V(G) = 1$
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2 - 3</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2 - 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	
✓ Model Tambah Data User [m_user.php/insert_user]	

<pre> 16 function insert_user(\$input) { 17 1\$result = \$this->db->insert('account', \$input); 18 return \$result; 2 19 } 20 </pre>	
 <pre> graph TD 1((1)) --> 2((2)) </pre>	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 1 - 2 + 2$ $V(G) = 1$
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	
<p>B. Edit Data User</p>	
<p>Controller Edit Data User [cont.php/edit_user]</p>	
<pre> 55 function edit_user() { 56 1 \$input = array(57 "id_ac" => \$this->input->post('id_ac'), 58 "nama_ac" => \$this->input->post('nama_ac'), 59 "username" => \$this->input->post('username'), 60 "password" => \$this->input->post('password'), 61 "level" => \$this->input->post('level'), 62 "alamat" => \$this->input->post('alamat'), 63 "tlp" => \$this->input->post('tlp') 64); 65 2 \$logic = \$this->m_user->update_user(\$input); 66 3 redirect(base_url() . "index.php/cont/data_user?status=\$logic"); </pre>	

	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 2 - 3 + 2$ $V(G) = 1$
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2 - 3</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2 - 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	
<p>Model Edit Data User [m_user.php/update_user]</p>	
<pre> 28 function update_user(\$input) { 29 1 \$this->db->where('id_ac', \$input['id_ac']); 30 2 return \$this->db->update('account', \$input); 31 } 32 </pre>	
	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 1 - 2 + 2$ $V(G) = 1$
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p>	

Jalur 1 = 1 - 2

2. *Test Case*

Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

C. Hapus Data User

Controller Hapus Data User [cont.php/hapus_user]

```

67 | function hapus_user($id_ac) {
68 |     1 $this->m_user->hapus_user($id_ac);
69 |     2 redirect(base_url() . "index.php/cont/data_user?status=$id_ac");
70 | }
    
```



$V(G) = e - n + 2$ (e = edge, n = node)

$V(G) = 1 - 2 + 2$

$V(G) = 1$

1. *Basis Set*

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2

2. *Test Case*

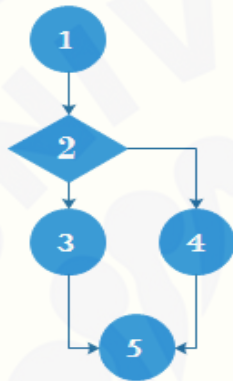
Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

Model Hapus Data User [m_user.php/hapus_user]


```

33 function hapus_user($id_ac) {
34     1 $query = $this->db->query(
35         "DELETE FROM account WHERE id_ac = '" . $id_ac . "'");
36
37     2 if ($query) {
38         3 RETURN TRUE;
39     } else {
40         4 RETURN FALSE;
41     }
42     5 }

```



$$V(G) = e - n + 2 \quad (e = \text{edge}, n = \text{node})$$

$$V(G) = 5 - 5 + 2$$

$$V(G) = 2$$

1. Basis Set

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2 - 3 - 5

Jalur 2 = 1 - 2 - 4 - 5

2. Test Case

Basis set dicoba dan basis set sukses yang dihasilkan 1 - 2 - 3 - 5, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

b. Whitebox Testing Buat Akun

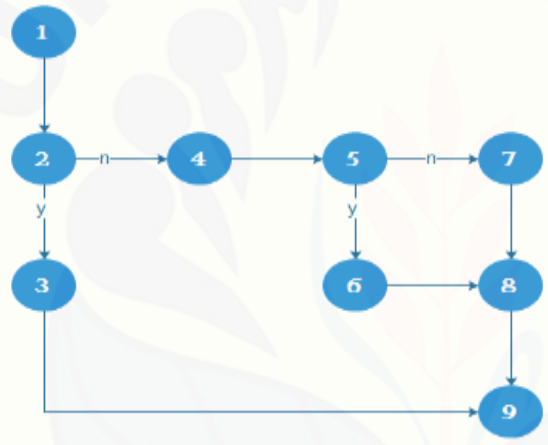
A. Buat Akun

✓ Controller Buat Akun [cont_registrasio.php/insert_akun]

```

36 public function insert_akun(){
37     $input = array(
38         1 "nama_ac" => $this->input->post('nama_ac'),
39         "email" => $this->input->post('email'),
40         "username" => $this->input->post('username'),
41         "password" => $this->input->post('password'),
42         "level" => 7,
43         "alamat" => $this->input->post('alamat'),
44         "tip" => $this->input->post('tip'),
45         "status" => "No" );
46     2if($this->m_user->cek_username($input['username']) == TRUE){
47         $this->session->set_flashdata("message","Username Telah Digunakan!");
48         3redirect("cont_registrasio/create_account");
49     }
50     else{
51         4$logica = $this->m_user->register($input);
52     }
53     5if($logica){
54         $this->session->set_flashdata("message",$input['username'].". Silahkan aktivasi melalui email anda");
55         6redirect("cont_registrasio/login");
56     }
57     else{
58         7$this->session->set_flashdata("message","Registrasi gagal");
59         redirect("cont_registrasio/create_account");
60     }
61     8
62     9
63 }

```



$V(G) = e - n + 2$ (e = edge, n = node)
 $V(G) = 10 - 9 + 2$
 $V(G) = 3$

1. **Basis Set**
Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

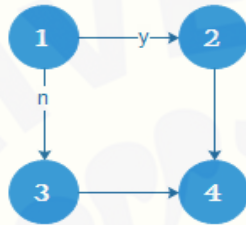
Jalur 1 = 1 - 2 - 3 - 9
 Jalur 2 = 1 - 2 - 4 - 5 - 6 - 8 - 9
 Jalur 3 = 1 - 2 - 4 - 5 - 7 - 8 - 9

2. **Test Case**
Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2 - 3 - 9, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

✓ Model Buat Akun [m_user.php/register]

```

31 function register($input){
32     // email
33     if($this->send_email($input)){
34         // insert database
35         return $this->insert_user($input);
36     }
37     else{
38         3 return false;
39     }
40 }
    
```



$$V(G) = e - n + 2 \text{ (e = edge, n = node)}$$

$$V(G) = 4 - 4 + 2$$

$$V(G) = 2$$

1. *Basis Set*

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

$$\text{Jalur 1} = 1 - 2 - 4$$

$$\text{Jalur 2} = 1 - 2 - 3$$

2. *Test Case*

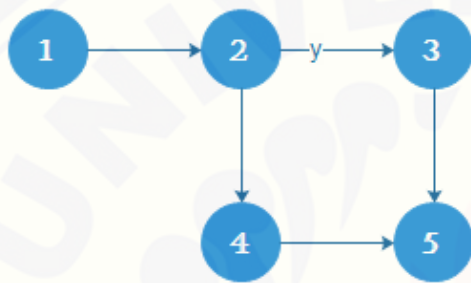
Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2 - 4, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

c. *Whitebox Testing* Aktivasi Akun Lewat Email

B. Aktivasi Akun Lewat Email
✓ Controller Aktivasi Akun Lewat Email [cont_registrasio.php/confirm]

```

62 public function confirm($username){
63     1$result = $this->m_user->confirm($username);
64     2if($result){
65         3$this->session->set_flashdata("message",$username." telah aktif");
66     }
67     4else{
68         $this->session->set_flashdata("message",$username." gagal di aktivasi");
69     }
70     5redirect("cont_registrasio/login");
71 }
    
```



$V(G) = e - n + 2$ (e = edge, n = node)
 $V(G) = 5 - 5 + 2$
 $V(G) = 2$

3. Basis Set

Basis set menghasilkan dua jalur independent dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2 - 3 - 5

Jalur 2 = 1 - 2 - 4 - 5

4. Test Case

Basis set dicoba dan basis set sukses yang dihasilkan 1 - 2 - 3 - 5, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

✓ Model Aktivasi Akun Lewat Email [m_user.php/confirm]

```

91 public function confirm($username){
92     1$this->db->where('username', $username);
93     2return $this->db->update("account",array("status"=>"Yes"));
94 }
    
```



$V(G) = e - n + 2$ (e = edge, n = node)
 $V(G) = 1 - 2 + 2$

	V(G) = 1
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 – 2</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 – 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	

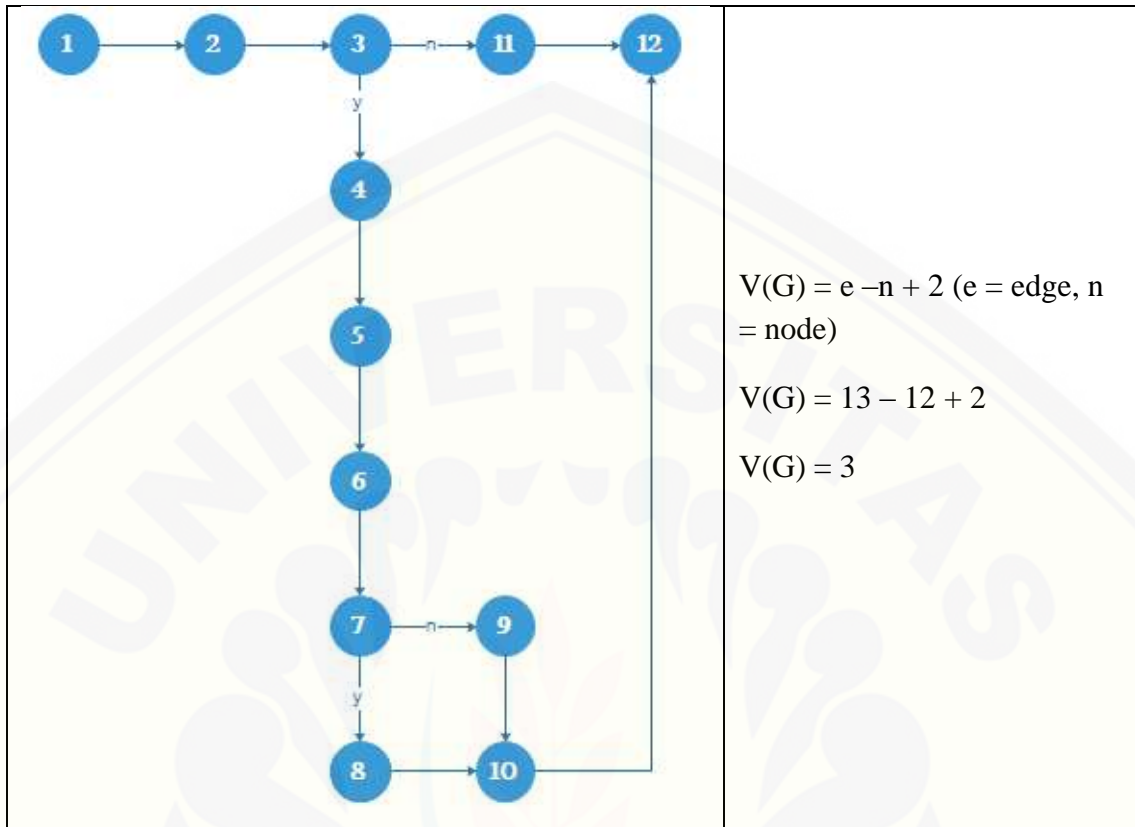
d. *Whitebox Testing* Registrasi Pasien Baru

A. Registrasi Pasien Baru

✓ Controller Registrasi Pasien Baru [cont_registrasio.php/registrasi_baru]

```

function registrasi_baru(){
    $input_pasien = array("id_pasien" => $this->input->post('id_pasien'),
    1  "tgl_masuk" => date('Y-m-d', strtotime($this->input->post('tgl_masuk'))),
    "nama_pasien" => $this->input->post('nama_pasien'),
    "tgl_lahir" => date('Y-m-d', strtotime($this->input->post('tgl_lahir'))),
    "kota_lahir" => $this->input->post('kota_lahir'),
    "jenis_kelamin" => $this->input->post('jk_pasien'),
    "alamat" => $this->input->post('alamat_pasien'),
    "kecamatan" => $this->input->post('kecamatan'),
    "kabupaten" => $this->input->post('kabupaten'),
    "no_tlp" => $this->input->post('no_tlp'),
    "pekerjaan" => $this->input->post('pekerjaan'),
    "nama_wali" => $this->input->post('nama_wali'),
    "no_tlp_wali" => $this->input->post('no_tlp_wali'));
    $insert_pasien = $this->m_pasien->insert_pasien($input_pasien);
    2  if($insert_pasien){
    3  $input_reg = array(
        "no_registrasi" => $this->input->post('no_registrasi'),
        "tgl_registrasi" => date('Y-m-d', strtotime($this->input->post('tgl_registrasi'))),
        "id_pasien" => $this->input->post('id_pasien') );
    4  $keluhan = $this->input->post('keluhan');
    $reg = $this->m_registrasi->prediksi($input_reg, $keluhan);
    5  if($reg){
    6  $this->session->set_flashdata("message",$input_pasien['nama_pasien'].", Telah Terdaftar");
        redirect(base_url() . "index.php/cont_registrasio/lihat_waktu?status=sukses");
    7  }else{
        $this->session->set_flashdata("message","Pendaftaran Gagal");
        8  redirect(base_url() . "index.php/cont_registrasio/reg_pasien_baru?status=gagal");
    }
    9  $this->session->set_flashdata("message",$input_pasien['nama_pasien'].", Telah Terdaftar");
        10  redirect(base_url() . "index.php/cont_registrasio/lihat_waktu?status=sukses");
    }else{
        $this->session->set_flashdata("message","Pendaftaran Gagal");
        11  redirect(base_url() . "index.php/cont_registrasio/reg_pasien_baru?status=gagal");
    }
}
    
```

1. Basis Set

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2 - 3 - 11 - 12

Jalur 2 = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 12

Jalur 3 = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 9 - 10 - 12

2. Test Case

Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2 - 3 - 11 - 12, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

e. Whitebox Testing Data Pasien

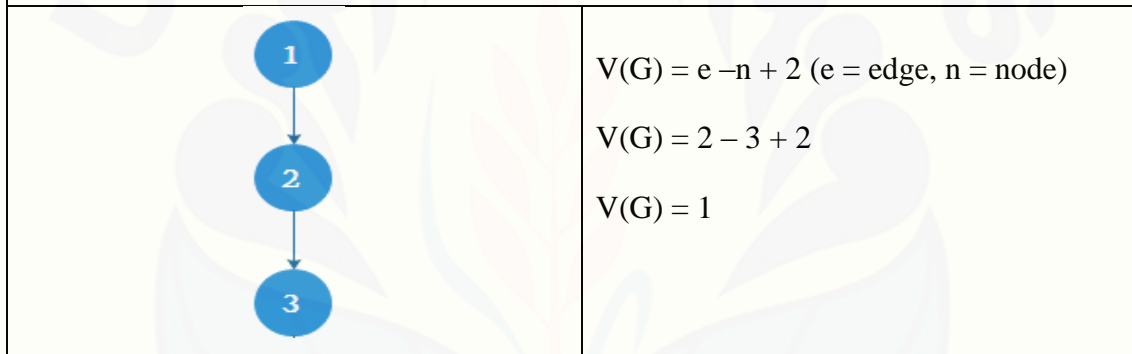
A. Tambah Data Pasien

✓ Controller Tambah Data Pasien [cont.php/insert_pasien]

```

function insert_pasien() {
86     $input = array(
87         "id_pasien" => $this->input->post('id_pasien'),
88         "tgl_masuk" => date('Y-m-d', strtotime($this->input->post('tgl_masuk'))),
89         "nama_pasien" => $this->input->post('nama_pasien'),
90         "tgl_lahir" => date('Y-m-d', strtotime($this->input->post('tgl_lahir'))),
91         "kota_lahir" => $this->input->post('kota_lahir'),
92         "jenis_kelamin" => $this->input->post('jk_pasien'),
93         "alamat" => $this->input->post('alamat_pasien'),
94         "kecamatan" => $this->input->post('kecamatan'),
95         "kabupaten" => $this->input->post('kabupaten'),
96         "no_tlp" => $this->input->post('no_tlp'),
97         "pekerjaan" => $this->input->post('pekerjaan'),
98         "nama_wali" => $this->input->post('nama_wali'),
99         "no_tlp_wali" => $this->input->post('no_tlp_wali')
100     );
101     $logic = $this->m_pasien->insert_pasien($input);
102     redirect(base_url() . "index.php/cont/data_pasien?status=$logic");3
103 }
104

```



$V(G) = e - n + 2$ (e = edge, n = node)

$V(G) = 2 - 3 + 2$

$V(G) = 1$

3. *Basis Set*
Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2 - 3


4. *Test Case*
Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2 - 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.



✓ Model Tambah Data Pasien [m_pasien.php/insert_pasien]

```

13     function insert_pasien($input) {
14         1 $result = $this->db->insert('tb_pasien', $input);
15         return $result;2
16     }

```

	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 1 - 2 + 2$ $V(G) = 1$
<p>3. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2</p> <p>4. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	
<p>B. Edit Data Pasien</p>	
<p>Controller Edit Data Pasien [cont.php/edit_pasien]</p>	
<pre> function edit_pasien() { 119 120 \$input = array("id_pasien" => \$this->input->post('id_pasien'), 121 1 "tgl_masuk" => \$this->input->post('tgl_masuk'), 122 "nama_pasien" => \$this->input->post('nama_pasien'), 123 "tgl_lahir" => \$this->input->post('tgl_lahir'), 124 "kota_lahir" => \$this->input->post('kota_lahir'), 125 "jenis_kelamin" => \$this->input->post('jk_pasien'), 126 "alamat" => \$this->input->post('alamat_pasien'), 127 "kecamatan" => \$this->input->post('kecamatan'), 128 "kabupaten" => \$this->input->post('kabupaten'), 129 "no_tlp" => \$this->input->post('no_tlp'), 130 "pekerjaan" => \$this->input->post('pekerjaan'), 131 "nama_wali" => \$this->input->post('nama_wali'), 132 2 "no_tlp_wali" => \$this->input->post('no_tlp_wali')); 133 \$logic = \$this->m_pasien->update_pasien(\$input); 134 redirect(base_url() . "index.php/cont/data_pasien?status=\$logic" 3); 135 } </pre>	

	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 3 - 2 + 2$ $V(G) = 3$
<p>3. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu: Jalur 1 = 1 - 2 - 3</p> <p>4. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2 - 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	
<p>Model Edit Data Pasien [m_pasien.php/update_pasien]</p>	
<pre> 45 function update_pasien(\$input) { 46 \$this->db->where('id_pasien', \$input['id_pasien']); 47 return \$this->db->update('tb_pasien', \$input); 48 }</pre>	
	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 2 - 1 + 2$ $V(G) = 3$
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu: Jalur 1 = 1 - 2</p>	

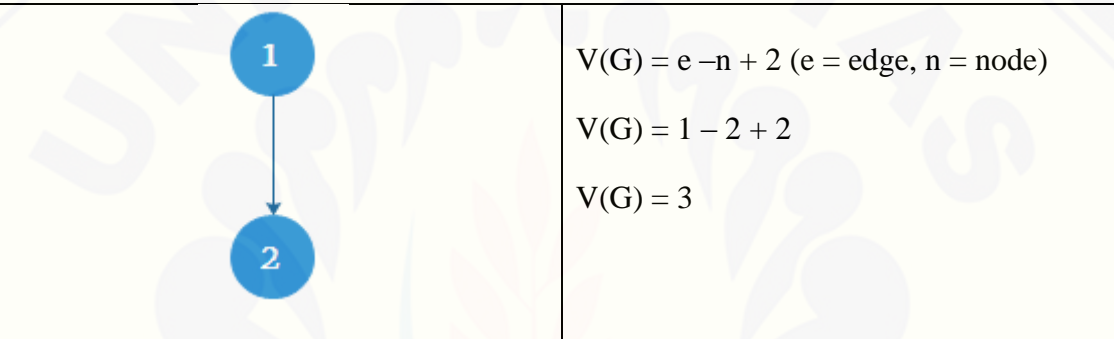
2. *Test Case*
Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

C. View Data Pasien

Controller View Data Pasien [cont.php/detail_pasien]

```

137 function detail_pasien($id_pasien) {
138     1 $data = $this->m_pasien->get_pasien_id($id_pasien)->row_array();
139     $this->load->view('include/datapasien/detail_pasien', $data); 2
140 }
    
```



1. *Basis Set*
Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:


Jalur 1 = 1 - 2

2. *Test Case*
Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

Model View Data Pasien [m_pasien.php/get_pasien_id]

```



38 function get_pasien_id($id_pasien) {
39     1 $query = 'select * from tb_pasien
40         where id_pasien = ' . $id_pasien;
41     $data = $this->db->query($query); 2
42     3 return $data;
43 }
    
```


	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 2 - 3 + 2$ $V(G) = 1$
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 – 2 – 3</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 – 2 – 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	

E.2. Whitebox Testing Tahap Increment Dua

a. Whitebox Testing Data Obat

1. Tambah Data Obat	
✓ Controller Tambah Data Obat [cont.php/insert_obat]	
<pre> 157 158 159 160 161 162 163 164 165 166 167 168 169 </pre>	<pre> function insert_obat() { \$input = array(1 "kode_obat" => \$this->input->post('kode_obat'), "nama_obat" => \$this->input->post('nama_obat'), "id_supplier" => \$this->input->post('id_supplier'), "harga_jual" => \$this->input->post('harga_jual'), "harga_beli" => \$this->input->post('harga_beli'), "stok" => \$this->input->post('stok')); 2 \$logic = \$this->m_obat->insert_obat(\$input); 3 redirect(base_url() . "index.php/cont/data_obat?status=\$logic"); } </pre>

	$V(G) = e - n + 2$ (e = edge, n = node) $V(G) = 2 - 3 + 2$ $V(G) = 1$
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2 - 3</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2 - 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	
<p>✓ Model Tambah Data Obat [m_obat.php/insert_obat]</p>	
<pre> 14 function insert_obat(\$input){ 15 16 \$result = \$this->db->insert('tb_obat', \$input); 17 18 return \$result; 19 }</pre>	
	$V(G) = e - n + 2$ (e = edge, n = node) $V(G) = 1 - 2 + 2$ $V(G) = 1$
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2, dapat dilihat bahwa</p>	

simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

2. Edit Data Obat

Controller Edit Data Obat [cont.php/edit_obat]

```

171 function edit_obat() {
172
173     $input = array(
174         1 "kode_obat" => $this->input->post('kode_obat'),
175         "nama_obat" => $this->input->post('nama_obat'),
176         "id_supplier" => $this->input->post('id_supplier'),
177         "harga_jual" => $this->input->post('harga_jual'),
178         "harga_beli" => $this->input->post('harga_beli'),
179         "stok" => $this->input->post('stok')
180
181     2 $logic = $this->m_obat->update_obat($input);
182     3 redirect(base_url() . "index.php/cont/data_obat?status=$logic");
183 }

```



$$V(G) = e - n + 2 \quad (e = \text{edge}, n = \text{node})$$

$$V(G) = 2 - 3 + 2$$

$$V(G) = 1$$

1. Basis Set



Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2 - 3

2. Test Case

Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2 - 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

Model Edit Data Obat [m_obat.php/update_obat]

<pre> 34 function update_obat(\$input) { 35 \$this->db->where('kode_obat', \$input['kode_obat']); 36 return \$this->db->update('tb_obat', \$input); 37 }</pre>	
	<p>$V(G) = e - n + 2$ (e = edge, n = node)</p> <p>$V(G) = 1 - 2 + 2$</p> <p>$V(G) = 1$</p>
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	
<p>3. Hapus Data Obat</p>	
<p>Controller Hapus Data Obat[cont.php/hapus_obat]</p>	
<pre> 189 function hapus_obat(\$kode_obat) { 190 \$this->m_obat->hapus_obat(\$kode_obat); 191 redirect(base_url() . "index.php/cont/data_obat?status=\$kode_obat"); 192 }</pre>	
	<p>$V(G) = e - n + 2$ (e = edge, n = node)</p> <p>$V(G) = 1 - 2 + 2$</p> <p>$V(G) = 1$</p>
<p>1. <i>Basis Set</i></p>	

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2

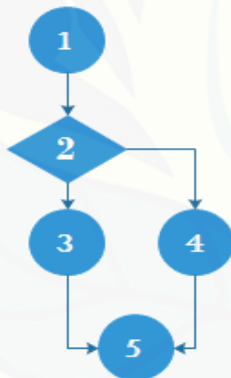
2. *Test Case*

Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

Model Hapus Data Obat[m_obat.php/hapus_obat]

```

38 function hapus_obat($kode_obat) {
39     $query = $this->db->query("DELETE FROM tb_obat WHERE kode_obat = '".$kode_obat."'");
40
41     if($query) {
42         RETURN TRUE;
43     } else {
44         RETURN FALSE;
45     }
46 }
    
```



$V(G) = e - n + 2$ (e = edge, n = node)

$V(G) = 5 - 5 + 2$

$V(G) = 2$

1. *Basis Set*

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2 - 3 - 5

Jalur 2 = 1 - 2 - 4 - 5

2. *Test Case*

Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2 - 3 - 5, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem

telah memenuhi syarat kelayakan.

4. View Data Obat

Controller View Data Obat[cont.php/detail_obat]

```

185 function detail_obat($kode_obat) {
186     $data = $this->m_obat->get_obat_id($kode_obat)->row_array();
187     $this->load->view('include/data_obat/detail', $data);
188 }

```



$V(G) = e - n + 2$ (e = edge, n = node)

$V(G) = 1 - 2 + 2$

$V(G) = 1$

1. Basis Set

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2

2. Test Case


Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

Model View Data Obat[m_obat.php/get_obat_id]

```



20 function get_obat_id($kode_obat) {
21     $query = "select * from tb_obat
22             where kode_obat = '$kode_obat'";
23     $result = $this->db->query($query);
24     return $result;
25 }

```

	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 2 - 3 + 2$ $V(G) = 1$
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 – 2 – 3</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 – 2 – 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	

b. *Whitebox Testing* Data Supplier

A. Tambah Data Supplier	
✓ Controller Tambah Data Supplier [cont.php/insert_Supplier]	
<pre> 206 207 208 209 210 211 212 213 214 215 216 217 218 </pre>	<pre> function insert_supplier() { \$input = array(1 'id_supplier' => \$this->input->post('id_supplier'), 'nama_supplier' => \$this->input->post('nama_supplier'), 'alamat_supplier' => \$this->input->post('alamat_supplier'), 'tlp_supplier' => \$this->input->post('tlp_supplier'), 'nama_sales' => \$this->input->post('nama_sales'), 'tlp_sales' => \$this->input->post('tlp_sales')); \$insert = \$this->m_obat->insert_supplier(\$input); redirect(base_url() . "index.php/cont/data_supplier?status=\$insert"); 3 } </pre>

	<p>$V(G) = e - n + 2$ (e = edge, n = node)</p> <p>$V(G) = 2 - 3 + 2$</p> <p>$V(G) = 1$</p>
<p>3. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2 - 3</p> <p>4. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2 - 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	
<p>✓ Model Tambah Data Supplier [m_obat.php/insert_supplier]</p>	
<pre> 264 function insert_supplier(\$input){ 265 \$query = \$this->db->insert('tb_supplier', \$input); 266 267 return \$query; 268 }</pre>	
	<p>$V(G) = e - n + 2$ (e = edge, n = node)</p> <p>$V(G) = 1 - 2 + 2$</p> <p>$V(G) = 1$</p>
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2</p>	

2. *Test Case*
Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

B. Edit Data Supplier

✓ Controller Edit Data Supplier [cont.php/edit_supplier]

```

227 function edit_supplier() {
228
229     $input = array(
230         "id_supplier" => $this->input->post('id_supplier'),
231         "nama_supplier" => $this->input->post('nama_supplier'),
232         "alamat_supplier" => $this->input->post('alamat_supplier'),
233         "tlp_supplier" => $this->input->post('tlp_supplier'),
234         "nama_sales" => $this->input->post('nama_sales'),
235         "tlp_sales" => $this->input->post('tlp_sales')
236     );
237     $logic = $this->model->update_supplier($input);
238     redirect(base_url() . "index.php/cont/data_obat?status=$logic");
239 }
    
```



$V(G) = e - n + 2$ (e = edge, n = node)

$V(G) = 2 - 3 + 2$

$V(G) = 1$

1. *Basis Set*

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2 - 3

2. *Test Case*

Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2 - 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

<p>✓ Model Edit Data Supplier [m_obat.php/update_supplier]</p> <pre> 275 function update_supplier(\$input) { 276 \$this->db->where('id_supplier', \$input['id_supplier']) 277 return \$this->db->update('tb_supplier', \$input); 278 }</pre>	
<pre> graph TD 1((1)) --> 2((2))</pre>	<p>$V(G) = e - n + 2$ (e = edge, n = node)</p> <p>$V(G) = 1 - 2 + 2$</p> <p>$V(G) = 1$</p>
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	
<p>C. Hapus Data Supplier</p> <p>Controller Hapus Data Supplier[cont.php/hapus_Supplier]</p> <pre> 245 function hapus_supplier(\$id_supplier){ 246 \$this->m_obat->hapus_supplier(\$id_supplier); 247 redirect(base_url() . "index.php/cont/data_supplier?status=\$id_supplier"); 248 }</pre>	
<pre> graph TD 1((1)) --> 2((2))</pre>	<p>$V(G) = e - n + 2$ (e = edge, n = node)</p> <p>$V(G) = 1 - 2 + 2$</p> <p>$V(G) = 1$</p>
<p>1. <i>Basis Set</i></p>	

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2

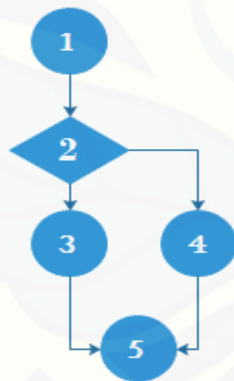
2. *Test Case*

Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

Model Hapus Data Supplier[m_obat.php/hapus_Supplier]

```

279 function hapus_supplier($id_supplier) {
280     $query = $this->db->query("DELETE FROM tb_supplier "
281         . "WHERE id_supplier = '". $id_supplier. "'");
282
283     if($query) {
284         RETURN TRUE; 3
285     } else {
286         RETURN FALSE; 4
287     }
288 }
    
```



$V(G) = e - n + 2$ (e = edge, n = node)

$V(G) = 5 - 5 + 2$

$V(G) = 2$

1. *Basis Set*

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2 - 3 - 5

Jalur 2 = 1 - 2 - 4 - 5

2. *Test Case*

Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2 - 3 - 5, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

D. View Data Supplier

Controller View Data Supplier[cont.php/detail_Supplier]

```

241 function detail_supplier($id_supplier) {
242     $data = $this->m_obat->get_supplier_id($id_supplier)->row_array();
243     $this->load->view('include/data_supplier/detail_supplier', $data);
244 }
  
```



$V(G) = e - n + 2$ (e = edge, n = node)

$V(G) = 1 - 2 + 2$

$V(G) = 1$

1. Basis Set

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2

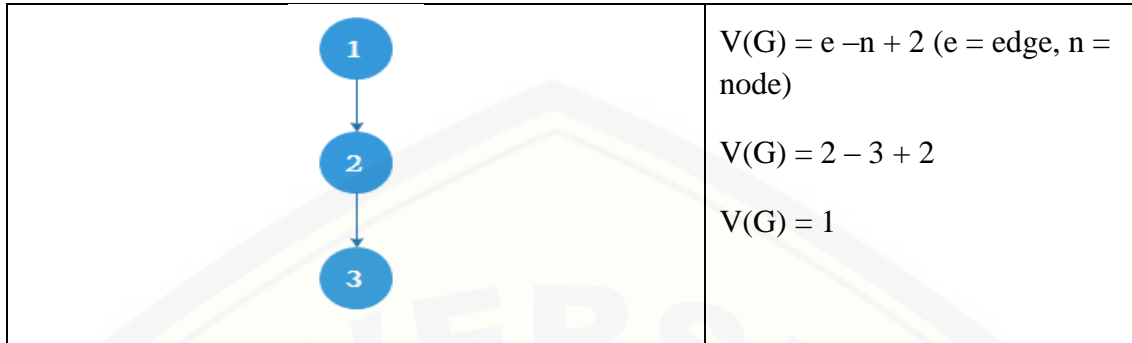
2. Test Case

Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

Model View Data Supplier[m_obat.php/get_supplier_id]

```

269 function get_supplier_id($id_supplier) {
270     $query = 'select * from tb_supplier
271             where id_supplier = ' . $id_supplier;
272     $result = $this->db->query($query);
273     return $result;
274 }
  
```



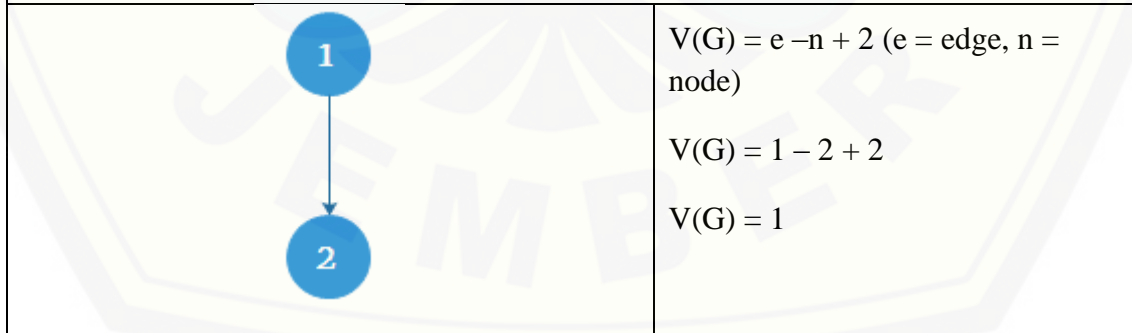
1. *Basis Set*
Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 – 2 – 3

2. *Test Case*
Basis set dicoba dan *basis set* sukses yang dihasilkan 1 – 2 – 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

c. *Whitebox Testing* Daftar Obat Habis

B. Daftar Obat Habis	
✓ Controller Daftar Obat Habis [cont.php/daftar_obat_habis]	
<pre> 300 public function daftar_obat_habis() { 301 \$data['obat_habis'] = \$this->m_obat->get_obat_habis(); 302 \$this->page('include/data_obat/daftar_obat_habis', \$data); 303 } </pre>	



3. *Basis Set*
Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas

siklomatik, yaitu:

Jalur 1 = 1 - 2

4. Test Case

Basis set dicoba dan basis set sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

✓ Model Daftar Obat Habis [m_obat.php/daftar_obat_habis]

```

238 function get_obat_habis() {
239     $query = 'select a.kode_obat, a.nama_obat, a.stok, b.nama_supplier,
240             b.nama_sales,b.tlp_sales
241             from tb_obat a
242             join tb_supplier b
243             on a.id_supplier = b.id_supplier
244             where stok <= 50';
245     $result = $this->db->query($query);
246     return $result->result_array();
247 }
    
```



$V(G) = e - n + 2$ (e = edge, n = node)

$V(G) = 2 - 3 + 2$

$V(G) = 1$

3. Basis Set

Basis set menghasilkan dua jalur independent dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2 - 3

4. Test Case

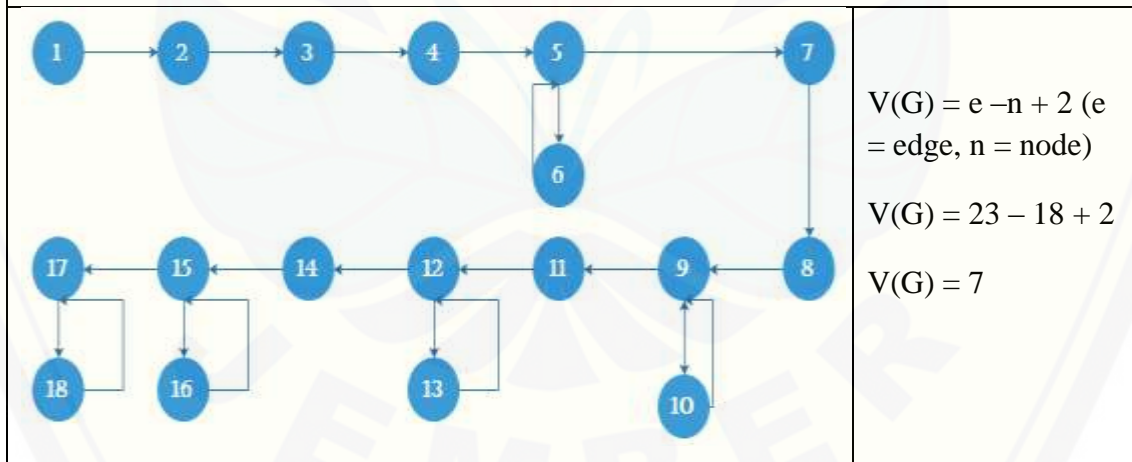
Basis set dicoba dan basis set sukses yang dihasilkan 1 - 2 - 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

d. *Whitebox Testing* Transaksi Pembelian Obat

C. Transaksi Pembelian Obat	
✓ Controller Transaksi Pembelian Obat [cont.php/simpan_pembelian_obat]	
<pre> 257 public function simpan_pembelian_obat() { 258 1 \$tgl_transaksi = date('Y-m-d', strtotime(\$this->input->post('tgl_transaksi'))); 259 2 \$harga_total = \$this->input->post('total'); 260 261 //detailnya 262 3 \$kode_obat = \$this->input->post('kode_obat'); 263 \$qty = \$this->input->post('qty'); 4 264 5 \$harga = \$this->input->post('harga'); 265 266 6 \$this->m_obat->simpan_pembelian_obat(267 \$tgl_transaksi, \$harga_total, 268 \$kode_obat, \$qty, \$harga); 269 270 7 \$this->m_obat->update_stok_obat_beli(\$kode_obat, \$qty); 271 272 redirect(base_url()."index.php/cont/trans_pembelian_obat"); 8 273 } </pre>	
	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 7 - 8 + 2$ $V(G) = 1$
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	
✓ Model Transaksi Pembelian Obat [m_obat.php/simpan_pembelian_obat]	


```

110 public function simpan_pembelian_obat($tgl_transaksi, $harga_total, $kode_obat, $qty, $harga) {
111     $data = array(
112         1 "tgl_pembelian" => $tgl_transaksi,
113         "harga_total" => $harga_total);
114
115     $insert_1 = $this->db->insert('tb_transaksi_pembelian_obat', $data); 2
116
117     $id_transaksi = ""; 3
118     4 $selectID = $this->db->query("select id_transaksi from tb_transaksi_pembelian_obat order by id_transaksi desc limit 1");
119     5 foreach ($selectID->result() as $row) {
120         $id_transaksi = $row->id_transaksi; 6
121     }
122     $arr_kode_obat = array();
123     $arr_qty = array();
124     $arr_harga = array();
125     $i = 0; 8
126     9 foreach ($kode_obat as $kode_obat => $a) {
127         $arr_kode_obat[$i] = $a;
128         $i++;
129     }
130     11 $i = 0;
131     foreach ($qty as $qty => $b) { 12
132         $arr_qty[$i] = $b;
133         $i++;
134     }
135     14 $i = 0;
136     foreach ($harga as $harga => $c) { 15
137         $arr_harga[$i] = $c;
138         $i++;
139     }
140     17 for ($k = 0; $k < count($arr_harga); $k++) {
141         $insert_detail = $this->db->query("INSERT INTO tb_detail_transaksi_pembelian_obat(
142             18 id_transaksi, kode_obat, qty ) values (
143                 ' . $id_transaksi . ',
144                 ' . $arr_kode_obat[$k] . ',
145                 ' . $arr_qty[$k] . '
146             )"");
147     }
148 }
    
```



$V(G) = e - n + 2$ (e = edge, n = node)
 $V(G) = 23 - 18 + 2$
 $V(G) = 7$

1. Basis Set

Basis set menghasilkan dua jalur independent dari perhitungan kompleksitas siklotatik, yaitu:

Jalur 1 = 1 - 2 - 3 - 4 - 5 - 6 - 5 - 7 - 8 - 9 - 10 - 9 - 11 - 12 - 13 - 12 - 14 - 15 - 16 - 15 - 17 - 18 - 17

Jalur 2 = 1 - 2 - 3 - 4 - 5 - 6 - 5 - 7 - 8 - 9 - 11 - 12 - 14 - 15 - 17

Jalur 3 = 1 - 2 - 3 - 4 - 5 - 6 - 5 - 7 - 8 - 9 - 11 - 12 - 14 - 15 - 17

Jalur 4 = 1 - 2 - 3 - 4 - 5 - 7 - 8 - 9 - 10 - 9 - 11 - 12 - 14 - 15 - 17

Jalur 5 = 1 - 2 - 3 - 4 - 5 - 7 - 8 - 9 - 11 - 12 - 13 - 12 - 14 - 15 - 17

Jalur 6 = 1 - 2 - 3 - 4 - 5 - 7 - 8 - 9 - 11 - 12 - 14 - 15 - 16 - 15 - 17

Jalur 7 = 1 - 2 - 3 - 4 - 5 - 7 - 8 - 9 - 11 - 12 - 14 - 15 - 17 - 18 - 17

2. Test Case

Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2 - 3 - 4 - 5 - 6 - 5 - 7 - 8 - 9 - 10 - 9 - 11 - 12 - 13 - 12 - 14 - 15 - 16 - 15 - 17 - 18 - 17, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

e. Whitebox Testing Transaksi Penjualan Obat

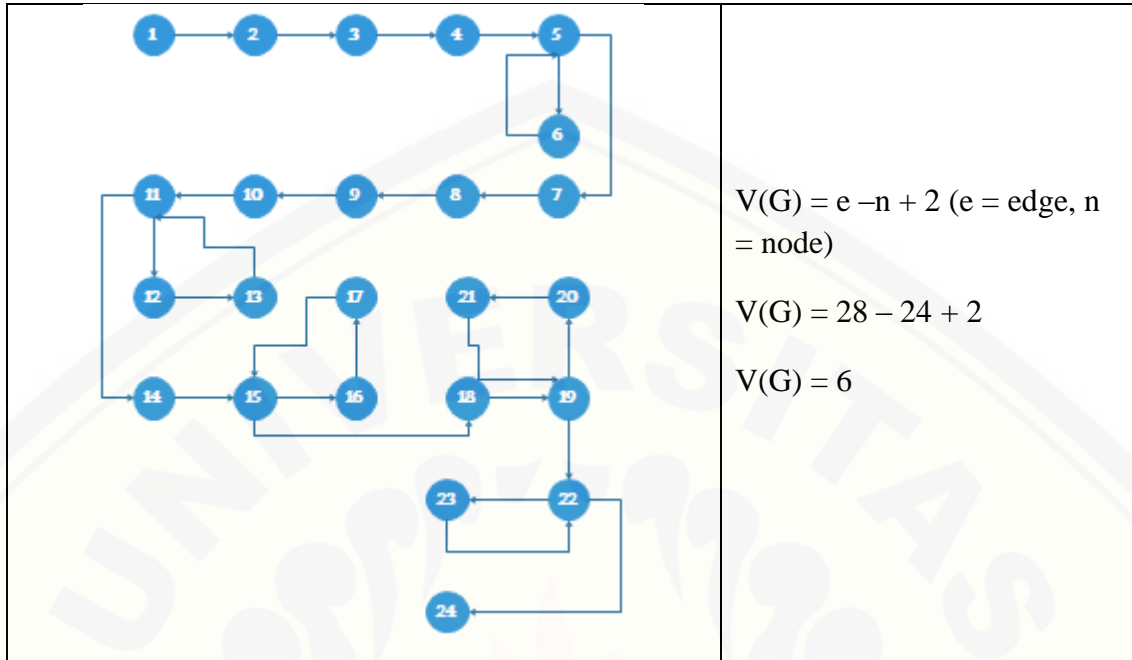
D. Transaksi Penjualan Obat	
✓ Controller Transaksi Penjualan Obat [cont.php/simpan_penjualan_obat]	
<pre> 277 278 279 280 281 282 283 284 285 286 287 288 289 291 292 293 294 295 296 </pre>	<pre> public function simpan_penjualan_obat() { 1 \$tgl_transaksi = date('Y-m-d', strtotime(\$this->input->post('tgl_transaksi'))); 2 \$id_pasien = \$this->input->post('id_pasien'); 3 \$biaya_pemeriksaan = \$this->input->post('biaya_pemeriksaan'); 4 \$harga_total = \$this->input->post('total'); //detailnya 5 \$kode_obat = \$this->input->post('kode_obat'); 6 \$qty = \$this->input->post('qty'); 7 \$harga = \$this->input->post('harga'); 8 \$this->m_obat->simpan_penjualan_obat(\$tgl_transaksi, \$id_pasien, \$biaya_pemeriksaan, \$harga_total, \$kode_obat, \$qty, \$harga); 9 \$this->m_obat->update_stok_obat_jual(\$kode_obat, \$qty); \$this->load->view('include/cetak/cetak/cetakNotaPenjObat'); //redirect(base_url()."index.php/cont/trans_penjualan_obat"); } </pre>

	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 9 - 10 + 2$ $V(G) = 1$
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	
<p>✓ Model Transaksi Penjualan Obat [m_obat.php/simpan_penjualan_obat]</p>	

```

54 public function simpan_penjualan_obat($tgl_transaksi,$id_pasien,
55                                     $biaya_pemeriksaan,$harga_total,$kode_obat,
56                                     $qty,$harga){
57     1 $data = array(
58         "tgl_transaksi" => $tgl_transaksi,
59         "id_pasien" => $id_pasien,
60         "biaya_pemeriksaan" => $biaya_pemeriksaan,
61         "harga_total" => $harga_total);
62     2 $insert_1 = $this->db->insert('tb_transaksipenjualanobat', $data);
63     $id_transaksi = "3";
64     3 $selectID = $this->db->query("select id_transaksi from "
65     4     . "tb_transaksipenjualanobat order by id_transaksi desc limit 1");
66     5 foreach($selectID -> result() as $row)
67     {
68         $id_transaksi = $row -> id_transaksi; 6
69     }
70     7 $arr_kode_obat = array();
71     $arr_qty = array(); 8
72     $arr_harga = array();
73     $i = 0; 1
74     1 foreach($kode_obat as $kode_obat => $a){
75         $arr_kode_obat[$i] = $a;
76         $i++;
77     }
78     $i = 0; 1
79     1 foreach($qty as $qty => $b){
80         $arr_qty[$i] = $b; 1
81         $i++;
82     }
83     $i = 0; 18
84     1 foreach($harga as $harga => $c){
85         $arr_harga[$i] = $c; 2
86         $i++;
87     }
88     2 foreach($k = 0;$k<count($arr_harga);$k++){
89         $insert_detail = $this->db->query("INSERT INTO tb_detail_transaksipenjualanobat(
90         id_transaksi,kode_obat, qty,harga) values (
91         ".$id_transaksi.",
92         ".$arr_kode_obat[$k].", 23
93         ".$arr_qty[$k].",
94         ".$arr_harga[$k].
95         )");
96     }
97 }
98 }
99 }

```



$$V(G) = e - n + 2 \text{ (e = edge, n = node)}$$

$$V(G) = 28 - 24 + 2$$

$$V(G) = 6$$

3. Basis Set

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 – 2 – 3 – 4 – 5 – 6 – 5 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 11 – 14 – 15 – 16 – 17 – 15 – 18 – 19 – 20 – 21 – 19 – 22 – 23 – 22 – 24

Jalur 2 = 1 – 2 – 3 – 4 – 5 – 6 – 5 – 7 – 8 – 9 – 10 – 11 – 14 – 15 – 18 – 19 – 22 – 24

Jalur 3 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 11 – 14 – 15 – 18 – 19 – 22 – 24

Jalur 4 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 9 – 10 – 11 – 14 – 15 – 16 – 17 – 15 – 18 – 19 – 22 – 24

Jalur 5 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 9 – 10 – 11 – 14 – 15 – 18 – 19 – 20 – 21 – 19 – 22 – 24

Jalur 6 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 9 – 10 – 11 – 14 – 15 – 18 – 19 – 22 – 23 – 22 – 24

4. Test Case

Basis set dicoba dan basis set sukses yang dihasilkan 1 – 2 – 3 – 4 – 5 – 6 – 5 – 7 –

8 – 9 – 10 – 11 – 12 – 13 – 11 – 14 – 15 – 16 – 17 – 15 – 18 – 19 – 20 – 21 – 19 – 22 – 23 – 22 – 24, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

E.3. Whitebox Testing Tahap Increment Tiga

a. Whitebox Testing Data Frame

1. Tambah Data Frame	
✓ Controller Tambah Data frame [cont.php/insert_frame]	
<pre> function insert_frame() { 340 341 \$input = array(342 1 "id_frame" => \$this->input->post('id_frame'), 343 "merk_frame" => \$this->input->post('merk_frame'), 344 "detail_frame" => \$this->input->post('detail_frame'), 345 "harga_beli" => \$this->input->post('harga_beli'), 346 "harga_jual" => \$this->input->post('harga_jual'), 347 "stok" => \$this->input->post('stok') 348); 349 2 \$insert = \$this->m_kacamata->insert_frame(\$input); 350 redirect(base_url() . "index.php/cont/data_frame?status=\$insert"); 3 351 } 352 </pre>	
<pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) </pre>	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 2 - 3 + 2$ $V(G) = 1$
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2 - 3</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2 - 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah</p>	

memenuhi syarat kelayakan.

✓ Model Tambah Data frame [m_kacamata.php/insert_frame]

```

6   function insert_frame($input){
7       $query = $this->db->insert('tb_frame', $input);
8
9       return $query; 2
10  }
    
```



$V(G) = e - n + 2$ (e = edge, n = node)

$V(G) = 1 - 2 + 2$

$V(G) = 1$

1. Basis Set

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2

2. Test Case


Basis set dicoba dan basis set sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

2. Edit Data Frame

Controller Edit Data Frame [cont.php/edit_frame]

```

361 function edit_frame() {
362
363     $input = array(
364         1 "id_frame" => $this->input->post('id_frame'),
365         "merk_frame" => $this->input->post('merk_frame'),
366         "detail_frame" => $this->input->post('detail_frame'),
367         "harga_beli" => $this->input->post('harga_beli'),
368         "harga_jual" => $this->input->post('harga_jual'),
369         "stok" => $this->input->post('stok')
370     );
371     2 $logic = $this->m_kacamata->update_frame($input);
372     redirect(base_url() . "index.php/cont/data_frame?status=$logic"); 3
373 }
    
```

	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 2 - 3 + 2$ $V(G) = 1$
---	---

3. *Basis Set*
Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:


Jalur 1 = 1 - 2 - 3

4. *Test Case*
Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2 - 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

Model Edit Data Frame [m_kacamata.php/update_frame]

```

23 function update_frame($input) {
24     1 $this->db->where('id_frame', $input['id_frame']);
25     return $this->db->update('tb_frame', $input); 2
26 }
    
```

	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 1 - 2 + 2$ $V(G) = 1$
---	---

3. *Basis Set*
Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2

4. *Test Case*

Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

3. Hapus Data Frame

Controller Hapus Data Frame[cont.php/hapus_frame]

```

374 function hapus_frame($id_frame) {
375     $this->m_kacamata->hapus_frame($id_frame);
376     redirect(base_url() . "index.php/cont/data_frame?status=$id_frame");
377 }
    
```



$$V(G) = e - n + 2 \text{ (e = edge, n = node)}$$

$$V(G) = 1 - 2 + 2$$

$$V(G) = 1$$

1. *Basis Set*

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2

2. *Test Case*

Basis set dicoba dan *basis set* sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

Model Hapus Data Frame[m_kacamata.php/hapus_frame]

```

27 function hapus_frame($id_frame) {
28     $query = $this->db->query("DELETE FROM tb_frame WHERE id_frame = '".$id_frame.'");
29
30     if($query) {
31         RETURN TRUE;
32     } else {
33         RETURN FALSE;
34     }
35 }
    
```

<pre> graph TD 1((1)) --> 2{2} 2 --> 3((3)) 2 --> 4((4)) 3 --> 5((5)) 4 --> 5 </pre>	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 5 - 5 + 2$ $V(G) = 2$
<p>3. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2 - 3 - 5 Jalur 2 = 1 - 2 - 4 - 5</p> <p>4. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2 - 3 - 5, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	

b. *Whitebox Testing* Transaksi Penjualan Kacamata

E. Transaksi Penjualan Kacamata

✓ Controller Transaksi Penjualan Kacamata [cont.php/simpan_penjualan_kacamata]


```

public function simpan_penjualan_kacamata() {
    $id_transaksi = $this->input->post('id_transaksi');
    $tgl_transaksi = date('Y-m-d', strtotime($this->input->post('tgl_transaksi')));
    $biaya_pemeriksaan = $this->input->post('biaya_pemeriksaan');
    $id_pasien = $this->input->post('id_pasien');
    $harga_total = $this->input->post('total');
    $dp = $this->input->post('dp');
    $status = $this->input->post('status');

    $kurang = $this->input->post('kurang');

    $tgl_pengambilan = date('Y-m-d', strtotime($this->input->post('tgl_pengambilan')));

    //detailnya
    $id_frame = $this->input->post('id_frame');
    $jenis_lensa = $this->input->post('jenis_lensa');
    $jenis_kanan = $this->input->post('jenis_kanan');
    $ukuran_kanan = $this->input->post('ukuran_kanan');
    $jenis_kiri = $this->input->post('jenis_kiri');
    $ukuran_kiri = $this->input->post('ukuran_kiri');
    $harga = $this->input->post('harga');

    $this->m_kacamata->simpan_penjualan_kacamata(
        $tgl_transaksi, $id_pasien, $biaya_pemeriksaan, $harga_total,
        $dp, $status, $kurang, $tgl_pengambilan, $id_frame, $jenis_lensa,
        $jenis_kanan, $ukuran_kanan, $jenis_kiri, $ukuran_kiri, $harga);
    $this->m_kacamata->update_stok_frame_jual($id_frame);
    redirect(base_url() . "index.php/cont/trans_penjualan_kcmt");
}

```



$$V(G) = e - n + 2 \quad (e = \text{edge}, n = \text{node})$$

$$V(G) = 2 - 3 + 2$$

$$V(G) = 1$$

3. Basis Set

Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:

$$\text{Jalur 1} = 1 - 2 - 3$$

4. Test Case

Basis set dicoba dan basis set sukses yang dihasilkan 1 - 2 - 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

✓ Model Transaksi Penjualan Kacamata

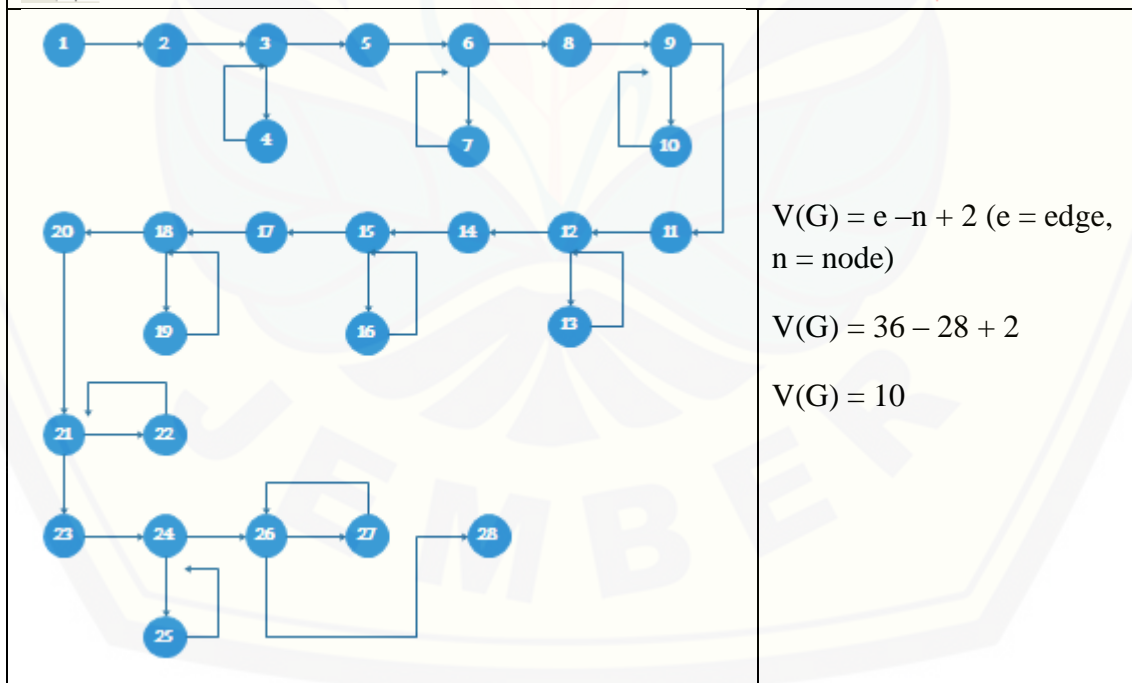
```

[m_kacamata.php/simpan_penjualan_kacamata]
70 public function simpan_penjualan_kacamata(
71     $tgl_transaksi, $id_pasien, $biaya_pemeriksaan, $harga_total, $dp, $status,
72     $kurang, $tgl_pengambilan, $frame, $jenis_lensa, $jenis_kanan, $ukuran_kanan,
73     $jenis_kiri, $ukuran_kiri, $harga
74 ) {
75     $data = array(
76         1 "tgl_transaksi" => $tgl_transaksi,
77         "id_pasien" => $id_pasien,
78         "biaya_pemeriksaan" => $biaya_pemeriksaan,
79         "harga_total" => $harga_total,
80         "dp" => $dp,
81         "status" => $status,
82         "kurang" => $kurang,
83         "tgl_pengambilan" => $tgl_pengambilan);
84
85     $insert_id = $this->db->insert('tb_transaksipenjualankacamata', $data);
86     2
87     $id_transaksi = "";
88     $selectID = $this->db->query("select id_transaksi from "
89         . "tb_transaksipenjualankacamata order by id_transaksi desc limit 1");
90     3foreach ($selectID->result() as $row) {
91         $id_transaksi = $row->id_transaksi; 4
92     }
93
94     $arr_frame = array();
95     5$rr_jenis_lensa = array();
96     $arr_jenis_kanan = array();
97     $arr_ukuran_kanan = array();
98     $arr_jenis_kiri = array();
99     $arr_ukuran_kiri = array();
100    $arr_harga = array();
101
102    $i = 0;
103    6foreach ($frame as $frame => $a) {
104        $arr_frame[$i] = $a;
105        $i++;
106    }
107    8$i = 0;
108    9foreach ($jenis_lensa as $jenis_lensa => $b) {
109        $arr_jenis_lensa[$i] = $b;
110        $i++;
111    }
112    11$i = 0;
113    12foreach ($jenis_kanan as $jenis_kanan => $c) {
114        $arr_jenis_kanan[$i] = $c;
115        $i++;
116    }
117    14$i = 0;
118    15foreach ($ukuran_kanan as $ukuran_kanan => $d) {
119        $arr_ukuran_kanan[$i] = $d;
120        $i++;
121    }
122    17$i = 0;
123    18foreach ($jenis_kiri as $jenis_kiri => $e) {
124        $arr_jenis_kiri[$i] = $e;
125        $i++;
126    }
127    20$i = 0;
128    21foreach ($ukuran_kiri as $ukuran_kiri => $f) {
129        $arr_ukuran_kiri[$i] = $f;
130        $i++;
131    }
132
133    $i = 0;

```

```

132     $i = 0;
133     24 foreach ($harga as $harga => $g) {
134         $arr_harga[$i] = $g;
135         25 $i++;
136     }
137     26 for ($k = 0; $k < count($arr_harga); $k++) {
138         $insert_detail = $this->db->query("INSERT INTO tb_detail_transaksipenjualanlacamata(
139             27 id_transaksi,
140             id_frame,
141             lensa,
142             jenis_kanan,
143             ukuran_kanan,
144             jenis_kiri,
145             ukuran_kiri,
146             harga
147         ) values (
148             "" . $id_transaksi . "",
149             "" . $arr_frame[$k] . "",
150             "" . $arr_jenis_lensa[$k] . "",
151             "" . $arr_jenis_kanan[$k] . "",
152             "" . $arr_ukuran_kanan[$k] . "",
153             "" . $arr_jenis_kiri[$k] . "",
154             "" . $arr_ukuran_kiri[$k] . "",
155             "" . $arr_harga[$k] . "
156         )");
157     }
158     //print_r($arr_frame);
159 }
160 }
    
```



$V(G) = e - n + 2$ (e = edge, n = node)

$V(G) = 36 - 28 + 2$

$V(G) = 10$

1. Basis Set

Basis set menghasilkan dua jalur independent dari perhitungan kompleksitas

siklomatik, yaitu:

Jalur 1 = 1 – 2 – 3 – 4 – 3 – 5 – 6 – 7 – 6 – 8 – 9 – 10 – 9 – 11 – 12 – 13 – 12 – 14 – 15 – 16 – 15 – 17 – 18 – 19 – 18 – 20 – 21 – 22 – 21 – 23 – 24 – 25 – 24 – 26 – 27 – 26 – 28

Jalur 2 = 1 – 2 – 3 – 4 – 3 – 5 – 6 – 8 – 9 – 11 – 12 – 14 – 15 – 17 – 18 – 20 – 21 – 23 – 24 – 26 – 28

Jalur 3 = 1 – 2 – 3 – 5 – 6 – 7 – 6 – 8 – 9 – 11 – 12 – 14 – 15 – 17 – 18 – 20 – 21 – 23 – 24 – 26 – 28

Jalur 4 = 1 – 2 – 3 – 5 – 6 – 8 – 9 – 10 – 9 – 11 – 12 – 14 – 15 – 17 – 18 – 20 – 21 – 23 – 24 – 26 – 28

Jalur 5 = 1 – 2 – 3 – 5 – 6 – 8 – 9 – 11 – 12 – 13 – 12 – 14 – 15 – 17 – 18 – 20 – 21 – 23 – 24 – 26 – 28

Jalur 6 = 1 – 2 – 3 – 5 – 6 – 8 – 9 – 11 – 12 – 14 – 15 – 16 – 15 – 17 – 18 – 20 – 21 – 23 – 24 – 26 – 28

Jalur 7 = 1 – 2 – 3 – 5 – 6 – 8 – 9 – 11 – 12 – 14 – 15 – 17 – 18 – 19 – 18 – 20 – 21 – 23 – 24 – 26 – 28

Jalur 8 = 1 – 2 – 3 – 5 – 6 – 8 – 9 – 11 – 12 – 14 – 15 – 17 – 18 – 20 – 21 – 22 – 21 – 23 – 24 – 26 – 28


Jalur 9 = 1 – 2 – 3 – 5 – 6 – 8 – 9 – 11 – 12 – 14 – 15 – 17 – 18 – 20 – 21 – 23 – 24 – 25 – 24 – 26 – 28


Jalur 10 = 1 – 2 – 3 – 5 – 6 – 8 – 9 – 11 – 12 – 14 – 15 – 17 – 18 – 20 – 21 – 23 – 24 – 26 – 27 – 26 – 28

2. *Test Case*

Basis set dicoba dan *basis set* sukses yang dihasilkan 1 – 2 – 3 – 4 – 3 – 5 – 6 – 7 – 6 – 8 – 9 – 10 – 9 – 11 – 12 – 13 – 12 – 14 – 15 – 16 – 15 – 17 – 18 – 19 – 18 – 20 – 21 – 22 – 21 – 23 – 24 – 25 – 24 – 26 – 27 – 26 – 28, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

c. *Whitebox Testing* Transaksi Pelunasan Kacamata

A. Transaksi Pelunasan Kacamata	
✓ Controller Transaksi Pelunasan Kacamata [cont.phptransaksi_pelunasan_kacamata]	
<pre> 447 □ 448 449 450 451 </pre>	<pre> public function transaksi_pelunasan_kacamata(\$id_transaksi){ 1 \$this->m_kacamata->update_status_lunas(\$id_transaksi); 2 redirect("cont/trans_pelunasan_kcmt"); } </pre>
 <pre> graph TD 1((1)) --> 2((2)) </pre>	<p>$V(G) = e - n + 2$ (e = edge, n = node)</p> <p>$V(G) = 1 - 2 + 2$</p> <p>$V(G) = 1$</p>
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 – 2</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 – 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	
✓ Model Transaksi Pelunasan Kacamata [m_kacamata.php/update_status_lunas]	
<pre> 199 □ 200 201 202 203 </pre>	<pre> public function update_status_lunas(\$id_transaksi){ 1 \$this->db->where('id_transaksi', \$id_transaksi); 2 return \$this->db->update('tb_transaksipenjualankacamata', array("status" => "LUNAS", "kurang" => 0)); } </pre>

	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 1 - 2 + 2$ $V(G) = 1$
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 – 2</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 – 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	

E.4. Whitebox Testing Tahap Increment Empat

a. Whitebox Testing Input Data Pemeriksaan



B. Input Data Pemeriksaan

✓ Controller Input Data Pemeriksaan [cont.php/insert_pemeriksaan]

```

537 function insert_pemeriksaan() {
538     $input = array(
539         //AUTO INCREMENT GIMANA
540         "id_pasien" => $this->input->post('id_pasien'),
541         "tgl_pemeriksaan" => date('Y-m-d', strtotime($this->input->post('tgl_periksa'))),
542         "status_pasien" => $this->input->post('status_pasien'),
543         "subyektif" => $this->input->post('subyektif'),
544         "obyektif" => $this->input->post('obyektif'),
545         "assesment" => $this->input->post('assesment'),
546         "planning" => $this->input->post('planning')
547     );
548     $insert = $this->m_rekammedis->insert_pemeriksaan($input);
549     3redirect(base_url() . "index.php/cont/add_data_pemeriksaan?status=$insert");
550 }

```

	<p>$V(G) = e - n + 2$ (e = edge, n = node)</p> <p>$V(G) = 2 - 3 + 2$</p> <p>$V(G) = 1$</p>
<p>3. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2 - 3</p> <p>4. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2 - 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	
<p>✓ Model Input Data Pemeriksaan [m_rekammedis.php/insert_pemeriksaan]</p>	
<pre> 8 function insert_pemeriksaan(\$input){ 9 1 \$query = \$this->db->insert('tb_pemeriksaan', \$input); 10 11 return \$query;2 12 }</pre>	
	<p>$V(G) = e - n + 2$ (e = edge, n = node)</p> <p>$V(G) = 1 - 2 + 2$</p> <p>$V(G) = 1$</p>
<p>3. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 - 2</p> <p>4. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 - 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah</p>	

memenuhi syarat kelayakan.

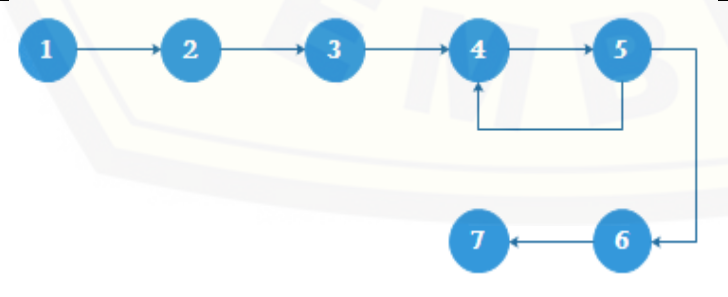
b. *Whitebox Testing* Lihat Data Rekam Medis

A. Lihat Data Rekam Medis

✓ *Controller* Lihat Data Rekam Medis [cont.php/get_data_rekammedis]

```

493 public function get_data_rekammedis() {
494     1 $id = $this->input->post('nama_pasien');
495     2 $dataa = $this->m_rekammedis->get_data_pemeriksaan($id);
496     3 echo "Nama : ".$dataa[0]['nama_pasien']."<br>";
497     echo "TTL : ".$dataa[0]['ttl']."<br>";
498     echo "Umur : ".$dataa[0]['umur']."<br>";
499     echo "Alamat : ".$dataa[0]['alamat']."<br>";
500     $string = "<table class='table table-striped table-bordered table-hover'>
501         <thead>
502             <tr>
503                 <th style=''>No</th>
504                 <th style='width: 20%'>Tanggal</th>
505                 <th style='width: 10%'>Status</th>
506                 <th style='width: 15%'>Subyektif</th>
507                 <th style='width: 15%'>Obyektif</th>
508                 <th style='width: 15%'>Assesment</th>
509                 <th style='width: 15%'>Planning</th>
510             </tr>
511         </thead>
512         <tbody id='listdata1'>";
513     $i = 0;
514     4 foreach($dataa as $d) :
515         $string .= '<tr>
516             5 <td>' . ++$i . '</td>
517             <td>' . $d['tgl_pemeriksaan'] . '</td>
518             <td>' . $d['status_pasien'] . '</td>
519             <td>' . $d['subyektif'] . '</td>
520             <td>' . $d['obyektif'] . '</td>
521             <td>' . $d['assesment'] . '</td>
522             <td>' . $d['planning'] . '</td>
523         </tr>';
524     6 endforeach;
525     $string.="</tbody></table>";
526     7 echo $string;
    
```



$V(G) = e - n + 2$ (e = edge, n = node)

$V(G) = 7 - 7 + 2$

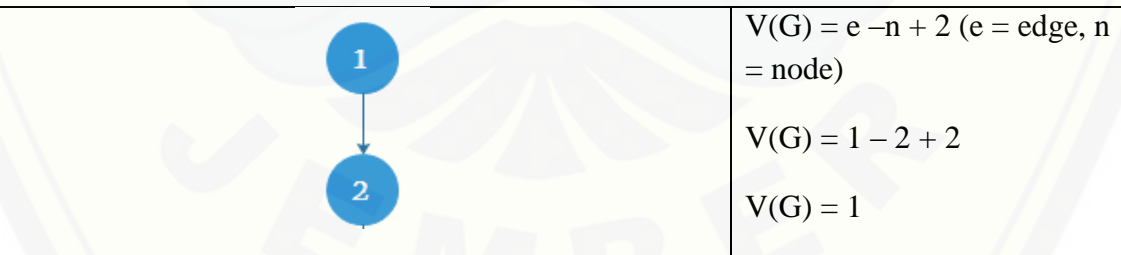
$V(G) = 2$

5. *Basis Set*
Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:
- Jalur 1 = 1 – 2 – 3 – 4 – 5 – 4 – 5 – 6 – 7
- Jalur 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7
6. *Test Case*
Basis set dicoba dan *basis set* sukses yang dihasilkan Jalur 1 = 1 – 2 – 3 – 4 – 5 – 4 – 5 – 6 – 7, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

✓ Model Input Data Pemeriksaan [m_rekammedis.php/insert_pemeriksaan]

```

function get_data_pemeriksaan($id_pasien){
14     $query = $this->db->query('SELECT a.id_pasien, a.nama_pasien,
15         concat(a.kota_lahir, " ", a.tgl_lahir) as ttl,
16         YEAR(CURDATE()) - YEAR(a.tgl_lahir) as umur,
17         concat(a.alamat, " ", a.kecamatan, " ", a.kabupaten) as alamat,
18         a.no_tlp,
19         a.pekerjaan,
20         a.nama_wali,
21         a.no_tlp_wali,
22         b.tgl_pemeriksaan,
23         b.status_pasien,
24         b.subyektif,
25         b.obyektif,
26         b.assessment,
27         b.planning
28     FROM tb_pasien a join tb_pemeriksaan b on a.id_pasien = b.id_pasien where a.id_pasien='.$id_pasien);
29
30     2 return $query->result_array();
31 }
    
```




5. *Basis Set*
Basis set menghasilkan dua jalur *independent* dari perhitungan kompleksitas siklomatik, yaitu:
- Jalur 1 = 1 – 2
6. *Test Case*

Basis set dicoba dan *basis set* sukses yang dihasilkan 1 – 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

c. Whitebox Testing Pendaftaran Operasi

A. Pendaftaran Operasi	
✓ Controller Pendaftaran Operasi [cont.php/insert_pend_operasi]	
<pre> 569 function insert_pend_operasi() { 570 \$input = array(571 "id_pasien" => \$this->input->post('id_pasien'), 572 "tgl_operasi" => \$this->input->post('tgl_operasi'), 573 "no_urut" => \$this->input->post('kuota') 574); 575 576 2 \$insert = \$this->m_operasi->insert_pend_operasi(\$input); 577 578 3 redirect(base_url() . "index.php/cont/pendaftaran_operasi?status=\$insert"); 579 }</pre>	
<pre> graph LR 1((1)) --> 2((2)) 2((2)) --> 3((3))</pre>	$V(G) = e - n + 2 \text{ (e = edge, n = node)}$ $V(G) = 2 - 3 + 2$ $V(G) = 1$
<p>1. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 – 2 – 3</p> <p>2. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan Jalur 1 = 1 – 2 – 3, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	
✓ Model Input Data Pemeriksaan [m_rekammedis.php/insert_pemeriksaan]	
<pre> 9 function insert_pend_operasi(\$input) { 10 \$query = \$this->db->insert('tb_operasi', \$input); 11 12 return \$query; 13 }</pre>	

	$V(G) = e - n + 2$ (e = edge, n = node) $V(G) = 1 - 2 + 2$ $V(G) = 1$
<p>7. <i>Basis Set</i> <i>Basis set</i> menghasilkan dua jalur <i>independent</i> dari perhitungan kompleksitas siklomatik, yaitu:</p> <p>Jalur 1 = 1 – 2</p> <p>8. <i>Test Case</i> <i>Basis set</i> dicoba dan <i>basis set</i> sukses yang dihasilkan 1 – 2, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.</p>	

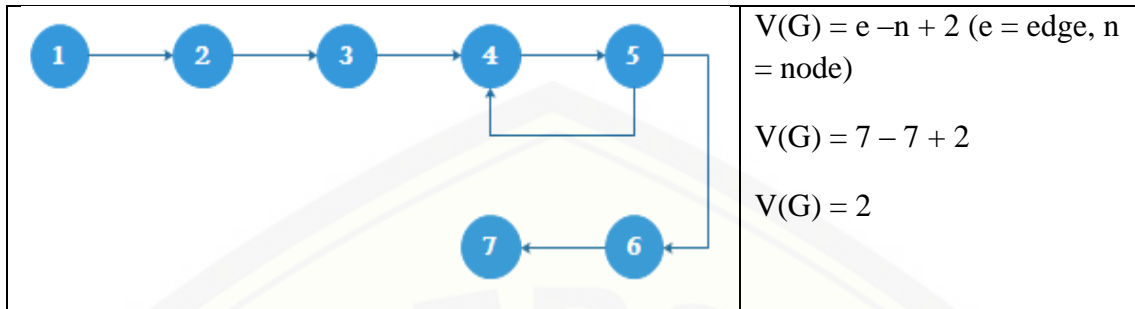
d. *Whitebox Testing* Lihat Jadwal Operasi

A. Lihat Data Rekam Medis

✓ *Controller* Lihat Data Rekam Medis [cont.php/get_data_rekammedis]

```

584 public function get_jadwal_operasi() {
585     1 $tgl = $this->input->post('tgl_operasi');
586     2 $dataaa = $this->m_operasi->get_jadwal_operasi($tgl);
587     3 $string = "";
588     3 $i = 0;
589
590     4 foreach($dataaa as $d) :
591         5 $string .= '
592             <tr>
593                 <td>' . ++$i . '</td>
594                 <td>' . $d['nama_pasien'] . '</td>
595                 <td>' . $d['no_urut'] . '</td>
596             </tr>';
597     6 endforeach;
598     7 echo $string;
599
600 }
```



$V(G) = e - n + 2$ (e = edge, n = node)

$V(G) = 7 - 7 + 2$

$V(G) = 2$

3. Basis Set

Basis set menghasilkan dua jalur independent dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2 - 3 - 4 - 5 - 4 - 5 - 6 - 7

Jalur 1 = 1 - 2 - 3 - 4 - 5 - 6 - 7

4. Test Case

Basis set dicoba dan basis set sukses yang dihasilkan Jalur 1 = 1 - 2 - 3 - 4 - 5 - 4 - 5 - 6 - 7, dapat dilihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.

✓ Model Input Data Pemeriksaan [m_rekammedis.php/insert_pemeriksaan]

```

21 function get_jadwal_operasi($tgl) {
22     $query = $this->db->query("SELECT a.nama_pasien, b.no_urut
23     FROM tb_pasien a join tb_operasi b on a.id_pasien = b.id_pasien where b.tgl_operasi = '". $tgl.'";
24
25     return $query->result_array();
26 }
  
```



$V(G) = e - n + 2$ (e = edge, n = node)

$V(G) = 1 - 2 + 2$

$V(G) = 1$

9. Basis Set

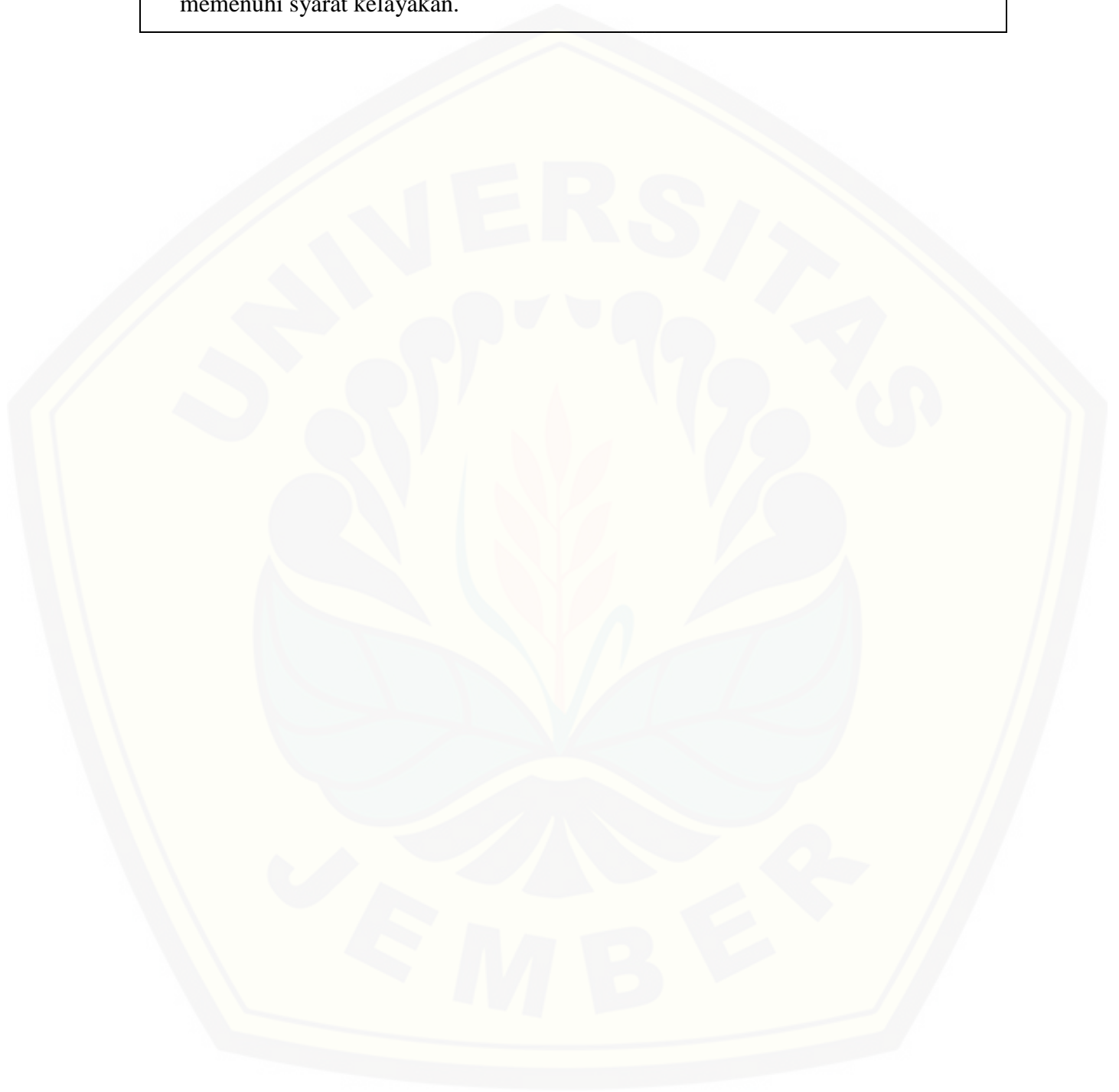
Basis set menghasilkan dua jalur independent dari perhitungan kompleksitas siklomatik, yaitu:

Jalur 1 = 1 - 2

10. Test Case

Basis set dicoba dan basis set sukses yang dihasilkan 1 - 2, dapat dilihat bahwa

simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut sistem telah memenuhi syarat kelayakan.



46	Wahib	25	Laki-laki	Tidak	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
47	M.Romli	32	Laki-laki	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
48	Zulfa	12	Perempuan	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
49	Manaf	16	Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
50	Abd Majid	14	Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
51	Yanti	20	Perempuan	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
52	Hasan	18	Laki-laki	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
53	Suhairiyah	48	Perempuan	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
54	Jamilah	5	Perempuan	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
55	jumawah	70	Laki-laki	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
56	ipniyah	17	Perempuan	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
57	rosidi	53	Laki-laki	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
58	haliyah	5	Perempuan	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
59	hosen	60	Laki-laki	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
60	hamidah	55	Perempuan	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
61	maisa	42	Perempuan	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
62	Zainab	32	Perempuan	Tidak	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
63	Ati	60	Perempuan	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Cepat
64	Mahmuda	16	Perempuan	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
65	Syaifulloh	15	Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
66	Masaid	37	Laki-laki	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Sedang
67	Ahmad	45	Laki-laki	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Ya	Sedang
68	Hasiyah	65	Perempuan	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
69	Maria	72	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
70	Jumiyah	52	Perempuan	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
71	Fadilatul	17	Perempuan	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat

72	Nur Halima	19	Perempuan	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
73	Umi Kulsum	16	Perempuan	Tidak	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
74	Maisu	60	Laki-laki	Ya	Tidak	Ya	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
75	Zainal A	19	Laki-laki	Tidak	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
76	Gilang R	8	Laki-laki	Ya	Tidak	Ya	Tidak	Ya	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
77	Firdaus	24	Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
78	Afrina	8	Perempuan	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
79	Rosida	14	Perempuan	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
80	Masluha	18	Perempuan	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
81	Farisa	15	Perempuan	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
82	Fafa	1	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
83	Zakia	19	Perempuan	Ya	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Sedang
84	Eka K.	26	Perempuan	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
85	Mukmin	16	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
86	Mahmuri	17	Perempuan	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
87	Siti Mimuna	17	perempuan	Ya	Tidak	Tidak	Tidak	Ya	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
88	Elisa	34	Laki-laki	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
89	Abd Aziz	26	Perempuan	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
90	Anjani	13	Laki-laki	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
91	Zainab	17	Perempuan	Tidak	Tidak	Ya	Tidak	Ya	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Sedang
92	Abd Latif	26	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
93	Muis	17	Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
94	Heri	23	Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
95	Aisyah	16	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
96	Siti Nadiya	15	Perempuan	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
97	Rahatul H	14	Perempuan	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat

98	Jumriyah	17	Perempuan	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
99	Dedi	18	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
100	Haris	12	Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
101	Fatimah	15	Perempuan	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Sedang
102	Aisyah	21	Perempuan	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
103	Rifatul	16	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
104	Sumriyah	23	Perempuan	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
105	Saimun	75	Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
106	Mariah	65	Perempuan	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
107	Romlah	15	Perempuan	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Lama
108	Khusnul	13	Perempuan	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
109	Zuhri	16	Laki-laki	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
110	Yulianti	16	Perempuan	Ya	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
111	Fauziyah	22	Perempuan	Tidak	Tidak	Ya	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
112	Hatimah	60	Perempuan	Tidak	Tidak	Ya	Tidak	Ya	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Lama
113	Holili	16	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
114	Nova	30	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Lama
115	Ifa	13	Perempuan	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
116	Marsu	50	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
117	Ainiyah	40	Perempuan	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
118	Heri	40	Laki-laki	Ya	Tidak	Ya	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
119	Saibah	18	Perempuan	Tidak	Tidak	Ya	Ya	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Sedang
120	Ulya	15	Perempuan	Ya	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Sedang
121	Sofiyah	17	Perempuan	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
122	Zainudin	16	Laki-laki	Ya	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
123	Abdul Ghofur	17	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat

124	sholihin	31	Laki-laki	Ya	Tidak	Ya	Tidak	Ya	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Lama
125	Laila	90	Perempuan	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Ya	Ya	Lama
126	desi	29	Perempuan	ya	tidak	ya	tidak	ya	ya	ya	tidak	ya	tidak	tidak	tidak	tidak	Sedang
127	susi	36	Perempuan	Tidak	Tidak	Ya	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
128	Dika	45	Laki-laki	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Sedang
129	Mila	19	Perempuan	Ya	Ya	Ya	Tidak	Ya	Ya	Ya	Tidak	Ya	Tidak	Ya	Ya	Tidak	Lama
130	Putra	27	Laki-laki	Tidak	Tidak	Ya	Tidak	Ya	Ya	Ya	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Lama
131	Firman	37	Laki-laki	Tidak	Ya	Ya	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Sedang
132	Ricky	5	Laki-laki	Ya	Tidak	Ya	Tidak	Ya	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Lama
133	Mardhi	24	Laki-laki	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
134	Dio	24	Laki-laki	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Sedang
135	Dedi	36	Laki-laki	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
136	Niki	23	Perempuan	Tidak	Ya	Ya	Tidak	Ya	Ya	Ya	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Lama
137	Wulan	46	Perempuan	Tidak	Ya	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
138	rere	10	Perempuan	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
139	Teguh	38	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Ya	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Sedang
140	Akmal	7	Laki-laki	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
141	Fathur	69	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Sedang
142	Aris	7	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
143	Iftahul	12	Perempuan	Ya	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Sedang
144	Burawi	58	Laki-laki	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
145	Umi	43	Perempuan	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
146	Ahmad	19	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
147	Rudi	16	Laki-laki	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
148	Faridatul	14	Perempuan	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
149	Farah	7	Perempuan	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat

150	Andre	20	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
151	Mualifah	15	Perempuan	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
152	Fatmawati	14	Perempuan	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
153	Lu'luul	16	Perempuan	Ya	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Cepat
154	Siti M	52	Perempuan	Tidak	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
155	Maisanti	25	Perempuan	Tidak	Tidak	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Sedang
156	Wizarotul	15	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Sedang
157	Sumariyah	16	Perempuan	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
158	Arofi	19	Laki-laki	Tidak	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
159	Yuniar	30	Perempuan	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
160	Lilis	29	Perempuan	Tidak	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
161	Zaini	47	Laki-laki	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
162	Martiyem	47	Laki-laki	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
163	Hari	35	Laki-laki	Ya	Tidak	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Sedang
164	Riseh .U	13	Perempuan	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat