



**SISTEM PAKAR DIAGNOSIS PENYAKIT DEMAM TIFOID
MENGUNAKAN METODE *CERTAINTY FACTOR*
(Studi Kasus RSD Balung)**

SKRIPSI

Oleh

**Lintang Harianti
NIM 122410101051**

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER
2016**



**SISTEM PAKAR DIAGNOSIS PENYAKIT DEMAM TIFOID
MENGUNAKAN METODE *CERTAINTY FACTOR*
(Studi Kasus RSD Balung)**

SKRIPSI

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

Oleh
Lintang Harianti
NIM 122410101051

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2016

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT, yang telah memberikan kelancaran dan kemudahan dalam menyelesaikan skripsi ini;
2. Ibunda Pariyatun serta Alm. Ayahanda Djumana tercinta atas segala kesabaran, keikhlasan, limpahan kasih sayang, doa, dan motivasi serta dukungan yang luar biasa;
3. Tante Emy Kusmiati tercinta yang selalu memberikan dukungan;
4. Kakak Nona Citra P.S dan adikku Nur Lela F tercinta.;
5. Saudara-saudaraku berserta seluruh keluarga besar;
6. Sahabat kontrakan Puri Bunga Nirwana yaitu Puput, Seftya, Ara, Gau, Dwi, Jeje, Esa, Delia dan sahabat seperjuangan yaitu Deasy, Qoqom yang selalu memberikan dukungan;
7. Teman-teman seperjuangan kuliah di PSSI yang bersama berjuang menuntut ilmu;
8. Calon ayah dari anak-anak saya kelak;
9. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi;
10. Almamater Program Studi Sistem Informasi Universitas Jember

MOTTO

“Hai orang-orang beriman apabila dikatakan kepadamu: “berlapang-lapanglah dalam majlis” maka lapangkanlah niscaya Allah akan memberi kelapangan untukmu dan apabila dikatakan: “berdirilah kamu” maka berdirilah, niscaya Allah akan meninggikan orang-orang yang beriman diantaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat. Dan Allah Maha mengetahui apa yang kamu kerjakan”. (Terjemahan Surat *QS. Al-Mujadilah* ayat 11)¹

¹ Kementerian Agama Republik Indonesia. 2014. Ummul Mukminin : Al Qur'an dan Terjemahannya untuk Wanita. Jakarta: OASIS Terrace Resident

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Lintang Harianti

NIM : 122410101051

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Pakar Diagnosis Penyakit Demam Tifoid Menggunakan Metode *Certainty Factor* (Studi Kasus RSD Balung)” adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, 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 sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 22 Juni 2016

Yang menyatakan,

Lintang Harianti

NIM 122410101051

SKRIPSI

**SISTEM PAKAR DIAGNOSIS PENYAKIT DEMAM TIFOID
MENGUNAKAN METODE *CERTAINTY FACTOR*
(Studi Kasus RSD Balung)**

Oleh

Lintang Harianti

NIM 122410101051

Pembimbing :

Dosen Pembimbing Utama : Drs. Antonius C.P., M.App.,Sc.,Ph.D

Dosen Pembimbing Pendamping : Muhamad Arief H.,S.Kom.,M.Kom

PENGESAHAN PEMBIMBING

Skripsi berjudul “Sistem Pakar Diagnosis Penyakit Demam Tifoid Menggunakan Metode *Certainty Factor* (Studi Kasus RSD Balung)” telah diuji dan disahkan pada:

hari, tanggal : Rabu, 22 Juni 2016

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh :

Pembimbing I,

Pembimbing II,

Drs. Antonius C.P., M.App.,Sc.,Ph.D

Muhamad Arief H.,S.Kom.,M.Kom

NIP 196909281993021001

NIP 198101232010121003

PENGESAHAN

Skripsi berjudul “Sistem Pakar Diagnosis Penyakit Demam Tifoid Menggunakan Metode *Certainty Factor* (Studi Kasus RSD Balung)” telah diuji dan disahkan pada:

hari, tanggal : Rabu, 22 Juni 2016

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim penguji:

Penguji I,

Penguji II,

Anang Andrianto, S.T., M.T.
NIP 196906151997021002

Windi Eka Yulia R, S.Kom, MT
NIP 198403052010122002

Mengesahkan

Ketua Program Studi,

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

NIP 196704201992011001

RINGKASAN

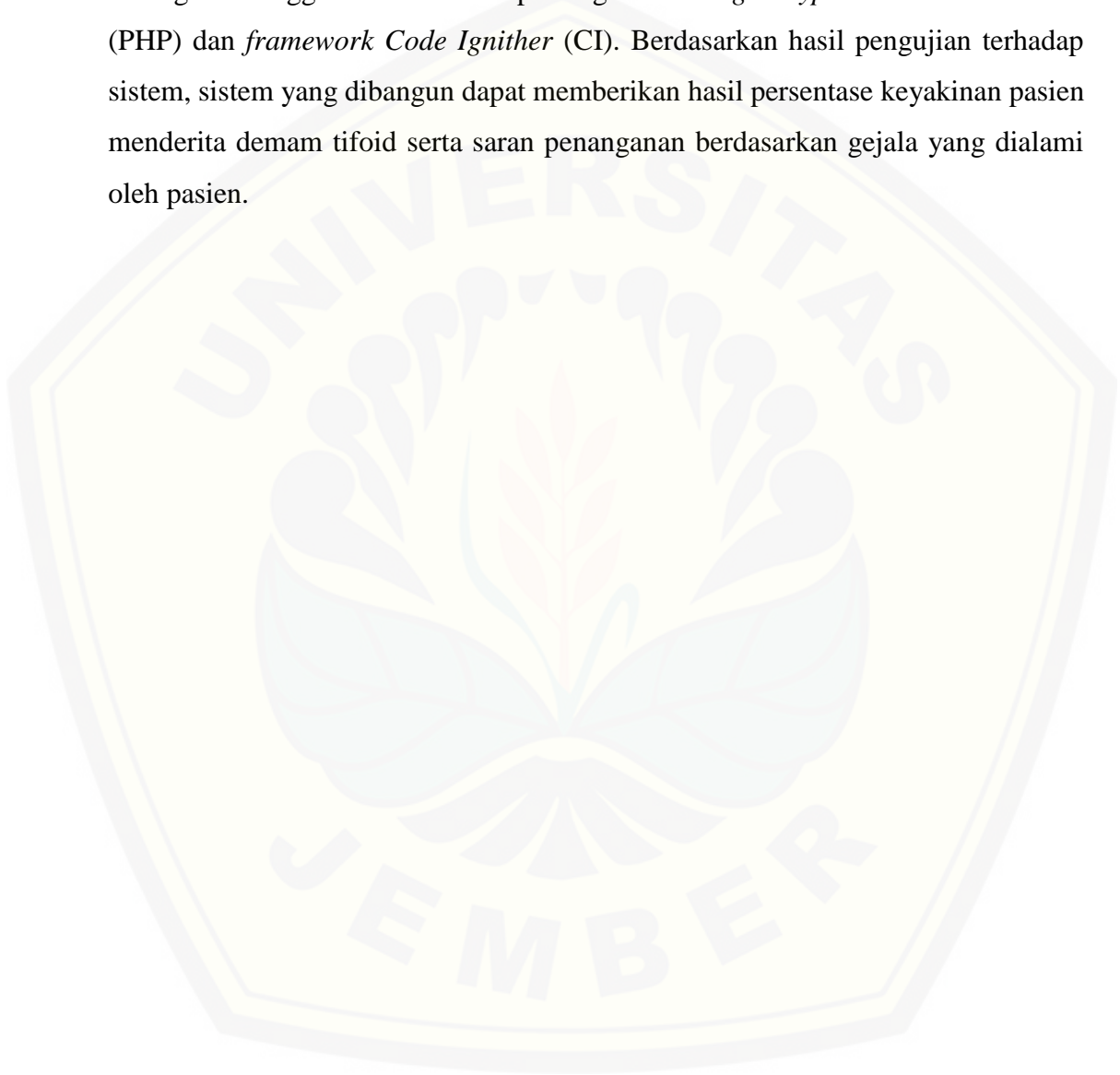
Sistem Pakar Diagnosis Penyakit Demam Tifoid Menggunakan Metode *Certainty Factor* (Studi Kasus RSD Balung); Lintang Harianti, 122410101051; 2016: 164 halaman; Progam Studi Sistem Informasi Universitas Jember.

RSD Balung merupakan rumah sakit umum daerah yang ada di Kecamatan Balung Kabupaten Jember. Pada tahun 2015 penderita penyakit demam tifoid di RSD Balung mengalami peningkatan dari tahun 2014. Minimnya pemahaman masyarakat akan penyakit demam tifoid ini menjadi salah satu penyebab jumlah penderita demam tifoid di RSD Balung semakin meningkat. Kesulitan penanganan pada pasien demam tifoid ialah penanganan pada pasien yang sudah memiliki komplikasi akibat keterlambatan pengobatan. Oleh karena itu, untuk membantu memberikan pengetahuan kepada masyarakat dan membantu mendiagnosis awal agar tidak sampai terjadi komplikasi, maka diperlukan suatu sistem pakar diagnosis penyakit demam tifoid.

Pada penelitian ini dibangun sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor*. sistem yang dibangun berisikan basis pengetahuan tentang penyakit demam tifoid serta sesi konsultasi. Pada basis pengetahuan berisikan tentang gejala, penanganan, serta kaidah-kaidah yang digunakan untuk mendiagnosis demam tifoid. Pada sesi konsultasi berisikan pertanyaan tentang keluhan gejala yang dialami dan akan digunakan pada perhitungan *certainty factor* untuk penentuan persentase keyakinan apakah pengguna benar-benar menderita penyakit demam tifoid atau tidak. Hasil diagnosis yang ditampilkan berupa persentase keyakinan dan saran penanganannya. Sistem pakar ini dibangun berbasis website agar dapat digunakan dengan mudah oleh pasien.

Penelitian dilaksanakan dalam tiga tahap yaitu tahap pengumpulan data, tahap analisis dan tahap pengembangan sistem. Tahap pengumpulan data dilakukan melalui studi literatur dan wawancara kepada narasumber. Tahap analisis dilakukan

dengan menganalisis data yang didapatkan ke dalam metode sistem. Tahap pengembangan sistem dilakukan dengan mengadopsi model *waterfall* yang dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program, pengujian sistem dan perawatan. Sistem ini dibangun menggunakan bahasa pemrograman *Page Hypertext Pre-Processor* (PHP) dan *framework Code Igniter* (CI). Berdasarkan hasil pengujian terhadap sistem, sistem yang dibangun dapat memberikan hasil persentase keyakinan pasien menderita demam tifoid serta saran penanganan berdasarkan gejala yang dialami oleh pasien.



PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan Sistem Pakar Diagnosis Penyakit Demam Tifoid Menggunakan Metode *Certainty Factor* (Studi Kasus RSD Balung). Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

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

1. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Bapak Drs. Antonius C.P., M.App.,Sc.,Ph.D., selaku Dosen Pembimbing Utama dan Bapak Muhamad Arief Hidayat S.Kom, M.Kom. selaku Dosen Pembimbing Pendamping yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Bapak Drs. Antonius C.P., M.App.,Sc.,Ph.D., dan Bapak Dwi retno Istiadi Swasono S.T.,M.Kom. selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Alm. Ayah, Ibu, Tante Emy Kusmiati, kakak tercinta Nona Citra P.S dan adik tersayang Nur Lela F yang telah sangat mendukung baik dalam hal moril dan materil serta selalu memotivasi penulis;
6. RSD Balung selaku objek penelitian yang telah meluangkan waktu yang membantu dalam melancarkan pengerjaan skripsi ini.
7. Sahabat kontrakan Puri Bunga Nirwana yaitu Puput, Seftya, Ara, Gau, Dwi, Jeje, Esa, Delia dan sahabat seperjuangan yaitu Deasy, Qoqom yang selalu menghibur dan memberi dukungan;

8. Keluarga besar Himpunan Mahasiswa Sistem Informasi (HIMASIF) periode 2013-2014 dan 2014-2015;
9. Keluarga besar FORMATION semua mahasiswa angkatan 2012 yang telah menjadi keluarga selama menempuh pendidikan S1;
10. Kelompok KKN 137 Kapongan
11. Ustad Muhammad Hafi yang selalu memberikan dukungan;
12. Direktorat Pembelajaran dan Kemahasiswaan Ditjen Dikti yang telah memberikan beasiswa bidikmisi kepada penulis sehingga dapat menyelesaikan pendidikan S1;
13. Semua pihak yang tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, 22 Juni 2016

Penulis

DAFTAR ISI

	Halaman
SKRIPSI	i
PERSEMBAHAN	ii
MOTTO	iii
PERNYATAAN	iv
SKRIPSI	v
PENGESAHAN PEMBIMBING	vi
PENGESAHAN	vii
RINGKASAN	viii
PRAKATA	x
DAFTAR ISI	xii
DAFTAR TABEL	xvi
DAFTAR GAMBAR	xvii
DAFTAR LAMPIRAN	xix
BAB I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan dan Manfaat	3
1.3.2 Tujuan	3
1.3.2 Manfaat	3
1.4 Batasan Masalah	4
1.5 Sistematika Penulisan	4
BAB 2. TINJAUAN PUSTAKA	6
2.1 Penelitian Terdahulu	6
2.2 Penyakit Demam Tifoid	7
2.3 Metode <i>Certainty Factor</i>	8
2.4 Sistem Pakar	10
2.4.1 Definisi Sistem Pakar	10
2.4.2 Keuntungan Sistem Pakar	12

2.4.3	Kelemahan Sistem Pakar.....	12
BAB 3. METODOLOGI PENELITIAN		13
3.1	Jenis Penelitian	13
3.2	Tempat dan Waktu Penelitian	13
3.3	Metode Pengumpulan Data	14
3.4	Metode Analisis Data	15
3.5	Teknik Pengembangan Sistem	13
3.5.1	Analisis Kebutuhan	14
3.5.2	Desain.....	16
3.5.3	Penulisan Kode Program.....	16
3.5.4	Pengujian.....	17
3.5.5	Pemeliharaan	17
BAB 4. ANALISIS DAN PERANCANGAN SISTEM		18
4.1	Deskripsi Umum Sistem.....	18
4.1.1	SOP (Statement of purpose).....	18
4.1.2	Fungsi sistem.....	19
4.2	Analisis Kebutuhan	19
4.2.1	Kebutuhan Fungsional	20
4.2.2	Kebutuhan Non-Fungsional	20
4.3	Desain Sistem	21
4.3.1.	Bussiness Process.....	21
4.3.2.	<i>Use Case</i> Diagram.....	22
4.3.3.	<i>Use Case</i> Skenario	25
4.3.4.	<i>Activity</i> Diagram.....	31
4.3.5.	<i>Sequence</i> Diagram.....	36
4.3.6.	<i>Class</i> Diagram.....	41
4.3.7.	<i>Entity Relationship</i> Diagram (ERD)	41
4.4	Implementasi.....	43
4.5	Pengujian.....	47
4.5.1.	Metode <i>White Box</i>	48
4.5.2.	Metode <i>Black Box</i>	58

BAB 5. HASIL DAN PEMBAHASAN	59
5.1 Hasil Penerapan Perhitungan Metode <i>Certainty Factor</i>	59
5.1.1 Data Gejala Penyakit Demam Tifoid	59
5.1.2 Data Nilai <i>Certainty Factor</i> Pakar (CF Pakar)	61
5.1.3 Data Nilai <i>Certainty Factor</i> Pengguna (CF User)	61
5.1.4 Data Penanganan	62
5.2 Pengujian Sistem Pakar Diagnosis Penyakit Demam Tifoid	
Menggunakan Metode <i>Certainty Factor</i>	64
5.2.1. Kasus 1	65
5.2.2. Kasus 2	68
5.2.3. Kasus 3	72
5.2.4. Hasil Perbandingan	75
5.3 Hasil Pembuatan Sistem Pakar Diagnosis Penyakit Demam Tifoid	
Menggunakan Metode <i>Certainty Factor</i>	76
5.3.1 Tampilan Awal Sistem	76
5.3.2 Fitur Login	77
5.3.3 Halaman <i>Home Admin</i>	77
5.3.4 Fitur Melihat Data Gejala.....	78
5.3.5 Fitur Menambah Data Gejala	79
5.3.6 Fitur Mengubah Data Gejala	79
5.3.7 Fitur Menghapus Data Gejala	80
5.3.8 Fitur Melihat Data CF	81
5.3.9 Fitur Menambah Data CF	81
5.3.10 Fitur Mengubah Data CF	82
5.3.11 Fitur Menghapus Data CF	83
5.3.12 Fitur Melihat Rekap Data Pasien	83
5.3.13 Fitur Melihat Data Diagnosis	84
5.3.14 Fitur Mendaftar Anggota.....	85
5.3.15 Fitur Melihat Data Pasien Anggota.....	86
5.3.16 Fitur Melihat <i>Record</i> Pemeriksaan.....	87
5.3.17 Fitur Melihat Detail <i>Record</i> Pemeriksaan.....	87

5.3.18	Fitur Memasukkan Pemeriksaan Pasien Lama	88
5.3.19	Fitur Memasukkan Pemeriksaan Pasien Baru	89
5.4	Implementasi Metode <i>Certainty Factor</i> Pada Sistem Pakar Diagnosis Penyakit Demam Tifoid	90
5.5	Pembahasan Pada Sistem Pakar Diagnosis Penyakit Demam Tifoid .	92
5.5.1	Kelebihan Sistem	92
5.5.2	Kelemahan Sistem.....	93
BAB 6. PENUTUP	94
6.1	Kesimpulan	94
6.2	Saran.....	95
DAFTAR PUSTAKA	96
LAMPIRAN	97

DAFTAR TABEL

	Halaman
Tabel 4.1 Deskripsi pembagian aktor sistem	23
Tabel 4.2 Deskripsi use case sistem	23
Tabel 4.3 Skenario memasukkan pemeriksaan pasien baru	26
Tabel 4.4 Kode program function checkbaru()	44
Tabel 4.5 Kode program function tambah_gejala()	45
Tabel 4.6 Kode program function cfcombine().....	46
Tabel 4.7 Kode program function tambah_diagnosis().....	47
Tabel 4.8 Test case function checkbaru()	54
Tabel 4.9 Test case function tambah_gejala().....	55
Tabel 4.10 Test case function cfcombine()	56
Tabel 4.11 Test case function tambah_diagnosis()	57
Tabel 5.1 Data gejala penyakit demam tifoid	60
Tabel 5.2 Data pertanyaan	60
Tabel 5.3 Data nilai certainty factor pakar (CF pakar)	61
Tabel 5.4 Data nilai certainty factor pakar (CF user)	62
Tabel 5.5 Data penentuan penanganan	62
Tabel 5.6 Data penanganan demam tifoid ringan	63
Tabel 5.7 Data penanganan demam tifoid berat	64
Tabel 5.8 Data keluhan pasien pertama	65
Tabel 5.9 Nilai CF user dan nilai CF pakar pasien pertama	65
Tabel 5.10 Data keluhan pasien kedua	68
Tabel 5.11 Nilai CF user dan nilai CF pakar pasien kedua	69
Tabel 5.12 Data keluhan pasien ketiga	72
Tabel 5.13 Nilai CF user dan nilai CF pakar pasien ketiga	72
Tabel 5.14 Perbandingan perhitungan manual dengan sistem	75

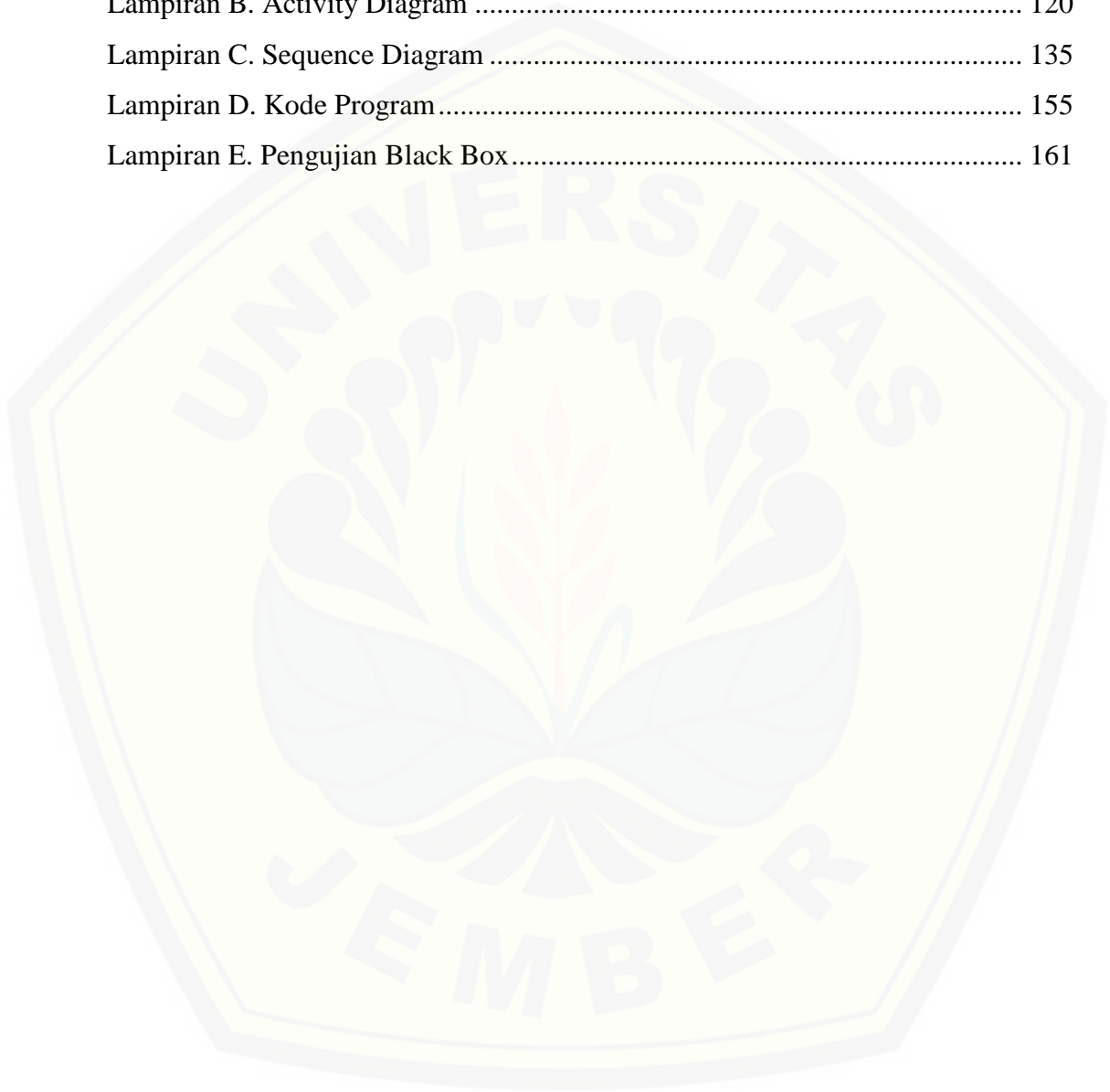
DAFTAR GAMBAR

	Halaman
Gambar 2.1 Tahapan Perhitungan Metode Certainty Factor	9
Gambar 2.2 Struktur Sistem Pakar)	11
Gambar 3.1 Diagram Alir Penerapan Metode Certainty Factor	15
Gambar 3.2 Model Waterfall	14
Gambar 4.1 Bussiness Process Sistem	21
Gambar 4.2 Use Case Diagram	22
Gambar 4.3 Activity Diagram Memasukkan Pemeriksaan Pasien Baru.....	32
Gambar 4.4 Sequence diagram memasukkan pemeriksaan pasien baru.....	37
Gambar 4 5 Class diagram	42
Gambar 4.6 Entity Relationship Diagram	43
Gambar 4.7 Listing program function checkbaru().....	49
Gambar 4 8 Listing program function tambah_gejala().....	50
Gambar 4.9 Listing program function tambah_diagnosis()	50
Gambar 4.10 Diagram alir function checkbaru()	51
Gambar 4.11 Diagram alir function tambah_gejala()	51
Gambar 4.12 Diagram Alir function cfcombine().....	51
Gambar 4.13 Diagram Alir function tambah_diagnosis().....	52
Gambar 4.14 Grafik alir function checkbaru().....	52
Gambar 4.15 Grafik alir function tambah_gejala().....	52
Gambar 4.16 Grafik alir function cfcombine()	53
Gambar 4.17 Grafik alir function tambah_diagnosis()	53
Gambar 5.1 Tampilan awal sistem.....	76
Gambar 5 2 Tampilan halaman login.....	77
Gambar 5.3 Halaman home admin.....	78
Gambar 5.4 Halaman data gejala	78
Gambar 5.5 Halaman form tambah gejala	79
Gambar 5 6 Halaman form edit gejala	80
Gambar 5.7 Fitur menghapus data gejala.....	80

Gambar 5.8 Halaman Data CF	81
Gambar 5.9 Halaman form tambah data CF	82
Gambar 5.10 Form edit data CF	82
Gambar 5.11 Fitur menghapus data CF	83
Gambar 5.12 Halaman rekap data pasien	84
Gambar 5.13 Halaman data record	85
Gambar 5.14 Halaman data detail pemeriksaan	85
Gambar 5.15 Halaman form pendaftaran	86
Gambar 5.16 Halaman dashboard	86
Gambar 5.17 Halaman record pemeriksaan	87
Gambar 5.18 Halaman detail record pemeriksaan	88
Gambar 5.19 Halaman form keluhan	88
Gambar 5.20 Halaman hasil diagnosis	89
Gambar 5.21 Halaman form keluhan	89
Gambar 5.22 Kode program menghitung nilai CF pakar	90
Gambar 5.23 Implementasi kode program menghitung nilai CF[H,E]	90
Gambar 5.24 kode program menghitung nilai CFCombine	91
Gambar 5.25 Kode program penentuan penanganan	91
Gambar 5.26 Kode program query mengambil nilai CFCombine	91
Gambar 5.27 Kode program menghitung persentase keyakinan	92

DAFTAR LAMPIRAN

	Halaman
Lampiran A. Use Case Skenario	97
Lampiran B. Activity Diagram	120
Lampiran C. Sequence Diagram	135
Lampiran D. Kode Program	155
Lampiran E. Pengujian Black Box	161



BAB I. PENDAHULUAN

Bab ini merupakan bab awal dari laporan tugas akhir. Pada bab ini akan dibahas tentang latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

1.1 Latar Belakang

RSD Balung merupakan rumah sakit umum daerah yang ada di Kecamatan Balung Kabupaten Jember. Pada tahun 2015, sebanyak 30.306 pasien dirawat di RSD Balung terdiri atas 4.987 orang rawat inap, 25.319 orang rawat jalan. Dari data tsb, sebanyak 204 pasien merupakan penderita penyakit demam tifoid di RSD Balung yakni mengalami peningkatan 1% dari tahun 2014 dimana sebanyak 201 pasien penderita demam tifoid pada tahun 2014. Pasien demam tifoid tersebut ditangani oleh dokter umum. Kesulitan penanganan pasien demam tifoid ialah penanganan pada pasien yang sudah memiliki komplikasi dikarenakan penyakit yang sudah akut akibat keterlambatan pengobatan, sehingga dokter umum harus mengarahkan pasien untuk diperiksa oleh dokter spesialis.

Minimnya pemahaman masyarakat akan penyakit demam tifoid ini menyebabkan jumlah penderita demam tifoid di RSD Balung semakin meningkat. Kurangnya menjaga kebersihan lingkungan menyebabkan proses penularan penyakit demam tifoid ini menjadi lebih mudah. Selain faktor kebersihan, masyarakat juga kurang memahami akan gejala-gejala yang timbul. Hal itu dikarenakan gejala yang timbul tidak memiliki gejala yang khas sehingga sering disalah artikan sebagai penyakit demam biasa, sehingga enggan berkonsultasi dan tidak segera mendapat penanganan dokter. Akibatnya banyak pasien penderita demam tifoid yang mengalami komplikasi akibat keterlambatan penanganan dirujuk dari puskesmas setempat ke RSD Balung untuk mendapatkan penanganan lebih intensif.

Untuk membantu memberikan pengetahuan kepada masyarakat akan penyakit demam tifoid ini dan juga membantu mendiagnosis awal agar tidak sampai terjadi komplikasi, maka di perlukan suatu sistem pakar untuk mendiagnosis penyakit demam tifoid diderita oleh pasien. Sistem pakar ini bukan untuk menggantikan peran dokter, melainkan untuk membantu memberikan pengetahuan kepada pasien sebelum berkonsultasi kepada dokter, sehingga sebelum berkonsultasi ke dokter, pasien sudah mengetahui penyakit yang diderita serta mampu memberikan penanganan dini.

Sistem pakar yang dibutuhkan yakni sistem pakar diagnosis penyakit demam tifoid dengan menerapkan suatu metode yang digunakan untuk mengetahui hasil diagnosis. Salah satu metode yang dapat diterapkan ialah metode *certainty factor* (CF). Metode *certainty factor* merupakan metode yang mendefinisikan ukuran kepastian terhadap suatu fakta atau aturan, untuk menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi. Alasan penggunaan metode *certainty factor* ini dikarenakan memiliki keunggulan dengan metode inferensi biasa seperti *forward chaining* atau *backward chaining* yakni dapat menggambarkan tingkat keyakinan pakar. Sistem pakar ini dibangun berbasis webiste agar dapat digunakan dengan mudah. Dengan adanya sistem pakar ini diharapkan mampu memberikan pengetahuan kepada masyarakat tentang penyakit demam tifoid.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dipaparkan diatas maka rumusan masalah yang timbul adalah sbb:

- a. Bagaimana penerapan metode *certainty factor* untuk mendiagnosis penyakit demam tifoid?
- b. Bagaimana membangun sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor*?

1.3 Tujuan dan Manfaat

Berikut merupakan tujuan yang ingin dicapai dan manfaat yang ingin diperoleh dalam penelitian ini.

1.3.2 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah:

- a. Mengetahui penerapan metode *certainty factor* untuk mendiagnosis penyakit demam tifoid.
- b. Membangun sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor*.

1.3.2 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

- a. Diharapkan dapat membantu menekan peningkatan penderita demam tifoid setiap tahunnya.
- b. Dengan menggunakan sistem pakar diagnosis penyakit demam tifoid, masyarakat mampu mengetahui penyakit yang diderita serta mampu memberikan penanganan dini sehingga penyakit tidak menjadi semakin parah.
- c. Diharapkan dapat menjadi dasar bagi peneliti pada penelitian selanjutnya untuk pengembangan sistem pakar diagnosis penyakit demam tifoid yang lebih baik.
- d. Dapat meningkatkan keilmuan tentang sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor*, sekaligus sebagai media untuk penyelesaian tugas akhir jenjang S1 pada Program Studi Sistem Informasi, Universitas Jember.

1.4 Batasasan Masalah

Agar tidak terjadi penyimpangan dalam proses penelitian dan pembangunan sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor*, maka ditetapkan beberapa batasan permasalahan. Adapun batasan masalah yang timbul adalah sbb:

- a. Sistem pakar ini hanya digunakan untuk mendiagnosis penyakit *Demam tifoid*.
- b. Sistem pakar diagnosis penyakit demam tifoid ini dikembangkan menggunakan metode *Certainty factor*.
- c. Sistem pakar ini dikembangkan dengan berbasis website.

1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

a. Pendahuluan

Bab kesatu ini memuat uraian tentang latar belakang, perumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan skripsi yang masing-masing tertuang secara eksplisit dalam subbab tersendiri.

b. Tinjauan Pustaka

Bab ini memaparkan tinjauan terhadap hasil-hasil penelitian terdahulu berkaitan dengan masalah yang dibahas, landasan materi dan konsep prediksi, dan kajian teori metode analisis data yang berkaitan dengan masalah dalam penelitian.

c. Metodologi Penelitian

Bab ini menguraikan tentang tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem yang digunakan dalam penelitian.

d. Pengembangan Sistem

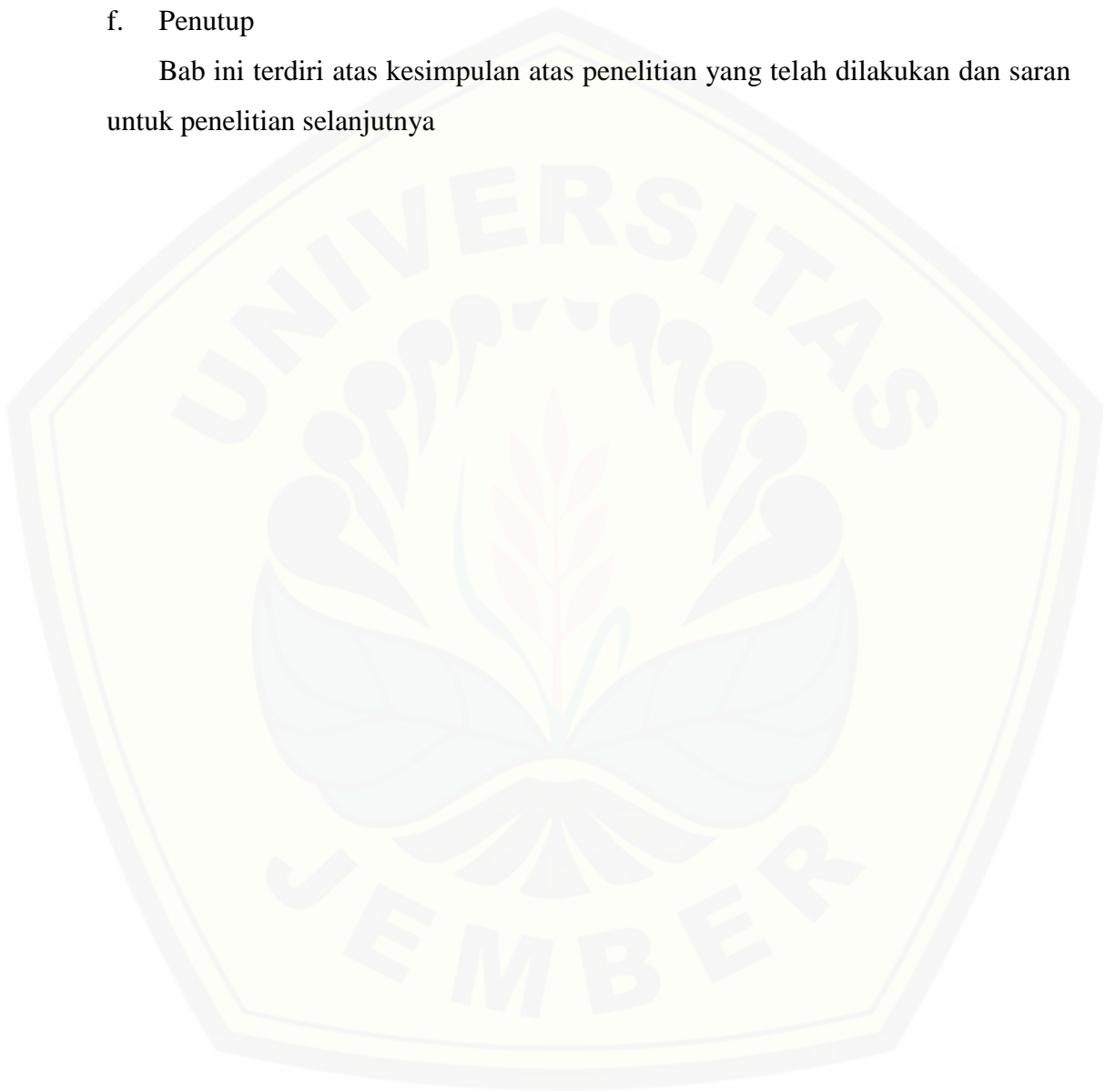
Bab ini berisi uraian tentang langkah-langkah yang ditempuh dalam proses menganalisis dan merancang sistem yang hendak dibangun meliputi desain, pengkodean, dan pengujian sistem.

e. Hasil dan Pembahasan

Bab ini memaparkan secara rinci pemecahan masalah melalui analisis yang disajikan dalam bentuk deskripsi dibantu dengan ilustrasi berupa tabel dan gambar untuk memperjelas hasil penelitian.

f. Penutup

Bab ini terdiri atas kesimpulan atas penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya



BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, dan juga penelitian-penelitian terdahulu.

2.1 Penelitian Terdahulu

Penelitian pertama dilakukan oleh (Wijaya & Prasetyowati, 2012) tentang sistem pakar diagnosis penyakit demam tifoid dan demam berdarah dengan menggunakan metode *forward chaining*. Sistem yang dikembangkan bertujuan menangani kesalahan analisa dokter antara kedua penyakit tersebut saat memeriksa pasien. Kesalahan yang sering terjadi dikarenakan gejala yang dimiliki kedua penyakit tersebut hampir sama. Untuk mengurangi kesalahan diagnosis, akan lebih baik jika diimbangi dengan tingkat keyakinan akan diagnosis suatu penyakit seperti pada metode *certainty factor* sehingga tidak menimbulkan keraguan bagi dokter untuk mengambil keputusan.

Pada penelitian kedua, metode *certainty factor* telah digunakan sebelumnya pada sistem pakar yang dikembangkan oleh (Sari, 2013). Aplikasi sistem pakar yang dikembangkan bertujuan mendiagnosis satu jenis penyakit yakni penyakit demam berdarah. Gejala penyakit demam berdarah direpersentasekan ke dalam bentuk rule agar dapat dimengerti oleh komputer. Terdapat dua nilai *Certainty Factor* (CF) yang dikombinasikan yakni CF dari pakar dan nilai CF dari pengguna yang didapat dari hasil jawaban setelah konsultasi. Penerapan metode *certainty factor* dapat memberikan perhitungan penyelesaian seberapa pasti para pengguna atau pasien menderita penyakit demam berdarah dan mampu menghasilkan diagnosis yang akurat. Namun, pada sistem tersebut perlu dilakukan penambahan data untuk gejala penyakit demam berdarah beserta solusi dan cara pencegahannya, sehingga informasi yang dimiliki oleh sistem semakin banyak.

Penelitian lain juga dilakukan oleh (Sahara, Tanpa Tahun) tentang sistem pakar yang dikembangkan menggunakan metode *Certainty Factor* untuk mendiagnosis penyakit usus. Gejala-gejala dari beberapa jenis penyakit usus direpresentasikan ke dalam rule dengan bobot yang telah ditentukan oleh pakar terhadap gejala-gejala yang mempengaruhi terjadinya suatu penyakit usus diantaranya usus buntu, disentri, sembelit, diare, termasuk penyakit yang sama dengan yang penulis kembangkan yakni demam tifoid. Tahapan pembangunan sistem dimulai dengan mengakuisisi pengetahuan dari dokter atau pakar kemudian membangun basis pengetahuan dan memberikan nilai CF pada setiap gejala yang terkait dengan suatu penyakit usus dalam range nilai 0-1. Dengan memilih gejala-gejala penyakit yang dilihat atau dirasakan maka sistem dapat mendiagnosis penyakit usus dengan menampilkan beberapa penyakit dengan nilai CF terbesar yang diurutkan secara *descending*.

2.2 Penyakit Demam Tifoid

Demam tifoid adalah penyakit infeksi pencernaan yang disebabkan oleh bakteri *Salmonella typhimurium* atau yang lebih banyak dikenal dengan *Salmonella typhi*. Penyakit ini ditularkan melalui konsumsi makanan atau minuman yang terkontaminasi oleh tinja atau urin orang yang terinfeksi. Gejala biasanya muncul 1-3 minggu setelah terkena, dan mungkin ringan atau berat (Mandal, Wilkins, Dunbar, & Mayon-White, 2006).

Gejala demam tifoid yang timbul meliputi demam yang semakin meningkat di minggu pertama hingga minggu ketiga, nyeri kepala, malaise, konstipasi, batuk non produktif, bradikardia relative, apatis, diare, distensi abdomen, bercak-bercak merah didada (rose spot), splenomegalli, delirium, mengantuk, distensi abdomen. Demam tifoid (termasuk para-tifoid) disebabkan oleh kuman *Salmonella typhi*, *S paratyphi A*, *S paratyphi B* dan *S paratyphi C*. Jika penyebabnya adalah *S paratyphi*, gejalanya lebih ringan dibanding dengan yang disebabkan oleh *S typhi* (Inawati, 2009)

Komplikasi sering kali terjadi pada penderita demam tifoid yang sudah cukup akut, jenis komplikasi demam tifoid dibagi menjadi dua yakni komplikasi intestinal dan ekstra-intestinal (Mandal, Wilkins, Dunbar, & Mayon-White, 2006). Komplikasi intestinal terdiri atas pendarahan usus dan perforasi usus. Sedangkan komplikasi ekstra-intestinal terdiri atas komplikasi kardiovaskuler, komplikasi darah, komplikasi paru, komplikasi hepar dan kandung empedu, komplikasi ginjal, komplikasi tulang, komplikasi neuropsikiatrik.

2.3 Metode *Certainty Factor*

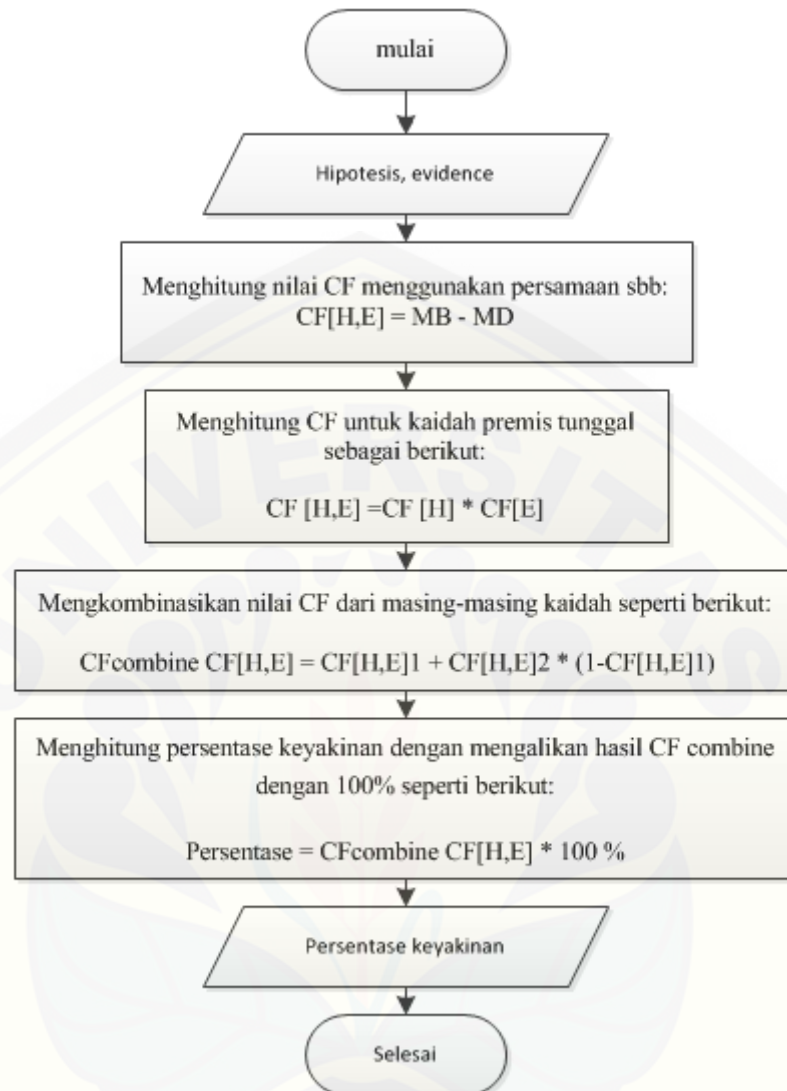
Metode *certainty factor* pertama kali diperkenalkan oleh Shortliffe Buchanan pada tahun 1975 untuk mengakomodasi ketidakpastian pemikiran (inexact reasoning) seorang pakar. Metode *certainty factor* (CF) adalah metode yang mendefinisikan ukuran kepastian terhadap suatu fakta atau aturan, untuk menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi, dengan menggunakan *certainty factor* ini dapat menggambarkan tingkat keyakinan pakar (Kusumadewi, 2003). Rumus dasar *certainty factor* adalah sebagai berikut:

$$CF[H,E] = MB[H,E] - MD[H,E] \quad [2.1]$$

Keterangan:

- CF[H,E] = Faktor kepastian.
- MB[H,E] = Ukuran kepercayaan terhadap hipotesa H, jika diberikan evidence E (antara 0 dan 1) .
- MD[H,E] = Ukuran ketidakpercayaan terhadap evidence H, jika diberikan evidence E (antara 0 dan 1).

Suatu sistem pakar seringkali memiliki kaidah lebih dari satu dan terdiri atas beberapa premis yang dihubungkan dengan AND atau OR. Pengetahuan mengenai premis dapat juga tidak pasti, hal ini dikarenakan besarnya nilai CF yang diberikan oleh pasien saat menjawab pertanyaan sistem atas premis (gejala) yang dialami pasien. Masing-masing kaidah tersebut dihitung nilai *certainty factor*nya sehingga diperoleh nilai *certainty factor* untuk masing-masing aturan. Langkah-langkah perhitungan metode *certainty factor* digambarkan pada Gambar 2.1.



Gambar 2.1 Tahapan Perhitungan Metode *Certainty Factor*

Gambar 2.1 merupakan tahapan perhitungan metode *Certainty Factor*. Dimulai dari menentukan nilai CF dari setiap hipotesa yang dipengaruhi oleh suatu *evidence* kemudian nilai CF tersebut masing masing dikombinasikan. Langkah terakhir menghitung nilai persentase keyakinannya dan keluaran yang dihasilkan adalah persentase keyakinan.

Nilai *certainty factor* yang digunakan pada penelitian ini ada 2, yaitu:

- a. Nilai *certainty factor* kaidah yang nilainya melekat pada suatu kaidah/*rule* tertentu dan besarnya nilai diberikan oleh pakar.
- b. Nilai *certainty factor* yang diberikan oleh pengguna untuk mewakili derajat kepastian/keyakinan atas premis (gejala) yang dialami pengguna.

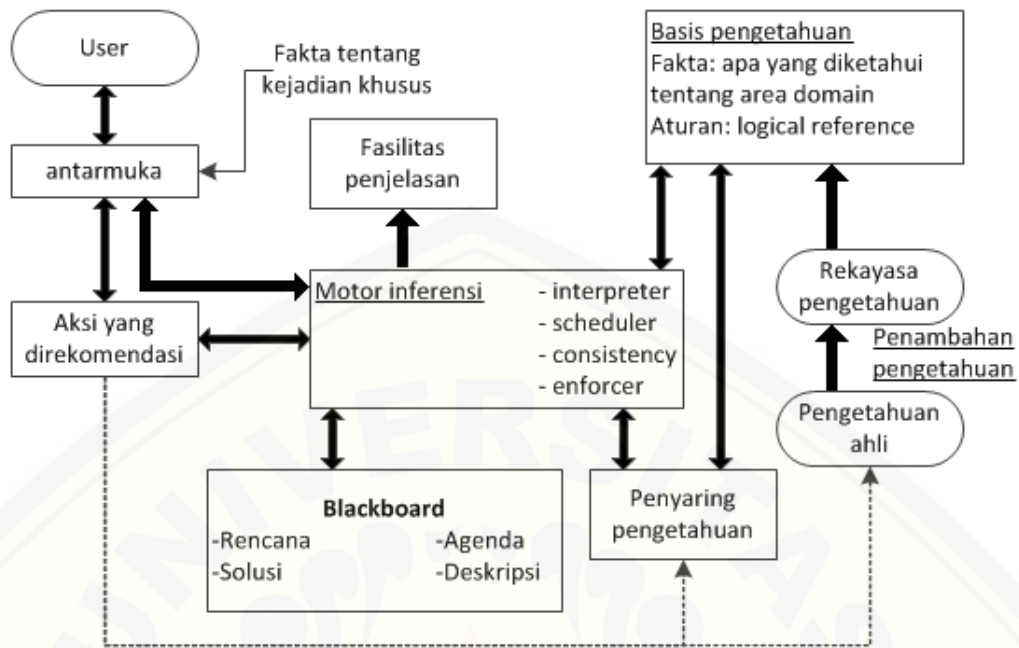
Pada penelitian kedua yang telah yang dipaparkan sebelumnya dijelaskan bahwa pengembangan sistem yang dilakukan hanya mendiagnosis satu penyakit saja yakni demam berdarah dengan metode *certainty factor*. Karakteristik penyakit demam berdarah sendiri memiliki kesamaan dengan penyakit demam tifoid. Sehingga, metode *certainty factor* yang diterapkan pada sistem pakar penyakit demam berdarah dapat juga diterapkan untuk mendiagnosis penyakit demam tifoid.

2.4 Sistem Pakar

2.4.1 Definisi Sistem Pakar

Sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah yang biasa dilakukan oleh para ahli. Definisi lain menurut Durkin, sistem pakar adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan seorang pakar (Kusumadewi, 2003) .

Sistem pakar pertama kali dikembangkan oleh komunitas AI pada pertengahan tahun 1960. Sistem pakar memiliki dua bagian utama. yang pertama yaitu Lingkungan pengembangan (*development environment*), yaitu bagian yang digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar. Yang kedua adalah Lingkungan konsultasi (*consultation environment*), yaitu bagian yang digunakan oleh pengguna yang bukan pakar untuk memperoleh pengetahuan. Gambaran struktur sistem pakar dapat dilihat pada Gambar 2.2.



Gambar 2.2 Struktur Sistem Pakar (Sumber: Kusumadewi, 2003)

Gambar 2.2 merupakan struktur sistem pakar berisi beberapa komponen diantaranya :

- Subsistem penambahan pengetahuan. Bagian ini digunakan untuk memasukkan pengetahuan, mengkonstruksi atau memperluas pengetahuan dalam basis pengetahuan.
- Basis pengetahuan. Berisi pengetahuan-pengetahuan yang dibutuhkan untuk memahami, mempersamaasikan dan menyelesaikan masalah.
- Motor inferensi. Berisi metodologi untuk melakukan penalaran serta digunakan untuk mempersamaasikan konklusi.
- Blackboard. Merupakan area pada memori yang digunakan untuk merekam kejadian yang sedang berlangsung termasuk keputusan sementara.
- Antarmuka. Digunakan untuk media komunikasi antara pengguna dan program
- Subsistem penjelasan. Digunakan untuk menjelaskan kelakuan sistem pakar secara interaktif
- Subsistem penyaring pengetahuan

2.4.2 Keuntungan Sistem Pakar

Secara garis besar, banyak manfaat yang dapat diambil dengan adanya sistem pakar antara lain

- a. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli
- b. Bisa melakukan proses secara berulang secara otomatis
- c. Menyimpan pengetahuan dan keahlian para pakar
- d. Menyimpan pengetahuan dan keahlian para pakar
- e. Meningkatkan output dan produktivitas
- f. Meningkatkan kualitas
- g. Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka)
- h. Mampu beroperasi dalam lingkungan yang berbahaya
- i. Memiliki kemampuan untuk mengakses pengetahuan
- j. Memiliki reliabilitas
- k. Meningkatkan kapabilitas sistem komputer
- l. Memiliki kemampuan untuk bekerja dengan informasi tidak lengkap dan mengandung ketidak pastian
- m. Sebagai media pelengkap dalam pelatihan
- n. Meningkatkan kapabilitas dalam penyelesaian masalah
- o. Menghemat waktu dalam mengambil keputusan

2.4.3 Kelemahan Sistem Pakar

Disamping memiliki beberapa keuntungan, sistem pakar juga memiliki beberapa kelemahan, antara lain:

- a. Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal
- b. Sulit dikembangkan. Hal ini tentu saja erat kaitannya dengan ketersediaannya pakar di bidangnya
- c. Sistem pakar tidak 100% benar

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode yang digunakan dalam merancang dan membangun sistem menggunakan metode pengembangan.

3.1 Jenis Penelitian

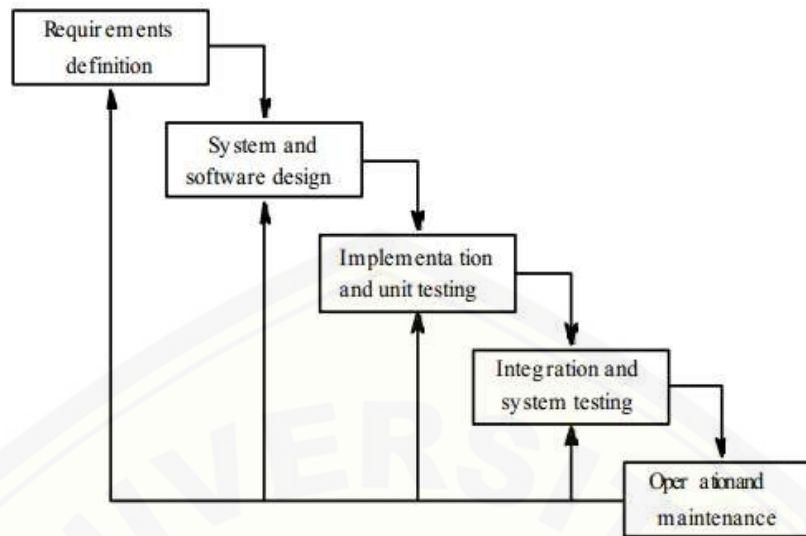
Jenis penelitian yang digunakan pada penelitian ini merupakan jenis penelitian pengembangan, karena tujuan penelitian adalah untuk membangun sebuah sistem pakar. Penelitian pengembangan bertujuan untuk membuat dan mengembangkan suatu produk yang efektif untuk digunakan. Penelitian ini bukan jenis penelitian yang ditunjukkan untuk menemukan teori atau menguji kebenaran dari suatu teori dalam bentuk eksperimentasi..

3.2 Tempat dan Waktu Penelitian

Penelitian dilakukan di RSD Balung Kabupaten Jember yang beralamat di Jalan Rambipuji No 19 Balung. Waktu penelitian dilakukan selama 4 (empat) bulan, dimulai pada bulan Maret 2016 sampai dengan bulan Juni 2016.

3.3 Tahapan Penelitian

Tahapan penelitian pengembangan sistem mengadopsi dari model waterfall dengan tahapan-tahapan seperti pada gambar berikut:



Gambar 3.1 Model *Waterfall* (Sommerville, 2001)

3.5.1 Analisis Kebutuhan

Analisis kebutuhan merupakan tahap untuk mengumpulkan data, informasi, serta mencari kebutuhan fungsional dan non fungsional sistem. Pada tahap ini, peneliti mencari permasalahan yang ada untuk dapat dianalisis kebutuhan yang diperlukan, sebagai solusi dari permasalahan yang muncul. Data-data yang telah didapat kemudian dikelompokkan menjadi kebutuhan fungsional dan non-fungsional sistem. Metode yang digunakan adalah sebagai berikut:

a. Metode Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam mencapai tujuan penelitian. Pengumpulan data pada penelitian ini dilakukan melalui beberapa teknik pengumpulan data yaitu:

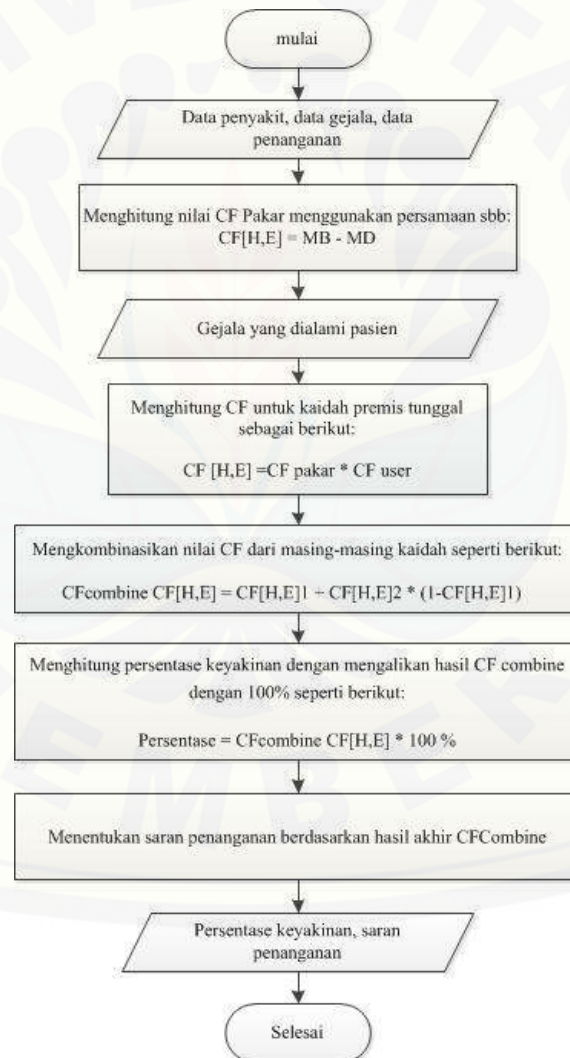
1. Melakukan wawancara kepada dokter RSD Balung untuk mendapatkan data mengenai penyakit demam tifoid.
2. Melakukan studi literature, jurnal, media, maupun internet mengenai penyakit demam tifoid.

Jenis data yang digunakan dalam penelitian adalah data primer dan data sekunder. Data primer merupakan data yang diperoleh melalui metode wawancara.

Data sekunder merupakan data yang diperoleh dari studi literatur. Pengumpulan data dan informasi yang dilakukan antara lain kriteria penyakit, gejala, nilai keyakinan, dan penanganan.

b. Metode Analisis Data

Tahap analisis data dimulai dengan menelaah data secara keseluruhan yang telah diperoleh dari tahap pengumpulan data. Langkah selanjutnya adalah menganalisa data dengan menggunakan metode *certainty factor*. Algoritma metode *certainty factor* pada sistem pakar diagnosis penyakit demam tifoid dapat dilihat pada Gambar 3.1



Gambar 3.2 Diagram Alir Penerapan Metode *Certainty Factor*

3.5.2 Desain

Tahap ini meliputi tahap desain sistem dengan pembuatan diagram menggunakan tools *UML Visual Paradigm*. Diagram-diagram tersebut akan digunakan sebagai acuan pembuatan sistem pada tahap implementasi. Diagram-diagram yang digunakan yang digunakan sbb:

a. *Business Process*

Business process merupakan diagram yang menggambarkan proses dari sebuah sistem yang meliputi *input*, *output*, dan *goal* yang merupakan tujuan dari sebuah sistem yang dibangun.

b. *Use Case Diagram*

Use Case Diagram menggambarkan fungsionalitas dari sebuah sistem dan interaksi antara pengguna dengan sistem untuk melakukan suatu pekerjaan tertentu. *Use Case Diagram* juga menggambarkan hak akses dari pengguna.

c. *Scenario*

Scenario digunakan untuk menjelaskan fitur sistem yang terdapat pada *Use Case Diagram*.

d. *Sequence Diagram*

Sequence Diagram digunakan untuk menunjukkan rangkaian pesan yang dikirim antar object juga interaksi antar object.

e. *Class Diagram*

Class Diagram digunakan untuk menggambarkan struktur statis class dalam sistem.

f. *Entity Realtionship Diagram*

Entity Realtionship Diagram digunakan untuk menunjukkan relasi antar object.

3.5.3 Penulisan Kode Program

Setelah desain sistem telah selesai dilakukan maka selanjutnya pada tahap ini akan dilakukan pembuatan sistem dan implementasiannya. Pembuatan sistem meliputi penulisan kode program, dan pembuatan basis data. Penulisan kode program dilakukan menggunakan tools *Netbeans IDE 8.0* sebagai editor dengan

bahasa pemrograman *PHP* sedangkan untuk manajemen basis data menggunakan tools *Xampp for windows*.

3.5.4 Pengujian

Pengujian digunakan untuk mengetahui sejauh mana sistem ini dapat berjalan. Testing berfungsi untuk mengetahui apakah sistem ini dapat berfungsi dengan baik sesuai dengan yang diharapkan. Serta untuk mengetahui letak kekurangan yang ada pada sistem ini. Terdapat dua metode yang digunakan untuk pengujian ini yakni :

a. *White box testing*

White box testing merupakan cara pengujian dengan melihat modul untuk yang telah dibuat dengan program – program yang ada. Pengujian ini dilakukan oleh (*develeoper*) pembuat program. Jika ada modul yang menghasilkan output yang tidak sesuai, maka baris-baris program, variabel dan parameter yang terlibat pada unit tersebut satu persatu akan di cek dan diperbaiki, kemudian di compile ulang. (Pressman, 2001)

b. *Black box testing*

Berbeda dengan *white box textng*, *Black box testing* merupakan metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja. Metode ini memfokuskan pada keperluan fungsionalitas dari software. (Pressman, 2001)

3.5.5 Pemeliharaan

Pemeliharaan merupakan proses perawatan sistem setelah sistem digunakan oleh pengguna. Pemeliharaan dilakukan dengan mengecek kinerja sistem secara berkala. Pengecekan dilakukan apakah kinerja sistem masih berjalan dengan baik dan memperbaiki apakah terdapat kerusakan atau tidak.

BAB 4. PENGEMBANGAN SISTEM

Bab ini menguraikan mengenai analisis kebutuhan, desain, implementasi, dan pengujian sistem yang digunakan dalam proses pengembangan sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor*. Dimana tahapan analisis hingga pengujian yang dilakukan sesuai dengan metode pengembangan *waterfall*.

4.1 Deskripsi Umum Sistem

Deskripsi umum dari sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* yang dibangun dalam penelitian ini akan dijelaskan lebih detail pada SOP (statement of purpose) sistem dan fungsi sistem.

4.1.1 SOP (Statement of purpose)

Sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* ini berisikan basis pengetahuan tentang penyakit demam tifoid serta sesi konsultasi. Pada basis pengetahuan berisikan tentang gejala, penanganan, serta kaidah-kaidah yang digunakan untuk mendiagnosis demam tifoid. Pada sesi konsultasi berisikan pertanyaan tentang keluhan gejala yang dialami dan akan digunakan untuk perhitungan *certainty factor* untuk penentuan persentase keyakinan apakah pengguna benar-benar menderita penyakit demam tifoid atau tidak. Hasil diagnosis akan ditampilkan berupa persentase keyakinan dan saran penanganannya. Sistem pakar ini dibangun berbasis website agar dapat digunakan dengan mudah oleh pasien.

4.1.2 Fungsi sistem

Fungsi utama dari sistem yang dibangun dalam penelitian ini terletak pada fitur login yang dapat menentukan hak akses dari setiap pengguna dari sistem ini sendiri. Ketika pengguna melakukan login, maka sistem akan melakukan autentifikasi *username* dan *password* dari pengguna. Selanjutnya sistem akan menyajikan tampilan sistem yang sesuai dengan hak akses dari pengguna yang meliputi:

a. Admin

Admin merupakan pengguna sistem yang telah terdaftar, dan ketika login berhasil maka system akan menampilkan halaman dashboard dan halaman-halaman yang sesuai dengan level admin yakni level satu. Admin dapat mengakses halaman data gejala, data CF, dan juga halaman data diagnosis. Admin bertanggung jawab atas data kaidah yang digunakan untuk mendiagnosis penyakit demam tifoid oleh pasien.

b. Anggota

Anggota merupakan pengguna sistem yang telah terdaftar melalui proses pendaftaran sebagai anggota, dan ketika login berhasil maka sistem akan menampilkan halaman dashboard dan halaman-halaman yang sesuai dengan level anggota yakni level dua.. Anggota dapat memasukkan biodata dan melakukan pemeriksaan pasien baik bagi pasien yang telah terdaftar maupun pasien baru. Anggota juga dapat melihat hasil diagnosis, penanganan, serta pemeriksaan setiap pasien yang telah didaftarkan.

4.2 Analisis Kebutuhan

Berdasarkan metode pengembangan sistem model *waterfall*, tahapan awal yang dilakukan adalah tahapan analisis kebutuhan. Tahapan ini dilakukan analisis terhadap kebutuhan-kebutuhan dari sistem yang dibangun, baik berupa kebutuhan fungsional maupun kebutuhan nonfungsional. Hasil analisa tersebut sangat mempengaruhi fungsionalitas sistem yang dibangun untuk dapat digunakan sesuai dengan fungsi dan kebutuhan pengguna.

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem berisi fitur-fitur inti yang harus dipenuhi dalam sistem agar sistem mampu difungsikan sesuai dengan tujuan dan kebutuhan pengguna terhadap sistem itu sendiri. Kebutuhan fungsional dari sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* yaitu:

- a. Sistem mampu mendiagnosis penyakit demam tifoid berdasarkan gejala yang dimasukkan oleh pasien menggunakan metode *certainty factor*
- b. Sistem mampu mengelola data pasien (input, view)
- c. Sistem mampu mengelola data gejala penyakit demam tifoid (input, edit, delete)
- d. Sistem mampu mengelola data kaidah diagnosis (input, edit, delete)
- e. Sistem mampu menampilkan saran penanganan sesuai tingkat penyakit demam tifoid yang diderita oleh pasien
- f. Sistem mampu merekam pemeriksaan pasien baru (input, view)
- g. Sistem mampu merekam pemeriksaan pasien lama (input, view)
- h. Sistem mampu melakukan pendaftaran hak akses sebagai anggota
- i. Sistem mampu menampilkan total pasien

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan fitur-fitur yang dimiliki untuk mendukung sistem dalam memenuhi fungsionalitasnya untuk dapat memenuhi kebutuhan dari pengguna. Kebutuhan non-fungsional sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* yaitu:

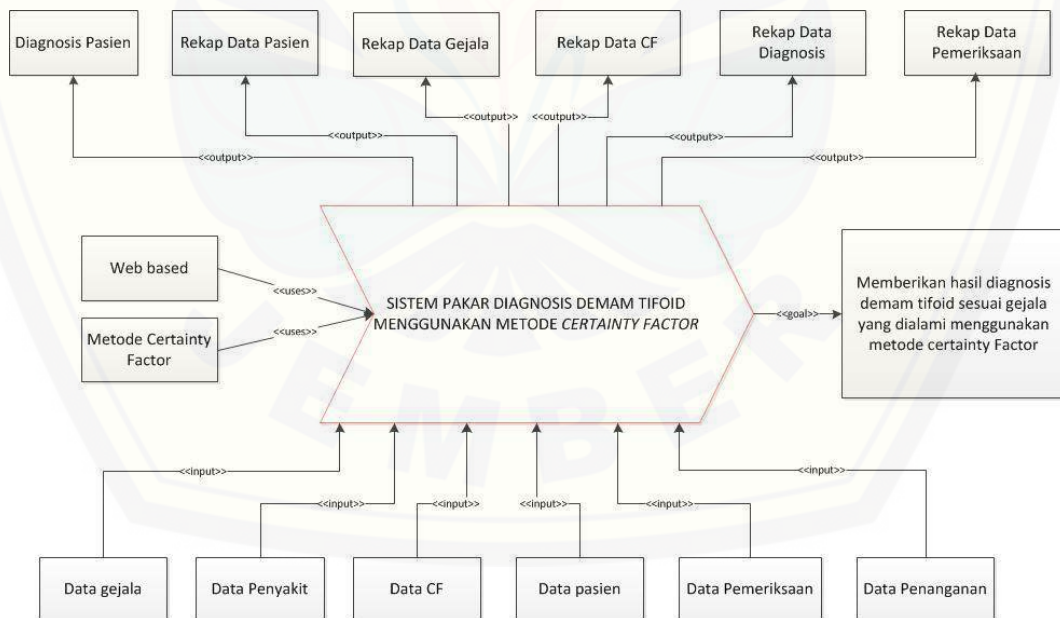
- a. Sistem dapat digunakan oleh banyak pengguna pada saat yang bersamaan dan pada komputer yang berbeda.
- b. Tampilan dan bahasa komunikasi sistem mudah dimengerti oleh pengguna untuk memberikan kenyamanan pemakaian dan memudahkan pengoperasian.
- c. Sistem menggunakan *username* dan *password* untuk autentifikasi akses pengguna terhadap sistem.

4.3 Desain Sistem

Tahapan yang dilakukan setelah melakukan analisis kebutuhan sistem yaitu tahap perencanaan pembangunan sistem yang dapat digambarkan dengan desain sistem. Desain sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* ini meliputi *business process*, *use case diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram*.

4.3.1. Bussiness Process

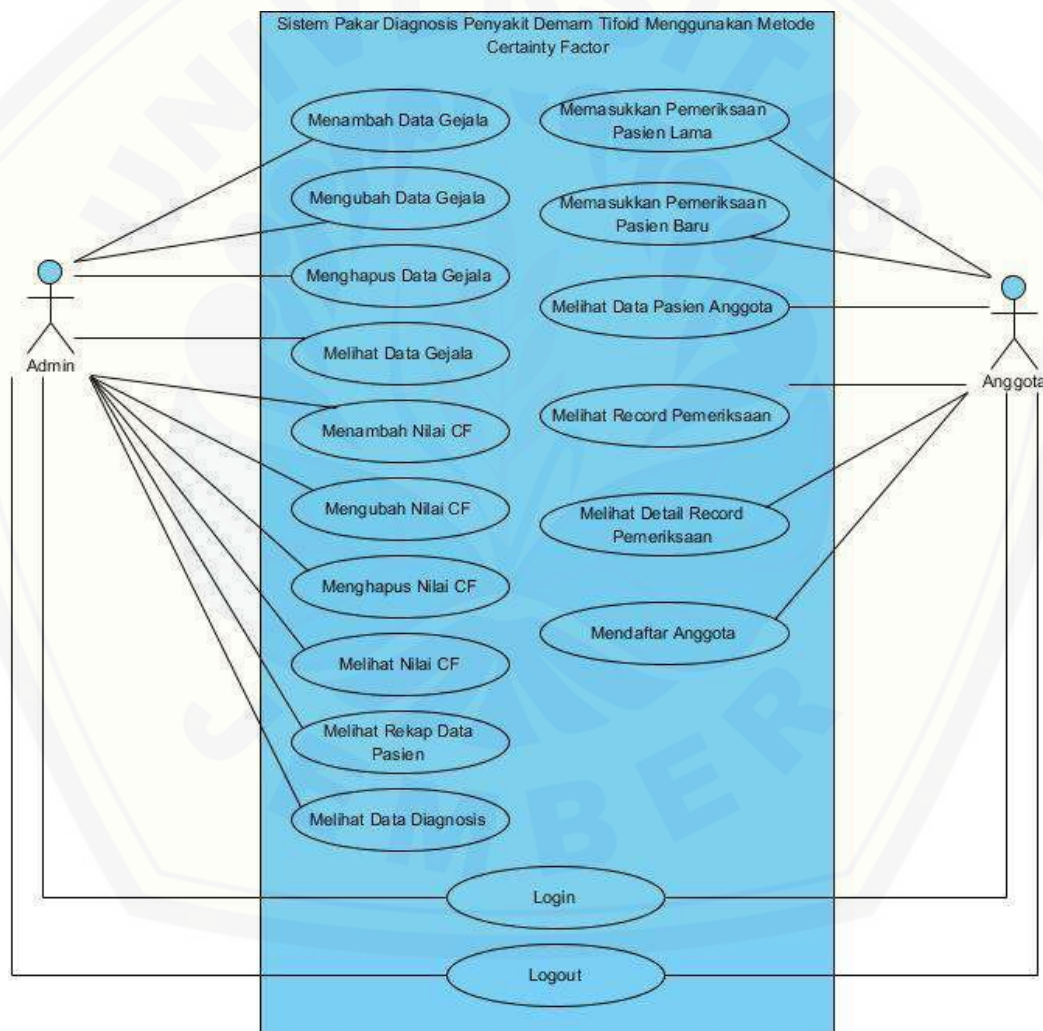
Selain dapat dideskripsikan dalam sebuah SOP (Statement of Purpouse), gambaran umum sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* dapat digambarkan melalui sebuah business process. Seperti yang dapat kita lihat pada Gambar 4.1 menggambarkan data-data yang digunakan sebagai masukan, data keluaran, uses sistem yang dibangun, hingga goal dari dibangunnya sistem sendiri.



Gambar 4.1 *Bussiness Process* Sistem

4.3.2. Use Case Diagram

Use case diagram merupakan pemodelan yang dibuat untuk dapat menggambarkan interaksi antara aktor dengan sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* yang akan dibangun. Melalui use case diagram dapat diketahui interaksi yang dapat dilakukan aktor terhadap sistem sesuai dengan hak akses yang dimiliki oleh masing masing aktor atau pengguna. Pada Gambar 4.2 digambarkan use case diagram yang terdiri atas dua aktor dengan enam belas use case.



Gambar 4.2 Use Case Diagram

Berdasarkan use case diagram pada Gambar 4.2 terdapat dua aktor atau pengguna, yaitu Admin, dan Anggota. Adapun deskripsi dari masing-masing aktor dapat dilihat pada Tabel 4.1.

Tabel 4.1 Deskripsi pembagian aktor sistem

No	Aktor	Deskripsi
1.	Admin	Aktor yang memiliki hak penuh terhadap data gejala, data nilai CF (kaidah), rekap data pasien dan data diagnosis.
2.	Anggota	Aktor yang memiliki hak memasukkan pemeriksaan pasien lama maupun pasien baru, melihat data pasien serta melihat record pemeriksaan setiap pasien yang diperiksa.

Selain memiliki dua aktor, dalam use case diagram juga terdapat enam belas use case. Deskripsi dari use case tersebut dapat dilihat pada Tabel 4.2.

Tabel 4.2 Deskripsi *use case* sistem

No	Use Case	Deskripsi
1.	menambah data gejala	Fitur yang dapat diakses oleh pengguna level satu untuk menambahkan data gejala penyakit demam tifoid ke dalam sistem
2.	mengubah data gejala	Fitur yang dapat diakses oleh pengguna level satu untuk mengubah data gejala penyakit demam tifoid yang telah ada dalam sistem
3.	menghapus data gejala	Fitur yang dapat diakses oleh pengguna level satu untuk menghapus data gejala penyakit demam tifoid dari dalam sistem

Dilanjutkan..

Lanjutan..

No	Use Case	Deskripsi
4.	melihat data gejala	Fitur yang dapat diakses oleh pengguna level satu untuk melihat data gejala penyakit demam tifoid dari dalam sistem
5.	menambah data cf	Fitur yang dapat diakses oleh pengguna level satu untuk menambahkan data kaidah diagnosis ke dalam sistem
6.	mengubah data cf	Fitur yang dapat diakses oleh pengguna level satu untuk mengubah data kaidah diagnosis yang telah ada dalam sistem
7.	menghapus data cf	Fitur yang dapat diakses oleh pengguna level satu untuk menghapus data kaidah diagnosis dari dalam sistem.
8.	melihat data cf	Fitur yang dapat diakses oleh pengguna level satu untuk melihat data kaidah diagnosis dari dalam sistem.
9.	melihat rekap data pasien	Fitur yang dapat diakses oleh pengguna level satu untuk melihat data pasien yang telah dimasukkan ke dalam sistem oleh pengguna level dua.
10.	melihat data diagnosis	Fitur yang dapat diakses oleh pengguna level satu untuk melihat data diagnosis dari pemeriksaan pasien yang telah dimasukkan ke dalam sistem oleh pengguna level dua.
11.	memasukkan pemeriksaan pasien lama	Fitur yang dapat diakses oleh pengguna level dua untuk melanjutkan pemeriksaan pasien lama yang telah terdaftar ke dalam sistem.

Dilanjutkan...

Lanjutan..

No	Use Case	Deskripsi
12.	memasukkan pemeriksaan pasien baru	Fitur yang dapat diakses oleh pengguna level dua untuk melakukan pemeriksaan pasien baru yang belum terdaftar ke dalam sistem.
13.	melihat data pasien anggota	Fitur yang dapat diakses oleh pengguna level dua untuk menampilkan data pasien yang telah didaftarkan ke dalam sistem.
14.	melihat record pemeriksaan	Fitur yang dapat diakses oleh pengguna level dua untuk menampilkan data pemeriksaan pasien yang berisi waktu pemeriksaan dan hasil diagnosis.
15.	melihat detail record pemeriksaan	Fitur yang dapat diakses oleh pengguna level dua untuk menampilkan data detail pemeriksaan pasien pada waktu tertentu
16.	mendaftar anggota	Fitur yang dapat diakses oleh pengguna level dua untuk mendaftarkan diri sebagai anggota dan mendapatkan hak akses level dua
17.	Login	Fitur yang berfungsi melakukan autentifikasi hak akses semua aktor dalam menggunakan sistem
18.	Logout	Fitur yang berfungsi melakukan proses keluar dari sistem bagi semua aktor

4.3.3. Use Case Skenario

Use case skenario adalah dokumentasi terhadap kebutuhan fungsional sistem. *Use case* skenario sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* adalah sebagai berikut:

A. Skenario Memasukkan Pemeriksaan Pasien Baru

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario memasukkan pemeriksaan pasien baru dapat dilihat pada Tabel 4.3.

Tabel 4.3 Skenario memasukkan pemeriksaan pasien baru

ID	USC-12
Name	Memasukkan Pemeriksaan Pasien Baru
Participating Actor	Anggota
Entry Condition	Anggota akan melakukan pemeriksaan untuk pasien baru
Exit Condition	Anggota telah melakukan pemeriksaan untuk pasien baru dan mengetahui hasil diagnosis
SKENARIO UTAMA	
“Memasukkan pemeriksaan pasien baru ”	
Aksi Aktor	Reaksi Sistem
1. Klik menu dashboard	2. Menampilkan data pasien
3. Klik tombol periksa pasien baru	4. Menampilkan halaman form keluhan
5. Mengisi kolom <ol style="list-style-type: none"> a. Nama b. Umur 	
6. Mengisi form konsultasi yang berisi pilihan jawaban pada setiap gejala yang diajukan. Pilihan jawaban yang diajukan meliputi:	

Dilanjutkan...

Lanjutan...

Aksi Aktor	Reaksi Sistem
a. Sangat Yakin b. Yakin c. Cukup Yakin d. Sedikit Yakin e. Tidak Tahu Tidak	
7. Klik tombol check	
	8. Menghitung hasil diagnosis berdasarkan jawaban yang dipilih menggunakan metode <i>certainty factor</i>
	9. Menampilkan halaman hasil diagnosis yang berisi:
	a. Nama Pasien
	b. Persentase
	c. Saran Penanganan

SKENARIO ALTERNATIF
“kolom belum diisi ”

Aksi Aktor	Reaksi Sistem
5. Mengisi form biodata berisi:	
	a. Nama
	b. Umur
6. Mengisi form konsultasi yang berisi pilihan jawaban pada setiap gejala yang diajukan. Pilihan jawaban yang diajukan meliputi:	
	a. Sangat Yakin
	b. Yakin
	c. Cukup Yakin
	d. Sedikit Yakin
	e. Tidak Tahu
	f. Tidak
7. Klik tombol check	

Dilanjutkan...

Lanjutan...

5. Menampilkan pesan dengan menunjuk pada kolom yang belum diisi

SKENARIO ALTERNATIF

“Terdapat jawaban yang belum dipilih”

Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 5. Mengisi form biodata berisi: <ol style="list-style-type: none"> a. Nama b. Umur 6. Mengisi form konsultasi yang berisi pilihan jawaban pada setiap gejala yang diajukan. Pilihan jawaban yang diajukan meliputi: <ol style="list-style-type: none"> a. Sangat Yakin b. Yakin c. Cukup Yakin d. Sedikit Yakin e. Tidak Tahu f. Tidak 7. Klik tombol check 	<ol style="list-style-type: none"> 8. Menampilkan pesan dengan menunjuk pada pertanyaan yang belum diisi

B. Skenario Menambah Data Gejala

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario menambah data gejala dapat dilihat pada Lampiran A (*Usecase* Skenario).

C. Skenario Mengubah Data Gejala

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengubah data gejala dapat dilihat pada Lampiran A (*Usecase* Skenario).

D. Skenario Menghapus Data Gejala

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario menghapus data gejala dapat dilihat pada Lampiran A (*Usecase* Skenario).

E. Skenario Melihat Data Gejala

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat data gejala dapat dilihat pada Lampiran A (*Usecase* Skenario).

F. Skenario Menambah Data CF

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario menambah data CF dapat dilihat pada Lampiran A (*Usecase* Skenario).

G. Skenario Mengubah Data CF

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengubah data CF dapat dilihat pada Lampiran A (*Usecase* Skenario).

H. Skenario Menghapus Data CF

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario menghapus data CF dapat dilihat pada Lampiran A (*Usecase* Skenario).

I. Skenario Melihat Data CF

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat data CF dapat dilihat pada Lampiran A (*Usecase* Skenario).

J. Skenario Melihat Rekap Data Pasien

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat rekap data pasien dapat dilihat pada Lampiran A (*Usecase* Skenario).

K. Skenario Melihat Data Diagnosis

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat data diagnosis dapat dilihat pada Lampiran A (*Usecase* Skenario).

L. Skenario Memasukkan Pemeriksaan Pasien Lama

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario memasukkan pemeriksaan pasien lama dapat dilihat pada Lampiran A (*Usecase* Skenario).

M. Skenario Melihat Data Pasien Anggota

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat data pasien anggota dapat dilihat pada Lampiran A (*Usecase* Skenario).

N. Skenario Melihat Record Pemeriksaan

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat record pemeriksaan dapat dilihat pada Lampiran A (*Usecase* Skenario).

O. Skenario Melihat Detail Record Pemeriksaan

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat detail record pemeriksaan dapat dilihat pada Lampiran A (*Usecase* Skenario).

P. Skenario Mendaftar Anggota

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mendaftar anggota dapat dilihat pada Lampiran A (*Usecase* Skenario).

Q. Skenario *Login*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario *login* dapat dilihat pada Lampiran A (*Usecase* Skenario).

R. Skenario *Logout*

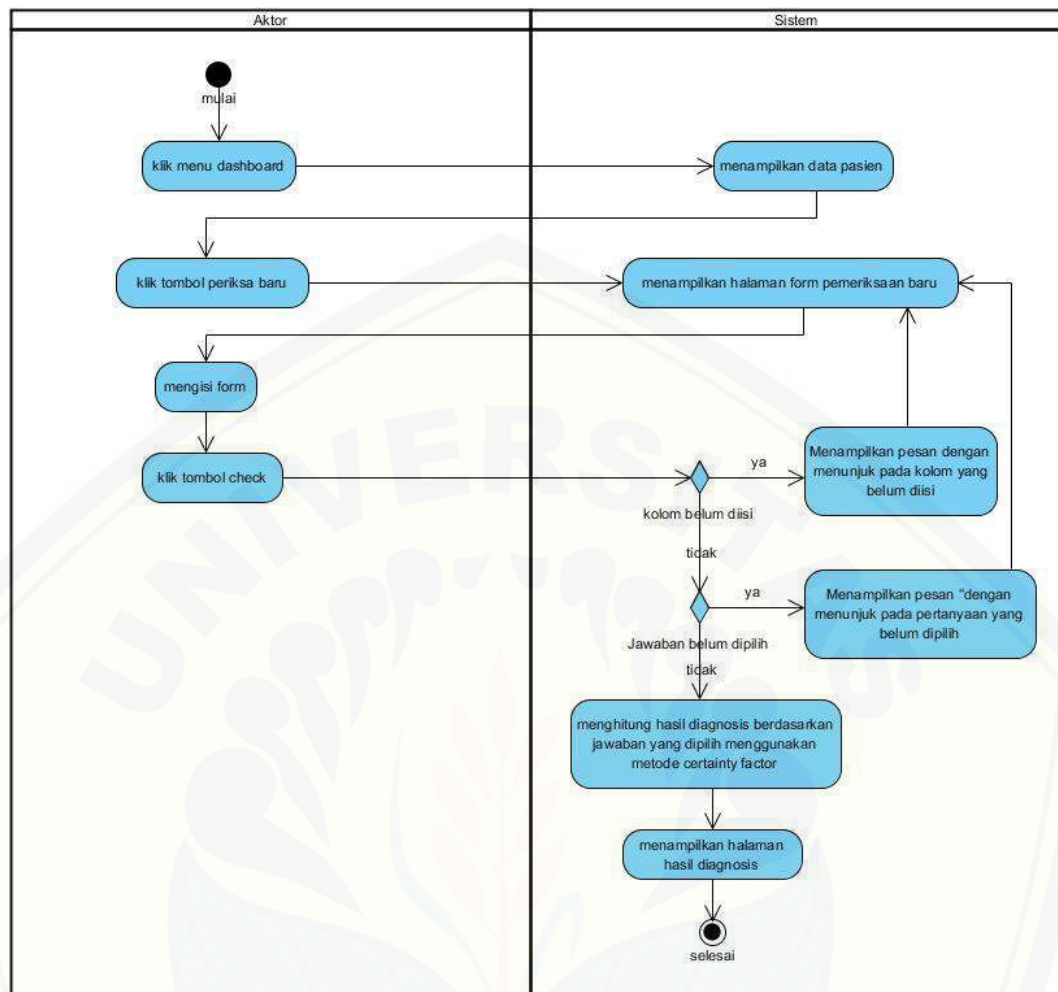
Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario *logout* dapat dilihat pada Lampiran A (*Usecase* Skenario).

4.3.4. *Activity* Diagram

Activity Diagram menggambarkan aliran aktivitas dalam sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* yang akan dibangun. sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* ini memiliki enam belas *activity* diagram yaitu sebagai berikut:

A. *Activity* Diagram Memasukkan Pemeriksaan Pasien Baru

Activity diagram memasukkan pemeriksaan pasien baru dilakukan oleh pengguna level dua yakni anggota. *Activity* diagram memasukkan pemeriksaan pasien baru menjelaskan tentang pengguna level dua dapat memasukkan pemeriksaan pasien yang belum terdaftar dalam sistem. *Activity* diagram memasukkan pemeriksaan pasien baru dapat dilihat pada Gambar 4.3



Gambar 4.3 *Activity Diagram* Memasukkan Pemeriksaan Pasien Baru

B. *Activity Diagram* Menambah Data Gejala

Activity diagram menambah data gejala dapat dilakukan oleh pengguna level satu yakni admin. *Activity diagram* menambah data gejala menjelaskan tentang bagaimana admin dapat menambahkan data gejala pada sistem. *Activity diagram* menambah data gejala dapat dilihat pada Lampiran B (*Activity Diagram*).

C. *Activity Diagram* Mengubah Data Gejala

Activity diagram mengubah data gejala dapat dilakukan oleh pengguna level satu yakni admin. *Activity diagram* mengubah data gejala menjelaskan tentang

bagaimana admin dapat menngubah data gejala pada sistem. *Activity diagram* Mengubah Data Gejala dapat dilihat pada Lampiran B (*Activity Diagram*).

D. *Activity Diagram* Menghapus Data Gejala

Activity diagram menghapus data gejala dapat dilakukan oleh pengguna level satu yakni admin. *Activity diagram* menghapus data gejala menjelaskan tentang bagaimana admin dapat menghapus data gejala pada sistem. *Activity diagram* menghapus data gejala dapat dilihat pada Lampiran B (*Activity Diagram*).

E. *Activity Diagram* Melihat Data Gejala

Activity diagram melihat data gejala dapat dilakukan oleh pengguna level satu yakni admin. *Activity diagram* melihat data gejala menjelaskan tentang bagaimana admin dapat melihat data gejala pada sistem. *Activity diagram* melihat data gejala dapat dilihat pada Lampiran B (*Activity Diagram*).

F. *Activity Diagram* Menambah Data CF

Activity diagram menambah data CF dapat dilakukan oleh pengguna level satu yakni admin. *Activity diagram* menambah data CF menjelaskan tentang bagaimana admin dapat menambah kaidah diagnosis pada sistem. *Activity diagram* menambah data CF dapat dilihat pada Lampiran B (*Activity Diagram*).

G. *Activity Diagram* Mengubah Data CF

Activity diagram mengubah data CF dapat dilakukan oleh pengguna level satu yakni admin. *Activity diagram* mengubah data CF menjelaskan tentang bagaimana admin dapat mengubah kaidah diagnosis pada sistem. *Activity diagram* mengubah data CF dapat dilihat pada Lampiran B (*Activity Diagram*).

H. *Activity Diagram* Menghapus Data CF

Activity diagram menghapus data CF dapat dilakukan oleh pengguna level satu yakni admin. *Activity diagram* menghapus data CF menjelaskan tentang bagaimana admin dapat menghapus kaidah diagnosis pada sistem. *Activity diagram* menghapus data CF dapat dilihat pada Lampiran B (*Activity Diagram*).

I. *Activity Diagram* Melihat Data CF

Activity diagram melihat data CF dapat dilakukan oleh pengguna level satu yakni admin. *Activity diagram* melihat data CF menjelaskan tentang bagaimana admin dapat melihat kaidah diagnosis pada sistem. *Activity diagram* melihat data CF dapat dilihat pada Lampiran B (*Activity Diagram*).

J. *Activity Diagram* Melihat Rekap Data Pasien

Activity diagram melihat rekap data pasien dapat dilakukan oleh pengguna level satu yakni admin. *Activity diagram* melihat rekap data pasien menjelaskan tentang bagaimana admin dapat menampilkan semua data pasien. *Activity diagram* melihat rekap data pasien dapat dilihat pada Lampiran B (*Activity Diagram*).

K. *Activity Diagram* Melihat Data Diagnosis

Activity diagram melihat data diagnosis dapat dilakukan oleh pengguna level satu yakni admin. *Activity diagram* melihat data diagnosis menjelaskan tentang bagaimana admin dapat menampilkan semua data diagnosis dari pemeriksaan pasien. *Activity diagram* melihat melihat data diagnosis dapat dilihat pada Lampiran B (*Activity Diagram*).

L. *Activity Diagram* Memasukkan Pemeriksaan Pasien Lama

Activity diagram memasukkan pemeriksaan pasien lama dapat dilakukan oleh pengguna level dua yakni anggota. *Activity diagram* memasukkan pemeriksaan pasien lama menjelaskan tentang bagaimana anggota dapat melanjutkan pemeriksaan pasien yang telah terdaftar dalam sistem. *Activity diagram* Memasukkan Pemeriksaan Pasien Lama dapat dilihat pada Lampiran B (*Activity Diagram*).

M. *Activity Diagram* Melihat Data Pasien Anggota

Activity diagram melihat data pasien anggota dapat dilakukan oleh pengguna level dua yakni anggota. *Activity diagram* melihat data pasien anggota menjelaskan tentang bagaimana anggota dapat melihat data pasien yang telah didaftarkan

kedalam sistem. *Activity* diagram Memasukkan Pemeriksaan Pasien Lama dapat dilihat pada Lampiran B (*Activity* Diagram).

N. *Activity* Diagram Melihat Record Pemeriksaan

Activity diagram melihat record pemeriksaan dapat dilakukan oleh pengguna level dua yakni anggota. *Activity* diagram melihat record pemeriksaan menjelaskan tentang bagaimana anggota dapat menampilkan data pemeriksaan pasien yang berisi waktu pemeriksaan dan hasil diagnosis. *Activity* diagram melihat record pemeriksaan dapat dilihat pada Lampiran B (*Activity* Diagram).

O. *Activity* Diagram Melihat Detail Record Pemeriksaan

Activity diagram melihat detail record pemeriksaan dapat dilakukan oleh pengguna level dua yakni anggota. *Activity* diagram melihat detail record pemeriksaan menjelaskan tentang bagaimana anggota dapat menampilkan data detail pemeriksaan pasien pada waktu tertentu. *Activity* diagram melihat detail record pemeriksaan dapat dilihat pada Lampiran B (*Activity* Diagram).

P. *Activity* Diagram Mendaftar Anggota

Activity diagram mendaftar anggota dapat dilakukan oleh pengguna level dua yakni anggota. *Activity* diagram mendaftar anggota menjelaskan tentang bagaimana anggota dapat mendaftarkan diri sebagai anggota dan mendapatkan hak akses level dua. *Activity* diagram mendaftar anggota dapat dilihat pada Lampiran B (*Activity* Diagram).

Q. *Activity* Diagram *Login*

Activity diagram *login* menjelaskan tentang bagaimana sistem dapat menjalankan fungsi melakukan autentifikasi hak akses semua aktor dalam menggunakan sistem. *Activity* diagram *login* dapat dilihat pada Lampiran B (*Activity* Diagram).

R. *Activity Diagram Logout*

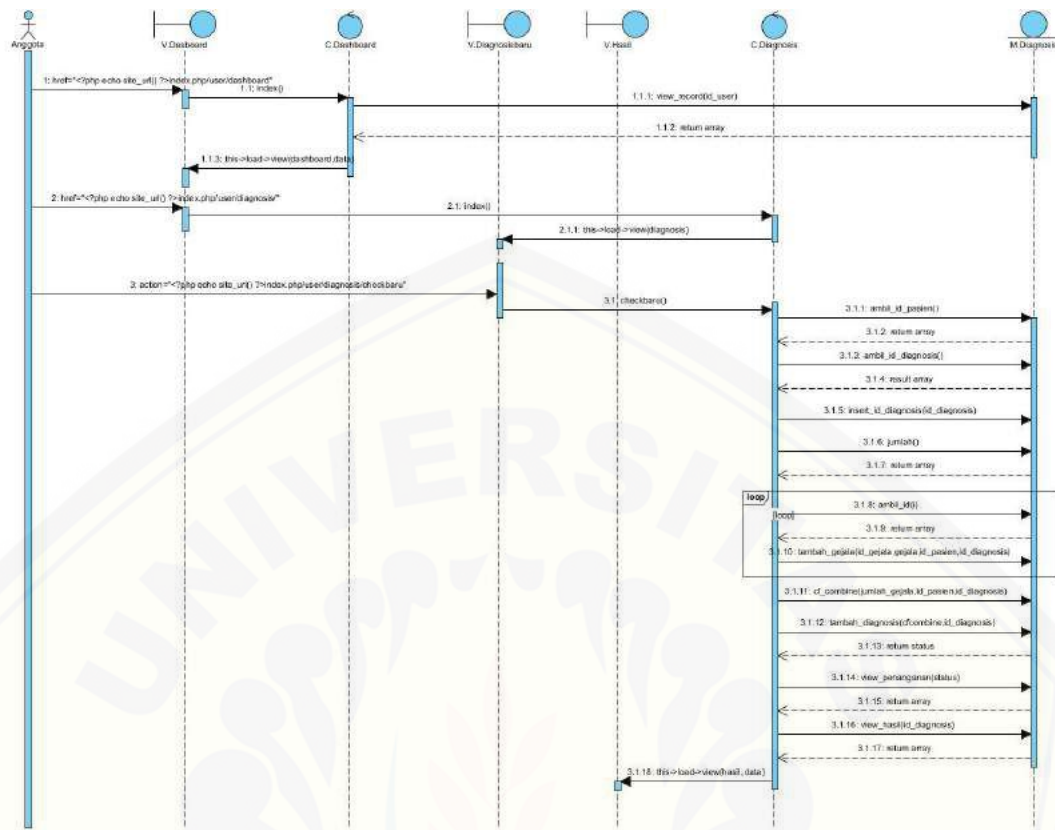
Activity diagram logout menjelaskan tentang bagaimana sistem dapat menjalankan fungsi melakukan proses keluar dari sistem. *Activity diagram logout* dapat dilihat pada Lampiran B (*Activity Diagram*).

4.3.5. *Sequence Diagram*

Sequence diagram adalah dokumentasi suatu diagram terurut yang menampilkan interaksi - interaksi antar objek di dalam sistem. *Sequence diagram* digunakan untuk menggambarkan skenario dan memodelkan aliran logika dalam sistem dengan cara *visual*. *Sequence diagram* dari sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* adalah sebagai berikut:

A. *Sequence Diagram Memasukkan Pemeriksaan Pasien Baru*

Penggambaran *sequence diagram* memasukkan pemeriksaan pasien baru digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence diagram* Memasukkan Pemeriksaan Pasien Baru adalah *view dashboard*, *view diagnosisbaru*, *view hasil*, *controller c_dashboard*, *controller c.diagnosis*, dan *model m_diagnosis*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil *method* yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil *method* pada *class* yang berhubungan baik dengan *class model* jika terdapat *query* yang diambil maupun *class view* jika akan menampilkan sebuah *view* selanjutnya. *Sequence diagram* memasukkan pemeriksaan pasien baru lebih lengkap dijelaskan pada Gambar 4.4



Gambar 4.4 *Sequence diagram* memasukkan pemeriksaan pasien baru

B. *Sequence Diagram* Menambah Data Gejala

Penggambaran *sequence diagram* menambah data gejala digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menambah data gejala. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence diagram* menambah data gejala lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

C. *Sequence Diagram* Mengubah Data Gejala

Penggambaran *sequence diagram* mengubah data gejala digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menambah data gejala. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence diagram* mengubah data gejala lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

D. *Sequence* Diagram Menghapus Data Gejala

Penggambaran *sequence* diagram menghapus data gejala digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menghapus data gejala. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengubah data gejala lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

E. *Sequence* Diagram Melihat Data Gejala

Penggambaran *sequence* diagram melihat data gejala digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melihat data gejala. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram melihat data gejala lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

F. *Sequence* Diagram Menambah Data CF

Penggambaran *sequence* diagram mengubah data CF digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menghapus data CF. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengubah data CF lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

G. *Sequence* Diagram Mengubah Data CF

Penggambaran *sequence* diagram mengubah data CF digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menghapus data CF. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengubah data CF lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

H. *Sequence* Diagram Menghapus Data CF

Penggambaran *sequence* diagram menghapus data CF digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menghapus data

CF. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram menghapus data CF lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

I. *Sequence* Diagram Melihat Data CF

Penggambaran *sequence* diagram melihat data CF digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melihat data CF. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram menghapus data CF lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

J. *Sequence* Diagram Melihat Rekap Data Pasien

Penggambaran *sequence* diagram melihat rekap data pasien digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melihat rekap data pasien. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram melihat rekap data pasien lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

K. *Sequence* Diagram Melihat Data Diagnosis

Penggambaran *sequence* diagram melihat data diagnosis digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melihat data diagnosis. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram melihat data diagnosis lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

L. *Sequence* Diagram Memasukkan Pemeriksaan Pasien Lama

Penggambaran *sequence* diagram memasukkan pemeriksaan pasien lama digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* memasukkan pemeriksaan pasien lama. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram memasukkan pemeriksaan pasien lama lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

M. *Sequence Diagram* Melihat Data Pasien Anggota

Penggambaran *sequence diagram* melihat data pasien anggota digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melihat data pasien anggota. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence diagram* melihat data pasien anggota lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

N. *Sequence Diagram* Melihat Record Pemeriksaan

Penggambaran *sequence diagram* melihat record pemeriksaan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melihat record pemeriksaan. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence diagram* melihat record pemeriksaan lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

O. *Sequence Diagram* Melihat Detail Record Pemeriksaan

Penggambaran *sequence diagram* melihat detail record pemeriksaan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melihat detail record pemeriksaan. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence diagram* melihat record pemeriksaan lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

P. *Sequence Diagram* Mendaftar Anggota

Penggambaran *sequence diagram* mendaftar anggota digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* mendaftar anggota. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence diagram* mendaftar anggota lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

Q. *Sequence Diagram Login*

Penggambaran *sequence diagram login* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case login*. Masing - masing *class* akan

ditampilkan secara *visual* dengan gambar. *Sequence* diagram *login* lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

R. *Sequence* Diagram *Logout*

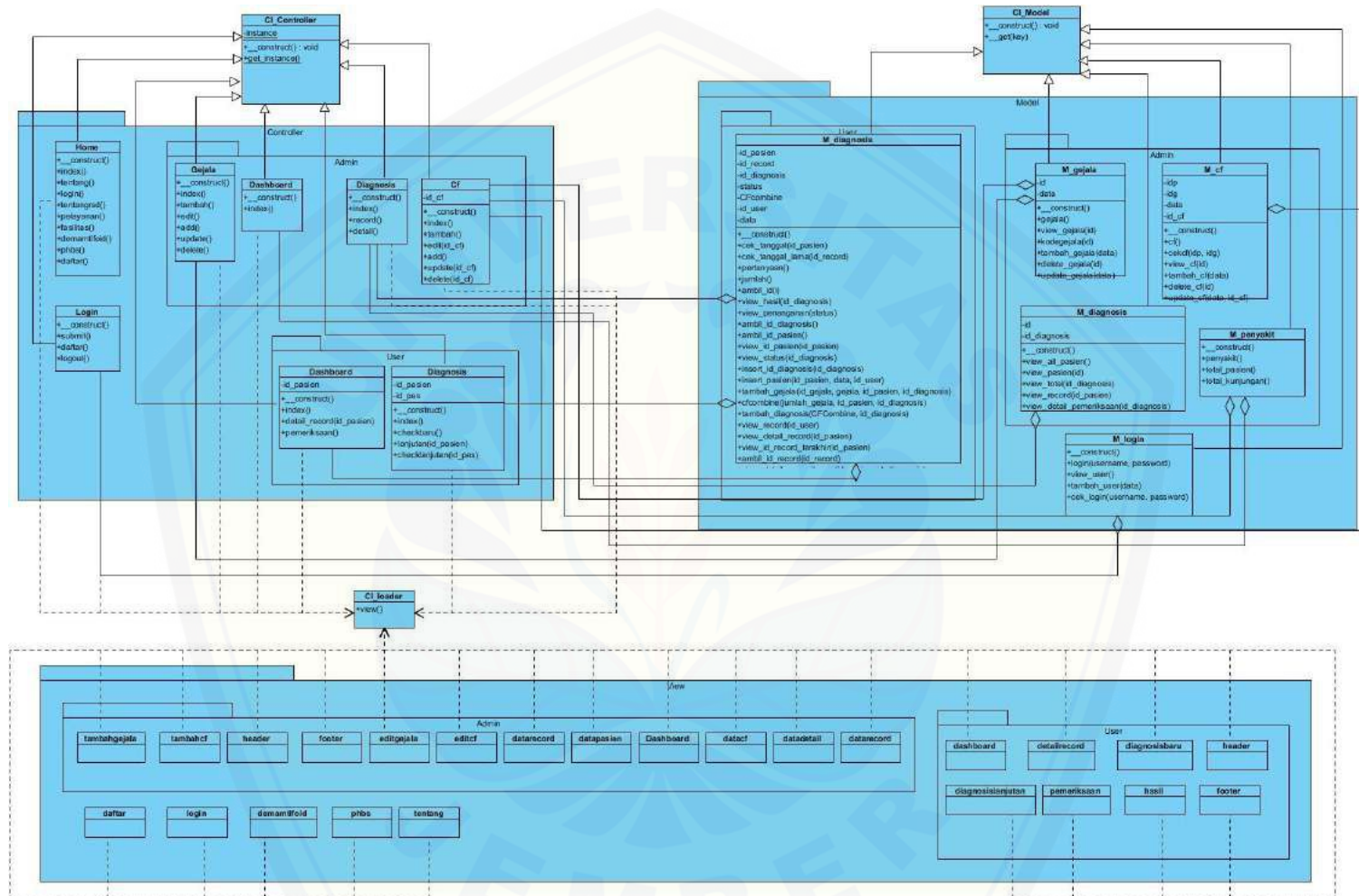
Penggambaran *sequence* diagram *logout* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* *logout*. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram *logout* lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

4.3.6. *Class* Diagram

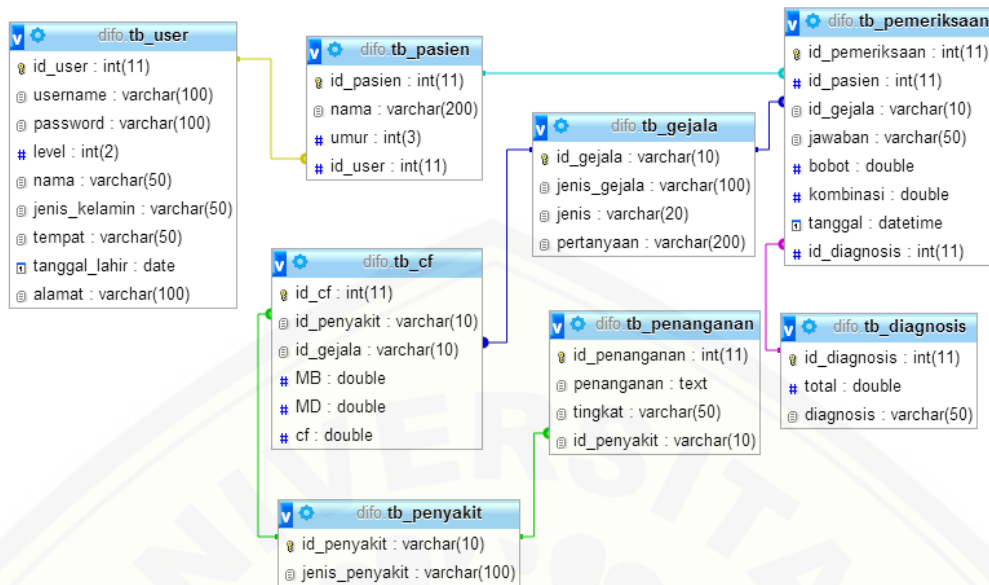
Class diagram menggambarkan hubungan antara kelas yang digunakan untuk membangun suatu sistem. Dalam paradigma OOP (*Object Oriented Programming*) terdapat 3 jenis kelas yaitu *model*, *view* dan *controller*. Berdasarkan *sequence* diagram yang telah dibangun, *class* diagram sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* terdiri dari 6 kelas *model*, 24 kelas *view* dan 8 kelas *controller*. *Class* diagram sistem dapat dilihat pada Gambar 4.5.

4.3.7. *Entity Relationship* Diagram (ERD)

Entity Relationship Diagram (ERD) sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* merupakan gambaran komponen dan struktur *database* yang digunakan dalam pembuatan sistem. ERD yang diimplementasikan pada sistem ini terdiri dari 8 entitas yang dapat dilihat pada Gambar 4.6.



Gambar 4 5 Class diagram



Gambar 4.6 Entity Relationship Diagram

4.4 Implementasi

Desain yang telah dibuat akan diimplementasikan ke dalam kode program. Beberapa hal yang dilakukan dalam tahap implementasi antara lain:

- Penulisan kode program (*coding*) menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor (PHP)* dengan bantuan *framework Code Igniter (CI)*.
- Manajemen basisdata menggunakan *DBMS MySQL*.

Kode program perhitungan metode *certainty factor* terdapat di *class* diagnosis pada *package controllers/user* dan pada *class* m_diagnosis pada *package models/user*. Penulisan kode program perhitungan metode tersebut dapat dilihat pada Tabel 4.4, Tabel 4.5, Tabel 4.6 dan Tabel 4.7.

Tabel 4.4 Kode program *function* checkbaru()

No	Kode
41.	public function checkbaru() {
42	\$ <i>id_user</i> = \$this->session->userdata('id_user');
43	\$ <i>data</i> ['nama'] = \$this->input->post('nama');
44	\$ <i>data</i> ['umur'] = \$this->input->post('umur');
45	
46	\$ <i>id_pasien</i> = \$this->m_diagnosis->ambil_id_pasien();
47	\$this->m_diagnosis->insert_pasien(\$ <i>id_pasien</i> , \$ <i>data</i> , \$ <i>id_user</i>);
48	
49	\$ <i>id_diagnosis</i> = \$this->m_diagnosis->ambil_id_diagnosis();
50	\$this->m_diagnosis->insert_id_diagnosis(\$ <i>id_diagnosis</i>);
51	
52	\$ <i>jumlah</i> = \$this->m_diagnosis->jumlah();
53	foreach (\$ <i>jumlah</i> ->result_array() as \$ <i>row</i>) {
54	\$ <i>jumlah_gejala</i> = \$ <i>row</i> ['jumlah'];
55	}
56	
57	for (\$ <i>i</i> =0; \$ <i>i</i> < \$ <i>jumlah_gejala</i> ; \$ <i>i</i> ++) {
58	\$ <i>sambil2</i> = \$this->m_diagnosis->ambil_id(\$ <i>i</i>);
59	foreach (\$ <i>sambil2</i> ->result_array() as \$ <i>p</i>) {
60	\$ <i>gejala</i> = \$this->input->post(\$ <i>p</i> ['id_gejala']);
61	\$this->m_diagnosis->tambah_gejala(\$ <i>p</i> ['id_gejala'], \$ <i>gejala</i> , \$ <i>id_pasien</i> , \$ <i>id_diagnosis</i>);
62	}
63	}
64	
65	\$ <i>CFCombine</i> = \$this->m_diagnosis->cfcombine(\$ <i>jumlah_gejala</i> , \$ <i>id_pasien</i> , \$ <i>id_diagnosis</i>);
66	\$ <i>status</i> = \$this->m_diagnosis->tambah_diagnosis(\$ <i>CFCombine</i> , \$ <i>id_diagnosis</i>);
67	\$ <i>pasien</i> ['penanganan'] = \$this->m_diagnosis->view_penanganan(\$ <i>status</i>);
68	\$ <i>pasien</i> ['pasien'] = \$this->m_diagnosis->view_hasil(\$ <i>id_diagnosis</i>);
69	\$ <i>data</i> ['username'] = \$this->session->userdata('username');
70	\$this->load->view('user/header',\$ <i>data</i>);
71	\$this->load->view('user/hasil', \$ <i>pasien</i>);
72	\$this->load->view('user/footer');
73	
74	}

Pada Tabel 4.4 merupakan kode program *function* checkbaru() yang berfungsi untuk melakukan pemeriksaan pasien baru yang belum terdaftar dalam sistem untuk mengetahui hasil diagnosis, persentase dan saran penanganan. Tahapan menyimpan perhitungan metode *certainty factor* pada *function* checkbaru() terdapat pada baris 57-62. Pada baris 57-62 merupakan perulangan untuk memanggil nilai CF pada setiap gejala yang ada dan menjalankan *function* tambah_gejala() pada *class* m_diagnosis yang akan dijelaskan pada Tabel 4.5 sesuai dengan banyaknya gejala, sedangkan untuk menghitung hasil *CFcombine* terdapat pada *class* m_diagnosis.

Hasil perhitungan dari *CFcombine* yang terdapat pada class *m_diagnosis* di simpan ke dalam variabel *\$CFCombine* seperti pada baris 65. Setelah nilai *CFcombine* diketahui, maka *function* akan menyimpan diagnosis pada class *m_diagnosis* dengan mengambil nilai balik berupa status penanganan dan menyimpannya pada variabel *\$status* seperti pada baris 66. Selanjutnya *function* akan menampilkan penanganan berdasarkan status penanganan, data pasien dan persentase keyakinan seperti pada baris 67-73.

Tahap perhitungan dimulai dari mencari nilai *CF[H,E]* pergejala. Kode program untuk menghitung nilai *CF[H,E]* yang terdapat pada *function* *tambah_gejala()* dapat dilihat pada Tabel 4.5.

Tabel 4.5 Kode program *function* *tambah_gejala()*

No	Kode
80.	<code>public function tambah_gejala(\$id_gejala,\$gejala, \$id_pasien, \$id_diagnosis) {</code>
81.	
82.	<code>for (\$i = 0; \$i < count(\$gejala); \$i++) {</code>
83.	<code> \$jawaban = "";</code>
84.	<code> if(\$gejala[\$i] == 0){</code>
85.	<code> \$jawaban = "Tidak";</code>
86.	<code> } else if (\$gejala[\$i] == 0.2){</code>
87.	<code> \$jawaban = "Tidak Tahu";</code>
88.	<code> } else if (\$gejala[\$i] == 0.4){</code>
89.	<code> \$jawaban = "Sedikit Yakin";</code>
90.	<code> } else if (\$gejala[\$i] == 0.6){</code>
91.	<code> \$jawaban = "Cukup Yakin";</code>
92.	<code> } else if (\$gejala[\$i] == 0.8){</code>
93.	<code> \$jawaban = "Yakin";</code>
94.	<code> } else if (\$gejala[\$i] == 1){</code>
95.	<code> \$jawaban = "Sangat Yakin";</code>
96.	<code> }</code>
97.	<code> \$query1[\$i] = \$this->db->query("SELECT cf from tb_cf where id_gejala = '\$id_gejala'");</code>
98.	<code> foreach (\$query1[\$i]->result_array() as \$row) {</code>
99.	<code> \$cf_pakar[\$i] = \$row['cf'];</code>
100.	<code> }</code>
101.	<code> \$CFHE[\$i] = \$cf_pakar[\$i] * \$gejala[\$i];</code>
102.	<code> \$result = \$this->db->query("insert into tb_pemeriksaan values ('', '\$id_pasien', '\$id_gejala', '\$jawaban', '\$gejala[\$i]', \$CFHE[\$i], NOW(), \$id_diagnosis)");</code>
103.	<code> }</code>
104.	<code> return \$result;</code>
105.	<code>}</code>

Pada Tabel 4.5 merupakan kode program *function* *tambah_gejala()* yang berfungsi menghitung nilai $CF[H,E]$ pergejala dan menyimpan data pemeriksaan pasien. Tahapan menghitung nilai $CF[H,E]$ pergejala dimulai dari mengubah nilai CF setiap jawaban pertanyaan (CF_{user}) yang berupa angka menjadi sebuah kalimat dan disimpan pada variabel $\$jawaban$ yang selanjutnya digunakan untuk menyimpan jawaban pemeriksaan ke dalam database seperti pada baris 83-96. Pada baris 97-100 merupakan kode yang digunakan untuk menjalankan *query* mengambil nilai CF pakar setiap gejala dan disimpan kedalam variabel *array* $\$cf_pakar$. Perhitungan nilai $CF[H,E]$ pergejala didapat dari perkalian antara nilai CF pakar pergejala dengan nilai CF_{user} pergejala seperti pada baris 101. Hasil perhitungan nilai $CF[H,E]$ pergejala tersebut disimpan secara berulang seperti pada baris 102.

Tahap perhitungan metode *certainty factor* selanjutnya adalah menghitung nilai $CF_{Combine}$. Kode program untuk menghitung $CF_{Combine}$ yang terdapat pada *function* *cfcombine()* dapat dilihat pada Tabel 4.6.

Tabel 4.6 Kode program *function* *cfcombine()*

No	Kode
107.	<code>public function cfcombine(\$jumlah_gejala, \$id_pasien, \$id_diagnosis) {</code>
108	<code> \$CFH1 = \$this->db->query("SELECT kombinasi FROM tb_pemeriksaan where id_pasien =</code>
	<code> '\$id_pasien' and id_diagnosis = \$id_diagnosis LIMIT 0,1");</code>
109	<code> foreach (\$CFH1->result_array() as \$row) {</code>
110	<code> \$CFCombine = \$row['kombinasi'];</code>
111	<code> }</code>
112	
113	<code> for (\$i = 1; \$i < \$jumlah_gejala; \$i++) {</code>
114	<code> \$query = \$this->db->query("SELECT * FROM tb_gejala where jenis = 'demam tifoid' order</code>
	<code> by id_gejala asc limit \$i,1");</code>
115	<code> foreach (\$query->result_array() as \$row) {</code>
116	<code> \$id[\$i] = \$row['id_gejala'];</code>
117	<code> }</code>
118	<code> \$result[\$i] = \$this->db->query("SELECT kombinasi FROM tb_pemeriksaan where id_pasien</code>
	<code> = '\$id_pasien' and id_diagnosis = \$id_diagnosis and id_gejala = '\$id[\$i]'");</code>
119	<code> foreach (\$result[\$i]->result_array() as \$row) {</code>
120	<code> \$com[\$i] = \$row['kombinasi'];</code>
121	<code> }</code>
122	<code> \$CFCombine = \$CFCombine + (\$com[\$i] * (1 - \$CFCombine));</code>
123	<code> }</code>
124.	<code> return \$CFCombine;}</code>

Pada Tabel 4.6 merupakan kode program *function* *cfcombine()* yang berfungsi untuk menghitung *CFCCombine*. Tahapan perhitungan dimulai dari mengambil nilai *CF[H,E]* gejala pertama sebagai nilai awal dari *CFCCombine* dengan menjalankan *query* seperti pada baris 108-111. Tahap berikutnya mengambil nilai *CF[H,E]* pergejala dengan menjalankan *query* seperti pada baris 113-121 kemudian mengkombinasikan *CF[H,E]* pergejala tersebut dan mendapatkan nilai *CFCCombine* seperti pada baris 122.

Kode program penentuan penanganan terdapat pada *function* *tambah_diagnosis()* yang dapat dilihat pada Tabel 4.7. Nilai *CFCCombine* yang telah didapat akan dicari status penanganannya seperti pada baris 128-132. Penulisan kode program lainnya dapat dilihat pada Lampiran D (Kode Program).

Tabel 4.7 Kode program *function* *tambah_diagnosis()*

No	Kode
127	<code>public function tambah_diagnosis(\$CFCCombine, \$id_diagnosis) {</code>
128	<code> if (\$CFCCombine >= 0.9){</code>
129	<code> \$status = 'Demam Tifoid Berat';</code>
130	<code> }else if (\$CFCCombine > 0 && \$CFCCombine < 0.9){</code>
131	<code> \$status = 'Demam Tifoid Ringan';</code>
132	<code> } else{</code>
133	<code> \$status = 'negatif';</code>
134	<code> }</code>
135	<code> \$result = \$this->db->query("update tb_diagnosis set total = \$CFCCombine, diagnosis = '\$status' where id_diagnosis = \$id_diagnosis");</code>
136	<code> return \$status;}</code>

4.5 Pengujian

Tahapan pengujian sistem merupakan suatu tahapan yang dilakukan secara sistematis untuk menguji dan mengevaluasi sistem dengan menggunakan sebuah metode pengujian sistem untuk mengetahui apakah kebutuhan sistem telah terpenuhi dan sistem layak untuk digunakan oleh pengguna. Agar pengujian yang dilakukan lebih valid, maka tahap pengujian sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* ini dilakukan dengan menggunakan dua metode, yaitu *white box* dan *black box*.

4.5.1. Metode *White Box*

Pengujian sistem dengan metode *white box* dilakukan untuk menguji sistem dari segi desain dan kode program. Hal tersebut bertujuan untuk mengevaluasi apakah sistem mampu menghasilkan fungsi-fungsi, inputan, dan keluaran yang sesuai dengan spesifikasi dari kebutuhan sistem itu sendiri. Pengujian dengan metode *white box* dilakukan oleh peneliti dengan cara menghitung *independent path* yaitu dengan menggunakan suatu pengukuran kuantitatif *cyclomatic complexity*, listing program, penentuan jalur independen, dan *test case*. Tahapan pengujian menggunakan metode *white box* meliputi:

1. *Listing Program*

Listing program merupakan baris-baris kode yang nantinya akan diuji. Setiap langkah dari kode-kode yang ada diberi nomor baik menjalankan *statement* biasa atau penggunaan kondisi dalam program.

2. Diagram alir

Diagram alir merupakan notasi yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari listing program. Diagram alir digambarkan dengan node-node (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program.

3. Kompleksitas siklomatik (*cyclomatic complexity*)

Kompleksitas siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Bila digunakan dalam konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam basis set suatu program. Perhitungan kompleksitas siklomatik menggunakan rumus [3.1].

$$V(G) = E - N + 2 \quad [3.1]$$

Keterangan :

V(G) : Kompleksitas siklomatik

E : Jumlah *edge*

N : Jumlah *node*

4. Jalur Independen (*Independent Path*)

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru. Bila dinyatakan dalam grafik alir, jalur independen harus bergerak setidaknya sepanjang satu *edge* yang belum dilintasi sebelum jalur tersebut didefinisi.

5. Pengujian Basis Set (*Test Case*)

Pada bagian ini diberikan contoh data yang akan memaksa pelaksanaan jalur di basis set. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah melewati basis set yang tersedia. Sistem telah memenuhi syarat kelayakan perangkat lunak jika salah satu jalur yang dieksekusi setidaknya satu kali.

Pengujian sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* menggunakan metode *white box* akan diterapkan pada fitur yang dinilai dapat mewakili sistem pakar diagnosis penyakit demam tifoid sebagai berikut.

A. Listing Program Metode *Certainty Factor*

```

41     public function checkbaru() {
42         $id_user = $this->session->userdata('id_user');
43         $data['nama'] = $this->input->post('nama');
44         $data['umur'] = $this->input->post('umur');
45
46         $id_pasien = $this->m_diagnosis->ambil_id_pasien();
47         $this->m_diagnosis->insert_pasien($id_pasien, $data, $id_user);
48
49         $id_diagnosis = $this->m_diagnosis->ambil_id_diagnosis();
50         $this->m_diagnosis->insert_id_diagnosis($id_diagnosis);
51
52         $jumlah = $this->m_diagnosis->jumlah();
53         foreach ($jumlah->result_array() as $row) {
54             $jumlah_gejala = $row['jumlah'];
55         }
56
57         for ($i=0; $i < $jumlah_gejala ; $i++) {
58             $ambil2 = $this->m_diagnosis->ambil_id($i);
59             foreach ($ambil2->result_array() as $p) {
60                 $gejala = $this->input->post($p['id_gejala']);
61                 $this->m_diagnosis->tambah_gejala($p['id_gejala'], $gejala, $id_pasien, $id_diagnosis);
62             }
63         }
64
65         $CFCombine = $this->m_diagnosis->cfcombine($jumlah_gejala, $id_pasien, $id_diagnosis);
66         $status = $this->m_diagnosis->tambah_diagnosis($CFCombine, $id_diagnosis);
67         $pasien['penanganan'] = $this->m_diagnosis->view_penanganan($status);
68         $pasien['pasien'] = $this->m_diagnosis->view_hasil($id_diagnosis);
69         $data['username'] = $this->session->userdata('username');
70         $this->load->view('user/header', $data);
71         $this->load->view('user/hasil', $pasien);
72         $this->load->view('user/footer');
73     }
74 }

```

Gambar 4.7 Listing program *function* checkbaru()

```

80 public function tambah_gejala($id_gejala,$gejala, $id_pasien, $id_diagnosis) {
81
82     for ($i = 0; $i < count($gejala); $i++) {
83         $jawaban = "";
84         if($gejala[$i] == 0){
85             $jawaban = "Tidak";
86         }else if ($gejala[$i] == 0.2){
87             $jawaban = "Tidak Tahu";
88         }else if ($gejala[$i] == 0.4){
89             $jawaban = "Sedikit Yakin";
90         }else if ($gejala[$i] == 0.6){
91             $jawaban = "Cukup Yakin";
92         }else if ($gejala[$i] == 0.8){
93             $jawaban = "Yakin";
94         }else if ($gejala[$i] == 1){
95             $jawaban = "Sangat Yakin";
96         }
97         $query1[$i] = $this->db->query("SELECT cf from tb_cf where id_gejala = '$id_gejala'");
98         foreach ($query1[$i]->result_array() as $row) {
99             $cf_pakar[$i] = $row['cf'];
100         }
101         $CFHE[$i] = $cf_pakar[$i] * $gejala[$i];
102         $result = $this->db->query("insert into tb_pemeriksaan values ( ' ', '$id_pasien', '$id_gejala', '$jawaban', '$gejala[$i]
103             ]', $CFHE[$i], NOW(), $id_diagnosis)");
104     }
105     return $result;
}

```

Gambar 4.8 Listing program function tambah_gejala()

```

107 public function cfcombine($jumlah_gejala, $id_pasien, $id_diagnosis) { /* Load Data golongan Semua */
108     $CFH1 = $this->db->query("SELECT kombinasi FROM tb_pemeriksaan where id_pasien = '$id_pasien' and id_diagnosis = $
109         id_diagnosis LIMIT 0,1");
110     foreach ($CFH1->result_array() as $row) {
111         $CFCombine = $row['kombinasi'];
112     }
113     for ($i = 1; $i < $jumlah_gejala; $i++) {
114         $query = $this->db->query("SELECT * FROM tb_gejala where jenis = 'demam tifoid' order by id_gejala asc limit $i,1");
115         foreach ($query->result_array() as $row) {
116             $id[$i] = $row['id_gejala'];
117         }
118         $result[$i] = $this->db->query("SELECT kombinasi FROM tb_pemeriksaan where id_pasien = '$id_pasien' and id_diagnosis = $
119             id_diagnosis and id_gejala = '$id[$i]'");
120         foreach ($result[$i]->result_array() as $row) {
121             $com[$i] = $row['kombinasi'];
122         }
123         $CFCombine = $CFCombine + ($com[$i] * (1 - $CFCombine));
124     }
125     return $CFCombine;
}

```

Gambar 4.9 Listing program function cfcombine()

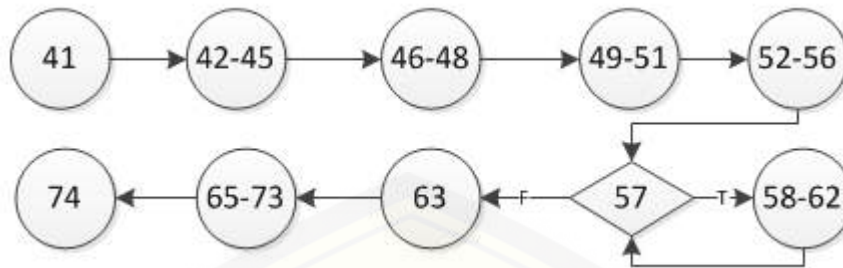
```

127 public function tambah_diagnosis($CFCombine, $id_diagnosis) {
128     if ($CFCombine >= 0.9){
129         $status = 'Demam Tifoid Berat';
130     }else if ($CFCombine > 0 && $CFCombine < 0.9){
131         $status = 'Demam Tifoid Ringan';
132     }else{
133         $status = 'negatif';
134     }
135     $result = $this->db->query("update tb_diagnosis set total = $CFCombine, diagnosis = '$status' where id_diagnosis = $
136         id_diagnosis");
137     return $status;
}

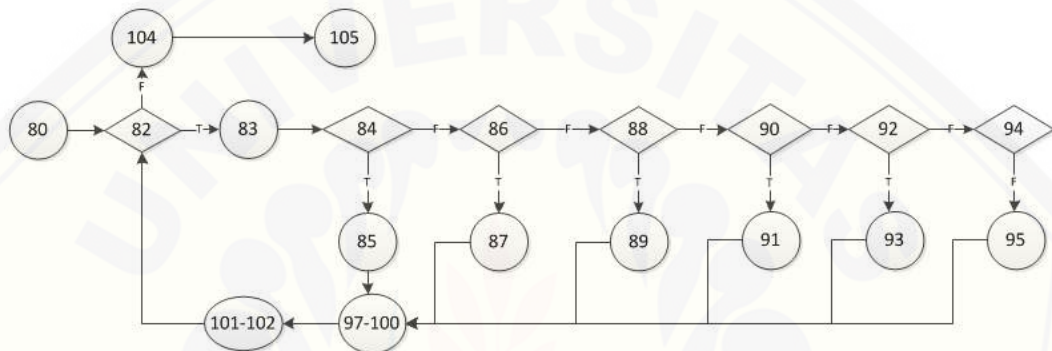
```

Gambar 4.10 Listing program function tambah_diagnosis()

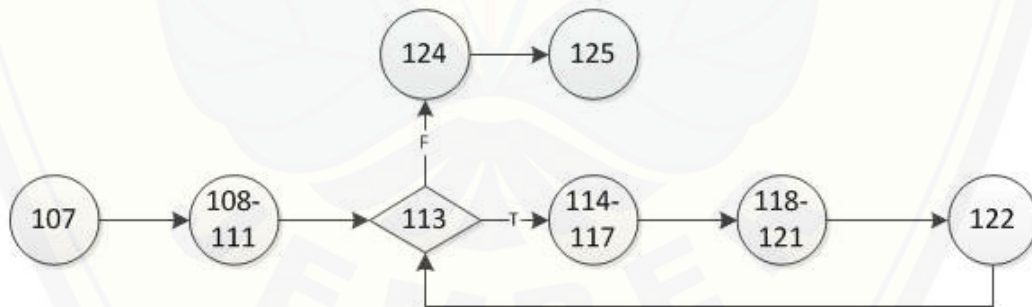
B. Diagram Alir



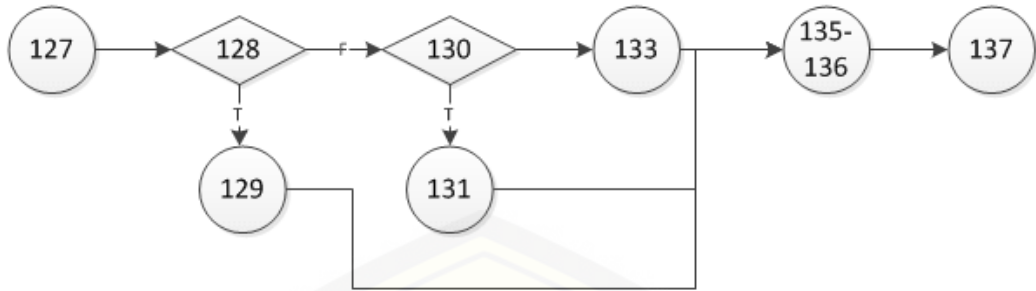
Gambar 4.11 Diagram alir function checkbaru()



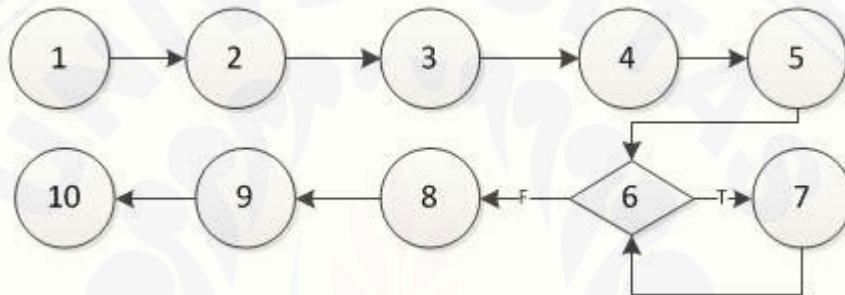
Gambar 4.12 Diagram alir function tambah_gejala()



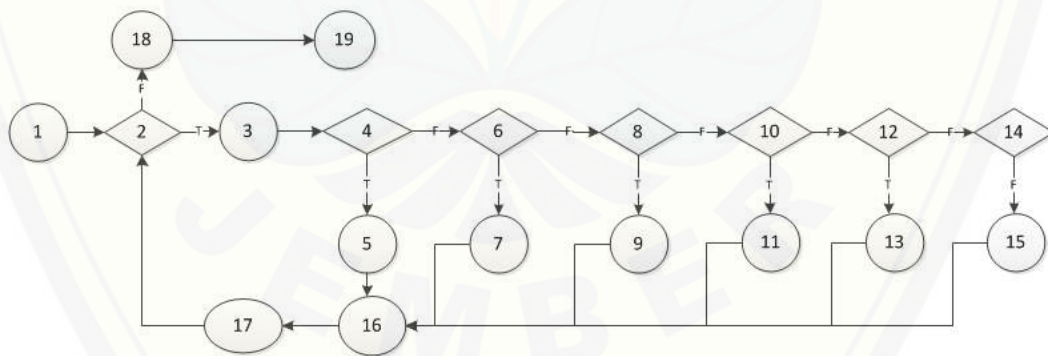
Gambar 4.13 Diagram Alir function cfcombine()



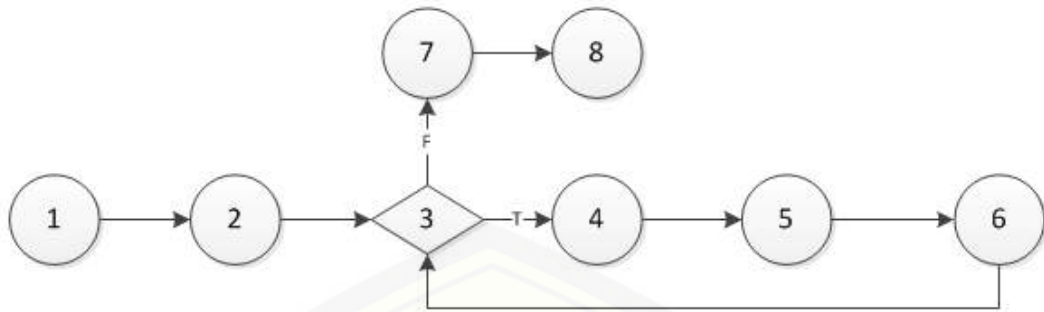
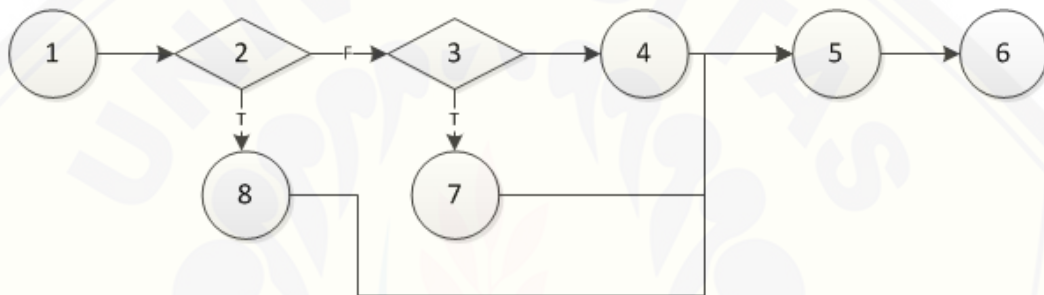
Gambar 4.14 Diagram Alir *function* tambah_diagnosis()



Gambar 4.15 Grafik alir *function* checkbaru()



Gambar 4.16 Grafik alir *function* tambah_gejala()

Gambar 4.17 Grafik alir *function* cfcombine()Gambar 4.18 Grafik alir *function* tambah_diagnosis()

C. Kompleksitas Siklomatik (*Cyclomatic Complexity*)

1. Function checkbaru()

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

$$V(G) = 11 - 10 + 2 = 2$$

2. Function tambah_gejala()

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

$$V(G) = 24 - 19 + 2 = 7$$

3. Function cfcombine()

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

$$V(G) = 8 - 8 + 2 = 2$$

4. Function tambah_diagnosis()

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

$$V(G) = 9 - 8 + 2 = 3$$

D. Jalur Independen (*Independent Path*)

1. Function checkbaru()

Jalur 1 = 1-2-3-4-5-6-8-9-10

Jalur 2 = 1-2-3-4-5-6-7-8-9-10

2. Function tambah_gejala()

Jalur 1 = 1-2-18-19

Jalur 2 = 1-2-3-4-5-16-17-18-19

Jalur 3 = 1-2-3-4-6-7-16-17-18-19

Jalur 4 = 1-2-3-4-6-8-9-16-17-18-19

Jalur 5 = 1-2-3-4-6-8-10-11-16-17-18-19

Jalur 6 = 1-2-3-4-6-8-10-12-13-16-17-18-19

Jalur 7 = 1-2-3-4-6-8-10-12-14-15-16-17-18-19

3. Function cfcombine()

Jalur 1 = 1-2-3-7-8

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

4. Function tambah_diagnosis()

Jalur 1 = 1-2-8-5-6

Jalur 2 = 1-2-3-7-5-6

Jalur 3 = 1-2-3-4-5-6

D. Pengujian Basis Set (*Test Case*)Tabel 4.8 *Test case function checkbaru()*

<i>Test case function checkbaru()</i>	
Jalur 1	
<i>Test Case</i>	Jika <i>for</i> ($\$i=0; \$j < \$jumlah_gejala; \$i++$) pada <i>node</i>
	6 tidak dilakukan
Target yang diharapkan	Gagal menghitung hasil diagnosis dan menampilkan persentase keyakinan 0% dan status penanganan 'negatif'
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-8-9-10

Dilanjutkan..

Lanjutan...

Jalur 2	
<i>Test Case</i>	Jika <i>for (\$i=0;\$j<\$jumlahgejala;\$i++)</i> pada <i>node</i> 6 dilakukan
Target yang diharapkan	Berhasil menghitung hasil diagnosis dan menampilkan persentase keyakinan dan saran penanganan serta berhasil menyimpan data
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-7-8-9-10

Tabel 4.9 *Test case function tambah_gejala()*

Test case function tambah_gejala()	
Jalur 1	
<i>Test Case</i>	Jika <i>for (\$i=0;\$j<count(\$gejala);\$i++)</i> pada <i>node</i> 2 tidak dilakukan
Target yang diharapkan	Gagal menghitung CF[H,E] dan gagal menyimpan data
Hasil pengujian	Benar
Path/Jalur	1-2-18-19
Jalur 2	
<i>Test Case</i>	Jika <i>if(\$gejala[i] == 0)</i>
Target yang diharapkan	Berhasil menghitung CF[H,E], berhasil menyimpan data dengan data Jawaban = 'tidak'
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-16-17-18-19
Jalur 3	
<i>Test Case</i>	Jika <i>if(\$gejala[i] == 0.2)</i>
Target yang diharapkan	Berhasil menghitung CF[H,E], berhasil menyimpan data dengan data Jawaban = 'tidak tahu'
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-13-8-9-10

Dilanjutkan...

Lanjutan...

Jalur 4	
<i>Test Case</i>	Jika $if(\$gejala[i] == 0.4)$
Target yang diharapkan	Berhasil menghitung CF[H,E], berhasil menyimpan data dengan data <i>Jawaban</i> = 'sedikit yakin'
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-6-8-9-16-17-18-19
Jalur 5	
<i>Test Case</i>	Jika $if(\$gejala[i] == 0.6)$
Target yang diharapkan	Berhasil menghitung CF[H,E], berhasil menyimpan data dengan data <i>Jawaban</i> = 'cukup yakin'
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-6-8-10-11-16-17-18-19
Jalur 6	
<i>Test Case</i>	Jika $if(\$gejala[i] == 0.8)$
Target yang diharapkan	Berhasil menghitung CF[H,E], berhasil menyimpan data dengan data <i>Jawaban</i> = 'yakin'
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-6-8-10-12-13-16-17-18-19
Jalur 7	
<i>Test Case</i>	Jika $if(\$gejala[i] == 1)$
Target yang diharapkan	Berhasil menghitung CF[H,E], berhasil menyimpan data dengan data <i>Jawaban</i> = 'sangat yakin'
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-6-8-10-12-14-15-16-17-18-19

Tabel 4.10 *Test case function cfcombine()*

Test case function cfcombine()	
Jalur 1	
<i>Test Case</i>	Jika $for (\$i=1; \$j < \$jumlah_gejala; \$i++)$ pada <i>node</i> 3 tidak dilakukan
Target yang diharapkan	Gagal menghitung CFCombine dan serta gagal menyimpan data

Dilanjutkan...

Lanjutan...

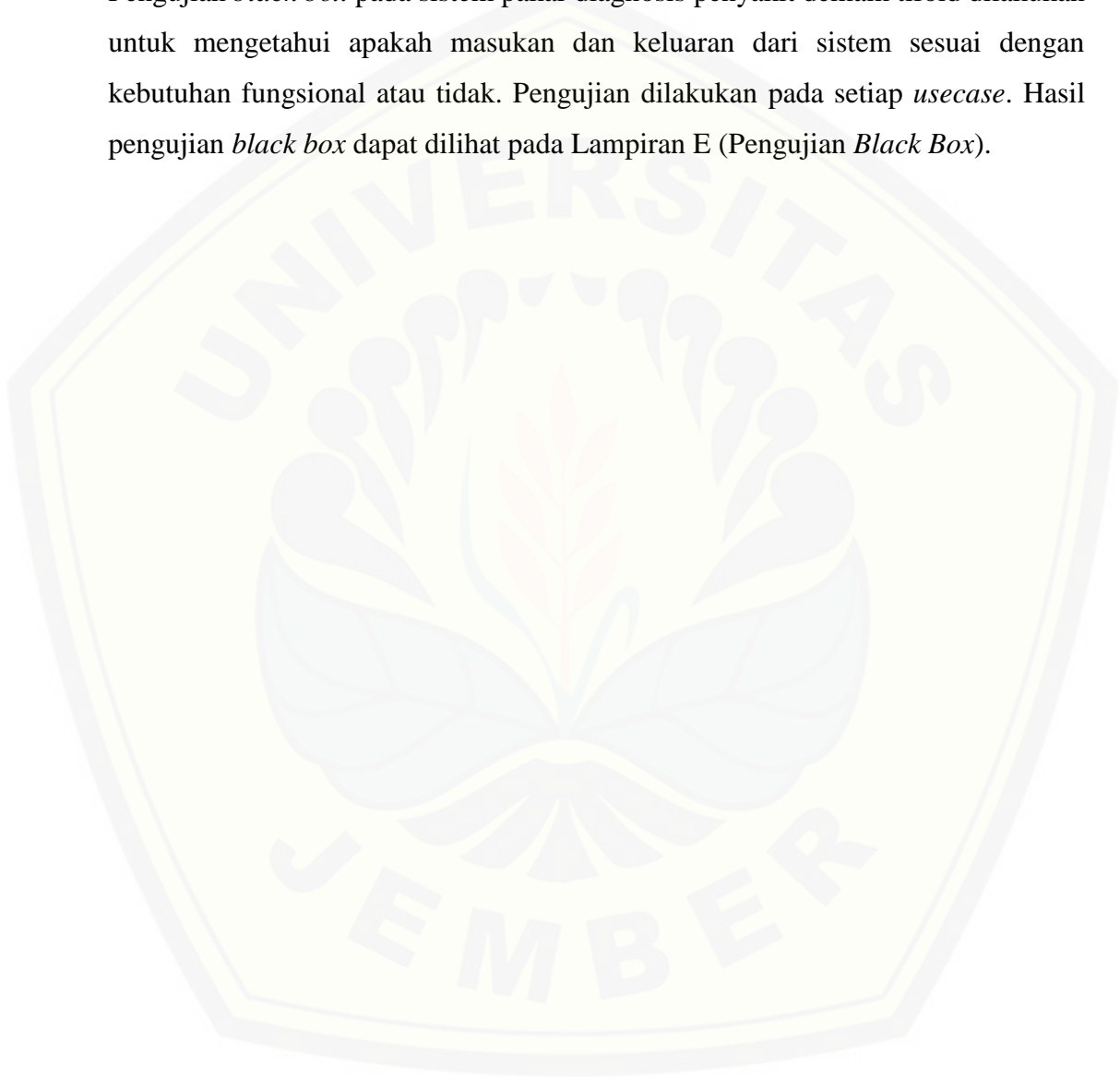
Jalur 1	
Hasil pengujian	Benar
Path/Jalur	1-2-3-7-8
Jalur 2	
<i>Test Case</i>	Jika <i>for (\$i=1;\$j<\$jumlah_gejala;\$i++)</i> pada <i>node</i> 3 dilakukan
Target yang diharapkan	Berhasil menghitung CFCombine dan serta berhasil menyimpan data
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-7-8

Tabel 4.11 *Test case function* tambah_diagnosis()

Test case function tambah_diagnosis()	
Jalur 1	
<i>Test Case</i>	Jika <i>if(\$CFCombine >= 0.9)</i>
Target yang diharapkan	Berhasil menyimpan data dan menampilkan data penanganan demam tifoid berat
Hasil pengujian	Benar
Path/Jalur	1-2-8-5-6
Jalur 2	
<i>Test Case</i>	Jika <i>if(\$CFCombine >0 && \$CFCombine <0.9)</i>
Target yang diharapkan	Berhasil menyimpan data dan menampilkan data penanganan demam tifoid ringan
Hasil pengujian	Benar
Path/Jalur	1-2-3-7-5-6
Jalur 3	
<i>Test Case</i>	Jika <i>\$CFCombine</i> selain range tersebut
Target yang diharapkan	Berhasil menyimpan data dan menampilkan data status penanganan negatif
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6

4.5.2. Metode *Black Box*

Black box testing merupakan metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja. Metode ini memfokuskan pada keperluan fungsionalitas dari software. Pengujian *black box* pada sistem pakar diagnosis penyakit demam tifoid dilakukan untuk mengetahui apakah masukan dan keluaran dari sistem sesuai dengan kebutuhan fungsional atau tidak. Pengujian dilakukan pada setiap *usecase*. Hasil pengujian *black box* dapat dilihat pada Lampiran E (Pengujian *Black Box*).



BAB 6. PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah:

- a. Penerapan metode *certainty factor* digunakan pada diagnosis demam tifoid untuk memberikan perhitungan penyelesaian seberapa pasti pasien menderita penyakit demam tifoid berdasarkan gejala yang dialami. Langkah awal yakni menentukan nilai CF pakar yang didapat dari wawancara dengan pakar dan nilai CF *user* untuk mewakili tingkat keyakinan gejala yang dialami pasien dalam rentang 0-1. Selanjutnya pasien memasukkan gejala yang dialami. Dari gejala yang dialami tersebut selanjutnya dihitung *CFCombine* serta persentase keyakinannya menggunakan metode *certainty factor*. Hasil dari perhitungan yang didapatkan adalah persentase keyakinan dan saran penanganan. Hal yang mempengaruhi hasil persentase keyakinan adalah tingkat keyakinan dari gejala yang dialami oleh pasien. Semakin pasien yakin akan gejala yang dialami maka persentase keyakinan yang dihasilkan akan semakin tinggi.
- b. Sistem pakar diagnosis penyakit demam tifoid menggunakan metode *certainty factor* telah berhasil dibangun berbasis *website* dengan 2 hak akses yaitu admin dan anggota dengan fitur utama adalah fitur pemeriksaan pasien yang dilakukan oleh anggota. Sistem dikembangkan dengan mengadopsi model *waterfall*. Model *waterfall* digunakan karena kebutuhan sistem telah bisa didefinisikan di awal. Tahap pengembangan dilakukan dengan tahap analisis kebutuhan, desain, penulisan kode program, pengujian dan pemeliharaan. Analisis kebutuhan dilakukan dengan cara mencari kebutuhan fungsional dan non fungsional yang dibutuhkan untuk membangun sistem. Tahap desain

sistem dilakukan dengan melakukan perancangan sistem dengan membuat beberapa diagram yang akan digunakan sebagai acuan dalam penulisan kode program. Selanjutnya dilakukan penulisan kode program dengan menggunakan bahasa pemrograman *PHP*, *framework CodeIgniter*, dan database manajemen *MySQL*. Tahap pengujian dilakukan setelah penulisan kode program yakni dengan melakukan pengujian *black box* dan *white box*. Tahap terakhir yakni melakukan pemeliharaan untuk mengetahui fungsionalitas sistem secara berkala.

6.2 Saran

Adapun saran yang ditujukan untuk memberikan masukan yang lebih baik yaitu :

- a. Sistem pakar yang akan dikembangkan selanjutnya perlu diberikan peringatan ketika masukan sistem tidak sesuai dengan *rule* agar hasil yang didapat akurat.
- b. Sistem pakar yang akan dikembangkan selanjutnya akan lebih baik jika menggunakan lebih dari satu penyakit dan menggunakan diagnosis penunjang agar hasil yang didapat lebih akurat.

DAFTAR PUSTAKA

- Inawati. (2009). Demam Tifoid. *Jurnal Ilmiah Kedokteran, Edisi Khusus Desember 2009*, 31-36.
- Kusumadewi, S. (2003). *Artificial Intelligence (Teknik dan Aplikasinya)* (1 ed.). Yogyakarta, Yogyakarta, Indonesia: Graha Ilmu.
- Mandal, B. K., Wilkins, E. G., Dunbar, E. M., & Mayon-White, R. T. (2006). *Penyakit Infeksi*. Jakarta: Penerbit Erlangga.
- Pressman, R. S. (2001). *Software Engineering a practitioner's approach* (5 ed.). New York, America: McGraw-Hill.
- Sahara, Q. A. (Tanpa Tahun). *Aplikasi Android Sistem Pakar Guna Mendiagnosa Penyakit Usus Pada Manusia Menggunakan Metode Certainty Factor*. Universitas Dian Nuswantoro, Fakultas Ilmu Komputer. Semarang: Universitas Dian Nuswantoro.
- Sari, N. A. (2013). Sistem Pakar Mendiagnosa Penyakit Demam Berdarah Menggunakan Metode Certainty Factor. *Jurnal Ilmiah Pelita Informatika Budi Darma*, 4, 100-103.
- Wijaya, B., & Prasetiyowati, M. I. (2012). Rancang Bangun Sistem Pakar Pendiagnosa Penyakit Demam Typhoid dan Demam Berdarah Menggunakan Metode Forward Chaining. *Jurnal Teknik Informatika*, 4, 17-23.
- World Health Organization. (2003). *Background document: The diagnosis, treatment and prevention of typhoid fever*. Geneva, Switzerland: Department of Vaccines and Biologicals.

LAMPIRAN

Lampiran A. Use Case Skenario

A.1 Use Case Skenario Menambah Data Gejala

Tabel A.1 Use case skenario menambah data gejala

ID	USC-01
Name	Menambah Data Gejala
Participating Actor	Admin
Entry Condition	Admin akan menambah data gejala
Exit Condition	Admin telah menambah data gejala
SKENARIO UTAMA “Menambah Data Gejala”	
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Klik menu penyakit 2. Klik submenu data gejala 4. Klik tombol tambah gejala 6. Mengisi kolom <ol style="list-style-type: none"> a. kode gejala b. gejala c. pertanyaan 7. Klik tombol simpan 	<ol style="list-style-type: none"> 3. Menampilkan halaman data gejala yang berisi: <ol style="list-style-type: none"> a. No b. Kode c. Gejala d. Action 5. Menampilkan halaman form tambah gejala 8. Menyimpan data

Dilanjutkan ...

Lanjutan...

9. Kembali ke halaman data gejala dan menampilkan pesan “berhasil di tambah”

**SKENARIO UTAMA
“Kode Gejala Telah digunakan”**

Aksi Aktor	Reaksi Sistem
6. Mengisi kolom <ol style="list-style-type: none"> a. kode gejala b. gejala c. pertanyaan 	
5. Klik tombol simpan	6. menampilkan pesan “kode telah digunakan”

**SKENARIO UTAMA
“Kolom Belum Diisi”**

Aksi Aktor	Reaksi Sistem
5. Klik tombol simpan	6. Menampilkan pesan “dengan menunjuk kolom pada form yang belum diisi”

A.2 Use Case Skenario Mengubah Data Gejala

Tabel A.2 Use case skenario mengubah data gejala

ID	USC-02
Name	Mengubah Data Gejala
Participating Actor	Admin
Entry Condition	Admin akan mengubah data gejala
Exit Condition	Admin telah mengubah data gejala

**SKENARIO UTAMA
“Mengubah Data Gejala”**

Dilanjutkan...

Lanjutan...

Aksi Aktor	Reaksi Sistem
1. Klik menu penyakit	
2. Klik submenu data gejala	
	3. Menampilkan halaman data gejala yang berisi: <ol style="list-style-type: none"> No Kode Gejala Gejala Action
4. Klik tombol <i>edit</i> pada kolom <i>Action</i>	5. Menampilkan halaman form edit gejala
6. Mengisi kolom <ol style="list-style-type: none"> Gejala Pertanyaan 	
7. Klik tombol simpan	8. Menyimpan data
	9. Kembali ke halaman data gejala dan menampilkan pesan “berhasil diubah”

A.3 Use Case Skenario Menghapus Data Gejala

Tabel A.3 Use case skenario menghapus data gejala

ID	USC-03
Name	Menghapus Data Gejala
Participating Actor	Admin
Entry Condition	Admin akan menghapus data gejala
Exit Condition	Admin telah menghapus data gejala

SKENARIO UTAMA “Menghapus Data Gejala”

Dilanjutkan...

Lanjutan...

Aksi Aktor	Reaksi Sistem
1. Klik menu penyakit	
2. Klik submenu data gejala	
	3. Menampilkan halaman data gejala yang berisi: <ol style="list-style-type: none"> No Kode Gejala Gejala Action
4. Klik tombol <i>delete</i> pada kolom Action	5. Menampilkan pesan “apakah anda yakin?” dengan tombol “ya” dan “tidak”
6. Klik tombol “ya”	7. Menghapus gejala
	8. Kembali ke halaman data gejala dan menampilkan pesan “berhasil dihapus”

SKENARIO UTAMA
“Batal Menghapus Data Gejala”

Aksi Aktor	Reaksi Sistem
4. Klik tombol <i>delete</i> pada kolom Action	
	5. Menampilkan pesan “apakah anda yakin?” dengan tombol “ya” dan “tidak”
6. Klik tombol “tidak”	7. Tidak menghapus gejala
	8. Tetap di halaman data gejala

A.4 Use Case Skenario Melihat Data Gejala

Tabel A.4 Use case skenario melihat data gejala

ID	USC-04
Name	Melihat Data Gejala
Participating Actor	Admin
Entry Condition	Admin akan melihat data gejala
Exit Condition	Admin telah melihat data gejala
SKENARIO UTAMA “Menambah Data Gejala”	
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Klik menu penyakit 2. Klik submenu data gejala 	<ol style="list-style-type: none"> 3. Menampilkan halaman data gejala yang berisi: <ol style="list-style-type: none"> a. No b. Kode c. Gejala d. Action

A.5 Use Case Skenario Menambah Data CF

Tabel A.5 Use case skenario menambah data CF

ID	USC-05
Name	Menambah Data CF
Participating Actor	Admin
Entry Condition	Admin akan menambah data CF
Exit Condition	Admin telah menambah data CF

Dilanjutkan...

Lanjutan...

SKENARIO UTAMA “Menambah Data CF”	
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Klik menu penyakit 2. Klik submenu data CF 4. Klik tombol tambah data CF 6. Memilih <ol style="list-style-type: none"> a. Jenis penyakit b. Jenis gejala 7. Mengisi kolom <ol style="list-style-type: none"> a. MB b. MD 8. Klik tombol simpan 	<ol style="list-style-type: none"> 3. Menampilkan halaman data CF yang berisi: <ol style="list-style-type: none"> a. No b. Kode Penyakit c. Kode Gejala d. MB e. MD f. <i>Certainty Factor</i> g. <i>Action</i> 5. Menampilkan halaman form tambah data CF 9. Menghitung nilai Certainty Factor dan Menyimpan data 10. Kembali ke halaman data CF dan menampilkan pesan “berhasil di tambah”
SKENARIO ALTERNATIF “nilai CF telah ada”	
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 6. Memilih <ol style="list-style-type: none"> a. Jenis penyakit b. Jenis gejala 7. Mengisi kolom <ol style="list-style-type: none"> a. MB b. MD 	

Dilanjutkan...

Lanjutan...

Aksi Aktor	Reaksi Sistem
8. Klik tombol simpan	9. Menampilkan pesan “data CF telah ada”

**SKENARIO ALTERNATIF
“kolom belum diisi”**

Aksi Aktor	Reaksi Sistem
8. Klik tombol simpan	9. Menampilkan pesan “dengan menunjuk kolom pada form yang belum diisi”

A.6 Use Case Skenario Mengubah Data CF

Tabel A.6 Use Case Skenario Mengubah Data CF

ID	USC-06
Name	Mengubah Data CF
Participating Actor	Admin
Entry Condition	Admin akan mengubah data CF
Exit Condition	Admin telah mengubah data CF

**SKENARIO UTAMA
“Megubah Data CF”**

Aksi Aktor	Reaksi Sistem
1. Klik menu penyakit	
2. Klik submenu data CF	

Dilanjutkan...

Lanjutan...

Aksi Aktor	Reaksi Sistem
	3. Menampilkan halaman data CF yang berisi: <ol style="list-style-type: none"> No Kode Penyakit Kode Gejala MB MD <i>Certainty Factor</i> <i>Action</i>
4. Klik tombol <i>edit</i> kolom Action	5. Menampilkan halaman form edit data CF
6. Mengisi kolom <ol style="list-style-type: none"> MB MD 	
7. Klik tombol simpan	8. Menghitung nilai <i>Certainty Factor</i> dan Menyimpan data 9. Kembali ke halaman data CF dan menampilkan pesan “berhasil di ubah”

A.7 Use Case Skenario Menghapus Data CF

Tabel A.7 Use case skenario menghapus data CF

ID	USC-07
Name	Menghapus Nilai CF
Participating Actor	Admin
Entry Condition	Admin akan menghapus data nilai CF
Exit Condition	Admin telah menghapus data nilai CF

SKENARIO UTAMA
“Menghapus Data CF”

Dilanjutkan...

Lanjutan...

Aksi Aktor	Reaksi Sistem
1. Klik menu penyakit	
2. Klik submenu data CF	
	3. Menampilkan halaman data CF a. No b. Kode Penyakit c. Kode Gejala d. MB e. MD f. <i>Certainty Factor</i> g. <i>Action</i>
4. Klik tombol <i>delete</i> kolom <i>Action</i>	
6. Klik tombol “ya”	5. Menampilkan pesan “apakah anda yakin?” dengan tombol “ya” dan “tidak”
	7. Menghapus nilai CF
	8. Kembali ke halaman data CF dan menampilkan pesan “berhasil di hapus”

SKENARIO UTAMA
“Batal Hapus Data CF”

Aksi Aktor	Reaksi Sistem
4. Klik tombol <i>delete</i> kolom <i>Action</i>	
6. Klik tombol “tidak”	5. Menampilkan pesan “apakah anda yakin?” dengan tombol “ya” dan “tidak”
	7. Tidak menghapus nilai CF
	8. Tetap di halaman data CF

A.8 Use Case Skenario Melihat Data CF

Tabel A.8 Use case skenario melihat data CF

ID	USC-08
Name	Melihat Data CF
Participating Actor	Admin
Entry Condition	Admin akan melihat data CF
Exit Condition	Admin telah melihat data CF

SKENARIO UTAMA
“Melihat Data CF”

Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Klik menu penyakit 2. Klik submenu data CF 	<ol style="list-style-type: none"> 3. Menampilkan halaman data CF <ol style="list-style-type: none"> a. No b. Kode Penyakit c. Kode Gejala d. MB e. MD f. <i>Certainty Factor</i> g. <i>Action</i>

A.9 Use Case Skenario Melihat Rekap Data Pasien

Tabel A.9 Use case skenario melihat rekap data pasien

ID	USC-09
Name	Melihat Rekap Data Pasien
Participating Actor	Admin
Entry Condition	Admin akan melihat data pasien
Exit Condition	Admin telah melihat data pasien

Dilanjutkan....

Lanjutan...

SKENARIO UTAMA “Melihat Rekap Data Pasien”	
Aksi Aktor	Reaksi Sistem
1. Klik menu diagnosis	
2. Klik submenu data diagnosis	
	3. Menampilkan halaman data pasien yang berisi: <ol style="list-style-type: none"> a. No b. Nama c. Umur d. <i>User</i> e. Detail

A.10 *Use Case* Skenario Melihat Data Diagnosis

Tabel A.10 *Use case* skenario melihat data diagnosis

ID	USC-10
Name	Melihat Data Diagnosis
Participating Actor	Admin
Entry Condition	Admin akan melihat data diagnosis
Exit Condition	Admin telah melihat data diagnosis

SKENARIO UTAMA “Melihat Data Diagnosis”	
Aksi Aktor	Reaksi Sistem
1. Klik menu diagnosis	
2. Klik submenu data diagnosis	

Dilanjutkan...

Lanjutan...

Aksi Aktor	Reaksi Sistem
4. Klik tombol lihat <i>record</i> pada kolom Detail	3. Menampilkan halaman data pasien yang berisi: <ol style="list-style-type: none"> No Nama Umur <i>User</i> Detail
6. Klik tombol Detail pada kolom Detail	5. Menampilkan halaman data <i>record</i> pasien yang berisi <ol style="list-style-type: none"> No Nama Umur Tanggal Persentase Detail
	7. Menampilkan halaman detail diagnosis pasien yang dipilih berisi: <ol style="list-style-type: none"> No ID Pasien Nama pasien Umur Jawaban CF <i>user</i> CF pakar Total <i>CFCombine</i> Status

A.11 Use Case Skenario Memasukkan Pemeriksaan Pasien Lama

Tabel A.11 Use case skenario memasukkan pemeriksaan pasien lama

ID	USC-11
Name	Memasukkan Pemeriksaan Pasien Lama
Participating Actor	Anggota
Entry Condition	Anggota akan melakukan pemeriksaan untuk pasien lama
Exit Condition	Anggota telah melakukan pemeriksaan untuk pasien lama dan mengetahui hasil diagnosis

SKENARIO UTAMA**“Memasukkan pemeriksaan pasien lama ”**

Aksi Aktor	Reaksi Sistem
1. Klik menu dashboard	2. Menampilkan data pasien yang berisi: a. No b. Nama c. Aksi
3. Klik tombol lihat record pada kolom aksi	4. Menampilkan halaman data record pasien yang berisi: a. No b. Tanggal c. Persentase d. Aksi
5. Klik tombol periksa lanjutan	6. Menampilkan halaman form keluhan
7. Mengisi form konsultasi yang berisi pilihan jawaban pada setiap gejala yang diajukan. Pilihan jawaban yang diajukan meliputi:	

Dilanjutkan...

Lanjutan...

Aksi Aktor	Reaksi Sistem
a. Sangat Yakin b. Yakin c. Cukup Yakin d. Sedikit Yakin e. Tidak Tahu f. Tidak	
8. Klik tombol check	9. Menghitung hasil diagnosis berdasarkan jawaban yang dipilih menggunakan metode <i>certainty factor</i> 10. Menampilkan halaman hasil diagnosis yang berisi: <ol style="list-style-type: none"> No Nama Pasien Persentase Saran Penanganan

SKENARIO ALTERNATIF
“terdapat jawaban yang belum dipilih ”

Aksi Aktor	Reaksi Sistem
7. Mengisi form konsultasi yang berisi pilihan jawaban pada setiap gejala yang diajukan. Pilihan jawaban yang diajukan meliputi: <ol style="list-style-type: none"> Sangat Yakin Yakin Cukup Yakin Sedikit Yakin Tidak Tahu Tidak 	
8. Klik tombol check	9. Menampilkan pesan “dengan menunjuk pada pertanyaan yang belum dipilih”

A.12 Use Case Skenario Memasukkan Pemeriksaan Pasien Baru

Tabel A.12 Use case skenario memasukkan pemeriksaan pasien baru

ID	USC-12
Name	Memasukkan Pemeriksaan Pasien Baru
Participating Actor	Anggota
Entry Condition	Anggota akan melakukan pemeriksaan untuk pasien baru
Exit Condition	Anggota telah melakukan pemeriksaan untuk pasien baru dan mengetahui hasil diagnosis

SKENARIO UTAMA**“Memasukkan pemeriksaan pasien baru ”**

Aksi Aktor	Reaksi Sistem
1. Klik menu dashboard	2. Menampilkan data pasien
3. Klik tombol periksa pasien baru	4. Menampilkan halaman form keluhan
5. Mengisi kolom <ul style="list-style-type: none"> a. Nama b. Umur 	
6. Mengisi form konsultasi yang berisi pilihan jawaban pada setiap gejala yang diajukan. Pilihan jawaban yang diajukan meliputi: <ul style="list-style-type: none"> a. Sangat Yakin b. Yakin c. Cukup Yakin d. Sedikit Yakin e. Tidak Tahu f. Tidak 	
7. Klik tombol check	

Dilanjutkan...

Lanjutan..

Aksi Aktor	Reaksi Sistem
	8. Menghitung hasil diagnosis berdasarkan jawaban yang dipilih menggunakan metode <i>certainty factor</i> 9. Menampilkan halaman hasil diagnosis yang berisi: <ol style="list-style-type: none"> Nama Pasien Persentase Saran Penanganan

SKENARIO ALTERNATIF
“kolom belum diisi ”

Aksi Aktor	Reaksi Sistem
6. Mengisi form biodata berisi: <ol style="list-style-type: none"> Nama Umur 7. Mengisi form konsultasi yang berisi pilihan jawaban pada setiap gejala yang diajukan. Pilihan jawaban yang diajukan meliputi: <ol style="list-style-type: none"> Sangat Yakin Yakin Cukup Yakin Sedikit Yakin Tidak Tahu Tidak 8. Klik tombol check	9. Menampilkan pesan dengan menunjuk pada kolom yang belum diisi

Dilanjutkan...

Lanjutan...

SKENARIO ALTERNATIF	
“Terdapat jawaban yang belum dipilih”	
Aksi Aktor	Reaksi Sistem
10. Mengisi form biodata berisi: <ol style="list-style-type: none"> a. Nama b. Umur 11. Mengisi form konsultasi yang berisi pilihan jawaban pada setiap gejala yang diajukan. Pilihan jawaban yang diajukan meliputi: <ol style="list-style-type: none"> a. Sangat Yakin b. Yakin c. Cukup Yakin d. Sedikit Yakin e. Tidak Tahu f. Tidak 12. Klik tombol check	13. Menampilkan pesan dengan menunjuk pada pertanyaan yang belum diisi

A.13 *Use Case* Skenario Melihat Data Pasien Anggota

Tabel A.13 *Use case* skenario melihat data pasien anggota

ID	USC-13
Name	Melihat Data Pasien Anggota
Participating Actor	Anggota
Entry Condition	Anggota akan melihat data pasien anggota
Exit Condition	Anggota telah melihat data pasien anggota

Dilanjutkan...

Lanjutan...

SKENARIO UTAMA “Melihat data pasien anggota”	
Aksi Aktor	Reaksi Sistem
1. Klik menu dashboard	2. Menampilkan data pasien yang berisi: <ol style="list-style-type: none"> a. No b. Nama c. Umur d. Aksi

A.14 *Use Case* Skenario Melihat *Record* Pemeriksaan

Tabel A.14 *Use case* skenario melihat *record* pemeriksaan

ID	USC-14
Name	Melihat <i>Record</i> Pemeriksaan
Participating Actor	Anggota
Entry Condition	Anggota akan melihat data <i>record</i> pemeriksaan pasien anggota
Exit Condition	Anggota telah melihat data <i>record</i> pemeriksaan pasien anggota

SKENARIO UTAMA “Melihat record pemeriksaan ”	
Aksi Aktor	Reaksi Sistem
1. Klik menu dashboard	2. Menampilkan data pasien yang berisi: <ol style="list-style-type: none"> a. No b. Nama c. Umur d. Aksi

Dilanjutkan...

Lanjutan...

Aksi Aktor	Reaksi Sistem
3. Klik tombol lihat <i>record</i> pada kolom aksi	4. Menampilkan data record pasien yang berisi: <ol style="list-style-type: none"> No Tanggal Persentase Aksi

A.15 Use Case Skenario Melihat Detail Record Pemeriksaan

Tabel A.15 Use case skenario melihat detail record pemeriksaan

ID	USC-15
Name	Melihat Detail <i>Record</i> Pemeriksaan
Participating Actor	Anggota
Entry Condition	Anggota akan melihat detail data <i>record</i> pemeriksaan pasien anggota
Exit Condition	Anggota telah melihat detail data <i>record</i> pemeriksaan pasien anggota

SKENARIO UTAMA
“Melihat detail record pemeriksaan ”

Aksi Aktor	Reaksi Sistem
1. Klik menu dashboard	2. Menampilkan data pasien yang berisi: <ol style="list-style-type: none"> No Nama Umur Aksi
3. Klik tombol lihat <i>record</i> pada kolom aksi	

Dilanjutkan...

Lanjutan...

Aksi Aktor	Reaksi Sistem
	4. Menampilkan data record pasien yang berisi: <ol style="list-style-type: none"> No Tanggal Persentase Aksi
5. Klik tombol lihat detail pada kolom aksi	6. Menampilkan data detail record pemeriksaan pasien yang berisi: <ol style="list-style-type: none"> No Gejala Jawaban

A.16 Use Case Skenario Mendaftar Anggota

Tabel A.16 Use case skenario mendaftar anggota

ID	USC-16
Name	Mendaftar Anggota
Participating Actor	Anggota
Entry Condition	Aktor akan melakukan pendaftaran
Exit Condition	Akktor telah melakukan pendaftaran

SKENARIO UTAMA
“Mendaftar Anggota”

Aksi Aktor	Reaksi Sistem
1. Buka halaman web sistem	2. Menampilkan halaman tampilan awal
3. Klik menu <i>login</i>	4. Menampilkan halaman form <i>login</i>

Dilanjutkan...

Lanjutan...

Aksi Aktor	Reaksi Sistem
5. Klik tombol daftar	6. Menampilkan halaman form pendaftaran
7. Mengisi kolom <ol style="list-style-type: none"> Nama Jenis kelamin Tempat lahir Tanggal lahir Alamat <i>Username</i> <i>Password</i> 	
8. Klik tombol daftar	9. Menampilkan halaman fom <i>login</i> dan menampilkan pesan “pendaftaran berhasil”

SKENARIO ALTERNATIF
“Username telah digunakan”

Aksi Aktor	Reaksi Sistem
7. Mengisi kolom <ol style="list-style-type: none"> Nama Jenis kelamin Tempat lahir Tanggal lahir Alamat <i>Username</i> <i>Password</i> 	
8. Klik tombol daftar	9. Menampilkan pesan “Gagal! Username telah digunakan”

SKENARIO ALTERNATIF
“Belum Mengisi Kolom ”

Aksi Aktor	Reaksi Sistem
8. Klik tombol daftar	

Dilanjutkan...

Lanjutan...

Aksi Aktor	Reaksi Sistem
	9. Menampilkan pesan dengan menunjuk kolom pada form pendaftaran yang belum diisi

A.17 Use Case Skenario Login

Tabel A.17 Use Case Skenario Login

ID	USC-17
Name	<i>Login</i>
Participating Actor	Admin, Anggota
Entry Condition	Aktor akan melakukan <i>login</i>
Exit Condition	Aktor telah melakukan <i>login</i>

SKENARIO UTAMA
"Login"

Aksi Aktor	Reaksi Sistem
1. Buka halaman web sistem	
	2. Menampilkan halaman tampilan awal
3. Klik menu <i>login</i>	
	4. Menampilkan halaman form <i>login</i>
5. Mengisi kolom a. <i>Username</i> b. <i>Password</i>	
6. Klik tombol <i>login</i>	
	7. Menampilkan halaman <i>dashboard</i>

Dilanjutkan...

Lanjutan...

SKENARIO ALTERNATIF “Username dan Password Salah”	
Aksi Aktor	Reaksi Sistem
5. Mengisi kolom a. <i>Username</i> b. <i>Password</i>	
6. Klik tombol <i>login</i>	7. Menampilkan pesan “Login gagal! Username atau Password Salah

SKENARIO ALTERNATIF “Belum Mengisi Kolom ”	
Aksi Aktor	Reaksi Sistem
6. Klik tombol <i>login</i>	7. Menampilkan pesan “dengan menunjuk kolom pada form <i>login</i> yang belum diisi”

A.18 Use Case Skenario Logout

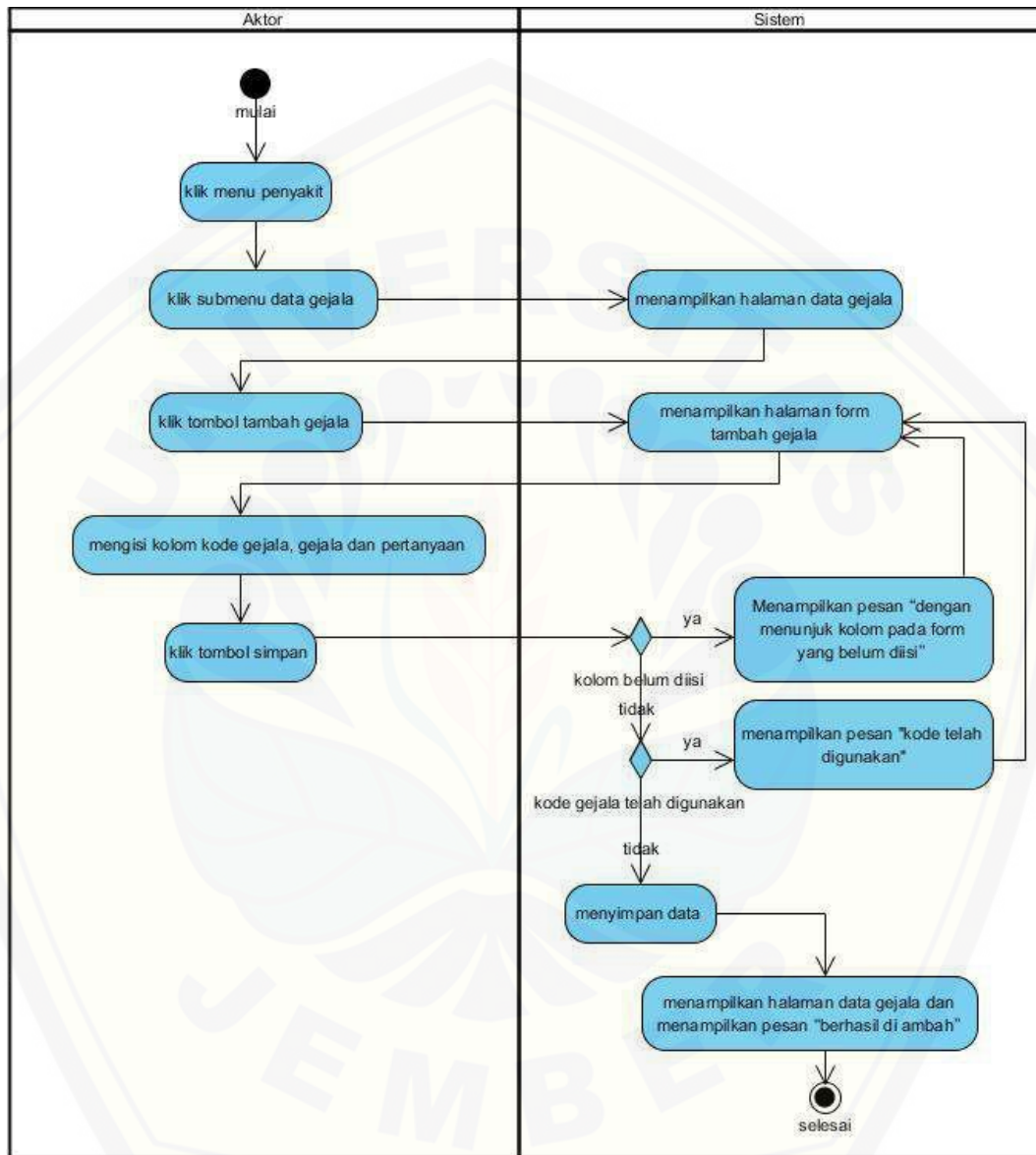
Tabel A.18 Use case skenario *Logout*

ID	USC-18
Name	<i>Logout</i>
Participating Actor	Admin, Anggota
Entry Condition	Aktor akan keluar dari sistem
Exit Condition	Aktor telah keluar dari sistem

SKENARIO UTAMA “Logout”	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>logout</i>	2. Menampilkan halaman tampilan awal

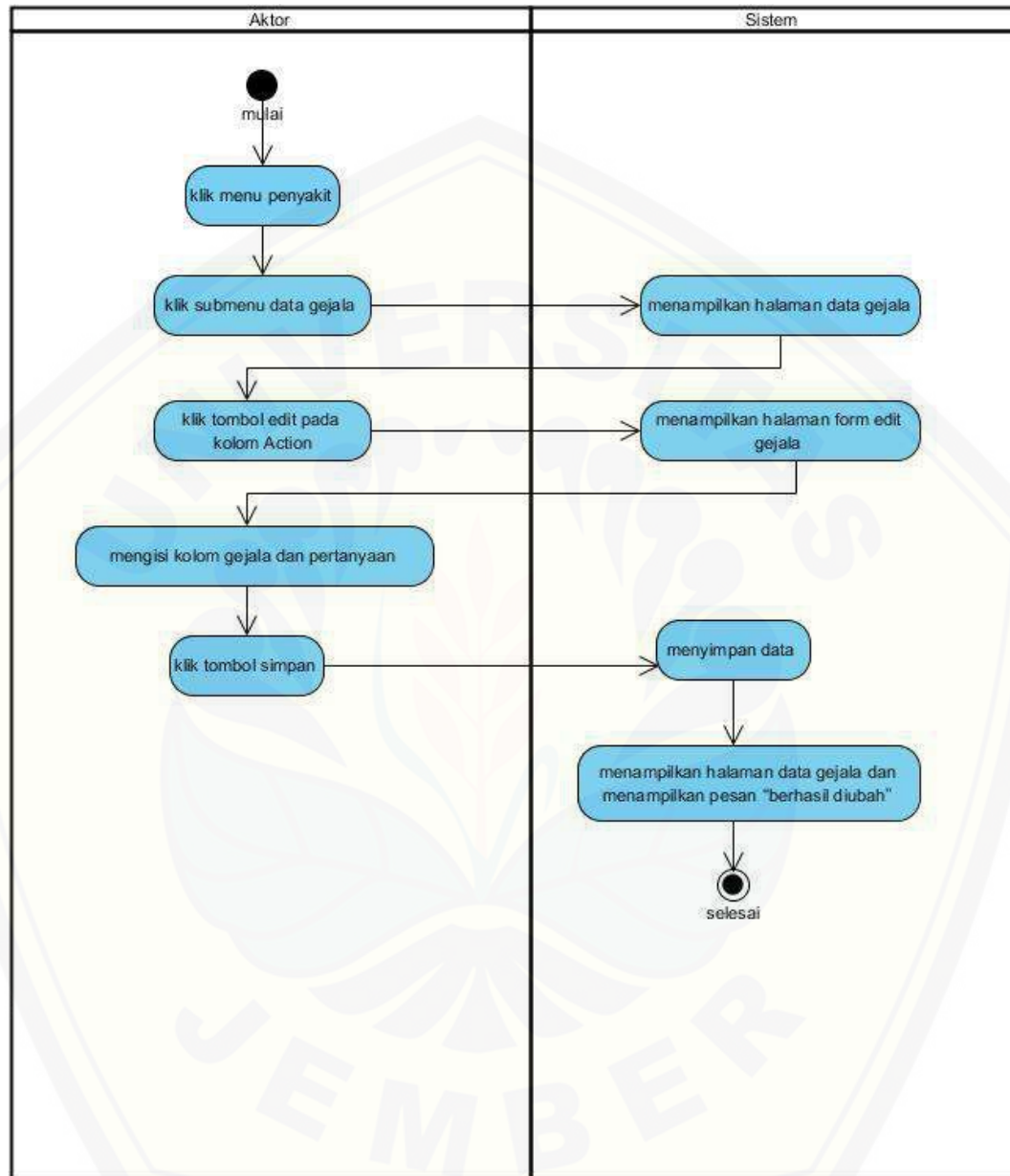
Lampiran B. Activity Diagram

B.1 Activity Diagram Menambah Data Gejala



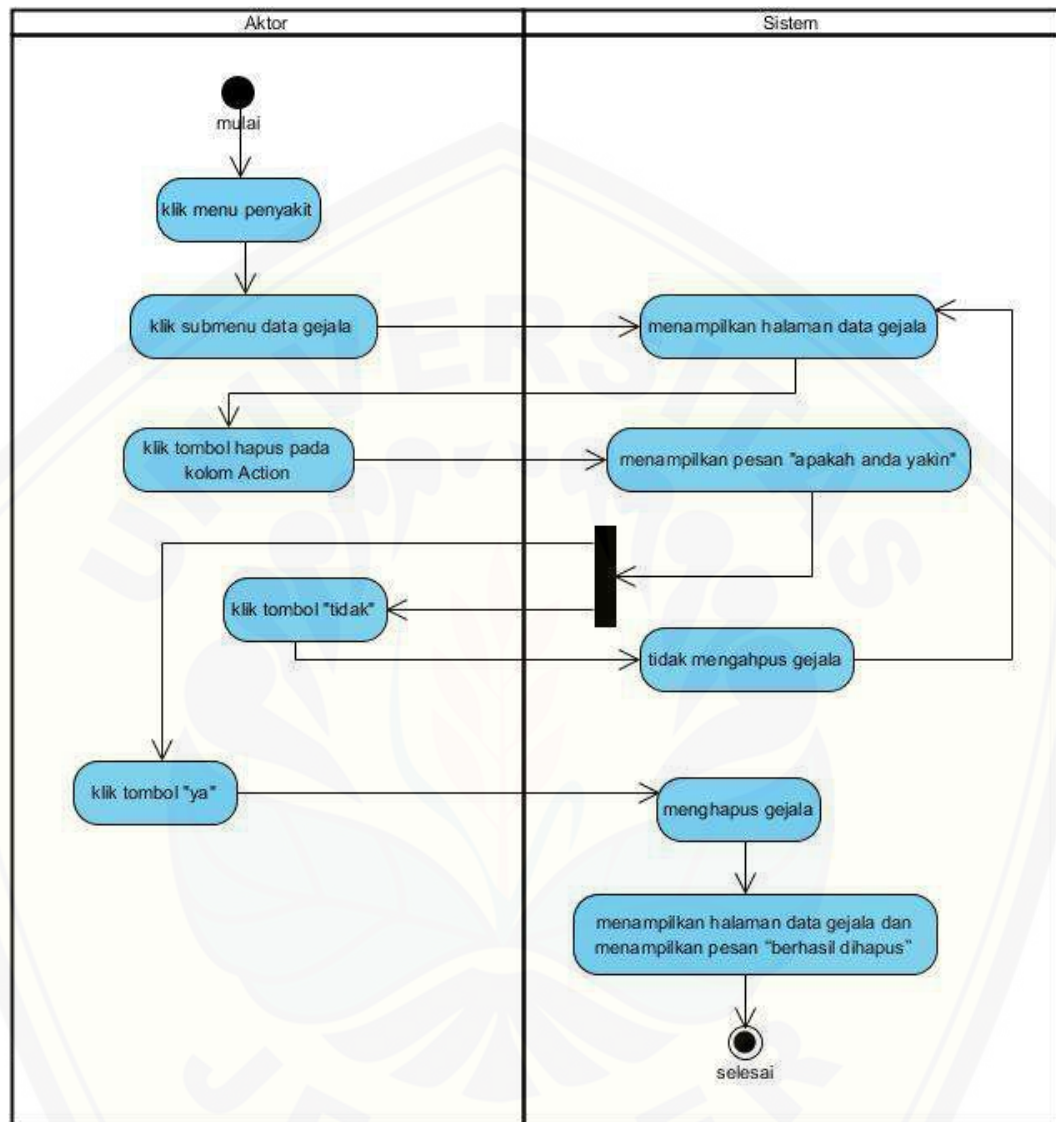
Gambar B.1 Activity diagram menambah data gejala

B.2 Activity Diagram Mengubah Data Gejala



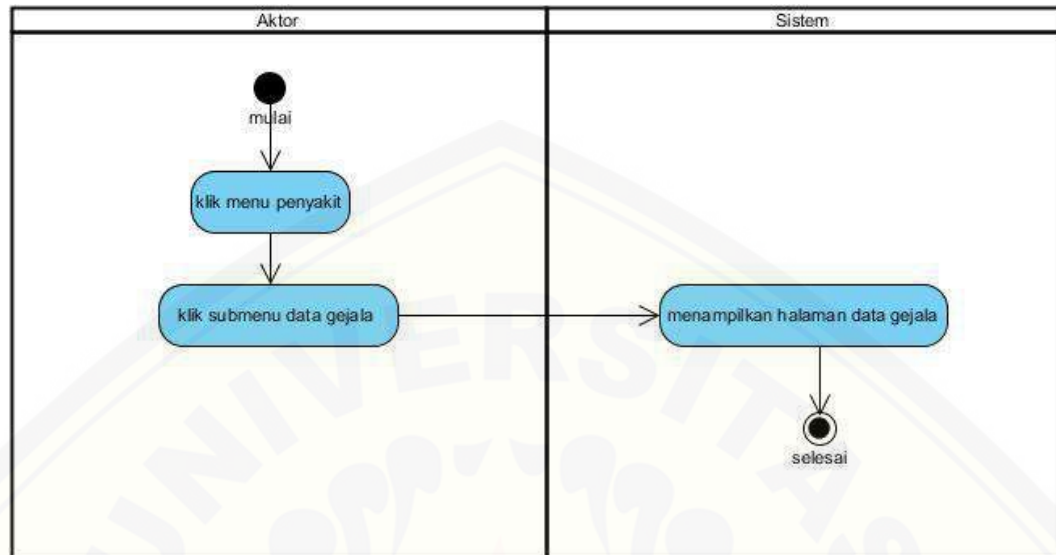
Gambar B.2 Activity diagram mengubah data gejala

B.3 Activity Diagram Menghapus Data Gejala



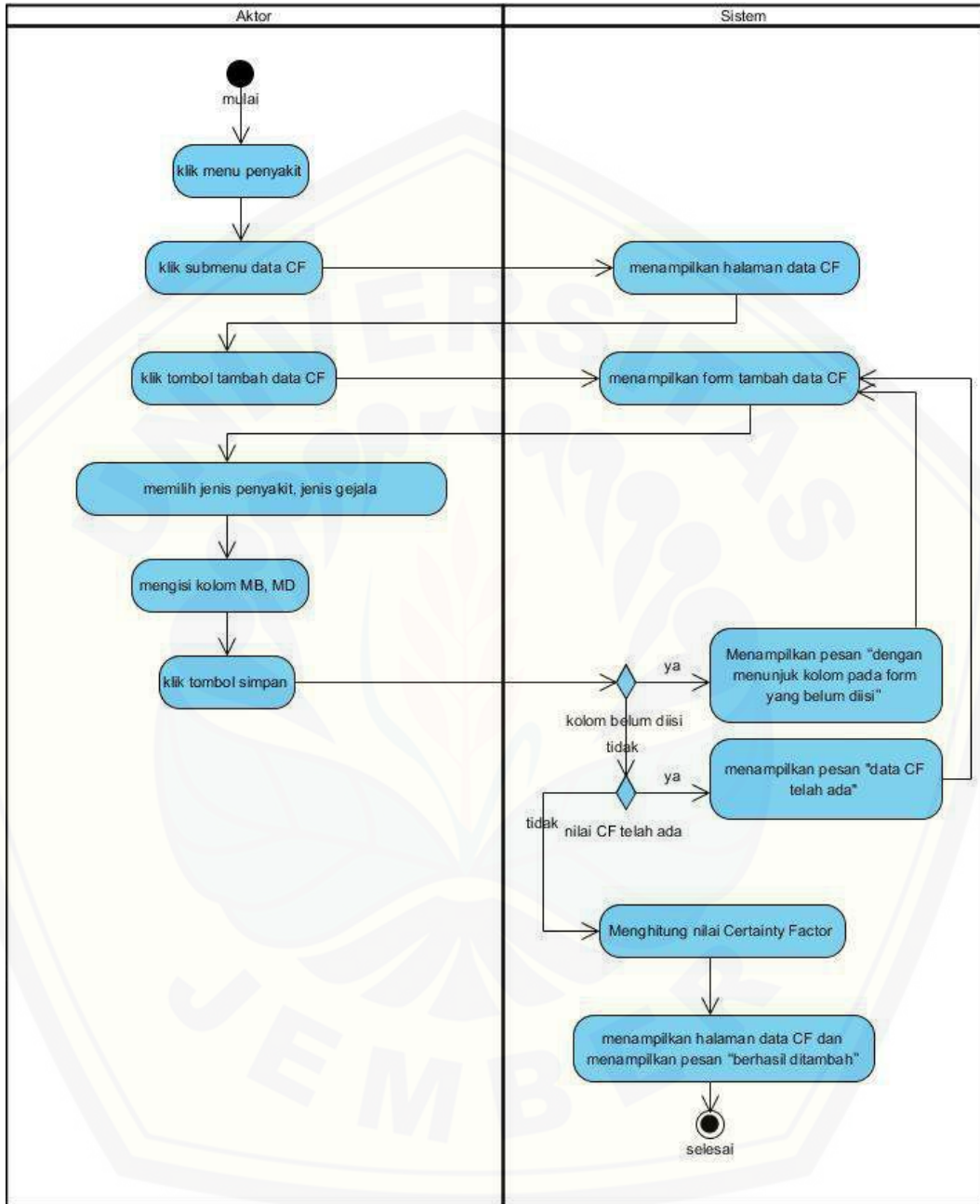
Gambar B.3 Activity diagram menghapus data gejala

B.4 Activity Diagram Melihat Data Gejala



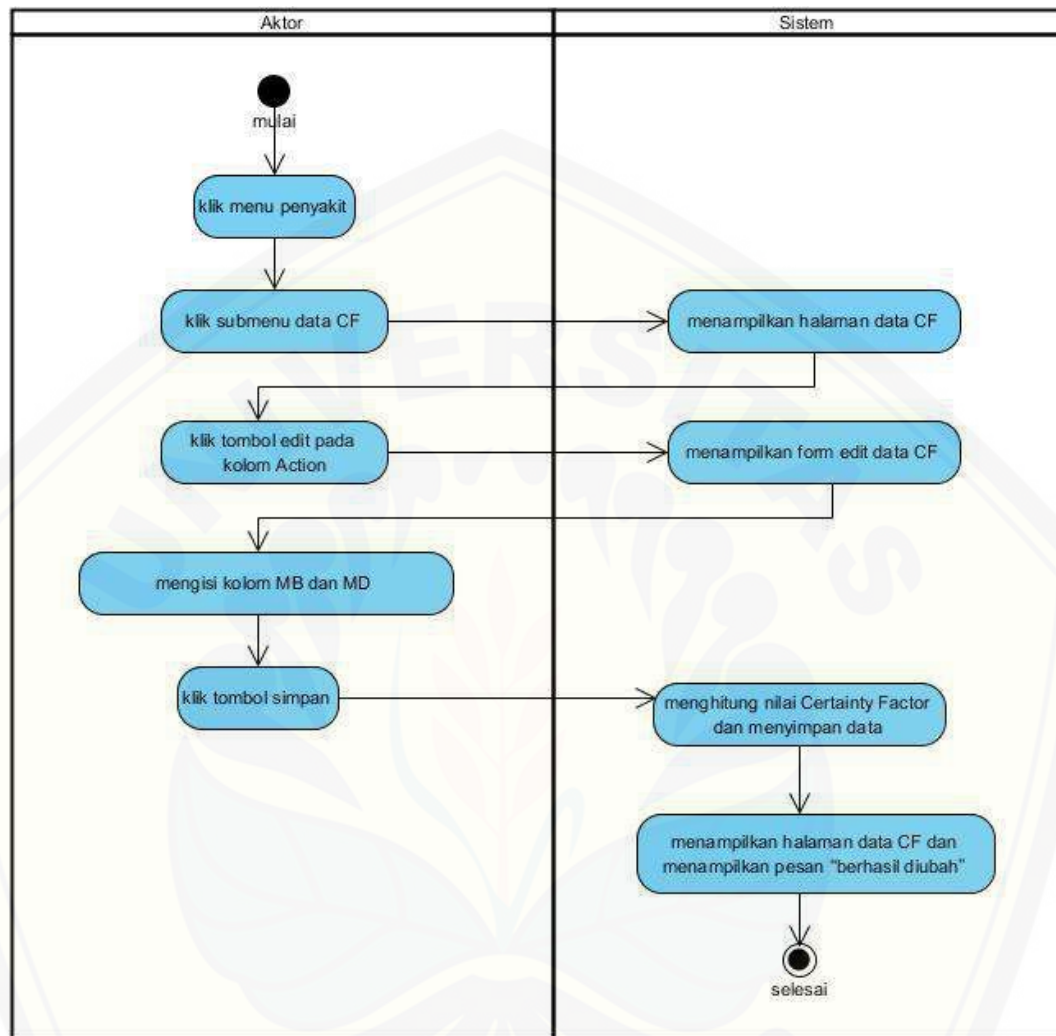
Gambar B.4 Activity diagram melihat data gejala

B.5 Activity Diagram Menambah Data CF



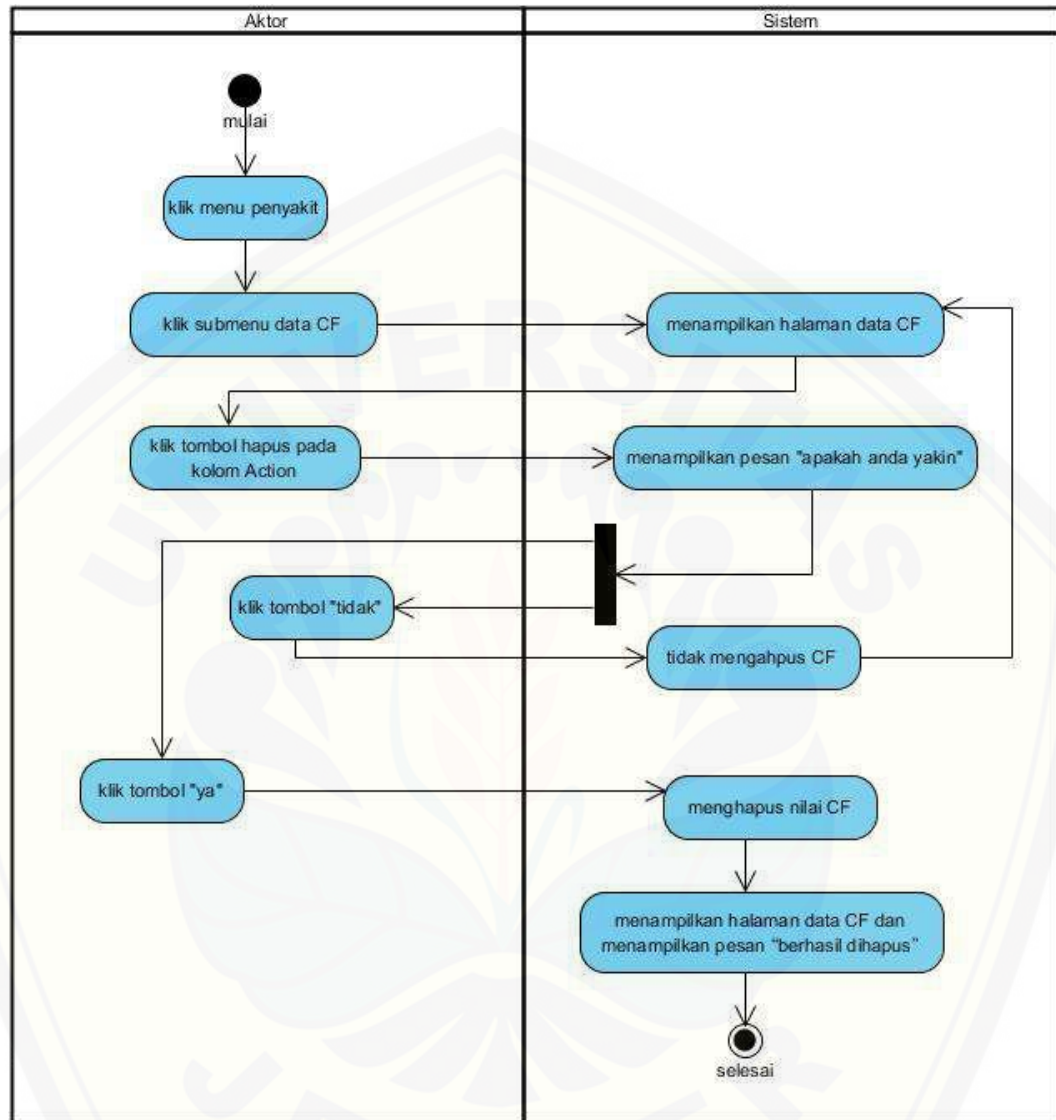
Gambar B.5 Activity diagram menambah data CF

B.6 Activity Diagram Mengubah Data CF



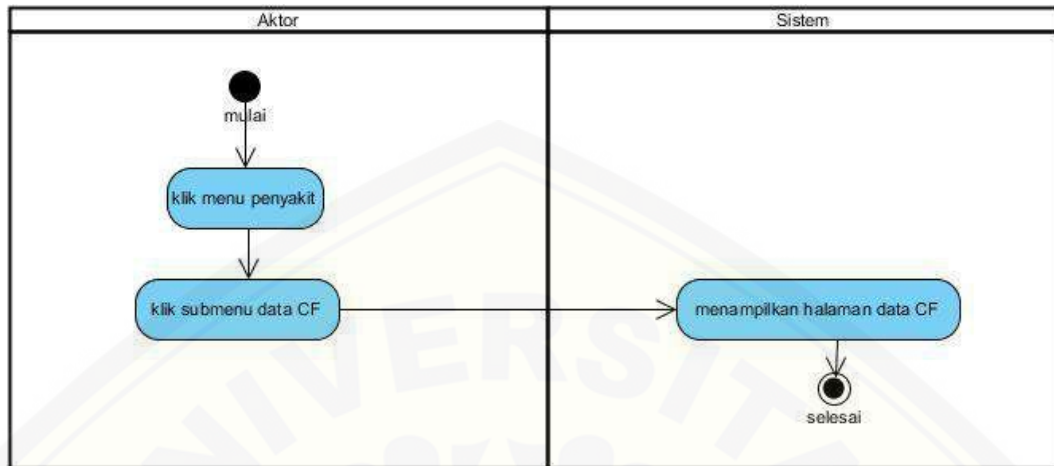
Gambar B 6 Activity diagram mengubah data CF

B.7 Activity Diagram Menghapus Data CF



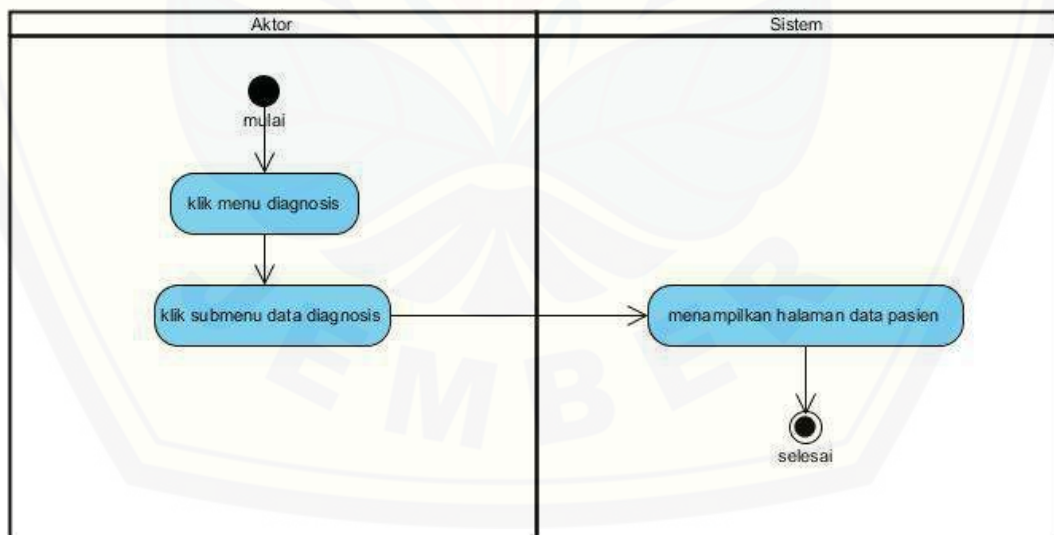
Gambar B.7 Activity diagram menghapus data CF

B.8 Activity Diagram Melihat Data CF



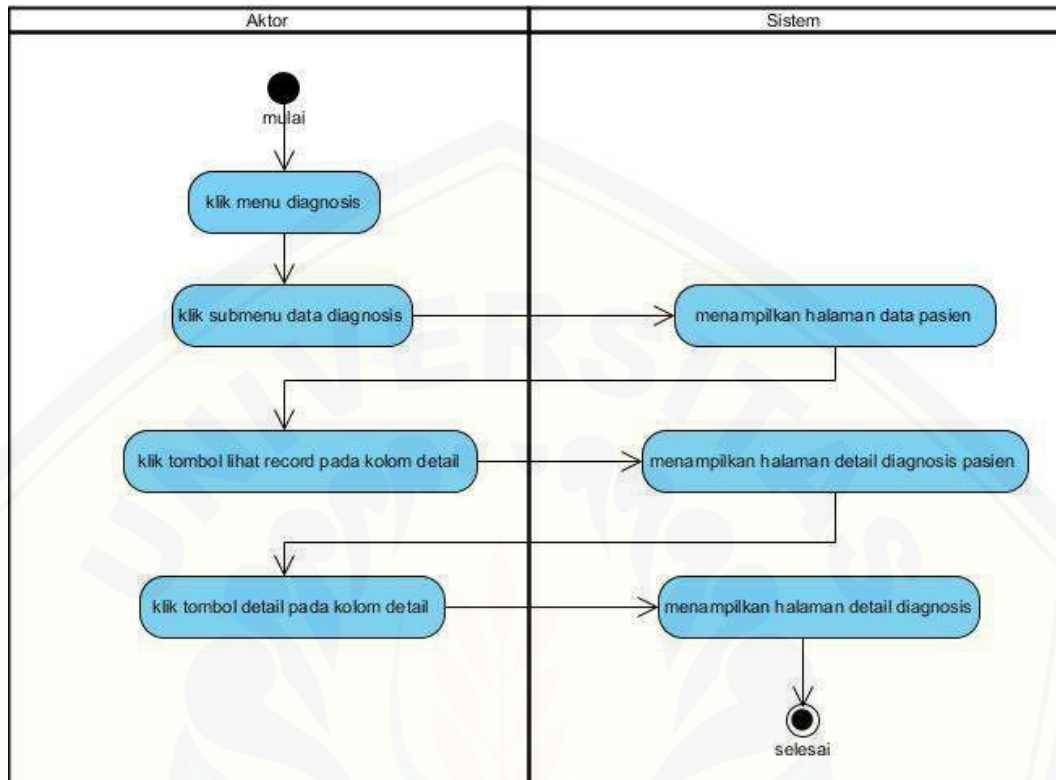
Gambar B.8 Activity diagram melihat data CF

B.9 Activity Diagram Melihat Rekap Data Pasien



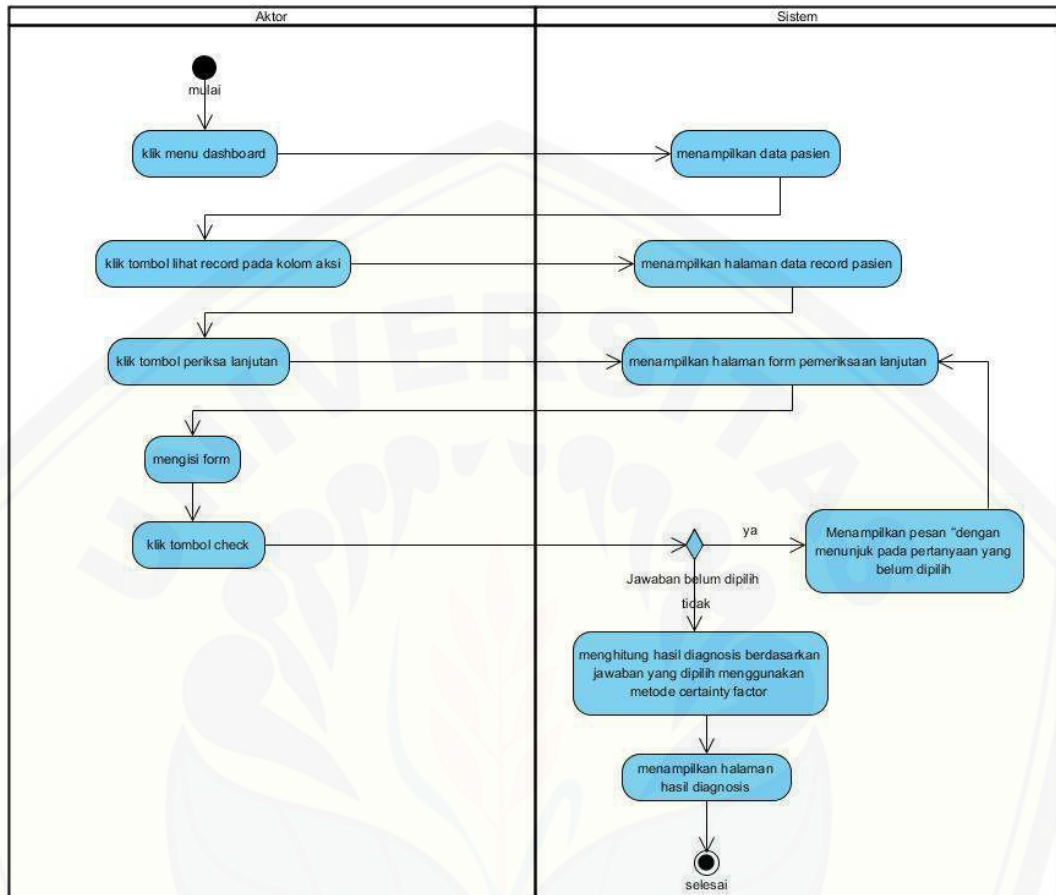
Gambar B.9 Activity diagram melihat rekap data pasien

B.10 Activity Diagram Melihat Data Diagnosis



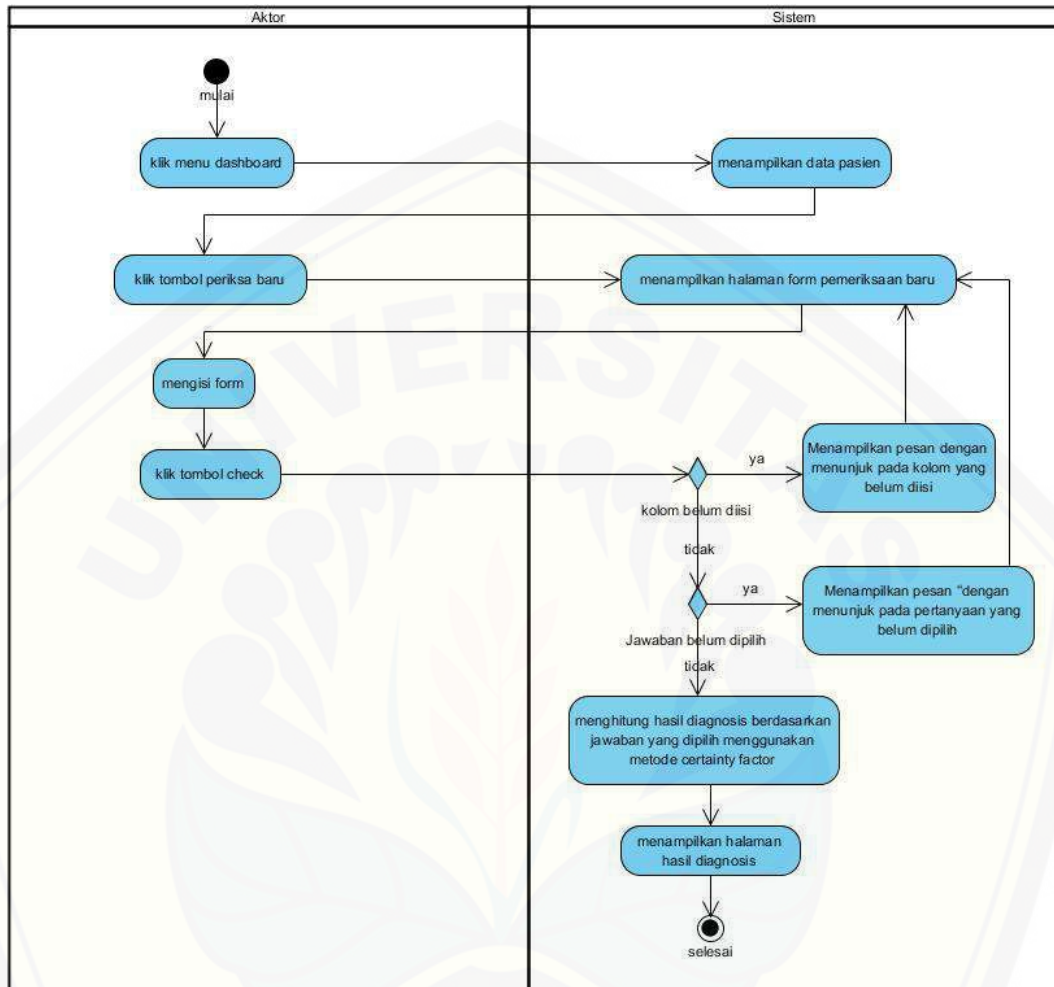
Gambar B.10 Activity diagram melihat data diagnosis

B.11 Activity Diagram Memasukkan Pemeriksaan Pasien Lama



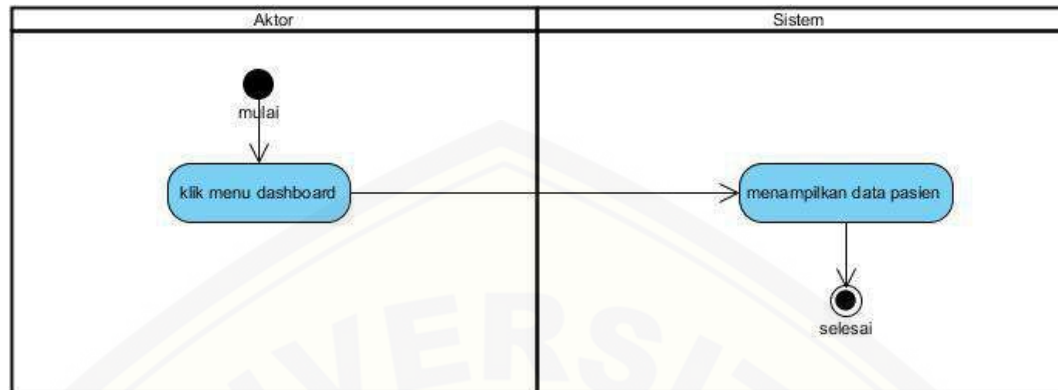
Gambar B.11 Activity diagram memasukkan pemeriksaan pasien lama

B.12 Activity Diagram Memasukkan Pemeriksaan Pasien Baru



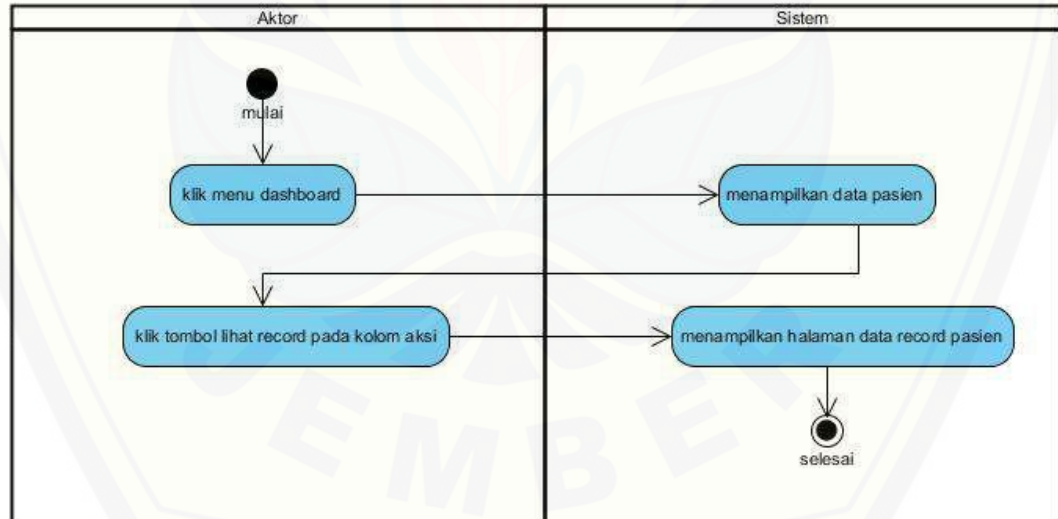
Gambar B.12 Activity diagram memasukkan pemeriksaan pasien baru

B.13 Activity Diagram Melihat Data Pasien Anggota



Gambar B.13 Activity diagram melihat data pasien anggota

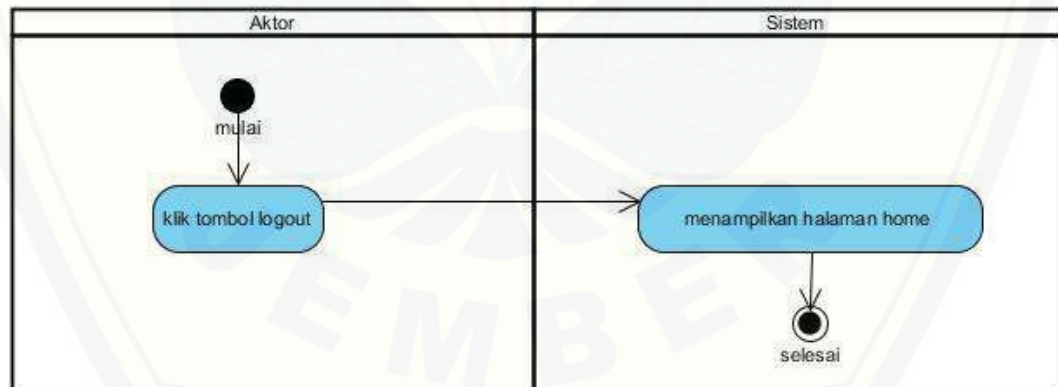
B.14 Activity Diagram Melihat Record Pemeriksaan



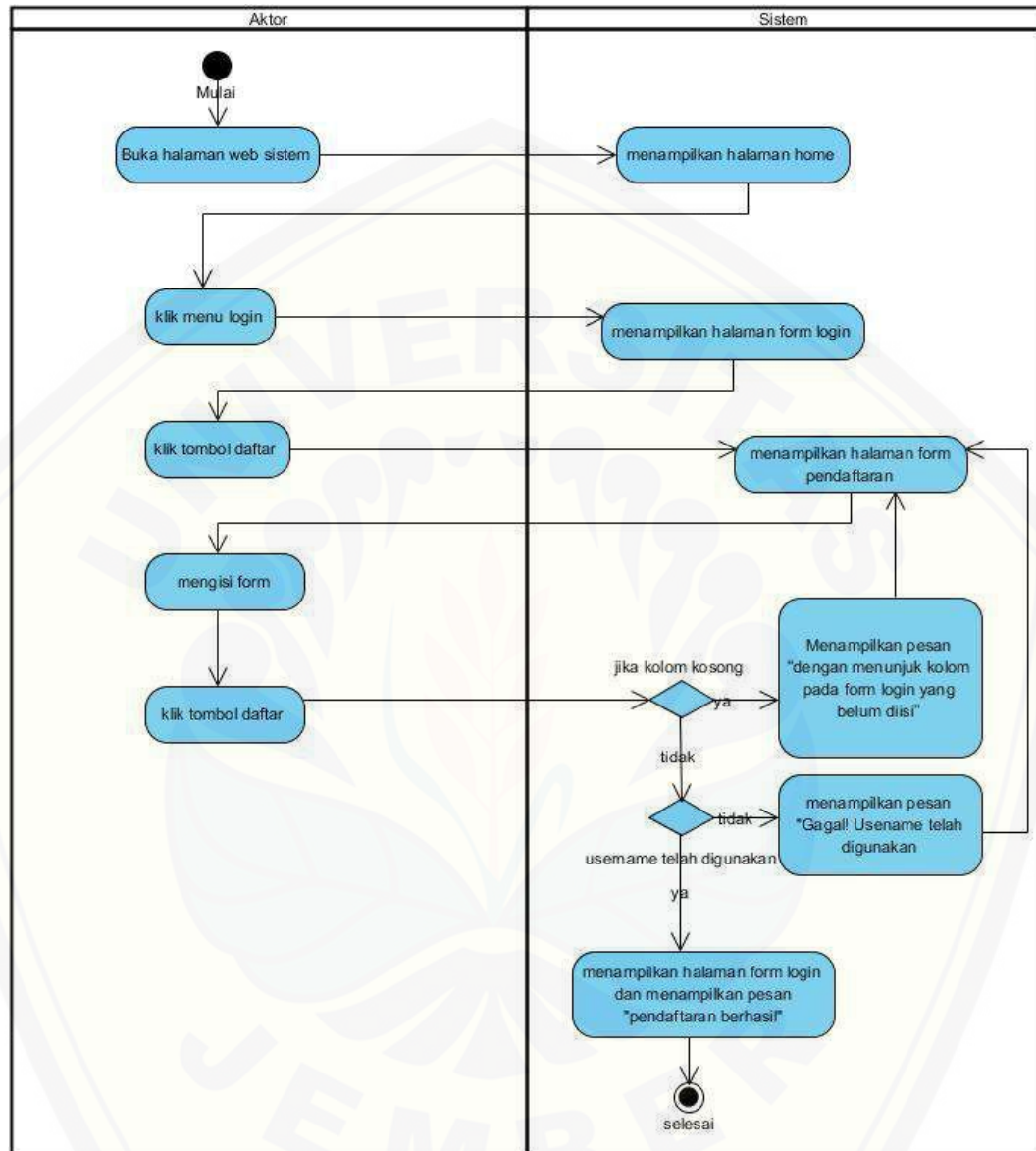
Gambar B.14 Activity diagram melihat record pemeriksaan

B.15 Activity Diagram Melihat Detail *Record* PemeriksaanGambar B.15 Activity diagram melihat detail *record* pemeriksaan

B.16 Activity Diagram Logout

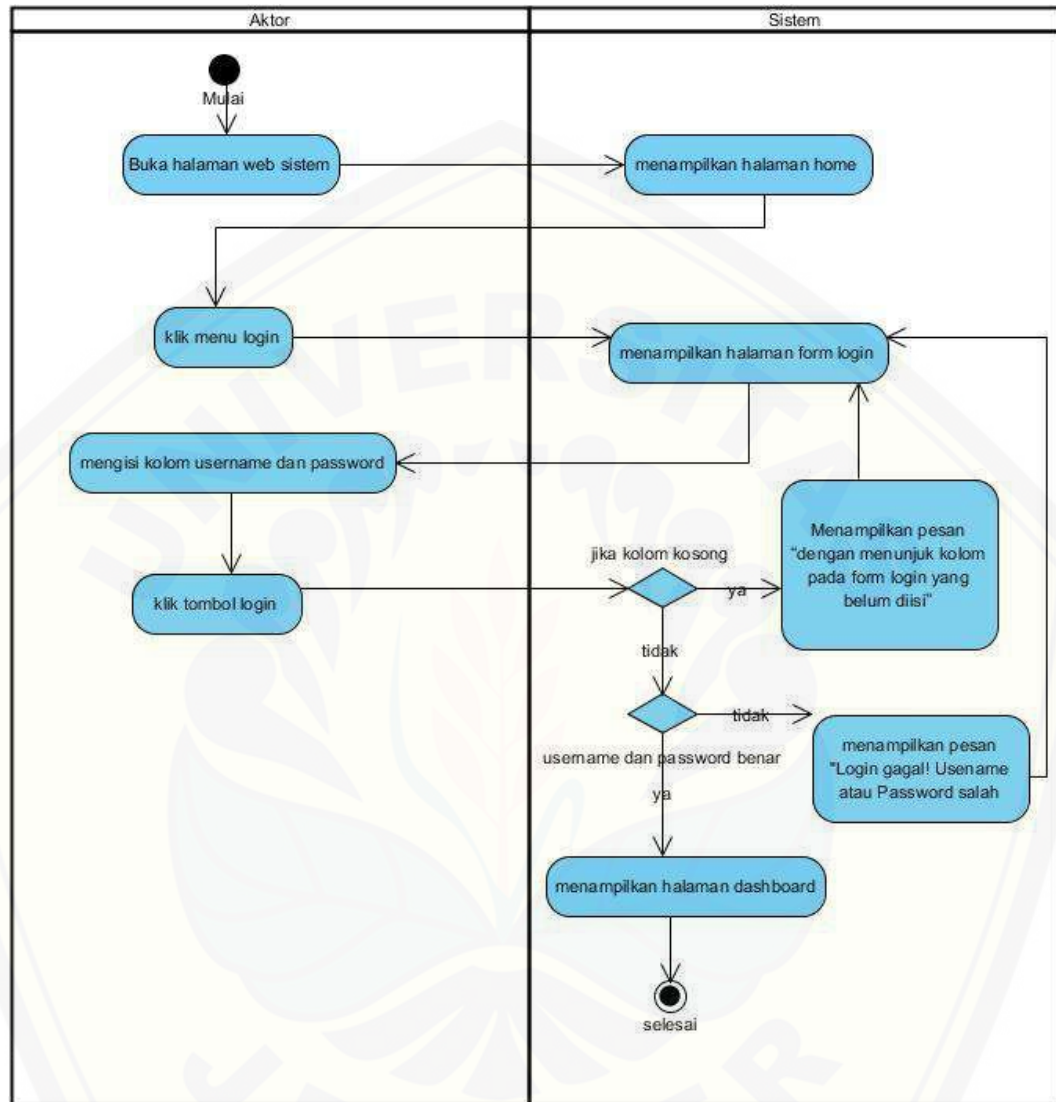
Gambar B.16 Activity diagram *logout*

B.17 Activity Diagram Mendaftar Anggota



Gambar B.17 Activity diagram Mendaftar Anggota

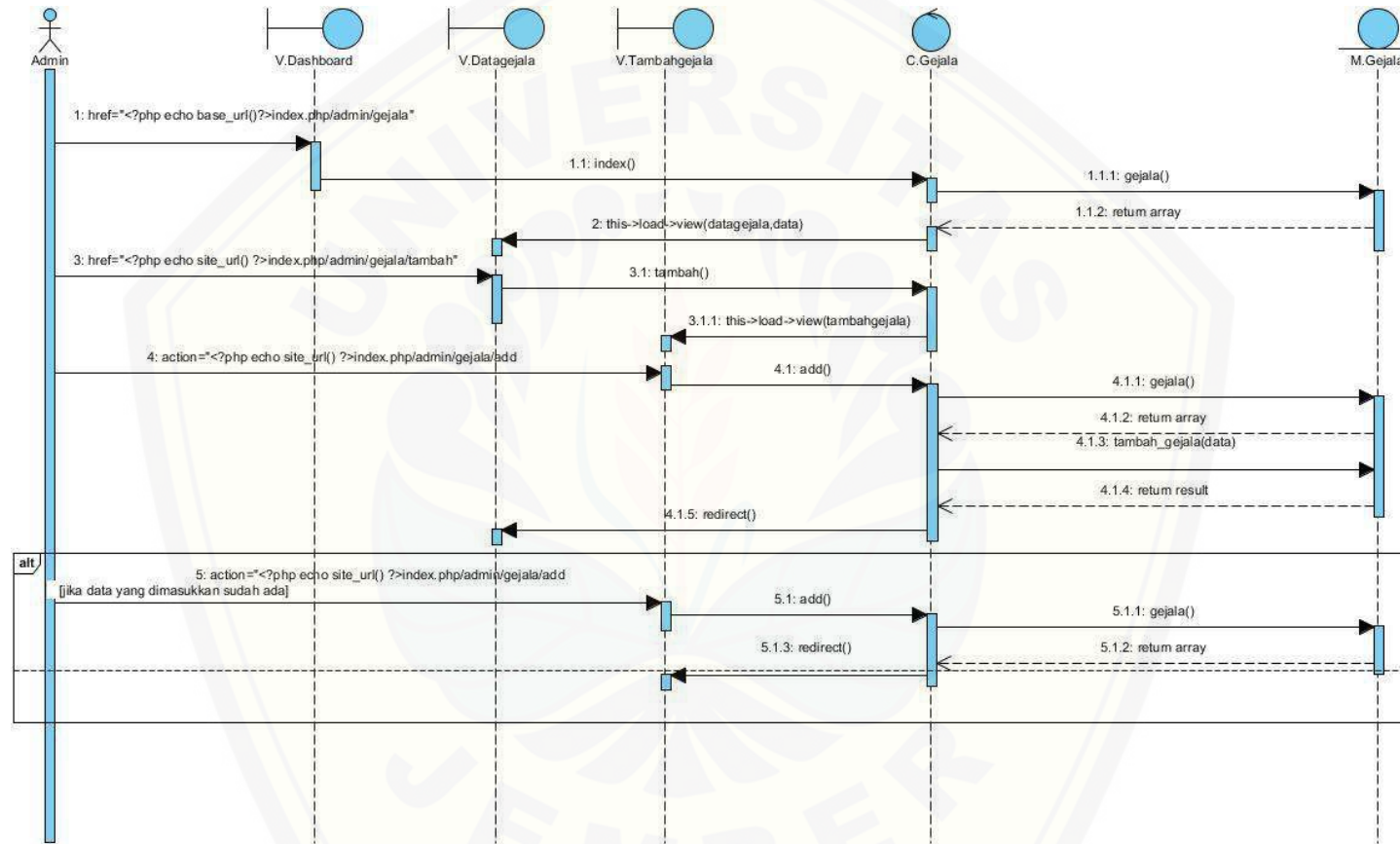
B.18 Activity Diagram Login



Gambar B.18 Activity diagram login

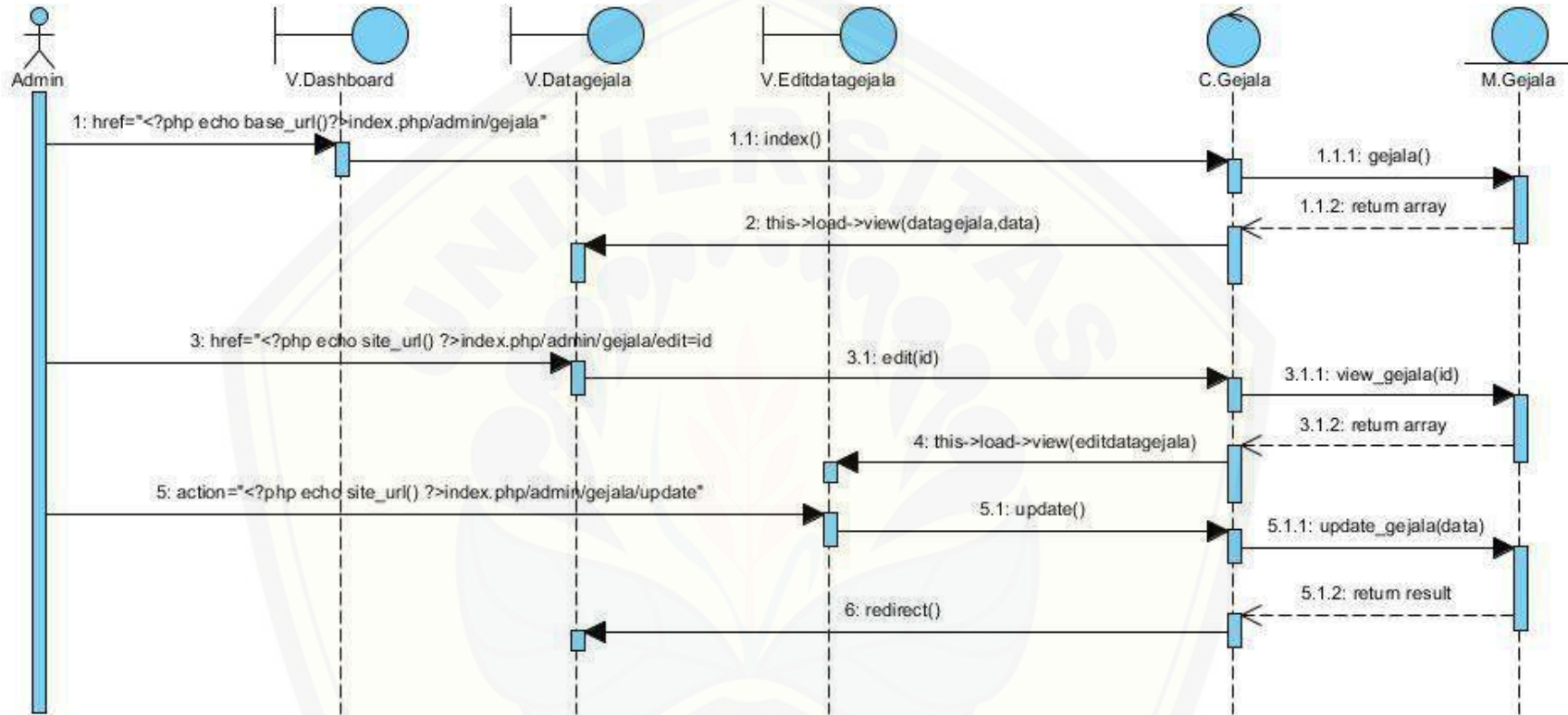
Lampiran C. Sequence Diagram

C.1 Sequence Diagram Menambah Data Gejala



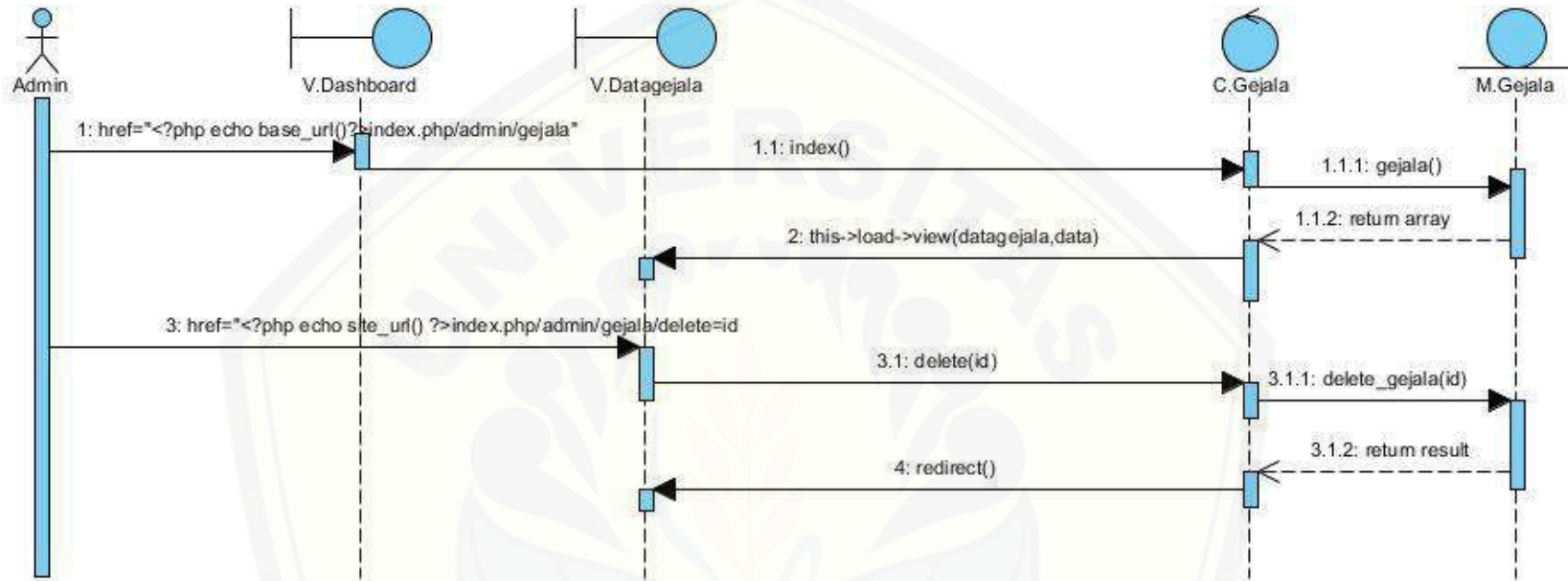
Gambar C.1 Sequence diagram menambah data gejala

C.2 Sequence Diagram Mengubah Data Gejala



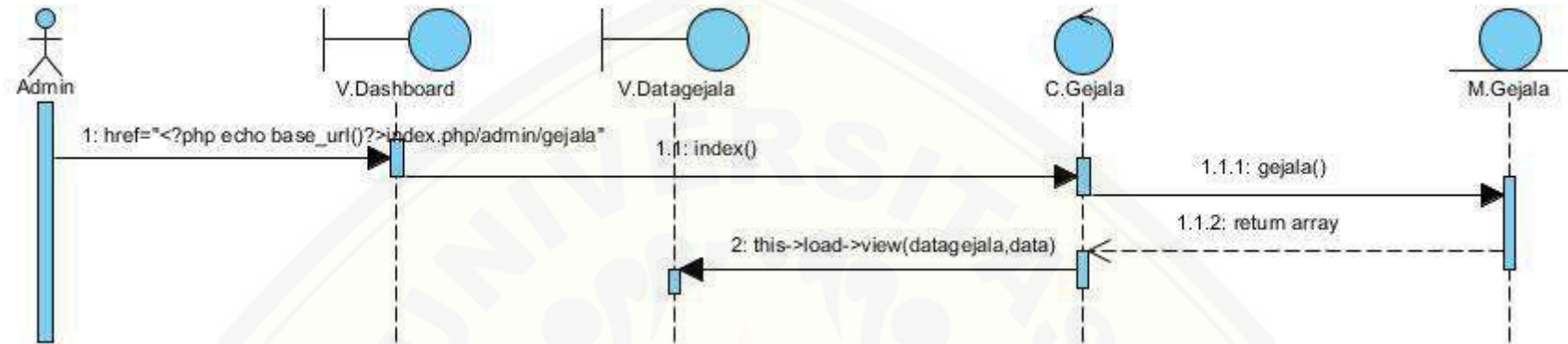
Gambar C.2 Sequence diagram mengubah data gejala

C.3 Sequence Diagram Menghapus Data Gejala



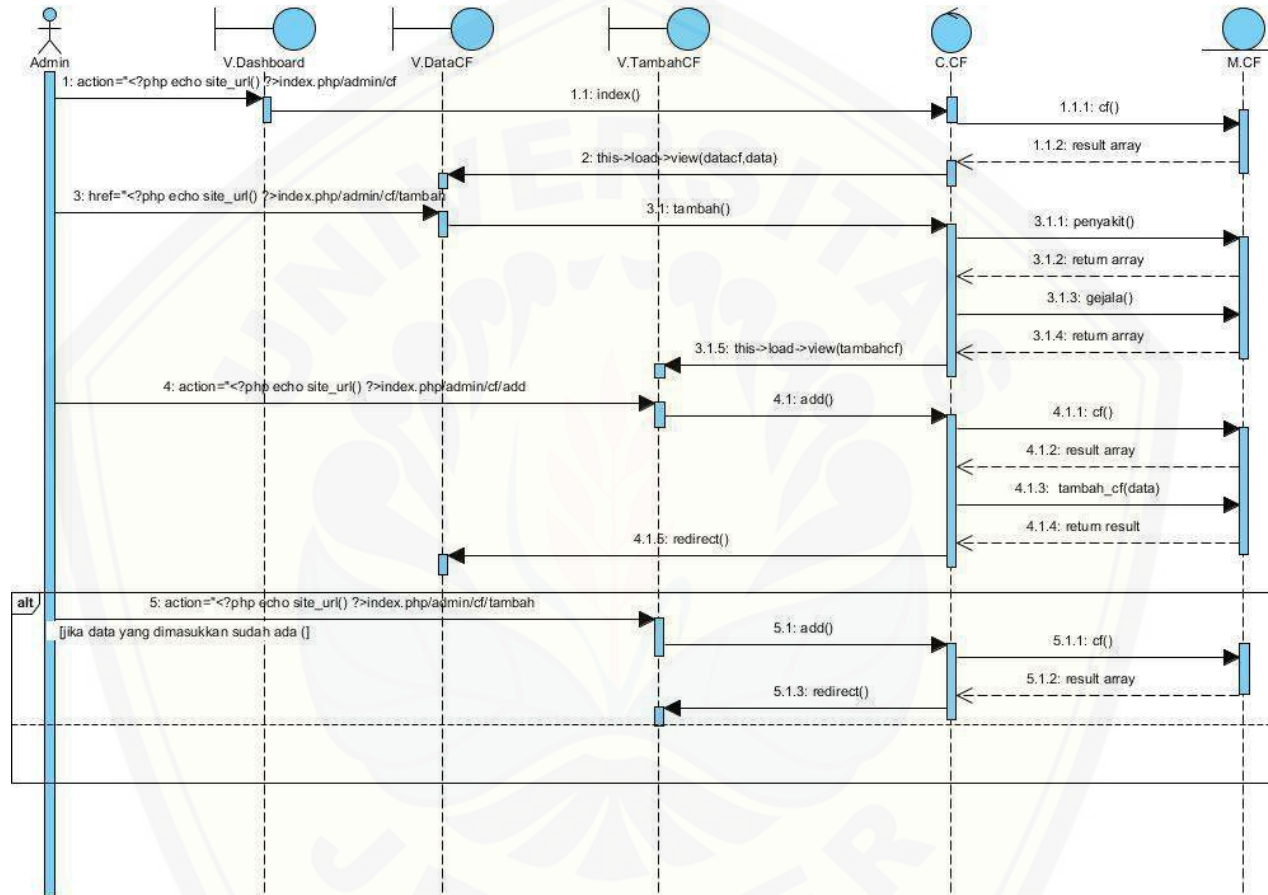
Gambar C.3 Sequence diagram menghapus data gejala

C.4 Sequence Diagram Melihat Data Gejala



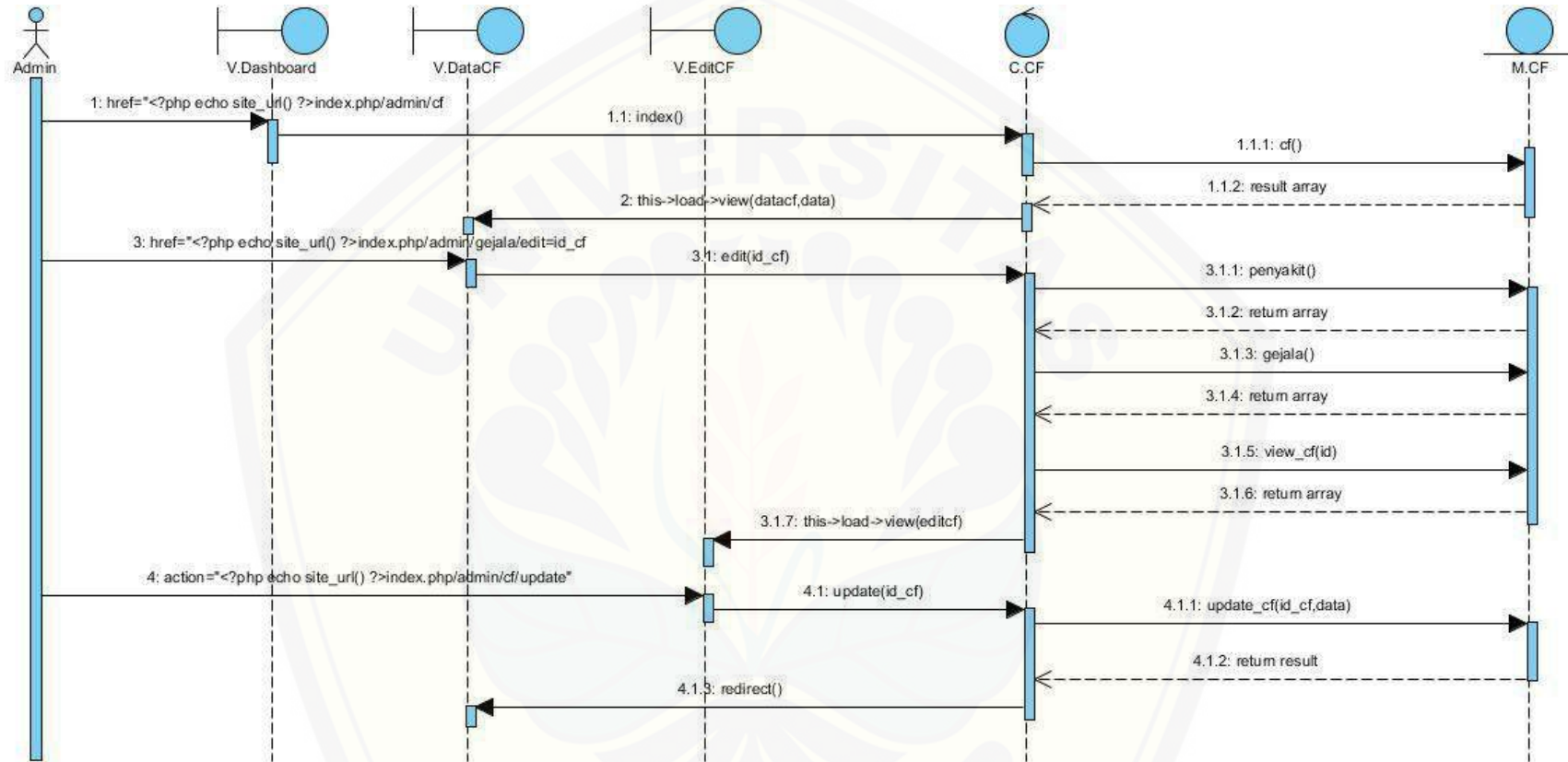
Gambar C.4 Sequence diagram melihat data gejala

C.5 Sequence Diagram Menambah Data CF



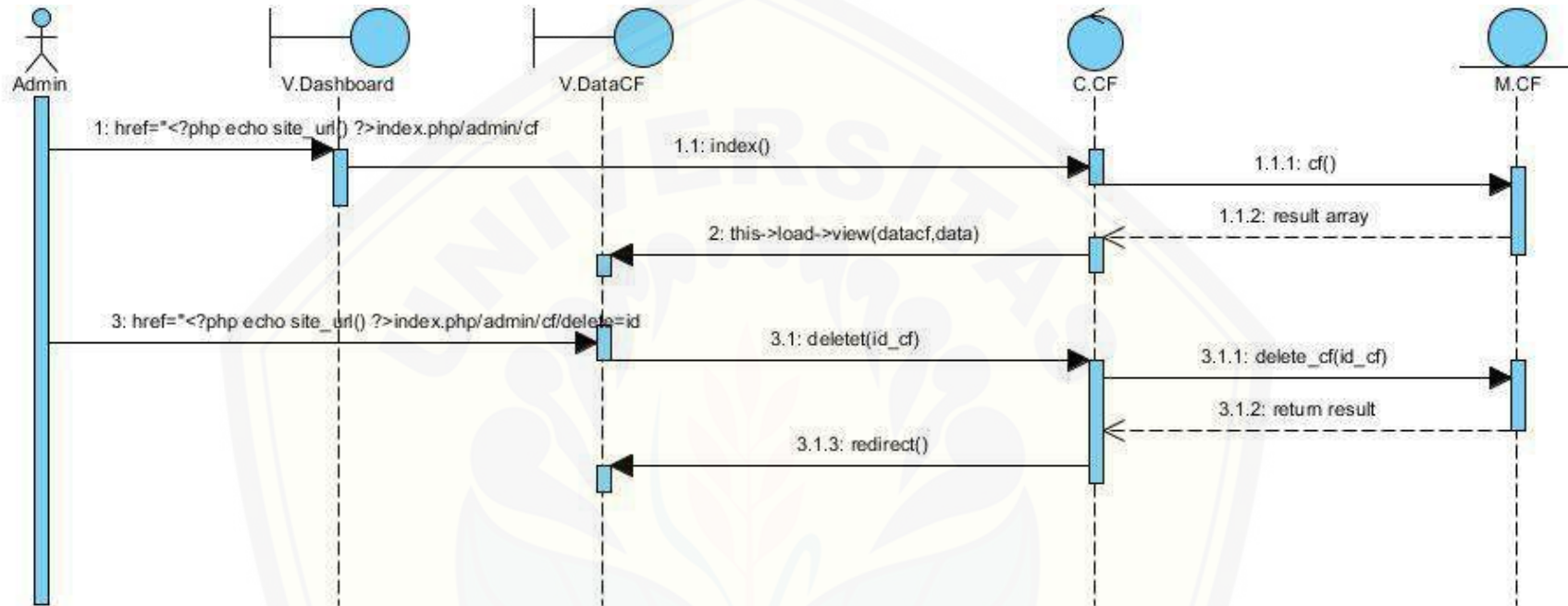
Gambar C.5 Sequence diagram menambah data CF

C.6 Sequence Diagram Mengubah Data CF



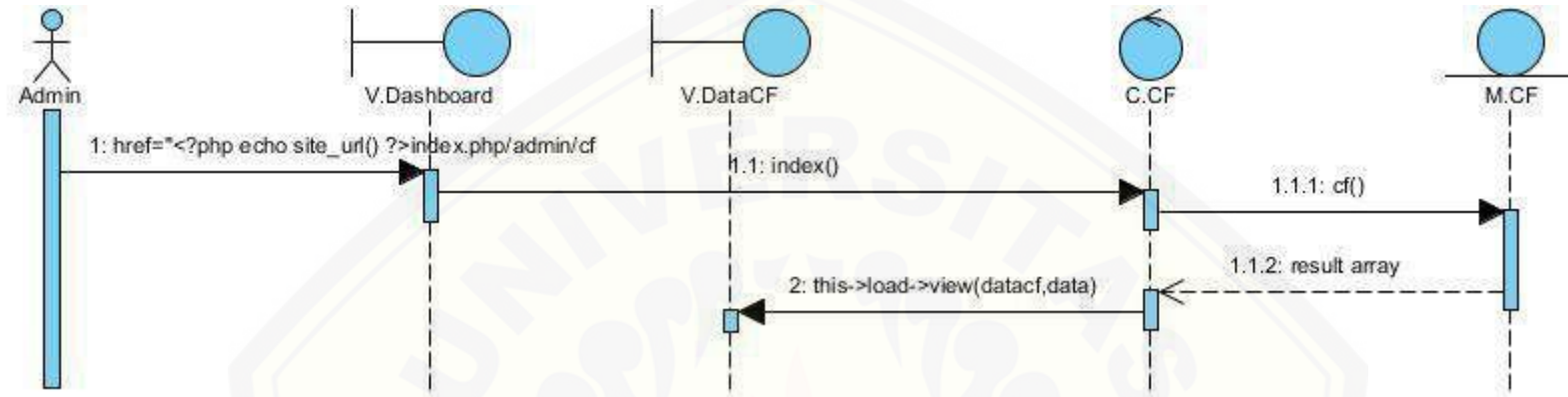
Gambar C.6 Sequence diagram mengubah data CF

C.7 Sequence Diagram Menghapus Data CF



Gambar C.7 Sequence diagram menghapus data CF

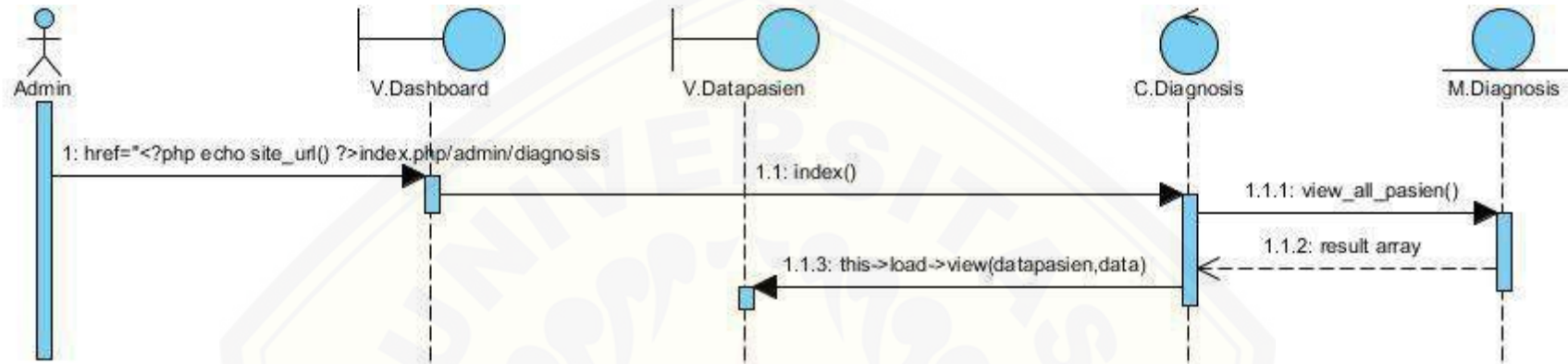
C.8 Sequence Diagram Melihat Data CF



C.8 Sequence diagram melihat data CF

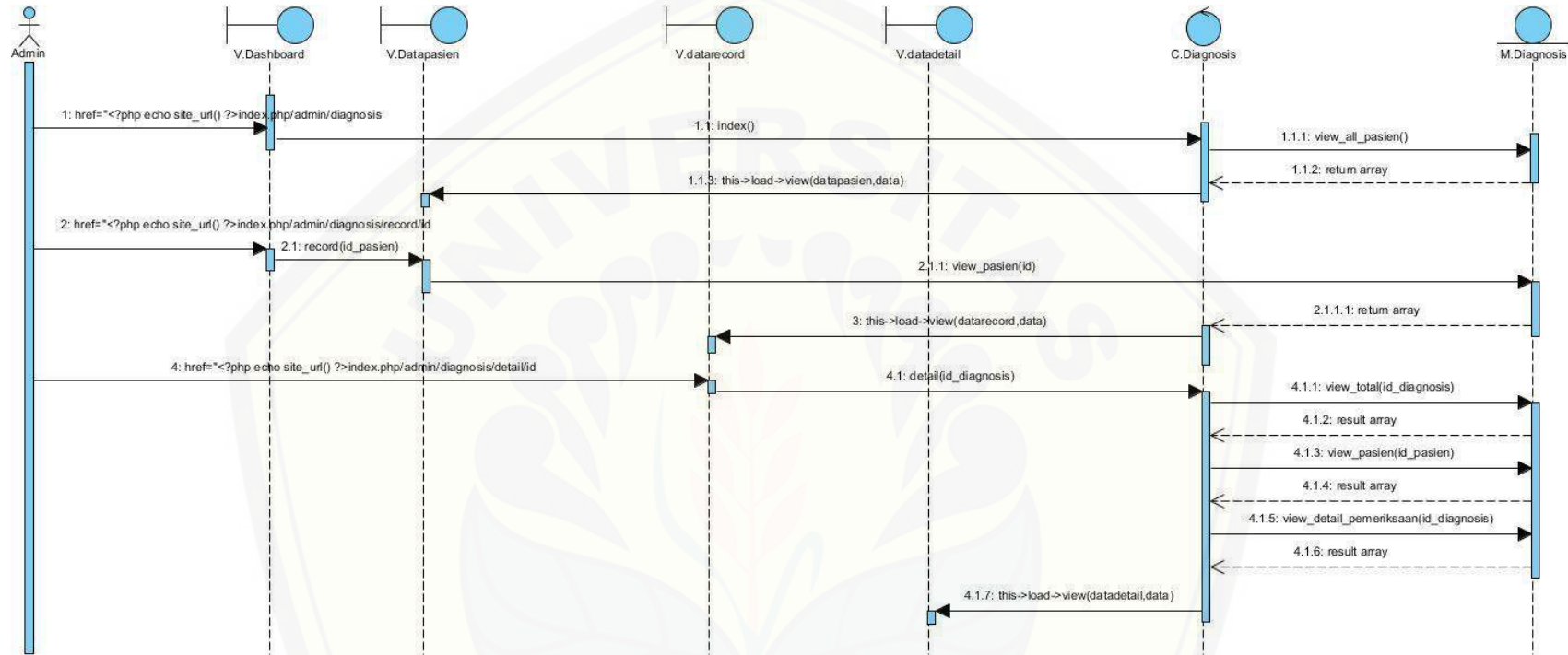
Gambar

C.9 Sequence Diagram Melihat Rekap Data Pasien



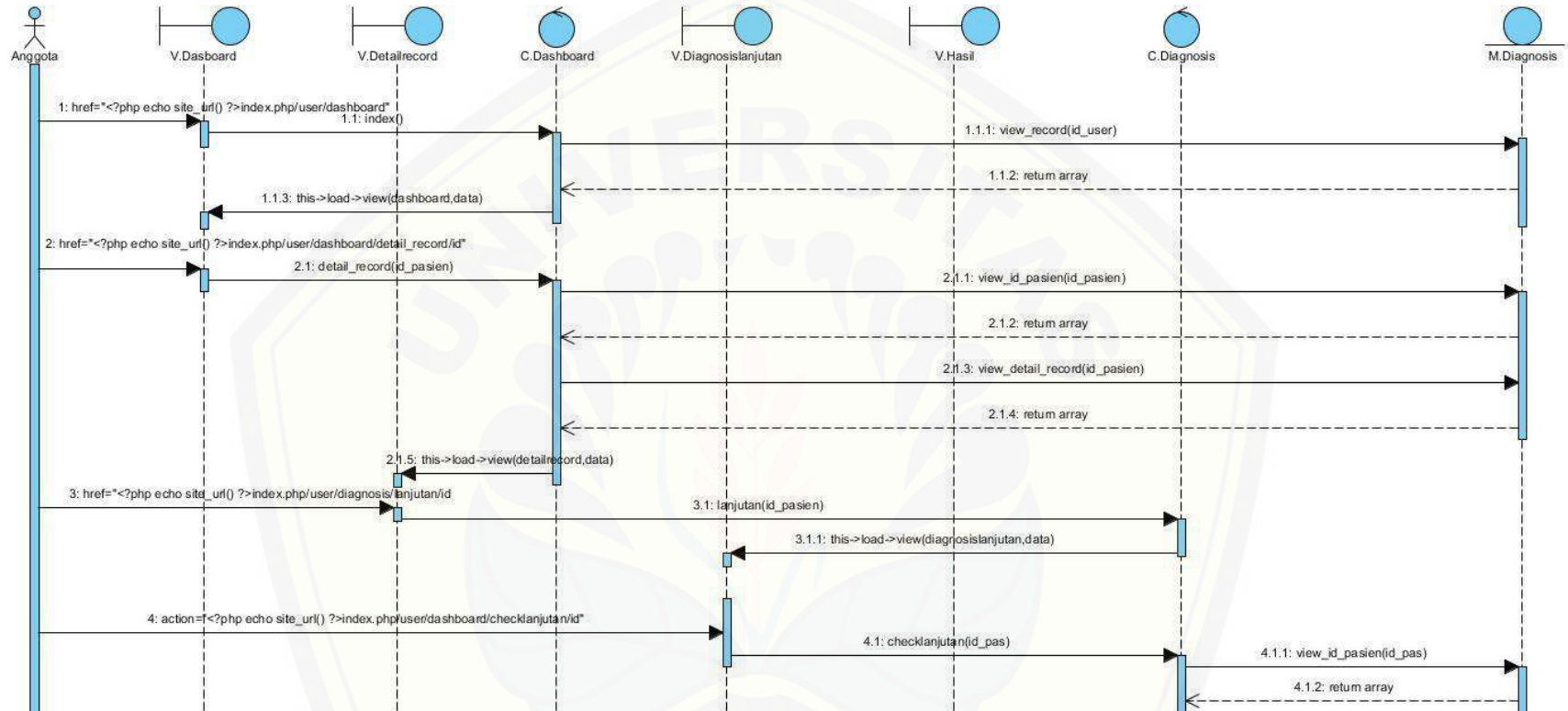
Gambar C.9 Sequence diagram melihat rekap data pasien

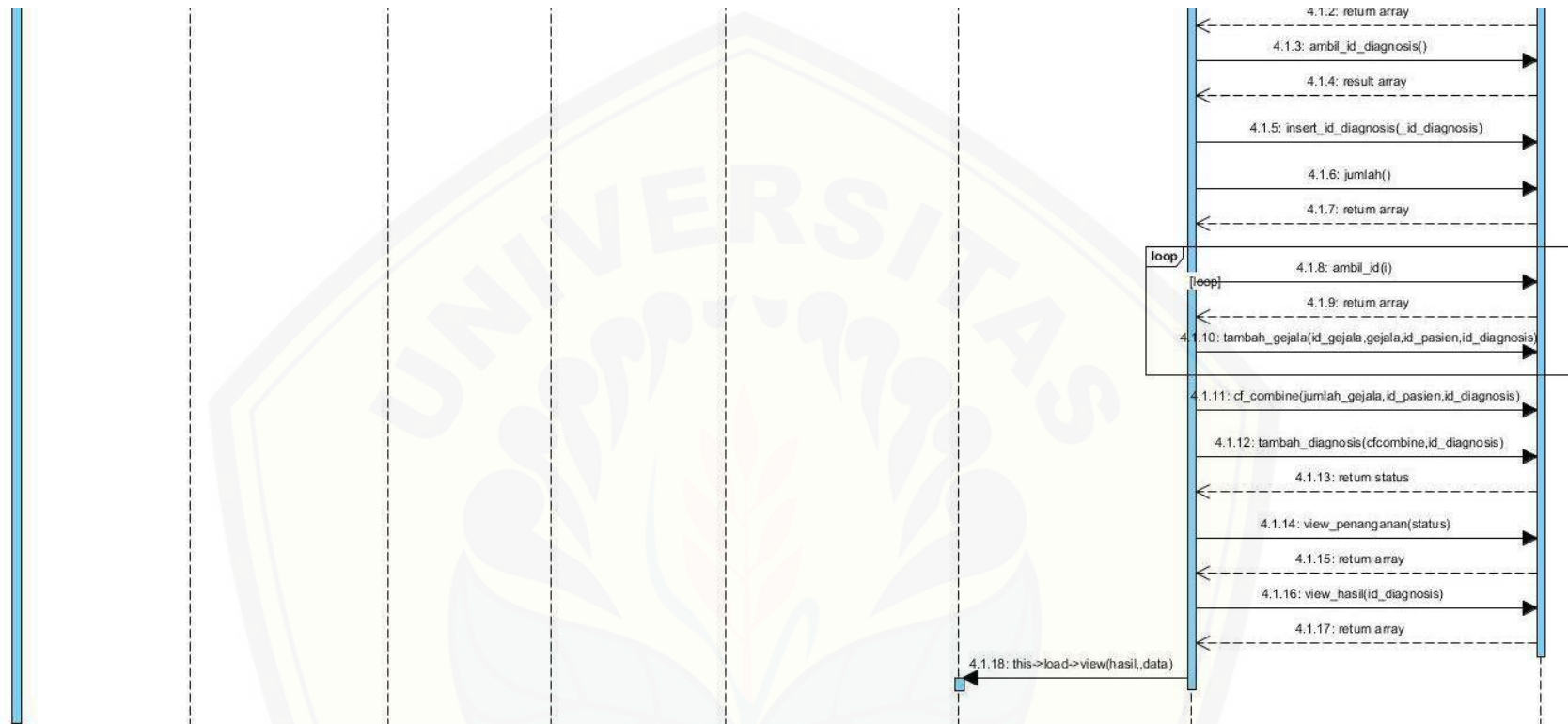
C.10 Sequence Diagram Melihat Data Diagnosis



Gambar C.10 Sequence diagram melihat data diagnosis

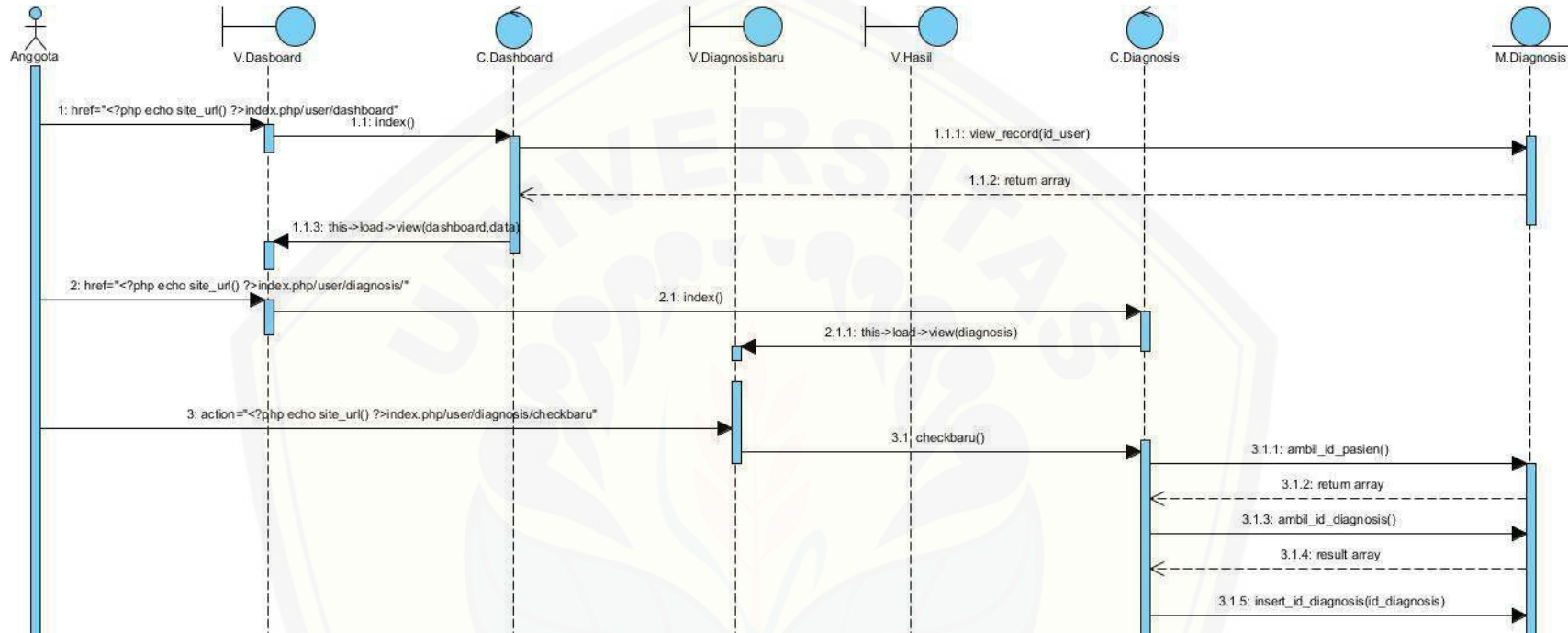
C.11 Sequence Diagram Memasukkan Pemeriksaan Pasien Lama

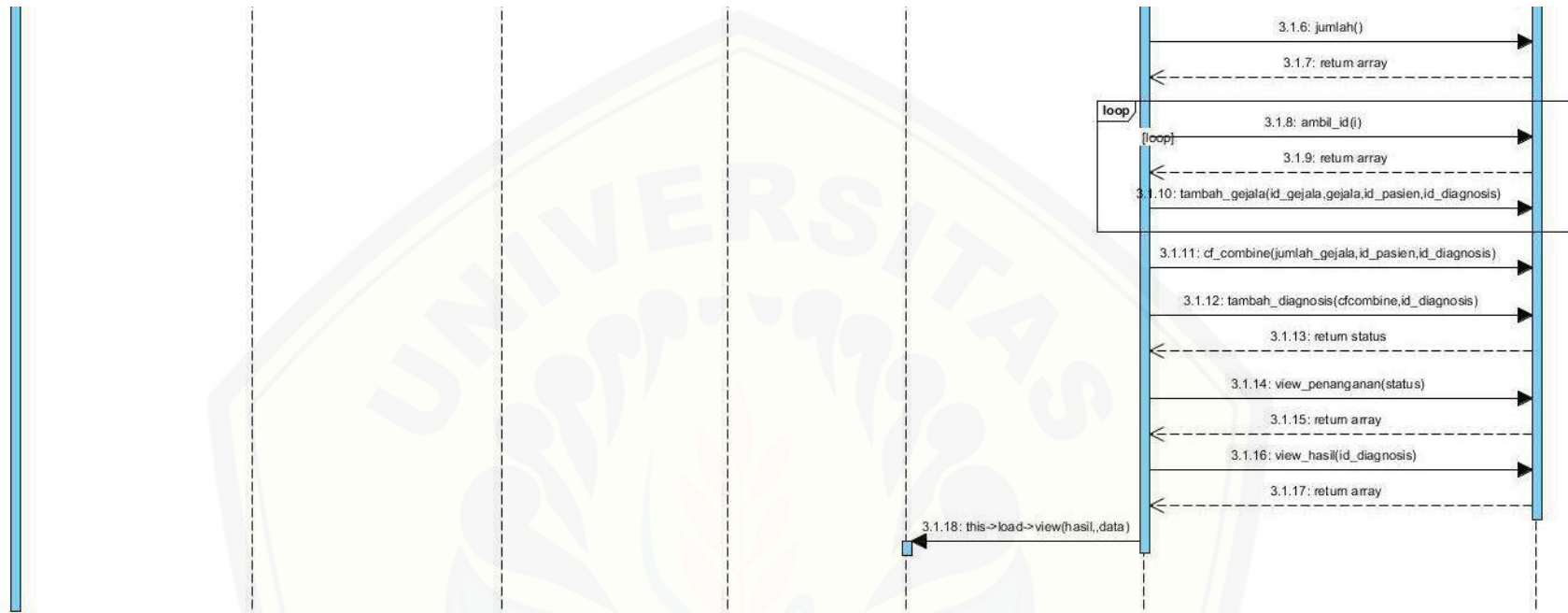




Gambar C.11 *Sequence* diagram memasukkan pemeriksaan pasien lama

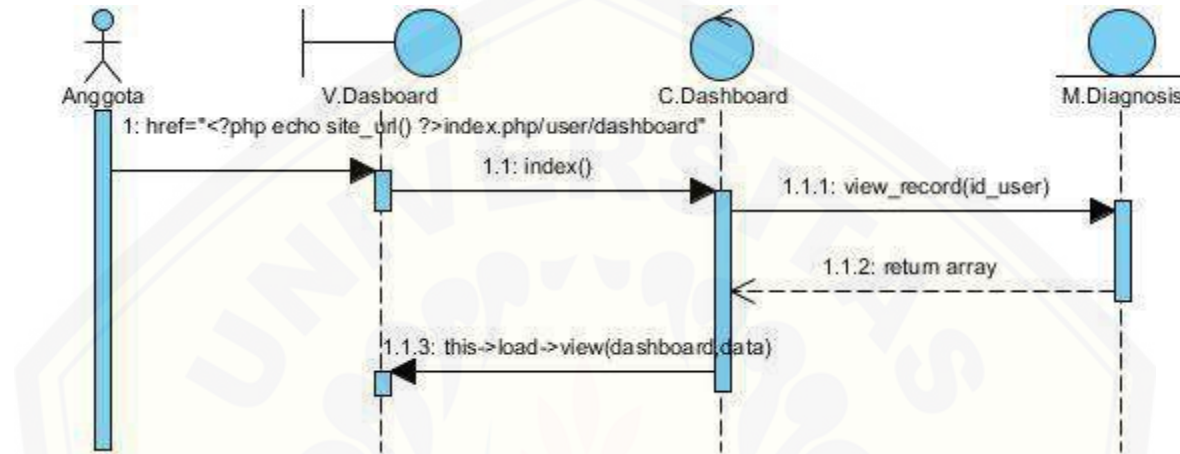
C.12 Sequence Diagram Memasukkan Pemeriksaan Pasien Baru





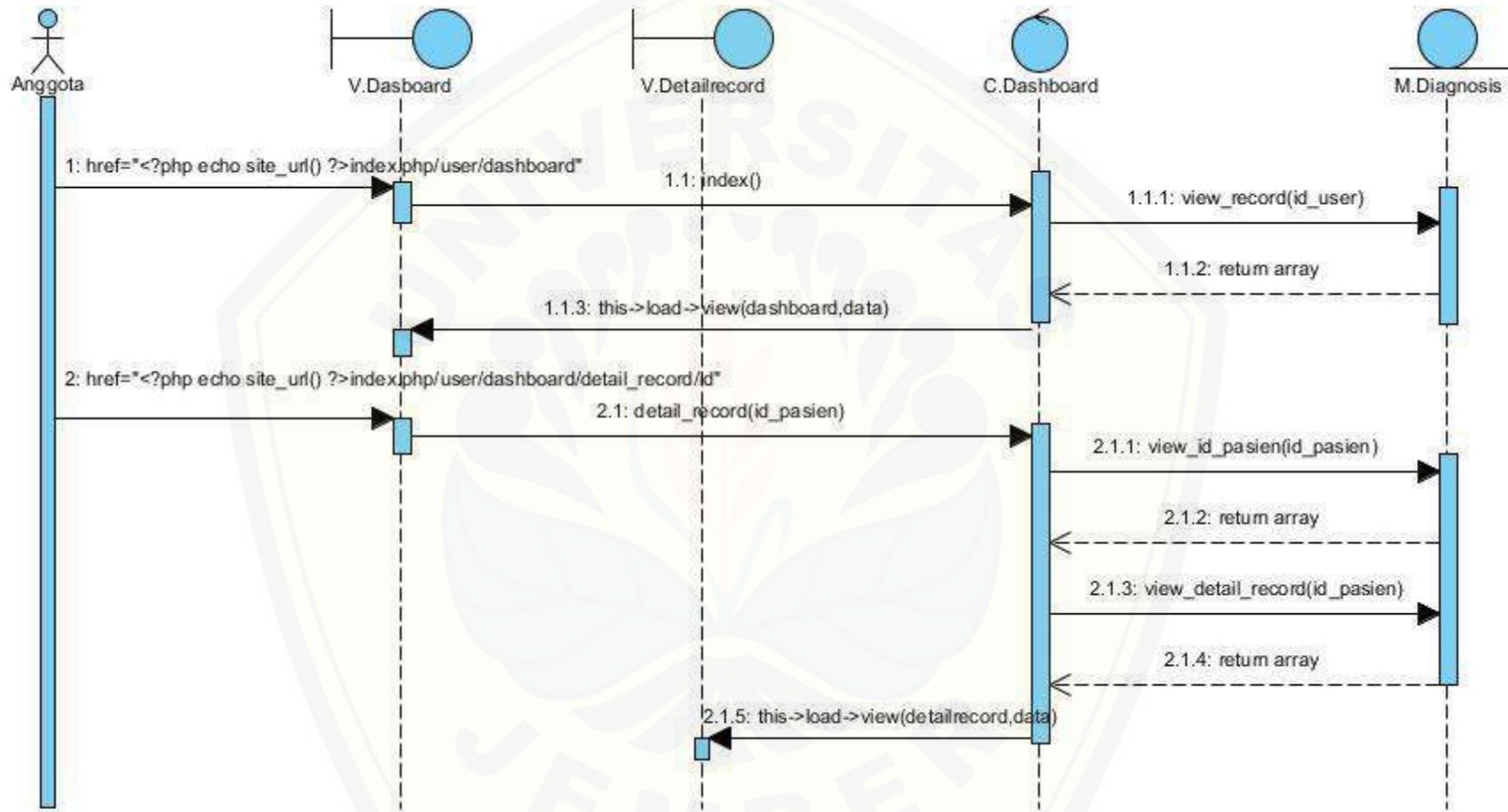
Gambar C.12 *Sequence* diagram memasukkan pemeriksaan pasien baru

C.13 *Sequence Diagram* Melihat Data Pasien Anggota



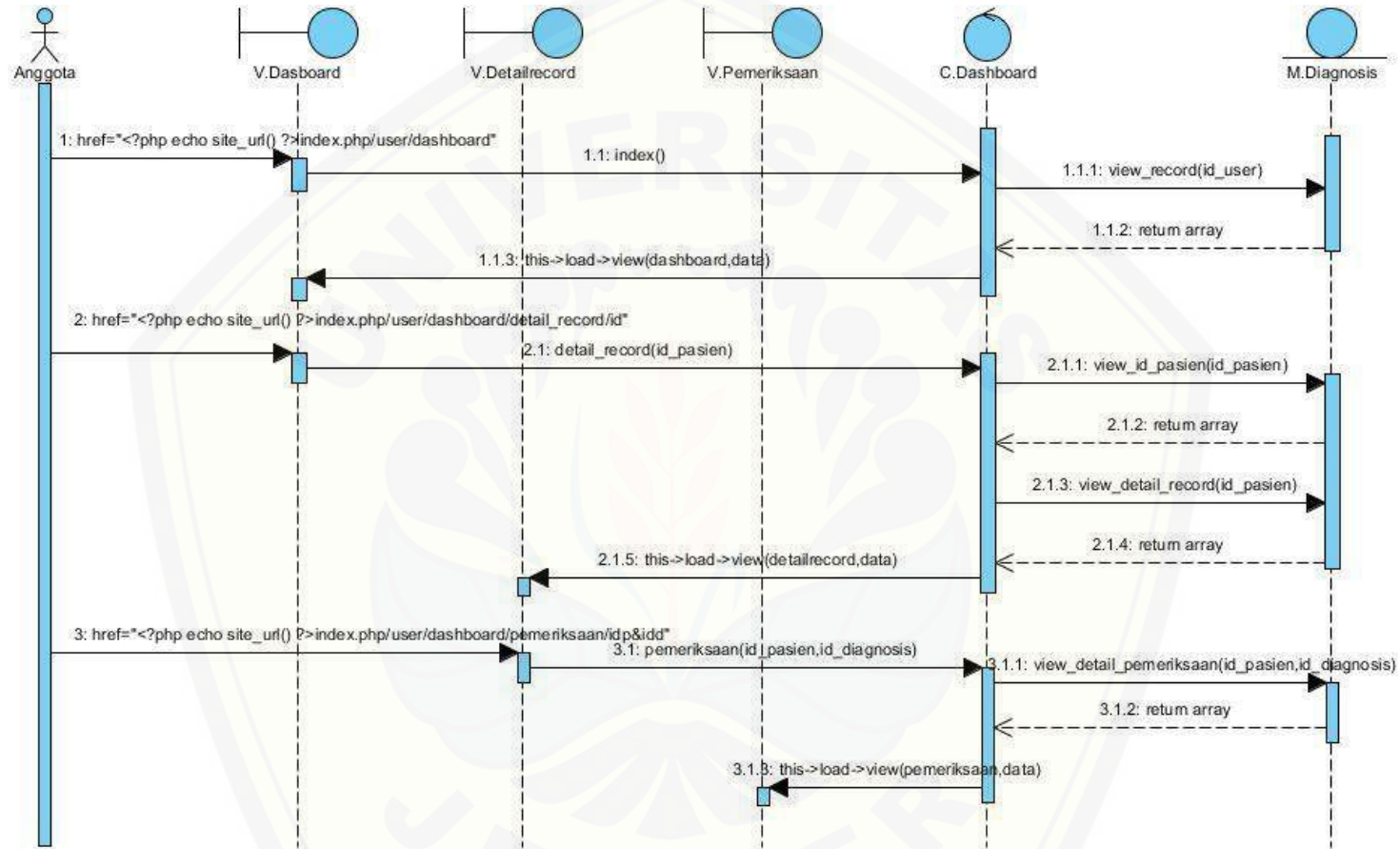
Gambar C.13 *Sequence diagram* melihat data pasien anggota

C.14 Sequence Diagram Melihat Record Pemeriksaan



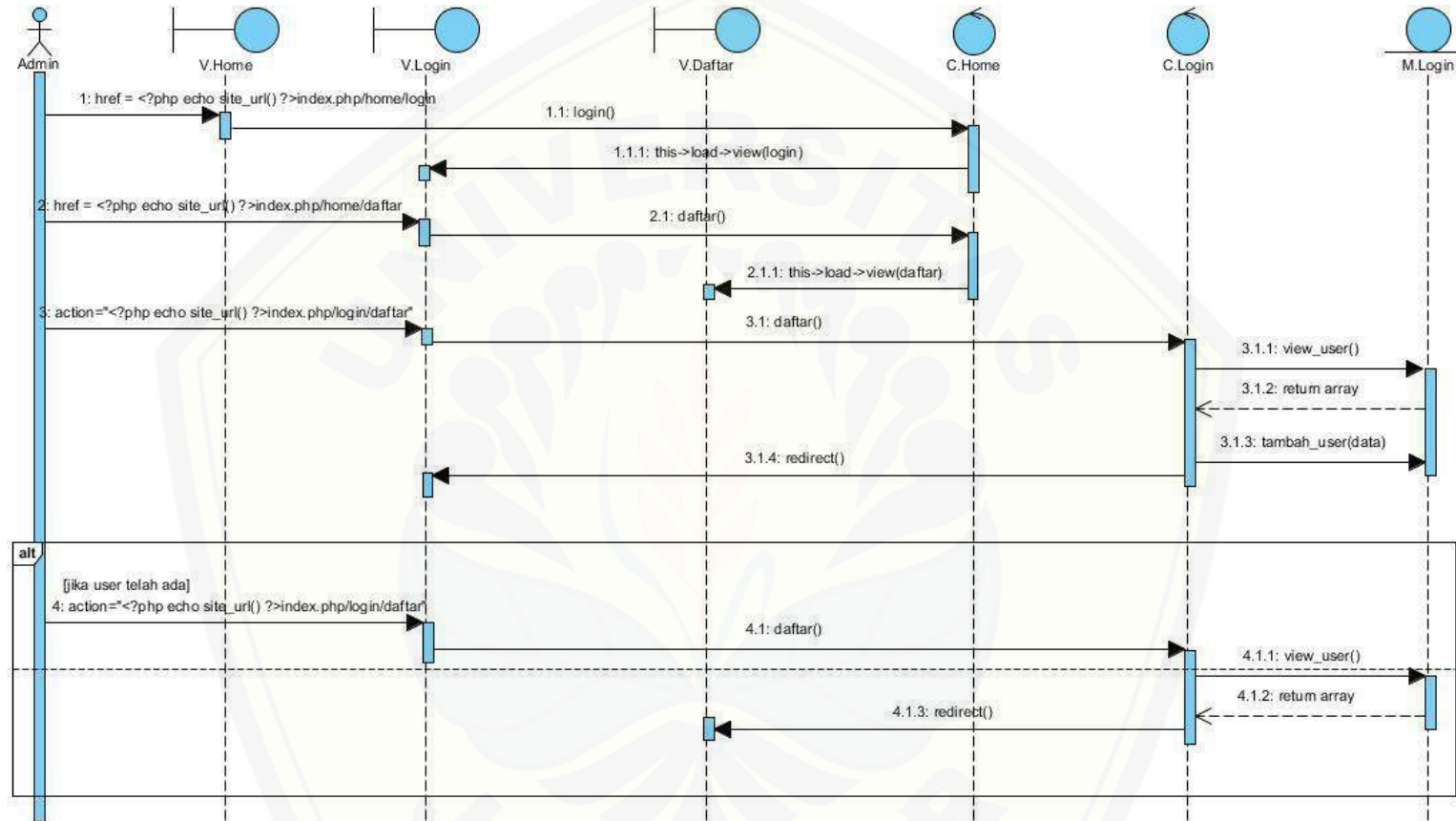
Gambar C.14 Sequence diagram melihat record pemeriksaan

C.15 Sequence Diagram Detail Melihat Detail Record Pemeriksaan



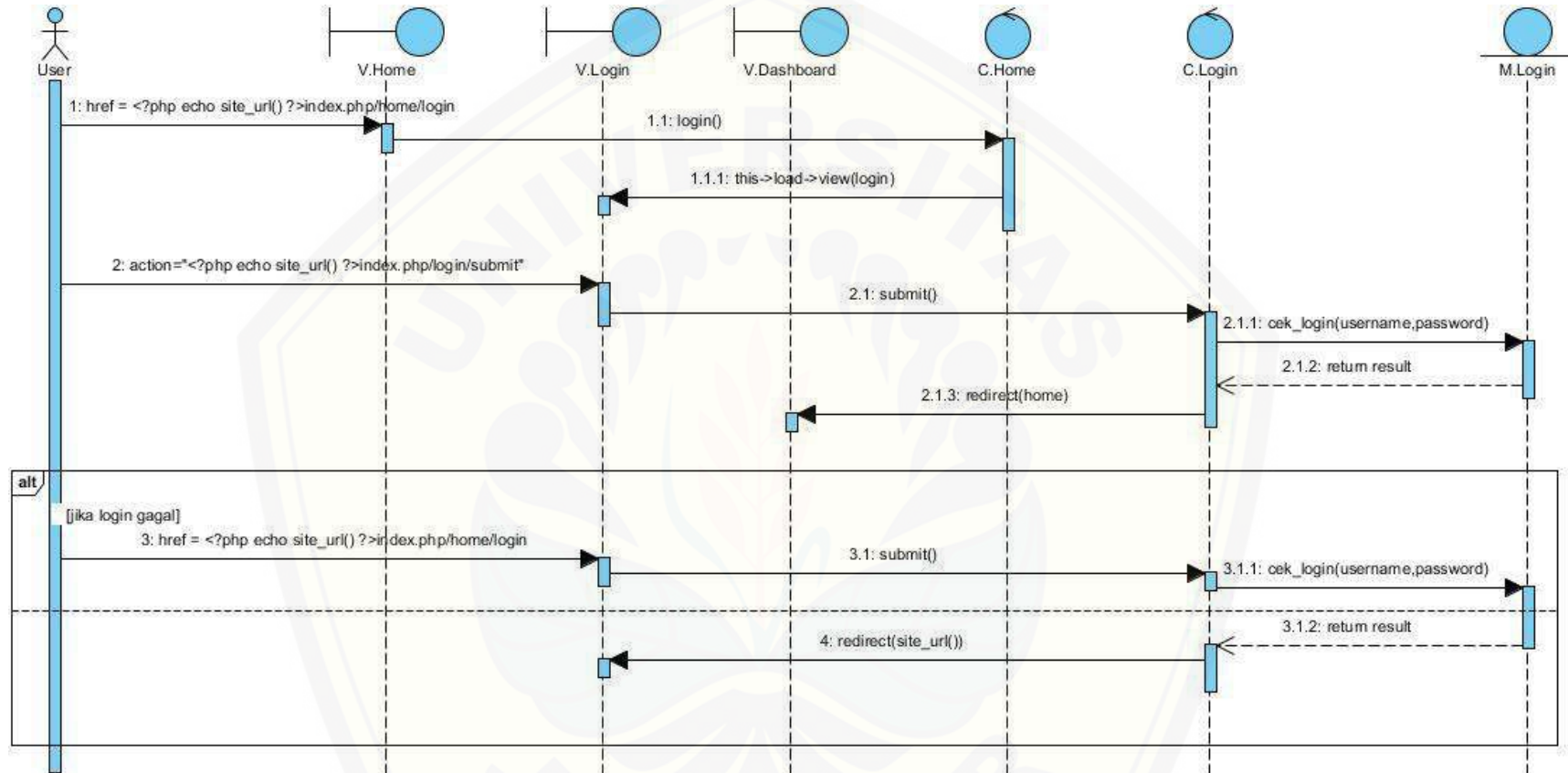
Gambar C.15 Sequence diagram melihat detail record pemeriksaan

C.16 Sequence Diagram Mendaftar Anggota



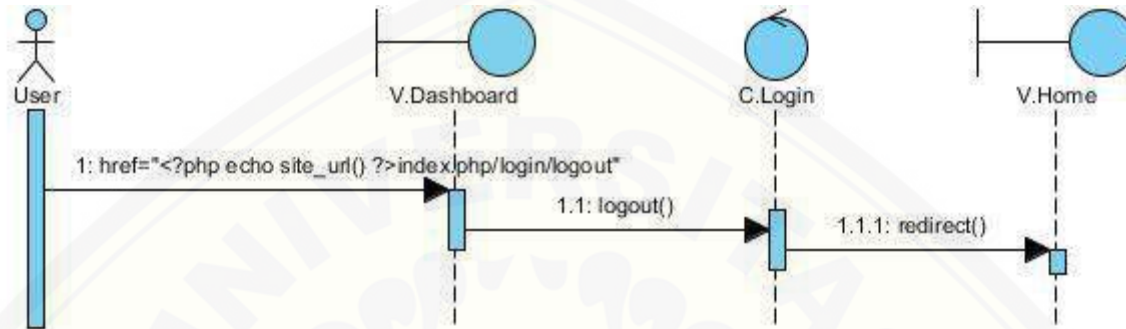
Gambar C.16 Sequence diagram mendaftar anggota

C.17 Sequence Diagram login



Gambar C.17 Sequence diagram login

C.18 *Sequence Diagram logout*



Gambar C.18 *Sequence diagram logout*

Lampiran D. Kode Program

D.1. Kode program class controller/user/diagnosis

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Diagnosis extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->helper('url');
        $this->load->model('user/m_diagnosis');
        $isLoggedIn = $this->session->userdata('level');
        if($isLoggedIn == null || $isLoggedIn != "2"){
            redirect("/");
        }
    }

    public function index() {
        $id_pasien = $this->session->userdata('id_pasien');
        $data['pertanyaan'] = $this->m_diagnosis->pertanyaan();
        $data['username'] = $this->session->userdata('username');
        $this->load->view('user/header', $data);
        $this->load->view('user/diagnosisbaru', $data);
        $this->load->view('user/footer');
    }

    public function checkbaru() {
        $id_user = $this->session->userdata('id_user');
        $data['nama'] = $this->input->post('nama');
        $data['umur'] = $this->input->post('umur');

        $id_pasien = $this->m_diagnosis->ambil_id_pasien();
        $this->m_diagnosis->insert_pasien($id_pasien, $data, $id_user);

        $id_diagnosis = $this->m_diagnosis->ambil_id_diagnosis();
        $this->m_diagnosis->insert_id_diagnosis($id_diagnosis);

        $jumlah = $this->m_diagnosis->jumlah();
        foreach ($jumlah->result_array() as $row) {
            $jumlah_gejala = $row['jumlah'];
        }

        for ($i=0; $i < $jumlah_gejala ; $i++) {
            $ambil2 = $this->m_diagnosis->ambil_id($i);
            foreach ($ambil2->result_array() as $p) {
                $gejala = $this->input->post($p['id_gejala']);
                $this->m_diagnosis->tambah_gejala($p['id_gejala'], $gejala, $id_pasien, $id_diagnosis);
            }
        }

        $CFCombine = $this->m_diagnosis->cfcombine($jumlah_gejala, $id_pasien, $id_diagnosis);
        $status = $this->m_diagnosis->tambah_diagnosis($CFCombine, $id_diagnosis);
    }
}
```

```
$pasien['penanganan'] = $this->m_diagnosis->view_penanganan($status);
$pasien['pasien'] = $this->m_diagnosis->view_hasil($id_diagnosis);
$data['username'] = $this->session->userdata('username');
$this->load->view('user/header',$data);
$this->load->view('user/hasil', $pasien);
$this->load->view('user/footer');

}

public function lanjutan($id_pasien) {
    $data['id_pasien'] = $this->m_diagnosis->view_id_pasien($id_pasien);
    $data['pertanyaan'] = $this->m_diagnosis->pertanyaan();
    $data['username'] = $this->session->userdata('username');
    $this->load->view('user/header', $data);
    $this->load->view('user/diagnosislanjutan', $data);
    $this->load->view('user/footer');
}

public function checklanjutan($id_pas) {
    $id_user = $this->session->userdata('id_user');
    $idp = $this->m_diagnosis->view_id_pasien($id_pas);
    foreach ($idp->result_array() as $row) {
        $id_pasien = $row['id_pasien'];
    }
    $id_diagnosis = $this->m_diagnosis->ambil_id_diagnosis();
    $this->m_diagnosis->insert_id_diagnosis($id_diagnosis);

    $jumlah = $this->m_diagnosis->jumlah();
    foreach ($jumlah->result_array() as $row) {
        $jumlah_gejala = $row['jumlah'];
    }

    for ($i=0; $i < $jumlah_gejala ; $i++) {
        $ambil2 = $this->m_diagnosis->ambil_id($i);
        foreach ($ambil2->result_array() as $p) {
            $gejala = $this->input->post($p['id_gejala']);
            $this->m_diagnosis->tambah_gejala($p['id_gejala'], $gejala, $id_pasien, $id_diagnosis);
        }
    }
}

$CFCombine = $this->m_diagnosis->cfcombine($jumlah_gejala, $id_pasien, $id_diagnosis);
$status = $this->m_diagnosis->tambah_diagnosis($CFCombine, $id_diagnosis);
$pasien['penanganan'] = $this->m_diagnosis->view_penanganan($status);
$pasien['pasien'] = $this->m_diagnosis->view_hasil($id_diagnosis);
$data['username'] = $this->session->userdata('username');
$this->load->view('user/header',$data);
$this->load->view('user/hasil', $pasien);
$this->load->view('user/footer');
}
}
```

D.2. Kode program class model/user/m_diagnosis

```
<?php

class M_diagnosis extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    public function cek_tanggal($id_pasien) {
        $query = $this->db->query("select pm.tanggal, d.id_record,
            datediff(now(),min(pm.tanggal)) as selisih from tb_diagnosis d join tb_pemeriksaan pm
            on (pm.id_diagnosis = d.id_diagnosis) where pm.id_pasien = $id_pasien and id_record
            in (select max(id_record) from tb_diagnosis d join tb_pemeriksaan pm on
            (pm.id_diagnosis = d.id_diagnosis) where pm.id_pasien = $id_pasien)");
        return $query;
    }

    public function cek_tanggal_lama($id_record) {
        $query = $this->db->query("select datediff(now(),min(tanggal)) as selisih from
            tb_pemeriksaan p join tb_diagnosis d on (p.id_diagnosis = p.id_diagnosis) where
            id_record = $id_record order by tanggal desc limit 1");
        return $query;
    }

    public function pertanyaan() {
        $query = $this->db->query("SELECT * FROM tb_gejala order by id_gejala asc");
        return $query;
    }

    public function jumlah() {
        $query = $this->db->query("SELECT count(*) as jumlah FROM tb_gejala where jenis =
        'demam tifoid' order by id_gejala asc");
        return $query;
    }

    public function ambil_id($i) {
        $query = $this->db->query("SELECT * FROM tb_gejala where jenis = 'demam tifoid'
        order by id_gejala asc limit $i,1");
        return $query;
    }

    public function view_hasil($id_diagnosis) {
        $query = $this->db->query("SELECT left(d.total,4) as t, d.diagnosis, ps.nama FROM
        tb_pasien ps join tb_pemeriksaan pm on (ps.id_pasien = pm.id_pasien) join tb_diagnosis d on
        (pm.id_diagnosis = d.id_diagnosis) where d.id_diagnosis = $id_diagnosis group by
        d.id_diagnosis");
        return $query;
    }
}
```

```
public function view_penanganan($status) {
    $query = $this->db->query("SELECT * FROM tb_penanganan where tingkat = '$status'");
    return $query;
}

public function ambil_id_diagnosis() {
    $reqid = $this->db->query("SELECT MAX(id_diagnosis) as id_diagnosis FROM
tb_pemeriksaan");
    foreach ($reqid->result_array() as $row) {
        $id_diag = $row['id_diagnosis'];
    }
    $id = $id_diag + 1;
    return $id;
}

public function ambil_id_pasien() {
    $reqid = $this->db->query("SELECT MAX(id_pasien) as id_pasien FROM tb_pasien");
    foreach ($reqid->result_array() as $row) {
        $id_pass = $row['id_pasien'];
    }
    $id_pasien = $id_pass + 1;
    return $id_pasien;
}

public function view_id_pasien($id_pasien) {
    $reqid = $this->db->query("SELECT id_pasien FROM tb_pasien where id_pasien =
$id_pasien");
    return $reqid;
}

public function view_status($id_diagnosis) { /* Load Data golongan Semua */
    $reqid = $this->db->query("SELECT diagnosis FROM tb_diagnosis where id_diagnosis =
$id_diagnosis");
    return $reqid;
}

public function insert_id_diagnosis($id_diagnosis) { /* Load Data golongan Semua */
    $result = $this->db->query("insert into tb_diagnosis values ($id_diagnosis, '')");
    return $result;
}

public function insert_pasien($id_pasien,$data, $id_user) {
    $result = $this->db->query("insert into tb_pasien values
($id_pasien,'$data[nama]',$data[umur],$id_user)");
    return $result;
}
```

```

public function tambah_gejala($id_gejala,$gejala, $id_pasien, $id_diagnosis) {

    for ($i = 0; $i < count($gejala); $i++) {
        $jawaban = "";
        if($gejala[$i] == 0){
            $jawaban = "Tidak";
        }else if ($gejala[$i] == 0.2){
            $jawaban = "Tidak Tahu";
        }else if ($gejala[$i] == 0.4){
            $jawaban = "Sedikit Yakin";
        }else if ($gejala[$i] == 0.6){
            $jawaban = "Cukup Yakin";
        }else if ($gejala[$i] == 0.8){
            $jawaban = "Yakin";
        }else if ($gejala[$i] == 1){
            $jawaban = "Sangat Yakin";
        }
        $query1[$i] = $this->db->query("SELECT cf from tb_cf where id_gejala = '$id_gejala'");
        foreach ($query1[$i]->result_array() as $row) {
            $cf_pakar[$i] = $row['cf'];
        }
        $CFHE[$i] = $cf_pakar[$i] * $gejala[$i];
        $result = $this->db->query("insert into tb_pemeriksaan values ('', '$id_pasien', '$id_gejala', '$jawaban', '$gejala[$i]', $CFHE[$i], NOW(), $id_diagnosis)");
    }
    return $result;
}

public function cfcombine($jumlah_gejala, $id_pasien, $id_diagnosis) {
    $CFH1 = $this->db->query("SELECT kombinasi FROM tb_pemeriksaan where id_pasien = '$id_pasien' and id_diagnosis = '$id_diagnosis' LIMIT 0,1");
    foreach ($CFH1->result_array() as $row) {
        $CFCombine = $row['kombinasi'];
    }
    for ($i = 1; $i < $jumlah_gejala; $i++) {
        $query = $this->db->query("SELECT * FROM tb_gejala where jenis = 'demam tifoid' order by id_gejala asc limit $i,1");
        foreach ($query->result_array() as $row) {
            $id[$i] = $row['id_gejala'];
        }
        $result[$i] = $this->db->query("SELECT kombinasi FROM tb_pemeriksaan where id_pasien = '$id_pasien' and id_diagnosis = '$id_diagnosis' and id_gejala = '$id[$i]'");
        foreach ($result[$i]->result_array() as $row) {
            $com[$i] = $row['kombinasi'];
        }
        $CFCombine = $CFCombine + ($com[$i] * (1 - $CFCombine));
    }
}

```



```

return $CFCombine;
}
public function tambah_diagnosis($CFCombine, $id_diagnosis) {
    if ($CFCombine >= 0.9){
        $status = 'Demam Tifoid Berat';
    }else if ($CFCombine > 0 && $CFCombine < 0.9){
        $status = 'Demam Tifoid Ringan';
    }else{
        $status = 'negatif';
    }
    $result = $this->db->query("update tb_diagnosis set total = $CFCombine, diagnosis =
'Status' where id_diagnosis = $id_diagnosis");
    return $status;
}
public function view_record($id_user) {
    $query = $this->db->query("SELECT p.id_pasien, p.nama, p.umur from tb_diagnosis d
join tb_pemeriksaan pm on (pm.id_diagnosis = d.id_diagnosis) join tb_pasien p on (p.id_pasien
= pm.id_pasien) where p.id_user = $id_user group by id_pasien");
    return $query;
}
public function view_detail_record($id_pasien) { /* Load Data golongan Semua */
    $query = $this->db->query("SELECT pm.id_pasien, left(d.total,4) as t, d.diagnosis,
pm.id_diagnosis, max(pm.tanggal) as tanggal from tb_diagnosis d join tb_pemeriksaan pm on
(pm.id_diagnosis = d.id_diagnosis) where pm.id_pasien = $id_pasien group by pm.id_diagnosis
order by pm.tanggal desc");
    return $query;
}

public function view_id_record_terakhir($id_pasien) { /* Load Data golongan Semua */
    $query = $this->db->query("SELECT d.id_record, max(d.id_diagnosis) as id_diagnosis,
max(pm.tanggal) as tanggal from tb_diagnosis d join tb_pemeriksaan pm on (pm.id_diagnosis =
d.id_diagnosis) where id_pasien = $id_pasien group by pm.id_diagnosis order by tanggal
desc");
    return $query;
}
public function ambil_id_record($id_record) { /* Load Data golongan Semua */
    $query = $this->db->query("SELECT id_record from tb_diagnosis where id_record =
$id_record limit 1");
    return $query;
}
public function view_detail_pemeriksaan($id_pasien, $id_diagnosis) { /* Load Data
golongan Semua */
    $query = $this->db->query("SELECT g.jenis_gejala, pm.jawaban, pm.tanggal FROM
tb_pasien ps join tb_pemeriksaan pm on (ps.id_pasien = pm.id_pasien) join tb_gejala g on
(g.id_gejala = pm.id_gejala) where pm.id_pasien = $id_pasien and pm.id_diagnosis =
$id_diagnosis");
    return $query;
}}

```

Lampiran E. Pengujian *Black Box*

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Menambah data gejala	Mengisi kode gejala, jenis gejala lalu Mengklik tombol simpan	Menyimpan gejala dan menampilkan pesan “berhasil ditambah”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Kode gejala telah digunakan	Menampilkan pesan “kode gejala telah digunakan”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Kolom belum diisi	Menampilkan pesan dengan menunjuk pada kolom yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
2.	Mengubah data gejala	Mengisi jenis gejala lalu Mengklik tombol simpan	Menyimpan gejala dan menampilkan pesan “berhasil diubah”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
3.	Menghapus data gejala	Mengklik tombol delete Klik tombol “ya”	Menghapus gejala dan menampilkan pesan “berhasil dihapus”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol “tidak”	Tidak menghapus gejala dan tetap pada halaman data gejala	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
4.	Melihat data gejala	Klik submenu data gejala	Menampilkan data gejala	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
5.	Menambah data CF	Memilih gejala dan mengisi MB, MD lalu mengklik tombol simpan	Menyimpan CF dan menampilkan pesan “berhasil ditambah”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

		Data CF telah ada	Menampilkan pesan “data CF telah ada”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Kolom belum diisi	Menampilkan pesan dengan menunjuk pada kolom yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
6.	Mengubah data CF	Mengisi MB dan MD Mengklik tombol simpan	Menyimpan CF dan menampilkan pesan “berhasil diubah”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
7.	Menghapus data gejala	Mengklik tombol delete Klik tombol “ya” Klik tombol “tidak”	Menghapus CF dan menampilkan pesan “berhasil dihapus” Tidak menghapus CF dan tetap pada halaman data CF	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
8.	Melihat data CF	Klik submenu data CF	Menampilkan data CF	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
9.	Melihat Rekap Data Pasien	Klik submenu data diagnosis	Menampilkan rekap data pasien	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
10.	Melihat Data diagnosis	Klik tombol lihat <i>record</i> data diagnosis Klik tombol detail	Menampilkan data <i>record</i> pemeriksaan Menampilkan detail pemeriksaan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

11.	Memasukkan pemeriksaan pasien lama	Mengisi pilihan jawaban pada form keluhan lalu mengklik tombol <i>check</i>	Menampilkan persentase keyakinan dan saran penanganan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Pertanyaan belum diisi	Menampilkan pesan dengan menunjuk pada pertanyaan yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
12.	Memasukkan pemeriksaan pasien baru	Mengisi kolom nama, umur dan pilihan form keluhan lalu mengklik tombol <i>check</i>	Menampilkan persentase keyakinan dan saran penanganan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Kolom belum diisi	Menampilkan pesan dengan menunjuk pada kolom yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Pertanyaan belum diisi	Menampilkan pesan dengan menunjuk pada pertanyaan yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
13.	Melihat Data pasien anggota	Klik menu <i>dashboard</i>	Menampilkan data pasien anggota	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
14.	Melihat <i>record</i> pemeriksaan	Klik tombol lihat <i>record</i> pada halaman <i>dashboard</i>	Menampilkan <i>record</i> pemeriksaan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
15.	Melihat detail <i>record</i> pemeriksaan	Klik tombol lihat <i>detail</i> pada halaman <i>record</i> pemeriksaan	Menampilkan detail <i>record</i> pemeriksaan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
16.	Mendaftar Anggota	Mengisi form pendaftaran lalu klik tombol daftar	Menyimpan data dan menampilkan pesan “pendaftaran berhasil”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

	<i>Username</i> telah digunakan	Menampilkan pesan “Gagal! <i>Username</i> telah digunakan”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	Kolom belum diisi	Menampilkan pesan dengan menunjuk pada kolom yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
17.	<i>Login</i>	Mengisi <i>username</i> dan <i>password</i> lalu klik tombol <i>login</i>	<i>Login</i> berhasil dan menampilkan halaman awal masing-masing aktor	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Username atau password salah	Menampilkan pesan “Login gagal! <i>Username</i> atau <i>password</i> salah	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	Kolom belum diisi	Menampilkan pesan dengan menunjuk pada kolom yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
18.	<i>Logout</i>	Klik tombol <i>logout</i>	Berhasil keluar sistem dan menampilkan halaman awal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal