



**RANCANG BANGUN SISTEM PAKAR PREDIKSI TINGKATAN  
ASMA PASIEN DAN PENANGANANNYA, MENGGUNAKAN  
METODE *CERTAINTY FACTOR***

**SKRIPSI**

Oleh

**Rinaldi Dwi Sayoga**

**NIM 122410101028**

**PROGRAM STUDI SISTEM INFORMASI  
UNIVERSITAS JEMBER**

**2017**



**RANCANG BANGUN SISTEM PAKAR PREDIKSI TINGKATAN  
ASMA PASIEN DAN PENANGANANNYA, MENGGUNAKAN  
METODE *CERTAINTY FACTOR***

**SKRIPSI**

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

Oleh

**Rinaldi Dwi Sayoga**

**NIM 122410101028**

**PROGRAM STUDI SISTEM INFORMASI  
UNIVERSITAS JEMBER**

**2017**

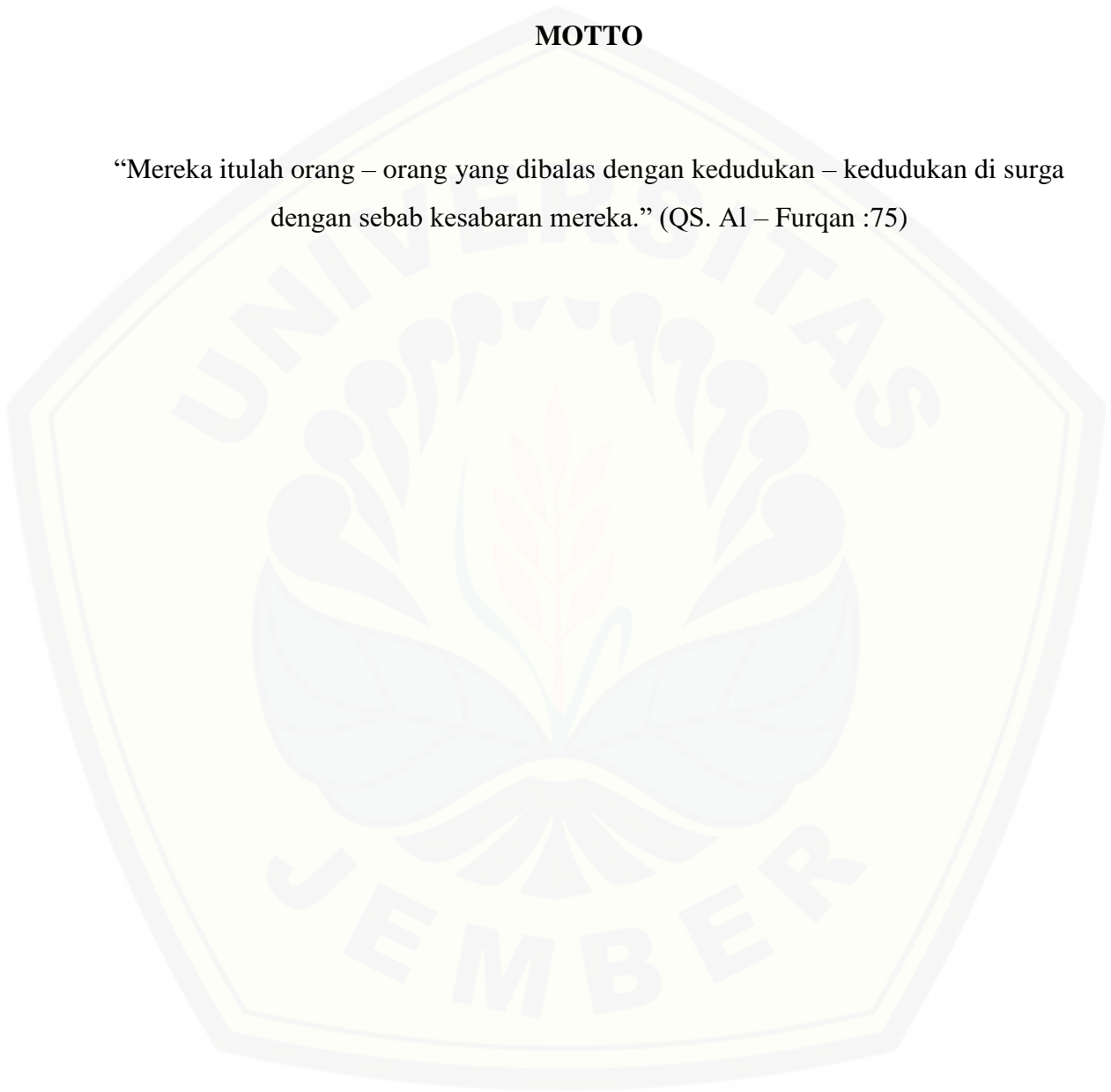
## PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi;
2. Ibunda Muryati dan Ayahanda Suryadi tersayang yang selalu memberikan dukungan do'a, kasih sayang, motivasi yang luar biasa;
3. Kakak Renita Mardiyasari tercinta yang selalu memberi dukungan;
4. Diah Ayu dan Mama Ina yang selalu mengingatkan dan memberikan semangat dalam menyelesaikan skripsi.
5. Seluruh keluarga besar tersayang yang ada di Jember, Sidoarjo, Situbondo, Malang dan Banyuwangi;
6. Sahabat terdekat saya Yohanis Permadi, Bagus Cahyo, Riska Arimanudin, Dimas Arifianto, Bagus Ananda, Yoko, dan Gede Bagoes yang selalu meberikan dukungan dan motivasi;
7. Yohanis Permadi, mas Arifin, Marceli Aditya, bapak Yanuar yang telah membantu dalam penyusunan skripsi.
8. Adik – adik angkatan yang selalu memberikan semangat agar terus bersabar;
9. Sahabatku bersama doa dan bantuannya;
10. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi;
11. Almamater program studi sistem informasi universitas jember.

**MOTTO**

“Mereka itulah orang – orang yang dibalas dengan kedudukan – kedudukan di surga dengan sebab kesabaran mereka.” (QS. Al – Furqan :75)



**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Rinaldi Dwi Sayoga

NIM : 122410101028

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Pakar Prediksi Tingkatan Asma Pasien dan Penanganannya”, 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, 4 Mei 2017

Yang menyatakan,

Rinaldi Dwi Sayoga

NIM 122410101028

**SKRIPSI**

**RANCANG BANGUN SISTEM PAKAR PREDIKSI TINGKATAN  
ASMA PASIEN DAN PENANGANANNYA, MENGGUNAKAN  
METODE *CERTAINTY FACTOR***

Oleh

**Rinaldi Dwi Sayoga**

**NIM 122410101028**

Pembimbing :

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

Dosen Pembimbing Pendamping : Nelly Oktavia Adiwijaya, S.Si., M.T.

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “Rancang Bangun Sistem Pakar Prediksi Tingkatan Asma Pasien dan Penanganannya, Menggunakan Metode *Certainty Factor*” telah diuji dan disahkan pada:

hari, tanggal : Kamis, 4 Mei 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh :

Pembimbing I,

Pembimbing II,

Anang Andrianto.,S.T.,M.T

Nelly Oktavia Adiwijaya, S.Si., M.T.

NIP 196906151997021002

NIP 198410242009122008

**PENGESAHAN PENGUJI**

Skripsi berjudul “Rancang Bangun Sistem Pakar Prediksi Tingkatan Asma Pasien dan Penanganannya, Menggunakan Metode *Certainty Factor*” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim penguji:

Penguji I,

Penguji II,

Dr. Saiful Bukhori, ST., M.Kom  
NIP. 196811131994121001

Fajrin Nurman Arifin, ST., M.Eng  
NIP. 198511282015041002

Mengesahkan

Ketua Program Studi,

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

NIP 196704201992011001



## RINGKASAN

**Rancang Bangun Sistem Pakar Prediksi Tingkatan Asma Pasien dan Penanganannya, Menggunakan Metode *Certainty Factor***; Rinaldi Dwi Sayoga, 122410101028; 2017: 135 halaman; Progam Studi Sistem Informasi Universitas Jember.

Asma adalah suatu penyakit yang menyerang pernapasan manusia yang memiliki ciri – ciri susah bernapas, batuk, ataupun napas berbunyi(mengi). Asma dapat mempengaruhi kualitas hidup serta beban sosial ekonomi. Asma mempunyai tingkat kematian yang rendah namun kasusnya cukup banyak di negara dengan pendapatan menengah kebawah. Penyakit asma pada dasarnya adalah termasuk salah satu golongan dari penyakit ringan, tetapi penyakit ini tidak jarang juga menimbulkan korban jiwa jika tidak segera ditangani atau mengalami perlakuan yang kurang dalam pengobatannya. Penyebab dari timbulnya penyakit asma pun bermacam- macam, diantaranya adalah alergi yang merupakan faktor pencetus asma yang paling penting. Macam-macam penyebab alergi ini dapat berupa kutu debu, kecoak, binatang berbulu, dan polen/tepung sari. Kutu debu umumnya ditemukan pada lantai rumah, karpet dan tempat tidur yang kotor. Udara pun dapat menjadi pencetus asma jika terdapat suhu udara yang terlalu dingin atau pun terlalu panas. Dan asma terdiri atas beberapa tingkatan, diantaranya adalah asma rendah, asma sedang dan asma berat. Tingkatan asma tersebut berbeda – beda dari tingkat keparahan pasien sampai cara penanganan yang baik untuk pasien.

Pada penelitian ini dibangun sistem pakar prediksi tingkatan asma pasien menggunakan metode *certainty factor*, sehingga dapat membantu dan menggantikan tugas dokter sebagai pakar dalam melakukan diagnosis tingkatan asma pasien. Sehingga akan didapatkan hasil diagnosis sistem pakar yang hasilnya akan sama dengan hasil diagnosis dokter dan mampu dipertanggung jawabkan kebenarannya.

## PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Rancang Bangun Sistem Pakar Prediksi Tingkatan Asma Pasien dan Penanganannya, Menggunakan Metode *Certainty Factor*”. 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:

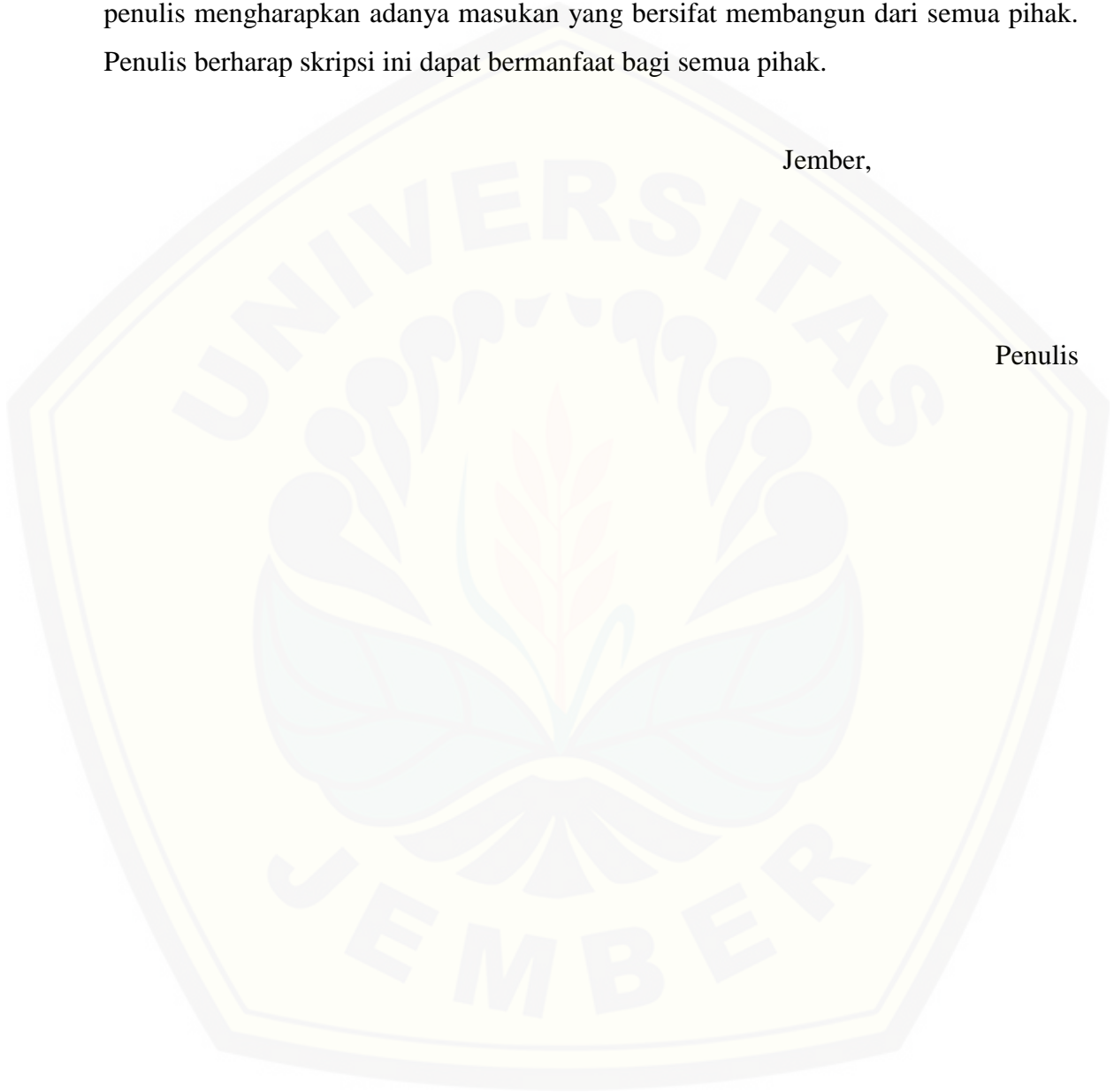
1. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Anang Andrianto, S.T., M.T., selaku Dosen Pembimbing Utama dan Nelly Oktavia Adiwijaya, S.Si., M.T., selaku Dosen Pembimbing Pendamping yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Syaiful Bukhori, S.T., M.Kom., sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di program studi sistem informasi;
5. Ibunda Muryati dan Ayahanda Suryadi tersayang yang selalu memberikan dukungan do'a, kasih sayang, motivasi yang luar biasa;
6. Kakak Renita Mardiyasari tercinta yang selalu memberi dukungan;
7. Seluruh keluarga besar tersayang yang ada di Jember, Sidoarjo, Situbondo, Malang dan Banyuwangi;
8. Sahabat terdekat saya Yohanis Permadi, Bagus Cahyo, Riska Arimanudin, Dimas Arifianto, Bagus Ananda, Yoko, dan Gede Bagoes yang selalu memberikan dukungan dan motivasi;
9. Keluarga besar Fakultas Ilmu Komputer, Program Studi Sistem Informasi Universitas Jember;

10. 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,

Penulis



**DAFTAR ISI**

	Halaman
PERSEMBAHAN.....	iii
MOTTO .....	iv
PERNYATAAN.....	v
SKRIPSI.....	vi
PENGESAHAN PEMBIMBING.....	vii
PENGESAHAN PENGUJI.....	viii
RINGKASAN .....	ix
PRAKATA.....	x
DAFTAR ISI.....	xii
DAFTAR TABEL.....	xvii
DAFTAR GAMBAR .....	xviii
DAFTAR LAMPIRAN.....	xx
BAB 1. PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan dan Manfaat.....	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan.....	4
BAB 2. TINJAUAN PUSTAKA .....	5
2.1 Penelitian Terdahulu .....	5
2.2 Dasar Teori .....	6
2.2.1 Asma.....	6
2.2.2 Kecerdasan Buatan .....	10
2.2.3 <i>Certainty Factor</i> .....	12

BAB 3. METODOLOGI PENELITIAN.....	15
3.1 Jenis Penelitian .....	15
3.2 Waktu dan Tempat Penelitian .....	15
3.3 Alat Penelitian .....	15
3.4 Tahapan Penelitian .....	16
3.4.1 Tahapan Analisa Kebutuhan .....	17
3.4.2 Tahapan Desain Sistem .....	18
3.4.3 Tahapan Implementasi Sistem .....	19
3.4.4 Tahapan Pengujian Sistem .....	20
3.4.5 Tahapan Pemeliharaan Sistem .....	21
3.5 Gambaran Umum Sistem .....	21
BAB 4. PENGEMBANGAN SISTEM.....	23
4.1 Analisis Kebutuhan Sistem .....	23
4.1.1 Kebutuhan Fungsional.....	23
4.1.2 Kebutuhan Non-Fungsional .....	24
4.2 Desain Sistem .....	24
4.2.1 Business process.....	24
4.2.2 <i>Usecase Diagram</i> .....	25
4.2.3 Skenario Sistem.....	28
4.2.4 Activity Diagram.....	36
4.2.5 <i>Sequence Diagram</i> .....	47
4.2.6 <i>Class Diagram</i> .....	56
4.2.7 <i>Entity Relationship Diagram</i> .....	57
4.3 Penulisan Kode Program .....	58
4.3.1 Kode Program Controller C_Diagnosa.....	59

4.3.2 Kode Program <i>Controller</i> Cdata_Gabungan_Pakar .....	61
4.3.3 Kode Program <i>Controller</i> C_info.....	62
4.3.4 Kode Program <i>Controller</i> login.....	62
4.3.5 Kode Program <i>Controller</i> Cdata_penyakit_pakar .....	63
4.3.6 Kode Program <i>Controller</i> Cdata_pencegahan .....	63
4.3.7 Kode Program <i>Controller</i> Cdata_Ge.....	63
4.4 Pengujian sistem.....	63
4.4.1 White Box Testing .....	63
4.4.2 <i>Black box Testing</i> .....	70
<b>BAB 5. HASIL DAN PEMBAHASAN.....</b>	<b>77</b>
5.1 Sistem Pakar Prediksi Tingkatan Asma Pasien dan Penanganannya .....	77
5.2 Hasil Implementasi Sistem Pakar Prediksi Tingkatan Asma Pasien Dan Penanganannya .....	77
5.2.1 Tampilan halaman awal sistem .....	78
5.2.2 Tampilan <i>login screen</i> .....	79
5.2.3 Tampilan halaman awal home pakar.....	79
5.2.4 Tampilan Halaman Data Pencegahan .....	80
5.2.5 Tampilan Form Tambah Data Pencegahan .....	81
5.2.6 Tampilan Peringatan Hapus Data Pencegahan.....	82
5.2.7 Tampilan Halaman Data Tingkatan Asma.....	82
5.2.8 Tampilan <i>Form</i> Tambah Data Tingkatan Asma .....	83
5.2.9 Tampilan <i>Form Edit</i> Data Tingkatan Asma.....	84
5.2.10 Tampilan Peringatan Hapus data Tingkatan Asma .....	85
5.2.11 Tampilan Halaman Data Gejala .....	85

5.2.12 Tampilan <i>Form Edit</i> Data Gejala.....	86
5.2.13 Tampilan Peringatan Hapus Data Gejala.....	87
5.2.14 Tampilan Halaman Data Gabungan.....	88
5.2.15 Tampilan <i>Form</i> Tambah Data Gabungan.....	88
5.2.16 Tampilan Peringatan Hapus Data Gabungan.....	89
5.2.17 Tampilan <i>Form</i> Tambah Data Gejala.....	89
5.2.18 Tampilan <i>Form</i> Diagnosa Asma Pasien.....	90
5.2.19 Tampilan Halaman Pencegahan.....	91
5.3 Pembahasan Metode <i>Certainty Factor</i> Untuk Sistem Pakar Prediksi Tingkatan Asma Pasien Dan Penanganannya.....	92
5.3.1 Implementasi Metode <i>Certainty Factor</i> .....	92
5.4 Pengujian Sistem.....	94
5.4.1 Pengujian Perhitungan Sistem Menggunakan Metode <i>Certainty Factor</i> .....	95
5.4.2 Uji Keakuratan Data.....	100
BAB 6. PENUTUP.....	102
5.1 Kesimpulan.....	102
5.2 Saran.....	102
DAFTAR PUSTAKA.....	104
LAMPIRAN.....	105
Lampiran <i>screenshot</i> kode program C_info.....	105
Lampiran <i>screenshot</i> kode program login.....	106
Lampiran <i>screenshot</i> kode program Cdata_penyakit_pakar.....	106
Lampiran <i>screenshot</i> kode program Cdata_gabungan_pakar.....	108
Lampiran <i>screenshot</i> kode program controller Cdata_gejala_pakar.....	110
Lampiran <i>screenshot</i> kode program M_data_cegah.....	111
Lampiran <i>screenshot</i> kode program M_data_gabungan.....	112

Lampiran <i>screenshot</i> kode program M_data_gejala .....	113
Lampiran <i>screenshot</i> kode program M_data_pakar .....	114
Lampiran <i>screenshot</i> kode program M_data_penyakit .....	115





## DAFTAR TABEL

	Halaman
Tabel 2. 1 Tabel Tingkatan Asma .....	8
Tabel 2. 2 Penanganan .....	9
Tabel 4. 1 Macam - macam user .....	27
Tabel 4. 2 Definisi Usecase.....	27
Tabel 4. 3 Skenario Login.....	29
Tabel 4. 4 Skenario Mengelola Data Gejala Asma .....	30
Tabel 4. 5 Skenario Mengelola Data Nilai CF .....	32
Tabel 4. 6 Skenario Mengelola Data Pencegahan Asma Pasien .....	33
Tabel 4. 7 Skenario Melihat Halaman Home.....	35
Tabel 4. 8 Skenario Mendiagnosis Gejala Penyakit Asma .....	35
Tabel 4. 9 Skenario Mendiagnosis Pencegahan Asma Pasien .....	36
Tabel 4. 10 Test Case Fitur Tambah Gejala Asma .....	66
Tabel 4. 11 Test Case Perhitungan Metode Certainty Factor .....	68
Tabel 4. 12 Black Box Login .....	70
Tabel 4. 13 Tabel Pengujian Black Box Mengelola Data Gejala .....	70
Tabel 4. 14 Pengujian Black Box Manajemen Data Tingkatan Asma.....	71
Tabel 4. 15 Pengujian Black Box Mengelola Data Pencegahan Asma.....	73
Tabel 4. 16 Pengujian Black Box Mengelola Data Gabungan.....	74
Tabel 4. 17 Pengujian Black Box Mendiagnosa Asma Pasien .....	75
Tabel 4. 18 Pengujian Black Box Pencegahan Asma Pasien.....	76
Tabel 5. 1 Range Tingkatan Asma Pasien .....	94
Tabel 5. 2 Gejala Terpilih Pertama .....	96
Tabel 5. 3 Gejala Terpilih Kedua.....	98
Tabel 5. 4 Tabel Perbandingan Akurasi Diagnosis Sistem dan Dokter .....	100

**DAFTAR GAMBAR**

	Halaman
Gambar 4. 1 Business Process .....	25
Gambar 4. 2 Usecase Diagram.....	26
Gambar 4. 3 Activity Diagram Diagnosis Pencegahan Asma Pasien.....	37
Gambar 4. 4 Activity Diagram Diagnosis Tingkatan Asma Pasien.....	38
Gambar 4. 5 Activity Diagram Login .....	39
Gambar 4. 6 Activity Diagram Mengelola Data Gejala Penyakit Asma .....	40
Gambar 4. 7 Activity Diagram Mengelola Data Pencegahan Asma.....	42
Gambar 4. 8 Activity Diagram Mengelola Data Gabungan.....	45
Gambar 4. 9 Activity Diagram Mengelola Data Tingkatan Asma.....	46
Gambar 4. 10 Sequence Diagram Login .....	47
Gambar 4. 11 Activity Diagram Diagnosa Asma Pasien .....	48
Gambar 4. 12 Sequence Diagram Diagnosa Pencegahan Asma Pasien.....	49
Gambar 4. 13 Sequence Diagram Mengelola Data Gejala Asma .....	50
Gambar 4. 14 Sequence Diagram Mengelola Data Pencegahan Asma .....	52
Gambar 4. 15 Sequence Diagram Mengelola Data Tingkatan Asma .....	54
Gambar 4. 16 Sequence Diagram Mengelola Data Nilai CF .....	55
Gambar 4. 17 Class Diagram .....	57
Gambar 4. 18 Entity Relationship Diagram .....	58
Gambar 4. 19 Kode Program Controller Diagnosa.....	60
Gambar 4. 20 Controller Tambah Data Gabungan .....	61
Gambar 4. 21 Edit Data Gabungan .....	62
Gambar 4. 22 Hapus Data Gabungan.....	62

Gambar 4. 23 White Box Testing Mengelola Data Gejala Asma .....	64
Gambar 4. 24 White Box Testing Proses Tambah Gejala .....	64
Gambar 4. 25 Diagram Alir Tambah Data Gejala .....	65
Gambar 4. 26 Diagram Alir Proses Tambah Gejala .....	65
Gambar 4. 27 Listing Kode Program Metode Certainty Factor .....	67
Gambar 4. 28 Diagram Alir Perhitungan Metode Certainty Factor .....	68
Gambar 5. 1 Tampilan Halaman Awal Sistem .....	78
Gambar 5. 2 Tampilan Login Screen .....	79
Gambar 5. 3 Tampilan Halaman Awal Home Pakar.....	80
Gambar 5. 4 Tampilan Halaman Data Pencegahan .....	81
Gambar 5. 5 Tampilan Form Tambah Data Pencegahan .....	82
Gambar 5. 6 Tampilan Peringatan Hapus Data Pencegahan.....	82
Gambar 5. 7 Tampilan Halaman Data Tingkatan Asma .....	83
Gambar 5. 8 Tampilan Form Tambah Tingkatan Asma .....	84
Gambar 5. 9 Tampilan Form Edit Data Tingkatan Asma .....	85
Gambar 5. 10 Tampilan Peringatan Hapus Data Tingkatan Asma .....	85
Gambar 5. 11 Tampilan Halaman Data Gejala .....	86
Gambar 5. 12 Tampilan Form Edit Gejala.....	87
Gambar 5. 13 Tampilan Peringatan Hapus Data Gejala .....	87
Gambar 5. 14 Tampilan Halaman Data Gabungan .....	88
Gambar 5. 15 Tampilan Form Tambah Data Gabungan.....	89
Gambar 5. 16 Tampilan Form Peringatan Hapus Data Gabungan.....	89
Gambar 5. 17 Tampilan Form Tambah Data Gejala.....	90
Gambar 5. 18 Tampilan Form Diagnosa Asma Pasien .....	91
Gambar 5. 19 Tampilan Halaman Pencegahan .....	92

**DAFTAR LAMPIRAN**

	Halaman
Lampiran screenshot kode program C_info.....	105
Lampiran screenshot kode program login.....	106
Lampiran screenshot kode program Cdata_penyakit_pakar.....	106
Lampiran screenshot kode program Cdata_gabungan_pakar.....	108
Lampiran screenshot kode program Cdata_gejala_pakar.....	110
Lampiran screenshot kode program M_data_cegah.....	111
Lampiran screenshot kode program M_data_gabungan.....	112
Lampiran screenshot kode program M_data_gejala.....	113
Lampiran screenshot kode program M_data_pakar.....	114
Lampiran screenshot kode program M_data_penyakit.....	115

## BAB 1. PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir. Bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian, dan sistematika penulisan.

### 1.1 Latar Belakang

Asma dapat mempengaruhi kualitas hidup serta beban sosial ekonomi. Asma mempunyai tingkat kematian yang rendah namun kasusnya cukup banyak di negara dengan pendapatan menengah kebawah. Menurut *World Health Organization* memperkirakan 235 juta penduduk dunia menderita asma dan jumlahnya diperkirakan akan terus bertambah (WHO, 2013) Penyakit asma pada dasarnya adalah termasuk salah satu golongan dari penyakit ringan, tetapi penyakit ini tidak jarang juga menimbulkan korban jiwa jika tidak segera ditangani atau mengalami perlakuan yang kurang dalam pengobatannya.

Menurut data studi Survey Kesehatan Rumah Tangga (SKRT) di berbagai propinsi di Indonesia, pada tahun 1986 asma menduduki urutan kelima dari sepuluh penyebab kesakitan (*morbiditas*) bersama-sama dengan *bronkitis kronik* dan *emfisema*. Pada SKRT 1992, asma, *bronkitis kronik*, dan *emfisema* sebagai penyebab kematian (*mortalitas*) keempat di Indonesia atau sebesar 5,6%. Lalu pada SKRT 1995, dilaporkan prevalensi asma di seluruh Indonesia sebesar 13 per 1.000 penduduk ((PDPI), 2015).

Penyebab dari timbulnya penyakit asma pun bermacam- macam, penyebab dari asal muasal penderita asma itu dapat berupa dari penyakit turunan yang diderita dari orang tuanya. Penyebab lain dari adalah alergi yang merupakan faktor pencetus asma yang paling penting. Macam-macam penyebab alergi ini dapat berupa kutu debu, kecoak, binatang berbulu, dan polen/tepung sari. Kutu debu umumnya ditemukan pada

lantai rumah, karpet dan tempat tidur yang kotor. Udara pun dapat menjadi pencetus asma jika terdapat suhu udara yang terlalu dingin atau pun terlalu panas.

Kemudian menurut Achmad Hudoyo (Hudoyo, 2014) medikasi asma ditujukan untuk mengatasi dan mencegah gejala obstruksi jalan napas. Didalamnya terdiri atas pelega dan pengontrol. Medikasi asma yang ditujukan kepada pengidap asma pun berbeda beda bergantung berdasarkan tingkatan asma yang dialami oleh pasien.

Diagnosis asma untuk penentuan tingkatan asma dan penanganan pasien didasari oleh gejala yang bersifat episodik, gejala berupa batuk, sesak napas, mengi, rasa berat di dada dan variabiliti yang berkaitan dengan cuaca. Anamnesis yang baik cukup untuk menegakkan diagnosis, ditambah dengan pemeriksaan jasmani dan pengukuran faal paru terutama reversibiliti kelainan faal paru, akan lebih meningkatkan nilai diagnostik (Hudoyo, 2014). Diagnosis asma berdasarkan dokter ini pun berhubungan dengan tabel tingkatan asma pasien, tetapi meskipun sudah tercantum didalam tabel tersebut, terkadang juga terjadi kesalahan diagnosis karena kemiripan data diagnosis dari hasil pemeriksaan terhadap pasien sehingga dapat muncul nilai ketidakpastian pada diagnosis asma pasien.

Logika pada *certainty factor* telah menjadi area riset yang mengagumkan karena kemampuannya dalam menghubungkan bahasa mesin yang serba presisi dengan bahasa manusia yang cenderung tidak presisi serta menekankan pada makna atau arti. Dengan demikian, sistem ini adalah sebuah mesin penerjemah bahasa manusia sehingga bisa dimengerti oleh bahasa mesin dan juga sebaliknya. Logika *certainty factor* telah diaplikasikan dalam bidang kedokteran, yang didalamnya terdapat ketidakpastian dan bidang kedokteran merupakan contoh permasalahan yang cocok untuk aplikasi logika *certainty factor*, karena terdapat ketidakpastian, ketidak tepatan pengukuran, keanekaragaman dan subjektivitas yang secara jelas hadir dalam melakukan diagnosa medis.

## 1.2 Rumusan Masalah

1. Apakah sistem mampu menentukan tingkatan asma pada pasien?
2. Apa penyebab serangan asma pasien?
3. Bagaimana cara penanganan pasien tersebut?

## 1.3 Tujuan dan Manfaat

Penyakit asma memiliki beberapa penyebab yang bermacam – macam, diantaranya yang paling sering ditemukan di lingkungan masyarakat adalah berasal dari alergi debu dan penyakit keturunan yang diidap oleh orang tuanya. Dengan adanya sistem pakar penentuan tingkatan penyakit asma, ada beberapa hal yang nantinya akan menjadi tujuan dan manfaat dari sistem ini jika sistem telah dijalankan. Manfaatnya antara lain :

1. Dapat ditemukan tingkatan asma yang dialami oleh pasien.
2. Dapat diketahui apa penyebab serangan asma pasien tersebut.
3. Ditemukannya bagaimana cara untuk penanganan yang tepat dan cepat bagi pasien yang secara tiba – tiba mengalami serangan asma.

## 1.4 Batasan Masalah

Beberapa hal yang membatasi penelitian ini diantaranya adalah:

1. Sistem informasi ini hanya terfokus kepada pasien dan dokter.
2. Proses perhitungan sistem menggunakan metode *certainty factor*
3. Sistem dapat membantu pengolahan data pemeriksaan pasien melalui perhitungan data sehingga mendapatkan tingkatan asma pasien.
4. Kriteria yang digunakan untuk menentukan tingkatan asma pasien didapatkan dari banyaknya serangan setiap hari, minggu, atau bulan dan dari kekuatan paru – paru pasien dalam bernapas.

### 1.5 Sistematika Penulisan

Sistematika penulisan dan keruntutan skripsi ini disusun sebagai berikut:

1. Pendahuluan

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

2. Tinjauan Pustaka

Bab ini menjelaskan tentang materi, informasi, tinjauan pustaka, dan studi terdahulu yang menjadi kerangka pemikiran dalam penelitian.

3. Metodologi Penelitian

Bab ini menjelaskan tentang metode penelitian yang digunakan dalam penelitian.

4. Pengembangan Sistem

Bab ini menjelaskan tentang pengembangan sistem yang dikembangkan.

5. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang dilakukan.

6. Penutup

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



## BAB 2. TINJAUAN PUSTAKA

Pada bagian ini akan dijelaskan tinjauan terhadap hasil-hasil penelitian terdahulu berkaitan dengan masalah yang dibahas, kajian teori berkaitan dengan masalah, kerangka pemikiran yang merupakan sintesis dari kajian teori yang dikaitkan dengan permasalahan yang dihadapi.

### 2.1 Penelitian Terdahulu

Penelitian sebelumnya yang berjudul “Sistem Pakar Diagnosa Level Penyakit Asma Menggunakan Metode Fuzzy Tsukamoto” yang dilakukan oleh Muhyiddin Ali, Mahasiswa Program Studi Teknik Informatika, Jurusan Ilmu Komputer, Universitas Dian Nuswantoro Semarang, telah menghasilkan sebuah aplikasi diagnosa penyakit asma untuk mengetahui tingkat atau level penyakit asma. Penelitian sistem pakar diagnosa tingkatan asma pasien ini menggunakan metode *Fuzzy Logic* Tsukamoto. Kelemahan dari metode *Fuzzy Logic* Tsukamoto ini adalah, dari perhitungan menggunakan seluruh data yang ada dengan dihitung sekaligus, hal ini juga memungkinkan untuk terjadinya kesalahan pada penghitungan sehingga menyebabkan hasil prediksi menjadi tidak akurat. Berbeda dengan metode *Certainty Factor* yang menggunakan perhitungan tiap 2 kategori indikator untuk meminimalisir kemungkinan terjadinya kesalahan perhitungan, sehingga hasil nilai dari perhitungan yang dikeluarkan oleh metode *certainty factor* menjadi lebih akurat dibandingkan metode *fuzzy*. Selain itu, pada metode fuzzy, kita harus merubah nilai crisp menjadi nilai linguistik. Lalu juga harus dicari fungsi keanggotaannya, jika bentuk fungsi keanggotaan pada metode fuzzy salah sedikit saja, maka akan mempengaruhi hasil

prediksi. Oleh karena itulah metode certainty factor akan lebih baik penggunaannya dari segi akurasi dan efisiensi.

Penelitian lain mengenai metode certainty factor juga dilakukan oleh Stephanie Halim, mahasiswi Program Studi Teknik Informatika, Universitas Multimedia Nusantara, Tangerang pada pembahasannya yang berjudul “Penerapan Metode Certainty Factor dalam Sistem Pakar Pendeteksi Resiko Osteoporosis dan Osteoarthritis”. Pada penelitian ini, telah dibuat sebuah aplikasi untuk mendeteksi pasien penderita osteoporosis dan osteoarthritis. Menurutnya kelebihan dari metode ini adalah cocok digunakan pada sistem pakar yang mengukur sesuatu yang pasti atau tidak pasti seperti mendiagnosis penyakit dan perhitungan dari metode ini hanya berlaku untuk sekali hitung, serta hanya dapat mengolah dua data sehingga keakuratannya terjaga.

## 2.2 Dasar Teori

### 2.2.1 Asma

Asma merupakan sebuah penyakit kronik saluran napas yang terdapat di seluruh dunia dengan kekerapan bervariasi yang berhubungan dengan peningkatan kepekaan saluran napas sehingga memicu episode mengi berulang (*wheezing*), sesak napas (*breathlessness*), dada rasa tertekan (*chest tightness*), dispnea, dan batuk (*cough*) terutama pada malam atau dini hari. ((GINA), 2013) Adapun faktor-faktor yang berhubungan dengan kejadian asma adalah:

#### 1. Imunitas dasar

Mekanisme imunitas terhadap kejadian peradangan pada asma kemungkinan terjadi ekspresi sel Th2 yang berlebihan ((NHLBI), 2007). Menurut (Dr. Moffat, 2007), gen kromosom bernama ORMDL3 mempunyai hubungan kuat sebagai faktor penyebab asma.

#### 2. Umur

Kasus tertinggi asma biasanya mengenai anak-anak (7-10%), yaitu umur 5 sampai dengan 14 tahun. Sedangkan pada orang dewasa, angka kejadian asma lebih kecil yaitu sekitar 3-5% (Asthma and Allergy Foundation of America, 2010). Menurut studi yang dilakukan oleh Australian Institute of Health and Welfare (2007), kejadian asma pada kelompok umur 18 sampai dengan 34 tahun adalah 14% sedangkan >65 tahun menurun menjadi 8.8%. Di Jakarta, sebuah studi pada RSUP Persahabatan menyimpulkan rerata angka kejadian asma adalah umur 46 tahun (Pratama, 2009).

### 3. Jenis Kelamin

Menurut ((GINA), 2013) dan ((NHLBI), 2007) jenis kelamin laki-laki merupakan sebuah faktor resiko terjadinya asma pada anak-anak. Akan tetapi, pada masa pubertas, rasio penderita bergeser dan menjadi lebih sering terjadi pada perempuan ((NHLBI), 2007)

### 4. Faktor pencetus

Paparan terhadap penyebab alergi merupakan faktor pencetus asma yang paling penting. Macam-macam penyebab alergi ini dapat berupa kutu debu, kecoak, binatang berbulu, dan polen/tepung sari. Kutu debu umumnya ditemukan pada lantai rumah, karpet dan tempat tidur yang kotor. Kecoa telah dibuktikan menyebabkan tingkat sensitif pada alergi menjadi bertambah, terutama pada rumah di perkotaan ((NHLBI), 2007). Menurut ((GINA), 2013), paparan terhadap binatang, khususnya bulu anjing dan kucing juga dapat meningkatkan tingkat sensitif alergi seseorang.

Selain itu derajat penyakit asma menurut tingkat gejala, keterbatasan aliran udara, dan fungsi paru dikategorikan ke dalam 3 kategori yaitu asma ringan, asma sedang, dan asma berat. Yang membedakan dari macam asma tersebut terutama adalah nilai APE dan VEPnya. APE yaitu Arus Puncak Ekspirasi, adalah kecepatan arus ekspirasi maksimal yang dapat dicapai saat ekspirasi paksa. Tarik napas semaksimal mungkin, hembuskan dengan kekuatan maksimal segera setelah kedua bibir dirapatkan pada *mouthpiece*. Sedangkan pengertian VEP yaitu jumlah udara yang dapat dikeluarkan dalam waktu 1 detik, diukur dalam liter. Berdasarkan jurnal dari (PDPI, 2011) berikut tabel yang menyatakan kategori asma pasien :

Tabel 2. 1 Tabel Tingkatan Asma

Derajat Asma	Gejala	Gejala Malam	Faal paru
<b>I. Persisten Ringan</b>	<b>Mingguan</b>		<b>APE <math>\geq</math> 80 %</b>
	Gejala $>$ 1x / minggu, tetapi $<$ 1x / hari	$>$ 2 kali sebulan	VEP <sub>1</sub> $\geq$ 80 % nilai prediksi
<b>II. Persisten Sedang</b>	<b>Harian</b>		<b>APE <math>\geq</math> 80 %</b>
	Gejala setiap hari	$>$ 1x / seminggu	VEP <sub>1</sub> 60 - 80 % nilai prediksi
<b>III. Persisten Berat</b>	<b>Berkelanjutan</b>		<b>APE 60 - 80 %</b>
	Gejala terus menerus	Sering	VEP <sub>1</sub> $\leq$ 60 % nilai prediksi

Pengertian dari macam – macam asma dari 3 kategori diatas, diantaranya :

1. Asma Ringan

Yang pertama yaitu asma ringan dengan ciri – ciri penderita mengalami serangan paling tidak sekali seminggu, tetapi kurang dari sekali sehari dalam waktu 3 bulan dan beberapa serangan mempengaruhi tidur dan aktivitas, dan atau jika pasien memiliki gejala kronis yang memerlukan pengobatan hampir setiap hari dan kejadian gejala asma nokturnal lebih dari 2 kali sebulan. Nilai APE lebih atau sama dengan 80%

prediksi dan lebih VEP lebih atau sama dengan 80% dengan variabiliti APE 20% - 30%.

### 2. Asma Sedang

Selanjutnya adalah asma sedang dengan ciri – ciri ditandai gejala harian dalam jangka waktu lama atau serangan asma nokturnal lebih dari sekali seminggu. Nilai APE berada pada kisaran 60% sampai dengan 80%, dan nilai VEP pada kisaran 60% - 80% dengan variabiliti APE > 30%.

### 3. Asma Berat

Kemudian yang terakhir yaitu asma berat dengan ciri – ciri penderita mengalami serangan yang besar, gejala yang terus menerus dan gejala nokturnal yang sering, mempunyai aktivitas yang terbatas, dan kadang mengalami serangan berat walaupun sedang dalam pengobatan. Nilai APE kurang dari 60% dan nilai VEP kurang dari 60% dengan variabiliti APE > 30%.

Kemudian menurut Achmad Hudoyo (Hudoyo, 2014) berdasarkan tiga macam tingkatan asma tersebut, terdapat bentuk penanganan asma yang berbeda – beda diantaranya adalah sebagai berikut :

Tabel 2. 2 Penanganan

<b>Berat Asma</b>	<b>Medikasi Pengontrol Harian</b>	<b>Alternatif</b>
Asma Persisten Ringan	Glukokortikosteroid inhalasi (200-400 ug BD/hari <i>atau</i> ekivalennya)	Teofilin lepas lambat
Asma Persisten Sedang	Kombinasi inhalasi glukokortikosteroid (400-800 ug BD/hari <i>atau</i> ekivalennya) dan agonis beta-2 kerja lama	Glukokortikosteroid inhalasi (400-800 ug BD <i>atau</i> ekivalennya) ditambah Teofilin lepas lambat

Asma Berat	Persisten	Kombinasi inhalasi glukokortikosteroid (> 800 ug BD atau ekivalennya) dan agonis beta-2 kerja lama, ditambah $\geq 1$ di bawah ini: - teofilin lepas lambat - <i>leukotriene modifiers</i> - glukokortikosteroid oral	Prednisolon/ metilprednisolon oral selang sehari 10 mg ditambah agonis beta-2 kerja lama oral, ditambah teofilin lepas lambat
---------------	-----------	--	--

### 2.2.2 Kecerdasan Buatan

Menurut (T.Sutojo, 2010) kecerdasan buatan atau yang lebih dikenal dengan *Artificial Intelligence* (AI) merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil dan mampu mengambil keputusan seperti yang dilakukan oleh manusia. AI dikembangkan pertama kali pada tahun 1960-an ketika John McCarthy dari *Massachusetts Institute of Technology* (MIT) menciptakan bahasa pemrograman LISP. Kemudian berkembang dengan dibuatnya program komputer yang “berpikir” seperti permainan catur dan pembuktian perhitungan matematis secara komputasi. Pada tahun 1964, Joseph Weizenbaurn juga dari MIT membuat ELIZA, sebuah program yang menggambarkan konsultasi seorang psikiater dengan pasiennya.

Dengan adanya kecerdasan buatan, diharapkan tidak menutup kemungkinan hanya dengan data pengetahuan yang terbatas, sebuah komputer dapat berpikir seperti manusia dalam menghadapi masalah.

Implementasi kecerdasan buatan dapat diterapkan pada bidang:

#### 1. Visualisasi komputer

Kecerdasan buatan pada bidang visualisasi komputer ini memungkinkan sebuah sistem komputer mengenali gambar sebagai input. Contohnya mengenali sebuah pola pada suatu gambar.

#### 2. Pengenalan Suara

Kecerdasan buatan pada pengenalan suara ini dapat mengenali suara manusia. Cara mengenali suara ini dengan mencocokkannya pada acuan yang telah diprogramkan terlebih dahulu. Contohnya perintah komputer dengan menggunakan suara user.

### 3. Sistem Pakar

Menurut (Siswanto, 2010) sistem pakar atau sistem berbasis pengetahuan merupakan yang paling banyak aplikasinya dalam membantu menyelesaikan masalah-masalah dalam dunia nyata. Kecerdasan buatan pada sistem pakar ini memungkinkan sebuah sistem komputer memiliki cara berpikir dan penalaran seorang ahli untuk membantu dan mempermudah dalam mengambil keputusan yang digunakan sebagai pemecah masalah yang ada pada saat itu. Contohnya program komputer yang dapat mendiagnosa penyakit dengan memasukan gejala-gejala yang dialami pasien.

Sistem pakar adalah ilmu yang berdasarkan proses manusia berpikir, maka penelitian bagaimana proses manusia berpikir adalah hal yang pokok. Hasil akhir dari semua proses berpikir tersebut disebut tujuan (goal). Berdasarkan fakta-fakta yang ada keterangan diatas, maka komputer yang menggunakan sistem pakar dapat digunakan sebagai alat bantu dan pengganti pakar dalam pengambilan keputusan dari suatu masalah.

Menurut (Fitriastuti, 2009) suatu aplikasi sistem pakar memanfaatkan teknologi sistem pakar yang berfungsi sebagai pengganti organisasi/seorang ahli dalam bidangnya. Sistem pakar timbul karena adanya permasalahan pada suatu bidang khusus yang spesifik dimana *user* menginginkan suatu solusi dari permasalahan tersebut diselesaikan dengan mendekati cara – cara pakar dalam menyelesaikan masalah.

### 4. Permainan

Kecerdasan buatan pada permainan ini memungkinkan sebuah sistem komputer untuk memiliki cara berpikir manusia dalam bermain. Contohnya permainan yang memiliki fasilitas orang melawan komputer. Komputer sudah di program sedemikian rupa agar memiliki cara bermain seperti seorang manusia bahkan bisa melebihi seorang manusia.

Menurut (Jusak, 2010) secara umum struktur sebuah sistem pakar terdiri atas tiga komponen utama, yaitu; knowledge base, working memory dan inference engine.

1. Knowledge base (basis pengetahuan) adalah bagian dari sebuah sistem pakar yang mengandung/menyimpan pengetahuan (domain knowledge). Knowledge base yang dikandung oleh sebuah sistem pakar berbeda antara satu dengan yang lain tergantung pada bidang kepakaran dari sistem yang dibangun. Misalnya, medical expert system akan memiliki basis pengetahuan tentang hal-hal yang berkaitan dengan medis. Knowledge base direpresentasikan dalam berbagai macam bentuk, salah satunya adalah dalam bentuk sistem berbasis aturan (ruled-based system).

2. Working memory mengandung/menyimpan fakta-fakta yang ditemukan selama proses konsultasi dengan sistem pakar. Selama proses konsultasi, user memasukkan fakta-fakta yang dibutuhkan. Kemudian sistem akan mencari padanan tentang fakta tersebut dengan informasi yang ada dalam knowledge base untuk menghasilkan fakta baru. Sistem akan memasukkan fakta baru ini ke dalam working memory. Jadi working memory akan menyimpan informasi tentang fakta-fakta yang dimasukkan oleh user ataupun fakta baru hasil kesimpulan dari sistem.

3. Inference engine bertugas mencari padanan antara fakta yang ada di dalam working memory dengan fakta-fakta tentang domain knowledge tertentu yang ada di dalam knowledge base, selanjutnya inference engine akan menarik/mengambil kesimpulan dari problem yang diajukan kepada system.

### 2.2.3 Certainty Factor

Menurut (T.Sutojo, 2010) awal mula teori certainty factor (CF) diusulkan oleh Shortlife dan Buchanan pada 1975 untuk mengakomodasi ketidakpastian pemikiran seorang pakar. Seorang pakar/ahli dalam hal ini biasanya dokter sering kali menganalisis informasi yang ada dengan ungkapan seperti “mungkin”, “kemungkinan besar”, “hampir pasti”. Untuk mengakomodasi hal ini kita menggunakan certainty factor guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi.



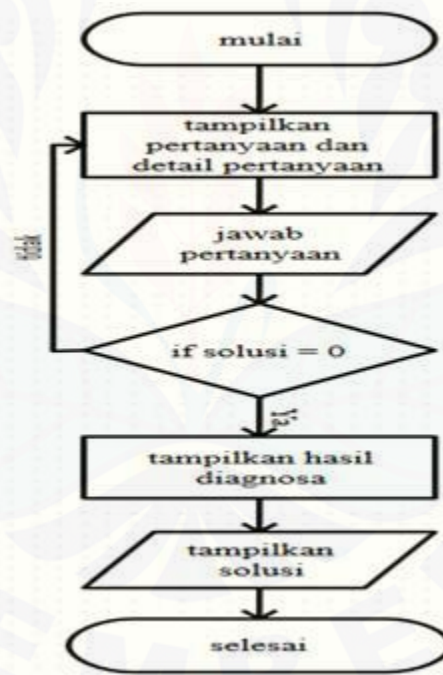
Berdasarkan pengertian tersebut, keakuratan dari *certainty factor* ini memiliki tingkat akurasi yang bagus dikarenakan metode ini memiliki beberapa keuntungan yaitu :

1. Metode ini cocok dipakai dalam sistem pakar yang mengandung ketidakpastian.
2. Dalam sekali proses perhitungan hanya dapat mengolah 2 data saja sehingga keakuratan data tetap terjaga.

Sedangkan kekurangan metode *certainty factor* adalah :

1. Untuk data lebih dari 2 buah, harus dilakukan beberapa kali pengolahan data.

Berikut adalah gambaran flowchart dari metode *certainty factor* :



Gambar 2.2 1 Flowchart Metode Certainty Factor

Pada gambar 1 proses input masuk dari pasien atau pun dari dokter, setelah itu akan dimunculkan beberapa pertanyaan yang bersangkutan dengan diagnosa dari kemungkinan nilai indikator berapa pasien menderita asma. Setelah itu sistem akan menghitung nilai CF, rumus dari menghitung nilai CF tiap gejala adalah :

$$CF(H,E) = MB(H,E) - MD(H,E).....persamaan 1$$

Keterangan :

CF(H,E) : CF dari nilai sebuah gejala dari hipotesis H, terhadap *evidence* E

MB(H,E) : Nilai kepercayaan gejala dari hipotesis H, terhadap *evidence* E

MD(H,E) : Nilai ketidakpercayaan gejala dari hipotesis H, terhadap *evidence* E

Setelah ditemukannya nilai CF dari setiap gejala yang telah dan kemudian dicarilah nilai kombinasi dari nilai CF tersebut yang menggunakan rumus :

$$CF_{combine} = CF[H,E]_n + CF[H,E]_{n+1} * (1 - CF[H,E]_n).....persamaan2$$

Keterangan :

CFcombine : CF kombinasi dari beberapa gejala

CF[H,E]<sub>n</sub> : CF dari H(hipotesis H) terhadap E(evidence) ke n

CF[H,E]<sub>n+1</sub> : CF dari H(hipotesis H) terhadap E(evidence) ke (n + 1)

### **BAB 3. METODOLOGI PENELITIAN**

Pada bagian ini membahas mengenai metode penelitian pembangunan Sistem Pakar Prediksi Tingkatan Asma Pasien. Pada bab ini pula akan memaparkan langkah dan prosedur yang dilakukan dalam mengumpulkan data atau informasi empiris guna memecahkan permasalahan dalam penelitian ini.

#### **3.1 Jenis Penelitian**

Pada penelitian ini digunakan dua jenis penelitian, yaitu penelitian kuantitatif dan penelitian kualitatif. Jenis kuantitatif digunakan karena model analisa menggunakan angka, data yang digunakan merupakan data hasil observasi dan studi literatur dan jenis penelitian kualitatif digunakan karena dalam penelitian ini menerapkan serta mengkaji teori yang telah ada sebelumnya.

#### **3.2 Waktu dan Tempat Penelitian**

Tempat yang dilaksanakan untuk penelitian ini adalah puskesmas Bondowoso, Jawa Timur. Waktu penelitian dilakukan selama 3 bulan, dimulai bulan Januari 2016 sampai dengan bulan Maret 2016

#### **3.3 Alat Penelitian**

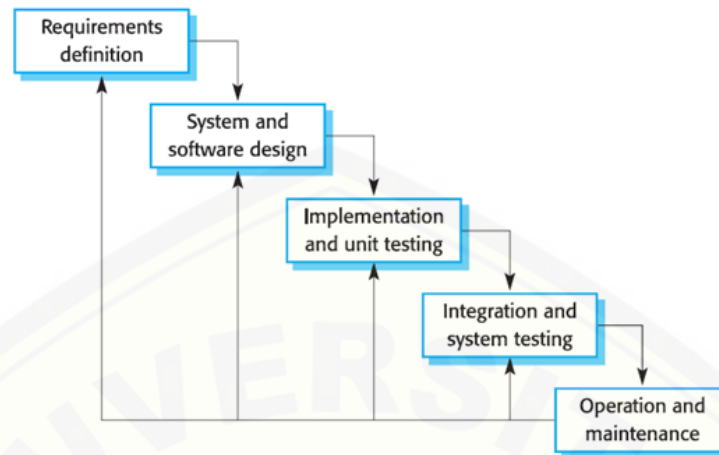
Alat yang digunakan untuk proses penelitian ini meliputi *hardware* berupa satu laptop dan *software* sebagai berikut:

1. *Windows 7*
2. *Sublime Text 2*

3. *DBMS MySQL*
4. *Xampp*
5. *Google Chrome*
6. *Ms. Office*
7. *Ms. Visio*
8. *Ms. Excel*
9. *Unified Modeling Language*
10. *yEd Graph Editor*

#### **3.4 Tahapan Penelitian**

Penelitian ini akan dilakukan dalam beberapa tahap yang disesuaikan dengan metode Software Defelopment Life Cycle (SDLC) waterfall, yang dibagi menjadi beberapa tahap yaitu : analisis kebutuhan, desain sistem, implementasi, dan pengujian. Waterfall model merupakan salah satu model proses perangkat lunak yang mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi dan evolusi dengan mempresentasikannya sebagai fase-fase proses yang berbeda seperti analisis dan definisi persyaratan, perancangan perangkat lunak, implementasi dan pengujian unit, integrasi dan pengujian sistem, operasi dan pemeliharaan (Sommerville, 2011). Sebagaimana terlihat pada gambar berikut.



Gambar 3. 1 Tahapan Penelitian

### 3.4.1 Tahapan Analisa Kebutuhan

Tahapan analisis kebutuhan pada penelitian yang telah dilakukan tersebut dimulai dengan tahap pengumpulan data meliputi studi pustaka, observasi, dan pencarian jurnal. Tahapan yang selanjutnya yaitu tahap pengolahan data dan gambaran umum sistem yang akan dibangun.

#### 3.4.1.1 Tahap Pengumpulan Data

Pada tahap ini menganalisa kebutuhan yang dilakukan adalah mengumpulkan data yang diperlukan untuk memprediksi penyakit asma pada pasien, serta menentukan kriteria apa saja yang layak untuk dijadikan parameter. Teknik pengumpulan data yang dilakukan antara lain :

1. Studi literatur

Teknik ini dilakukan untuk menyusun dasar teori yang digunakan dalam penelitian. Sumber yang digunakan untuk memperoleh studi literatur antara lain buku, jurnal, karya ilmiah, dan situs web.

2. Wawancara

Wawancara merupakan teknik dimana peneliti bertemu langsung dengan narasumber yang berkaitan untuk menggali informasi yang dibutuhkan melalui pertanyaan. Dalam penelitian ini peneliti bertanya langsung pada narasumber untuk memperoleh data serta informasi dalam penyelesaian penelitian.

### 3. Pengolahan Data

Pengolahan data dilakukan untuk menentukan kriteria apa saja yang layak untuk dijadikan parameter, dan menentukan kebutuhan sistem berdasarkan studi literatur dan wawancara yang telah dilakukan.

#### 3.4.2 Tahapan Desain Sistem

Tahapan selanjutnya adalah desain sistem, desain yang dibuat harus mengimplementasikan kebutuhan sistem. Untuk proses perancangan desain ini, penulis menggunakan *software Unified Modeling Language (UML)*. Konsep program menggunakan *Object-Oriented Programming (OOP)*. Pada bagian desain, pengerjaan yang dilakukan diantaranya adalah:

1. *Business Process*

*Business Process* merupakan gambaran dari masuknya data serta data yang dihasilkan dari proses yang dijalankan sistem. Data yang dibutuhkan oleh sistem (*input*), keluaran data yang dihasilkan (*output*), media dari sistem (*uses*), dan tujuan dari pembuatan sistem (*goal*).

2. *Use Case Diagram*

*Use Case Diagram* merupakan dokumentasi yang menggambarkan fitur dan aktor yang mampu mengakses fitur-fitur pada sistem yang akan dibangun. Aktor yang dapat mengakses aplikasi ini yaitu pihak dokter sebagai admin atau pakar.

3. *Use Case Scenario*

*Use Case Scenario* merupakan penjelasan alur sistem sesuai dengan yang tertera pada *use case diagram*, skenario juga menjelaskan reaksi yang akan terjadi pada sistem setelah menerima perlakuan dari aktor. *Use case scenario* menggambarkan

keadaan normal saat setiap aktor mengakses aplikasi dan keadaan alternatif yang terjadi pada kondisi tertentu.

#### 4. *Sequence Diagram*

*Sequence diagram* merupakan diagram yang digunakan untuk menggambarkan interaksi yang terjadi di dalam sistem sesuai dengan urutan dijalankannya sistem tersebut. Interaksi tersebut meliputi, aktor yang mengakses fitur tersebut sesuai dengan yang ada pada *use case diagram*, tampilan sistem, *controller*, model, dan pesan yang disampaikan jika terjadi suatu kondisi tertentu.

#### 5. *Activity Diagram*

*Activity diagram* menggambarkan alur aktivitas dalam sistem yang sedang dirancang, meliputi awal alur dimulai, *decision* yang terjadi, dan bagaimana alur berakhir yang dapat dilakukan oleh masing-masing aktor. *Activity diagram* juga menggambarkan aktivitas sistem setelah menerima perlakuan dari aktor tersebut.

#### 6. *Class Diagram*

*Class diagram* merupakan sebuah spesifikasi yang menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan kelas, *function*, dan atribut yang dibutuhkan oleh seorang programmer dalam membangun sistem.

#### 7. *Entity Relationship Diagram (ERD)*

*Entity relationship diagram* merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

### 3.4.3 Tahapan Implementasi Sistem

Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain yang sudah dirancang harus diubah bentuknya ke dalam bahasa pemrograman melalui proses *coding*, yang akhirnya menjadi sebuah sistem. Tahap ini merupakan

implementasi dari tahap desain yang secara teknis. Dalam pengkodean ini, penulis menggunakan atau berbasis *website*. Bentuk *coding* yang digunakan penulis yaitu *HTML*, *CSS*, dan *PHP* untuk *website* dengan *framework codeigniter* (CI). Manajemen basis data yang digunakan dalam membangun sistem yaitu *DBMS MySQL*.

#### 3.4.4 Tahapan Pengujian Sistem

Setelah pengkodean selesai, maka yang dilakukan selanjutnya adalah pengujian terhadap aplikasi yang telah dibangun. Pengujian aplikasi dilakukan dengan dua cara yaitu:

1. *White Box Testing*

*White box testing* merupakan pengujian perangkat lunak yang dilakukan dengan cara melihat modul yang telah dibuat dengan program yang ada. Pengujian *white box* ini menggunakan *Cyclomatic Complexity* (CC). CC merupakan alat pengukuran untuk mengindikasikan kekompleksitasan program dengan menelusuri jalur-jalur yang ada melalui *source codenya*. Rumus CC dapat dilihat pada persamaan berikut

$$CC = (Edge - Node) + 2$$

2. *Black Box Testing*

*Black box testing* merupakan pengujian perangkat lunak dengan cara memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja saat program dijalankan. Pengujian dilakukan dengan menjalankan program secara langsung dan menganalisis *input* serta *output* yang dihasilkan aplikasi. *Black box testing* dilakukan oleh *user* yang menjalankan aplikasi. Yang terakhir dilakukan dalam tahapan model *waterfall* ini adalah tahapan pemeliharaan. Sistem yang sudah selesai akan mengalami perubahan. Perubahan biasanya berupa *error* sehingga diperlukan perbaikan dan pemeliharaan pada sistem.

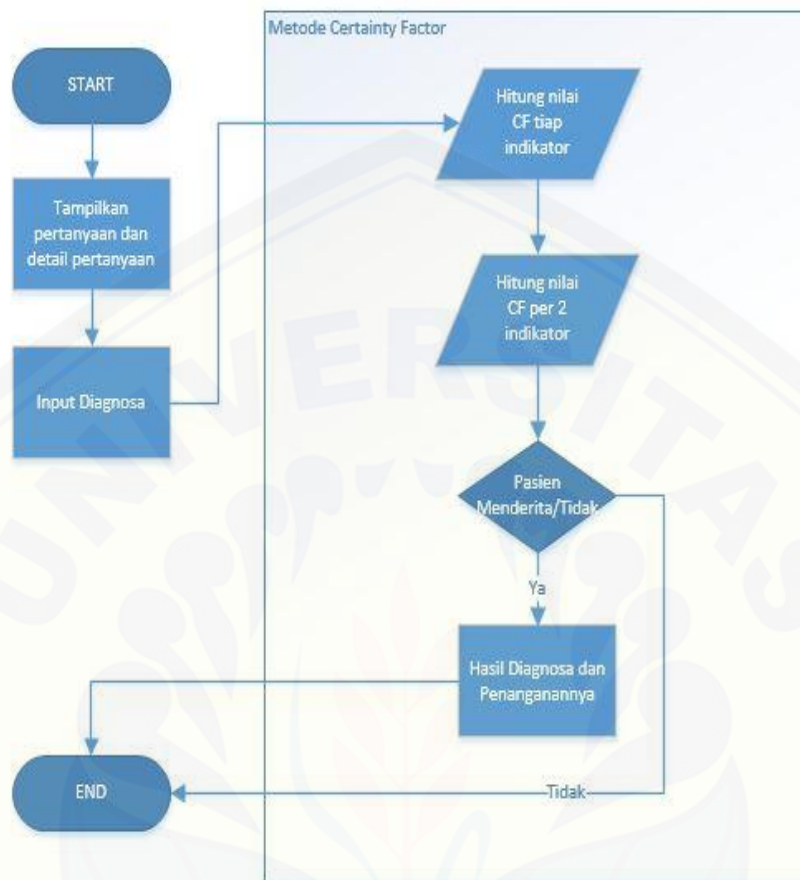


#### 3.4.5 Tahapan Pemeliharaan Sistem

Sistem yang sudah selesai akan mengalami perubahan. Perubahan biasanya berupa *error* sehingga diperlukan perbaikan dan pemeliharaan pada sistem.

### 3.5 Gambaran Umum Sistem

Sistem Informasi yang akan dibangun dalam penelitian ini merupakan sistem informasi berbasis *website* yang akan membantu dokter dan pasien dalam menentukan dan memprediksi termasuk di dalam kategori apakah pasien asma pasien tersebut. Dalam sistem informasi ini membutuhkan input data berupa kategori gejala, dan di dalam setiap kategori terdiri atas tiga gejala yang setiap gejalanya memiliki nilai untuk memberikan nilai atas asma yang di derita oleh pasien. Dari sistem yang dibangun kita mampu melihat termasuk di dalam kategori apakah pasien tersebut dan bagaimana penanganannya. Juga dalam sistem ini nantinya akan di munculkan pertanyaan untuk pasien yang merupakan pertanyaan untuk mengetahui Jadi nantinya dengan adanya sistem ini, mampu membantu pasien dalam menghindari apa saja yang harus dijauhinya agar penyakit asmanya bisa terhindar. Selain itu, sistem ini juga mampu membantu dan menghemat kerja dokter dan pihak rumah sakit maupun puskesmas dalam menangani pasien, jadi pasien penderita asma dapat ditangani dengan tepat dan cepat. Berdasarkan gambaran sistem, Alur berjalannya sistem secara umum dapat dilihat pada Gambar berikut.



Gambar 3. 2 *Flowchart* Sistem

## BAB 4. PENGEMBANGAN SISTEM

Bab ini akan menguraikan tentang pengembangan sistem serta implementasi metode certainty factor. Tahap pengembangan sistem dilaksanakan berdasarkan mode *waterfall*, dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program, dan pengujian sistem.

### 4.1 Analisis Kebutuhan Sistem

Analisis kebutuhan perangkat lunak yaitu mengidentifikasi kebutuhan yang diperlukan sehingga menjadi landasan untuk membangun sebuah website untuk memprediksi tingkatan penyakit asma dari pasien. Dari hasil pengumpulan data yang dilakukan, dapat diketahui bahwa proses perhitungan ini membutuhkan nilai dan gejala yang dimasukkan oleh dokter sebagai admin atau pakar, kemudian nilai dari suatu gejala akan dihitung dengan gejala lainnya, sehingga nantinya akan ditemukan hasil berupa analisis pasien tersebut menderita asma kategori apa, dan bagaimana pula penanganannya. Analisis kebutuhan perangkat lunak melingkupi data kebutuhan fungsional dan kebutuhan non-fungsional.

#### 4.1.1 Kebutuhan Fungsional

Kebutuhan fungsional berisi proses-proses yang akan dilakukan oleh sistem. Kebutuhan fungsional dari aplikasi ini adalah sebagai berikut:

1. Sistem mampu mengetahui kategori asma dari pasien
2. Sistem mampu mengelola data gejala
3. Sistem mampu membantu dalam pencegahan asma pasien
4. Sistem mampu membantu dalam mengatasi jika pasien mengalami serangan
5. Sistem dibangun dengan menggunakan metode certainty factor

#### 4.1.2 Kebutuhan Non-Fungsional

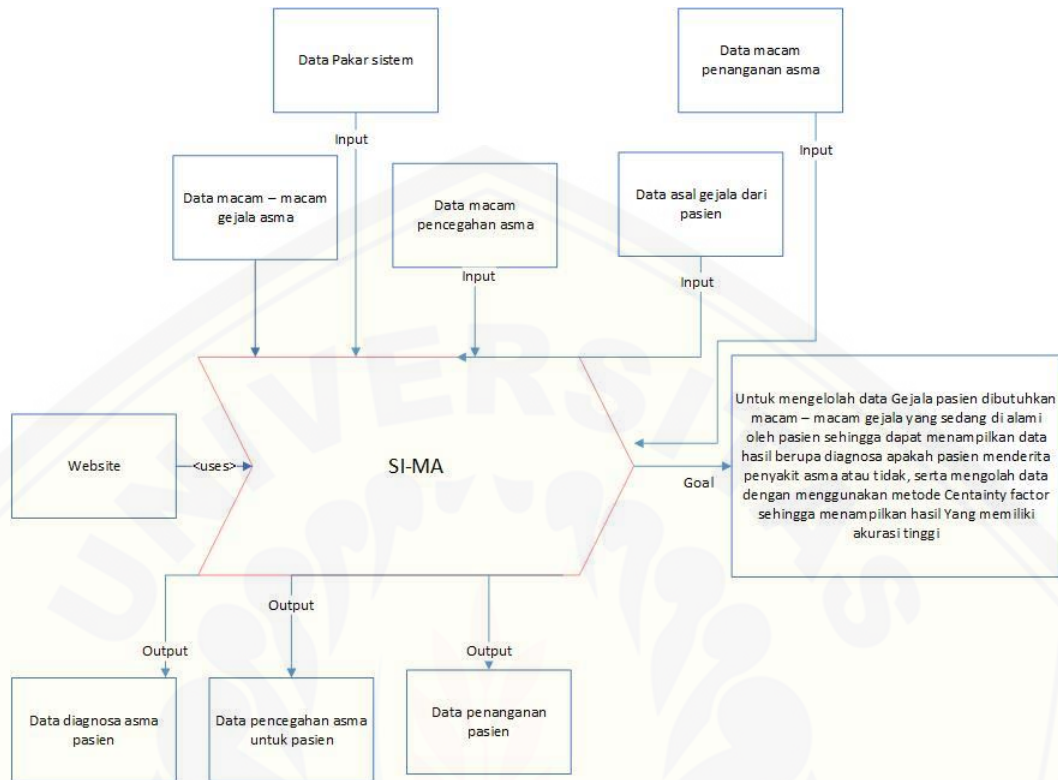
Kebutuhan non-fungsional merupakan hal yang dibutuhkan oleh sistem untuk mendukung aktivitas sistem sesuai dengan kebutuhan fungsional yang ada. Kebutuhan non fungsional sistem pada penelitian ini adalah menampilkan *user interface* yang *user friendly* dengan berbasis website sehingga pengguna bisa menggunakan dengan mudah dan nyaman. Sistem memiliki batasan hak akses pengguna dengan menggunakan *username* dan *password*.

## 4.2 Desain Sistem

Desain sistem yang dibuat berdasarkan gambaran sistem yang telah dijelaskan pada bab metodologi penelitian. Desain yang dibuat meliputi *business process*, *use case diagram*, *use case scenario*, *sequence diagram*, *activity diagram*, *class diagram*, dan ERD.

### 4.2.1 Business process

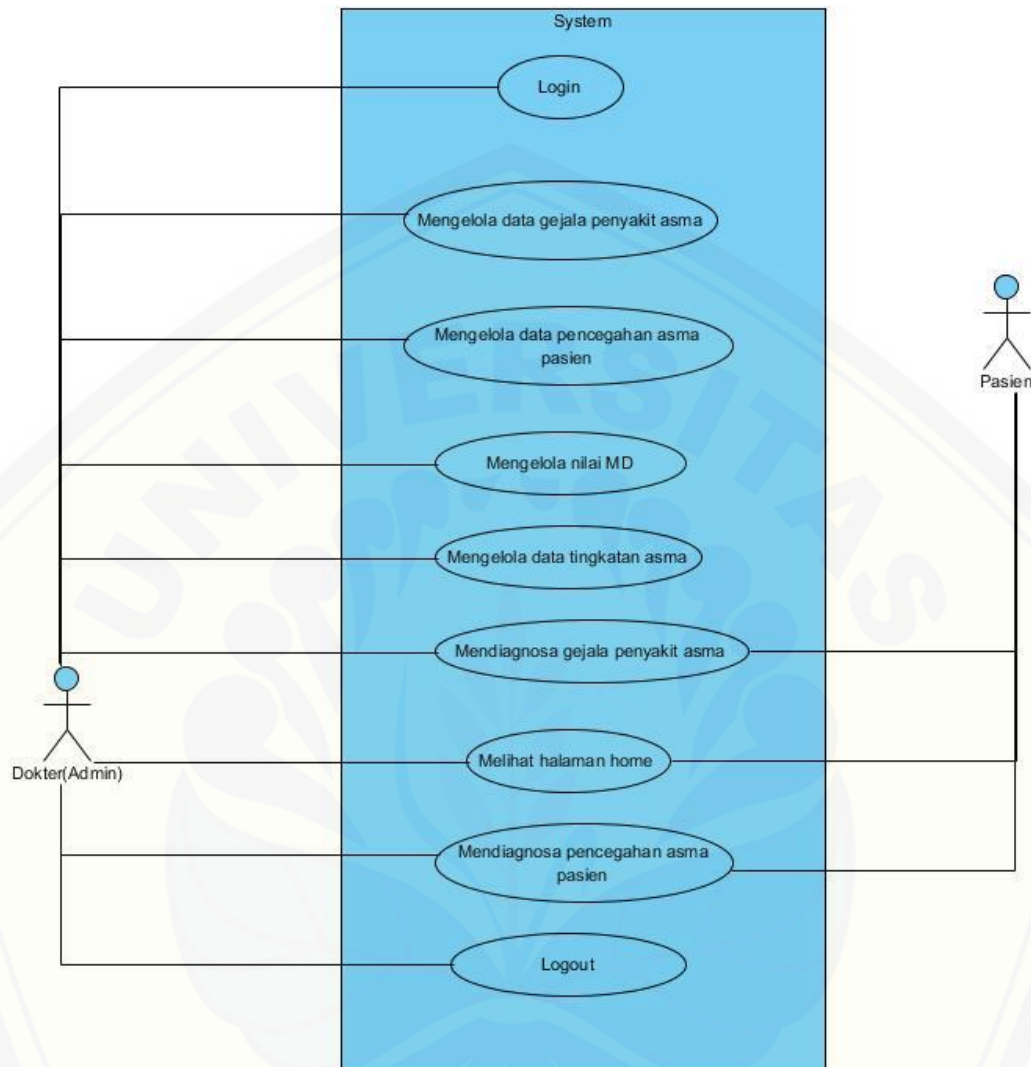
*Business process* merupakan diagram yang menggambarkan kebutuhan data yang dibutuhkan oleh sistem. Ada beberapa komponen di dalamnya, meliputi data yang menjadi masukan (*input*), data masukan yang kemudian diolah menjadi data keluaran (*output*), media yang digunakan (*uses*), dan tujuan yang ingin dicapai (*goal*). *Business process* pada aplikasi ini dapat dilihat pada gambar berikut :



Gambar 4. 1 Business Process

#### 4.2.2 Usecase Diagram

*Usecase diagram* digunakan untuk menggambarkan fitur dan aktor yang terdapat dalam sistem yang dikembangkan. *Usecase diagram* mendeskripsikan interaksi antara suatu aktor dengan sistem yang akan dibuat. *Usecase diagram* bertujuan untuk mengetahui fungsi apa saja yang dapat diakses oleh suatu aktor. *Usecase diagram* sistem dapat dilihat pada gambar berikut



Gambar 4. 2 Usecase Diagram

*Usecase* sistem pakar prediksi tingkatan asma pasien dan penanganannya terdapat penjelasan berupa definisi aktor yang menjelaskan aktor siapa saja yang terdapat pada sistem tersebut dapat dilihat pada table 1, selain itu terdapat definisi *usecase* yang menggambarkan fungsionalitas dari setiap *usecase*. Definisi *usecase* dari *usecase* sistem pakar prediksi tingkatan asma pasien dan penanganannya dapat dilihat pada tabel berikut:

Tabel 4. 1 Macam - macam *user*

No.	Aktor	Definisi Tugas
1	Dokter(Admin)	Melakukan manajemen data pakar, manajemen data pencegahan, manajemen data gejala dan manajemen data gabungan, update nilai
3	<i>User</i>	Melakukan konsultasi asma dan pencegahannya

Tabel 4. 2 Definisi *Usecase*

No.	Aktor	Tugas	Definisi Tugas
1	Dokter(Admin)	Login	Melakukan <i>login</i> untuk masuk ke sistem sebagai admin
2	Dokter(Admin)	Mengelola data gejala penyakit asma	<i>Usecase</i> yang digunakan untuk menambah, mengedit, atau menghapus data gejala penyakit asma
3	Dokter(Admin)	Mengelola data pencegahan	<i>Usecase</i> yang digunakan untuk menambah, mengedit, atau menghapus data tentang pencegahan asma pasien
4	Dokter(Admin)	Melihat halaman home	<i>Usecase</i> yang digunakan untuk masuk ke dalam sistem menu home
5	Dokter(Admin)	Diagnosa gejala penyakit asma	<i>Usecase</i> yang digunakan untuk melakukan diagnosa terhadap pasien

			berdasarkan gejala – gejala yang terjadi pada pasien
6	Dokter(Admin)	Diagnosa pencegahan asma pasien	<i>Usecase</i> yang digunakan untuk melakukan diagnosa pencegahan asma bagi pasien
7	Dokter(Admin)	Logout	<i>Usecase</i> yang digunakan untuk dokter keluar dari hak akses admin
8	Pasien	Melihat halaman home	<i>Usecase</i> yang digunakan untuk masuk ke dalam sistem menu home
9	Pasien	Diagnosa gejala penyakit asma	<i>Usecase</i> yang digunakan untuk melakukan diagnosa terhadap pasien berdasarkan gejala – gejala yang terjadi pada pasien
10	Pasien	Diagnosa pencegahan asma pasien	<i>Usecase</i> yang digunakan untuk melakukan diagnosa

#### 4.2.3 Skenario Sistem

Skenario sistem merupakan gambaran lebih rinci dari *usecase*, skenario berfungsi menggambarkan alur dari sebuah sistem beserta alternatif yang dilakukan oleh aktor. Skenario sistem pakar prediksi tingkatan asma pasien dan penanganannya dapat diketahui pada tabel – tabel dibawah ini.

##### 4.2.3.1. Skenario Login

Skenario ini menjelaskan alur proses *login* dari dokter sebagai admin



Tabel 4. 3 Skenario *Login*

ID	01
Nama Use Case	Login
Aktor	Dokter
PreKondisi	- Sistem online - Sistem menemukan data dokter yang bersangkutan dalam database
PraKondisi	- Data dokter pada sistem telah disimpan - Sistem berada pada halaman dashboard
<b>Normal usecase Login</b>	
<b>Aksi Aktor(dokter)</b>	<b>Reaksi Sistem</b>
1. Membuka web diagnosis asma	
	2. Menampilkan halaman dashboard
3. Mengklik Login	
	4. Menampilkan form login
5. Mengisi form login	
6. Klik OK	
	7. Melakukan pengecekan id dan password dari tabel login
	8. Menampilkan halaman hak ases dokter
<b>Alternatif(ID atau password salah)</b>	
	8. Menampilkan pop up “id atau password salah”
	9. Menampilkan halaman login kembali
10. Mengisi form login	
11. Klik OK	
	12. Menampilkan halaman hak akses dokter

## 4.2.3.2. Skenario mengelola data gejala asma

Skenario ini menjelaskan alur proses mengelola data gejala asma oleh dokter sebagai admin

Tabel 4. 4 Skenario Mengelola Data Gejala Asma

Id	02
Nama Use Case	Mengelola data gejala asma
Aktor	Dokter
PreKondisi	- Sistem online - Sistem berada pada sesi aktif (user telah melakukan login) - Sistem menemukan data gejala asma dalam database (Edit & Hapus)
PraKondisi	- Data pada sistem bertambah / berubah - Sistem berada pada halaman gejala
<b>Normal usecase Tambah Data</b>	
<b>Aksi Aktor(Dokter)</b>	<b>Reaksi Sistem</b>
1. Memilih menu gejala	
	2. Mengambil data gejala asma dari tabel gejala
	3. Menampilkan halaman gejala
4. klik Tambah	
	5. Menampilkan form input
6. Menginputkan nama gejala	
7. Klik OK	
	8. Data baru gejala asma tersimpan di database di dalam tabel gejala
	9. Mengambil semua data gejala asma dari tabel gejala dan menampilkan halaman gejala
<b>Alternatif data isian kosong</b>	

	8. Menampilkan pop up “data tidak boleh kosong”
	9. Kembali ke form input
<b>Alternatif batal</b>	
7. Klik batal	
	8. Menampilkan halaman gejala
<b>Normal usecase Edit data</b>	
<b>Aksi Aktor edit</b>	<b>Reaksi Sistem</b>
1. Memilih menu gejala	
	2. Mengambil data gejala asma dari tabel gejala
	2. Menampilkan gejala
3. Memilih edit data	
4. Klik Simpan	
	5. Kembali ke halaman gejala
<b>Alternatif data kosong</b>	
	5. Menampilkan pop up “data tidak boleh kosong”
6. klik OK	
	7. Menampilkan form input gejala
<b>Alternatif klik back</b>	
4. Klik back	
	5. Kembali ke halaman gejala
<b>Normal usecase delete data</b>	
1. Memilih menu gejala	
	2. Mengambil data gejala asma dari tabel gejala
	2. Menampilkan gejala
3. Memilih delete gejala	
	4. Data pilihan dihapus dari database
	5. Kembali ke halaman gejala

## 4.3.2.3 Skenario mengelola data nilai CF

Skenario ini menjelaskan alur proses mengelola nilai CF oleh dokter sebagai admin

Tabel 4. 5 Skenario Mengelola Data Nilai CF

Id	03
Nama Use Case	Mengelola data nilai CF
Aktor	Dokter
PreKondisi	- Sistem online - Sistem berada pada sesi aktif (user telah melakukan login) - Sistem menemukan data gabungan dalam database (Edit & Hapus)
PraKondisi	- Data pada sistem bertambah / berubah - Sistem berada pada halaman data gabungan
<b>Normal usecase Tambah Data</b>	
<b>Aksi Aktor(Dokter)</b>	<b>Reaksi Sistem</b>
1. Memilih menu data gabungan	
	Menampilkan halaman data gabungan
klik Tambah	
	Menampilkan form input
Menginputkan data nilai	
6. Klik OK	
	7. Data tersimpan di database
	8. Kembali ke halaman data gabungan
<b>Alternatif data isian kosong</b>	
	7. Menampilkan pop up “data tidak boleh kosong”
	8. Kembali ke form input

Alternatif batal	
Klik batal	
	7. Menampilkan halaman data gabungan
Normal usecase delete data	
1. Memilih menu data gabungan	
	2. Menampilkan data nilai
3. Memilih delete	
	4. Data pilihan dihapus dari database
	5. Kembali ke halaman data gabungan

#### 4.3.2.4 Skenario mengelola data pencegahan asma pasien

Skenario ini menjelaskan alur proses melakukan input edit dan delete yang dilakukan oleh dokter sebagai admin

Tabel 4. 6 Skenario Mengelola Data Pencegahan Asma Pasien

Id	04
Nama Use Case	Mengelola data pencegahan asma pasien
Aktor	Dokter
PreKondisi	- Sistem online - Sistem berada pada sesi aktif (user telah melakukan login) - Sistem menemukan data pencegahan asma dalam database (Edit & Hapus)
PraKondisi	- Data pada sistem bertambah / berubah - Sistem berada pada halaman pencegahan
Normal usecase Tambah Data	
Aksi Aktor(Dokter)	Reaksi Sistem
1. Memilih menu pencegahan	
	Menampilkan halaman pencegahan
klik Tambah	
	Menampilkan form input

Mengisi form pencegahan	
6. Klik OK	
	7. Data tersimpan di database
	8. Kembali ke halaman pencegahan
<b>Alternatif data isian kosong</b>	
	7. Menampilkan pop up “data tidak boleh kosong”
	8. Kembali ke form input
<b>Alternatif batal</b>	
Klik batal	
	7. Menampilkan halaman pencegahan
<b>Normal usecase Edit data</b>	
<b>Aksi Aktor edit</b>	<b>Reaksi Sistem</b>
Memilih menu pencegahan	
	2. Menampilkan pencegahan
3. Memilih edit data	
4. Klik Simpan	
	5. Kembali ke halaman pencegahan
<b>Alternatif data kosong</b>	
	5. Menampilkan pesan data kosong atau salah
6. klik OK	
	7. Menampilkan form input gejala
<b>Alternatif klik back</b>	
4. Klik back	
	5. Kembali ke halaman pencegahan
<b>Normal usecase delete data</b>	
1. Memilih menu pencegahan	
	2. Menampilkan pencegahan
3. Memilih delete	
	4. Data pilihan dihapus dari database
	5. Kembali ke halaman pencegahan

## 4.3.2.5 Skenario melihat halaman home

Skenario ini menjelaskan alur proses melakukan *view* halaman *home* dari dokter dan pasien

Tabel 4. 7 Skenario Melihat Halaman Home

ID	05
Nama Use Case	Melihat halaman home
Aktor	Dokter, pasien
PreKondisi	- Sistem online
PraKondisi	- Sistem berada pada halaman dashboard
<b>Normal usecase melihat halaman home</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Membuka web diagnosis asma	
2. Klik home	
	2. Menampilkan halaman dashboard

## 4.3.2.6 Skenario mendiagnosis gejala penyakit asma

Skenario ini menjelaskan alur proses melakukan diagnosa asma pasien berdasarkan bermacam – macam gejala yang telah dimasukkan dan dilakukan oleh dokter atau pasien

Tabel 4. 8 Skenario Mendiagnosis Gejala Penyakit Asma

ID	07
Nama Use Case	Mendiagnosis gejala penyakit asma
Aktor	Dokter, pasien
PreKondisi	- Sistem online
PraKondisi	- Sistem berada pada halaman dashboard
<b>Normal usecase diagnosa gejala penyakit asma</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>

1. Memilih menu diagnosa	
	2. Menampilkan halaman diagnosa
3. Memilih macam gejala	
4. Klik diagnosa	
	5. Menampilkan hasil diagnosa

#### 4.3.2.7 Skenario mendiagnosis pencegahan asma pasien

Skenario ini menjelaskan alur proses melakukan diagnosa untuk pencegahan asma pasien berdasarkan bermacam – macam jawaban dari pertanyaan yang telah dimasukkan dan dilakukan oleh dokter atau pasien

Tabel 4. 9 Skenario Mendiagnosis Pencegahan Asma Pasien

ID	08
Nama Use Case	Mendiagnosis pencegahan asma pasien
Aktor	Dokter, pasien
PreKondisi	- Sistem online
PraKondisi	- Sistem berada pada halaman dashboard
<b>Normal usecase diagnosa gejala penyakit asma</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Memilih menu pencegahan	
	2. Menampilkan halaman pencegahan
Menjawab macam pertanyaan	
4. Klik diagnosa	
	5. Menampilkan hasil diagnosa

#### 4.2.4 Activity Diagram

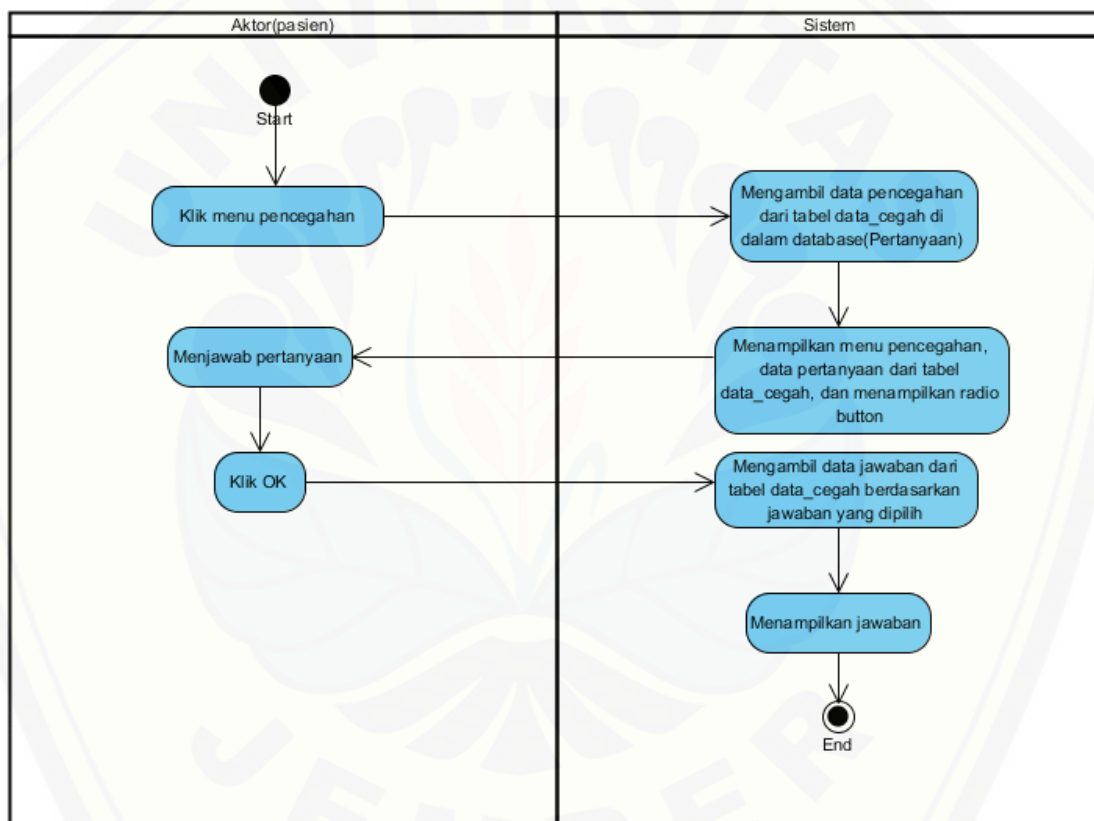
*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang akan dibangun, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan



bagaimana mereka berakhir. *Activity diagram* sistem sistem prediksi penyakit asma dan penanganannya dapat diketahui pada tabel berikut.

#### 4.2.4.1 *Activity diagram* diagnosa pencegahan asma pasien

*Activity diagram* ini menggambarkan alur kegiatan aktor(dokter dan pasien) melakukan diagnosa pencegahan asma pada pasien. Pada kegiatan ini nantinya akan diketahui dari mana pasien dapat mengidap penyakit asma, dan ditentukan juga cara pencegahannya.

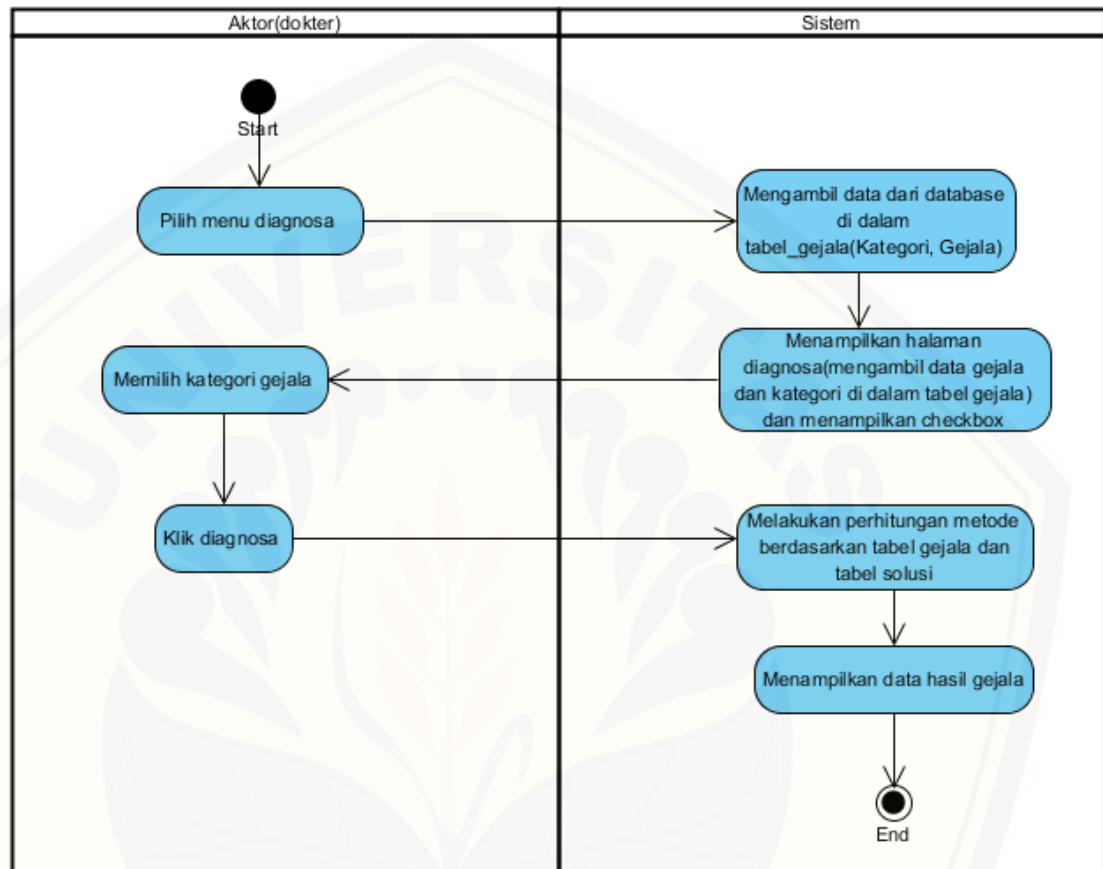


Gambar 4. 3 *Activity Diagram* Diagnosis Pencegahan Asma Pasien

#### 4.2.4.2 *Activity diagram* diagnosa tingkatan asma pasien

*Activity diagram* ini menggambarkan alur kegiatan aktor(dokter dan pasien) melakukan diagnosa asma pasien. Pada kegiatan ini nantinya akan diketahui pasien

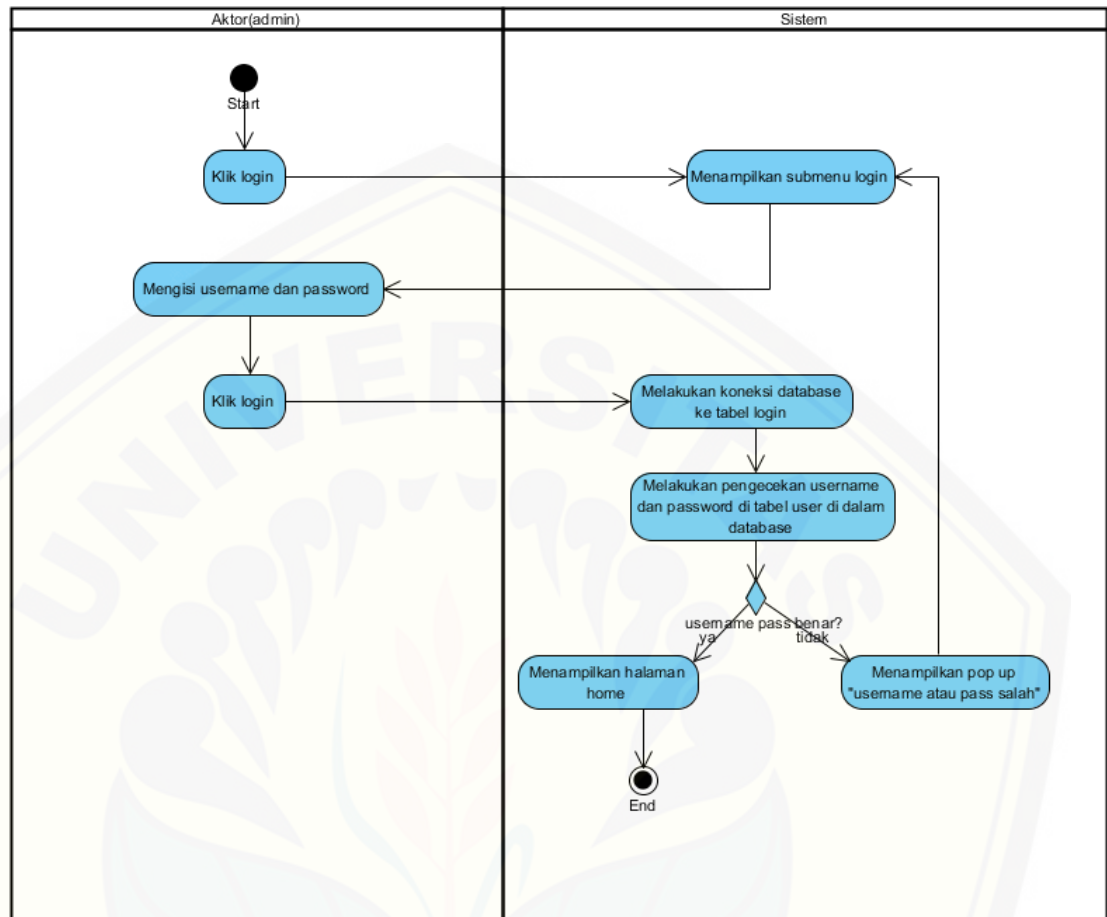
mengidap asma dengan tingkat kategori apa dan bagaimana penanganan yang seharusnya dilakukan jika terjadi serangan.



Gambar 4. 4 *Activity Diagram* Diagnosis Tingkatan Asma Pasien

#### 4.2.4.3 *Activity diagram login(dokter)*

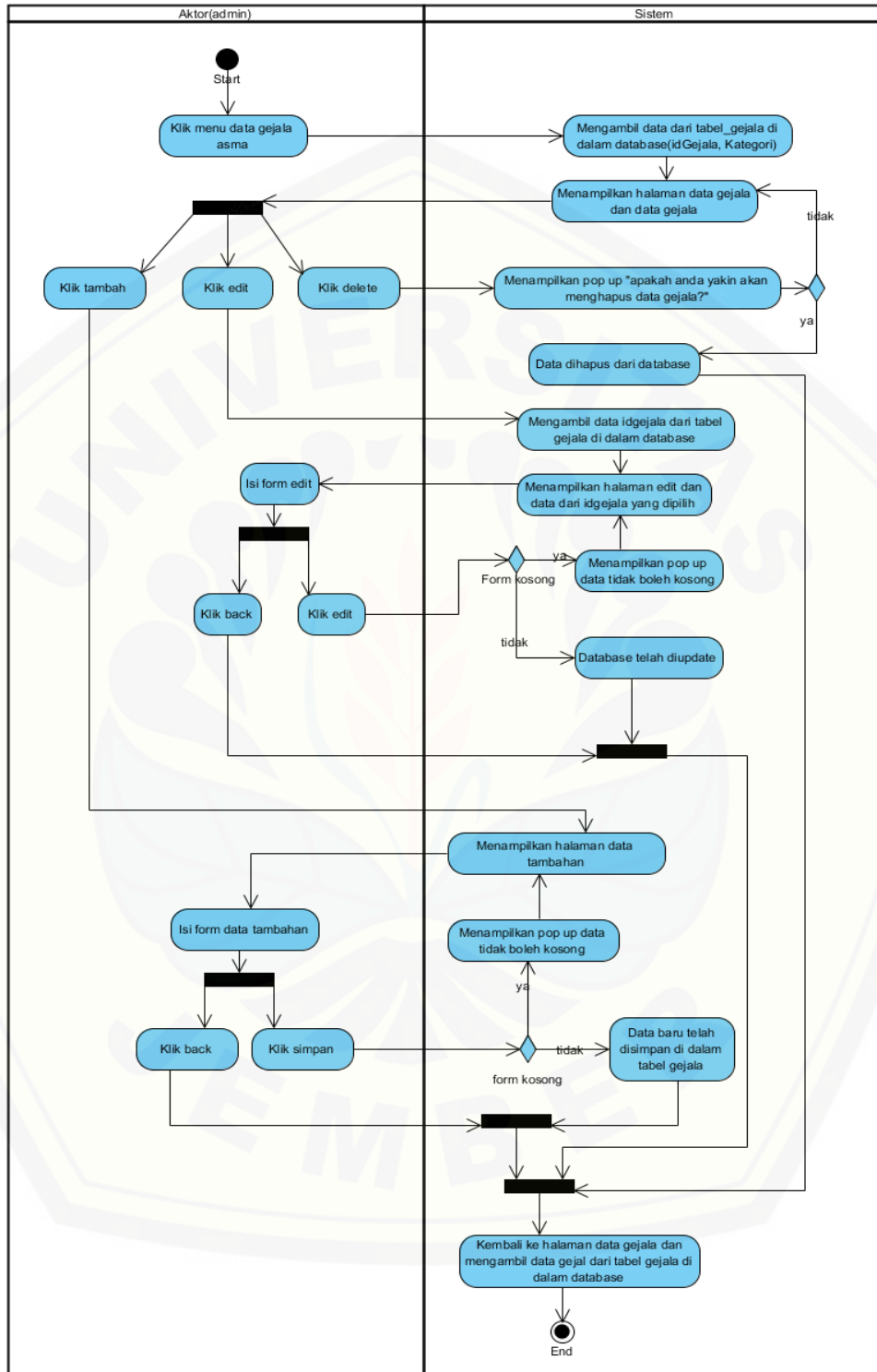
*Activity diagram* ini menggambarkan alur kegiatan dokter melakukan login untuk masuk ke dalam sistem dengan hak ases sebagai admin.



Gambar 4. 5 Activity Diagram Login

#### 4.2.4.4 Activity diagram mengelola data gejala penyakit asma(dokter)

Activity diagram ini menggambarkan alur kegiatan dokter melakukan tambah data, edit data, maupun hapus data gejala penyakit asma. Pada saat membuka menu ini dokter telah melakukan login sehingga dokter mendapatkan hak akses sebagai admin untuk merubah nilai data.

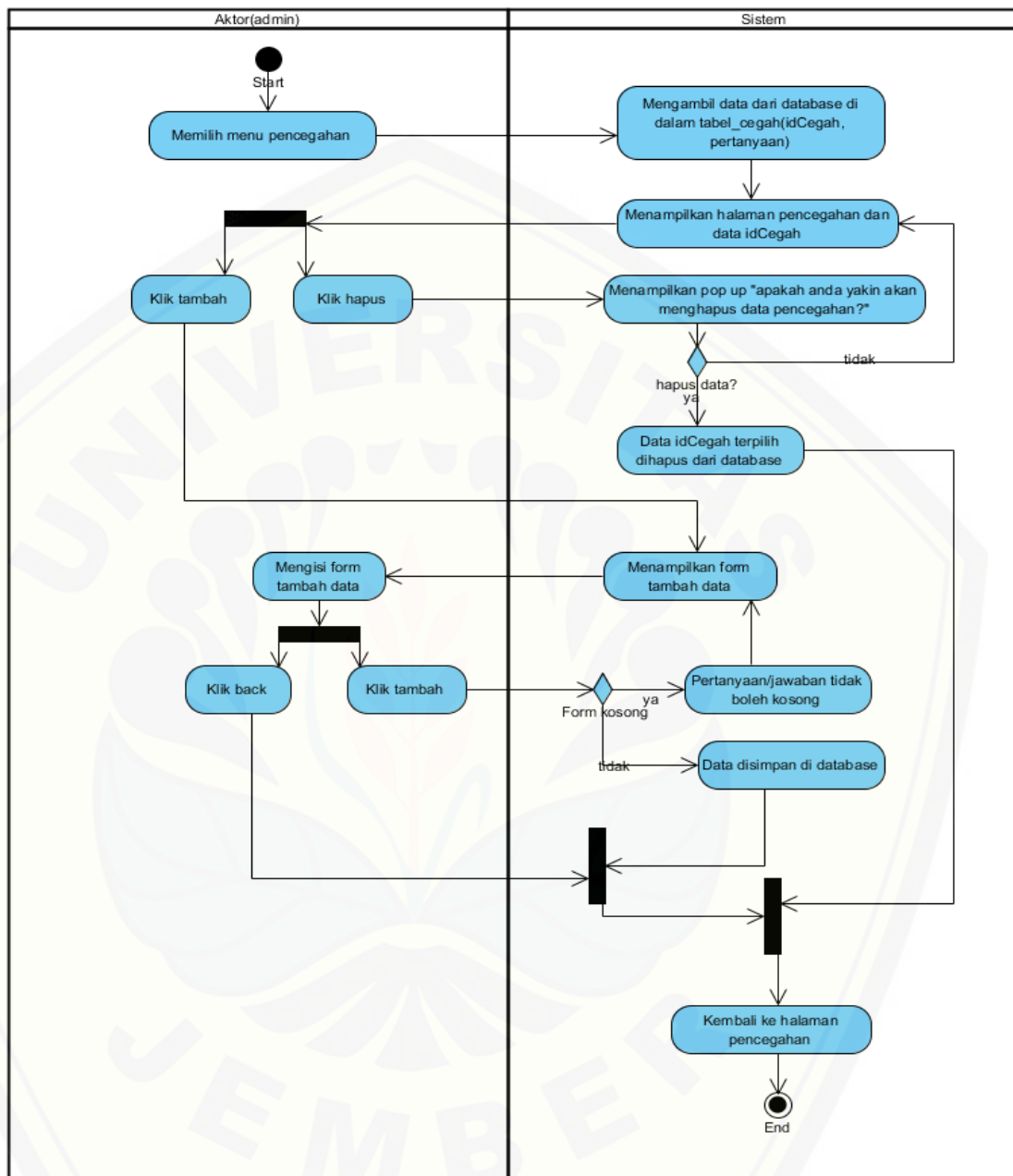


Gambar 4. 6 Activity Diagram Mengelola Data Gejala Penyakit Asma

#### 4.2.4.5 *Activity diagram* mengelola data pencegahan asma(dokter)

*Activity diagram* ini menggambarkan alur kegiatan dokter melakukan tambah data, edit data, maupun hapus data pencegahan asma pada pasien. Pada saat membuka menu ini dokter telah melakukan login sehingga dokter mendapatkan hak akses sebagai admin untuk merubah nilai data.



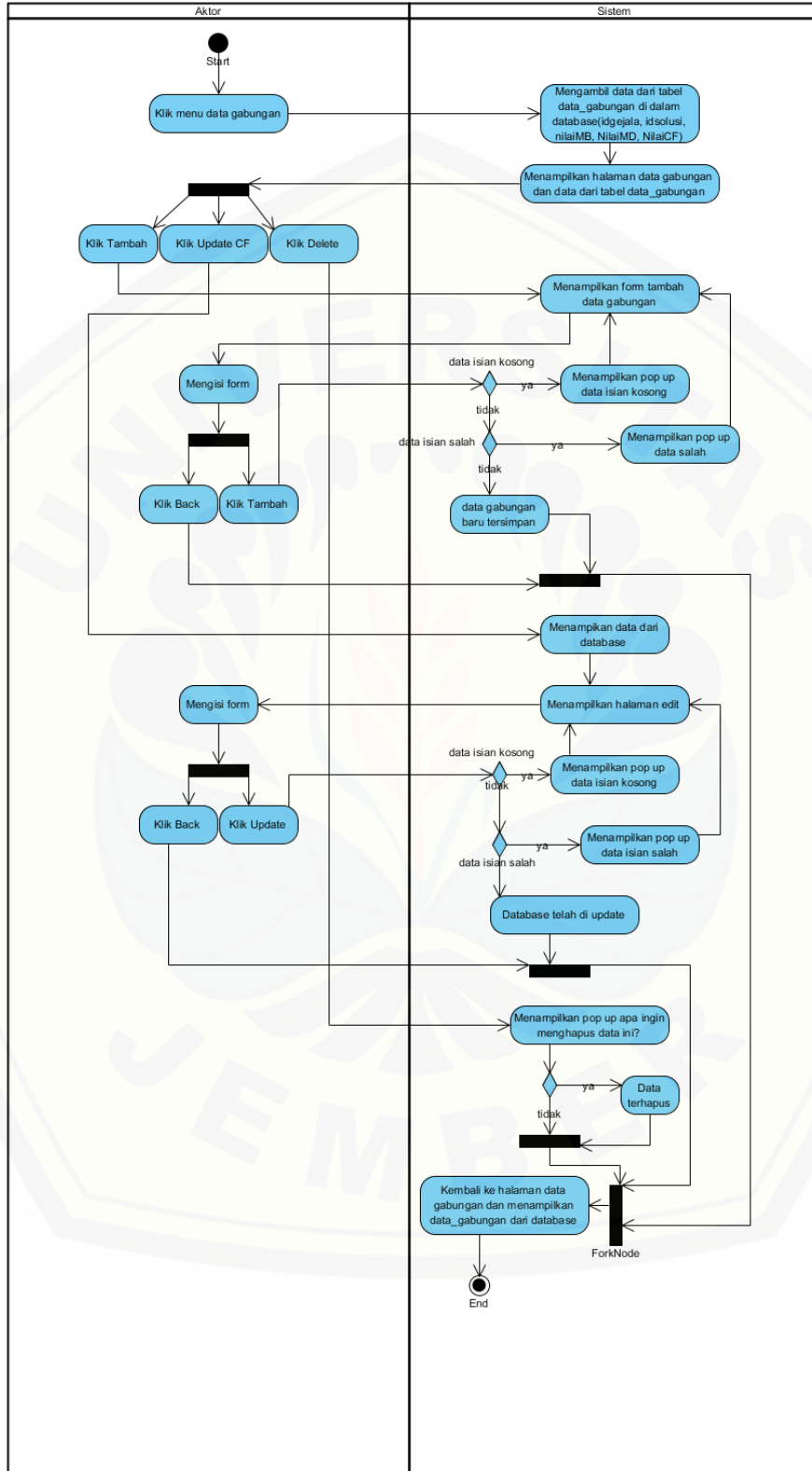


Gambar 4. 7 Activity Diagram Mengelola Data Pencegahan Asma

#### 4.2.4.6 *Activity diagram* mengelola data gabungan(dokter)

*Activity diagram* ini menggambarkan alur kegiatan dokter melakukan tambah data, edit data, maupun hapus data nilai MB, nilai MD, dan nilai CF. Pada saat membuka menu ini dokter telah melakukan login sehingga dokter mendapatkan hak akses sebagai admin untuk merubah nilai data.





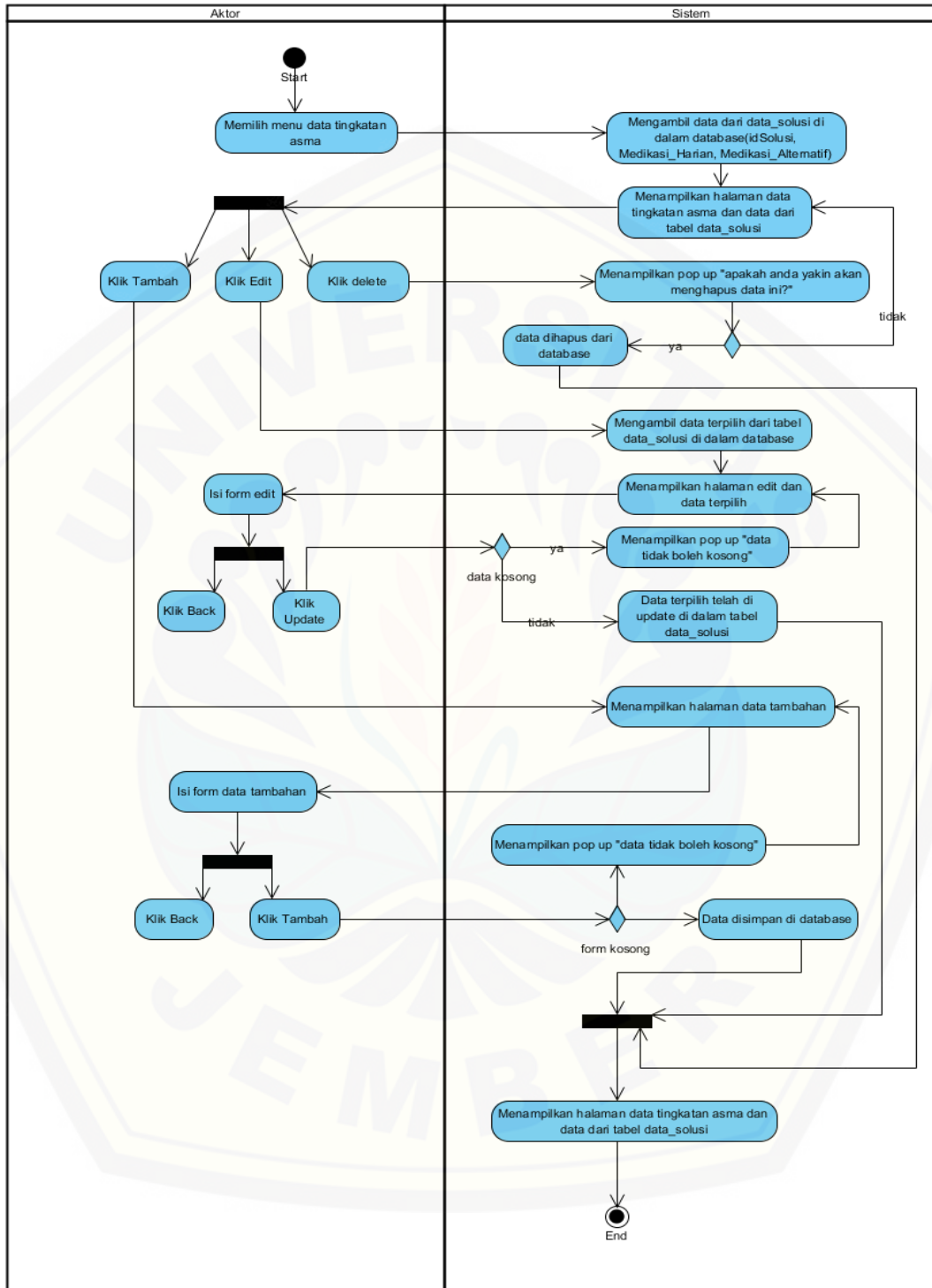


Gambar 4. 8 *Activity Diagram* Mengelola Data Gabungan

#### 4.2.4.7 *Activity diagram* mengelola data tingkatan asma(dokter)

*Activity diagram* ini menggambarkan alur kegiatan dokter melakukan tambah data, edit data, maupun hapus data tingkatan asma. Pada saat membuka menu ini dokter telah melakukan login sehingga dokter mendapatkan hak akses sebagai admin untuk merubah nilai data.





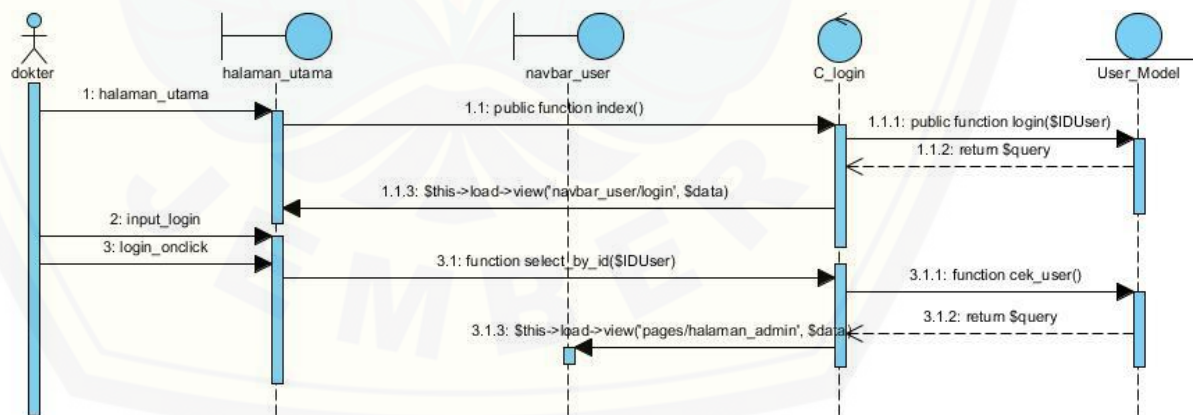
Gambar 4. 9 Activity Diagram Mengelola Data Tingkatan Asma

4.2.5 Sequence Diagram

Sequence diagram merupakan diagram yang digunakan untuk menggambarkan interaksi yang terjadi di dalam sistem sesuai dengan urutan dijalankannya sistem tersebut. Interaksi tersebut meliputi, aktor yang mengakses fitur tersebut sesuai dengan yang ada pada usecase diagram, tampilan sistem, controller, model, dan pesan yang disampaikan jika terjadi suatu kondisi tertentu. Sequence diagram dirancang berdasar usecase skenario sehingga tahapan yang dibangun sudah jelas dan teratur. Pada tahap ini juga bisa dijelaskan fungsi atau method apa saja yang akan digunakan untuk membangun aplikasi pada penelitian ini.

4.2.5.1 Sequence Diagram login dokter

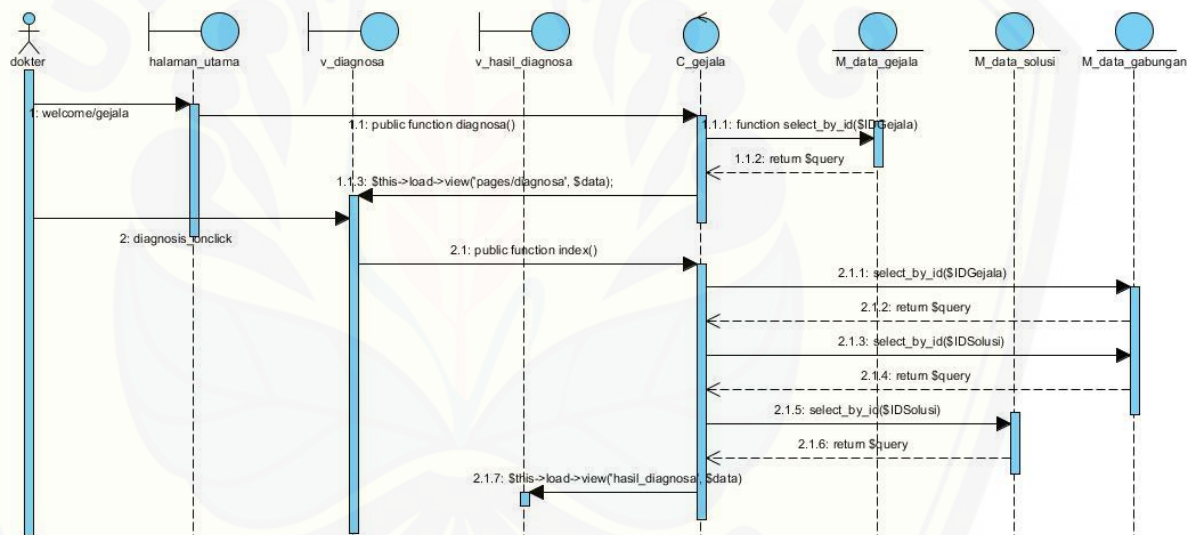
Pada sequence login dapat dilihat alur perjalanan suatu fitur login aktor(pakar) dengan pertama – tama melakukan login melalui klik menu login yang ada pada menu dan kemudian controller login akan melakukan pengecekan apakah username dan password dari aktor sudah benar atau tidak, jika benar maka akan menuju ke menu halaman home milik pakar, jika tidak maka akan kembali ke halaman utama dengan menampilkan tulisan login gagal, username atau password salah. Gambaran alur sequence diagram login dokter ada pada gambar berikut.



Gambar 4. 10 Sequence Diagram Login

4.2.5.2 Sequence Diagram diagnosa asma pasien

Pada menu diagnosa asma pasien pertama setelah aktor klik menu diagnosa maka database akan mengambil data gejala yang ada dan lalu aktor akan memilih gejala – gejala yang diinginkan. Setelah gejala diinputkan sistem akan melakukan pengecekan *query* berdasarkan data – data yang telah dipilih oleh aktor dan lalu menampilkan hasil diagnosis asma pasien pada halaman hasil diagnosa. Penggambaran *sequence diagram* diagnosa asma pasien seperti pada gambar dibawah menjelaskan fungsi atau method apa saja yang akan dibuat. Pada gambar tersebut dapat dilihat alur perjalanan suatu fitur antara aksi yang dilakukan oleh aktor, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut.

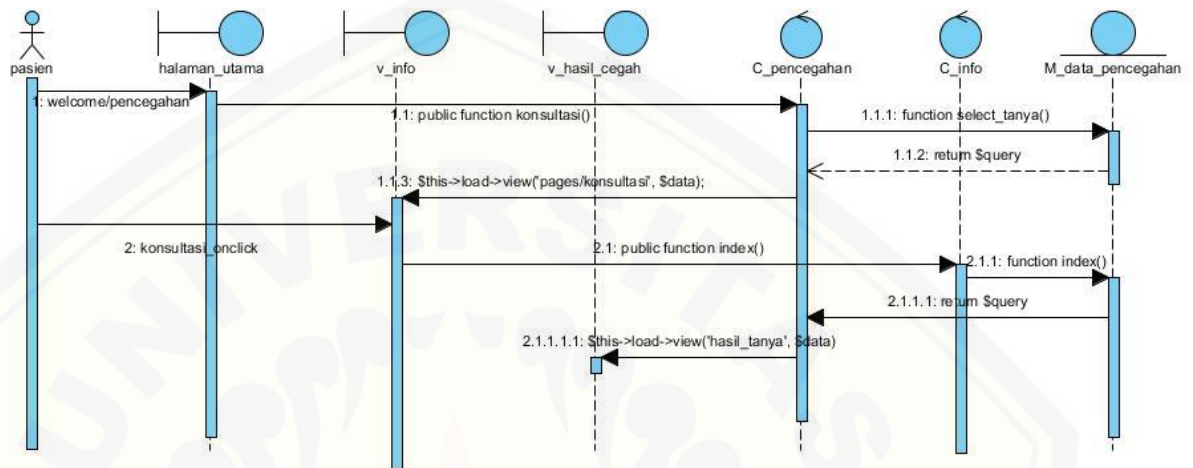


Gambar 4. 11 Activity Diagram Diagnosa Asma Pasien

#### 4.2.5.3 Sequence Diagram diagnosa pencegahan asma pasien

Menu diagnosa pencegahan asma pasien menampilkan beberapa pertanyaan untuk pasien. Pada menu ini pasien berhak memilih satu pertanyaan atau lebih. Nantinya setelah dipilih pertanyaan – pertanyaan yang ada maka akan dikembalikan jawaban – jawaban berdasarkan *query* pertanyaan yang dipilih oleh *user*. Penggambaran *sequence diagram* diagnosa pencegahan asma pasien menjelaskan fungsi atau method apa saja

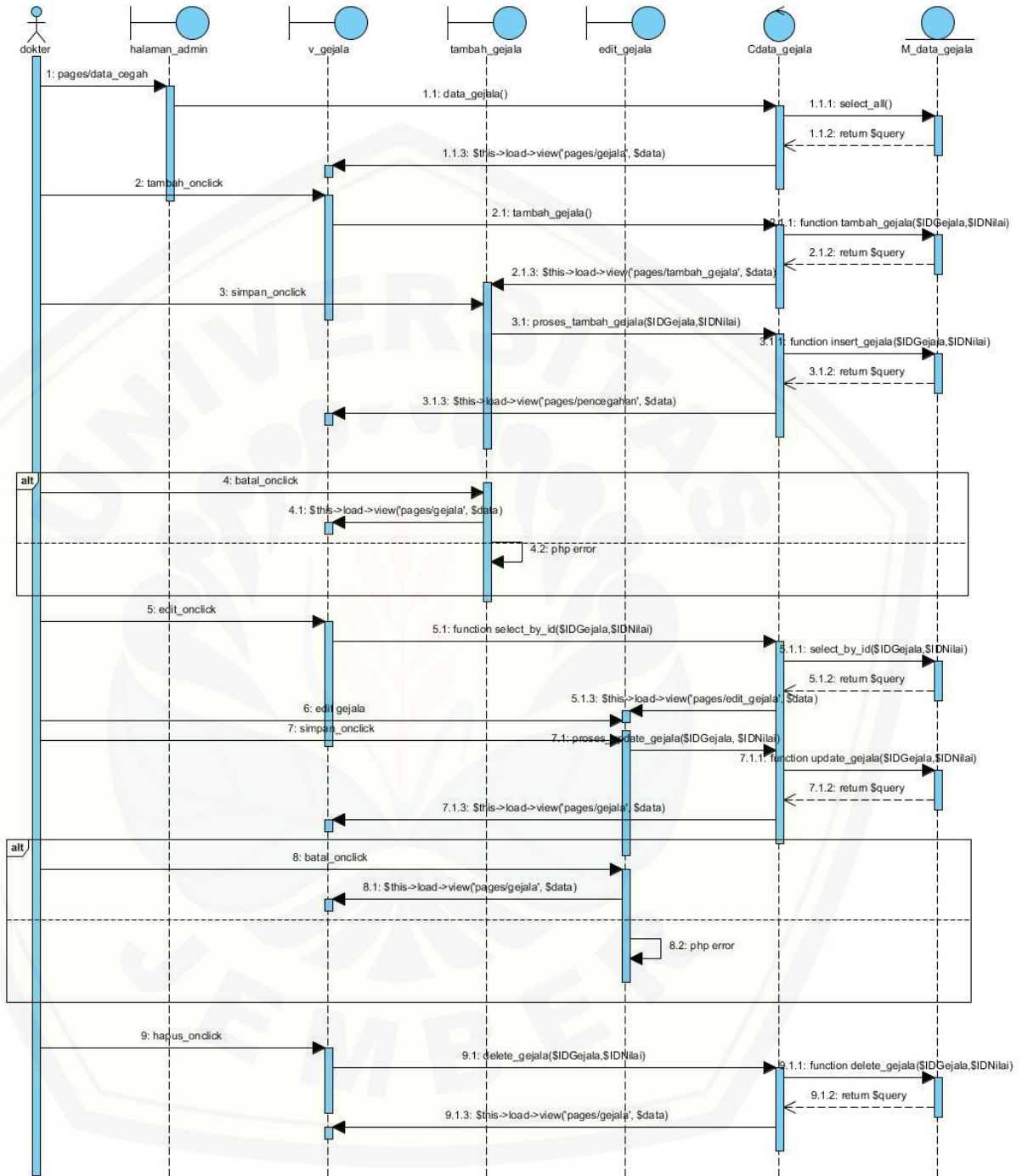
yang akan dibuat. Pada gambar dibawah dapat dilihat alur perjalanan suatu fitur antara aksi yang dilakukan oleh aktor, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut.



Gambar 4. 12 *Sequence Diagram* Diagnosa Pencegahan Asma Pasien

#### 4.2.5.4 *Sequence Diagram* mengelola data gejala asma

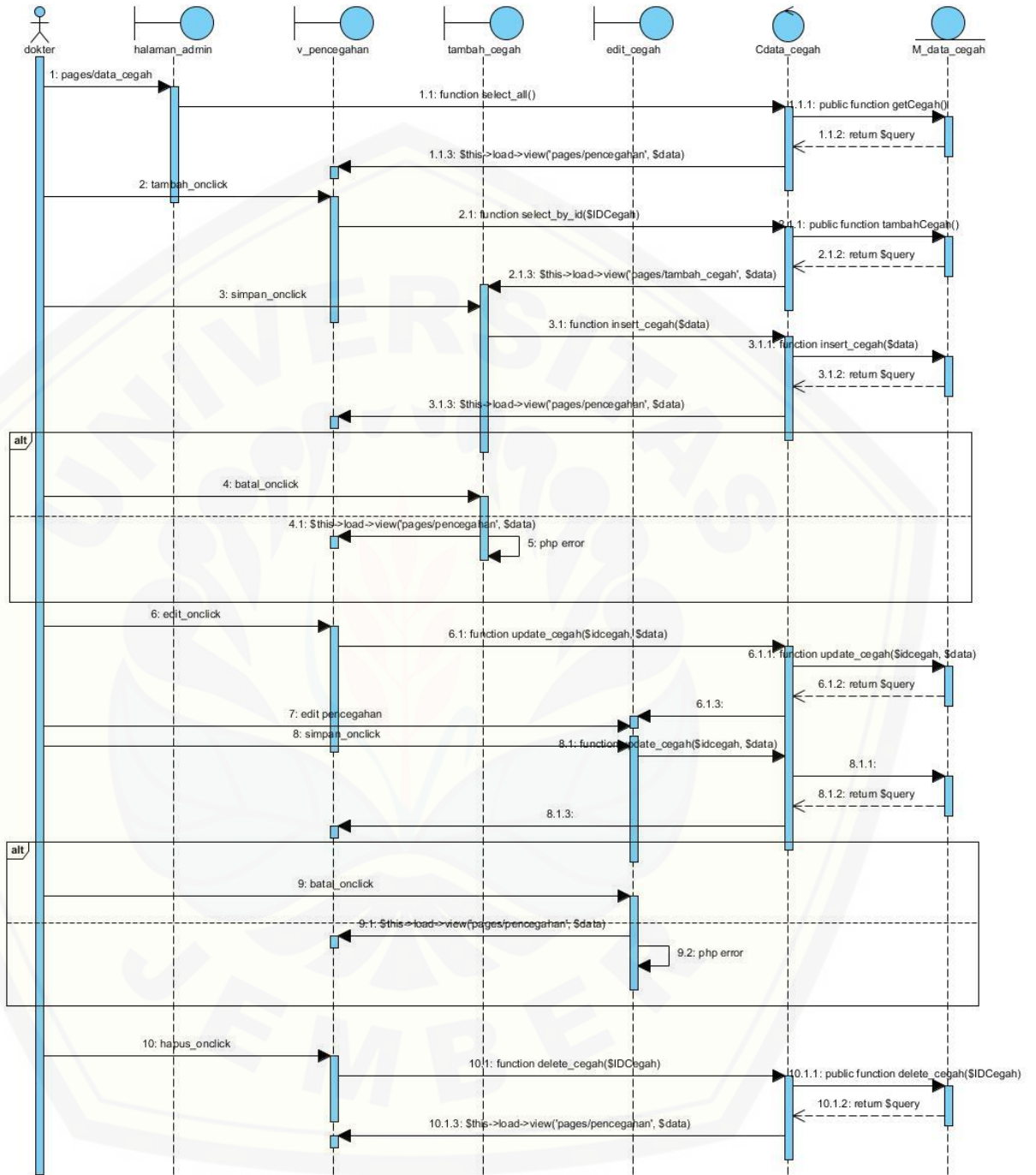
Mengelola data gejala asma yang dilakukan oleh pakar, sebelumnya pakar harus melakukan login terlebih dahulu. Setelah itu pakar baru bisa melakukan *input*, *edit*, *delete* data yang diinginkan. Setelah data dilakukan perubahan berupa input atau *edit* data baru tersebut kembali disimpan dalam *database*. Begitu pun ketika data dihapus, data akan dilakukan *update* sehingga data yang terpilih tadi sudah tidak ada di dalam *database*. Penggambaran *sequence diagram* mengelola data gejala asma seperti pada gambar dibawah menjelaskan fungsi atau method apa saja yang akan dibuat. Pada gambar tersebut dapat dilihat alur perjalanan suatu fitur antara aksi yang dilakukan oleh aktor antara lain seperti tambah data, edit data, dan hapus data kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut.



Gambar 4. 13 Sequence Diagram Mengelola Data Gejala Asma

#### 4.2.5.5 *Sequence Diagram* mengelola data pencegahan asma

Mengelola data pencegahan asma yang dilakukan oleh pakar, sebelumnya pakar harus melakukan login terlebih dahulu. Setelah itu pakar baru bisa melakukan *input*, *edit*, *delete* data yang diinginkan. Setelah data dilakukan perubahan berupa *input* atau *edit* data baru tersebut kembali disimpan dalam database. Begitu pun ketika data dihapus, data akan dilakukan *update* sehingga data yang terpilih tadi sudah tidak ada di dalam *database*. Penggambaran *sequence diagram* mengelola data pencegahan asma menjelaskan fungsi atau method apa saja yang akan dibuat. Pada gambar dibawah dapat dilihat alur perjalanan suatu fitur antara aksi yang dilakukan oleh aktor antara lain seperti tambah data, edit data, dan hapus data kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut.

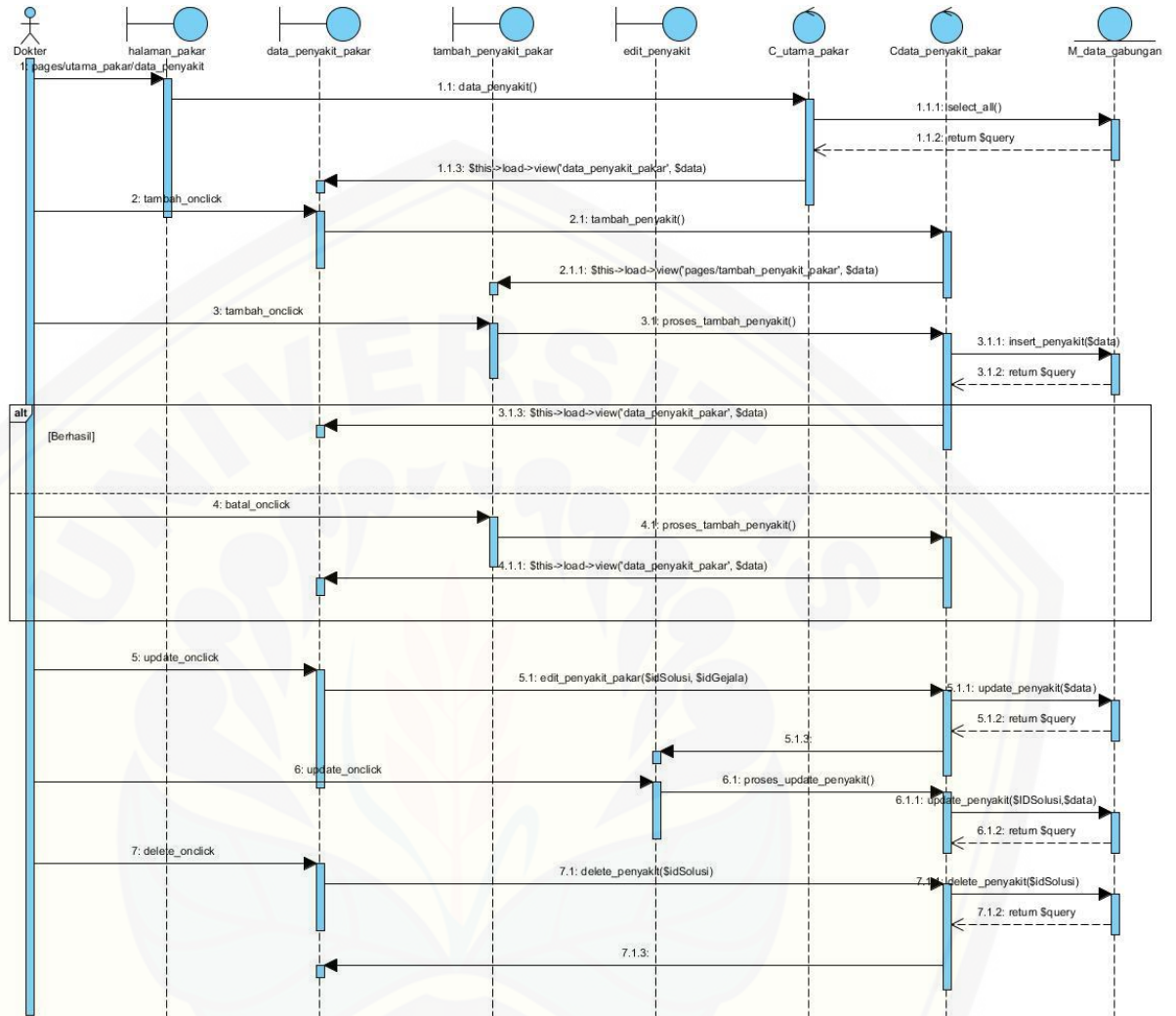


Gambar 4. 14 Sequence Diagram Mengelola Data Pencegahan Asma



#### 4.2.5.6 *Sequence Diagram* mengelola data tingkatan asma

Mengelola data tingkatan asma yang dilakukan oleh pakar, sebelumnya pakar harus melakukan login terlebih dahulu. Setelah itu pakar baru bisa melakukan *input*, *edit*, *delete* data yang diinginkan. Setelah data dilakukan perubahan berupa input atau *edit* data baru tersebut kembali disimpan dalam database. Begitu pun ketika data dihapus, data akan dilakukan *update* sehingga data yang terpilih tadi sudah tidak ada di dalam *database*. Penggambaran *sequence diagram* mengelola data tingkatan asma menjelaskan fungsi atau method apa saja yang akan dibuat. Pada gambar dibawah dapat dilihat alur perjalanan suatu fitur antara aksi yang dilakukan oleh aktor antara lain seperti tambah data, edit data, dan hapus data kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut.

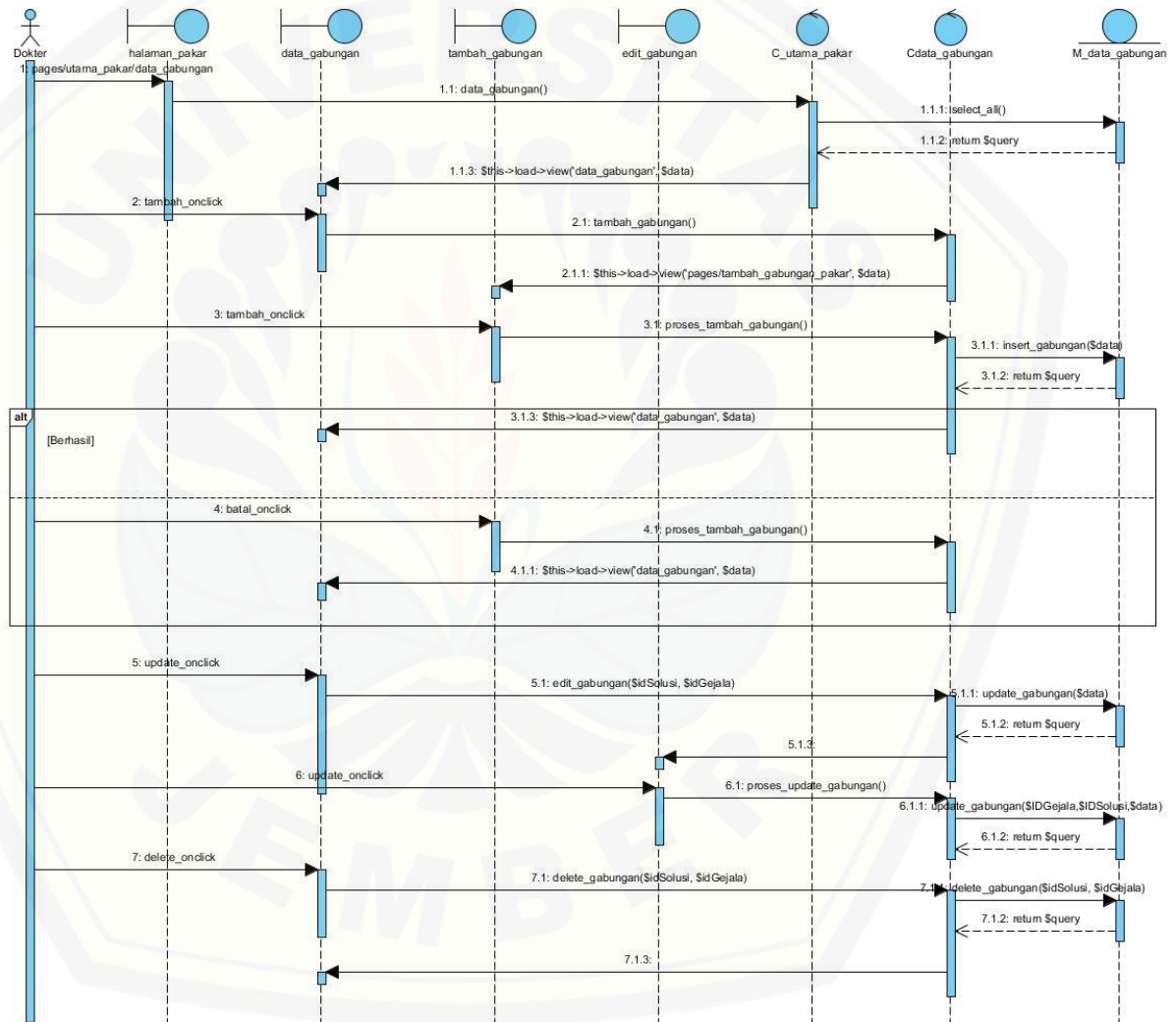


Gambar 4. 15 Sequence Diagram Mengelola Data Tingkatan Asma

4.2.5.7 Sequence Diagram mengelola data nilai CF

Mengelola data nilai CF yang dilakukan oleh pakar, sebelumnya pakar harus melakukan login terlebih dahulu. Setelah itu pakar baru bisa melakukan *input*, *edit*, *delete* data yang diinginkan. Data yang ditambahkan, diedit dan dihapus berhubungan dengan data gejala dan data tingkatan asma pada menu sebelumnya. Setelah data dilakukan perubahan berupa input atau *edit* data baru tersebut kembali disimpan dalam database. Begitu pun ketika data dihapus, data akan dilakukan *update* sehingga data

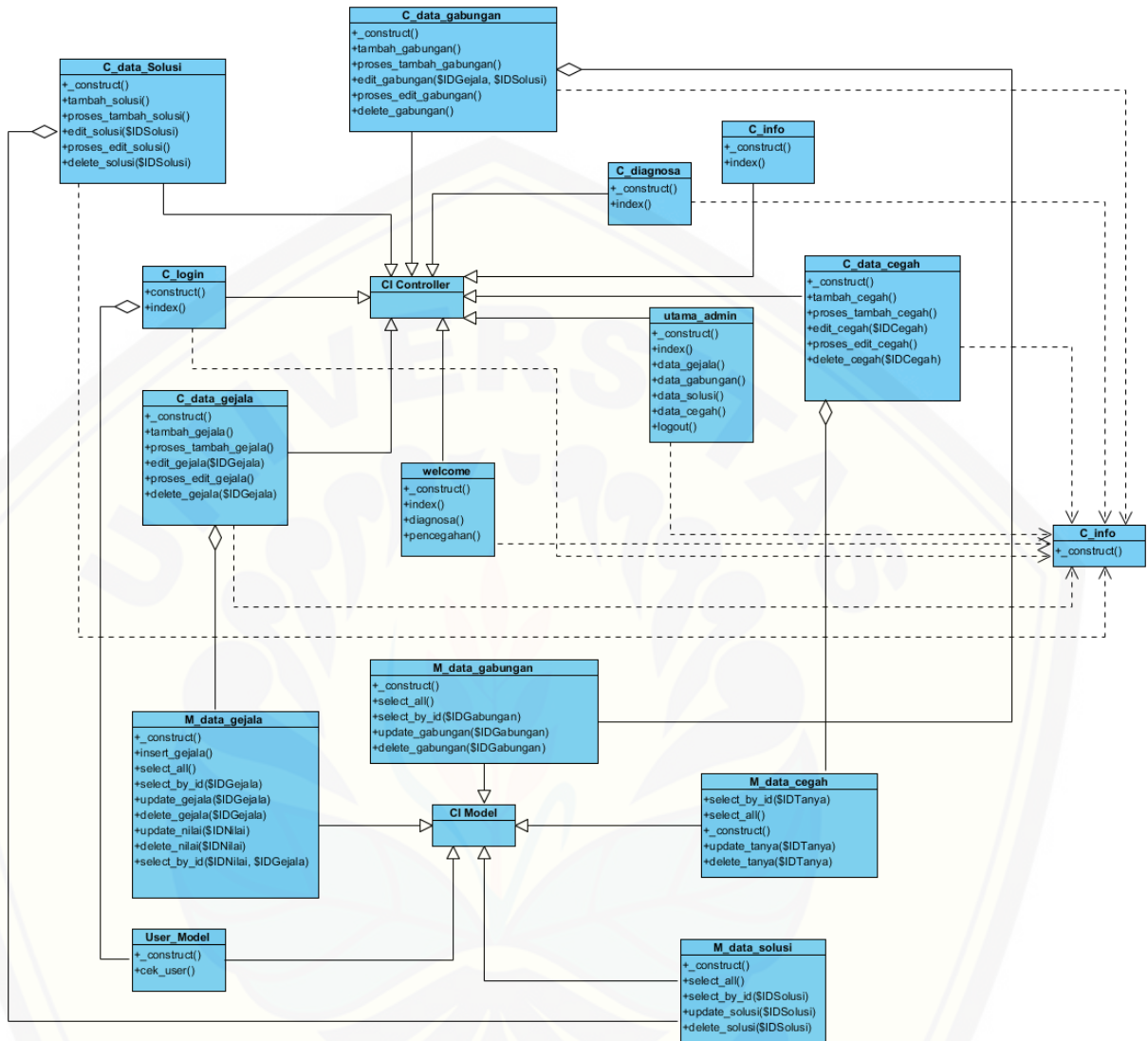
yang terpilih tadi sudah tidak ada di dalam *database*. Penggambaran *sequence diagram* mengelola data tingkatan asma menjelaskan fungsi atau method apa saja yang akan dibuat. Pada gambar dibawah dapat dilihat alur perjalanan suatu fitur antara aksi yang dilakukan oleh aktor antara lain seperti tambah data, edit data, dan hapus data kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut.



Gambar 4. 16 *Sequence Diagram* Mengelola Data Nilai CF

#### 4.2.6 Class Diagram

*Class diagram* merupakan suatu diagram yang menggambarkan struktur dan penjelasan *class*, *package*, dan objek serta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain. Pada *class diagram* sistem ini dibagi menjadi tiga yaitu *model*, *view*, dan *controller*. *Model* menggambarkan struktur *class* yang menghubungkan sistem dengan *database*, *view* menggambarkan struktur *class* yang menangani tampilan dari sistem, sedangkan *controller* menggambarkan struktur *class* dari logika sistem dan menghubungkan antara *model* dan *view*. Pada sistem prediksi penyakit asma dan penanganannya, class diagram terdiri atas 3 class utama yaitu CI\_controller, CI Model dan loader\_view. Pada class utama tersebut memiliki fungsi masing – masing yaitu pada CI\_controller menghubungkan *controller* satu dengan lainnya. *Controller* pada sistem prediksi penyakit asma dan penanganannya terdiri dari c\_login, c\_data\_gabungan, dan lain lain. Sedangkan pada *class* utama model berfungsi untuk menghubungkan model. Macam – macam model yang ada pada sistem prediksi penyakit asma dan penanganannya antara lain M\_data\_gabungan, M\_data\_cegah, dan lain lain. Gambaran *class* diagram pada penelitian ini dapat dilihat pada gambar berikut.

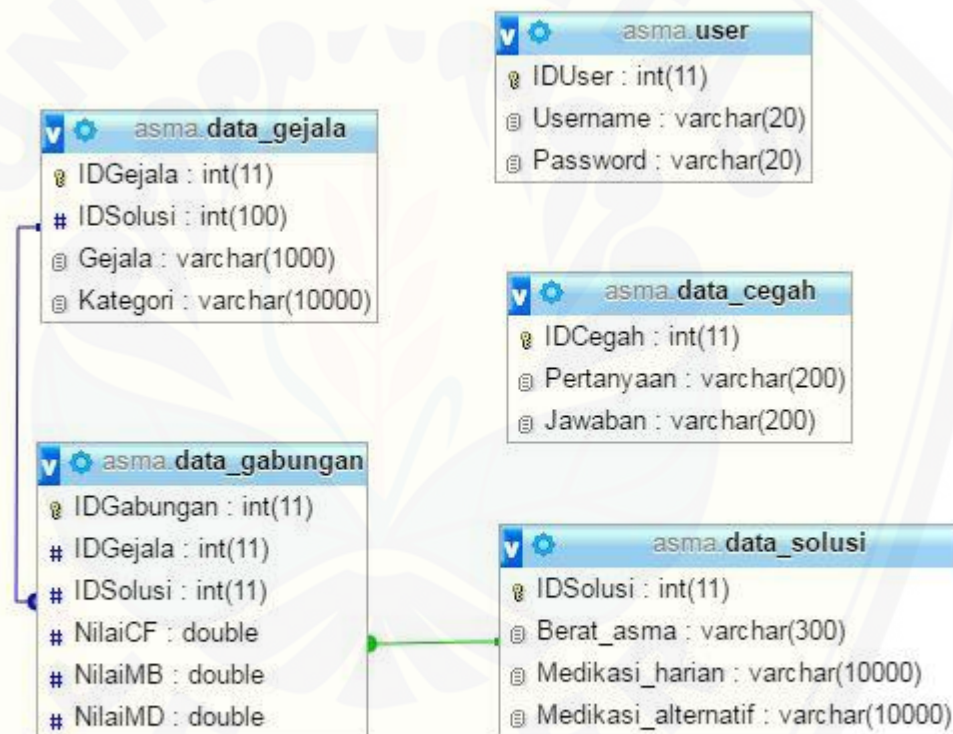


Gambar 4. 17 Class Diagram

#### 4.2.7 Entity Relationship Diagram

Entity relationship diagram merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Entity relationship diagram digunakan untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol. Pada sistem prediksi penyakit asma dan penanganannya terdapat 5 tabel yaitu tabel user, tabel data\_cegah, tabel data\_gejala,

tabel data\_solusi, dan tabel data\_gabungan. Pada *database* sistem prediksi penyakit asma dan penanganannya terdapat 3 database yang berelasi yaitu tabel data\_gejala, tabel data\_gabungan, dan tabel data\_solusi. Tabel data\_gejala saling berelasi dengan *foreign key* IDGejala dan IDGejala data\_gabungan sebagai *primary key*, dan tabel data\_solusi, IDSolusi sebagai *primary key* berelasi dengan IDSolusi pada tabel data\_gabungan sebagai *primary key*. *Class diagram* pada penelitian ini dapat dilihat pada gambar berikut.



Gambar 4. 18 Entity Relationship Diagram

### 4.3 Penulisan Kode Program

Tahap penulisan kode program merupakan tahap lanjutan dari desain sistem. Desain sistem dari semua fitur yang telah dibuat menggunakan pemodelan UML akan

diimplementasikan ke dalam kode program. Penulisan kode program menggunakan bahasa pemrograman *Page Hypertext Pre-Processor* (PHP) dengan *framework Code Igniter* (CI) untuk website dan menggunakan *javascript*. Pemograman dilakukan menggunakan *Sublime Text* dan manajemen basis data yang digunakan adalah MySQL.

Implementasi perhitungan dari metode *certainty factor* pada sistem pakar prediksi tingkatan asma pasien dan penanganannya terdapat pada *view* hasil\_konsultasi. Gambaran baris kode program perhitungan metode *certainty factor* adalah sebagai berikut :

#### 4.3.1 Kode Program Controller C\_Diagnosa

Kode program pada *controller* C\_diagnosa merupakan *controller* yang didalamnya terdapat kode program berupa perhitungan data gabungan yang mengambil data dari nilai CF setiap gejala terpilih dan IDPenyakit berdasarkan tabel di database yang digunakan untuk perhitungan penentuan kategori asma pasien.

```

1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class C_diagnosa extends CI_Controller {
4
5     public function __construct(){
6         parent::__construct();
7         $this->load->helper('url');
8         $this->load->library('input');
9         $this->load->model('M_data_gejala');
10        $this->load->model('M_data_penyakit');
11        $this->load->model('M_data_gabungan');
12    }
13
14    public function index()
15    {
16        $IDPenyakit = array();
17        $persentase = array();
18        $cek1 = $this->input->post('cek1');
19        $jumlah1 = count($cek1);
20        $cek2 = $this->input->post('cek2');
21        $jumlah2 = count($cek2);
22        $cek3 = $this->input->post('cek3');
23        $jumlah3 = count($cek3);
24        for($i=0; $i<$jumlah1; $i++){
25            $data["daftar_gabungan1"] = $this->M_data_gabungan->select_id_gejala($cek1[$i])->result();
26            foreach ($data["daftar_gabungan1"] as $gabungan1) {
27                array_push($IDPenyakit, $gabungan1->NilaiCF);
28            }
29        }
30        for($i=0; $i<$jumlah2; $i++){
31            $data["daftar_gabungan2"] = $this->M_data_gabungan->select_id_gejala($cek2[$i])->result();
32            foreach ($data["daftar_gabungan2"] as $gabungan2) {
33                array_push($IDPenyakit, $gabungan2->NilaiCF);
34            }
35        }
36        for($i=0; $i<$jumlah3; $i++){
37            $data["daftar_gabungan3"] = $this->M_data_gabungan->select_id_gejala($cek3[$i])->result();
38            foreach ($data["daftar_gabungan3"] as $gabungan3) {
39                array_push($IDPenyakit, $gabungan3->NilaiCF);
40            }
41        }
42        $hasil = ($IDPenyakit[0] + ($IDPenyakit[1] * (1 - $IDPenyakit[0])));
43        $hasil2 = $hasil + ($IDPenyakit[2] * (1 - $hasil));
44        if ( $hasil2 >= 0.90 && $hasil2 <= 1.00) {
45            $data7['daftar_penyakit1'] = $this->M_data_gabungan->select_solusi("3")->result();
46            $data7['hasil'] = $hasil2;
47        }
48        else if( $hasil2 >= 0.80 && $hasil2 <= 0.89) {
49            $data7['daftar_penyakit1'] = $this->M_data_gabungan->select_solusi("2")->result();
50            $data7['hasil'] = $hasil2;
51        }
52        else if( $hasil2 >= 0 && $hasil2 <= 0.79) {
53            $data7['daftar_penyakit1'] = $this->M_data_gabungan->select_solusi("1")->result();
54            $data7['hasil'] = $hasil2;
55        }
56        $data=array(
57            'title'=>'diagnosa',
58            'active_diagnosa'=>'active');
59        $this->load->view('element/header',$data);
60        $this->load->view('element/navbar_user');
61        $this->load->view('pages/hasil_konsultasi', $data7);
62        $this->load->view('element/footer');
63    }
64 }
65
66 /* End of file welcome.php */
67 /* Location: ./application/controllers/welcome.php */

```

Gambar 4. 19 Kode Program *Controller* Diagnosa



#### 4.3.2 Kode Program *Controller* Cdata\_Gabungan\_Pakar

Pada kode program controller data gabungan pakar terdiri atas beberapa *function* yang berbeda tugas dan fungsinya. Diantaranya adalah :

##### 4.3.2.1 Tambah gabungan

Baris Kode program pada *function* ini berfungsi untuk melakukan tambahan data gabungan yang dimasukkan oleh pakar.

```
15 public function tambah_gabungan(){
16     if ($this->session->userdata('IDUser')) {
17         $data=array(
18             'title'=>'Data Gabungan',
19             'active_about'=>'active');
20
21         $this->load->view('element/header',$data);
22         $this->load->view('element/navbar_pakar');
23
24         $data['jangka'] = $this->M_data_gejala->select_jangka()->result();
25         $data['faal'] = $this->M_data_gejala->select_faal()->result();
26         $data['malam'] = $this->M_data_gejala->select_malam()->result();
27         $data['daftar_penyakit'] = $this->M_data_penyakit->select_all()->result();
28         $this->load->view('pages/tambah_gabungan_pakar', $data);
29         $this->load->view('element/footer');
30     }
31     else{
32         redirect(site_url('welcome/'));
33     }
34 }
```

Gambar 4. 20 *Controller* Tambah Data Gabungan

##### 4.3.2.2 Edit data gabungan

Baris kode program pada *function* ini berfungsi untuk melakukan edit data gabungan yang dimasukkan oleh pakar.

```

80 public function edit_gabungan($id_penyakit, $id_gejala){
81     if ($this->session->userdata('IDUser')) {
82         $data=array(
83             'title'=>'Data Gabungan',
84             'active_about'=>'active');
85
86         $this->load->view('element/header',$data);
87         $this->load->view('element/navbar_pakar');
88
89         $data['IDSolusi']         = $id_penyakit;
90         $data['IDGejala']        = $id_gejala;
91         $data['cf']              = $this->M_data_gabungan->select_by_id($id_penyakit, $id_gejala)->row();
92         $this->load->view('pages/edit_cf', $data);
93         $this->load->view('element/footer');
94     }
95     else{
96         redirect(site_url('welcome/'));
97     }
98 }

```

Gambar 4. 21 Edit Data Gabungan

#### 4.3.2.3 Hapus data gabungan

Pada kode program pada *function* ini berfungsi untuk menghapus data gabungan yang dipilih oleh pakar.

```

112 public function delete_gabungan($id_penyakit, $id_gejala){
113     $this->M_data_gabungan->delete_gabungan($id_penyakit, $id_gejala);
114     redirect(site_url('utama_pakar/data_gabungan'));
115 }
116 }

```

Gambar 4. 22 Hapus Data Gabungan

#### 4.3.3 Kode Program Controller C\_info

*Listing screenshot* kode program controller info telah dilampirkan pada lampiran kode program C\_info.

#### 4.3.4 Kode Program Controller login

*Listing screenshot* kode program controller info telah dilampirkan pada lampiran kode program login.

#### 4.3.5 Kode Program Controller Cdata\_penyakit\_pakar

*Listing screenshot* kode program controller info telah dilampirkan pada lampiran kode program Cdata\_Penyakit Pakar.

#### 4.3.6 Kode Program Controller Cdata\_pencegahan

#### 4.3.7 Kode Program Controller Cdata\_Ge

### 4.4 Pengujian sistem

Tahap pengujian dilakukan untuk mengevaluasi kinerja dari sistem yang telah dibuat. Pengujian yang dilakukan dengan cara *White box testing* dan *Black box testing*. Pengujian *white box testing* adalah pengujian dengan metode *test case* yang digambarkan berdasarkan notasi diagram alir. Pengujian *black box testing* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika *internal* perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *Black box testing* merupakan metode perancangan data uji yang didasarkan pada spesifikasi sistem pakar prediksi tingkatan asma pasien dan penanganannya.

#### 4.4.1 White Box Testing

White Box Testing merupakan pengujian pada modul pengkodean program untuk menjamin kode program bebas dari kesalahan sintaks maupun logika. Dalam pengujian white box terdapat beberapa tahapan pembuatan dokumentasi pengujian yaitu *cyclomatic complexity (CC)*, *listing program*, penentuan jalur *independen*, dan *test case*. Gambaran detail mengenai tahapan pengujian dengan metode *white box testing* dapat dilihat pada penjabaran berikut.

## 1. Pengujian *White Box Testing* Mengelola Data Gejala Asma

Pengujian *White Box Testing* manajemen data gejala meliputi fitur tambah data gejala asma, *edit* data gejala asma, dan hapus data gejala asma. Pengujian *White Box Testing* mengelola data gejala asma adalah sebagai berikut.

### a. Listing program fitur tambah data gejala asma

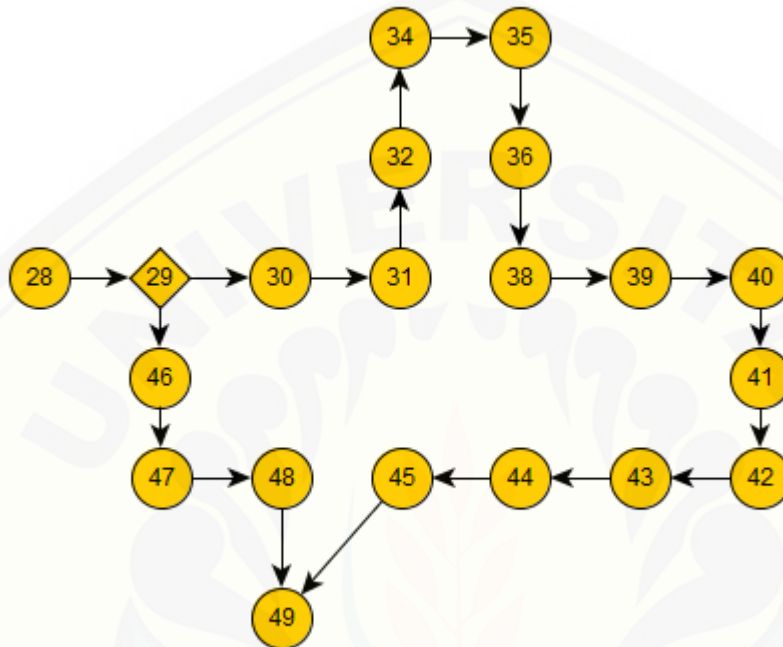
```
28 public function tambah_gejala(){
29     if ($this->session->userdata('IDUser')) {
30         $data2=array(
31             'title'=>'Data Gejala',
32             'active_konsultasi'=>'active');
33
34         $data = array(
35             'id_otomatis'=> $this->M_data_gejala->id_otomatis(),
36             );
37
38         $this->load->view('element/header',$data2);
39         $this->load->view('element/navbar_pakar');
40         $data['jangka'] = $this->M_data_gejala->select_jangka()->result();
41         $data['faal'] = $this->M_data_gejala->select_faal()->result();
42         $data['malam'] = $this->M_data_gejala->select_malam()->result();
43         $this->load->view('pages/tambah_gejala_pakar',$data);
44         $this->load->view('element/footer');
45     }
46     else{
47         redirect(site_url('welcome/'));
48     }
49 }
50
```

Gambar 4. 23 *White Box Testing* Mengelola Data Gejala Asma

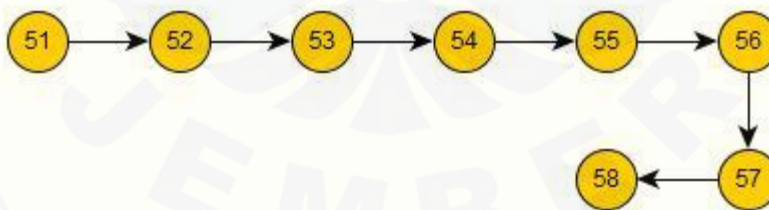
```
51 public function proses_tambah_gejala(){
52     $data['IDGejala'] = $this->input->post('IDGejala');
53     $data['Gejala'] = $this->input->post('Gejala');
54     $data['Kategori'] = $this->input->post('Kategori');
55     |
56
57     $this->M_data_gejala->insert_gejala($data);
58     redirect(site_url('utama_pakar/data_gejala'));
59 }
60
```

Gambar 4. 24 *White Box Testing* Proses Tambah Gejala

b. Diagram alir fitur tambah data gejala



Gambar 4. 25 Diagram Alir Tambah Data Gejala



Gambar 4. 26 Diagram Alir Proses Tambah Gejala

c. Perhitungan *Cyclomatic Complexity* fitur tambah gejala asma

Perhitungan diagram alir pada fitur tambah data gejala asma menggunakan *Cyclomatic Complexity* adalah sebagai berikut:

*function* tambah\_gejala() :  $V(G) = E - N + 2 = 20 - 20 + 2 = 2$

*function* proses\_tambah\_gejala() :  $V(G) = E - N + 2 = 7 - 8 + 2 = 1$

d. *Test case* fitur tambah gejala asma

Tabel 4. 10 *Test Case* Fitur Tambah Gejala Asma

<i>Test Case function</i> tambah_gejala()	
<i>Test case</i>	Jika form tambah gejala mengload data id otomatis
Target yang diharapkan	Menampilkan form tambah data dengan id otomatis
Hasil pengujian	Benar
<i>Path/Jalur</i>	28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-49
<i>Test case function</i> tambah_gejala()	
<i>Test case</i>	Jika form tambah gejala mengload data id otomatis
Target yang diharapkan	Menampilkan form tambah data dengan id otomatis
Hasil pengujian	Benar
<i>Path/Jalur</i>	28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-49

. 2. Pengujian *White Box Testing* Perhitungan Metode *Certainty Factor*

Pengujian *White Box Testing* perhitungan metode *certainty factor* meliputi fitur diagnosa asma pasien. Pengujian *White Box Testing* perhitungan metode *certainty factor* adalah sebagai berikut.

a. *Listing* program perhitungan metode *certainty factor*

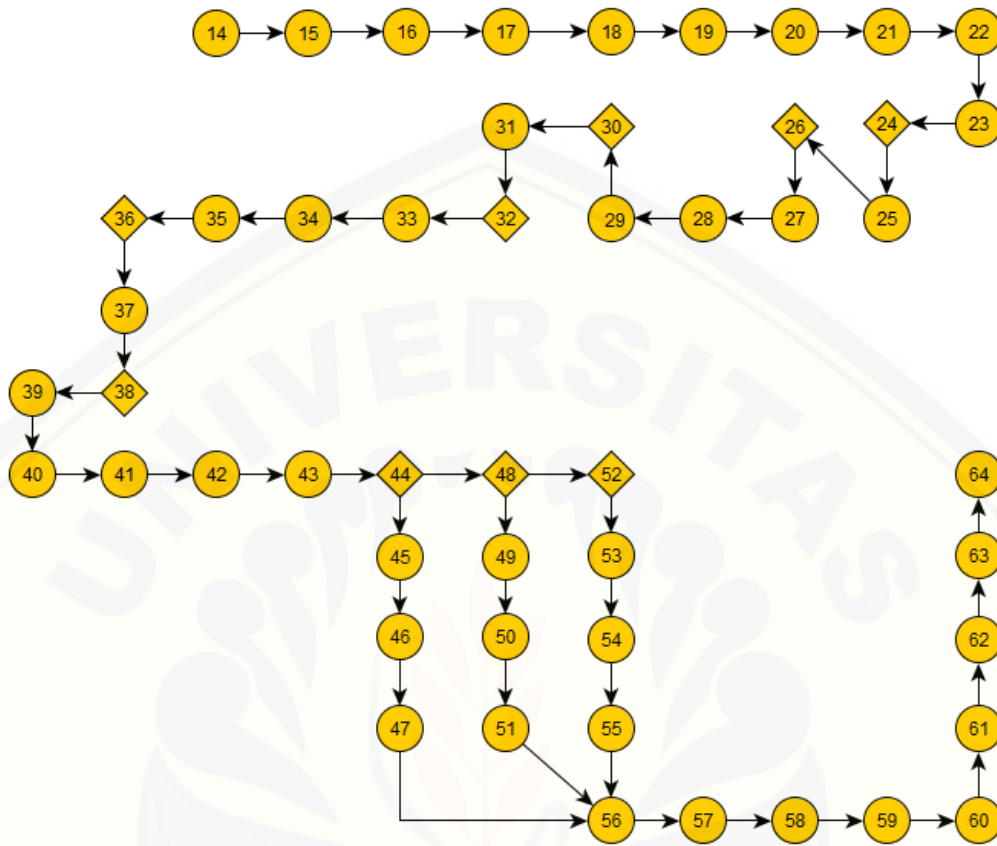
```

14 public function index()
15 {
16     $IDPenyakit = array();
17     $persentase = array();
18     $cek1 = $this->input->post('cek1');
19     $jumlah1 = count($cek1);
20     $cek2 = $this->input->post('cek2');
21     $jumlah2 = count($cek2);
22     $cek3 = $this->input->post('cek3');
23     $jumlah3 = count($cek3);
24     for($i=0; $i<$jumlah1; $i++){
25         $data["daftar_gabungan1"] = $this->M_data_gabungan->select_id_gejala($cek1[$i])->result();
26         foreach ($data["daftar_gabungan1"] as $gabungan1) {
27             array_push($IDPenyakit, $gabungan1->NilaiCF);
28         }
29     }
30     for($i=0; $i<$jumlah2; $i++){
31         $data["daftar_gabungan2"] = $this->M_data_gabungan->select_id_gejala($cek2[$i])->result();
32         foreach ($data["daftar_gabungan2"] as $gabungan2) {
33             array_push($IDPenyakit, $gabungan2->NilaiCF);
34         }
35     }
36     for($i=0; $i<$jumlah3; $i++){
37         $data["daftar_gabungan3"] = $this->M_data_gabungan->select_id_gejala($cek3[$i])->result();
38         foreach ($data["daftar_gabungan3"] as $gabungan3) {
39             array_push($IDPenyakit, $gabungan3->NilaiCF);
40         }
41     }
42     $hasil = ($IDPenyakit[0] + ($IDPenyakit[1] * (1 - $IDPenyakit[0])));
43     $hasil2 = $hasil + ($IDPenyakit[2] * (1 - $hasil));
44     if ( $hasil2 >= 0.90 && $hasil2 <= 1.00) {
45         $data7['daftar_penyakit1'] = $this->M_data_gabungan->select_solusi("3")->result();
46         $data7['hasil'] = $hasil2;
47     }
48     else if( $hasil2 >= 0.80 && $hasil2 <= 0.89) {
49         $data7['daftar_penyakit1'] = $this->M_data_gabungan->select_solusi("2")->result();
50         $data7['hasil'] = $hasil2;
51     }
52     else if( $hasil2 >= 0 && $hasil2 <= 0.79) {
53         $data7['daftar_penyakit1'] = $this->M_data_gabungan->select_solusi("1")->result();
54         $data7['hasil'] = $hasil2;
55     }
56     $data=array(
57         'title'=>'diagnosa',
58         'active_diagnosa'=>'active');
59     $this->load->view('element/header',$data);
60     $this->load->view('element/navbar_user');
61     $this->load->view('pages/hasil_konsultasi', $data7);
62     $this->load->view('element/footer');
63 }
64 }

```

Gambar 4. 27 Listing Kode Program Metode *Certainty Factor*

- b. Diagram alir perhitungan metode *certainty factor*



Gambar 4. 28 Diagram Alir Perhitungan Metode *Certainty Factor*

c. Perhitungan *Cyclomatic Complexity* perhitungan metode *certainty factor*

Perhitungan diagram alir pada perhitungan metode *certainty factor* menggunakan *Cyclomatic Complexity* adalah sebagai berikut:

*function index()* :  $V(G) = E - N + 2 = 52 - 51 + 2 = 3$

d. *Test case* perhitungan metode *certainty factor*

Tabel 4. 11 *Test Case* Perhitungan Metode *Certainty Factor*

<i>Test Case function index()</i>	
<i>Test case</i>	Jika ada data yang di centang oleh user



Target yang diharapkan	Menampilkan hasil perhitungan dari rumus yang tersedia
Hasil pengujian	Benar
<i>Path/Jalur</i>	14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-56-57-58-59-60-61-62-63-64
<i>Test case function index()</i>	
<i>Test case</i>	Jika ada data yang di centang oleh user
Target yang diharapkan	Menampilkan hasil perhitungan dari rumus yang tersedia
Hasil pengujian	Benar
<i>Path/Jalur</i>	14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-48-49-50-51-56-57-58-59-60-61-62-63-64
<i>Test case function index()</i>	
<i>Test case</i>	Jika ada data yang di centang oleh user
Target yang diharapkan	Menampilkan hasil perhitungan dari rumus yang tersedia
Hasil pengujian	Benar
<i>Path/Jalur</i>	14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-48-52-53-54-55-56-57-58-59-60-61-62-63-64

#### 4.4.2 Black box Testing

*Black Box Testing* merupakan pengujian yang menekankan pada pengujian fungsionalitas sistem agar keluaran sesuai dengan apa yang diharapkan pengguna. Dokumentasi hasil pengujian dengan metode *black box testing* dapat dilihat pada tabel – tabel berikut.

Tabel 4. 12 *Black Box Login*

No	Fitur	Kasus	Hasil	Keterangan
1.	Login (Pakar)	<ul style="list-style-type: none"> <li>Ketika klik tombol <i>login</i> dan semua <i>field</i> pada form login sudah terisi dengan benar</li> <li>Ketika klik tombol login dan belum mengisi form secara lengkap / <i>field</i> masih ada yang kosong</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan halaman awal admin</li> <li>Menampilkan error message username atau password salah</li> </ul>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal  <input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

Tabel diatas merupakan hasil pengujian *black box* pada fitur hak akses login. Hasil pengujian menunjukkan bahwa implementasi sistem pada fitur login sesuai dengan harapan admin dan pakar.

Tabel 4. 13 Tabel Pengujian *Black Box* Mengelola Data Gejala

No	Fitur	Kasus	Hasil	Keterangan
1.	Tambah data gejala	<ul style="list-style-type: none"> <li>Ketika klik tombol tambah dan semua <i>field</i></li> </ul>	<ul style="list-style-type: none"> <li>Menambah data gejala sesuai yang</li> </ul>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

		<p>pada form tambah gejala sudah terisi dengan benar</p> <ul style="list-style-type: none"> <li>• Ketika klik tombol simpan dan belum mengisi form tambah gejala secara lengkap / <i>field</i> masih ada yang kosong</li> </ul>	<p>diinputkan oleh pakar</p> <ul style="list-style-type: none"> <li>• Menampilkan pesan data tidak boleh kosong</li> </ul>	<p>[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal</p>
2.	Edit data gejala	<ul style="list-style-type: none"> <li>• Ketika klik tombol <i>update</i> dan semua <i>field</i> pada form edit data gejala sudah terisi dengan benar</li> <li>• Ketika klik tombol <i>update</i> dan belum mengisi secara lengkap / <i>field</i> masih ada yang kosong</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Update</i> data gejala sesuai yang diinputkan oleh pakar</li> <li>• Menampilkan pesan data tidak boleh kosong</li> </ul>	<p>[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal</p> <p>[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal</p>
3.	Hapus data gejala	<ul style="list-style-type: none"> <li>• Ketika klik tombol delete data yang akan dihapus dan klik tombol delete pada pop – up</li> <li>• Ketika klik tombol delete dan memilih cancel</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Delete</i> data gejala sesuai yang dipilih oleh pakar</li> <li>• Menampilkan halaman sebelumnya</li> </ul>	<p>[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal</p> <p>[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal</p>

Tabel diatas merupakan hasil pengujian *black box* pada fitur mengelola data gejala. Hasil pengujian menunjukkan impementasi sistem pada menejemen data gejala sesuai dengan harapan pakar.

Tabel 4. 14 Pengujian *Black Box* Manajemen Data Tingkatan Asma

No	Fitur	Kasus	Hasil	Keterangan
----	-------	-------	-------	------------

1.	Tambah data tingkatan asma pasien	<ul style="list-style-type: none"> <li>• Ketika klik tombol tambah dan semua <i>field</i> pada form tambah data tingkatan asma pasien sudah terisi dengan benar</li> <li>• Ketika klik tombol simpan dan belum mengisi form tambah tingkatan asma pasien secara lengkap / <i>field</i> masih ada yang kosong</li> </ul>	<ul style="list-style-type: none"> <li>• Menambah tingkatan asma pasien sesuai yang diinputkan oleh pakar</li> <li>• Menampilkan pesan data tidak boleh kosong</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal  <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
2.	Edit data tingkatan asma pasien	<ul style="list-style-type: none"> <li>• Ketika klik tombol <i>update</i> dan semua <i>field</i> pada form edit data tingkatan asma sudah terisi dengan benar</li> <li>• Ketika klik tombol <i>update</i> dan belum mengisi form edit data tingkatan asma pasien secara lengkap / <i>field</i> masih ada yang kosong</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Update</i> data tingkatan asma sesuai yang diinputkan oleh pakar</li> <li>• Menampilkan pesan data tidak boleh kosong</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal  <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
3.	Hapus data tingkatan asma	<ul style="list-style-type: none"> <li>• Ketika klik tombol delete data yang akan dihapus dan klik tombol delete pada pop – up</li> <li>• Ketika klik tombol delete dan memilih cancel</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Delete</i> data tingkatan asma sesuai yang dipilih pakar</li> <li>• Menampilkan halaman sebelumnya</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal  <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

Tabel diatas merupakan hasil pengujian *black box* pada fitur mengelola data tingkatan asma pasien. Hasil pengujian menunjukkan impementasi sistem pada menejemen data tingkatan asma pasien sesuai dengan harapan pakar.

Tabel 4. 15 Pengujian *Black Box* Mengelola Data Pencegahan Asma

No	Fitur	Kasus	Hasil	Keterangan
1.	Tambah data pencegahan asma pasien	<ul style="list-style-type: none"> <li>• Ketika klik tombol tambah dan semua <i>field</i> pada form tambah data pencegahan asma pasien sudah terisi dengan benar</li> <li>• Ketika klik tombol simpan dan belum mengisi form tambah pencegahan asma pasien secara lengkap / <i>field</i> masih ada yang kosong</li> </ul>	<ul style="list-style-type: none"> <li>• Menambah pencegahan asma pasien sesuai yang diinputkan oleh pakar</li> <li>• Menampilkan pesan data tidak boleh kosong</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal  <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
2.	Edit data pencegahan asma pasien	<ul style="list-style-type: none"> <li>• Ketika klik tombol <i>update</i> dan semua <i>field</i> pada form edit data pencegahan asma sudah terisi dengan benar</li> <li>• Ketika klik tombol <i>update</i> dan belum mengisi form edit data pencegahan asma pasien secara lengkap / <i>field</i> masih ada yang kosong</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Update</i> data solusi asma sesuai yang diinputkan oleh pakar</li> <li>• Menampilkan pesan data tidak boleh kosong</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal  <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
3.	Hapus data data pencegahan asma pasien	<ul style="list-style-type: none"> <li>• Ketika klik tombol delete data yang akan dihapus dan klik tombol delete pada pop – up</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Delete</i> data solusi asma sesuai yang dipilih pakar</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

		<ul style="list-style-type: none"> <li>• Ketika klik tombol delete dan memlih cancel</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan halaman sebelumnya</li> </ul>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
--	--	---	--	---

Tabel diatas merupakan hasil pengujian *black box* pada fitur mengelola data pencegahan asma pasien. Hasil pengujian menunjukkan impementasi sistem pada menejemem data pencegahan asma pasien sesuai dengan harapan pakar.

Tabel 4. 16 Pengujian *Black Box* Mengelola Data Gabungan

No	Fitur	Kasus	Hasil	Keterangan
1.	Tambah data gabungan	<ul style="list-style-type: none"> <li>• Ketika klik tombol tambah dan semua <i>field</i> pada form tambah data gabungan sudah terisi dengan benar</li> <li>• Ketika klik tombol simpan dan belum mengisi form tambah data gabungan secara lengkap / <i>field</i> masih ada yang kosong</li> </ul>	<ul style="list-style-type: none"> <li>• Menambah pencegahan asma pasien sesuai yang diinputkan oleh pakar</li> <li>• Menampilkan pesan data tidak boleh kosong</li> </ul>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal  <input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
2.	Edit data gabungan	<ul style="list-style-type: none"> <li>• Ketika klik tombol <i>update</i> dan semua <i>field</i> pada form edit data gabungan sudah terisi dengan benar</li> <li>• Ketika klik tombol <i>update</i> dan belum mengisi form edit data gabungan secara lengkap / <i>field</i> masih ada yang kosong</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Update</i> data solusi asma sesuai yang diinputkan oleh pakar</li> <li>• Menampilkan pesan data tidak boleh kosong</li> </ul>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal  <input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

3.	Hapus data gabungan	<ul style="list-style-type: none"> <li>• Ketika klik tombol delete data yang akan dihapus dan klik tombol delete pada pop – up</li> <li>• Ketika klik tombol delete dan memilih cancel</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Delete</i> data solusi asma sesuai yang dipilih pakar</li> <li>• Menampilkan halaman sebelumnya</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal  <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
----	---------------------	---	--	--

Tabel diatas merupakan hasil pengujian *black box* pada fitur mengelola data gabungan. Hasil pengujian menunjukkan impementasi sistem pada menejemen data gabungan sesuai dengan harapan pakar.

Tabel 4. 17 Pengujian *Black Box* Mendiagnosa Asma Pasien

No	Fitur	Kasus	Hasil	Keterangan
1.	diagnosa	<ul style="list-style-type: none"> <li>• Ketika klik tombol konsultasi dan user telah memilih gejala – gejala yang ada pada form diagnosa</li> <li>• Ketika klik tombol diagnosa dan user belum memilih gejala – gejala yang ada pada form diagnosa</li> <li>• Ketika klik tombol konsultasi dan user memilih semua data gejala yang ada pada form diagnosa</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan hasil perhitungan konsultasi user berupa data gejala</li> <li>• Menampilkan pesan pilih gejala terlebih dahulu</li> <li>• Menampilkan hasil perhitungan konsultasi user berupa tingkatan asma.</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal  <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal  <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

Tabel diatas merupakan hasil pengujian *black box* pada fitur diagnosa asma pasien. Fitur diagnosa merupakan fitur yang dapat diakses user, user disini merupakan siapa saja yang dapat mengakses sistem. Fitur diagnosa merupakan fitur utama dalam sistem yang melakukan perhitungan metode *certainty factor*.

Tabel 4. 18 Pengujian *Black Box* Pencegahan Asma Pasien

No	Fitur	Kasus	Hasil	Keterangan
1.	diagnosa	<ul style="list-style-type: none"> <li>• Ketika klik tombol konsultasi dan user telah memilih pertanyaan yang ada pada form pencegahan</li> <li>• Ketika klik tombol konsultasi dan user belum memilih pertanyaan yang ada pada form pencegahan</li> <li>• Ketika klik tombol konsultasi dan user memilih pertanyaan yang ada pada form pencegahan</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan jawaban dari pertanyaan yang telah dipilih oleh user</li> <li>• Menampilkan pesan pilih gejala terlebih dahulu</li> <li>• Menampilkan jawaban dari pertanyaan yang dipilih oleh user</li> </ul>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal  <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal  <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

Tabel diatas merupakan hasil pengujian *black box* pada fitur pencegahan asma pasien. Fitur pencegahan asma pasien merupakan fitur yang dapat diakses user, user disini merupakan siapa saja yang dapat mengakses sistem.



## BAB 6. PENUTUP

### 6.1 Kesimpulan

Kesimpulan yang dapat ditarik dari penelitian yang telah dilakukan sebagai berikut :

1. Sistem mampu membantu melakukan diagnosis tingkatan asma pasien dan penanganannya dengan tepat berdasarkan gejala – gejala yang ada dan perbandingan antara diagnosis sistem dengan diagnosis dokter tepat.
2. Dapat diketahui bagaimanakah cara meminimalisir serangan dengan melakukan pencegahan yang tepat untuk pasien agar terhindar dari serangan asma
3. Berdasarkan studi literatur, wawancara, dan penelitian dari beberapa jurnal, terdapat 3 macam kategori gejala dengan 3 macam gejala pada setiap kategori. Dan berdasarkan penelitian pakar data macam – macam gejala asma ini mampu dipertanggung jawabkan kebenarannya karena sudah melalui berbagai penelitian dan telah menjadi acuan internasional oleh banyak dokter spesialis.
4. Sistem pakar dapat berfungsi sebagai pembantu dan pengganti pakar dalam menyelesaikan masalah yang ada. Pada sistem ini, sistem pakar digunakan sebagai pengganti pakar dalam melakukan diagnosis.

### 6.2 Saran

Beberapa saran dan masukan berikut diharapkan dapat memberikan perbaikan dalam penelitian selanjutnya, yaitu :

1. Pengembangan lebih lanjut pada penelitian ini diharapkan dapat menambahkan berbagai macam gejala jika memang terdapat jurnal baru yang menjadi kesepakatan banyak dokter dan dapat tampilkan daftar gejala yang telah dipilih untuk didiagnosa berdasarkan pilihan dari dokter atau pun pasien.
2. Penerapan metode *certainty factor* dapat diterapkan pada komoditas lain hasil dengan memodifikasi data set pada database data gejala, data tingkatan asma dan data gabungan, perubahan data set ini dimaksudkan untuk memudahkan pemanggilan database pada sistem sesuai dengan penelitian yang sedang dilakukan.

**DAFTAR PUSTAKA**

- (GINA), G. I. (2013). Global Burden of Asthma-Global Initiative for Asthma.
- (NHLBI), N. H. (2007). Guidelines for the Diagnosis and Management of Asthma.
- (PDPI), P. D. (2015). Tuberkulosis Paru.
- Dr. Moffat, M. (2007). Applying the Preferred Physical Therapist Practice.
- Fitriastuti, F. (2009). Aplikasi Sistem Pakar Berbasis Web Untuk Mendeteksi Kerusakan Perangkat Komputer Dengan Metode Backward Chaining.
- Hudoyo, A. (2014). Penatalaksanaan Asma & PPOK Pada Orang Dewasa berdasar Pedoman GINA & GOLD.
- Jusak. (2010). Sistem Pakar : Buku. Pegangan Kuliah.
- PDPI. (2011). PPOK (Penyakit Paru Obstruktif Kronik), pedoman praktis diagnosis dan penatalaksanaan di Indonesia.
- Siswanto. (2010). Kecerdasan Tiruan.
- T.Sutojo, E. M. (2010). Kecerdasan Buatan.
- WHO. (2013). World Health Statistic 2013.

## LAMPIRAN

Lampiran *screenshot* kode program C\_info

```
1  <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3  class C_info extends CI_Controller {
4
5      public function __construct(){
6          parent::__construct();
7          $this->load->helper('url');
8          $this->load->library('input');
9          $this->load->model('M_data_cegah');
10     }
11
12     public function index()
13     {
14         $pertanyaan = array();
15         $pert = $this->input->post();
16
17
18         $this->db->from('data_cegah');
19         foreach($pert['pertanyaan'] as $row)
20         {
21             $this->db->or_where('IDCegah', $row);
22         }
23
24         $data3['pertanyaan'] = $this->db->get()->result();
25
26         $data1=array(
27             'title'=>'pertanyaan',
28             'active_diagnosa'=>'active');
29
30
31         $this->load->view('element/header',$data1);
32         $this->load->view('element/navbar_user');
33
34         $this->load->view('pages/hasil_tanya', $data3);
35         $this->load->view('element/footer');
36     }
37 }
```

Lampiran *screenshot* kode program login

```
1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class Login extends CI_Controller {
4
5
6 public function __construct(){
7     parent::__construct();
8     $this->load->helper('url');
9     $this->load->library('input');
10    $this->load->library('session');
11    $this->load->model('User_Model');
12 }
13
14 public function index()
15 {
16     // mengecek masukan dari form
17     $username = $this->input->post('username');
18     $password = $this->input->post('password');
19     $temp_account = $this->User_Model->cek_user($username, $password)->row();
20
21
22     $num_account = count($temp_account);
23     if ($num_account > 0){
24
25         $array_items = array(
26             'IDUser' => $temp_account->IDUser,
27             'username' => $temp_account->username,
28             'logged_in' => true
29         );
30
31         if ($temp_account->IDUser == "1") {
32             $this->session->set_userdata($array_items);
33             redirect(base_url('index.php/utama_pakar'));
34         }
35
36     } else {
37         |
38         redirect(base_url()."?pesan=Login Gagal. Username atau Password Anda Salah !");
39     }
40 }
41 }
42 }
```

Lampiran *screenshot* kode program Cdata\_penyakit\_pakar

```
1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class CData_penyakit_pakar extends CI_Controller {
4
5     public function __construct(){
6         parent::__construct();
7         $this->load->helper('url');
8         $this->load->library('input');
9         $this->load->library('session');
10        $this->load->model('M_data_penyakit');
11    }
12    public function tambah_penyakit(){
13        if ($this->session->userdata('IDUser')) {
14            $data=array(
15                'title'=>'Data Hama dan Penyakit',
16                'active_info'=>'active');
17
18            $data = array(
19                'id_otomatis'=> $this->M_data_penyakit->id_otomatis(),
20            );
21
22            $this->load->view('element/header',$data);
23            $this->load->view('element/navbar_pakar');
24
25            $this->load->view('pages/tambah_penyakit_pakar',$data);
26            $this->load->view('element/footer');
27        }
28        else{
29            redirect(site_url('welcome/'));
30        }
31    }
32
33    public function proses_tambah_penyakit(){
34        $data['IDSolusi'] = $this->input->post('IDSolusi');
35        $data['Berat_asma'] = $this->input->post('Berat_asma');
36        $data['Medikasi_harian'] = $this->input->post('Medikasi_harian');
37        $data['Medikasi_alternatif'] = $this->input->post('Medikasi_alternatif');
38        $this->M_data_penyakit->insert_penyakit($data);
39        redirect(site_url('utama_pakar/data_penyakit'));
40    }
41
42    public function edit_penyakit($ID_penyakit){
43        if ($this->session->userdata('IDUser')) {
44            $data=array(
45                'title'=>'Data Peyakit',
46                'active_info'=>'active');
47
48            $this->load->view('element/header',$data);
49            $this->load->view('element/navbar_pakar');
50
51            $data['penyakit'] = $this->M_data_penyakit->select_by_id($ID_penyakit)->row();
52            $this->load->view('pages/edit_penyakit_pakar', $data);
53            $this->load->view('element/footer');
54        }
55        else{
56            redirect(site_url('welcome/'));
57        }
58    }
59
60    public function proses_edit_penyakit(){
61
62        $data['Berat_asma'] = $this->input->post('Nama');
63        $data['Medikasi_harian'] = $this->input->post('Keterangan');
64        $id_penyakit = $this->input->post('IDPenyakitdanHama');
65        $this->M_data_penyakit->update_penyakit($id_penyakit, $data);
66        redirect(site_url('utama_pakar/data_penyakit'));
67    }
68
69    public function delete_penyakit($id_penyakit){
70        $this->M_data_penyakit->delete_penyakit($id_penyakit);
71        redirect(site_url('utama_pakar/data_penyakit'));
72    }
73 }
```

Lampiran *screenshot* kode program Cdata\_gabungan\_pakar

```

1  <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3  class Cdata_gabungan_pakar extends CI_Controller {
4
5      public function __construct(){
6          parent::__construct();
7          $this->load->helper('url');
8          $this->load->library('input');
9          $this->load->library('session');
10         $this->load->model('M_data_gejala');
11         $this->load->model('M_data_penyakit');
12         $this->load->model('M_data_gabungan');
13     }
14
15     public function tambah_gabungan(){
16         if ($this->session->userdata('IDUser')) {
17             $data=array(
18                 'title'=>'Data Gabungan',
19                 'active_about'=>'active');
20
21             $this->load->view('element/header',$data);
22             $this->load->view('element/navbar_pakar');
23
24             $data['jangka'] = $this->M_data_gejala->select_jangka()->result();
25             $data['faal'] = $this->M_data_gejala->select_faal()->result();
26             $data['malam'] = $this->M_data_gejala->select_malam()->result();
27             $data['daftar_penyakit'] = $this->M_data_penyakit->select_all()->result();
28             $this->load->view('pages/tambah_gabungan_pakar', $data);
29             $this->load->view('element/footer');
30         }
31         else{
32             redirect(site_url('welcome/'));
33         }
34     }
35
36     public function proses_tambah_gabungan(){
37         $penyakit = $this->input->post('penyakit');
38         $gejala = $this->input->post('gejala');
39         $data['NilaiMD'] = $this->input->post('NilaiMD');
40
41         $query2 = mysql_query("select * from data_solusi where IDSolusi = '".$penyakit."'");
42         while($data2 = mysql_fetch_array($query2)){
43             $data['IDSolusi'] = $data2['IDSolusi'];
44         }
45         $query3 = mysql_query("select * from data_gejala where Gejala = '".$gejala."'");
46         while($data3 = mysql_fetch_array($query3)){
47             $data['IDGejala'] = $data3['IDGejala'];
48         }
49
50         $querycek = mysql_num_rows( mysql_query("select * from data_gabungan where IDSolusi = '".$data['IDSolusi']."'
51 AND IDGejala = '".$data['IDGejala']."'"));
52         if ($querycek){
53             redirect(site_url('Cdata_gabungan_pakar/tambah_gabungan?pesan=gagal'));
54         }
55         else{
56
57             $this->M_data_gabungan->insert_gabungan($data);
58             $nilai_MB = 0.90;
59             $nilai_MD = $data['NilaiMD'];
60             $query4 = mysql_query("select * from data_gabungan");
61             while($data4 = mysql_fetch_array($query4)){
62
63                 if( $data4['IDGejala'] == $data['IDGejala']){
64                     $nilai_CF = $nilai_MB - $nilai_MD ;
65                 }
66             }
67
68             $data2['nilaiCF'] = $nilai_CF;
69             $data2['nilaiMB'] = $nilai_MB;
70             $this->M_data_gabungan->update_nilaiMB($data['IDGejala'], $data2);
71             redirect(site_url('utama_pakar/data_gabungan'));
72         }
73     }
74 }
75
76

```

```
77
78
79 public function edit_gabungan($id_penyakit, $id_gejala){
80     if ($this->session->userdata('IDUser')) {
81         $data=array(
82             'title'=>'Data Gabungan',
83             'active_about'=>'active');
84
85         $this->load->view('element/header',$data);
86         $this->load->view('element/navbar_pakar');
87
88         $data['IDSolusi']         = $id_penyakit;
89         $data['IDGejala']        = $id_gejala;
90         $data['cf']              = $this->M_data_gabungan->select_by_id($id_penyakit, $id_gejala)->row();
91         $this->load->view('pages/edit_cf', $data);
92         $this->load->view('element/footer');
93     }
94     else{
95         redirect(site_url('welcome/'));
96     }
97 }
98
99 public function proses_edit_gabungan(){
100     $nilai_MB         = $this->input->post('NilaiMB');
101     $nilai_MD         = $this->input->post('NilaiMD');
102     $data['NilaiMD']  = $nilai_MD;
103     $id_penyakit      = $this->input->post('IDPenyakit');
104     $id_gejala        = $this->input->post('IDGejala');
105     $data['NilaiCF']  = $nilai_MB - $nilai_MD;
106     // $data['IDUser'] = $this->session->userdata('IDUser');
107     $this->M_data_gabungan->update_gabungan($id_penyakit, $id_gejala, $data);
108     redirect(site_url('utama_pakar/data_gabungan'));
109 }
110
111 public function delete_gabungan($id_penyakit, $id_gejala){
112     $this->M_data_gabungan->delete_gabungan($id_penyakit, $id_gejala);
113     redirect(site_url('utama_pakar/data_gabungan'));
114 }
115 }
```



Lampiran *screenshot* kode program controller Cdata\_gejala\_pakar

```
1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class Cdata_gejala_pakar extends CI_Controller {
4
5     public function __construct(){
6         parent::__construct();
7         $this->load->helper('url');
8         $this->load->library('input');
9         $this->load->library('session');
10        $this->load->model('M_data_gejala');
11    }
12    public function tambah_gejala(){
13        if ($this->session->userdata('IDUser')) {
14            $data2=array(
15                'title'=>'Data Gejala',
16                'active_konsultasi'=>'active');
17
18            $data = array(
19                'id_otomatis'=> $this->M_data_gejala->id_otomatis(),
20            );
21
22            $this->load->view('element/header',$data2);
23            $this->load->view('element/navbar_pakar');
24            $data['jangka'] = $this->M_data_gejala->select_jangka()->result();
25            $data['faal'] = $this->M_data_gejala->select_faal()->result();
26            $data['malam'] = $this->M_data_gejala->select_malam()->result();
27            $this->load->view('pages/tambah_gejala_pakar',$data);
28            $this->load->view('element/footer');
29        }
30        else{
31            redirect(site_url('welcome/'));
32        }
33    }
34
35    public function proses_tambah_gejala(){
36        $data['IDGejala'] = $this->input->post('IDGejala');
37        $data['Gejala'] = $this->input->post('Gejala');
38        $data['Kategori'] = $this->input->post('Kategori');
39
40
41        $this->M_data_gejala->insert_gejala($data);
42        redirect(site_url('utama_pakar/data_gejala'));
43    }
44
45    public function edit_gejala($ID_gejala){
46        if ($this->session->userdata('IDUser')) {
47            $data=array(
48                'title'=>'Data gejala',
49                'active_konsultasi'=>'active');
50
51            $this->load->view('element/header',$data);
52            $this->load->view('element/navbar_pakar');
53
54            $data['gejala'] = $this->M_data_gejala->select_by_id($ID_gejala)->row();
55            $this->load->view('pages/edit_gejala_pakar', $data);
56            $this->load->view('element/footer');
57        }
58        else{
59            redirect(site_url('welcome/'));
60        }
61    }
62
63    public function proses_edit_gejala(){
64        $id_gejala = $this->input->post('IDGejala');
65        $data['Gejala'] = $this->input->post('Gejala');
66        // $data['IDUser'] = $this->session->userdata('IDUser');
67        $this->M_data_gejala->update_gejala($id_gejala, $data);
68        redirect(site_url('utama_pakar/data_gejala'));
69    }
70
71    public function delete_gejala($id_gejala){
72        $this->M_data_gejala->delete_gejala($id_gejala);
73        redirect(site_url('utama_pakar/data_gejala'));
74    }
75 }
```

Lampiran *screenshot* kode program M\_data\_cegah

```
1 |<?php
2 | class M_data_cegah extends CI_Model {
3 |     function __construct(){
4 |         parent::__construct();
5 |     }
6 |
7 |     function select_all(){
8 |         $this->db->select('*');
9 |         $this->db->from('data_cegah');
10 |         return $this->db->get();
11 |     }
12 |
13 |     function insert_cegah($data){
14 |         $this->db->insert('data_cegah', $data);
15 |     }
16 |
17 |     function select_tanya(){
18 |         $this->db->select('*');
19 |         $this->db->from('data_cegah');
20 |         return $this->db->get();
21 |     }
22 |
23 |     function select_by_id($idCegah){
24 |         $this->db->select('*');
25 |         $this->db->from('data_cegah');
26 |         return $this->db->get();
27 |     }
28 |
29 |     function select_id($idCegah){
30 |         $this->db->select('*');
31 |         $this->db->from('data_cegah');
32 |         $this->db->where('IDCegah', $idCegah);
33 |         return $this->db->get();
34 |     }
35 |
36 |     function update_cegah($idCegah, $data){
37 |         $this->db->where('IDCegah', $id_cegah);
38 |         $this->db->update('data_cegah', $data);
39 |     }
40 |
41 |     function delete_cegah($id_cegah){
42 |         $this->db->where('IDCegah', $id_cegah);
43 |         $this->db->delete('data_cegah');
44 |     }
45 |
46 |     function id_otomatis(){
47 |         $id = $this->db->query("select MAX(IDCegah) as id_max from data_cegah");
48 |         $kd = '';
49 |         if ($id->num_rows() > 0) {
50 |             foreach ($id->result_array() as $k) {
51 |                 $kd = ((int) $k['id_max']) + 1;
52 |             }
53 |         } else {
54 |             $kd = '1';
55 |         }
56 |         return $kd;
57 |     }
58 | }
```

## Lampiran screenshot kode program M\_data\_gabungan

```
1 |<?php
2 | class M_data_gabungan extends CI_Model {
3 |     function __construct(){
4 |         parent::__construct();
5 |     }
6 |     function insert_gabungan($data){
7 |         $this->db->insert('data_gabungan', $data);
8 |     }
9 |     function select_all(){
10 |         $this->db->select('*');
11 |         $this->db->from('data_gabungan');
12 |         $this->db->join('data_gejala', 'data_gejala.IDGejala = data_gabungan.IDGejala');
13 |         $this->db->join('data_solusi', 'data_solusi.IDSolusi = data_gabungan.IDSolusi');
14 |         return $this->db->get();
15 |     }
16 |     function select_by_id($id_penyakit, $id_gejala){
17 |         $this->db->select('*');
18 |         $this->db->from('data_gabungan');
19 |         $this->db->where('IDSolusi', $id_penyakit);
20 |         $this->db->where('IDGejala', $id_gejala);
21 |         return $this->db->get();
22 |     }
23 |     function update_gabungan($id_penyakit, $id_gejala, $data){
24 |         $this->db->where('IDSolusi', $id_penyakit);
25 |         $this->db->where('IDGejala', $id_gejala);
26 |         $this->db->update('data_gabungan', $data);
27 |     }
28 |     function update_nilaiMB($id_gejala, $data){
29 |         $this->db->where('IDGejala', $id_gejala);
30 |         $this->db->update('data_gabungan', $data);
31 |     }
32 |     function delete_gabungan($id_penyakit, $id_gejala){
33 |         $this->db->where('IDSolusi', $id_penyakit);
34 |         $this->db->where('IDGejala', $id_gejala);
35 |         $this->db->delete('data_gabungan');
36 |     }
37 |     function select_id_gejala($id_gejala){
38 |         $this->db->select('*');
39 |         $this->db->from('data_gabungan');
40 |         $this->db->where('IDGejala', $id_gejala);
41 |         return $this->db->get();
42 |     }
43 |     function select_id_solusi($id_penyakit){
44 |         $this->db->select('*');
45 |         $this->db->from('data_gabungan');
46 |         $this->db->where('IDSolusi', $id_penyakit);
47 |         return $this->db->get();
48 |     }
49 |     function select_solusi($idsolusi){
50 |         $this->db->select('*');
51 |         $this->db->from('data_solusi');
52 |         $this->db->where('IDSolusi', $idsolusi);
53 |         return $this->db->get();
54 |     }
55 | }
56 | ?>
```

Lampiran *screenshot* kode program M\_data\_gejala

```
1 <?php
2 class M_data_gejala extends CI_Model {
3     function __construct(){
4         parent::__construct();
5     }
6     function select_all(){
7         $this->db->select('*');
8         $this->db->from('data_gejala');
9         return $this->db->get();
10    }
11    function insert_gejala($data){
12        $this->db->insert('data_gejala', $data);
13    }
14    function select_jangka(){
15        $this->db->select('*');
16        $this->db->from('data_gejala');
17        $this->db->where('Kategori', 'Jangka Serangan');
18        return $this->db->get();
19    }
20    function select_faal(){
21        $this->db->select('*');
22        $this->db->from('data_gejala');
23        $this->db->where('Kategori', 'Faal Paru');
24        return $this->db->get();
25    }
26    function select_malam(){
27        $this->db->select('*');
28        $this->db->from('data_gejala');
29        $this->db->where('Kategori', 'Gejala Malam');
30        return $this->db->get();
31    }
32    function select_by_id($id_gejala){
33        $this->db->select('*');
34        $this->db->from('data_gejala');
35        $this->db->where('IDGejala', $id_gejala);
36        return $this->db->get();
37    }
38    function update_gejala($id_gejala, $data){
39        $this->db->where('IDGejala', $id_gejala);
40        $this->db->update('data_gejala', $data);
41    }
42    function delete_gejala($id_gejala){
43        $this->db->where('IDGejala', $id_gejala);
44        $this->db->delete('data_gejala');
45    }
46    function id_otomatis(){
47        $id = $this->db->query("select MAX(IDGejala) as id_max from data_gejala");
48        $skd = '';
49        if ($id->num_rows() > 0) {
50            foreach ($id->result_array() as $k) {
51                $skd = ((int) $k['id_max']) + 1;
52            }
53        } else {
54            $skd = '1';
55        }
56        return $skd;
57    }
58 }
59 ?>
```

Lampiran *screenshot* kode program M\_data\_pakar

```
1 |<?php
2 | class M_data_pakar extends CI_Model {
3 |     function __construct(){
4 |         parent::__construct();
5 |     }
6 |     function cek_user($username, $password){
7 |         $this->db->select('*');
8 |         $this->db->from('user');
9 |         $this->db->where('username', $username);
10 |        $this->db->where('password', $password);
11 |        return $this->db->get();
12 |    }
13 |    function select_all(){
14 |        $this->db->select('*');
15 |        $this->db->from('user');
16 |        $this->db->where('IDUser = 2');
17 |        return $this->db->get();
18 |    }
19 |    function select_by_id($id_pakar){
20 |        $this->db->select('*');
21 |        $this->db->from('user');
22 |        $this->db->where('IDUser', $id_pakar);
23 |        return $this->db->get();
24 |    }
25 |    function update_pakar($id_pakar, $data){
26 |        $this->db->where('IDUser', $id_pakar);
27 |        $this->db->update('user', $data);
28 |    }
29 |    function id_otomatis(){
30 |        $id = $this->db->query("select MAX(IDUser) as id_max from user");
31 |        $skd = '';
32 |        if ($id->num_rows() > 0) {
33 |            foreach ($id->result_array() as $k) {
34 |                $skd = ((int) $k['id_max']) + 1;
35 |            }
36 |        } else {
37 |            $skd = '1';
38 |        }
39 |        return $skd;
40 |    }
41 | }
42 | ?>
```

Lampiran *screenshot* kode program M\_data\_penyakit

```
1 <?php
2 class M_data_penyakit extends CI_Model {
3     function __construct(){
4         parent::__construct();
5     }
6     function insert_penyakit($data){
7         $this->db->insert('data_solusi', $data);
8     }
9     function select_all(){
10        $this->db->select('*');
11        $this->db->from('data_solusi');
12        return $this->db->get();
13    }
14    function select_by_id($id_penyakit){
15        $this->db->select('*');
16        $this->db->from('data_solusi');
17        $this->db->where('IDSolusi', $id_penyakit);
18        return $this->db->get();
19    }
20    function update_penyakit($id_penyakit, $data){
21        $this->db->where('IDSolusi', $id_penyakit);
22        $this->db->update('data_solusi', $data);
23    }
24    function delete_penyakit($id_penyakit){
25        $this->db->where('IDSolusi', $id_penyakit);
26        $this->db->delete('data_solusi');
27    }
28    function id_otomatis(){
29        $id = $this->db->query("select MAX(IDSolusi) as id_max from data_solusi");
30        $kd = '';
31        if ($id->num_rows() > 0) {
32            foreach ($id->result_array() as $k) {
33                $kd = ((int) $k['id_max']) + 1;
34            }
35        } else {
36            $kd = '1';
37        }
38        return $kd;
39    }
40 }
41 ?>
```