

**SISTEM PAKAR DIAGNOSIS PENYAKIT TANAMAN CABAI
MENGUNAKAN METODE *FORWARD CHAINING*
BERBASIS ANDROID**

(Skripsi)

Oleh

MITA FULJANA



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2017**

ABSTRAK

SISTEM PAKAR DIAGNOSIS PENYAKIT TANAMAN CABAI MENGUNAKAN METODE *FORWARD CHAINING* BERBASIS ANDROID

Oleh

MITA FULJANA

Penelitian ini dilakukan untuk membuat suatu sistem pakar yang mampu mendiagnosis penyakit pada tanaman cabai berdasarkan pengetahuan yang diberikan langsung dari pakar/ahlinya. Penelitian ini menggunakan metode perhitungan probabilitas klasik dalam menghitung presentase diagnosis dan dibuat pada *mobile device platform* Android. Pada penelitian ini terdiri dari 37 data gejala, 10 data penyakit cabai yang disebabkan oleh jamur, dan 10 data aturan. Sistem pakar ini menggunakan metode inferensi *forward chaining*. Hasil pengujian menunjukkan bahwa: (1) Pengujian fungsional dengan menggunakan metode *Black Box Equivalence Partitioning* (EP) mendapatkan hasil sesuai dengan yang diharapkan pada skenario uji di setiap kelas uji. (2) Pengujian kepakaran dengan membandingkan hasil perhitungan manual dan sistem sudah sesuai dan berjalan baik. (3) Pengujian kuesioner dengan 53 responden yang dibagi menjadi empat kelompok responden menunjukkan; kelompok responden pertama yang terdiri dari pakar penyakit cabai mendapatkan rata-rata nilai sebesar 85,14 % (dikategorikan sangat baik), kelompok responden kedua yang terdiri mahasiswa jurusan Pertanian 84,13 % (dikategorikan sangat baik), kelompok responden ketiga yang terdiri dari mahasiswa jurusan Ilmu Komputer 84,28 % (dikategorikan sangat baik), dan kelompok responden keempat yang terdiri dari petani cabai 86 % (dikategorikan sangat baik).

Kata Kunci: Sistem Pakar, Probabilitas Klasik, *Forward Chaining*, Penyakit Cabai, Skala *Likert*, Android

ABSTRACT

EXPERT SYSTEM OF CHILI PLANT DISEASE DIAGNOSIS USING *FORWARD CHAINING* METHOD BASED ANDROID

By

MITA FULJANA

This research was conducted to make an expert system that is able to diagnose disease in chili plants based on knowledge that provided directly from the experts. This research uses classical probability calculation method in calculating the percentage of diagnoses and created on the Android mobile device platform. In this research consisted of 37 symptoms data, 10 data of chili disease caused by fungi, and 10 rules of data. This expert system uses inference forward chaining method . Test results show that: (1) Functional testing using the Black Box Equivalence Partitioning (EP) method get the results as expected on the test scenario in each test class. (2) Expert testing by comparing the results of manual and system calculations are already appropriate and running well. (3) Tests using questionnaires with 53 respondents which is divided into four groups of respondents shows; the first respondents group that is consisting of experts of chili disease got an average value of 85.14% (categorized excellent), the second respondents group that is consisting of students of Agriculture Department 84.13% (categorized excellent), third respondent group that is consisting of students of Computer Science Department 84.28% (categorized excellent), and fourth respondent group that is consisting of chili farmers 86% (categorized excellent).

Key Words : Android, Expert System, Chili Disease, Classic Probability, Forward Chaining, Likert Scale

**SISTEM PAKAR DIAGNOSIS PENYAKIT TANAMAN CABAI
MENGUNAKAN METODE *FORWARD CHAINING*
BERBASIS ANDROID**

Oleh

MITA FULJANA

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA ILMU KOMPUTER**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2017**

Judul Skripsi : **SISTEM PAKAR DIAGNOSIS PENYAKIT TANAMAN CABAI MENGGUNAKAN METODE FORWARD CHAINING BERBASIS ANDROID**

Nama Mahasiswa : **MITA FULJANA**

Nomor Pokok Mahasiswa : **1347051007**

Program Studi : **Ilmu Komputer**

Fakultas : **Matematika dan Ilmu Pengetahuan Alam**



Aristoteles, S.Si.,M.Si.
NIP.19810521 200604 1 002

Ir. Joko Prasetyo, M.P
NIP.19590214198902 1 001

2. Ketua Jurusan Ilmu Komputer


Dr. Ir. Kurnia Muludi, M.Sc.
NIP. 19640616 198902 1 001

MENGESAHKAN

1. Tim Penguji

Ketua : **Aristoteles, S.Si., M.Si.**

Sekretaris : **Ir. Joko Prasetyo, M.P.**


Penguji
Bukan Pembimbing: **Dr. Ir Kurnia Muludi, M.Sc.**

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Prof. Warsito, S.Si., D.E.A., Ph.D.

NIP. 19710212 199512 1 001

Tanggal Lulus Ujian Skripsi: **13 OKTOBER 2017**



PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul “Sistem Pakar Diagnosis Penyakit Tanaman Cabai Menggunakan Metode *Forward Chaining* Berbasis Android” merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila dikemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 13 Oktober 2017



MITA FULJANA

NPM. 1347051007

RIWAYAT HIDUP



Penulis dilahirkan di Jakarta pada tanggal 01 Mei 1995, sebagai anak kelima dari enam bersaudara, dari bapak Muhammad Nasir dan Ibu Yurnalis.

Penulis memiliki 4 orang kakak bernama Joni, Sasrini, Novi, dan Fadli serta memiliki 1 orang adik bernama Iqbal. Penulis menyelesaikan Pendidikan Sekolah Dasar (SD) diselesaikan di SDN 07 Cengkareng pada tahun 2007, Sekolah Menengah Pertama (SMP) di SMP Negeri 2 Sragi pada tahun 2010, dan Sekolah Menengah Atas (SMA) di SMA Negeri 1 Kalianda pada tahun 2013.

Pada tahun 2013, penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam melalui jalur ujian Paralel. Selama kuliah, penulis terdaftar dalam organisasi HIMAKOM (Himpunan Mahasiswa Jurusan Ilmu Komputer) periode 2013/2014. Pada bulan Januari-Maret 2016, penulis melakukan Kuliah Kerja Nyata selama 60 hari di Desa Bakung Udik, Kecamatan Gedung Meneng, Kabupaten Tulang Bawang. Pada bulan Juli 2016, penulis melakukan kerja praktik di Direktorat Reserse Narkoba Polda Lampung, Kecamatan Teluk Betung, Kabupaten Bandar Lampung.

Selama menjadi mahasiswa penulis pernah mendapat Program Kreatifitas Mahasiswa Kewirausahaan (PKM-K) pada tahun 2016.

PERSEMBAHAN

Puji dan syukur saya panjatkan kepada Allah SWT atas segala berkah-Nya sehingga skripsi ini dapat terselesaikan.

Teruntuk Bapak Mamaku yang sangat kucintai, kupersembahkan skripsi ini
Terimakasih untuk kasih sayang, perhatian, pengorbanan, usaha, dukungan moril maupun materi, motivasi dan do'a-do'a yang tiada henti untuk kesuksesanku...

Teruntuk Kakak-kakakku dan adikku yang sangat aku sayangi Uda Joni, Teta , Bang Novi,
Bang Fadli, dan Iqbal serta keluarga besar tercinta.

Terimakasih untuk pembimbingku yang selalu sabar memberikan bimbingan dan arahan.

Teruntuk sahabat dan teman-teman tersayang,
Terimakasih untuk canda tawa, tangis dan perjuangan yang telah kita lewati bersama....

Teruntuk Keluarga Ilmu Komputer 2013,

Serta Almamater Tercinta, Universitas Lampung.

MOTO

“Percayalah, Tuhan tak pernah salah memberi rezeki”

(Mita Fuljana)

*"Lakukan yang terbaik, sehingga aku tak akan menyalahkan
diriku sendiri atas segalanya*

(Magdalena Neuner)

*“Jadikanlah sabar dan sholat sebagai penolongmu,
Sesungguhnya Allah bersama orang-orang yang sabar”.(Q.S*

Al-Baqarah: 153)

SANWACANA

Assalamualaikum wr. wb.

Alhamdulillah, segala puji bagi Allah SWT yang telah melimpahkan rahmat, hidayah, kesehatan dan karunia-Nya sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul “Sistem Pakar Diagnosa Penyakit Tanaman Cabai Menggunakan Metode *Forward Chaining* Berbasis Android” dengan baik

Dalam penyusunan skripsi ini, penulis banyak menghadapi kesulitan. Namun, berkat bantuan dan dorongan dari berbagai pihak, akhirnya penulis dapat menyelesaikan skripsi ini. Untuk itu pada kesempatan ini, penulis mengucapkan terimakasih kepada :

1. Kedua orang tua tercinta, Bapak M.Nasir dan Ibu Yurnalis yang telah memberikan doa, semangat, cinta dan kasih sayang, dukungan moril maupun materil untuk menyelesaikan skripsi ini.
2. Kakak-kakaku dan adikku tersayang, Uda Joni, Teta, Bang novi, Bang Fadli, dan Iqbal yang selalu memberikan dan motivasi dalam penyusunan skripsi ini
3. Bapak Aristoteles, S.Si., M.Si., sebagai pembimbing I yang telah memberikan ide dan masukan dalam pengerjaan skripsi serta memberikan dorongan sehingga skripsi ini dapat diselesaikan.
4. Bapak Ir. Joko Prasetyo, M.P sebagai pembimbing II penulis, yang telah membimbing dan memberikan bantuan, ide, kritik serta saran dalam penyusunan skripsi ini.

5. Bapak Ir. Dr. Kurnia Muludi, M.S.Sc sebagai pembahas sekaligus ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan masukan-masukan dan saran yang bermanfaat dalam skripsi ini.
6. Bapak Didik Kurniawan, S.Si.,M.T, selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
7. Bapak Prof. Warsito, S.Si., D.E.A., Ph.D selaku Dekan FMIPA Universitas Lampung.
8. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang telah memberikan ilmu pengetahuan yang bermanfaat bagi penulis.
9. Ibu Ade Nora dan Pak Irshan selaku staf administrasi di Jurusan Ilmu Komputer yang telah membantu segala urusan administrasi selama kuliah.
10. Mahasiswa dan Dosen Jurusan Agroteknologi Fakultas Pertanian Universitas Lampung yang telah memberikan masukan-masukan dalam skripsi ini.
11. Sahabat-sahabatku Cikmay, Revy, dan Upe yang tak pernah henti menemani, memberikan semangat, dan menghibur penulis dalam kondisi apapun.
12. Sahabat-sahabat seperjuangan Rina, Ici, Tika, Navia, Gita, Ncen, Ratu Bunga, dan Rizka yang telah menemani selama masa perkuliahan.
13. Teman-temanku tersayang Winda, Ayu, Bunga, Della, Debi, dan Reni yang telah memberikan semangat dan menghibur penulis dalam kondisi apapun.
14. Teman-teman seperjuangan, Keluarga Besar Ilmu Komputer angkatan 2013
15. Almamater tercinta, Universitas Lampung

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, akan tetapi sedikit harapan semoga skripsi ini dapat bermanfaat bagi perkembangan ilmu pengetahuan terutama bagi teman-teman Ilmu Komputer.

Bandar Lampung, 28 September 2017

Mita Fuljana

DAFTAR ISI

	Halaman
DAFTAR ISI	xiv
DAFTAR TABEL	xvii
DAFTAR GAMBAR	xviii
DAFTAR LAMPIRAN	xx
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
II. TINJAUAN PUSTAKA	6
2.1 Cabai	6
2.1.1 Kandungan Cabai	7
2.1.2 Budidaya Tanaman Cabai	7
2.1.3 Penyakit Pada Tanaman Cabai	8
2.2 Sistem Pakar	12
2.2.1 Definisi Sistem Pakar	13
2.2.2 Konsep Dasar Sistem Pakar	13
2.2.3 Ciri-ciri Sistem Pakar	15
2.2.4 Keuntungan Sistem Pakar	15
2.2.5 Kelemahan Sistem Pakar	15
2.2.6 Struktur Sistem Pakar	16
2.2.7 Basis Pengetahuan	18
2.2.8 Teknik Inferensi	19
2.2.9 <i>Forward Chainig</i>	19
2.2.9.1 Metode <i>Forward Chaining</i>	20
2.2.9.2 Contoh Kasus <i>Forward Chaining</i>	23
2.3 UML (<i>Unified Modelling Language</i>)	24

2.4	Pengujian Perangkat Lunak	29
2.4.1	<i>Equivalence Partioning</i>	30
2.4.2	Probabilitas Klasik	30
2.4.3	Skala <i>Likert</i>	31
III.	METODE PENELITIAN	33
3.1	Waktu dan Tempat Penelitian	33
3.2	Alat Pendukung	33
3.3	Tahapan Penelitian	34
3.4	Tahap Perumusan Masalah	34
3.4.1	Tahap Pengumpulan Data	34
3.4.2	Perancangan Sistem	35
3.4.2.1	Perancangan UML	36
3.4.2.2	Perancangan Antarmuka	46
3.5	Pengujian Sistem	51
IV.	PEMBAHASAN	55
4.1	Analisis Kebutuhan	55
4.2	Representasi Pengetahuan	56
4.3	Implementasi Sistem	56
4.4	Analisa Persentase Penyakit	57
4.5	Tampilan Aplikasi Sistem Pakar Diagnosis Penyakit Cabai	58
4.5.1	Tampilan Halaman <i>Splash Screen</i>	58
4.5.2	Tampilan Menu Utama	59
4.5.3	Tampilan Menu Data Penyakit	59
4.5.4	Tampilan Menu Konsultasi	61
4.5.5	Tampilan Halaman Tentang	62
4.5.6	Tampilan Menu Kritik dan Saran	62
4.5.7	Tampilan Menu Bantuan	64
4.5.8	Tampilan Menu <i>Exit</i>	64
4.6	Hasil Pengujian	65
4.6.1	Pengujian Internal	65
4.6.1.1	Pengujian Fungsional	65
4.6.1.2	Pengujian Kepakaran Sistem	71

4.6.2 Pengujian Eksternal	74
4.6.2.1 Analisis Hasil Kuisisioner	79
V. KESIMPULAN DAN SARAN	86
5.1 Kesimpulan	86
5.2 Saran	86

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Tabel	Halaman
2.1 Penyakit Tanaman Cabai yang Disebabkan Oleh Cendawan	10
2.2 Notasi Activity Diagram	27
2.3 Notasi <i>Sequence</i> Diagram	28
2.4 Tabel Kemungkinan Penyakit	31
2.5 Tingkat Preferensi Jawaban Skala <i>Likert</i>	32
3.1 Tabel Daftar Pengujian	52
4.1 Hasil Pengujian Versi Android	66
4.2 Hasil Pengujian Resolusi Layar dan Densitas Layar	67
4.3 Pengujian <i>User Interface</i>	68
4.4 Pengujian Fungsi dari Menu Aplikasi	69
4.5 Hasil Pengujian Kepakaran Sistem	71
4.6 Hasil Penilaian Kategori Responden I	75
4.7 Hasil Penilaian Kategori Responden II	76
4.8 Hasil Penilaian Kategori Responden III	77
4.9 Hasil Penilaian Kategori Responden IV	78
4.10 Kriteria <i>Index</i> Penilaian Hasil Kuisisioner	79
4.11 Hasil Kuisisioner dari Responden	85

DAFTAR GAMBAR

Gambar	Halaman
1.1 Perkembangan produksi cabai provinsi Lampung, tahun 2011-2013	1
2.1 Cabai Merah	6
2.2 Infeksi <i>Colletotrichum</i> spp	9
2.3 Kematian semaian cabai	9
2.4 Struktur Sistem Pakar	17
2.5 Proses <i>Forward Chaining</i>	20
2.6 Iterasi ke-1	21
2.7 Iterasi ke- 2	22
2.8 Iterasi ke- 3	22
2.9 Penyelesaian dengan <i>Forward Chaining</i>	24
2.10 Contoh Aktor	25
2.11 <i>Use Case</i>	25
3.1 Tahapan Penelitian	35
3.2 <i>Use Case Diagram</i>	37
3.3 <i>Activity Diagram</i> Data Penyakit	38
3.4 <i>Activity Diagram</i> Konsultasi Penyakit	38
3.5 <i>Activity Diagram</i> Bantuan	39
3.6 <i>Activity Diagram</i> Tentang	40
3.7 <i>Activity Diagram Exit</i>	41
3.8 <i>Sequence Diagram</i> Data Penyakit	42
3.9 <i>Sequence Diagram</i> Konsultasi Penyakit	43
3.10 <i>Sequence diagram</i> Tentang	44

3.11	<i>Sequence</i> Diagram Bantuan	45
3.12	<i>Sequence</i> Diagram <i>Exit</i>	45
3.13	Rancangan Halaman Menu Utama	46
3.14	Halaman Menu Data Penyakit	47
3.15	Halaman Menu Konsultasi Gejala	48
3.16	Halaman Menu Hasil Diagnosis	48
3.17	Halaman Detail Penyakit	49
3.18	Halaman Menu Tentang	49
3.19	Halaman Menu Bantuan	50
3.20	Halaman Kritik dan Saran	50
4.1	Tampilan <i>Splash Screen</i>	58
4.2	Tampilan Menu Utama	59
4.3	Halaman list Data Penyakit	60
4.4	Halaman Detail Penyakit	60
4.5	Halaman Menu Konsultasi	61
4.6	Halaman Hasil Diagnosis	62
4.7	Halaman Tentang	63
4.8	Halaman Kritik dan Saran	63
4.9	Halaman Bantuan	64
4.10	Tampilan Menu <i>Exit</i>	64
4.11	Grafik Jawaban untuk Pertanyaan 1	80
4.12	Grafik Jawaban untuk Pertanyaan 2	81
4.13	Grafik Jawaban untuk Pertanyaan 3	81
4.14	Grafik Jawaban untuk Pertanyaan 4	82
4.15	Grafik Jawaban untuk Pertanyaan 5	83
4.16	Grafik Jawaban untuk Pertanyaan 6	83
4.17	Grafik Jawaban untuk Pertanyaan 7	84

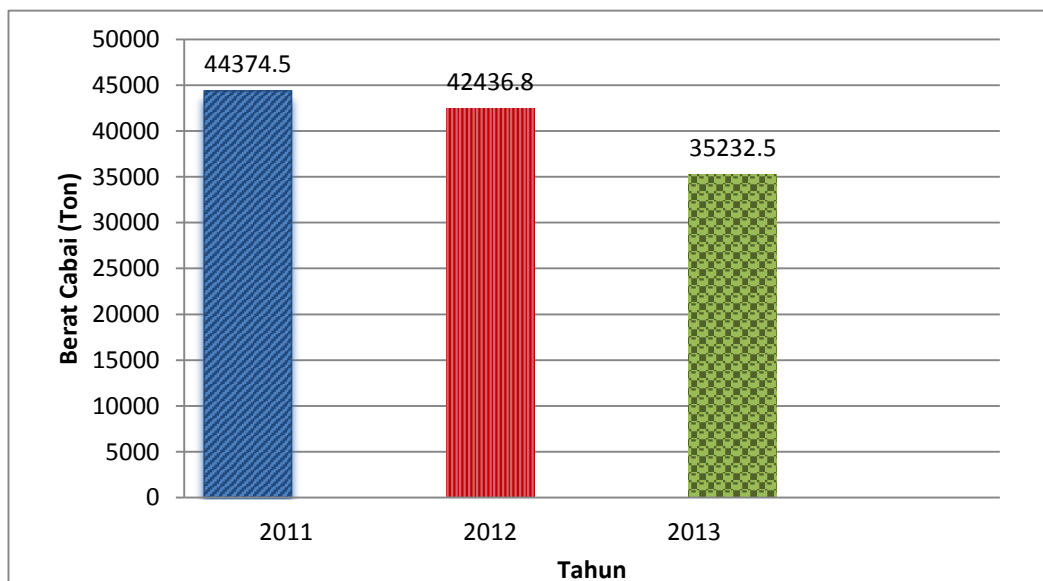
DAFTAR LAMPIRAN

Lampiran	Halaman
1 Data Penyakit	91
2 Data Gejala	92
3 Tabel Keputusan	94
4 Pohon Keputusan	97
5 Basis Aturan	98
6 Data Penyakit Dalam Bahasa Inggris	100
7 Data Gejala Dalam Bahasa Inggris	101
8 Tabel Keputusan Dalam Bahasa Inggris	103
9 Basis Aturan Dalam Bahasa Inggris	106

I. PENDAHULUAN

1.1 Latar Belakang

Cabai merupakan tanaman hortikultura cukup penting di Indonesia. Cabai tergolong tanaman buah dan sayuran yang mempunyai potensial untuk dikembangkan dan memiliki nilai ekonomi tinggi (Ralahalu et al, 2003). Menurut Badan Pusat Statistik Lampung (2014) produksi cabai selama tahun 2011 – 2013 cenderung menurun. Penurunan produksi cabai ini diikuti dengan kebutuhan konsumen yang tinggi. Pada Gambar 1. 1 merupakan grafik perkembangan produksi cabai diprovinsi Lampung.



Gambar 1.1 Perkembangan produksi cabai provinsi Lampung, tahun 2011-2013. Sumber : Badan Pusat Statistik Lampung, 2014.

Pada saat musim tertentu (musim hujan), produksi cabai berkurang, sementara permintaan konstan dan kontinu setiap hari, bahkan meningkat pada musim tertentu. Salah satu penyebab produksi menurun adalah penyakit tanaman cabai.

Adanya patogen yang menyerang tanaman cabai dapat menimbulkan gagal panen. Untuk itu diperlukan upaya pengendalian yang tepat agar tidak menggagalkan panen. Pengendalian tidak hanya saat serangan sudah ada, tetapi yang paling penting adalah tindakan mencegah agar penyakit tidak datang menyerang. Karena kurangnya pengetahuan petani dalam mengetahui jenis penyakit yang menyerang tanaman cabai dan bagaimana cara pengendaliannya sesuai dengan ciri-ciri penyakit yang terdapat pada tanaman tersebut membuat terlambatnya proses diagnosa (Muslim, 2015).

Proses diagnosa membutuhkan seorang pakar yang ahli dan berpengalaman agar menghasilkan diagnosa yang tepat. Namun demikian, keterbatasan waktu yang dimiliki seorang pakar terkadang menjadi kendala bagi para petani yang akan melakukan konsultasi guna menyelesaikan suatu permasalahan untuk mendapatkan solusi terbaik. Dalam hal ini sistem pakar dihadirkan sebagai alternatif kedua dalam memecahkan permasalahan setelah seorang pakar. Berdasarkan pemaparan yang dijelaskan maka dibutuhkan sistem pakar untuk mendiagnosa penyakit tanaman cabai dengan harapan dapat memodelkan sistem pakar diagnosa penyakit tanaman cabai yang dapat memberikan informasi mengenai gejala, penyakit, dan cara pengendaliannya.

Pada penelitian Anshori (2013) menghasilkan sistem pakar yang dapat mendiagnosa penyakit tanaman cabai berdasarkan gejala-gejalanya dan cara pengendaliannya. Penelitian tersebut memiliki kekurangan yaitu sistem pakar yang dibangun masih berbasis desktop dengan menggunakan PC sehingga kurang praktis dalam penggunaannya, karena hanya dapat digunakan pada komputer *stand alone*. Sedangkan sistem pakar berbasis android lebih efisien karena lebih mudah diakses di mana saja.

Pada penelitian lain (Muslim, 2015) menghasilkan sistem pakar yang dapat mendiagnosa hama dan penyakit pada tanaman cabai menggunakan Metode Teorema Bayes berbasis web. Berdasarkan hasil uji pretest dan posttest hasil keakurasian adalah sebesar 100%. Penelitian tersebut memiliki kekurangan yaitu sistem pakar yang dibangun masih berbasis web, sehingga membutuhkan koneksi internet yang stabil agar aplikasi berjalan dengan lancar dan data yang di gunakan masih sedikit. Berdasarkan hasil uji *pre test* dan *post test* hasil keakurasian adalah sebesar 100%.

Berdasarkan beberapa penelitian tersebut, maka dilakukan penelitian mengenai sistem pakar diagnosis penyakit tanaman cabai berbasis android dengan harapan dapat memodelkan sistem pakar diagnosis penyakit tanaman cabai yang dapat memberikan solusi secara tepat.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah bagaimana membangun suatu aplikasi sistem pakar untuk mendiagnosis penyakit tanaman cabai menggunakan metode *forward chaining* berbasis android.

1.3 Batasan Masalah

Batasan masalah pada penelitian ini antara lain:

1. Mendiagnosis penyakit tanaman cabai yang disebabkan oleh cendawan.
2. Aplikasi ini berbasis android.
3. Metode penalaran yang digunakan adalah *forward chaining*.
4. Penyakit yang dapat diidentifikasi sebanyak 10 penyakit dengan 37 gejala.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah untuk membangun aplikasi sistem pakar yang dapat mendiagnosis penyakit pada tanaman cabai menggunakan metode *forward chaining* berbasis android.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini antara lain:

1. Membantu masyarakat, petani atau penyuluh dalam mendapatkan informasi penyakit tanaman cabai.

2. Membantu masyarakat, petani atau penyuluh tanaman cabai dalam mendapatkan informasi penanganannya berdasarkan gejala yang terlihat.

II. TINJAUAN PUSTAKA

2.1 Cabai

Tanaman cabai merupakan salah satu komoditas hortikultura yang tergolong tanaman semusim. Tanaman cabai diperkirakan ada sekitar 20 spesies yang sebagian besar tumbuh di tempat asalnya, Amerika. Adapun klasifikasi tanaman cabai adalah sebagai berikut (Pitojo, 2003):

Kerajaan	: Plantae (Tumbuhan)
Divisi	: Spermathophyta
Sub divisi	: Angiospermae
Kelas	: Dicotyledoneae (Berkeping dua/dikotil)
Sub kelas	: Metachlamydeae
Ordo	: Solanales
Famili	: Solanaceae
Genus	: <i>Capsicum</i>
Spesies	: <i>Capsicum annuum</i> L.

Adapun cabai merah disajikan pada Gambar 2.1.



Gambar 2.1 Cabai Merah (Setiadi, 2015).

2.1.1 Kandungan Cabai

Cabai mengandung $\pm 1,5\%$ (biasanya antara 0,1 – 1%) rasa pedas. Rasa pedas tersebut terutama disebabkan oleh kandungan capsaicin. (Sumpena, 2013). Capsaicin terdapat pada biji cabai dan pada plasenta, yaitu kulit cabai bagian dalam yang berwarna putih tempat melekatnya biji. Rasa pedas tersebut bermanfaat untuk mengatur peredaran darah, memperkuat jantung, nadi dan syaraf, mencegah flu dan demam, membangkitkan semangat dalam tubuh (tanpa efek narkotik), serta mengurangi nyeri encok dan rematik (Prajnanta, 2001). Cabai juga memiliki manfaat bagi kesehatan tubuh, diantaranya adalah:

1. Penelitian terbaru mengindikasikan bahwa *capcaisin* mampu menurunkan berat badan pada orang yang menderita obesitas.
2. Makanan mengandung cabai mampu memperlambat proses terjadinya resiko penyakit kardiovaskular.
3. Cabai dianggap mampu mengendalikan pencemaran mikroba pada makanan.
4. Beberapa penelitian mengatakan bahwa *capcaisin* memiliki manfaat perlindungan antiulcer (obat penghambat produksi asam lambung) pada lambung yang terinfeksi bakteri *Helicobacter pylori* (Setiadi, 2015).

2.1.2 Budidaya Tanaman Cabai

Budidaya merupakan kegiatan terencana pemeliharaan sumber daya hayati yang dilakukan pada suatu areal lahan untuk diambil manfaat/hasil panennya (Pratama et al, 2015). Ada dua pola budidaya cabai di Indonesia yaitu budidaya ala brebes dan budidaya modern. Budidaya ala brebes petani menggunakan benih produksi

sendiri tanpa mulsa plastik, pengolahan lahan, dan pupuk. Sedangkan budidaya petani cabai modern menggunakan benih impor, mulsa plastik, pupuk, dan pestisida. Mereka juga mengolah lahan khusus untuk budidaya cabai. Biaya yang digunakan untuk budidaya cabai modern ini lebih mahal dibandingkan budidaya ala brebes (Alex, 2015).

2.1.3 Penyakit Pada Tanaman Cabai

Penyakit tanaman cabai yang disebabkan oleh cendawan (jamur) antara lain:

1. Rebah Kecambah atau *Damping Off*

Patogen : Salah satu dari *Rhizoctonia solani*, *Pythium* spp. *Fusarium* spp. *Phytophthora* sp. atau *Colletotrichum* spp.

Gejala : Semaian cabai gagal tumbuh, biji yang sudah berkecambah mati tiba-tiba (Gambar 2.2) atau semaian kerdil karena batang bawah atau leher akar busuk dan mengering (Gambar 2.3). Pada bedengan persemaian nampak kebotakan kecambah atau semaian cabai secara sporadis dan menyebar tidak beraturan.

Pencegahan dan pengendalian :

- a. Media untuk penyemaian menggunakan lapisan sub soil (1,5-2 m dibawah permukaan tanah), pupuk kandang matang yang halus dan pasir kali pada perbandingan 1 : 1 : 1. Campuran media ini dipasteurisasi selama 2 jam.
- b. Semaian yang terinfeksi penyakit harus dicabut dan dimusnahkan, media tanah yang terkontaminasi dibuang.
- c. Naungan persemaian secara bertahap dibuka agar matahari masuk dan tanaman menjadi lebih kuat.
- d. Penggunaan fungisida selektif dengan dosis batas terendah.



Gambar 2.2 Infeksi *Colletotrichum* spp. pada biji kadang-kadang berwarna hitam atau coklat dan biji tidak bernas (Duriat, 2007).



Gambar 2.3 Kematian semaian cabai/ tanaman muda karena penyakit rebah kecambah (Duriat, 2007).

2. Antraknosa

Patogen : *Colletotrichum* spp.

Gejala : Mati pucuk yang berlanjut ke bagian bawah. Daun, ranting dan cabang busuk kering berwarna coklat kehitam-hitaman. Pada batang acervuli cendawan terlihat berupa benjolan.

Pencegahan dan pengendalian :

- a. Pemupukan yang berimbang, yaitu Urea 150-200 kg, ZA 450-500 kg, TSP 100-150 kg, KCl 100-150 kg, dan pupuk organik 20-30 ton per hektar.

- b. *Intercropping* cabai di dataran tinggi dapat mengurangi serangan hama dan penyakit serta menaikkan hasil panen.
- c. Penggunaan mulsa plastik perak di dataran tinggi, dan jerami di dataran rendah mengurangi infestasi antraknos dan penyakit tanah, terutama pada musim hujan.
- d. Penyakit antraknosa *Colletotrichum* spp. dikendalikan dengan fungisida klorotalonil (Daconil ® 500 F, 2g/l) atau Propineb (Antracol ® 70 WP, 2g/l). Kedua fungisida ini digunakan secara bergantian.
- e. Untuk mengurangi penggunaan pestisida ($\pm 30\%$) dianjurkan untuk menggunakan nozel kipas yang butiran semprotannya berupa kabut dan merata.

Tabel 2.1 Penyakit Tanaman Cabai yang disebabkan oleh Cendawan

No.	Data Penyakit	Penyebab
1.	<ul style="list-style-type: none"> a. Penyakit Antraknosa b. Gejala <ul style="list-style-type: none"> 1. Mati pucuk. 2. Daun, ranting, dan cabang busuk kering berwarna coklat kehitam-hitaman. 3. Buah timbul bercak lunak berwarna hitam dan busuk lunak. 4. Pada batang <i>acervuli</i> cendawan terlihat berupa benjolan. c. Solusi <ul style="list-style-type: none"> 1. Pemupukan yang berimbang, yaitu Urea 150-200 kg, ZA 450-500 kg, TSP 100-150 kg, KCl 100-150 kg, dan pupuk organik 20-30 ton per hektar. 2. <i>Intercropping</i> cabai di dataran tinggi dapat mengurangi serangan hama dan penyakit serta menaikkan hasil panen. 3. Penggunaan mulsa plastik perak di dataran tinggi, dan jerami di dataran rendah mengurangi infestasi antraknos dan penyakit tanah, terutama di musim hujan. 	Penyebab penyakit ini adalah <i>Colletotrichum</i> spp.

Tabel 2.1 (lanjutan)

No.	Data Penyakit	Penyebab
	4. Untuk mengurangi penggunaan pestisida (\pm 30%) dianjurkan untuk menggunakan nozel kipas yang butiran semprotannya berupa kabut dan merata.	
2.	<p>a. Penyakit Bercak Daun Serkospora</p> <p>b. Gejala</p> <ol style="list-style-type: none"> 1. Bercak berbentuk bulat berwarna abu-abu tua dan warna coklat di pinggirannya. 2. Daun menjadi tua (menguning) sebelum waktunya. 3. Bercak daun berukuran sekitar 0,25 cm. 4. Sering terjadi sobekan dipusat daun <p>c. Solusi</p> <ol style="list-style-type: none"> 1. Pemupukan yang berimbang, yaitu Urea 150-200 kg, ZA 450-500 kg, TSP 100-150 kg, KCl 100-150 kg, dan pupuk organik 20-30 ton per hektar. 2. Penggunaan mulsa plastik perak di dataran tinggi, dan jerami di dataran rendah mengurangi infestasi bercak dan penyakit tanah, terutama di musim hujan. 3. Untuk bercak <i>Cercospora</i> dianjurkan menggunakan daun mindi (<i>Melia azederach</i>) pada konsentrasi 1: 20 (berat/volume). 4. Penyakit bercak daun <i>Cercospora capsici</i> dikendalikan dengan fungisida difenoconazole (Score ® 250 EC dengan konsentrasi 0,5 ml/l). Interval penyemprotan 7 hari. 5. Untuk mengurangi penggunaan pestisida (\pm 30%) dianjurkan untuk menggunakan nozel kipas yang butiran semprotannya berupa kabut dan merata. 	Penyakit ini disebabkan oleh <i>Cercospora capsici</i>
3.	<p>a. Penyakit Busuk daun Fitofthora.</p> <p>b. Gejala</p> <ol style="list-style-type: none"> 1. Busuk batang menjadi kering dan mengeras 2. Seluruh daun menjadi layu 3. Pada daun timbul bercak putih seperti tersiram air 4. Serangan pada akar mengakibatkan tanaman layu, mengering, dan mati. 	Penyakit ini disebabkan oleh <i>Phytophthora capsici</i>

Tabel 2.1 (lanjutan)

No	Data Penyakit	Penyebab
	<p>c. Solusi</p> <ol style="list-style-type: none"> 1. Pemupukan yang berimbang, yaitu Urea 150-200 kg, ZA 450-500 kg, TSP 100-150 kg, KCl 100-150 kg, dan pupuk organik 20-30 ton per hektar. 2. Penggunaan mulsa plastik perak di dataran tinggi, dan jerami di dataran rendah mengurangi infestasi penyakit, terutama di musim hujan. 3. Tanaman muda yang terinfeksi penyakit di lapangan dimusnahkan dan disulam dengan yang sehat. 4. Cendawan <i>Phytophthora capsici</i> dapat dikendalikan dengan fungisida sistemik Metalaksil-M 4% + Mancozeb 64% (Ridomil Gold MZ ® 4/64 WP) pada konsentrasi 3 g/l air, bergantian dengan fungisida kontak seperti klorotalonil (Daconil ® 500 F, 2g/l). Fungisida sistemik digunakan maksimal empat kali per musim. 5. Untuk mengurangi penggunaan pestisida (\pm 30%) dianjurkan untuk menggunakan nozel kipas yang butiran semprotannya berupa kabut dan merata. 	

2.2 Sistem Pakar

Sistem pakar merupakan sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem pakar biasanya digunakan untuk melakukan interpretasi dan analisa, diagnosa, dan membantu pengambilan keputusan (Oktaviani, 2012).

2.2.1 Definisi Sistem Pakar

Sistem pakar sebagai sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut (Hartati dan Iswanti, 2008).

Sistem Pakar (*expert system*) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli. Dengan sistem pakar ini, orang awampun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli, sistem pakar ini juga akan membantu aktivitasnya sebagai asisten yang sangat berpengalaman (Kusumadewi, 2003).

2.2.2 Konsep Dasar Sistem Pakar

Menurut Kusumadewi (2003) konsep dasar sistem pakar mengandung keahlian, ahli, pengalihan keahlian, inferensi, aturan, dan kemampuan menjelaskan. Keahlian adalah suatu kelebihan penguasaan pengetahuan dibidang tertentu yang diperoleh dari pelatihan, membaca atau pengalaman. Contoh bentuk pengetahuan yang termasuk keahlian adalah:

1. Fakta-fakta pada lingkup permasalahan tertentu.
2. Teori-teori pada lingkup permasalahan tertentu.

3. Prosedur-prosedur dan aturan-aturan bekeanan dengan lingkup permasalahan tertentu.
4. Strategi-strategi global untuk menyelesaikan masalah.
5. *Meta-knowledge* (pengetahuan tentang pengetahuan).

Bentuk-bentuk ini memungkinkan para ahli untuk dapat mengambil keputusan lebih cepat dan lebih baik daripada seseorang yang bukan ahli. Seorang ahli adalah seseorang yang mampu menjelaskan suatu tanggapan, mempelajari hal-hal baru seputar topik permasalahan (domain), menyusun kembali pengetahuan jika dipandang perlu, memecah aturan-aturan jika dibutuhkan, dan menentukan relevan tidaknya keahlian mereka. Pengalihan keahlian dari para ahli komputer untuk kemudian dialihkan lagi ke orang lain yang bukan ahli, merupakan tujuan utama dari sistem pakar. Proses ini membutuhkan 4 aktivitas yaitu tambahan pengetahuan (ke komputer), inferensi pengetahuan, dan pengalihan pengetahuan ke user. Pengetahuan yang disimpan di komputer disebut dengan nama basis pengetahuan.

Salah satu fitur yang harus dimiliki oleh sistem pakar adalah kemampuan untuk menalar. Jika keahlian-keahlian sudah tersimpan sebagai basis pengetahuan dan sudah tersedia program yang mampu mengakses basis data, maka komputer harus dapat diprogram untuk membuat inferensi. Proses inferensi ini dikemas dalam bentuk motor inferensi (*inference engine*). Sebagian besar sistem pakar komersial dibuat dalam bentuk *rule-based system*, yang mana pengetahuan disimpan dalam bentuk aturan-aturan. Aturan tersebut biasanya berbentuk IF-THEN.

2.2.3 Ciri-ciri Sistem Pakar

Ciri-ciri sistem pakar menurut Kusumadewi (2003), antara lain:

1. Memiliki fasilitas informasi yang handal.
2. Mudah dimodifikasi.
3. Dapat digunakan dalam berbagai jenis komputer.
4. Memiliki kemampuan untuk belajar beradaptasi.

2.2.4 Keuntungan Sistem Pakar

Menurut Dewi (2015), banyak manfaat yang dapat diambil dengan adanya sistem pakar, antara lain:

1. Membuat seseorang yang awam dapat bekerja seperti layaknya seorang pakar.
2. Menyimpan pengetahuan dan keahlian para pakar.
3. Meningkatkan output dan produktivitas.
4. Meningkatkan kualitas.
5. Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka).
6. Memiliki kemampuan untuk mengakses pengetahuan.
7. Memiliki kemampuan memecahkan masalah yang kompleks
8. Menghemat waktu dalam pengambilan keputusan.

2.2.5 Kelemahan Sistem Pakar

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

1. Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal.
2. Sulit dikembangkan. Hal ini tentu saja erat kaitannya dengan ketersediaan pakar dibidangnya.
3. Sistem pakar tidak 100% bernilai benar (Kusumadewi, 2003)

2.2.6 Struktur Sistem Pakar

Sistem pakar terdiri dari 2 bagian pokok, yaitu lingkungan pengembang (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan sebagai pembangun sistem pakar baik dari segi pembangun komponen maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh seseorang bukan ahli untuk berkonsultasi.

Komponen-komponen yang ada pada sistem pakar terdapat pada Gambar 2.4:

1. Antar Muka Pengguna

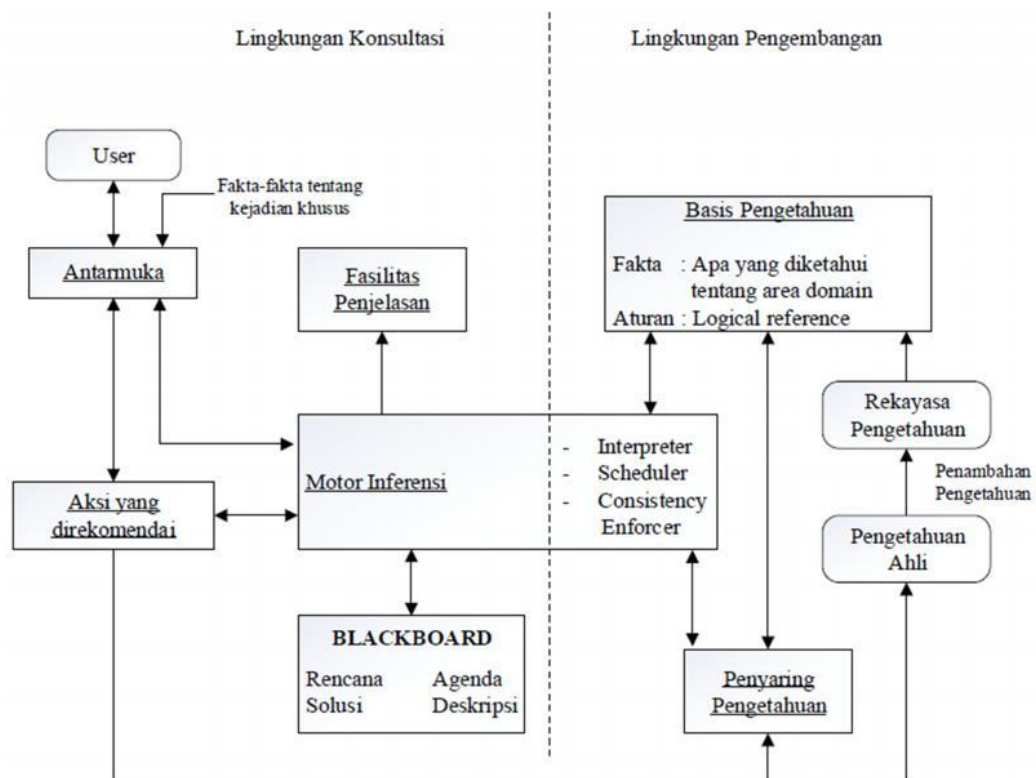
Antarmuka pengguna merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem.

Pada bagian ini terjadi dialog antara program dan pemakai, yang memungkinkan sistem pakar menerima instruksi dan informasi (*input*) dari pemakai, juga memberikan informasi (*output*) kepada pemakai.

2. Motor inferensi (*inference engine*).

Program yang berisi metodologi yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan dan blackboard serta digunakan untuk memformulasikan konklusi. Ada 3 elemen utama dalam motor inferensi, yaitu:

- *Interpreter*: mengeksekusi item-item agenda yang terpilih dengan menggunakan aturan-aturan dalam basis pengetahuan yang sesuai.
- *Scheduler*: akan mengontrol agenda.
- *Consistency enforcer*: akan berusaha memelihara kekonsistenan dalam merepresentasikan solusi yang bersifat darurat.



Gambar 2.4 Struktur Sistem Pakar (Kusumadewi, 2003).

3. Blackboard

Blackboard merupakan area kerja memori tempat pendeskripsian masalah yang diberikan oleh data input, digunakan juga untuk perekaman hipotesis dan keputusan sementara. Ada 3 tipe keputusan yang dapat direkam, yaitu:

- a. Rencana: bagaimana menghadapi masalah.
- b. Agenda: aksi-aksi yang potensial yang sedang menunggu untuk dieksekusi.

- c. Solusi: hipotesis kandidat dan arah tindakan alternatif yang telah dihasilkan sistem sampai dengan saat ini
- 4. Subsistem penjelasan. Digunakan untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif melalui pertanyaan:
 - a. Mengapa suatu pertanyaan ditanyakan oleh sistem pakar?
 - b. Bagaimana konklusi dicapai?
 - c. Mengapa ada alternative yang dibatalkan?
 - d. Rencana apa yang digunakan untuk mendapatkan solusi?
- 5. Subsistem penambahan pengetahuan.
 Bagian ini digunakan untuk memasukkan pengetahuan, mengonstruksi atau memperluas pengetahuan dalam basis pengetahuan. Pengetahuan itu bisa berasal dari ahli, buku, basis data, penelitian, dan gambar.
- 6. Sistem penyaring pengetahuan.
 Sistem ini digunakan untuk mengevaluasi kinerja sistem pakar itu sendiri untuk melihat apakah pengetahuan-pengetahuan yang ada masih cocok untuk digunakan di masa datang.

2.2.7 Basis Pengetahuan

Basis pengetahuan berisi pengetahuan-pengetahuan dalam menyelesaikan masalah, tentu di dalam domain tertentu. Ada dua bentuk pendekatan basis pengetahuan yang umum digunakan, yaitu:

1. Penalaran berbasis aturan (Pengetahuan direpresentasikan dengan menggunakan aturan berbentuk **:IF-THEN**. Bentuk ini digunakan apabila kita memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu dan si pakar dapat

menyelesaikan masalah tersebut secara berurutan. Disamping itu juga digunakan apabila dibutuhkan penjelasan tentang langkah-langkah pencapaian solusi.

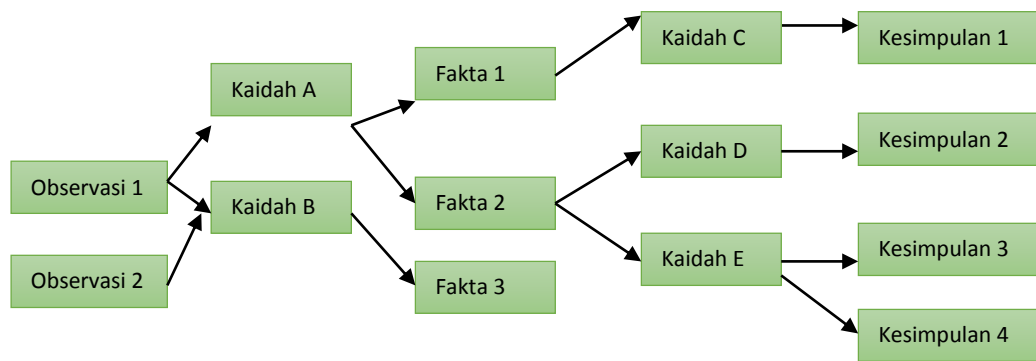
2. Penalaran berbasis kasus (Basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang sekarang. Bentuk ini akan digunakan apabila *user* menginginkan untuk tahu lebih banyak lagi pada kasus-kasus yang hampir sama. Selain itu, bentuk ini juga digunakan apabila kita telah memiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan (Kusumadewi, 2003).

2.2.8 Teknik Inferensi

Mesin inferensi adalah bagian yang mengandung mekanisme fungsi berpikir dan pola-pola penalaran sistem yang digunakan seorang pakar. Mekanisme ini akan menganalisa suatu masalah tertentu dan selanjutnya mencari kesimpulan yang terbaik (Siswanto, 2010).

2.2.9 Forward Chaining

Menurut Arhami (2005), *forward chaining* merupakan proses peruntukan yang dimulai dengan menampilkan kumpulan data atau fakta yang meyakinkan menuju konklusi akhir. Dalam pendekatan ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Sehingga metode ini juga sering disebut “*data driven*” yang dimulai dari premis-premis atau informasi masukan (*if*) dahulu kemudian menuju konklusi atau kesimpulan (*then*).



Gambar 2.5. Proses *Forward Chaining* (Arhami, 2005).

2.2.9.1 Metode *Forward Chaining*

Menurut Sutojo (2011) metode *forward chaining* merupakan teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari rules IF-THEN. Bila ada fakta yang cocok dengan bagian IF, maka rule tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru pencocokan berhenti bila tidak ada lagi rule yang bisa dieksekusi.

Contoh:

Misalkan diketahui sistem pakar menggunakan 5 buah rule berikut.

R1 : IF (putih AND hijau) THEN hitam

R2 : IF (coklat AND biru AND ungu) THEN putih

R3 : IF merah THEN coklat

R4 : IF kuning THEN pink

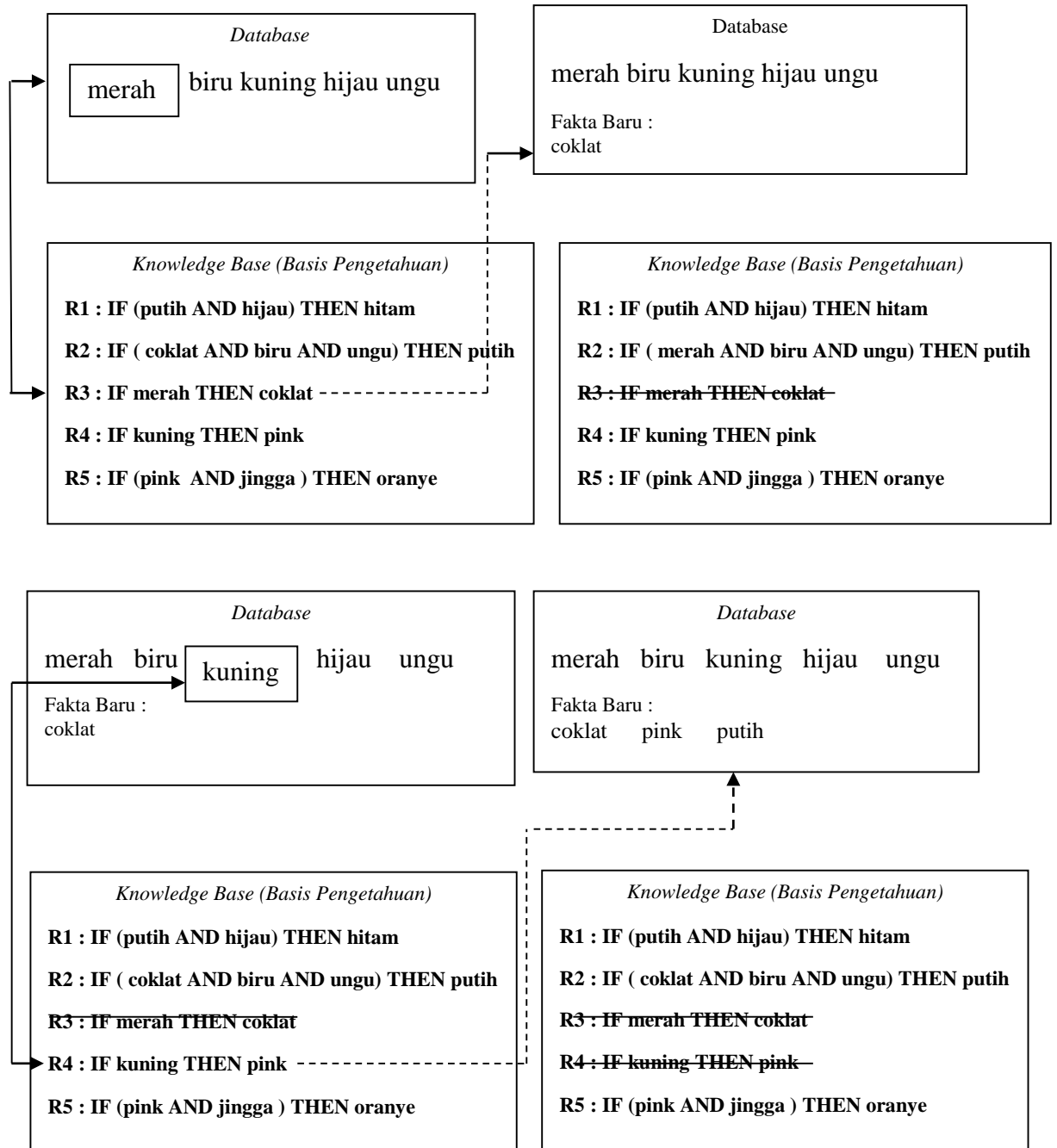
R5 : IF (pink AND jingga) THEN oranye

Fakta-fakta : merah, biru, kuning, hijau, dan ungu bernilai benar

Goal : menentukan apakah hitam bernilai benar atau salah.

Iterasi ke-1

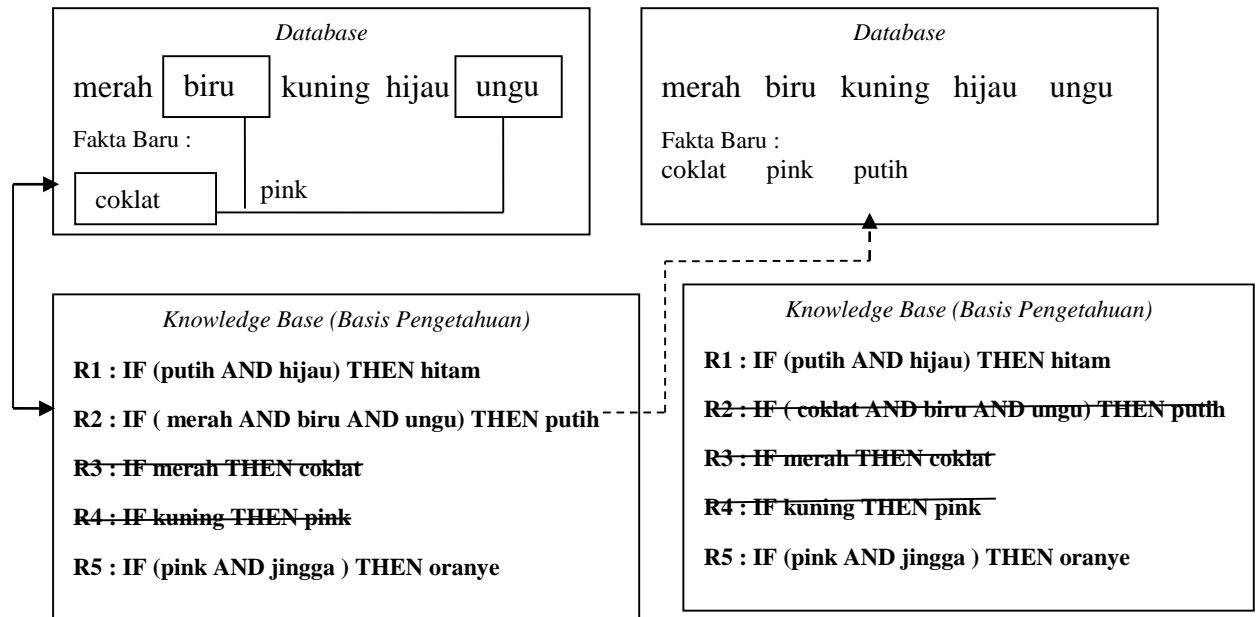
Iterasi ke-1 dapat dilihat pada Gambar 2.6.



Gambar 2.6. Iterasi ke-1

Iterasi ke-2

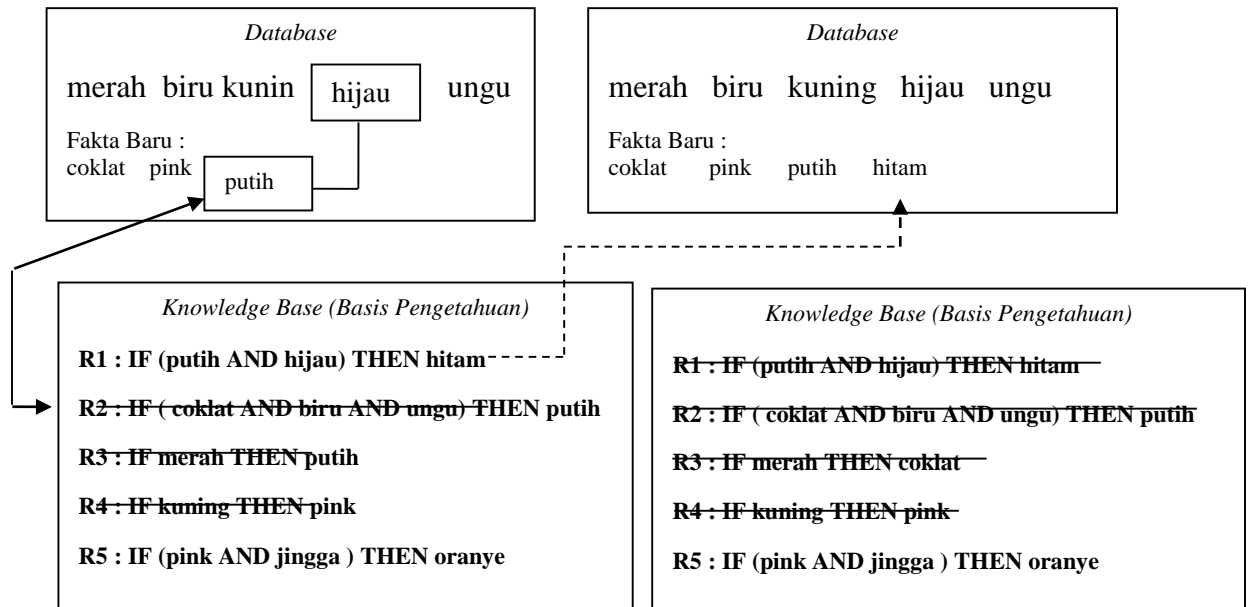
Iterasi ke-2 dapat dilihat pada Gambar 2.7.



Gambar 2.7. Iterasi ke-2

Iterasi ke-3

Iterasi ke-3 dapat dilihat pada Gambar 2.8.



Gambar 2.8. Iterasi ke-3

Sampai di sini proses dihentikan karena sudah tidak ada lagi rule yang bisa dieksekusi. Hasil pencarian adalah hitam bernilai benar (lihat database di bagian fakta baru).

2.2.9.2 Contoh kasus *Forward Chaining*:

Diketahui sistem pakar dengan aturan-aturan sebagai berikut:

R1 : IF suku bunga turun THEN harga obligasi naik

R2 : IF suku bunga naik THEN harga obligasi turun

R3 : IF suku bunga tidak berubah THEN harga obligasi tidak berubah

R4 : IF dolar naik THEN suku bunga turun

R5 : IF dolar turun THEN suku bunga naik

R6 : If harga obligasi turun THEN beli obligasi

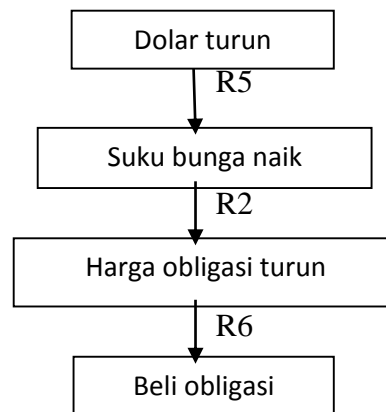
Apabila diketahui bahwa dolar turun, maka untuk memutuskan apakah akan membeli obligasi atau tidak dapat?

Untuk dapat menyelesaikan aturan-aturan yang sudah disediakan, maka langkah-langkah yang harus dilakukan adalah sebagai berikut.

Dari fakta dolar turun, berdasarkan aturan-5, diperoleh konklusi suku bunga naik.

Dari aturan-2, suku bunga naik menyebabkan harga obligasi turun. Dengan menggunakan aturan R-6, jika harga obligasi turun, maka kesimpulan yang diambil adalah membeli obligasi (Kusumadewi, 2003).

Solusi dengan menggunakan *forward chaining* dapat dilihat pada Gambar 2.9.



Gambar 2.9 Penyelesaian dengan *Forward Chaining* (Kusumadewi, 2003)

2.3 UML (*Unified Modelling Language*)

Menurut Nugroho (2010), UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

UML dideskripsikan oleh beberapa diagram, yaitu:

1. *Use Case* Diagram

Use case Diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case* diagram lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case* diagram merepresentasikan sebuah interaksi antara aktor dengan sistem yang akan dikembangkan (Fowler, 2004).

Komponen-komponen dalam *use case* diagram (Fowler, 2004):

a. Aktor

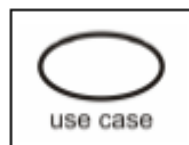
Pada dasarnya aktor bukanlah bagian dari *use case* diagram, namun untuk dapat terciptanya suatu *use case* diagram diperlukan aktor, dimana aktor tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem yang dibuat. Sebuah aktor mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima dan memberi informasi pada sistem. Aktor hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Aktor digambarkan dengan *stick pan* seperti yang terdapat pada Gambar 2.10.



Gambar 2.10. Contoh Aktor (Fowler, 2004).

b. *Use Case*

Gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti kegunaan sistem yang akan dibangun. Bentuk *use case* dapat terlihat pada Gambar 2.11.



Gambar 2.11. *Use Case* (Fowler, 2004).

Ada beberapa relasi yang terdapat pada *use case* diagram:

1. *Association*, menghubungkan link antar element.
2. *Generalization*, disebut juga pewarisan (*inheritance*), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
3. *Dependency*, sebuah element bergantung dalam beberapa cara ke element lainnya.
4. *Aggregation*, bentuk *association* dimana sebuah elemen berisi elemen lainnya.

Tipe relasi yang mungkin terjadi pada *use case* diagram:

1. <<*include*>>, yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
2. <<*extends*>>, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan peringatan.
3. <<*communicates*>>, merupakan pilihan selama asosiasi hanya tipe relationship yang dibolehkan antara aktor dan *use case*.

2. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat digunakan untuk aktifitas lainnya (Fowler, 2004). Berikut ini adalah tabel Notasi Activity Diagram yang diilustrasikan pada Tabel 2.2.

Tabel 2.2 Notasi Activity Diagram (Fowler, 2004).



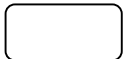
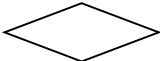


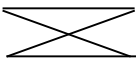
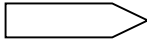
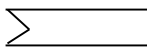

Simbol	Keterangan
	Titik Awal
	Titik Akhir
	<i>Activity</i>
	Pilihan untuk mengambil keputusan
	<i>Fork</i> ; Digunakan untuk menunjukan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Rake</i> ; Menunjukan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (<i>Flow Final</i>)

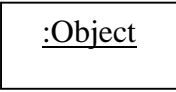



Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan

proses. *Activity* diagram juga digunakan untuk menggambarkan interaksi antara beberapa *use case* (Fowler, 2004).


3. *Sequence* Diagram

Sequence diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem (Fowler, 2004). Berikut ini adalah Notasi *Sequence* Diagram yang disajikan pada Tabel 2.3.

Tabel 2.3 Notasi *Sequence* Diagram (Fowler, 2004).

Simbol	Nama	Keterangan
	<i>Object</i>	<i>Object</i> merupakan <i>instance</i> dari sebuah class dan dituliskan secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama obyek didalamnya yang diawali dengan sebuah titik koma
	<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom. Simbol <i>actor</i> sama dengan symbol pada <i>actor Use Case</i> Diagram.
	<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertical yang ditarik dari sebuah obyek.
	<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . <i>Activation</i> mengindikasikan sebuah obyek yang akan melakukan sebuah aksi.

Tabel 2.3 Notasi Sequence Diagram (lanjutan)

Simbol	Nama	Keterangan
	<i>Message</i>	<p>Message, digambarkan dengan anak panah <i>horizontal</i> antara <i>Activation</i>. <i>Message</i> mengindikasikan komunikasi antara <i>object-object</i>.</p>

2.4 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses menjalankan dan mengevaluasi sebuah perangkat lunak secara manual maupun otomatis untuk menguji apakah perangkat lunak sudah memenuhi persyaratan atau belum. Pengertian lain, pengujian adalah aktivitas untuk menemukan dan menentukan perbedaan antara hasil yang diharapkan dengan hasil sebenarnya. Pendekatan *Black-Box* merupakan pendekatan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan (Jiang, 2012).

Kasus uji ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya. Teknik pengujian ini berfokus pada domain informasi dari perangkat lunak, yaitu melakukan kasus uji dengan mempartisi domain *input* dan *output* program. Metode *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian ini berusaha menemukan kesalahan dalam kategori fungsi-fungsi yang tidak benar atau hilang, kesalahan

interface, kesalahan dalam struktur data atau akses basis data eksternal, kesalahan kinerja, dan inisialisasi dan kesalahan terminal (Pressman, 2001). Teknik pengujian yang digunakan dalam penelitian ini terbagi atas pengujian fungsional dengan menggunakan metode *Black Box* yaitu *Equivalence Partitioning*.

2.4.1 Equivalence Partitioning

Equivalence Partitioning (EP) merupakan metode *Black Box testing* yang membagi domain masukan dari program kedalam kelas-kelas sehingga *test case* dapat diperoleh. *Equivalence Partitioning* berusaha untuk mendefinisikan kasus uji yang menemukan sejumlah jenis kesalahan, dan mengurangi jumlah kasus uji yang harus dibuat. Kasus uji yang didesain untuk *Equivalence Partitioning* berdasarkan pada evaluasi dari kelas ekuivalensi untuk kondisi masukan yang menggambarkan kumpulan keadaan yang valid atau tidak. Kondisi masukan dapat berupa spesifikasi nilai numerik, kisaran nilai, kumpulan nilai yang berhubungan atau kondisi *Boolean* (Pressman, 2001).

2.4.2 Probabilitas Klasik

Menurut Arhami (2005) probabilitas merupakan suatu cara kuantitatif yang berhubungan dengan ketidakpastian yang telah ada. Probabilitas klasik disebut juga *a priori probability* karena berhubungan dengan suatu permainan (*games*) atau sistem. Seperti yang telah disebutkan sebelumnya, istilah *a priori* berarti “sebelum” (Arhami, 2005). Probabilitas ini dianggap sebagai suatu jenis permainan seperti pelemparan dadu, permainan kartu, dan pelemparan koin.

Rumus umum untuk probabilitas klasik di definisikan sebagai peluang $P(A)$ dengan n adalah banyaknya kejadian, $n(A)$ merupakan banyaknya hasil mendapatkan A . Frekuensi relative terjadinya A adalah $\frac{n(A)}{n}$ maka (Arhami, 2005) :

$$P(A) = \frac{n(A)}{n} \dots\dots\dots \text{(Persamaan 1)}$$

Probabilitas klasik ini digunakan untuk mendapatkan peluang kemungkinan penyakit, sehingga untuk menghitung persentase penyakit adalah :

$$\text{Persentase (A)} = P(A) \times 100\% \dots\dots\dots \text{(Persamaan 2)}$$

Tabel persentase penyakit disajikan pada Tabel 2.4.

Tabel 2.4. Tabel Kemungkinan Penyakit

Kondisi	Persentase
Pasti Tidak	<10%
Tidak Tahu	10-19%
Hampir Mungkin	20 – 39%
Mungkin	40 – 59%
Kemungkinan Besar	60 - 79%
Hampir Pasti	80 – 99%
Pasti	100%

2.4.3 Skala Likert

Menurut Likert dalam buku Azwar S (2011, p. 139), sikap dapat diukur dengan metode rating yang dijumlahkan (Method of Summated Ratings). Metode ini merupakan metode penskalaan pernyataan sikap yang menggunakan distribusi respons sebagai dasar penentuan nilai skalanya. Nilai skala setiap pernyataan tidak ditentukan oleh derajat *favourable* nya masing-masing akan tetapi ditentukan oleh distribusi *respons* setuju dan tidak setuju dari sekelompok responden yang bertindak sebagai kelompok uji coba (*pilot study*) (Azwar, 2011).

Skala Likert, yaitu skala yang berisi lima tingkat preferensi jawaban dengan pilihan sebagai berikut: 1 = tidak baik; 2 = kurang baik; 3 = cukup baik; 4 = baik; 5 = sangat

baik. Skala *Likert* berisi lima tingkat preferensi jawaban dengan pilihan pada Tabel 2.5.

Tabel 2.5 Tingkat Preferensi Jawaban Skala *Likert*

Angka Preferensi	Keterangan Preferensi
1	Tidak Baik
2	Kurang Baik
3	Cukup Baik
4	Baik
5	Sangat Baik

Rumus penentuan presentase penilaian berdasarkan kriteria pada Skala *Likert* adalah menggunakan rumus aritmatika *mean* berikut (Aristoteles dkk, 2016):

$$P = \frac{Xi}{n(N)} (100\%)$$

Keterangan:

P = Presentase pernyataan

Xi = Nilai kualitatif total

n = Jumlah responden

N = Nilai kategori pernyataan terbaik

Selanjutnya, penentuan interval per kategori dihitung menggunakan rumus berikut:

$$I = \frac{100\%}{K}$$

Keterangan:

I = Interval

K = Kategori interval

III. METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam dan Jurusan Agroteknologi bidang Hama dan Penyakit Tanaman Fakultas Pertanian Universitas Lampung. Waktu penelitian dilaksanakan pada periode semester genap Tahun Ajaran 2016/2017.

3.2 Alat Pendukung

Alat pendukung yang digunakan dalam penelitian ini adalah sebagai berikut:

a. Perangkat Keras

1. *System Manufacturer* : ASUSTeK COMPUTER INC.
2. *System Model*: S451LN
3. *Processor* : Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz (4 CPUs), ~2.3GH
4. *Memory* : 8192 MB RAM dan hardisk : 1 TB

b. Perangkat Lunak

1. Sistem operasi Windows 8 Home Premium 64-bit, merupakan sistem operasi yang terpasang pada laptop yang digunakan dalam pengembangan sistem.

2. Android Studio, merupakan sebuah *Integrated Development Environment* (IDE) untuk pengembangan aplikasi di platform Android.
3. *Java Development Kit* (JDK), merupakan *tools* pengembang bahasa pemrograman java.
4. *Photoshop CS5*, digunakan untuk pembuatan tampilan *user interface* dan *editing* atribut gambar.

3.3 Tahapan Penelitian

Berikut merupakan tahapan penelitian yang dijelaskan pada Gambar 3.1. Tahapan penelitian merupakan langkah-langkah yang akan dilakukan peneliti dalam melakukan penelitian. Langkah-langkah dalam penelitian ini meliputi perumusan masalah, pengumpulan data, perancangan sistem, pengembangan sistem, dan pengujian sistem. Tahapan penelitian ini dapat dilihat pada Gambar 3.1.

3.3.1 Tahap Perumusan Masalah

Tahapan ini merupakan proses merumuskan dan membatasi masalah yang akan diteliti. Perumusan dan pembatasan masalah diperlukan agar dapat lebih mengarahkan peneliti dalam membuat sistem sehingga penelitian yang dikerjakan tidak keluar dari batasan yang telah ditetapkan sebelumnya.

3.3.2 Tahap Pengumpulan Data

Tahapan pengumpulan data dilakukan dengan menggunakan dua metode yaitu melalui studi pustaka dan wawancara.

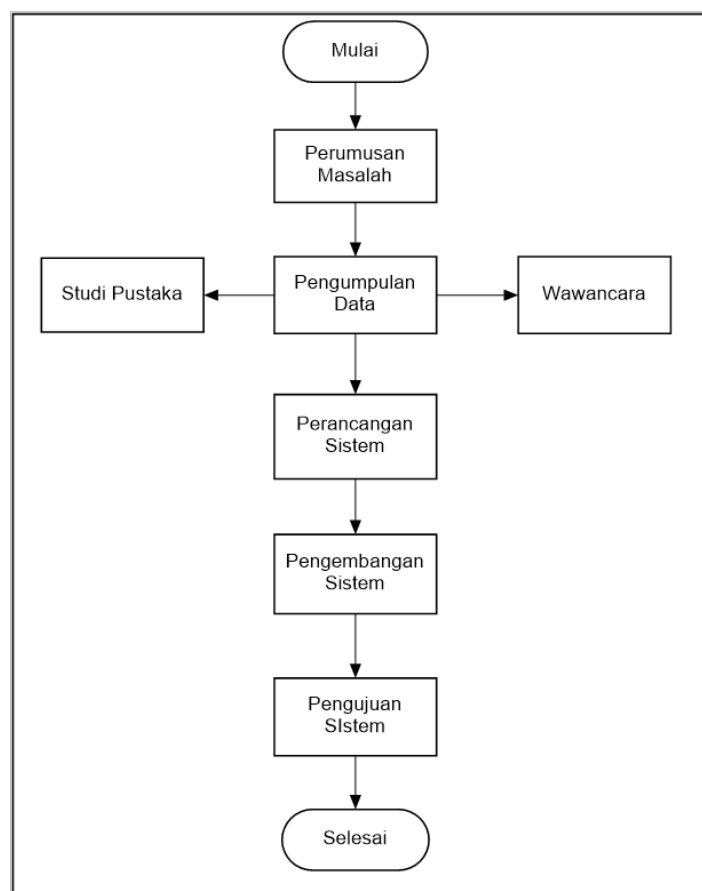
a. Studi Pustaka

Pada tahap ini data dikumpulkan melalui berbagai literatur seperti pada buku, jurnal, ataupun dokumen yang berkaitan dengan tema penelitian.

b. Wawancara

Pada metode ini dilakukan proses *interview* atau wawancara kepada para ahli/pakar.

Hal ini bertujuan untuk mendapatkan data yang tidak ditemukan pada metode studi pustaka.



Gambar 3.1. Tahapan Penelitian

3.3.3 Perancangan Sistem

Perancangan sistem adalah tahap setelah analisis dari siklus pengembangan sistem.

Perancangan sistem di sini berupa penggambaran, perencanaan, dan pembuatan

sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Perancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan. Tahap ini termasuk mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah dilakukan instalasi akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem.

3.3.3.1 Perancangan UML (*Unified Modelling Language*)

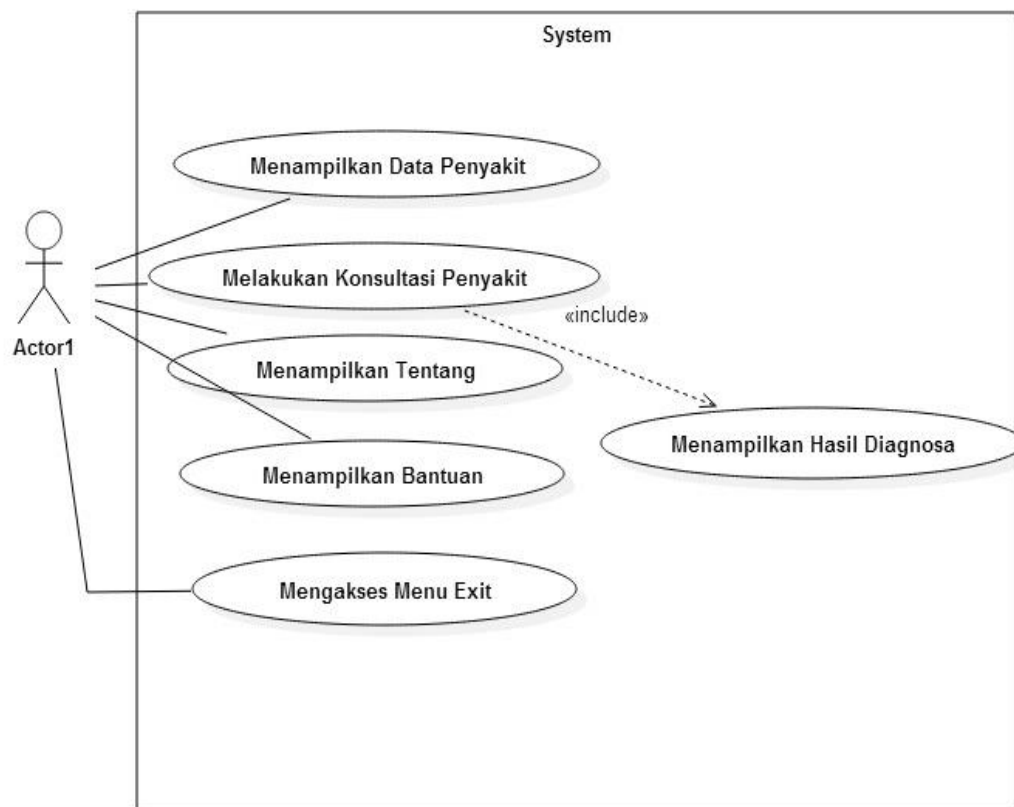
Pemodelan (modeling) adalah tahap merancang perangkat lunak sebelum melakukan tahap pembuatan program (koding). Pada penelitian ini, perancangan sistem dilakukan dengan memodelkan permasalahan dalam bentuk diagram-diagram UML sebagai berikut:

1. *Use Case* Diagram

Dalam membangun aplikasi sistem pakar berbasis android ini telah dirancang model *use case diagram* untuk menginterpretasikan fungsi *interface* dari sisi pengguna (*user*). Desain *use case diagram* sistem dapat dilihat pada Gambar 3.2.

2. *Activity* Diagram

Activity diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam satu operasi sehingga dapat juga untuk aktivitas lainnya.



Gambar 3.2 Use Case Diagram

Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Pada aplikasi Sistem Pakar Diagnosis Penyakit Cabai terdapat 5 (lima) *activity* diagram yaitu sebagai berikut:

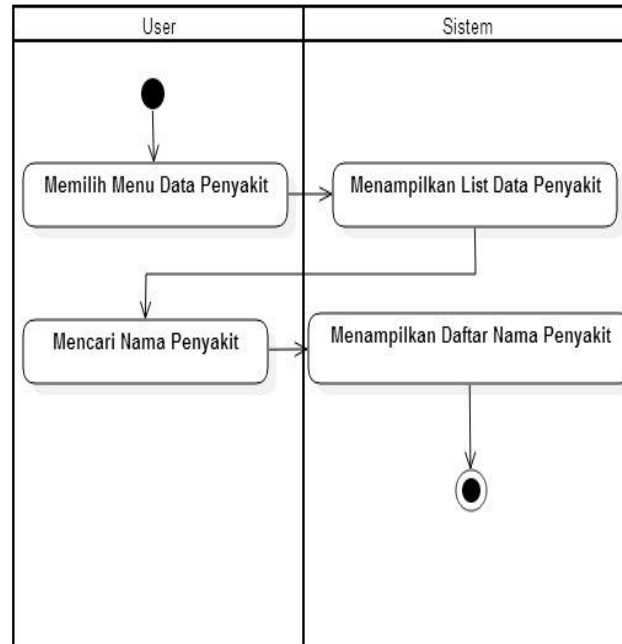
a. *Activity* Diagram Data Penyakit

Activity Diagram Data Penyakit pada Gambar 3.3 terdiri dari *user* dan sistem.

Proses pada Gambar 3.3 dijelaskan sebagai berikut:

1. *User* mengawali aktivitas, memilih menu data penyakit dan sistem akan menampilkan *list* penyakit.
2. Kemudian *user* mencari nama penyakit yang dibutuhkan
3. Selanjutnya sistem menampilkan nama penyakit yang dicari.

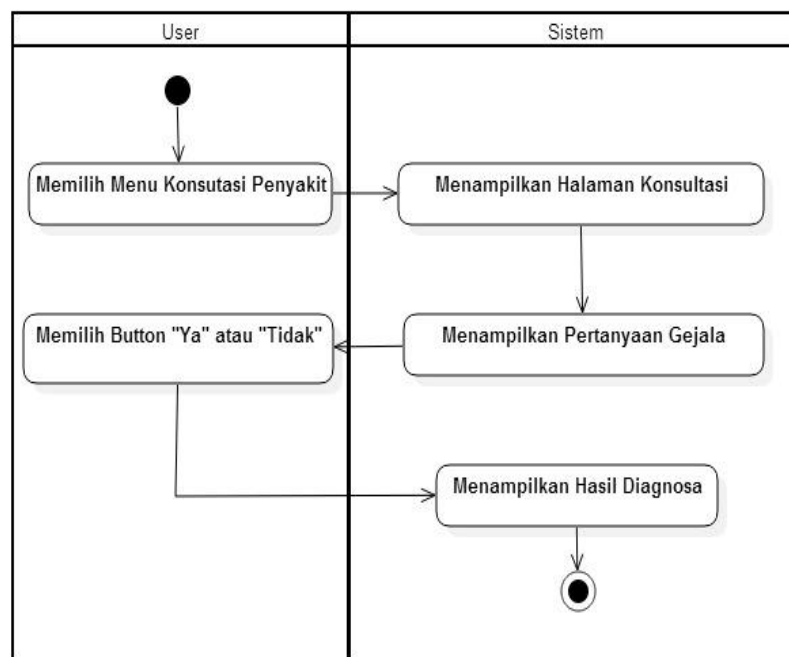
Activity diagram Data Penyakit disajikan pada Gambar 3.3.



Gambar 3.3 Activity Diagram Data Penyakit

b. Activity Diagram Konsultasi Penyakit

Activity diagram Konsultasi Penyakit disajikan pada Gambar 3.4.



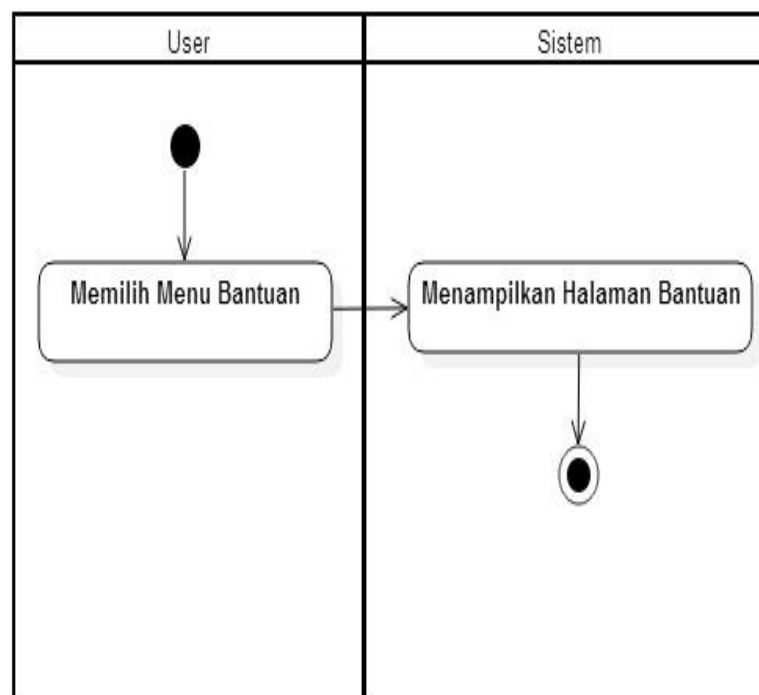
Gambar 3.4 Activity Diagram Konsultasi Penyakit

Activity Diagram pada Gambar 3.4 terdiri dari *user* dan sistem. Proses pada Gambar 3.4 dijelaskan sebagai berikut:

1. *User* memilih menu konsultasi dan sistem akan menampilkan halaman konsultasi.
2. Kemudian sistem menampilkan pertanyaan gejala.
3. Selanjutnya *user* menjawab pertanyaan dengan memilih button “Ya” atau Tidak”
4. Kemudian sistem menampilkan halaman diagnosis.

c. *Activity Diagram* Bantuan

Activity diagram Bantuan disajikan pada Gambar 3.5.



Gambar 3.5 *Activity Diagram* Bantuan

Activity Diagram pada Gambar 3.5 terdiri dari *user* dan sistem. Proses pada Gambar 3.5 dijelaskan sebagai berikut:

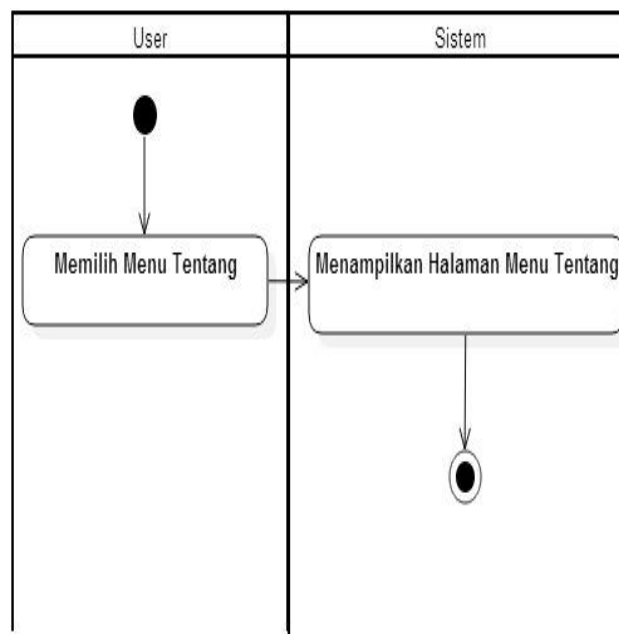
1. *User* mengawali aktivitas, memilih menu bantuan .
2. Kemudian sistem akan menampilkan halaman bantuan.

d. *Activity Diagram Tentang*

Activity Diagram Tentang pada Gambar 3.6 terdiri dari *user* dan sistem. Proses pada Gambar 3.6 dijelaskan sebagai berikut:

1. *User* mengawali aktivitas, memili menu tentang.
2. Kemudian sistem akan menampilkan halaman tentang.

Activity diagram Tentang disajikan pada Gambar 3.6.



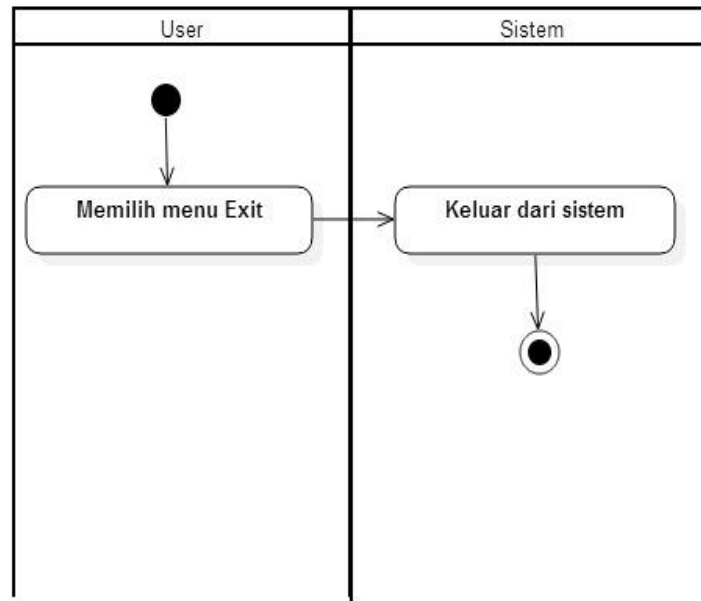
Gambar 3.6 *Activity Diagram Tentang*

e. *Activity Diagram Exit*

Activity Diagram Exit pada Gambar 3.7 terdiri dari *user* dan sistem. Proses pada Gambar 3.7 dijelaskan sebagai berikut:

1. *User* mengawali aktivitas, memilih menu Exit.
2. Kemudian sistem akan keluar dari sistem yang sedang dijalankan.

Activity diagram Exit disajikan pada Gambar 3.7.



Gambar 3.7 Activity Diagram Exit

3. Sequence Diagram

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. Pada aplikasi Sistem Pakar Diagnosis Penyakit Tanaman Cabai terdapat 5 (empat) *sequence* diagram yaitu sebagai berikut:

a. Sequence Diagram Data Penyakit

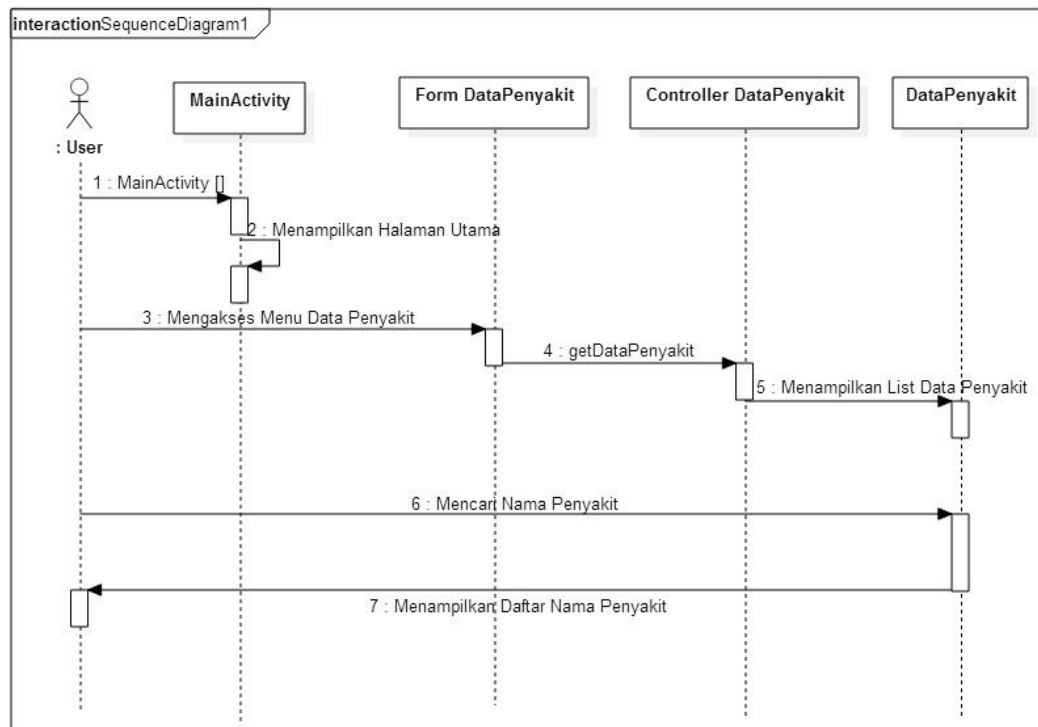
Sequence Diagram Data Penyakit pada Gambar 3.8 terdiri dari 1 *user* dan 4 objek yaitu *main activity*, *form* data penyakit, *controller* data penyakit, dan data penyakit. Proses Gambar 3.8 dijelaskan sebagai berikut:

1. *User* masuk ke halaman utama (*main activity*). Objek melakukan *self stimulus* dengan menampilkan halaman utama sistem.
2. *User* mengakses menu data penyakit dan objek *controller* data penyakit

mendapatkan halaman yang dituju

3. Objek *controller* menampilkan *list* penyakit.
4. *User* mencari nama penyakit yang diinginkan.
5. Kemudian objek data penyakit menampilkan daftar nama penyakit.

Sequence diagram Data Penyakit disajikan pada Gambar 3.8.



Gambar 3.8 *Sequence* Diagram Data Penyakit

b. *Sequence* Diagram Konsultasi Penyakit

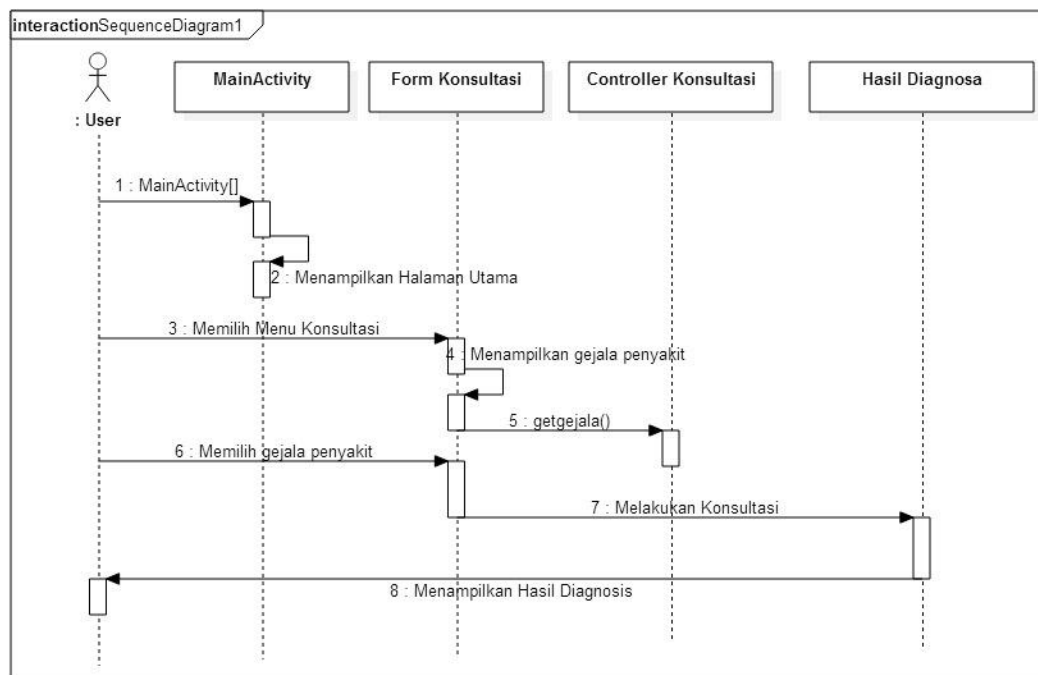
Sequence Diagram pada Gambar 3.9 terdiri dari 1 *user* dan 4 objek yaitu *main activity*, *form* konsultasi, *controller* konsultasi, dan hasil diagnosa. Proses Gambar 3.9 dijelaskan sebagai berikut:

1. *User* masuk ke halaman utama (*main activity*). *User* memilih menu konsultasi
2. *Form* konsultasi menampilkan gejala penyakit dan objek *controller* konsultasi

mendapatkan gejala

3. *User* memilih gejala penyakit yang sesuai.
4. Objek *controller* konsultasi melakukan proses diagnosis.
5. Kemudian objek hasil diagnosis menampilkan hasil diagnosis penyakit.

Sequence diagram Konsultasi Penyakit disajikan pada Gambar 3.9.



Gambar 3.9 *Sequence* Diagram Konsultasi Penyakit

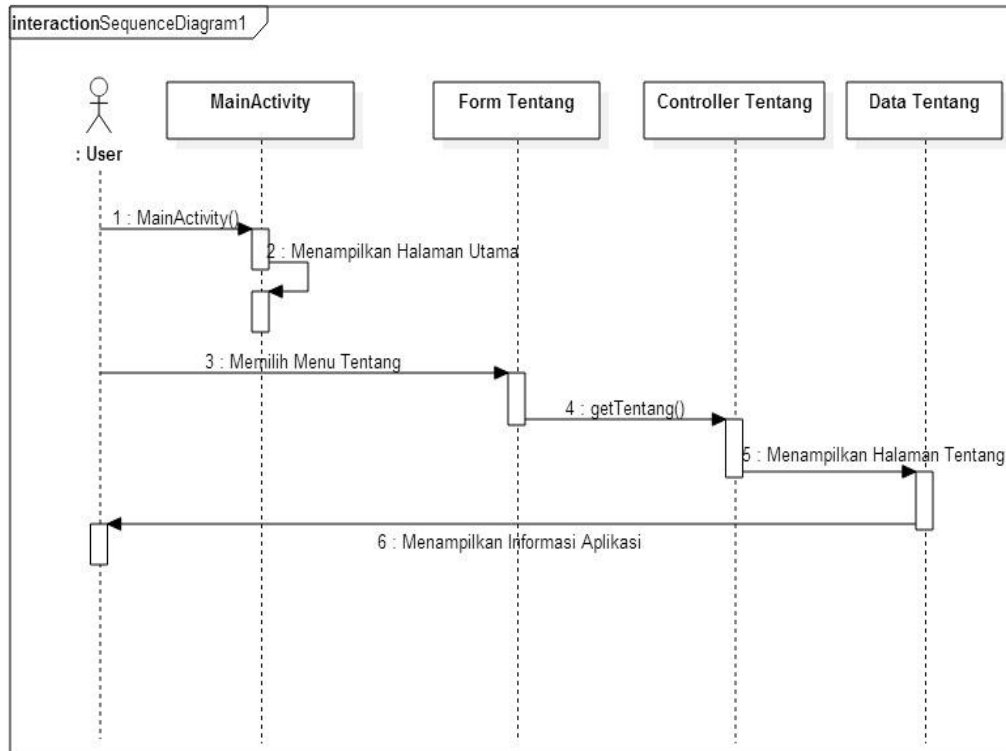
c. *Sequence* Diagram Tentang

Sequence Diagram Tentang pada Gambar 3.10 terdiri dari 1 *user* dan 4 objek yaitu *main activity*, *form* tentang, *controller* tentang, dan *data* tentang. Proses pada Gambar 3.10 dijelaskan sebagai berikut:

1. *User* masuk ke halaman utama (*main activity*). Objek melakukan *self stimulus* dengan menampilkan halaman utama sistem.
2. *User* memilih menu tentang dan objek *controller* tentang mendapatkan halaman yang dituju dan kemudian menampilkan halaman tentang.

3. Kemudian objek data tentang menampilkan informasi aplikasi kepada *user*.

Sequence diagram Tentang disajikan pada Gambar 3.10.



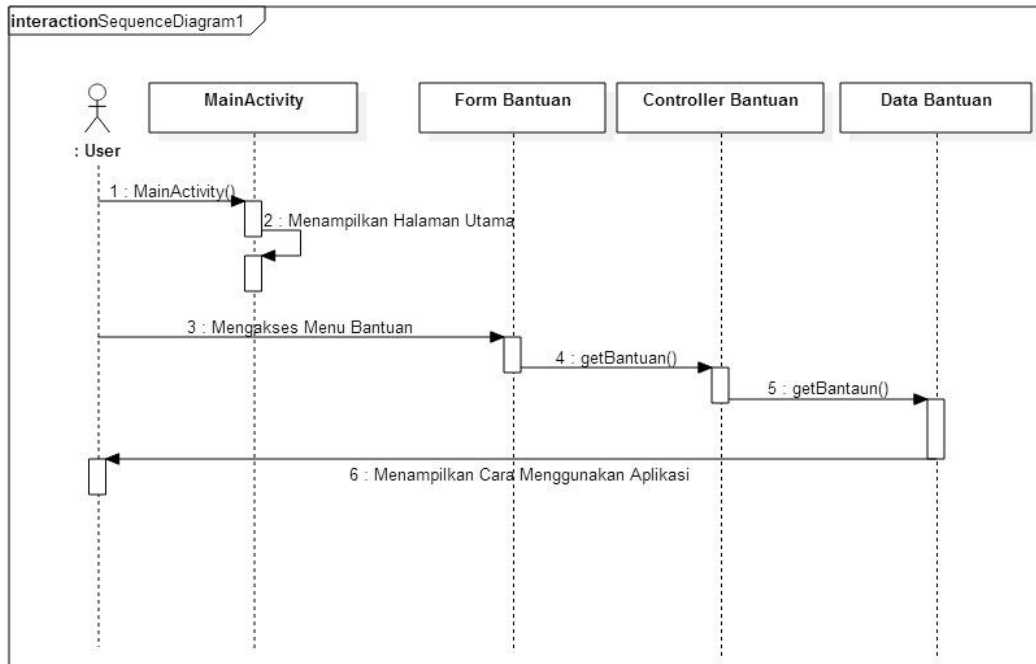
Gambar 3.10 *Sequence* diagram Tentang

d. *Sequence* Diagram Bantuan

Sequence Diagram Bantuan pada Gambar 3.11 terdiri dari 1 *user* dan 4 objek yaitu *main activity*, *form* bantuan, *controller* bantuan, dan data bantuan. Proses pada Gambar 3.11 dijelaskan sebagai berikut.

1. *User* masuk ke halaman utama (*main activity*). Objek melakukan *self stimulus* dengan menampilkan halaman utama sistem.
2. *User* mengakses menu bantuan dan objek *controller* bantuan mendapatkan halaman yang dituju.
3. Objek data bantuan menampilkan halaman bantuan.

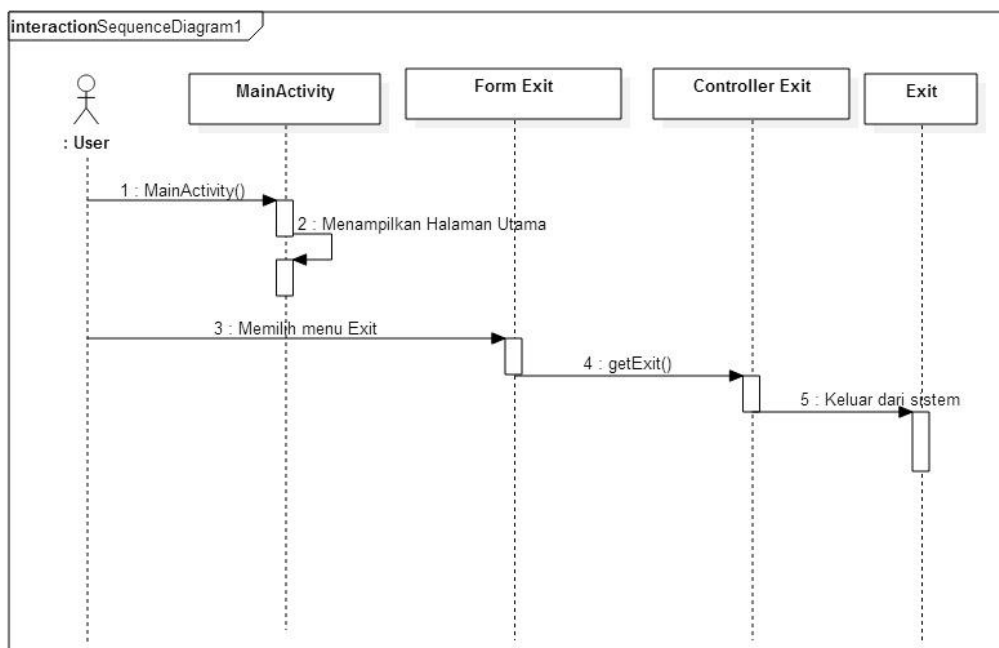
Sequence diagram Bantuan disajikan pada Gambar 3.11



Gambar 3.11 Sequence Diagram Bantuan

e. Sequence Diagram Exit

Sequence diagram Exit di sajikan pada gambar 3.12.



Gambar 3.12 Sequence Diagram Exit

Sequence Diagram pada Gambar 3.12 terdiri dari 1 *user* dan 4 objek yaitu *main activity*, *form exit*, *controller exit*, dan *exit*. Proses pada Gambar 3.12 dijelaskan sebagai berikut.

1. *User* masuk ke halaman utama (*main activity*).
2. *User* mengakses menu *exit* dan objek *controller exit* mendapatkan halaman yang dituju.
3. Kemudian objek *exit* akan keluar dari sistem.

3.3.3.2 Perancangan Antarmuka

Perancangan antarmuka merupakan proses penggambaran bagaimana sebuah tampilan (interface) sistem dibentuk. Antarmuka sistem dibuat sebagai penghubung antara sistem pakar dengan pengguna (*user*). Pada rancangan antarmuka sistem pakar ini terdapat beberapa halaman yang dapat diakses oleh pengguna (*user*), seperti berikut:

1. Rancangan Halaman Utama

Rancangan halaman utama aplikasi dapat dilihat pada Gambar 3.13.

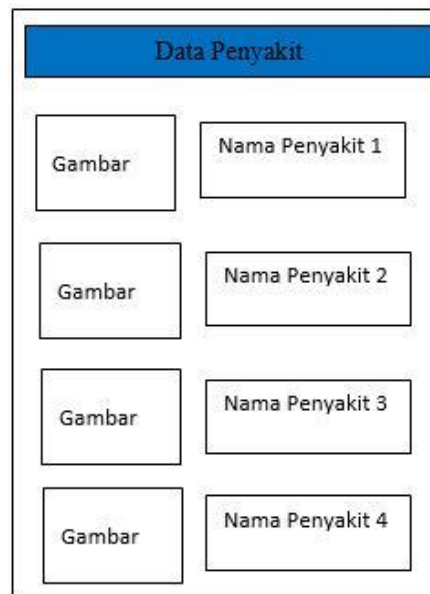


Gambar 3.13 Rancangan Halaman Menu Utama

Pada halaman utama terdapat dua pilihan bahasa yang dapat digunakan yaitu bahasa Indonesia dan bahasa Inggris. Halaman utama berisikan menu-menu pilihan yang dapat digunakan oleh *user*. Menu yang terdapat pada halaman utama antara lain: menu Data Penyakit, menu Konsultasi Gejala, menu Bantuan, dan menu *Exit*.

2. Halaman Data Penyakit

Rancangan halaman menu Data Penyakit dapat dilihat ketika *user* memilih menu ini, *user* dapat langsung melihat daftar nama penyakit tanaman cabai. Rancangan halaman Data Penyakit dapat dilihat pada Gambar 3.14.



Data Penyakit	
Gambar	Nama Penyakit 1
Gambar	Nama Penyakit 2
Gambar	Nama Penyakit 3
Gambar	Nama Penyakit 4

Gambar 3.14 Halaman Menu Data Penyakit

3. Halaman Menu Konsultasi Gejala

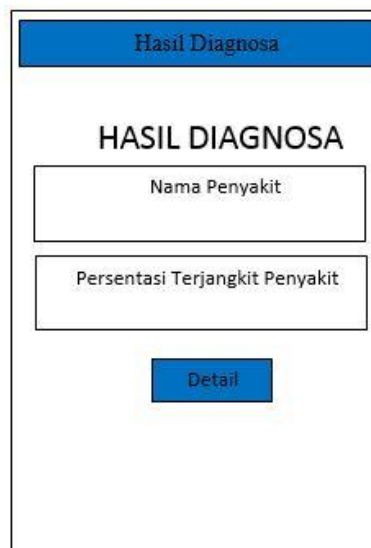
Halaman Menu Konsultasi Gejala merupakan halaman yang menampilkan pertanyaan-pertanyaan gejala yang harus dijawab oleh *user* dengan memilih *button* “YA” atau “TIDAK”. Rancangan halaman Menu Konsultasi Gejala dapat dilihat pada Gambar 3.15.



Gambar 3.15 Halaman Menu Konsultasi Gejala

4. Halaman Menu Hasil Diagnosis

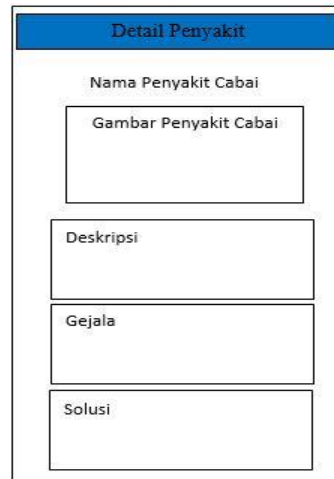
Halaman Menu Hasil Diagnosis merupakan halaman yang menampilkan menu Hasil Diagnosis. Halaman menu Hasil Diagnosis merupakan halaman yang ditampilkan sistem setelah user memilih gejala. Rancangan halaman Menu Hasil Diagnosis dapat dilihat pada Gambar 3.16.



Gambar 3.16 Halaman Menu Hasil Diagnosis

5. Halaman Detail Penyakit

Halaman Detail Penyakit menampilkan detail penyakit yang diderita dan memberikan informasi tentang deskripsi, gejala penyakit, dan memberikan solusi penanganannya. *User* dapat melihat apa penyakit yang di derita cabai dan cara penanganannya. Rancangan halaman Detail Penyakit disajikan pada Gambar 3.17.



Gambar 3.17 Halaman Detail Penyakit

6. Halaman Menu Tentang

Rancangan halaman menu Tentang dapat dilihat pada Gambar 3.18.

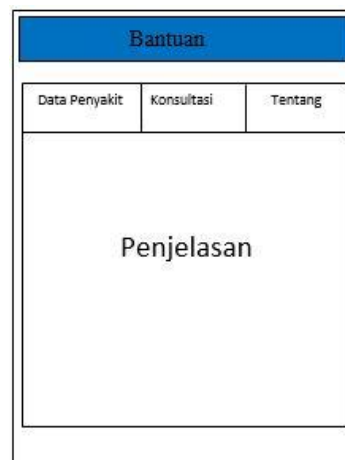


Gambar 3.18 Halaman Menu Tentang

Ketika *user* memilih menu “Tentang” pengguna akan melihat informasi mengenai aplikasi Sistem Pakar Diagnosis Penyakit Tanaman Cabai.

7. Halaman Bantuan

Halaman Bantuan merupakan halaman yang menampilkan informasi bantuan dalam penggunaan aplikasi. Rancangan halaman bantuan penyakit disajikan pada Gambar 3.19.



Gambar 3.19 Halaman Menu Bantuan

8. Halaman Kritik dan Saran

Rancangan halaman kritik dan saran disajikan pada Gambar 3.20.



Gambar 3.20 Halaman Kritik dan Saran

Halaman Kritik dan Saran berisi tentang kritik dan saran yang di miliki *user* terhadap aplikasi Sistem Pakar Cabai ini.

3.4 Metode Pengujian Sistem

Pengujian sistem dimaksudkan untuk menguji semua elemen–elemen perangkat lunak yang dibuat apakah sudah sesuai dengan yang diharapkan. Pendekatan kasus uji dalam penelitian ini adalah pengujian *black box* dengan metode *Equivalence Partitioning* (EP). Pengujian ini dilakukan dengan membagi domain masukan dari program ke dalam kelas-kelas sehingga *test case* dapat diperoleh. EP berusaha untuk mendefinisikan kasus uji yang menemukan sejumlah jenis kesalahan, dan mengurangi jumlah kasus uji yang harus dibuat. EP didasarkan pada premis masukan dan keluaran dari suatu komponen yang dipartisi ke dalam kelas-kelas, menurut spesifikasi dari komponen tersebut, yang akan diperlakukan sama (ekuivalen) oleh komponen tersebut. Pada pengujian ini harus diyakinkan bahwa masukan yang sama akan menghasilkan respon yang sama pula. Alasan menggunakan metode EP pada pengujian aplikasi Sistem Pakar Penyakit Cabai ini adalah karena metode ini dapat digunakan untuk mencari kesalahan pada fungsi yang diberikan ke aplikasi dan dapat mengetahui kesalahan pada *interface* aplikasi sehingga dapat mengurangi masalah terhadap nilai masukan.

Berikut ini merupakan rancangan daftar pengujian yang disajikan pada Tabel 3.1.

Tabel 3.1 Tabel Daftar Pengujian

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
1.	Versi Android	Pengujian kompatibilitas versi <i>operatif system</i> android	Pengujian pada android versi 4.3 (<i>Jelly Bean</i>)	Kompatibel dengan android versi 4.3 (<i>Jelly Bean</i>)
			Pengujian pada android versi 4.4 (<i>Kit-Kat</i>)	Kompatibel dengan android versi android versi 4.4 (<i>Kit-Kat</i>)
			Pengujian pada android Versi 5.0 (<i>Lollipop</i>)	Kompatibel dengan android Versi 5.0 (<i>Lollipop</i>)
			Pengujian pada android versi 6.0 (<i>Marshmallow</i>)	Kompatibel dengan android versi 6.0 (<i>Marshmallow</i>)
2.	Versi Android	Pengujian Resolusi Layar dan Densitas Layar pada android	Pengujian pada android dengan resolusi 4 inch	Tampilan terlihat sesuai atau baik pada android dengan resolusi 4 inch
			Pengujian pada android dengan resolusi 4,5 inch	Tampilan terlihat sesuai atau baik pada android dengan resolusi 4,5 inch
			Pengujian pada android dengan resolusi 5 inch	Tampilan terlihat sesuai atau baik pada android dengan resolusi 5 inch
			Pengujian pada android dengan resolusi 5,5 inch	Tampilan terlihat sesuai atau baik pada android dengan resolusi 5,5 inch

Tabel 3.1 Tabel Daftar Pengujian (lanjutan)

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
3.	<i>User Interface</i>	Pengujian pada <i>icon Pakar Cabai</i>	Klik <i>icon Pakar Cabai</i> pada perangkat android	Menampilkan <i>splash screen</i> dan diikuti halaman utama aplikasi
		Pengujian pada menu utama aplikasi PAKAR CABAI	Klik tombol menu "Data Penyakit"	Menampilkan <i>layout</i> Data Penyakit
			Klik tombol menu "Konsultasi Gejala"	Menampilkan <i>layout</i> Konsultasi Gejala
			Klik tombol menu "Bantuan"	Menampilkan <i>layout</i> Bantuan
			Klik tombol menu "Tentang"	Menampilkan <i>layout</i> Tentang
			Klik tombol menu "Kritik dan Saran"	Menampilkan <i>layout</i> Kritik dan Saran
	Fungsi <i>layout</i> Data Penyakit	Pengujian Pencarian database Penyakit Cabai	Klik tombol menu "Cari Penyakit"	Menampilkan kolom teks search dan <i>layout</i> masukan
			Klik tombol "Cari"	Menampilkan daftar nama penyakit
	Fungsi Layout Konsultasi	Pengujian pada <i>layout</i> Konsultasi	Klik tombol menu "Konsultasi Gejala"	Menampilkan <i>Layout</i> sub menu Konsultasi Gejala.
		Pengujian pada menu Konsultasi	Klik tombol menu "Ya"	Menampilkan <i>Layout</i> pertanyaan sistem atau hasil konsultasi

Tabel 3.1 Tabel Daftar Pengujian (lanjutan)

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
			Klik tombol menu "Tidak"	Menampilkan layout pertanyaan selanjutnya atau hasil konsultasi
			Klik Sub Menu Hasil Diagnosis Penyakit	Menampilkan detail penyakit cabai hasil konsultasi.
6	Fungsi pada menu Bantuan	Pengujian pada menu Bantuan	Klik tombol menu "Bantuan"	Menampilkan Layout bantuan
7	Fungsi pada menu Tentang	Pengujian pada menu Tentang	Klik tombol menu "Tentang"	Menampilkan Layout infomasi tentang aplikasi pakar cabai
8.	Fungsi pada menu Kritik dan Saran	Pengujian pada menu Kritik dan Saran	Klik icon "Email"	Menampilkan halaman email pengembang
			Klik icon "Google Playstore"	Menampilkan aplikasi pada halaman playstore
			Klik <i>icon</i> "Sosial Media	Menampilkan halaman sosial media pengembang

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat ditarik kesimpulan bahwa:

1. Telah berhasil dibangun aplikasi “Sistem Pakar Cabai” untuk membantu masyarakat umum dan petani dalam mengidentifikasi penyakit yang disebabkan oleh cendawan (jamur) berdasarkan gejala-gejala yang diberikan.
2. Sistem pakar yang dibangun dapat bermanfaat bagi masyarakat umum dan petani untuk mengetahui informasi tentang penyakit cabai.
3. Hasil pengujian fungsional menunjukkan bahwa sistem pakar yang dibangun telah berjalan sesuai yang diharapkan.
4. Berdasarkan penilaian penggunaan aplikasi melalui pengisian kuisioner, dapat disimpulkan bahwa aplikasi “Aplikasi Pakar Cabai” memperoleh presentase penilaian rata-rata sebesar 85,14% (sangat baik) menurut pakar cabai, 84,13% (sangat baik) menurut mahasiswa pertanian, 84,28% (sangat baik) menurut mahasiswa ilmu komputer, dan 86% (sangat baik) menurut petani cabai.

5.2 Saran

Beberapa saran untuk pengembangan lebih lanjut yang dapat diberikan setelah dilakukannya penelitian ini adalah sebagai berikut:

1. Penambahan data penyakit cabai yang lebih lengkap.
2. Penyempurnaan desain *user interface* aplikasi.
3. Penyederhanaan bahasa pada gejala dan penyakit cabai agar mudah dimengerti.

DAFTAR PUSTAKA

- Alex, S., 2015. *Jenis dan Budidaya Cabai*. Jakarta: Penebar Swadaya.
- Anshori, A. 2013. *Rancang Bangun Sistem Pakar Untuk Mendiagnosa Penyakit Tanaman Cabai Dengan Metode Forward Chaining* (Skripsi). Fakultas Teknik. Universitas Lampung. Bandar Lampung.
- Pratama, A.A., Aristoteles, and Wardianto, "Sistem Pakar Diagnosa Penyakit pada Ikan Budidaya Air Tawar dengan Metode Forward Chaining," *Jurnal Komputasi*, vol. 3, no. 2, pp. 92-98, 2015.
- Ardhika Praseda Ageng Putra, Aristoteles, and Rara Diantari, "Sistem Pakar Diagnosa Penyakit pada Ikan Budidaya Air Tawar dengan Metode Forward Chaining Berbasis Android," *Jurnal Komputasi*, vol. 4, no. 1, pp. 92-98, 2016.
- Arhami, M. 2005. *Konsep Dasar Sistem Pakar*. Yogyakarta: Andi.
- Azwar, S. 2011. *Sikap dan Perilaku. Dalam: Sikap Manusia Teori dan Pengukurannya 2nd ed.* Yogyakarta, Pustaka Pelajar.
- Badan Pusat Statistik Lampung. 2014. *Berita Resmi Statistik Provinsi Lampung*. Lampung: Badan Pusat Statistik Provinsi Lampung.
- Duriat, A. S., Gunaeni, N., and Wulandari, A. W. 2007. *Penyakit Penting Tanaman Cabai dan Pengendaliannya. Monografi, 1* (31).
- Fowler, Martin. 2004. *UML Distilled Panduan Singkat Bahasa pemodelan Objek Standar, Edisi 3*. Yogyakarta: Andi Publishing.
- Hartati, Sri and Iswanti, Sari. 2008. *Sistem Pakar dan Pengembangannya*. Yogyakarta : Graha Ilmu.
- Jiang, F., Y. Lu. 2012. Software testing model selection research based on yin- yang testing theory. In: *IEEE Proceeding of International Conference on Computer Science and Information Processing (CISP)*, pp. 590-594.
- Kusumadewi, Sri. 2003. *Artificial Intelligence: Teknik dan Aplikasinya*. Edisi 1. Yogyakarta: Graha Ilmu.

- Lowell, L. B., Sylvia, K.G., Glen, L. H., and Jean, M. P., 1991. *Penyakit-Penyakit Utama Cabai*, AVRDC The World Vegetable Center
- Muslim, A.A, 2015. “Sistem Pakar Diagnosa Hama Dan Penyakit Cabai Berbasis *Teorema Bayes*.” *JUTISI*, Vol. 4, No. 3, Desember 2015: 797 – 876.
- Nugroho, B. 2014. *Aplikasi Sistem Pakar*. Yogyakarta: Gava Media.
- Suci Oktaviani, Satria Perdana Arifin, and Ibnu Surya, “ Sistem Pakar Diagnosa Penyakit Ginjal Menggunakan Metode Hill Climbing”, *Jurnal Teknik Informatika*, Vol. 1, September 2012.
- Pitojo, S., 2003. *Benih Cabai*. Yogyakarta: Kanisius.
- Prajnanta. F, 2001. *Agribisnis Cabai Hibrida*. Jakarta: Penebar Swadaya
- Pressman, R. S. 2001. *Software Engineering A Practitioner’s Approach Fifth Edition*. McGraw-Hill Companies. Inc, New York.
- Puput Sinta Dewi, Ryana Dwi Lestari, and Ryani Tri Lestari, “ Sistem Pakar Diagnosis Penyakit Ikan Koi Dengan Metode Bayes”, *Jurnal Ilmiah Komputer dan Informatika*, vol. 4, no. 1, Maret 2015.
- Ralahalu, M. S., Hehanusa, M. L., and Oszaer, L.L, “Respon Tanaman Cabai Besar (*Capsicum annum L*) Terhadap Pemberian Pupuk Organik Hormon Tanaman Ungul”, *Jurnal Ilmu Budaya Tanaman*, vol. 2, no. 2, Oktober 2013.
- Setiadi, 2015. *Bertanam Cabai*, Edisi Revisi. Jakarta: Penebar Swadaya.
- Siswanto, 2010. *Kecerdasan Tiruan*. Edisi 2. Yogyakarta: Graha Ilmu.
- Sumpena U, “Penetapan Kadar Capsaicin Beberapa Jenis Cabai (*Capsicum sp*) di Indonesia”, *Jurnal Balai Penelitian Tanaman Sayuran*, Vol 9, No. 2, 2013.
- Sutojo, T., Edy M., dan Vincent. 2011. *Kecerdasan Buatan*. Yogyakarta: Andi.