



**APLIKASI SISTEM PAKAR IDENTIFIKASI HAMA DAN PENYAKIT
PADA TANAMAN KEDELAI MENGGUNAKAN METODE
TEOREMA BAYES**

SKRIPSI

oleh

Hofi Atmajaya

NIM 122410101067

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2017



**APLIKASI SISTEM PAKAR IDENTIFIKASI HAMA DAN PENYAKIT
PADA TANAMAN KEDELAI MENGGUNAKAN
METODE TEOREMA BAYES**

SKRIPSI

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

oleh
Hofi Atmajaya
NIM 122410101067

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER
2017**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT yang senantiasa memberikan rahmat, taufik, hidayah, dan bimbingannya sehingga bisa menyelesaikan tugas akhir ini dengan baik.
2. Ibu dan Bapak tercinta yang selalu memberikan do'a, kasih sayang, dan dukungan.
3. Kakak, Adik, dan seluruh keluarga yang telah memberikan dorongan positif.
4. Dosen Pembimbing.
5. Almameter Program Studi Sistem Informasi Universitas Jember.
6. Balai Penelitian Aneka Kacang dan Umbi (Balitkabi) Malang.

MOTTO

Allah menghendaki kemudahan bagimu, dan tidak menghendaki kesukaran bagimu. (terjemahan Q.S. Al-Baqoroh ayat 185) *)

Kelambatan adalah penya-nyiaan. Cepat – cepatlah selagi ada kesempatan sebelum ia berubah menjadi kesedihan. **)



*) Al Hidayah House of Quran SDN. BHD. 2010. *Terjemahan Bahasa Melayu Al Qur'an Al Karim (Rash Utmani)*. Selangor Darul Ehsan: Percetakan Zafar SDN. BHD.

**) Ali bin Abi Thalib. 2012. *Tanyalah Aku Sebelum Kau Kehilangan Aku Kata-kata Mutiara Ali bin Abi Thalib*. Bandung: Pustaka Hidayah.

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama: Hofi Atmajaya

NIM: 122410101067

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit Kedelai Menggunakan Metode Teorema *Bayes*”, adalah benar-benar hasil karya sendiri, kecuali dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 14 Desember 2017

Yang menyatakan,

Hofi Atmajaya

NIM 122410101067

SKRIPSI

**APLIKASI SISTEM PAKAR IDENTIFIKASI HAMA DAN PENYAKIT
KEDELAI MENGGUNAKAN TEOREMA BAYES**

Oleh

Hofi Atmajaya

NIM 122410101067

Pembimbing:

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

Dosen Pembimbing Pendamping : Oktalia Juwita, S.Kom., M.MT.

PENGESAHAN PEMBIMBING

Skripsi berjudul “Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit Kedelai Menggunakan Teorema Bayes” telah diuji dan disahkan pada:

hari, tanggal : Jumat, 22 Desember 2017

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

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

NIP 196909281993021002

Oktalia Juwita, S.Kom., M.MT.

NIP 198110202014042001

PENGESAHAN PENGUJI

Skripsi berjudul “Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit Kedelai Menggunakan Metode Teorema *Bayes*” telah diuji dan disahkan pada:

hari, tanggal : Jumat, 22 Desember 2017

tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji,

Penguji I,

Penguji II,

Achmad Maududie, S.T., M.Sc.

197004221995121001

Nova El Maidah, S.Si., M.Cs.

198411012015042001

Mengesahkan

Ketua Program Studi,

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

NIP 196704201992011001

RINGKASAN

Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit Kedelai

Menggunakan Metode Teorema *Bayes*; Hofi Atmajaya, 122410101067; 2016:
138 halaman; Progam Studi Sistem Informasi Universitas Jember.

Kedelai merupakan salah satu tanaman pangan yang sangat penting bagi penduduk Indonesia sebagai sumber protein nabati, bahan baku industri pakan ternak, dan bahan baku aneka industri olahan pangan. Seiring kebutuhannya, peningkatan mutu maupun jumlah produksi kedelai dihadapkan pada masalah hama, penyakit, dan kerentanan pada tanaman kedelai, oleh sebab itu Aplikasi Identifikasi Hama dan Penyakit Pada Tanaman Kedelai Menggunakan Metode Teorema Bayes dikembangkan untuk membantu pengguna khususnya petani kedelai dalam mengidentifikasi hama dan penyakit kedelai beserta saran penanggulangan yang dapat direkomendasikan. Pada penelitian ini, menggunakan sebanyak 24 hipotesa yang terdiri dari 15 hama dan 9 penyakit yang di dalamnya memiliki gejala sebanyak 76. Proses pengidentifikasian hama atau penyakit dilakukan dengan memilih gejala yang tampak pada tanaman dan disesuaikan dengan gejala yang terjadi pada tanaman. Nilai probabilitas yang dihasilkan dari perhitungan Teorema Bayes yang diterapkan pada aplikasi akan diurutkan berdasarkan nilai *bayes* pada tiap-tiap hipotesa yang terbesar, kemudian mengambil nilai yang terbesar sebagai hasil identifikasi. Hasil pengujian akurasi sistem dengan menginputkan beberapa gejala yang berbeda, didapatkan sejumlah 22 hipotesa yang valid dari 24 jumlah hipotesa dengan persentase akurasi sistem sebesar 91,66% dan memiliki reliabilitas yang baik dengan koefisien reliabilitas sebesar 0,780165.

PRAKATA

Alhamdulillahirobbil'alamin, rasa syukur kita panjatkan kehadirat Allah SWT karena limpahan rahmat, kasih, dan sayangnya, penulis dapat menyelesaikan skripsi dengan judul “Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit Kedelai Menggunakan Teorema Bayes”. Sholawat serta salam tetap tercurah limpahkan kepada baginda Nabi besar Muhammad SAW. Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penulisan dan penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan banyak terima kasih kepada:

1. Bapak Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Bapak Anang Andrianto, S.T., M.T., selaku Dosen Pembimbing Utama dan Ibu Oktalia Juwita, S.Kom., M.MT., selaku Dosen Pembimbing Pendamping yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Ibu Nelly Oktavia Adiwijaya, S.Si., M.T. selaku Dosen Pembimbing Akademik yang telah membimbing selama menjadi mahasiswa.
4. Seluruh Bapak dan Ibu Dosen beserta Staff dan Karyawan di Program Studi Sistem Informasi Universitas Jember.
5. Bapak, Ibu, Kakak, Adik, dan keluarga di rumah yang telah memberikan dorongan dan dukungan untuk menyelesaikan skripsi ini.
6. Yuwaka, Nata, Puput, teman kontrakan, dan teman-teman seperjuangan.

Penulis menyadari bahwasanya dalam penulisan skripsi ini jauh dari kesempurnaan dan oleh sebab itu penulis mengharapkan kritikan maupun saran yang membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat untuk kedepannya.

Jember, 7 September 2017

Penulis

DAFTAR ISI

	Halaman
SKRIPSI.....	i
PERSEMBAHAN.....	ii
MOTTO	iii
PERNYATAAN.....	iv
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN PENGUJI.....	vii
RINGKASAN	viii
PRAKATA.....	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xvi
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan dan Manfaat	3
1.1.1 Tujuan	3
1.1.2 Manfaat	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan	4
BAB 2. TINJUAN PUSTAKA	6
2.1 Penelitian Terdahulu	6
2.2 Kedelai	7

2.4 Sistem Pakar.....	12
2.5 System Developent Life Cycle (SDLC) Waterfall	14
2.6 Teorema <i>Bayes</i>	14
2.7 Penerapan Metode Teorema Bayes Pada Sistem Pakar Identifikasi Hama Kedelai	15
BAB 3. METODOLOGI PENELITIAN	17
3.1 Jenis Penelitian.....	17
3.2 Tempat dan Waktu Penelitian	17
3.3 Tahapan Penelitian	17
3.3.1 Analisis Kebutuhan	18
3.3.2 Desain Sistem.....	19
3.3.3 Implementasi	19
3.3.4 <i>Testing</i> dan Evaluasi	19
3.3.5 Pemeliharaan	21
BAB 4. PENGEMBANGAN SISTEM.....	22
4.1 Analisis Kebutuhan	22
4.1.1 SOP (Statement of Purpose)	22
4.1.2 Kebutuhan Fungsional	23
4.1.3 Kebutuhan Non-Fungsional	23
4.1.4 Fungsi Sistem.....	24
4.2 Desain Sistem.....	25
4.2.1 <i>Business Process Diagram</i>	25
4.2.2 <i>Use Case Diagram</i>	26
4.2.3 <i>Use Case Skenario</i>	28
4.2.4 <i>Activity Diagram</i>	32

4.2.5 <i>Sequence</i> Diagram.....	37
4.2.6 <i>Class</i> Diagram.....	41
4.2.7 <i>Entity Relationship</i> Diagram (ERD)	42
4.3 Penulisan Kode Program.....	43
4.4 Pengujian Sistem.....	48
4.4.1 <i>White Box</i>	48
4.4.2 Metode <i>Black Box</i>	52
4.4.3 Uji Validitas dan Reliabilitas	57
BAB 5. HASIL DAN PEMBAHASAN	59
5.1 Hasil Pengembangan Sistem	59
5.1.1 Tampilan Login Area	59
5.1.2 Tampilan Menu Utama	59
5.1.3 Tampilan <i>List</i> Gejala.....	61
5.1.4 Tampilan Form Tambah Gejala	61
5.1.5 Tampilan List Hipotesa (Hama dan Penyakit).....	62
5.1.6 Tampilan Form Tambah Hipotesa	63
5.1.7 Tampilan Detail Hipotesa (Hama dan Penyakit)	63
5.1.8 Tampilan Tambah Gejala pada Salah Satu Hipotesa	64
5.1.9 Tampilan Credits.....	65
5.1.10 Tampilan Identifikasi	65
5.1.11 Tampilan Hasil Identifikasi.....	66
5.2 Hasil Implementasi Metode Teorema <i>Bayes</i> pada Sistem Pakar Identifikasi Hama dan Penyakit pada Tanaman Kedelai	67
5.2.1 Hasil Penerapan Metode Teorema <i>Bayes</i> ke dalam Kode Program.....	67

5.2.2 Hasil Identifikasi pada Sistem Pakar Identifikasi Hama dan Penyakit pada Tanaman Kedelai	72
5.3 Pembahasan Perancangan dan Pengembangan Sistem	72
5.4 Pembahasan Implementasi Metode Teorema Bayes	73
5.4.1 Pembahasan Metode Teorema <i>Bayes</i> dalam Sistem Pakar	73
5.4.2 Pembahasan Perhitungan Manual dengan satu sampel gejala	79
5.4.3 Pembahasan Perhitungan Manual dengan dua sampel gejala	83
5.4.4 Pembahasan Perhitungan Manual dengan sampel tiga gejala.....	88
5.5 Pengujian Sistem	93
5.5.1 Uji Akurasi Sistem	93
5.5.2 Uji Validitas	100
5.5.3 Uji Reliabilitas	102
BAB 6. PENUTUP	105
6.1 Kesimpulan	105
6.2 Saran.....	106
DAFTAR PUSTAKA	107
LAMPIRAN.....	108

DAFTAR GAMBAR

Gambar 2.1 Struktur Sistem Pakar (Kusumadewi, 2003).....	13
Gambar 2.2 <i>Flowchart</i> Sistem Pakar Identifikasi Kedelai menggunakan Teorema Bayes	16
Gambar 3.1 Diagram <i>SDLC</i> Waterfall (A.S. & Salahuddin, 2011).....	18
Gambar 4.1 Business Process Diagram Sistem Pakar Identifikasi Hama dan Penyakit Kedelai	25
Gambar 4.2 <i>Use Case</i> Diagram Sistem Pakar Identifikasi Hama atau Penyakit pada Tanaman Kedelai.....	26
Gambar 4.3 <i>Activity</i> Diagram Pengelolaan Data Gejala (<i>View</i>).....	33
Gambar 4.4 <i>Activity</i> Diagram Pengelolaan Data Gejala (<i>Tambah</i>)	33
Gambar 4.5 <i>Activity</i> Diagram Pengelolaan Data Gejala (<i>Delete</i>)	34
Gambar 4.6 <i>Activity</i> Diagram Identifikasi Hama atau Penyakit	36
Gambar 4.7 <i>Sequence</i> Diagram Pengelolaan Data Gejala (<i>View</i>)	38
Gambar 4.8 <i>Sequence</i> Diagram Pengelolaan Data Gejala (<i>Tambah</i>)	39
Gambar 4.9 <i>Sequence</i> Diagram Pengelolaan Data Gejala (<i>Delete</i>)	40
Gambar 4.10 <i>Sequence</i> Diagram Identifikasi Hama dan Penyakit	41
Gambar 4.11 <i>Class</i> Diagram Aplikasi Identifikasi Hama dan Penyakit Kedelai .	42
Gambar 4.12 <i>Entitiy Relationship</i> Diagram Aplikasi Identifikasi Hama dan Penyakit Kedelai	43
Gambar 4.13 Diagram Alir <i>function</i> hitung_bayes	51
Gambar 5.1 Tampilan <i>Login Area</i>	60
Gambar 5.2 Tampilan menu utama untuk masing-masing hak akses (dari kiri hak akses <i>user</i> , hak akses pakar, dan hak akses admin)	60
Gambar 5.3 Tampilan list gejala pada user dan pada hak akses pakar dan admin	61
Gambar 5.4 Tampilan form input gejala	62
Gambar 5.5 Tampilan list hipotesa	62
Gambar 5.6 Tampilan form tambah hipotesa.....	63
Gambar 5.7 Tampilan detail salah satu hipotesa.....	64
Gambar 5.8 Tampilan form tambah gejala pada salah satu hipotesa.....	64

Gambar 5.9 Tampilan credits	65
Gambar 5.10 Tampilan view identifikasi disertai checklist beberapa gejala	66
Gambar 5.11 Tampilan hasil identifikasi	67



DAFTAR TABEL

Tabel 2.1 Jenis Hama Penting pada Tanaman Kedelai (Marwoto, 2013).....	7
Tabel 2.2 Jenis Penyakit pada Tanaman Kedelai (Marwoto, 2013)	11
Tabel 4.1 Deskripsi Aktor	27
Tabel 4.2 Deskripsi <i>Use Case</i>	27
Tabel 4.3 <i>Use Case</i> Skenario Pengelolaan Data Gejala.....	29
Tabel 4.4 <i>Use Case</i> Skenario Identifikasi Hama dan Penyakit Kedelai	31
Tabel 4.5 Kode Program <i>Class</i> Teorema_bayes	43
Tabel 4.6 Kode Program <i>Class</i> Hitung_bayes_tabel	45
Tabel 4.7 Kode Program <i>Class</i> Hasil.....	47
Tabel 4.8 <i>Listing</i> program menghitung pembilang.....	49
Tabel 4.9 <i>Listing</i> program menghitung penyebut	49
Tabel 4.10 <i>Listing</i> Program menghitung nilai bayes masing-masing hipotesa.....	49
Tabel 4.11 <i>Listing</i> Program menghitung total nilai bayes	50
Tabel 4.12 <i>Listing</i> Program mengurutkan nilai bayes terbesar ke terkecil dengan menggunakan metode bubble sort.....	50
Tabel 4.13 <i>Test Case function</i> hitung_bayes.....	51
Tabel 4.14 Pengujian sistem dengan metode <i>Black Box</i>	52
Tabel 4.15 Daftar pertanyaan pada angket atau kuisisioner	57
Tabel 4.16 Hasil Rekap Kuisisioner untuk aplikasi Identifikasi Hama dan Penyakit pada Kedelai.....	58
Tabel 5.1 Kode program menghitung bayes_tabel	68
Tabel 5.2 Potongan kode program menghitung pembilang	69
Tabel 5.3 Potongan kode program menghitung penyebut pada rumus teorema bayes.....	70
Tabel 5.4 Potongan kode program untuk menghitung nilai bayes pada seluruh hipotesa	70
Tabel 5.5 Potongan kode program menghitung total nilai bayes masing-masing hipotesa	71

Tabel 5.6 Function pengurutan nilai bayes seluruh hipotesa menggunakan metode <i>Bubble Sort</i>	72
Tabel 5.7 Tabel Bayes pada perhitungan teorema <i>bayes</i> sistem pakar identifikasi hama dan penyakit kedelai	75
Tabel 5.8 Sampel inputan satu gejala.....	79
Tabel 5.9 Hasil perhitungan pembilang dan penyebut menggunakan sampel inputan satu gejala.....	81
Tabel 5.10 Hasil nilai bayes dan total nilai bayes masing-masing hipotesa pada sampel inputan satu gejala	83
Tabel 5.11 Sampel inputan dua gejala	84
Tabel 5.12 Hasil perhitungan pembilang dan penyebut menggunakan sampel dua gejala	86
Tabel 5.13 Hasil nilai bayes dan total nilai bayes masing-masing hipotesa	87
Tabel 5.14 Sampel inputan gejala	88
Tabel 5.15 Hasil perhitungan pembilang dan penyebut menggunakan sampel 3 gejala	90
Tabel 5.16 Total nilai bayes pada masing-masing hipotesa menggunakan sampel 3 gejala	92
Tabel 5.17 Hasil Validitas Sistem.....	93
Tabel 5.18 Hasil perhitungan validitas dari rekap skor kuisisioner.....	101
Tabel 5.19 Hasil perhitungan reliabilitas dari rekap skor kuisisioner	103

BAB 1. PENDAHULUAN

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

1.1 Latar Belakang

Kedelai merupakan salah satu tanaman pangan yang sangat penting bagi penduduk Indonesia, karena fungsinya sebagai sumber protein nabati, bahan baku industri pakan ternak, dan bahan baku aneka industri olahan pangan (Rukmana & Yuniarsih, 2006). Kedelai mempunyai peran dan sumbangan yang besar bagi penyediaan bahan pangan bergizi bagi penduduk Indonesia karena kandungan proteinnya yang kaya akan asam amino.

“Upaya peningkatan produksi kedelai dihadapkan kepada masalah hama, penyakit, dan ketidakseimbangan hara di tanah. Serangan hama dan penyakit juga berpotensi menurunkan kualitas hasil dan ketidakseimbangan hara pada tanah tidak hanya berdampak terhadap penurunan produksi dan mutu hasil, tetapi juga menyebabkan tanaman lebih rentan terhadap serangan hama dan penyakit. Serangan hama tertentu pada tanaman kedelai seringkali menampilkan gejala serupa dengan penyakit gejala dan ketidakseimbangan hara. Oleh karena itu, gejala tersebut perlu diidentifikasi agar penyebabnya dapat diketahui dengan tepat untuk menentukan cara pengendalian atau pemulihan tanaman dengan efisien dan efektif” (Marwoto, Sri Hardaningsih, & Taufiq, 2006).

Selain itu, dalam proses penanaman kedelai terdapat beberapa kendala yang terjadi disebabkan perubahan iklim hingga mengakibatkan meningkatnya intensitas serangan hama dan penyakit yang diiringi kurangnya tenaga penyuluh pertanian (Wisnu Mahendra, Achmad Ridok, & Nurul Hidayat, 2010). Petani kedelai merupakan pihak yang paling membutuhkan informasi mengenai jenis hama dan penyakit beserta cara penanggulangan untuk mengatasi serangan hama atau penyakit tersebut. Bentuk informasi maupun data-data yang berkaitan tentang hama dan penyakit pada tanaman kedelai hanya dikemas dalam bentuk booklet. Selain

itu, frekuensi penyuluh pertanian kepada petani kedelai juga tidak menentu. Hal tersebut menjadi salah satu penyebab para petani kedelai kekurangan dalam memperoleh pengetahuan tentang pengendalian maupun pemulihan yang disebabkan oleh hama dan penyakit tanaman kedelai.

Berdasarkan kurangnya para petani kedelai memperoleh pengetahuan tentang pengendalian maupun pemulihan sehingga dilakukan sebuah penelitian dengan mengembangkan sebuah sistem pakar identifikasi hama dan penyakit pada tanaman kedelai berbasis Android. Setiap sistem pakar sebagai bagian dari kecerdasan buatan memerlukan mesin inferensi yang salah satunya adalah Teorema Bayes untuk memproses data masukan hingga menghasilkan keluaran yang diharapkan. Teorema Bayes memiliki kelebihan dalam penggunaan *bayes* tabel yang dinamis sebagai nilai bayes dasar dalam memperhitungkan gejala yang berkaitan tanpa menggunakan *rule* atau kaidah penelusuran sehingga penggunaan metode Teorema Bayes memungkinkan untuk dilakukannya penambahan jumlah hama dan penyakit tanpa mengatur ulang *rule* atau kaidah penelusuran.

Kontribusi dalam pengembangan kedelai juga pernah dilakukan oleh penelitian sebelumnya yaitu pengembangan aplikasi sistem pakar dalam bentuk platform web (Wisnu Mahendra dkk., 2010), namun memiliki kekurangan dalam hal mobilitas dan menggunakan sejumlah 9 penyakit dan 23 gejala sehingga mendorong penulis untuk mengembangkan aplikasi dalam bentuk platform mobile apps (Android) dengan melakukan penambahan jumlah gejala dan hama serta penyakit.

1.2 Rumusan Masalah

Dengan mempertimbangkan latar belakang di atas, dapat dirumuskan beberapa rumusan masalah yaitu:

- a. Bagaimana menerapkan metode Teorema *bayes* sebagai mesin inferensi untuk sistem pakar identifikasi hama dan penyakit pada tanaman kedelai?
- b. Bagaimana membangun aplikasi yang berbasis android untuk sistem pakar identifikasi hama dan penyakit pada tanaman kedelai?

1.3 Tujuan dan Manfaat

Tujuan dan manfaat yang dipaparkan merupakan berbagai hasil yang ingin dicapai dan dapat digunakan dari beberapa aspek penelitian ini terselesaikan.

1.1.1 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah:

- a. Merancang dan membangun aplikasi berupa sistem pakar untuk mengidentifikasi hama dan penyakit pada tanaman kedelai dengan menggunakan metode teorema *bayes*.
- b. Membantu para petani kedelai untuk mendapat informasi maupun pemulihan tanaman kedelai yang disebabkan oleh hama dan penyakit pada tanaman kedelai.

1.1.2 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat diantara lain:

- a. Bagi Petani Kedelai
Dapat dijadikan sebagai media konsultasi dan informasi untuk pengidentifikasian dan saran penanggulangan terhadap gejala-gejala yang ditimbulkan oleh hama dan penyakit pada tanaman kedelai.
- b. Bagi Penulis
Dapat melatih kemampuan dan menerapkan ilmu pengetahuan yang telah diperoleh di Program Studi Sistem Informasi untuk membantu petani kedelai dan masyarakat umum.
- c. Bagi Perkembangan Ilmu Pengetahuan
Penelitian ini dapat digunakan sebagai bahan referensi dalam menambah pengetahuan maupun pengembangan lebih lanjut bagi peneliti lain.

1.4 Batasan Masalah

Dalam proses penelitian dan pengembangan “Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit Kedelai Menggunakan Metode Teorema *Bayes*” diperlukan suatu batasan masalah untuk menghindari ketidakkonsistenan konsep baik dari cakupan penelitian maupun pengembangan yang diantaranya:

- a. Aplikasi sistem pakar ini menggunakan metode Teorema *bayes* sebagai mesin inferensinya dalam merepresentasikan nilai dari suatu pakar kedalam aplikasi/sistem.
- b. Sistem ini fokus untuk menampilkan referensi tentang hama dan penyakit pada tanaman kedelai, hasil identifikasi dari hama atau penyakit, dan saran penanggulangannya.
- c. Aplikasi sistem pakar ini dirancang untuk diterapkan pada *device* berbasis Android.
- d. Penelitian ini tidak membahas hasil keluaran nilai *bayes* yang memiliki lebih dari satu hasil identifikasi.

1.5 Sistematika Penulisan

Sistematika penulisan skripsi ini diantaranya sebagai berikut:

- a. Pendahuluan
Bab pendahuluan menguraikan tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan skripsi yang masing-masing diuraikan didalam subbab tersendiri.
- b. Tinjauan Pustaka
Bab tinjauan pustaka memaparkan tentang penelitian terdahulu yang berkaitan dengan penelitian yang dilakukan penulis, pengertian dari sistem pakar, membahas tentang kedelai sebagai objek penelitian, menjelaskan tentang teorema *bayes* yang merupakan metode yang digunakan dalam perhitungan penentuan Hama dan Penyakit Kedelai, dan menjelaskan tentang penerapan teorema *bayes* kedalam sistem.
- c. Metodologi Penelitian
Bab metodologi penelitian menguraikan tentang jenis penelitian, tempat dan waktu penelitian, tahap penelitian dalam proses pengembangan sistem, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem terhadap penelitian yang dilakukan.
- d. Pengembangan Sistem

Bab pengembangan sistem menguraikan tentang teknik pengembangan sistem yang diawali dari analisa kebutuhan fungsional dan nonfungsional, merancang *Business Process diagram*, *use case diagram*, *use case scenario*, *activity diagram*, *sequence diagram*, *class diagram*, *entity relationship diagram* (ERD), penulisan kode program, dan proses pengujian sistem.

e. Hasil dan pembahasan

Bab hasil dan pembahasan menguraikan tentang pemecahan masalah yang telah dirumuskan sebelumnya melalui analisis yang disajikan dalam bentuk deskripsi dan ilustrasi berupa tabel dan gambar untuk memperjelas hasil penelitian.

f. Penutup

Bab penutup berisi tentang kesimpulan dari hasil analisa dan saran terhadap penelitian selanjutnya yang memiliki pembahasan dibidang yang sama.

BAB 2. TINJUAN PUSTAKA

Bab ini menjelaskan teori – teori serta pustaka yang digunakan untuk penelitian. Teori – teori ini diambil dari berbagai literature, buku dan jurnal.

2.1 Penelitian Terdahulu

Penelitian mengenai identifikasi hama dan penyakit tanaman kedelai sudah pernah dilakukan oleh peneliti asal Indonesia. Salah satunya yaitu penelitian yang dilakukan oleh Prawidya Destarianto dengan judul Penerapan Metode *Inference Tree* dan *Forward Chaining* Dalam Sistem Pakar Diagnosis Hama dan Penyakit Kedelai *Edamame* Berdasarkan Gejala Kerusakannya (Prawidya Destarianto, Erni Yudanigtiyas, & Sholeh Hadi Pramono, 2013). Pada Jurnal tersebut peneliti memaparkan bahwa sistem yang dikembangkan dapat mendiagnosis 12 jenis hama dan 4 penyakit kedelai dengan menggunakan metode tersebut mempunyai tingkat ketepatan yang sesuai dengan data jenis hama dan penyakit dengan nilai 100 yang didapatkan dari perhitungan jumlah jawaban dibagi dengan jumlah gejala dikalikan dengan 100%. Penggunaan metode (*Inference Tree* dan *Forward Craining*) pada penelitian di atas memberikan dorongan dalam skripsi ini peneliti untuk menggunakan metode yang berbeda (*Teorema Bayes*) tetapi dengan objek yang sama (Hama dan Penyakit Kedelai) untuk mengetahui seberapa baik metode *Teorema Bayes* dalam penerapannya.

Penelitian lain yaitu penelitian yang pernah dilakukan oleh Mohammad Johan Wahyudi dengan Judul Sistem Pakar Untuk Mengidentifikasi Penyakit Udag Galah Dengan Metode *Theorema Bayes* (Muhammad Johan Wahyudi & Abdul Fadlil, 2013). Pada Jurnal Tersebut peneliti menjelaskan bahwa *Teorema Bayes* mampu dijadikan alat penarik kesimpulan dalam mengidentifikasi penyakit Udag Galah berdasarkan gejala yang dimasukkan serta memberikan solusi layaknya seorang pakar. Selain itu, hal lain yang mendorong penulis menggunakan metode *teorema bayes* yaitu kedelai memiliki gejala-gejala yang nampak dan memiliki tahapan yang sama dalam hal mengidentifikasi penyakit seperti halnya saat mengidentifikasi penyakit pada udang galah.

2.2 Kedelai

Kedelai dikenal dengan beberapa nama lokal, diantaranya adalah kedele, kacang kepung, kacang bulu, gadela, dan demokam. Di Jepang dikenal dengan adanya kedelai rebus atau *Edamame* dan juga dikenal sebagai kedelai manis, serta kedelai hitam atau *Koramame*, sedangkan nama umum di Dunia disebut *Soybean* (Rukmana & Yuniarsih, 2006).

Para Ahli Botani mencatat suku kacang-kacangan atau *Papilionaceae* yang tumbuh di dunia mempunyai 690 genus dan sekitar 18.000 spesies. Kerabat dekat tanaman kedelai yang ditanam secara komersial di dunia diperkirakan keturunan atau kerabat jenis kedelai liar *G. soya* atau *G.usuriensis* (Rukmana & Yuniarsih, 2006).

Produksi kedelai terkendala oleh serangan hama dan penyakit yang dapat menurunkan kualitas maupun produksi kedelai. Serangan hama maupun penyakit ditandai oleh beberapa gejala yang tampak pada daun, batang, maupun polong atau biji kedelai. Di Indonesia tercatat lebih dari 111 spesies Arthropoda merupakan hama, 53 spesies merupakan bukan sasaran, 61 spesies predator, dan 41 spesies parasitoid (Okada, Wedanimbi Tengkano, & T. Djuwarso, 1998). Namun tidak semua jenis hama yang dapat menimbulkan kerugian pada tanaman kedelai (Wedanimbi Tengkano & M. Soehardjan, 1985). Jenis Hama penting dan merugikan pada tanaman kedelai disajikan pada Tabel 2.1.

Tabel 2.1 Jenis Hama Penting pada Tanaman Kedelai (Marwoto dkk., 2006)

No.	Nama	Gejala
1.	Hama Lalat Bibit Kacang (<i>Ophiomya phaseoli</i>)	<ol style="list-style-type: none"> 1. Terdapat lubang tusukan antara epidermis atas dan bawah keping biji 2. Terdapat telur dalam jaringan mesofil 3. Telur berwarna putih 4. Telur berbentuk lonjong 5. Terdapat lubang gerakan pada keping biji atau pangkal helai daun 6. Bintik-bintik putih pada keping biji 7. Bintik-bintik putih pada daun
2.	Hama Lalat Batang (<i>Melanagromyza sojae</i>)	<ol style="list-style-type: none"> 1. Bintik-bintik bekas tusukan alat peletak telur pada daun muda

No.	Nama	Gejala
		<ol style="list-style-type: none"> 2. Lubang gerekkan larva pada daun menyebabkan tanaman layu, mengering, dan mati 3. Terdapat telur pada bagian bawah daun yang lebih muda
3.	Lalat Pucuk (<i>Melanagromyza dolicostigma</i>)	<ol style="list-style-type: none"> 1. Seluruh helai daun layu 2. Telur keputih-putihan 3. Telur berbentuk lonjong 4. Telur diletakkan pada permukaan bawah daun bagian pucuk yang belum membuka
4.	Hama Aphis (<i>Aphis glycines Matsumura</i>)	<ol style="list-style-type: none"> 1. Tubuh hama berukuran kecil 2. Tubuh Hama lunak 3. Tubuh hama berwarna hijau kekuning-kuningan 4. Pertumbuhan tanaman kerdil 5. Tanaman terserang pada cuaca panas dan musim kemarau
5.	Hama Kutu Bemisia (<i>Bemisia tabaci Gennadius</i>)	<ol style="list-style-type: none"> 1. Telur terletak di permukaan bawah daun muda 2. Telur berwarna kuning terang dan bertangkai seperti kerucut 3. Daun menjadi kering dan layu
6.	Hama Tungau Merah (<i>Tetranychus cinnabarius Boisduval</i>)	<ol style="list-style-type: none"> 1. tubuh tungau berwarna merah dengan tungkai putih 2. telur diletakkan dibawah permukaan daun 3. Daun berwarna kekuning-kuningan 4. Terdapat jaringan benang halus pada daun
7.	Hama Kumbang Kedelai (<i>Phaedonia inclusa Stall</i>)	<ol style="list-style-type: none"> 1. Kumbang kedelai dewasa berbentuk kubah 2. Tubuh kumbang berwarna hitam mengkilap 3. Telur diletakkan dibawah permukaan daun secara berkelompok 4. Telur berbentuk bulat panjang 5. Telur berwarna kuning pucat
8.	Hama Ulat Grayak (<i>Spodoptera litura Fabricius</i>)	<ol style="list-style-type: none"> 1. Serangga dewasa berupa ngengat abu-abu 2. Telur diletakkan secara berkelompok 3. Daun yang terserang terlihat dari jauh berwarna putih

No.	Nama	Gejala
9.	Hama Ulat Jengkal (<i>Chrysodeixis chalsites</i>)	<ol style="list-style-type: none"> 1. Telur berwarna kuning 2. Hama berwarna hijau 3. Daun termakan hingga tersisa tulang daun 4. Kerusakan daun dari arah pinggir 5. Serangan larva terjadi pada stadia vegetatif
10.	Hama Ulat penggulung daun (<i>Lamprosema indicata</i>)	<ol style="list-style-type: none"> 1. Hama berukuran kecil berwarna coklat kekuningan 2. Telur diletakkan berkelompok pada daun muda 3. Ulat membentuk gulungan daun 4. Daun termakan hingga bersisa tulang daun 5. Kotoran berwarna coklat didalam gulungan daun
11.	Hama Ulat Helicoverpa (<i>Helicoverpa spp.</i>)	<ol style="list-style-type: none"> 1. Telur diletakkan satu persatu pada daun 2. Telur diletakkan satu persatu pada pucuk 3. Telur diletakkan satu persatu pada bunga 4. Telur berwarna kuning muda 5. Warna ulat tua hijau kekuningan 6. Warna ulat tua hijau 7. Warna ulat tua coklat 8. Warna ulat tua agak hitam kecokelatan
12.	Hama Kepik Polong (<i>Riptortus linearis</i>)	<ol style="list-style-type: none"> 1. Hama mirip dengan walang sanggit 2. Hama berwarna kuning coklat dengan garis putih kekuningan 3. Telur diletakkan berkelompok dipermukaan daun 4. Cairan polong dan biji terhisap oleh hama 5. Polong mengempis 6. Biji mengempis 7. Polong dan biji mengering 8. Polong gugur 9. Biji gugur
13.	Hama Kepik Hijau (<i>Nezara viridula Linnaeus</i>)	<ol style="list-style-type: none"> 1. Telur diletakkan secara berkelompok rata-rata 80 butir

No.	Nama	Gejala
		<ol style="list-style-type: none"> 2. Bentuk telur seperti cangkir berwarna kuning dan berubah merah bata setelah menetas 3. Pagi hari hama tinggal dipermukaan daun bagian atas 4. Siang hari hama akan turun untuk makan polong 5. Hasil dan kualitas biji menurun 6. Polong mengempis 7. Biji mengempis 8. Polong gugur 9. Biji menjadi busuk 10. Kulit biji keriput dan bercak coklat
14.	Hama Kepik Piezodorus (<i>Piezodorus rubrofasciatus</i>)	<ol style="list-style-type: none"> 1. Hama berwarna hijau 2. Telur berbentuk silinder, berwarna abu-abu kehitaman dan strip putih ditengah 3. Kepik jantan mempunyai garis melintang merah muda pada lehernya 4. Kepik betina mempunyai garis melintang putih pada lehernya 5. Telur diletakkan berkelompok pada permukaan daun bagian atas, polong, batang, atau rumput 6. Kepik menghisap cairan polong dan biji 7. Penurunan hasil dan kualitas biji
15.	Hama Penggerek Polong Kedelai (<i>Etiella spp.</i>)	<ol style="list-style-type: none"> 1. Serangga berwarna keabu-abuan dan mempunyai garis putih pada sayap depan 2. Telur diletakkan berkelompok 4-15 butir dibagian bawah daun, kelopak bunga, atau polong 3. Telur berbentuk lonjong 4. Tanda serangan berupa lubang gerek berbentuk bundar pada kulit polong 5. Bintik coklat tua pada kulit polong Lubang coklat tua pada kulit polong 6. Ada bekas jalan masuk larva kedalam biji

No.	Nama	Gejala
		7. Bekas lubang terdapat butir-butir kotoran cokelat muda

Selain hama yang dapat menurunkan kualitas maupun produksi pada tanaman kedelai, kendala lainnya yaitu serangan penyakit. Berbeda dengan hama yang dapat berupa serangga ataupun hewan pengganggu, penyakit dapat berupa serangan dari bakteri dan virus. Namun penyakit yang menyerang tanaman kedelai dapat memiliki gejala yang tampak seperti halnya gejala yang tampak oleh serangan hama. Daftar penyakit yang dapat menyerang tanaman kedelai disajikan pada Tabel 2.2.

Tabel 2.2 Jenis Penyakit pada Tanaman Kedelai (Marwoto dkk., 2006)

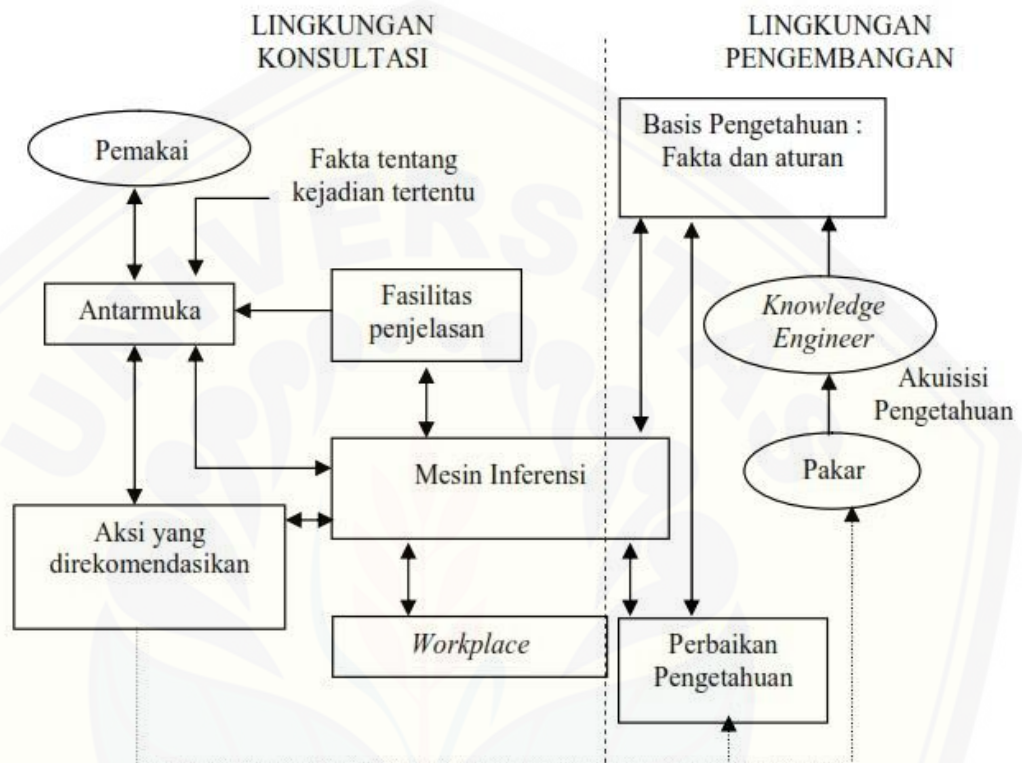
No.	Nama	Gejala
1.	Penyakit Karat Kedelai (<i>Phakopsora pachyrhizi</i>)	1. Bercak-bercak berwarna cokelat kemerahan bagian bawah daun seperti karat 2. Daun gugur 3. Polong Hampa
2.	Penyakit Pustul Bakteri (<i>Xanthomonas axonopodis</i>) (Pada Daun)	1. Bercak kecil berwarna hijau pucat pada kedua permukaan daun 2. Bisul warna cokelat muda atau putih pada permukaan bawah daun 3. Bercak terlihat lebih besar bervariasi tak beraturan berwarna kecokelatan
3.	Penyakit Antraknosa (<i>Colletotrichum dematium</i> <i>var truncatum</i> dan <i>C. destructivum</i>) (Pada Daun)	1. Tulang daun pada permukaan bawah menebal warna kecokelatan 2. Batang timbul bintik-bintik hitam berupa duri jamur 3. Perkecambahan biji terganggu
4.	Penyakit Downy Mildew (<i>Peronospora manshurica</i>) (Pada Daun)	1. Pada bawah daun timbul bercak warna putih kekuningan 2. Bercak berbentuk bulat dengan batas yang jelas antar bercak 3. Permukaan bawah daun timbul miselium dan konidium
5.	Penyakit Target Spot (<i>Corynespora cassicola</i>) (Pada Daun)	1. Bercak cokelat kemerahan pada daun, batang, biji, hipokotil, atau akar dengan diameter 10-15mm 2. Bercak membentuk lingkaran seperti papan tembak

No.	Nama	Gejala
6.	Rebah Kecambah, Busuk daun dan Polong (<i>Rhizoctonia solani</i>) (Penyakit Tular Tanah)	<ol style="list-style-type: none"> 1. Busuk di dekat akar pada tanaman baru tumbuh 2. Tanaman mati karena rebah 3. Timbul hawar (Busuk) dari bawah (Akar) ke atas (Daun) 4. Pada Daun timbul miselium menyebabkan daun-daun lengket menyerupai sarang laba-laba
7.	Penyakit Hawar Batang (<i>Sclerotium rolfsii</i>) (Penyakit Tular Tanah)	<ol style="list-style-type: none"> 1. Infeksi terjadi pada pangkal batang 2. Infeksi berwarna bercak cokelat muda hingga cokelat tua hingga mencapai hipokotil 3. Layu mendadak pada tanaman 4. Daun bercak bulat dari warna merah hingga cokelat tua 5. Daun mengering atau menempel pada batang mati 6. Terdapat miselium putih pada pangkal batang, sisa daun, dan pada tanah sekeliling tanaman
8.	Penyakit Hawar, Bercak Daun, dan Bercak Biji Ungu (<i>Cercospora kikuchii</i>) (Penyakit pada Benih)	<ol style="list-style-type: none"> 1. Daun berwarna ungu muda muda 2. Daun menjadi kasar, kaku, dan berwarna ungu kemerahan 3. Biji bercak berwarna ungu 4. Biji tidak beraturan dan membesar dengan warna variasi dari merah muda atau ungu pucat sampai ungu tua
9.	Penyakit Virus Mosaik (Penyakit pada Benih)	<ol style="list-style-type: none"> 1. Tulang daun pada daun muda menjadi kurang jernih 2. Daun berkerut dan mempunyai gambaran mosaik dengan warna hijau gelap 3. Tepi daun mengalami klorosis 4. Ukuran biji mengecil dan jumlah biji berkurang

2.4 Sistem Pakar

Secara umum, sistem pakar atau *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. Struktur sistem pakar terdiri dari 2 pokok yaitu lingkungan pengembang atau *development environment* dan lingkungan konsultasi atau *consultation environment* (Sri Kusumadewi, 2003).



Gambar 2.1 Struktur Sistem Pakar (Sri Kusumadewi, 2003)

Gambar 2.1 mewakili gambaran struktur sistem pakar diantaranya yaitu lingkungan pengembang yang digunakan sebagai pembangunan sistem pakar baik dari segi pembangunan komponen maupun basis pengetahuan. Akuisisi pengetahuan pakar akan dikonversi kedalam bentuk basis pengetahuan yang kemudian akan digunakan oleh mesin inferensi dalam melayani pemakai pada lingkungan konsultasi. Lingkungan konsultasi digunakan oleh seseorang bukan ahli untuk berkonsultasi yang didalamnya memiliki antarmuka yang akan dihadapkan kepada pemakai yang dilengkapi aksi yang direkomendasikan untuk berkonsultasi. Antarmuka tersebut dibantu oleh mesin inferensi dalam mengolah data inputan yang diberikan oleh pemakai melalui antarmuka yang memiliki fasilitas penjelas sebagai output keluaran hasil inferensi. Sistem pakar dapat berfungsi sebagai

pembantu keputusan seorang pakar dalam menentukan suatu keputusan dari inputan yang diberikan oleh user. Selain itu, sistem pakar dapat juga berfungsi sebagai pengganti pakar jika pengguna sistem adalah seorang user biasa.

2.5 System Development Life Cycle (SDLC) Waterfall

System Development Life Cycle atau Siklus Hidup Pengembangan Sistem dalam rekayasa perangkat lunak adalah fase-fase pendekatan yang dilakukan untuk menganalisis dan merancang perangkat lunak (Kenneth E. Kendall, 2010). SDLC Model *waterfall* melakukan pendekatan yang sistematis dan sekuensial dimulai dengan analisis kebutuhan sistem, desain sistem, implementasi kedalam kode program, pengujian dan evaluasi, serta pemeliharaan sistem.

Kelebihan dari SDLC model *waterfall* adalah metode ini masih lebih baik digunakan meskipun tergolong kuno karena ketika kebutuhan telah terdefinisi dengan baik maka alur kerja (*workflow*) dari analisis kebutuhan hingga pemeliharaan dapat dilakukan secara berurutan. Model *waterfall* sangat dianjurkan pada pengembangan perangkat lunak yang memiliki kebutuhan terdefinisi secara konsisten. Perubahan yang terjadi sewaktu waktu pada analisis kebutuhan, model *waterfall* dapat diiterasi menyesuaikan kebutuhan baru.

Disisi lain, model *waterfall* memiliki kelemahan diantaranya yaitu pengembangan harus dilakukan secara bertahap sehingga dapat membingungkan pengembang jika terjadi perubahan kebutuhan meskipun model *waterfall* memungkinkan untuk dilakukannya iterasi dalam pengembangannya. Tahapan-tahapan pada model *waterfall* sangat bergantung pada penyelesaian tahap sebelumnya (Pressman, 2015).

2.6 Teorema Bayes

Teorema *bayes* merupakan suatu metode yang digunakan untuk menghitung ketidakpastian data menjadi data yang pasti dengan membandingkan antara data ya dan tidak. Probabilitas *bayes* merupakan salah satu cara untuk mengatasi ketidakpastian data dengan menggunakan perhitungan formula *bayes* yang dapat dilihat pada Persamaan 2.1.

$$P(H_i|E) = \frac{P(E|H_i) \times P(H_i)}{\sum_{k=1}^n P(E|H_k) \times P(H_k)} \quad (2.1)$$

$P(H_i | E)$ = probabilitas hipotesis H_i (hama/penyakit) jika diberikan *evidence* E (gejala)

$P(E | H_i)$ = probabilitas munculnya *evidence* E (gejala) dengan memperhatikan hipotesis H_i (hama/penyakit)

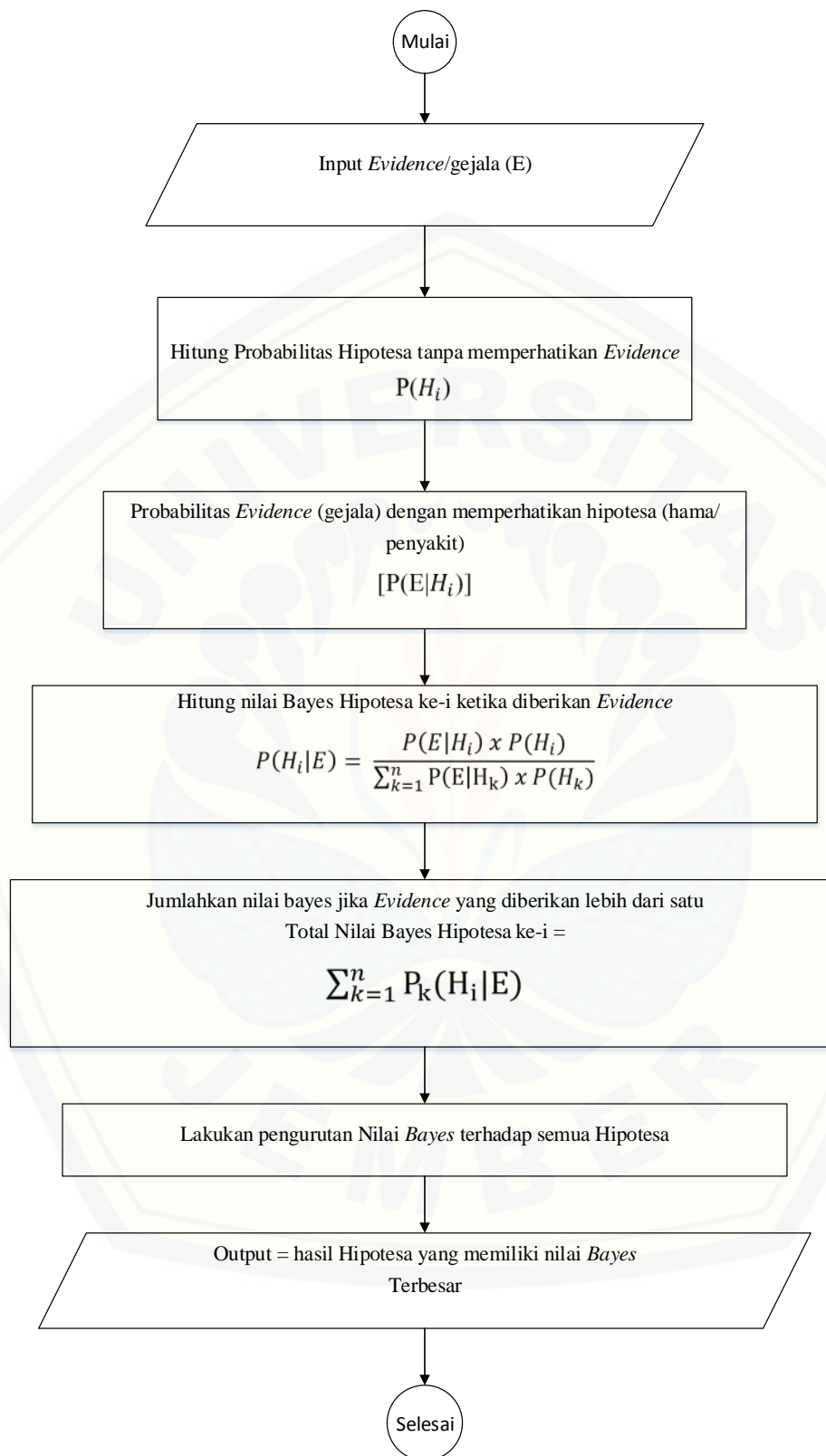
$P(H_i)$ = probabilitas H_i tanpa memperhatikan *evidence* E apapun

n = jumlah hipotesis yang terjadi

Proses perhitungan diawali dengan menghitung probabilitas tiap-tiap Hipotesis (hama/penyakit) tanpa memperhatikan *evidence* (gejala-gejalanya) [$P(H_i)$]. Kemudian menghitung probabilitas *evidence* (gejala) yang diberikan dengan memperhatikan Hipotesis (hama/penyakit) [$P(E|H_i)$]. Kemudian menjumlahkan hasil perkalian probabilitas $P(E|H_k) \times P(H_k)$ sebanyak hipotesis yang diduga. Hasil Identifikasi akan ditunjukkan dari hasil perhitungan Hipotesis yang memiliki nilai *bayes* terbesar.

2.7 Penerapan Metode Teorema Bayes Pada Sistem Pakar Identifikasi Hama Kedelai

Pengembangan Sistem Pakar Identifikasi Hama Kedelai diawali dengan membangun Basis Pengetahuan dengan mengakuisisi pengetahuan dari pakar hama dan penyakit kedelai kemudian mencari nilai *bayes* pada setiap gejala yang diinputkan oleh user pada aplikasi terhadap tiap-tiap hama pada basis pengetahuan. Nilai *bayes* yang terbesar pada tiap hama yang diperoleh dari total nilai *bayes* gejala yang bersangkutan akan dijadikan hasil identifikasi akhir. Proses penerapan Teorema *Bayes* pada Aplikasi Sistem Pakar Identifikasi Hama Kedelai dapat dilihat pada Gambar 2.2



Gambar 2.2 Flowchart Sistem Pakar Identifikasi Kedelai menggunakan Teorema Bayes

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode yang digunakan terhadap penelitian yang akan digunakan untuk mengembangkan sistem.

3.1 Jenis Penelitian

Jenis penelitian yang akan dilakukan merupakan pengembangan. Penelitian pengembangan dilakukan untuk menghasilkan suatu produk atau mengembangkan suatu produk. Produk dalam penelitian ini adalah “Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit pada Tanaman Kedelai menggunakan Teorema Bayes”. Aplikasi ini dikembangkan dengan menggunakan metode perhitungan yang berbeda dengan penelitian sebelumnya dalam mengeluarkan hasil identifikasi. Pada pengembangan aplikasi ini juga terdapat penambahan jumlah hama dan gejala serta perbaikan dalam sisi mobilitas sistem yaitu dikembangkan dalam bentuk *platform mobile* Android. Studi kasus dan pengumpulan data hama dan penyakit pada tanaman kedelai didapat dari booklet pertanian yang dikeluarkan oleh Balitkabi maupun booklet yang dikeluarkan oleh Balitbang Pertanian yang merupakan suatu rangkuman dari ilmu kepakaran tentang hama dan penyakit kedelai yang merujuk pada para peneliti sebagai pakar tanaman kedelai yang bertempat di Balai Penelitian Aneka Kacang dan Umbi (Balitkabi) Malang.

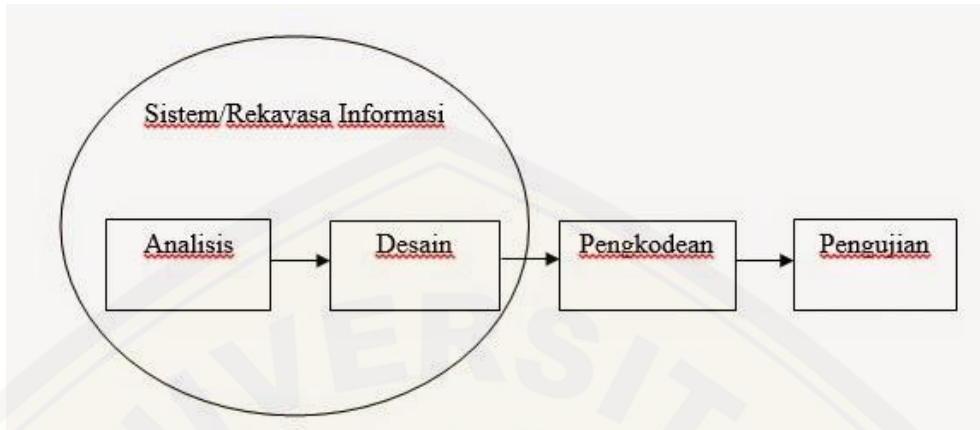
3.2 Tempat dan Waktu Penelitian

Tempat yang dilaksanakan untuk penelitian adalah Program Studi Sistem Informasi. Waktu penelitian dimulai pada bulan Juli 2016 sampai dengan selesai pengerjaan skripsi dan pengembangan sistem.

3.3 Tahapan Penelitian

Penelitian ini akan dilakukan dalam beberapa tahap yang disesuaikan dengan metode *Software Development Life Cycle (SDLC) waterfall* yang dibagi menjadi beberapa tahapan, yaitu analisis kebutuhan, desain sistem, implementasi,

pengujian dan pemeliharaan. Tahapan-tahapan tersebut ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram *SDLC* Waterfall (Rosa A.S & M. Salahuddin, 2011)

3.3.1 Analisis Kebutuhan

Tahap pertama pada proses perancangan sistem adalah tahap analisis kebutuhan. Pada tahap ini dilakukan penelitian mencari permasalahan yang ada untuk dapat dianalisis kebutuhan yang diperlukan sebagai solusi dari permasalahan yang muncul. Data-data yang telah didapat kemudian dikelompokkan menjadi kebutuhan fungsional dan non-fungsional. Teknik pengumpulan data yang dilakukan antara lain:

a. Studi Pustaka

Teknik ini dilakukan dengan tujuan sebagai dasar pembahasan penyusunan dasar teori yang dapat mendukung penelitian ini. Sumber yang digunakan dalam studi pustaka berupa buku, booklet, jurnal, dan penelitian terdahulu. Sumber studi pustaka dan teori yang dapat mendukung penelitian ini dapat dilihat pada bagian daftar pustaka dan jurnal penelitian terdahulu.

b. Wawancara

Wawancara merupakan teknik mengumpulkan data dengan cara mengajukan pertanyaan kepada narasumber untuk memperoleh data yang dibutuhkan guna menyelesaikan penelitian ini. Pertanyaan yang diajukan melalui surat elektronik atau email yang diajukan kepada Seksi Jasa Penelitian Balitkabi

(jaslit.balitikabi@gmail.com). Informasi detail mengenai wawancara dapat dilihat pada Lampiran D.

3.3.2 Desain Sistem

Desain sistem merupakan tahap pembuatan desain sistem, dimana pada penelitian ini menggunakan *Unified Modeling Language* (UML) yang dirancang dengan konsep *Object-Oriented Programming* (OOP). Pemodelan UML yang digunakan sebagai berikut:

- a. *Business Process*
- b. *Usecase Diagram*
- c. *Usecase Scenario*
- d. *Sequence Diagram*
- e. *Activity Diagram*
- f. *Class Diagram*
- g. *Entity Relationship Diagram*

Setelah perancangan sistem telah selesai dilakukan maka selanjutnya dilakukan tahap pembuatan sistem dan implementasi. Pembuatan sistem meliputi pembuatan desain interface, dan penulisan kode program menggunakan tools Android Studio sebagai editor.

3.3.3 Implementasi

Tahap implementasi merupakan tahapan untuk mengimplementasikan desain yang telah dirancang ke dalam kode program. Dengan melakukan penulisan kode program menggunakan bahasa pemrograman Java dan xml sebagai unit interfacenya.

3.3.4 *Testing* dan Evaluasi

Testing dan evaluasi digunakan untuk mengetahui sejauh mana sistem ini dapat berjalan. *Testing* berfungsi untuk mengetahui apakah sistem ini dapat berfungsi dengan baik sesuai dengan yang diharapkan. Serta untuk mengetahui letak kekurangan yang ada pada sistem. Pengujian dilakukan oleh tim penguji dari pengembang sistem. Selanjutnya dilakukan evaluasi serta perbaikan terhadap

kekurangan-kekurangan yang ada pada sistem ini dilakukan beberapa metode untuk pengujian yaitu diantaranya:

a. *White Box Testing*

White box testing merupakan cara pengujian dengan melihat modul yang telah dibuat dengan program-program yang ada. Pengujian ini, dilakukan oleh *developer*. Jika ada modul yang menghasilkan output yang tidak sesuai maka baris-baris program, variabel dan parameter yang terlibat pada unit tersebut satu persatu akan dicek dan diperbaiki, kemudian akan di *compile* ulang. Teknik pengujian ini menggunakan pengujian jalur dasar (*basis path testing*) dimana kompleksitas dari perangkat lunak yang dibangun akan dihitung menggunakan *Cyclomatic Complexity*.

b. *Black Box Testing*

Pengujian *Black Box* melibatkan pengguna/*user*, dimana hanya memperhatikan fungsionalitas yang berkaitan dengan masukan/keluaran (I/O) apakah sesuai dengan sistem yang dijalankan.

c. Uji Akurasi Sistem

Uji Akurasi sistem dilakukan setelah diterapkannya metode teorema *bayes* pada sistem pakar identifikasi hama dan penyakit pada tanaman kedelai bertujuan untuk mengetahui akurasi yang didapatkan dengan membandingkan dengan hasil yang diporelah dengan pakar.

d. Uji Validitas dan Reliabilitas

Uji Validitas merupakan cara pengujian yang dilakukan untuk mengetahui kevalidan atau kesahihan suatu instrument. Uji validitas ini menggunakan metode *Bivariate Pearson* yang dapat dilakukan dengan menggunakan angket/kuisisioner dan dapat dihitung menggunakan rumus:

$$r_{xy} = \frac{n \sum XY - (\sum X)(\sum Y)}{\sqrt{\{n \sum X^2 - (\sum X)^2\}\{n \sum Y^2 - (\sum Y)^2\}}}$$

r_{xy} = Koefisien Korelasi

N = Jumlah responden uji coba

X = Skor tiap Item

Y = Skor seluruh Item Responden

Sedangkan Uji Reliabilitas bertujuan untuk mengetahui apakah alat ukur/instrument memiliki tingkat konsistensi dan dapat dipercaya. Tinggi rendahnya reliabilitas secara empirik dapat ditunjukkan dengan nilai koefisien reliabilitas mendekati angka 1. Kesepakatan secara umum reliabilitas dianggap cukup memuaskan jika $\geq 0,7$. Pengujian reliabilitas instrumen dengan menggunakan rumus *Alpha Cronbach* sebagai berikut:

$$r = \left(\frac{n}{n-1} \right) \left(1 - \frac{\sum \sigma_t^2}{\sigma_t^2} \right)$$

r = reliabilitas yang dicari

n = jumlah item pertanyaan yang diuji

$\sum \sigma_t^2$ = jumlah varians skor tiap-tiap item

σ_t^2 = varians total

3.3.5 Pemeliharaan

Perangkat lunak yang sudah selesai akan mengalami perubahan. Perubahan biasanya berupa *error* sehingga diperlukan perbaikan dan pemeliharaan kepada sistem. Perubahan ini dilakukan agar sistem bersifat dinamis.

BAB 4. PENGEMBANGAN SISTEM

Bab ini menguraikan tentang analisis kebutuhan, desain sistem, implementasi, dan pengujian sistem yang digunakan dalam aplikasi sistem pakar identifikasi hama dan penyakit pada tanaman kedelai menggunakan metode teorema *bayes*. Tahap analisis hingga tahap pengujian menggunakan metode pengembangan *waterfall*.

4.1 Analisis Kebutuhan

Berdasarkan metode pengembangan *waterfall*, tahap awal yang dilakukan adalah tahap analisis kebutuhan sistem. Tahap analisis ini dilakukan terhadap objek penelitian untuk memperoleh kebutuhan sistem yang diantaranya berupa kebutuhan fungsional dan kebutuhan nonfungsional. Hasil analisis kebutuhan sistem tersebut akan digunakan untuk pembuatan sistem agar fungsi yang terbentuk dapat sesuai dengan kebutuhan awal.

Berdasarkan hasil dari studi pustaka tentang kedelai dan hasil wawancara, pengembangan Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit pada Tanaman Kedelai menggunakan Teorema *Bayes* membutuhkan data hama, data penyakit, data gejala, dan data penanganan terkait hama dan penyakit yang telah teridentifikasi. Data-data yang didapat merupakan hasil rangkuman para peneliti pertanian untuk hama dan penyakit tanaman kedelai yang diunduh melalui website resmi Jasa Penelitian Balitkabi.

Tujuan dikembangkannya sistem ini adalah untuk membantu Petani Kedelai sebagai *user* khususnya untuk mengidentifikasi hama ataupun penyakit dengan menginputkan gejala-gejala yang tampak pada tanaman dan menyesuaikan data gejala tersebut sebagai pilihan pada gejala yang tersedia pada sistem.

4.1.1 SOP (Statement of Purpose)

Sistem ini diharapkan mampu memproses kumpulan data gejala yang diinputkan oleh user menggunakan Teorema *Bayes* untuk menghasilkan nilai *Bayes* pada tiap-tiap hama dan penyakit dengan mempertimbangkan hasil nilai *Bayes* terbesar sebagai *output* atau hasil keluaran identifikasi. Terdapat 3 hak akses pada

sistem ini diantaranya adalah *User*, Admin, dan Pakar. Sistem dapat mengelola data gejala, data Hama dan Penyakit beserta saran penanggulangan pada tiap-tiap hama dan penyakit. Pengembangan sistem dikemas dalam platform Android untuk memudahkan user dalam hal mobilitas penggunaan fitur-fitur yang dikembangkan.

4.1.2 Kebutuhan Fungsional

Kebutuhan Fungsional sistem merupakan fitur-fitur inti yang harus tersedia di dalam sistem agar sistem dapat berfungsi dan berjalan sesuai dengan tujuan dan kebutuhan user. Kebutuhan fungsional sistem dari Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit Kedelai Menggunakan Teorema *Bayes* diantaranya yaitu:

- a. Sistem mampu mengelola Data Gejala yang diinputkan oleh Admin.
- b. Sistem mampu mengelola Data Hipotesa berupa Data Hama dan Penyakit pada Tanaman Kedelai yang diinputkan oleh Admin.
- c. Sistem mampu mengidentifikasi Hipotesa berupa Hama atau Penyakit berdasarkan hasil perhitungan nilai Teorema *Bayes* terbesar.
- d. Sistem mampu menampilkan hasil identifikasi kepada *user* berupa solusi atau saran penanggulangan hama dan penyakit kedelai.
- e. Sistem mampu memverifikasi data yang dapat diakses oleh Pakar dalam pengelolaan Data Gejala dan Data Hipotesa.
- f. Sistem mampu memberikan informasi pengembang sistem dan pihak yang terkait melalui tampilan Credits.

4.1.3 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan fitur-fitur yang dikembangkan untuk bersamaan dengan fitur inti untuk mendukung sistem dalam memenuhi kebutuhan fungsionalnya. Kebutuhan non-fungsional dari Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit pada Tanaman Kedelai menggunakan Teorema *Bayes* diantaranya yaitu:

- a. Sistem memiliki tampilan dengan penggunaan ikon dan kata yang sederhana pada beberapa tampilan sistem.

- b. Sistem memiliki tampilan judul aplikasi untuk menunjukkan identitas aplikasi dan fungsi sistem secara tidak langsung.
- c. Sistem menggunakan gambar tombol menu yang mewakili data ataupun informasi yang disediakan sistem guna mewakili maksud data yang akan ditampilkan.

4.1.4 Fungsi Sistem

Sistem ini dikembangkan dengan memiliki fungsi utama yang terletak pada fitur login sesuai dengan hak akses dari setiap pengguna dalam sistem. Sistem dapat menyajikan tampilan yang disesuaikan dengan kebutuhan hak akses *user* setelah sistem berhasil melakukan proses autentikasi *username* dan *password*, hak akses tersebut diantaranya:

- a. Admin

Admin merupakan hak akses yang utama dalam hal pengelolaan data yang bergerak dalam sistem. Sistem akan menyajikan fitur-fitur yang disesuaikan dengan kebutuhan hak akses sebagai admin antara lain yaitu mengelola data gejala, mengelola data hipotesa yang terdiri dari Hama dan Penyakit, dan mengelola data solusi yang terdiri dari saran pengendalian atau penanggulangan sebagai hasil dari identifikasi.

- b. Pakar

Pakar merupakan hak akses yang berfungsi untuk memverifikasi data gejala maupun data hipotesa yang telah diinputkan oleh hak akses Admin. Data tersebut perlu diverifikasi sebelum dapat dilihat oleh hak akses user dalam melihat data gejala, data hipotesa, maupun menggunakan fitur identifikasi.

- c. *User*

User dapat terdiri dari pengguna pada umumnya tetapi dikhususkan untuk Petani Kedelai dalam hal informasi mengenai pama dan penyakit kedelai. Hak Akses *User* memiliki fitur antara lain yaitu melihat daftar informasi gejala yang dapat dijangkit oleh tanaman kedelai, melihat informasi hama dan penyakit beserta cara penaggulangan, dan menggunakan fitur identifikasi dalam mencari

hama dan penyakit dengan informasi gejala yang tampak pada tanam kedelai sebagai inputan utama.

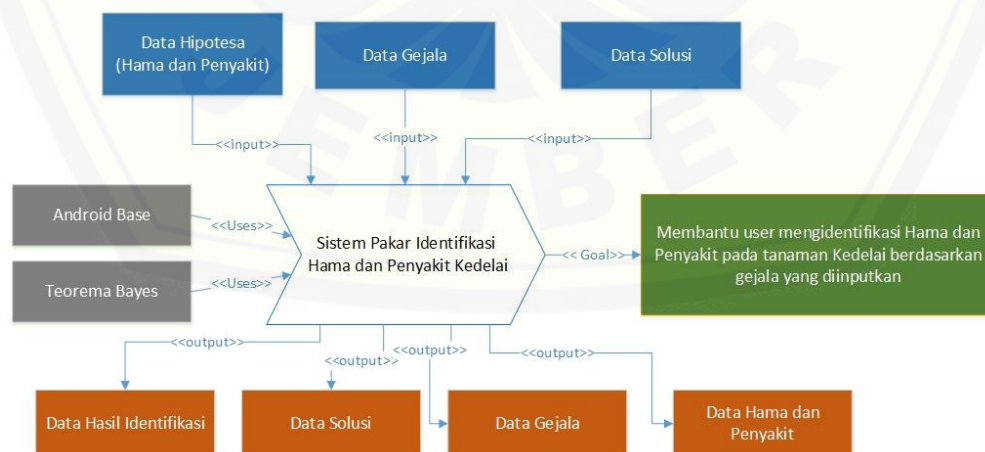
4.2 Desain Sistem

Pengembangan sistem dalam tahapan berikutnya setelah analisis kebutuhan adalah melakukan tahapan desain sistem yaitu proses pengembangan yang menggambarkan perencanaan dalam pembangunan sistem. Desain sistem dari Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit Kedelai Menggunakan Metode Teorema *Bayes* meliputi *Business Process Diagram*, *Use Case Diagram*, *Use Case Skenario*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dan *Entity Relationship Diagram*.

4.2.1 Business Process Diagram

Business Process diagram dapat menggambarkan kebutuhan sistem secara umum yang diwakili oleh beberapa komponen yang terdiri dari:

- Input*: data yang dibutuhkan dan dimasukkan ke dalam sistem
- Output*: data yang dihasilkan oleh sistem
- Goal*: tujuan dikembangkannya sistem
- Uses*: platform yang menjadi basis sistem
- Process*: sistem yang bekerja

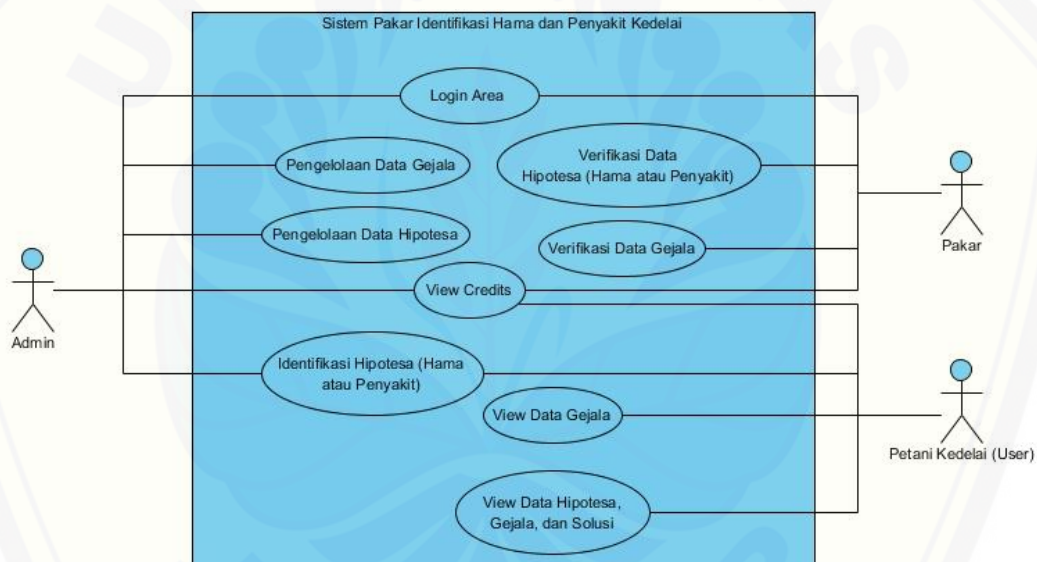


Gambar 4.1 Business Process Diagram Sistem Pakar Identifikasi Hama dan Penyakit Kedelai

Gambar 4.1 merupakan *business process* dari sistem pakar identifikasi hama dan penyakit pada tanaman kedelai menggunakan teorema *bayes* yang berbasis android dalam pengembangannya. Business process menjelaskan proses *input*, *output*, *goal*, dan *uses* yang diaplikasikan pada sistem.

4.2.2 Use Case Diagram

Use Case Diagram merupakan suatu pemodelan yang digunakan untuk menggambarkan interaksi antara aktor dengan sistem yang dibangun. *Use Case* diagram dapat menunjukkan interaksi yang dapat dilakukan di dalam sistem dengan memperhatikan hak akses pada tiap-tiap aktor. *Use Case Diagram* pada sistem yang dikembangkan ini ditunjukkan pada gambar 4.2.



Gambar 4.2 *Use Case Diagram* Sistem Pakar Identifikasi Hama atau Penyakit pada Tanaman Kedelai

Gambar 4.2 menunjukkan *Use Case* diagram pada sistem pakar identifikasi hama atau penyakit pada tanaman kedelai menggunakan teorema *bayes* yang terdiri dari 3 Aktor dan 9 *use case*. Berdasarkan *Use Case* diagram pada gambar tersebut, terdapat 3 Aktor yaitu Admin, Petani Kedelai sebagai *User*, dan Pakar. Deskripsi dari masing-masing Aktor tersebut dapat dilihat pada Tabel 4.1.

Tabel 4.1 Deskripsi Aktor

No.	Aktor	Deskripsi
1.	Admin	Aktor Admin merupakan aktor yang memiliki hak akses utama untuk mengatur data pada sistem secara penuh. Aktor ini dapat melakukan <i>login Admin Area</i> , pengelolaan Data Gejala, pngelolaan Data Hipotesa (Data Hama dan Penyakit), identifikasi Hipotesa, dan melihat <i>Credits</i> .
2.	User (Petani Kedelai)	Aktor User (Petani Kedelai) merupakan aktor yang memiliki hak akses dasar yang digunakan untuk menggunakan fitur dasar yang terdapat pada aplikasi diantaranya yaitu, mengidentifikasi Hipotesa (Hama dan Penyakit) dengan menggunakan data gejala sebagai inputan, melihat Daftar Gejala, melihat daftar Hipotesa (Hama dan Penyakit), dan melihat <i>Credits</i> .
3.	Pakar	Aktor Pakar merupakan aktor yang memiliki hak akses yang memiliki otoritas untuk memverifikasi Data Hama dan Penyakit yang diinputkan oleh Admin yang merupakan Data utama untuk mengidentifikasi suatu hama dan Penyakit pada tanaman Kedelai.

Selain memiliki 3 aktor, dalam *Use Case Diagram* juga terdapat 9 *Use Case*. Deskripsi dari 9 *Use Case* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Deskripsi *Use Case*

No.	Use Case	Deskripsi
1.	Pengelolaan Data Gejala	<i>Use Case</i> yang menggambarkan tentang proses pengelolaan gejala diantaranya menambahkan data gejala dan menghapus data gejala. <i>Use Case</i> ini hanya dapat diakses oleh Hak Akses Admin.
2.	Pengelolaan Data Hipotesa (Hama atau Penyakit)	<i>Use Case</i> yang menggambarkan tentang proses pengelolaan Hama dan Penyakit diantaranya yaitu menambahkan data Hama dan Penyakit dan menghapus data Hama dan Penyakit. <i>Use Case</i> ini hanya dapat diakses oleh Hak Akses Admin.
3.	View Data Gejala	<i>Use Case</i> yang menggambarkan proses untuk menampilkan data-data gejala pada tanaman

No.	Use Case	Deskripsi
		kedelai yang dapat dilihat oleh hak akses admin maupun <i>user</i> petani.
4.	View Data Hipotesa (Hama atau Penyakit), Gejala, dan Solusi	<i>Use Case</i> yang menggambarkan proses untuk menampilkan data Hama dan Penyakit beserta Gejala dan Solusi pada suatu hipotesa yang dapat dilihat oleh hak akses <i>user</i> petani
5.	Identifikasi Hipotesa (Hama atau Penyakit)	<i>Use Case</i> yang menggambarkan tentang proses pengidentifikasian Hama dan Penyakit tanaman kedelai. <i>Use Case</i> ini dapat digunakan oleh hak akses admin dan diperuntukkan secara khusus untuk hak akses <i>user</i> petani.
6.	Login Area	<i>Use Case</i> yang menggambarkan tentang proses autentifikasi hak akses yang diperuntukkan kepada hak akses admin dan hak akses pakar agar dapat menggunakan fitur-fitur pengelolaan data gejala maupun data hama dan penyakit.
7.	View Credits	<i>Use Case</i> yang menggambarkan tentang proses untuk menampilkan data atau informasi tentang pengembang sistem. <i>Use Case</i> ini dapat diakses oleh admin, pakar, maupun <i>user</i> petani.
8.	Verifikasi Data Gejala	<i>Use Case</i> yang menggambarkan tentang aktifitas dari hak akses pakar untuk memverifikasi Data Gejala yang telah diinputkan oleh Admin kedalam sistem
9.	Verifikasi Data Hama dan Penyakit	<i>Use Case</i> yang menggambarkan tentang aktifitas dari hak akses pakar untuk memverifikasi Data Hipotesa yang telah diinputkan oleh Admin kedalam sistem

4.2.3 Use Case Skenario

Use Case Skenario merupakan dokumentasi sistem informasi yang menjelaskan alur kebutuhan fungsional sistem yang didalamnya menggambarkan aksi antara aktor dan sistem. *Use Case* Skenario Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit pada tanaman Kedelai sebagai berikut:

a. *Use Case* Skenario Pengelolaan Data Gejala

Use Case Skenario Pengelolaan Data Gejala merupakan penjelasan urutan aksi Aktor untuk pengelolaan data gejala meliputi view, tambah, dan delete gejala.

Reaksi sistem pada skenario normal dan skenario alternatif untuk pengelolaan data gejala terdapat pada Tabel 4.3.

Tabel 4.3 *Use Case* Skenario Pengelolaan Data Gejala

Name	Pengelolaan Data Gejala
Actor	Admin
Entry Condition	Aktor ingin mengelola Data Gejala
Exit Condition	Aktor berhasil mengelola Data Gejala
Skenario Utama “View Data Gejala”	
Aksi Aktor	Aksi Sistem
1. Menekan tombol menu “Gejala” pada <i>view</i> menu utama	2. Mengambil data nama gejala dan status verifikasi dari tabel gejala <i>database</i> kedelai
	3. Menampilkan <i>view list</i> Gejala yang terdiri data nama gejala, status verifikasi, tombol logout dan tombol “Tambah Gejala”
Skenario Utama “Tambah Data Gejala”	
4. Menekan tombol “Tambah Gejala” pada <i>view list</i> gejala	5. Menampilkan form tambah gejala yang berisi <i>textfield</i> gejala dan tombol <i>submit</i>
6. Mengisi <i>textfield</i> gejala	
7. Menekan tombol <i>submit</i>	8. Menyimpan data nama gejala kedalam tabel gejala pada <i>database</i> kedelai dan menampilkan Alert “Input data gejala berhasil!”
	9. Mengambil data gejala dari tabel gejala <i>database</i> kedelai
	10. Menampilkan <i>view list</i> Gejala yang terdiri data nama gejala, status verifikasi, tombol logout dan tombol “Tambah Gejala”
Skenario Alternatif “Tambah Data Gejala” form tidak lengkap	
6a. Tidak mengisi <i>textfield</i> gejala	
7a. Menekan tombol <i>Submit</i>	8a. Menampilkan <i>Alert</i> "Textfield gejala tidak boleh kosong!"
Skenario Alternatif “Tambah Data Gejala” data gejala sudah tersedia	
6b. Mengisi <i>textfield</i> gejala	

7b. Menekan tombol <i>Submit</i>	8b. Mengecek inputan <i>id_gejala</i> dengan <i>id_gejala</i> pada tabel <i>gejala</i> dari <i>database</i> kedelai
	9b. Menampilkan <i>Alert</i> “Data gejala sudah tersedia”
Skenario Utama “Delete Data Gejala”	
1. Menekan tombol menu “Gejala” pada <i>view</i> menu utama	2. Mengambil data nama gejala dan status verifikasi dari tabel <i>gejala database</i> kedelai
	3. Menampilkan <i>view list</i> Gejala yang terdiri data nama gejala, status verifikasi, tombol logout dan tombol “Tambah Gejala”
4. Memilih salah satu gejala pada <i>view list</i> gejala	5. Menampilkan <i>view</i> Dialog Opsi dan tombol <i>delete</i>
6. Menekan tombol <i>delete</i>	7. Menghapus data gejala pada tabel <i>gejala</i> dari <i>database</i> kedelai sesuai <i>id_gejala</i> yang dipilih aktor
	8. Menampilkan <i>Alert</i> “Gejala dihapus!”

b. *Use Case* Skenario Pengelolaan Data Hipotesa

Use Case Skenario Pengelolaan Data Hipotesa merupakan penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dalam melakukan pengelolaan data hipotesa yang terdiri dari hama dan penyakit kedelai. Detail mengenai *use case* skenario ini terdapat pada lampiran A.1.

c. *Use Case* Skenario Identifikasi Hama atau Penyakit Kedelai

Use case skenario Identifikasi Hama atau Penyakit Kedelai merupakan penjelasan urutan aksi antara aktor dan reaksi sistem pada skenario normal dalam menggunakan fitur identifikasi guna menghasilkan hasil identifikasi dari perhitungan teorema bayes. Detail mengenai *use case* skenario ini terdapat pada tabel 4.4.

Tabel 4.4 *Use Case* Skenario Identifikasi Hama dan Penyakit Kedelai

Name	Identifikasi Hipotesa (Hama atau Penyakit)
Actor	User
Entry Condition	Aktor ingin mengidentifikasi Hipotesa pada tanaman kedelai
Exit Condition	Aktor melihat <i>view</i> hasil identifikasi beserta data <i>list</i> gejala dan saran penanggulangannya
Skenario Utama “Aktor Mengidentifikasi Hama dan Penyakit pada Tanaman Kedelai”	
Aksi Aktor	Aksi Sistem
1. Menekan tombol menu “Identifikasi” pada <i>view</i> menu utama	2. Mengambil data gejala dari tabel gejala <i>database</i> kedelai
	3. Menampilkan <i>view</i> Identifikasi yang terdiri dari <i>checkbox list</i> gejala, dan tombol mulai
4. <i>Checklist checkbox</i> gejala pada <i>view</i> Identifikasi disesuaikan dengan gejala yang nampak pada tanaman	
5. Menekan tombol mulai	6. Menampilkan hasil identifikasi yang terdiri dari hipotesa, gambar hipotesa, <i>list</i> gejala dan saran penanggulangannya

d. *Use Case* Skenario Login Area

Use Case Skenario Login Area merupakan penjelasan urutan aksi aktor dengan reaksi sistem pada skenario normal dan skenario alternatif dengan tujuan untuk dapat mendapat otoritas sebagai hak akses dalam menggunakan fitur-fitur yang telah ditentukan. Detail mengenai use case skenario ini terdapat pada lampiran A.2.

e. *Use Case* Skenario View Credits

Use Case Skenario View Credits merupakan penjelasan urutan aksi aktor dengan reaksi sistem pada skenario normal dengan tujuan untuk dapat melihat

informasi mengenai penulis dan kreditor. Detail mengenai use case skenario ini terdapat pada lampiran A.3.

f. *Use Case* Verifikasi Data Gejala

Use Case Skenario Verifikasi Data Gejala merupakan penjelasan urutan aksi aktor dengan reaksi sistem pada skenario normal yang bertujuan untuk dapat memverifikasi data gejala yang sebelumnya telah diinputkan oleh admin untuk ditinjau kembali data tersebut. Detail mengenai use case skenario ini terdapat pada lampiran A.4.

g. *Use Case* Verifikasi Data Hipotesa

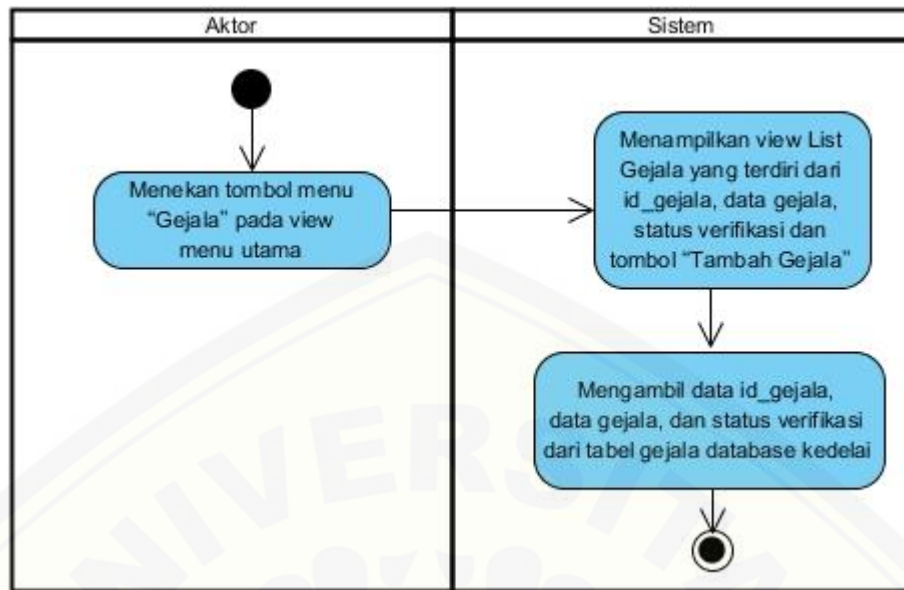
Use Case Skenario Verifikasi Data Hipotesa merupakan penjelasan urutan aksi aktor dengan reaksi sistem pada skenario normal dengan tujuan untuk dapat memverifikasi data hipotesa yang sebelumnya telah diinputkan oleh admin untuk ditinjau kembali detail dari data tersebut. Detail mengenai use case skenario ini terdapat pada lampiran A.5.

4.2.4 *Activity* Diagram

Activity Diagram merupakan dokumentasi desain sistem yang menggambarkan aliran aktifitas yang dilakukan antara aktor dan sistem pada Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit pada Tanaman Kedelai Menggunakan Metode Teorema Bayes. Sistem pakar ini memiliki *activity* diagram diantaranya sebagai berikut:

a. *Activity* Diagram Pengelolaan Data Gejala (View)

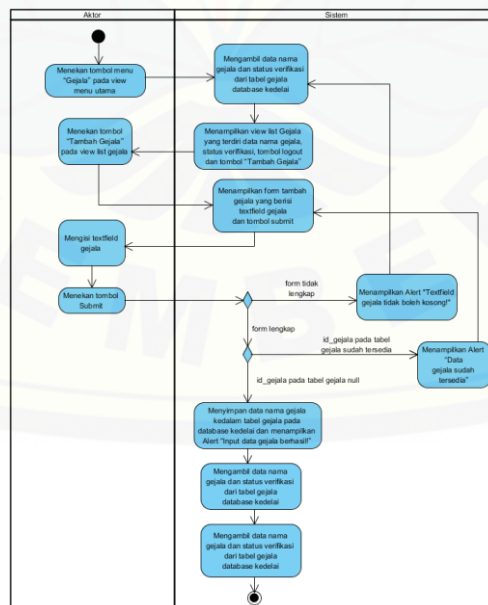
Activity Diagram Pengelolaan Data Gejala (View) merupakan aliran aktifitas yang dilakukan aktor dan sistem yang bertujuan untuk menampilkan data gejala. Adapun hak akses yang dapat menggunakan fitur ini diantaranya user, admin, dan pakar. Detail aliran aktifitas pada *activity* ini terdapat pada Gambar 4.3.



Gambar 4.3 Activity Diagram Pengelolaan Data Gejala (View)

b. Activity Diagram Pengelolaan Data Gejala (Tambah)

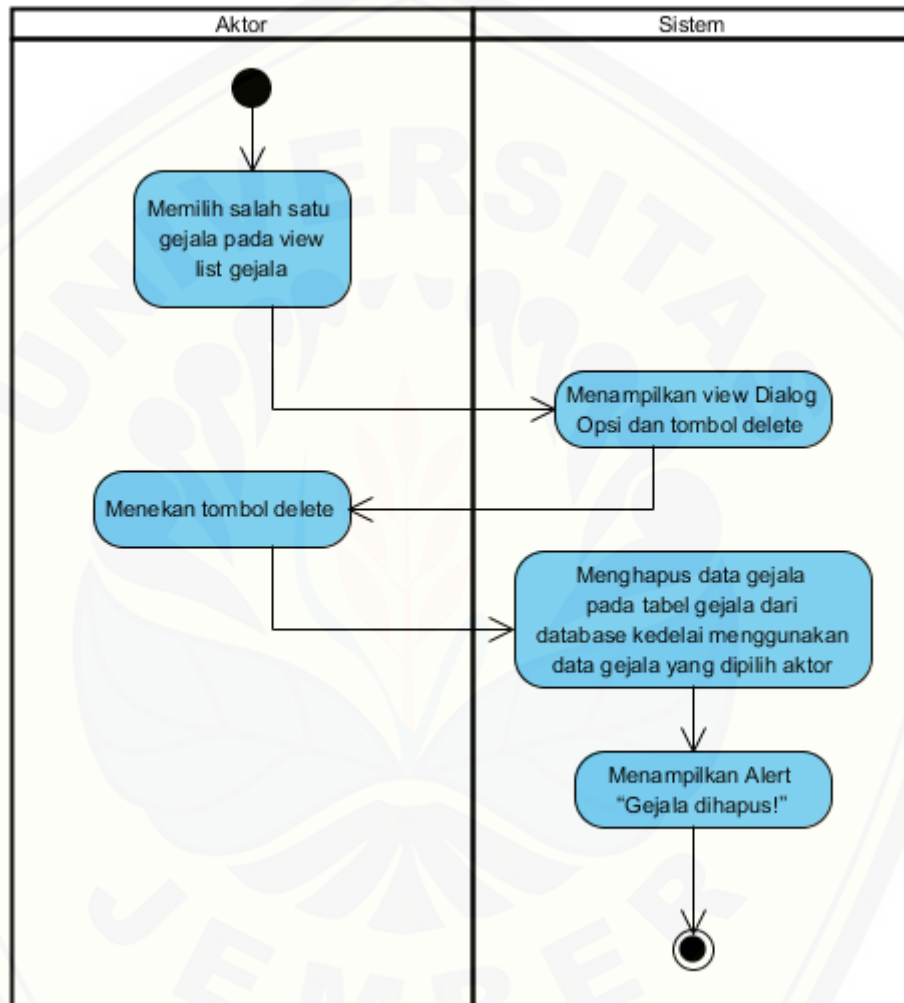
Activity Diagram Pengelolaan Data Gejala (Tambah) merupakan aliran aktifitas yang dilakukan aktor dan sistem yang bertujuan untuk menginputkan data gejala. Adapun hak akses yang dapat menggunakan fitur ini adalah Admin. Detail aliran aktifitas pada activity ini terdapat pada Gambar 4.4.



Gambar 4.4 Activity Diagram Pengelolaan Data Gejala (Tambah)

c. *Activity Diagram* Pengelolaan Data Gejala (Delete)

Activity Diagram Pengelolaan Data Gejala (Delete) merupakan aliran aktifitas yang dilakukan aktor dan sistem yang bertujuan menghapus data gejala. Adapun hak akses yang dapat menggunakan fitur ini adalah Admin. Detail aliran aktifitas pada *activity* ini terdapat pada Gambar 4.5.



Gambar 4.5 *Activity Diagram* Pengelolaan Data Gejala (Delete)

d. *Activity Diagram* Pengelolaan Data Hipotesa (View)

Activity Diagram Pengelolaan Data Hipotesa (View) merupakan aliran aktifitas yang dilakukan aktor dan sistem yang bertujuan menampilkan data hipotesa, gejala yang berkaitan dengan hipotesa, gambar hipotesa, dan solusi yang dapat dilakukan untuk penganggulangan. Adapun hak akses yang dapat

menggunakan fitur ini adalah Admin, User, dan Pakar. Detail aliran aktifitas pada *activity* ini terdapat pada Lampiran B.1.

e. *Activity* Diagram Pengelolaan Data Hipotesa (Tambah)

Activity Diagram Pengelolaan Data Hipotesa (Tambah) merupakan aliran aktifitas yang dilakukan aktor dan sistem yang bertujuan menginputkan data hipotesa, gejala yang berkaitan dengan hipotesa, gambar hipotesa, dan solusi yang dapat dilakukan untuk penganggulangan. Adapun hak akses yang dapat menggunakan fitur ini adalah Admin. Detail aliran aktifitas pada *activity* ini terdapat pada Lampiran B.2.

f. *Activity* Diagram Pengelolaan Data Hipotesa (Delete)

Activity Diagram Pengelolaan Data Hipotesa (Delete) merupakan aliran aktifitas yang dilakukan aktor dan sistem yang bertujuan menghapus data hipotesa. Adapun hak akses yang dapat menggunakan fitur ini adalah Admin. Detail aliran aktifitas pada *activity* ini terdapat pada Lampiran B.3.

g. *Activity* Diagram Pengelolaan Data Hipotesa (View Detail salah satu Hipotesa)

Activity Diagram Pengelolaan Data Hipotesa (View Detail salah satu Hipotesa) merupakan aliran aktifitas yang dilakukan aktor dan sistem yang bertujuan menampilkan detail hipotesa diantaranya gejala yang berkaitan dengan hipotesa, gambar hipotesa, dan solusi yang dapat dilakukan sebagai saran penganggulangan. Adapun hak akses yang dapat menggunakan fitur ini adalah Admin. Detail aliran aktifitas pada *activity* ini terdapat pada Lampiran B.4.

h. *Activity* Diagram Pengelolaan Data Hipotesa (Tambah Data Gejala pada salah satu Hipotesa)

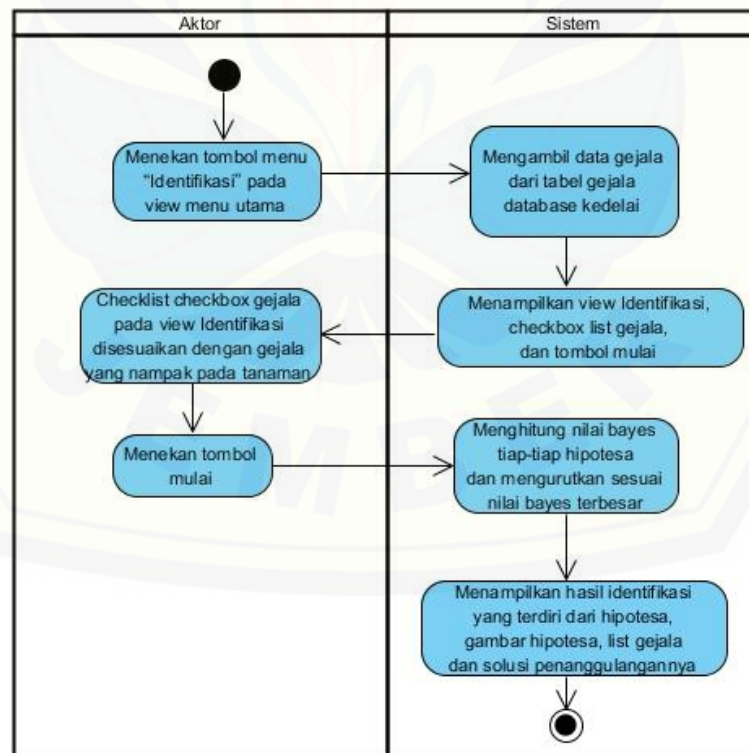
Activity Diagram Pengelolaan Data Hipotesa (Tambah Data Gejala pada salah satu Hipotesa) merupakan aliran aktifitas yang dilakukan aktor dan sistem yang bertujuan menginputkan data gejala yang berkaitan dengan hipotesa. Adapun hak akses yang dapat menggunakan fitur ini adalah Admin. Detail aliran aktifitas pada *activity* ini terdapat pada Lampiran B.5.

i. *Activity Diagram* Pengelolaan Data Hipotesa (Delete Gejala pada salah satu Hipotesa)

Activity Diagram Pengelolaan Data Hipotesa (Delete Gejala pada salah satu Hipotesa) merupakan aliran aktifitas yang dilakukan aktor dan sistem yang bertujuan menghapus data gejala yang berkaitan dengan hipotesa. Adapun hak akses yang dapat menggunakan fitur ini adalah Admin. Detail aliran aktifitas pada *activity* ini terdapat pada Lampiran B.6.

j. *Activity Diagram* Identifikasi Hama atau Penyakit

Activity Diagram Identifikasi Hama atau Penyakit menggambarkan aliran aktifitas yang dilakukan oleh Admin atau User untuk menjalankan fitur identifikasi hama atau penyakit dengan reaksi dari sistem berupa keluaran nilai bayes terbesar sebagai hasil identifikasi beserta solusi atau saran penanggulangan yang dapat dilakukan terhadap hama atau penyakit tersebut. Untuk mengetahui aliran aktifitas pada *Activity Diagram* Identifikasi Hama atau Penyakit dapat dilihat pada gambar 4.6.



Gambar 4.6 *Activity Diagram* Identifikasi Hama atau Penyakit

k. *Activity Diagram Login Area*

Activity Diagram Login Area menggambarkan aliran aktifitas yang dilakukan oleh Aktor baik admin maupun pakar untuk menjalankan fitur login hingga mengakses sistem dengan hak akses yang sesuai untuk dapat melakukan aktifitas lainnya sebagai hak akses yang telah ditentukan. *Activity Diagram Login Area* juga menggambarkan reaksi sistem terhadap Aktor ketika menjalankan fitur Login Area. Untuk mengetahui aliran aktifitas pada *Activity Diagram Login Area* dapat dilihat pada Lampiran B.7.

l. *Activity Diagram View Credits*

Activity Diagram View Credits menggambarkan aliran aktifitas yang dilakukan oleh Aktor baik admin, user, maupun pakar untuk menampilkan halaman yang berisi informasi penulis dan pengembang sistem. Detail aliran aktifitas pada *Activity View Credits* dapat dilihat pada Lampiran B.8.

m. *Activity Diagram Verifikasi Data Gejala*

Activity Diagram Verifikasi Data Gejala merupakan aliran aktifitas yang dilakukan aktor dan sistem yang bertujuan untuk memverifikasi data gejala pada sistem yang telah diinputkan oleh admin. Adapun hak akses yang dapat menggunakan fitur ini adalah Pakar. Detail aliran aktifitas pada *activity* ini terdapat pada Lampiran B.9.

n. *Activity Diagram Verifikasi Data Hipotesa*

Activity Diagram Verifikasi Data Hipotesa merupakan aliran aktifitas yang dilakukan aktor dan sistem yang bertujuan untuk memverifikasi data hipotesa pada sistem yang telah diinputkan oleh admin. Adapun hak akses yang dapat menggunakan fitur ini adalah Pakar. Detail aliran aktifitas pada *activity* ini terdapat pada Lampiran B.10

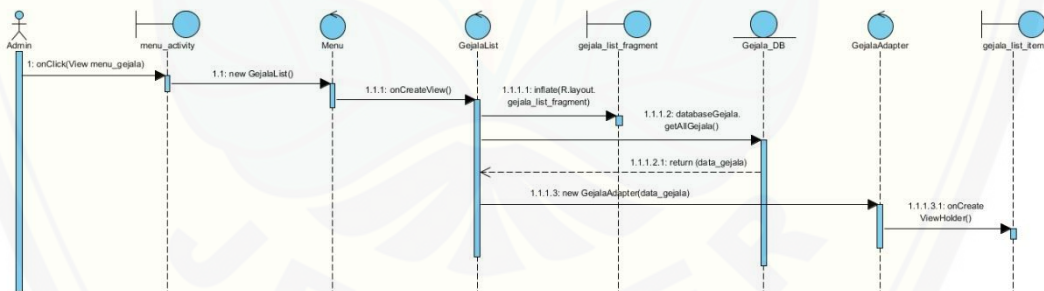
4.2.5 *Sequence Diagram*

Sequence diagram merupakan dokumentasi sistem berbentuk diagram berurut yang menggambarkan interaksi aktor dan objek dalam sistem. *Sequence* diagram digunakan untuk menggambarkan aliran sistem pada usecase skenario dan aliran logika yang diterapkan pada sistem. Interaksi antar objek pada *sequence*

diagram tersebut meliputi aktor yang berinteraksi dengan sistem, *view* berfungsi menampilkan data, *controller* yang berfungsi mengatur jalannya data, *function* atau *method* yang bertugas menangani data tertentu, dan model untuk berinteraksi dengan database. *Sequence* diagram yang akan dibahas pada bagian ini antara lain *sequence* diagram pengelolaan data gejala (*view*), *sequence* diagram pengelolaan data gejala (tambah), *sequence* diagram pengelolaan data gejala (*delete*), dan *sequence* diagram identifikasi hama penyakit kedelai. Detail *sequence* diagram yang lain dapat dilihat pada lampiran C.

a. *Sequence* Diagram Pengelolaan Data Gejala (*View*)

Sequence Diagram ini merupakan urutan langkah-langkah yang dilakukan antar objek sebagai suatu respon atas kejadian/*event* didalam sistem yang bertujuan untuk menampilkan data gejala. Aktor yang berperan dalam *sequence* diagram ini adalah user, admin, dan pakar. Pada *sequence* ini terdapat class *menu_activity*, *gejala_list_fragment*, *gejala_list_item*, *Menu*, *GejalaList*, *GejalaAdapter*, *Gejala_DB* serta berbagai *method* yang dibutuhkan dalam proses ini. Detail *sequence* diagram dapat dilihat pada Gambar 4.7.

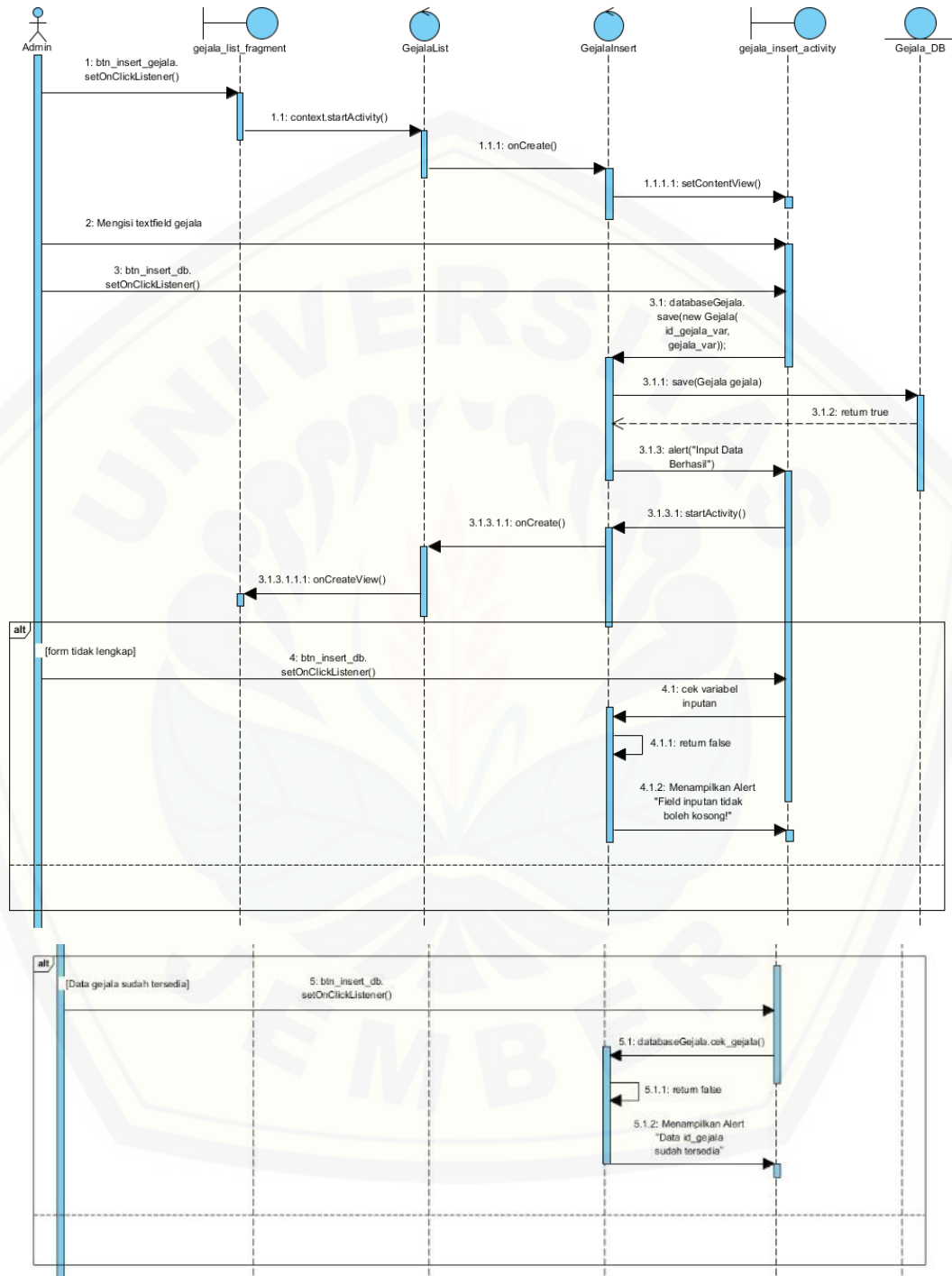


Gambar 4.7 *Sequence* Diagram Pengelolaan Data Gejala (*View*)

b. *Sequence* Diagram Pengelolaan Data Gejala (Tambah)

Sequence Diagram ini merupakan urutan langkah-langkah yang dilakukan antar objek sebagai suatu respon atas kejadian/*event* didalam sistem yang bertujuan untuk menginputkan data gejala. Aktor yang berperan dalam *sequence* diagram ini adalah admin. Pada *sequence* ini terdapat class *gejala_list_fragment*, *gejala_insert_activity*, *GejalaList*, *GejalaInsert*, *Gejala_DB*, serta berbagai *method*

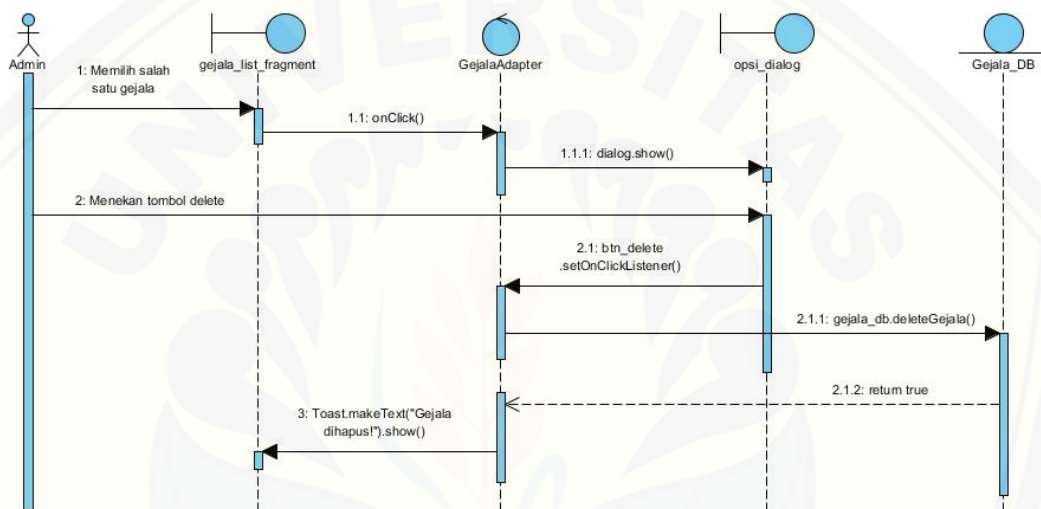
yang dibutuhkan dalam proses ini. Detail *sequence* diagram dapat dilihat pada Gambar 4.8.



Gambar 4.8 Sequence Diagram Pengelolaan Data Gejala (Tambah)

c. *Sequence* Diagram Pengelolaan Data Gejala (Delete)

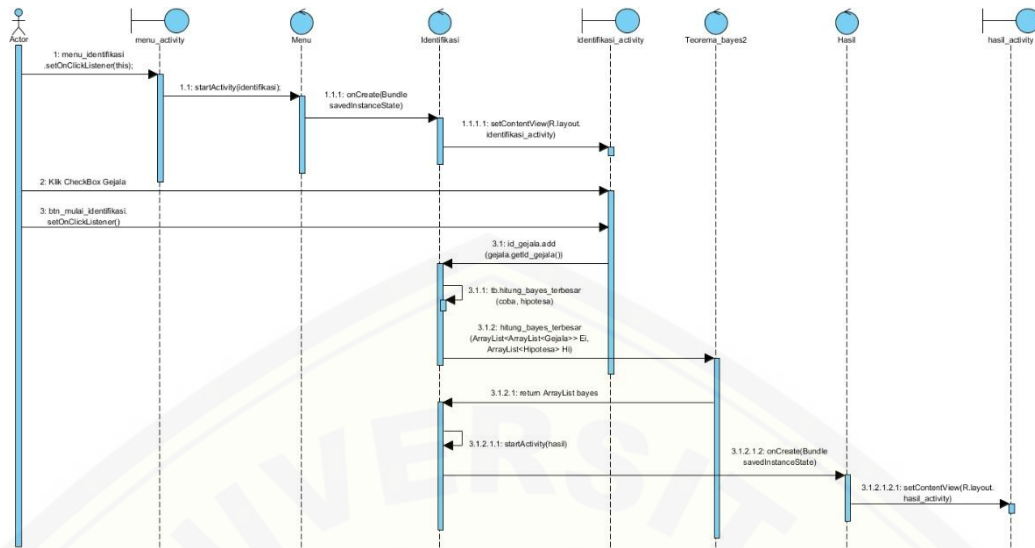
Sequence Diagram ini merupakan urutan langkah-langkah yang dilakukan antar objek sebagai suatu respon atas kejadian/*event* didalam sistem yang bertujuan untuk menghapus data gejala. Aktor yang berperan dalam *sequence* diagram ini adalah admin. Pada *sequence* ini terdapat class gejala_list_fragment, opsi_dialog, GejalaAdapter, Gejala_DB, serta berbagai method yang dibutuhkan dalam proses ini. Detail *sequence* diagram dapat dilihat pada Gambar 4.9.



Gambar 4.9 *Sequence* Diagram Pengelolaan Data Gejala (Delete)

d. *Sequence* Diagram Identifikasi Hama dan Penyakit

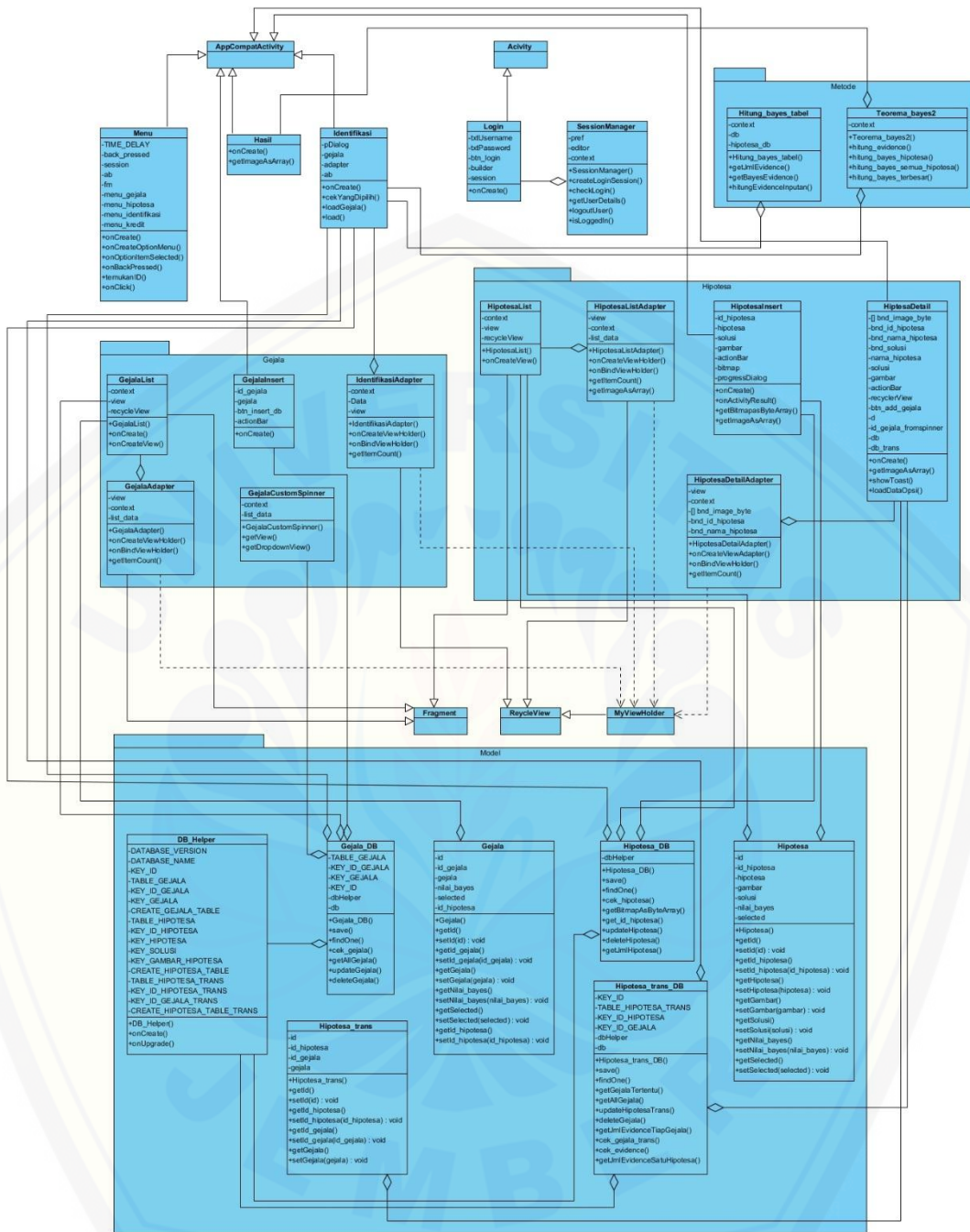
Sequence Diagram ini menggambarkan skenario atau rangkaian langkah yang dilakukan sebagai respon atas kejadian/*event* didalam sistem dengan tujuan untuk mengidentifikasi hama atau penyakit pada tanaman kedelai dengan memasukkan gejala-gejala yang disesuaikan dengan gejala yang terjadi pada tanaman. Aktor yang terdapat pada *sequence* ini adalah user, admin, dan pakar. Pada *sequence* ini terdapat class menu activity, identifikasi_activity, hasil_activity, Menu, Identifikasi, Teorema_bayes2, Hasil, dan berbagai method yang diperlukan dalam proses ini. Detail *Sequence* Diagram Identifikasi Hama dan Penyakit dapat dilihat pada gambar 4.10.



Gambar 4.10 *Sequence Diagram* Identifikasi Hama dan Penyakit

4.2.6 *Class Diagram*

Class diagram menggambarkan hubungan antar kelas yang digunakan untuk membangun suatu sistem, kelas yang digambarkan berisi atribut dan nama method pada setiap kelas yang digunakan didalamnya. Berdasarkan *sequence diagram* yang dibangun *class diagram* ini terdiri dari 17 class backend, 9 class model, dan 21 class yang diwakili oleh class *AppCompatActivity*, *Activity*, dan *Fragment* yang berfungsi sebagai penampil atau *unit interface*. *Class diagram* Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit Kedelai menggunakan Teorema *Bayes* dapat dilihat pada Tabel 4.11.

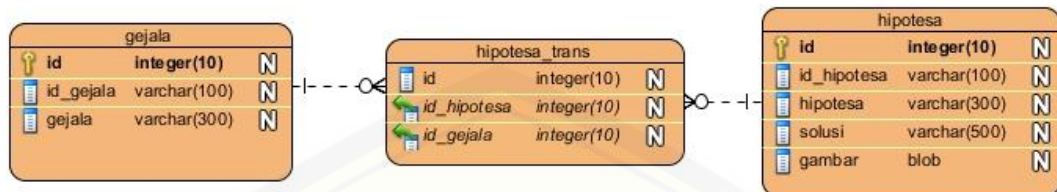


Gambar 4.11 Class Diagram Aplikasi Identifikasi Hama dan Penyakit Kedelai

4.2.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) pada Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit Kedelai menggunakan Teorema Bayes merupakan gambaran dari model hubungan antar data atribut dalam media penyimpanan database yang

dikelompokkan berdasarkan entitas-entitas yang memiliki atribut yang dibutuhkan. ERD pada aplikasi ini dapat dilihat pada Gambar 4.12.



Gambar 4.12 *Entitiy Relationship* Diagram Aplikasi Identifikasi Hama dan Penyakit Kedelai

4.3 Penulisan Kode Program

Desain sistem yang telah dirancang akan diimplementasikan kedalam kode program. Beberapa tahap implementasi kedalam kode program diantaranya yaitu:

- Penulisan kode program atau *Coding* menggunakan bahasa pemrograman *Java* sebagai *Backend* Program dan *XML* sebagai *Unit Interface* atau tampilan Program.
- Manajemen Basisdata menggunakan *SQLite*.

Kode Program perhitungan metode Teorema *Bayes* terdapat pada *class* *Teorema_bayes* pada *package* *Metode*, *class* *Hitung_bayes_tabel* pada *package* *Metode*, dan *class* *Hasil* sebagai *class* utama untuk menampilkan hasil kepada aktor. Kode program perhitungan metode Teorema *Bayes* tersebut terdapat pada Tabel 4.5, Tabel 4.6, dan Tabel 4.7.

Tabel 4.5 Kode Program *Class* *Teorema_bayes*

```

File:
E:\AndroidStudioProjects\Kedelai\app\src\main\java\com\project\hofi\kedelai\Metode\Teorema_bayes.java
01: package com.project.hofi.kedelai.Metode;
02:
03: import com.project.hofi.kedelai.Model.Gejala;
04: import com.project.hofi.kedelai.Model.Hipotesa;
05:
06: import java.lang.reflect.Array;
07: import java.util.ArrayList;
08:
09: /**
10:  * Created by hofio on 16/10/2017.
11:  */
12:
  
```

```
13: public class Teorema_bayes {
14:     public ArrayList<Hipotesa> hitung_bayes(ArrayList<ArrayList<Gejala>> Ei,
ArrayList<Hipotesa> Hi){
15:         //menghitung nilai Hi tanpa memandang evidence
16:         double one=1;
17:         double pHi=one/Hi.size();
18:
19:         //menghitung pembilang
20:         ArrayList<ArrayList<Hipotesa>> pembilang=new ArrayList<>();
21:         for (int i = 0; i < Ei.size(); i++) {
22:             ArrayList<Hipotesa> per_hipotesa=new ArrayList<>();
23:             for (int j = 0; j < Ei.get(i).size(); j++) {
24:                 Hipotesa hipotesa=new Hipotesa();
25:                 hipotesa.setPembilang(Ei.get(i).get(j).getNilai_bayes()*pHi);
26:                 per_hipotesa.add(hipotesa);
27:             }
28:             pembilang.add(per_hipotesa);
29:         }
30:
31:         //menghitung penyebut
32:         ArrayList<Hipotesa> penyebut=new ArrayList<>();
33:
34:         for (int i = 0; i < pembilang.get(0).size(); i++) {
35:             Hipotesa hipotesa=new Hipotesa();
36:             double temp=0;
37:             for (int j = 0; j < pembilang.size(); j++) {
38:                 temp +=pembilang.get(j).get(i).getPembilang();
39:             }
40:             hipotesa.setPenyebut(temp);
41:             penyebut.add(hipotesa);
42:         }
43:
44:         //menghitung bayes hipotesa
45:         ArrayList<ArrayList<Hipotesa>> bayes_hipotesa=new ArrayList<>();
46:         for (int i = 0; i < pembilang.size(); i++) {
47:             ArrayList<Hipotesa> per_gejala=new ArrayList<>();
48:             for (int j = 0; j < pembilang.get(i).size(); j++) {
49:                 Hipotesa hipotesa=new Hipotesa();
50:                 hipotesa.setNilai_bayes(pembilang.get(i).get(j).getPembilang()/penyebut.get(j).getPenyebut());
51:                 per_gejala.add(hipotesa);
52:             }
53:             bayes_hipotesa.add(per_gejala);
54:         }
55:
56:         //menghitung total nilai bayes hipotesa
57:         ArrayList<Hipotesa> tot_bayes_hipotesa=new ArrayList<>();
58:         for (int i = 0; i < bayes_hipotesa.size(); i++) {
59:             double total=0;
60:             Hipotesa hipotesa=new Hipotesa();
61:             for (int j = 0; j < bayes_hipotesa.get(i).size(); j++) {
62:                 total += bayes_hipotesa.get(i).get(j).getNilai_bayes();
63:             }
64:             hipotesa.setNilai_bayes(total/bayes_hipotesa.get(i).size());
65:             hipotesa.setId_hipotesa(Hi.get(i).getId_hipotesa());
```

```

66:     hipotesa.setHipotesa(Hi.get(i).getHipotesa());
67:     tot_bayes_hipotesa.add(hipotesa);
68: }
69:
70: //mengurutkan nilai bayes dg bubble sort terbesar ke terkecil
71: ArrayList<Hipotesa> temp = new ArrayList<>();
72: for (int i = 0; i < tot_bayes_hipotesa.size(); i++) {
73:     for (int j = 0; j < tot_bayes_hipotesa.size() - 1; j++) {
74:         if (tot_bayes_hipotesa.get(j).getNilai_bayes() < tot_bayes_hipotesa.get(j +
1).getNilai_bayes()) {
75:             temp.add(tot_bayes_hipotesa.get(j + 1));
76:             tot_bayes_hipotesa.set(j + 1, tot_bayes_hipotesa.get(j));
77:             tot_bayes_hipotesa.set(j, temp.get(0));
78:         }
79:         temp.clear();
80:     }
81: }
82: return tot_bayes_hipotesa;
83: }
84: }
85:

```

Tabel 4.6 Kode Program *Class* Hitung_bayes_tabel

```

01: package com.project.hofi.kedelai.Metode;
02:
03: import android.content.Context;
04: import android.widget.Toast;
05:
06: import com.project.hofi.kedelai.Model.Gejala;
07: import com.project.hofi.kedelai.Model.Hipotesa;
08: import com.project.hofi.kedelai.Model.Hipotesa_DB;
09: import com.project.hofi.kedelai.Model.Hipotesa_trans_DB;
10:
11: import java.lang.reflect.Array;
12: import java.util.ArrayList;
13: import java.util.Arrays;
14:
15: /**
16:  * Created by Hofi on 05/02/2017.
17:  */
18:
19: public class Hitung_bayes_tabel {
20:
21:     /*1. Menyiapkan query kembalian nilai brp jml evidence utk tiap penyakit
22:     * 2. Menyiapkan query kembalian nilai brp jml hipotesa/penyakit*/
23:
24:     Context context;
25:     Hipotesa_trans_DB db;
26:     Hipotesa_DB hipotesa_db;
27:
28:
29:     public Hitung_bayes_tabel(Context context) {
30:         this.context = context;
31:         db = new Hipotesa_trans_DB(context);

```

```

32:     hipotesa_db = new Hipotesa_DB(context);
33:
34: }
35:
36: /*menghitung jml evidence 1 hipotesa*/
37: public int getJml_evidence(String id_hipotesa) {
38:     /*Ambil dg Query pd tabel Gejala
39:     * kembalikan nilai berupa jml_evidence per 1 penyakit*/
40:
41:     return db.getJmlEvidenceSatuHipotesa(id_hipotesa);
42: }
43:
44: /*menghitung bayes 1 evidence 1 hipotesa (bayes evidence pasti sama dlm satu
hipotesa)*/
45: public double get_bayes_evidence(String id_hipotesa){
46:     double bayes=0;
47:     double one=1;
48:     for (int i = 0; i < getJml_evidence(id_hipotesa); i++) {
49:         bayes=one/getJml_evidence(id_hipotesa);
50:     }
51:     return bayes;
52: }
53:
54: /*CARA PENGGUNAAN FUNCTION HITUNG_EVIDENCE_INPUTAN
55: * 1. Ambil kumpulan id_gejala yg telah diinputkan oleh user jadikan dalam bentuk array
56: * 2. Masukkan data tsb sebagai parameter
57: * 3. Kembalikan nilai akan berbentuk array 2 dimensi berindeks*/
58:
59: /*menghitung bayes banyak evidence 1 hipotesa*/
60: public ArrayList<Gejala> hitung_evidence_inputan(ArrayList<String> id_gejala, String
id_hipotesa) {
61:
62:     ArrayList<Gejala> gejala=new ArrayList<>();
63:
64:     for (int i = 0; i < id_gejala.size(); i++) {
65:         boolean cek=db.cek_evidence(id_gejala.get(i), id_hipotesa);
66:         Gejala objek=new Gejala();
67:         if (cek) {
68:
69:             /*hipotesa.get(i).add(bayes_evidence_semua_hipotesa.get(i));*/
70:
71:             objek.setNilai_bayes(get_bayes_evidence(id_hipotesa));
72:             objek.setId_gejala(id_gejala.get(i));
73:             objek.setId_hipotesa(id_hipotesa);
74:             gejala.add(objek);
75:
76:             /*Toast.makeText(context, "Evidence "+gejala.get(i).getId_gejala()+" pada
Hipotesa "+id_hipotesa+" = "+gejala.get(i).getNilai_bayes(),
Toast.LENGTH_LONG).show();*/
77:         } else {
78:             double zero=0;
79:             objek.setNilai_bayes(zero);
80:             objek.setId_gejala(id_gejala.get(i));
81:             objek.setId_hipotesa(id_hipotesa);
82:             gejala.add(objek);

```



```

83:         /*Toast.makeText(context, "Evidence "+gejala.get(i).getId_gejala()+" pada
Hipotesa "+id_hipotesa+" = "+gejala.get(i).getNilai_bayes(),
Toast.LENGTH_LONG).show();*/
84:     }
85: }
86:     return gejala;
87: }
88: }

```

Tabel 4.7 Kode Program *Class* Hasil

```

01: package com.project.hofi.kedelai;
02:
03: import android.content.Intent;
04: import android.graphics.Bitmap;
05: import android.graphics.BitmapFactory;
06: import android.support.design.widget.TabLayout;
07: import android.support.v4.view.ViewPager;
08: import android.support.v7.app.AppCompatActivity;
09: import android.os.Bundle;
10: import android.support.v7.widget.RecyclerView;
11: import android.view.View;
12: import android.widget.ImageView;
13: import android.widget.TextView;
14:
15: import com.project.hofi.kedelai.Model.Hipotesa;
16: import com.project.hofi.kedelai.Model.Hipotesa_DB;
17:
18: import java.util.ArrayList;
19:
20: public class Hasil extends AppCompatActivity {
21:
22:     @Override
23:     protected void onCreate(Bundle savedInstanceState) {
24:         super.onCreate(savedInstanceState);
25:         setContentView(R.layout.hasil_activity);
26:
27:         ArrayList<Hipotesa> hasil_identifikasi = (ArrayList<Hipotesa>)
getIntent().getSerializableExtra("hasil_identifikasi");
28:
29:         ArrayList<Hipotesa> hasil = new ArrayList<>();
30:
31:
32:
33:         hasil.add(hasil_identifikasi.get(0));
34:         for (int i = 0; i < hasil_identifikasi.size() - 1; i++) {
35:             if (hasil_identifikasi.get(0).getNilai_bayes() == hasil_identifikasi.get(i +
1).getNilai_bayes()) {
36:                 hasil.add(hasil_identifikasi.get(i + 1));
37:             }
38:         }
39:
40:         ViewPager viewPager=(ViewPager)findViewById(R.id.view_pager);
41:         HasilPagerAdapter adapter=new HasilPagerAdapter(Hasil.this, hasil);
42:         viewPager.setAdapter(adapter);

```

```
43:
44:     TabLayout tab = (TabLayout) findViewById(R.id.tab);
45:     if (hasil.size(>1) {
46:         tab.setupWithViewPager(viewPager);
47:     }else{
48:         tab.setVisibility(View.GONE);
49:     }
50: }
51:
52: }
53:
```

4.4 Pengujian Sistem

Pengujian sistem pada Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit pada tanaman Kedelai menggunakan Teorema *Bayes* merupakan proses pengujian untuk mengetahui kesesuaian antara implementasi kode program dan rancangan sistem. Pengujian Sistem dapat dilakukan dengan menggunakan beberapa metode yaitu metode *White Box*, metode *Black Box*, Uji Validitas dan Reliabilitas.

4.4.1 *White Box*

Pengujian sistem dengan menggunakan metode *White Box* merupakan suatu pengujian sistem yang dilakukan dari segi desain sistem dan kode program. Pengujian pada sistem ini dengan cara menggambar diagram alir, menghitung *Cyclomatic Complexity* (CC), dan membuat tabel pengujian *test case*. Tahap-tahap dalam pengujian sistem dengan menggunakan metode *White Box* diantaranya yaitu *Listing Program* yang ditunjukkan pada Poin a, Diagram Alir pada poin b, Perhitungan *Cyclomatic Complexity* pada poin c, dan tabel pengujian Alur *Test Case* pada poin d.

a. Listing Program Metode Teorema Bayes

Tabel 4.8 Listing program menghitung pembilang

```

19: //menghitung pembilang
20: ArrayList<ArrayList<Hipotesa>> pembilang=new ArrayList<>();
21: for (int i = 0; i < Ei.size(); i++) {
22:     ArrayList<Hipotesa> per_hipotesa=new ArrayList<>();
23:     for (int j = 0; j < Ei.get(i).size(); j++) {
24:         Hipotesa hipotesa=new Hipotesa();
25:         hipotesa.setPembilang(Ei.get(i).get(j).getNilai_bayes()*pHi);
26:         per_hipotesa.add(hipotesa);
27:     }
28:     pembilang.add(per_hipotesa);
29: }

```

Tabel 4.9 Listing program menghitung penyebut

```

31: //menghitung penyebut
32: ArrayList<Hipotesa> penyebut=new ArrayList<>();
33:
34: for (int i = 0; i < pembilang.get(0).size(); i++) {
35:     Hipotesa hipotesa=new Hipotesa();
36:     double temp=0;
37:     for (int j = 0; j < pembilang.size(); j++) {
38:         temp +=pembilang.get(j).get(i).getPembilang();
39:     }
40:     hipotesa.setPenyebut(temp);
41:     penyebut.add(hipotesa);
42: }

```

Tabel 4.10 Listing Program menghitung nilai bayes masing-masing hipotesa

```

44: //menghitung bayes hipotesa
45: ArrayList<ArrayList<Hipotesa>> bayes_hipotesa=new ArrayList<>();
46: for (int i = 0; i < pembilang.size(); i++) {
47:     ArrayList<Hipotesa> per_gejala=new ArrayList<>();
48:     for (int j = 0; j < pembilang.get(i).size(); j++) {
49:         Hipotesa hipotesa=new Hipotesa();
50:         hipotesa.setNilai_bayes(pembilang.get(i).get(j).getPembilang()/penyebut.get(j).
getPenyebut());
51:         per_gejala.add(hipotesa);
52:     }
53:     bayes_hipotesa.add(per_gejala);
54: }

```

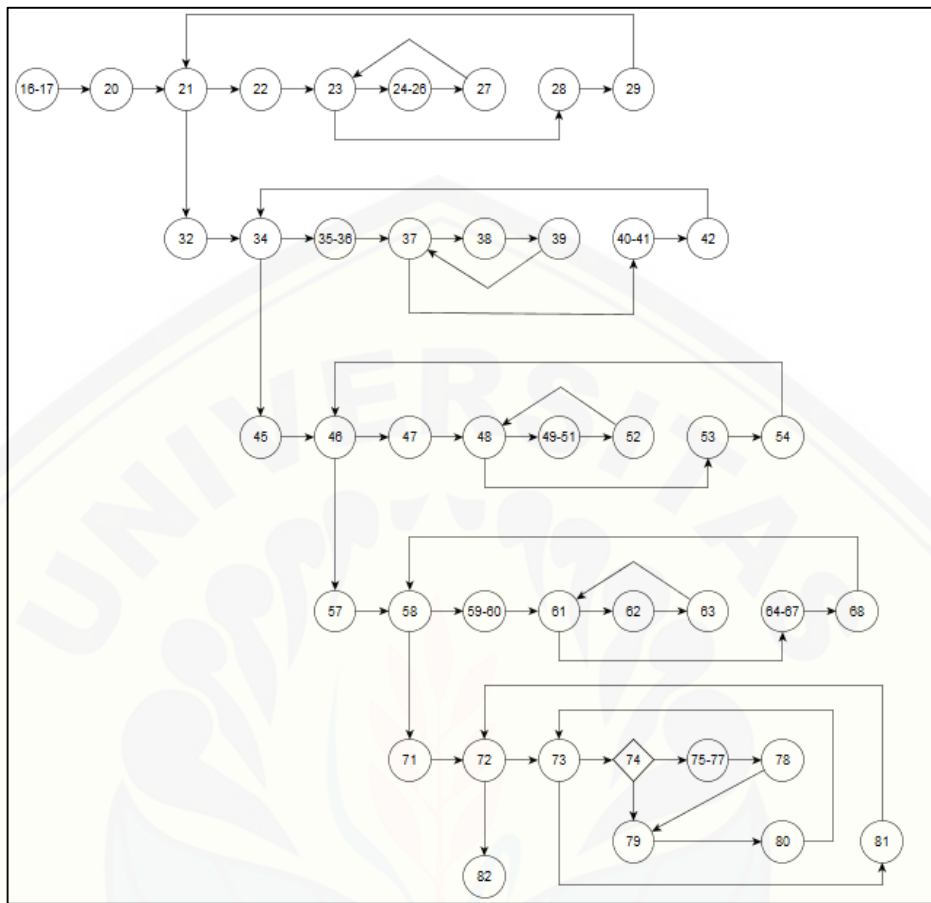
Tabel 4.11 *Listing* Program menghitung total nilai bayes

```
56: //menghitung total nilai bayes hipotesa
57: ArrayList<Hipotesa> tot_bayes_hipotesa=new ArrayList<>();
58: for (int i = 0; i < bayes_hipotesa.size(); i++) {
59:     double total=0;
60:     Hipotesa hipotesa=new Hipotesa();
61:     for (int j = 0; j < bayes_hipotesa.get(i).size(); j++) {
62:         total += bayes_hipotesa.get(i).get(j).getNilai_bayes();
63:     }
64:     //hipotesa.setNilai_bayes(total/bayes_hipotesa.get(i).size());
65:     hipotesa.setNilai_bayes(total);
66:     hipotesa.setId_hipotesa(Hi.get(i).getId_hipotesa());
67:     hipotesa.setHipotesa(Hi.get(i).getHipotesa());
68:     tot_bayes_hipotesa.add(hipotesa);
69: }
```

Tabel 4.12 *Listing* Program mengurutkan nilai bayes terbesar ke terkecil dengan menggunakan metode bubble sort

```
File:
E:\AndroidStudioProjects\Kedelai\app\src\main\java\com\project\hofi\kedelai\
Metode\Teorema_bayes.java
71: //mengurutkan nilai bayes dg bubble sort terbesar ke terkecil
72: ArrayList<Hipotesa> temp = new ArrayList<>();
73: for (int i = 0; i < tot_bayes_hipotesa.size(); i++) {
74:     for (int j = 0; j < tot_bayes_hipotesa.size() - 1; j++) {
75:         if (tot_bayes_hipotesa.get(j).getNilai_bayes() <
tot_bayes_hipotesa.get(j + 1).getNilai_bayes()) {
76:             temp.add(tot_bayes_hipotesa.get(j + 1));
77:             tot_bayes_hipotesa.set(j + 1, tot_bayes_hipotesa.get(j));
78:             tot_bayes_hipotesa.set(j, temp.get(0));
79:         }
80:         temp.clear();
81:     }
82: }
83: return tot_bayes_hipotesa;
```

b. Diagram Alir



Gambar 4.13 Diagram Alir *function* hitung_bayes

c. Perhitungan *Cyclomatic Complexity* (CC)

$$CC \text{ function hitung_bayes} = E - N + 2 = 53 - 43 + 2 = 12$$

d. Pengujian Basis Tes

Tabel 4.13 *Test Case* *function* hitung_bayes

Jalur 1	
Path / Jalur	(16 – 17) – (20) – 21 – 22 – 23 – (24 – 26) – 27 – 23 – 28 – 29 – 21 – 32 – 34 – (35 – 36) – 37 – 38 – 39 – 37 – (40 – 41) – 42 – 34 – 45 – 46 – 47 – 48 – (49 – 51) – 52 – 48 – 53 – 54 – 46 – 57 – 58 – (59 – 60) – 61 – 62 – 63 – 61 – (64 – 67) – 68 – 58 – 71 – 72 – 73 – 74 – (75 – 77) – 78 – 79 – 80 – 73 – 81 – 72 – 82
<i>Test Case</i>	Menghitung nilai bayes masing-masing hipotesa berdasarkan gejala yang diinputkan terurut berdasarkan nilai bayes terbesar ke terkecil

Target yang diharapkan	Berhasil menghitung nilai bayes masing-masing hipotesa berdasarkan gejala yang diinputkan terurut berdasarkan nilai bayes terbesar ke terkecil
Hasil Pengujian	Benar
Jalur 2	
Path / Jalur	(16 – 17) – (20) – 21 – 22 – 23 – (24 – 26) – 27 – 23 – 28 – 29 – 21 – 32 – 34 – (35 – 36) – 37 – 38 – 39 – 37 – (40 – 41) – 42 – 34 – 45 – 46 – 47 – 48 – (49 – 51) – 52 – 48 – 53 – 54 – 46 – 57 – 58 – (59 – 60) – 61 – 62 – 63 – 61 – (64 – 67) – 68 – 58 – 71 – 72 – 73 – 74 – 79 – 80 – 73 – 81 – 72 - 82
Test Case	Menghitung nilai bayes masing-masing hipotesa berdasarkan gejala yang diinputkan terurut berdasarkan nilai bayes terbesar ke terkecil
Target yang diharapkan	Berhasil menghitung nilai bayes masing-masing hipotesa berdasarkan gejala yang diinputkan terurut berdasarkan nilai bayes terbesar ke terkecil
Hasil Pengujian	Benar

4.4.2 Metode *Black Box*

Pengujian *Black Box* merupakan pengujian sistem yang bertujuan untuk mengevaluasi inputan, fungsi, maupun keluaran terhadap kebutuhan fungsional yang dibutuhkan dari sistem yang telah dikembangkan. Hasil pengujian sistem dengan menggunakan metode *black box* dapat dilihat pada Tabel 4.8.

Tabel 4.14 Pengujian sistem dengan metode *Black Box*

No.	Fitur	Aksi	Hasil	Ket
1.	Login Area	Klik tombol login area pada action bar	Menampilkan view login yang berisi textfield username, password, dan tombol login	OK
		Mengisi textfield username dan password kemudian klik tombol login	Login berhasil dan menampilkan halaman utama berisi menu Gejala, Hipotesa, Identifikasi, <i>Credits</i> , dan	OK

No.	Fitur	Aksi	Hasil	Ket
			tombol <i>logout</i> pada <i>Action Bar</i> aplikasi	
		Memasukkan username dan password yang salah	Login gagal dan menampilkan alert dialog "Username dan Password salah!"	OK
		Klik tombol <i>logout</i> pada <i>action bar</i>	Menampilkan halaman utama untuk user biasa	OK
2.	Pengelolaan Data Gejala	Menekan tombol menu "Gejala" pada <i>view</i> menu utama	Menampilkan <i>view List</i> Gejala yang terdiri dari data gejala, status verifikasi dan tombol "Tambah Gejala"	OK
		Menekan tombol "Tambah Gejala" pada <i>view list</i> gejala	Menampilkan form tambah gejala yang berisi <i>textfield</i> gejala dan tombol <i>submit</i>	OK
		Mengisi <i>id_gejala</i> , nama gejala, dan menekan tombol <i>submit</i>	Menyimpan data gejala kedalam database dan menampilkan Alert "Input data gejala berhasil!"	OK
		Tidak lengkap mengisi <i>textfield id_gejala</i> dan gejala kemudian menekan tombol <i>submit</i>	Menampilkan <i>Alert</i> "Inputan form tidak lengkap" pada <i>view form input</i> gejala	OK
		Mengisi form inputan gejala dengan data yang sudah pernah diinputkan sebelumnya	Menampilkan <i>Alert</i> "Data <i>id_gejala</i> sudah tersedia"	OK
		Menghapus data gejala dengan	Menghapus data gejala pada	OK

No.	Fitur	Aksi	Hasil	Ket
		memilih salah satu data dan menekan tombol delete pada opsi dialog	database dan menampilkan Alert “Gejala dihapus!”	
3.	Pengelolaan Data Hipotesa	Menekan tombol menu “Hipotesa” pada <i>view</i> menu utama	Menampilkan <i>view List</i> Hipotesa yang terdiri dari nama hipotesa, gambar hipotesa, status verifikasi, dan tombol “Tambah Hipotesa”	OK
		Menekan tombol “Tambah Hipotesa” pada <i>view list</i> hipotesa	Menampilkan form tambah hipotesa yang berisi tombol “Pilih Gambar”, <i>textfield</i> id_hipotesa, <i>textfield</i> hipotesa, <i>textfield</i> solusi, dan tombol <i>submit</i>	OK
		Menekan tombol pilih gambar dan memilih gambar hipotesa yang sesuai, mengisi <i>textfield</i> id_hipotesa, <i>textfield</i> hipotesa, dan <i>textfield</i> solusi Kemudian menekan tombol <i>submit</i>	Menyimpan data hipotesa kedalam database dan menampilkan kumpulan data hipotesa	OK

No.	Fitur	Aksi	Hasil	Ket
		Tidak memilih gambar hipotesa dan tidak lengkap mengisi <i>textfield</i> <i>id_hipotesa</i> , <i>textfield</i> hipotesa, dan <i>textfield</i> solusi kemudian menekan tombol submit	Menampilkan <i>Alert</i> "Field Inputan tidak lengkap!"	OK
		Mengisi data hipotesa yang pernah diinputkan sebelumnya dan menekan tombol submit	Menampilkan <i>Alert</i> "Data <i>id_hipotesa</i> sudah ada"	OK
		Menghapus salah satu hipotesa dengan menekan tombol delete pada opsi dialog	Menghapus data hipotesa pada tabel hipotesa <i>database</i> kedelai sesuai dengan <i>id_hipotesa</i> dan menampilkan <i>Alert</i> "Hipotesa dihapus!"	OK
		Melihat detail salah satu hipotesa dengan menekan tombol detail pada opsi dialog	Menampilkan <i>view</i> detail hipotesa yang terdiri dari hipotesa, gambar, list gejala pada suatu hipotesa, solusi, dan tombol tambah gejala	OK
		Menekan tombol tambah gejala untuk menambahkan gejala pada detail salah satu hipotesa	Menampilkan <i>list</i> gejala pada Dialog spinner dan tombol tambahkan	OK

No.	Fitur	Aksi	Hasil	Ket
		Memilih gejala pada Dialog Spinner dan menekan tombol submit	Menampilkan Alert “Gejala berhasil ditambahkan!”	OK
		Tambah Data Gejala pada salah satu Hipotesa dan data gejala pernah diinputkan	Menampilkan Alert “Gejala pada Hipotesa ini sudah pernah diinputkan”	OK
		Delete Gejala pada salah satu detail Hipotesa	Menghapus id_gejala pada tabel hipotesa_trans <i>database</i> kedelai dan Menampilkan <i>view</i> detail hipotesa	OK
4.	View Credits	Menekan tombol menu “Credits” pada <i>view</i> menu utama	Menampilkan <i>view</i> Credits yang terdiri dari deskripsi aplikasi	OK
5.	Verifikasi Data Gejala	Menekan tombol unverified pada gejala yang akan diverifikasi	Merubah tombol unverified menjadi verified berwarna hijau	OK
6.	Verifikasi Data Hipotesa	Menekan tombol verifikasi pada salah satu detail hipotesa	Mengambil id_hipotesa dan merubah status verifikasi pada tabel hipotesa <i>database</i> kedelai dan menampilkan alert “Berhasil di-verifikasi!”	OK

4.4.3 Uji Validitas dan Reliabilitas

Pada Uji Validitas dan Reliabilitas dengan menggunakan angket atau kuisisioner sebagai alat ukur instrumen pada aplikasi identifikasi hama dan penyakit pada tanaman kedelai. Pada pertanyaan yang diajukan dengan menggunakan skala likert lima pilihan jawaban yaitu:

- a. Skor 1 = Sangat Tidak Setuju (STS)
- b. Skor 2 = Tidak Setuju (TS)
- c. Skor 3 = Cukup Setuju (CS)
- d. Skor 4 = Setuju (S)
- e. Skor 5 = Sangat Setuju (SS)

Adapun daftar pertanyaan yang diajukan terdapat pada Tabel 4.10.

Tabel 4.15 Daftar pertanyaan pada angket atau kuisisioner

No.	Pertanyaan	SS	S	CS	TS	STS
1.	Apakah dengan adanya Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit Kedelai dapat membantu petani mengenali hama maupun penyakit?					
2.	Apakah dengan menggunakan aplikasi sistem pakar identifikasi hama dan penyakit kedelai dapat dengan cepat mengidentifikasi hama atau penyakit dibandingkan tanpa sistem pakar?					
3.	Apakah aplikasi ini sangat bermanfaat?					
4.	Apakah aplikasi ini cukup baik dan mudah digunakan?					
5.	Apakah aplikasi ini layak untuk membantu mengidentifikasi hama atau penyakit kedelai?					
6.	Apakah aplikasi ini dapat memberikan informasi tentang hama dan penyakit kedelai?					
7.	Apakah anda setuju bahwa aplikasi ini dapat meniru seorang pakar/penyuluh pertanian tanaman kedelai dalam mengidentifikasi hama atau penyakit kedelai?					

Target responden untuk pengumpulan data yaitu para petani di Indonesia dengan memanfaatkan kuisisioner digital atau Google Form yang berisi daftar pertanyaan

tentang identitas responden dan 7 item pertanyaan terkait aplikasi yang telah dikembangkan. Pada kuisisioner yang diajukan, didapatkan sejumlah 25 responden dari anggota komunitas petani dengan yang masing-masing 7 item pertanyaan dengan skor yang didapatkan berdasarkan jawaban yang diberikan responden dapat dilihat pada Tabel 4.11.

Tabel 4.16 Hasil Rekap Kuisisioner untuk Aplikasi Identifikasi Hama dan Penyakit pada Kedelai

No.	Responden	No. Item						
		Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7
1	1	3	3	3	5	5	3	1
2	2	5	5	5	5	5	5	5
3	3	4	5	5	4	4	5	3
4	4	5	5	4	4	5	5	5
5	5	5	5	5	5	5	5	5
6	6	5	5	5	5	5	5	4
7	7	4	3	4	3	4	4	3
8	8	4	5	5	3	4	5	3
9	9	5	5	5	5	5	5	5
10	10	5	3	3	4	4	3	4
11	11	4	4	4	4	4	4	4
12	12	3	3	3	4	3	3	2
13	13	4	4	5	3	4	4	4
14	14	4	3	4	3	4	3	3
15	15	5	4	5	4	4	4	3
16	16	5	4	5	5	5	5	5
17	17	4	3	4	4	5	5	3
18	18	5	4	4	5	4	5	5
19	19	4	3	5	4	4	5	4
20	20	3	4	4	3	4	3	4
21	21	4	4	4	4	4	4	4
22	22	5	5	5	5	5	5	5
23	23	4	3	4	4	4	5	4
24	24	5	5	5	5	5	5	5
25	25	5	3	4	4	4	5	3

BAB 6. PENUTUP

Bab ini berisi tentang penjelasan mengenai kesimpulan dari penelitian yang telah dilakukan oleh penulis beserta saran yang dapat dilakukan untuk pengembangan penelitian lebih lanjut. Kesimpulan maupun saran diharapkan dapat dijadikan sebagai acuan bagi peneliti selanjutnya.

6.1 Kesimpulan

Kesimpulan yang didapat dari penelitian yang telah dilakukan yaitu:

1. Teorema Bayes pada penerapannya terhadap aplikasi identifikasi hama dan penyakit kedelai memiliki beberapa kesimpulan sebagai berikut:
 - a. Sistem ini menggunakan 24 hipotesa dan 76 gejala yang diinputkan ke dalam sistem sehingga memiliki data yang lebih besar dibandingkan penelitian sebelumnya yang menggunakan 9 hipotesa dan 23 gejala.
 - b. Hasil perhitungan akurasi sistem dengan menginputkan beberapa gejala yang berbeda, didapatkan sejumlah 22 hipotesa yang valid dari 24 jumlah hipotesa. Sehingga didapatkan persentase sebesar 91,66%. Hasil tersebut memiliki akurasi yang lebih baik dari penelitian sebelumnya yang memiliki nilai akurasi sistem sebesar 90% dari 20 sampel gejala.
 - c. Penambahan data hipotesa dan gejala dapat diinputkan ke dalam sistem secara dinamis tanpa mengatur ulang kaidah penelusuran (*rule*) yang merupakan kelebihan diterapkannya metode teorema bayes ke dalam sistem sehingga memberikan keleluasaan tanpa harus mengubah kode program.
2. Aplikasi Sistem Pakar Identifikasi Hama dan Penyakit pada Tanaman Kedelai dibangun dengan menggunakan *SDLC Waterfall* dan berdasarkan analisis kebutuhan fungsional memiliki 3 hak akses dengan aktor antara lain user (petani), pakar (verifikator), dan admin (pengembang). Sistem ini memiliki beberapa kesimpulan terkait tentang pengembangannya yaitu sebagai berikut:
 - a. Hak akses user memiliki fungsi antara lain identifikasi hama dan penyakit, view data gejala, view data solusi, dan view credits. Pakar berfungsi untuk memverifikasi data gejala, data hipotesa, melakukan login sistem, dan view

credits. Sedangkan admin berfungsi untuk mengelola data gejala, data hipotesa dan melakukan login sistem.

- b. Hasil uji validitas untuk mengetahui kevalidan instrument menunjukkan 7 item yang diajukan sebagai pertanyaan terhadap 25 petani sebagai responden menghasilkan nilai t hitung $>$ t tabel sehingga keseluruhan item tersebut dapat disimpulkan menghasilkan nilai yang valid.
- c. Hasil uji reliabilitas untuk mengetahui tingkat konsistensi dari aplikasi yang telah dibangun menunjukkan koefisien reliabilitas sebesar 0,780165 lebih besar daripada 0,6 yang merupakan nilai minimal tingkat reliabilitas suatu instrumen yang reliabel sehingga dapat disimpulkan bahwa instrumen tersebut memiliki tingkat reliabilitas yang baik.

6.2 Saran

Beberapa saran yang dapat diberikan sebagai masukan untuk penelitian selanjutnya yaitu:

1. Penelitian dan pengembangan lebih lanjut diharapkan mengkombinasikan metode teorema *bayes* dengan metode yang lain untuk mengatasi hasil probabilitas yang sama dan penambahan jumlah gejala maupun jumlah hipotesa yang lebih besar, sehingga nilai bayes yang dihasilkan sangat sensitif.
2. Penelitian selanjutnya diharapkan mengembangkan sistem dengan menggunakan persamaan teorema *bayes* probabilitas bersyarat sebagai pembandingan hasil penelitian dan keluaran yang dihasilkan sistem.
3. Pengembangan sistem lebih lanjut diharapkan dapat dilakukan pada platform yang berbeda seperti windows phone, ios, maupun platform mobile lainnya yang akan mendominasi pada waktu yang akan datang.

DAFTAR PUSTAKA

- Kenneth E. Kendall. (2010). *Analisis dan Perancangan Sistem*. Jakarta: Indeks.
- Marwoto, Sri Hardaningsih, & Taufiq, A. (2006). *Hama, penyakit, dan masalah hara pada tanaman kedelai: identifikasi dan pengendaliannya*. Bogor: Badan Penelitian dan Pengembangan Pertanian, Pusat Penelitian dan Pengembangan Tanaman Pangan.
- Muhammad Johan Wahyudi, & Abdul Fadlil. (2013). Sistem Pakar Untuk Mengidentifikasi Penyakit Udag Galah Dengan Metode Theorema Bayes, *I*(1), 10.
- Okada, Wedanimbi Tengkano, & T. Djuwarso. (1998). *An Outline of Soybean Pest in Indonesia in Faunestic Aspects*. Bogor: BORIF.
- Prawidya Destarianto, Erni Yudanigtiyas, & Sholeh Hadi Pramono. (2013). Penerapan Metode Inference Tree dan Forward Chaining dalam Sistem Pakar Diagnosis Hama dan Penyakit Kedelai Edamame Berdasarkan Gejala Kerusakannya, *7*, 7.
- Pressman, R. S. (2015). *Software engineering: a practitioner's approach* (Eighth edition). New York, NY: McGraw-Hill Education.
- Rosa A.S, & M. Salahuddin. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.
- Rukmana, I. R., & Yuniarsih, Y. (2006). *Kedelai Budidaya dan Pasca Panen*. Yogyakarta: KANISIUS.
- Sri Kusumadewi. (2003). *Artificial Intelligence*. Yogyakarta: Graha Ilmu.
- Wedanimbi Tengkano, & M. Soehardjan. (1985). *Jenis-jenis hama pada berbagai fase pertumbuhan tanaman kedelai*. Bogor: Puslitbangtan.
- Wisnu Mahendra, Achmad Ridok, & Nurul Hidayat. (2010). Penerapan Teorema Bayes Untuk Identifikasi Penyakit Pada Tanaman Kedelai, *1*, 5.

LAMPIRAN

Lampiran A. Use Case Skenario

A.1 Use Case Skenario Pengelolaan Data Hipotesa

Tabel A.1 Use Case Skenario Pengelolaan Data Hipotesa

Name	Mengelola Data Hipotesa
Actor	Admin
Entry Condition	Aktor ingin mengelola Data Hipotesa
Exit Condition	Aktor berhasil mengelola Data Hipotesa
Skenario Utama “View Data Hipotesa”	
Aksi Aktor	Aksi Sistem
1. Menekan tombol menu “Hipotesa” pada <i>view</i> menu utama	2. Mengambil data <i>id_hipotesa</i> , <i>hipotesa</i> , <i>solusi</i> , <i>status verifikasi</i> , dan gambar dari tabel <i>hipotesa database</i> kedelai
	3. Menampilkan <i>view List</i> Hipotesa yang terdiri dari <i>hipotesa</i> , gambar <i>hipotesa</i> , <i>status verifikasi</i> , tombol <i>logout</i> dan tombol “Tambah Hipotesa”
Skenario Utama “Tambah Data Hipotesa”	
Aksi Aktor	Aksi Sistem
4. Menekan tombol “Tambah Hipotesa” pada <i>view list</i> hipotesa	5. Menampilkan form tambah hipotesa yang berisi tombol “Pilih Gambar”, <i>textfield</i> nama hipotesa, <i>textfield</i> solusi, dan tombol <i>submit</i>
6. Menekan tombol pilih gambar dan memilih gambar hipotesa yang sesuai, mengisi <i>textfield id_hipotesa</i> , <i>textfield</i> hipotesa, dan <i>textfield</i> solusi	
7. Menekan tombol <i>submit</i>	8. Menyimpan data gambar hipotesa, <i>id_hipotesa</i> , <i>hipotesa</i> dan <i>solusi</i> kedalam tabel <i>hipotesa database</i> kedelai

	9. Mengambil data <i>id_hipotesa</i> , <i>hipotesa</i> , <i>solusi</i> , <i>status verifikasi</i> , dan <i>gambar</i> dari tabel <i>hipotesa database</i> kedelai
	10. Menampilkan <i>view List Hipotesa</i> yang terdiri dari <i>hipotesa</i> , <i>gambar hipotesa</i> , <i>status verifikasi</i> , <i>tombol logout</i> dan <i>tombol “Tambah Hipotesa”</i>
Skenario Alternatif “Tambah Data Hipotesa” form tidak lengkap	
6a. Tidak memilih gambar hipotesa dan tidak lengkap mengisi <i>textfield id_hipotesa</i> , <i>textfield hipotesa</i> , dan <i>textfield solusi</i>	
7a. Menekan tombol <i>Submit</i>	8a. Menampilkan <i>Alert “Inputan form tidak lengkap/gambar tidak ada!”</i>
Skenario Alternatif “Tambah Data Hipotesa” data hipotesa sudah tersedia	
6b. Menekan tombol pilih gambar dan memilih gambar hipotesa yang sesuai, mengisi <i>textfield id_hipotesa</i> , <i>textfield hipotesa</i> , dan <i>textfield solusi</i>	
7b. Menekan tombol <i>Submit</i>	8b. Mengecek inputan nama hipotesa dengan nama hipotesa pada tabel <i>hipotesa</i> dari <i>database</i> kedelai
	9b. Menampilkan <i>Alert “Hipotesa sudah ada!”</i>
Skenario Utama “Delete Hipotesa”	
4. Memilih salah satu hipotesa dengan menekan salah satu hipotesa pada <i>view List Hipotesa</i>	
	5. Menampilkan <i>view Dialog Opsi</i> yang terdiri dari <i>tombol delete</i> dan <i>tombol Detail</i>
6. Menekan tombol <i>delete</i> pada <i>view Dialog Opsi</i>	
	7. Menghapus data hipotesa pada tabel <i>hipotesa database</i> kedelai sesuai dengan <i>id_hipotesa</i> yang akan dihapus oleh <i>Aktor</i>
	8. Menampilkan <i>Alert “Hipotesa telah dihapus!”</i>

Skenario Utama “View Detail salah satu Hipotesa”

- | | |
|--|---|
| 4. Memilih salah satu hipotesa dengan menekan salah satu hipotesa pada <i>view List Hipotesa</i> | |
| | 5. Menampilkan <i>view</i> Dialog Opsi yang terdiri dari tombol delete dan tombol Detail |
| 6. Menekan tombol Detail | |
| | 7. Mengambil data <i>id_hipotesa</i> , <i>hipotesa</i> , <i>solusi</i> , dan gambar dari tabel <i>hipotesa</i> |
| | 8. Mengambil data <i>id_gejala</i> dan data gejala berelasi dengan <i>id_hipotesa</i> dari tabel <i>hipotesa_trans database</i> kedelai |
| | 9. Menampilkan <i>view List Hipotesa</i> yang terdiri dari hipotesa, gambar hipotesa, status verifikasi, tombol logout dan tombol “Tambah Hipotesa” |

Skenario Utama “Tambah Data Gejala pada salah satu Hipotesa”

- | | |
|--|--|
| 4. Memilih salah satu hipotesa dengan menekan salah satu hipotesa pada <i>view List Hipotesa</i> | |
| | 5. Menampilkan <i>view</i> Dialog Opsi yang terdiri dari tombol delete dan tombol Detail |
| 6. Menekan tombol Detail | |
| | 7. Mengambil data <i>id_hipotesa</i> , <i>hipotesa</i> , <i>solusi</i> , dan gambar dari tabel <i>hipotesa</i> |
| | 8. Mengambil data <i>id_gejala</i> dan data gejala berelasi dengan <i>id_hipotesa</i> dari tabel <i>hipotesa_trans database</i> kedelai |
| | 9. Menampilkan <i>view</i> detail hipotesa yang terdiri dari hipotesa, gambar, list gejala pada suatu hipotesa, <i>solusi</i> , dan tombol tambah gejala |
| 10. Menekan tombol tambah gejala | |
| | 11. Mengambil data <i>id_gejala</i> dan data gejala pada tabel gejala <i>database</i> kedelai |
-

	12. Menampilkan <i>list</i> gejala pada Dialog spinner dan tombol tambahkan
13. Memilih gejala pada Dialog Spinner dan menekan tombol submit	
	14. Menyimpan <i>id_gejala</i> yang dipilih kedalam tabel <i>hipotesa_trans database</i> kedelai
	15. Menampilkan Alert “Gejala berhasil ditambahkan!”
Skenario Alternatif “Tambah Data Gejala pada salah satu Hipotesa” data gejala pernah diinputkan	
13a. Memilih gejala pada Dialog Spinner dan menekan tombol tambahkan	
	14a. Mengecek <i>id_gejala</i> pada spinner dengan <i>id_gejala</i> satu hipotesa pada tabel <i>hipotesa_trans</i>
	15a. Menampilkan Alert “Gejala pada Hipotesa ini sudah pernah diinputkan”
Skenario Utama “Delete Gejala pada salah satu Hipotesa”	
4. Memilih salah satu hipotesa dengan menekan salah satu hipotesa pada <i>view List</i> Hipotesa	
	5. Menampilkan <i>view</i> Dialog Opsi yang terdiri dari tombol delete dan tombol Detail
6. Menekan tombol Detail	
	7. Mengambil data <i>id_hipotesa</i> , <i>hipotesa</i> , <i>solusi</i> , dan gambar dari tabel <i>hipotesa</i>
	8. Mengambil data <i>id_gejala</i> dan data gejala berelasi dengan <i>id_hipotesa</i> dari tabel <i>hipotesa_trans database</i> kedelai
	9. Menampilkan <i>view</i> detail hipotesa yang terdiri dari hipotesa, gambar, list gejala pada suatu hipotesa, solusi, dan tombol tambah gejala
10. Memilih salah satu gejala pada detail hipotesa	

	11. Menampilkan dialog opsi dan tombol delete
12. Menekan tombol delete	
	13. Menghapus <i>id_gejala</i> pada tabel <i>hipotesa_trans database</i> kedelai
	14. Mengambil data <i>id_hipotesa</i> , <i>hipotesa</i> , <i>solusi</i> , dan gambar dari tabel <i>hipotesa</i>
	15. Mengambil data <i>id_gejala</i> dan data gejala berelasi dengan <i>id_hipotesa</i> dari tabel <i>hipotesa_trans database</i> kedelai
	16. Menampilkan <i>view</i> detail <i>hipotesa</i> yang terdiri dari <i>hipotesa</i> , gambar, list gejala pada suatu <i>hipotesa</i> , <i>solusi</i> , dan tombol tambah gejala

A.2 Use Case Skenario Login Area

Tabel A.2 Use Case Skenario Login Area

Name	<i>Login</i>
Actor	Admin dan Pakar
Entry Condition	Admin ingin mengakses aplikasi sistem pakar identifikasi hama dan penyakit pada tanaman kedelai
Exit Condition	Berhasil <i>Login</i>
Skenario Utama "Login"	
Aksi Aktor	Aksi Sistem
1. Membuka Aplikasi	2. Menampilkan menu Gejala, Hipotesa, Identifikasi, <i>Credits</i> dan tombol <i>login Area</i>
3. Menekan tombol " <i>Login Area</i> " pada <i>Action Bar</i> Aplikasi	4. Menampilkan <i>view</i> <i>Login</i> , <i>textfield</i> <i>username</i> , <i>textfield</i> <i>password</i> , dan tombol <i>login</i>
5. Memasukkan <i>Username</i> , <i>Password</i> , dan menekan tombol <i>login</i>	6. Verifikasi <i>username</i> dan <i>password</i> beserta membuat <i>session</i> dan menyimpannya kedalam <i>SharedPreference</i>

	7. Menampilkan Halaman utama berisi menu Gejala, Hipotesa, Identifikasi, <i>Credits</i> , dan tombol <i>logout</i> pada <i>Action Bar</i> aplikasi
Skenario Alternatif “Username dan Password salah”	
Aksi Aktor	Aksi Sistem
5a. Admin memasukkan <i>Username</i> dan <i>Password</i> yang salah	6a. Verifikasi <i>username</i> dan <i>password</i>
	7a. Menampilkan <i>Alert</i> Dialog “ <i>Username</i> atau <i>Password</i> Salah!”
Skenario Alternatif “Username dan Password tidak diisi”	
Aksi Aktor	Aksi Sistem
5b. Admin tidak lengkap dalam memasukkan <i>Username</i> atau <i>Password</i>	6b. Menampilkan <i>Alert</i> Dialog “Masukkan <i>Username</i> dan <i>Password</i> tidak lengkap!”
Skenario Utama “Memilih tombol Logout”	
Aksi Aktor	Aksi Sistem
2. Menekan tombol <i>Logout</i>	3. Menghapus <i>session</i> pada <i>SharedPreferences</i> dan menampilkan halaman utama berisi menu Gejala, Hipotesa, Identifikasi, <i>Credits</i> , dan tombol <i>login</i> area

A.3 Use Case Skenario View Credits

Tabel A.3 Use Case Skenario View Credits

Name	View Credits
Actor	User, Admin, dan Pakar
Entry Condition	Aktor ingin data Credits
Exit Condition	Aktor berhasil melihat data Credits
Skenario Utama “View Credits”	
Aksi Aktor	Aksi Sistem
1. Menekan tombol menu “Credits” pada <i>view</i> menu utama	2. Menampilkan view Credits yang terdiri dari deskripsi aplikasi



A.4 Use Case Skenario Verifikasi Data Gejala

Tabel A.4 Use Case Skenario Verifikasi Data Gejala

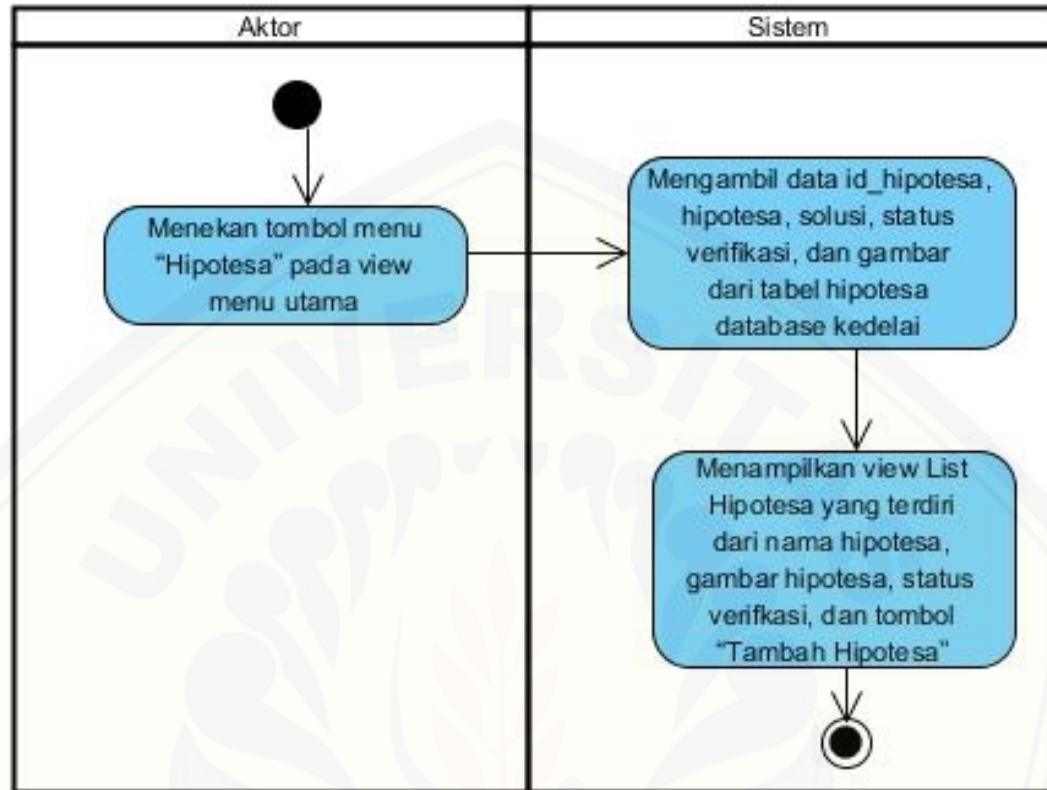
Name	Verifikasi Data Gejala
Actor	Pakar
Entry Condition	Aktor ingin memverifikasi data Gejala yang berisi id_gejala dan gejala
Exit Condition	Aktor berhasil memverifikasi data Gejala yang berisi id_gejala dan gejala
Skenario Utama “Verifikasi Data Gejala”	
Aksi Aktor	Aksi Sistem
1. Menekan tombol menu “Gejala” pada <i>view</i> menu utama	2. Mengambil data id_gejala, data gejala, dan status verifikasi dari tabel gejala <i>database</i> kedelai
	3. Menampilkan data id_gejala, data gejala, dan tombol unverified berwarna merah
4. Menekan tombol unverified pada gejala yang akan diverifikasi	5. Mengambil id_gejala dan merubah status verifikasi gejala verified
	6. Merubah tombol unverified menjadi verified berwarna hijau

A.5 Use Case Skenario Verifikasi Data Hipotesa

Tabel A.5 Use Case Skenario Verifikasi Data Hipotesa

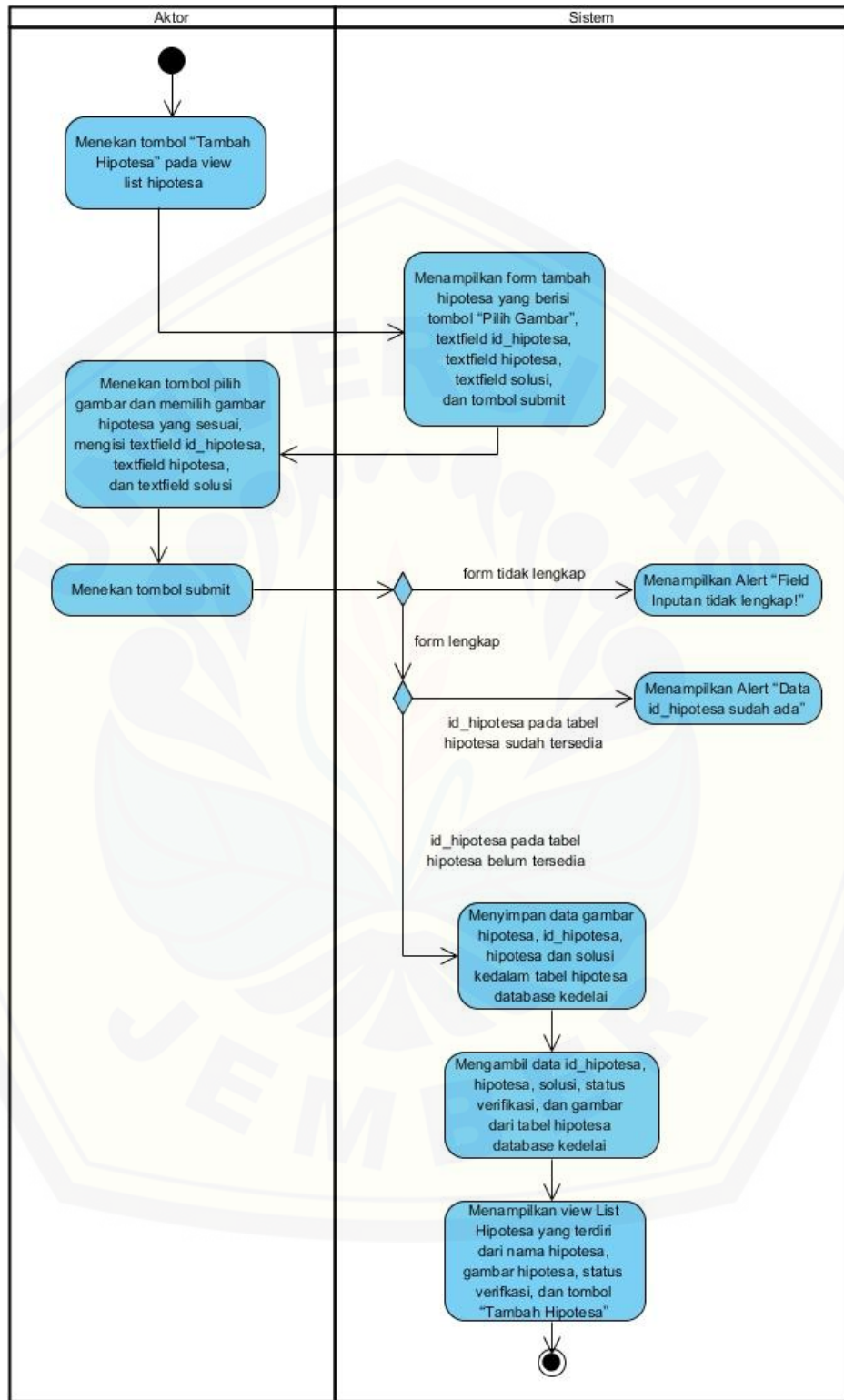
Name	Verifikasi Data Hipotesa
Actor	Pakar
Entry Condition	Aktor ingin memverifikasi data hipotesa yang berisi nama hipotesa, gambar, list gejala satu hipotesa, dan solusi
Exit Condition	Aktor berhasil memverifikasi data hipotesa yang berisi nama hipotesa, gambar, list gejala satu hipotesa, dan solusi
Skenario Utama “Verifikasi Data Hipotesa”	
Aksi Aktor	Aksi Sistem
1. Menekan tombol menu hipotesa pada <i>view</i> menu utama	

	2. Mengambil data <i>id_hipotesa</i> , <i>hipotesa</i> , <i>solusi</i> , <i>gambar</i> , dan status verifikasi dari tabel <i>hipotesa database</i> kedelai
	3. Menampilkan <i>view List Hipotesa</i> yang terdiri dari nama <i>hipotesa</i> , <i>gambar hipotesa</i> , dan status verifikasi
4. Memilih salah satu <i>hipotesa</i> dengan menekan salah satu <i>hipotesa</i> pada <i>view List Hipotesa</i>	
	5. Menampilkan <i>view Dialog Opsi</i> yang terdiri dari tombol <i>delete</i> dan tombol <i>Detail</i>
6. Menekan tombol <i>Detail</i>	
	7. Mengambil data <i>id_hipotesa</i> , <i>hipotesa</i> , <i>solusi</i> , dan <i>gambar</i> dari tabel <i>hipotesa</i>
	8. Mengambil data <i>id_gejala</i> dan data <i>gejala</i> berelasi dengan <i>id_hipotesa</i> dari tabel <i>hipotesa_trans database</i> kedelai
	9. Menampilkan <i>view detail hipotesa</i> yang terdiri dari <i>hipotesa</i> , <i>gambar</i> , <i>list gejala</i> pada suatu <i>hipotesa</i> , <i>solusi</i> , dan tombol verifikasi
9. Menekan tombol verifikasi	
	10. Mengambil <i>id_hipotesa</i> dan merubah status verifikasi pada tabel <i>hipotesa database</i> kedelai dan menampilkan alert “Berhasil di-verifikasi!”

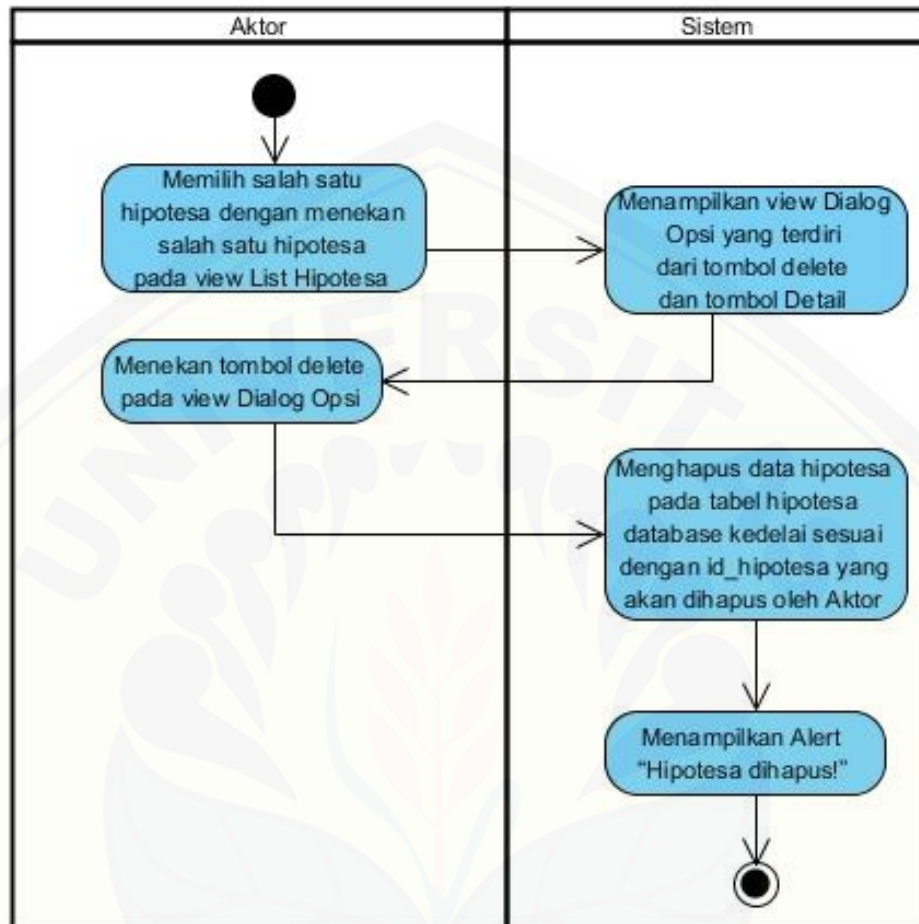
Lampiran B. Activity Diagram*B.1 Activity Diagram Pengelolaan Data Hipotesa (View)*

Gambar B.1 Activity Diagram Pengelolaan Data Hipotesa (View)

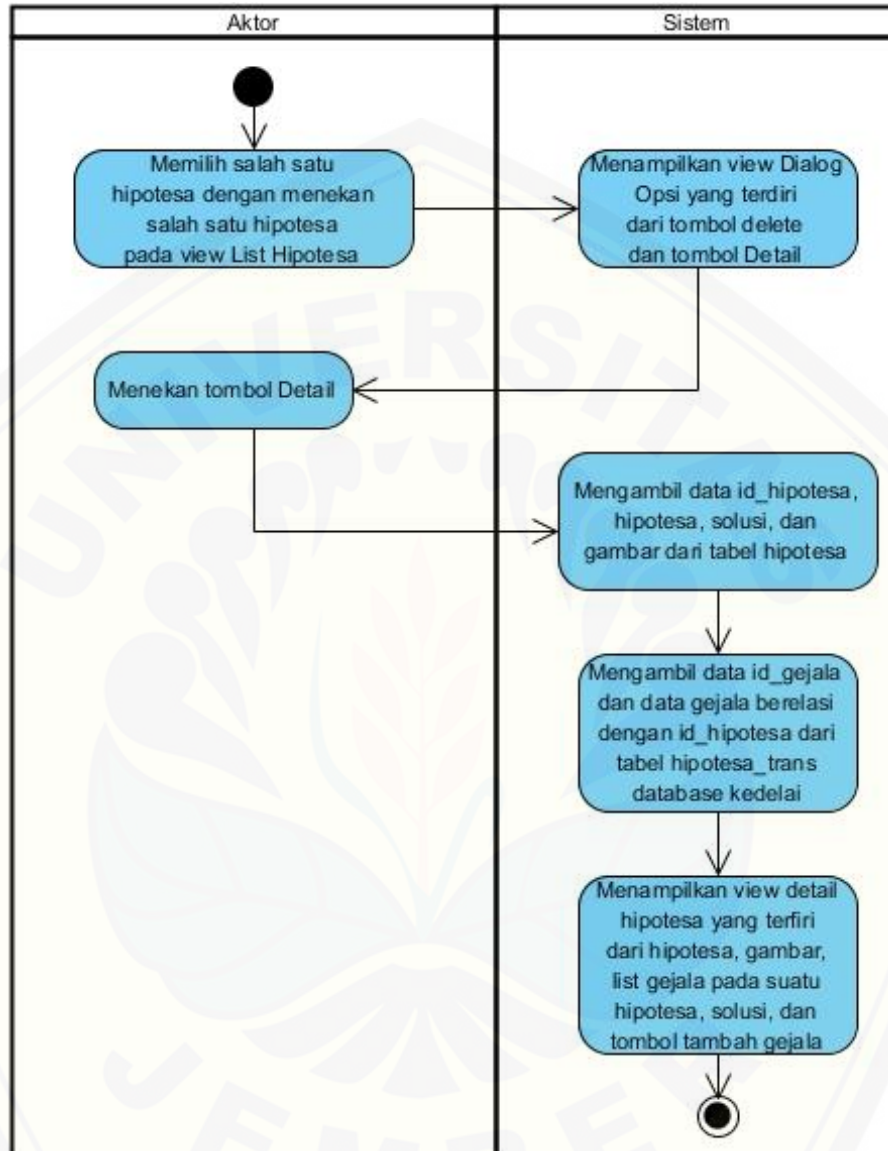
B.2 Activity Diagram Pengelolaan Data Hipotesa (Tambah)



Gambar B.2 Activity Diagram Pengelolaan Data Hipotesa (Tambah)

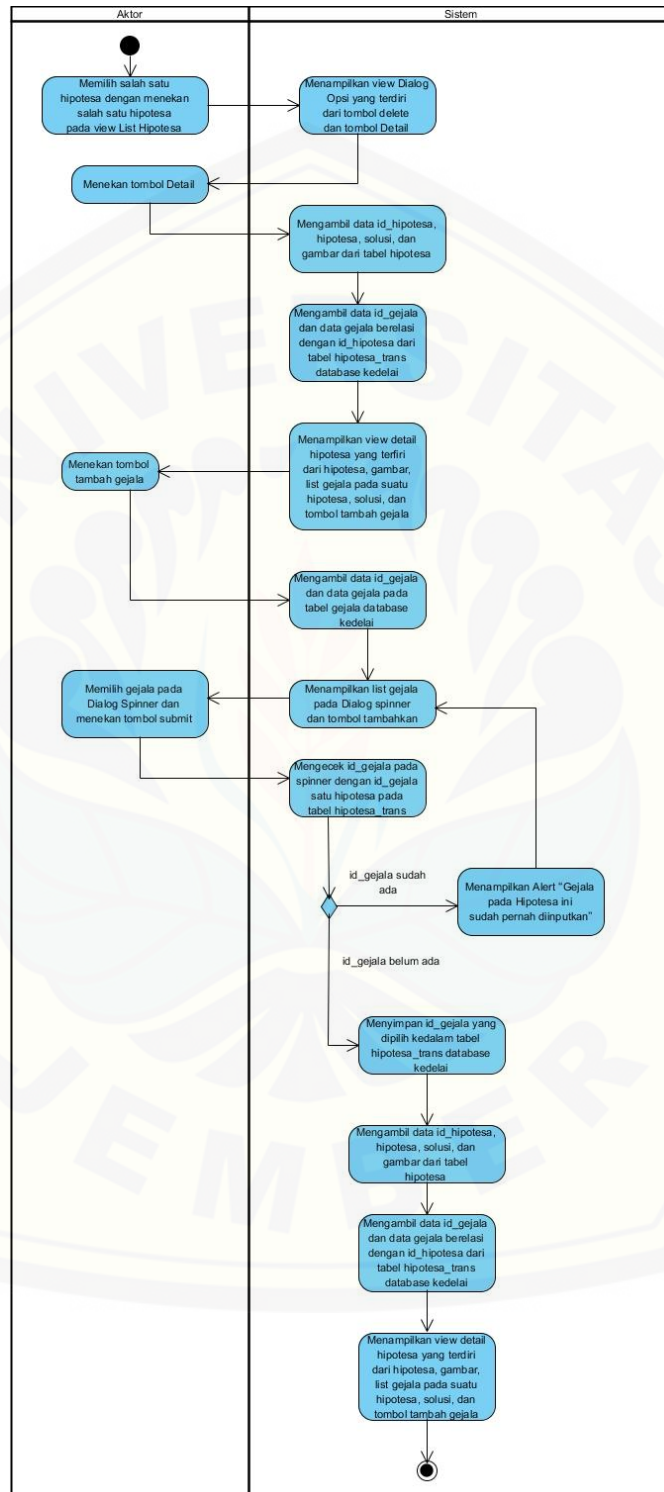
B.3 Activity Diagram Pengelolaan Data Hipotesa (*Delete*)Gambar B.3 Activity Diagram Pengelolaan Data Hipotesa (*Delete*)

B.4 Activity Diagram Pengelolaan Data Hipotesa (View Detail salah satu Hipotesa)



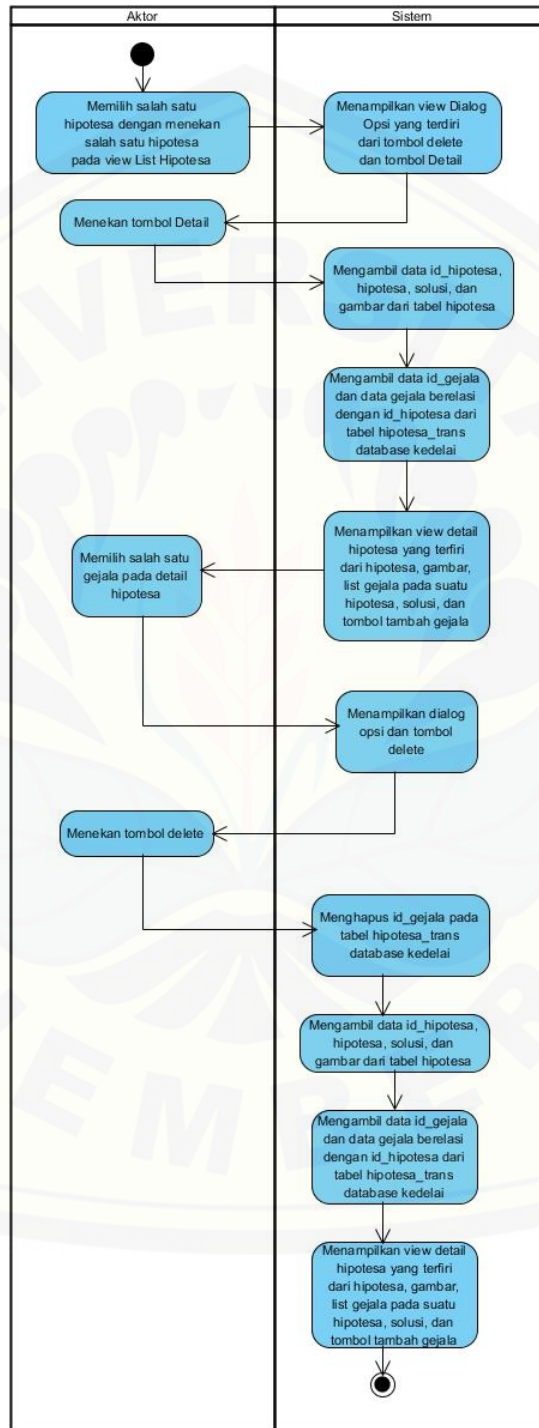
Gambar B.4 Activity Diagram Pengelolaan Data Hipotesa (View Detail salah satu Hipotesa)

B.5 Activity Diagram Pengelolaan Data Hipotesa (Tambah Data Gejala pada salah satu Hipotesa)



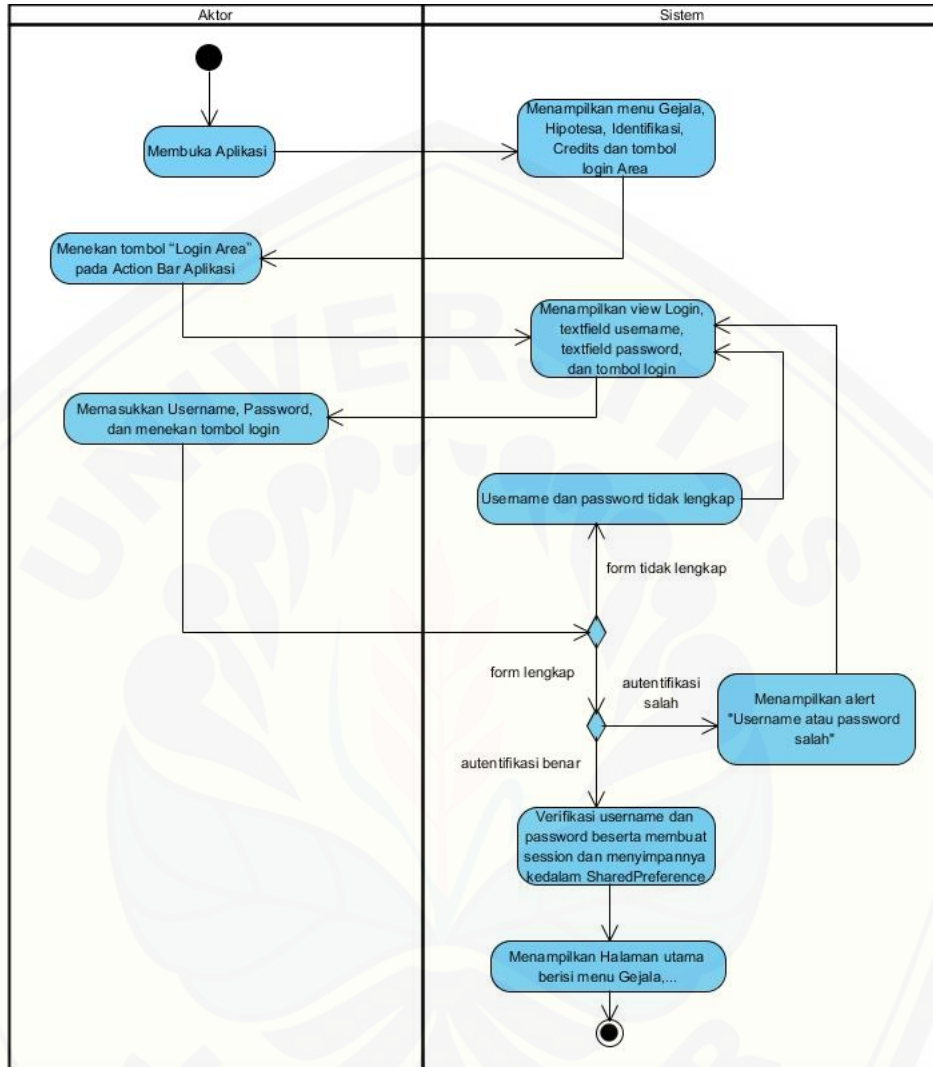
Gambar B.5 Activity Diagram Pengelolaan Data Hipotesa (Tambah Data Gejala pada salah satu Hipotesa)

B.6 Activity Diagram Pengelolaan Data Hipotesa (*Delete* Gejala pada salah satu Hipotesa)



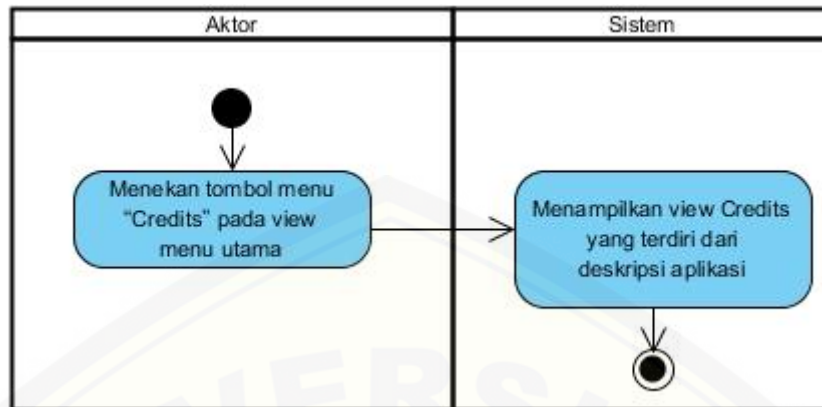
Gambar B.6 Activity Diagram Pengelolaan Data Hipotesa (*Delete* Gejala pada salah satu Hipotesa)

B.7 Activity Diagram Login Area



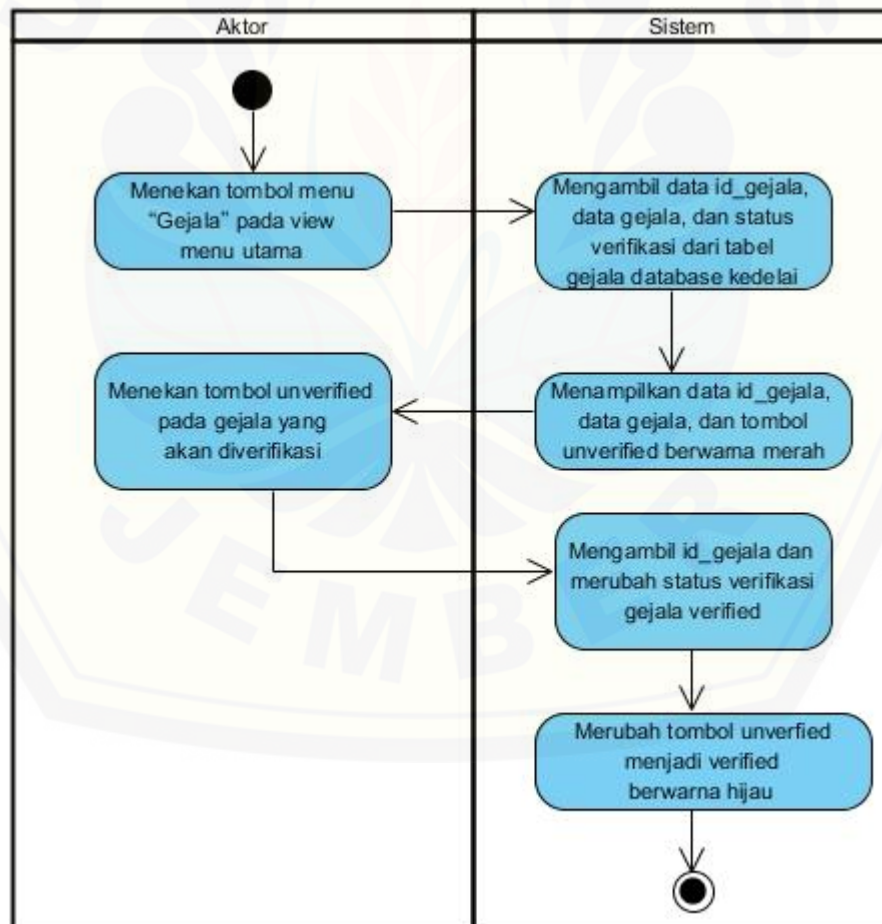
Gambar B.7 Activity Diagram Login Area

B.8 Activity Diagram View Credits



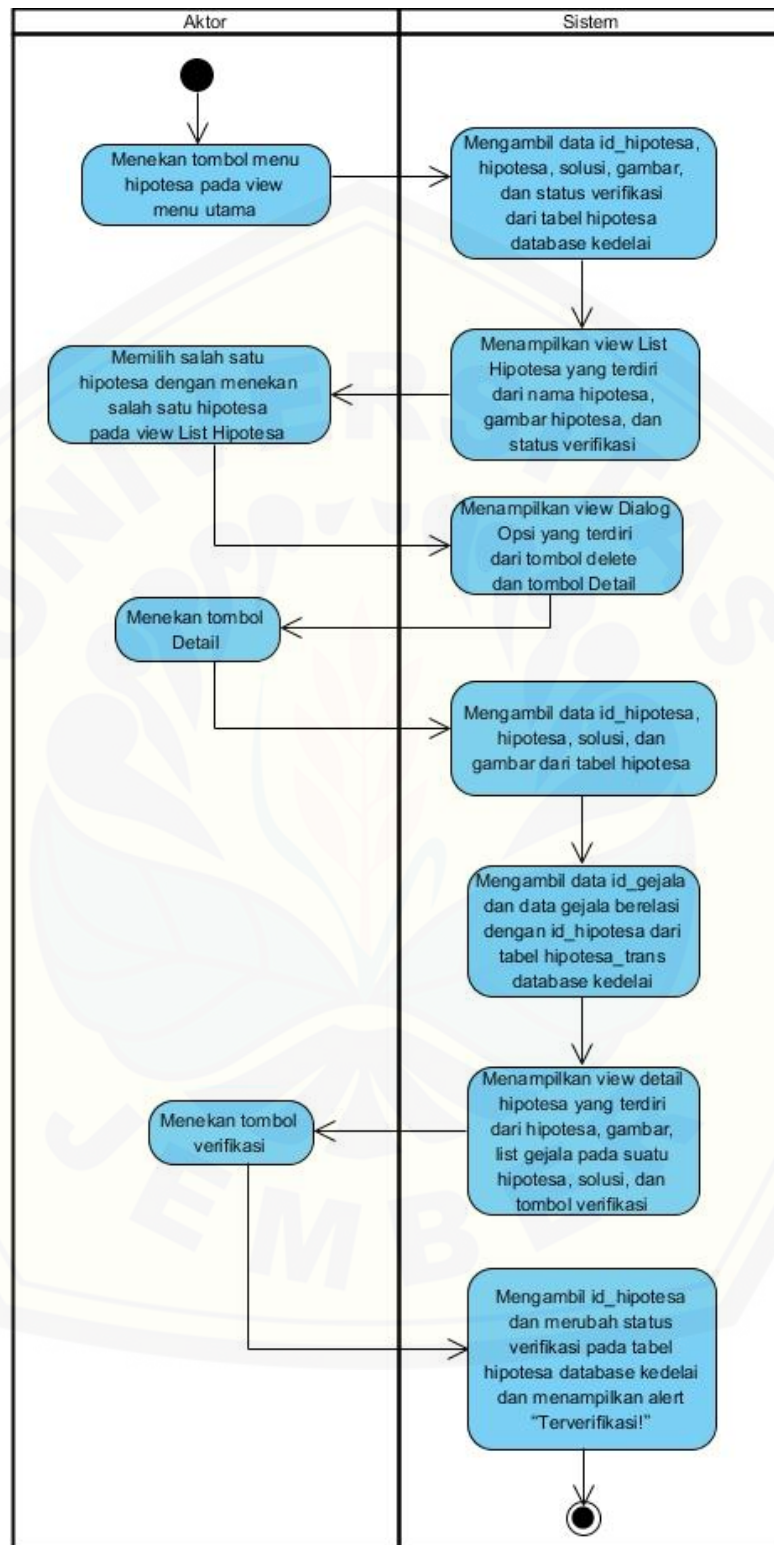
Gambar B.8 Activity Diagram View Credits

B.9 Activity Diagram Verifikasi Data Gejala



Gambar B.9 Activity Diagram Verifikasi Data Gejala

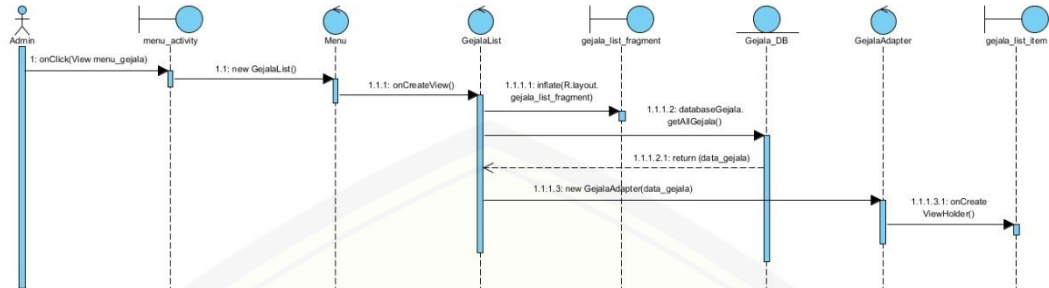
B.10 Activity Diagram Verifikasi Data Hipotesa



Gambar B.10 Activity Diagram Verifikasi Data Hipotesa

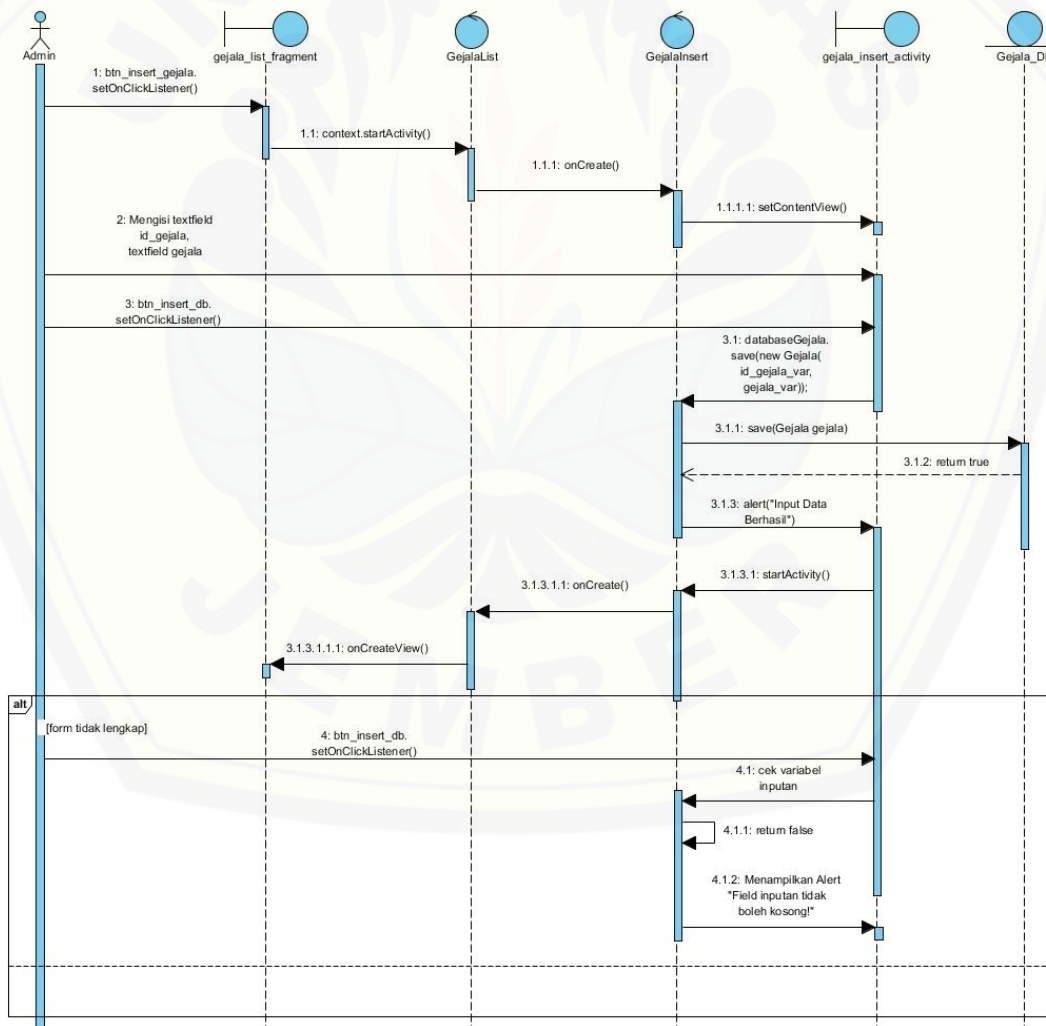
Lampiran C. Sequence Diagram

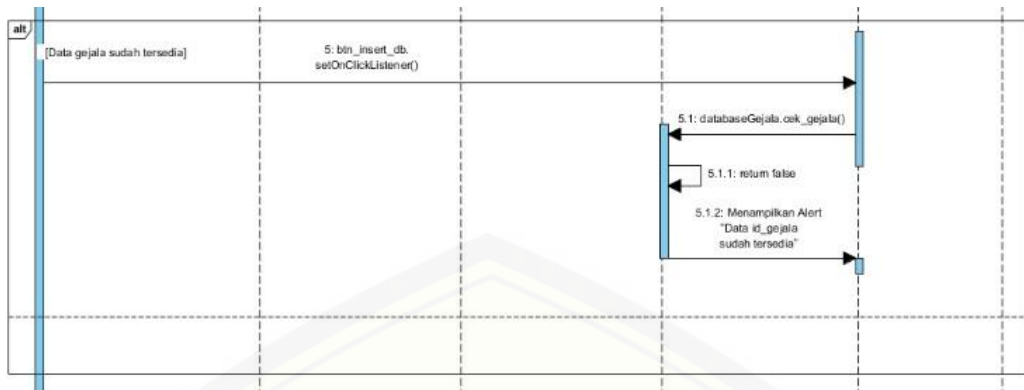
C.1 Sequence Diagram Pengelolaan Data Hipotesa (View)



Gambar C.1 Sequence Diagram Pengelolaan Data Hipotesa (View)

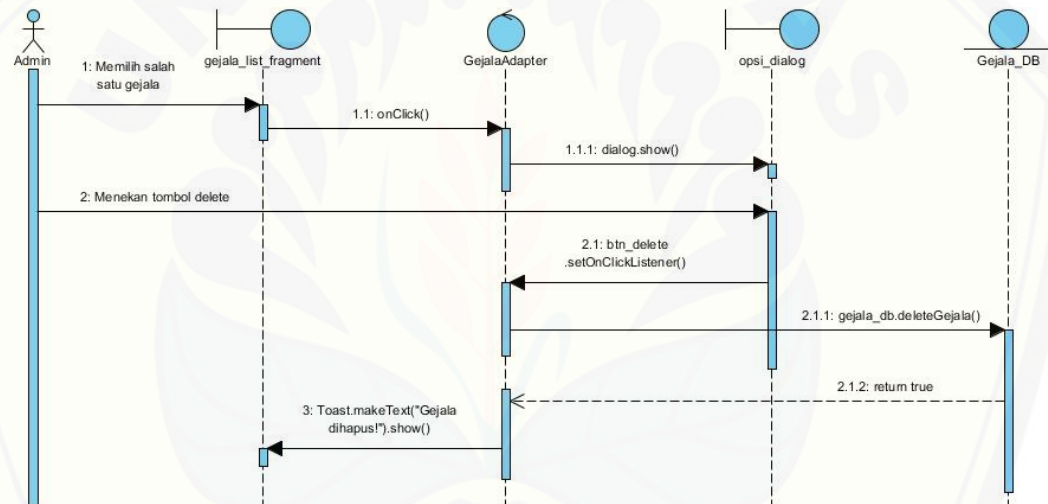
C.2 Sequence Diagram Pengelolaan Data Hipotesa (Tambah)





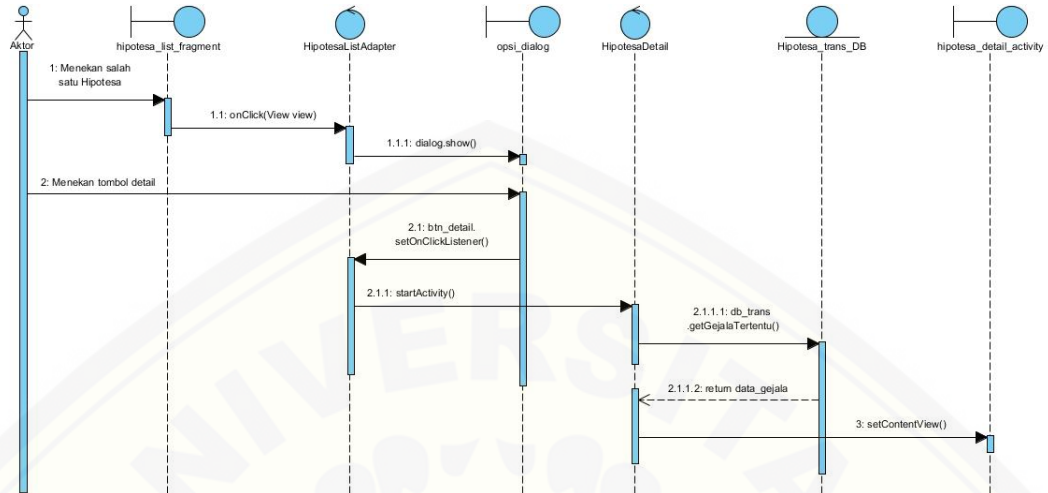
Gambar C.2 Sequence Diagram Pengelolaan Data Hipotesa (Tambah)

C.3 Sequence Diagram Pengelolaan Data Hipotesa (Delete)



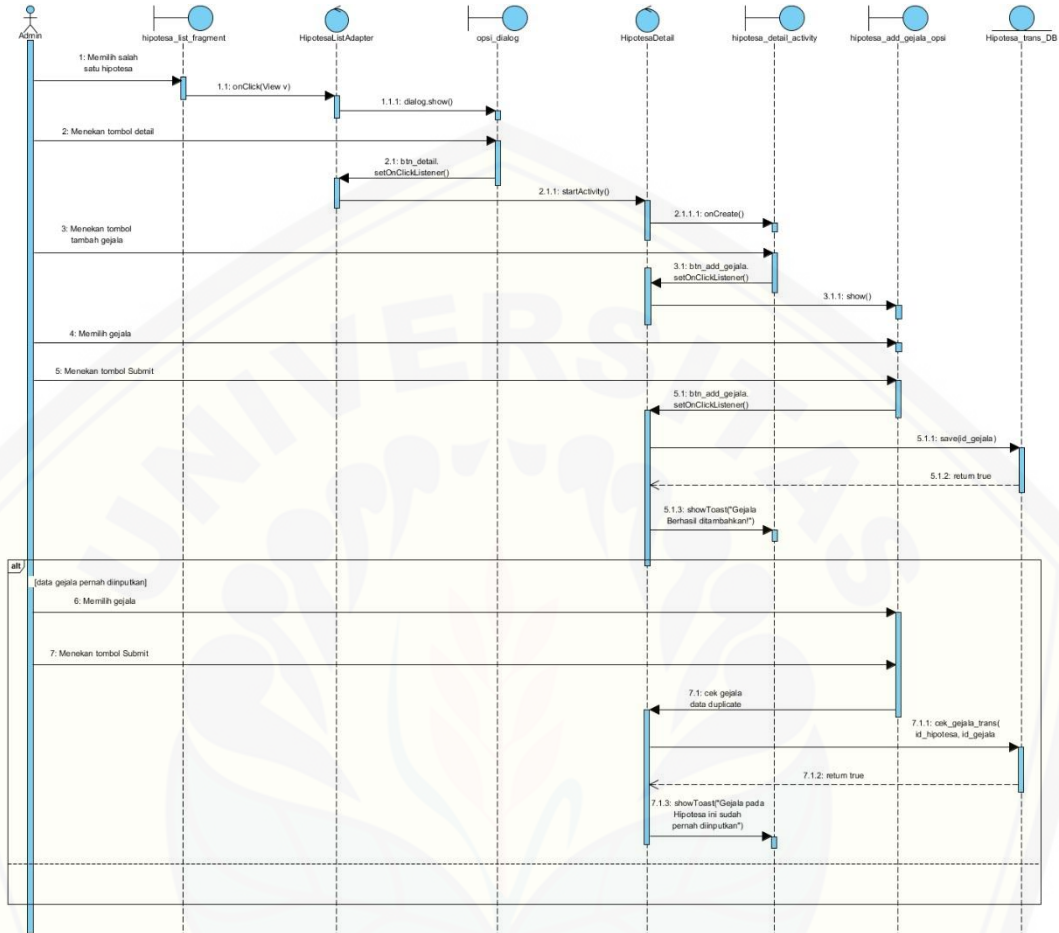
Gambar C.3 Sequence Diagram Pengelolaan Data Hipotesa (Delete)

C.4 Sequence Diagram Pengelolaan Data Hipotesa (View Detail salah satu Hipotesa)



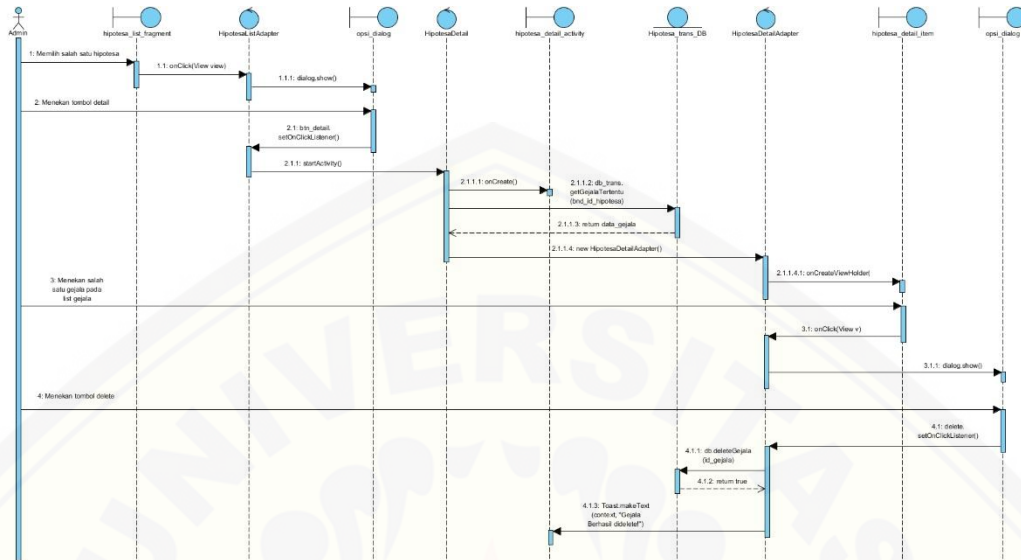
Gambar C.4 Sequence Diagram Pengelolaan Data Hipotesa (View Detail salah satu Hipotesa)

C.5 Sequence Diagram Pengelolaan Data Hipotesa (Tambah Gejala salah satu Hipotesa)



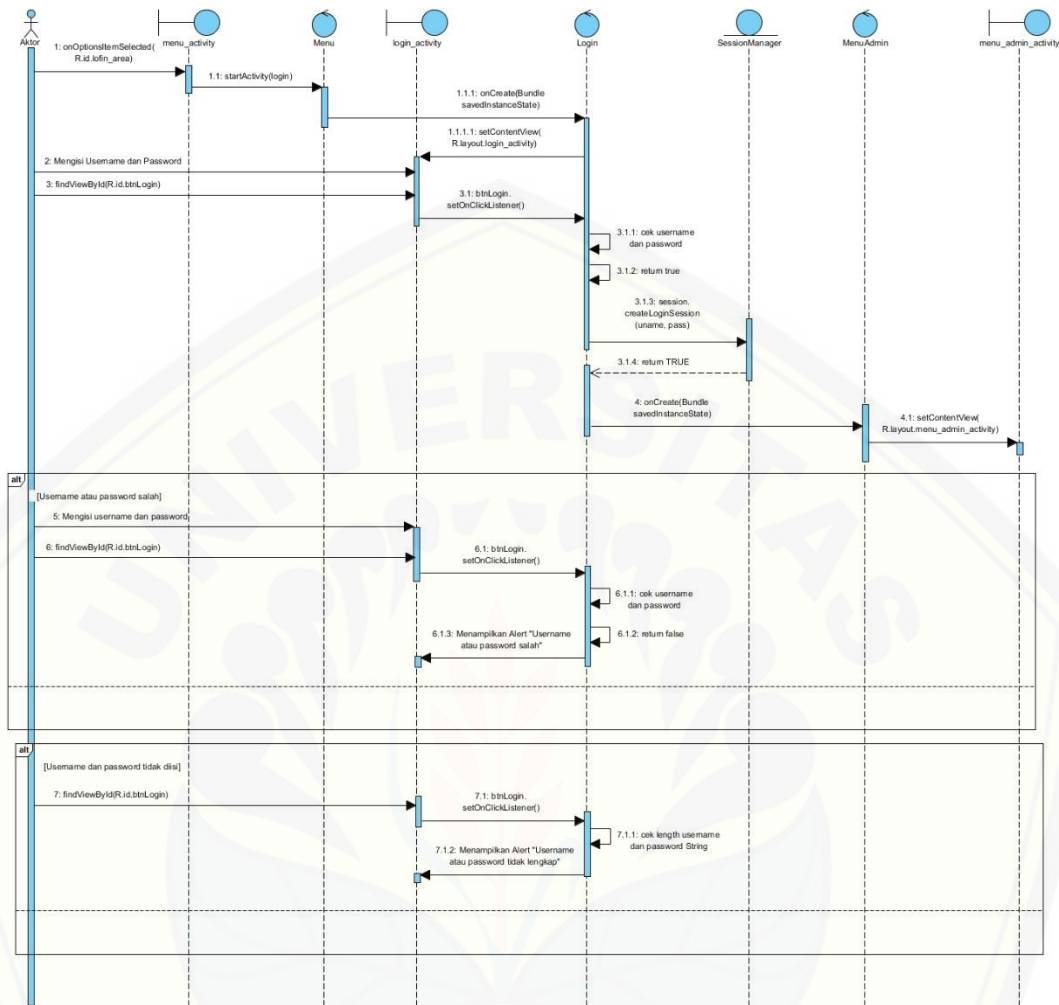
Gambar C.5 Sequence Diagram Pengelolaan Data Hipotesa (Tambah Gejala salah satu Hipotesa)

C.6 Sequence Diagram Pengelolaan Data Hipotesa (Delete Gejala salah satu Hipotesa)



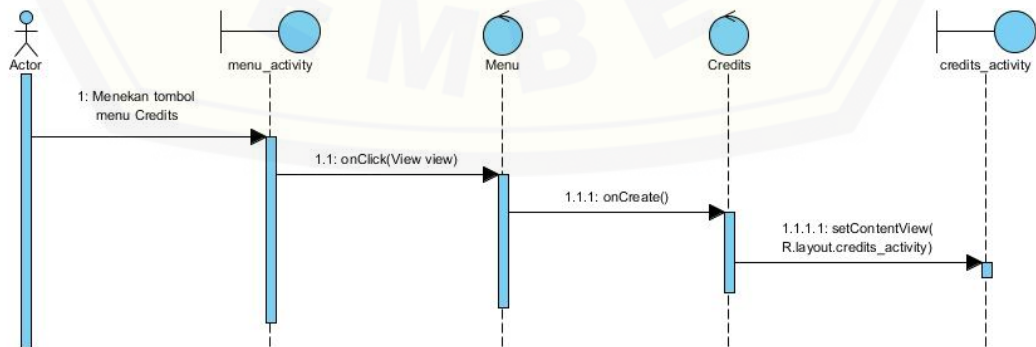
Gambar C.6 Sequence Diagram Pengelolaan Data Hipotesa (Delete Gejala salah satu Hipotesa)

C.7 Sequence Diagram Login Area



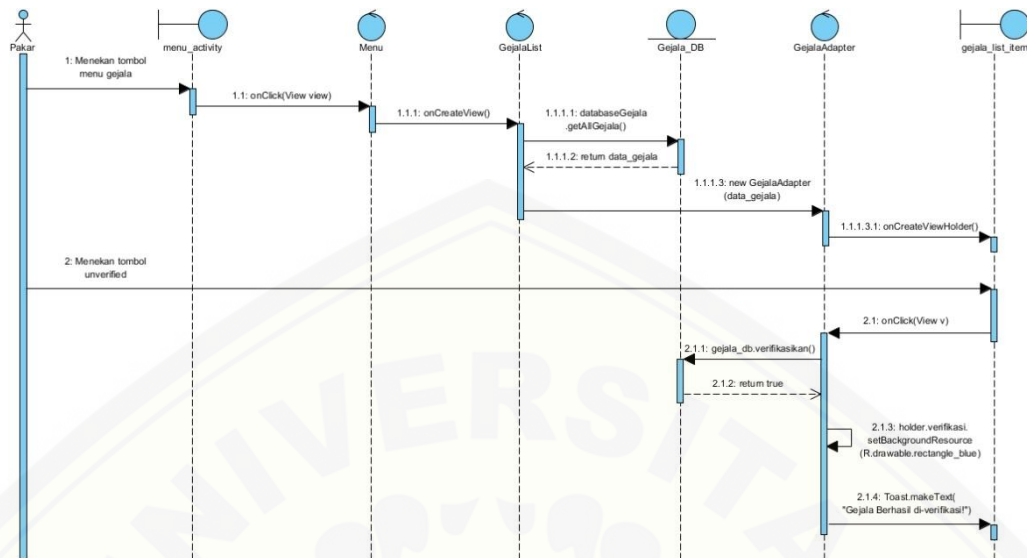
Gambar C.7 Sequence Diagram Login Area

C.8 Sequence Diagram View Credits



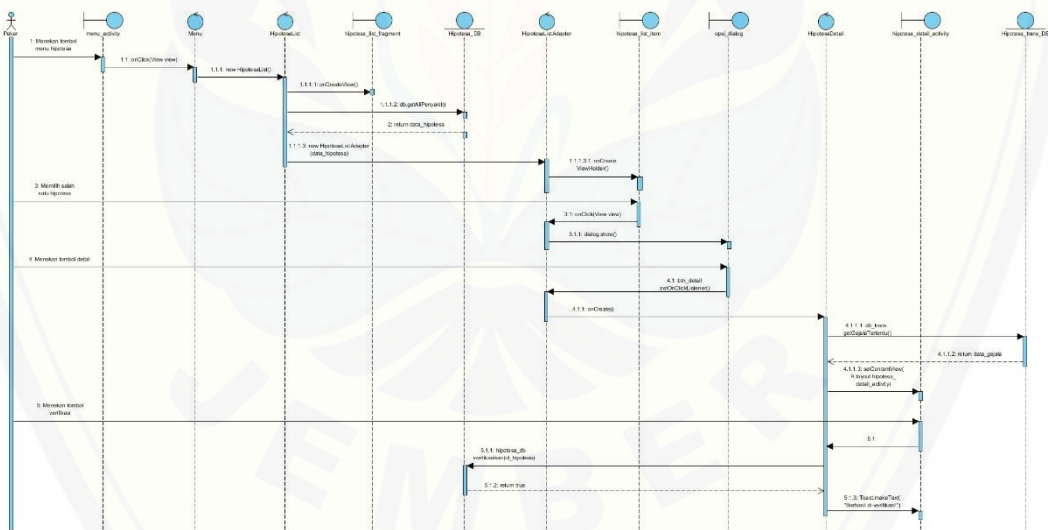
Gambar C.8 Sequence Diagram View Credits

C.9 Sequence Diagram Verifikasi Data Gejala



Gambar C.9 Sequence Diagram Verifikasi Data Gejala

C.10 Sequence Diagram Verifikasi Data Hipotesa



Gambar C.10 Sequence Diagram Verifikasi Data Hipotesa

Lampiran D. Surat Elektronik (Email) Wawancara Penelitian

D.1 Pertanyaan Tahap I



hofi atmajaya <hofiatmajaya2@gmail.com>

(tanpa subjek)

1 pesan

hofi atmajaya <hofiatmajaya2@gmail.com>

24 Oktober 2015 20.46

Kepada: balitkabi@litbang.pertanian.go.id

Assalamualaikum Balitkabi,,
perkenalkan nama saya Hofi Atmajaya, mahasiswa Univeritas Jember Prodi Sistem Informasi

Berhubung saya masih kesulitan dalam mencari ide skripsi pada smstr. 7 skrg, akan sangat membahagiakan sekali jika dari Balitkabi memberikan bantuan berupa Dataset penelitian dari thun sbelumnya hingga skrg. Data tersebut akan saya gunakan untuk memilih metode yang tepat untuk diterapkan kedalam aplikasi saya.

Pertanyaan saya, bagaimana saya memperoleh laporan tahunan (ex. 2014) dari hasil penelitian baik berupa buku yg tercantum pada WEB resmi Balitkabi maupun Dataset berbentuk tabel.

Sehubungan dengan ini, Data penelitian tersebut akan saya coba ajukan sebagai proposal skripsi dengan judul Sistem Informasi Pendeteksian Hama pada Tanaman Kacang dan Umbi berbasis Web,
untuk metodenya sementara masih mencari yg relevan dengan pengembangan sistem...

Sistem saya akan dirancang dengan memiliki fitur:

1. Processing data inputan yg akan dijadikan sebagai variabel dan pembobotan dalam menghasilkan informasi hasil deteksi hama
2. Processing Gambar tanaman maupun produk untuk menghasilkan informasi kualitas dari Tanaman maupun produk yg diinputkan gambarnya
3. Menampilkan metode yang tepat untuk pencegahan maupun solusi terkait dengan hasil informasi dari aplikasi yang Dasar-Dasar ilmu pencegahannya diambil dari Penelitian Balitkabi

Demikian dari saya, sangat berterima kasih dan sengan sekali apabila Balitkabi memberikan respon yang baik bagi saya.

Terima Kasih & Mohon Maaf apabila ada kekurangan.

Assalamualaikum WR. WB.

D.2 Jawaban Tahap I



hofi atmajaya <hofiatmajaya2@gmail.com>

re: informasi tentang laporan tahunan.

3 pesan

Seksi Jasa Penelitian Balitkabi <jaslit.balitkabi@gmail.com>

30 Oktober 2015

07.44

Kepada: hofiatmajaya2@gmail.com

assalamualaikum wr.wb

saudara Hofi,
pertama kami mohon maaf atas keterlambatan dalam menjawab e-mail yang Sudara kirim. berikut adalah tanggapan kami atas pertanyaan-pertanyaan Saudara.

Untuk dataset penelitian tidak dapat kami berikan karena merupakan hak kekayaan intelektual dari peneliti yang bersangkutan.

Laporan tahunan mulai dari 2010 sampai 2014 sudah ada di Web balitkabi pada menu hasil penelitian. Dapat secara langsung diunduh sebagai bahan referensi.

Skripsi dengan judul "Sistem Informasi Pendeteksian Hama pada Tanaman kacang dan Umbi Berbasis Web" terlalu luas. Mungkin cakupan bisa dipersempit.

Coba pelajari publikasi-publikasi yang kami muat dalam website kami, untuk mencari judul/topik apa apa yang berisi cukup lengkap informasinya.

jika tetap dengan topik seperti yang Saudara inginkan,

Terkait fitur aplikasi nomor 1:

Perdalam informasi mengenai metode perhitungan ambang kendali untuk diimplementasikan dalam aplikasi .

Terkait fitur nomor 2:

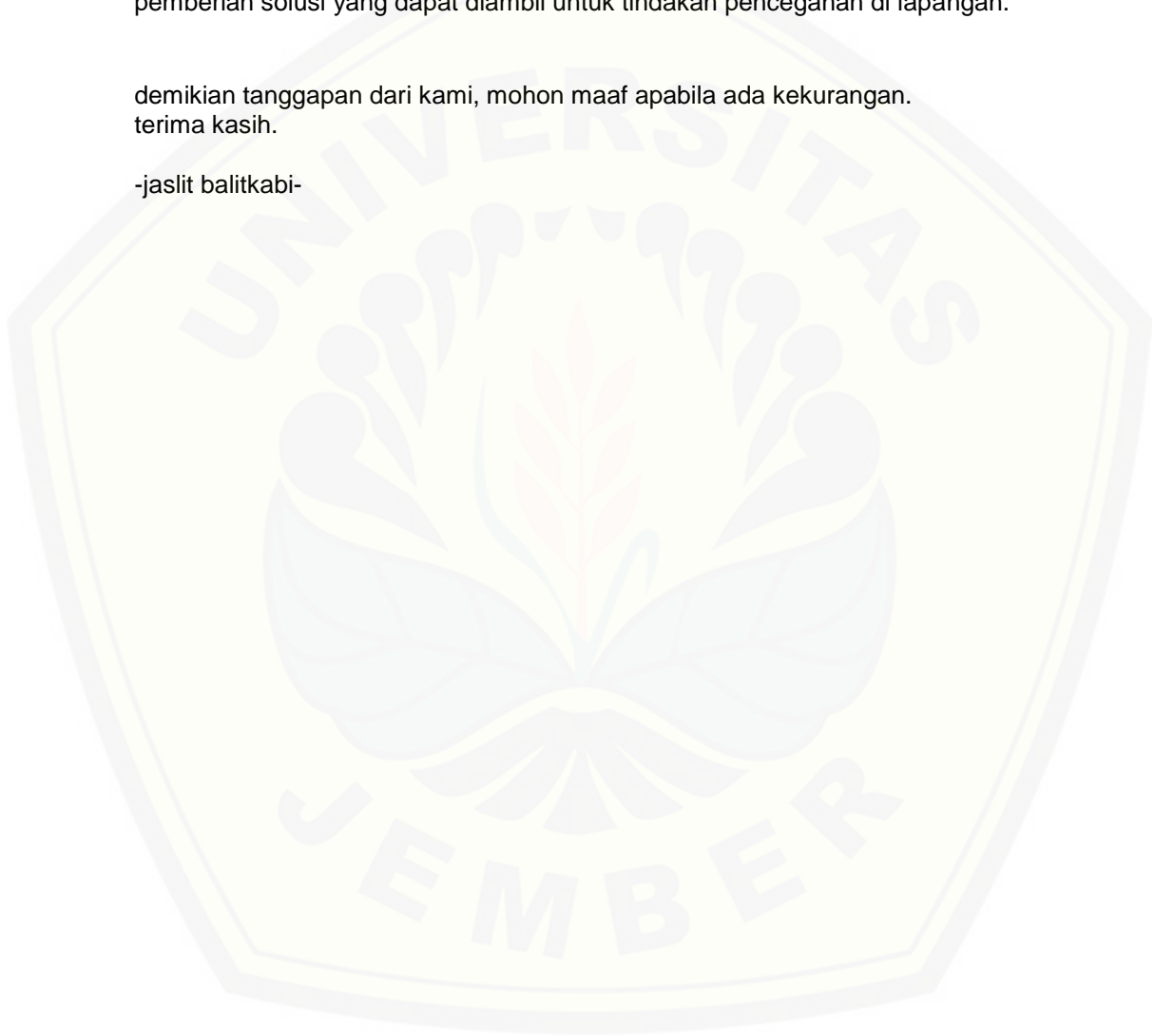
Hanya sebagai informasi tambahan saja, bukan sebagai input dari aplikasi. Memuat foto dan deskripsi mengenai penyakit yang terdapat di lingkup yang dipilih.

Terkait fitur nomor 3:

Bisa diunduh dan dibaca booklet mengenai “masalah hama, penyakit, dan hara pada tanaman kedelai” jika yang diambil merupakan hama mengenai tanaman kedelai. Salahsatu bagian di dalam buku tersebut berisi deskripsi mengenai hama dalam penyakit tanaman kedelai, beserta solusi pengendalian seperti apa yang bisa diterapkan. Akan tetapi booklet ini kami rasa memiliki kekurangan dalam hal pemberian solusi yang dapat diambil untuk tindakan pencegahan di lapangan.

demikian tanggapan dari kami, mohon maaf apabila ada kekurangan.
terima kasih.

-jaslit balitkabi-



D.3 Pertanyaan Tahap II



hofi atmajaya <hofiatmajaya2@gmail.com>

(tanpa subjek)

hofi atmajaya <hofiatmajaya2@gmail.com>
Kepada: balitkabi@gmail.com

5 September 2017 05.39

Assalamualaikum wr. wb. salam sejahtera. Saya Hofi Atmajaya dari PS. Sistem Informasi Universitas Jember. Berhubung saya sedang melaksanakan skripsi yg didalamnya ttg Identifikasi Hama dan penyakit pada tanaman Kedelai berbasis Android, dan sedang membutuhkan pengujian kebenaran hasil dari aplikasi yg dibangun. Yang ingin ditanyakan adalah,

1. Apakah memungkinkan dari Balitkabi bersedia membantu saya untuk pengujian keabsahan app yg dibangun? (Dokumentasi pengujian saya sediakan juga dlm bentuk kuisisioner sbg bukti saya kepada dosen untuk pengujian dg pakar dari Balitkabi)

2. Jika dari Bapak/Ibu dari Balitkabi bersedia membantu, apakah syarat administrasi yang harus saya penuhi? (seperti surat dari fakultas, bakesbangpol, biaya untuk Balitkabi, dsb.)

Demikian dari saya Bapak/Ibu, Terima kasih banyak, wassalamualaikum Wr. Wb.

D.4. Jawaban Tahap II



hofi atmajaya <hofiatmajaya2@gmail.com>

(tanpa subjek)

Seksi Jasa Penelitian Balitkabi <jaslit.balitkabi@gmail.com>6 September 2017
07.59

Kepada: hofi atmajaya <hofiatmajaya2@gmail.com>

Waalaikum salam wr wb.

Menanggapi email saudara perihal validasi hasil aplikasi identifikasi hama dan penyakit pada tanaman kedelai, apakah aplikasi saudara tersebut dlm bentuk sistem pakar, DSS, aplikasi murni berbasis android. untuk jelasnya silakan saudara datang ke kantor kami dengan membawa (proposal, surat pengantar dari kampus).

Demikian yang bisa saya informasikan.

Jaslit Balitkabi