

**PENGEMBANGAN SISTEM PAKAR PENYAKIT TANAMAN
PERKEBUNAN BERBASIS ANDROID**

SKRIPSI



Disusun Oleh :

**Dedy Ega Pratama Cahyo
12.18.268**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2016**

3412

UNCLASSIFIED//FOR OFFICIAL USE ONLY
EXCLUDED FROM AUTOMATIC DOWNGRADING
AND DECLASSIFICATION

12/12/2000
DATE OF REVIEW
12/12/2000

3412

EXCLUDED FROM AUTOMATIC DOWNGRADING
AND DECLASSIFICATION

LEMBAR PERSETUJUAN

PENGEMBANGAN SISTEM PAKAR PENYAKIT TANAMAN PERKEBUNAN BERBASIS ANDROID

SKRIPSI

*Disusun dan Diajukan untuk melengkapi dan memenuhi persyaratan guna
mencapai Gelar Sarjana Komputer Strata Satu (S-1)*

Disusun Oleh :

Dedy Ega Pratama Cahyo

12.18.268

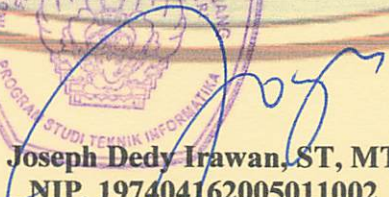
Diperiksa dan disetujui oleh

Dosen Pembimbing I

Dosen Pembimbing II


Karina Auliasari, ST, M. Eng
NIP.P 1031000426


Febriana Santi Wahyuni, S.Kom, M.Kom
NIP.P 1031000425


Mengetahui,
Ketua Program Studi Teknik Informatika S-1

Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2016

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama : Dedy Ega Pratama Cahyo
NIM : 12.18.268
Program Studi : Teknik Informatika S-1
Fakultas : Fakultas Teknologi Industri

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam skripsi yang berjudul : **“Pengembangan Sistem Pakar Penyakit Tanaman Perkebunan Berbasis Android”** Adalah skripsi sendiri bukan duplikasi serta mengutip atau menyadur seluruhnya karya orang lain kecuali sumber lainnya

Malang, Januari 2016



Dedy Ega Pratama Cahyo
NIM 12.18.268

Abstrak

Penyakit merupakan serangan terhadap tanaman yang diakibatkan oleh hama atau pathogen. Apabila tanaman yang terserang oleh penyakit dapat mengakibatkan tanaman itu mati dan penyakit akan menyebar ke tanaman yang lain jika tidak segera ditanggulangi oleh petani.

Sistem pakar adalah sistem yang menggunakan pengetahuan manusia yang terekam dalam komputer untuk memecahkan persoalan yang biasanya memerlukan keahlian manusia. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja para ahli. Sistem pakar terdiri dari dua pokok yaitu: pengembangan dan konsultasi. Lingkungan pengembangan digunakan oleh pembangun sistem pakar untuk membangun komponen dan memasukan pengetahuan ke dalam basis pengetahuan Metode Forward Chaining adalah metode pencarian atau teknik pelacakan kedepan yang dimulai dengan informasi yang ada dan penggabungan rule untuk menghasilkan suatu kesimpulan atau tujuan. Metode inferensi runut maju cocok digunakan untuk menangani masalah pengendalian (controlling) dan peramalan Teknik Forward Chaining merupakan teknik yang sering digunakan untuk proses inferensia yang memulai penalarannya dan sekumpulan data menuju kesimpulan yang dapat ditarik. Teknik Forward Chaining yaitu metode penalaran yang bergerak dan IF part menuju THEN part.

Hasil pengujian fungsional sistem untuk admin mulai dari halaman login, tambah penyakit, edit penyakit, hapus penyakit, tambah gejala, edit gejala, hapus gejala, tambah relasi dan hapus relasi dengan 100% berhasil. Hasil pengujian fungsional sistem untuk client mulai dari menampilkan splash screen, menampilkan gejala, proses diagnosa, menampilkan penyakit, menampilkan detail penyakit dan menampilkan tentang dengan 100% berhasil. Hasil pengujian pada sistem operasi mulai dari android Jellybean 4.2.2, Android Kitkat 4.4.2, dan android Lolipop 5.0.0 dengan 100% berhasil. Hasil pengujian akurasi dengan gejala yang dimasukan dan hasil penyakit didapat dari sistem penyakit tanaman perkebunan dan hasil penyakit tanaman perkebunan dari pakar 100% berhasil.

Kata Kunci : penyakit, Sistem Pakar , Forward Chaining

KATA PENGANTAR

Puji syukur terhadap kehadiran Allah SWT atas limpahan rahmat serta hidayah yang diberikan untuk menuntaskan Skripsi dengan lancar.

Skripsi yang berjudul **“PENGEMBANGAN SISTEM PAKAR PENYAKIT TANAMAN PERKEBUNAN BERBASIS ANDROID”** ini dilakukan untuk memenuhi salah satu persyaratan kelulusan di Institut Teknologi Nasional Malang Jurusan Teknologi Industri, Program Studi Teknik Informatika. Namun demikian, sangat disadari bahwa skripsi ini masih jauh dari kesempurnaan yang tak lepas dari kesalahan dan kekurangan, sehingga diharapkan dapat diperbaiki dan disempurnakan dikemudian hari.

Pada kesempatan ini penulis mengucapkan terimakasih kepada :

1. Bapak Dr. Ir. Lalu Mulyadi, MT, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Anang Subardi, MT, selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Joseph Dedy Irawan, ST, MT, selaku Ketua Program Studi Teknik Informatika Institut Teknologi Nasional Malang.
4. Bapak Sonny Prasetyo, ST, MT, selaku Sekretaris Program Studi Teknik Informatika Institut Teknologi Nasional Malang.
5. Ibu Karina Auliasari, ST. M.Eng, selaku Dosen Wali, yang telah memberikan motivasi, bimbingan, masukan, dan saran selama masa perkuliahan.
6. Ibu Karina Auliasari, ST. M.Eng, selaku Pembimbing Utama dan Ibu Febriana Santi Wahyuni, S.Kom, M.Kom, selaku Pembimbing pendamping, yang dengan sabar telah banyak memberikan bimbingan, motivasi, dan saran dalam proses pembuatan skripsi ini.
7. Bapak dan Ibu dosen Program Studi Teknik Informatika Institut Teknologi Nasional Malang yang telah memberikan pengetahuan selama masa perkuliahan.

8. Seluruh staf dan karyawan Program Studi Teknik Informatika Institut Teknologi Nasional Malang yang telah banyak memberikan bantuan selama perkuliahan.
9. Orang tua yang selalu memberikan dukungan terhadap penulis di Institut Teknologi Nasional Malang, yaitu dukungan moral dan financial selama ini.
10. Teman-teman Teknik Informatika yang telah membantu dalam proses penyelesaian skripsi ini.
11. Serta pihak-pihak lain yang tak dapat disebutkan satu persatu disini yang telah banyak memberikan bantuan demi terselesaikannya skripsi ini.

Akhir kata, semoga skripsi ini banyak memberikan manfaat kepada penulis sendiri khususnya dan pembaca sekalian pada umumnya.

Malang, Januari 2016

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN DAN PENGESAHAN.....	ii
SURAT PERNYATAAN KEASLIAN SKRIPSI.....	iii
ABSTRAK	iv
KATA PENGANTAR.....	v
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Metode Penelitian.....	3
1.5.1 Landasan Teori	3
1.5.2 Desain Aplikasi.....	3
1.5.3 Impelementas.....	3
1.5.4 Uji Coba.....	3
1.6 Sistemmatika Penulisan.....	4
BAB II LANDASAN TEORI	5
2.1 Sistem Pakar	5
2.2 Kompone Sistem Pakar	5
2.2.1 Subsistem Penambahan Pengetahuan	5
2.2.2 Basis Pengetahuan.....	5
2.2.3 Motor Inferensi.....	6
2.2.4 Workspace.....	6
2.2.5 Antarmuka.....	7
2.2.6 Subsistem	7
2.2.7 Perbaikan.....	7

2.3	Ciri – Ciri Sistem Pakar	7
2.3.1	Manfaat Sistem Pakar	8
2.4	Forward Chaining.....	8
2.5	Eclips	10
2.6	JSON	11
BAB III ANALISA DAN PERANCANGAN		13
3.1	Analisa.....	13
3.1.1	Analisa Kebutuhan.....	13
3.1.1.1	Akuisisi dan Representasi.....	13
3.2	Perancangan.....	27
3.2.1	Mesin Inferensi.....	27
3.2.2	Basis Data	28
3.2.3	Antarmuka.....	30
BAB IV IMPLEMENTASI DAN PENGUJIAN		39
4.1	Implementasi Sistem	39
4.1.1	Proses Login Pakar.....	39
4.1.2	Halaman Utama Pakar	40
4.1.3	Penginputan Data	40
4.1.3.1	Form Input Data Gejala.....	40
4.1.3.2	Form Input Data Penyakit	41
4.1.3.3	Form Input Relasi.....	41
4.1.4	Halaman Menu Utama	42
4.1.5	Halaman Edit Gejala	42
4.1.6	Halaman Edit Penyakit.....	43
4.1.7	Halaman Menu Relasi	43
4.1.8	Halaman Edit Data Relasi	44
4.1.9	Halaman Utama User	44
4.1.10	Halaman Diagnosa	45
4.1.11	Halaman Gejala.....	45
4.1.12	Halaman Hasil.....	46
4.1.13	Halaman Macam – Macam Pantogen	46
4.1.14	Halaman Detail Pantogen.....	47

4.2 Pengujian Sistem	47
4.2.1 Pengujian Fungsional Admin	47
4.2.2 Pengujian Fungsional Aplikasi	48
4.2.3 Pengujian Akurasi.....	49
BAB V PENUTUP.....	51
5.1 Kesimpulan.....	51
5.2 Saran.....	51
DAFTAR PUSTAKA	52

DAFTAR TABEL

Tabel 3.1 Tabel Data Gejala.....	13
Tabel 3.2 Tabel Data Penyakit	16
Tabel 3.3 Tabel Rule Tanaman Kelapa.....	18
Tabel 3.4 Tabel Rule Tanaman Kopi	18
Tabel 3.5 Tabel Rule Tanaman Cengkeh	19
Tabel 3.6 Tabel Rule Tanaman Karet	19
Tabel 3.7 Tabel Rule Tanaman Teh	20
Tabel 3.8 Tabel Rule Tanaman Coklat.....	20
Tabel 3.9 Tabel Tanaman.....	29
Tabel 3.10 Tabel Data Gejala.....	29
Tabel 3.11 Tabel Data Penyakit	29
Tabel 3.12 Tabel Data Pantogen	30
Tabel 3.13 Tabel Rule	30
Tabel 3.14 Tabel User	30
Tabel 4.1 Hasil Pengujian Fungsional Web Admin.....	47
Tabel 4.3 Hasil Pengujian Fungsional Aplikasi	48
Tabel 4.3 Hasil Pengujian Pada Beberapa sistem Opreasi.....	49
Tabel 4.4 Hasil Pengujian Akurasi.....	49

DAFTAR GAMBAR

Gambar 2.1 Penyelesaian Forward Chaining.....	9
Gambar 3.1 Pohon Keputusan Tanaman Kelapa	22
Gambar 3.2 Pohon Keputusan Tanaman Kopi.....	23
Gambar 3.3 Pohon Keputusan Tanaman Cengkeh.....	24
Gambar 3.4 Pohon Keputusan Tanaman Karet.....	25
Gambar 3.5 Pohon Keputusan Tanaman Teh.....	26
Gambar 3.6 Pohon Keputusan Tanaman Coklat	27
Gambar 3.7 Flowchart Mesin Inferensi.....	28
Gambar 3.8 Perancangan halaman Login Admin	31
Gambar 3.9 Perancangan Halaman Utaman Admin	31
Gambar 3.10 Perancangan Halaman Tanaman	32
Gambar 3.11 Perancangan Halaman Edit Gejala	32
Gambar 3.12 Perancangan Halaman Edit Penyakit	33
Gambar 3.13 Perancangan Halaman Tambah Gejala.....	33
Gambar 3.14 Perancangan Halaman Tambah Penyakit	34
Gambar 3.15 Perancangan Halaman Edit Rule.....	34
Gambar 3.16 Perancangan Halaman Tambah Rule	35
Gambar 3.17 Perancangan Halaman Utama User	35
Gambar 3.18 Perancangan Halaman Diagnosa	36
Gambar 3.19 Perancangan Halaman Gejala.....	36
Gambar 3.20 Perancangan Halaman Hasil.....	37
Gambar 3.21 Perancangan Halaman Macam – Macam Pantogen	37
Gambar 3.22 Perancangan Halaman Detail Pantogen.....	38
Gambar 3.23 Perancangan Halaman About	38
Gambar 4.1 Halaman Login Pakar.....	39
Gambar 4.2 Halaman Utama Pakar	40
Gambar 4.3 Halaman Input Data Gejala	40
Gambar 4.4 Halaman Input Data Penyakit.....	41
Gambar 4.5 Halaman Input Data Relasi.....	41

Gambar 4.6 Halaman Menu Tanaman	42
Gambar 4.7 Halaman Edit Data Gejala	42
Gambar 4.8 Halaman Edit Data Penyakit	42
Gambar 4.9 Menu Relasi.....	42
Gambar 4.10 Edit Data relasi	42
Gambar 4.11 Halaman Utama User	43
Gambar 4.12 Halaman Diagnosa.....	44
Gambar 4.13 Halaman Gejala	45
Gambar 4.14 Halaman Hasil	46
Gambar 4.15 Halaman Macam – Macam Pantogen.....	46
Gambar 4.16 Halaman Detail Pantogen	47

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pertumbuhan dan perkembangan tanaman dari sejak benih, pembibitan, pemanenan, hingga di gudang penyimpanan selalu tidak luput dari gangguan hama, pantoge, gulma atau karena faktor-faktor lingkungan yang tidak sesuai bagi tanaman. Akibat gangguan tersebut seorang peneliti dari India pernah menyatakan bahwa kerugian tanaman akibat gangguan gulma 33%, pantogen 26%, serangga hama 20%, tikus 6%, dan kerusakan di penyimpanan sekitar 7% (Nur Tjahjadi, 1996). Jika penggaggu tanaman tersebut mengganggu secara serentak maka kerugian tanaman akan lebih besar. Hal ini belum termasuk gangguan karena faktor lingkungan. Demikian besarnya peran pengganggu tanaman terhadap penghasilan petani, tetapi masih banyak petani yang kurang tentang hama, pantogen, gulma, dan cara pengendaliannya yang kurang diketahui oleh petani.

Dalam hal ini ahli pertanian mempunyai kemampuan untuk menganalisa gejala-gejala yang penyakit pada tanaman perkebunan. Tetapi untuk mengatasi semua persoalan yang dihadapi oleh para petani terkendala oleh waktu dan banyak petani yang mempunyai masalah dengan tanaman perkebunan, dan kebanyakan aplikasi sistem pakar yang sudah ada hanya berbasis aplikasi desktop. Apabila ada timbul gejala – gejala pada tanaman perkebunan petani tidak dapat mengetahui penyakit yang terjadi pada tanaman perkebunan. Dan tidak semua petani akan selalu berada didepan komputer karena petani akan selalu berkeliling ke lahan pertanian untuk melihat tanaman perkebunan yang dimiliki mengalami pertumbuhan atau mengalami penyakit yang disebabkan oleh gulma, pantogen, atau hama.

Mengacu pada kasus tersebut, penulis merasa tergugah untuk membuat suatu aplikasi sistem pakar berbasis android untuk kepentingan petani mengetahui secara cepat penyakit yang terjadi pada tanaman perkebunan tersebut dan cara mengatasi penyakit yang terjadi tanpa harus menunggu para ahli pertanian melakukan

sosialisasi tentang penyakit apa aja yang bisa menyerang tanaman perkebunan dan cara menanggulangi penyakit yang terjadi pada tanaman perkebunan

1.2 Rumusan Masalah

Bagaimana menerapkan metode *forward chaining* dalam pengembangan sistem pakar untuk mengidentifikasi penyakit tanaman perkebunan berbasis android ?

1.3 Tujuan Penelitian

Adapun yang menjadi tujuan penulisan dalam penyusunan proposal skripsi adalah sebagai berikut :

1. Membuat aplikasi pengembangan sistem pakar penyakit tanaman perkebunan berbasis android dengan menggunakan metode *Forward Chaining*.
2. Menganalisa hasil *forward chaining* harus sesuai dengan fakta – fakta yang telah di *input*.

1.4 Batasan Masalah

1. Jenis - jenis tanaman yang dapat teridentifikasi meliputi tanaman kelapa, teh, kopi, coklat, cengkeh.
2. Sistem dikembangkan dengan bahasa pemrograman Java.
3. Data penyakit dan gejala untuk membangun basis pengetahuan diambil dari buku Hama dan Penyakit Tanaman yang berisi detail penyakit,gejala dan solusi untuk penyakit tanaman kelapa, kopi, cengkeh, teh, karet,coklat (Nur Tjahjadi,1996).
4. Pada penelitian ini jumlah total penyakit untuk tanaman ada 34 penyakit dan gejala ada 55 gejala.
5. Sistem dikembangkan berbasis Android yang compatible dengan sistem operasi Android *version* Jellybin.

1.5 Metode Penelitian

Dalam penyusunan penelitian Skripsi, untuk dapat mencapai keinginan penulis membutuhkan informasi dan metode yang berhubungan dengan tema yang akan di kupas oleh penulis, yaitu mengenai konsep dalam pembuatan Sistem Pakar Pentuan penyakit Pada Tanaman Perkebunan. Maka perlu dilakukan dengan langkah – langkah sebagai berikut :

1.5.1 Landasan Teori

Pada tahap tinjauan pustaka, penulis melakukan pembelajaran melalui buku dan jurnal mengenai sistem pakar. Pada tahap tinjauan pustaka penulis mencari landasan teori dan metode yang digunakan pada penelitian yaitu metode *Forward Chaining*.

1.5.2 Desain Aplikasi

Pada tahapan desain aplikasi, penulis akan mengidentifikasi komponen-komponen sistem yang akan digunakan secara rinci yang meliputi perancangan algoritma matematis sistem, desain *flowchart*, desain blok diagram penentuan *input*, *process* dan *output*, dan lain-lain yang dapat mendukung perancangan

1.5.3 Implementasi

Setelah proses desain aplikasi telah dilakukan. Langkah selanjutnya adalah melakukan implementasi pada desain yang telah dibuat. Tahap ini adalah penerapan terhadap rancangan algoritma, *flowchart* dan blok diagram pada sebuah bahasa pemrograman. Digunakan untuk implementasi aplikasi ini adalah *Eclips*.

1.5.4 Uji Coba

Pada tahap uji coba untuk mencari kesalahan atau kekurangan dari program atau aplikasi yang telah dibuat agar dapat dicapai tujuan yang diinginkan. Pada tahap uji coba, akan dilakukan diagnosa dengan memasukan fakta – fakta yang ada dan akan diproses.

1.6 Sistematika Penulisan

Dalam penyusunan skripsi ini agar lebih mudah dipahami maka dibuatlah suatu sistematika penulisan sebagai berikut:

BAB I : PENDAHULUAN

Berisi latar belakang, perumusan masalah, batasan masalah, tujuan dan metodologi penelitian, desain aplikasi, implementasi, uji coba.

BAB II : LANDASAN TEORI

Berisi teori-teori yang didapat dari studi literature dan konsep-konsep yang menunjang dalam proses pembuatan tugas akhir ini, beserta dengan penyelesaian masalah yang diambil dalam penyusunan tugas akhir.

BAB III : PERANCANGAN APLIKASI

Bab ini berisi tentang desain dan perancangan sistem yang akan dibangun meliputi analisa sistem komponen sistem pencarian lokasi, rancangan basis data dan perancangan antarmuka.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Berisi tentang implementasi dan uji coba dari aplikasi.

BAB V : PENUTUP

Merupakan bab terakhir yang memuat inti sari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

BAB II

LANDASAN TEORI

2.1 Sistem Pakar

Sistem pakar adalah sistem yang menggunakan pengetahuan manusia yang terekam dalam komputer untuk memecahkan persoalan yang biasanya memerlukan keahlian manusia. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja para ahli. Sistem pakar terdiri dari dua pokok yaitu: pengembangan dan konsultasi. Lingkungan pengembangan digunakan oleh pembangun sistem pakar untuk membangun komponen dan memasukan pengetahuan ke dalam basis pengetahuan. Lingkungan konsultasi digunakan oleh orang yang bukan ahli untuk memperoleh pengetahuan (Honggowibowo, 2009).

2.2 Komponen Sistem Pakar

2.2.1 Subsistem Penambahan Pengetahuan

Bagian ini digunakan untuk memasukkan pengetahuan, mengkonstruksi atau memperluas pengetahuan dalam basis pengetahuan. Pengetahuan itu bisa berasal dari: ahli, buku, basis data, penelitian, dan gambar. (Ulysses, 2012).

2.2.2 Basis Pengetahuan

- a. Berisi pengetahuan-pengetahuan yang dibutuhkan untuk memahami, memformulasikan dan menyelesaikan masalah.
- b. Basis pengetahuan tersusun atas fakta dan kaidah.
- c. Fakta adalah informasi tentang objek, peristiwa, atau situasi. Kaidah adalah cara untuk membangkitkan suatu fakta baru dari fakta yang sudah diketahui.

2.2.3 Motor Inferensi

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

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

Dalam prosesnya, mesin/motor inferensi menggunakan strategi penalaran

- a. Strategi penalaran pasti (Exact Reasoning)

Dilakukan jika semua data yang dibutuhkan untuk menarik suatu kesimpulan tersedia.

- b. Strategi penalaran tak pasti (Inexact Reasoning)

Dilakukan jika semua data yang dibutuhkan untuk menarik suatu kesimpulan tidak tersedia.

- c. Strategi pengendalian.

Berfungsi sebagai panduan arah dalam melakukan proses penalaran. Terdapat tiga teknik pengendalian yang sering digunakan, yaitu:

- a. Forward chaining
- b. Backward chaining
- c. Gabungan kedua teknik

2.2.4 Workspace

Merupakan area dalam memori yang digunakan untuk merekam kejadian yang sedang berlangsung termasuk keputusan sementara. Ada 3 tipe keputusan yang dapat direkam yaitu:

- a. Rencana: bagaimana menghadapi masalah
- b. Agenda: aksi-aksi yang potensial yang sedang menunggu untuk dieksekusi
- c. Solusi: calon aksi yang akan dibangkitkan

2.2.5 Antarmuka

Merupakan batasan antara dua entitas, berfungsi sebagai penengah untuk media komunikasi antara user dan program Contoh antarmuka untuk pengguna:

- a. GUI (*graphical user interface*).
- b. WUI (*web based user interface*).

2.2.6 Subsistem

Digunakan untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif melalui pertanyaan:

- a. Mengapa suatu pertanyaan ditanyakan oleh sistem pakar?
- b. Bagaimana konklusi dicapai?
- c. Mengapa ada alternatif yang dibatalkan?
- d. Rencana apa yang digunakan untuk mendapatkan solusi?

2.2.8 Perbaikan

Sistem ini digunakan untuk mengevaluasi kinerja sistem pakar itu sendiri untuk melihat apakah pengetahuan-pengetahuan yang ada masih cocok untuk digunakan di masa mendatang.

2.3 Ciri – Ciri Sistem Pakar

Ciri – ciri sistem pakar (Siswanto, 2000) antara lain :

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data yang tidak pasti.
3. Dapat mengemukakan rangkaian alasan – alasan yang diberikannya dengan cara yang dapat dipahami.
4. Berdasarkan pada kaidah/ketentuan/*rule* tertentu.

5. Dirancang untuk dapat dikembangkan secara bertahap.
6. Pengetahuan & mekanisme penalaran (*inference*) jelas terpisah.
7. Keluarannya bersifat anjuran.
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai dituntun oleh dialog dengan user.

2.3.1 Manfaat Sistem Pakar

Banyaknya kemampuan dan manfaat yang diberikan oleh sistem pakar (Achmad, 2006) diantaranya :

1. Meningkatkan output dan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari manusia
2. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
3. Mampu menangkap kepakaran yang sangat terbatas.
4. Dapat beroperasi dilingkungan yang berbahaya.
5. Memudahkan akses ke pengetahuan.
6. Handal. Sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit. Sistem pakar juga secara konsisten melihat semua detil dan tidak akan melewatkan informasi yang relevan dan solusi yang potensial.
7. Mampu menyediakan pelatihan. Pengguna permula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman. Fasilitas penjelas dapat berfungsi sebagai guru.
8. Meningkatkan kemampuan *problem solving*, karena mengambil sumber pengetahuan dari banyak pakar.

2.4 Forward Chaining

Metode *Forward Chaining* adalah metode pencarian atau teknik pelacakan kedepan yang dimulai dengan informasi yang ada dan penggabungan rule untuk menghasilkan suatu kesimpulan atau tujuan. Metode inferensi runut maju cocok digunakan untuk menangani masalah pengendalian (controlling) dan peramalan. Teknik *Forward Chaining* merupakan teknik yang sering digunakan untuk proses inferensia yang memulai penalarannya dan sekumpulan data menuju

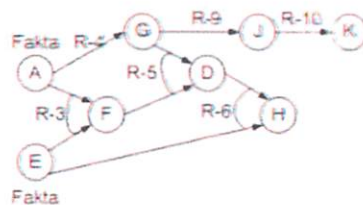
kesimpulan yang dapat ditarik. Teknik Forward Chaining yaitu metode penalaran yang bergerak dari IF part menuju THEN part (Wicaksono, 2014).

Dicontohkan pada tabel di bawah ini terlihat 10 aturan yang tersimpan dalam basis pengetahuan. Fakta awal yang diberikan hanya A & F (artinya A dan F bernilai benar). Ingin dibuktikan apakah K bernilai benar (hipotesis : K) ?

Tabel 2.1 Contoh Sistem Perantaraan Maju

No	Aturan
R-1	IF A & B THEN C
R-2	IF C THEN D
R-3	IF A & D THEN F
R-4	IF A THEN G
R-5	IF F & G THEN D
R-6	IF G & E THEN H
R-7	IF C & H THEN I
R-8	IF I & A THEN J
R-9	IF G THEN J
R-10	IF J THEN K

Penyelesaian dengan Forward Chaining pada Gambar berikut.



Gambar 2.1 Penyelesaian *forward chaining*

Langkah-langkah yang harus dilakukan dalam membuat sistem forward chaining berbasis aturan, yaitu:

a. Pendefinisian Masalah

Tahap ini meliputi pemilihan domain masalah dan akusisi pengetahuan.

b. Pendefinisian Data Input

Sistem forward chaining memerlukan data awal untuk memulai inferensi.

c. Pendefinisian Struktur Pengendalian Data.

Aplikasi yang kompleks memerlukan premis tambahan untuk membantu mengendalikan pengaktifan suatu aturan.

d. Penulisan Kode Awal

Tahap ini berguna untuk menentukan apakah sistem telah menangkap domain pengetahuan secara efektif dalam struktur

aturan yang baik.

e. Pengujian Sistem

Pengujian sistem dilakukan dengan beberapa aturan untuk menguji sejauh mana sistem berjalan dengan benar.

f. Perancangan Antarmuka

Antarmuka adalah salah satu komponen penting dari suatu sistem. Perancangan antarmuka dibuat bersama-sama dengan pembuatan basis pengetahuan.

g. Pengembangan Sistem

Pengembangan sistem meliputi penambahan antarmuka dan pengetahuan sesuai dengan prototipe sistem.

h. Evaluasi Sistem.

Pada tahap ini dilakukan pengujian sistem dengan masalah yang sebenarnya. Jika sistem belum berjalan dengan baik maka akan dilakukan pengembangan kembali

2.5 Eclipse

Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform

(platform-independent). Berikut ini adalah sifat dari Eclipse Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X. Multi-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya. Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan open source, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in (Kadir, 2013).

2.6 JSON

JSON adalah kependekan dari Javascript Object Notation. JSON dipakai untuk dua hal; yang pertama untuk format transfer data antar device/OS/bahasa pemrograman yang berbeda dan sebagai media penyimpanan data seperti MongoDB. JSON dalam artikel ini lebih menekankan pada transfer data. Sebenarnya selain JSON, format transfer data antar sistem bisa dilakukan XML, plaintext ataupun HTML, namun 3 format tersebut punya banyak kekurangan, terutama dari sisi ukuran. Saingan terdekat dari JSON adalah XML, dalam layanan Web API atau webservice, sebagian layanan menggunakan JSON, sebagian yang lain menggunakan format XML. Dua format ini bisa saling menggantikan. berikut ini adalah contoh data JSON tentang event kegiatan (Kadir, 2013).

Source code

```
{ "list_event":
[
  { "eventID": "1",
    "judul": "Pengajian Akbar menyambut ramadhan",
```

```

    "tanggal": "2014-05-19",
    "jam": "20:00",
    "lokasi": "Masjid Kampus UGM"
  },
  { "eventID": "2",
    "judul": "Bedah buku kesesatan syiah",
    "tanggal": "2014-05-21",
    "jam": "09:00",
    "lokasi": "Masjid Kampus UII "
  }
]
}

```

list_event adalah adalah Array JSON. Array terdiri dari beberapa object. Dalam kasus ini, kita punya 2 object event. Masing masing object event ini mempunyai 5 field

BAB III

ANALISIS DAN PERANCANGAN

3.1. Analisis

Pada bagian analisis dilaksanakan kajian untuk meneliti struktur secara lebih dalam, kajian yang dilaksanakan meliputi : Analisa Kebutuhan.

3.1.1. Analisa Kebutuhan

Analisa kebutuhan merupakan penguraian kebutuhan untuk membangun sebuah aplikasi sistem pakar, kajian yang dilaksanakan meliputi : Akuisisi dan Representasi Pengetahuan.

3.1.1.1. Akuisisi dan Representasi Pengetahuan

Akuisisi dan Representasi Pengetahuan merupakan pengumpulan sebuah data – data dari dokumen atau pakar yang meliputi : data gejala , data penyakit

1. Data Gejala

Data gejala merupakan gejala – gejala yang terjadi pada sebuah tanaman untuk mengidentifikasi penyakit yang dialami oleh tanaman.

Tabel 3.1 Data gejala

Kode Gejala	Uraian	Jenis Tanaman
G001	Timbul bercak pada daun masih muda	Kelapa
G002	Daun berguguran	Kelapa
G003	Daun bewarna coklat pada bagian luar	Kelapa
G004	Tunas lepas karena pangkal membusuk	Kelapa
G005	Buah muda berguguran	Kelapa
G006	Akar kelapa busuk	Kelapa
G007	Daun muda lebih pendek	Kelapa
G008	Pohon mengecil	Kelapa

G009	Produksi buah menurun	Kelapa
G010	Daun menguning	Kelapa
G011	Daun menjadi layu	Kelapa
G012	Pelepah daun bergelantungan	Kelapa
G013	Ujung daun mengerut dan kering	Kelapa
G001	Akar muncul di permukaan tanah	Kopi
G002	Batang terdapat cendawan berwarna hitam	Kopi
G003	Pangkal batang menjadi busuk	Kopi
G004	Cendawan terdapat pada ranting	Kopi
G005	Daun gugur	Kopi
G006	Ranting – ranting pohon mati	Kopi
G007	Daun timbul bercak kuning kecoklatan	Kopi
G008	Daun diselimuti cendawan	Kopi
G009	Terdapat cendawan berwarna coklat	Kopi
G001	Daun menguning	Cengkeh
G002	Tanaman menjadi layu	Cengkeh
G003	Daun timbul bercak	Cengkeh
G004	Daun lebih cepat gugur	Cengkeh
G005	Daun yang tersisa berwarna hijau suram	Cengkeh
G001	Daun timbul bercak	Karet
G002	Daun berwarna pucat	Karet
G003	Ranting – Ranting mati	Karet
G004	Akar dililit cendawan berwarna putih	Karet
G005	Daun terdapat massa tepung berwarna putih pada permukaan daun	Karet
G006	Pada daerah bercak timbul warna kuning	Karet
G007	Daun muda berwarna kehitaman dan gugur	Karet
G008	Daun terdapat tepung berwarna hitam	Karet
G001	Daun timbul bercak selanjutnya bercak menjadi lubang	Teh

G002	Tanaman menjadi layu dan daun tidak gugur	Teh
G003	Daun bewarna merah tembaga	Teh
G004	Induk daun timbul bercak coklat	Teh
G005	Ujung tunas sampai bawah tunas mengering	Teh
G006	Ranting terdapat bercak coklat	Teh
G007	Ranting bewarna putih	Teh
G008	Ranting mengering	Teh
G009	Tanaman layu	Teh
G001	Buah timbul bercak hitam dan meluas ke seluruh buah	Coklat
G002	Bercak hitam pada batang	Coklat
G003	Lapisan kulit dibelah bewarna merah	Coklat
G004	Akar terdapat Rizomorfa tipis bewarna putih	Coklat
G005	Daun menguning dan akhirnya layu	Coklat
G006	Daun bergantung pada pohon sudah mati	Coklat
G007	Cendawan membentuk lapisan hifa warna putih	Coklat
G008	Batang dan ranting cabang mengering	Coklat
G009	Pangkal daun timbul 3 titik warna coklat	Coklat
G010	Ranting dibelah timbul garis coklat	Coklat

2. Data Penyakit

Data penyakit merupakan macam – macam penyakit yang dapat dialami oleh tanaman – tanaman perkebunan.

Tabel 3.2 Data Penyakit

Kode Penyakit	Uraian	Kode Tanaman
P01	Bercak daun	Kelapa
P02	Busuk tunas	Kelapa
P03	Busuk buah	Kelapa
P04	Cadang	Kelapa
P05	Penyakit kuning	Kelapa
P06	Penyakit busuk akar	Kelapa
P07	Penyakit layu	Kelapa
P01	Penyakit akar hitam	Kopi
P02	Penyakit akar coklat	Kopi
P03	Jamur upas	Kopi
P04	Busuk batang	Kopi
P05	Karat daun	Kopi
P06	Bercak daun	Kopi
P07	Embun jelaga	Kopi
P01	Busuk akar	Cengkeh
P02	Bercak daun	Cengkeh
P03	Cacar daun	Cengkeh
P04	Penyakit Mati Ranting	Cengkeh
P05	Penyakit Mati Bujang	Cengkeh
P01	Gugur daun Oidium	Karet
P02	Cendawan akar putih	Karet
P03	Gugur daun Corynespora	Karet
P04	Gugur Daun Colletotrichum	Karet

P05	Helminthosporium	Karet
P01	Cacar teh	Teh
P02	Penyakit akar	Teh
P03	Penyakit busuk daun	Teh
P04	Penyakit mati ujung	Teh
P05	Jamur upas	Teh
P01	Penyakit VSD	Coklat
P02	Busuk buah	Coklat
P03	Jamur upas	Coklat
P04	Kanker batang	Coklat
P05	Cendawan akar merah	Coklat

3. Tabel Rule

Tabel rule merupakan aturan untuk menentukan jenis penyakit yang terjadi pada tanaman perkebunan berdasarkan data gejala.

a. Tabel Rule Tanaman Kelapa

Pada tabel rule tanaman kelapa G001 sampai dengan G010 merupakan kode gejala yang terjadi pada tanaman kelapa. P01 sampai dengan P07 merupakan kode penyakit dari tanaman kelapa. Struktur tabel rule tanaman kelapa dapat dilihat pada tabel 3.3.

Tabel 3.3 Rule Tanaman Kelapa

Penyakit Gejala	P01	P02	P03	P04	P05	P06	P07
G001	X			X			
G002	X						
G003		X					
G004		X					
G005			X		X	X	
G006			X				
G007				X			

G008				X			
G009				X	X		
G010					X		X
G011						X	
G012						X	
G013							X

b. Tabel Rule Tanaman Kopi

Pada tabel rule tanaman kopi G001 sampai dengan G010 merupakan kode gejala yang terjadi pada tanaman kopi. P01 sampai dengan P07 merupakan kode penyakit dari tanaman kopi. Struktur tabel rule tanaman kopi dapat dilihat pada tabel 3.4.

Tabel 3.4 Rule Tanaman Kopi

Penyakit Gejala	P01	P02	P03	P04	P05	P06	P07
G001	X						
G002	X						
G003							
G005			X				
G006				X		X	
G007				X	X		
G008					X	X	
G009							X
G010		X					

c. Tabel Rule Tanaman Cengkeh

Pada tabel rule tanaman cengkeh G001 sampai dengan G005 merupakan kode gejala yang terjadi pada tanaman cengkeh. P01 sampai dengan P05 merupakan kode penyakit dari tanaman cengkeh. Struktur tabel rule tanaman cengkeh dapat dilihat pada tabel 3.5.

Tabel 3.5 Rule Tanaman Cengkeh

Penyakit Gejala	P01	P02	P03	P04	P05
G001	X			X	
G002	X			X	
G003		X	X		
G004			X	X	X
G005					X

d. Tabel Rule Tanaman Karet

Pada tabel rule tanaman karet G001 sampai dengan G008 merupakan kode gejala yang terjadi pada tanaman karet. P01 sampai dengan P05 merupakan kode penyakit dari tanaman karet. Struktur tabel rule tanaman karet dapat dilihat pada tabel 3.6.

Tabel 3.6 Rule Tanaman Karet

Penyakit Gejala	P01	P02	P03	P04	P05
G001	X		X	X	X
G002		X			
G003		X			
G004		X			
G005	X				
G006			X		
G007				X	
G008					X

e. Tabel Rule Tanaman Teh

Pada tabel rule tanaman teh G001 sampai dengan G009 merupakan kode gejala yang terjadi pada tanaman teh. P01 sampai dengan P05 merupakan kode penyakit dari tanaman teh. Struktur tabel rule tanaman teh dapat dilihat pada tabel 3.7.

Tabel 3.7 Rule Tanaman Teh

Penyakit Gejala	P01	P02	P03	P04	P05
G001	X				X
G002		X			
G003		X			
G004				X	
G005				X	
G006					X
G007			X		
G008					X
G009			X		

f. Tabel Rule Tanaman Coklat

Pada tabel rule tanaman coklat G001 sampai dengan G010 merupakan kode gejala yang terjadi pada tanaman coklat. P01 sampai dengan P05 merupakan kode penyakit dari tanaman coklat. Struktur tabel rule tanaman coklat dapat dilihat pada tabel 3.8.

Tabel 3.8 Rule Tanaman Coklat

Penyakit Gejala	P01	P02	P03	P04	P05
G001		X			
G002				X	

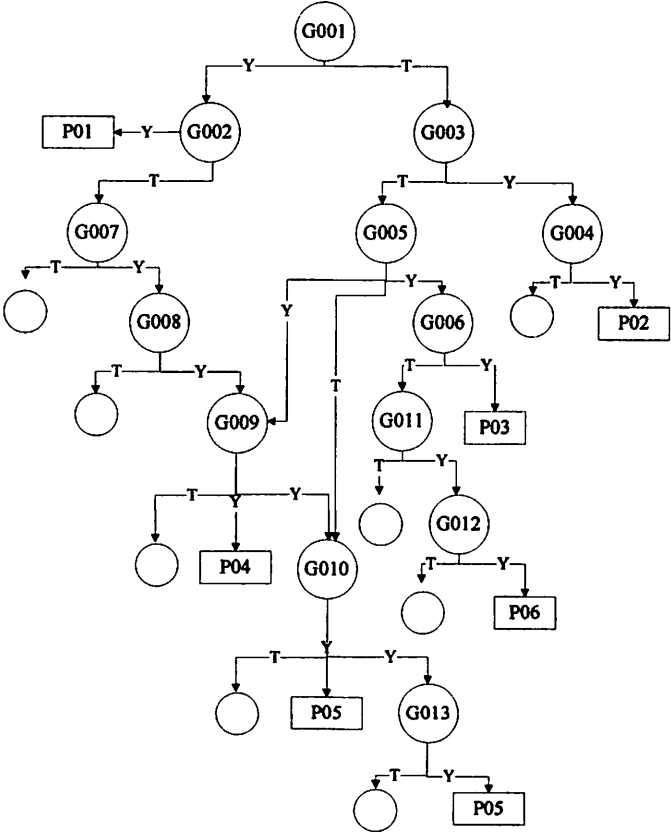
G003				X	
G004					X
G005					X
G006					X
G007			X		
G008	X				
G009	X				
G010	X				

5. Pohon Keputusan

Pohon Keputusan merupakan hal yang menentukan keberhasilan sistem yang dibangun, pohon keputusan yang digunakan untuk mendiagnosa penyakit. Pohon keputusan terdiri dari gejala dan penyakit yang menunjukkan hubungan antar objek. Berikut adalah gambar pohon keputusan yang terbentuk, dapat dilihat pada gambar berikut.

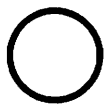
a. Pohon Keputusan Tanaman Kelapa

Pohon keputusan tanaman kelapa terdiri dari gejala dan penyakit pada tanaman kelapa yang menunjukkan hubungan antara gejala 1 dengan gejala yang lain akan menghasilkan sebuah penyakit sebagai contoh : G001 dengan G002 akan menghasilkan penyakit P01. Berikut adalah pohon keputusan yang terbentuk, dapat dilihat pada gambar 3.1.



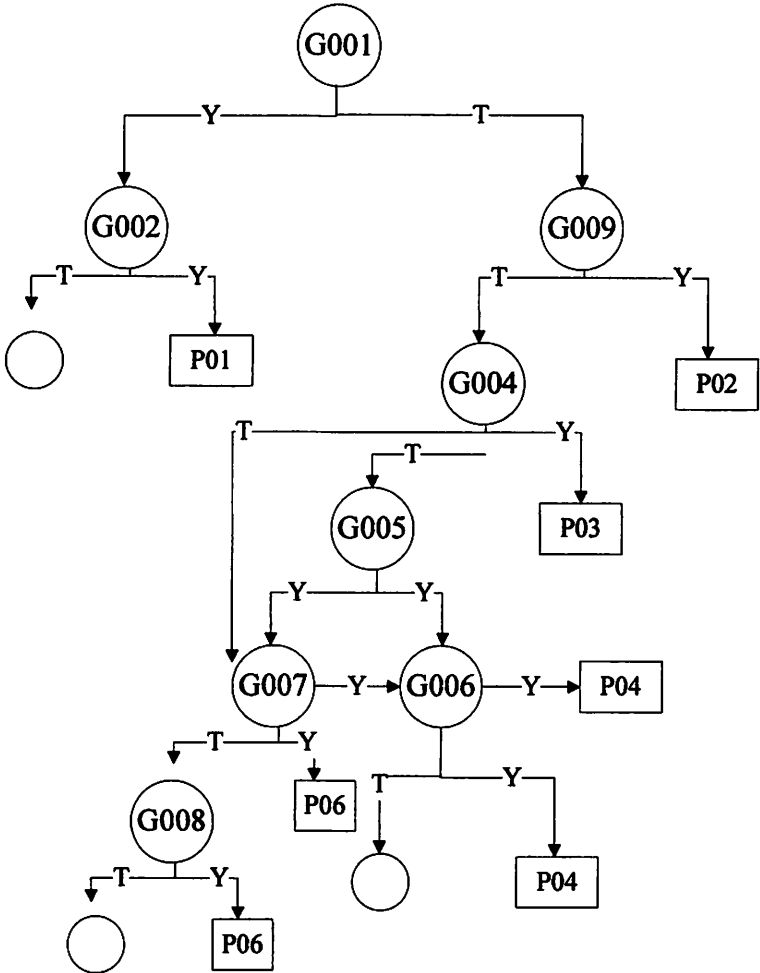
Gambar 3.1. Pohon Keputusan Tanaman Kelapa

Keterangan :

-  Lingkaran besar : Gejala pada tanaman
-  Lingkaran kecil : Penyakit tidak ditemukan
-  Persegi : Penyakit pada tanaman

b. Pohon Keputusan Tanaman Kopi

Pohon keputusan tanaman kopi terdiri dari gejala dan penyakit pada tanaman kopi yang menunjukkan hubungan antara gejala 1 dengan gejala yang lain akan menghasilkan sebuah penyakit sebagai contoh : G001 dengan G002 akan menghasilkan penyakit P01. Berikut adalah pohon keputusan yang terbentuk, dapat dilihat pada gambar 3.2.



Gambar 3.2. Pohon Keputusan Tanaman Kopi

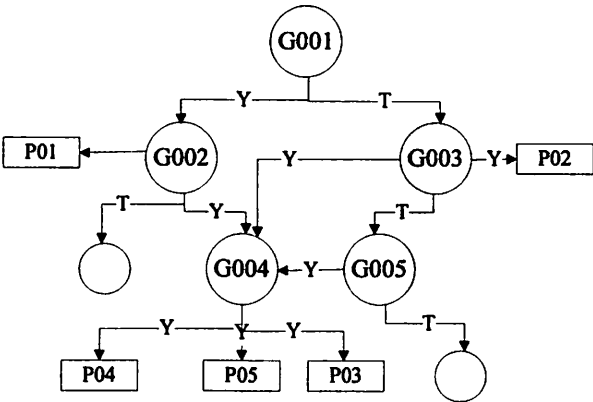
Keterangan :

-  Lingkaran besar : Gejala pada tanaman
-  Lingkaran kecil : Penyakit tidak ditemukan
-  Persegi : Penyakit pada tanaman

c. Pohon Keputusan Tanaman Cengkeh

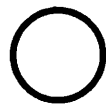
Pohon keputusan tanaman cengkeh terdiri dari gejala dan penyakit pada tanaman cengkeh yang menunjukkan hubungan antara gejala 1 dengan gejala yang lain akan menghasilkan sebuah penyakit sebagai contoh : G001 dengan G002 akan

menghasilkan penyakit P01. Berikut adalah pohon keputusan yang terbentuk, dapat dilihat pada gambar 3.3.



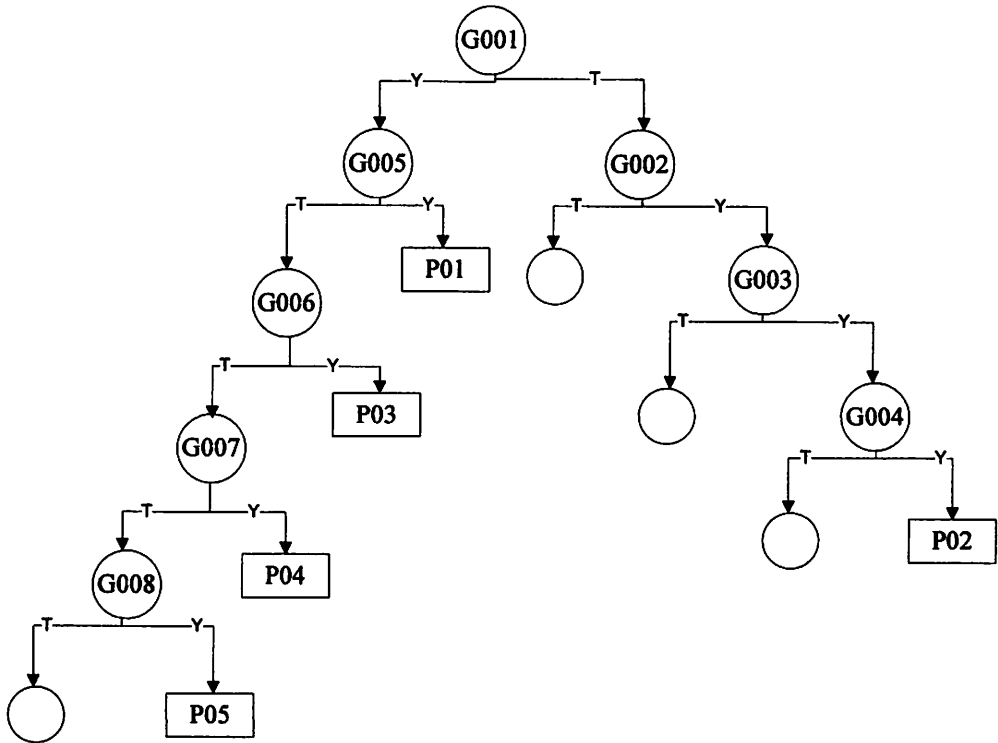
Gambar 3.3. Pohon Keputusan Tanaman Cengkeh

Keterangan :

-  Lingkaran besar : Gejala pada tanaman
-  Lingkaran kecil : Penyakit tidak ditemukan
-  Persegi : Penyakit pada tanaman

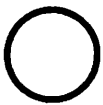
d. Pohon Keputusan Tanaman Karet

Pohon keputusan tanaman karet terdiri dari gejala dan penyakit pada tanaman karet yang menunjukkan hubungan antara gejala 1 dengan gejala yang lain akan menghasilkan sebuah penyakit sebagai contoh : G001 dengan G005 akan menghasilkan penyakit P01. Berikut adalah pohon keputusan yang terbentuk, dapat dilihat pada gambar 3.4.



Gambar 3.4. Pohon Keputusan Tanaman Karet

Keterangan :



Lingkaran besar : Gejala pada tanaman



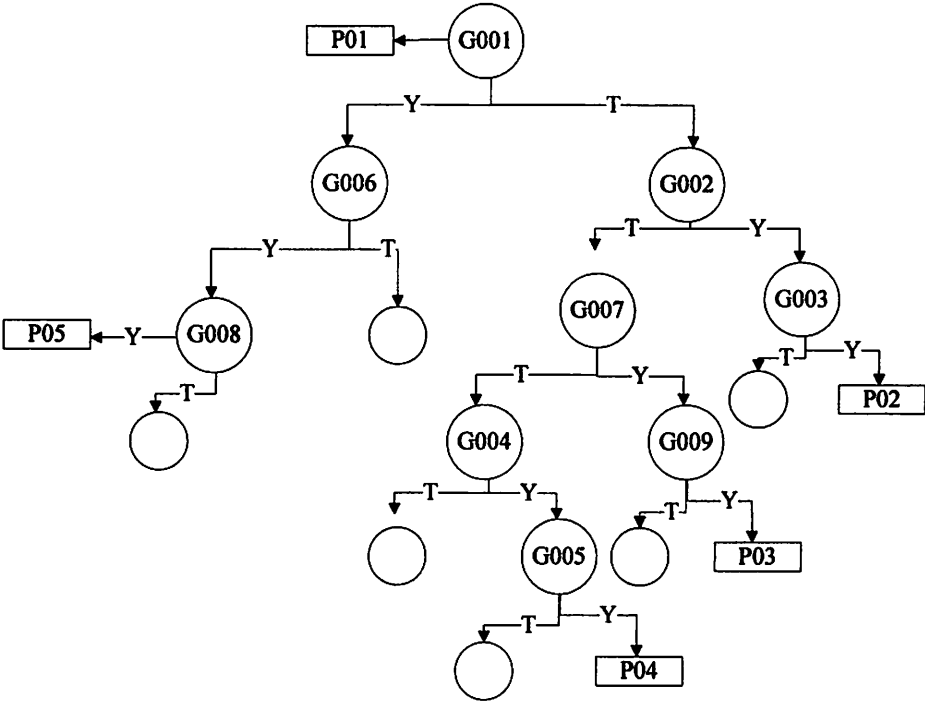
Lingkaran kecil : Penyakit tidak ditemukan



Persegi : Penyakit pada tanaman

e. Pohon Keputusan Tanaman Teh

Pohon keputusan tanaman teh terdiri dari gejala dan penyakit pada tanaman teh yang menunjukkan hubungan antara gejala 1 dengan gejala yang lain akan menghasilkan sebuah penyakit sebagai contoh : G001 dengan G006 dan G008 akan menghasilkan penyakit P05. Berikut adalah pohon keputusan yang terbentuk, dapat dilihat pada gambar 3.5.



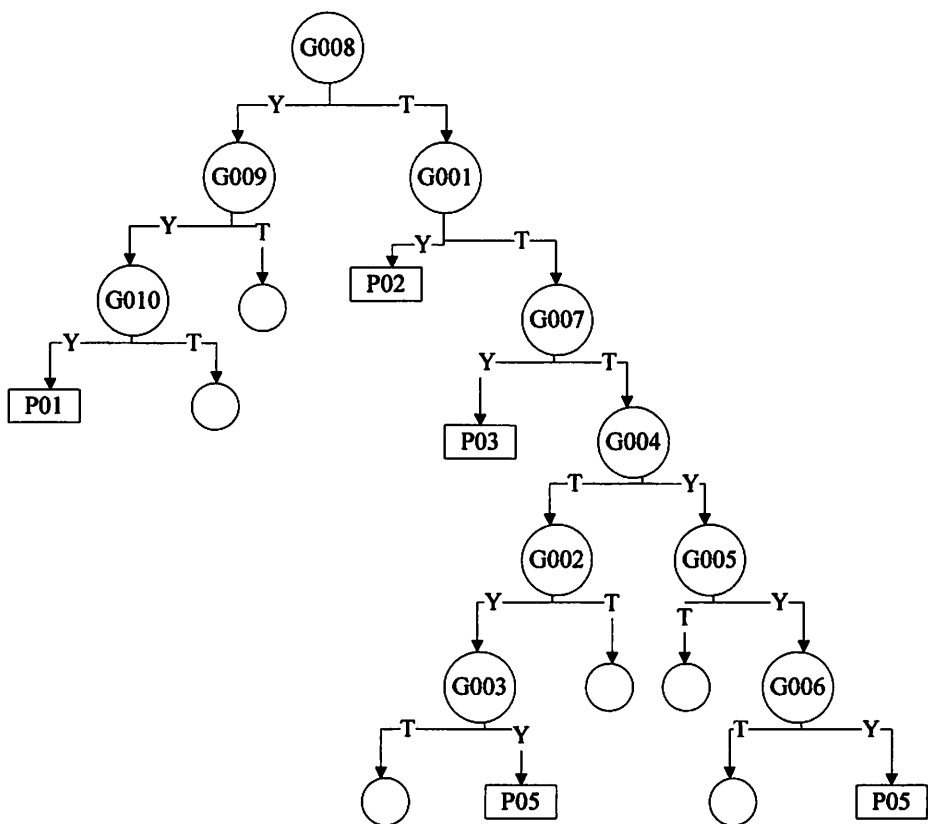
Gambar 3.5. Pohon Keputusan Tanaman Teh

Keterangan :

-  Lingkaran besar : Gejala pada tanaman
-  Lingkaran kecil : Penyakit tidak ditemukan
-  Persegi : Penyakit pada tanaman

f. Pohon Keputusan Tanaman Coklat

Pohon keputusan tanaman coklat terdiri dari gejala dan penyakit pada tanaman coklat yang menunjukkan hubungan antara gejala 1 dengan gejala yang lain akan menghasilkan sebuah penyakit sebagai contoh : G008 dengan G009 dan G010 akan menghasilkan penyakit P01. Berikut adalah pohon keputusan yang terbentuk, dapat dilihat pada gambar 3.6.



Gambar 3.6 Pohon Keputusan Tanaman Coklat

Keterangan :

-  Lingkaran besar : Gejala pada tanaman
-  Lingkaran kecil : Penyakit tidak ditemukan
-  Persegi : Penyakit pada tanaman

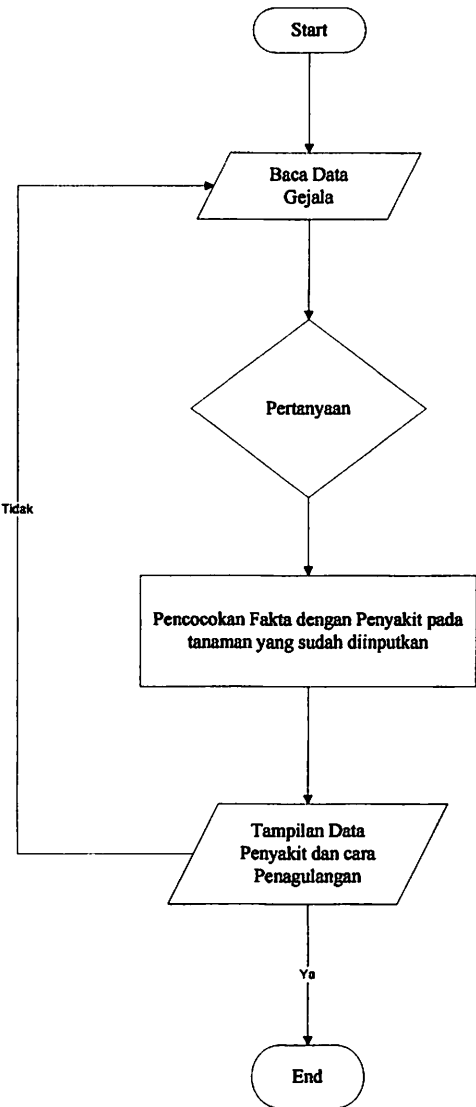
3.2. Perancangan

3.2.1. Mesin Inferensi

Pada perancangan sistem pakar untuk penyakit tanaman perkebunan dilakukan dengan menggunakan mesin inferensi. Mesin infrensi *forward chaining* digunakan dalam sistem. Untuk menentukan penyakit setelah *user* memasukan gejala – gejala yang terjadi pada tanaman perkebunan.

Setiap gejala yang dipilih oleh *user* akan diproses oleh sistem dengan mencari kesesuaian aturan yang telah dibuat dalam *knowledge-based* , sehingga nantinya

akan diperoleh suatu penyakit yang dialami dari gejala – gejala yang telah dipilih. Untuk lebih jelasnya dapat dilihat pada *flowchart* pada gambar 3.7.



Gambar 3.7. *flowchart* Mesin Inferensi

Didalam *flowchart* mesin inferensi dimulai dengan membaca data gejala kemudian *user* memilih gejala, pencocokan gejala – gejala yang sudah diinputkan dengan penyakit pada tanaman yang sudah ditentukan dengan *knowledge-based* kemudian didapatkan penyakit dan cara penaggulangan dan selesai.

3.2.2. Basis Data

Perancangan basis data merupakan proses untuk menentukan isi data yang dibutuhkan mendukung rancangan sistem. Model rancangan basis data yang dibangun adalah model relationship dimana seluruh tabel saling berhubungan satu

dengan yang lainnya. Rancangan basis data yang berisi tabel data yang digunakan adalah tabel tanaman , tabel data gejala , tabel data penyakit , tabel pantogen , tabel rule , tabel user.

1. Tabel Tanaman

Tabel tanaman digunakan untuk menyimpan data tanaman, struktur tabel tanaman dapat dilihat pada tabel 3.9.

Tabel 3.9 Tabel Tanaman

Nama Field	Type	Length	Keterangan
kode_tanaman	Varchar	4	Primary Key
jenis_tanaman	Char	20	

2. Tabel Data Gejala

Tabel data gejala digunakan untuk menyimpan data tentang gejala – gejala pada tanaman perkebunan, struktur tabel data gejala dapat dilihat pada tabel 3.10.

Tabel 3.10 Tabel Data Gejala

Nama Field	Type	Length	Keterangan
kode_gejala	Varchar	4	Primary Key
kode_tanaman	Varchar	4	
gejala	Char	30	

3. Tabel Data Penyakit

Tabel data penyakit digunakan untuk menyimpan data tentang penyakit pada tanaman perkebunan, struktur tabel data penyakit dapat dilihat pada tabel 3.11.

Tabel 3.11 Tabel Data Penyakit

Nama Field	Type	Length	Keterangan
kode_penyakit	Varchar	4	Primary Key
kode_tanaman	Varchar	4	
jenis_penyakit	Char	30	
cara_penanggulangan	Char	50	

4. Tabel Pantogen

Tabel Pantogen digunakan untuk menyimpan data tentang jenis pantogen yang menyerang tanaman perkebunan, struktur tabel pantogen dapat dilihat pada tabel 3.12.

Tabel 3.12 Tabel Pantogen

Nama Field	Type	Length	Keterangan
kode_pantogen	Varchar	4	Primary Key
pantogen	Char	30	

5. Tabel Rule

Tabel Rule digunakan untuk menyimpan data tentang aturan untuk menentukan jenis penyakit pada tanaman , struktur tabel aturan dapat dilihat pada tabel 3.13.

Tabel 3.13. Tabel Rule

Nama Field	Type	Length	Keterangan
kode_rule	Varchar	4	Primary Key
kode_gejala	Varchar	4	
kode_penyakit	Varchar	4	
kode_tanaman	Varchar	4	

6. Tabel User

Tabel User digunakan untuk menyimpan data *username* dan *password*, struktur tabel user dapat dilihat pada tabel 3.14.

Tabel 3.14 Tabel User

Nama Field	Type	Length	Keterangan
id	Varchar	4	Primary Key
username	Char	30	
password	Char	30	

3.2.3 Antar Muka

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang sistem pakar yang akan dibangun. Sehingga akan mempermudah dalam mengimplementasikan sistem pakar. Berikut ini adalah tampilan antarmuka yang dirancang pada pengembangan sistem pakar penyakit tanaman perkebunan berbasis android.

1. Halaman Login Admin

Pada perancangan halaman login hanya terdapat *Input username* dan *password*. Digunakan pakar untuk menambah dan menghapus data – data gejala , penyakit , rule dan macam – macam pantogen. Perancangan halaman login dapat dilihat pada gambar 3.8.

Username :

Password :

LOGIN

Gambar 3.8 Perancangan halaman login

2. Halaman Utama Admin

Pada perancangan halaman utama admin terdapat home , tanaman , input , logout, rule. Menu tanaman terdapat *dropdwon* untuk memilih jenis tanaman yang akan memunculkan gejala dan penyakit pada tanaman perkebunan yang dipilih. Menu input tedapat *dropdown* untuk memilih data gejala atau data penyakit yang akan diinputkan. Perancangan halaman utama admin dapat dilihat pada gambar 3.9.

Wellcome to Panel Admin Sistem Pakar Penyakit Tanaman
Perkebunan

Home Tanaman Input Logout Rule

Gambar 3.9 Perancangan halaman utama

3. Halaman Tanaman

Pada perancangan halaman tanaman terdapat menu gejala dan penyakit untuk menampilkan data dari gejala atau penyakit sesuai dengan jenis tanaman yang dipilih pada menu tanaman. Perancangan halaman tanaman dapat dilihat pada gambar 3.10.

Gejala

Penyakit

Kode Gejala	Gejala	Tanaman

Gambar 3.10 Perancangan halaman tanaman

4. Halaman Edit Gejala

Pada perancangan halaman edit gejala terdapat *field* untuk memasukan kode gejala dan gejalannya. Perancangan halaman edit gejala dapat dilihat pada gambar 3.11.

Kode Tanaman :

Kode Gejala :

Gejala :

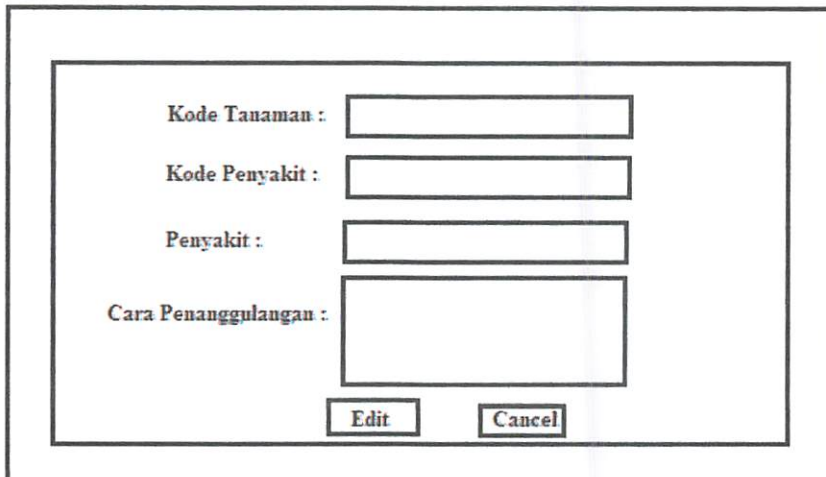
Edit

Cancel

Gambar 3.11 Perancangan halaman edit gejala

5. Halaman Edit Penyakit

Pada perancangan halaman edit penyakit terdapat *field* untuk memasukan kode penyakit, nama penyakit dan cara penanggulangan. Perancangan halaman edit penyakit dapat dilihat pada gambar 3.12.



Kode Tanaman :

Kode Penyakit :

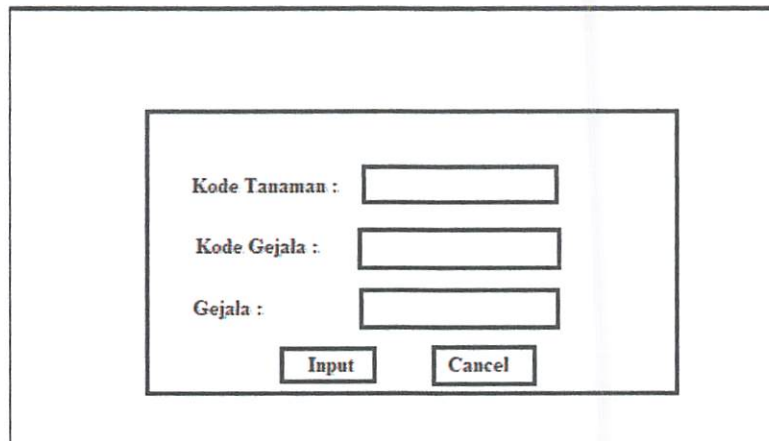
Penyakit :

Cara Penanggulangan :

Gambar 3.12 Perancangan halaman edit penyakit

6. Halaman Tambah Gejala

Pada perancangan halaman tambah penyakit terdapat *field* untuk memasukan jenis tanaman, kode gejala dan gejala. Perancangan halaman tambah gejala dapat dilihat pada gambar 3.13.



Kode Tanaman :

Kode Gejala :

Gejala :

Gambar 3.13 Perancangan halaman tambah gejala

7. Halaman Tambah Penyakit

Pada perancangan halaman tambah penyakit terdapat *field* untuk memasukan jenis tanaman, kode penyakit, nama penyakit dan cara penanggulangan. Perancangan halaman edit penyakit dapat dilihat pada gambar 3.14.

Kode Tanaman :

Kode Penyakit :

Penyakit :

Cara Penanggulangan :

Gambar 3.14 Perancangan halaman tambah penyakit

8. Halaman Edit Rule

Pada perancangan halaman edit rule terdapat *field* untuk memasukan kode penyakit, kode penyakit , dan kode tanaman. Perancangan halaman edit rule dapat dilihat pada gambar 3.15.

Kode Tanaman :

Kode Penyakit :

Kode Gejala :

Gambar 3.15 Perancangan halaman edit rule

9. Halaman Tambah Rule

Pada perancangan halaman tambah rule terdapat *field* untuk memasukan kode penyakit, kode gejala dan kode tanaman. Perancangan halaman tamabh rule dapat dilihat pada gambar 3.16.

Kode Tanaman :

Kode Penyakit

Kode Gejala :

Input

Cancel

Gambar 3.16 Perancangan halaman tambah rule

10. Halaman Utama user

Pada perancangan halaman utama user terdapat 3 menu yaitu diagnosa , macam - macam pantogen , about. Perancangan halaman utama dapat dilihat pada gambar 3.17.

Diagnosa

macam - macam
pantogen

About

Gambar 3.17 Perancangan halaman utama user

11. Halaman Diagnosa

Pada perancangan halaman diagnosa terdapat 6 menu jenis – jenis tanaman yaitu kelapa , kopi , cengkeh , karet, teh , coklat. Perancangan halaman diagnosa dapat dilihat pada gambar 3.18.

The diagram shows a vertical stack of buttons within a rectangular frame. The buttons are labeled from top to bottom: Kelapa, Kopi, Karet, Cengkeh, Teh, Coklat, and Kembali.

Gambar 3.18 Perancangan halaman Diagnosa

12. Halaman Pilih Gejala

Pada perancangan halaman pilih gejala terdapat banyak gejala sesuai dengan jenis tanaman yang telah dipilih oleh user. Perancangan halaman pilih gejala dapat dilihat pada gambar 3.19.

The diagram shows a vertical stack of checkboxes with labels 'gejala 1' through 'gejala 5' to their right. To the right of the list is a vertical scrollbar. At the bottom of the frame are two buttons labeled 'kembali' and 'Selanjutnya'.

Gambar 3.19 Perancangan halaman gejala

13. Halaman Hasil

Pada perancangan halaman hasil terdapat penyakit apa yang dialami oleh tanaman sesuai dengan gejala yang sudah dipilih. Perancangan halaman hasil dapat dilihat pada gambar 3.20.

Jenis penyakit :

Cara penanggulangan :

Kembali

Gambar 3.20 Perancangan halaman hasil

14. Halaman Macam – Macam Pantogen

Pada perancangan halaman macam – macam pantogen terdapat nama pantogen yang menyerang tanaman perkebunan. Perancangan halaman macam – macam pantogen dapat dilihat pada gambar 3.21.

Pantogen 1

Pantogen 2

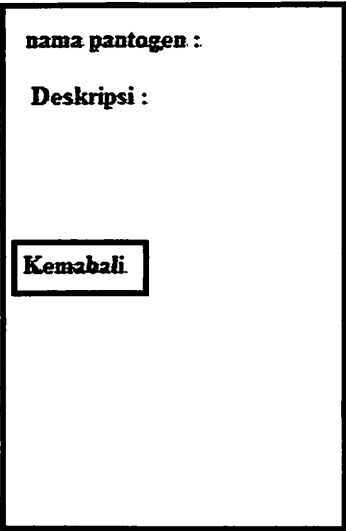
Pantogen 3

Kembali

Gambar 3.21 Perancangan halaman macam – macam pantogen

15. Halaman Detail Pantogen

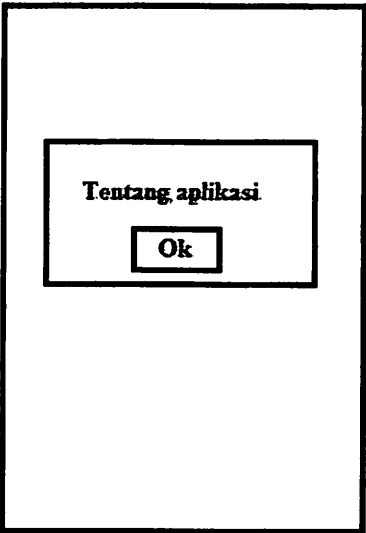
Pada perancangan halaman detail pantogen terdapat detail dari pantogen yang telah dipilih oleh *user* . Perancangan halaman utama dapat dilihat pada gambar 3.22.



Gambar 3.22 Perancangan halaman detail pantogen

16. Halaman About

Perancangan halaman about dapat dilihat pada gambar 3.23



Gambar 3.23 Perancangan halaman about

BAB IV

IMPLEMENTASI DAN PENGUJIAN

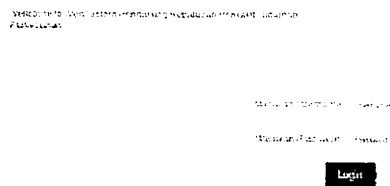
4.1 Implementasi Sistem

Dalam bab ini akan membahas penerapan perancangan sistem pada form – form dalam aplikasi sistem pakar tanaman perkebunan. Aplikasi sistem pakar ini menggunakan basis data MySQL dan bahasa pemrograman Eclips dan pada panel admin menggunakan bahasa pemrograman PHP. Sistem pakar penyakit tanaman perkebunan untuk berkonsultasi gejala – gejala penyakit yang terjadi pada tanaman perkebunan. Pada fitur admin pakar dapat menambahkan, mengubah, menghapus data seperti gejala, jenis tanaman, penyakit, dan rule atau aturan.

Berikut merupakan penjelasan bagian-bagian implementasi sistem yang terbagi menjadi beberapa proses aplikasi, yaitu :

4.1.1 Proses Login pakar

Halaman login untuk pakar hanya terdapat dua entri untuk username dan password. Kemudian ada terdapat 1 tombol. Setiap user yang sudah terdaftar dan memasukan (*username* dan *password*) dengan benar, akan dapat mengakses panel admin.



Gambar 4.1 Halaman Login Pakar

4.1.2 Halaman Utama Pakar

Setelah *user* sukses melakukan *login*, muncul halaman menu utama dengan menu-menu proses perijinan yang ditampilkan sesuai dengan hak akses *user*. Berikut ini tampilan halaman menu terdapat beberapa menu yaitu tanaman, input, rule,logout.

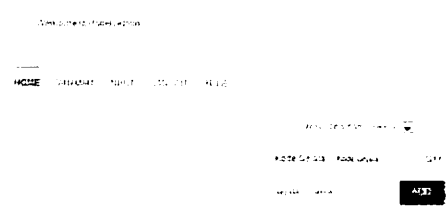


Gambar 4.2 Halaman Utama Pakar

4.1.3 Penginputan Data

4.1.3.1 Form Input Data Gejala

Form input pakar mengisi variabel inputan yaitu memasukan jenis tanaman terlebih dahulu setelah itu memasukan kode gejala setelah itu pakar memasukan gejala setelah tekan tombol input data secara langsung masuk ke database.



Gambar 4.3 Halaman Input Data Gejala

4.1.3.2 Form Input Data Penyakit

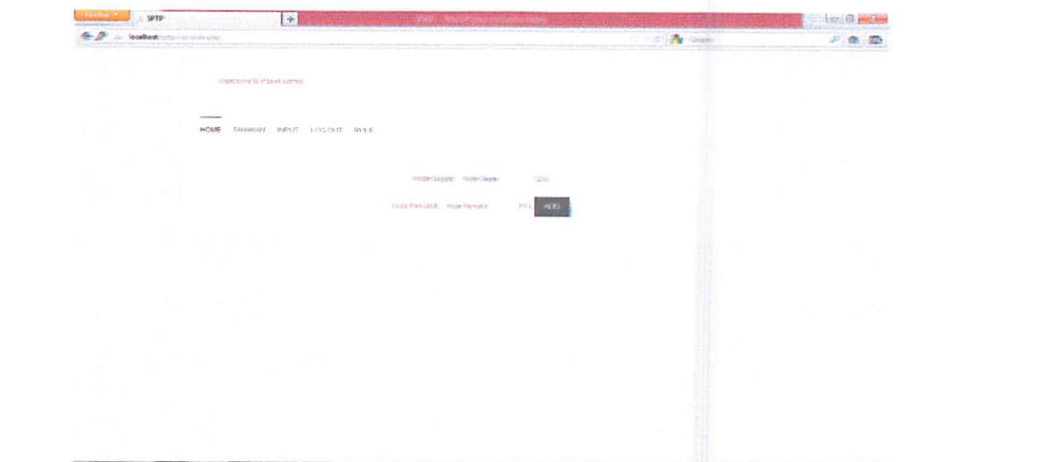
Form input pakar mengisi variabel inputan yaitu memasukan jenis tanaman terlebih dahulu setelah itu memasukan kode penyakit setelah itu pakar memasukan penyakit setelah memasukan penyakit pakar harus memasukan cara penanggulangan setelah tekan tombol input data secara langsung masuk ke database.



Gambar 4.4 Halaman Input Data Penyakit

4.1.3.3 Form Input Relasi

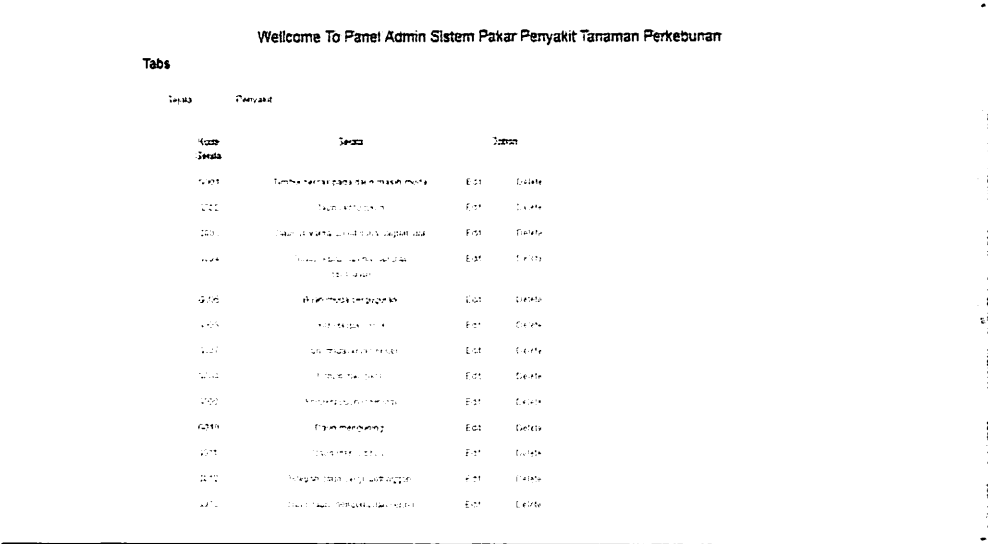
Form input pakar mengisi variabel inputan yaitu memasukan kode penyakit setelah itu pakar memasukan kode_gejala setelah setelah tekan tombol input data secara langsung masuk ke database.



Gambar 4.5 Halaman Input Data Relasi

4.1.4 Halaman Menu Tanaman

Didalam menu utama pakar memilih jenis tanaman setelah pakar memilih tanaman. Halaman tanaman terdapat 2 menu yaitu gejala dan penyakit apabila pakar memilih menu gejala akan muncul isi dari tabel gejala sesuai dengan jenis tanaman yang dipilih oleh pakar.



Gambar 4.6 Halaman Menu Tanaman

4.1.5 Halaman Edit Data gejala

Didalam menu tanaman terdapat menu edit gejala, pakar dapat mengedit gejala pada tanaman sesuai dengan jenis gejala apa yang dipilih oleh pakar.



Gambar 4.7 Halaman Edit Data Gejala

4.1.6 Halaman Edit Data Penyakit

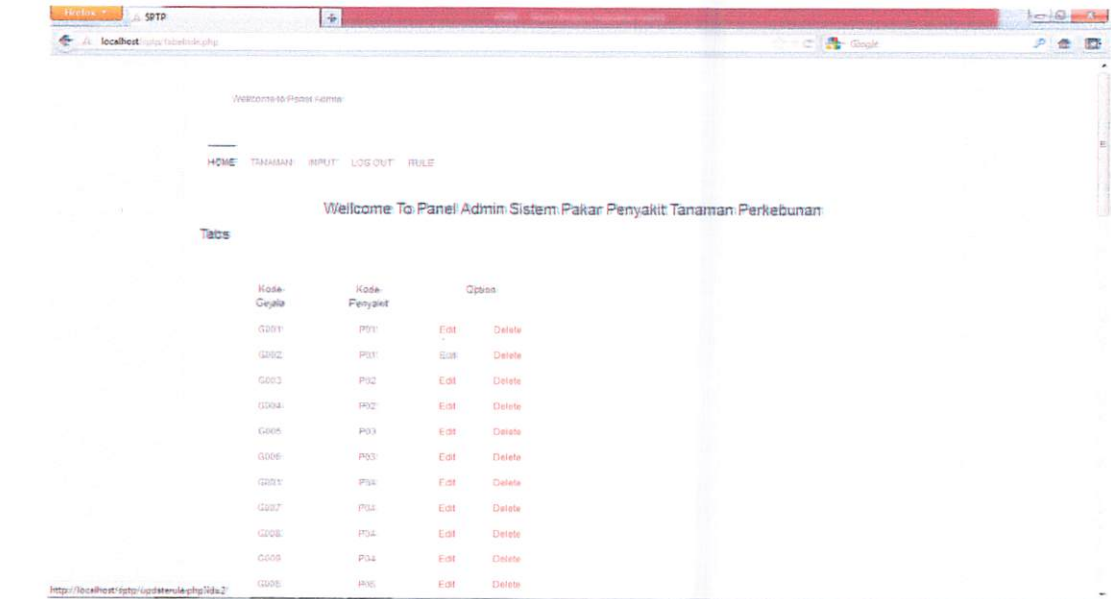
Didalam menu tanaman terdapat menu edit penyakit, pakar dapat mengedit penyakit pada tanaman sesuai dengan jenis penyakit apa yang dipilih oleh pakar.



Gambar 4.8 Halaman Edit Data Penyakit

4.1.7 Halaman Menu Relasi

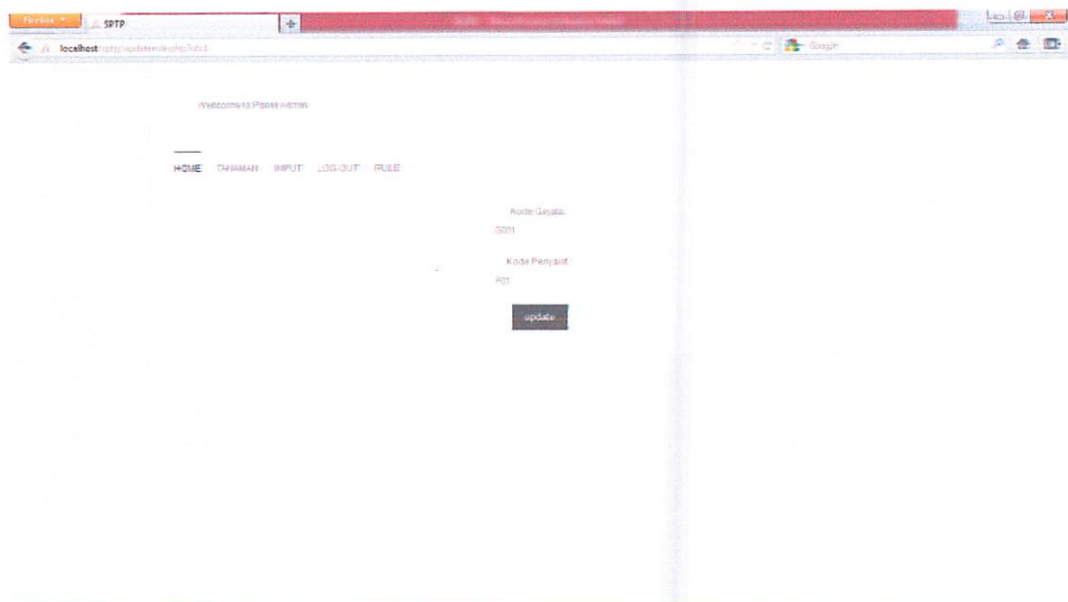
Didalam menu relasi berisi data – data relasi yang terdapat pada tabel relasi.



Gambar 4.9 Halaman Menu Relasi

4.1.8 Halaman Edit Data Relasi

Didalam menu relasi terdapat menu edit relasi, pakar dapat mengedit relasi pada tabel relasi sesuai dengan id relasi apa yang dipilih oleh pakar.



Gambar 4.10 Halaman Edit Data Relasi Penyakit

4.1.9 Halaman Utama User

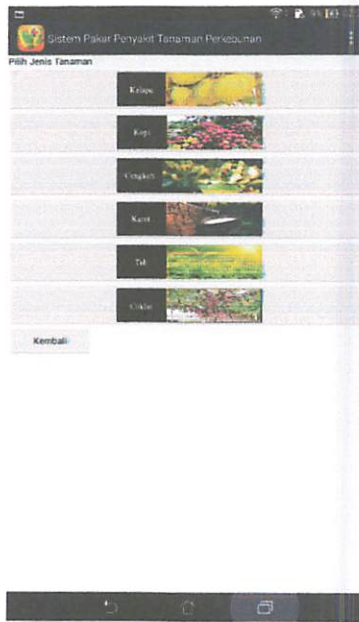
Halaman utama user akan terdapat 2 menu yaitu diagnosa dan macam-macam pantogen.



Gambar 4.11 Halaman Utama User

4.1.10 Halaman Diagnosa

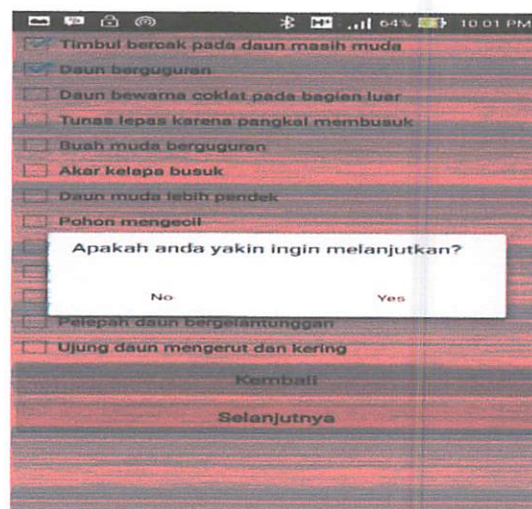
Halaman diagnosa user akan terdapat menu lagi untuk memilih jenis tanaman yaitu kelapa, kopi, cengkeh, karet, teh dan coklat.



Gambar 4.12 Halaman Diagnosa

4.1.11 Halaman Gejala

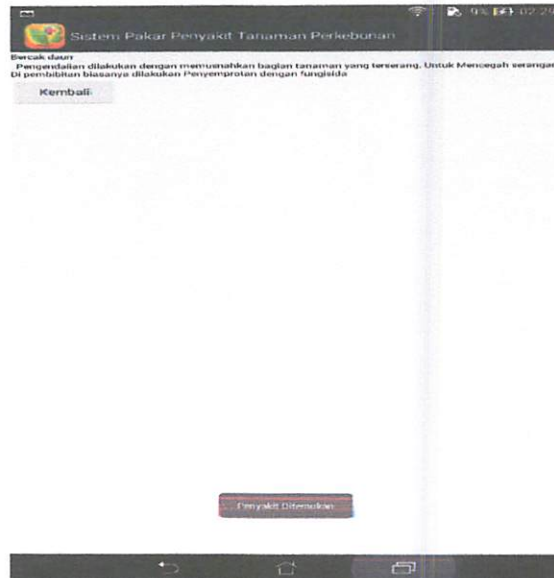
Halaman gejala pada setiap jenis tanaman sama yang membedakan hanya macam – macam gejala. Dan user dapat memilih gejala-gejala yang terjadi pada tanaman setelah memilih gejala user menekan tombol proses.



Gambar 4.13 Halaman Gejala

4.1.12 Halaman Hasil

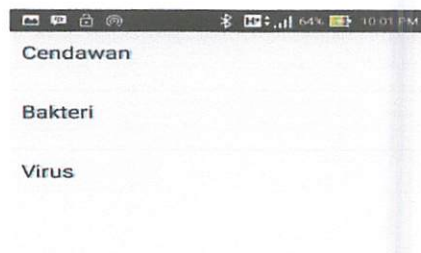
Halaman hasil akan muncul penyakit dan cara penanggulangan sesuai dengan gejala yang telah dimasukan atau setelah dipilih.



Gambar 4.14 Halaman Hasil

4.1.13 Halaman Macam – Macam Pantogen

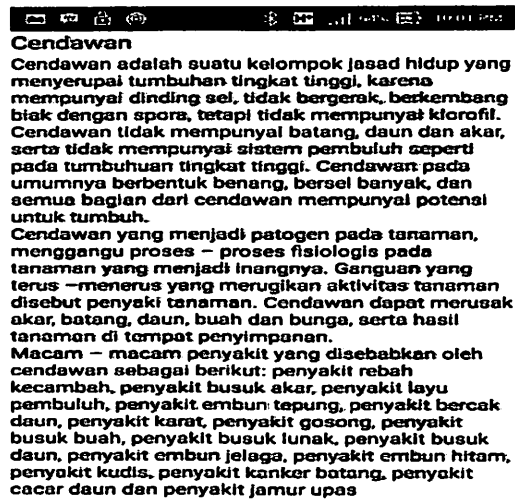
Halaman ini terdapat jenis-jenis pantogen.



Gambar 4.15 Halaman Macam – Macam Pantogen

4.1.14 Halaman Detail Pantogen

Halaman ini detail dari pantogen yang dipilih oleh user. Pada halaman ini user dapat membaca tentang deskripsi pantogen yang dipilih serta user dapat membaca penyakit apa aja yang ditimbulkan oleh pantogen



Gambar 4.16 Halaman Detail Pantogen

4.2 Pengujian Sistem

Pengujian merupakan tahap uji coba dari sistem untuk menguji setiap fungsi vital agar nantinya dapat diambil kesimpulan apakah sistem berjalan dengan baik sesuai tujuan awal pembuatan

4.2.1 Pengujian Fungsional Admin

Pengujian fungsional admin dilakukan untuk menguji fitur-fitur yang ada pada web admin diagnosa penyakit tanaman perkebunan berbasis android. Hasil pengujian fungsional web admin ditunjukkan dalam Tabel 4.1.

Tabel 4.1. Hasil Pengujian Fungsional Web Admin

No	Fungsi yang diuji	Hasil	
		Berjalan	Tidak Berjalan
1	Login web admin	✓	
2	Tambah penyakit	✓	
3	Edit penyakit	✓	
4	Hapus penyakit	✓	

5	Tambah gejala	✓	
6	Edit gejala	✓	
7	Hapus gejala	✓	
8	Tambah relasi	✓	
9	Hapus relasi	✓	

Dari hasil pengujian fungsional web admin menunjukkan bahwa fitur-fitur yang ada pada sistem dapat berjalan 100%.

4.2.2 Pengujian Fungsional Client

Pengujian fungsional client aplikasi dilakukan untuk menguji fitur-fitur yang ada pada aplikasi untuk penyakit tanaman perkebunan berbasis android. Hasil pengujian fungsional client aplikasi ditunjukkan dalam Tabel 4.2.

Tabel 4.2. Hasil Pengujian Fungsional aplikasi

No	Fungsi yang diuji	Hasil	
		Berjalan	Tidak Berjalan
1	Menampilkan <i>Splash Screen</i>	✓	
2	Menampilkan gejala	✓	
3	Proses diagnosa	✓	
4	Menampilkan penyakit	✓	
5	Menampilkan detail penyakit	✓	
6	Menampilkan tentang	✓	

Dari hasil pengujian fungsional aplikasi menunjukkan bahwa fitur-fitur yang ada pada sistem dapat berjalan 100%

Pada tahapan pengujian aplikasi bagian *android* ini dilakukan dengan menggunakan beberapa sistem operasi. Dimana pengujian ini dilakukan untuk mengetahui tingkat keberhasilan dalam bagian *android*. Dan berikut adalah tabel hasil pengujian dari pembuatan aplikasi menggunakan beberapa *system operasi* yang ditunjukkan pada Tabel 4.3 dibawah ini.

Tabel 4.3 Hasil pengujian pada beberapa sistem operasi

Proses	Android 4.2.2 (Jelly Bean)	Android 4.4.2 (KitKat)	Android 5.0.0 (Lolipop)
Menampilkan splash screen	√	√	√
Menampilkan gejala	√	√	√
Proses diagnosa	√	√	√
Menampilkan penyakit	√	√	√
Menampilkan detail penyakit	√	√	√
Menampilkan tentang	√	√	√

Keterangan

√ : sesuai

x : tidak sesuai

Pada hasil pengujian aplikasi dengan menggunakan 3 sistem operasi yang berbeda, semua tampilan dan fungsi aplikasi dapat berjalan 100% pada 3 sistem operasi yaitu OS, Android, Jelly Bean, KitKat, Lolipop.

4.2.3 Pengujian Akurasi

Pengujian akurasi merupakan perbandingan hasil pengujian yang bertujuan untuk membandingkan hasil menurut pakar dengan hasil pengujian sistem pakar penyakit tanaman perkebunan. Dan pada tabel 4.4 adalah hasil pengujian sistem aplikasi untuk penyakit tanaman perkebunan dengan hasil menurut pakar.

Tabel 4.3 Hasil pengujian pada beberapa sistem operasi

Gejala	Hasil aplikasi penyakit tanaman perkebunan	Hasil pakar
G001	P01	P01
G002		
G001	P04	P04
G007		
G008		

G009		
G003	P02	P02
G004		
G005	P05	P05
G009		
G005	P03	P03
G006		

Pada hasil pengujian aplikasi penyakit tanaman perkebunan penyakit yang dihasilkan oleh sistem dari gejala yang diinputkan sama dengan hasil penyakit tanaman perkebunan dari pakar 100% berhasil.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil-hasil pengujian fungsional yang telah dilakukan, aplikasi untuk diagnosa penyakit tenggorokan dan pernafasan memiliki karakteristik-karakteristik sebagai berikut :

1. Hasil pengujian fungsional sistem untuk admin mulai dari halaman *login*, tambah penyakit, edit penyakit, hapus penyakit, tambah gejala, edit gejala, hapus gejala, tambah relasi dan hapus relasi dengan 100% berhasil.
2. Hasil pengujian fungsional sistem untuk *client* mulai dari menampilkan *splash screen*, menampilkan gejala, proses diagnosa, menampilkan penyakit, menampilkan detail penyakit dan menampilkan tentang dengan 100% berhasil.
3. Hasil pengujian pada sistem operasi mulai dari android Jellybean 4.2.2, Android Kitkat 4.4.2, dan android Lolipop 5.0.0 dengan 100% berhasil.
4. Hasil pengujian akurasi dengan gejala yang dimasukan dan hasil penyakit didapat dari sistem penyakit tanaman perkebunan dan hasil penyakit tanaman perkebunan dari pakar 100% berhasil.

5.2 Saran

Saran untuk pengembangan lebih lanjut adalah :

1. Pada pengembangan selanjutnya aplikasi dapat diupload di playstore.
2. Mengembangkan aplikasi sistem pakar agar dapat digunakan pada ios dan windows phone.

DAFTAR PUSTAKA

- Achmad, Balza. 2006. Diktat Mata Kuliah Kecerdasan Buatan. Universitas Gadjah Mada. Yogyakarta.
- Honggiwibowo, Anton Setiawan. 2009. Sistem Pakar Penyakit Tanaman Padi Berbasis Web Dengan Foward dan Backward Chaining (Vol 7 No 2, Desember).
- Kadir, Abdul. 2013. Pemrograman Aplikasi Android From Zero To a Pro. Yogyakarta : Andi.
- Siswanto. 2000. Kecerdasan buatan tiruan jilid 2. Stimik budi luhur. Cileduk.
- Tjahjadi, Ir Nur. 1996. Hama Dan Penyakit Tanaman. Yogyakarta : Kansius.
- Ulysses, John Fredik. 2012. Struktur Sistem Pakar. Stimik.Palangkaraya
- Wicaksono, Arga Dian Setyo. 2014 Sistem Pakar Analisa Penyakit Ikan Lele Berbasis Web Menggunakan Metode Forward Chaining . Stekom. Semarang

LAMPIRAN

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

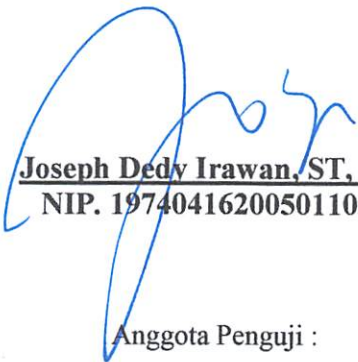
NAMA : Dedy Ega Pratama Cahyo
NIM : 1218268
JURUSAN : Teknik Informatika S-1
JUDUL : PENGEMBANGAN SISTEM PAKAR PENYAKIT TANAMAN
PERKEBUNAN BERBASIS ANDROID

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Sabtu
Tanggal : 16 Januari 2016
Nilai : 78,95 (B+)

Panitia Ujian Skripsi :

Ketua Majelis Penguji


Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

Anggota Penguji :

Dosen Penguji I


Yosep Agus Pranoto, ST. MT
NIP. P. 1031000432

Dosen Penguji II


Moh. Miftakhur Rokhman, S.Kom. M.Kom
NIP.P. 1031500479

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Program Studi Teknik Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Dedy Ega Pratama Cahyo
NIM : 1218268
JURUSAN : Teknik Informatika S-1
JUDUL : PENGEMBANGAN SISTEM PAKAR PENYAKIT TANAMAN
PERKEBUNAN BERBASIS ANDROID

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	16 Januari 2016	1. Pahami Coding Java	
2.	Penguji II	16 Januari 2016	1. Penambahan Header 2. Menu Jenis Tanaman Text Dirapikan 3. Ketika Penyakit Tidak Ditemukan Kembali Ke Menu Sebelumnya 4. Sitasi Tidak Ada 5. Penulisan Daftar Pustaka Salah	

Dosen Penguji I



Yosep Agus Pranoto, ST. MT
NIP. P. 1031000432

Dosen Penguji II



Moh. Miftakhur Rokhman, S.Kom. M.Kom
NIP. P. 1031500479

Dosen Pembimbing I



Karina Auliasari, ST. M.Eng
NIP. P. 1031000426

Dosen Pembimbing II



Febriana Santi W, S.Kom, M.Kom
NIP. P. 1031000425



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

**FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK**

PT. BNI (RSERO) MALANG
BAN JAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 23 Oktober 2015

Nomor : ITN-593/I.INF/TA/2015

Lampiran : ---

Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Karina Auliasari, ST. M.Eng
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,

Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : DEDY EGA PRATAMA CAHYO
Nim : 1218268
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

23 Oktober 2015 S/D 23 Maret 2016

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
Ketua,

Joseph Dedy Irawan, ST., MT.
NIP : 197404162005021002

Form S-4a



BNI (PERS) MALANG
BANK NIASI MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

**FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK**

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 23 Oktober 2015

Nomor : ITN-593/I.INF/TA/2015
Inspirasi : ---
Judul : Bimbingan Skripsi
Kepada : Yth. Bpk/Ibu Febriana Santi W, S.Kom, M.Kom
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : DEDY EGA PRATAMA CAHYO
Nim : 1218268
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

23 Oktober 2015 S/D 23 Maret 2016

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
Ketua,



Joseph Dedy Irawan, ST., MT.
NIP : 197404162005021002

Form S-4a



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1

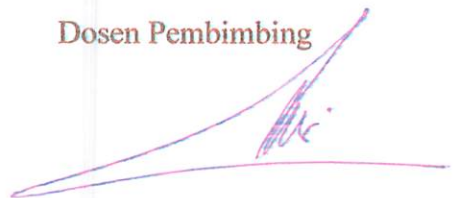
FORMULIR BIMBINGAN SKRIPSI

Nama : Dedy Ega Pratama Cahyo
Nim : 1218268
Masa Bimbingan : 23 Oktober 2015 Sampai 23 Maret 2016
Judul Skripsi : PENGEMBANGAN SISTEM PAKAR PENYAKIT TANAMAN
PERKEBUNAN BERBASIS ANDROID

No	Tanggal	Uraian	Paraf Pembimbing
1	5-11-15	Pembahasan Bab I dan Bab II	
2	12-11-15	Perancangan Knowledge Based System	
3	19-11-15	Coding Rute Based Pada Android	
4	26-11-15	Coding Halaman Administrator	
5	27-12-15	Pembahasan Bab III	
6	3-12-15	Hasil Implementasi	
7	4-12-15	Pengujian Sistem	
8	5-01-16	Revisi Bab IV	
9	8-01-16	Fix Bab IV	
10	11-01-16	Fix Laporan Dan Coding	

Malang, 5-1-2016

Dosen Pembimbing



(Karina Auliasari, ST, M.Eng)

NIP.P. 1031000426



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1

FORMULIR BIMBINGAN SKRIPSI

Nama : Dedy Ega Pratama Cahyo
Nim : 1218268
Mass Bimbingan : 23 Oktober 2015 Sampai 23 Maret 2016
Judu skripsi : PENGEMBANGAN SISTEM PAKAR PENYAKIT TANAMAN
PERKEBUNAN BERBASIS ANDROID

No	Tanggal	Uraian	Paraf Pembimbing
1	18-11-15	Revisi Program (Implementasi Metode)	<i>hni</i>
2	20-11-15	Revisi Bab I (Rumusan Masalah), ACC Program	<i>hni</i>
3	21-11-15	Revisi Bab II (Landasan Teori)	<i>hni</i>
4	23-11-15	ACC Seminar Progres	<i>hni</i>
5	3-12-15	Revisi Bab III	<i>hni</i>
6	10-12-15	Revisi Bab IV	<i>hni</i>
7	11-12-15	Revisi Bab V	<i>hni</i>
8	14-12-15	ACC Seminar Hasil	<i>hni</i>
9	23-12-15	Revisi Bab III (Pohon Keputusan)	<i>hni</i>
10	14-1-16	ACC Kompre	<i>hni</i>

Malang,

Dosen Pembimbing

(Febriana Santi Wahyuni, S.Kom, M.Kom)

NIP.P. 1031000425

AMPIRAN 1. Script Menampilkan Data Gejala Pada User

```
package com.example.sptp;

import java.util.ArrayList;
import java.util.HashMap;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import android.R.integer;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.*;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.RadioGroup.OnCheckedChangeListener;

public class Kopi extends Activity {
    LinearLayout wrapper_checkbox;
    ConnectionDetector cd;
    Boolean isInternetPresent = false;
    AlertDialogManager alert = new AlertDialogManager();
    String[] isi_gejala, idnya;
    int jumlah_gejala;
    ProgressDialog pDialog;
    String status = "1";
    JSONArray college = null;
    CheckBox cbk;
    CheckBox cb_detail;
    String isi_checkbox;
    int tampungarray[];
    LinearLayout.LayoutParams lParams_detail;
    private StringBuilder str;
```



```

    private TextView txt_tampil;
    ArrayList<String> daftar_jawaban;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.layout_kopi);
        daftar_jawaban = new ArrayList<String>();
        jumlah_gejala = 0;
        isi_gejala = new String[200];
        idnya = new String[200];

        lParams_detail = new
        LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,

        LinearLayout.LayoutParams.WRAP_CONTENT);
        lParams_detail.width =
        LinearLayout.LayoutParams.WRAP_CONTENT;
        lParams_detail.height =
        LinearLayout.LayoutParams.WRAP_CONTENT;

        wrapper_checkbox = (LinearLayout)
        findViewById(R.id.wrapper_checkbox);

        cekInternet();

        Button back = (Button) findViewById(R.id.back_kopi);
        back.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                Intent intent = new Intent ();
                setResult(RESULT_OK, intent);
                finish();
            }
        });
    }
    public void cekInternet() {
        cd = new ConnectionDetector(getApplicationContext());
        isInternetPresent = cd.isConnectingToInternet();

        if (isInternetPresent) {

            new AmbilData().execute();

        } else {

```

```

        alert.showAlertDialog(Kopi.this, "Peringatan", "Internet
tidak tersedia, Silakn cek koneksi internet.",
        false);
    }
}

public class AmbilData extends AsyncTask<String, String, String> {

    ArrayList<HashMap<String, String>> dataList = new
ArrayList<HashMap<String, String>>();

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(Kopi.this);
        pDialog.setMessage("Loading Data ...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(true);
        pDialog.show();
    }

    @Override
    protected String doInBackground(String... arg0) {
        // TODO Auto-generated method stub
        String url;
        url = "http://dptp.esy.es/data_gejala_kopi.php";

        JSONParser jParser = new JSONParser();

        JSONObject json = jParser.getJSONFromUrl(url);
        try {
            college = json.getJSONArray("kopi");

            String success = json.getString("success");

            if (success.equals("1")) {

                for (int i = 0; i < college.length(); i++) {
                    JSONObject c =
college.getJSONObject(i);

                    HashMap<String, String> map =
new HashMap<String, String>();

                    String id =
c.getString("kode_gejala").trim();

                    String keterangan =
c.getString("gejala");

```

```

        map.put("kode_gejala", id);
        map.put("gejala", keterangan);
        jumlah_gejala++;
        idnya[i] = id;
        isi_gejala[i] = keterangan;
        dataList.add(map);
    }
} else {

    pDialog.dismiss();
    status = "0";

}

} catch (JSONException e) {

    pDialog.dismiss();

}

return null;
}

@Override
protected void onPostExecute(String result) {
    // TODO Auto-generated method stub
    super.onPostExecute(result);
    pDialog.dismiss();
    if (status.equals("0")) {
        Toast.makeText(getApplicationContext(), "data
tidak ada", Toast.LENGTH_SHORT).show();

    }

    for (int x = 0; x < jumlah_gejala; x++) {
        tampilkan_checkbox(isi_gejala[x], idnya[x]);
    }

    ArrayList<Integer> isicheckbok = new
rrayList<Integer>();
    for (int x = 0; x < jumlah_gejala; x++) {
        isicheckbok.add(jumlah_gejala);
    }

}

}

```

```

public void tampilkan_checkbox(String isi_checkbox, String id) {
    try {
        final CheckBox cbk_detail = new CheckBox(this);
        cbk_detail.setText(isi_checkbox);
        cbk_detail.setTextColor(Color.BLACK);
        cbk_detail.setChecked(false);
        cbk_detail.setEnabled(true);
        cbk_detail.setTag(id);

        cbk_detail.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void
onCheckedChanged(CompoundButton compoundButton, boolean b) {
                if (b) {

                    daftar_jawaban.add(cbk_detail.getTag().toString());
                } else {

                    daftar_jawaban.remove(cbk_detail.getTag().toString());
                }
            }
        });
        wrapper_checkbox.addView(cbk_detail, lParams_detail);

    } catch (Exception e) {
        alert.showAlertDialog(Kopi.this, "Peringatan",
        .toString(), false);
    }
    Button lanjut = (Button) findViewById(R.id.next_kopi);
    lanjut.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            box_konfirmasi(v);
        }
    });
}

public void alert(String pesan) {
    new AlertDialog.Builder(this).setTitle("ID pertanyaan sing
bok centang")
        .setMessage("Adalah : " +
pesan.toString()).setPositiveButton("Yes", null).setNegativeButton("No", null)
        .show();
}

```



```

    }

    public void box_konfirmasi(final View v) {
        new AlertDialog.Builder(this).setTitle("").setMessage("Apakah
        da yakin ingin melanjutkan?")
        .setPositiveButton("Yes", new
        DialogInterface.OnClickListener() {
            public void onClick(DialogInterface
            dialog, int which) {

                Bundle b = new Bundle();

                b.putStringArrayList("Data_Jawaban", daftar_jawaban);
                Intent intent = new
                Intent(v.getContext(), Hasil_gejala_kopi.class);
                intent.putExtras(b);
                startActivity(intent);
            }
        }).setNegativeButton("No", null).show();

    }

    public void pindah_halaman() {
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

```


IMPLEMENTASI 2. Script Menampilkan Hasil Penyakit Pada User

```
package com.example.sptp;

import java.util.ArrayList;
import java.util.HashMap;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.graphics.Color;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Window;
import android.widget.CheckBox;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

public class Hasil_gejala_kopi extends Activity {
    ArrayList<String> daftar_jawaban;
    TextView hasil;
    ProgressDialog progressDialog;
    String request_php;
    JSONArray college = null;
    int jumlah_gejala = 0;
    String[] isi_gejala, idnya, gulang;
    String status = "1";
    LinearLayout wrapper_textview;
    LinearLayout.LayoutParams lParams_detail;

    LinearLayout wrapper_checkbox;
    ConnectionDetector cd;
    Boolean isInternetPresent = false;
    AlertDialogManager alert = new AlertDialogManager();
    CheckBox cbk;
    CheckBox cb_detail;
    String isi_checkbox;
    int tampungarray[];
    private StringBuilder str;
    private TextView txt_tampil;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

        this.requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.hasil_gejala_kopi);
        isi_gejala = new String[200];
        idnya = new String[200];
        jumlah_gejala = 0;
        isi_gejala = new String[200];
        gulang = new String[200];
        idnya = new String[200];

        daftar_jawaban =
        Intent().getStringArrayListExtra("Data_Jawaban");
        request_php = "?kode_tanaman=t02&";
        for (int i = 0; i < daftar_jawaban.size(); i++) {
            request_php = request_php + "kode_gejala[" + i + "]=" +
            daftar_jawaban.get(i) + "&";
        }
        lParams_detail = new
        LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,

        LinearLayout.LayoutParams.WRAP_CONTENT);
        lParams_detail.width =
        LinearLayout.LayoutParams.WRAP_CONTENT;
        lParams_detail.height =
        LinearLayout.LayoutParams.WRAP_CONTENT;

        wrapper_textview = (LinearLayout)
        findViewById(R.id.wrapper_textview);
        cekInternet();

    }

    public void cekInternet() {
        cd = new ConnectionDetector(getApplicationContext());
        isInternetPresent = cd.isConnectingToInternet();

        if (isInternetPresent) {

            new AmbilData().execute();

        } else {
            alert.showAlertDialog(Hasil_gejala_kopi.this,
            "Peringatan", "Internetmu gak konak", false);
        }
    }

    public class AmbilData extends AsyncTask<String, String, String> {

```

```

        ArrayList<HashMap<String, String>> dataList = new
ArrayList<HashMap<String, String>>();

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            pDialog = new ProgressDialog(Hasil_gejala_kopi.this);
            pDialog.setMessage("Loading Data ...");
            pDialog.setIndeterminate(false);
            pDialog.setCancelable(true);
            pDialog.show();

        }

        @Override
        protected String doInBackground(String... arg0) {
            // TODO Auto-generated method stub
            String url;
            url = "http://dptp.esy.es/una.php" + request_php;

            JSONParser jParser = new JSONParser();

            JSONObject json = jParser.getJSONFromUrl(url);
            try {
                college = json.getJSONArray("kelapa");

                String success = json.getString("success");

                if (success.equals("1")) {

                    for (int i = 0; i < college.length(); i++) {
                        JSONObject c =
college.getJSONObject(i);

                        HashMap<String, String> map =

                        String id =

                        String keterangan =

                        String gulangan =

                        map.put("kode_gejala", id);
                        map.put("gejala", keterangan);
                        map.put("gulang", gulangan);

                        jumlah_gejala++;

```

```

        idnya[i] = id;
        isi_gejala[i] = keterangan;
        gulang[i] = gulangan;
        dataList.add(map);
    } else {

        pDialog.dismiss();
        status = "0";

    }

} catch (JSONException e) {
    pDialog.dismiss();
}

return null;
}

@Override
protected void onPostExecute(String result) {

    // TODO Auto-generated method stub
    super.onPostExecute(result);
    pDialog.dismiss();
    if (status.equals("0")) {
        Toast.makeText(getApplicationContext(), "data
tidak ada", Toast.LENGTH_SHORT).show();

    }

    for (int x = 0; x < jumlah_gejala; x++) {
        tampilkan_textview(isi_gejala[x],
        id[x],gulang[x]);
    }

    ArrayList<Integer> isicheckbok = new
    ArrayList<Integer>();
    for (int x = 0; x < jumlah_gejala; x++) {
        isicheckbok.add(jumlah_gejala);
    }

}

}

```



```

    public void tampilkan_textview(String isi_checkbox, String id,String
    gulang) {
        try {
            final TextView text_view_gejala = new TextView(this);
            text_view_gejala.setText(isi_checkbox);
            text_view_gejala.setTextColor(Color.BLACK);
            text_view_gejala.setTag(id);

            wrapper_textview.addView(text_view_gejala,
            0, 0);

            final TextView text_view_cara_penanggulangan = new
            TextView(this);
            text_view_cara_penanggulangan.setText(gulang);

            text_view_cara_penanggulangan.setTextColor(Color.BLACK);
            text_view_cara_penanggulangan.setTag(id);

            wrapper_textview.addView(text_view_cara_penanggulangan,
            1, 0);

        } catch (Exception e) {
            alert.showAlertDialog(Hasil_gejala_kopi.this,
            "Penanganan", e.getMessage(), false);
        }
    }

    public void alert(String pesan) {
        new AlertDialog.Builder(this).setTitle("ID pertanyaan sing
        mcentang")
            .setMessage("Adalah : " +
            pesan.toString()).setPositiveButton("Yes", null).setNegativeButton("No", null)
            .show();
    }
}

```

LAMPIRAN 3. Script Proses Cari Penyakit

```
hp
set($_GET['kode_gejala'])){
    $mysqli = new mysqli("mysql.idhostinger.com",
62981893_dptp", "aremania1108", "u762981893_dptp");

    $kode_tanaman=$_GET['kode_tanaman'];
    $array_gejala=$_GET['kode_gejala'];
    $dimana="";
    $jumlah_gejala=count($array_gejala);
    $array_penakit=array();

    function cari_penakit($gejala_masukan,$penyakit_masukan=0){
        $kode_tanaman=$_GET['kode_tanaman'];
        $mysqli = new mysqli("mysql.idhostinger.com",
"u762981893_dptp", "aremania1108", "u762981893_dptp");
        $keluaran_array=array();
        if(count($penyakit_masukan)>0){
            $query="SELECT * FROM tb_relasi WHERE
kode_tanaman = '$kode_tanaman' AND kode_gejala='$gejala_masukan'
".decode_kode_penakit($penyakit_masukan);
        }else{
            $query="SELECT * FROM tb_relasi WHERE
kode_tanaman = '$kode_tanaman' AND kode_gejala='$gejala_masukan'";
        }
        //echo($query."<br />");
        $hasil=$mysqli->query($query);
        if($hasil->num_rows > 0) {
            while ($data=$hasil->fetch_array(MYSQLI_BOTH)){

                array_push($keluaran_array,$data['kode_penakit']);
            }
        }
    }
}
```

```

    }
    return $keluaran_array;
}

function decode_kode_penyakit($array_masukan_penyakit){
    $dimana = "";

    for($stampan=0;$stampan<count($array_masukan_penyakit);$stampan++){
        if($stampan==0){
            $dimana .= " AND ( kode_penyakit =
'Sarray_masukan_penyakit[$stampan]' ";
        }else{
            $dimana .= " OR kode_penyakit=
'Sarray_masukan_penyakit[$stampan]' ";
        }
    }
    if(count($array_masukan_penyakit)>0){
        $dimana .= ")";
    }
    return $dimana;
}

for($sasik=0; $sasik<$jumlah_gejala; $sasik++){
    if($sasik==0){

        $sarray_penyakit=cari_penyakit($sarray_gejala[$sasik],null);
    }else{
        if(count($sarray_penyakit)==0){
            break;
        }

        $sarray_penyakit=cari_penyakit($sarray_gejala[$sasik],$sarray_penyakit);
    }
}

```

```

    }

}

$dimana_akhir="";
if ($jumlah_gejala>1 && $jumlah_gejala<8)
{
for($xyz=0; $xyz<count($array_penyakit); $xyz++){
    if($xyz==0){

        $dimana_akhir .= "AND (kode_penyakit =
'Sarray_penyakit[$xyz]' ";
    }else{
        $dimana_akhir .= " OR kode_penyakit=
'Sarray_penyakit[$xyz]' ";
    }
}
}
if ($xyz>0){
    $dimana_akhir .= ")";
}

$query_akhir = "SELECT * FROM `tb_penyakit` WHERE
kode_tanaman = '$kode_tanaman' ".$dimana_akhir;
//echo ($query_akhir);

$mysqli = new mysqli("mysql.idhostinger.com",
"u762981893_dptp", "aremania1108", "u762981893_dptp");
$hasil = $mysqli->query($query_akhir);
if($hasil->num_rows > 0 && $dimana_akhir<>"") {
    $response =array();
    $response["kelapa"] =array();
    while ($kelapa=$hasil->fetch_array(MYSQLI_BOTH)){
        $h['kode_gejala']=$kelapa['kode_penyakit'];
        $h['gejala']=$kelapa['penyakit'];
    }
}

```



```
        $h['kode_tanaman']=$kelapa['kode_tanaman'];

        $h['cara_penanggulangan']=$kelapa['cara_penanggulangan'];
        array_push($response["kelapa"], $h);
    }

    $response["success"]="1";
    echo json_encode($response);
}
else{
    $response["success"]=0;
    $response["message"]="tidak ada data";
    echo json_encode($response);
}
}
?>
```