

**MEMBANGUN SISTEM PAKAR
UNTUK MENGIDENTIFIKASI JENIS PENYAKIT PADA TANAMAN JERUK
BERBASIS WIRELESS APPLICATION PROTOCOL (WAP)
DENGAN WIRELESS MARKUP LANGUAGE (WML)
DAN PHP HYPERTEXT PREPROCESSOR (PHP)**

Titania Dwi Andini, S.Kom
Jaenal Arifin

Abstraksi

Untuk meningkatkan mutu dan mencegah penurunan hasil produksi jeruk yang disebabkan oleh penyakit jeruk, maka petani memerlukan informasi yang akurat mengenai penyakit jeruk serta cara pengendalian yang tepat. Teknologi kecerdasan buatan telah membuka wacana baru dalam dunia teknologi komputer. Dengan sistem pakar yang merupakan bagian dari teknologi kecerdasan buatan telah mampu memberi solusi dalam mendapatkan informasi yang diperlukan petani tentang jenis penyakit jeruk dan cara pengendaliannya. Satu lagi teknologi yang saat ini telah berkembang pesat dan memasyarakat adalah perangkat nirkabel yang diantaranya adalah telepon seluler (ponsel). Dengan teknologi *wireless application protocol* (WAP), informasi-informasi penting dan aplikasi-aplikasi berorientasi internet dapat disajikan ke dalam ponsel.

Perkembangan teknologi yang terjadi saat sekarang telah memungkinkan sistem pakar untuk diaplikasikan penggunaannya pada ponsel. Salah satu-nya dapat dimanfaatkan dalam pemberian informasi mengenai penyakit pada tanaman jeruk, penyebab serta cara pengendaliannya. Sistem pakar menggunakan teknik pencarian *heuristik* dan Metode inferensi *forward* dan *backward chaining*. Pengguna dari sistem pakar dibagi menjadi dua kategori yaitu umum dan administrator. Fasilitas yang diberikan untuk pengguna umum dan administrator dibedakan sesuai dengan kebutuhannya masing-masing. Pengguna umum diberi kemudahan dalam mendapatkan informasi berbagai jenis penyakit pada tanaman jeruk beserta gejala-gejala yang ditimbulkan, penyebab penyakit dan cara pengendaliannya serta dapat melakukan konsultasi layaknya dengan seorang pakar tanaman jeruk melalui tanya jawab antara pengguna dengan sistem guna mengidentifikasi jenis penyakit. Sedangkan administrator dimudahkan dalam mengelola sistem, baik proses tambah, hapus maupun update data.

Kata-kata kunci : Penyakit jeruk, Sistem pakar, *Heuristik*, *Forward* dan *Backward chaining*, WAP.

1. Pendahuluan

Kemajuan teknologi komputer yang pesat dapat membantu kehidupan manusia bahkan di dalam bidang di luar disiplin ilmu komputer. Sistem pakar merupakan salah satu cabang kecerdasan buatan (*artificial intelligence*) yang menggabungkan pengetahuan, pengalaman dan penelusuran data dari satu atau banyak pakar ke dalam bentuk sistem sehingga dapat digunakan untuk memecahkan berbagai masalah yang bersifat spesifik, dalam hal ini adalah permasalahan tentang mengidentifikasi penyakit pada tanaman jeruk.

Seiring dengan kemajuan teknologi yang semakin pesat, berpengaruh pula terhadap perkembangan perangkat komunikasi telepon seluler (ponsel), sehingga penggunaan perangkat ini semakin luas dan memasyarakat. Bahkan perkembangan saat ini, fungsi dari ponsel bukan hanya sebagai perangkat komunikasi saja melainkan sudah banyak fasilitas lain yang ada di dalamnya, diantaranya adalah untuk mengakses internet. Dengan pengaksesan internet melalui perangkat ini,

tentu sangat membantu dalam penyajian informasi yang cepat dan mudah serta bisa diakses dimanapun pengguna berada.

2. Tujuan Penelitian

Membangun sistem pakar yang berfungsi untuk mengidentifikasi penyakit pada tanaman jeruk berbasis WAP (*Wireless Application Protocol*), yang dapat diakses dengan ponsel yang sudah dilengkapi dengan *WAP browser*, sehingga petani jeruk dapat memanfaatkan sistem pakar ini untuk melakukan antisipasi atau menanggulangi penyakit pada tanamannya secara tepat dan lebih awal.

3. Rumusan Masalah

Beberapa permasalahan yang timbul adalah :

1. Bagaimana membangun sistem pakar yang dapat membantu petani untuk mengidentifikasi jenis penyakit pada tanaman jeruk?
2. Bagaimana menerapkan fasilitas penjelasan pada sistem pakar yang berhubungan dengan penyakit pada tanaman jeruk?
3. Bagaimana menyajikan sistem pakar secara untuk ditampilkan ke dalam telepon seluler (ponsel) dengan pemrograman WAP?

4. Metodologi Penelitian

Metodologi yang dipakai dalam penelitian ini adalah:

1. Studi Kepustakaan : melakukan studi literatur atau tinjauan pustaka tentang konsep dan teori dasar sistem pakar serta pengembangan program *Wireless Application Protocol* (WAP) dengan bahasa pemrograman *Wireless Markup Language* (WML) dan *Php Hypertext Preprocessor* (PHP).
2. Pengumpulan data : melakukan proses pengumpulan data-data mengenai gejala dan jenis-jenis penyakit pada tanaman jeruk dari laboratorium data dan perpustakaan Balai Penelitian Tanaman Jeruk dan Buah Subtropika (Balitjestro) serta dari beberapa situs di internet.
3. Penyusunan basis pengetahuan : data utama dan data penunjang yang didapatkan berupa fakta dari pakar tanaman jeruk. Aturan dan kesimpulan yang mengatur proses pencarian data yang saling berhubungan antara tabel satu dengan yang lain relasinya dirancang dengan pemodelan *entity relationship diagram*. Kemudian fakta dan aturan tersebut disimpan ke dalam basis data MySQL sebagai media penyimpanan.

5. Tinjauan Pustaka

a. Artificial Intelligence

1. Kecerdasan Buatan

Kecerdasan Buatan atau *Artificial Intelligence* adalah pemikiran atau ide untuk membuat suatu perangkat lunak komputer agar memiliki kecerdasan layaknya manusia, sehingga perangkat lunak tersebut dapat melakukan suatu pekerjaan yang biasanya dilakukan oleh manusia. Diantaranya pekerjaan itu adalah berupa konsultasi yang dapat memberikan suatu informasi berupa saran-saran yang akan sangat berguna.

Kecerdasan Buatan memungkinkan komputer untuk bisa berpikir dan mampu mengolah pengetahuan tertentu. Dengan cara ini, Kecerdasan Buatan dapat menirukan proses belajar dan berpikir seperti cara yang dilakukan oleh manusia, sehingga informasi baru dapat diserap dan digunakan sebagai acuan di masa-masa mendatang.

Kecerdasan atau kepandaian itu didapat berdasarkan pengetahuan dan pengalaman. Untuk itu agar perangkat lunak bisa mempunyai kecerdasan, maka perangkat lunak tersebut harus diberi pengetahuan dan kemampuan untuk menalar. Dari pengetahuan dan kemampuan yang telah didapat akan digunakan dalam menemukan solusi atau kesimpulan.

Beberapa bidang atau implementasi kecerdasan buatan antara lain: Sistem Pendukung Keputusan (*Decision Support System*), Sistem Pakar (*Expert System*), Robotika (*Robotics*), Pengolahan bahasa alami (*Natural Language processing*), Jaringan Saraf Tiruan (*Neural Network*), dan lain-lain.

2. Sistem Pakar

Istilah sistem pakar (*Expert System*) berasal dari istilah sistem pakar berbasis pengetahuan. Sistem pakar adalah sistem yang menggunakan pengetahuan manusia yang terekam dalam komputer untuk memecahkan persoalan yang biasanya memerlukan keahlian manusia (Turban, 2005:708)

Menurut Martin dan Oxman (1988) sistem pakar adalah sistem yang berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut (Kusrini, 2006:11)

Edward Feigenbaum (1982) mendefinisikan sistem pakar sebagai suatu program komputer cerdas yang menggunakan *knowledge* (pengetahuan) dan prosedur inferensi untuk menyelesaikan masalah yang cukup rumit yang membutuhkan seorang ahli untuk menyelesaikannya (Arhami, 2005:2).

Dari definisi-definisi diatas dapat disimpulkan bahwa sistem pakar adalah sebuah perangkat lunak komputer yang memiliki basis pengetahuan untuk domain tertentu dan menggunakan penalaran menyerupai seorang pakar dalam memecahkan masalah. Sistem pakar adalah salah satu jalan untuk mendapatkan pemecahan masalah secara lebih cepat dan mudah.

Dengan sistem pakar, seseorang yang awam pun dapat menyelesaikan masalah yang cukup rumit atau bisa juga hanya sekedar mencari informasi berkualitas yang sebenarnya hanya bisa diperoleh dengan bantuan para ahli. Sistem pakar juga dapat membantu aktifitas pakar, yang difungsikan sebagai asisten yang berpengalaman dan mempunyai pengetahuan yang dibutuhkan.

Sistem pakar merupakan sebuah terobosan dalam mengambil dan memadukan pengetahuan dengan teknologi. Dalam penyusunannya, sistem pakar mengkombinasikan aturan-aturan penarikan kesimpulan (*inference rules*) dengan basis pengetahuan tertentu yang berasal dari pengalaman maupun penelitian satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua hal tersebut kemudian diolah dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk menyelesaikan masalah tertentu.

3. Ciri-Ciri Sistem Pakar

Ciri-ciri sistem pakar yang baik adalah sebagai berikut :

1. Memiliki informasi yang handal.
2. Mudah dimodifikasi yaitu dengan menambah atau menghapus data.
3. Dapat digunakan dalam berbagai jenis komputer.
4. Memiliki kemampuan untuk belajar beradaptasi.

4. Keuntungan sistem pakar

Beberapa keuntungan sistem pakar adalah sebagai berikut:

1. Pekerjaan menjadi lebih sederhana dan waktu kerja menjadi lebih hemat (efisiensi waktu).
2. Mampu mengambil, menyimpan dan melestarikan pengetahuan dan keahlian para pakar (terutama yang termasuk keahlian langka).
3. Bisa melakukan proses secara berulang-ulang dan otomatis dengan hasil yang konsisten.
4. Menjadikan seorang yang masih awam, dapat bekerja layaknya seorang pakar.

5. Data-data yang terpercaya dari sebuah keahlian tertentu, sehingga pengguna sistem pakar seolah-olah berkonsultasi atau berkomunikasi langsung dengan pakar, walaupun mungkin pakar tersebut sudah meninggal.
6. Dapat meningkatkan output dan produktivitas dengan meningkatnya kualitas hasil pekerjaan/produksi, meningkatnya hal itu dikarenakan meningkatnya efisiensi kerja (menghemat waktu dalam pengambilan keputusan).
7. Memperluas jangkauan, dari keahlian seorang pakar. Dimana sebuah sistem pakar yang telah disahkan, akan sama juga artinya dengan menyediakan pakar dalam jumlah besar (dapat diperbanyak dengan kemampuan yang sama persis), dapat diperoleh dan dimanfaatkan sewaktu-waktu dan dimana saja pengguna berada.

5. Kelemahan Sistem Pakar

Di samping memiliki beberapa keuntungan, menurut Giarrantano, J. dan G.Riley (1989), sistem pakar memiliki beberapa kelemahan, yaitu :

1. Biaya yang diperlukan untuk membuat dan memelihara sangat mahal.
2. Sulit dikembangkan. Hal ini berkaitan erat dengan ketersediaan pakar/ ahli di bidangnya.
3. Sistem Pakar tidak 100% bernilai benar.

6. Alasan Pengembangan Sistem Pakar

Sistem pakar dikembangkan lebih lanjut dengan alasan :

1. Dapat menyediakan kepakaran setiap waktu dan di berbagai lokasi.
2. Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
3. Seorang pakar akan pensiun atau pergi.
4. Seorang pakar adalah mahal.
5. Kepakaran dibutuhkan juga pada lingkungan yang tidak bersahabat.

7. Struktur Sistem Pakar

Menurut Turban, sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan dan lingkungan konsultasi (Turban, 2005:721). Lingkungan pengembangan sistem pakar digunakan untuk membangun komponen dan memasukkan pengetahuan pakar ke dalam basis pengetahuan, sedangkan lingkungan konsultasi digunakan oleh pengguna bukan pakar guna memperoleh pengetahuan atau nasehat pakar.

Ada tiga komponen utama dalam sistem pakar yaitu basis penge-tahuan, mesin inferensi, dan antarmuka pengguna. Sistem pakar yang berinteraksi dengan pengguna dapat berisi komponen tambahan : akuisisi pengetahuan, tempat kerja, fasilitas penjelasan, perbaikan pengetahuan.

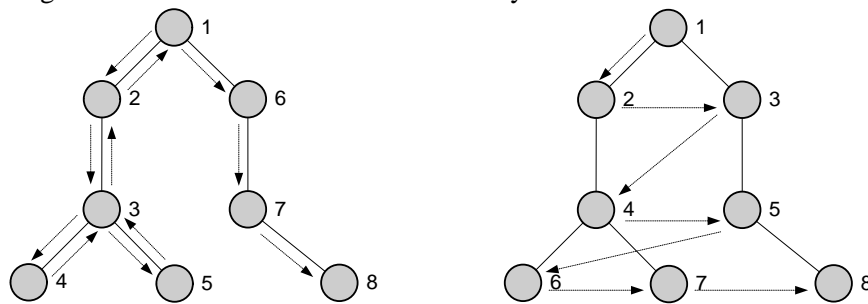
o Basis Pengetahuan (*Knowledge Base*)

Basis Pengetahuan merupakan inti program dari suatu sistem pakar dimana, basis pengetahuan yang mengandung pemahaman, formulasi, dan penyelesaian masalah, merupakan representasi pengetahuan dari seorang pakar yang ahli dibidangnya. Basis pengetahuan tersusun atas fakta dan aturan. Fakta adalah informasi tentang objek, peristiwa atau situasi. Sedangkan aturan adalah cara untuk membangkitkan atau menemukan fakta baru dari fakta yang telah ditemukan sebelumnya.

o Mesin Inferensi (*Inference Engine*)

Mesin inferensi merupakan mekanisme, pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan masalah. Mesin inferensi berperan sebagai 'otak' dalam memandu proses penalaran terhadap suatu kondisi, berdasar pada basis pengetahuan yang tersedia. Dalam mesin inferensi terjadi proses manipulasi dan mengarahkan

aturan, model, dan fakta yang ada untuk mencapai solusi atau kesimpulan. Dalam melakukan inferensi ada dua cara, yaitu *Forward Chaining*, pelacakan yang dimulai dari sekumpulan data (fakta) terlebih dahulu untuk menguji kebenaran hipotesis kemudian menuju kesimpulan. Dan *Backward chaining*, pelacakan yang dimulai dari hipotesis terlebih dahulu dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan. Selain teknik pelacakan tersebut, diperlukan juga teknik penelusuran data dalam bentuk jaringan yang terdiri atas node-node berbentuk pohon (*tree*). Ada tiga teknik penelusuran data yang digunakan yaitu : (a) *Depth-first search*: Merupakan teknik penelusuran data pada node-node secara vertikal dan mendalam.(b) *Breadth-first search*: Merupakan teknik penelusuran data pada semua node dalam satu level atau satu tingkatan sebelum level atau tingkatan berikutnya.(c) *Best-first search*: Merupakan teknik penelusuran data yang mengkombinasikan kedua metode sebelumnya.



a. *Depth-first search*

b. *Breadth-first search*

Gambar Teknik Penelusuran Data

o **Antarmuka Pemakai (*User Interface*)**

Merupakan perantara atau penghubung antara program aplikasi sistem pakar dengan pengguna. Biasanya pada bagian ini akan terjadi dialog atau menu-menu pilihan yang nantinya harus dijawab oleh pengguna agar sistem pakar dapat mengambil keputusan berdasarkan jawaban dari pengguna.

o **Akuisisi Pengetahuan (*Knowledge Aquisition*)**

Akuisisi pengetahuan adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Dalam tahap ini perekayasa sistem (*Knowledge engineer*) berusaha menyerap pengetahuan dari seorang pakar untuk selanjutnya ditransfer ke dalam basis pengetahuan.

o **Tempat kerja (*Workplace*)**

Workplace merupakan area dari sekumpulan memori kerja (*working memory*). *Workplace* digunakan untuk merekam fakta-fakta yang diperlukan, dimana fakta tersebut digunakan untuk memenuhi kondisi dari aturan dalam sistem. Memori kerja menyimpan semua fakta, baik fakta awal pada saat sistem mulai beroperasi, maupun fakta-fakta yang diperoleh pada saat proses penarikan kesimpulan sedang dilaksanakan. Memori kerja digunakan untuk menyimpan data hasil observasi dan data lain yang dibutuhkan selama pemrosesan.

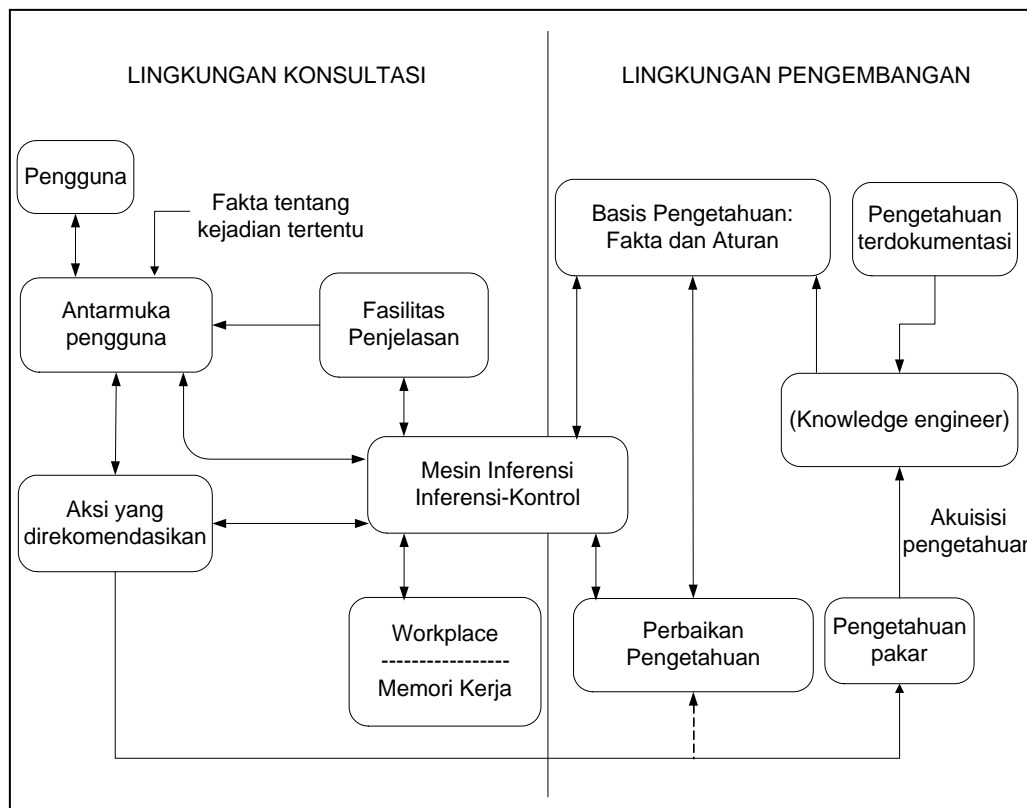
o **Fasilitas Penjelasan**

Fasilitas ini merupakan komponen tambahan yang akan meningkatkan kemampuan sistem pakar. Komponen ini menggambarkan suatu penalaran sistem kepada pengguna.

o **Perbaikan Pengetahuan**

Pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut adalah penting dalam

pembelajaran, sehingga mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya dan selalu memperbaiki pengetahuan berdasarkan pengalamannya.



Gambar Arsitektur sistem pakar (sumber: Turban(2005)).

8. Unsur Yang Terlibat Dalam Sistem Pakar

Dalam lingkungan sistem pakar terdapat beberapa unsur yang terlibat dan berinteraksi didalamnya, yaitu:

- o **Pakar (Expert)**

Pakar adalah orang yang memiliki pengetahuan, penilaian, pengalaman dan metode khusus, serta kemampuan untuk mengaplikasikan keahliannya tersebut dalam memberi nasehat dan memecahkan/menyelesaikan permasalahan.

- o **Perekayasa Sistem (Knowledge Engineer)**

Perekayasa Sistem adalah orang yang membantu pakar dalam menyusun area permasalahan dengan menginterpretasikan dan mengintegrasikan jawaban-jawaban pakar dari pertanyaan yang diajukan, dan menerangkan kesulitan-kesulitan konseptual.

- o **Pengguna (User)**

Pengguna adalah seseorang yang memanfaatkan sistem untuk mendapatkan saran yang disediakan oleh pakar. Sistem pakar memiliki beberapa pengguna, antara lain : pakar, pengguna bukan pakar, pelajar, dan pembangun sistem yang ingin meningkatkan dan menambah basis pengetahuan.

9. Bentuk Sistem Pakar

Ada 4 (empat) bentuk sistem pakar, yaitu :

- **Berdiri sendiri**

Sistem pakar jenis ini merupakan software yang berdiri sendiri tidak tergabung dengan software lainnya, artinya program utama tanpa mengandung subrutin yang memakai teknik algoritma konvensional.

- **Tergabung**

Sistem pakar jenis ini merupakan bagian programnya terkandung di dalam suatu algoritma konvensional atau merupakan program dimana di dalamnya memanggil algoritma subrutin lain konvensional, artinya program sistem pakar memiliki sebuah subrutin, misalnya untuk perhitungan matematik, membuat grafik, dan keperluan lain, dimana subrutin ini memakai algoritma konvensional atau bisa juga sebaliknya yaitu sebuah sistem pakar yang dikelilingi program lainnya, artinya sebuah subrutin yang akan dipanggil oleh program utama, dimana program utama ini memakai algoritma konvensional.

- **Terhubung**

Bentuk ini biasanya merupakan sistem pakar yang menghubungkan ke suatu program tertentu, artinya program dapat berhubungan dengan paket-paket program yang lain sebagai pendukung, seperti paket program database (Oracle, MySQL, dBase, FoxBase, dll), spreadsheet (Excel, Quatro Pro, dll), ataupun program pembuat grafik dan software aplikasi lain.

- **Sistem mengabdikan**

Sistem pakar ini merupakan bagian dari komputer khusus yang dihubungkan dengan suatu fungsi tertentu. Misalnya sistem pakar yang digunakan untuk membantu menganalisa data radar.

10. Aplikasi Sistem Pakar

Ada beberapa masalah yang menjadi area luas aplikasi sistem pakar antara lain :

- **Interpretasi**

Digunakan untuk menganalisa data yang tidak lengkap, tidak teratur dan data yang kontradiktif atau pengambilan keputusan dari hasil observasi. Termasuk diantaranya : pengawasan, pengenalan ucapan, analisa citra, interpretasi sinyal dan beberapa analisis kecerdasan.

- **Diagnosis**

Menentukan sebab fungsi yang tidak berjalan sebagaimana mestinya dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati, diantaranya medis, elektronis, mekanis dan diagnosis perangkat lunak.

- **Perencanaan**

Merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu, diantaranya digunakan untuk perencanaan keuangan, komunikasi, pengembangan produk, manajemen proyek dan militer.

- **Perancangan**

Digunakan untuk menentukan konfigurasi komponen-komponen yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala-kendala tertentu, diantaranya layout sirkuit dan perancangan bangunan.

- **Prediksi**

Digunakan untuk memprediksi akibat-akibat yang dimungkinkan dari situasi-situasi tertentu, diantaranya ramalan cuaca, ramalan ekonomi (bursa saham, keuangan, estimasi harga, pemasaran), prediksi lalu lintas, militer, dll. Prediksi biasanya dilakukan dengan melihat data-data ataupun informasi yang sebelumnya telah diperoleh.

- **Simulasi**

Digunakan untuk mensimulasikan sebuah permasalahan untuk dapat menemukan jawaban atas permasalahan tersebut.

- **Pengendalian**

Digunakan untuk mengatur tingkah laku suatu lingkungan yang kompleks seperti kontrol terhadap interpretasi-interpretasi, prediksi, perbaikan dan monitoring kelakuan sistem.

- **Monitoring**

Membandingkan suatu sistem yang teramati dengan tingkah laku yang diharapkan dari sistem, seperti *Computer Aided Monitoring System*.

- **Debugging dan repair**

Menentukan dan mengimplementasikan cara-cara untuk mengatasi malfungsi, diantaranya memberikan cara mengatasi suatu kegagalan.

11. Teknik Pencarian Heuristik

Heuristik adalah sebuah teknik yang mengembangkan efisiensi dalam proses pencarian, namun dengan kemungkinan mengorbankan kelengkapan (*completeness*). Untuk dapat menerapkan heuristik dengan baik dalam suatu domain tertentu, diperlukan suatu fungsi heuristik.

Fungsi heuristik digunakan untuk mengevaluasi keadaan-keadaan problema individual dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan.

Menurut Rich dan Knight (1991), dan Sri Kusumadewi, pencarian heuristik (*heuristic searching*) mempunyai beberapa jenis, antara lain:

- **Generate And Test**

Metode *generate and test* merupakan penggabungan antara *depth-first search* dengan pelacakan mundur (*backtracking*), yaitu bergerak kebelakang menuju pada suatu keadaan awal. Pendekatan ini meliputi langkah-langkah sebagai berikut :

- Bangkitkan suatu kemungkinan solusi (membangkitkan suatu titik tertentu atau lintasan tertentu dari keadaan awal).
- Uji untuk melihat apakah node tersebut benar-benar merupakan solusinya dengan cara membandingkan node tersebut atau node akhir dari suatu lintasan yang dipilih dengan kumpulan tujuan yang diharapkan.
- Jika solusi ditemukan, keluar. Jika tidak, ulangi kembali langkah pertama.
- Jika pembangkitan/pembuatan solusi-solusi yang dimungkinkan dapat dilakukan secara sistematis, maka prosedur ini akan dapat segera menemukan solusinya, (bila ada). Namun, jika ruang problema sangat besar, maka proses ini akan membutuhkan waktu yang lama. Metode *generate and test* ini kurang efisien untuk masalah yang kompleks.

- **Hill Climbing**

Metode *hill climbing* merupakan salah satu variasi metode (*generate and test*) dimana umpan balik yang berasal dari prosedur uji digunakan untuk memutuskan arah gerak dalam ruang pencarian (*search*). Dalam prosedur *generate and test* yang murni, respon fungsi uji hanyalah *ya* atau *tidak*. Dalam prosedur *hill climbing*, fungsi uji dikombinasikan dengan fungsi heuristik yang menyediakan pengukuran kedekatan suatu keadaan yang diberikan dengan tujuan (*goal*).

Algoritma *Simple Hill Climbing*:

- Cari operator yang belum pernah digunakan (gunakan operator ini untuk mendapatkan keadaan yang baru).
- Kerjakan langkah-langkah berikut sampai solusinya ditemukan atau sampai tidak ada operator baru yang akan diaplikasikan pada keadaan sekarang.
- Evaluasi keadaan baru tersebut, jika:
 - Keadaan baru merupakan tujuan, maka keluar.
 - Bukan tujuan, namun nilainya lebih baik dari sekarang, maka jadikan keadaan baru tersebut menjadi keadaan sekarang.
 - Keadaan baru tidak lebih baik dari pada keadaan sekarang, maka lanjutkan iterasi.

Masalah-masalah yang mungkin timbul pada *simple hill climbing*:

- Algoritma akan berhenti kalau mencapai nilai optimum lokal
- Urutan penggunaan operator akan sangat berpengaruh pada penemuan solusi.
- Tidak diijinkan untuk melihat satupun langkah sebelumnya.

o **Best-First Search**

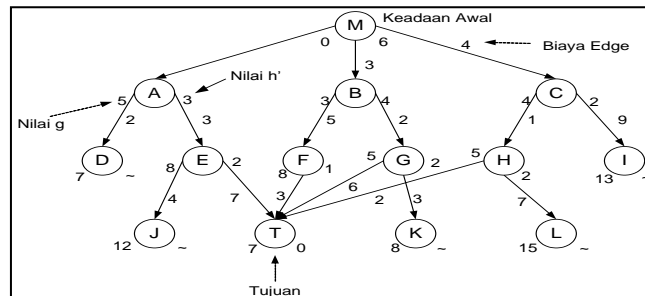
Metode *Best-First Search* merupakan kombinasi dari metode *depth-first search* dan *breadth-first search*. Pada metode *best-first search*, pencarian diperbolehkan mengunjungi node yang ada di level yang lebih rendah, jika ternyata node pada level yang lebih tinggi ternyata memiliki nilai heuristik yang lebih buruk.

Fungsi Heuristik yang digunakan merupakan prakiraan (estimasi) cost dari initial state ke goal state, yang dinyatakan dengan : $f^*(n) = g(n) + h'(n)$

Dimana: f^* = Fungsi evaluasi

g = cost dari initial state ke current state

h' = prakiraan cost dari current state ke goal state



Gambar Pohon Penelusuran (*Best First Search*)

Dari gambar diatas dapat dijelaskan bahwa node M merupakan keadaan awal dan node T merupakan tujuannya (goal). Biaya edge yang meng-hubungkan node M dengan node A adalah biaya yang dikeluarkan untuk bergerak dari kota M ke kota A. Nilai g diperoleh berdasarkan biaya edge minimal. Sedangkan nilai h' di node A merupakan hasil perkiraan terhadap biaya yang diperlukan dari node A untuk sampai ke tujuan. $h'(n)$ bernilai \sim jika sudah jelas tidak ada hubungan antara node n dengan node tujuan (jalan buntu). Kita bisa merunut nilai untuk setiap node.

- **Problem Reduction**

- **AND-OR Graphs**

AND-OR Graphs (tree), digunakan untuk memperlihatkan solusi dari permasalahan yang dapat diselesaikan dengan mendekomposisi permasalahan tersebut menjadi sekumpulan masalah yang kecil, dimana semuanya harus dapat diselesaikan. Pemecahan atau pereduksian ini membangkitkan *arcs* (busur) yang dikenal dengan *AND arcs*. Satu busur AND akan menghasilkan beberapa nomor dari simpul setelahnya. Semuanya harus diselesaikan supaya pancaran menghasilkan solusi. Hanya saja pada OR graph, banyak pancaran akan muncul dari beberapa node, mengindikasikan beberapa jalan yang permasalahannya bisa diselesaikan.

Untuk menemukan solusi dari sebuah AND-OR graph, diperlukan sebuah algoritma serupa dengan BFS tapi dengan kemampuan untuk menangani pancaran AND dengan tepat. Algoritma ini harus menemukan bagian dari simpul awal dari graph untuk menghimpun simpul-simpul yang menggambarkan *state* solusi. Perlu diketahui bahwa ini mungkin diperlukan untuk mendapatkan lebih dari satu solusi sejak setiap pancaran.

- **Constraint Satisfaction**

Constraint Satisfaction adalah sebuah prosedur pencarian yang dioperasikan pada sekumpulan ruang batas. *Initial state* terdiri dari batas-batas yang benar-benar diberikan dalam deskripsi masalah.

- **Means-Ends Analisis (MEA)**

Sejauh ini terdapat beberapa strategi pencarian *forward* dan *backward* tetapi untuk sebuah problem yang diberikan secara langsung atau harus dipilih, bagaimanapun gabungan dari dua arah tersebut adalah lebih tepat. *Means-Ends Analisis* adalah strategi penyelesaian masalah yang diperkenalkan pertama kali dalam GPS (*General Problem Solver*). MEA adalah pusat pemrosesan disekitar pendeteksian perbedaan antara *current state* dan *goal state*. Sekalipun perbedaan diisolasi, sebuah operator yang dapat mengurangi perbedaan harus ditemukan. Tetapi kemungkinan operator tidak dapat diterapkan pada *current state*.

b. Wireless Application Protocol (WAP)

Sejak pertengahan tahun 1990-an hingga kini, Internet telah mengubah cara manusia berkomunikasi dan berinteraksi. Perkembangan teknologi internet yang pesat memungkinkan untuk mendapatkan akses yang cepat terhadap pertukaran dan kebutuhan informasi dalam lingkup yang global, yaitu dunia. Saat ini internet menyajikan informasi tanpa dibatasi ruang dan waktu, informasi sudah menjadi wilayah publik, yang dapat diakses oleh siapa saja, kapan saja dan dimana saja.

Sementara itu dengan perkembangan teknologi nirkabel (*wireless*) yang demikian pesatnya saat ini, telah membangkitkan gagasan-gagasan tentang akses internet dan informasi dari perangkat komunikasi personal nirkabel dengan tingkat mobilitas tinggi, sehingga untuk mengakses informasi di internet dapat dilakukan melalui perangkat-perangkat nirkabel seperti telepon seluler atau *Personal Digital Assistant* (PDA). Teknologi ini memberikan keuntungan serta kemudahan dalam mendapatkan informasi karena dapat dilakukan dimana saja pengguna berada dan kapan saja.

Bagaimana perangkat-perangkat komunikasi personal nirkabel dapat mengakses internet? teknologi *Wireless Application Protocol* (WAP) adalah jawabannya, dengan teknologi ini perangkat komunikasi nirkabel tetap bisa melakukan komunikasi seperti biasa dan juga dapat mengakses internet. WAP merupakan sinergi dari kombinasi internet dan dunia komunikasi nirkabel. WAP merupakan protokol yang mengatur hubungan antara perangkat nirkabel dengan jaringan internet.

1. Apa itu WAP ?

WAP merupakan standar di seluruh dunia dalam menyediakan komunikasi internet dan mengedepankan layanan pada perangkat nirkabel (*wireless*) seperti telepon seluler, PDA, dan peralatan nirkabel lainnya. Protokol ini adalah suatu spesifikasi global yang mengijinkan bagi user yang memiliki perangkat nirkabel dapat dengan leluasa untuk mengakses dan saling berhubungan, baik dalam bentuk yang berhubungan dengan telekomunikasi maupun aplikasi-aplikasi berorientasi internet.

Struktur WAP mengadopsi topologi lapisan-lapisan yang ada pada *Internet Protocol* (model TCP/IP). Ini terkait dengan tujuan dibuatnya WAP, yaitu memberikan akses internet bagi perangkat nirkabel. Protokol mengatur bagaimana format paket data dan layanan terhadap paket data pada setiap lapisan, bagaimana suatu lapisan memberikan layanan kepada lapisan lain yang berada di atasnya.

2. Sejarah WAP

Sebelum dibangunnya protokol WAP, masing-masing organisasi maupun perusahaan komunikasi memiliki protokol yang mereka bangun sendiri-sendiri. Seperti Unwired Planet dengan *Handled Device Markup Language* (HDML) sebagai protokol untuk akses internet melalui paket data digital seluler. Lalu, Nokia dengan protokol *Tagged Text Markup Language* (TTML) yang dirancang untuk GSM. Ataupun Ericsson dengan *Intelligent Terminal Transfer Protocol* (ITTP).

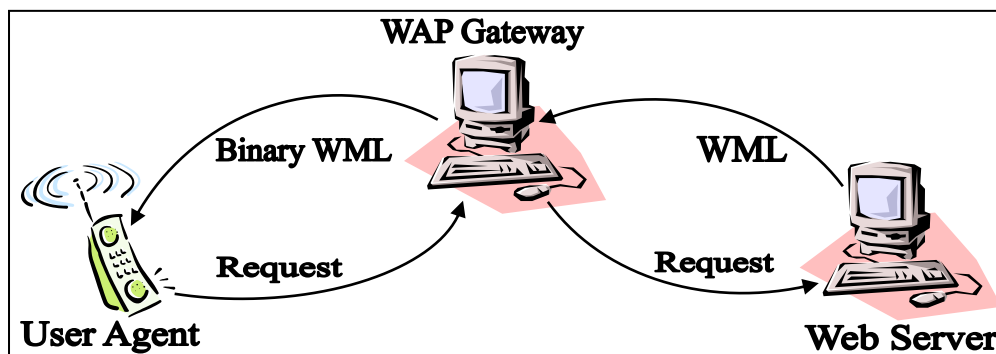
Pada pertengahan tahun 1997, empat perusahaan yaitu Nokia, Ericsson, Motorola, dan Phone.com sepakat untuk membangun protokol baru untuk perangkat *mobile* nirkabel, yaitu WAP. Tujuan utama dari pendefinisian protokol yang umum dan global ini adalah supaya protokol ini nantinya independen terhadap media seluler yang ada (misalnya GSM, D-AMPS, CDMA dan PCS).

Rancangan pertama dari spesifikasi WAP 1.0 diluncurkan pada bulan april 1998. Kemudian pada bulan mei tahun 1999, tepatnya 11 bulan setelah peluncuran spesifikasi WAP 1.0, versi komersial pertama dari WAP, yaitu WAP 1.1 telah diselesaikan. Perkembangan ponsel sampai dengan saat ini sudah banyak yang sudah dilengkapi dengan fasilitas WAP versi 2.0.

3. Mengembangkan Aplikasi WAP

Pengembangan aplikasi WAP dilakukan dalam suatu lingkungan kerja yang disebut *Wireless Application Environment* (WAE). Inti dari WAE ini terdiri dari *Wireless Markup Language* (WML) dan *Wireless Markup Language Script* (WMLScript).

Untuk menjangkau dunia internet, sebuah ponsel dengan teknologi WAP harus berjalan via WAP Gateway. WAP Gateway ini bertindak sebagai perantara, menghubungkan jaringan *mobile* dan internet dengan menerjemahkan *Hypertext Transfer Protocol* (HTTP) menjadi *Wireless Session Protocol* (WSP). Gambar di bawah ini menunjukkan skema sederhana hubungan antara web server, gateway dan ponsel dengan WAP.



Gambar Diagram network pada WAP

Web server melayani permintaan dari user melalui ponsel untuk sebuah aplikasi WAP melalui perantara WAP Gateway.

Aplikasi dalam WAP dibentuk dalam format WML. Untuk menjalankan suatu aplikasi WAP, sama halnya dengan internet biasa, yaitu tinggal mengetikkan alamat URL yang dikehendaki, misalnya : <http://wap.winwap.com>.

Untuk membuat aplikasi WAP yang dibutuhkan adalah sebuah web server untuk menangani permintaan user akan aplikasi WAP, misalnya Apache, *Microsoft Internet Information Service* (IIS), ataupun PWS (*Personal Web Server*).

4. Pengembangan WAP dengan WML

WML merupakan bahasa mark-up yang berbasis pada *Extensible Markup Language* (XML). WML adalah analogi dari HTML yang berjalan pada protokol nirkabel. Tag-tag pada WML mirip dengan tag-tag yang ada pada HTML. Data WML terstruktur dalam bentuk koleksi kartu atau *card*. Sebuah koleksi *card* disebut *deck*. Tiap *deck* tersusun dari isi yang terstruktur dan spesifikasi navigasi. Pengguna melakukan navigasi dalam susunan *card*, melihat isi tiap *card*, mengisi informasi yang dibutuhkan, membuat pilihan dan bernavigasi ke *card* selanjutnya atau kembali ke *card* sebelumnya.

Jika WML merupakan analogi dari HTML pada media nirkabel, maka WMLScript merupakan analogi yang tepat dari JavaScript. WMLScript, seperti halnya JavaScript, berjalan pada sisi client (*client side scripting*). Bedanya, WMLScript tidak dapat ditempatkan menjadi satu dengan halaman WML yang menggunakan fungsi dari WMLScript.

Fungsi-fungsi WMLScript yang akan digunakan oleh halaman WML ditempatkan dalam file yang terpisah. Pemisahan ini memberikan suatu keuntungan, yaitu dalam fokus pembuatan aplikasi. Jika bekerja dengan halaman WML, maka hanya berfokus pada isi atau *user interface* pada halaman yang diinginkan. Dengan WMLScript, fokus hanya pada pembuatan prosedur atau fungsi dari logika pemrograman.

Aplikasi WML yang telah dibuat dapat diakses menggunakan browser yang disebut dengan *user agent* (UA). UA mendownload halaman WML dan atau WMLScript yang dibutuhkan dan merender halaman tersebut. Hasil render halaman WML amat bergantung pada tipe perangkat yang digunakan. Dan tampilan yang diperoleh mungkin berbeda antara ponsel dengan kemampuan grafis yang baik dengan yang hanya mendukung modus teks.

5. Struktur Dasar WML

Sebuah halaman WML terdiri dari bagian header dan body. Dalam header terdapat dua hal yang harus di deklarasikan, yaitu deklarasi versi XML yang akan di gunakan dan deklarasi *Document Type Definition* (DTD). Deklarasi DTD ini bertujuan agar tipe data dalam dokumen yang dibuat dapat dikenali dengan benar. Deklarasi header WML adalah sebagai berikut:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

Pada bagian body file WML diawali dan diakhiri dengan pasangan tag `<wml>` dan `</wml>`. Pasangan tag inilah yang disebut sebagai *deck*. Dalam satu *deck* dapat terdiri dari satu atau lebih *card*. Sebuah *card* dapat berisi satu atau lebih komponen sebagai berikut :

1. Teks terformat, dapat berupa teks, gambar, dan link.
2. Elemen `<input>`, untuk menerima input dari user berupa string.
3. Elemen `<select>`, user dapat memilih dari daftar pilihan.
4. Elemen `<fieldset>`, bertindak sebagai wadah bagi elemen lain.

Level sintaks yang menyusun suatu deck adalah sebagai berikut:

```
<wml>
  <!--keterangan kode dengan tag ini-->
  <head>
    Kontrol akses dan meta information
  </head>
  <template>
    Event pada level deck
  </template>
  <card>
    Event pada level card
  </card>
</wml>
```

6. Pengembangan Aplikasi WAP dengan PHP

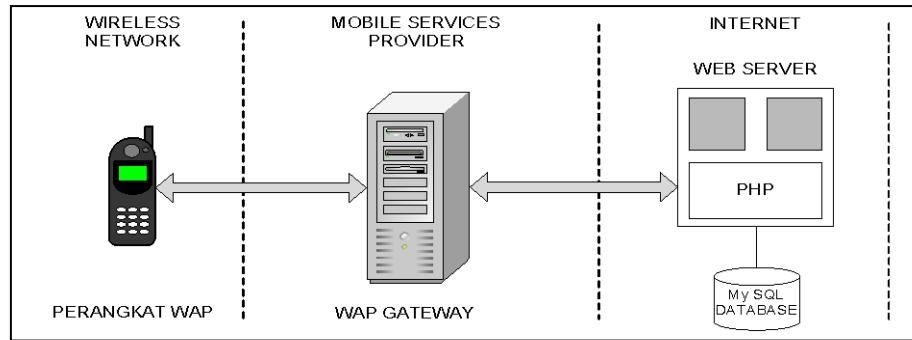
Untuk membuat aplikasi WAP menjadi lebih dinamis dan interaktif, yang mampu memberikan dan menerima respon dari dan ke *user*, dapat menggunakan bahasa-bahasa script yang berjalan pada sisi server (*server side-scripting*). Dalam hal ini penulis menggunakan PHP (*Hypertext Preprocessor*), bahasa *script serverside* yang tangguh, populer di internet dan gratis untuk memberi unsur dinamik dan interaktif pada aplikasi WAP.

Pada prinsipnya, komunikasi antara web server dengan perangkat WAP sama dengan hubungan antara web server dengan browser berbasis PC, hanya saja dalam hal ini ada satu tahap tambahan. Tahap ekstra yang dibutuhkan adalah transfer informasi oleh WAP gateway. WAP gateway bertindak sebagai perantara antara browser nirkabel dengan server tempat informasi berada.

Yang berperan sebagai WAP gateway biasanya adalah perusahaan telekomunikasi yang menyediakan layanan telepon seluler (*services provider*). Apabila sebuah *deck* telah terbuat, yang berisi dua buah *card*. Kemudian *user* dengan perangkat nirkabelnya melakukan permintaan terhadap *deck* tersebut. Maka, urutan *event* yang terjadi untuk permintaan *user* tersebut adalah sebagai berikut :

- Dari perangkat WAP permintaan dikirim ke WAP gateway dengan protokol WAP. WAP gateway melakukan *request* untuk URL tertentu dengan menggunakan protocol HTTP.
- Permintaan ditransmisikan via internet ke alamat IP dari perangkat WAP (alamat IP dari suatu perangkat WAP ditentukan oleh operator).
- Permintaan mencapai tujuan akhirnya, yaitu web server. Server membaca header dan memproses permintaan dokumen WAP. Kode program PHP yang terdapat dalam dokumen ini dikompilasi dan diformat sesuai dengan kebutuhan.
- Dokumen atau deck WAP yang telah diproses, dikirimkan kembali melalui WAP gateway. Pada gateway, isi dari deck dikompres menjadi data biner dan dikirimkan ke perangkat WAP (ponsel atau PDA).

Berikut ini adalah gambaran proses komunikasi antara browser nirkabel (ponsel atau PDA) dengan web server:



Gambar Proses komunikasi browser nirkabel dengan web server

Sangat memungkinkan dan mudah untuk menambahkan unsur dinamik ke dalam WML dengan PHP. Pengembangan aplikasi WAP dengan PHP memungkinkan untuk membuat aplikasi seperti database, mailserver, pengiriman pesan, dll. Supaya *script* PHP dapat didukung oleh perangkat WAP, *script* ini harus menghasilkan *output* header WML kepada *client*. Karena itu, setiap dokumen WML yang berisi kode PHP harus menyertakan baris-baris berikut yang ditempatkan pada awal *deck* :

```
<?php
//mengirim header
header("Content-type:text/vnd.wap.wml");
echo("<?xml version=\"1.0\"?>");
echo("<!DOCTYPE wml PUBLIC \"-//WAPFORUM//DTD WML
1.1//EN\" \"http://www.wapforum.org/DTD/wml_1.1.xml\">");
?>
```

Deklarasi ini diperlukan karena secara default PHP mengirim baris *Content-type: text/html*.

7. Interaksi PHP dengan MySQL

Komunikasi antara user dengan WAP browser dengan web server dapat menjadi lebih interaktif dengan penggunaan database. Dengan adanya PHP yang bekerja pada sisi server, komunikasi interaktif dapat dilakukan dengan antara *user* dengan *server*, baik Apache sebagai web server maupun database server MySQL. *User* yang mengakses dapat memperoleh data atau informasi dari *server* dan *server* dapat menyimpan data yang dikirimkan *user* dalam database MySQL.

Ada beberapa alasan menggunakan database MySQL, diantaranya adalah karena MySQL gratis serta mudah dipelajari. Dalam PHP terdapat banyak fungsi yang digunakan sebagai penghubung atau antarmuka dengan MySQL sehingga data dalam database dapat dilihat dari internet. Banyak situs di internet yang menggunakan PHP-MySQL dalam mengembangkan situsnya.

6. Rancangan Penelitian

a. Analisis Permasalahan

Dalam mengidentifikasi penyakit pada tanaman jeruk oleh petani, ada beberapa permasalahan yang muncul, antara lain :

1. Petani jeruk harus memiliki pengalaman dan pengetahuan agar dapat melakukan identifikasi pada tanamannya.
2. Untuk mendapatkan pengalaman dan pengetahuan dalam mengidentifikasi penyakit tanaman jeruk membutuhkan waktu serta latihan yang cukup panjang.

Solusi untuk mengatasi permasalahan diatas adalah dengan membuat perangkat lunak yang dapat digunakan untuk membantu petani jeruk dalam mengidentifikasi penyakit tanaman jeruk secara mudah, cepat dan efisien.

b. Analisis Kebutuhan Pengguna

Adapun pengguna (*user*) yang terlibat di dalam sistem pakar untuk mengidentifikasi penyakit pada tanaman jeruk ini adalah pengguna umum (masyarakat umum) dan administrator (pengelola sistem), aktifitas serta hak akses masing-masing pengguna dibedakan berdasarkan fasilitas yang dimiliki.

c. Analisis Kebutuhan Sistem Pakar

o Basis Pengetahuan

Basis pengetahuan pada sistem pakar untuk mengidentifikasi jenis penyakit pada tanaman jeruk terdiri dari dua macam sumber pengetahuan yaitu fakta dan aturan. Fakta pengetahuan yang disimpan dalam memori kerja adalah: pengetahuan mengenai jenis penyakit tanaman jeruk, gejala-gejala yang ditimbulkan, penyebab timbulnya penyakit serta pengendalian penyakit tersebut.

Informasi pengetahuan untuk jenis penyakit terdiri dari 18 jenis penyakit yaitu: CVPD (Citrus Vein Phloem Degeneration), Tristeza, dll (jenis penyakit selengkapnya pada halaman lampiran), selain itu ditambah-kan pula kedalam basis pengetahuan informasi lainnya meliputi penjelasan mengenai penyebab dan pengembangbiakan penyakit tersebut, serta cara pengendalian (pencegahan dan penanggulangan) yang dapat dilakukan.

Sedangkan informasi pengetahuan untuk gejala atau tanda yang muncul pada tanaman terdiri dari nama gejala yang berasal dari tanda-tanda yang terjadi pada tanaman jeruk, yang menunjukkan adanya kemungkinan adanya serangan penyakit.

Adapun aturan merupakan pengetahuan yang disimpan berdasarkan pada keterkaitan antara penyakit dengan gejalanya. Jadi basis aturan yang digunakan dalam sistem pakar ini akan menghubungkan antara jenis penyakit dengan gejala-gejala yang menyebabkan penyakit pada tanaman jeruk dan aturan tersebut akan digambarkan dengan *tree*.

o Mekanisme Inferensi

Mekanisme inferensi mengandung suatu mekanisme pola pikir dan penalaran yang digunakan dalam menyelesaikan suatu masalah. Dalam hal ini sistem akan mendiagnosa jenis penyakit pada tanaman jeruk menggunakan pelacakan dengan metode *forward chaining* (pelacakan kedepan) dan *Backward Chaining* (pelacakan ke belakang).

• *Forward chaining*

Forward chaining adalah metode bagaimana sistem dapat meng-ambil kesimpulan berdasarkan jawaban dari pengguna atas pertanyaan yang dimunculkan oleh sistem. Pertanyaan merupakan gejala-gejala penyakit yang terjadi pada tanaman.

Metode *forward chaining* yang dipakai dalam sistem pakar ini, dimana pelacakan tersebut diawali dengan pertanyaan tentang kondisi tanaman jeruk (apakah tanaman anda

sudah besar atau dewasa?) hal ini ditanyakan karena ada tiga kondisi tanaman dalam pembagian gejala penyakit yaitu dewasa, berbuah, dan kecil/masa pembibitan.

Kemudian pengguna diberi pilihan jawaban Ya atau Tidak dengan asumsi, 'ya' apabila sesuai dengan kondisi tanaman dan 'tidak' apabila tidak sesuai. Selanjutnya sistem akan memberikan pertanyaan mengenai gejala yang sesuai dengan kondisi tanaman dan harus terus dijawab dengan ya/tidak, sampai sistem memunculkan hasil diagnosa penyakit.

- **Backward chaining**

Backward Chaining adalah metode yang akan menampilkan semua gejala penyakit yang nama penyakitnya sudah diketahui dan di inputkan ke dalam sistem oleh pengguna.

Metode *backward chaining* yang dipakai dalam sistem pakar ini diawali dengan munculnya semua jenis penyakit pada tanaman jeruk, dan selanjutnya pengguna diminta untuk memilih jenis penyakit yang gejalanya akan dimunculkan. Setelah kode penyakit dimasukkan, pengguna tinggal memilih lihat gejala, maka sistem akan memunculkan semua gejala yang sesuai dengan jenis penyakit yang dipilih.

Pada penerapannya, metode *forward* dan *backward chaining* akan diimplementasikan kedalam serangkaian query database yang digunakan untuk melakukan penalaran, penelusuran, dan pencocokan data dari tabel-tabel yang saling berhubungan pada sistem pakar tersebut. Dan untuk mekanisme, pola pikir dan proses penalaran dalam mesin inferensi digambarkan dengan tree.

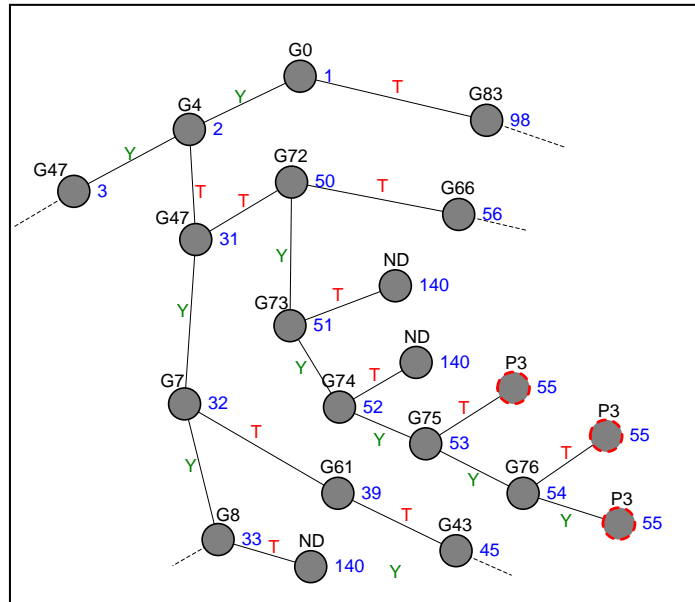
d. Perancangan Aturan

Pada perancangan aturan (*heuristik*) akan dikembangkan dengan pembuatan *tree* dan pengalihan aturan kedalam tabel (basis data) agar dapat berfungsi untuk mengarahkan penggunaan basis pengetahuan atau fakta yang ada, sehingga dapat dimanfaatkan untuk memecahkan suatu permasalahan dalam domain tertentu.

- **Pembuatan Tree (pohon penelusuran)**

Pembuatan *tree* pada perancangan aturan ini digunakan untuk mempermudah dalam proses penelusuran fakta yang akan dimasukkan ke dalam aturan pada perangkat lunak atau mesin inferensi yang akan dibuat. Pembuatan *tree* pada perancangan aturan ini menggunakan proses penalaran dengan metode *forward* dan *backward chaining*.

Dalam *tree* ini dapat dijelaskan bahwa aturan yang akan digunakan disesuaikan dengan alur sistem pakar (diagram alir sistem pakar) yang telah dirancang sebelumnya. Dibawah ini adalah sebagian hasil dari pembuatan *tree* (selengkapnya pada halaman lampiran).



Gambar Contoh hasil pembuatan tree

Dari contoh *tree* diatas, dapat dijelaskan bahwa aturan yang digunakan adalah sebagai berikut :

Sebagai langkah atau aturan awal (1) adalah G0 untuk menanyakan kondisi tanaman saat ini (apakah tanaman anda sudah besar atau dewasa?) hal ini ditanyakan karena ada tiga kondisi dalam pembagian gejala penyakit pada tanaman, yaitu : tanaman besar/dewasa, sedang berbuah, dan tanaman masih kecil/masa pembibitan.

Jika kondisi G0 memang benar, maka asumsi pilihan jawaban adalah “Ya”, dan gejala 4 (G4) pada aturan kedua (2) akan ditampilkan sebagai pertanyaan berikutnya. Tetapi jika G0 tidak benar, maka asumsi pilihan jawaban adalah “Tidak” dan kondisi G83 pada aturan ke 98 yang akan ditampilkan sebagai pertanyaan berikutnya.

Apabila gejala 4 memang benar (Ya), maka gejala 47 (G47) akan ditampilkan untuk pertanyaan selanjutnya, sebaliknya jika gejala 4 tidak benar (Tidak), maka akan sama gejala 47 (G47) aturan ke 31 juga yang akan ditampilkan sebagai pertanyaan selanjutnya. Hal ini dikarenakan ada jenis penyakit yang tidak mempunyai G4 tetapi mempunyai G47 dan ada yang memiliki dua gejala ini G4 dan G47.

Demikian proses-proses diatas akan berlangsung terus menerus hingga didapatkan hasil/kesimpulan dari setiap percabangan. Misalnya, jika pada gejala 71 (G71) aturan ke 54 memang benar (Ya) maka akan ditampilkan penyakit dengan kode P6 sebagai kesimpulan dan jika pada G71 tidak benar (tidak) juga akan ditampilkan penyakit dengan kode P6 sebagai kesimpulan, hal ini dikarenakan hasil penelusuran gejala sebelum-nya sudah mencukupi dalam mengidentifikasi penyakit tersebut jadi jawaban apapun akan mencapai kesimpulan.

○ **Pengalihan Aturan**

Pengalihan aturan ini merupakan implementasi dari pembuatan tree (pohon penelusuran). Jika tree dirancang dan dibuat dengan menunjukkan aturan berupa node-node (simpul), maka aturan tersebut dialihkan dalam bentuk tabel yang disimpan pada sebuah basis data. Pengalihan aturan ini dimaksudkan agar dapat digunakan untuk menjalankan mesin inferensi dalam sistem pakar.

Dibawah ini adalah contoh pengalihan aturan/rule dari rancangan aturan berupa pohon penelusuran (tree) kedalam sebuah tabel yang selanjut-nya akan disimpan kedalam

basis data agar dapat berfungsi sesuai yang diharapkan (tabel selengkapnya pada halaman lampiran).

Tabel 3.1 Tabel Aturan (heuristik)

| Aturan ke () | Kode gejala & penyakit | Jika 'ya' Aturan ke () | Jika 'tidak' Aturan ke () |
|--------------|------------------------|------------------------|---------------------------|
| 1 | G0 | 2 | 98 |
| 2 | G4 | 3 | 31 |
| 3 | G47 | 4 | 14 |
| 4 | G69 | 5 | 8 |
| 5 | G70 | 6 | 140 |
| 6 | G71 | 7 | 7 |
| 7 | P6 | 0 | 0 |
| ... | ... | ... | ... |

Dari contoh tabel aturan diatas dapat dijelaskan bahwa pada saat sistem dimulai, aturan pertama adalah aturan ke-1 yang akan menampilkan gejala penyakit atau kondisi tanaman dengan kode G0 yang merupakan pertanyaan pertama.

Selanjutnya jika pengguna memilih jawaban 'ya', maka sistem akan menuju pada aturan ke-2 yang akan menampilkan pertanyaan tentang gejala penyakit yang mempunyai kode G4. apabila dalam aturan pertama pengguna memilih jawaban tidak, maka sistem akan menuju aturan ke-98.

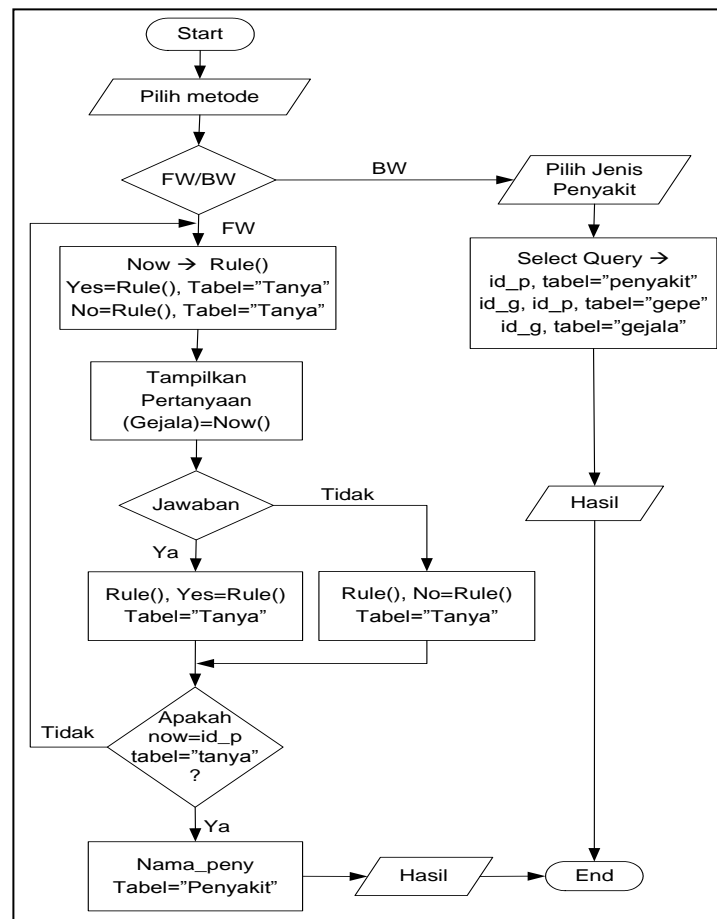
Sistem akan berjalan terus menerus sesuai jawaban dari pengguna sampai sistem menemukan kesimpulan yang merupakan hasil diagnosa atau identifikasi jenis penyakit dari gejala-gejala yang ditanyakan sebelumnya kepada pengguna.

e. Perancangan Proses

Perancangan proses akan menjelaskan bagaimana sistem pakar bekerja untuk mengolah data yang tersedia sehingga menjadi informasi yang berguna dengan fungsi-fungsi yang telah direncanakan yang digambarkan dengan diagram alir (*flowchart*). Perlu diketahui bahwa sistem ini nantinya akan digunakan oleh dua pengguna, yaitu pengguna umum dan administrator dan dibawah ini akan ditampilkan diagram alir untuk masing-masing pengguna tersebut.

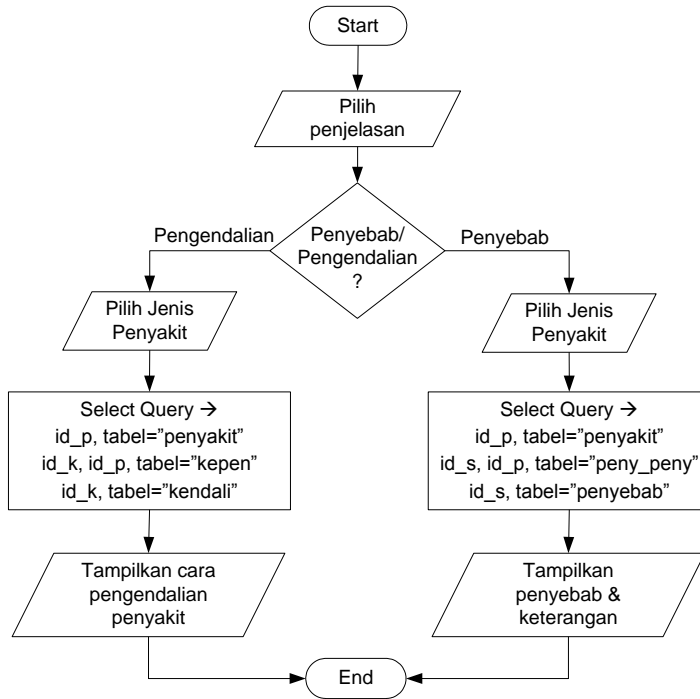
○ **Diagram Alir Sistem Pengguna Umum**

Diagram alir sistem pengguna umum adalah diagram alir yang menunjukkan bagaimana aliran proses yang terjadi dalam sistem pengguna umum. Bagaimana aliran proses jika menggunakan metode *backward chaining* atau *forward chaining*? Proses tersebut akan dijelaskan lebih lanjut dibawah ini.



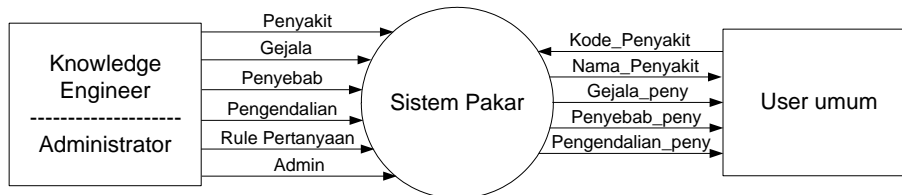
Gambar Diagram alir sistem pengguna umum

f. Perancangan Fasilitas Penjelasan

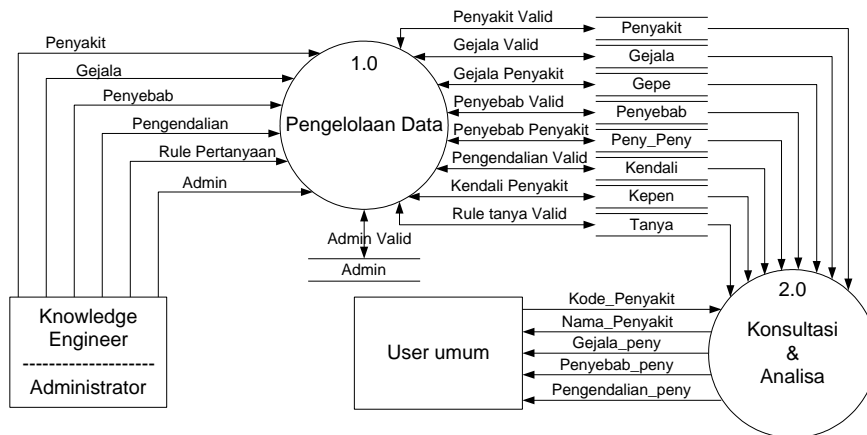


Gambar Diagram alir Fasilitas Penjelasan

g. Perancangan Sistem Context Diagram

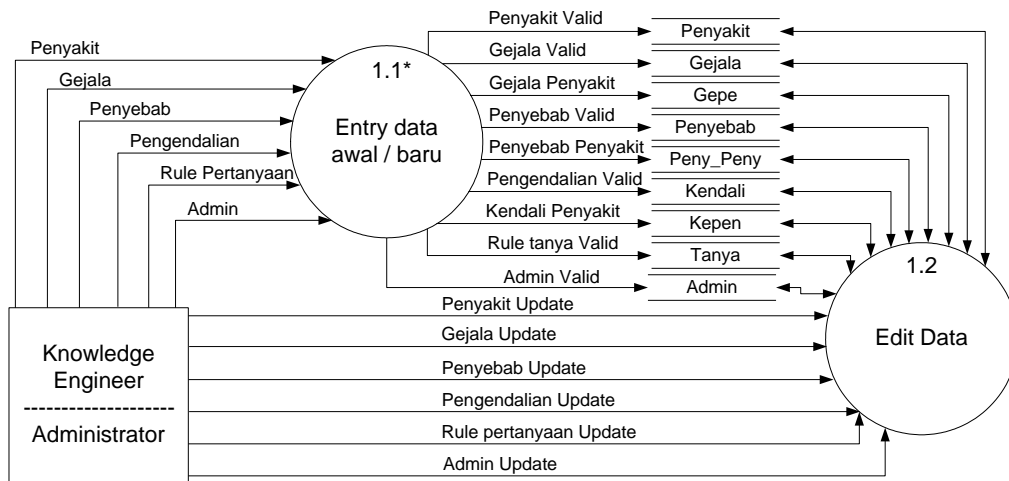


DFD Level 0



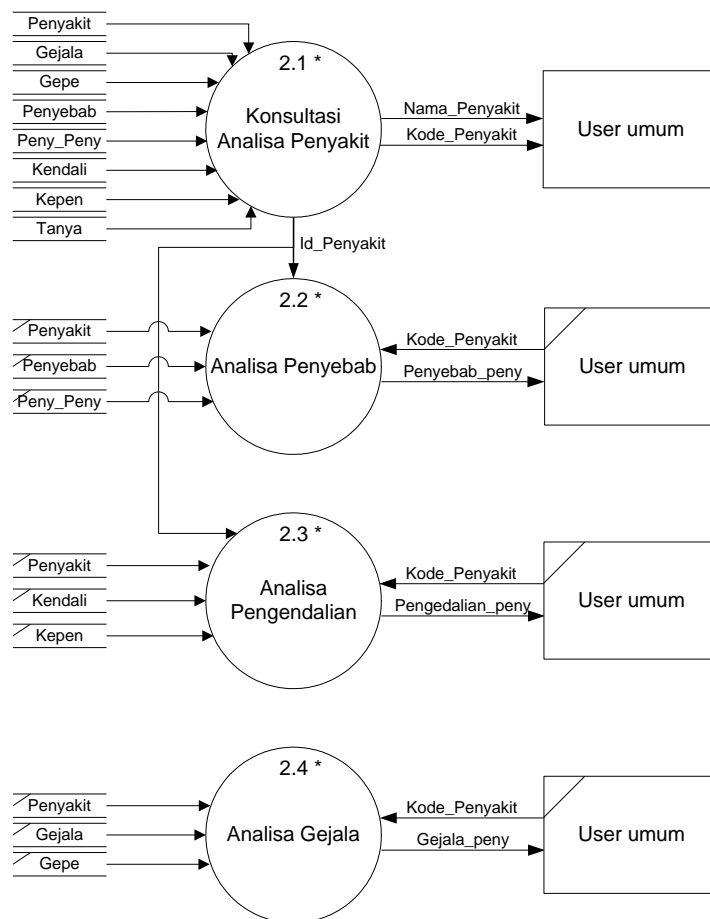
DFD Level 1

1. Level 1 proses 1.0 (pengelolaan data)

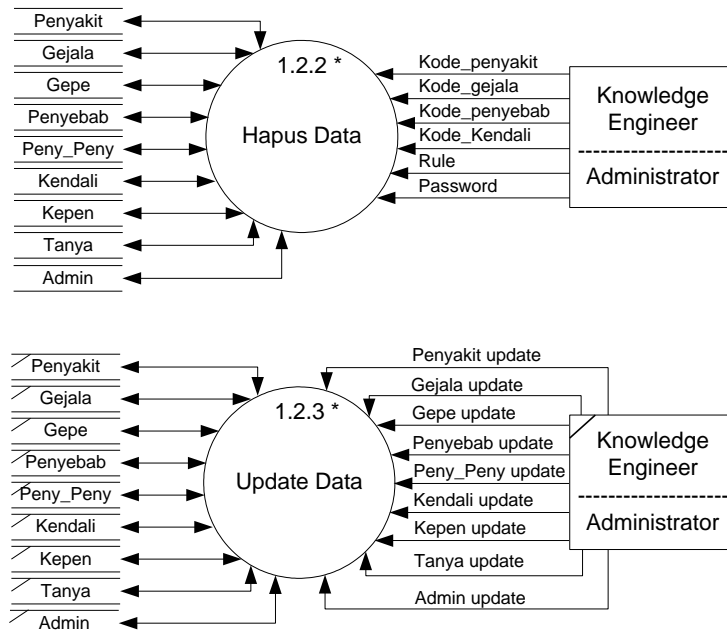


2. Level 1 proses 2.0 (konsultasi dan analisa)

DFD level 1 ini merupakan penjabaran dari level 0 proses 2



DFD Level 2



Gambar Data Flow Diagram Level 2

Perancangan Basis Data

1. Struktur Tabel

Tabel Penyakit

| No | Nama Field | Jenis | Panjang | Keterangan |
|----|------------|---------|---------|---------------|
| 1 | id_p PK | varchar | 5 | id_penyakit |
| 2 | nama_peny | varchar | 255 | nama penyakit |

Tabel Gejala

| No | Nama Field | Jenis | Panjang | Keterangan |
|----|-------------|---------|---------|-----------------|
| 1 | id_g PK | varchar | 5 | id_gejala |
| 2 | gejala_peny | varchar | 255 | gejala penyakit |

Tabel Gepe (gejala penyakit)

| No | Nama Field | Jenis | Panjang | Keterangan |
|----|------------|---------|---------|-------------|
| 1 | id_p FK | varchar | 5 | id_penyakit |
| 2 | id_g FK | varchar | 5 | id_gejala |

Tabel Penyebab

| No | Nama Field | Jenis | Panjang | Keterangan |
|----|------------|---------|---------|--|
| 1 | id_s PK | varchar | 5 | id_gejala |
| 2 | virus | varchar | 255 | nama virus / patogen penyebab penyakit |
| 3 | ket | text | - | keterangan |

Tabel Peny_peny (penyebab penyakit)

| No | Nama Field | Jenis | Panjang | Keterangan |
|----|------------|---------|---------|-------------|
| 1 | id_p FK | varchar | 5 | id_penyakit |
| 2 | id_s FK | varchar | 5 | id_penyebab |

Tabel Kendali

| No | Nama Field | Jenis | Panjang | Keterangan |
|----|------------|---------|---------|---------------------|
| 1 | id_k PK | varchar | 5 | id_pengendalian |
| 2 | kend | varchar | 255 | pengendali penyakit |

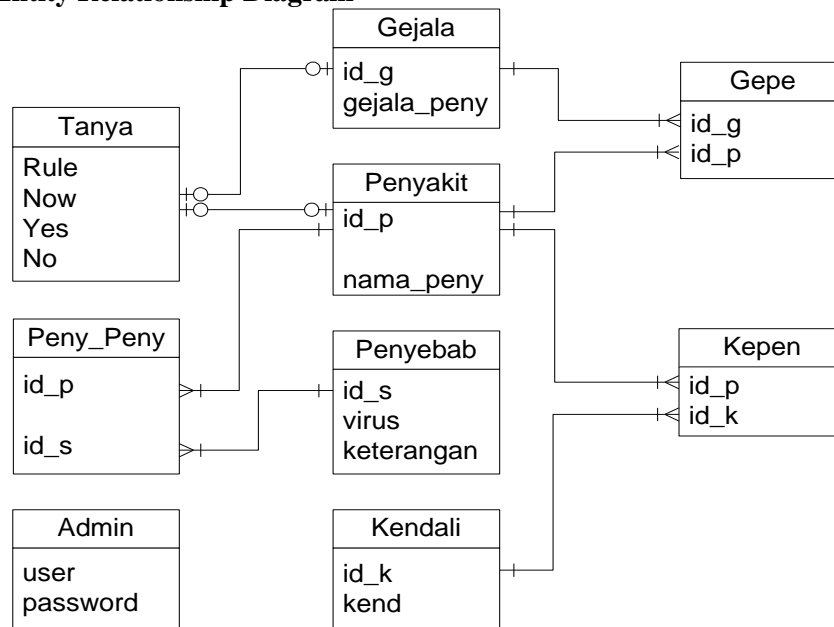
Tabel Kepen (kendali penyakit)

| No | Nama Field | Jenis | Panjang | Keterangan |
|----|------------|---------|---------|-----------------|
| 1 | id_p FK | varchar | 5 | id_penyakit |
| 2 | id_k FK | varchar | 5 | id_pengendalian |

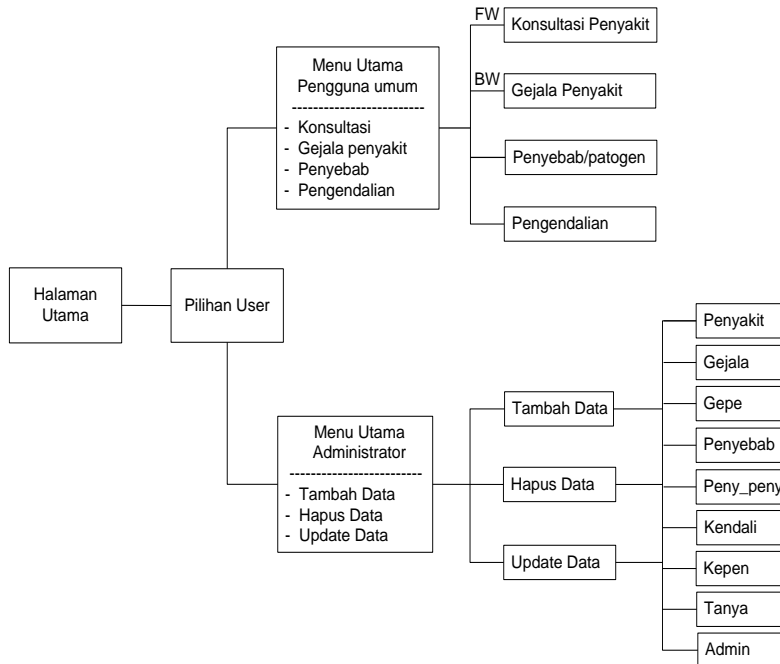
Tabel Tanya (rule pertanyaan)

| No | Nama Field | Jenis | Panjang | Keterangan |
|----|------------|---------|---------|--|
| 1 | rule PK | varchar | 5 | kode aturan/langkah |
| 2 | now | varchar | 5 | menampung id_ gejala/id_penyakit sebagai pertanyaan & kesimpulan |
| 3 | yes | varchar | 5 | menampung kode aturan berikutnya, apabila user memilih jawaban 'ya' |
| 4 | no | varchar | 5 | menampung kode aturan berikutnya, apabila user memilih jawaban 'tidak' |

2. Entity Relationship Diagram



Perancangan Antarmuka Pengguna



7. Rancangan Program

a. Implementasi Antarmuka Pengguna

Terdapat beberapa menu/halaman utama yang ada dalam aplikasi sistem pakar ini antara lain : halaman depan (utama), menu pilihan user, menu utama pengguna umum, dan menu utama administrator. Dibawah ini adalah tampilan menu/halaman utama pada emulator WAP.



Gambar Tampilan halaman utama pada emulator WAP



Gambar Halaman Pilihan User



Gambar Menu Utama User Umum



Gambar Menu Utama Administrator

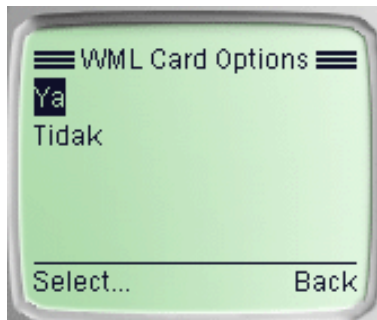
b. Pengujian Metode *Forward Chaining*



Gambar Halaman utama menu konsultasi



Gambar Tampilan pertanyaan 1



Gambar Menu pilihan jawaban pertanyaan



Gambar Tampilan pertanyaan 2

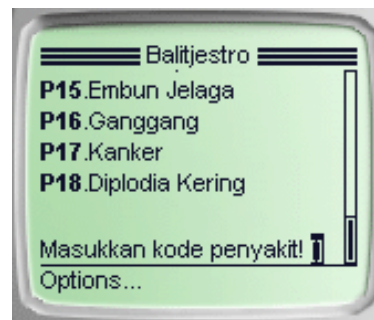


Gambar Hasil diagnosa penyakit

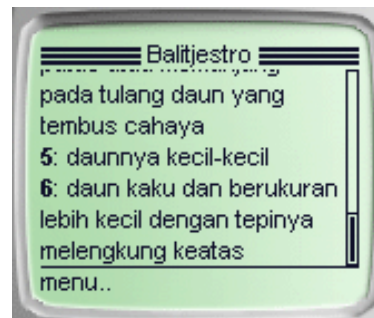


Gambar Kesimpulan penyakit tidak terdeteksi

c. Pengujian Metode *Backward Chaining*



Gambar Tampilan pilihan jenis penyakit



Gambar Tampilan informasi gejala penyakit

8. Kesimpulan

Setelah menguraikan secara keseluruhan perancangan serta melakukan implementasi dan pengujian aplikasi sistem pakar, maka dapat diperoleh kesimpulan sebagai berikut :

1. Sistem Pakar dibangun dengan metode *forward* dan *backward chaining* guna mendapatkan suatu kesimpulan atau untuk mengidentifikasi jenis penyakit pada tanaman jeruk serta pencarian gejala-gejala yang berhubungan dengan jenis penyakit.
2. Sistem Pakar dibangun berbasis *Wireless Aplication Protocol* (WAP) sehingga sistem dapat diakses menggunakan telepon seluler yang dapat digunakan dimanapun pengguna berada serta akan memudahkan dan menghemat waktu bagi pengguna.

3. Pengelolaan basis pengetahuan sistem pakar hanya dapat dilakukan oleh administrator. Pada sistem untuk administrator, dilengkapi dengan sistem login untuk menjaga kemungkinan agar perubahan data tidak dilakukan oleh selain administrator.
4. Dalam membangun sistem pakar berbasis *wireless application protocol* (WAP) menggunakan *wireless markup language* (WML) dan *php hypertext preprocessor* (PHP) dapat menyajikan aplikasi sistem secara optimal. Informasi untuk pengguna dapat ditampilkan serta berfungsi dengan baik dan sesuai dengan tujuan.

9. Daftar Pustaka

- Arhami, Muhammad. 2005. *Konsep Dasar Sistem Pakar*. Yogyakarta: Andi
- Dwiastuti, ME., dkk. 2004. *Pengenalan dan Pengendalian Hama Penyakit Tanaman Jeruk*. Batu : Loka Penelitian Tanaman Jeruk dan Hortikultura Subtropik, Pusat Penelitian dan Pengembangan Hortikultura, Badan Penelitian dan Pengembangan Pertanian, Departemen Pertanian.
- Suminar, farisi andri. *Cara Mudah Setting GPRS, WAP, dan MMS untuk semua ponsel*. www.serbadownload.web.id. Di download tanggal: 24 Maret 2006
- Imansyah, Muhammad. 2003. *PHP dan MySQL untuk orang awam*. Palembang : CV. Maxikom
- Karmilasari. *Sistem Inteleksi Buatan – Teknik Pencarian Heuristik*. <http://ocw.gunadarma.ac.id> . Di download tanggal: 15 Agustus 2008
- Kusrini. 2006. *Sistem Pakar, Teori dan Aplikasi*. Yogyakarta: Andi.
- Mobile Communication Laboratory STT Telkom Bandung. 2002. *Seri Penuntun Praktis Membangun Wireless Application Protocol (WAP)*. Jakarta : PT. Elex Media Komputindo.
- Nugroho, Bunafit. 2005. *Pengembangan Program WAP dengan WML dan PHP*. Yogyakarta: Gava Media.
- Nurhadi, Tyasno. 2003. *Pemrograman WML dan WMLS : Hadirkan Diri Anda di Mobile Internet*. Yogyakarta : Andi.
- Suyoto. 2004. *Inteleksi Buatan : Teori dan Pemrograman*. Yogyakarta: Gava Media.
- Syamsuddin, Aries. 2004. *Pengantar Sistem Pakar*. www.ilmukomputer.com . Di download tanggal: 19 Februari 2008.