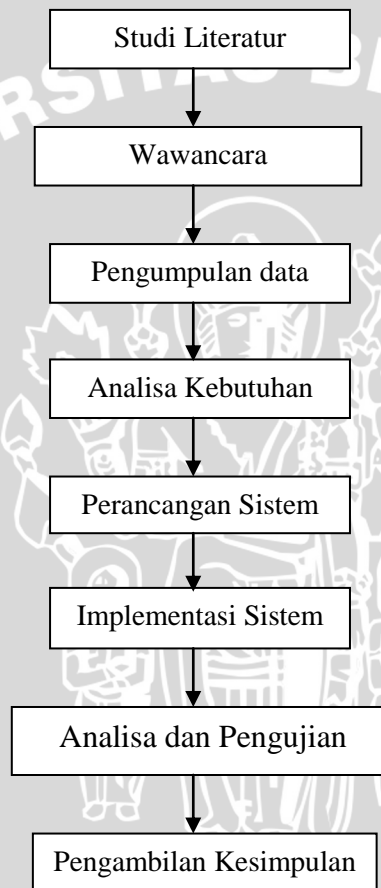


BAB III

METODE PENELITIAN

Bab ini menjelaskan langkah-langkah dalam pembuatan sistem, yaitu analisa kebutuhan, perancangan, implementasi dan pengujian aplikasi perangkat lunak. Berikut ini merupakan langkah-langkah pengerjaan yang dilustrasikan dalam diagram blok metodologi penelitian.



Gambar 3.1 Diagram Blok Metodologi Penelitian

3.1 Studi Literatur

Mempelajari *literature* dari beberapa bidang ilmu yang berhubungan dengan pembuatan sistem pakar untuk identifikasi hama penyakit pada budidaya tanaman jamur, diantaranya:

- Sistem Pakar

- Identifikasi berbagai jenis hama dan penyakit pada budidaya jamur
- *Certainty Factor*

3.2 Wawancara Narasumber

Wawancara dilakukan terhadap pakar untuk mendapatkan informasi tambahan, memastikan kebenaran informasi dari sumber lain (buku, internet), serta memastikan bentuk penerapan metode *certainty factor* terhadap proses identifikasi hama penyakit.

3.3 Pengumpulan Data

Sumber data diperoleh dari beberapa kali hasil wawancara terhadap ahli hama dan penyakit (*entomologi*), khususnya untuk data hasil pembobotan gejala yang menunjukkan besarnya pengaruh suatu gejala terhadap penyakit/hama yang menyerang. Sedangkan sumber data hama penyakit dengan gejalanya berdasarkan hasil *study literature* dengan pakar sebagai pihak yang memvalidkan kebenaran data.

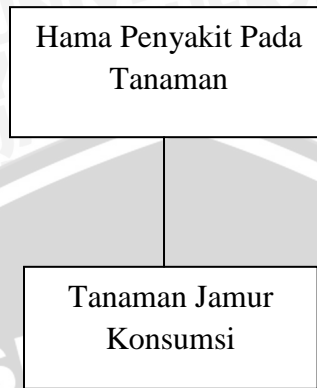
3.4 Analisa Kebutuhan

Analisa kebutuhan sistem bertujuan untuk mengidentifikasi apa saja yang dibutuhkan sistem agar tidak menyimpang dari permasalahan dan tujuan penelitian. Guna mendefinisikan kebutuhan sistem digunakan bentuk analisa terstruktur, yaitu menggunakan DFD (*Data Flow Diagram*).

3.5 Perancangan Sistem

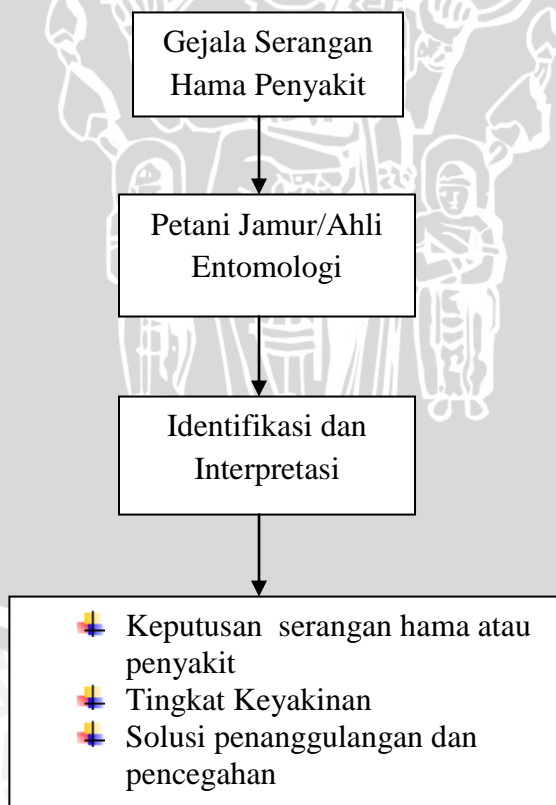
Sistem pakar yang akan dibangun digunakan untuk mengidentifikasi hama penyakit pada budidaya tanaman jamur konsumsi. Metode *certainty factor* digunakan untuk proses pengambilan keputusan, sedangkan penelusuran jawaban menggunakan metode *forward chaining*. Hasil output sistem terdiri dari: data hama atau penyakit yang menyerang, prosentase serangan berdasarkan tingkat keyakinan *certainty factor*, cara penanggulangan dan pencegahan serangan. Blok diagram area permasalahan ini menggambarkan tentang ruang lingkup permasalahan yang akan dibahas. Hama dan penyakit tanaman merupakan domain

yang lebih luas dari tanaman jamur konsumsi (*Edible Mushroom*). Domain inilah yang nantinya akan dibuat prototipenya.



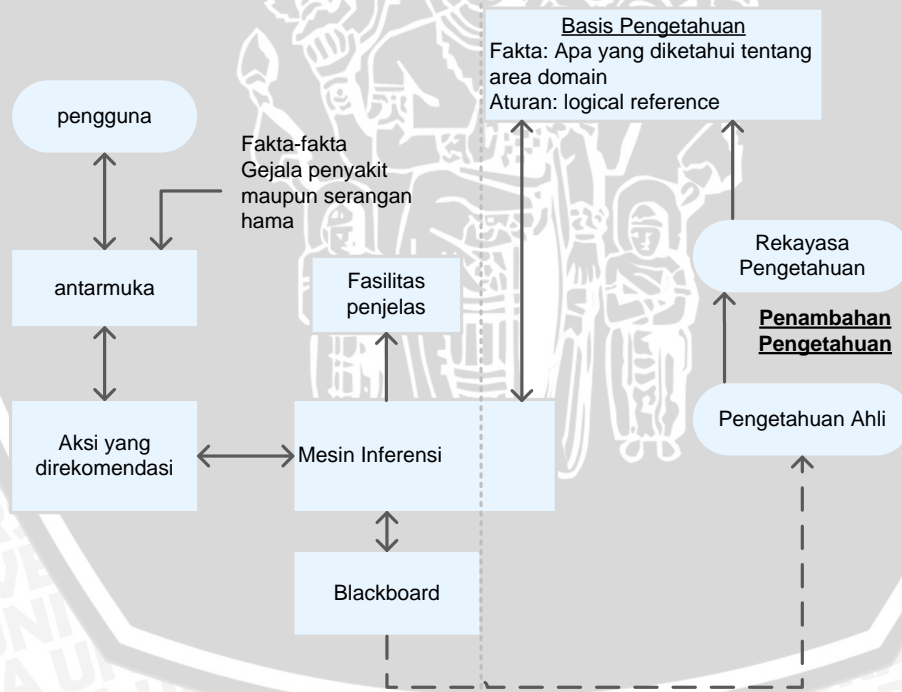
Gambar 3.2 Diagram Blok Area Permasalahan
Sumber: Perancangan

Tahapan-tahapan yang biasanya dilakukan dalam proses identifikasi hama penyakit diilustrasikan pada skema berikut.



Gambar 3.3 Skema Tahapan Identifikasi Hama Penyakit
Sumber: Perancangan

Tahapan yang biasa dilakukan baik oleh orang awam maupun seorang pakar dalam bidang hama penyakit dalam melakukan identifikasi hama penyakit adalah dengan melihat gejala fisik yang tampak pada tanaman jamur. Semakin spesifik yang dapat diamati, maka semakin besar tingkat keyakinan yang dapat dihasilkan. Konsep sistem pakar yang akan dibangun dengan menggunakan metode *certainty factor* mengadaptasi proses identifikasi secara manual. Sistem menerima inputan berupa keyakinan *user* terhadap gejala yang dapat diamati pada tanaman budidayanya, dengan begitu semakin besar tingkat keyakinan yang diinputkan dan semakin spesifik gejala yang dapat diamati maka diharapkan keputusannya pun dapat mencapai prosentase yang semakin tinggi. Hasil akhir berupa keputusan hama penyakit beserta prosentase tingkat keyakinan CF dan solusi untuk penanggulangan serta pencegahannya. Berikut merupakan gambar arsitektur sistem pakar yang telah mengacu pada konsep perancangan yang telah disebutkan diatas.



Gambar 3.4 Sistem Pakar Identifikasi Hama Penyakit Pada Budidaya Tanaman Jamur

Sumber: Perancangan

Bagian-bagian atau komponen yang membangun sistem pakar berbasis *web* untuk identifikasi hama penyakit pada budidaya tanaman jamur meliputi:

3.5.1 Antarmuka

Antarmuka merupakan mekanisme yang digunakan oleh *user* untuk berkomunikasi dengan sistem pakar seperti melihat informasi yang ada di dalam sistem, melakukan registrasi dan melakukan konsultasi. Antarmuka sistem pakar untuk identifikasi dirancang berdasarkan otoritas yang disediakan oleh sistem, baik untuk pakar maupun untuk user.

3.5.1.1 Basis Pengetahuan

Basis pengetahuan berisi pengetahuan relevan yang diperlukan untuk memahami, merumuskan, dan memecahkan persoalan. Basis tersebut mencakup dua elemen dasar yaitu fakta dan aturan khusus yang mengarahkan penggunaan pengetahuan untuk memecahkan persoalan khusus dalam domain tertentu.

Representasi pengetahuan dibutuhkan untuk menangkap sifat-sifat penting masalah dan mempermudah prosedur pemecahan masalah dalam mengakses informasi. Representasi pengetahuan yang digunakan pada skripsi ini yaitu aturan produksi yang dituliskan dalam bentuk jika-maka (*IF-THEN*). Struktur aturan produksi yang menghubungkan premis dengan konklusi untuk setiap rule dapat dijelaskan sebagai berikut:

R1: IF E1 and E2 THEN H CF(R1)

Setiap evidence dalam premis majemuk dipecah menjadi premis tunggal dengan representasi sebagai berikut:

R1.1: IF E1 THEN H dengan CF(R1.1)

R1.2: IF E2 THEN H dengan CF(R1.2)

3.5.1.2 Akuisisi pengetahuan

Akuisisi pengetahuan adalah akumulasi, transfer, dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Pada tahap ini *knowledge engineer* berusaha menyerap pengetahuan untuk selanjutnya ditransfer ke dalam basis pengetahuan. Pengetahuan yang diakuisisi adalah pengetahuan prosedural serta pengetahuan deklaratif. Pengetahuan yang ada diperoleh dari buku, internet maupun pengetahuan berasal dari pengalaman pakar.

Pada bagian ini, pengetahuan yang diinputkan dikodekan agar dapat diproses oleh komputer. Pakar dapat didampingi oleh seorang *knowledge engineer* untuk mempermudah dalam proses *maintanance* pengetahuan.

3.5.1.3 Mesin Inferensi

Metode penelusuran jawaban menggunakan metode *forward chaining*, dimana sistem menampilkan keseluruhan data gejala yang kemudian dari berbagai kemungkinan itu dipersempit berdasarkan inputan user. Setiap gejala yang ada dilakukan perhitungan menggunakan rumus pada metode *certainty factor* untuk mencari nilai *evidence* tunggal. Nilai *cf evidence* tunggal yang ada pada setiap *rule* kembali dihitung lagi menggunakan rumus *cf* kombinasi yang mana untuk setiap nilai *cf evidence* tunggal mendapat perlakuan sebagai nilai *cf1* dan *cf2*. Berikut ini rumus yang digunakan untuk menghitung nilai *cf evidence* tunggal maupun nilai *cf* kombinasi yang diterapkan untuk setiap premis tunggal hasil pecahan dari premis majemuk.

1. Aturan dengan *evidence* E tunggal dan hipotesis H tunggal

IF E THEN H (CF aturan)

$$CF(H,E)=CF(E)\times CF(\text{aturan}) \quad (3-1)$$

2. Kombinasi dua buah aturan dengan *evidence* berbeda (E_1 dan E_2), tetapi hipotesis sama.

IF E THEN H Aturan 1 $CF(H, E_1)=CF_1=C(E_1)\times CF(\text{Aturan1})$ (3-2)

IF E THEN H Aturan 2 $CF(H, E_2)=CF_2=C(E_2)\times CF(\text{Aturan2})$ (3-3)

$$CF(CF_1, CF_2)= \begin{cases} CF_1 + CF_2(1 - CF_1) & \text{jika } CF_1 \geq 0 \text{ dan } CF_2 \geq 0 \quad (3-4) \\ \frac{CF_1 + CF_2}{1 - \min[|CF_1|, |CF_2|]} & \text{jika } CF_1 < 0 \text{ atau } CF_2 < 0 \quad (3-5) \\ CF_1 + CF_2(1 + CF_1) & \text{jika } CF_1 \leq 0 \text{ dan } CF_2 \leq 0 \quad (3-6) \end{cases}$$

3.5.1.4 Blackboard (Daerah Kerja)

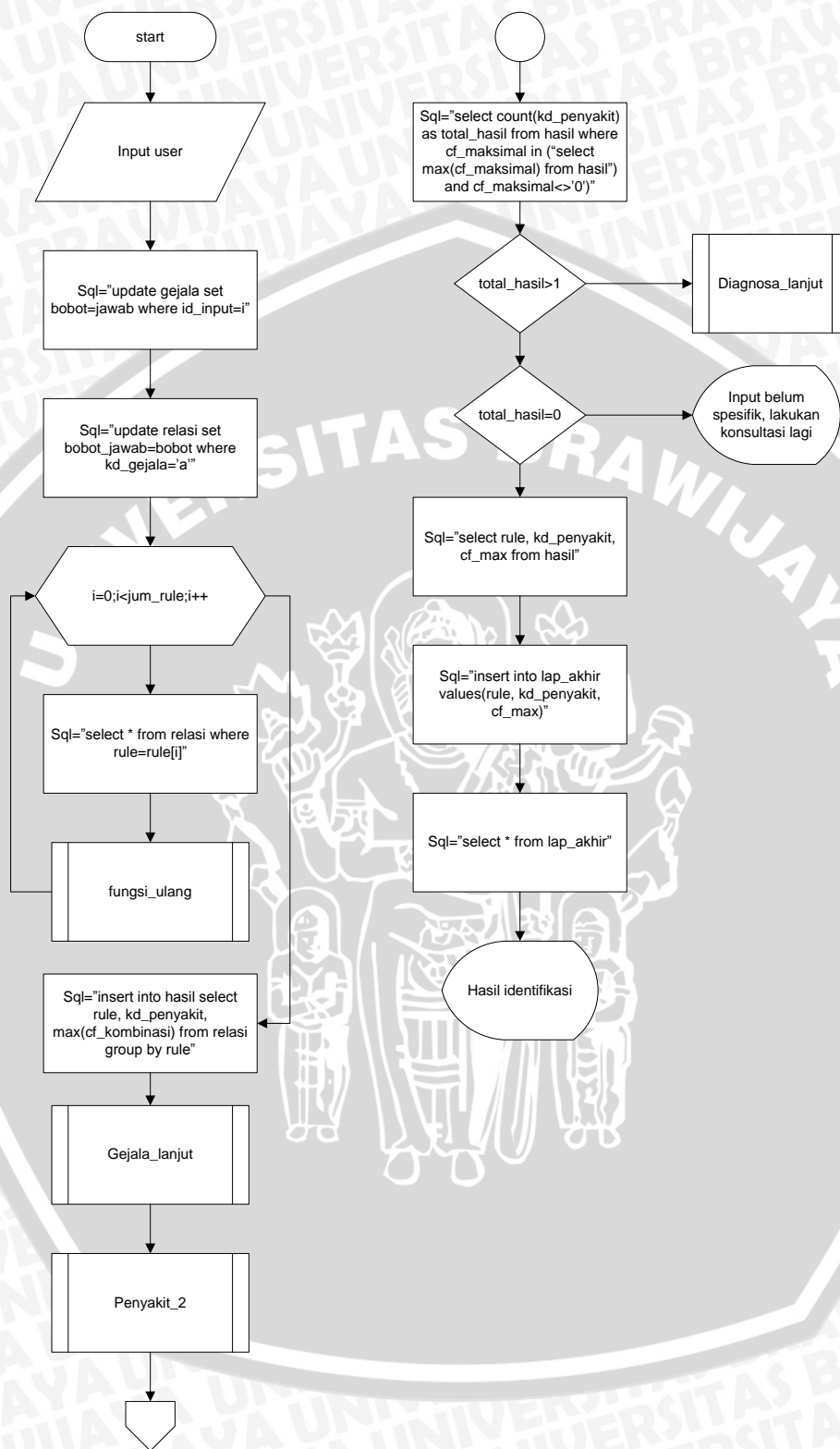
Blackboard merupakan area memori yang berfungsi sebagai basis data untuk merekam hasil sementara. *Blackboard* berisi rencana solusi yang berupa data yang digunakan sebagai bahan pertimbangan dalam memberikan kesimpulan akhir.

3.5.1.5 Fasilitas Penjelasan

Pada umumnya, fasilitas penjelasan berisi tuntunan penggunaan aplikasi sistem pakar dan bagaimana kesimpulan bisa diambil. Sistem menyediakan menu *help* untuk *user* yang masih belum mengerti bagaimana mekanisme konsultasi. Sistem juga menyediakan proses diagnosa lanjut apabila ternyata jawaban yang diberikan *user* masih belum spesifik. Diagnosa lanjut yang diberikan bertujuan memandu *user* agar dapat melakukan penambahan inputan gejala yang lebih spesifik.

3.6 Flowchart Aplikasi

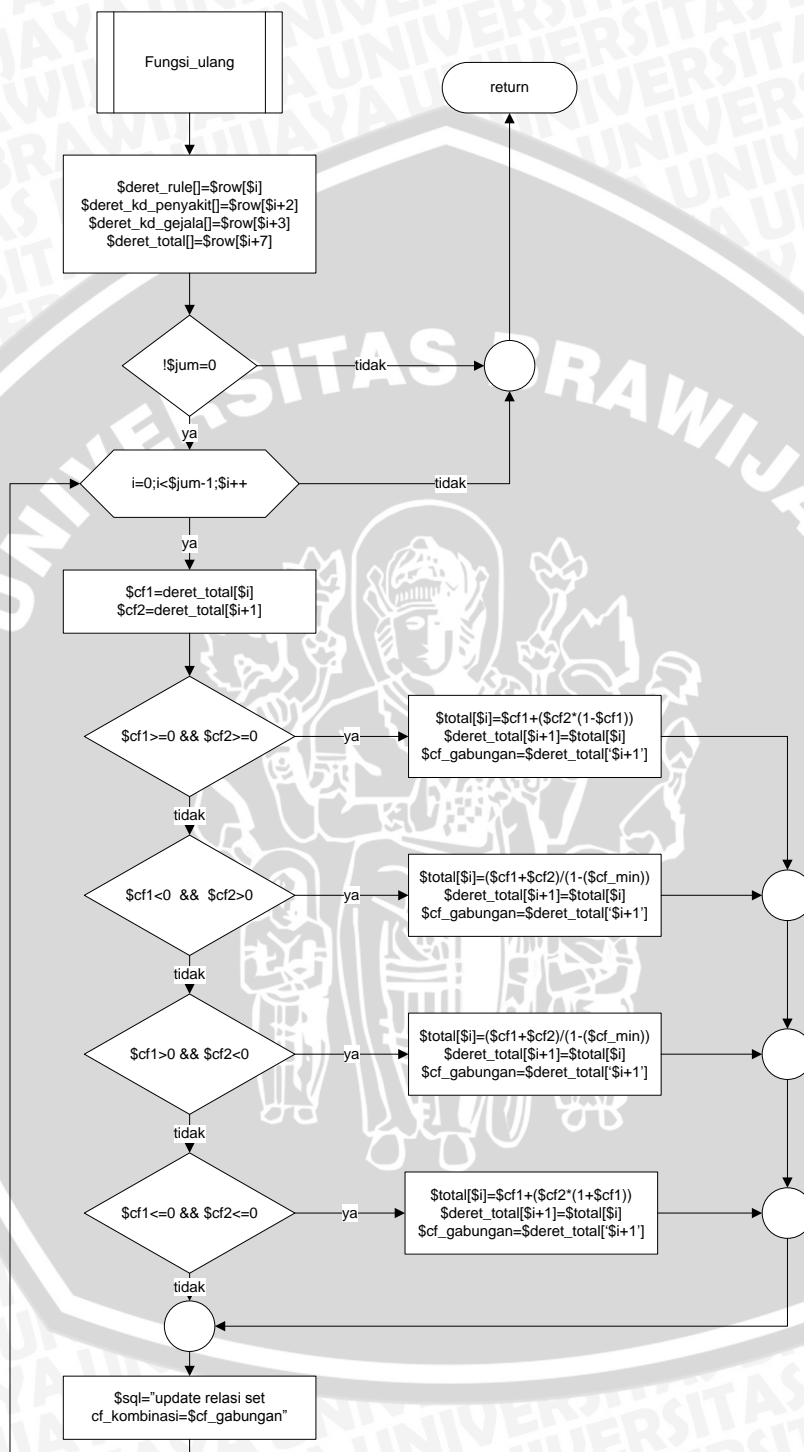
Diagram alir atau *flowchart* merupakan visualisasi dari algoritma yang diterapkan untuk memecahkan persoalan dalam sistem pakar. **Gambar 3.5** merupakan *flowchart* proses konsultasi pada sistem pakar berbasis *web* untuk identifikasi hama penyakit pada budidaya tanaman jamur menggunakan metode *certainty factor*. **Gambar 3.6** merupakan penjabaran dari fungsi ulang, dimana rumus cf kombinasi diterapkan dalam perhitungan sistem. Fungsi gejala lanjut berisi algoritma untuk menyelesaikan masalah apabila gejala berupa data hama penyakit, bila terjadi hama penyakit menjadi gejala bagi hama penyakit lain. Fungsi penyakit_2 sendiri merupakan fungsi untuk menyamakan hasil, apabila terdapat rule lain dengan gejala berbeda namun memiliki penyakit yang sama dengan rule lain. Perhitungan pada fungsi penyakit_2 menggunakan rumus cf kombinasi karena rumus ini ditujukan untuk dua rule atau lebih yang memiliki evidence berbeda satu sama lain namun memiliki hipotesa yang sama. Sedangkan untuk fungsi diagnosa_lanjut adalah apabila keputusan sementara dimiliki oleh lebih dari 1 hama penyakit dengan bobot *certainty factor* yang sama.



Gambar 3.5 Flowchart Utama Konsultasi Sistem Pakar

Sumber: Perancangan

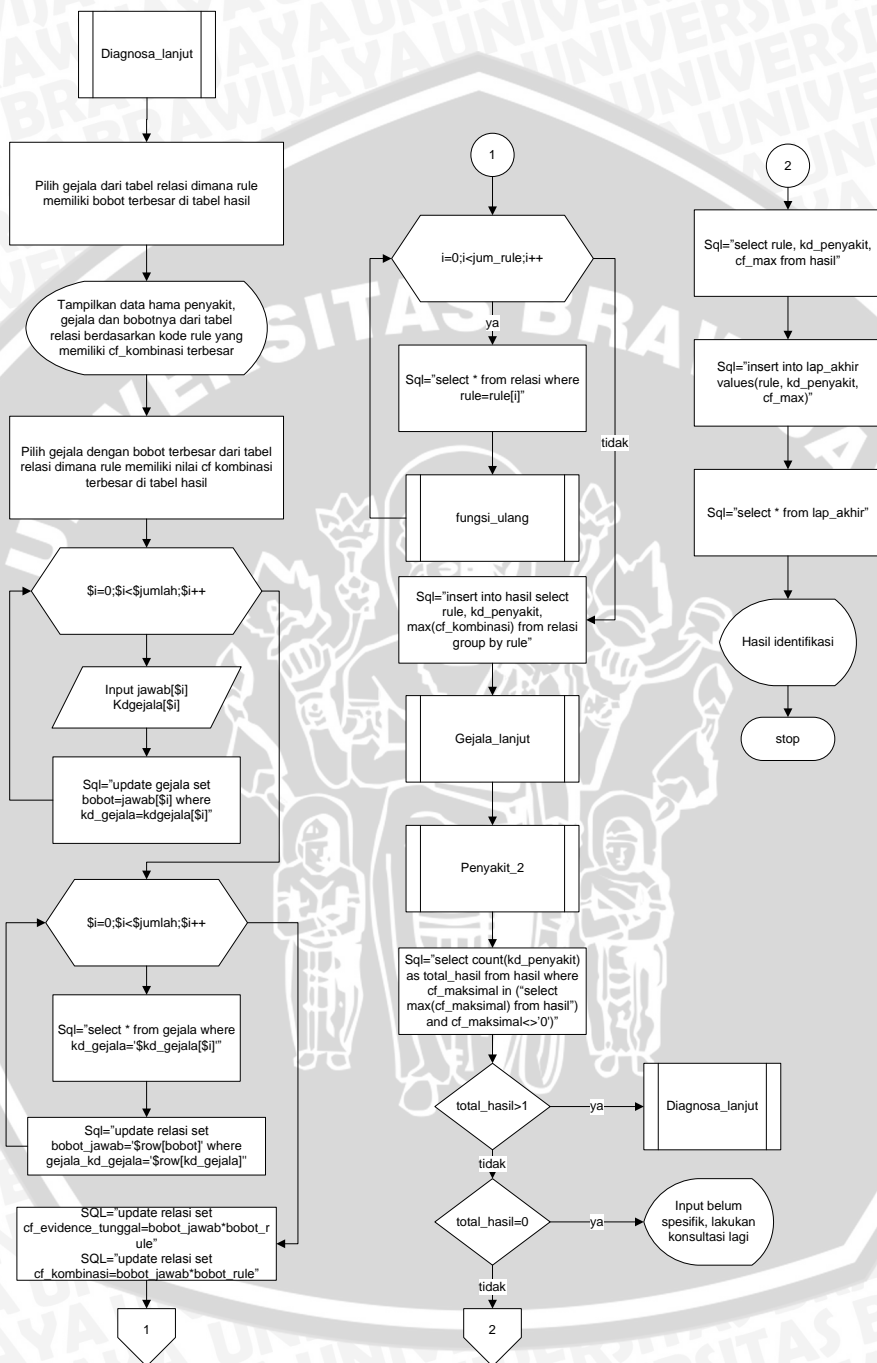
Fungsi ulang berisi alur logika perhitungan cf *evidence* tunggal hingga sampai dihasilkan nilai cf kombinasi untuk setiap *rule*.



Gambar 3.6 Flowchart Fungsi Ulang

Sumber: Perancangan

Fungsi diagnosa lanjut, berisi alur logika sistem pakar dalam mengatasi kasus dimana terdapat lebih dari 1 rule yang memiliki nilai cf kombinasi terbesar atau terdapat 2 calon keputusan akhir sistem pakar.



Gambar 3.7 Flowchart Diagnosa Lanjut

Sumber: Perancangan



Pada mulanya ketika jumlah keputusan dalam tabel hasil terdapat lebih dari satu, sistem menampilkan data keputusan hama penyakit sementara beserta gejala dan bobotnya. User diminta untuk melakukan penginputan gejala lagi, dimana pilihan yang disediakan untuk dipilih adalah data gejala yang memiliki bobot terbesar dari rule yang memiliki nilai cf terbesar. Secara garis besar, sistem menginformasikan data rule yang berhasil teridentifikasi oleh sistem beserta data gejala lengkap yang disertai bobotnya. Berikutnya user diminta untuk memilih gejala terbesar dari penyakit yang dianggap memiliki gejala yang lebih dominan dialami pada tanaman jamurnya.

3.7 Implementasi Sistem

Implementasi perangkat lunak dilakukan dengan mengacu kepada perancangan aplikasi. Implementasi perangkat lunak dilakukan dengan menggunakan PHP, MySQL dan tools pendukung lainnya. Implementasi sistem meliputi:

- Pembuatan antarmuka pengguna berupa halaman-halaman web.
- Memasukkan data penelitian ke *database* MySQL untuk diolah hingga mendapatkan referensi solusi
- Penerapan metode *Certainty Factor* dalam program yang dibuat dengan bahasa PHP.

3.8 Pengujian Sistem

Pada tahap ini dilakukan pengujian keberhasilan dan akurasi sistem yang telah dibuat pada tahap implementasi. Pengujian dilakukan dengan mengambil nilai rata-rata dari jumlah prosentase maksimal dari setiap *rule* yang diujicobakan dan dibandingkan dengan rata-rata dari hasil pembobotan pakar. Pengujian *black box* juga dilakukan, hal ini sebagai indikator keberhasilan pada setiap fungsionalitas yang ada pada sistem pakar.

3.9 Pengambilan Kesimpulan

Kesimpulan mengenai semua tahapan yang telah dilalui serta saran yang berkenaan dengan hasil yang telah dicapai. Kesimpulan diambil dari tahap

perancangan hingga analisa dan pengujian sistem. Saran berguna untuk memperbaiki kesalahan untuk pengembangan lebih lanjut.

