

BAB V IMPLEMENTASI

Pada bab ini dibahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah diperoleh dari analisis kebutuhan dan proses perancangan perangkat lunak yang dibuat. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi algoritma pada program mesin inferensi, dan implementasi antarmuka beserta fungsi proseduralnya.

5.1 Spesifikasi Sistem

Hasil analisis kebutuhan dan perancangan perangkat lunak yang telah diuraikan pada bab perancangan menjadi acuan untuk melakukan implementasi menjadi sistem yang dapat berfungsi sesuai dengan kebutuhannya. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Pembangunan sistem pakar identifikasi hama dan penyakit pada tanaman jamur ini menggunakan spesifikasi perangkat keras yang terdapat pada **Tabel 5.1**.

Tabel 5.1 Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
Prosesor	Intel(R) Core(TM) i5 CPU M 460 @ 2.53GHz (4 CPUs), ~2.5GHz
Memori(RAM)	2048MB DDR3
Hardisk	640 GB HDD
Kartu Grafis	ATI Mobility Radeon HD 5650 Up to 1655 MB HyperMemory (1GB)
Monitor	Monitor 14.0" LED LCD

Sumber: Implementasi

5.1.2 Spesifikasi Perangkat Lunak

Pembangunan sistem pakar identifikasi hama dan penyakit pada tanaman jamur menggunakan perangkat lunak dengan spesifikasi yang dijelaskan pada **Tabel 5.2**.

Tabel 5.2 Spesifikasi Perangkat Lunak Komputer

Sistem Operasi	Windows 7 ROG 32-bit (6.1, Build 7600) (7600.win7_gdr.091207-1941)
Bahasa Pemrograman	PHP
Tools pemrograman	Adobe Dreamweaver CS3
Web Server	Apache 2.2.17
Web Browser	Google Chrome
DBMS	MySQL

Sumber: Implementasi

5.2 Batasan-Batasan Implementasi

Beberapa batasan dalam mengimplementasikan sistem adalah sebagai berikut:

- Input yang diterima oleh sistem berupa jawaban *user* terhadap gejala yang cocok atau sesuai dengan yang dialami oleh tanaman.
- Jawaban *user* terdiri dari 3 pilihan jawaban yang memiliki bobot yang bervariasi, antara lain: ya (bernilai 1), tidak (bernilai 0), mungkin (bernilai 0.4).
- Bobot CF dari hasil wawancara pakar diberikan pada setiap gejala.
- Output yang diterima *user* berupa hasil keputusan hama atau penyakit yang menyerang tanaman jamurnya disertai dengan prosentase keyakinan CF terhadap serangan itu beserta cara pencegahan dan penanggulangannya.
- Versi XAMPP yang digunakan adalah versi 1.7.4.
- *Web browser* yang digunakan selama pengimplementasian adalah Google Chrome.

- *Database* sistem atau basis pengetahuan disimpan dalam DBMS MySQL.
- Metode yang digunakan dalam pengambilan keputusan yaitu *Certainty Factor*.
- Sistem pakar identifikasi hama dan penyakit pada tanaman jamur bersifat *rule base system*.
- Sistem ini bersifat dinamis, yaitu menyediakan fasilitas *insert* dan *update*.
- Logika yang digunakan untuk hubungan antara satu gejala dengan gejala yang lain dalam satu *rule* penyakit tertentu menggunakan logika *AND*.

5.3 Implementasi Algoritma

Sistem pakar mengadaptasi kaidah penelusuran dan pengambilan keputusan berdasarkan pada algoritma *Certainty Factor*.

5.3.1 Implementasi Algoritma CF Untuk Penelusuran Bobot CF Setiap Rule

Dalam *Certainty Factor*, setiap hasil gejala dalam setiap *rule* pada suatu penyakit tertentu harus memiliki bobot. Bobot CF berkisar antara -1 sampai dengan 1. Sistem mendapatkan bobot untuk setiap gejala dari hasil inputan nilai keyakinan *user* terhadap gejala yang dialami. Sistem membatasi pada tiga pilihan jawaban, diantaranya: ya, tidak, dan mungkin. Masing-masing pilihan memiliki bobot yang berbeda-beda. Semakin *user* yakin, maka tingkat keyakinannya pun semakin tinggi.

```
1. for ($i=0;$i<$jumlah;$i++){
2. //id nomor soal
3. $nomor=$id_soal[$i];
4. //jika user tidak memilih jawaban
5. if (empty($pilihan[$nomor]))
6. {$query=mysql_query("update gejala set bobot='0' where
7. id_input='$nomor'");}
```

```
8. else
9. { //jawaban dari user
10.     $jawaban=$pilihan[$nomor];
11.     //update bobot gejala dengan id input yang
12.     bersesuaian pada database
13.     $query=mysql_query("update gejala set
14.     bobot='$jawaban' where id_input='$nomor'"); } }
15.     //select bobot pada tabel gejala dan mengupdate
16.     bobot tabel relasi yang memiliki kode gejala yang
17.     sama dengan kode gejala pada tabel gejala
18.     $query1=mysql_query("select * from gejala");
19.     while($row =mysql_fetch_array($query1))
20.     {
21.         $sql=mysql_query("update relasi set
22.         bobot_jawab='$row[bobot]'
23.         where gejala_kd_gejala='$row[kd_gejala]'");
24.     }
25.     //update tabel relasi nilai cf_evidence_tunggal tiap
26.     Gejala untuk semua rule  $CF(H|E) = CF(E) \times CF(R)$ 
27.     $sql_1=mysql_query("update relasi set
28.     cf_evidence_tunggal=bobot_jawab*bobot_rule");
```

Gambar 5.1 Implementasi Algoritma *Certainty Factor* Dalam Penelusuran Bobot CF Setiap *Rule*

Sumber: Implementasi

Penjelasan implementasi algoritma *Certainty Factor* untuk mendapatkan bobot CF di setiap *rule* ditunjukkan pada Gambar 5.1 yaitu:

1. Baris 1-7 merupakan perintah untuk melakukan proses *update* bobot pada tabel gejala menjadi bernilai 0 (tidak) untuk gejala yang tidak dipilih sama sekali oleh user (default 'Tidak').
2. Baris 8-14 merupakan perintah untuk melakukan *update* bobot pada tabel gejala menjadi bernilai tertentu (nilai 1 atau 0.4) sesuai dengan gejala yang dipilih oleh *user*.
3. Baris 15-24 merupakan perintah untuk menyesuaikan bobot yang dimiliki oleh setiap gejala pada tabel relasi sehingga bersesuaian dengan isi bobot pada tabel gejala.

4. Baris 25-28 merupakan perintah untuk menghitung nilai cf evidence tunggal pada setiap gejala (pecahan premis majemuk).

5.3.2 Implementasi Algoritma *Certainty Factor* Untuk Pengambilan Keputusan

Setelah pada setiap premis tunggal dihitung nilai CF evidence tunggalnya, berikutnya setiap nilai cf evidence tunggal untuk setiap rule dihitung dengan nilai cf kombinasi.

```
1. $sql_3=mysql_query("select distinct rule from relasi");
2. $jum_rule=mysql_num_rows($sql_3);
3. $i=0;
4. while($row2=mysql_fetch_array($sql_3))
5. {
6. $deret_rule[]=$row2[$i];
7. }
8. function fungsi_ulang($sql_4)
9. {
10. $jum=mysql_num_rows($sql_4);
11. $i=0;
12. while($row1=mysql_fetch_array($sql_4))
13. {
14. $deret_rule1[]=$row1[$i];
15. $deret_kd_penyakit1[]=$row1[$i+2];
16. $deret_kd_gejala1[]=$row1[$i+3];
17. $deret_total1[]=$row1[$i+7];
18. }
19. if(!$jum==0){
20. for($i=0;$i<$jum-1;$i++){
21. $cf1=$deret_total1[$i];
22. $cf2=$deret_total1[$i+1];
```

```
23. if($cf1>=0 && $cf2>=0)
24. {$total[$i]=$cf1+($cf2*(1-$cf1));
25. $deret_total1[$i+1]=$total[$i];
26. $cf_gabungan=$deret_total1[$i+1];}
27. else if($cf1<0 && $cf2>0)
28. {$cf2abs=$cf2;
29. $cf1abs=abs($cf1);
30. $cf_min=min($cf1abs,$cf2abs);
31. $total[$i]=($cf1+$cf2)/(1-($cf_min));
32. $deret_total1[$i+1]=$total[$i];
33. $cf_gabungan=$deret_total1[$i+1];}
34. else if($cf1>0 && $cf2<0)
35. {$cf1abs=$cf1;
36. $cf2abs=abs($cf2);
37. $cf_min=min($cf1abs,$cf2abs);
38. $total[$i]=($cf1+$cf2)/(1-($cf_min));
39. $deret_total1[$i+1]=$total[$i];
40. $cf_gabungan=$deret_total1[$i+1];}
41. else if($cf1<=0 && $cf2<=0)
42. {$total[$i]=$cf1+($cf2*(1+$cf1));
43. $deret_total1[$i+1]=$total[$i];
44. $cf_gabungan=$deret_total1[$i+1];}
45. $kd_gejala[$i]=$deret_kd_gejala1[$i+1];
46. $query_6=mysql_query("update relasi set
47. cf_kombinasi='$cf_gabungan' where rule='$deret_rule1[$i]' and
48. gejala_kd_gejala='$kd_gejala[$i]'");
49. }}}
50. for($i=0;$i<$jum_rule;$i++){
51. $sql_4=mysql_query("select * from relasi where
52. rule='$deret_rule[$i]'");
53. fungsi_ulang($sql_4);}
```

Gambar 5.2 Implementasi Algoritma *Certainty Factor* Dalam Pengambilan Keputusan

Sumber: Implementasi

Penjelasan algoritma *Certainty Factor* untuk pengambilan keputusan hama dan penyakit pada Gambar 5.2, yaitu:

1. Baris 1-7 merupakan perintah untuk mendapatkan keseluruhan nilai yang terdapat pada tabel relasi.
2. Baris 8 merupakan pendeklarasian fungsi yang bernama fungsi_ulang.
3. Baris 10 merupakan perintah untuk mendapatkan jumlah baris yang ada di tabel relasi.
4. Baris 12-18 merupakan perintah untuk mendapatkan nilai per kolom dari tabel relasi.
5. Baris 21-22 merupakan perintah untuk mendapatkan nilai dari kolom yang berisi nilai cf evidence tunggal untuk setiap gejala, untuk setiap nilai pada baris 1 dianggap sebagai nilai cf1 dan nilai pada baris berikutnya dianggap sebagai nilai cf2.
6. Baris 23-26 merupakan perintah untuk menghitung nilai cf kombinasi apabila ditemukan nilai $cf1 \geq 0$ dan nilai $cf2 \geq 0$.
7. Baris 27-33 merupakan perintah untuk menghitung nilai cf kombinasi apabila ditemukan nilai $cf1 < 0$ dan nilai $cf2 > 0$.
8. Baris 34-40 merupakan perintah untuk mendapatkan nilai cf kombinasi apabila diketemukan nilai $cf1 > 0$ dan nilai $cf2 < 0$.
9. Baris 41-44 merupakan perintah untuk mendapatkan nilai cf kombinasi apabila diketemukan nilai $cf1 \leq 0$ dan nilai $cf2 \leq 0$.
10. Baris 45 merupakan perintah untuk menetapkan nilai dari baris ke-1 sebagai nilai cf kombinasi pada baris ke-2.
11. Baris 46-48 merupakan perintah untuk melakukan update nilai cf kombinasi pada tabel relasi.

12. Baris 50-53 merupakan perintah untuk memanggil fungsi ulang dengan rule tertentu.

5.4 Implementasi Antar Muka

5.4.1 Tampilan Halaman Utama Website

Halaman sistem pakar menjadi salah satu bagian menu dalam suatu *website* yang mengulas masalah budidaya jamur. Tampilan halaman utama diperlihatkan pada **Gambar 5.3**.

Pada halaman utama *website*, *user* dihadapkan pada tampilan utama berupa suatu *website* dengan isi artikel yang membahas masalah budidaya jamur. Untuk memasuki program sistem pakar, *user* harus memilih salah satu menu yang ada pada *website*, dengan nama menu Sistem Pakar.



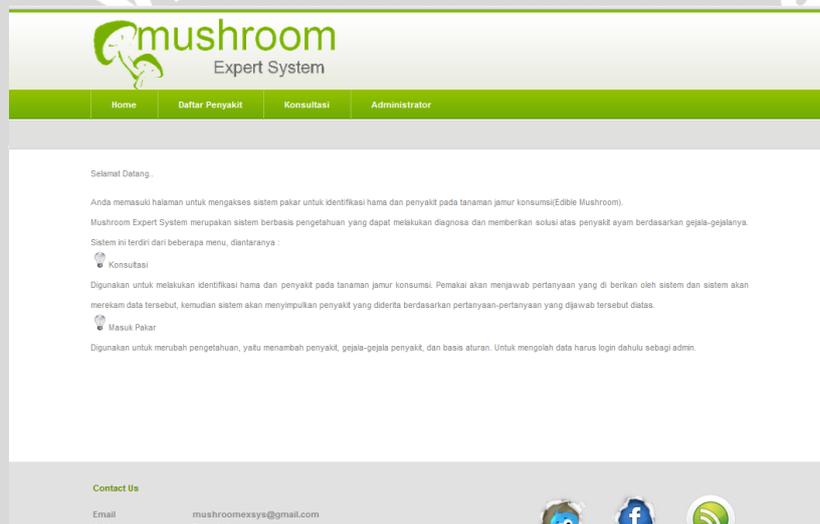
Gambar 5.3 Tampilan Halaman Utama Website

Sumber: Implementasi

5.4.2 Tampilan Halaman Utama Sistem Pakar

Pada halaman memasuki sistem pakar, *user* dihadapkan pada pilihan menu, diantaranya *Home* untuk kembali ke halaman utama *website*, Daftar

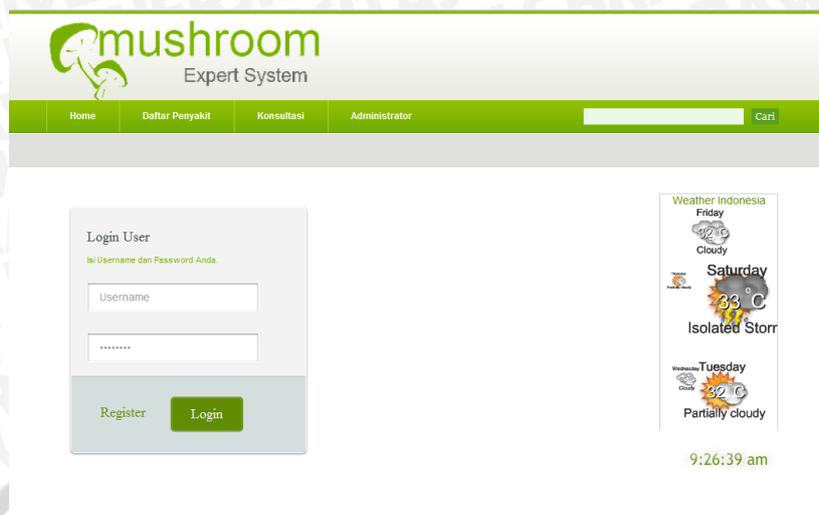
Penyakit untuk menampilkan daftar penyakit berdasarkan *rul*nya, sehingga *user* maupun pakar yang belum *login* pun dapat melihat kelengkapan data hama dan penyakit beserta gejala-gejala yang menyertai serangan. Halaman utama sistem pakar secara garis besar dibagi menjadi dua otoritas utama. Halaman konsultasi yang digunakan untuk *user* dapat berkonsultasi, menjawab pertanyaan yang diberikan oleh sistem dan kemudian *user* mendapatkan hasil prosentase, solusi berupa cara penanggulangan dan cara pencegahannya. Halaman pakar yang sebenarnya dapat digunakan baik oleh pakar. Dalam halaman pakar terdapat menu-menu yang digunakan untuk dapat melakukan proses *update* maupun *insert rule*, penyakit hama ataupun gejala.



Gambar 5.4 Tampilan Halaman Utama Sistem Pakar
Sumber: Implementasi

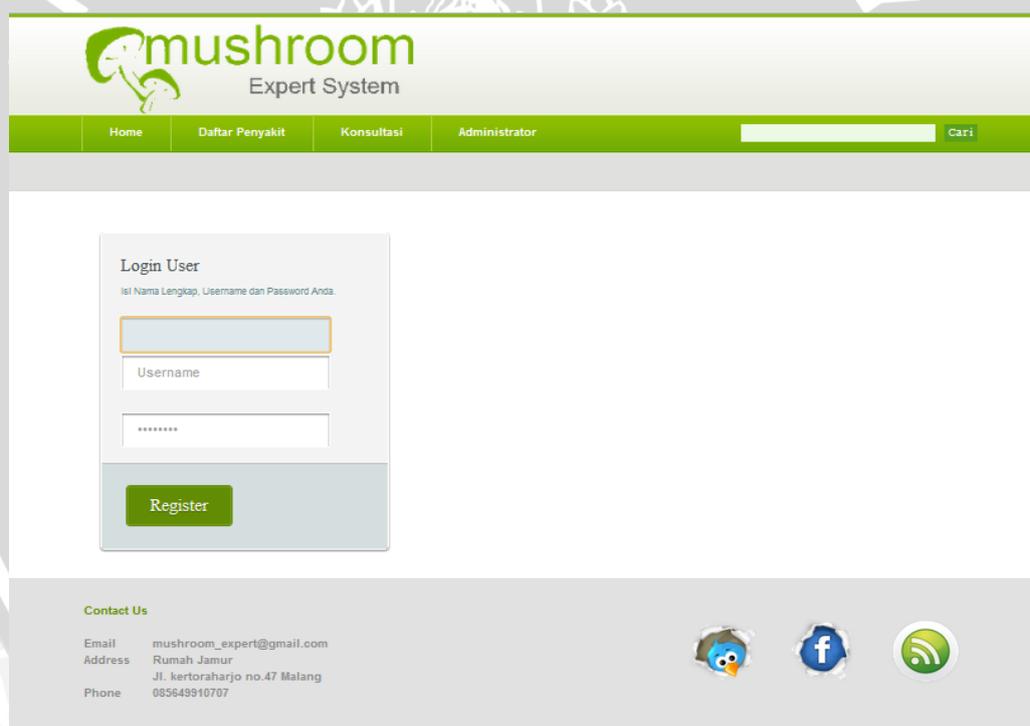
5.4.3 Tampilan Halaman *Login User*

Sebelum *user* dapat melakukan proses konsultasi, *user* diharuskan untuk melakukan *login*, dapat juga untuk pengguna baru melakukan proses registrasi terlebih dahulu sebelum dapat melakukan proses *login* dan kemudian melakukan konsultasi.



Gambar 5.5 Tampilan Halaman *Login User*

Sumber: Implementasi



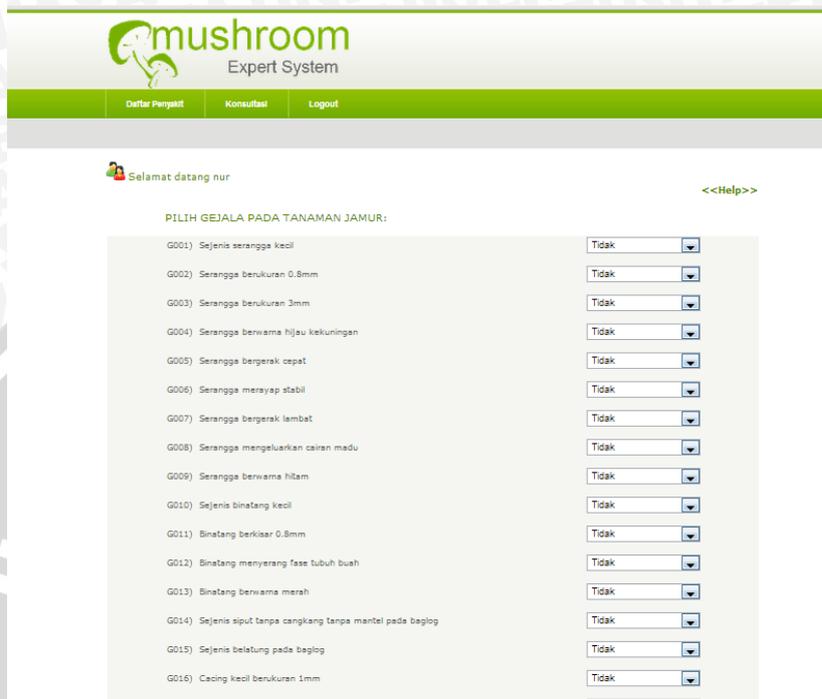
Gambar 5.6 Tampilan Halaman Registrasi *User Baru*

Sumber: Implementasi

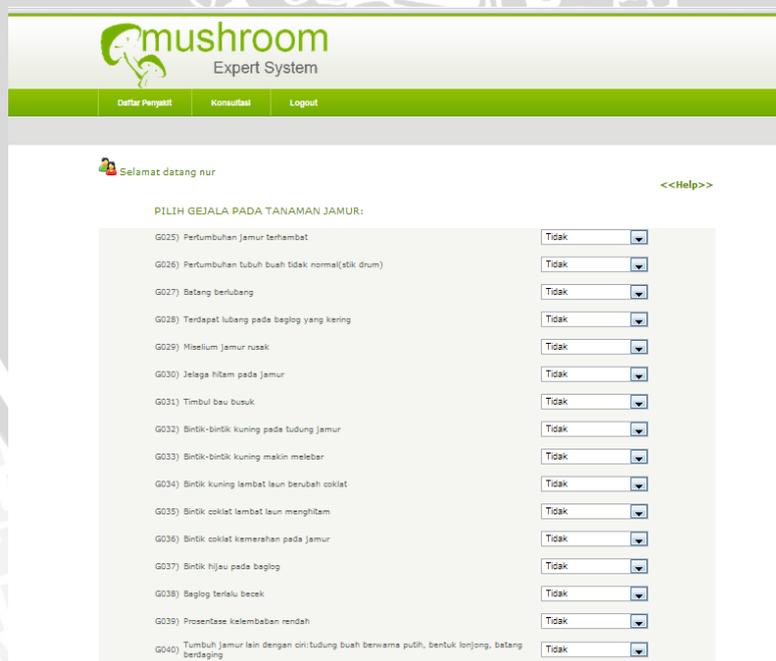
5.4.4 Tampilan Halaman Konsultasi

Selesai berhasil melakukan *login*, berikutnya *user* dihadapkan pada tampilan menu *form* konsultasi yang terdiri dari tampilan seluruh gejala serangan hama penyakit pada tanaman jamur disertai pilihan jawaban yang masing-masing

memiliki bobot yang berbeda-beda. Tampilan halaman konsultasi ditunjukkan pada **Gambar 5.7**.



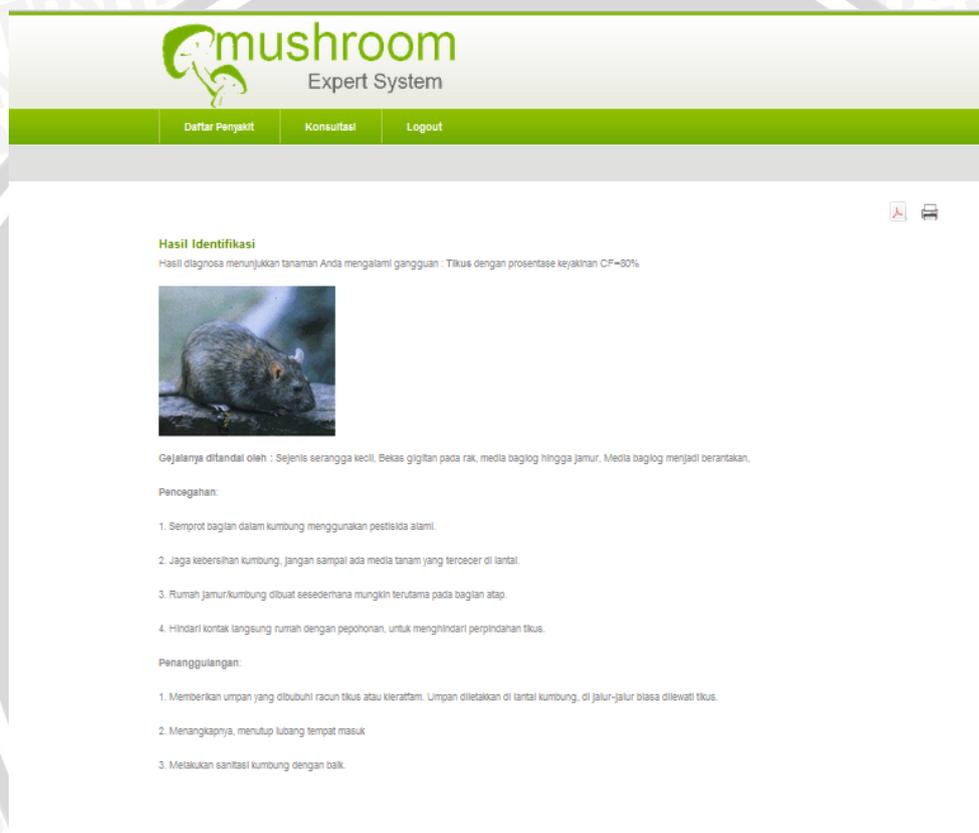
Gambar 5.7 Tampilan Halaman Konsultasi 1
Sumber: Implementasi



Gambar 5.8 Tampilan Halaman Konsultasi 2
Sumber: Implementasi

5.4.5 Tampilan Halaman Hasil Konsultasi

Setelah *user* melakukan proses input data, berikutnya *user* dapat mengetahui hasil dari proses konsultasinya. Pada halaman hasil konsultasi, *user* diberikan hasil nama hama atau penyakit yang menyerang, gambar untuk meyakinkan *user* terhadap hama yang menyerang tanamannya dan prosentase keyakinan serangan. Pada hasil konsultasi *user* juga diberikan solusi, atau cara pencegahan dan penanggulangannya.

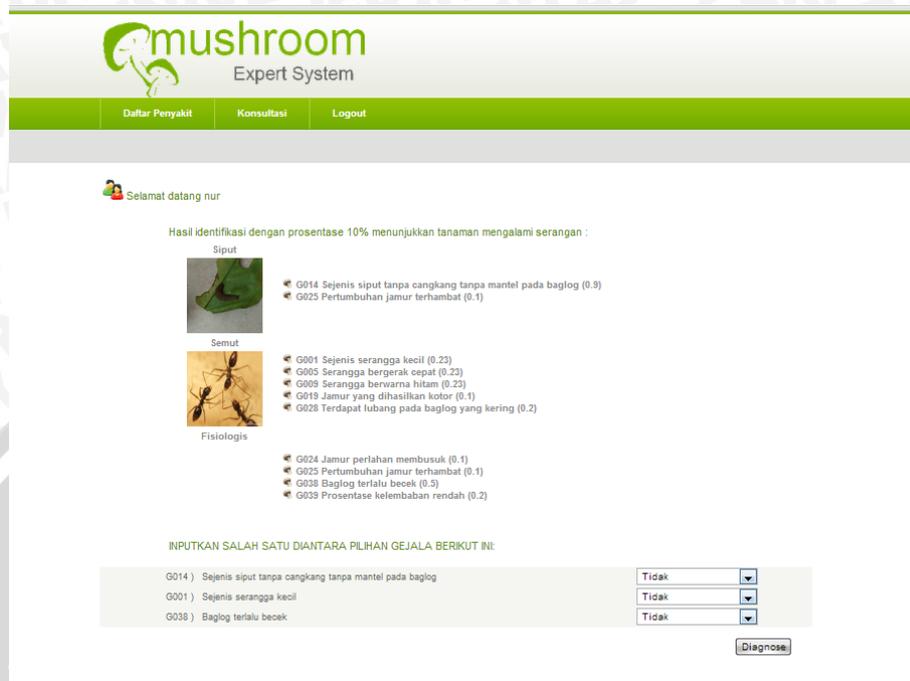


Gambar 5.9 Tampilan Halaman Hasil Konsultasi

Sumber: Implementasi

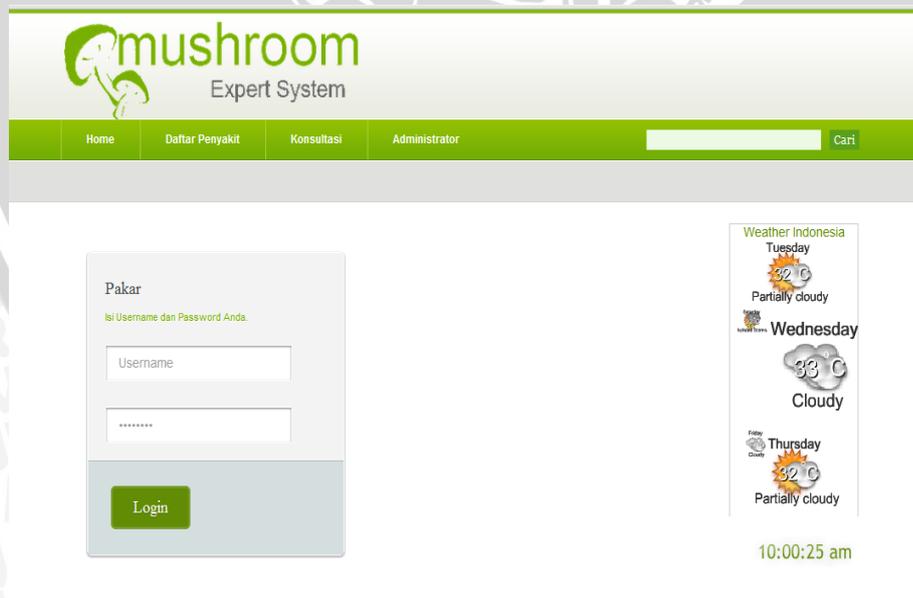
5.4.6 Tampilan Halaman Untuk Identifikasi Lanjut

Identifikasi lanjut digunakan apabila hasil sementara yang diperoleh dimiliki oleh lebih dari 1 hama penyakit sehingga diperlukan proses identifikasi untuk hasil yang lebih spesifik. Tampilan halaman identifikasi lanjut dapat dilihat pada **Gambar 5.10**.



Gambar 5.10 Tampilan Halaman Identifikasi Lanjut
Sumber: Implementasi

5.4.7 Tampilan Halaman *Login Pakar*

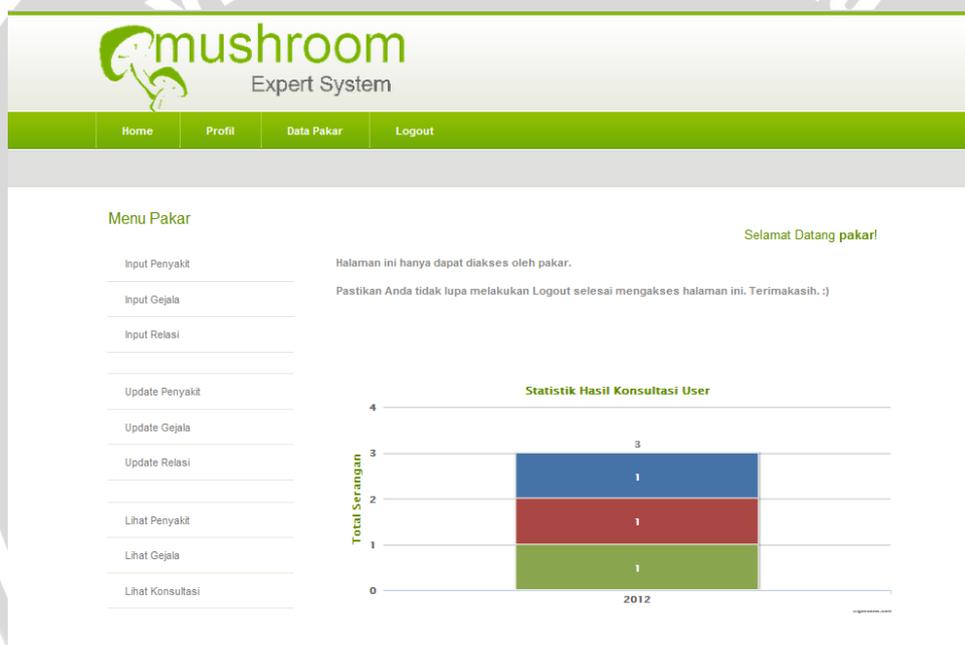


Gambar 5.11 Tampilan Halaman Data *Account*
Sumber: Implementasi

Sebelum pakar dapat mengakses halaman yang menjadi otoritasnya, pakar diharuskan melakukan *login* terlebih dahulu. Halaman *login* pakar ditampilkan pada **Gambar 5.11**.

5.4.8 Tampilan Halaman Utama Pakar

Selesai pakar sukses melakukan *login*, berikutnya pakar dihadapkan pada halaman utama pakar, di halaman tersebut terdapat beberapa pilihan menu yang bisa diakses, diantaranya: Profil, Data Pakar, Input Penyakit, Input Gejala, Input Relasi, *Update* Gejala, *Update* Penyakit, *Update* Relasi, Lihat Penyakit, Lihat Gejala dan Lihat Konsultasi.



Gambar 5.12 Tampilan Halaman Utama Pakar
Sumber: Implementasi

5.4.9 Tampilan Halaman Input Gejala

Halaman input gejala digunakan untuk pakar menambahkan gejala baru pada basis pengetahuan.

The screenshot displays the 'mushroom Expert System' interface. At the top, there is a navigation bar with 'Home', 'Profil', 'Data Pakar', and 'Logout' links. Below this is a 'Menu Pakar' sidebar with options like 'Input Penyakit', 'Input Gejala', 'Input Relasi', 'Update Penyakit', 'Update Gejala', 'Update Relasi', 'Lihat Penyakit', 'Lihat Gejala', and 'Lihat Konsultasi'. The main content area is titled 'INPUT GEJALA' and contains a form with the following fields: 'Kode Gejala' (with value 'G047'), 'Isi Gejala' (a large text area), and an 'Input' button. A user notification in the top right corner reads 'Anda telah login sebagai pakar'.

Gambar 5.13 Tampilan Halaman Input Gejala Baru
Sumber: Implementasi

5.4.10 Tampilan Halaman Input Penyakit Baru

Halaman input penyakit digunakan untuk menambah penyakit atau hama baru pada tanaman jamur. Tampilannya ditunjukkan pada **Gambar 5.14**.

The screenshot displays the 'mushroom Expert System' interface for adding a new disease. The navigation bar and 'Menu Pakar' sidebar are identical to the previous screenshot. The main content area is titled 'INPUT PENYAKIT' and contains a form with the following fields: 'Kode Penyakit' (with value 'P016'), 'Nama Penyakit' (a text input field), 'Gambar' (a file upload button labeled 'Choose File' with the text 'No file chosen'), 'Keterangan' (a large text area), and 'Solusi' (a large text area). An 'Input' button is located at the bottom of the form. A user notification in the top right corner reads 'Anda telah login sebagai pakar'.

Gambar 5.14 Tampilan Halaman Input Penyakit Baru
Sumber: Implementasi

5.4.11 Tampilan Halaman Input Aturan Baru

Penyakit baru yang belum memiliki *rule* belum dapat menjadi hasil dalam penelusuran gejala serangan penyakit atau hama. Halaman input *rule*, berfungsi untuk membuat *rule* baru, yang isinya bisa terdiri dari penyakit baru ataupun lama, yang jelas *rule* yang baru memiliki aturan baru, tidak menyamai aturan-aturan yang telah ada sebelumnya.

Aturan: R016
Penyakit: -Pilih Penyakit-
CF rule: (isi untuk inputan gejala berupa penyakit)
-Pilihan Nilai-
Pilihan Gejala:
G001 Sejenis serangga kecil -Pilihan Nilai-
G002 Serangga berukuran 0.8mm -Pilihan Nilai-
G003 Serangga berukuran 3mm -Pilihan Nilai-
G004 Serangga berwarna hijau kekuningan -Pilihan Nilai-
G005 Serangga bergerak cepat -Pilihan Nilai-
G006 Serangga merayap stabil -Pilihan Nilai-
G007 Serangga bergerak lambat -Pilihan Nilai-
G008 Serangga mengeluarkan cairan madu -Pilihan Nilai-
G009 Serangga berwarna hitam -Pilihan Nilai-
G010 Sejenis binatang kecil -Pilihan Nilai-
G011 Binatang berkisar 0.8mm -Pilihan Nilai-
G012 Binatang menyerang fase tubuh buah -Pilihan Nilai-

Gambar 5.15 Tampilan Halaman Input Aturan Baru
Sumber: Implementasi

5.4.12 Tampilan Halaman Update Gejala

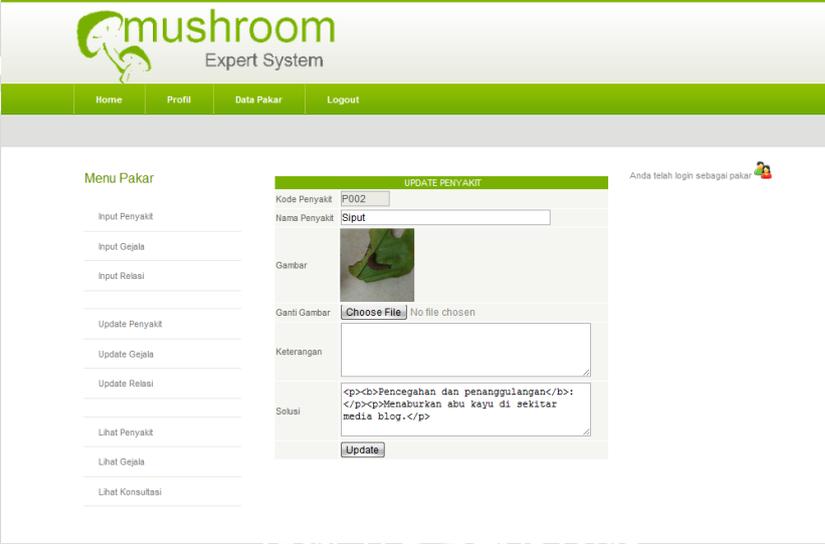
Kode: G007
Gejala: Serangga bergerak lambat
Update

Gambar 5.16 Tampilan Halaman Update Gejala
Sumber: Implementasi

Halaman *update* gejala digunakan untuk melakukan perubahan gejala isi gejala baik hanya untuk digunakan sebagai perbaikan isi gejala ataupun untuk memperbaiki kesalahan penulisan isi gejala. Sedangkan untuk penomoran gejala tidak dapat dilakukan perubahan karena berhubungan dengan gejala yang telah ada pada tabel aturan.

5.4.13 Tampilan Halaman *Update Penyakit*

Halaman *update* penyakit digunakan untuk melakukan perubahan terhadap data penyakit yang meliputi data nama penyakit, gambar penyakit atau hama, keterangan atau definisi penyakit hama, dan solusinya.



The screenshot displays the 'UPDATE PENYAKIT' form within the 'mushroom Expert System' interface. The form includes the following fields and content:

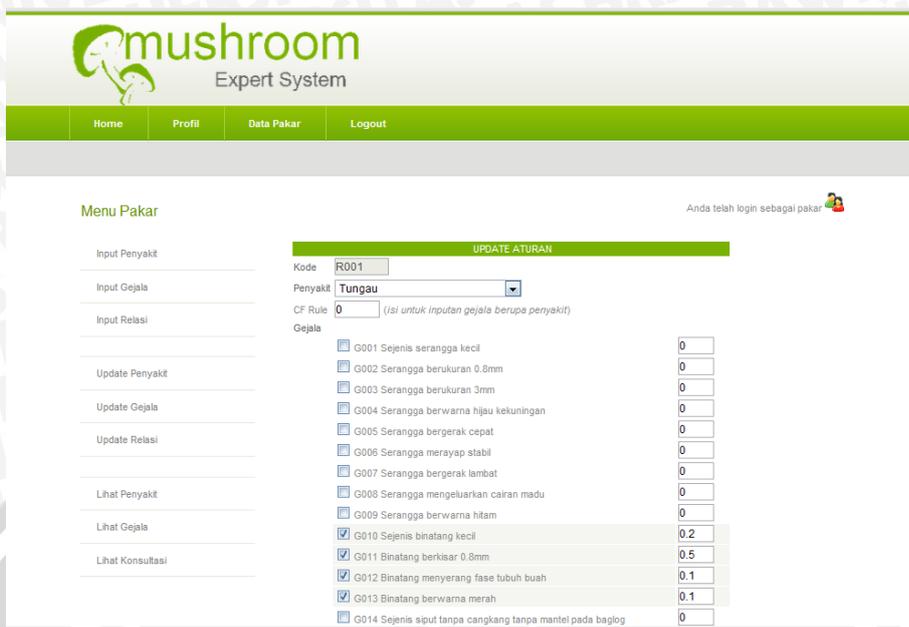
- Kode Penyakit:** P002
- Nama Penyakit:** Siput
- Gambar:** A small image of a green leaf with a white spot.
- Ganti Gambar:** A 'Choose File' button with the text 'No file chosen'.
- Keterangan:** A text area containing the text: '<p>Pencegahan dan penanggulangan:</p><p>Menaburkan abu kayu di sekitar medis blog.</p>'
- Solusi:** A text area containing the text: '<p>Pencegahan dan penanggulangan:</p><p>Menaburkan abu kayu di sekitar medis blog.</p>'
- Buttons:** An 'Update' button at the bottom of the solution field.

The left sidebar shows a 'Menu Pakar' with various navigation options like 'Input Penyakit', 'Input Gejala', 'Input Relasi', 'Update Penyakit', 'Update Gejala', 'Update Relasi', 'Lihat Penyakit', 'Lihat Gejala', and 'Lihat Konsultasi'. The top navigation bar includes 'Home', 'Profil', 'Data Pakar', and 'Logout'. The user is logged in as 'pakar'.

Gambar 5.17 Tampilan Halaman *Update* Hama Penyakit
Sumber: Implementasi

5.4.14 Tampilan Halaman *Update Aturan*

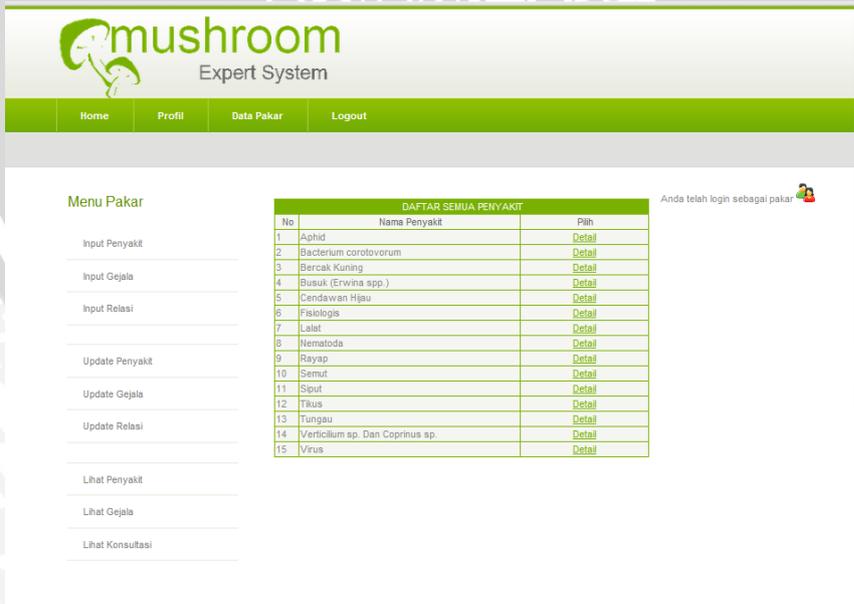
Halaman *update* aturan digunakan ketika pakar ingin melakukan perubahan data yang berhubungan dengan aturan. Data yang dapat dilakukan perubahan pada aturan, meliputi: data penyakit, gejala, maupun nilai bobot CF aturan.



Gambar 5.18 Tampilan Halaman *Update Aturan*
Sumber: Implementasi

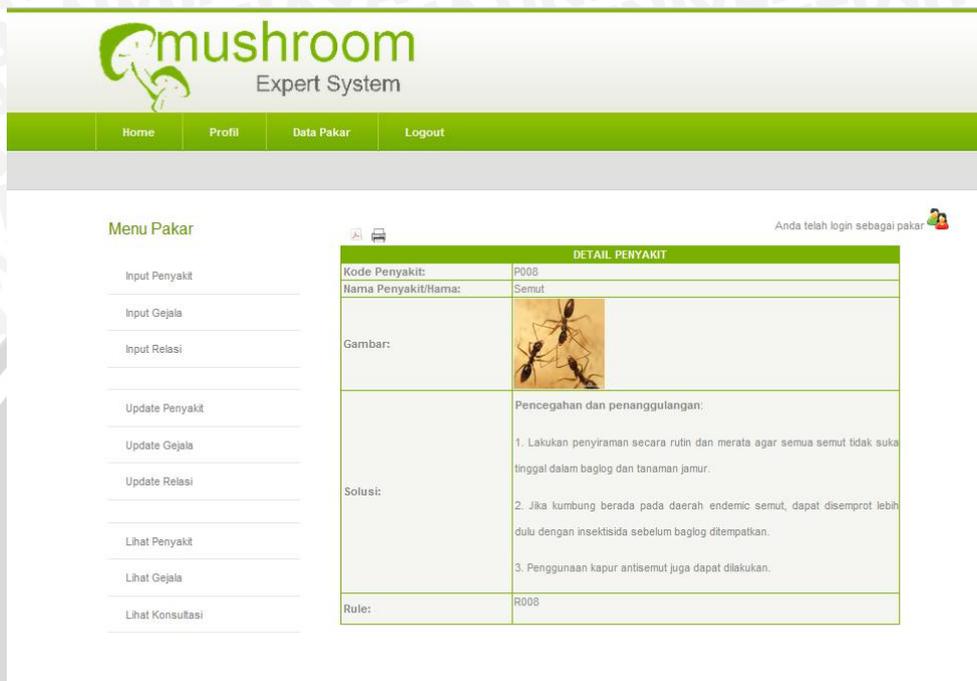
5.4.15 Tampilan Halaman Lihat Penyakit

Halaman ini digunakan untuk melihat data lengkap mengenai penyakit yang ada dalam basis pengetahuan. Data yang diperlihatkan meliputi: data kode hama penyakitnya, nama hama penyakit, gambar, solusi, dan *rule* apa saja yang melibatkan penyakit ini.



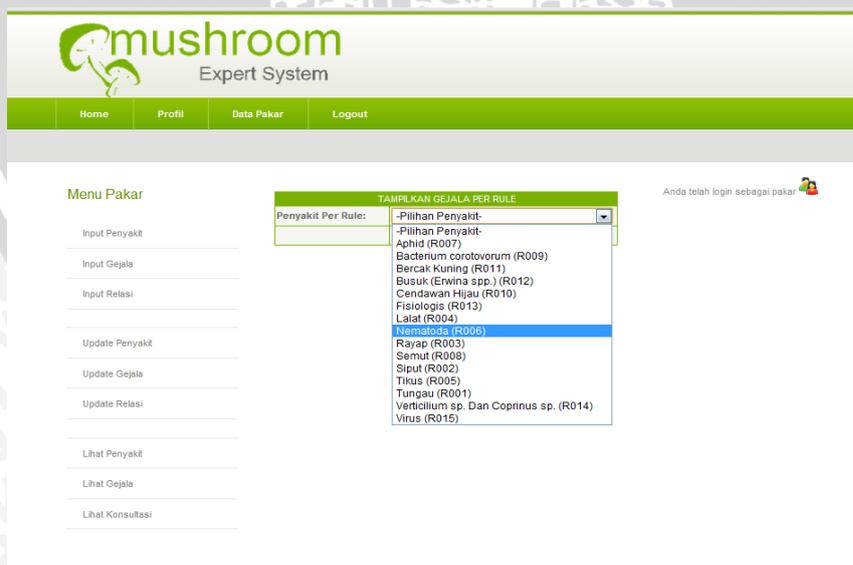
Gambar 5.19 Tampilan Halaman Lihat Penyakit
Sumber: Implementasi

Pada mulanya ditampilkan secara keseluruhan daftar hama penyakit yang ada, kemudian untuk lengkapnya, pakar dapat menekan menu *Detail*.

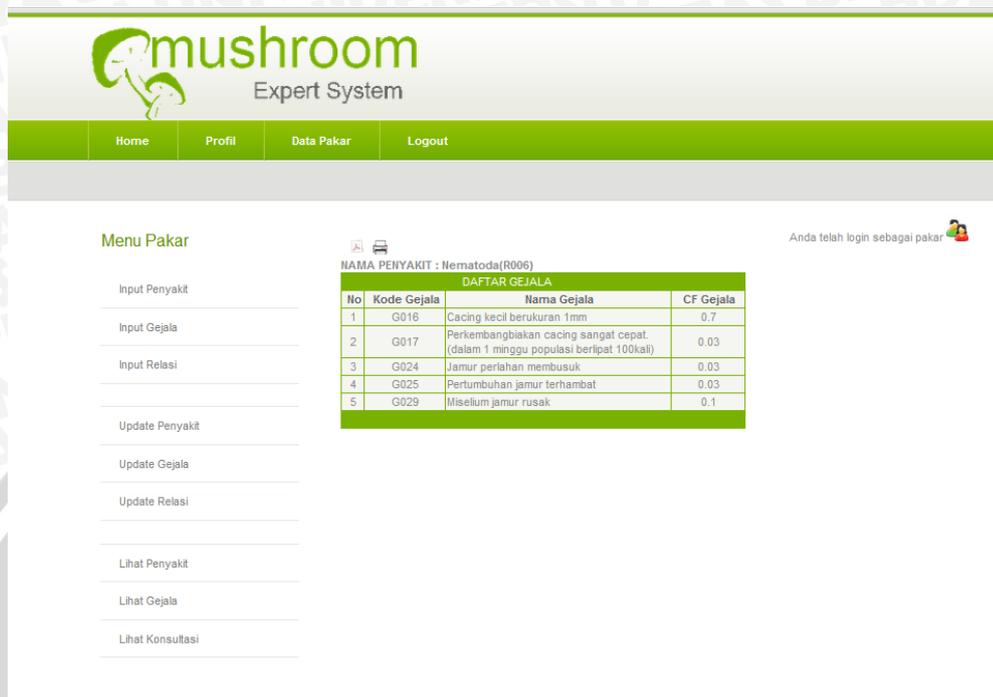


Gambar 5.20 Tampilan Halaman *Detail* Penyakit
Sumber: Implementasi

5.4.16 Tampilan Halaman Lihat Gejala



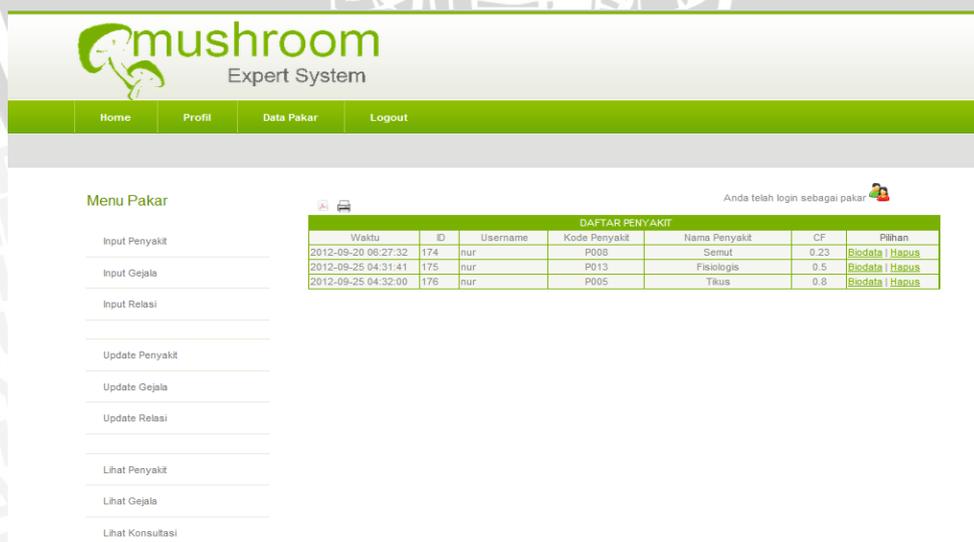
Gambar 5.21 Tampilan Halaman Gejala Per *Rule*
Sumber: Implementasi



Gambar 5.22 Tampilan Halaman Detail Gejala Per Rule
Sumber: Implementasi

Halaman ini digunakan untuk melihat informasi *detail* gejala untuk setiap *rule*. Data gejala lengkap dapat dicetak sesuai kebutuhan.

5.4.17 Tampilan Halaman Lihat Hasil Konsultasi



Gambar 5.23 Tampilan Halaman Hasil Konsultasi
Sumber: Implementasi

Halaman konsultasi dapat digunakan untuk *survey* hama penyakit yang banyak menyerang, ataupun melihat statistik pengunjung, maupun sebagai data rekap hasil konsultasi untuk dapat diolah kembali.

