**BAB IV**

**ANALISA DAN HASIL PERANCANGAN**

* 1. **Analisa Sistem**

Tahap analisa merupakan tahap yang paling penting dalam perancangan sebuah sistem, karena pada tahap inilah dilakukan evaluasi kinerja, identifikasi terhadap masalah yang ada, rancangan sistem dan langkah-langkah yang dibutuhkan untuk perancangan yang diinginkan sampai pada tahap analisa yang di harapkan.

Untuk itu penulis melakukan penganalisaan terhadap perancangan sistem yang dibuat, karena setiap sistem mempunyai keterbatasan atau kelemahan. Agar hal tersebut dapat diminimalisir penulis berusaha mengevaluasi setiap sistem yang dibuat, sesuai dengan judul yang penulis angkat dalam kasus ini.

Analisa sistem yang akan dibuat menggunakan metode *forward chaining* dengan mengumpulkan basis pengetahuan berupa data gejala penyakit pada cabai. Semua basis pengetahuan digunakan untuk membuat kesimpulan diagnosa penyakit pada tanaman cabai.

* + 1. **Analisa Sistem yang Sedang berjalan**

Berdasarkan informasi dari hasil pengamatan dan wawancara yang diperoleh dari Dinas pertanian Dharmasraya, dalam kegiatan pendataan kesehatan delakukan secara manual yang masih terkendala oleh beberapa hal seperti:

1. Diagnosa penyakit pada tanaman cabai masih dilakukan secara manual, sehingga membutuhkan waktu yang cukup lama.
2. Proses pencataan data dicatat dengan manual dengan bantuan *Microsoft Exel*.
3. Laporan data masih disimpan dalam format *exel*, sehingga tidak mampu menghasilkan suatu laporan yang terintegrasi dengan baik.

Dan berdasarkan permasalahan diatas adapun solusi yang dapat penulis usulkan untuk mengatasi permasalahan yang ada yaitu :

1. Merancang suatu sitem pakar yang dapat mendiagnosa penyakit pada tanaman cabai, sehingga dapat mempermudah tenaga pertanian dalam mendiagnosa penyakit pada tanaman cabai.
2. Pada sistem tersebut nantinya akan dibuatkan sebuah menu laporan riwayat konsultasi sistem pakar, sehingga dapat mempermudah tenaga pertanian dalam mengidentifikasi penyakit pada tanaman cabai.
   * + 1. **Analisa *Input***

Dalam data masukan (*input*)yang dibutuhkan untuk penyakit pada tanaman cabai pada Dinas Pertanian Dharmasraya yaitu, dari daun, warna akar, warna kulit buah dan batang. Namun pada sistem yang nantinya akan dirancang berdasarkan gejala-gejala dengan metode *Forward Chaining.*

* + - 1. **Analisa Proses**

Analisa proses merupakan analisa yang dilakukan terhadap proses di dalam aplikasi sistem pakar mengidentifikasi penyakit pada tanaman cabai. Adapun proses-proses yang terjadi dalam system ini nantinya adalah registrasi dan konsultasi penyakit yang akan dihubungkan nantinyan dengan beberapa master data lainnya.

* + - 1. **Analisa output**

Berdasarksn analisa yang dilakukan terhadap *output* dalam aplikasi sistem serta kebutuhan sietem menjelaskan *output* yang dihasilkan nantinya yaitu berupa laporan gejala-gejala yang dialami tanaman cabai dan laporan jenis penyakit yang tanaman cabai.

* + 1. **Basis Pengetahuan *(Kenowledge Base)***

Dalam perancangan sistem pakar, hal yang paling utama dilakukan adalah menentukan basis pengetahuan. Basis pengetahuan merupakan kumpulan-kumpulan fakta. Pendekatan basis pengetahuan yang dilakukan dengan menggunakan aturan (*rule*) karena perlu adanya langkah-langkah untuk mencapai tujuan.

Dalam kasus ini *user* (pengguna) akan menginputkan gejala-gejala penyakit yang diderita ke dalam sistem, selanjutnya sistem pakar dengan metode *forward chaining* akan mencocokkan data yang diinputkan dengan data pada basis pengetahuan.

* 1. **Data Jenis Penyakit pada Tanaman Cabai**

Pada basis pengetahuan berisikan data jenis penyakit pada tanaman cabai. Setelah melakukan pengambilan data dengan pakar diperoleh keterangan beberapa data penyakit. Berikut ini data jenis penyakit pada tanaman cabai pada Tabel **4.1** :

**Tabel 4.1 Data Jenis Penyakit pada Tanaman Cabai**

|  |  |
| --- | --- |
| **Kode Penyakit** | **Nama Penyakit** |
| P01 | Layu Fusarium (*Fusarium Oxysporum. Sp*) |
| P02 | Penyakit Layu Bakteri Ralstonia  (*Ralstonia solanacearum*) |
| P03 | Penyakit Busuk Buah  Antraknosa(*Collectrotichum gloeospoiroides*) |
| P04 | Penyakit Virus kuning (*Gemini Virus*) |
| P05 | Penyakit bercak daun (*Cercospora sp*.) |

* 1. **Data Gejala Penyakit pada Tanaman Cabai**

Berikut ini merupakan data-data basis pengetahuan tentang gejala-gejala penyakit pada tanaman cabai terdapat pada Tabel **4.2**.

**Tabel 4.2 Data Gejala Penyakit pada Tanaman Cabai**

|  |  |
| --- | --- |
| **Kode Gejala** | **Nama Gejala** |
| G01 | Daun mengalami kelayuan |
| G02 | Daun menguning dan menjalar ke ranting |
| G03 | Warna jaringan akar dan batang menjadi coklat |
| G04 | Layu secara tiba-tiba |
| G05 | Semua daun layu tetapi tidak berubah warna |
| G06 | Jaringan veskuler dari batang bagian bawah akar coklat |
| G07 | Munculnya bercak pada buah yang agak mengkilap |
| G08 | Seluruh buah keriput dan mengering |
| G09 | Warna kulit buah seperti jerami padi |
| G10 | Pucuk daun berubah menjadi kuning jelas |
| G11 | Tulang daun menebal dan daun menggulung ke atas |
| G12 | Tanaman kerdil dan tidak berbuah |
| G13 | Muncul bercak bulat berwarna coklat pada daun dan kering |
| G14 | Bercak berwarna pucat putih dengan warna tepi lebih tua |
| G15 | Terdapat lubang pada bercak tua |

* + 1. **Data Hubungan Penyakit dengan Gejala Penyakit pada Tanaman Cabai**

Berdasarkan data jenis penyakit dan gejala-gejala penyakit pada tanaman cabai maka dapat dibuat relasi sesuai dengan data yang didapatkan di lapangan. Relasi antara penyakit dengan gejala akan dijelaskan pada Tabel **4.3**.

**Tabel 4.3 Data Hubungan Penyakit dengan Gejala Penyakit pada Tanaman Cabai**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kode Penyakit** | **Nama Penyakit** | **Kode Gejala** | **Nama Gejala** |
| P01 | Layu Fusarium (Fusarium Oxysporum. Sp) | G01 | Daun mengalami kelayuan |
| G02 | Daun menguning dan menjalar ke ranting |
| G03 | Warna jaringan akar dan batang menjadi coklat |
| P02 | Penyakit Layu Bakteri Ralstonia (Ralstonia solanacearum) | G04 | Layu secara tiba-tiba |
| G05 | Semua daun layu tetapi tidak berubah warna |
| G06 | Jaringan veskuler dari batang bagian bawah akar coklat |
| P03 | Penyakit Busuk Buah Antraknosa(Collectrotichum gloeospoiroides) | G07 | Munculnya bercak pada buah yang agak mengkilap |
| G08 | Seluruh buah keriput dan mengering |
| G09 | Warna kulit buah seperti jerami padi |
| P04 | Penyakit Virus kuning (Gemini Virus) | G10 | Pucuk daun berubah menjadi kuning jelas |
| G11 | Tulang daun menebal dan daun menggulung ke atas |
| G12 | Tanaman kerdil dan tidak berbuah |
| P05 | Penyakit bercak daun (Cercospora sp.) | G13 | Muncul bercak bulat berwarna coklat pada daun dan kering |
| G14 | Bercak berwarna pucat putih dengan warna tepi lebih tua |
| G15 | Terdapat lubang pada bercak tua |
|  |  |  |  |

Pada Tabel 4.3 menjelaskan relasi atau hubungan antara jenis penyakit dengan gejala penyakit pada tanaman cabai, jenis dan gejala penyakit yang diberikan oleh pakar berdasarkan pengalaman dan teori ilmu yang ada pada pakar.

**4.4 Mesin Inferensi (*Inference Engine*)**

Langkah selanjutnya setelah mengetahui basis pengetahuan maka selanjutnya dilakukan analisa mesin inferensi. Mesin inferensi atau teknik yang digunakan dalam sistem pakar ini adalah dengan menggunakan metode *forward chaining* dengan aturan konsisi-aksi. Dimana dalam pengambilan kesimpulan dimulai dengan kondisi *IF* kemudian *THEN* untuk mendapatkan kesimpulan jenis penyakit yang diderita. Pada dasarnya *rule* terdiri dari dua bagian pokok, yaitu bagian *premis* atau kondisi dan bagian *conclution* atau kesimpulan. Struktur *rule* secara logika menghubungkan satu atau lebih kondisi (*premis*) pada bagian *IF* dengan satu atau lebih kesimpulan (*conclution*) yang terdapat pada bagian *THEN*.

Mesin inferensi akan menggambarkan secara grafis basis pengetahuan dan aturan-aturan dalam sistem pakar yang dibangun. Metode *inferensi* yang digunakan dalam penelusuran masalah pada sistem pakar dalam mendiagnosa pada tanaman cabai adalah metode *forward chaining* (runut maju). Metode *forward chaining* adalah strategi pencarian yang memulai proses pencarian dari sekumpulan data atau fakta, dari data-data tersebut dicari kesimpulan yang menjadi solusi dari permasalahan yang dihadapi. Didalam menemukan solusi dibutuhkan penyelesaian pada setiap tahapan, sebelum tahap yang satu selesai tidak dapat maju ke tahap berikutnya karena hal tersebut dapat berpengaruh dalam pencapaian solusi.

Pembuatan aturan atau *rule* digunakan untuk menentukan proses pencarian atau kesimpulan penyakit berdasarkan gejala-gejala yang diinputkan. Berdasarkan gambar pohon keputusan diatas dapat disimpulkan beberapa aturan atau *rule.* Berikut aturannya :

**Tabel 4.4 Rule Gejala dan Penyakit**

|  |  |  |
| --- | --- | --- |
| No | If | Then |
| 1 | G1, G2, G3 | P1 |
| 2 | G4, G5, G6 | P2 |
| 3 | G7, G8, G9 | P3 |
| 4 | G10, G11, G12 | P4 |
| 5 | G13, G14, G15 | P5 |

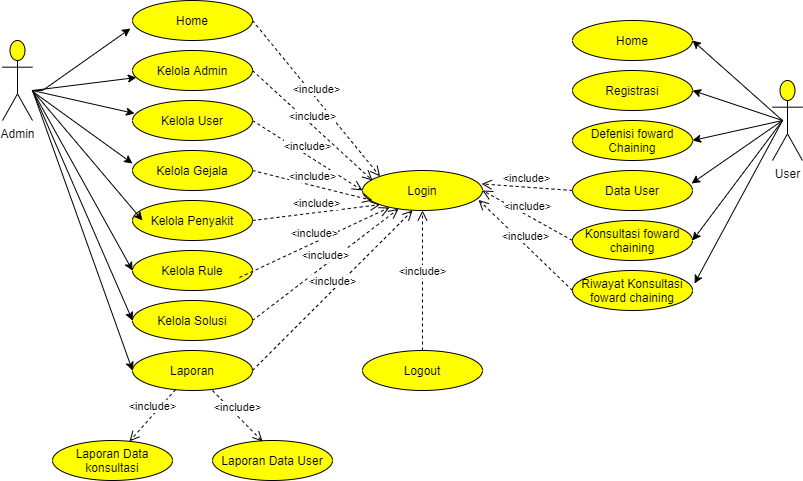
* 1. **Perancangan *UML***

Perancangan *UML* (*Unified Modelling Language*) digunakan untuk menentukan cara kerja program dalam pembangunan sistem pakar untuk mendiagnosa penyakit pada tanaman cabai berbasis *web* dengan menggunakan *use case diagram, class diagram, sequence diagram, dan activity diagram.* Untuk perancangan *UML* ini menggunakan program *Astah Community*. Adapun perancangan *UML* (*Unified Modelling Language*) untuk merancang diagram-diagram *UML* pada sistem pakar diagnosa penyakit pada tanaman cabai dijelaskan dibawah ini.

### *Use Case Diagram*

### *Use case diagram* adalah abtraksi dari interaksi antara sistem dengan *actor*. Oleh karena itu sangat penting untuk memilih abstraksi yang cocok. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *admin* dan *user* dari sebuah sistem dengan sistemnya sendiri melalui sebuah diagram bagaimana sebuah sistem dipakai.

Gambar diagram *use case diagram* untuk 2 aktor diatas dapat kita lihat pada Gambar 4.2 dibawah ini.



**Gambar 4.2 *Use Case Diagram***

* + 1. ***Class Diagram***

*Class* diagram menggambarkan struktur dan deskripsi *class* dan objek beserta hubungan antara data-data yang terdapat pada sistem *database*, seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Gambaran tentang *class* *diagram admin* dapat kita lihat pada Gambar 4.3 dibawah ini:



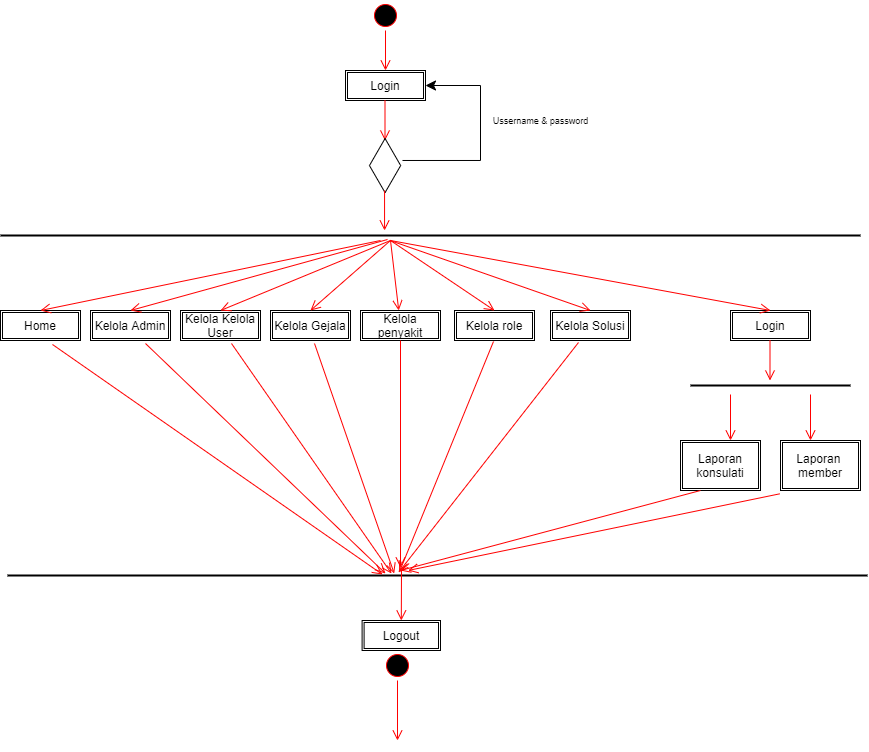
**Gambar 4.3 *Class* *Diagram***

* + 1. ***Activity Diagram***

*Activity diagram* merupakan sebuah diagram dimana dalam diagram tersebut terdapat semua aktivitas dalam sistem yang dirancang. Dalam *activity diagram* digambarkan aktivitas dari setiap aktor yang ada.

* 1. *Activity Diagram Admin*

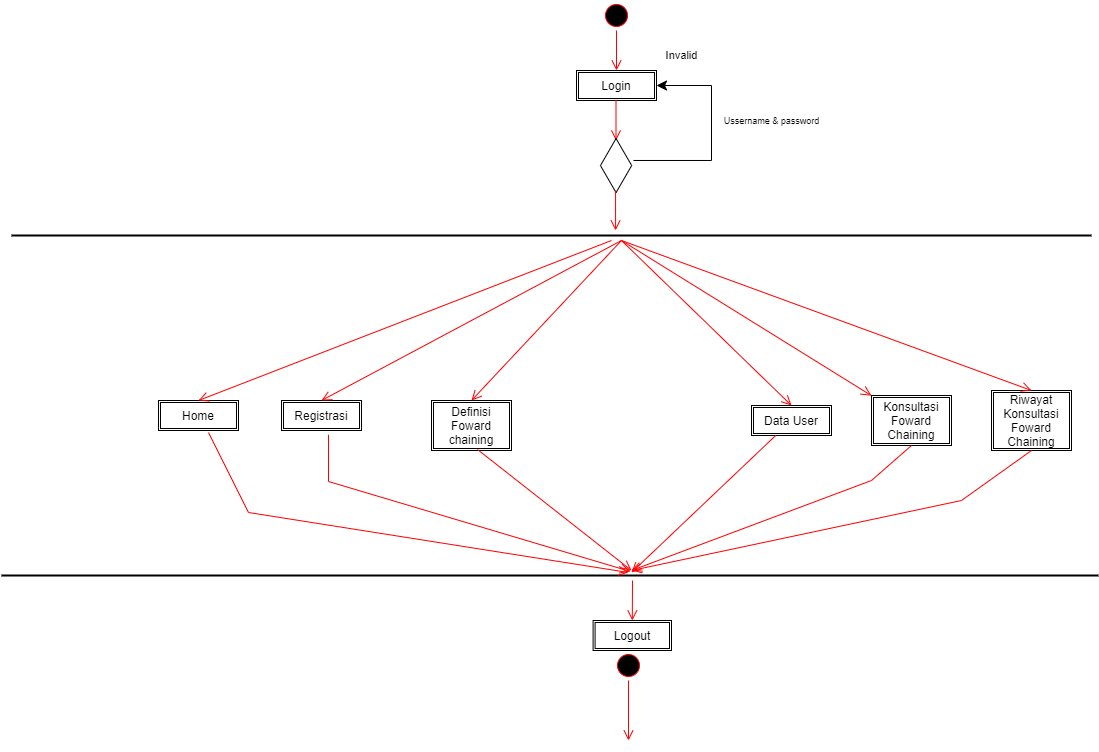
Berikut adalah gambaran *Activity* *diagram admin* menggambarkan segala aktivitas yang bisa dilakukan *admin* terhadap sistem yang dimulai dengan melakukan *login* dan memilih aktivitas yang akan dilakukan melalui menu pilihan yang ada. Gambar *activity diagram* untuk *activity admin* dapat kita lihat pada Gambar 4.4 berikut ini.

****

**Gambar 4.4 *Activity Diagram* Admin**

* 1. *Activity Diagram User*

Berikut adalah gambaran *Activity* *diagram* *user* menggambarkan segala aktivitas yang bisa dilakukan *userr* terhadap sistem yang dimulai dengan melakukan *login* kemudian *userr* dapat memilih aktivitas yang akan dilakukan melalui *menu* pilihan yang ada. Gambar *activity diagram* untuk *user* dapat kita lihat pada Gambar 4.5 dibawah ini.

****

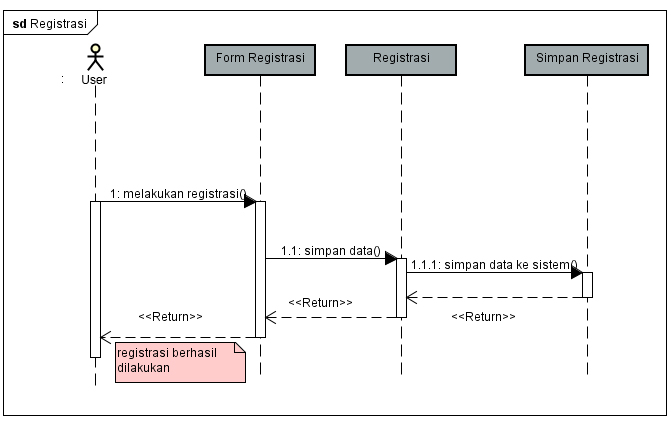
**Gambar 4.5 *Activity Diagram User***

* + 1. ***Sequence Diagram***

*Sequence diagram* digunakan untuk melacak eksekusi dari sebuah skenario *usecase*. *Sequence diagram* menggambarkan interaksi dengan menandai setiap partisipan dengan garis hidup yang berjalan secara vertikal ke bawah halaman dan urutan pesan dengan membaca ke arah bawah halaman (Andy Prasetyo Utomo, 2019). Pada laporan sistem diagnosa penyakit pada tanaman cabaiterdapat 8 *sequence diagram*. Disini dijelaskan secara rinci setiap proses yang terjadi.

1. *sequence Diagram Registrasi*

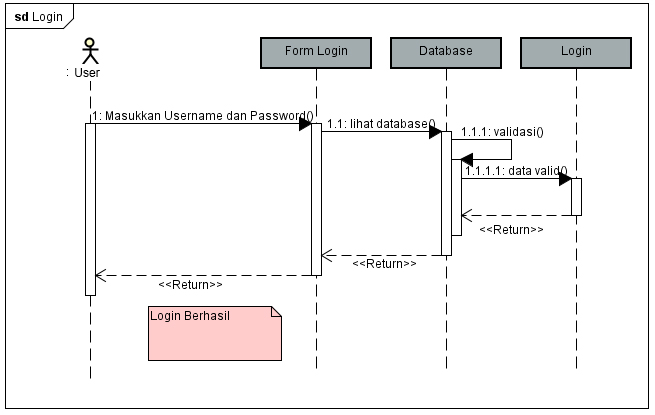
*Sequence diagram registrasi* bentuk rancangan program dapat dilihat seperti Gambar 4.6 berikut :



**Gambar 4.6 *Sequence* Diagram Registrasi**

1. *Sequence Diagram Login*

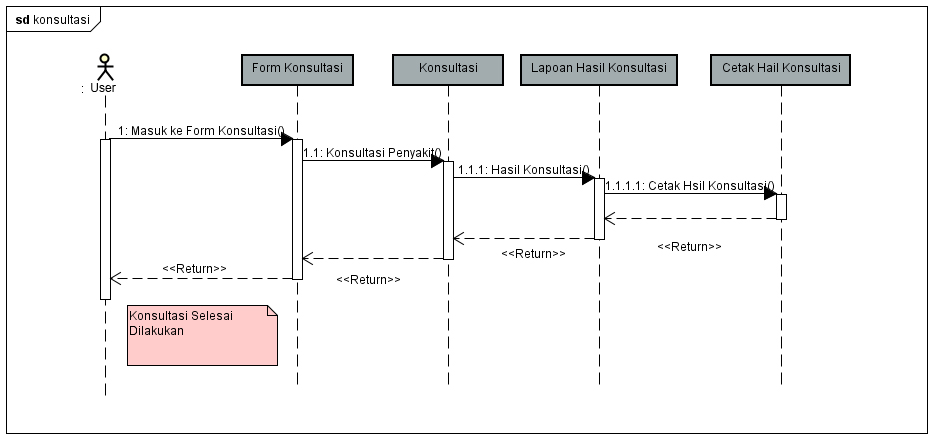
*Sequence diagram login admin* bentuk rancangan program dapat dilihat seperti Gambar 4.7 berikut :

****

**Gambar 4.7 *Sequence Diagram Login***

1. *Sequence Diagram* Konsultasi

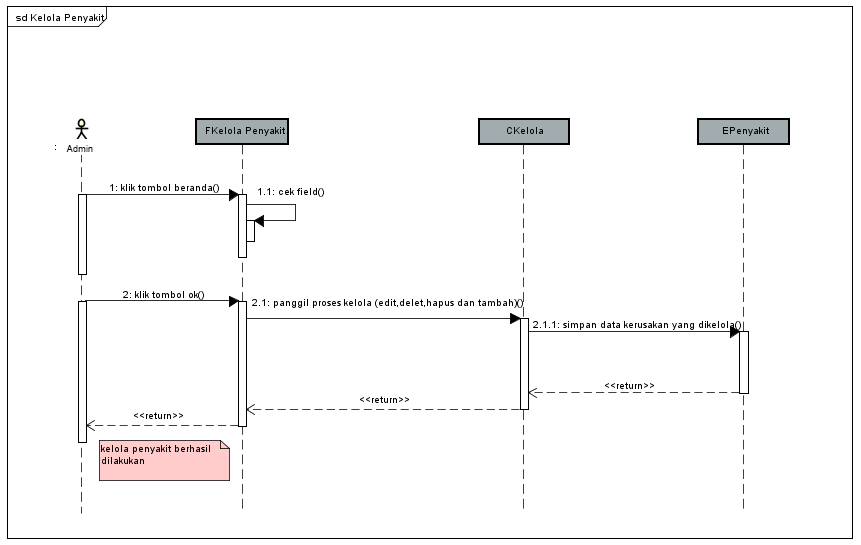
*Sequence diagram* konsultasibentuk rancangan program dapat dilihat seperti Gambar 4.8 berikut :

****

**Gambar 4.8 *Sequence Diagram* Konsultasi**

1. *Sequence Diagram* kelola Penyakit

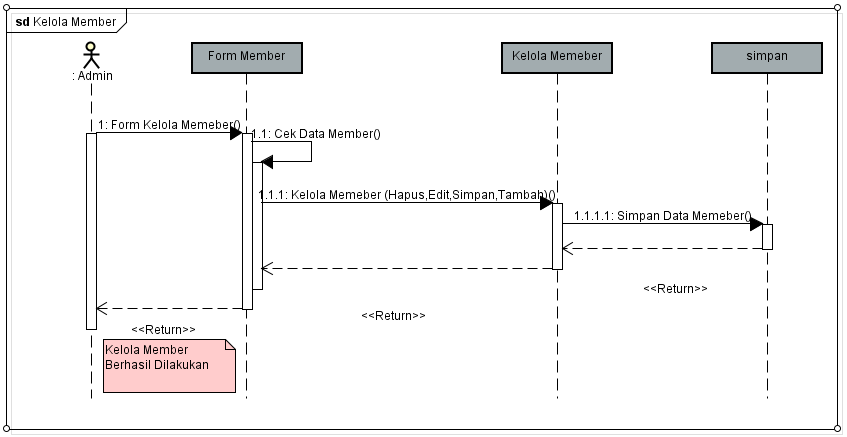
*Sequence diagram* kelola penyakitbentuk rancangan program dapat dilihat seperti Gambar 4.9 berikut :

****

**Gambar 4.9 *Sequence Diagram* Kelola Penyakit**

1. *Sequence Diagram* Kelola *User*

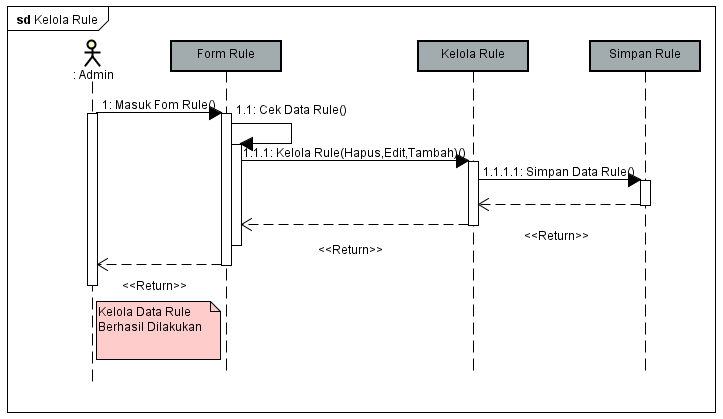
*Sequence diagram* kelola *member* bentuk rancangan program dapat dilihat seperti Gambar 4.10 berikut :

****

**Gambar 4.10 *Sequence* Diagram Kelola User**

1. *Sequence Diagram* Kelola *Rule*

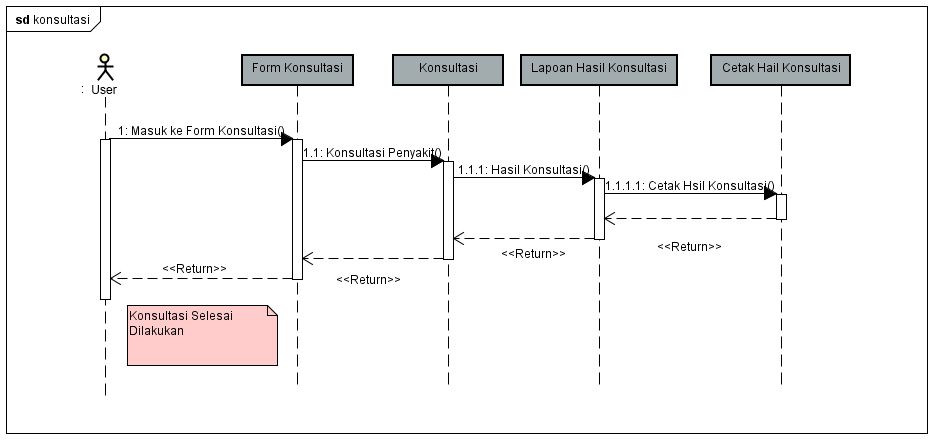
*Sequence diagram* kelola *Rule* bentuk rancangan program dapat dilihat seperti Gambar 4.11 berikut :

****

**Gambar 4.11 *Sequence* Diagram Kelola Rule**

1. *Sequence Diagram* Laporan Konsultasi User

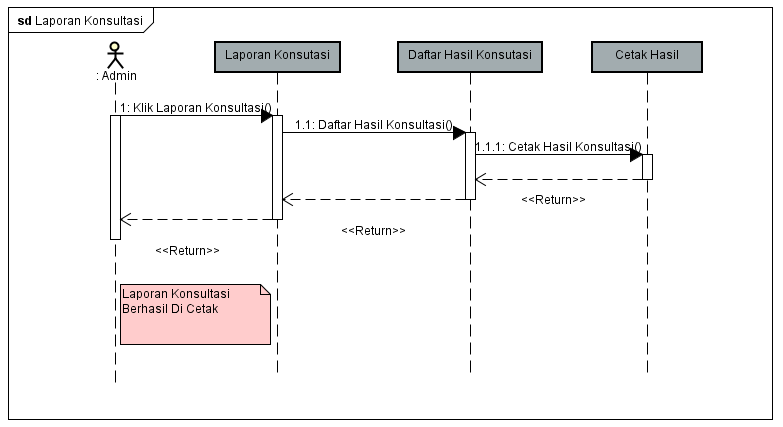
*Sequence diagram* laporan konsultasi user penyakitbentuk rancangan program dapat dilihat seperti Gambar 4.12 berikut :

****

**Gambar 4.12 *Sequence* Diagram Laporan Konsultasi User**

1. *Sequence Diagram* Laporan Data Konsultasi Keseluruhan

*Sequence diagram* laporan konsultasi keseluruhan penyakitbentuk rancangan program dapat dilihat seperti Gambar 4.13 berikut :



**Gambar 4.13 *Sequence* Diagram Laporan Data Konsultasi Keseluruhan**

* 1. **Desain Terinci**

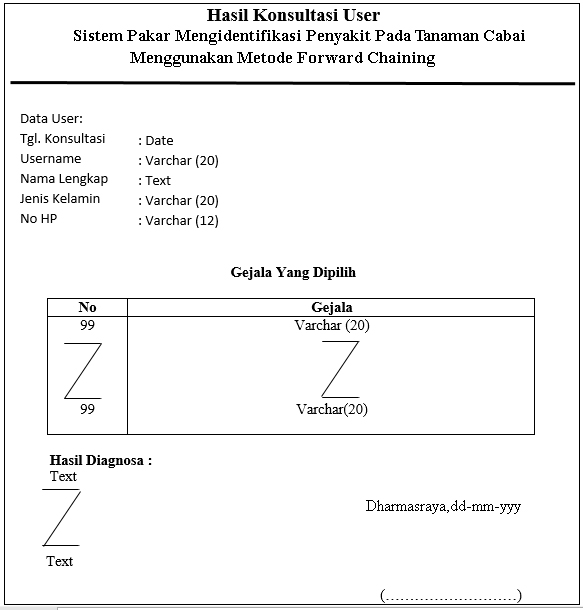
Desain terinci merupakan pengembangan lebih lanjut dari desain sistem secara global, dimana pada desain terinci ini dapat kita bagi atas empat rancangan yaitu desain *output*, desain *input*, desain *file*, dan logika program.

* + - 1. **Desain *Output***

Desain *output* merupakan suatu model keluaran dari sistem informasi dalam bentuk laporan yang ditampilkan ke layar monitor ataupun ke mesin cetak (*printer*). Berikut rancangan *output* dari sistem yang akan dibuat di sistem pakar diagnosa penyakit pada tanaman cabai.

**1. Laporan Data Hasil Konsultasi Member**

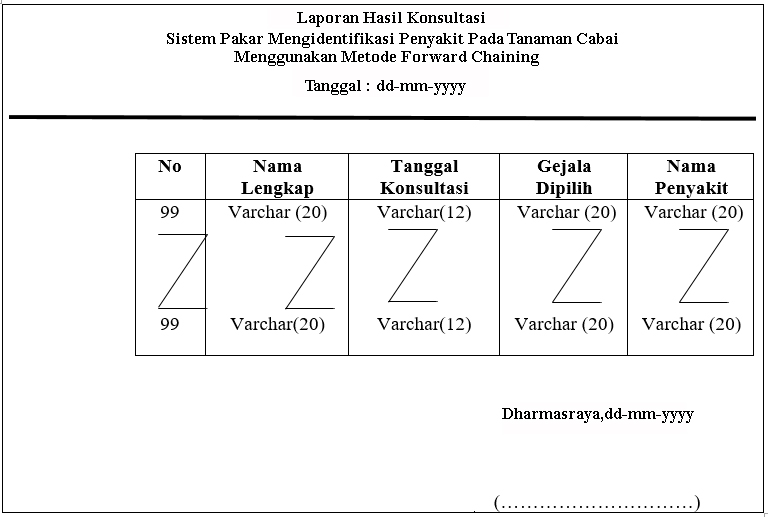
Pada laporan data hasil konsultasi member *admin* dapat melihat data member yang melakukan konsultasi. Analisa desain tampilan *output* konsultasi *member* dengan sistem dapat kita lihat pada Gambar 4.14 berikut ini.



**Gambar 4.14 Tampilan Laporan Data Hasil Konsultasi Member**

**2. Laporan Data Hasil Konsultasi**

Pada laporan data hasil konsultasi admin dapat melihat data member yang melakukan konsultasi. Analisa desain tampilan output laporan konsultasi dengan sistem dapat kita lihat pada Gambar 4.15 berikut ini.



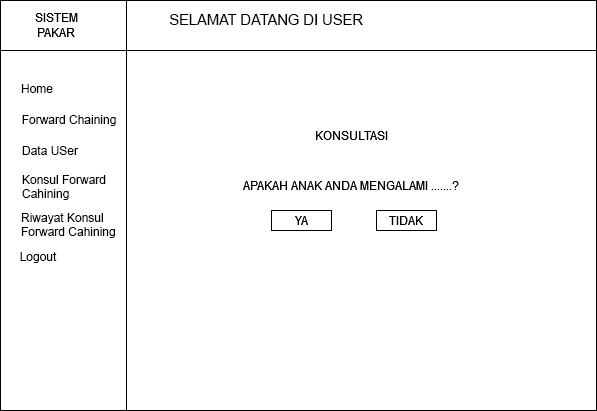
**Gambar 4.15 Tampilan Laporan Data Hasil Konsultasi**

* + 1. **Desain *Input***

Desain *Input* adalah halaman *user* melakukan proses *Enty Data*. Desain *input* merupakan suatu alat pemasukan data yang dibutuhkan dalam proses pembuatan laporan-laporan yang diinginkan perusahaan dalam mengambil keputusan. Yang dimaksud dengan *input* disini adalah *input* data yang langsung dihubungkan ke proses komputer melalui pengentrian dengan *keyboard*. Rancangandesain *input* sistem pakar diagnosa penyakit pada tanaman cabai bias dilihat seperti gambar berikut:

1. **Tampilan Input Konsultasi**

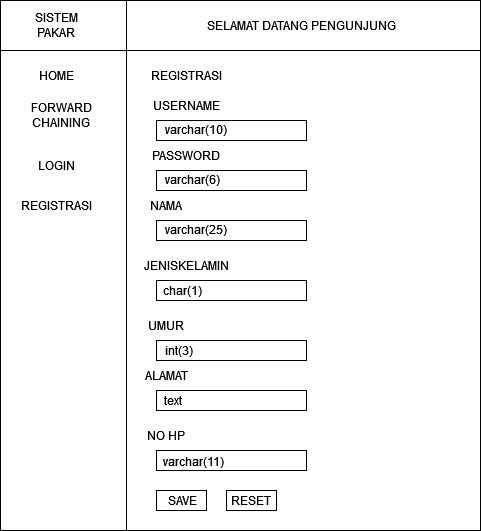
Gambaran tentang tampilan hasil konsultasi dapat dilihat dari Gambar 4.16

****

**Gambar 4.16 Tampilan *Input* Konsultasi**

1. **Tampilan Input *Registrasi***

Tampilan *form registrasi* sebagai tempat mengisi biodata pengunjung agar menjadi *member* dan dapat menjalankan sistem pakar serta berkonsultasi dengan sistem. Gambaran tentang tampilan *menu registrasi* userdapat dilihat dari Gambar 4.17 dibawah ini.

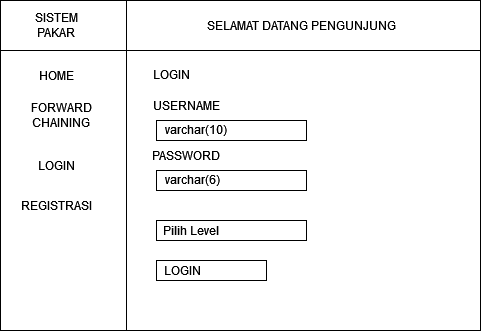


**Gambar 4.17 Tampilan Menu Registrasi**

1. **Tampilan *Form Login***

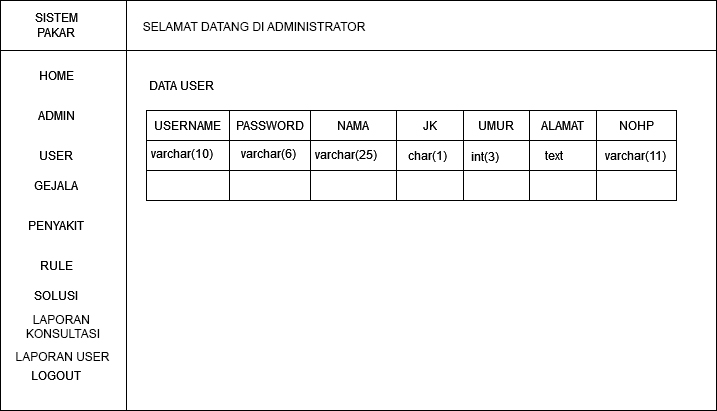
*Form login* disini terletak pada form *home*, dimana bagi *user member* dapat langsung memasukkan *username* dan *pasword* yang dimilikinya secara benar, maka akan tampil *form* konsultasi untuk mendiagnosa penyakit pada sistem, Sedangkan bagi pengunjung yang tidak memiliki *username* dan *password* untuk *login* namun ingin melakukan konsultasi dengan sistem maka wajibkan mendaftar terlebih dahulu. Setelah user mengisi *form* mendaftar dengan benar, maka *user* langsung mengisi *form login user* yang terletak pada tampilan *menu home* agar masuk kedalam *form* konsultasi.

Jika tidak mendaftar maka pengunjung hanya dapat melihat tampilan yang tersedia pada *menu home* saja, tanpa bisa melakukan konsultasi dengan sistem tentang masalah penyakit pada tanaman cabai. Gambaran tentang tampilan *login* dapat dilihat dari Gambar 4.18 dibawah ini.

**Gambar 4.18 Tampilan Form *Login***

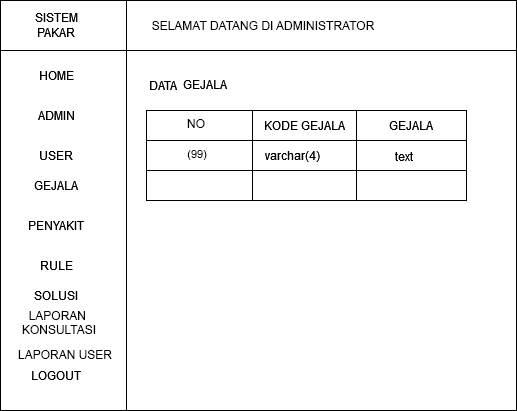
1. **Tampilan Input *User***

Gambaran tentang tampilan data *user* dapat dilihat pada Gambar 4.19.

**Gambar 4.19 Tampilan *Input user***

1. **Tampilan Input Gejala**

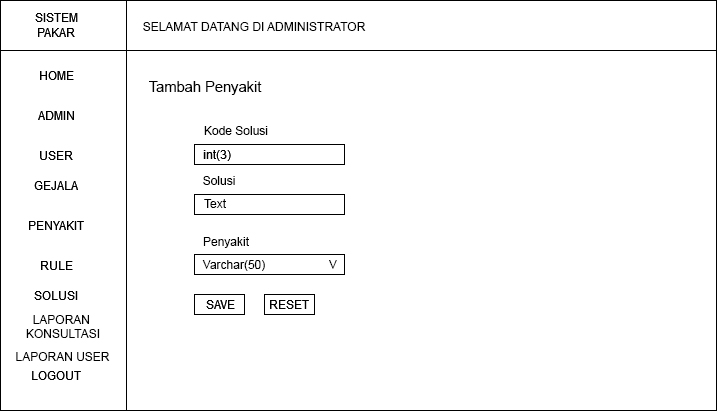
Gambaran tampilan gejala dapat dilihat pada Gambar 4.20 berikut.

****

**Gambar 4.20 Tampilan *Input* Gejala**

1. **Tampilan Input Solusi**

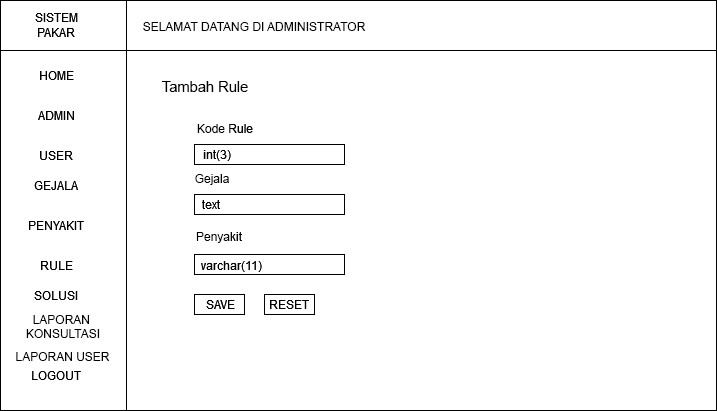
Gambaran tampilan gejala dapat dilihat pada Gambar 4.21 berikut.



**Gambar 4.21 Tampilan *Input* Solusi**

1. **Tampilan Input Rule**

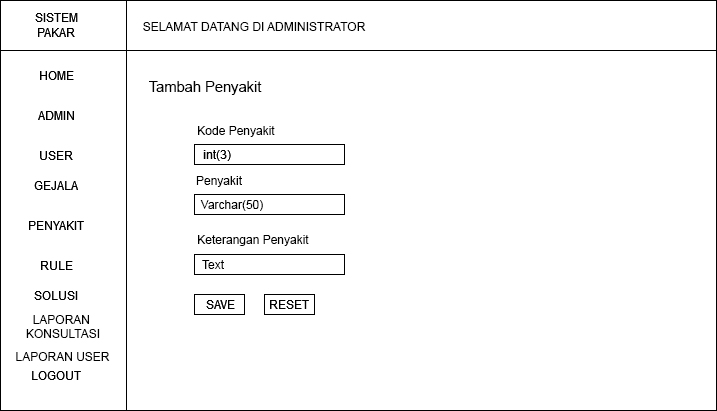
Gambaran tampilan rule dapat dilihat pada Gambar 4.22 berikut.

****

**Gambar 4.22 Tampilan *Input* Rule**

1. **Tampilan Input Penyakit**

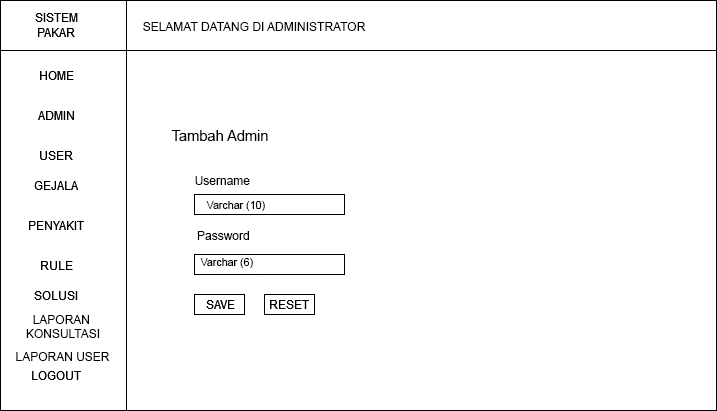
Gambaran tampilan penyakit dapat dilihat pada Gambar 4.23 berikut.

****

**Gambar 4.23 Tampilan *Input* Penyakit**

1. **Tampilan Input *Admin***

Gambaran tampilan *admin* dapat dilihat pada Gambar 4.24 berikut.

****

**Gambar 4.25 Tampilan *Input Admin***

* + 1. **Desain *File***

Database merupakan kumpulan dari beberapa file yang saling berhubungan pada *Class Diagram* , terdapat 6 tabel yang saling berhubungan diantaranya *admin, tbuser*, tbsaran, tbgejala, penyakit*, rule,* tbhasil\_konsultasi. File-file tersebut saling berhubungan berdasarkan kunci yang ada (*primary key*). Dan kunci tamu yang menjadi penghubung diantara satu tabel dengan tabel lainnya yang terdapat di dalam *database*.

Agar lebih jelas mengenai file-file pada sistem pakar mendiagnosa penyakit pada tanaman cabai dapat dilihat pada tabel-tabel berikut ini.

1. File *admin*

Merupakan struktur file tempat merekam data-data tipe *Admin* dengan bentuk struktur seperti Tabel 4.4 berikut ini.

Nama database : db\_cabai.sql

Nama tabel : admin

Primary key : idadmin

**Tabel 4.4 Desain *File*  Admin**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Width** | **Description** |
| idadmin | Int | (11) | Id admin |
| username | Varchar | |  |  | | --- | --- | | (10) |  | | Username |
| password | Varchar | (6) | Password |

1. File *user*

Merupakan struktur file tempat merekam data-data *user* dengan bentuk struktur seperti Tabel 4.5 berikut ini :

Nama database : db\_

cabai.sql

Nama tabel : user

Primary key : iduser

**Tabel 4.5 Desain *File*  *User***

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Width** | **Description** |
| iduser | Int | 11 | Id user |
| username | Varchar | 10 | Username |
| **Field name** | **Type** | **Width** | **Description** |
| password | Varchar | 6 | Password |
| nama | Varchar | 25 | Nama Lengkap |
| jeniskelamin | Char | 1 | Jenis kelamin |
| umur | Int | 3 |  |
| alamat | Text | - | Alamat |
| nohp | Varchar | 11 | Nomor handphone |

1. File gejala

Merupakan struktur file tempat merekam data-data gejala dengan bentuk struktur seperti Tabel 4.6 berikut ini :

Nama database : db\_cabai.sql

Nama tabel : gejala

Primary key : idgejala

**Tabel 4.6 Desain *File*  Gejala**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Width** | **Description** |
| idgejala | Int | 11 | Id gejala |
| kdgejala | Varchar | 4 | Kode Gejala |
| gejala | Text |  | Nama gejala |

1. File Penyakit

Merupakan struktur file tempat merekam data-data penyakit pada tanaman cabai dengan bentuk struktur seperti Tabel 4.7 berikut ini :

Nama database : db\_cabai.sql

Nama tabel : penyakit

Primary key : idpenyakit

**Tabel 4.7 Desain *File*  Penyakit**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Width** | **Description** |
| idpenyakit | Int | 11 | Id penyakit |
| kdpenyakit | Varchar | 4 | Kode Penyakit |
| penyakit | Text |  | Nama penyakit |
| keterangan | Text |  | Keterangan |

1. File *rule*

Merupakan struktur file tempat merekam data-data pertanyaan dengan bentuk struktur seperti Tabel 4.8 berikut ini :

Nama database : db\_ cabai.sql

Nama tabel : rule

Primary key : idrule

**Tabel 4.8 Desain *File*  *Rule***

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Width** | **Description** |
| idrule | Int | 11 | Id rule |
| kdrule | Varchar | 4 | Kode Rule |
| idgejala | Text |  | Id Gejala |
| idpenyakit | Int | 4 | Id Penyakit |

1. File hasil konsultasi

Merupakan struktur file tempat merekam data-data hasil konsultasi dengan bentuk struktur seperti Tabel 4.9 berikut ini :

Nama database : db\_ cabai.sql

Nama tabel : hasil

Primary key : idhasil

**Tabel 4.9 Desain *File*  Hasil Konsultasi**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Width** | **Description** |
| idhasil | Int | 11 | Id hasil konsultasi |
| iduser | Int | 11 | Id User |
| idpenyakit | Int | 11 | Id penyakit |
| tanggal | Date |  | Tanggal hasil konsultasi |
| gejalakonsul | Text |  | Gejala dipilih |

1. File solusi

Merupakan struktur file tempat solusi hasil konsultasi dengan bentuk struktur seperti Tabel 4.9 berikut ini :

Nama database : db\_cabai.sql

Nama tabel : solusi

Primary key : idsolusi

**Tabel 4.9 Desain Solusi**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Width** | **Description** |
| Idsolusi | Int | 11 | Id solusi |
| kdsolusi | Varchar | 4 | Kode solusi |
| solusi | Text |  | solusi |
| idpenyakit | Int |  | Id penyakit |