

BAB III METODE PENELITIAN

3.1 Bahan Penelitian

Bahan penelitian yang digunakan pada penelitian ini data terkait hama dan penyakit tanaman cabai rawit yang didapatkan dari hasil wawancara dengan seorang pakar di desa Mulangsari kecamatan pangkalan Jl. Raya pangkalan loji no.1.

Tabel 3. 1 Tabel Hama dan gejala tanaman cabai rawit

Hama	Gejala
Thrips	Daun mengeriting, Daun berwarna keperak-perakan pada bagian bawah, Daun berwarna coklat.
Kutu Kebul	Daun menguning, Bercak pada daun.
Kutu Daun	Daun keriput dan kerdil, Daun layu menguning, Daun keriting ke dalam
Tungau	Daun muda akan nampak terbakar pucuknya, Daun tua melengkung, Daun berwarna coklat, Daun menjadi kaku.

Tabel 3. 2 Tabel Penyakit dan gejala tanaman cabai rawit

Penyakit	Gejala
Gemini virus (Virus Kuning)	Daun pucuk berwarna kuning, Daun menggulung keatas, Tanaman kerdil dan tidak berbuah, Daun mengeriting ke atas.
Bercak Daun <i>Cercospora, sp</i>	Daun Bercak cokelat daun kering, Daun layu dan rontok, Bercak kecil berbentuk bulat pada daun.
Busuk Buah Antraknosa	Busuk buah jamur, Busuk buah bintik hitam, buah cabai mengerut dan keriting, Munculnya bercak yang agak mengkilap, Buah berwarna hitam dan orange, Warna kulit buah seperti jerami padi.
Layu Fusarium	warna akar dan batang tanaman menjadi coklat, Daun menjadi kaku dan melengkung ke bawah, Tunas dan bunga gugur, Daun layu dari bagian bawah, Pucuk tanaman layu dan mati.

3.2 Peralatan Penelitian

1. Perangkat keras yang digunakan adalah sebagai berikut :
 - a. Processor : Quad Core
 - b. Memory : 2 GB RAM
 - c. Hardisk : 500 GB HDD
 - d. OS : Windows 10
2. Perangkat lunak yang digunakan adalah sebagai berikut :
 - a. Xampp
 - b. Google Chrome
 - c. Sublime Text
 - d. PHP MyAdmin
 - e. MySQL

3.3 Lokasi Penelitian dan Waktu Penelitian

Penelitian ini dilaksanakan pada periode semester ganjil Tahun Ajaran 2019/2020. Bertempat di Dinas Pertanian Kabupaten Karawang yang bertempat di desa Mulangsari kecamatan pangkalan, kegiatan penelitian ini di laksanakan selama 4 bulan.

Tabel 3. 3 Jadwal penelitian

No	Bentuk Kegiatan	Minggu Pelaksanaan															
		Desember				Januari				Februari				Maret			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Izin Administrasi penelitian	■	■	■	■												
2.	Wawancara ke pakar			■	■												
3.	Pengumpulan data					■	■	■	■								
4.	Desain pemodelan aplikasi									■	■	■	■				
5.	Desain basis data													■	■	■	■
6.	Proses perancangan aplikasi																
7.	Uji coba aplikasi																
8.	Penyerahan sistem																

3.4 Prosedur Percobaan

1. Use case diagram

Berikut ini merupakan *use case diagram* yang di ajukan untuk sistem pakar diagnosa hama dan penyakit tanaman cabai rawit.



Gambar 3. 1 Use Case Diagram Hama dan Penyakit Tanaman Cabai Rawit

Tabel 3. 4 Skenario *use case login* admin / pakar

Nama use case	<i>Login admin</i>
<i>Triger event</i>	Admin memilih form <i>login</i>
Aktor	Admin, pakar
Kondisi sebelum	Aktor telah masuk halaman <i>login</i>
Kondisi sesudah	Aktor berada di halaman home sesuai hak akses
Normal course	Aktor Sistem 1.Mengisi data sesuai Form login yang ada. 2. Validasi data login 3. Sistem akan menampilkan pesan berhasil login 4.Menampilkan form beranda Sistem
Alternative course	Aktor Sistem 3a.Muncul pesan karena data login yang di isi tidak sesuai 3b.Admin dapat memperbaiki Data login yang tidak sesuai Untuk melanjutkan login
exception	Jika Admin salah memasukan nama pengguna dan sandi maka akan muncul pesan dan di kembalikan ke form login.

Tabel 3. 5 Skenario *use case* create data gejala

Nama <i>use case</i>	Data gejala
<i>Triger event</i>	Admin memilih form data gejala
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan data gejala
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu data gejala
	2. Menampilkan form data gejala
	3.Mengisi tambah data gejala Kemudian klik simpan
	4. Melakukan validasi data Gejala
	5.Jika data berhasil di validasi Maka tampil pesan data berhasil ditambahkan
<i>Alternative course</i>	Aktor Sistem
	3a.Jika data gagal di validasi Maka tampil pesan data gagal ditambahkan
exception	Jika Admin salah memasukan data gejala maka akan muncul pesan dan di kembalikan ke form data gejala.

Tabel 3. 6Skenario *use case* update data gejala

Nama <i>use case</i>	Data gejala
<i>Triger event</i>	Admin memilih form data gejala
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan data gejala
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu data gejala
	2. Menampilkan form data gejala
	3.Klik edit data gejala Kemudian mengisi data Gejala yang akan di edit Klik simpan
	4. Jika data berhasil di edit Maka muncul pesan data Gejala berhasil di perbarui
exception	-

Tabel 3. 9 Skenario *use case create data hama*

Nama <i>use case</i>	Data hama
<i>Triger event</i>	Admin memilih form data hama
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan data hama
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu data hama
	2. Menampilkan form data hama
	3.Mengisi tambah data hama Kemudian klik simpan
	4. Melakukan validasi data hama
	5.Jika data berhasil di validasi Maka tampil pesan data berhasil ditambahkan
<i>Alternative course</i>	Aktor Sistem
	3a.Jika data gagal di validasi Maka tampil pesan data gagal ditambahkan
<i>exception</i>	Jika Admin salah memasukan data hama maka akan muncul pesan dan di kembalikan ke form data hama.

Tabel 3. 10Skenario *use case update data hama*

Nama <i>use case</i>	Data hama
<i>Triger event</i>	Admin memilih form data hama
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan data hama
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu data hama
	2. Menampilkan form data hama
	3.Klik edit data hama Kemudian mengisi data Hama yang akan di edit Klik simpan
	4. Jika data berhasil di edit Maka muncul pesan data hama berhasil di perbarui
<i>exception</i>	-

Tabel 3. 11 Skenario *use case delete* data hama

<i>Nama use case</i>	Data hama
<i>Triger event</i>	Admin memilih form data gejala
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan data gejala
<i>Normal course</i>	Aktor Sistem
	1.Pilih menu data hama
	2. Menampilkan form data hama
	3.Klik hapus data hama
	4. Apakah anda yakin Akan menghapus data ini
<i>exception</i>	Jika Admin tidak jadi menghapus data hama maka akan di kembalikan ke form data hama.

Tabel 3. 12 Skenario *use case search* data hama

<i>Nama use case</i>	Cari data hama
<i>Triger event</i>	Admin memilih form cari data hama
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan cari data hama
<i>Normal course</i>	Aktor Sistem
	1.Pilih menu data hama
	2. Menampilkan form data hama
	3.Ketikan kata kunci pada kolom search, klik enter.
	4. Menampilkan data yang Dicari.
<i>Alternative course</i>	Aktor Sistem
<i>Exception</i>	-

Tabel 3. 13 Skenario *use case create* data penyakit

Nama <i>use case</i>	Data penyakit
<i>Triger event</i>	Admin memilih form data penyakit
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan data penyakit
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu data penyakit
	2. Menampilkan form data penyakit
	3.Mengisi tambah data penyakit klik simpan
	4. Melakukan validasi data penyakit
	5.Jika data berhasil di validasi Maka tampil pesan data berhasil ditambahkan
<i>Alternative course</i>	Aktor Sistem
	3a.Jika data gagal di validasi Maka tampil pesan data gagal ditambahkan
<i>exception</i>	Jika Admin salah memasukan data penyakit maka akan muncul pesan dan di kembalikan ke form data penyakit.

Tabel 3. 14Skenario *use case update* data penyakit

Nama <i>use case</i>	Data penyakit
<i>Triger event</i>	Admin memilih form data penyakit
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan data penyakit
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu data penyakit
	2. Menampilkan form data penyakit
	3.Klik edit data penyakit Kemudian mengisi data penyakit yang akan di edit Klik simpan
	4. Jika data berhasil di edit Maka muncul pesan data penyakit berhasil di perbarui
<i>exception</i>	-

Tabel 3. 15 Skenario *use case delete* data penyakit

Nama <i>use case</i>	Data penyakit
<i>Triger event</i>	Admin memilih form data penyakit
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan data penyakit
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu data penyakit
	2. Menampilkan form data penyakit
	3.Klik hapus data penyakit
	4. Apakah anda yakin Akan menghapus data ini
exception	Jika Admin tidak jadi menghapus data penyakit maka akan di kembalikan ke form data penyakit.

Tabel 3. 16 Skenario *use case create* data rule

Nama <i>use case</i>	Data rule
<i>Triger event</i>	Admin memilih form data rule
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan data rule
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu data rule
	2. Menampilkan form data rule
	3.Mengisi tambah data rule klik simpan
	4. Melakukan validasi data rule
	5.Jika data berhasil di validasi Maka tampil pesan data berhasil ditambahkan
<i>Alternative course</i>	Aktor Sistem
	3a.Jika data gagal di validasi Maka tampil pesan data gagal ditambahkan
exception	Jika Admin salah memasukan data rule maka akan muncul pesan dan di kembalikan ke form data rule.

Tabel 3. 17 Skenario *use case update* data rule

Nama <i>use case</i>	Data rule
<i>Triger event</i>	Admin memilih form data rule
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan data rule
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu data rule
	2. Menampilkan form data rule
	3.Klik edit data rule Kemudian mengisi data rule yang akan di edit Klik simpan
	4. Jika data berhasil di edit Maka muncul pesan data rule berhasil di perbarui
exception	-

Tabel 3. 18 Skenario *use case delete* data penyakit

Nama <i>use case</i>	Data penyakit
<i>Triger event</i>	Admin memilih form data penyakit
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan data penyakit
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu data penyakit
	2. Menampilkan form data penyakit
	3.Klik hapus data penyakit
	4. Apakah anda yakin Akan menghapus data ini
exception	Jika Admin tidak jadi menghapus data penyakit maka akan di kembalikan ke form data penyakit.

Tabel 3. 23 Skenario *use case update* identifikasi penyakit

Nama <i>use case</i>	Identifikasi penyakit
<i>Triger event</i>	Admin memilih form identifikasi penyakit
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan identifikasi penyakit
Normal <i>course</i>	Aktor Sistem
	1. Pilih menu identifikasi penyakit
	2. Menampilkan form Identifikasi penyakit
	3. Klik edit data identifikasi Penyakit Kemudian mengisi Data identifikasi penyakit yang akan di edit Klik simpan
	4. Jika data berhasil di edit Maka muncul pesan data penyakit berhasil di perbarui
exception	-

Tabel 3. 24 Skenario *use case delete* data penyakit

Nama <i>use case</i>	Identifikasi penyakit
<i>Triger event</i>	Admin memilih form Identifikasi penyakit
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan Identifikasi penyakit
Normal <i>course</i>	Aktor Sistem
	1. Pilih menu Identifikasi penyakit
	2. Menampilkan form Identifikasi penyakit
	3. Klik hapus Identifikasi penyakit
	4. Apakah anda yakin Akan menghapus data ini
exception	Jika Admin tidak jadi menghapus Identifikasi penyakit maka akan di kembalikan ke form Identifikasi penyakit.

Tabel 3. 27 Skenario *use case delete* solusi

Nama <i>use case</i>	Solusi
<i>Triger event</i>	Admin memilih form solusi
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan solusi
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu solusi
	2. Menampilkan form solusi
	3.Klik hapus solusi
	4. Apakah anda yakin Akan menghapus data ini
exception	Jika Admin tidak jadi menghapus solusi maka akan di kembalikan ke form solusi.

Tabel 3. 28 Skenario *use case create* master solusi

Nama <i>use case</i>	Master solusi
<i>Triger event</i>	Admin memilih form master solusi
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan master solusi
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu master solusi
	2. Menampilkan form Master solusi
	3.Mengisi tambah master solusi Kemudian klik simpan
	4. Melakukan validasi Master solusi
	5.Jika data berhasil di validasi Maka tampil pesan data berhasil ditambahkan
<i>Alternative course</i>	Aktor Sistem
	3a.Jika data gagal di validasi Maka tampil pesan data gagal ditambahkan
<i>Exception</i>	Jika Admin salah memasukan master solusi maka akan muncul pesan dan di kembalikan ke form master solusi.

Tabel 3. 29 Skenario *use case update* master solusi

Nama <i>use case</i>	Master solusi
<i>Triger event</i>	Admin memilih form Master solusi
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan Master solusi
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu Master solusi
	2. Menampilkan form Master solusi
	3.Klik edit data Master solusi Kemudian mengisi Data Master solusi yang akan di edit Klik simpan
	4. Jika data berhasil di edit Maka muncul pesan data Master solusi berhasil di perbarui
exception	-

Tabel 3. 30 Skenario *use case delete* master solusi

Nama <i>use case</i>	Master solusi
<i>Triger event</i>	Admin memilih form Master solusi
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan Master solusi
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu Master solusi
	2. Menampilkan form Master solusi
	3.Klik hapus Master solusi
	4. Apakah anda yakin Akan menghapus data ini
exception	Jika Admin tidak jadi menghapus Master solusi maka akan di kembalikan ke form Master solusi.

Tabel 3. 31 Skenario use case create pengguna

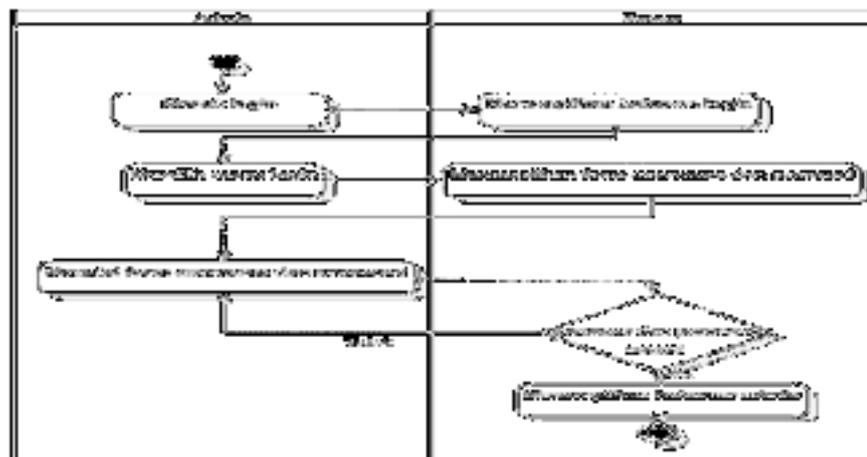
Nama <i>use case</i>	Pengguna
<i>Triger event</i>	Admin memilih form pengguna
Aktor	Admin, pakar
Kondisi sebelum	Aktor berada di halaman sesuai hak aksesnya
Kondisi sesudah	Aktor melakukan pengelolaan pengguna
Normal <i>course</i>	Aktor Sistem
	1.Pilih menu pegguna
	2. Menampilkan form pengguna
	3.Mengisi tambah pengguna klik simpan
	4. Melakukan validasi solusi
	5.Jika data berhasil di validasi Maka tampil pesan data berhasil ditambahkan
<i>Alternative course</i>	Aktor Sistem
	3a.Jika data gagal di validasi Maka tampil pesan data gagal ditambahkan
<i>exception</i>	Jika Admin salah memasukan pengguna maka akan muncul pesan dan di kembalikan ke form pengguna.

2. Activity diagram

Berikut ini merupakan beberapa *Activity Diagram* yang digunakan untuk menggambarkan alur kerja pada sebuah sistem yang akan di jalankan sesuai dengan ketentuan pembuatan sistem.

a. Activity diagram proses login admin

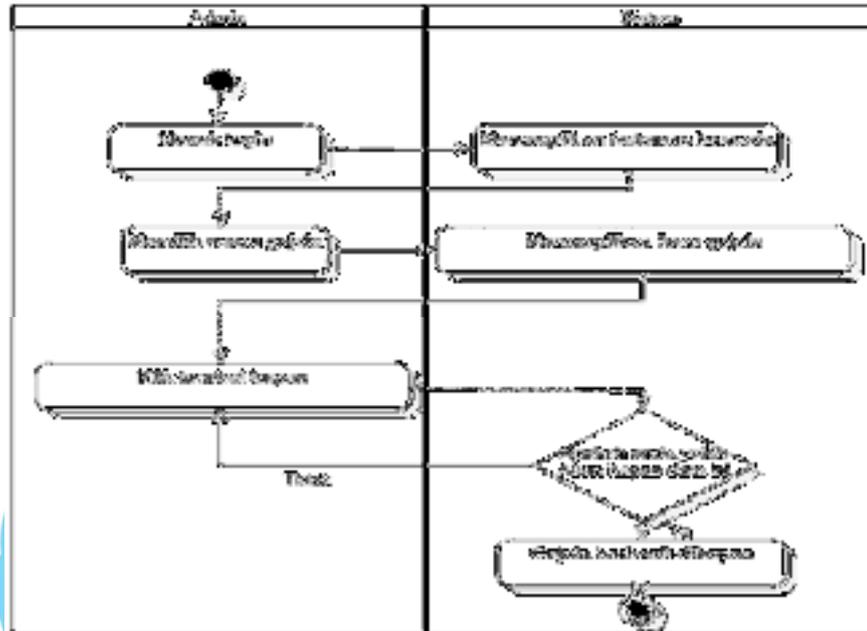
Aktifitas ini berfungsi untuk masuk ke sistem sebagai *admin* atau pakar.



Gambar 3. 2 Activity Diagram Login

d. *Activity diagram* proses *admin* hapus gejala

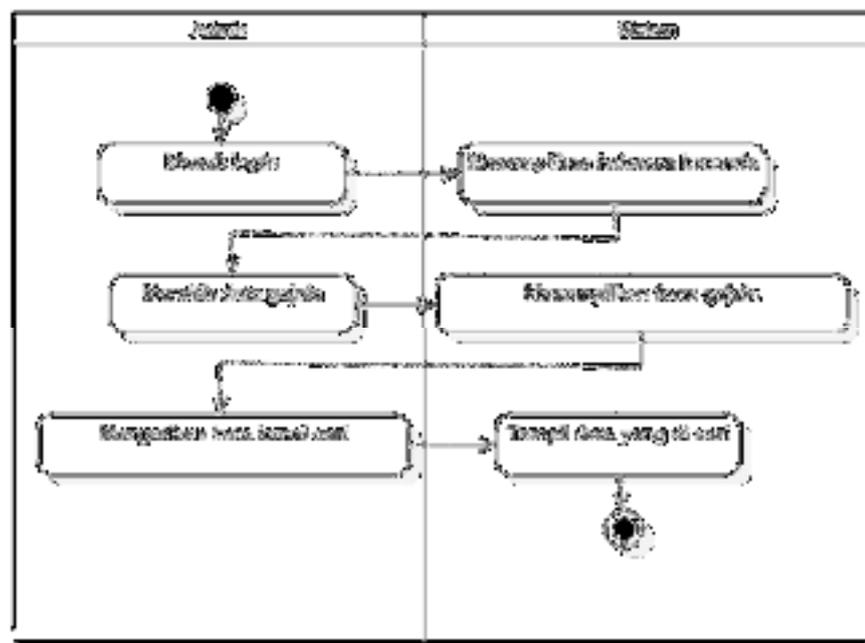
Aktifitas ini berfungsi untuk hapus gejala tanaman cabai rawit yang ada di sistem.



Gambar 3. 5 Activity Diagram Proses admin hapus gejala

e. *Activity diagram* proses *admin* cari Gejala

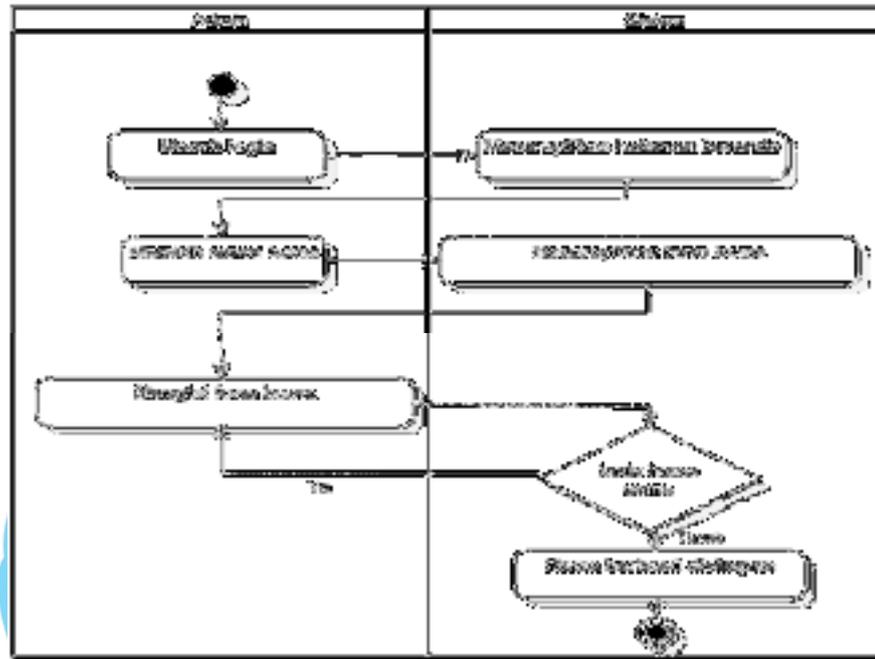
Aktifitas ini berfungsi untuk cari data gejala tanaman cabai rawit yang ada di sistem.



Gambar 3. 6 Activity Diagram Proses Admin cari Gejala

f. *Activity diagram* proses *admin input* data hama

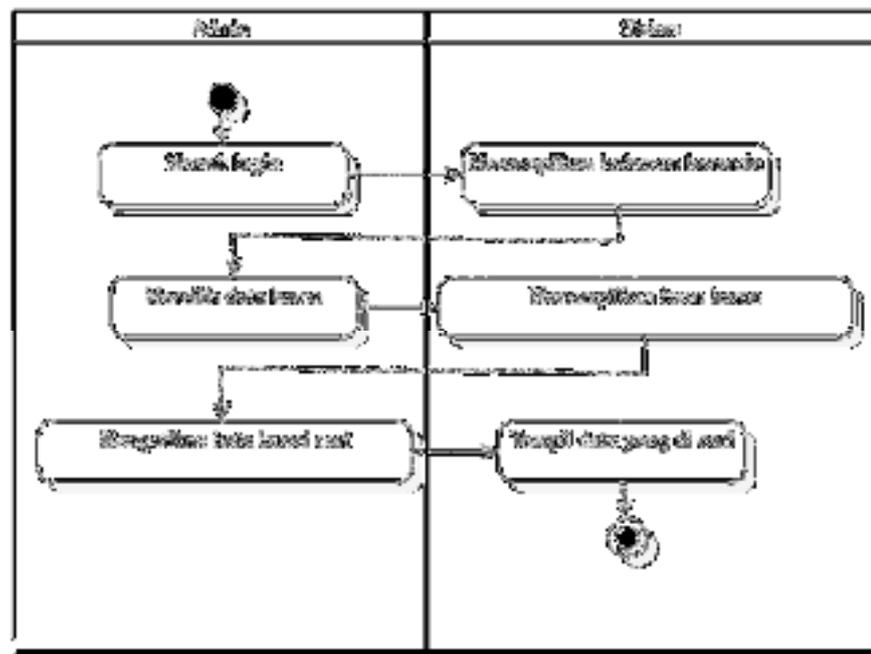
Aktivitas ini berfungsi untuk menginput data hama tanaman cabai rawit yang akan dimasukkan sistem.



Gambar 3. 7 *Activity Diagram* proses *admin Input* hama

g. *Activity diagram* proses *admin cari* hama

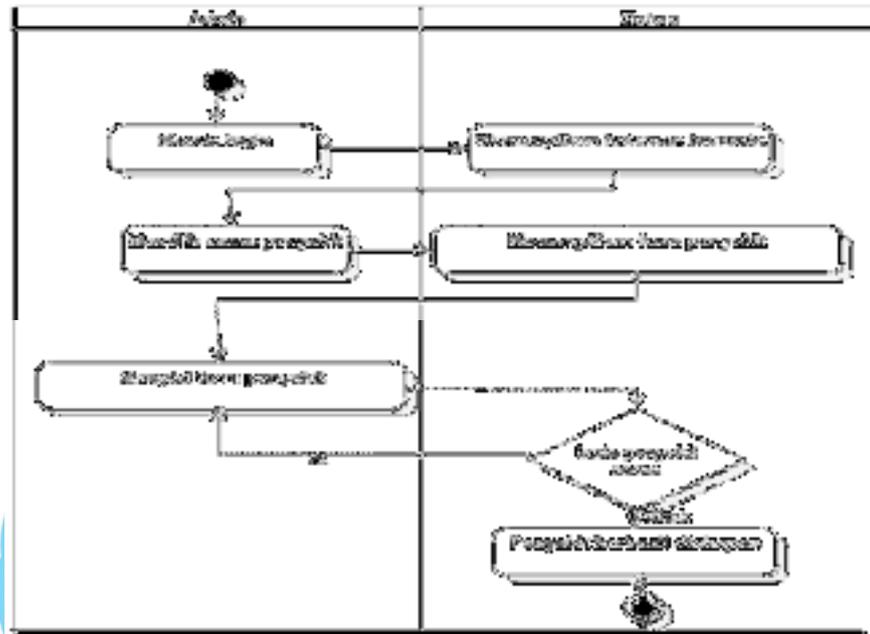
Aktivitas ini berfungsi untuk cari data hama tanaman cabai rawit yang ada di sistem.



Gambar 3. 8 *Activity Diagram* proses *admin cari* hama

h. *Activity diagram* proses *admin input* data penyakit

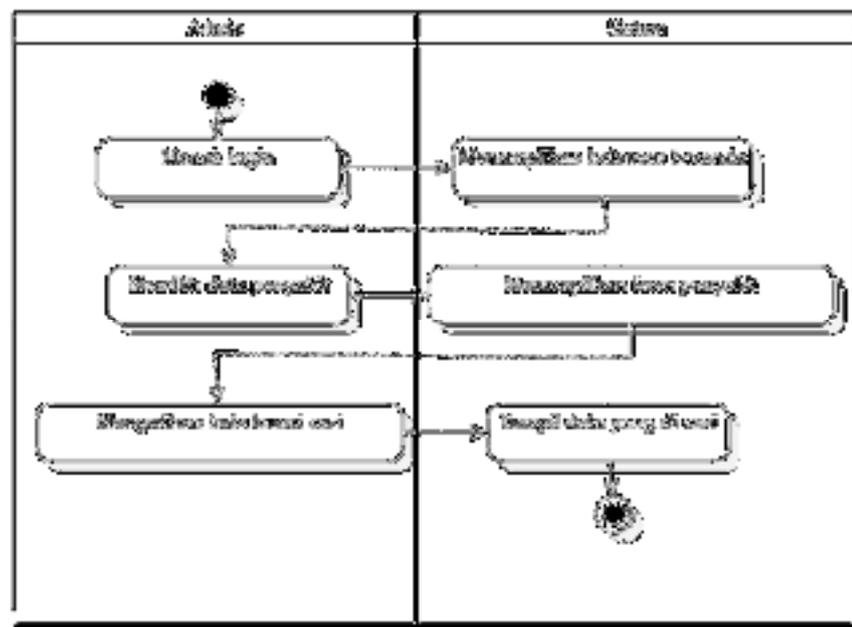
Aktifitas ini berfungsi untuk menginput data penyakit tanaman cabai rawit ke dalam sistem.



Gambar 3. 9 *Activity Diagram Input* penyakit

i. *Activity diagram* proses *admin cari* penyakit

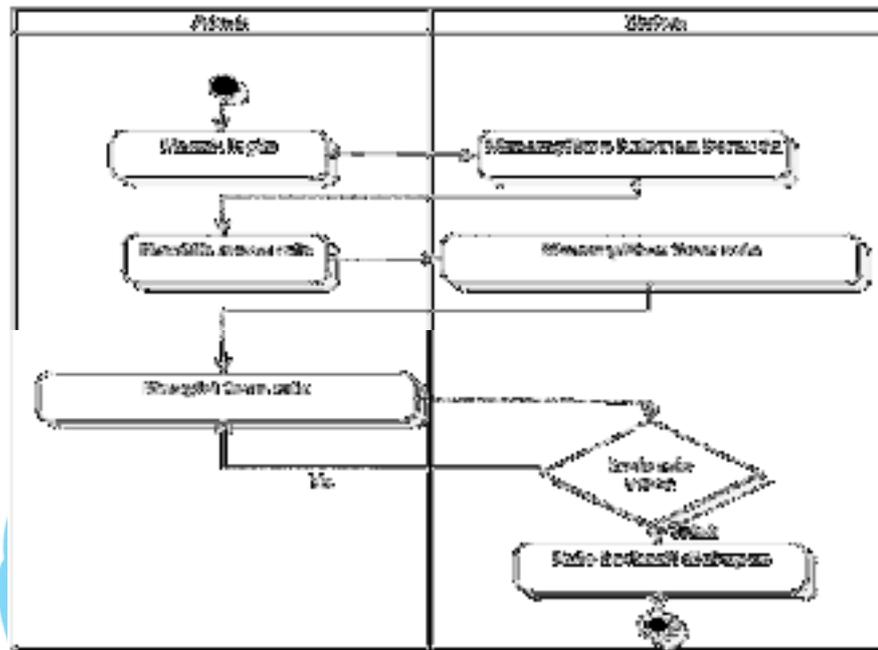
Aktifitas ini berfungsi untuk cari data penyakit tanaman cabai rawit yang ada di sistem.



Gambar 3. 10 *Activity Diagram Proses admin cari* penyakit

j. *Activity diagram proses admin input data rule*

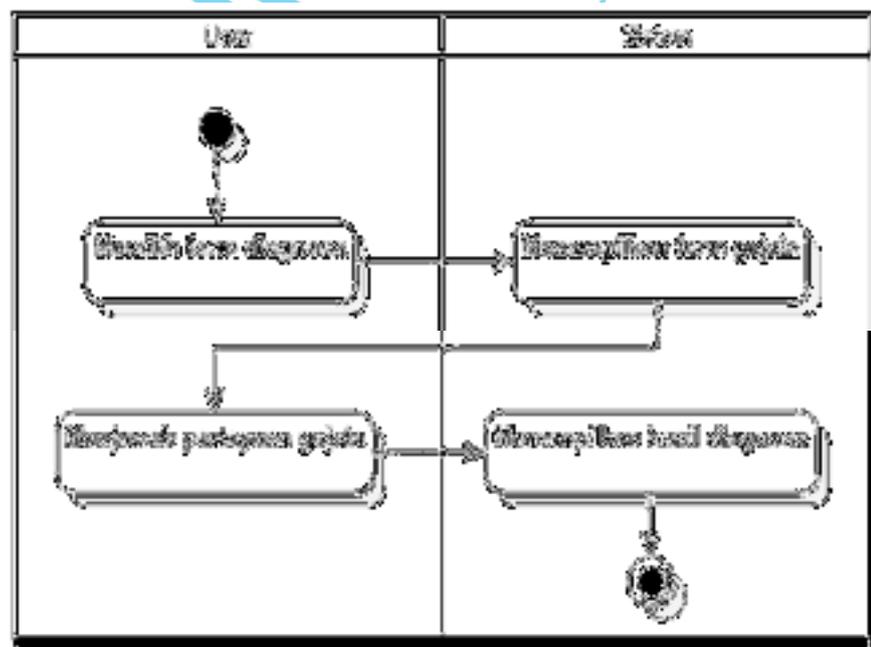
Aktifitas ini berfungsi untuk menginput data *rule* tanaman cabai rawit yang akan dimasukan sistem.



Gambar 3. 11 *Activity Diagram* Proses admin Input rule

k. *Activity diagram* diagnosa

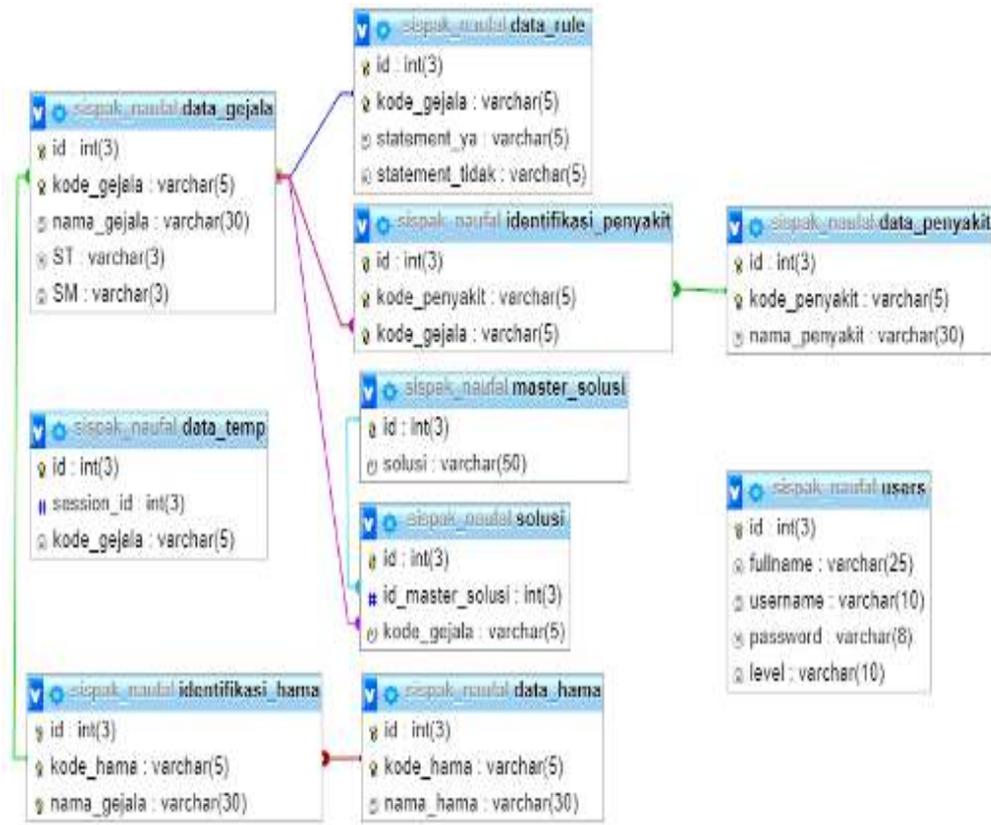
Aktifitas ini berfungsi untuk melakukan diagnosa hama dan penyakit tanaman cabai rawit.



Gambar 3. 12 *Activity Diagram* Diagnosa

3. Class Diagram

adalah untuk menggambarkan *class-class* pada sistem yang saling berhubungan satu sama lainnya.



Gambar 3. 13 Class diagram sistem pakar cabai rawit

4. Desain Basis Data

Perancangan yang dihasilkan meliputi desain yang terdiri dari desain basisdata (*database*), desain proses (algoritma pemrosesan data). Desain antar muka (*input dan output*).

Desain basisdata terdiri dari dua proses yaitu kamus data dan rancangan tabel.

1. Kamus data

Tahapan pertama dalam desain basis data adalah pembuatan kamus data, berikut daftar kamus data yang digunakan dalam sistem ini.

1. Data hama = {id, kode_hama, nama_hama}
2. Data penyakit = {id, kode_penyakit, nama_penyakit}
3. Data gejala = {id, kode_gejala, nama_gejala, ST, SM}
4. Identifikasi hama = {id, kode_hama, kode_gejala}

5. Identifikasi penyakit = {id, kode_penyakit, kode_gejala}
6. Solusi = {id, id_master_solusi, kode_gejala}
7. Master solusi = {id, solusi}
8. Data rule = {id, kode_gejala, statement_ya, st_tidak}
9. Data temp = {id, session_id, kode_gejala}
10. Data user = {fullname, username, password, level}

2. Rancangan tabel

Berikut adalah rancangan tabel yang digunakan untuk menyimpan data dalam sebuah sistem yang akan dibangun.

Nama database : sispak_naufal

Jumlah tabel : 10 tabel

Tabel 3. 32 Rancangan tabel data hama

No	Fields	Tipe	Deskripsi
1.	Id	Integer (3)	Primary key
2.	kode_hama	Varchar (5)	Kode_hama
3.	nama_hama	Varchar(30)	Nama_hama

Tabel 3. 33 Rancangan tabel data penyakit

No	Fields	Tipe	Deskripsi
1.	id	Integer (3)	Primary key
2.	kode_penyakit	Varchar (5)	Kode_penyakit
3.	nama_penyakit	Varchar (30)	Nama_penyakit

Tabel 3. 34 Rancangan tabel data gejala

No	Fields	Tipe	Deskripsi
1.	Id	Integer (3)	Primary key
2.	kode_gejala	Varchar (5)	kode_gejala
3.	nama_gejala	Varchar (30)	nama_gejala
4.	ST	Varchar (3)	ST
5.	SM	Varchar (3)	SM

Tabel 3. 35 Rancangan tabel identifikasi hama

No	Fields	Tipe	Deskripsi
1.	Id	Integer (3)	Primary key
2.	kode_hama	Varchar (5)	Kode_hama
3.	kode_gejala	Varchar(5)	Kode_gejala

Tabel 3. 36 Rancangan tabel identifikasi penyakit

No	Fields	Tipe	Deskripsi
1.	Id	Integer (3)	Primary key
2.	kode_penyakit	Varchar (5)	Kode_penyakit
3.	kode_gejala	Varchar(5)	Kode_gejala

Tabel 3. 37 Rancangan tabel data solusi

No	Fields	Tipe	Deskripsi
1.	id	Integer (3)	Primary key
2.	id_master_solusi	Varchar (3)	Id_master_solusi
3.	kode_gejala	Varchar(5)	Kode_gejala

Tabel 3. 38 Rancangan tabel master solusi

No	Fields	Tipe	Deskripsi
1.	id	Integer (3)	Primary key
2.	solusi	Varchar (50)	solusi

Tabel 3. 39 Rancangan tabel data rule

No	Fields	Tipe	Deskripsi
1.	id	Integer (3)	Primary key
2.	kode_gejala	Varchar (5)	kode_gejala
3.	statement_ya	Varchar (5)	statement_ya
4.	statement_tidak	Varchar(5)	statement_tidak

Tabel 3. 40 Rancangan tabel data temp

No	Fields	Tipe	Deskripsi
1.	id	Integer (3)	Primary key
2.	session_id	Varchar (3)	session_id
3.	kode_gejala	Varchar(5)	kode_gejala

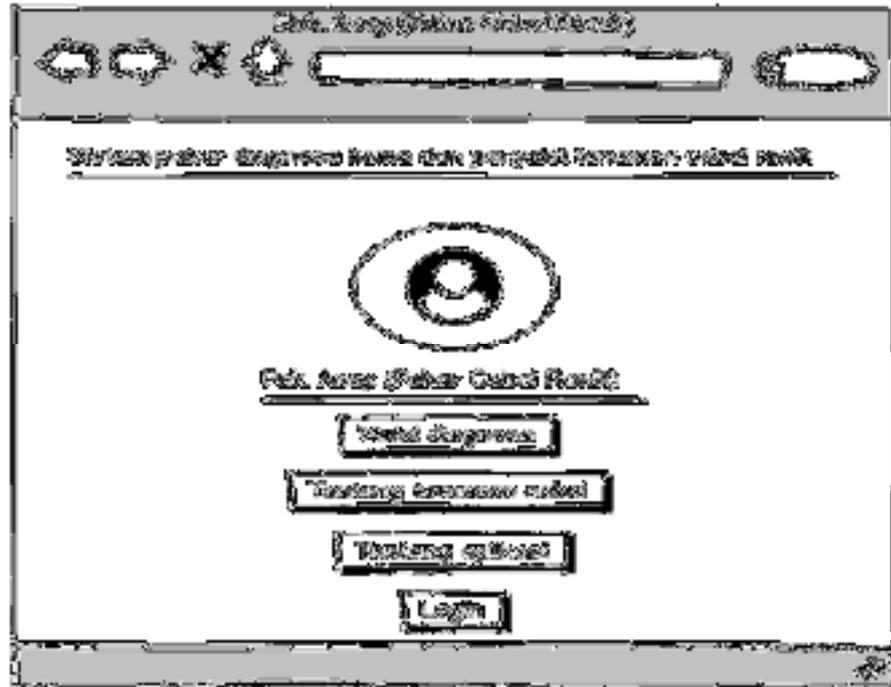
Tabel 3. 41 Rancangan tabel users

No	Fields	Tipe	Deskripsi
1.	Id	Integer(3)	Primary key
2.	Fullname	Varchar (25)	Fullname
3.	Username	Varchar (8)	username
4.	Pasword	Varchar(8)	Password
5.	level	Varchart(8)	level

5. Desain *Interface* Aplikasi

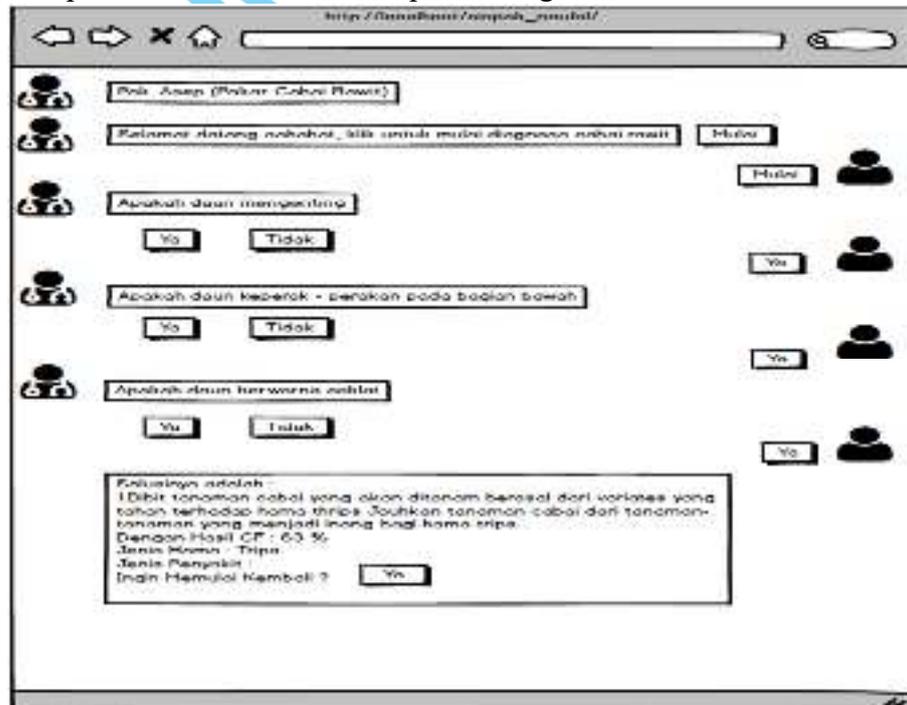
Berfungsi untuk menggambarkan sebuah aplikasi yang telah di buat pada sistem pakar diagnosa hama dan penyakit tanaman cabai rawit.

a. Tampilan beranda *user*



Gambar 3. 14 Tampilan *user*

b. Tampilan *user* saat melakukan proses diagnosa.



Gambar 3. 15 Tampilan saat proses diagnosa

g. Tambah *Rule*

Sistem pakar diagnosis demam dan penyakit lainnya berdasarkan suhu tubuh

Nama / Nama Rule
 Tambah Data Rule
 Nama gejala *
 [G 001] *
 Statement ID *
 G 002 *
 Statement TMSA *
 TMSA rule
 [G 003]
 [G 004]
 [G 005]

Gambar 3. 20 Tampilan tambah *rule*h. Daftar *list rule*

Sistem pakar diagnosis demam dan penyakit lainnya berdasarkan suhu tubuh

Nama / Nama Rule
 Daftar Rule
 Nama gejala *
 [G 001] *
 Statement ID *
 G 002 *
 Statement TMSA *
 TMSA rule
 [G 003]
 [G 004]
 [G 005]

No	Nama gejala	Statement ID	Statement TMSA	AKSI
1.	G 001	G 002	G 004	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
2.	G 002	G 003	-	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
3.	G 003	-	-	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
4.	G 004	G 005	G 006	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
5.	G 005	-	-	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>

Showing 1 to 10 of 31 entries

1 2 3

Gambar 3. 21 Tampilan daftar *list rule*

3.5 Analisis Data

1. Analisis Data Hama dan Penyakit Tanaman Cabai Rawit

Berisi tentang analisis data hama dan penyakit tanaman cabai rawit yang diperoleh dari desa Mulangsari kecamatan pangkalan Jl. Raya pangkalan loji no.1. Pada pengolahan data hama dan penyakit tanaman cabai rawit terdapat beberapa gejala untuk memudahkan maka akan di buat kode gejala.

Tabel 3. 42 Tabel Analisis Data Gejala Hama dan Peyakit

No	Hama	Penyakit	Gejala	Kode Gejala
1.	Thrips	-	Daun mengeriting, Daun berwarna keperak-perakan pada bagian bawah, Daun berwarna coklat.	G 01 G 02 G 03
2.	Kutu Kebul	-	Daun menguning, Bercak pada daun.	G 04 G 05
3.	Kutu Daun	-	Daun keriput dan kerdil, Daun layu menguning, Daun keriting ke dalam.	G 06 G 07 G 08
4.	Tungau	-	Daun muda akan nampak terbakar pucuknya, Daun tua melengkung, Daun berwarna coklat, Daun menjadi kaku.	G 09 G 10 G 11 G 12
5.	-	Geminivirus(Virus Kuning)	Daun pucuk berwarna kuning, Daun menggulung ke atas, Tanaman kerdil dan tidak berbuah, Daun mengeriting ke atas.	G 13 G 14 G 15 G 16
6.	-	Bercak Daun (Cercospora,sp)	Daun Bercak cokelat daun kering Daun layu dan rontok, Bercak kecil berbentuk bulat pada daun.	G 17 G 18 G 19

No	Hama	Penyakit	Gejala	Kode Gejala
7	-	Busuk Buah Antraknosa	Busuk buah jamur busuk buah bintik hitam Buah cabai mengerut dan mengering, Daun bercak yang agak mengkilap, Buah berwarna hitam dan orange Warna kulit buah seperti jerami padi.	G 20 G 21 G 22 G 23 G 24 G 25
8.	-	Layu Fusarium	warna akar dan batang tanaman menjadi cokelat, Daun menjadi kaku dan melengkung ke bawah, Tunas dan bunga gugur, Daun layu dari bagian bawah, Pucuk tanaman layu dan mati.	G 26 G 27 G 28 G 29 G 30

2. Analisis Data Rancangan Aplikasi Alur Kerja Sistem *Admin* Atau Pakar

- a. Pertama *admin* mengakses halaman sistem *admin*, maka akan dihadapkan dengan *form login* yang terdiri dari 2 kolom, yaitu *username* dan *password*.
- b. Kedua *Admin* mengisi *form login* yaitu dengan mengisi kolom *username* dan *password*
- c. Data *login* ini selanjutnya dicocokkan dengan data pengguna yang ada di dalam *database*, jika sesuai akan di cek hak aksesnya sebagai *admin login*.
- d. Setelah proses *login* sukses, *admin* dapat mengoperasikan sistem *admin*. Operasi yang dimaksud meliputi tambah data (*insert*), menampilkan data (*view*), memperbarui data (*update*), menghapus data (*delete*).

3. Analisis Data Rancangan Aplikasi Alur Kerja Sistem Petani Cabai Rawit

- a. Pertama petani mengakses halaman sistem, maka akan dihadapkan dengan *form dashboard* yang berisi diagnosa, tentang tanaman cabai rawit, tentang aplikasi dan login hanya untuk *admin* / pakar.

b. Kemudian *user* dapat melakukan diagnosa gejala tanaman cabai rawit, yang di alami *user* untuk mengetahui hasil hama atau penyakit.

4. Analisis Pengujian

Analisis pengujian dilakukan dengan mencocokkan data yang diberikan oleh pakar dengan aplikasi sistem pakar yang akan dibuat, sehingga pengujian tingkat efektifitas akurasi yang berjalan sesuai dengan aturan/ kaidah yang ditentukan. Analisis pengujian dilakukan untuk menguji seberapa efektif aplikasi yang akan dibuat. Data yang akan di uji akurasinya akan dibahas dalam bab hasil dan pembahasan. Adapun Rumus yang digunakan untuk mengukur akurasi sistem yang akan dibuat sebagai berikut :

$$\text{Nilai Akurasi} = \frac{\text{Jumlah Data Akurat}}{\text{Jumlah Semua data}} \times 100 \%$$

Keterangan :

Nilai Akurasi = Nilai Akurasi pengujian sistem dan pengetahuan pakar

Jumlah Data Akurat = Banyaknya data yang diuji sistem

Jumlah Semua Data = Keseluruhan data pengujian

