**SISTEM INFORMASI DIAGNOSA PENYAKIT PADA TANAMAN BUAH NANAS**

****

**Disusun oleh :**

**Nama : Titis Eka Setyawati**

**Nim : 18090049**

**Dosen Pengampu : La Ode Mohamad Zulfiqar, S.T,M.Kom**

**PROGRAM STUDI DIV TEKNIK INFORMATIKA**

**POLITEKNIK HARAPAN BERSAMA**

**TEGAL**

**2020**

**DAFTAR ISI**

**JUDUL** i

**DAFTAR ISI** ii

**DAFTAR TABEL** iii

**DAFTAR GAMBAR** iv

**BAB I PENDAHULUAN** 1

* 1. Latar Belakang 1
  2. Rumusan Masalah 2
  3. Maksud dan Tujuan 3
  4. Batasan Masalah 3

**BAB II PERANCANGAN SISTEM** 4

2.1. Deskripsi Sistem 4

2.2. Perancangan UML (*Unified Modelling Language*) 5

**BAB III PENUTUP** 34

3.1. Kesimpulan 34

**DAFTAR PUSTAKA** 35

**DAFTAR TABEL**

Tabel 1. Deskripsi Sistem 5

Tabel 2. Pendefinisian Usecase 6

Tabel 3. Skenario Login 7

Tabel 4. Skenario Mengelola Saran 7

Tabel 5. Skenario Mengelola Penyakit 8

Tabel 6. Skenario Mengelola Rule 9

Tabel 7. Skenario Mengelola Gejala 10

Tabel 8. Skenario Mengelola Pesan 11

Tabel 9. Skenario Mengelola Riawayat 12

Tabel 10. Skenario Mengelola Melakukan Diagnosa 13

Tabel 11. Skenario Mengelola Mengelola Profile 14

Tabel 12. Skenario Melihat Penyakit dan Saran 15

Tabel 13. Skenario Mengirim Pesan 16

**DAFTAR GAMBAR**

Gambar 1. Usercase Diagram 3

Gambar 2. Activity Admin Login 17

Gambar 3. Activity Kelola Penyakit 17

Gambar 4. Activity Kelola Saran Penyakit 18

Gambar 5. Activity Kelola Gejala 18

Gambar 6. Activity Kelola Rule 19

Gambar 7. Activity Kelola User 19

Gambar 8. Activity Kelola Pesan 20

Gambar 9. Activity Kelola Riwayat 20

Gambar 10. Activity Pendaftaran User 21

Gambar 11. Activity User Login 21

Gambar 12. Activity Diagnosa Penyakit 22

Gambar 13. Activity Riwayat Diagnosa 22

Gambar 14. Activity Saran Penyakit 23

Gambar 15. Activity Kirim Pesan 23

Gambar 16. Class Controler 24

Gambar 17. Sequence Kelola Penyakit 25

Gambar 18. Sequence Kelola Gejala 26

Gambar 19. Sequence Kelola Saran 27

Gambar 20. Sequence Kelola Rule 28

Gambar 21. Sequence Kelola User 29

Gambar 22. Sequence Kelola Profile User 30

Gambar 23. Sequence Diagnosa 31

Gambar 24. Sequence Riwayat 32

Gambar 25. Sequence Kirim Pesan 33

# BAB I

# PENDAHULUAN

* 1. **Latar Belakang**

Buah nanas (Ananas comosus) merupakan salah satu tanaman yang banyak dibudidaya di Indonesia, terutama didaerah Jawa dan Sumatra. Tiap tahun produksinya terus meningkat. Pada tahun 2005 produksi nanas di Indonesia mencapai 925,082 ton dan 1.427,781 ton pada thaun 2006. Semakin banyaknya permintaan terhadap buah ini memberikan prospek usaha yang sangat menjanjikan untuk para user buah di Indonesia. Peluang ini yang telah dimanfaatkan banyak user lokal yang membudidayakan buah berbentuk unik ini. penanaman suatu komoditas pertanian secara luas dan monokultur berpeluang terjadinya gangguan suatu hama atau penyakit. Di beberapa negara produsen buah nanas dilaporkan adanya beberapa hama dan penyakit berbahaya yang dapat mengancam produksi tanaman buah nanas [1].

Masalah gangguan penyakit tersebut juga dialami oleh user buah nanas di Indonesia. Untuk mengatasi masalah ini umumnya user melakukan pengandalian secara konvensional seperti menggunakan pestisida secara intensif. Useran pestisida secara berlebihan selain tidak efisien juga dapat menimbulkan berbagai masalah yang lain seperti pencemaran lingkungan [2]. Untuk mengatasi masalah tersebut dibutuhkan sistem informasi diagnosa atau aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. sistem informasi diagnosa juga dapat didefinisakan sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut [3].

UML (Unified Modeling Language) merupakan bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifacts (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (Object-Oriented) [4].

Berdasarkan permasalahan yang muncul, dalam penelitian ini dibuat suatu sistem informasi dengan judul “**SISTEM INFORMASI DIAGNOSA PENYAKIT PADA TANAMAN BUAH NANAS”** yang dapat membantu user dalam mendapatkan solusi atau penyakit yang menyerang tanaman buah nanas mereka tanpa harus berkonsultasi langsung dengan pakar dan dapat membantu pakar dengan cara menggantikan pakar jika pakar sedang tidak ditempat. Adapun metode perancangan yang akan dimodelkan melalui perancangan sistem dengan UML (*Unified Modelling Language*) untuk memvisualisasikan sistem yang akan dibuat.

* 1. **Rumusan Masalah**

Berdasarkan latar belakang yang diangkat, maka dibuat rumusan masalah yang akan diselesaikan dalam penelitian ini, antara lain :

Bagaimana cara merancang sistem informasi yang dapat mendeteksi secara dini penyakit pada Tanaman Buah Nanas menggunakan metode perancangan UML (*Unified Modelling Language*) ?

* 1. **Maksud dan Tujuan**

Maksud dan tujuan dari tugas ini dalah untuk mengetahui cara membuat sistem informasi deteksi dini penyakit pada tanaman Buah Nanas menggunakan metode perancangan UML (*Unified Modelling Language*).

* 1. **Batasan Masalah**

Untuk membatasi pembahasan agar terfokus pada aspek yang diangkat sebagai pembahasan utama dalam penelitian ini, maka perlu dibuat adanya batasan-batasan masalah, antara lain :

1. Jenis penyakit terbatas pada penyakit yang ada pada buah nanas.
2. Hasil akhir diagnosis menampilkan kemungkinan penyakit dan saran penanggulangan.
3. Metode Perancangan yang akan dibuat menggunakan UML (*Unified Modelling Language*) dengan *Usecase, Acktivity, Class* dan *Sequence* Diagram.
4. Sistem pakar yang dibuat berbasis website menggunakan *framework* codeigniter*.*

# 

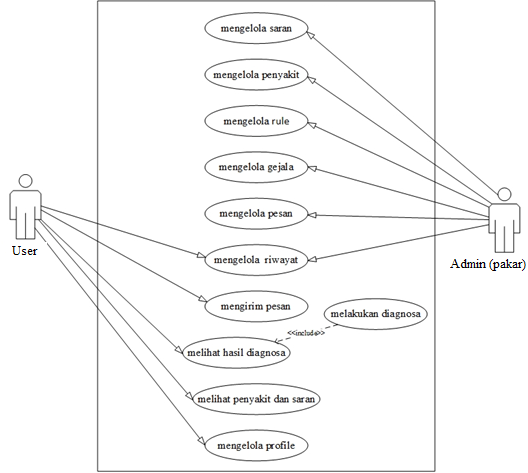
# BAB II

# PERANCANGAN SISTEM

1. **Deskripsi Sistem**

Sistem Informasi Diagnosa Penyakit Pada Tanaman Buah Nanas adalah sebuah sistem informasi hasil diagnosa penyakit yang terjadi pada tanaman buah nanas secara dini serta memberikan solusi pada masalah tersebut. Sistem Informasi Diagnosa Penyakit Pada Tanaman Buah Nanas yang dimodelkan melalui perancangan sistem dengan UML (*Unified Modelling Language*) untuk memvisualisasikan sistem yang akan dibuat.

1. **Perancangan UML (*Unified Modelling* *Language*)**
2. **Usecase Diagram**

****

**Gambar 1. Usercase Diagram**

**Deskripsi Aktor:**

**Tabel 1. Deskripsi Sistem**

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1 | Admin (pakar) | Admin (pakar) adalah orang yang melakukan login untuk dapat mengakses menu admin, dapat melakukan pengelolaan saran, dapat melakukan pengelolaan penyakit, dapat melakukan pengelolaan rule, dapat melakukan pengelolaan gejala, dapat melakukan pengelolaan pesan, dapta melakukan pengelolaan riwayat |
|
|
| 2 | User | User adalah orang yang melakukan login untuk dapat mengakses menu user, dapat melakukan diagnosa, dapat melihat hasil diagnosa, dapat melakukan pengelolaan profile, dapat melakukan pengelolaan riwayat, dapat mengirim pesan, dapat melihat penyakit dan saran |
|
|

**Pendefinisian Usecase:**

**Tabel 2. Pendefinisian Usecase**

|  |  |  |
| --- | --- | --- |
| No | Use Case | Deskripsi |
| 1 | Login | Merupakan proses untuk melakukan login admin |
| 2 | Mengelola saran | Mengelola saran merupakan proses pengelolaan data saran yang meliputi dapat melihat data saran, dapat melakukan penambahan data saran, dapat melakukan update data saran, dapat menghapus data saran |
| 3 | Mengelola Penyakit | Mengelola penyakit merupakan proses pengelolaan data penyakit yang meliputi dapat melihat data penyakit, dapat melakukan penambahan data penyakit, dapat melakukan update data penyakit, dapat menghapus data penyakit |
| 4 | Mengelola Rule | Mengelola rule merupakan proses pengelolaan data rule yang meliputi dapat melihat data rule, dapat melakukan penambahan data rule, dapat melakukan update data rule, dapat menghapus data rule |
| 5 | Mengelola Gejala | Mengelola gejala merupakan proses pengelolaan data gejala yang meliputi dapat melihat data gejala, dapat melakukan penambahan data gejala, dapat melakukan update data gejala, dapat menghapus data gejala |
| 6 | Mengelola pesan | Mengelola pesan merupakan proses pengelolaan pesan yang meliputi dapat melihat pesan, dapat menghapus pesan, mengubah pesan |
| 7 | Mengelola Riwayat | Mengelola riwayat merupakan proses pengelolaan riwayat yang meliputi dapat melihat detail riwayat, dapat menghapus riwayat |
| 8 | Mengirim pesan | Mengirim pesan merupakan proses pengiriman pesan yang meliputi dapat mengirim pesan |
| 9 | Melakukan Diagnosa | Melakukan Diagnosa merupakan proses melakukan diagnosa yang meliputi dapat memasukkan gejala, dapat melihat hasil diagnosa |
| 10 | Melihat Saran dan Penyakit | Melihat saran dan penyakit merupakan proses melihat saran dan penyakit yang meliputi dapat melihat saran, dapat melihat penyakit |
| 11 | Mengelola Profile | Mengelola profile merupakan proses pengelolaan profile yang meliputi dapat melihat profil, dapat melakukan penambahan data saran, dapat melakukan update profile, dapat menghapus profil |

**Skenario Usecase:**

**Tabel 3. Skenario Login**

|  |  |
| --- | --- |
| Usecase Skenario Login | |
| Aksi actor | Reaksi system |
| Skenario normal |  |
| 1. Masukkan username dan password |
|  | 2. Memeriksa valid dan tidak data yang dimasukkan dengan memeriksa ke basis data |
| 3. Masuk ke system |
| Skenario Alternatif |  |
| 4. Masukkan username dan password |
|  | 5. Memeriksa valid tidak data yang dimasukkan dengan memeriksa ke basis data |
| 6. Menampilkan notofikasi login gagal |
| 7. Masukkan username dan password yang valid |  |
|  | 8. Memeriksa valid tidak data yang dimasukkan dengan memerikksa ke basis data |
| 9. Masuk ke system |

**Tabel 4. Skenario Mengelola Saran**

|  |  |
| --- | --- |
| Usecase skenario Mengelola saran | |
| Aksi actor | Reaksi system |
| Skenario normal |  |
|  | 1. Memeriksa status login |
| 2. Membuka menu saran |  |
|  | 3. Mencari data saran yang akan dikelola |
| 4. Menampilkan data saran yang dicari (dengan tampilan belum detail |
| 5. Memilih data saran yang akan dikelola |  |
|  | 6. Menampilkan data saran yang dipilih(dengan tampilan belum detail) |
| 7. Mengelola data saran |  |
|  | 8. Memeriksa valid tidaknya data masukan |
| 9. Menyimpan data yang telah dikelola ke basis data |
| 10. Menampilkan pesan bahwa data sukses disimpan |
| Skenario Alternatif |  |
|  | 1. Memeriksa status login |
| 2. Memasukkan kata kunci |  |
|  | 3. Mencari data saran yang akan dikelola |
| 4. Menampilkan data saran yang dicari (dengan tampilan belum detail |
| 5. Memilih data saran yang akan dikelola |  |
|  | 6. Menampilkan data saran yang dipilih(dengan tampilan detail) |
| 7. Mengelola data saran |  |
|  | 8. Memeriksa valid tidaknya data masukan |
| 9.Menampilkan pesan bahwa data yang dimasukkan tidak valid |
| 10. Memasukkan data yang valid |  |
|  | 11. Memeriksa valid tidaknya data masukan |
| 12. Menyimpan data yang telah dikelola ke basis data |
| 13. Menampilkan pesan bahwa data sukse disimpan |

**Tabel 5. Skenario Mengelola Penyakit**

|  |  |
| --- | --- |
| Usecase skenario Mengelola Penyakit | |
| Aksi actor | Reaksi system |
| Skenario normal |  |
|  | 1. Memeriksa status login |
| 2. Membuka menu penyakit |  |
|  | 3. Mencari data penyakit yang akan dikelola |
| 4. Menampilkan data penyakit yang dicari (dengan tampilan belum detail) |
| 5. Memilih data penyakit yang akan dikelola |  |
|  | 6. Menampilkan data penyakit yang dipilih(dengan tampilan belum detail) |
| 7. Mengelola data penyakit |  |
|  | 8. Memeriksa valid tidaknya data masukan |
| 9. Menyimpan data yang telah dikelola ke basis data |
| 10. Menampilkan pesan bahwa data sukses disimpan |
| Skenario Alternatif |  |
|  | 1. Memeriksa status login |
| 2. Memasukkan kata kunci |  |
|  | 3. Mencari data penyakit yang akan dikelola |
| 4. Menampilkan data penyakit yang dicari (dengan tampilan belum detail) |
| 5. Memilih data penyakit yang akan dikelola |  |
|  | 6. Menampilkan data penyakit yang dipilih(dengan tampilan detail) |
| 7. Mengelola data penyakit |  |
|  | 8. Memeriksa valid tidaknya data masukan |
| 9.Menampilkan pesan bahwa data yang dimasukkan tidak valid |
| 10. Memasukkan data yang valid |  |
|  | 11. Memeriksa valid tidaknya data masukan |
| 12. Menyimpan data yang telah dikelola ke basis data |
| 13. Menampilkan pesan bahwa data sukse disimpan |

**Tabel 6. Skenario Mengelola Rule**

|  |  |
| --- | --- |
| Usecase skenario Mengelola Rule | |
| Aksi actor | Reaksi system |
| Skenario normal |  |
|  | 1. Memeriksa status login |
| 2. Membuka data rule |  |
|  | 3. Mencari data rule yang akan dikelola |
| 4. Menampilkan data rule yang dicari (dengan tampilan belum detail) |
| 5. Memilih data rule yang akan dikelola |  |
|  | 6. Menampilkan data rule yang dipilih(dengan tampilan belum detail) |
| 7. Mengelola data rule |  |
|  | 8. Memeriksa valid tidaknya data masukan |
| 9. Menyimpan data yang telah dikelola ke basis data |
| 10. Menampilkan pesan bahwa data sukses disimpan |
| Skenario Alternatif |  |
|  | 1. Memeriksa status login |
| 2. Memasukkan kata kunci |  |
|  | 3. Mencari data rule yang akan dikelola |
| 4. Menampilkan data rule yang dicari (dengan tampilan belum detail) |
| 5. Memilih data rule yang akan dikelola |  |
|  | 6. Menampilkan data rule yang dipilih(dengan tampilan detail) |
| 7. Mengelola data rule |  |
|  | 8. Memeriksa valid tidaknya data masukan |
| 9.Menampilkan pesan bahwa data yang dimasukkan tidak valid |
| 10. Memasukkan data yang valid |  |
|  | 11. Memeriksa valid tidaknya data masukan |
| 12. Menyimpan data yang telah dikelola ke basis data |
| 13. Menampilkan pesan bahwa data sukse disimpan |

**Tabel 7. Skenario Mengelola Gejala**

|  |  |
| --- | --- |
| Usecase skenario Mengelola Gejala | |
| Aksi actor | Reaksi system |
| Skenario normal |  |
|  | 1. Memeriksa status login |
| 2. Memasukkan kata kunci |  |
|  | 3. Mencari data gejala yang akan dikelola |
| 4. Menampilkan data gejala yang dicari (dengan tampilan belum detail) |
| 5. Memilih data gejala yang akan dikelola |  |
|  | 6. Menampilkan data gejala yang dipilih(dengan tampilan belum detail) |
| 7. Mengelola data gejala |  |
|  | 8. Memeriksa valid tidaknya data masukan |
| 9. Menyimpan data yang telah dikelola ke basis data |
| 10. Menampilkan pesan bahwa data sukses disimpan |
| Skenario Alternatif |  |
|  | 1. Memeriksa status login |
| 2. Memasukkan kata kunci |  |
|  | 3. Mencari data gejala yang akan dikelola |
| 4. Menampilkan data gejala yang dicari (dengan tampilan belum detail) |
| 5. Memilih data gejala yang akan dikelola |  |
|  | 6. Menampilkan data gejala yang dipilih(dengan tampilan detail) |
| 7. Mengelola data gejala |  |
|  | 8. Memeriksa valid tidaknya data masukan |
| 9.Menampilkan pesan bahwa data yang dimasukkan tidak valid |
| 10. Memasukkan data yang valid |  |
|  | 11. Memeriksa valid tidaknya data masukan |
| 12. Menyimpan data yang telah dikelola ke basis data |
| 13. Menampilkan pesan bahwa data sukse disimpan |

**Tabel 8. Skenario Mengelola Pesan**

|  |  |
| --- | --- |
| Usecase skenario Mengelola Pesan | |
| Aksi actor | Reaksi system |
| Skenario normal |  |
|  | 1. Memeriksa status login |
| 2. Membuka data pesan |  |
|  | 3. Mencari data pesan yang akan dikelola |
| 4. Menampilkan data pesan yang dicari (dengan tampilan belum detail) |
| 5. Memilih data pesan yang akan dikelola |  |
|  | 6. Menampilkan data pesan yang dipilih(dengan tampilan belum detail) |
| 7. Mengelola data pesan |  |
|  | 8. Memeriksa valid tidaknya data masukan |
| 9. Menyimpan data yang telah dikelola ke basis data |
| 10. Menampilkan pesan bahwa data sukses disimpan |
| Skenario Alternatif |  |
|  | 1. Memeriksa status login |
| 2. Membuka data pesan |  |
|  | 3. Mencari data pesan yang akan dikelola |
| 4. Menampilkan data pesan yang dicari (dengan tampilan belum detail) |
| 5. Memilih data rule yang akan dikelola |  |
|  | 6. Menampilkan data pesan yang dipilih(dengan tampilan detail) |
| 7. Mengelola data pesan |  |
|  | 8. Memeriksa valid tidaknya data masukan |
| 9.Menampilkan pesan bahwa data yang dimasukkan tidak valid |
| 10. Memasukkan data yang valid |  |
|  | 11. Memeriksa valid tidaknya data masukan |
| 12. Menyimpan data yang telah dikelola ke basis data |
| 13. Menampilkan pesan bahwa data sukse disimpan |

**Tabel 9. Skenario Mengelola Riwayat**

|  |  |
| --- | --- |
| Usecase skenario Mengelola Riwayat | |
| Aksi actor | Reaksi system |
| Skenario normal |  |
|  | 1. Memeriksa status login |
| 2. Memasukkan kata kunci |  |
|  | 3. Mencari data riwayat yang akan dikelola |
| 4. Menampilkan data riwayat yang dicari (dengan tampilan belum detail) |
| 5. Memilih data riwayat yang akan dikelola |  |
|  | 6. Menampilkan data riwayat yang dipilih(dengan tampilan belum detail) |
| 7. Mengelola data riwayat |  |
|  | 8. Memeriksa valid tidaknya data masukan |
| 9. Menyimpan data yang telah dikelola ke basis data |
| 10. Menampilkan pesan bahwa data sukses disimpan |
| Skenario Alternatif |  |
|  | 1. Memeriksa status login |
| 2. Memasukkan kata kunci |  |
|  | 3. Mencari data riwayat yang akan dikelola |
| 4. Menampilkan data riwayat yang dicari (dengan tampilan belum detail) |
| 5. Memilih data riwayat yang akan dikelola |  |
|  | 6. Menampilkan data riwayat yang dipilih(dengan tampilan detail) |
| 7. Mengelola data riwayat |  |
|  | 8. Memeriksa valid tidaknya data masukan |
| 9.Menampilkan pesan bahwa data yang dimasukkan tidak valid |
| 10. Memasukkan data yang valid |  |
|  | 11. Memeriksa valid tidaknya data masukan |
| 12. Menyimpan data yang telah dikelola ke basis data |
| 13. Menampilkan pesan bahwa data sukse disimpan |

**Tabel 10. Skenario Melakukan Diagnosa**

|  |  |
| --- | --- |
| Usecase skenario Melakukan Diagnosa | |
| Aksi actor | Reaksi system |
| Skenario normal |  |
|  | 1. Memeriksa status login |
| 2. Memilih Menu diagnosa |  |
|  | 3. Menampilkan menu diagnosa |
| 4. Memilih gejala |  |
|  | 5. Proses perhitungan bayes |
| 6. Menampilkan hasil diagnosa |
| 7. Melihat hasil diagnosa |  |
|  | 8. Menyimpan data yang telah dikelola ke basis data |
| Skenario Alternatif |  |
|  | 1. Memeriksa status login |
| 2. Memilih menu diagnosa |  |
|  | 3. Menampilkan menu diagnosa |
| 4. Memilih gejala |  |
|  | 5. Menampilkan data gejala yang dipilih(dengan tampilan belum detail |
| 6.Mengelola data gejala |  |
|  | 7. Memeriksa valid tidaknya data masukan |
| 8.Menampilkan pesan bahwa data yang dimasukkan tidak valid |
| 9. Memasukkan data yang valid |  |
|  | 10. Memeriksa valid tidaknya data masukan |
| 11. Menyimpan data yang telah dikelola ke basis data |
| 12. Menampilkan pesan bahwa data sukse disimpan |

**Tabel 11. Skenario Megelola Profile**

|  |  |
| --- | --- |
| Usecase skenario Mengelola Profile | |
| Aksi actor | Reaksi system |
| Skenario normal |  |
|  | 1. Memeriksa status login |
| 2. Membuka menu profile |  |
|  | 3. Proses select data profile |
| 4. Menampilkan data profile(dengan tampilan belum detail |
| 5. Memilih data profil yang akan dikelola |  |
|  | 6. Menampilkan data profil yang dipilih(dengan tampilan belum detail) |
| 7. Memeriksa valid tidaknya data masukan |
| 8. Menyimpan data yang telah dikelola ke basis data |
| 9. Menampilkan pesan bahwa data sukses disimpan |
|  |
| Skenario Alternatif |  |
|  | 1. Memeriksa status login |
| 2. Membuka menu profile |  |
|  | 3. Proses select data profile |
| 4. Menampilkan data profil (dengan tampilan belum detail) |
| 5. Memilih data profil yang akan dikelola |  |
|  | 6. Menampilkan data profil yang dipilih(dengan tampilan detail) |
| 7. Mengelola data profil |  |
|  | 8. Memeriksa valid tidaknya data masukan |
| 9.Menampilkan pesan bahwa data yang dimasukkan tidak valid |
| 10. Memasukkan data yang valid |  |
|  | 11. Memeriksa valid tidaknya data masukan |
| 12. Menyimpan data yang telah dikelola ke basis data |
| 13. Menampilkan pesan bahwa data sukse disimpan |

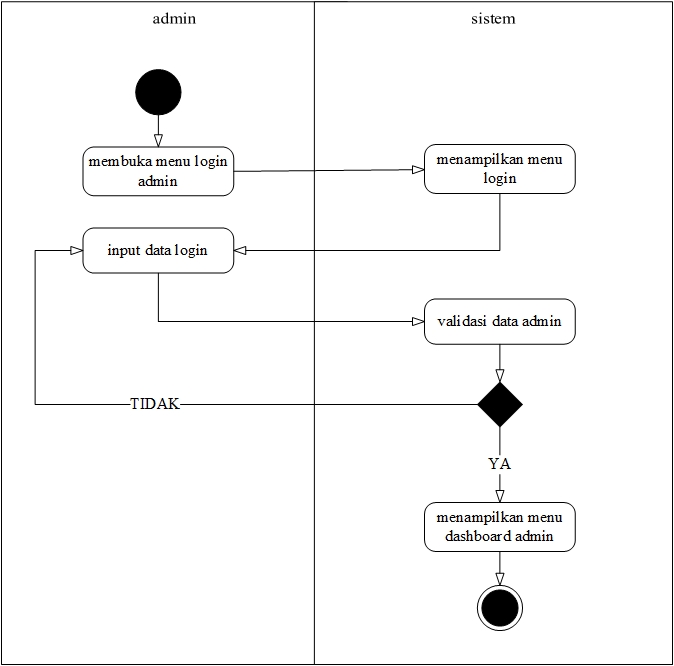
**Tabel 12. Skenario Melihat penyakit dan saran**

|  |  |
| --- | --- |
| Usecase skenario Melihat penyakit dan saran | |
| Aksi actor | Reaksi system |
| Skenario normal |  |
|  | 1. Memeriksa status login |
| 2. Memilih Menu penyakit |  |
|  | 3. Menampilkan menu penyakit where user |
| 4.Menampilkan data riwayat user |
| 5.Melihat data riwayat user |  |
| 6.Mengelola data riwayat user |
|  | 7. Menyimpan data yang telah dikelola ke basis data |
|  |
| Skenario Alternatif |
|  | 1. Memeriksa status login |
| 2. Memilih menu penyakit |  |
|  | 3. Menampilkan menu penyakit where user |
| 4. Menampilkan data riwayat user |
| 5. Melihat data riwayat user |  |
| 6.Mengelola data riwayat user |  |
|  | 7. Memeriksa valid tidaknya data masukan |
| 8.Menampilkan pesan bahwa data yang dimasukkan tidak valid |
| 9. Memasukkan data yang valid |  |
|  | 10. Memeriksa valid tidaknya data masukan |
| 11. Menyimpan data yang telah dikelola ke basis data |
| 12. Menampilkan pesan bahwa data sukse disimpan |

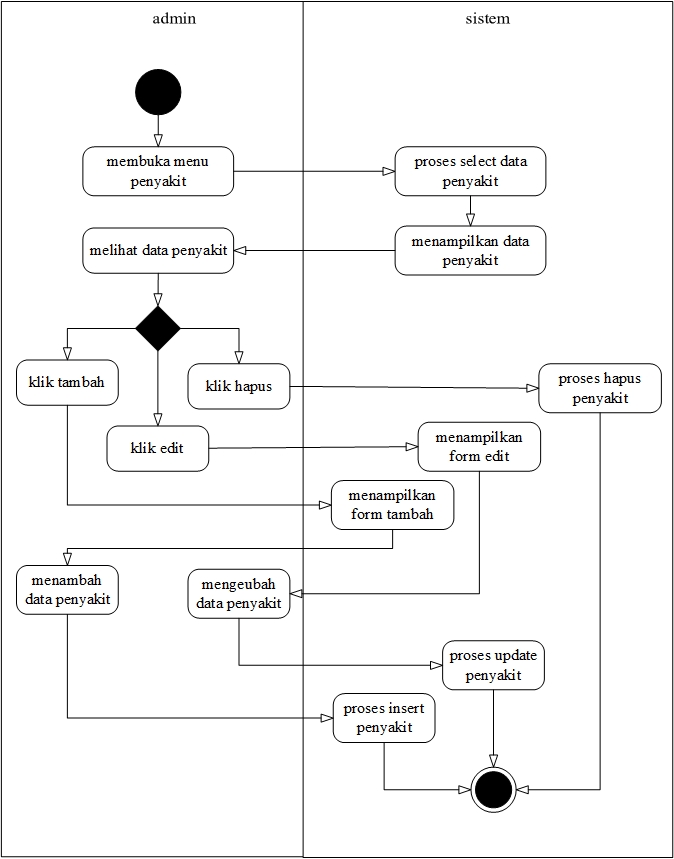
**Tabel 13. Skenario Mengirim Pesan**

|  |  |
| --- | --- |
| Usecase skenario Mengirim Pesan | |
| Aksi actor | Reaksi system |
| Skenario normal |  |
|  | 1. Memeriksa status login |
| 2. Memilih menu pesan |  |
|  | 3. Menampilkan form kirim pesan |
| 4. Input form kirim pesan |  |
|  | 5.Menampilkan pesan yang ditulis |
| 6. Memilih kirim |  |
|  | 7.Menyimpan pesan ke basis data |
|  | 8.Menampilkan pesan bahwa pesan sukses dikirim |
| Skenario Alternatif |  |
|  | 1. Memeriksa status login |
| 2. Memilih menu pesan |  |
|  | 3. Menampilkan form kirim pesan |
| 4. Input form kirim pesan |  |
|  | 5. Memeriksa valid tidaknya data masukan |
|  | 6. Menampilkan pesan bahwa data tidak valid |
| 7. Mengelola menu pesan |  |
|  | 8. Memeriksa valid tidaknya data masukan |
|  | 9.Menampilkan pesan bahwa data yang dimasukkan tidak valid |
| 10. Memasukkan form yang valid |  |
|  | 11. Memeriksa valid tidaknya data masukan |
| 12. Menyimpan data yang telah dikelola ke basis data |
| 13. Menampilkan pesan bahwa data sukse disimpan |

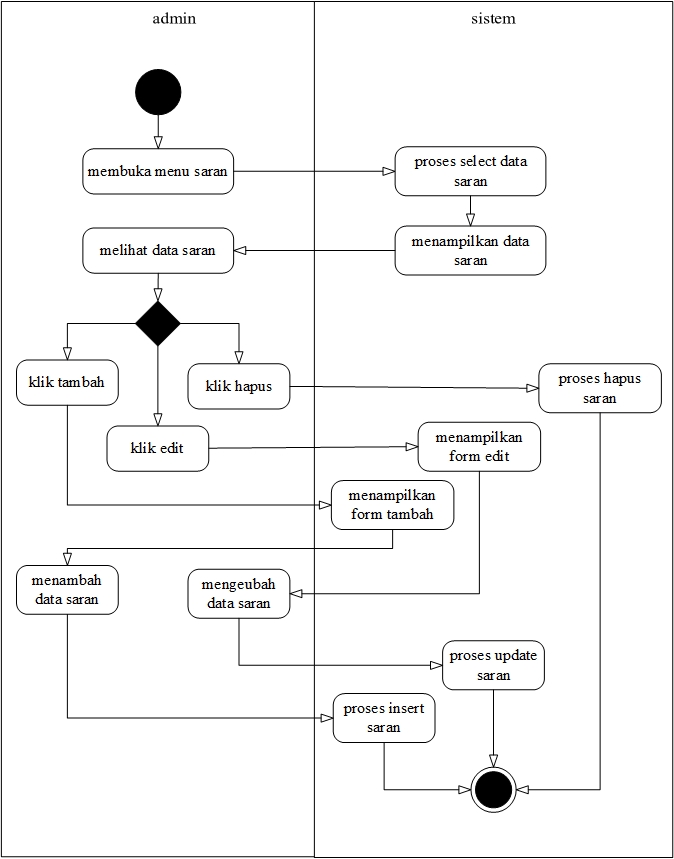
1. **Activity Diagram**
2. **Aktivitas Admin**

****

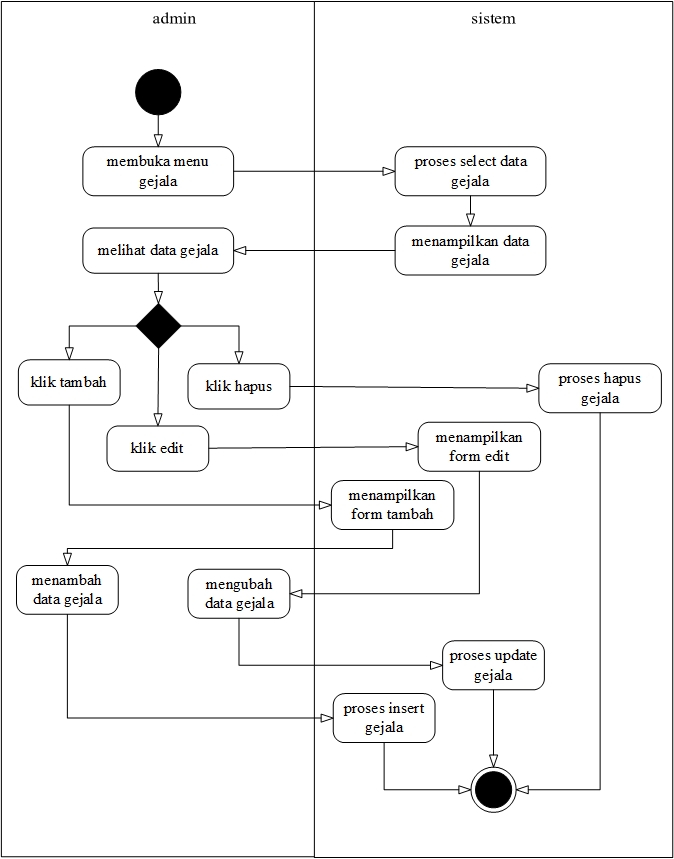
**Gambar 2. Activity Admin Login**

****

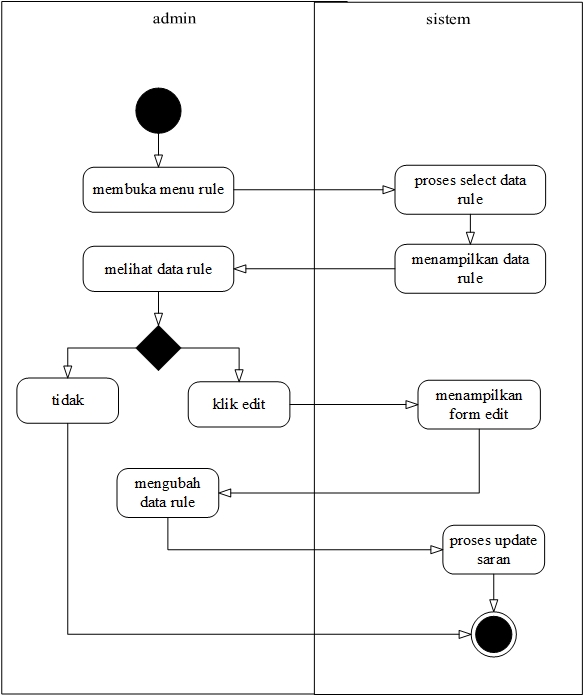
**Gambar 3. Activity Kelola Penyakit**

****

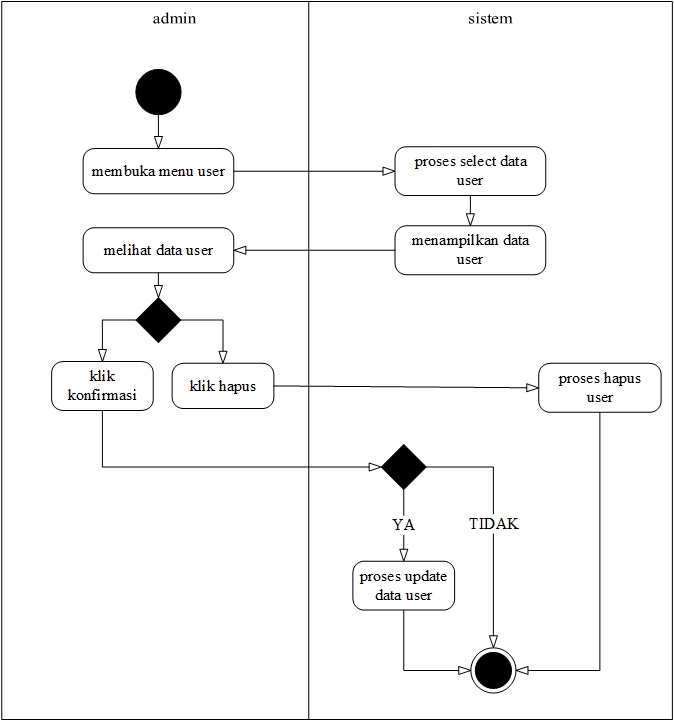
**Gambar 4. Activity Kelola Saran Penyakit**

****

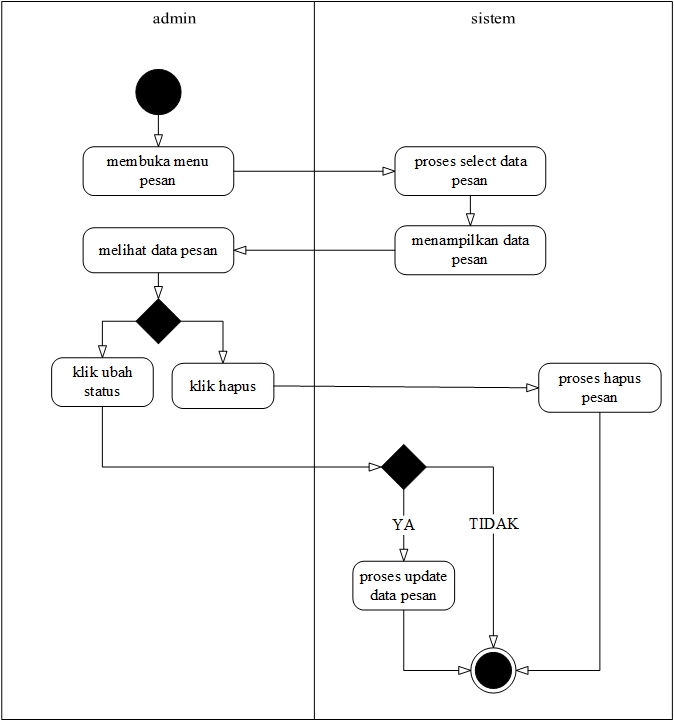
**Gambar 5. Activity Kelola Gejala**

****

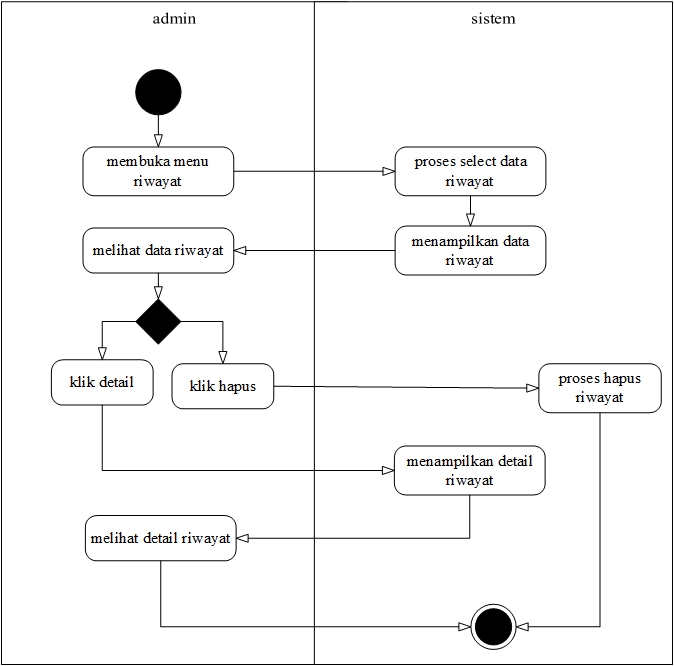
**Gambar 6. Activity Kelola Rule**

****

**Gambar 7. Activity Kelola User**

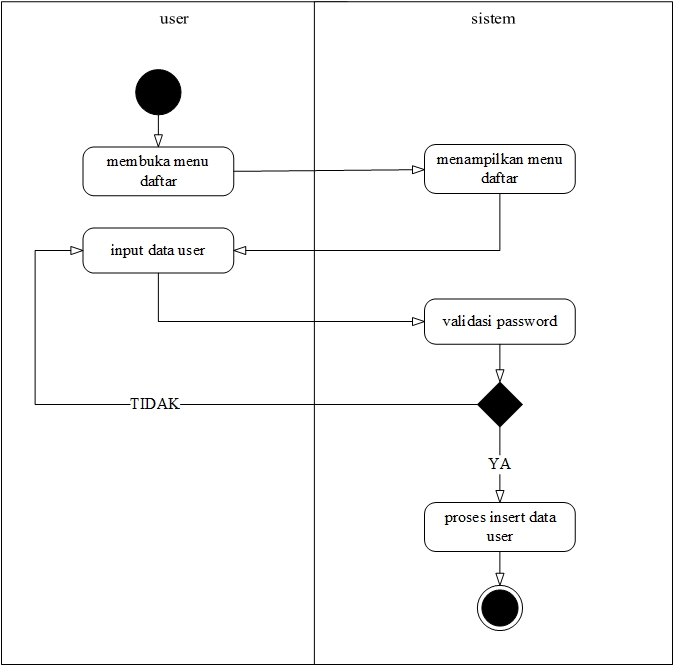
****

**Gambar 8. Activity Kelola Pesan**

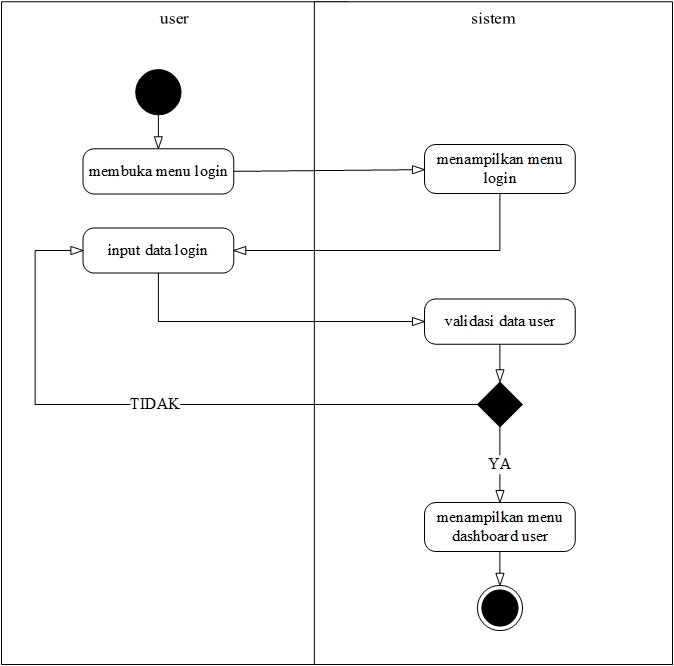
****

**Gambar 9. Activity Kelola Riwayat**

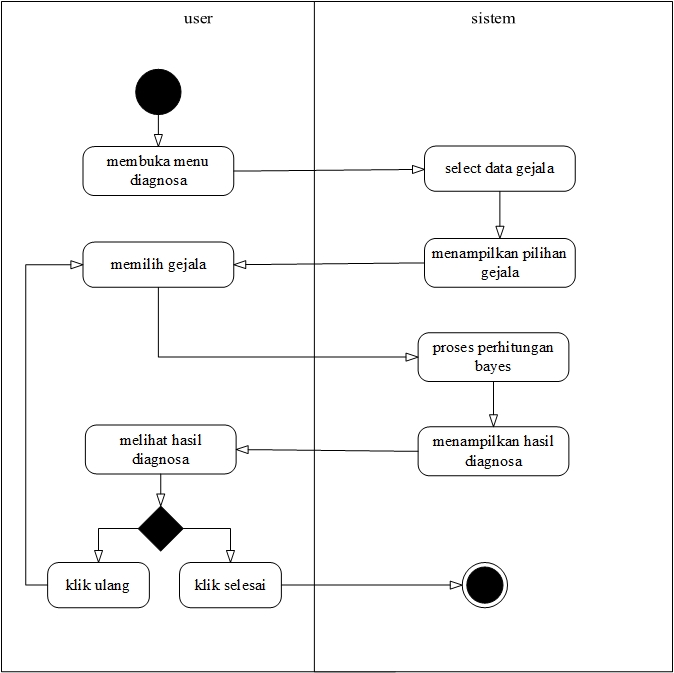
1. **Aktivitas User**

****

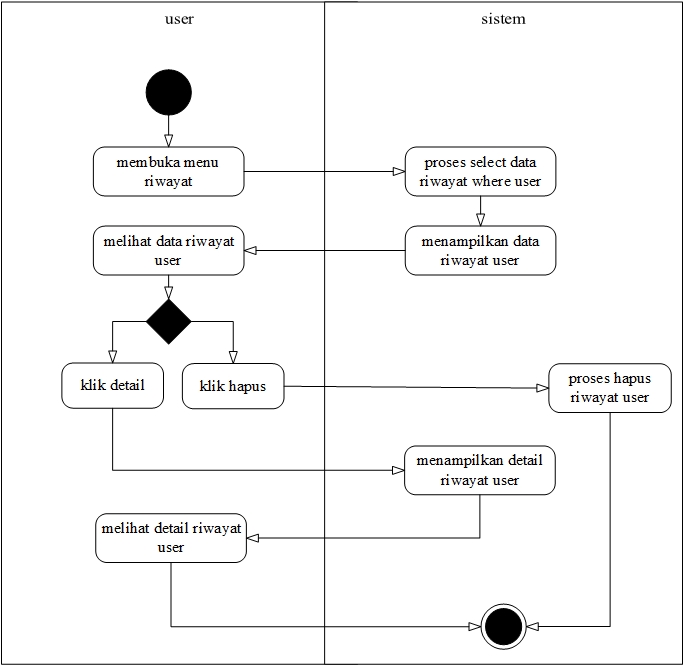
**Gambar 10. Activity Pendaftaran User**

****

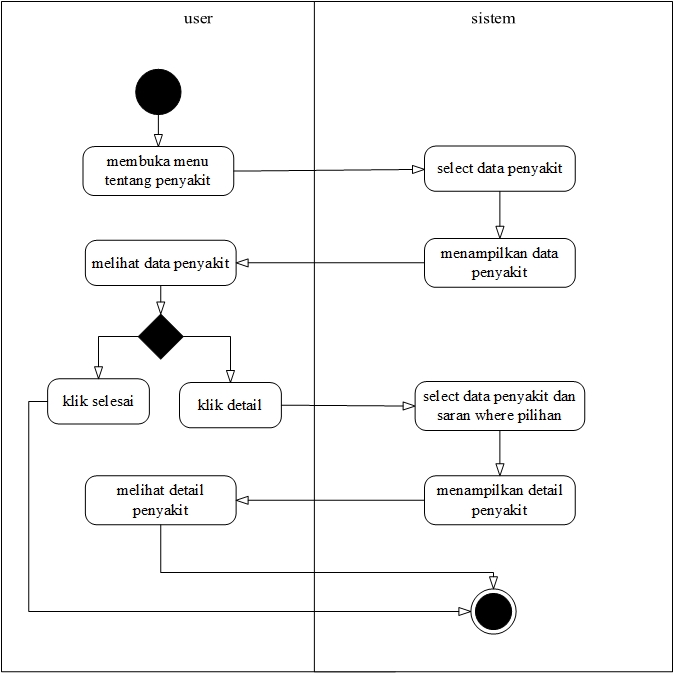
**Gambar 1. Activity User Login**

****

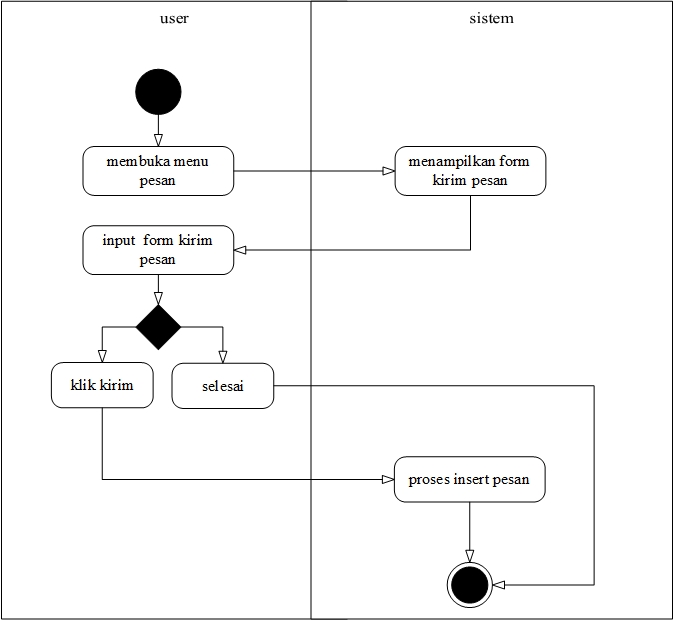
**Gambar 12. Activity Diagnosa Penyakit**

****

**Gambar 13. Activity Riwayat Diagnosa**

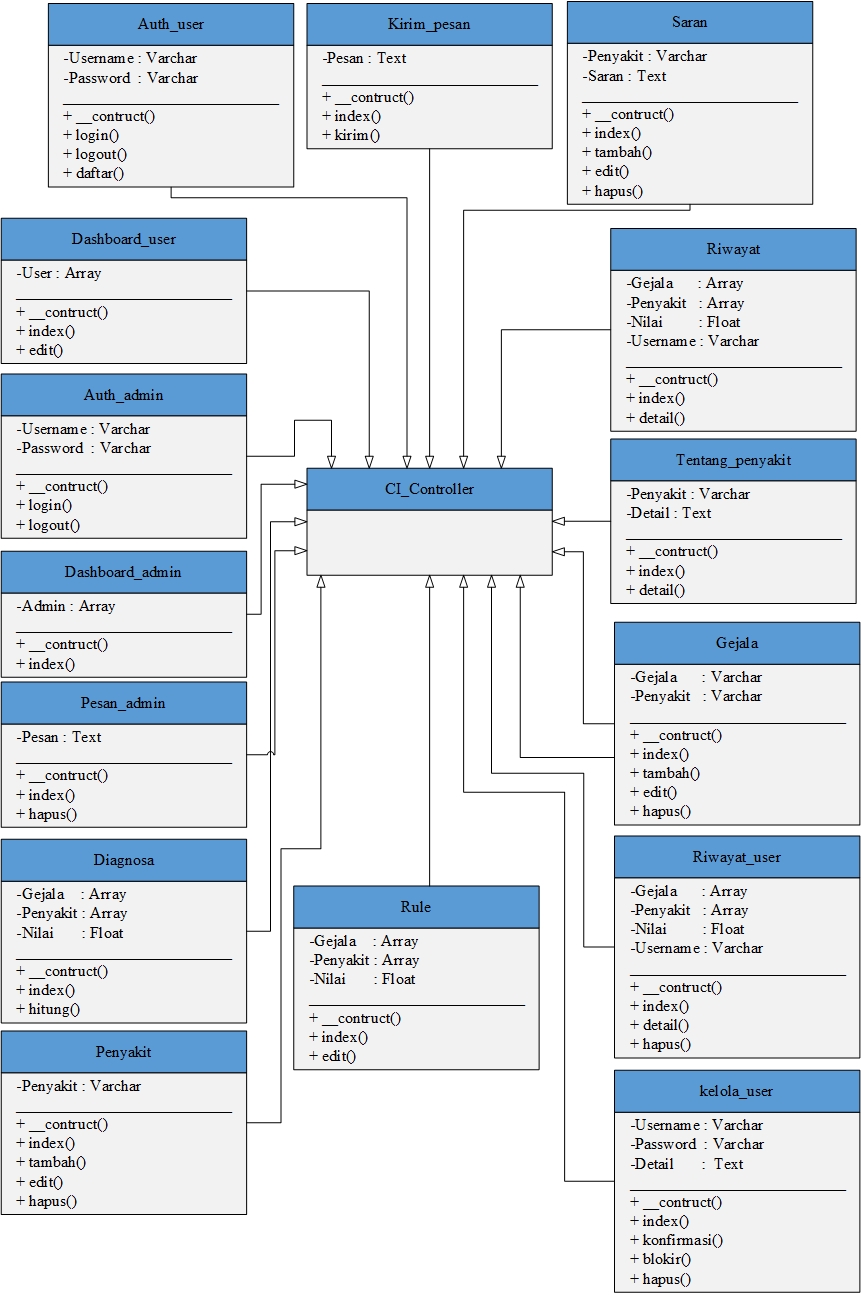
****

**Gambar 14. Activity Saran Penyakit**

****

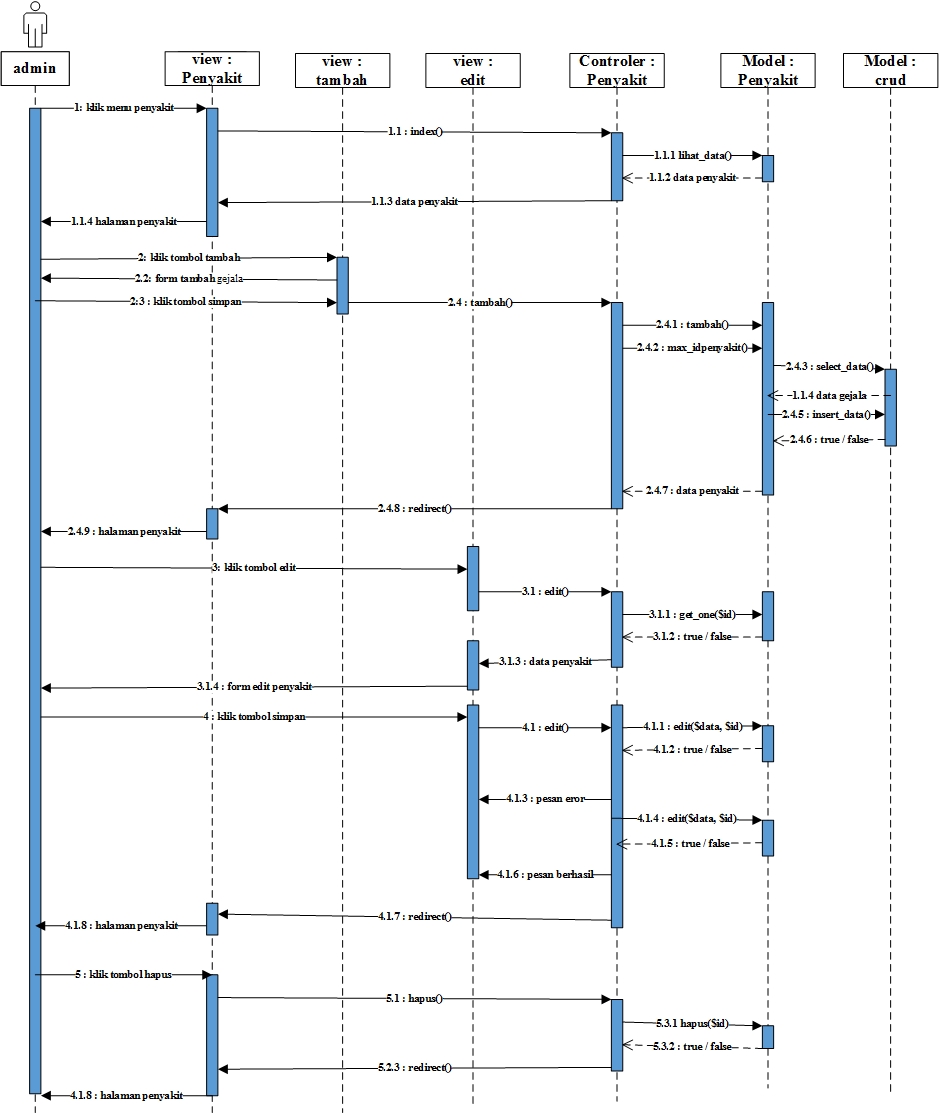
**Gambar 15. Activity Kirim Pesan**

1. **Class Diagram**
2. **Class Controler**

****

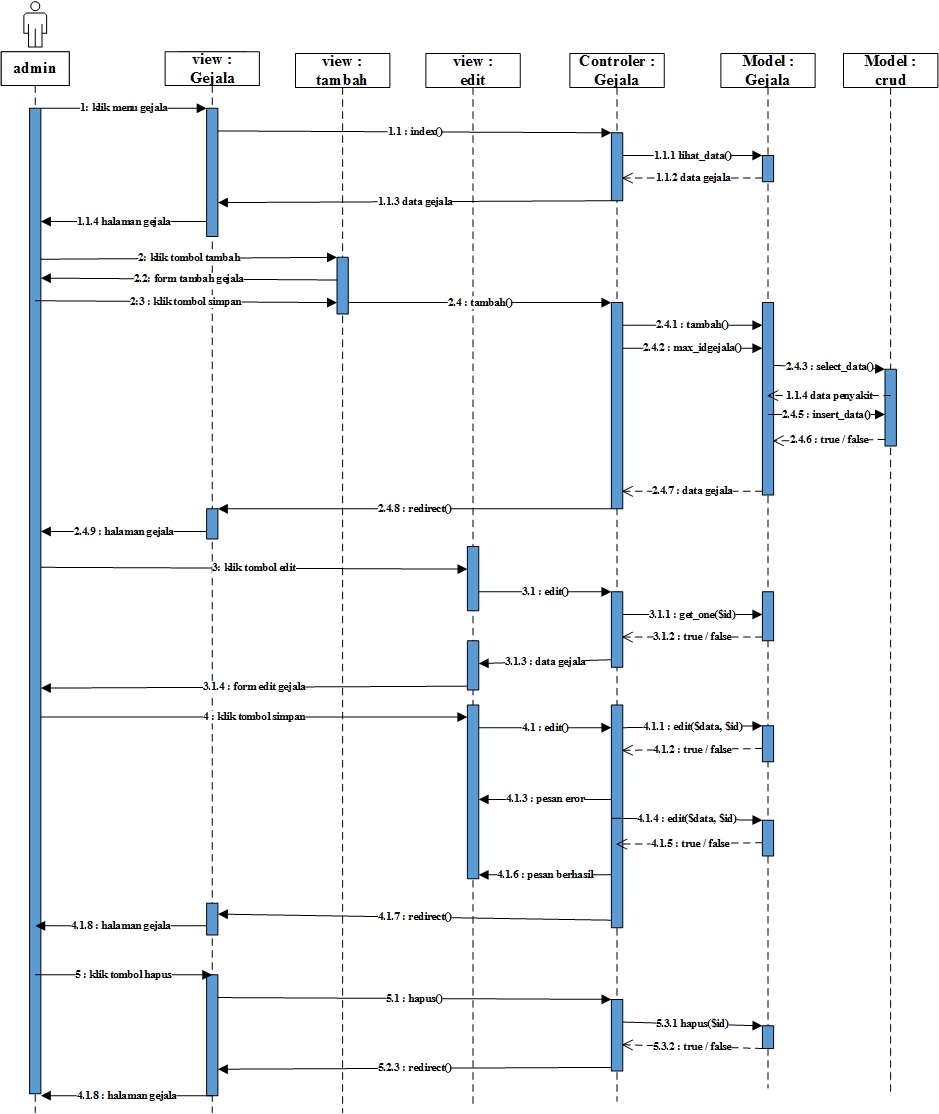
**Gambar 16. Class Controler**

1. **Squence Diagram**
2. **Squence Kelola Penyakit**

****

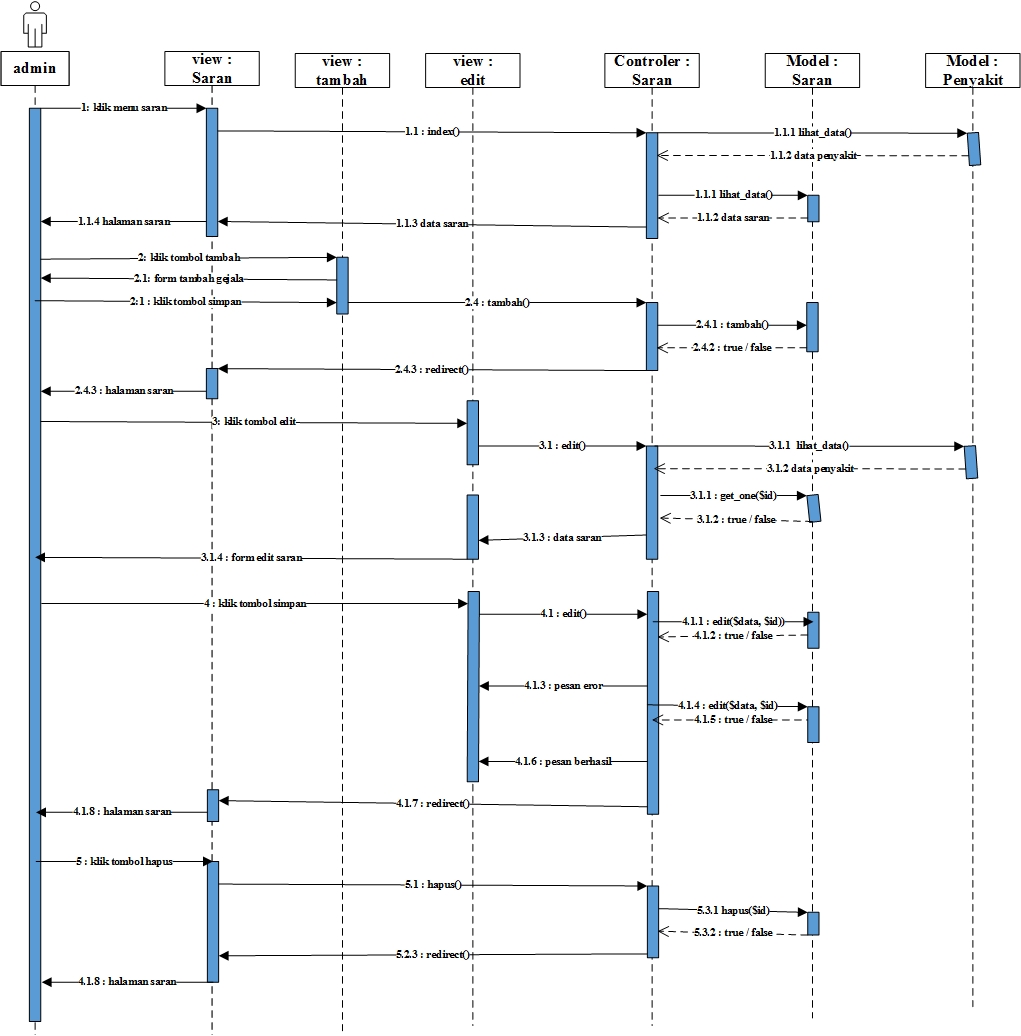
**Gambar 17. Squence Kelola Penyakit**

1. **Squence Kelola Gejala**

****

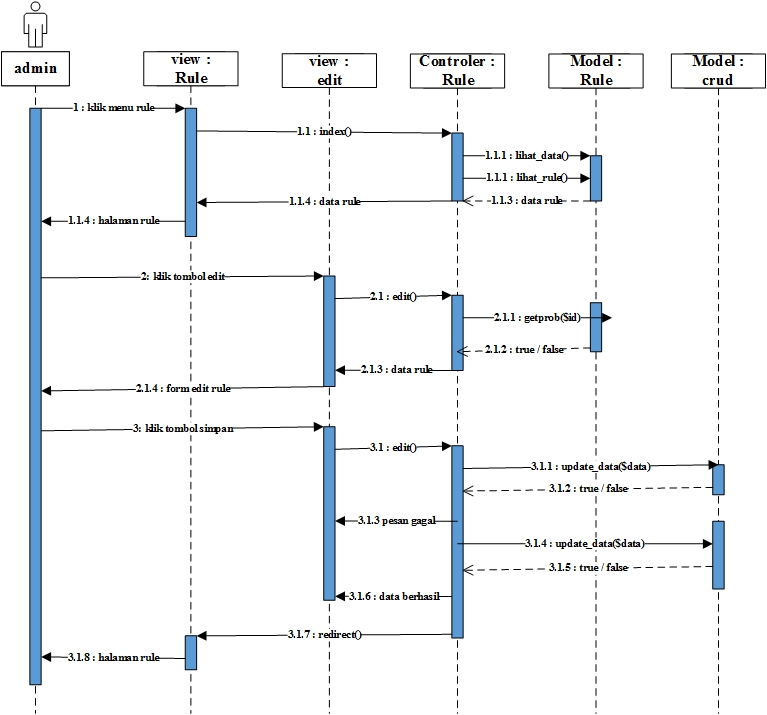
**Gambar 18. Squence Kelola Gejala**

1. **Squence Kelola Saran**

****

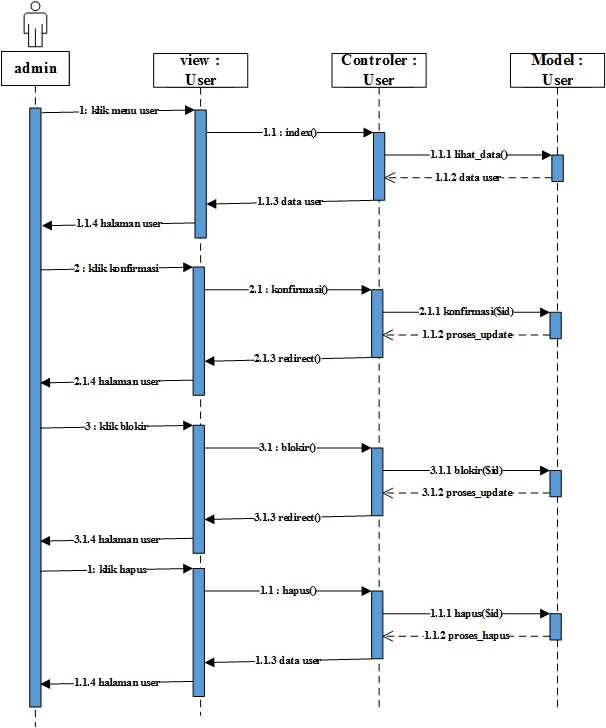
**Gambar 19. Squence Kelola Saran**

1. **Squence Kelola Rule**

****

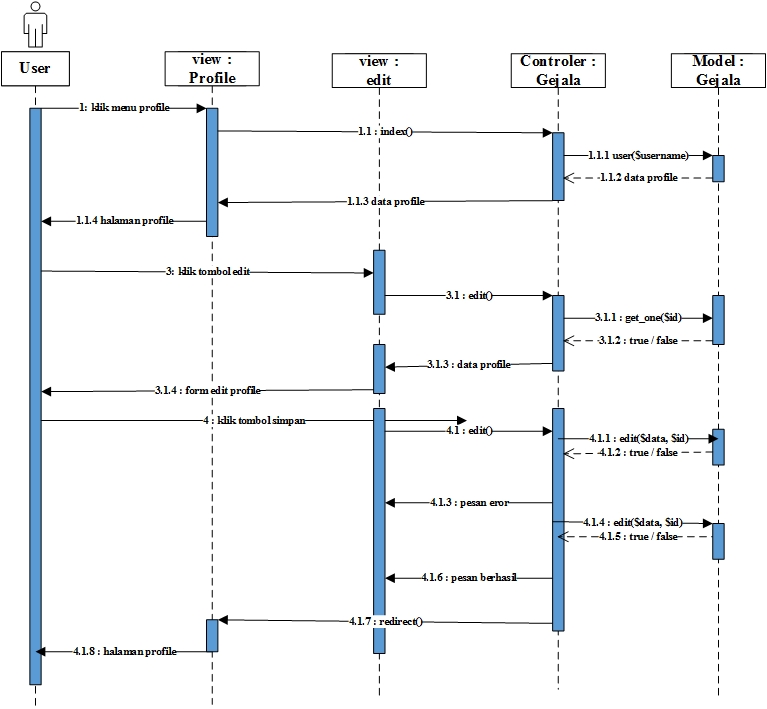
**Gambar 20. Squence Kelola Rule**

1. **Squence Kelola User**

****

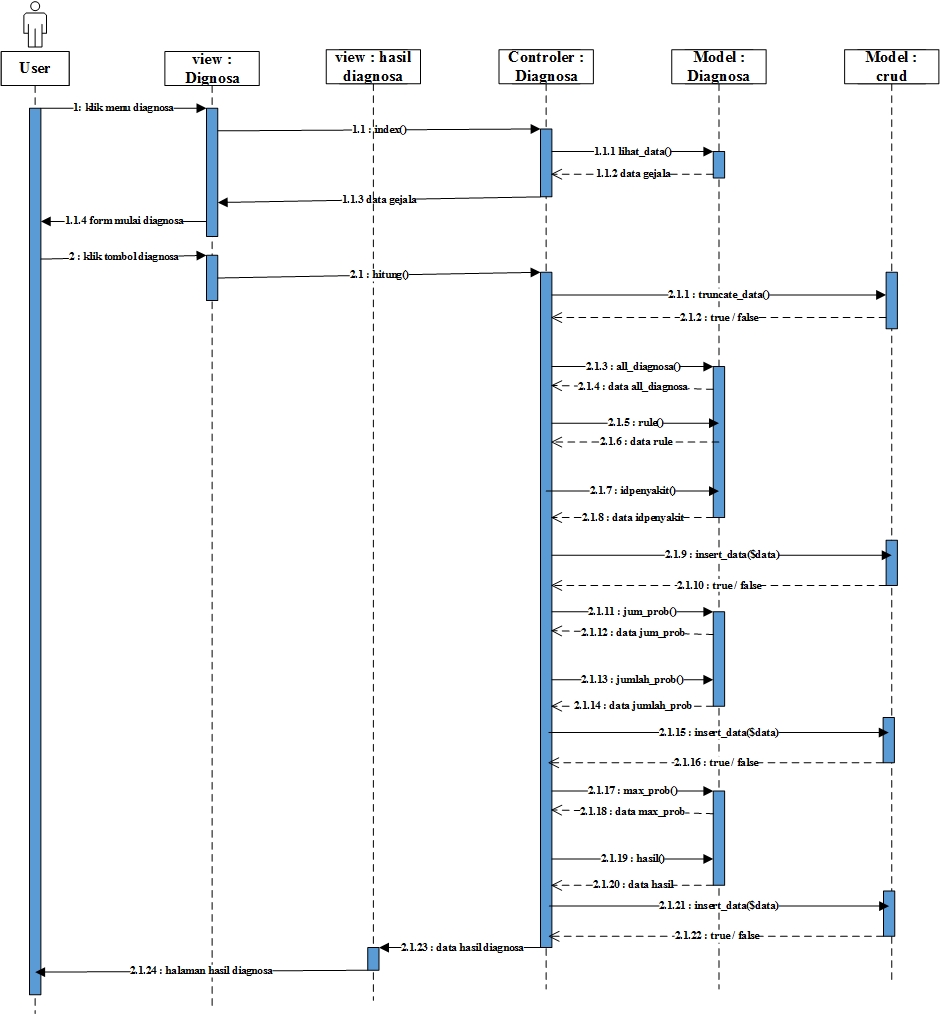
**Gambar 21. Squence Kelola User**

1. **Squence Profile User**

****

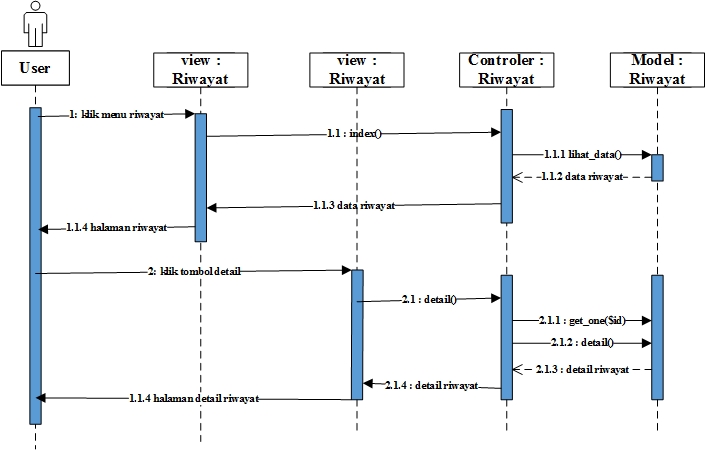
**Gambar 22. Squence Kelola Profile User**

1. **Squence Diagnosa**

****

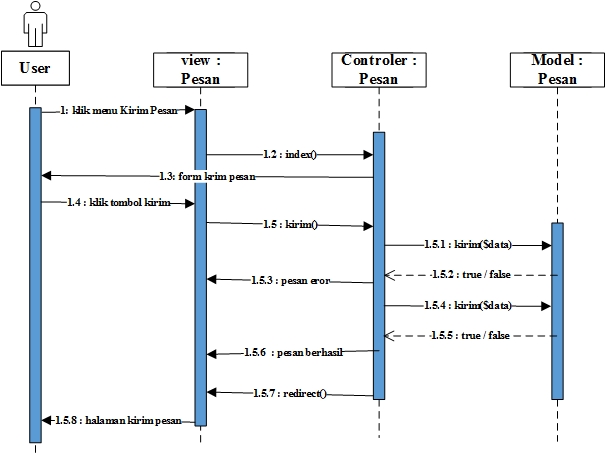
**Gambar 23. Squence Diagnosa**

1. **Squence Riwayat**

****

**Gambar 24. Squence Riwayat**

1. **Squence Kirim Pesan**

****

**Gambar 25. Squence Kirim Pesan**

**BAB III**

**PENUTUP**

* + 1. **Kesimpulan**

Setelah melalui tahap pengujian pada sistem pakar diagnosa penyakit tanaman nanas menggunakan UML (*Unified Modelling Language*), maka dapat diambil beberapa kesimpulan antara lain :

1. Telah berhasil dirancang sebuah sistem informasi untuk mendiagnosa penyakit tanaman buah nanas berserta saran atau solusi penanggulangan penyakit tersebut.
2. Sistem informasi diagnosa penyakit tanaman buah nanas dini dirancang menggunakan UML (*Unified Modelling Language*) dengan *usecase diagram, acktivity diagram, class diagram dan sequence diagram* untuk memvisualisasikan sistem yang akan dibuat

**DAFTAR PUSTAKA**

[1] Budi Samadi. 2013. Penentuan Untuk dan Budidaya Nanas Sistem Organik. Jakarta. Rineka Cipta.

[2] Tim Karya Tani Mandiri. 2011. Pedoman Bertanam Buah Nanas. Jakarta. Ghalia Indonesia.

[3] Kusrini. 2016. Sistem Pakar Teori Dan Aplikasi. Yogyakarta. Andi Offset.

[4] Munawar. 2005. Pemodelan Visual Dengan UML.Yogyakarta. Graha Ilmu.