# BAB IV

# HASIL DAN PEMBAHASAN

## 4.1 Perancangan Sistem

Perancangan merupakan suatu bagian dari metodologi pengembangan sistem untuk membangun perangkat lunak berupa tahapan-tahapan yang bertujuan untuk memberikan gambaran sistem secara terperinci.

Perancangan sistem juga dapat dikatakan sebagai tahapan dari siklus pengembangan sistem yang didefinisikan dari kebutuhan-kebutuhan fungsional dan persiapan untuk merancang, membangun, dan mengimplementasikan yang akan dijadikan gambaran bagaimana suatu sistem dibentuk, yang dapat berupa penggambaran, perancangan, dan pembuatan sketsa atau pengaturan dari beberapa elemen terpisah yang kemudian akan disatukan menjadi satu kesatuan yang utuh dan berfungsi dengan baik, serta menyangkut konfigurasi dari komponen-komponen perangkat keras dan juga perangkat lunak dari suatu sistem. Di bawah ini dijelaskan secara rinci tentang perancangan sistem yang diusulkan tersebut.

### 4.1.1 Tujuan Perancangan Sistem

Tujuan utama dari perancangan sistem baru yang diusulkan adalah untuk menyelesaikan masalah-masalah yang ada pada sistem yang sedang berjalan. Pada perancangan sistem ini juga dipaparkan secara rinci bagaimana sistem ini dirancang, prosedur-prosedur yang diusulkan sehingga dapat menggambarkan sistem yang diusulkan.

### 4.1.2 Gambaran Umum Sistem Yang Diusulkan

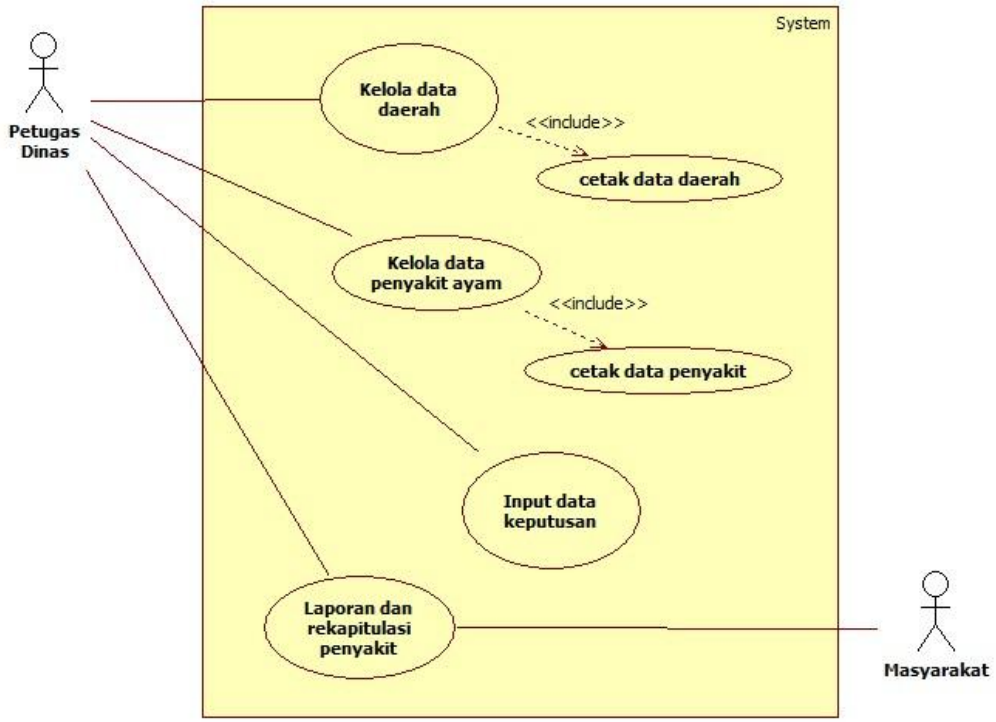
Sistem Informasi Geografis untuk memantau lokasi penyebaran penyakit ayam di Kota Bandung yang diusulkan merupakan sistem yang didukung dengan sebuah aplikasi berbasis *website online*. Pemilihan tipe aplikasi tersebut berdasarkan atas kebutuhan *user* untuk dapat mengakses secara 24 jam penuh. Sistem informasi geografis ini dapat digambarkan sebagai suatu bentuk pembelajaran yang dapat menjawab masalah atas kurangnya pengetahuan masyarakat mengenai penyakit ayam dan lokasi penyebarannya, diharapkan sistem baru yang diusulkan dapat berjalan lebih efektif.

### 4.1.3 Perancangan Prosedur Yang Diusulkan

Perancangan prosedur merupakan tahapan awal saat akan membuat sistem, di mana prosedur yang akan diusulkan ini dapat dilihat melalui proses-proses yang akan dilakukan ataupun dibutuhkan saat pembuatan sistem berlangsung.

#### 4.1.3.1 *Use Case Diagram*

*Use case diagram* adalah abstraksi dari interaksi antara *system* dan *actor*. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata *user*. *Use case diagram* sistem informasi geografis untuk memantau lokasi penyebaran penyakit ayam dapat dilihat pada gambar 4.1.




**Gambar 4.1** *Use Case Diagram* Sistem Informasi Geografis Penyakit Ayam

#### 4.1.3.1.1 *Scenario Use Case*

*Actor* mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya, menerima dan memberi informasi pada sistem. Adapun *actor* yang terlibat di dalam sistem informasi geografis untuk memantau lokasi penyebaran penyakit ayam hanyalah *user*.

1. Skenario Menentukan Daerah

**Tabel 4.1** Skenario Menentukan Daerah

| Identifikasi | |
|---|---|
| Nama | Menentukan daerah perkecamatan |
| Aktor | Admin |
| Deskripsi | Menggambarkan proses penambahan data daerah |
| Skenario Utama | |
| Aksi Aktor | Reaksi Sistem |
| 1. Memilih menu kelola data daerah | 2. Menampilkan menu input data daerah |
| 3. Input data daerah | 4. Update data daerah |
| 5. Lihat data daerah | 6. Menampilkan detail data daerah |

2. Skenario Kelola Data Penyakit

**Tabel 4.2** Skenario Kelola Data Penyakit

| Identifikasi | |
|---|---|
| Nama | Kelola Data Penyakit |
| Aktor | Admin |
| Deskripsi | Menggambarkan proses input data penyakit |
| Skenario Utama | |
| Aksi Aktor | Reaksi Sistem |
| 1. Memilih menu kelola data penyakit | 2. Menampilkan menu input data penyakit |
| 3. Input data penyakit | 4. Update data penyakit |
| 5. Lihat data penyakit | 6. Menampilkan detail data penyakit |

3. Skenario Pengambilan Keputusan

**Tabel 4.3** Skenario Pengambilan Keputusan

| Identifikasi | |
|---|---|
| Nama | Kelola Data Penyakit |
| Aktor | Admin |
| Deskripsi | Menggambarkan proses input data keputusan |
| Skenario Utama | |
| Aksi Aktor | Reaksi Sistem |
| 1. Memilih menu kelola data keputusan | 2. Menampilkan menu input data keputusan |
| 3. Input data keputusan | 4. Update data keputusan |
| 5. Lihat data keputusan | 6. Menampilkan detail data keputusan |

4. Skenario Laporan Penyakit Ayam

**Tabel 4.4** Skenario Laporan Penyakit Ayam

| Identifikasi | |
|---|---|
| Nama | Laporan penyakit ayam |
| Aktor | Admin |
| Deskripsi | Meliputi proses yang menggambarkan kejadian menampilkan laporan penyakit ayam dalam bentuk peta dan keputusan |
| Skenario Utama | |
| Aksi Aktor | Reaksi Sistem |
| 1. Memilih menu laporan | 2. Menampilkan data laporan penyakit ayam dalam bentuk peta |
| 3. Memilih kriteria meliputi bulan, tahun dan nama penyakit | 4. Menampilkan data laporan penyakit dalam bentuk peta dan keputusan berdasarkan kriteria |

| | |
|---|---|
| 5. Lihat rekapitulasi penyakit | 6. Menampilkan rekapitulasi detail penyakit yang dipilih |

5. Skenario Laporan Penyakit Ayam

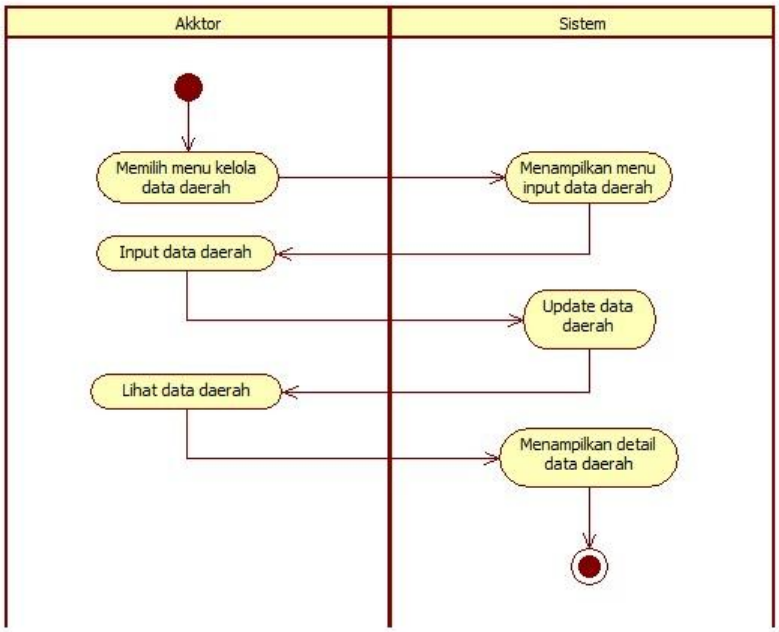
**Tabel 4.5** Skenario Laporan Penyakit Ayam

| Identifikasi | |
|---|---|
| Nama | Laporan penyakit ayam |
| Aktor | *User* |
| Deskripsi | Meliputi proses yang menggambarkan kejadian menampilkan laporan penyakit ayam dalam bentuk peta dan keputusan |
| Skenario Utama | |
| Aksi Aktor | Reaksi Sistem |
| 1. Memilih menu laporan | 2. Menampilkan data laporan penyakit ayam dalam bentuk peta |
| 3. Memilih kriteria meliputi bulan, tahun dan nama penyakit | 4. Menampilkan data laporan penyakit ayam dalam bentuk peta dan keputusan berdasarkan kriteria |
| 5. Lihat rekapitulasi penyakit | 6. Menampilkan rekapitulasi detail penyakit yang dipilih. |

### 4.1.3.2 Activity Diagram

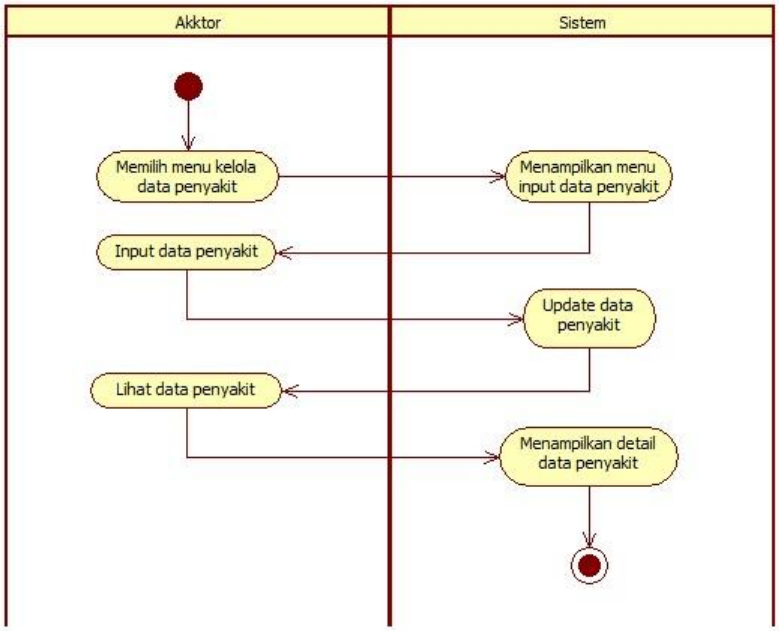
*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Berikut adalah *activity diagram* yang telah mengacu pada setiap skenario *use case* yang telah dibuat sebelumnya.

1. *Activity Diagram* Menentukan Daerah



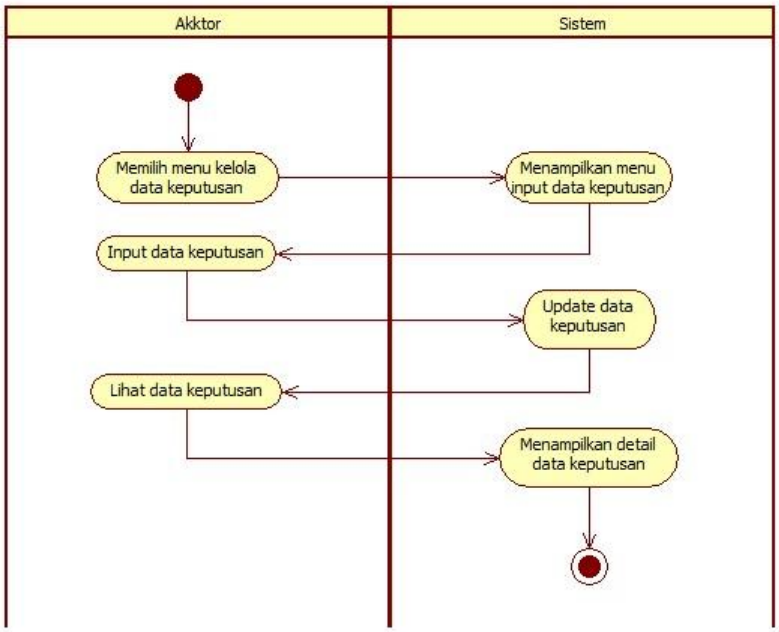

**Gambar 4.2** *Activity Diagram* Menentukan Daerah

2. *Activity Diagram* Kelola Data Penyakit



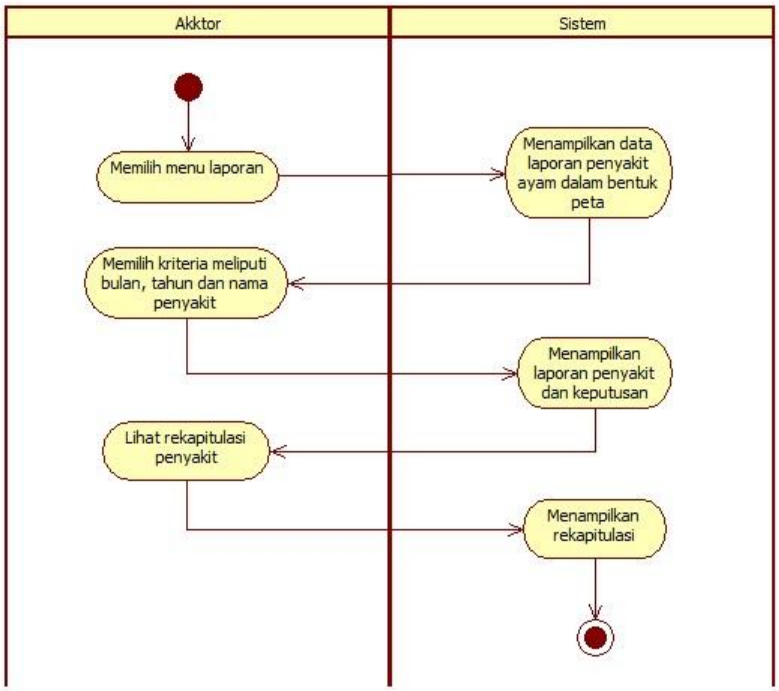

**Gambar 4.3** *Activity Diagram* Kelola Data Penyakit

3. *Activity Diagram* Pengambilan Keputusan




**Gambar 4.4** *Activity Diagram* Pengambilan Keputusan

4. *Activity Diagram* laporan Penyakit Ayam

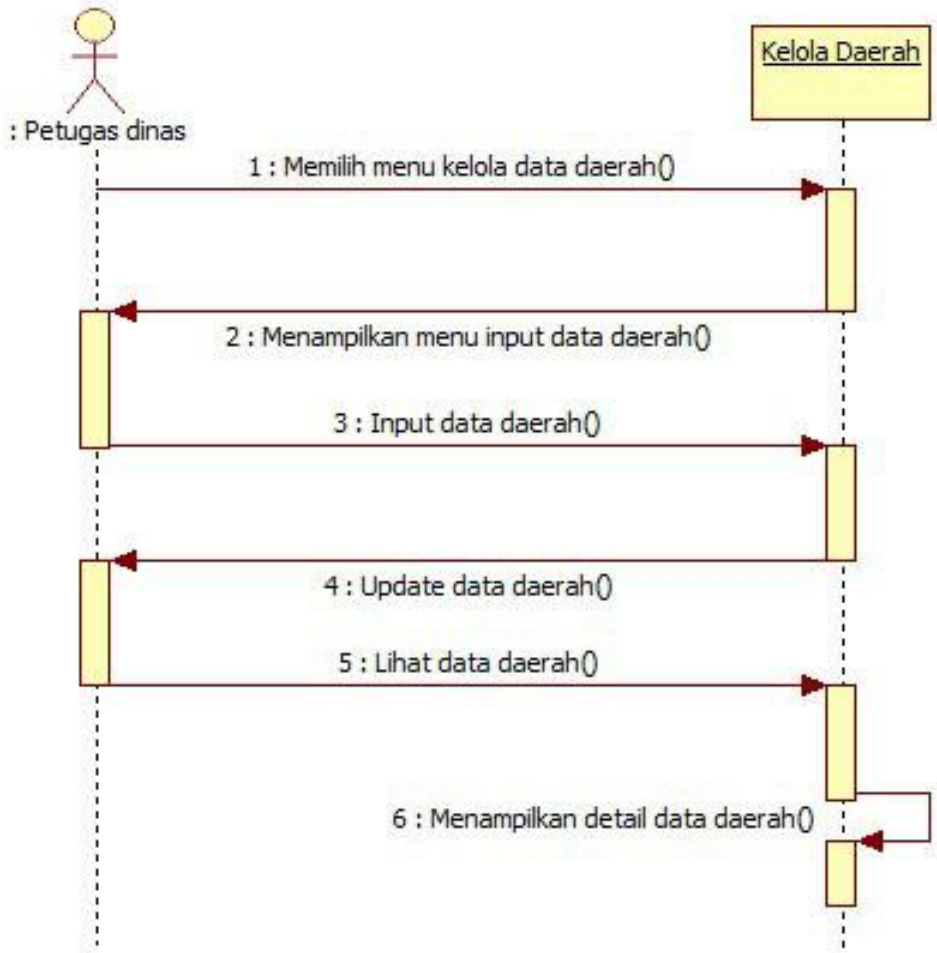



**Gambar 4.5** *Activity Diagram* laporan Penyakit Ayam

### 4.1.3.3 Sequence Diagram

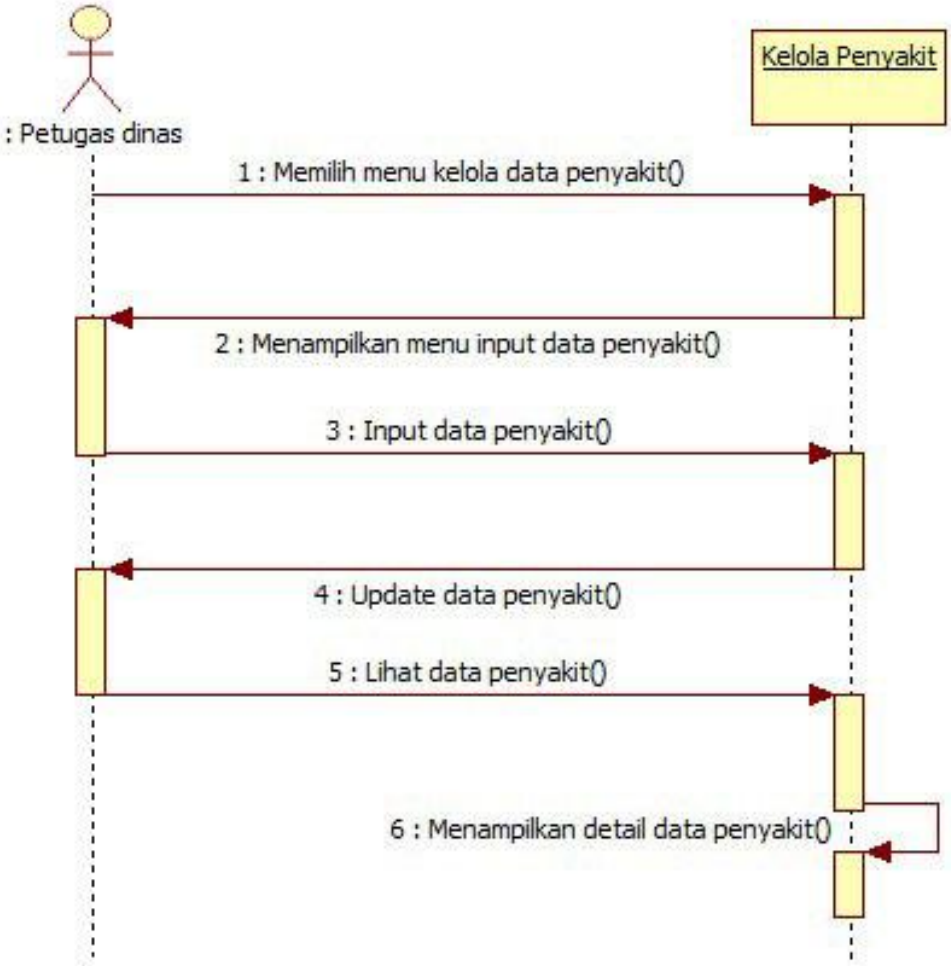
*Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah skenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Berikut adalah *sequence diagram* untuk sistem informasi geografis penyakit ayam.

1. *Sequence Diagram* Menentukan Daerah



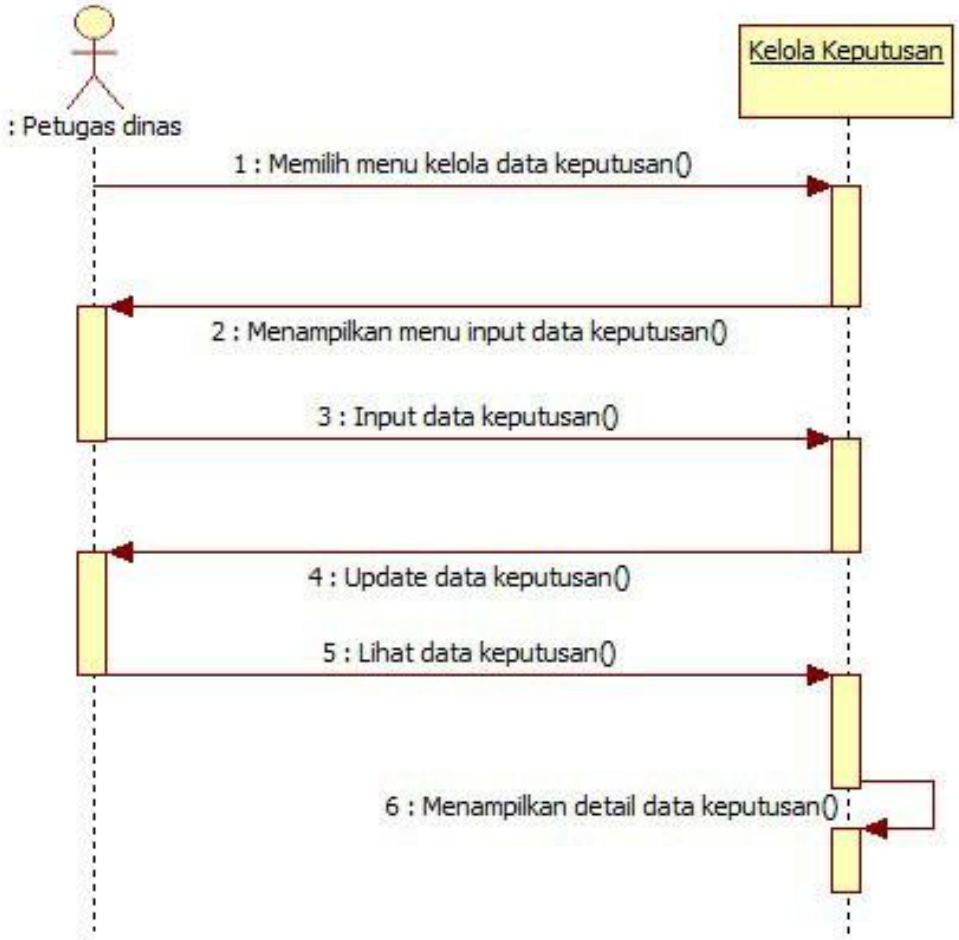

**Gambar 4.6** *Sequence Diagram* Menentukan Daerah

2. *Sequence Diagram* Kelola Data Penyakit



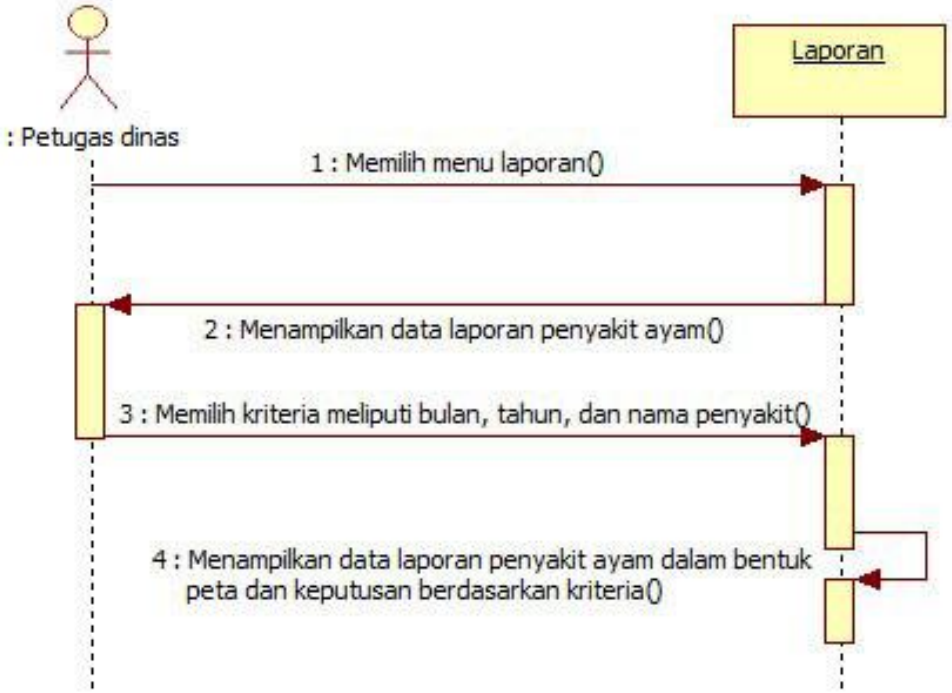

**Gambar 4.7** *Sequence Diagram* Kelola Data Penyakit

3. *Sequence Diagram* Pengambilan Keputusan




**Gambar 4.8** *Sequence Diagram* Pengambilan Keputusan

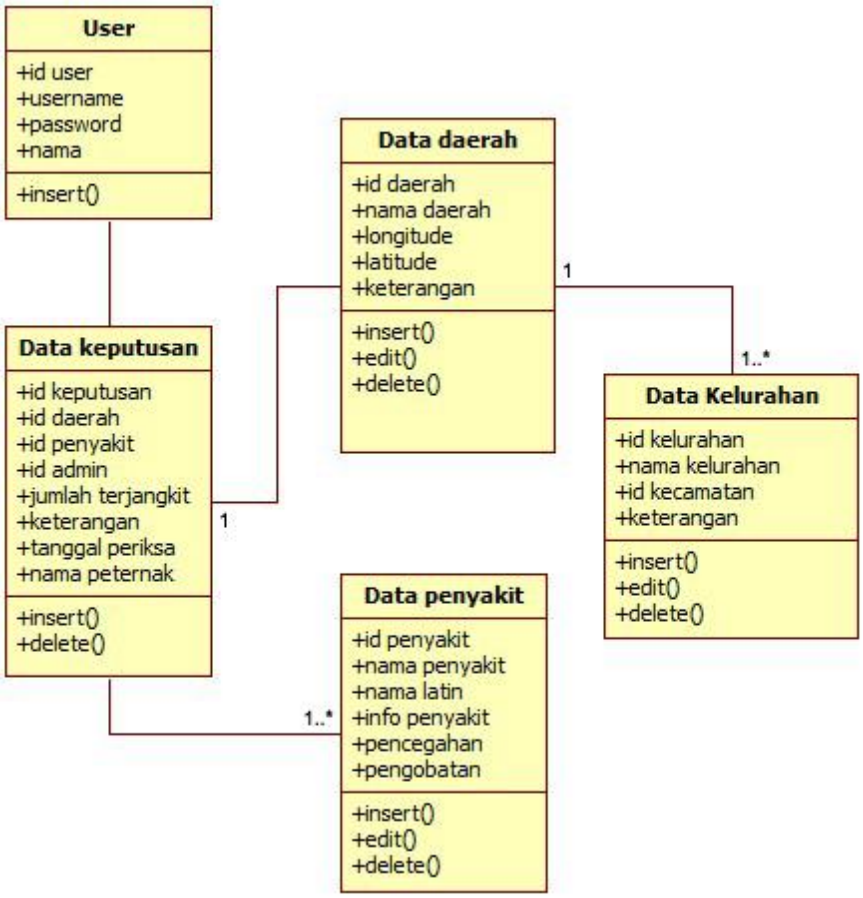
4. *Sequence Diagram* Laporan




**Gambar 4.9** *Sequence Diagram* Laporan

### 4.1.3.4 Class Diagram

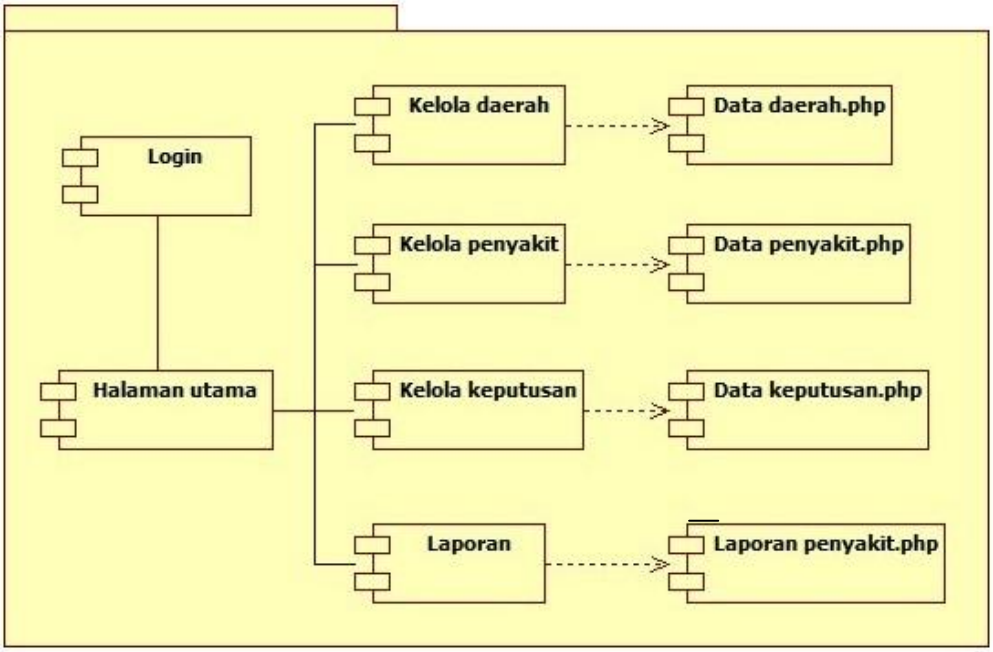
*Class* adalah deskripsi kelompok obyek-obyek dengan properti, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *class diagram* dapat memberikan pandangan global atas sebuah sistem.




**Gambar 4.10** *Class Diagram* Sistem Informasi Geografis Penyakit Ayam

### 4.1.3.5 Component Diagram

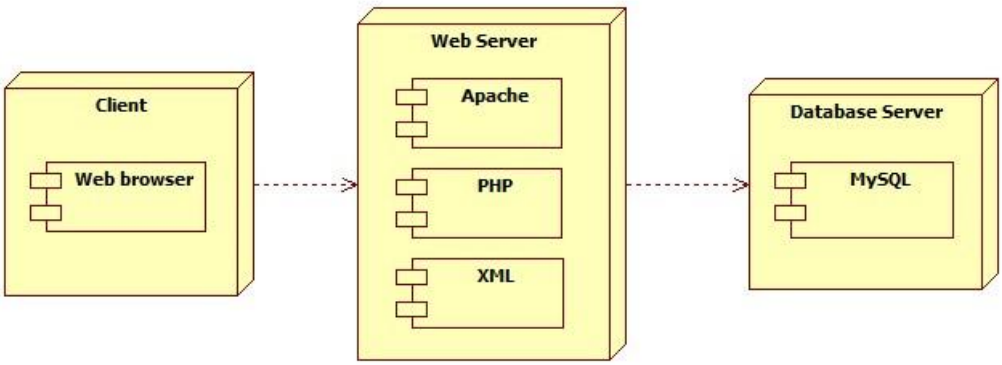
Komponen dapat berupa *source code*, komponen *biner*, atau *executable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari logical view ke *component view*. Sehingga *component diagram* merepresentasikan dunia nyata yaitu *component software* yang mengandung *component*, *interface*, dan *relationship*. *Component diagram* sistem informasi geografis untuk memantau lokasi penyebaran penyakit ayam dapat dilihat pada gambar 4.11.




**Gambar 4.11** *Component Diagram* Sistem Informasi Geografis Penyakit Ayam

### 4.1.3.6 Deployment Diagram

*Deployment diagram* menggambarkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. *Deployment diagram* sistem informasi geografis untuk memantau lokasi penyebaran penyakit ayam dapat dilihat pada gambar 4.12.




**Gambar 4.12** *Deployment Diagram* Sistem Informasi Geografis Penyakit Ayam

## 4.2 Perancangan Antar Muka

Perancangan antar muka atau *design interface* ini akan menggambarkan rancangan tampilan dalam sebuah program aplikasi. Halaman atau tampilan inilah yang nantinya akan berinteraksi langsung dengan pengguna program aplikasi. Perancangan antar muka ini berguna untuk memudahkan menentukan tampilan program aplikasi dan membantu merancang tampilan supaya sesuai dengan kebutuhan sistem yang diusulkan dengan mengadopsi kebutuhan sistem yang sedang berjalan.

Perancangan antar muka ini meliputi perancangan, yaitu perancangan struktur menu, perancangan *input* dan perancangan *output*. Berikut ini penjelasan dan rancangan dari ketiga perancangan tersebut.

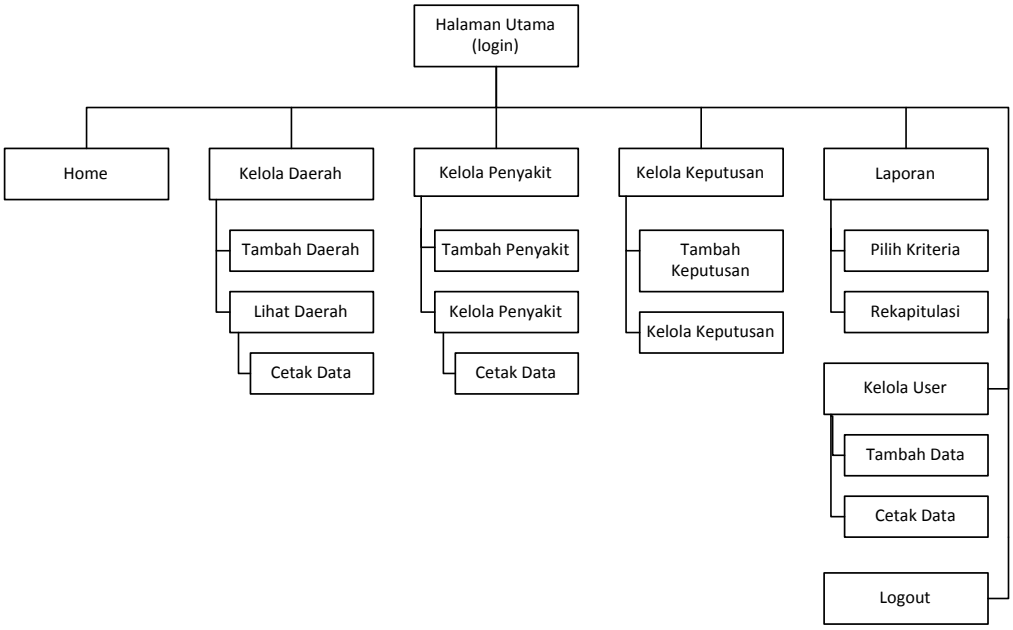
### 4.2.1 Struktur Menu

Perancangan struktur menu dibuat untuk memudahkan proses pembuatan menu pada program aplikasi. Tujuannya adalah agar menu-menu yang nantinya akan diakses oleh pengguna dapat tersusun dengan rapi dan terstruktur dengan

baik. Dengan harapan pengguna tidak akan menemukan kesulitan untuk menggunakan program aplikasi melalui menu-menu tersebut.

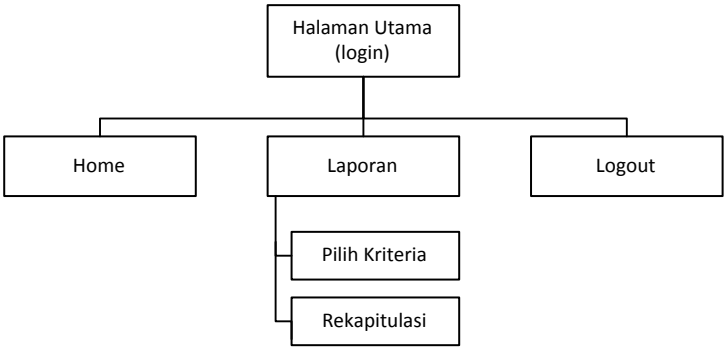
Pada program aplikasi yang diusulkan pada sistem informasi geografis untuk memantau lokasi penyebaran penyakit ayam, penulis membagi perancangan struktur menu sesuai authorisasi masing-masing *user*. Hal ini bertujuan untuk lebih memperjelas menu-menu apa saja yang dapat diakses oleh petugas dinas dan masyarakat.

a. Struktur Menu Administrator




**Gambar 4.13** Struktur Menu Administrator

b. Struktur Menu Masyarakat




**Gambar 4.14** Struktur Menu *User*

### 4.2.2 Perancangan Input

Perancangan input merupakan perancangan tampilan (*interface*) yang akan digunakan oleh user untuk memberikan masukan/inputan ke program untuk diproses. Berikut ini adalah gambar tampilan input:

a. Halaman Login

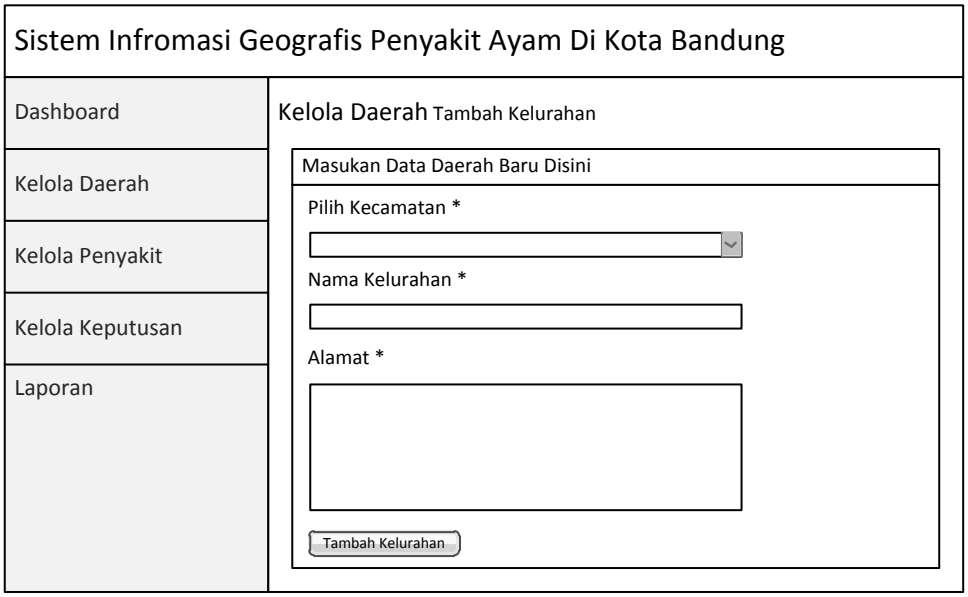
Halaman login merupakan halaman awal yang tampil ketika program pertama kali diakses. Halaman ini digunakan oleh semua user untuk melakukan login sesuai level authentifikasinya masing-masing. Berikut rancangan tampilannya:




**Gambar 4.15** Perancangan Halaman Login

b. Halaman Input Daerah

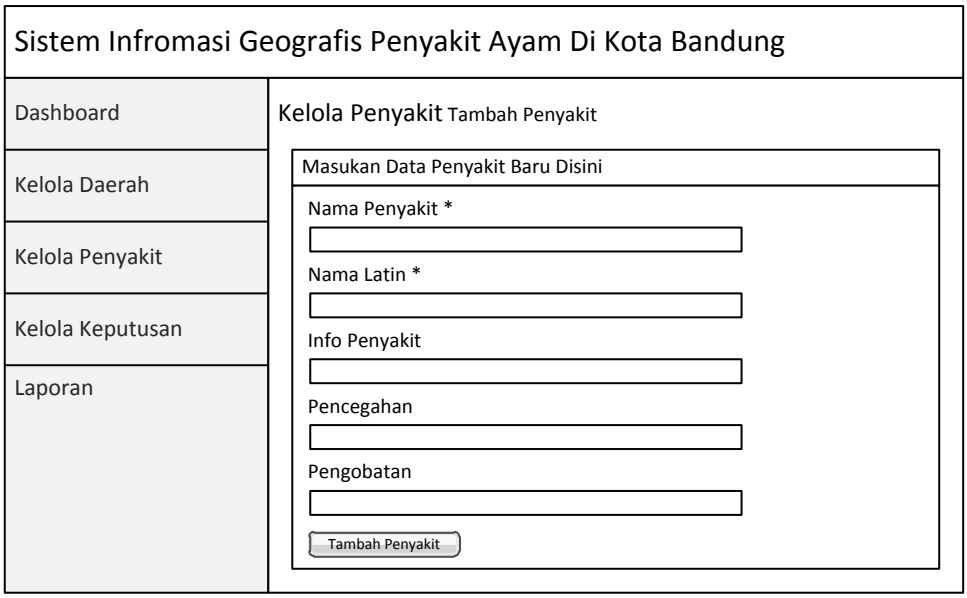
Halaman ini digunakan untuk menambahkan data daerah baru, apabila penambahan daerah. Berikut rancangan tampilannya:




**Gambar 4.16** Perancangan Halaman Input Data Daerah

c. Halaman Input Penyakit Ayam

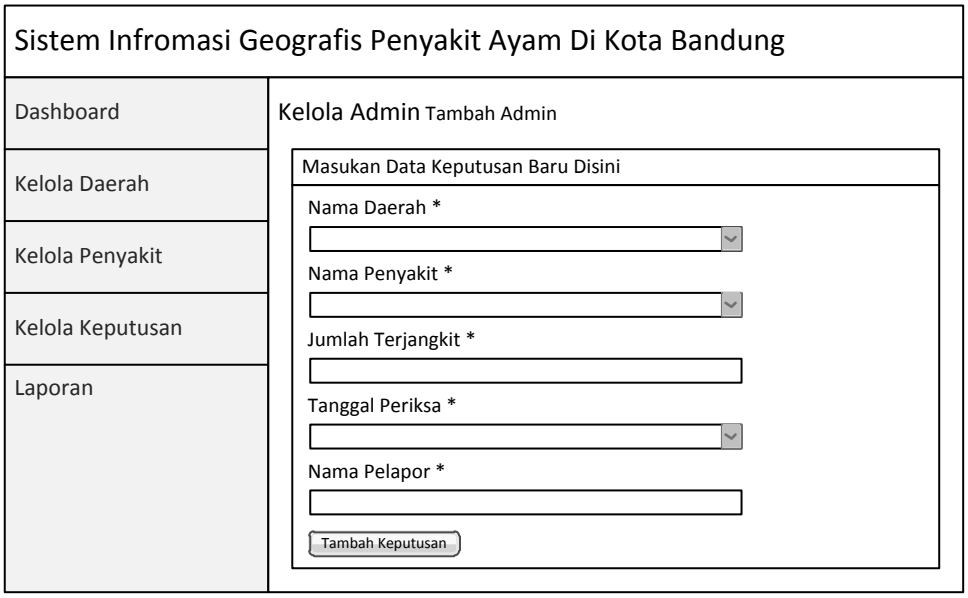
Halaman ini merupakan bagian dari halaman kelola penyakit yang berfungsi untuk menambahkan data penyakit ayam. Berikut rancangan tampilannya:




**Gambar 4.17** Perancangan Halaman Input Data Penyakit

d. Halaman Input Keputusan

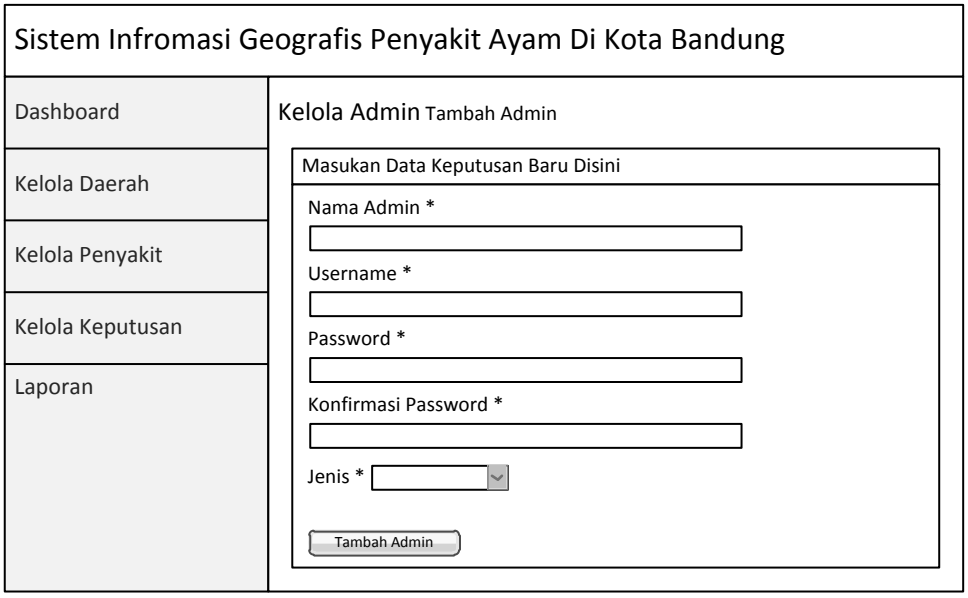
Halaman ini merupakan halaman yang digunakan untuk memasukan atau menginputkan keputusan. Halaman ini hanya dapat diakses oleh admin. Berikut desain tampilannya:




**Gambar 4.18** Perancangan Halaman Input Keputusan

e. Halaman Input Admin

Halaman ini merupakan bagian dari halaman admin yang berfungsi untuk menambahkan data *user*. Berikut rancangan tampilannya:




**Gambar 4.19** Perancangan Halaman Input Admin

### 4.2.3 Perancangan Output

Perancangan output merupakan suatu bentuk rancangan keluaran program aplikasi yang merupakan hasil pengolahan data. *Ouput* kemudian akan dicetak menjadi dokumen fisik yang memberikan informasi tertentu.

Berikut ini beberapa perancangan *output* yang akan dihasilkan program aplikasi pada sistem informasi geografis berbasis website pada dinas peternakan yang diusulkan:

a. Laporan Daerah

Laporan Data Kelurahan

Data Kelurahan (Diakses pada xx-xxx-xxxx xx:xx:xx pm)

| ID | Nama Kelurahan | Nama Kecamatan | Alamat |
|---|---|---|---|
| xx | xxx | xxx | xxx |
| xx | xxx | xxx | xxx |
| xx | xxx | xxx | xxx |
| xx | xxx | xxx | xxx |
| xx | xxx | xxx | xxx |

**Gambar 4.20** Perancangan Output Laporan Daerah

b. Laporan Penyakit

Laporan Data Penyakit

Data Penyakit (Diakses pada xx-xxx-xxxx xx:xx:xx pm)

| ID | Nama Penyakit | Nama Latin | Info Penyakit |
|---|---|---|---|
| xx | xx | xx | xx |
| xx | xx | xx | xx |
| xx | xx | xx | xx |
| xx | xx | xx | xx |

**Gambar 4.21** Perancangan Output Laporan Penyakit

c. Laporan Data Admin

Laporan Data Admin

Data Admin (Diakses pada xx-xxx-xxxx xx:xx:xx pm)

| Username | Password | Jenis Admin | Nama |
|---|---|---|---|
| xx | xx | xx | xx |
| xx | xx | xx | xx |
| xx | xx | xx | xx |
| xx | xx | xx | xx |

**Gambar 4.22** Perancangan Output Laporan Data Admin

## 4.3 Perancangan Arsitektur Jaringan

Arsitektur jaringan menggambarkan hubungan fisik antar komputer dan komponen-komponen jaringan, agar sistem yang diusulkan dapat berjalan dengan baik. Adapun kebutuhan untuk mendukung arsitektur jaringan ini adalah:

1. *User*, pengguna yang memakai Sistem Informasi Geografis Untuk Memantau Lokasi Penyebaran Penyakit Ayam Di Kota Bandung.

2. *Web Server*, penyimpanan *database* data daerah, data penyakit, data keputusan dan laporan.
3. Komputer, atau lebih dikenal sebagai perangkat keras yang sering digunakan untuk mengakses internet.

## 4.4 Implementasi

Implementasi merupakan tahap penerapan sistem yang diusulkan yang telah dirancang pada bab sebelumnya. Dalam proses implementasi harus disiapkan beberapa sumber daya yang akan digunakan oleh sistem. Sumber daya inilah yang akan menunjang kelancaran kinerja sistem.

Sistem informasi geografis berbasis *web* yang diusulkan pada Dinas Peternakan Provinsi Jawa Barat memiliki program aplikasi yang menggunakan PHP sebagai bahasa pemrograman dan MySQl sebagai *database*-nya. Pada tahan inilah dilakukan instalasi dari program aplikasi tersebut pada *server hosting* yang sesungguhnya.

### 4.4.1 Batasan Implementasi

Tahapan implementasi sistem ini memiliki batasan agar implementasi yang dilakukan dapat fokus dan tidak melebar pada kasus yang lain. Berikut ini batasan implementasi sistem informasi geografis yang diusulkan pada Dinas Peternakan Provinsi Jawa Barat:

1. Karena biaya untuk menyediakan akses internet dengan kecepatan *upstream* yang besar dan pengadaan *server web* lokal di gedung dinas peternakan yang cukup tinggi, dan adanya keterbatasan sumber daya

manusia untuk menjaga dan me-maintenance *server* tersebut, maka pada tahap implementasi ini, program aplikasi akan diinstal pada *server hosting* yang sudah disewa dari perusahaan penyedia jasa *hosting*.

2. Akses ke program aplikasi hanya bisa dilakukan melalui jaringan internet, hal ini dikarenakan keterbatasan pada poin nomor 1.

### 4.4.2 Implementasi Perangkat Lunak

Perangkat lunak merupakan komponen yang wajib ada dan sangat dibutuhkan untuk membuat dan mengimplementasikan sistem. Dalam proses pembuatan program aplikasi sistem informasi geografis untuk memantau lokasi penyebaran penyakit ayam di Kota Bandung, penulis menggunakan beberapa perangkat lunak berikut:

1. Sistem Operasi Window 7 Enterprise.
2. XAMPP 1.8.3 yang terdiri atas *apache* sebagai *web server*, PHP sebagai bahasa pemrograman dan MySQL sebagai *database server*.
3. CodeLobster PHP Edition sebagai layout dan *editor* penulisan *file-file* PHP, HTML, CSS dan JavaScript.
4. Navicat sebagai GUI dalam mengakses *database* MySQL
5. *Web Browser* Google Chrome, Mozilla Firefox dan Opera, sebagai media untuk mengakses program aplikasi yang dibuat.
6. Adobe Photoshop CS 3 sebagai perangkat lunak pengolah gambar untuk membuat desain *web* dan gambar-gambar yang dibutuhkan.

Sedangkan dalam proses implementasi, selain menggunakan beberapa perangkat lunas di atas, penulis juga menggunakan beberapa perangkat lunak di bawah ini:

1. Cpanel sebagai media untuk mengontrol seluruh kegiatan hosting.
2. PhpMyAdmin sebagai media untuk mengakses, *import* dan *export database.*
3. Filezilla FTP Client sebagai aplikasi untuk *transfer file* dengan menggunakan *port ftp*.

Setelah aplikasi ter-*install* di *server hosting*, maka *user* hanya membutuhkan *web browser* dan koneksi internet saja untuk mengakses program aplikasi ini.

### 4.4.3 Implementasi Perangkat Keras

Dalam mengimplementasikan program aplikasi sistem informasi geografis untuk memantau lokasi penyebaran penyakit ayam di Kota Bandung, dibutuhkan perangkat keras (*hardware*). Agar program aplikasi dapat berjalan dengan maksimal maka dibutuhkan komputer dengan spesifikasi minimum berikut ini:

1. Spesifikasi Komputer *Server*
    a. Processor Intel X3440 Xeon Quad Core 2.53Ghz
    b. RAM 8 GB
    c. Harddisk 160 GB
    d. VGA onboard
2. Spesifikasi Komputer Klien
    a. Processor Intel Pentium IV 1.8Ghz

b. RAM 512 MB

c. Harddisk minimal 160 GB

d. VGA onboard

Selain kedua komputer di atas, program aplikasi ini juga membutuhkan perangkat jaringan internet untuk menghubungkan komputer *server* dan klien. Tentu juga dibutuhkan printer, monitor, mouse dan keyboard standar.

### 4.4.4 Implementasi Basis Data

Di bawah ini merupakan *sintaks* SQL yang digunakan untuk membuat tabel-tabel di dalam *database* yang digunakan pada program aplikasi sistem informasi geografis untuk memantau lokasi penyebaran penyakit ayam di Kota Bandung:

```
CREATE TABLE IF NOT EXISTS `tb_daerah` (
  `id_daerah` int(11) NOT NULL AUTO_INCREMENT,
  `nama_daerah` varchar(50) NOT NULL,
  `longt` double NOT NULL,
  `lat` double NOT NULL,
  `keterangan` text NOT NULL,
  PRIMARY KEY (`id_daerah`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=16 ;

CREATE TABLE IF NOT EXISTS `tb_kelurahan` (
  `id_kelurahan` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `nama_kelurahan` varchar(200) NOT NULL,
  `id_kecamatan` int(11) NOT NULL,
  `keterangan` text NOT NULL,
  `longt` double NOT NULL,
  `lat` double NOT NULL,
  PRIMARY KEY (`id_kelurahan`),
  KEY `id_kecamatan` (`id_kecamatan`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=22 ;

CREATE TABLE IF NOT EXISTS `tb_keputusan` (
  `id_keputusan` int(11) NOT NULL AUTO_INCREMENT,
  `id_daerah` int(11) NOT NULL,
  `id_penyakit` int(11) NOT NULL,
  `id_admin` int(11) NOT NULL,
  `jumlah_terjangkit` int(11) NOT NULL,
  `keterangan` text NOT NULL,
  `tanggal_periksa` date NOT NULL,
  PRIMARY KEY (`id_keputusan`),
  KEY `id_daerah` (`id_daerah`),
  KEY `id_penyakit` (`id_penyakit`),
  KEY `id_admin` (`id_admin`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=16 ;

CREATE TABLE IF NOT EXISTS `tb_penyakit` (
```

```
  `id_penyakit` int(11) NOT NULL AUTO_INCREMENT,
  `nama_penyakit` varchar(100) NOT NULL,
  `nama_latin` varchar(100) NOT NULL,
  `info_penyakit` text NOT NULL,
  `pencegahan` text NOT NULL,
  `pengobatan` text NOT NULL,
  PRIMARY KEY (`id_penyakit`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ;

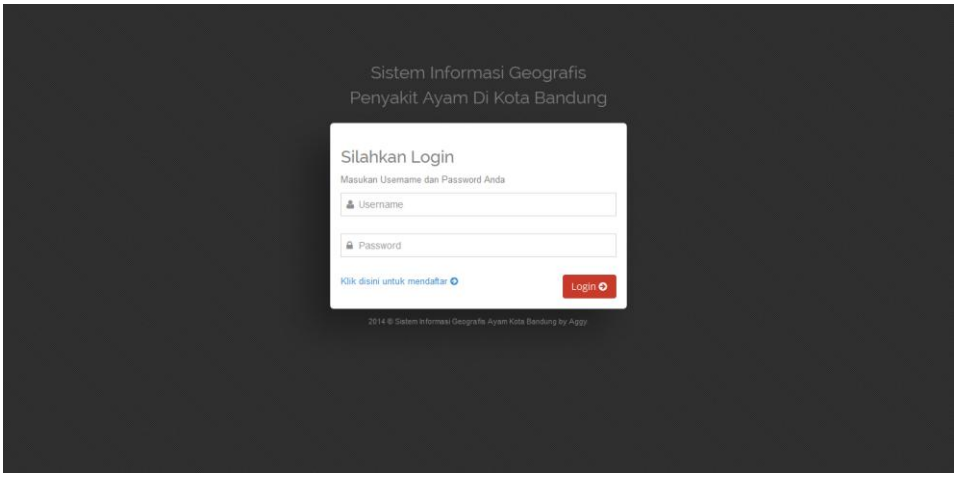
CREATE TABLE IF NOT EXISTS `tb_user` (
  `id_user` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(100) NOT NULL,
  `password` varchar(100) NOT NULL,
  `nama` varchar(100) NOT NULL,
  `peran` int(1) NOT NULL DEFAULT '3' COMMENT '1 : Admin, 2: Petugas Dinas 3: Rakyat',
  PRIMARY KEY (`id_user`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=12 ;
```

### 4.4.5 Implementasi Antar Muka

Tampilan antar muka (*interface*) merupakan tampilan program yang akan langsung berhadapan dengan *user*. *User* akan berinteraksi memberikan *input* atau menerima *output* melalui tampilan interface ini. Rancangan antar muka yang sudah dibuat pada bab sebelumnya diimplementasikan pada sebuah halaman

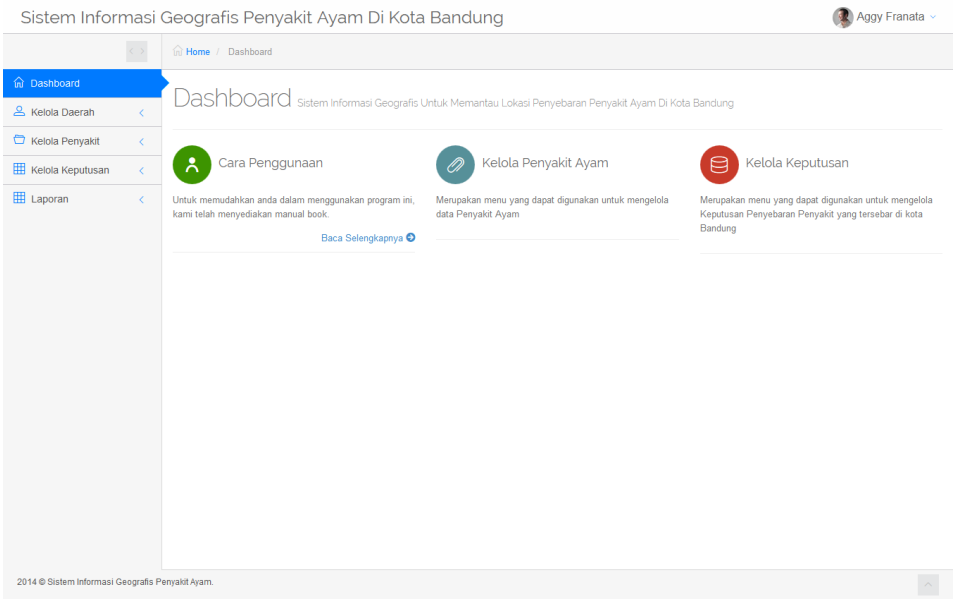
HTML yang dikelola oleh bahasa pemrograman PHP. Berikut tampilan antar muka program aplikasi sistem informasi geografis untuk memantau lokasi penyebaran penyakit ayam di Kota Bandung:

1. Halaman utama (Login)



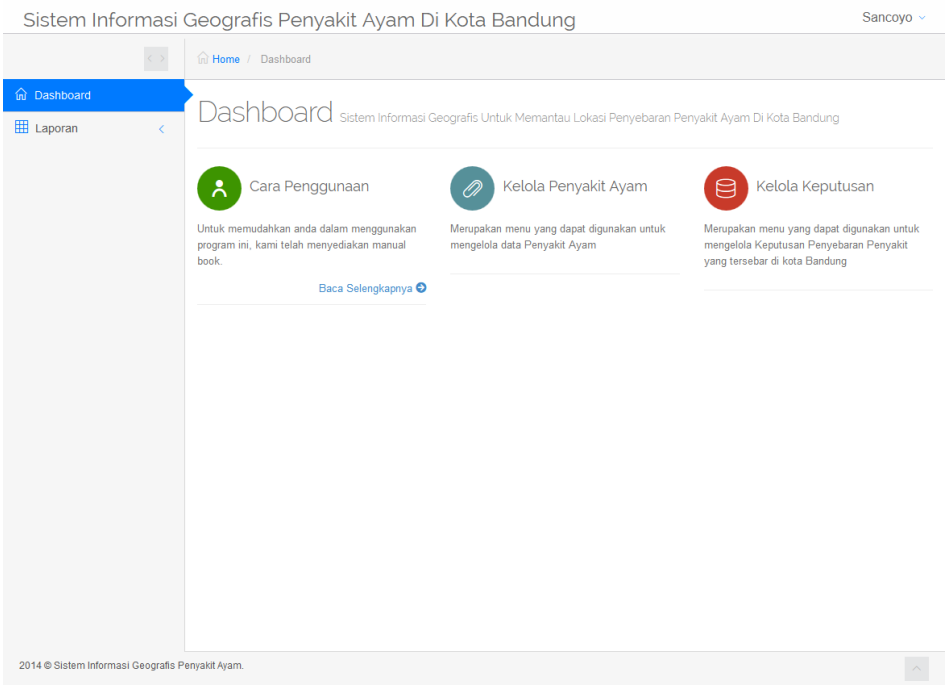

**Gambar 4.23** Halaman Login

2. Halaman Home Admin setelah Login



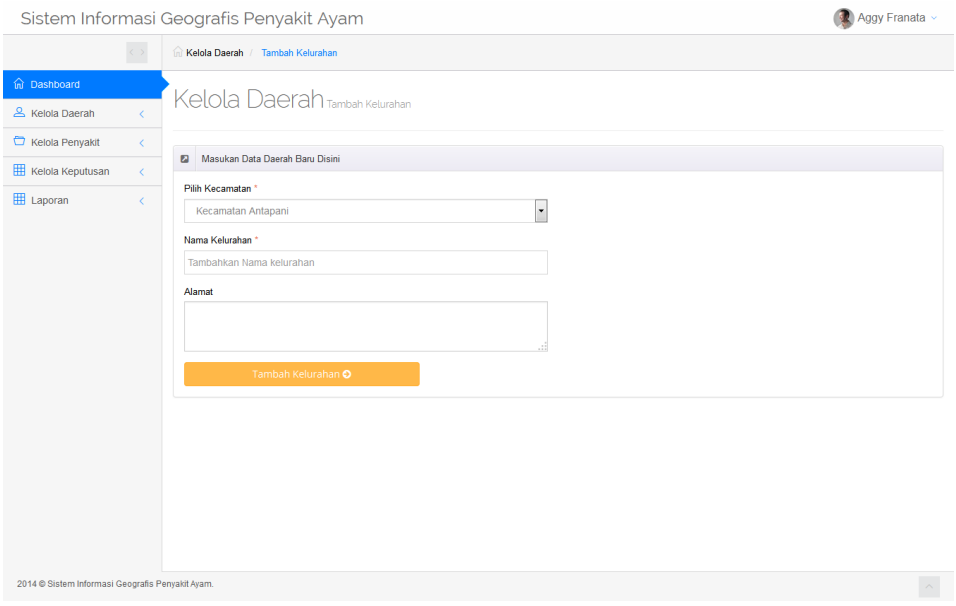

**Gambar 4.24** Halaman Home Admin

3. Halaman Home Masyarakat



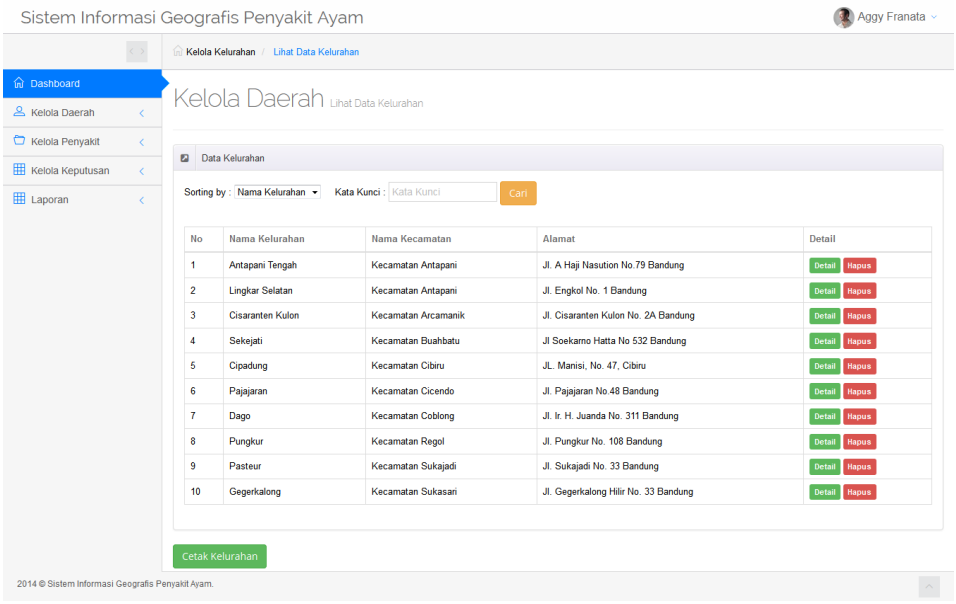

**Gambar 4.25** Halaman Home Masyarakat

4. Halaman Input Daerah



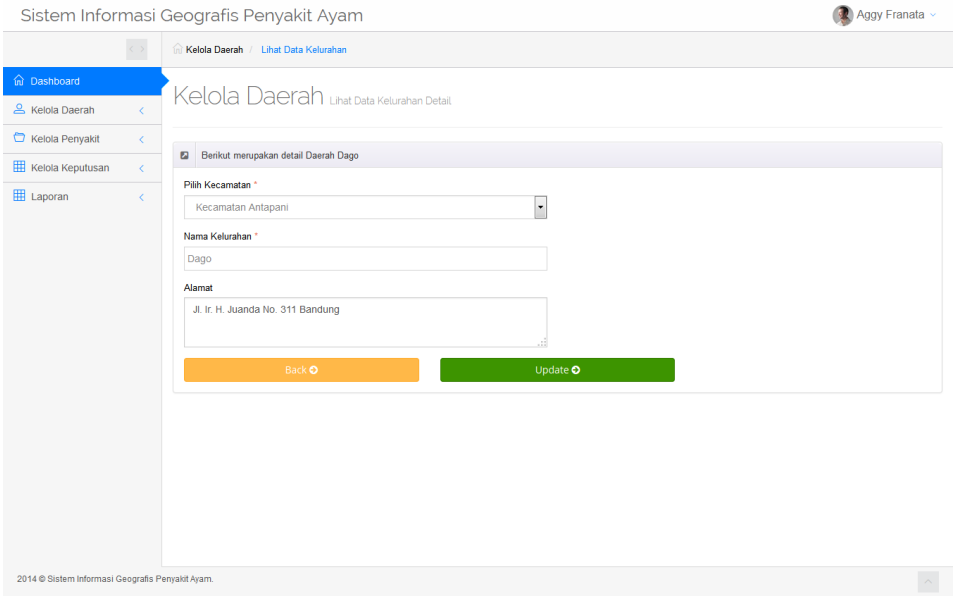

**Gambar 4.26** Halaman Input Data Daerah

5. Halaman Data Daerah



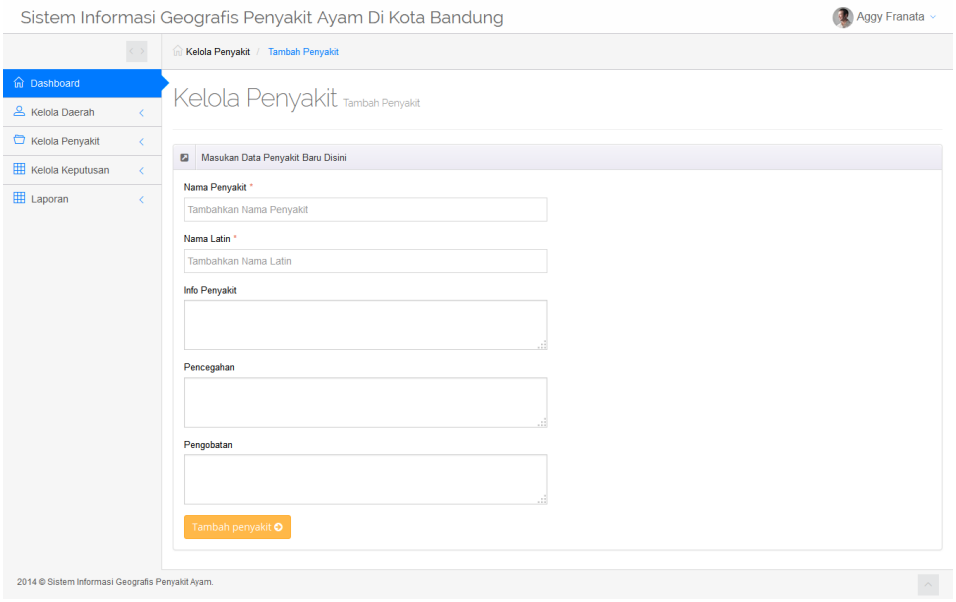

**Gambar 4.27** Halaman Data Daerah

6. Halaman Detail Data Daerah



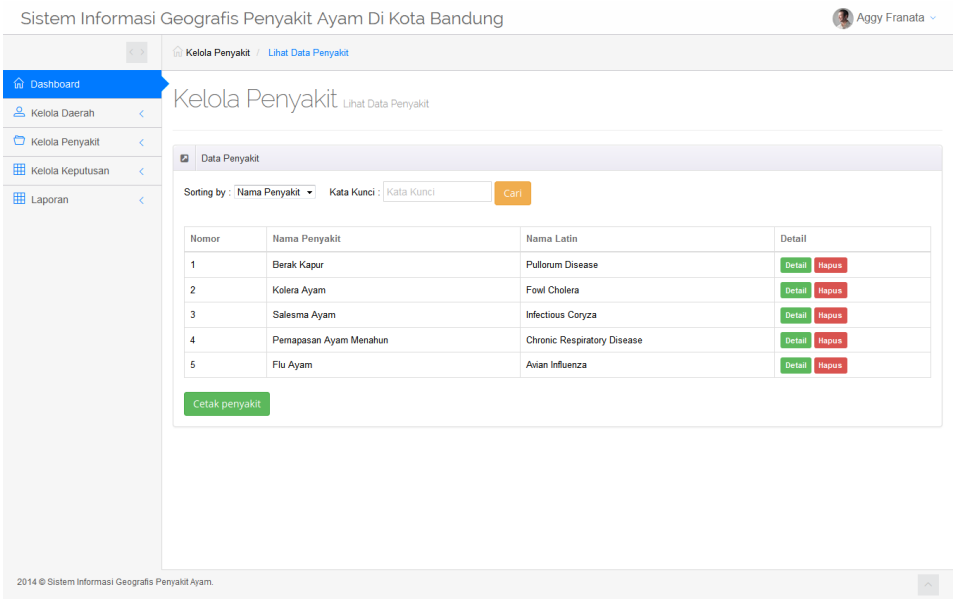

**Gambar 4.28** Halaman Detail Data Daerah

7. Halaman Input Penyakit



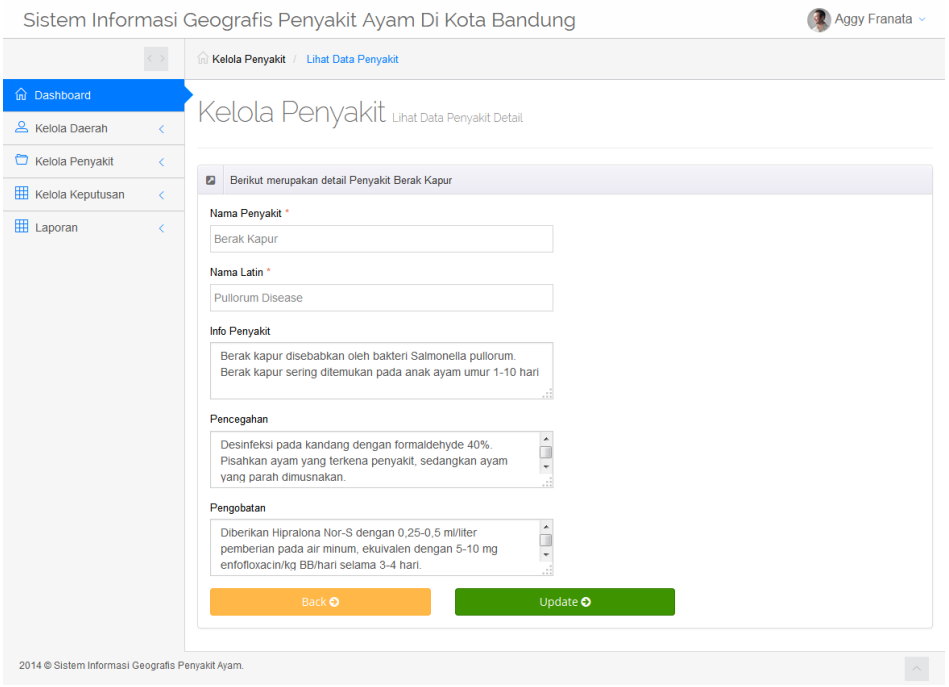

**Gambar 4.29** Halaman Input Data Penyakit

8. Halaman Data Penyakit



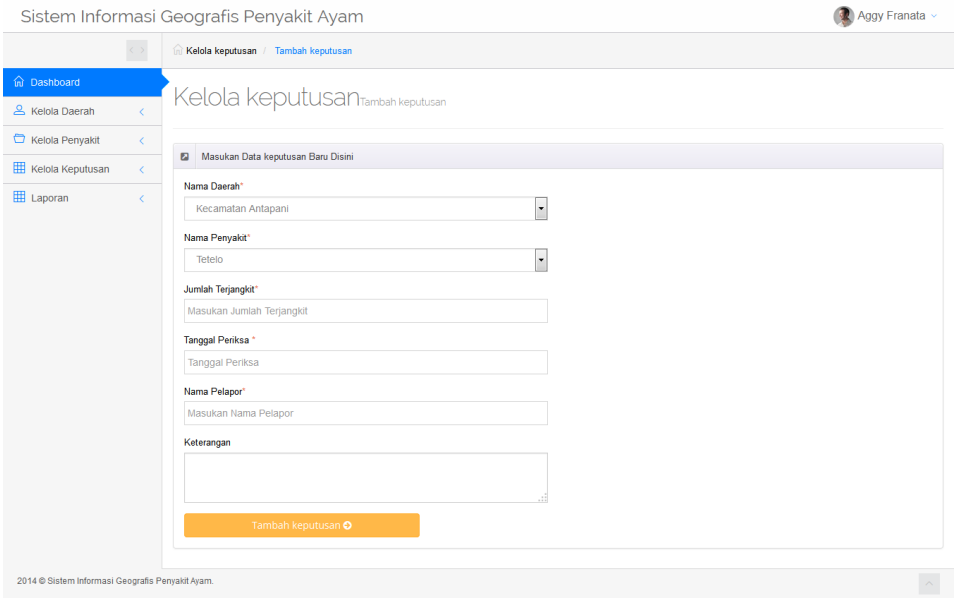

**Gambar 4.30** Halaman Data Penyakit

9. Halaman Detail Penyakit



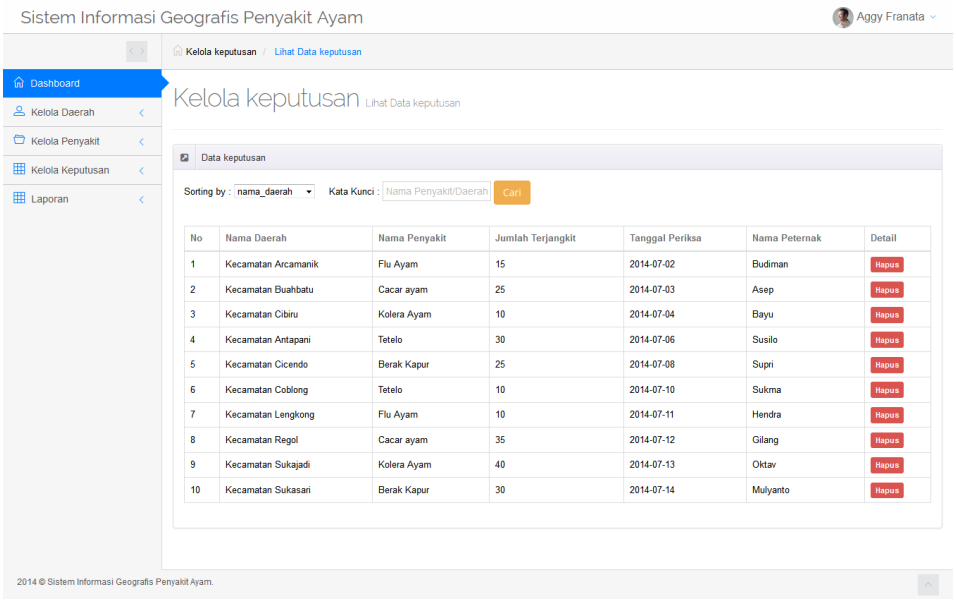

**Gambar 4.31** Halaman Detail Penyakit

10. Halaman Input Keputusan



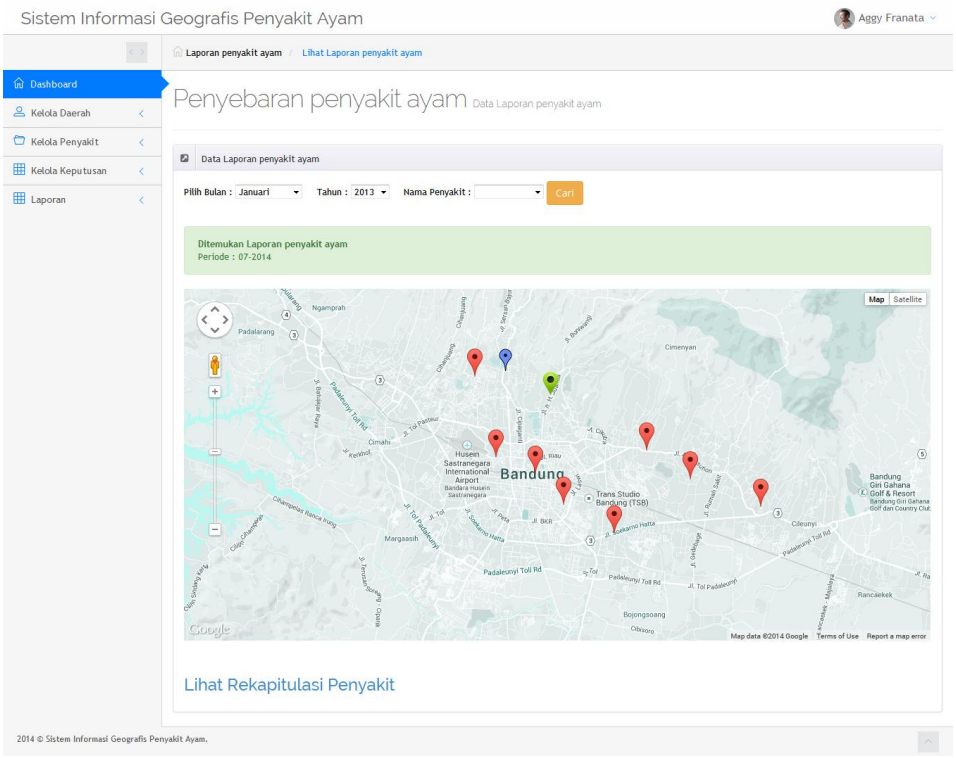

**Gambar 4.32** Halaman Input Keputusan

11. Halaman Data Keputusan



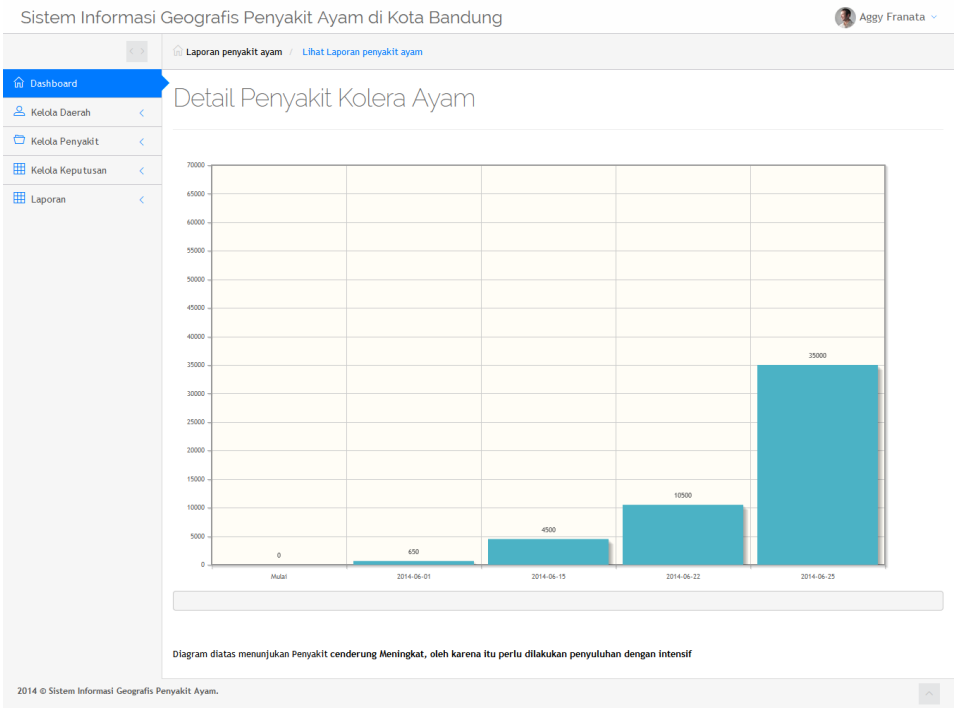

**Gambar 4.33** Halaman Data Keputusan

12. Halaman Laporan Penyakit Ayam



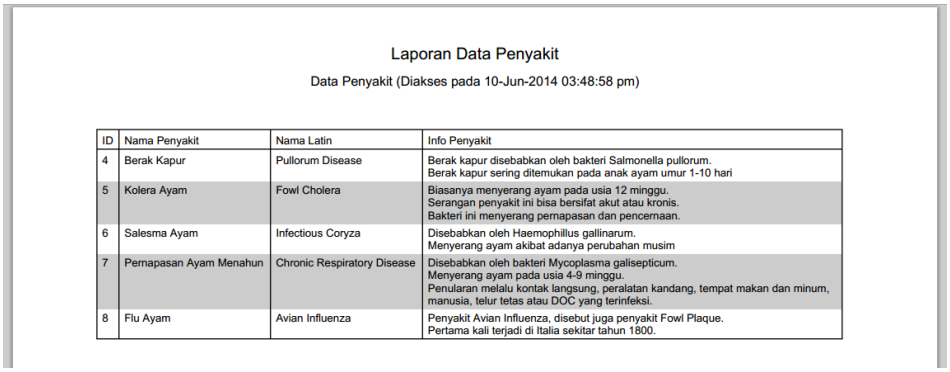

**Gambar 4.34** Halaman Laporan Penyakit Ayam

13. Tampilan Rekapitulasi Laporan




**Gambar 4.35** Tampilan Rekapitulasi Penyakit

14. Tampilan Laporan Data Penyakit



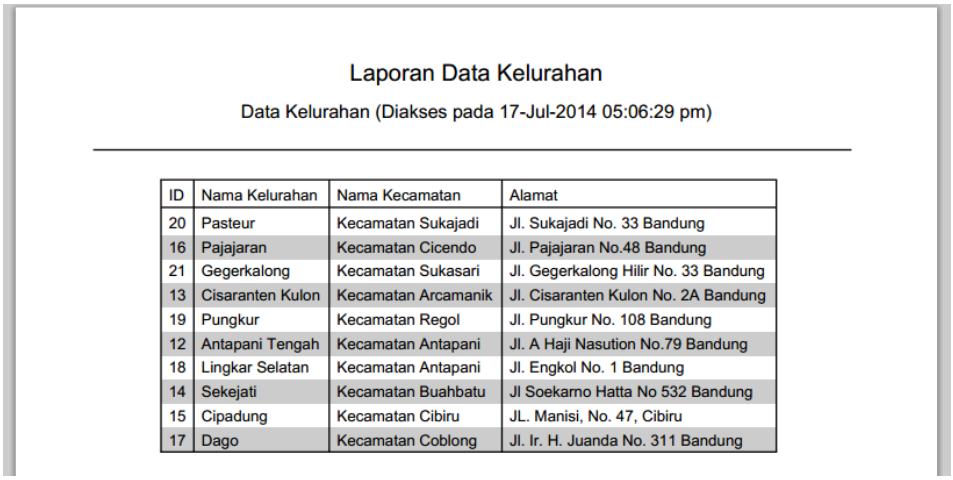
Laporan Data Penyakit

Data Penyakit (Diakses pada 10-Jun-2014 03:48:58 pm)

| ID | Nama Penyakit | Nama Latin | Info Penyakit |
|---|---|---|---|
| 4 | Berak Kapur | Pullorum Disease | Berak kapur disebabkan oleh bakteri Salmonella pullorum.<br>Berak kapur sering ditemukan pada anak ayam umur 1-10 hari |
| 5 | Kolera Ayam | Fowl Cholera | Biasanya menyerang ayam pada usia 12 minggu.<br>Serangan penyakit ini bisa bersifat akut atau kronis.<br>Bakteri ini menyerang pernapasan dan pencernaan. |
| 6 | Salesma Ayam | Infectious Coryza | Disebabkan oleh Haemophillus gallinarum.<br>Menyerang ayam akibat adanya perubahan musim |
| 7 | Pernapasan Ayam Menahun | Chronic Respiratory Disease | Disebabkan oleh bakteri Mycoplasma galisepticum.<br>Menyerang ayam pada usia 4-9 minggu.<br>Penularan melalu kontak langsung, peralatan kandang, tempat makan dan minum, manusia, telur tetas atau DOC yang terinfeksi. |
| 8 | Flu Ayam | Avian Influenza | Penyakit Avian Influenza, disebut juga penyakit Fowl Plaque.<br>Pertama kali terjadi di Italia sekitar tahun 1800. |

**Gambar 4.36** Tampilan Laporan Penyakit Ayam

15. Tampilan Laporan Data Daerah



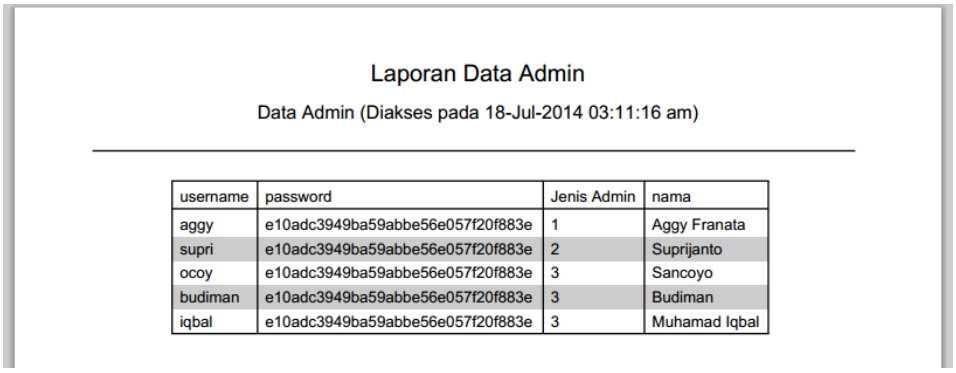
Laporan Data Kelurahan

Data Kelurahan (Diakses pada 17-Jul-2014 05:06:29 pm)

| ID | Nama Kelurahan | Nama Kecamatan | Alamat |
|---|---|---|---|
| 20 | Pasteur | Kecamatan Sukajadi | Jl. Sukajadi No. 33 Bandung |
| 16 | Pajajaran | Kecamatan Cicendo | Jl. Pajajaran No.48 Bandung |
| 21 | Gegerkalong | Kecamatan Sukasari | Jl. Gegerkalong Hilir No. 33 Bandung |
| 13 | Cisaranten Kulon | Kecamatan Arcamanik | Jl. Cisaranten Kulon No. 2A Bandung |
| 19 | Pungkur | Kecamatan Regol | Jl. Pungkur No. 108 Bandung |
| 12 | Antapani Tengah | Kecamatan Antapani | Jl. A Haji Nasution No.79 Bandung |
| 18 | Lingkar Selatan | Kecamatan Antapani | Jl. Engkol No. 1 Bandung |
| 14 | Sekejati | Kecamatan Buahbatu | Jl Soekarno Hatta No 532 Bandung |
| 15 | Cipadung | Kecamatan Cibiru | JL. Manisi, No. 47, Cibiru |
| 17 | Dago | Kecamatan Coblong | Jl. Ir. H. Juanda No. 311 Bandung |

**Gambar 4.37** Tampilan Laporan Data Daerah

16. Tampilan Laporan Data Admin



Laporan Data Admin

Data Admin (Diakses pada 18-Jul-2014 03:11:16 am)

| username | password | Jenis Admin | nama |
|---|---|---|---|
| aggy | e10adc3949ba59abbe56e057f20f883e | 1 | Aggy Franata |
| supri | e10adc3949ba59abbe56e057f20f883e | 2 | Suprijanto |
| ocoy | e10adc3949ba59abbe56e057f20f883e | 3 | Sancoyo |
| budiman | e10adc3949ba59abbe56e057f20f883e | 3 | Budiman |
| iqbal | e10adc3949ba59abbe56e057f20f883e | 3 | Muhamad Iqbal |

**Gambar 4.38** Tampilan Laporan Data Admin

### 4.4.5 Implementasi Instalasi Program

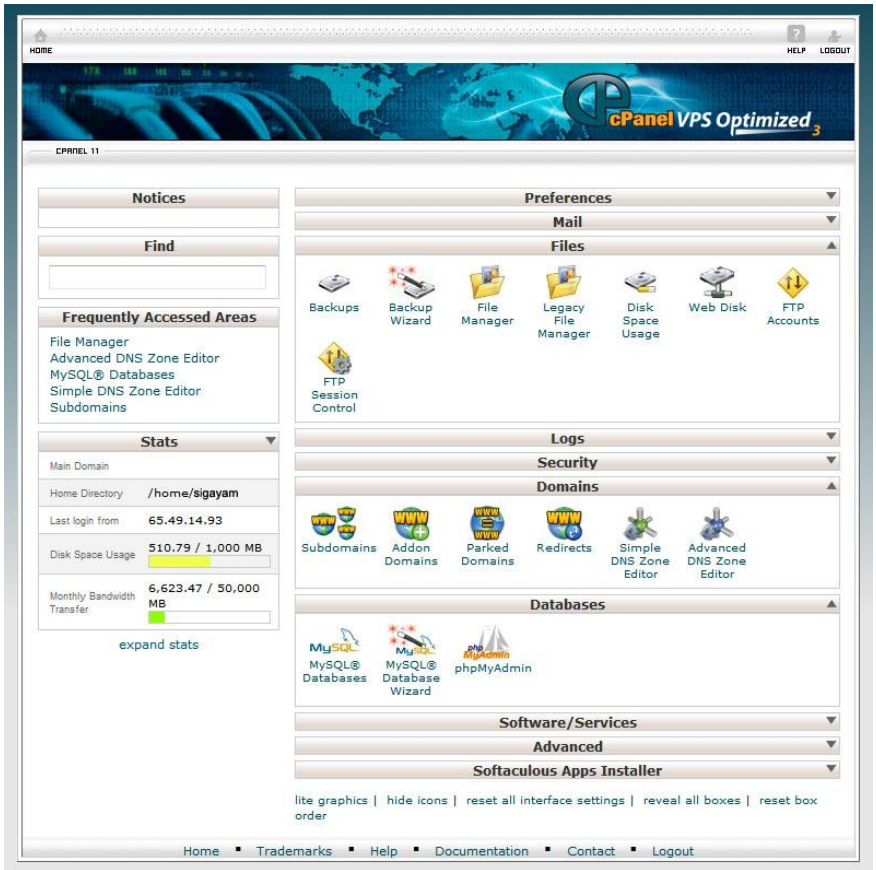
Pada bagian ini akan dijelaskan proses instalasi program ke server hosting. Berikut ini adalah langkah-langkah instalasinya:

1. Membuka halaman cPanel maka akan tampil halaman login berikut:



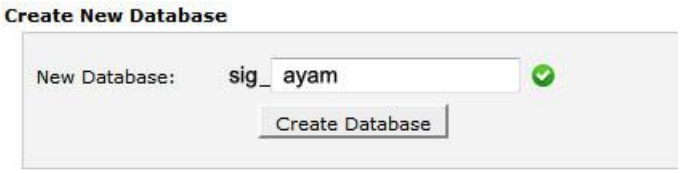

**Gambar 4.39** Tampilan Login cPanel

2. Jika login berhasil maka akan diarahkan ke halaman home cPanel berikut:



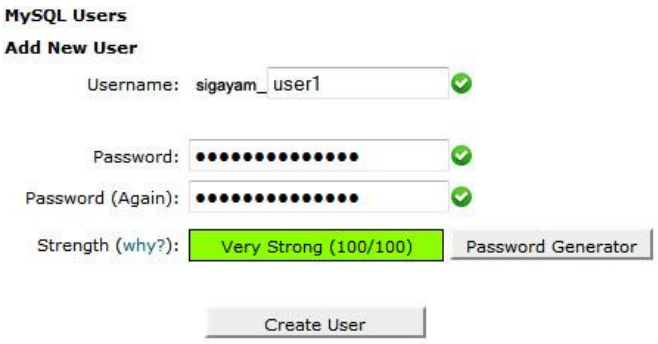

**Gambar 4.40** Halaman Home cPanel

3. Buat *database* melalui menu MySQL® Databases, ketik nama *database*-nya (sig_penyakit_ayam) pada bagian *Create New Database*:




**Gambar 4.41** Membuat database MySQL

Setelah database tercipta, buatlah user MySQL baru:




**Gambar 4.42** Membuat user MySQL

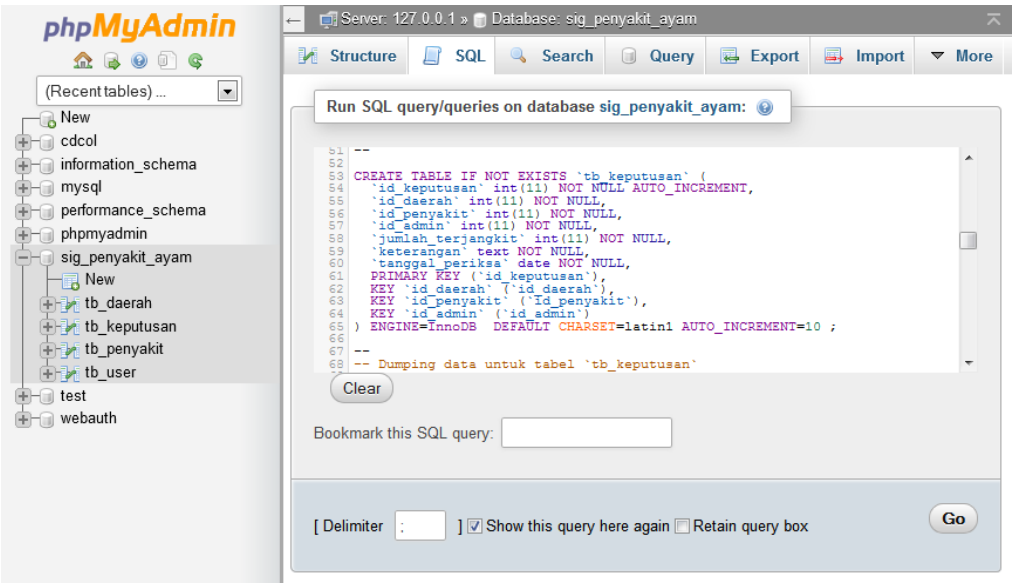
Kemudian set/atur user tersebut agar dapat mengakses database sig_ayam:




**Gambar 4.43** Mengatur user untuk database

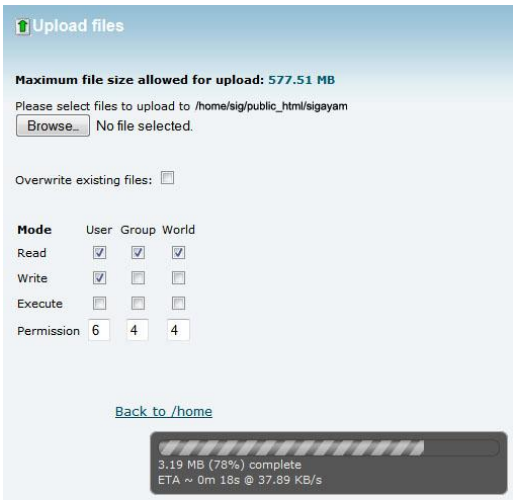
4. Langkah selanjutnya adalah melakukan dumping database melalui phpMyadmin. Klik tombol PhpMyadmin pada halaman home cPanel. Pilih

database sig_ayam, kemudian klik menu SQL, kemudian *paste* sintaks SQL pada Implementasi *Database* di atas:



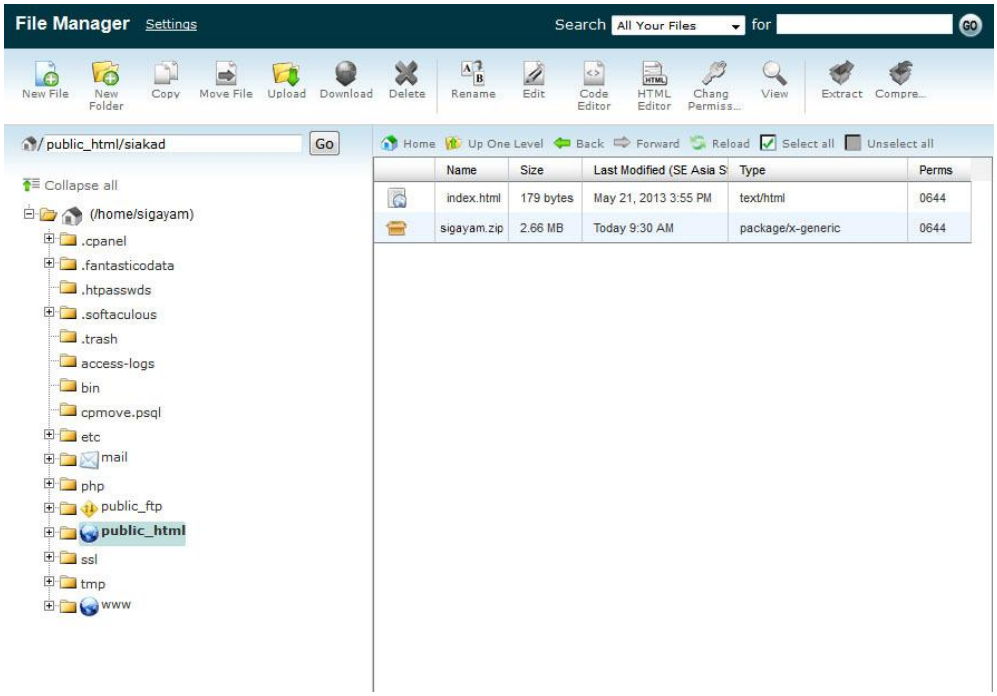

**Gambar 4.44** Dumping tabel-tabel pada databse

5. Berikut adalah meletakkan file-file program aplikasi sistem informasi geografis ayam pada direktori nama sigayam pada *root directory*. Pada home cPanel klik *file manager* kemudian masuk ke direktori sigayam, klik upload, dan berikut tampilan uploadnya:




**Gambar 4.45** Proses upload file-file program aplikasi

Setelah file berhasil terupload:

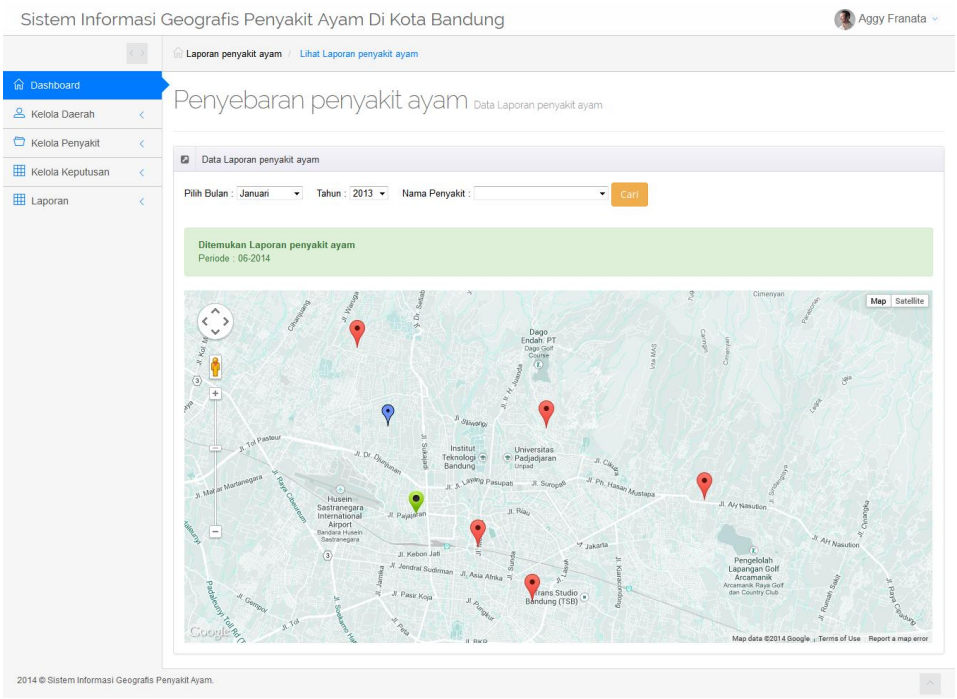



**Gambar 4.46** Tampilan file program pada *server hosting*

Extract file zip tersebut, kemudian setting koneksi ke mysql, setting authorisasi folder (chmod 775) pada folder gambar dan proses instalasi selesai.

### 4.4.6 Penggunaan Program

Pada bagian ini akan dijelaskan secara ringkas, cara penggunaan program. Program aplikasi ini dibuat dengan mempertimbangkan sisi *user friendly*, sehingga penulis meyakini *user* tidak akan menemukan banyak kesulitan dalam mengoperasikannya. Penulis hanya membatasi cara penggunaan program pada menu laporan berikut ini:

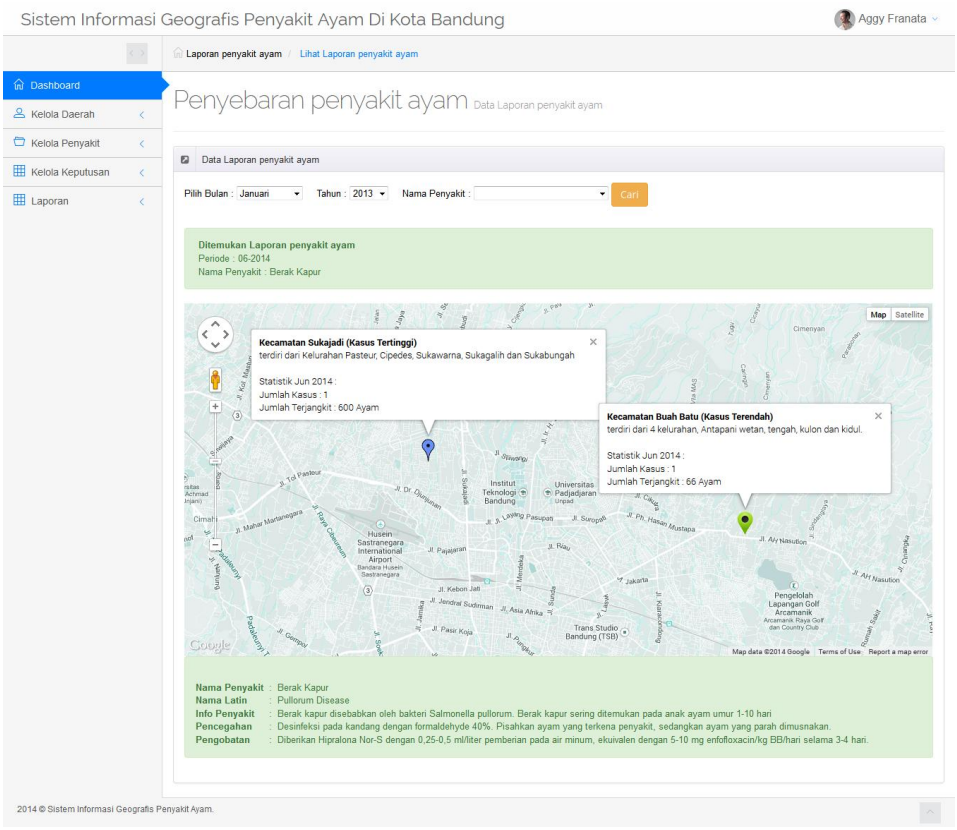



**Gambar 4.47** Laporan Penyakit Ayam

Gambar di atas merupakan tampilan awal ketika menu laporan diklik. Halaman tersebut menampilkan data laporan penyakit ayam periode bulan Juni 2014 dalam bentuk peta.

Di bagian atas peta terdapat kriteria pencarian, yaitu fasilitas untuk melakukan pemilihan bulan, tahun dan nama penyakit ayam. Pada fasilitas ini *user* dapat menentukan pencarian berdasarkan bulan, tahun dan jenis penyakit untuk melihat laporan penyebaran penyakit ayam lebih rinci.

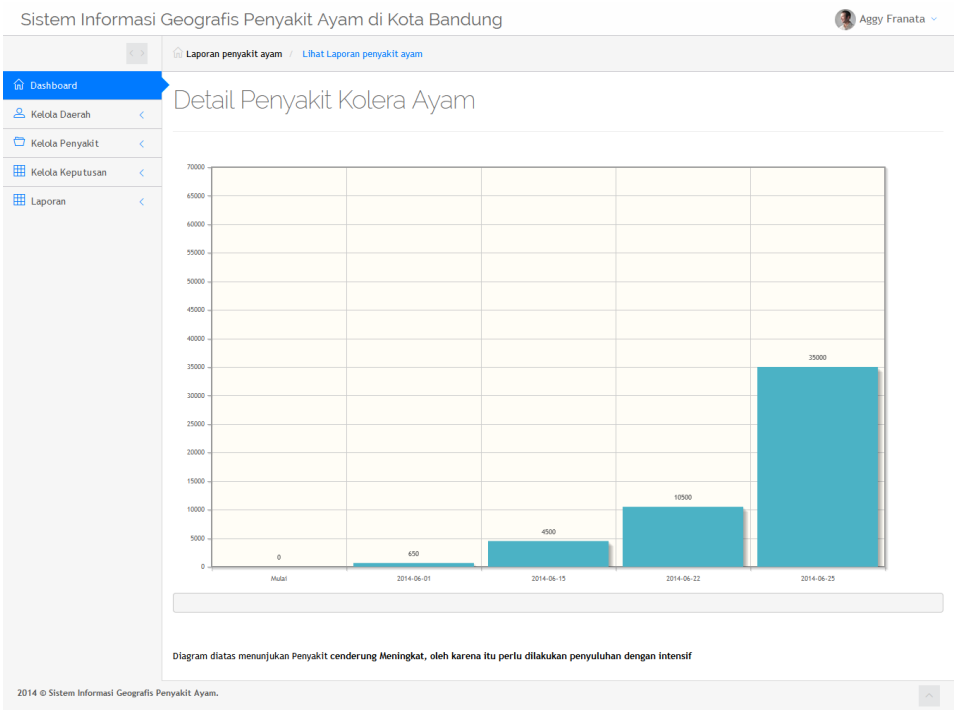
Berikut ini tampilan hasil pencarian data laporan penyakit ayam pada periode bulan Juni 2014 dengan nama penyakit ‘Berak Kapur’:




**Gambar 4.48** Hasil Pemilihan Kriteria Laporan Penyakit Ayam

Terdapat dua kecamatan dengan kasus yang sama, lokasi yang terdapat tanda berwarna biru merupakan kasus tertinggi dengan jumlah kasus dan jumlah ayam yang terjangkit sedangkan lokasi dengan tanda warna hijau daerah dengan kasus terendah. Pada peta juga muncul hasil keputusan yang diberikan oleh sistem mengenai cara pencegahan, pengobatan dan info penyakit ayam tersebut.

Di bagian bawah laporan terdapat rekapitulasi penyakit sesuai dengan jenis penyakit, yang mengindikasikan penyakit tersebut meningkat atau menurun dalam suatu daerah.




**Gambar 4.49** Hasil Rekapitulasi Penyakit

## 4.5 Pengujian

Pengujian merupakan bagian penting dari proses pengembangan sistem, karena pada tahap ini dilakukan pengujian dari sistem yang telah diimplementasikan sebelum benar-benar digunakan oleh *user*. Hal ini bertujuan untuk menentukan kesalahan, *error*, dan *bugs* yang mungkin saja terjadi pada saat proses pengolahan data yang tidak ditemukan pada proses pembuatan program.

### 4.5.1 Rencana Pengujian

Pengujian program aplikasi sistem informasi geografis untuk memantau lokasi penyebaran penyakit ayam di Kota Bandung dilakukan dengan menggunakan metode *black box*. Pengujian *black box* ini berfokus pada fungsi program dan hasil pengolahan (*output*) dari program tersebut.

Berikut ini beberapa tabel rencana pengujian yang dilakukan pada program aplikasi sistem informasi geografis untuk memantau lokasi penyebaran penyakit ayam di Kota Bandung:

**Tabel 4.6** Rencana Pengujian

| Item yang diuji | Deskripsi |
|---|---|
| Form Login | Melakukan pengujian apakah proses login berjalan sesuai dan authorisasi masing-masing *level user*. |
| Menu Kelola Daerah | Melakukan pengujian pada seluruh proses (tambah, ubah, hapus dan cetak) pada kelola daerah. |
| Menu Kelola Penyakit | Melakukan pengujian apakah seluruh proses (tambah, ubah, hapus dan cetak) pada kelola penyakit berjalan dengan baik. |
| Menu Kelola Keputusan | Melakukan pengujian apakah proses input data daerah dan data penyakit sesuai dengan prosedur system. Diuji apakah hasil input dapat memberikan keputusan. |
| Menu laporan | Melakukan pengujian apakah proses pemilihan daerah, penyakit dan keputusan sesuai dengan keluaran sistem. |

### 4.5.2 Kasus dan Hasil Pengujian

Pengujian program menggunakan metode *black box*, dilakukan dengan memberikan masukan (*input*) pada program yang kemudian disimpulkan hasilnya, apakah sesuai dengan yang diharapkan atau tidak. Di bawah ini dijelaskan beberapa kasus dan hasil pengujian program aplikasi sistem informasi geografis untuk memantau lokasi penyebaran penyakit ayam di Kota Bandung:

1. Pengujian Form Login

**Tabel 4.7** Pengujian Form Login

| Kasus dan Hasil Uji (Data Normal) | | |
|---|---|---|
| Data Masukan | Yang Diharapkan | Hasil Uji |
| *username* dan *password* benar | Dialihkan ke halaman home *user* dengan menu sesuai levelnya masing-masing. | [ ✔ ] Berhasil<br>[ ] Gagal |
| Kasus dan Hasil Uji (Data Salah) | | |
| Data Masukan | Yang Diharapkan | Hasil Uji |
| *username* dan *password* benar | Menampilkan kembali halaman *login* tersebut dan menampilkan pesan *login* gagal. | [ ✔ ] Berhasil<br>[ ] Gagal |

2. Pengujian Menu Kelola Daerah

   Menu Kelola Daerah merupakan menu yang berfungsi untuk memproses data-data dasar yang merupakan salah satu inputan utama sistem. Data ini merupakan data yang akan diolah pada aplikasi. Menu ini terdiri dari Tambah Daerah, Lihat Daerah

**Tabel 4.8** Pengujian Menu Kelola Daerah

| Kasus dan Hasil Uji (Data Normal) | | |
|---|---|---|
| Data Masukan | Yang Diharapkan | Hasil Uji |
| Nama kelurahan, alamat. | Data tersimpan ke *database* dan menampilkan pesan Berhasil Tambah Daerah. | [ ✔ ] Berhasil<br>[ ] Gagal |
| Data daerah yang akan diubah | Menampilkan data daerah pada form detail. Semua field tidak boleh kosong dan data daerah terupdate dengan baik. | [ ✔ ] Berhasil<br>[ ] Gagal |
| Data daerah | Jika ditekan tombol Hapus, sistem akan | [ ✔ ] Berhasil |

| | | |
|---|---|---|
| yang akan dihapus | menghapus data daerah tersebut. | [ ] Gagal |
| Kasus dan Hasil Uji (Data Salah) | | |
| Data Masukan | Yang Diharapkan | Hasil Uji |
| Nama kecamatan dan kelurahan kosong | Menampilkan pesan Gagal Tambah Daerah bahwa field harus diisi pada field yang tidak diisi. Proses simpan tidak dilanjutkan. | [ ✔ ] Berhasil<br>[ ] Gagal |

3. Pengujian Menu Kelola Penyakit

   Menu Kelola Penyakit merupakan salah satu inputan utama sistem. Data ini merupakan data yang akan diolah selanjutnya untuk pengambilan keputusan.

**Tabel 4.9** Pengujian Menu Kelola Penyakit

| Kasus dan Hasil Uji (Data Normal) | | |
|---|---|---|
| Data Masukan | Yang Diharapkan | Hasil Uji |
| Nama penyakit, Nama latin, Info penyakit, Pencegahan, Pengobatan | Data tersimpan ke *database* dan menampilkan pesan Berhasil Tambah Penyakit. | [ ✔ ] Berhasil<br>[ ] Gagal |
| Data penyakit yang akan diubah | Menampilkan data penyakit pada form detail. Semua field tidak boleh kosong dan data daerah terupdate dengan baik. | [ ✔ ] Berhasil<br>[ ] Gagal |
| Data penyakit yang akan dihapus | Jika ditekan tombol Hapus sistem akan menghapus data penyakit tersebut. | [ ✔ ] Berhasil<br>[ ] Gagal |
| Kasus dan Hasil Uji (Data Salah) | | |

| Data Masukan | Yang Diharapkan | Hasil Uji |
|---|---|---|
| Nama penyakit, Nama latin kosong | Menampilkan pesan Gagal Tambah Penyakit bahwa field harus diisi pada field yang tidak diisi. Proses simpan tidak dilanjutkan. | [ ✔ ] Berhasil<br>[ ] Gagal |

4. Pengujian Menu Kelola Keputusan

**Tabel 4.10** Pengujian Menu kelola Keputusan

| Kasus dan Hasil Uji (Data Normal) | | |
|---|---|---|
| Data Masukan | Yang Diharapkan | Hasil Uji |
| Nama daerah, nama penyakit, jumlah, tanggal, nama pelapor dan keterangan. | Data tersimpan ke *database* dan menampilkan pesan Berhasil Tambah Keputusan. | [ ✔ ] Berhasil<br>[ ] Gagal |
| Data keputusan yang akan dihapus | Jika ditekan tombol Hapus sistem akan menghapus data keputusan tersebut. | [ ✔ ] Berhasil<br>[ ] Gagal |
| Kasus dan Hasil Uji (Data Salah) | | |
| Data Masukan | Yang Diharapkan | Hasil Uji |
| Jumlah terjangkit, tanggal periksa kosong. | Menampilkan pesan Gagal Tambah Keputusan bahwa field harus diisi pada field yang tidak diisi. Proses simpan tidak dilanjutkan. | [ ✔ ] Berhasil<br>[ ] Gagal |

5. Pengujian Menu Laporan

**Tabel 4.11** Pengujian Menu Laporan

| Kasus dan Hasil Uji (Data Normal) | | |
|---|---|---|
| Data Masukan | Yang Diharapkan | Hasil Uji |
| Pilihan kriteria bulan, tahun periode dan nama penyakit. | Menampilkan laporan periode penyakit ayam, lokasi, jumlah kasus, jumlah terjangkit dan solusi sistem. | [ ✔ ] Berhasil<br>[ ] Gagal |
| Kasus dan Hasil Uji (Data Salah) | | |
| Data Masukan | Yang Diharapkan | Hasil Uji |
| Pilihan kriteria bulan, tahun dan penyakit kosong | Tidak ditemukan laporan penyakit ayam. | [ ✔ ] Berhasil<br>[ ] Gagal |

### 4.5.3 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian pada *item* dengan beberapa kasus di atas, maka dapat disimpulkan bahwa seluruh fungsi program yang diuji berjalan dengan baik dan sesuai dengan yang diharapkan.

Namun tidak menutup kemungkinan akan ditemukan kesalahan dan gangguan program pada waktu yang akan datang, sehingga akan dibutuhkan proses pemeliharaan dan perbaikan (*maintenance*) pada program aplikasi ini.