# BAB III
# PEMBAHASAN

## 3.1. Kegiatan Kerja Praktek

Kegiatan kerja praktek selama 20 hari di Grand Royal Panghegar Hotel Bandung mulai dari tanggal 16 Juli sampai tanggal 11 Agustus 2012, setiap hari Senin-Sabtu dari jam 08.00-16.00 WIB. Dalam melaksanakan kerja praktek ini penulis dibantu oleh pembimbing bernama **Iwan Darmawan, S.Kom.** sebagai Network and System Administrator. Kegiatan yang dilakukan ketika melakukan kerja praktek yaitu pengenalan lingkungan kerja, analisis sistem yang akan dibangun, perancangan, implementasi, pengujian, dan pemeliharaan sistem.

## 3.2. Analisis sistem

Analisis sistem adalah penguraian dari satu sistem yang utuh ke dalam bagian-bagian komponen dengan maksud untuk mengidentifikasi dan mengevaluasi masalah, sehingga diperoleh solusi. Analisis merupakan tahapan paling penting karena kesalahan pada tahap ini menyebabkan kesalahan ditahap selanjutnya.

### 3.2.1. Analisis Masalah

Departemen IT (*Information Technology*) adalah unit yang bertangggung jawab dalam hal pengelolaan teknologi informasi Grand Royal Panghegar Hotel Bandung. Salah satu diantaranya adalah membuat suatu sistem yang terintegrasi untuk mendeteksi adanya gejala kebakaran.

Pada saat ini, peralatan utama yang menjadi pengendali gejala kebakaran yaitu *Main Control Fire Alarm* (MCFA) dengan dengan bantuan interface pada komputer untuk memonitoringnya, dan HT (*Handy Talky)* sebagai alat komunikasi antara operator MCFA dengan *Firebrigade* yang ada di lapangan ketika terdeteksi gejala kebakaran untuk konfirmasi keadaan sebenarnya.

Namun komunikasi menggunakan HT yang persediaannya terbatas dan ketika menggunakan HT pada saluran yang sama tentunya semua yang sedang

berada pada saluran tersebut dapat mendengarnya, sehingga dapat memperlambat tindak lanjuti.

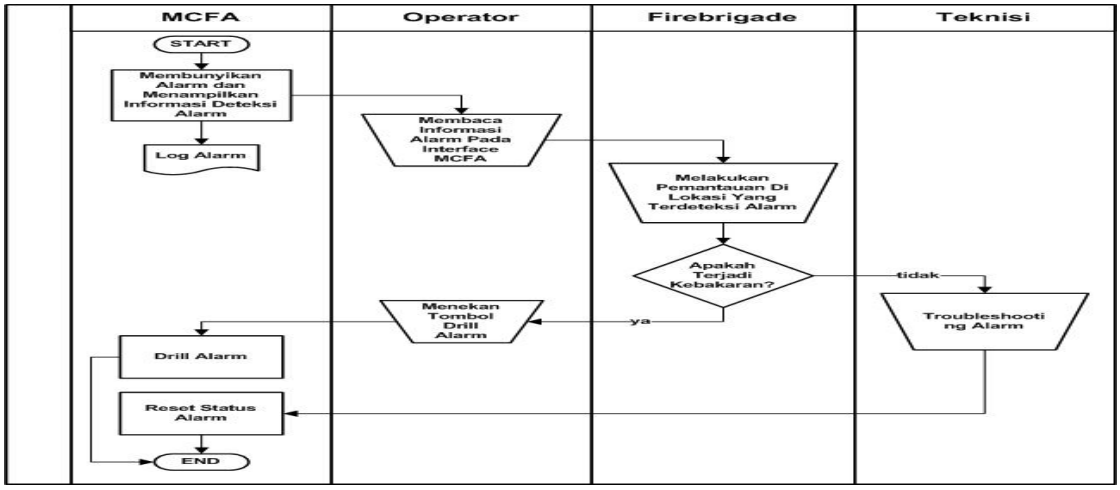
Melihat permasalah tersebut, Departemen IT membutuhkan aplikasi yang dapat mempercepat tindakan dan memudahkan komunikasi antara operator dengan *firebrigade* ketika terdeteksi adanya gejala kebakaran.

### 3.2.2. Analisis Sistem Yang Sedang Berjalan

**Analisis Sistem Yang Sedang Berjalan**

Analisis sistem yang sedang berjalan berisi tentang pemaparan prosedur tindakan ketika terdeteksi gejala kebakaran yang ada saat ini. Analisis ini dimaksudkan agar perangkat lunak yang dibangun tidak keluar dari prosedur yang telah ada. Adapun prosedur tindakan ketika terdeteksi adanya gejala kebakaran di Grand Royal Panghegar Hotel Bandung adalah :

1) MCFA membunyikan alarm peringatan terdeteksinya gejala kebakaran / gangguan pada perangkat alarm.
2) Operator membaca informasi pada interface MCFA kemudian menyampaikannya kepada *firebrigade.*
3) *Firebrigade* menuju lokasi yang terdeteksi adanya gejala kebakaran untuk memantau keadaan sebenarnya
4) *Firebrigade* melaporkan hasil pemantauan kepada operator MCFA. Jika hasil pemantauan adalah benar-benar terjadi gejala kebakaran, maka operator melakukan drill alarm dan *firebrigade* melakukan tindak lanjut penanganan kebakaran. Jika hasil pemantauan ternyata bukan gejala kebakaran, maka operator akan menghubungi teknisi untuk memperbaikin trouble pada perangkat alarm.




**Gambar 3.1 Prosedur Tindakan Gejala Kebakaran**

### 3.2.3. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan nonfungsional menggambarkan kebutuhan luar sistem yang diperlukan untuk menjalankan aplikasi yang dibangun. Adapun kebutuhan nonfungsional untuk menjalankan sistem ini meliputi: kebutuhan perangkat keras, kebutuhan perangkat lunak, dan pengguna. Analisis kebutuhan nonfungsional bertujuan agar aplikasi yang dibangun dapat digunakan sesuai dengan kebutuhan sistem.

#### 3.2.3.1. Analisis Perangkat Keras

Untuk menjalankan suatu aplikasi maka diperlukan perangkat keras yang dapat mendukung proses kerja dari sistem itu sendiri. Spesifikasi perangkat kerass yang digunakan dalam pembangunan sistem adalah sebagai berikut.

**Tabel 3.1 Spesifikasi Perangkat Keras (Pembangunan)**

| No | Perangkat Keras | Spesifikasi |
|---|---|---|
| 1 | Processor | Intel Core 2 Duo |
| 2 | Monitor | Monitor LCD 14.1” |
| 3 | VGA | Intel(R) Chipset 1GB |
| 4 | Memori | 4 GB |
| 5 | Keyboard | Standar |
| 6 | Mouse | Standar |
| 7 | Modem | Wavecom MB130B-Q2406B |

Spesifikasi perangkat keras yang digunakan oleh operator di ruangan kontrol tertera dalam tabel 3.2s.

**Tabel 3.2 Spesifikasi Perangkat Keras (Yang Digunakan di Ruangan Kontrol)**

| No | Perangkat Keras | Spesifikasi |
|---|---|---|
| 1 | Processor | Intel Core 2 Duo |
| 2 | Monitor | Monitor LCD 17” |
| 3 | VGA | Intel(R) Chipset |
| 4 | Memori | 1 GB |
| 5 | Keyboard | Standar |
| 6 | Mouse | Standar |
| 7 | Modem | Wavecom MB130B-Q2406B |

Berdasarkan perbadingan antara spesifikasi perangkat keras ketika pembangunan dengan perangkat keras yang sudah ada, spesifikasi perangkat keras yang ada sudah mencukupi untuk menjalankan sistem yang akan dibangun di Grand Royal Panghegar Hotel Bandung

### 3.2.3.2. Analisis Perangkat Lunak

Dalam analisis kebutuhan perangkat lunak, dibutuhkan beberapa *software* yang mendukung untuk digunakan dalam implementasi. *Software* yang sudah terpasang pada komputer di ruang kontrol Grand Royal Panghegar Hotel Bandung adalah sebagai berikut :

**Tabel 3.3 Spesifikasi Perangkat Lunak**

| No | Jenis |
|---|---|
| 1 | Adobe Reader 9 |
| 2 | FARO LS |
| 3 | Fireworks |
| 4 | Hauppage Windows NT Driver |
| 5 | Hauppage WinTV OCX |
| 6 | High Definition Audio Driver Package |
| 7 | Microsoft .NET Framework 4 |
| 8 | Microsoft Offie Enterprise 2007 |
| 9 | Microsoft SQL Server Desktop Engine |
| 10 | Microsoft Visual C++ |

Berdasarkan tabel 3.3 diatas, *software* yang telah terpasang pada komputer di ruangan kontrol belum memenuhi kriteria untuk implementasi sistem. Maka dari itu perlu penambahan *software* untuk implementasi sebagai berikut :

**Tabel 3.4 Spesifikasi Perangkat Lunak untuk Implementasi**

| No | Jenis | Perangkat Lunak |
|---|---|---|
| 1 | Bahasa Pemrograman | HTML |
| | | PHP |
| | | CSS |
| | | SQL |
| | | Delphi |
| 2 | DBMS | MySQL |
| | | Sybase SQL Anywhere |
| 3 | Development Tools | Notepad++ |
| | | StarUML |
| | | Microsoft Office Visio 2007 |
| | | CodeIgniter |
| | | XAMPP |
| | | Apache |
| 4 | SMS Gateway | Gammu |

### 3.2.3.3. Analisis Pengguna

Suatu aplikasi akan berjalan dengan optimal apabila ditunjang oleh perangkat pikir yang memliki kemampuan dalam menjalankan aplikasi yang bersangkutan. Karakteristik pengguna yang dibutuhkan adalah sebagai berikut sebagai berikut :

**Tabel 3.5 Karakteristik Pengguna**

| Pengguna | Tanggung Jawab | Hak Akses | Tingkat Pendidikan | Tingkat Keterampilan |
|---|---|---|---|---|
| Operator | Mengirim Notifikasi Alarm dan Mengelola data Firebrigade | Hak Akses Penuh | Min D3 Jurusan Teknik Informatika | Bisa mengikuti pentunjuk dan mengoperasikan komputer |

Karakteristik operator yang ada saat ini di Grand Royal Panghegar Hotel Bandung adalah sebagai berikut :

**Tabel 3.6 Karakteristik Pegawai**

| Jabatan | Tingkat Pendidikan | Tingkat Keterampilan |
|---|---|---|
| Staf IT | S1 Jurusan Teknik Informatika | Mahir mengikuti pentunjuk dan mengoperasikan komputer |

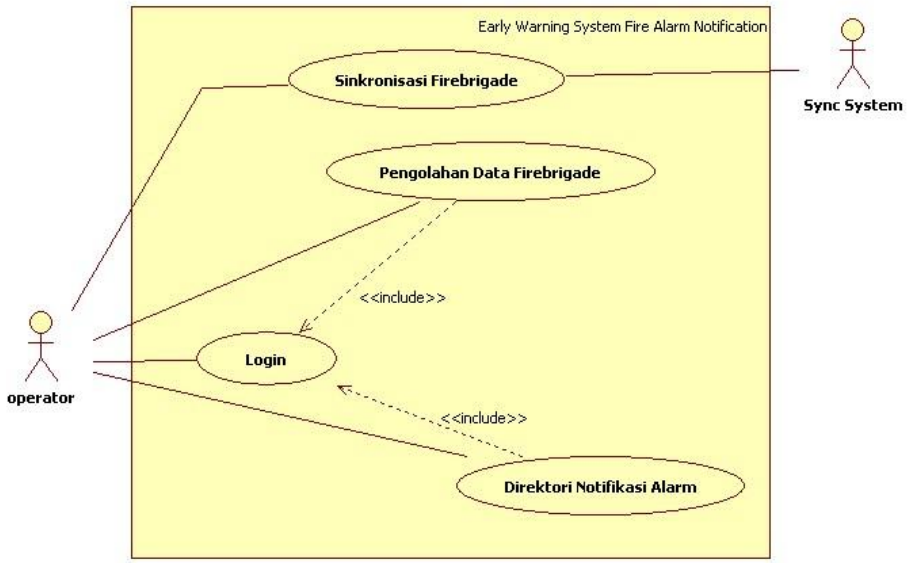
Berdasarkan perbandingan antara karakteristik pengguna yang dibutuhkan dengan karakteristik operator yang ada saat ini, operator sudah memenuhi syarat untuk menjalankan aplikasi yang akan dibangun di Grand Royal Panghegar Hotel Bandung ini.

### 3.2.4. Analisis Kebutuhan Fungsional

Analisis sistem yang dilakukan menggunakan tools UML, adapun tahapan analisis sistem menggunakan UML adalah *use case diagram, use case scenario, activity diagram, state diagram, sequence diagram,* dan *class diagram.*

#### 3.2.4.1. Use Case Duagram

*Use Case Diagram* merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antar aktor dengan aktifitas yang terdapat pada sistem. Sasaran pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan yang disepakati antara pemakai dan pengembang. Dari analisis pengguna aplikasi yang ada maka *use case* diagram untuk aplikasi *early warning system alarm notification* di Grand Royal Panghegar Hotel Bandung dapat dilihat dalam gambar 3.2.




**Gambar 3.2 Use Case Diagram Early Warning System Alarm Notification**

### 3.2.4.2. Use Case Scenario

*Use Case Scenario* mendeskripsikan urutan langkah-langkah dalam proses bisnis baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. Berdasarkan *use case diagram* pada gambar 3.2 maka *use case scenario* unutk aplikasi yang dibangun akan dijelaskan sebagai berikut :

#### 3.2.4.2.1. Use Case Scenario Login

*Use Case Scenario* login dapat dilihat pada tabel 3.7.

**Tabel 3.7 Use Case Scenario Login**

| **Identifikasi** | |
|---|---|
| **Nomor** | 1 |
| **Nama** | Login |
| **Tujuan** | Login untuk masuk ke sistem |
| **Deskripsi** | Proses ini untuk validasi user yang berhak menjalankan aplikasi |
| **Aktor** | Operator |
| **Use Case yang berkaitan** | Pengolahan Data Firebrigade, Direktori Notifikasi Alarm |
| **Skenario Utama** | |
| **Kondisi Awal** | *Form* menu login ditampilkan |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Mengisi username dan password | |
| 2. Menekan tombol login | |
| | 3. Menampilkan pesan bahwa berhasil login |
| **Skenario Alternatif (Proses Gagal)** | |
| | 1. Menampilkan pesan bahwa gagal login |
| 2. Mengisi kembali username dan password | |
| 3. Menekan tombol login | |
| **Kondisi Akhir** | Sistem akan menampilkan *form* menu utama (web) |

### 3.2.4.2.2. Use Case Scenario Sinkronisasi Firebrigade

*Use Case Scenario* sinkronisasi firebrigade dapat dilihat pada tabel 3.8.

**Tabel 3.8 Use Case Scenario Sinkronisasi Firebrigade**

| **Identifikasi** | |
|---|---|
| **Nomor** | 2 |
| **Nama** | Sinkronisasi Firebrigade |
| **Tujuan** | Mensinkronkan data firebrigade yang aktif |
| **Deskripsi** | Proses ini untuk mencari data firebrigade yang aktif kemudian mensinkronkannya dengan database |
| **Aktor** | Operator, Sync System |
| **Use Case yang berkaitan** | - |
| **Skenario Utama** | |
| **Kondisi Awal** | Form Menu Utama Sinkronisasi Firebrigade (Desktop) |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Sync System secara otomatis mencari data firebrigade yang aktif | |
| | 2. Menghasilkan data firebrigade yang aktif |
| **Skenario Alternatif (Manual Sinkron)** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Operator menekan tombol manual sync | |
| | 2. Menampilkan data firebrigade yang aktif |
| **Kondisi Akhir** | Sistem menyimpan data firebrigade yang aktif |

### 3.2.4.2.3. Use Case Scenario Pengolahan Data Firebrigade

*Use Case Scenario* pengolahan data firebrigade dapat dilihat pada tabel 3.9.

**Tabel 3.9 Use Case Scenario Pengolahan Data Firebrigade**

| **Identifikasi** | |
|---|---|
| **Nomor** | 3 |
| **Nama** | Pengolahan Data Firebrigade |
| **Tujuan** | Mengolah Data Firebrigade |
| **Deskripsi** | Proses ini untuk mengelola data firebrigade mulai dari menambah, mengubah, menghapus data firebrigade yang digunakan sebagai kode acuan untuk mengirim notifikasi alarm. |
| **Aktor** | Operator |
| **Use Case yang berkaitan** | Login |
| **Skenario Utama** | |
| **Kondisi Awal** | Form Menu Data Firebrigade |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Memilih tombol menu pilihan Firebrigade (view firebrigade/add firebrigade) | |
| | 2. Menampilkan form menu yang dipilih oleh aktor (view firebrigade/add firebrigade) |

### 3.2.4.2.4. Use Case Scenario Direktori Notifikasi Alarm

*Use Case Scenario* direktori notifikasi alarm dapat dilihat pada tabel 3.10.

**Tabel 3.10 Use Case Scenario Direktori Notifikasi Alarm**

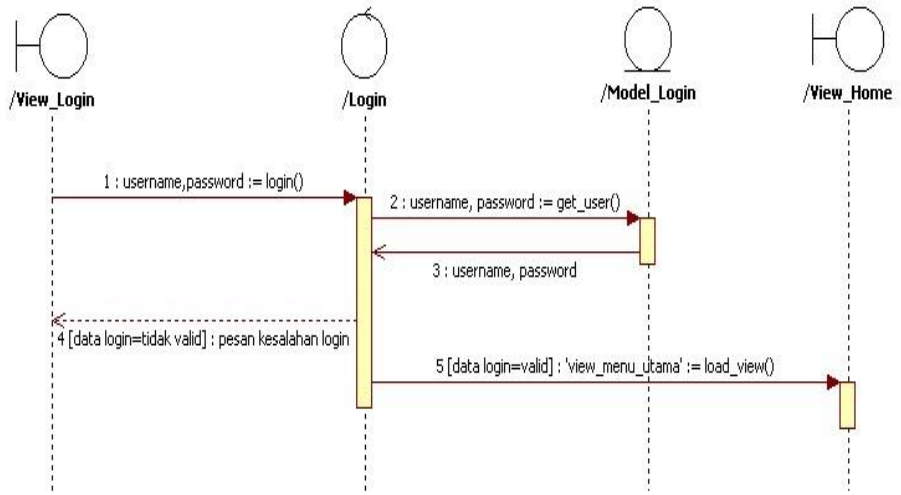
| **Identifikasi** | |
|---|---|
| **Nomor** | 4 |
| **Nama** | Direktori Notifikasi Alarm |
| **Tujuan** | Mengirim alarm, menampilkan data alarm |
| **Deskripsi** | Proses ini untuk mengirim notifikasi alarm, melihat alarm yang telah dibalas, melihat status alarm terkirim, dan melihat alarm yang gagal terkirim |
| **Aktor** | Operator |
| **Use Case yang berkaitan** | Login |
| **Skenario Utama** | |
| **Kondisi Awal** | Form Notifikasi Alarm |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Menilih tombol menu pilihan notifikasi alarm (send alarm/ replied alarm/sent alarm/ failed) | |
| | 2. Menampilkan form menu yang dipilih oleh aktor (send alarm/ replied alarm/sent alarm/ failed) |

### 3.2.4.3. Sequence Diagram

*Sequence Diagram* menggambarkan interaksi antar masing-masing objek pada setiap *use case* dalam urutan waktu. Interaksi ini berupa pengiriman data antar objek-objek yang saling berinteraksi. *Sequence Diagram* pada aplikasi Grand Royal Panghegar Early Warning System Fire Alarm Notification terdiri dari *sequence diagram* login, *sequence diagram* sinkronisasi firebrigade, *sequence diagram* pengolahan data firebrigade, dan *sequence diagram* direktori notifikasi alarm. Berikut ini penjelasan dari masing-masing *sequence diagram.*

### 3.2.4.3.1. Sequence Diagram Login

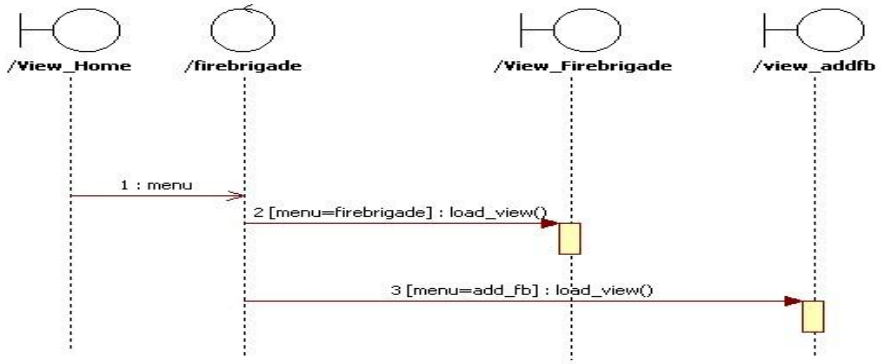
*Sequence Diagram Login* dapat dilihat pada gambar 3.3




**Gambar 3.3 Sequence Diagram Login**

### 3.2.4.3.2. Sequence Diagram Pengolahan Data Firebrigade

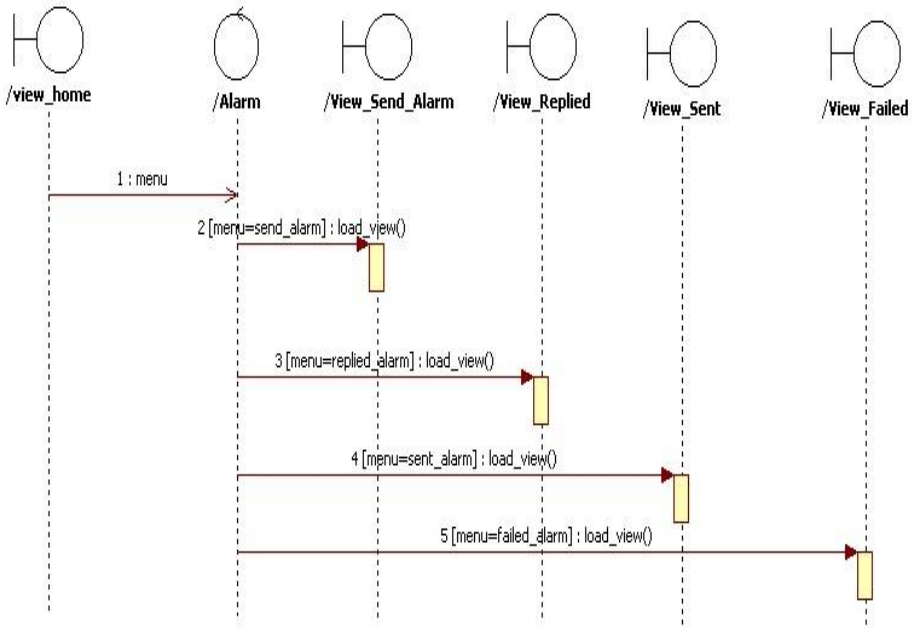
*Sequence Diagram* pengolahan data firebrigade dapat dilihat pada gambar 3.5.




**Gambar 3.5 Sequence Diagram Pengolahan Data Firebrigade**

### 3.2.4.3.3. Sequence Diagram Direktori Notifikasi Alarm

*Sequence Diagram* direktori notifikasi alarm dapat dilihat pada gambar 3.6




**Gambar 3.6 Sequence Diagram Direktori Notifikasi Alarm**

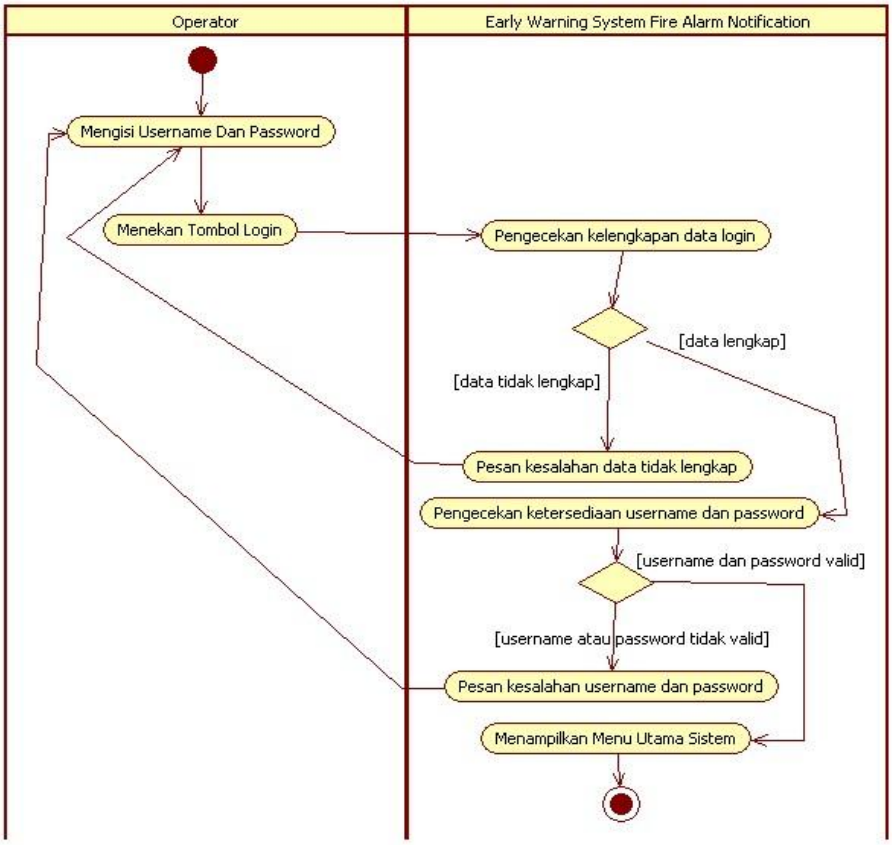
### 3.2.4.4. Activity Diagram

*Activity Diagram* menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses, dan digunakan pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. *Activity Diagram* sangat bermanfaat untuk memahami proses dari sistem secara keseluruhan. *Activity Diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram.*

*Activity Diagram* pada aplikasi Grand Royal Panghegar Early Warning System Fire Alarm Notification terdiri dari *activity diagram* login, *activity diagram* sinkronisasi firebrigade, *activity diagram* pengolahan data firebrigade, dan *activity diagram* direktori notifikasi alarm. Berikut ini penjelasan dari masing-masing *activity diagram.*

### 3.2.4.4.1. Activity Diagram Login

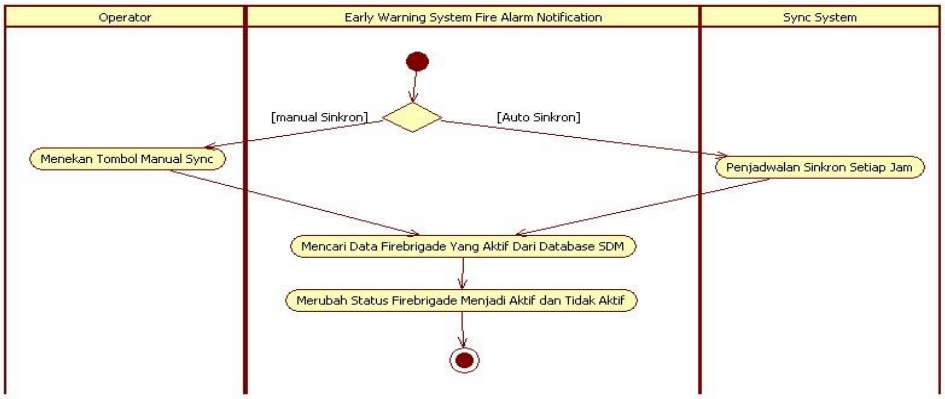
*Activity Diagram* Login dapat dilihat pada gambar 3.7




**Gambar 3.7 Activity Diagram Login**

### 3.2.4.4.2. Activity Diagram Sinkronisasi Firebrigade

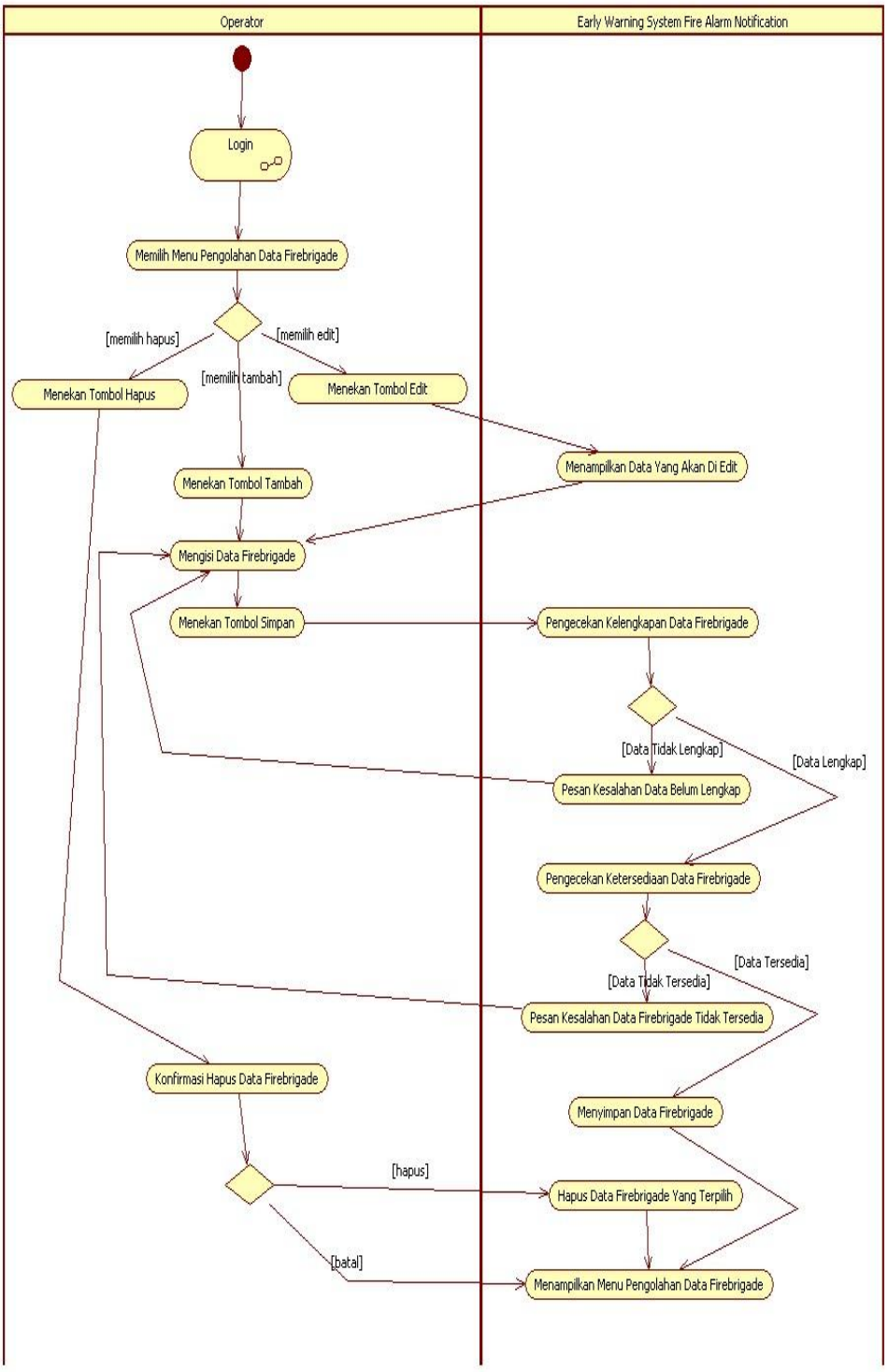
*Activity Diagram* Sinkronisasi Firebrigade dapat dilihat pada gambar 3.8




**Gambar 3.8 Activity Diagram Sinkronisasi Firebrigade**

### 3.2.4.4.3. Activity Diagram Pengolahan Data Firebrigade

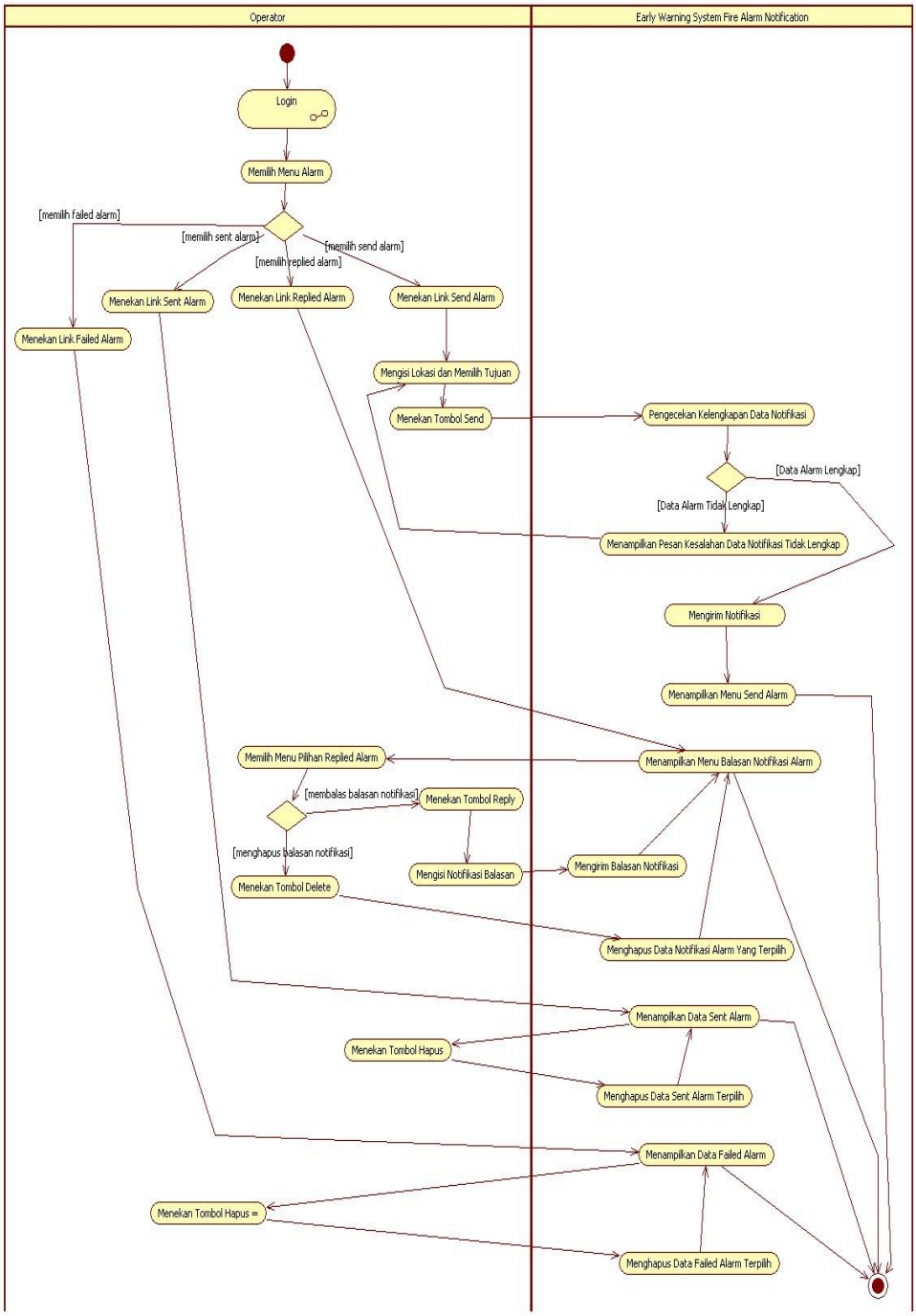
*Activity Diagram* pengolahan data firebrigade dapat dilihat pada gambar 3.9




**Gambar 3.9 Activity Diagram Pengolahan Data Firebrigade**

### 3.2.4.4.4. Activity Diagram Direktori Notifikasi Alarm

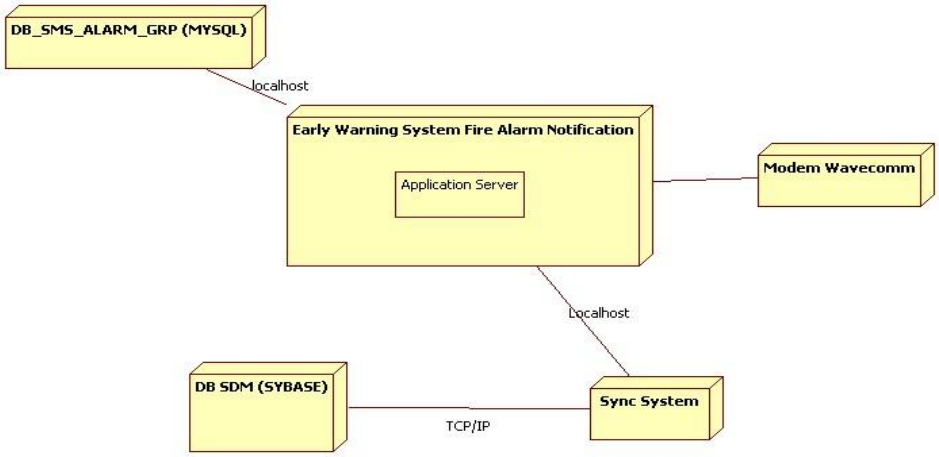
*Activity Diagram* direktori notifikasi alarm dapat dilihat pada gambar 3.10




**Gambar 3.10 Activity Diagram Notifikasi Alarm**

### 3.2.4.5. Deployment Diagram

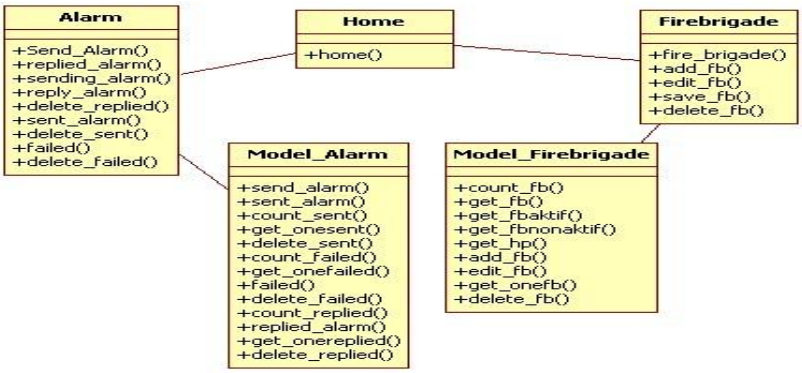
*Deployment diagram* menggambarkan proses-proses yang berbeda pada sistem yang berjalan dan bagaimana relasi di dalamnya. Adapun *deployment diagram* pada sistem ini dapat dilihat pada gambar 3.11




**Gambar 3.11 Delpoyment Diagram**

### 3.2.4.6. Class Diagram

*Class Diagram* menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. Struktur itu meliputi atribut dan *method* yang ada pada masing-masing *class*. Hubungannya dari masing-masing *class* yang ada tersebut digambarkan dengan menggunakan pewarisan dan generalisasi, hubungan statis / *associations*, agregasi, komposisi, *depedency*, dan realisasi. *Class Diagram* pada sistem ini terlihat pada gambar 3.12




**Gambar 3.12 Class Diagram**

## 3.3. Perancangan Sistem

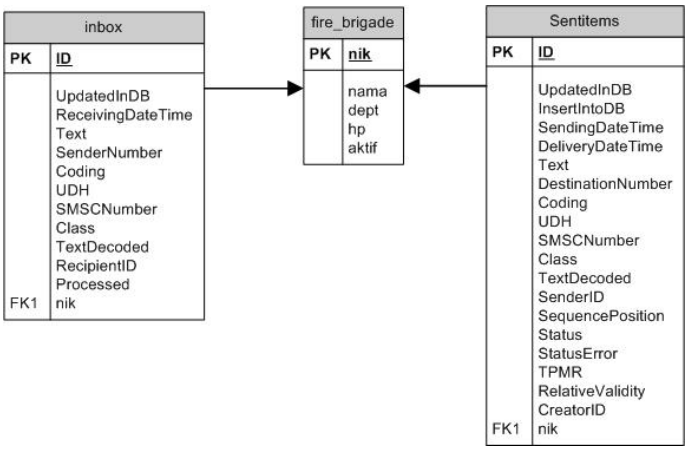
Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi.

### 3.3.1. Perancangan Data

Perancangan data merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai. Perencangan basis data terbagi menjadi dua yaitu skema relasi dan perancangan struktur tabel.

#### 3.3.1.1. Skema Relasi

Skema relasi merupakan rangkaian hubungan antara dua tabel atau lebih pada sistem basis data. Gambar 3.13 merupakan penjelasan rangkaian basis data pada aplikasi ini.




**Gambar 3.13 Skema Relasi**

#### 3.3.1.2. Struktur Tabel

Struktur tabel menggambarkan detail tabel yang berisi *field,* tipe data, panjang data, dan keterangan lainnya. Tabel yang terdapat pada basis data aplikasi ini adalah tabel fire_brigade, inbox, dan tabel sentitems. Berikut ini deskripsi dari masing-masing tabel tersebut.

1. **Tabel fire_brigade**

**Tabel 3.11 Struktur Tabel fire_brigade**

| Nama Field | Tipe Data | Panjang | Null | Kunci |
|---|---|---|---|---|
| Nik | Integer | 4 | No | Primary key |
| nama | Varchar | 100 | No | - |
| dept | Varchar | 100 | No | - |
| Hp | Varchar | 12 | No | - |
| aktif | Boolean | - | No | - |

2. **Tabel inbox**

**Tabel 3.12 Struktur Tabel inbox**

| Nama Field | Tipe Data | Panjang | Null | Kunci |
|---|---|---|---|---|
| ID | Integer | 10 | No | Primary key |
| ReceivingDateTime | Timestamp | - | No | - |
| Text | Text | - | No | - |
| SenderNumber | varchar | 20 | No | - |
| Coding | Enum | | No | - |
| UDH | Text | | No | - |
| SMSCNumber | varchar | 20 | No | - |
| Class | Integer | 11 | No | - |
| TextDecoded | varchar | 160 | No | - |
| UpdatedInDB | Timestamp | - | No | - |
| RecipientID | Text | - | No | - |
| Processed | Enum | - | No | - |
| Nik | Integer | 4 | - | Foreign key |

3. **Tabel sentitems**

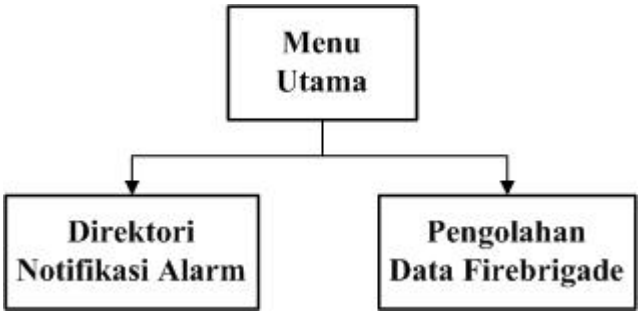
**Tabel 3.13 Struktur Tabel sentitems**

| Nama Field | Tipe Data | Panjang | Null | Kunci |
|---|---|---|---|---|
| ID | Integer | 10 | No | Primary key |
| InsertedIntoDB | Timestamp | - | No | - |
| SendingDateTime | Timestamp | - | No | - |
| DeliveryDateTime | Timestamp | - | Yes | - |
| Text | Text | - | No | - |
| DestinationNumber | Varchar | 20 | No | - |
| Coding | Enum | - | No | - |
| UDH | Text | - | No | - |
| SMSCNumber | Varchar | 20 | No | - |
| Class | Integer | 11 | No | - |
| TextDecoded | Varchar | 160 | No | - |
| UpdatedInDB | Timestamp | - | No | - |
| SenderID | Varchar | 255 | No | - |
| SequencePosition | Integer | 11 | No | - |

| | | | | |
|---|---|---|---|---|
| Status | Enum | - | No | - |
| StatusError | Integer | 11 | No | - |
| TPMR | Integer | 11 | No | - |
| RelativeValidity | Integer | 11 | No | - |
| CreatorID | Integer | 11 | No | - |
| nik | Integer | 4 | No | Foreign key |

### 3.3.2. Struktur Menu

Perancangan struktur menu merupakan gambaran jalur pemakaian aplikasi sehingga aplikasi yang dibangun mudah dipahami dan mudah digunakan. perancangan struktur menu dari Early Warning System Fire Alarm Notification Grand Royal Panghegar Hotel Bandung dapat dilihat pada gambar 3.14.




**Gambar 3.14 Struktur Menu**

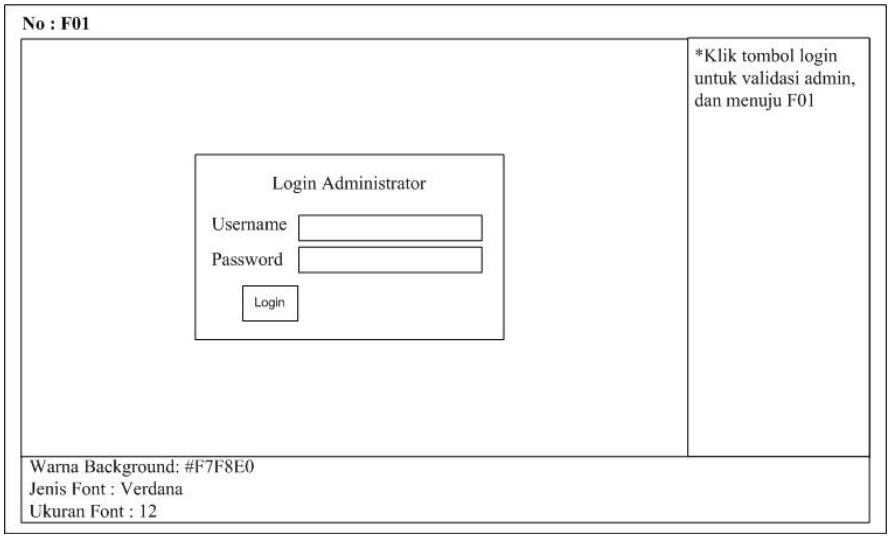
### 3.3.3. Perancangan Antar Muka

Perancangan antarmuka mendeskripsikan rencana tampilan dari setiap tampilan yang akan digunakan pada aplikasi ini. Perancangan antarmuka pada aplikasi ini terdiri dari perancangan *form* dan pesan.

#### 3.3.3.1. Perancangan Form

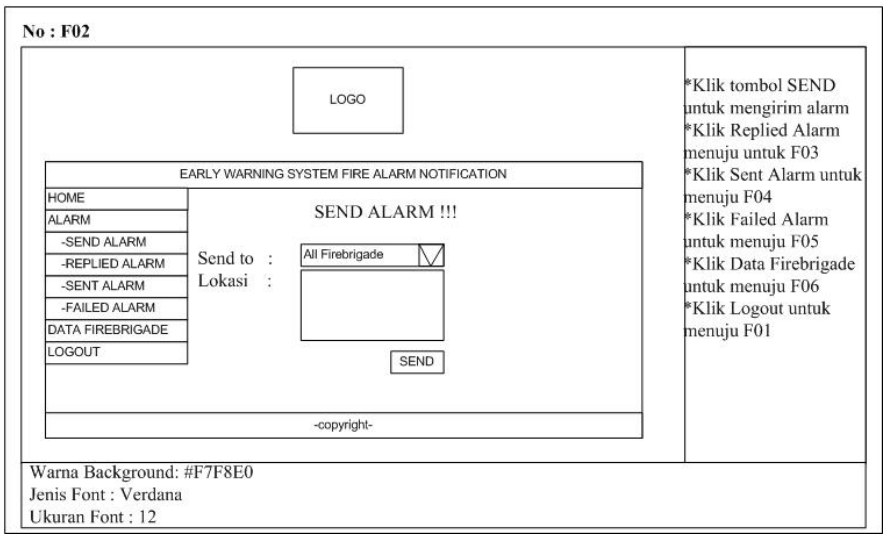
Perancangan antarmuka *form* pada aplikasi ini dapat dilihat pada Gambar 3.15 sampai dengan gambar 3.23 :

## 1. Form Login



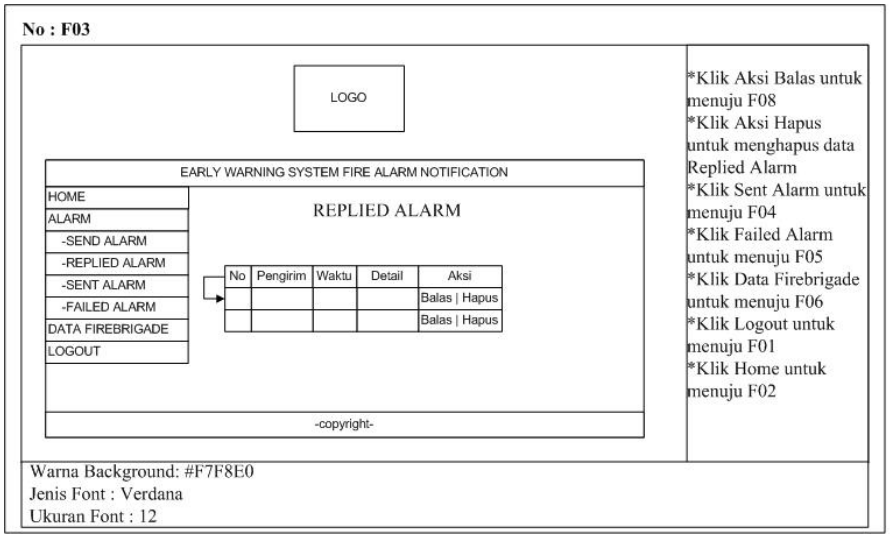

**Gambar 3.15 Form Login**

## 2. Form Send Alarm



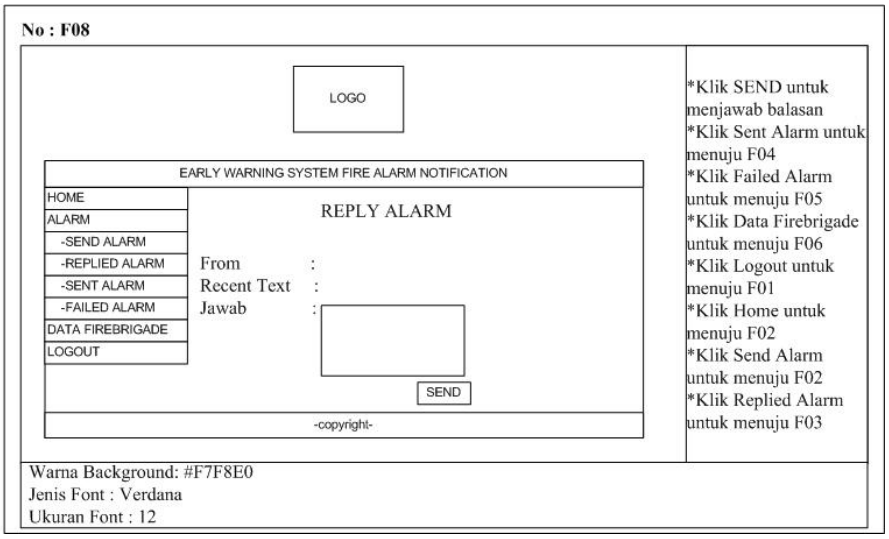

**Gambar 3.16 Form Send Alarm**

## 3. Form Replied Alarm



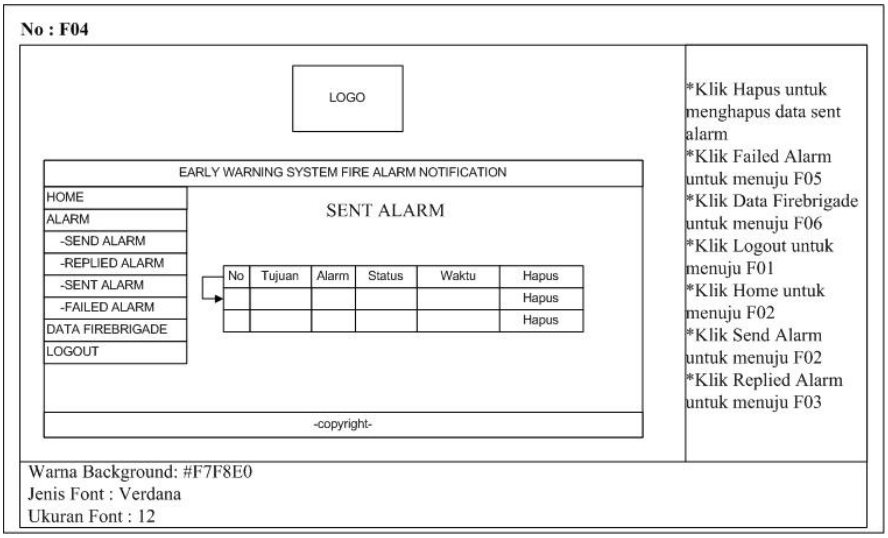

**Gambar 3.17 Form Replied Alarm**

## 4. Form Jawab Balasan Alarm



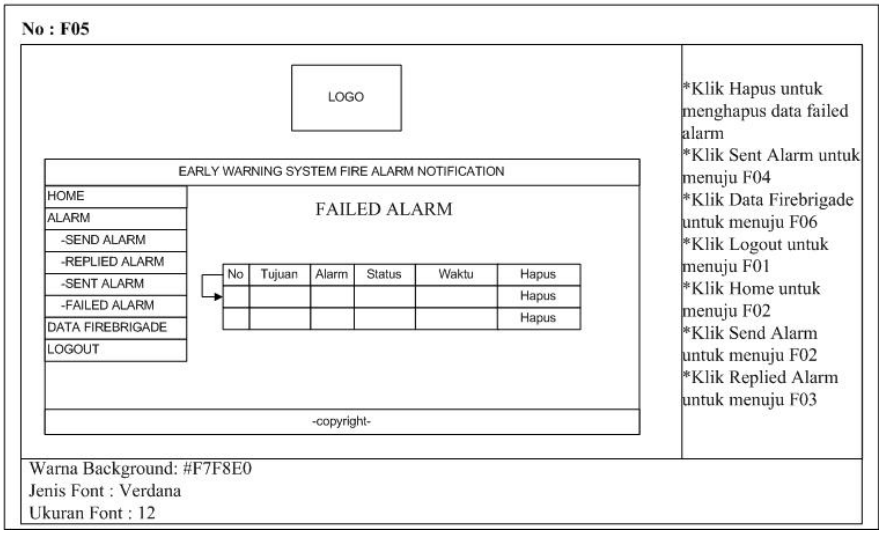

**Gambar 3.18 Form Balasan Alarm**

## 5. Form Sent Alarm



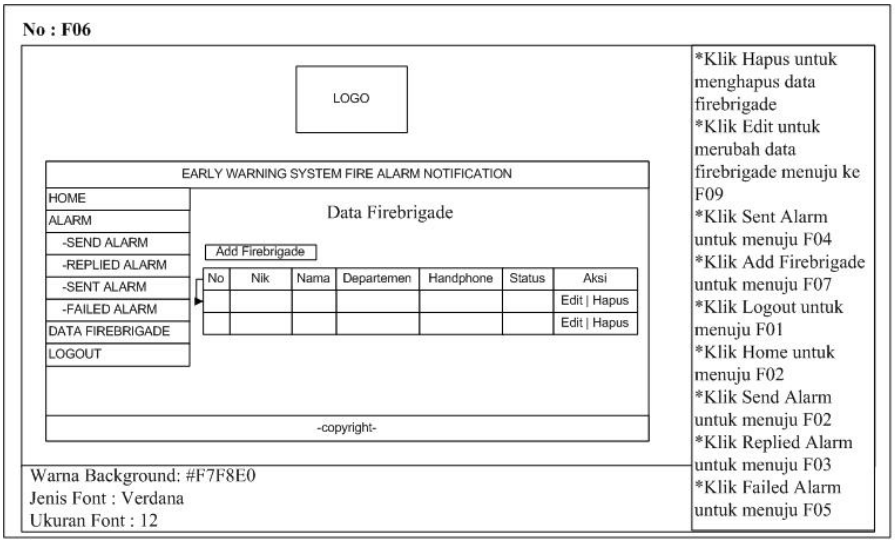

**Gambar 3.19 Form Sent Alarm**

## 6. Form Failed Alarm



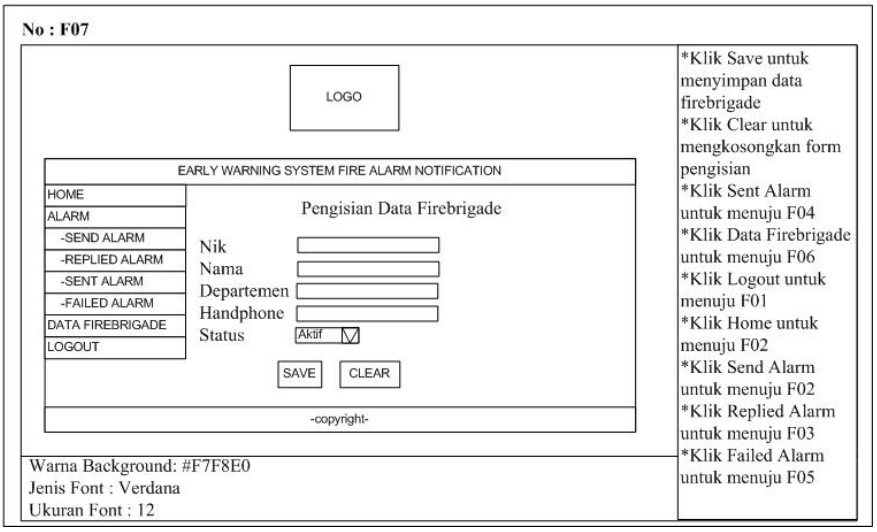

**Gambar 3.20 Form Failed Alarm**

## 7. Form Pengolahan Data Firebrigade



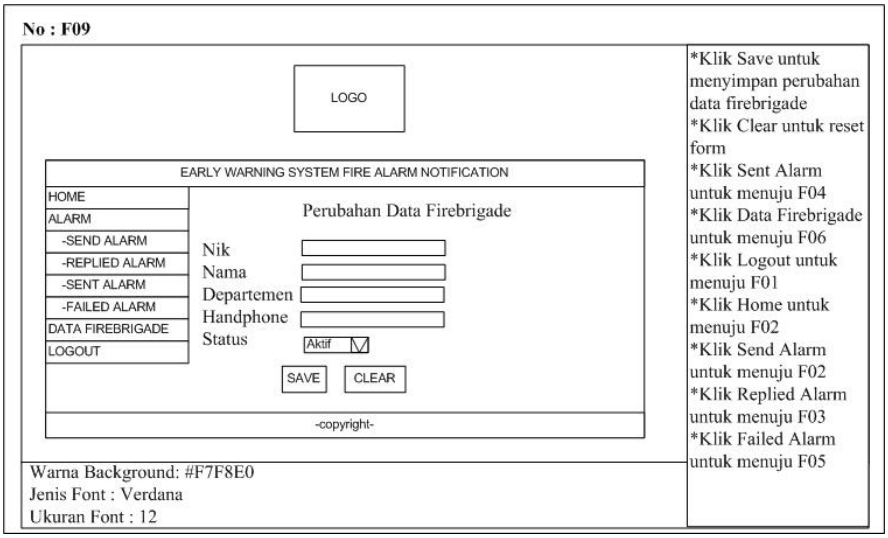

**Gambar 3.21 Form Pengolahan Data Firebrigade**

## 8. Form Tambah Firebrigade




**Gambar 3.22 Form Tambah Firebrigade**

## 9. Form Edit Firebrigade

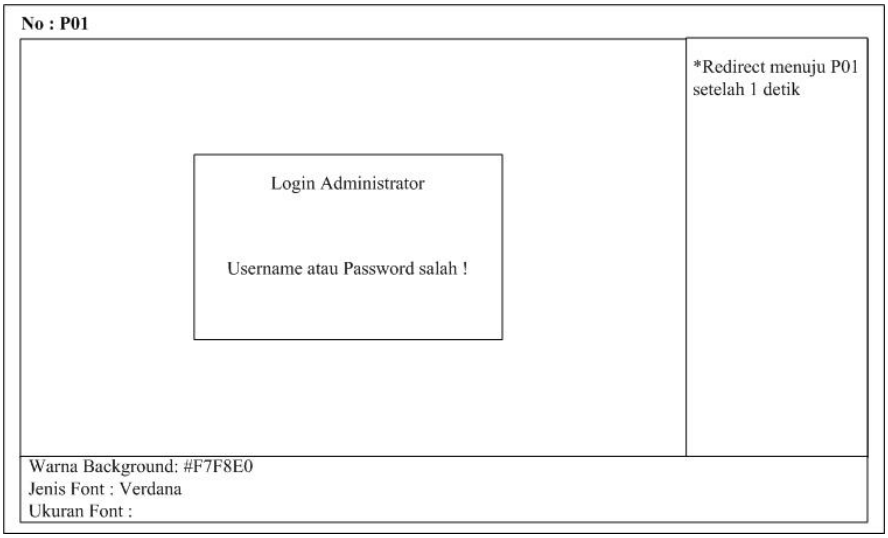



**Gambar 3.23 Form Edit Firebrigade**

### 3.3.3.2. Perancangan Pesan

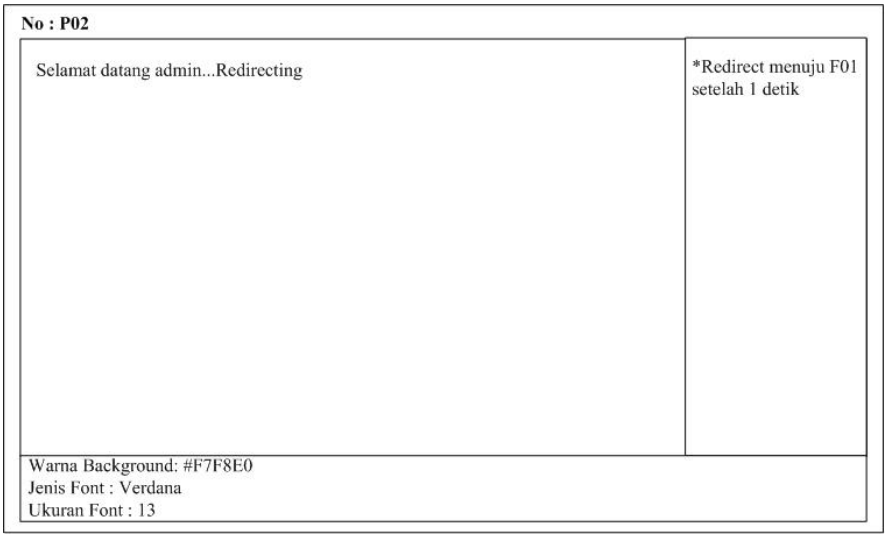
Perancangan antarmuka pesan pada aplikasi ini dapat dilihat pada Gambar 3.24 sampai dengan gambar 3.34

## 1. Perancangan Pesan P01



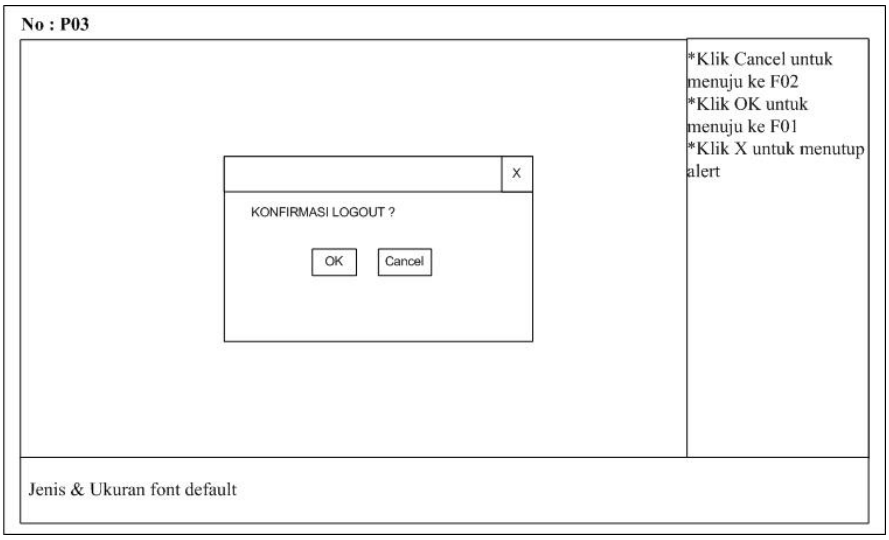

**Gambar 3.24 Perancangan Pesan P01**

## 2. Perancangan Pesan P02



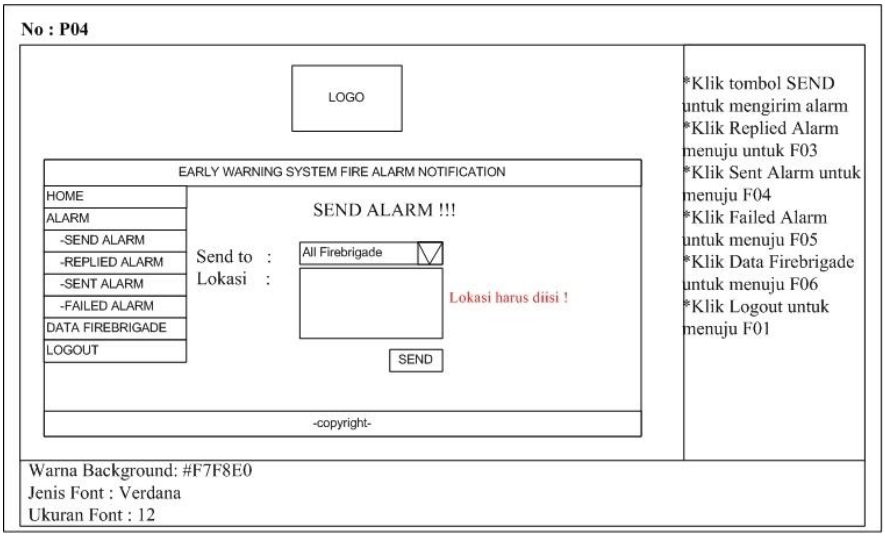

**Gambar 3.25 Perancangan Pesan P02**

## 3. Perancangan Pesan P03



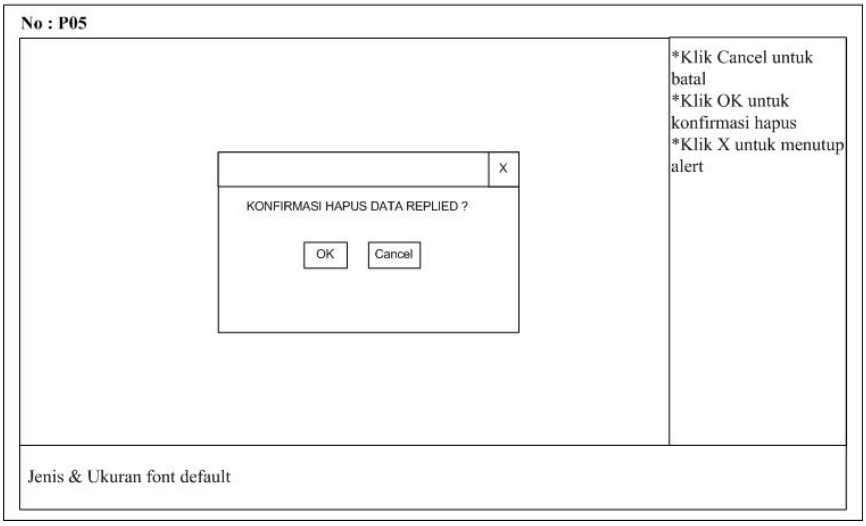

**Gambar 3.26 Perancangan Pesan P03**

## 4. Perancangan Pesan P04



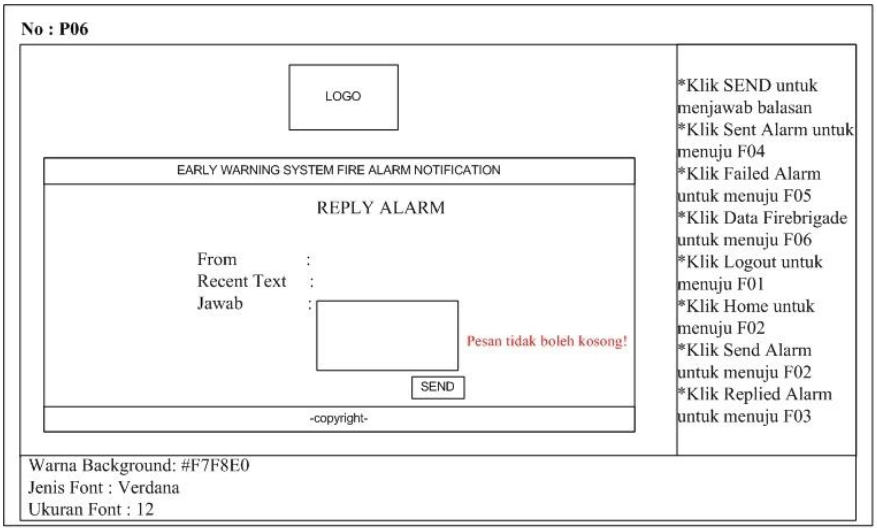

**Gambar 3.27 Perancangan Pesan P04**

## 5. Perancangan Pesan P05



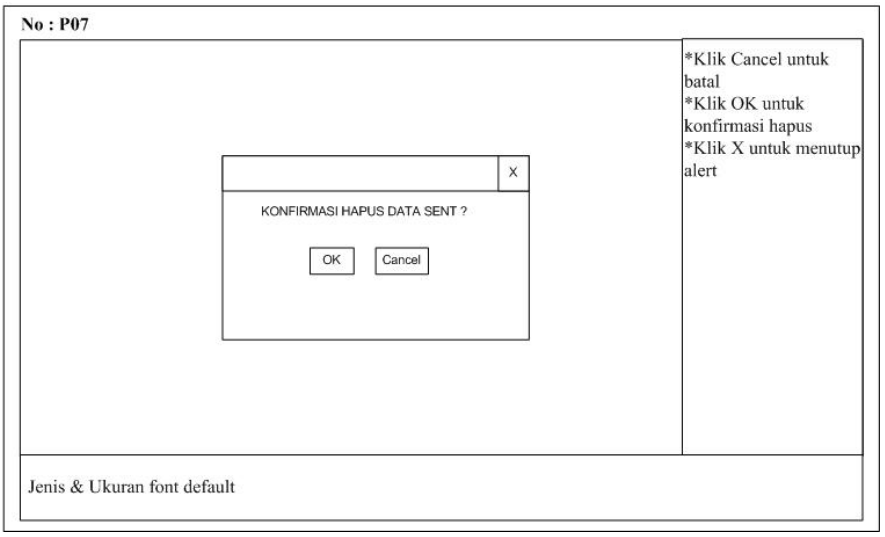

**Gambar 3.28 Perancangan Pesan P05**

## 6. Perancangan Pesan P06



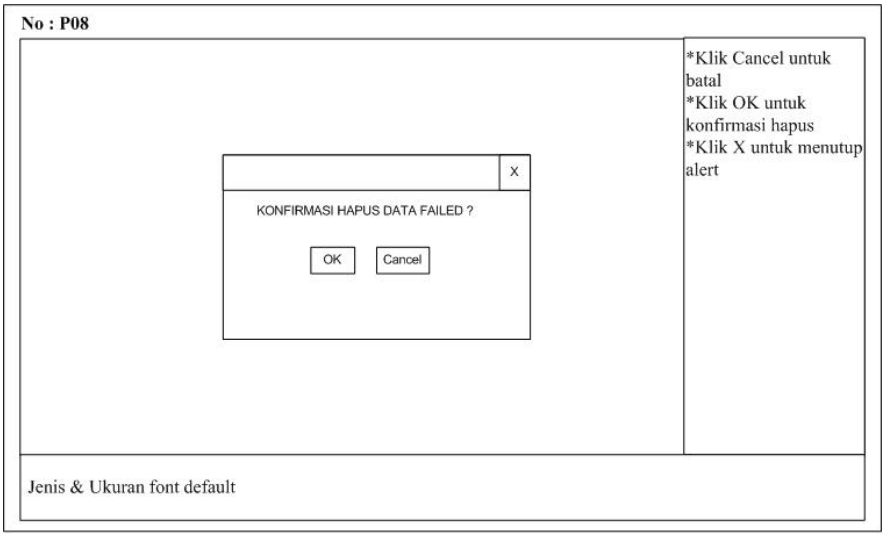

**Gambar 3.29 Perancangan Pesan P06**

## 7. Perancangan Pesan P07



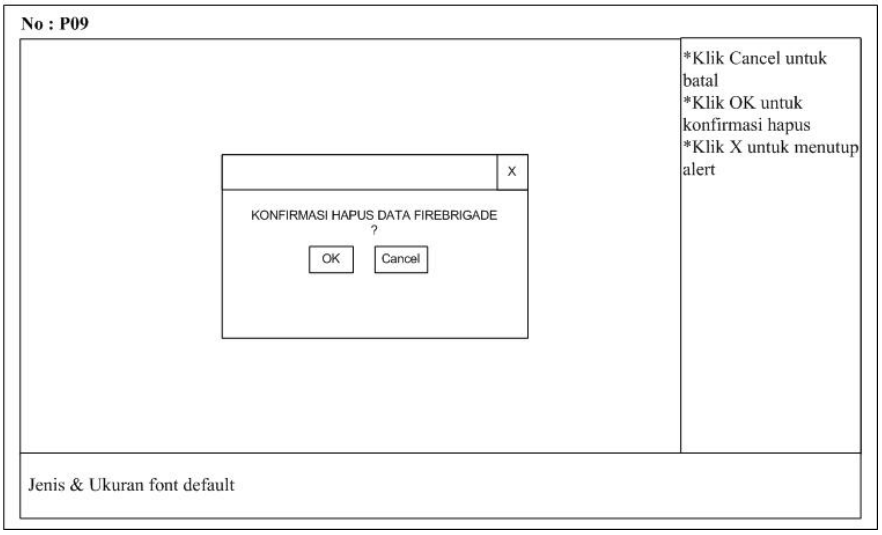

**Gambar 3.30 Perancangan Pesan P07**

## 8. Perancangan Pesan P08



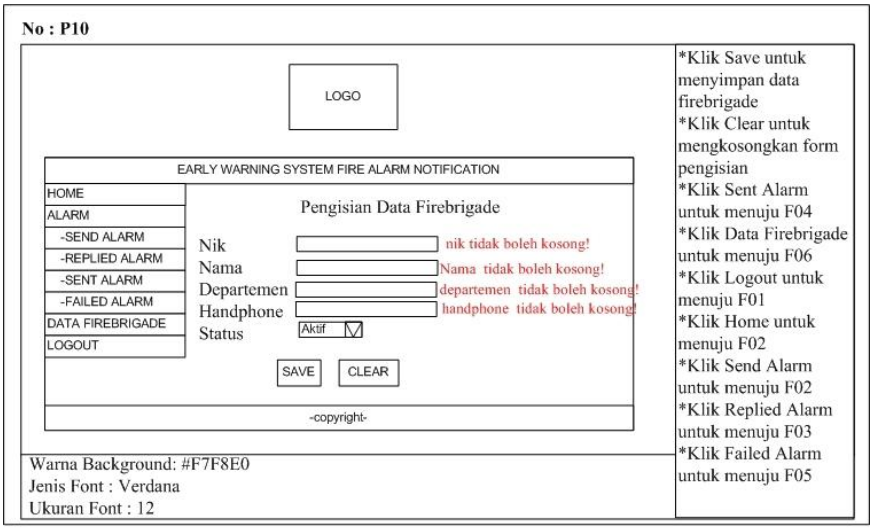

**Gambar 3.31 Perancangan Pesan P08**

## 9. Perancangan Pesan P09



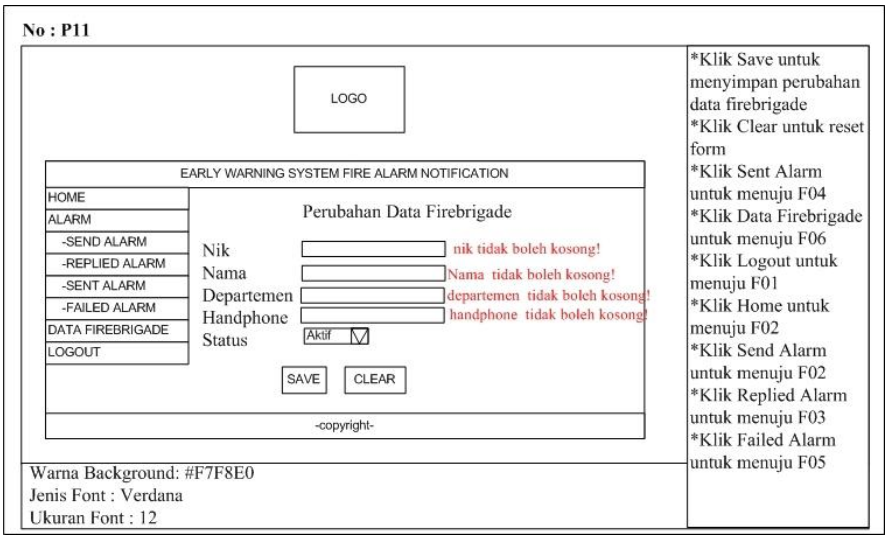

**Gambar 3.32 Perancangan Pesan P09**

## 10. Perancangan Pesan P10




**Gambar 3.33 Perancangan Pesan P10**

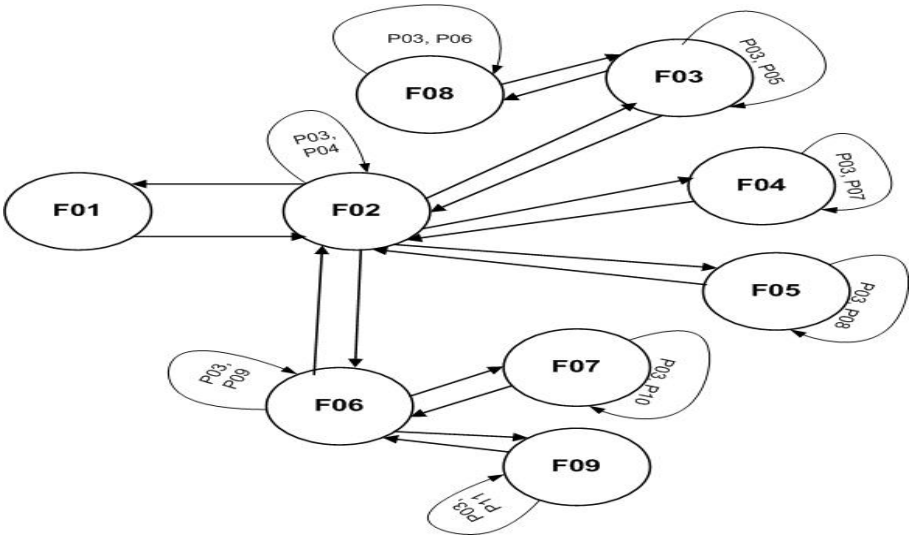
## 11. Perancangan Pesan P11




**Gambar 3.34 Perancangan Pesan P11**

### 3.3.4. Jaringan Semantik

Jaringan semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar berbagai objek. Jaringan semantik terdiri dari lingkaran-lingkaran yang menunjukkan objek informasi tentang objek-objek tertentu. Jaringan semantik untuk aplikasi ini dapat dilihat pada gambar 3.35

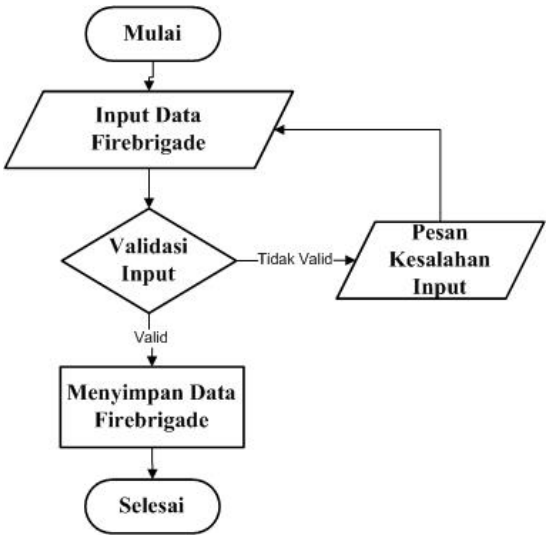



**Gambar 3.35 Jaringan Semantik**

### 3.3.5. Perancangan Prosedural

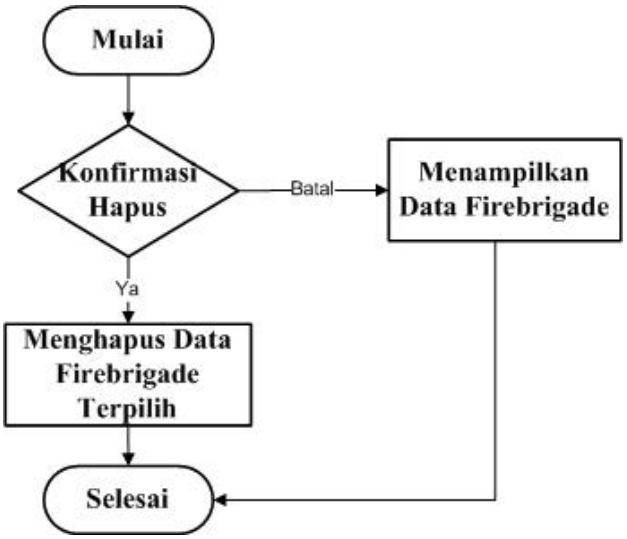
Perancangan proseduran merupakan representasi algoritma dari perangkat lunak yang dibangun. Perancangan proseduran pada aplikasi ini dapat dilihat dari gambar 3.36 sampai dengan gambar 3.44

**1. Perancangan prosedural penambahan data firebrigade**



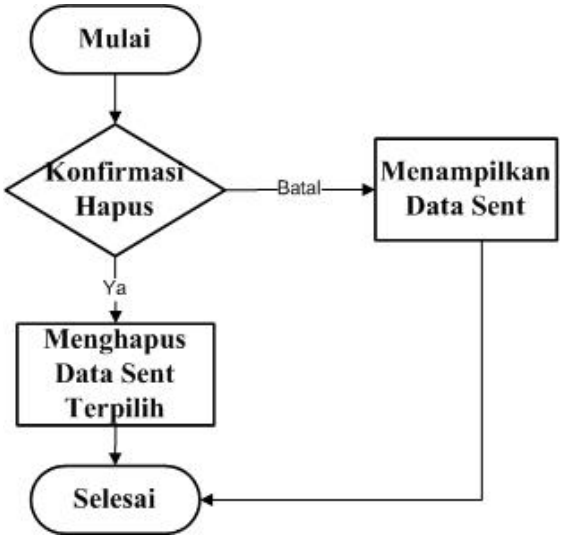

**Gambar 3.36 Flowchart penamahan data firebrigade**

**2. Perancangan prosedural hapus data firebrigade**



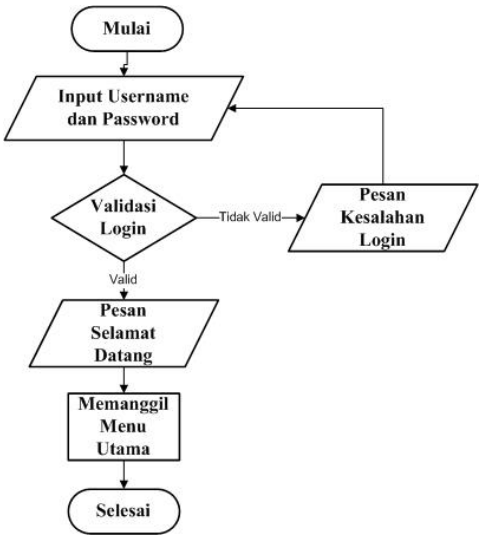

**Gambar 3.37 Flowchart hapus data firebrigade**

**3. Perancangan proseduran hapus sent alarm**



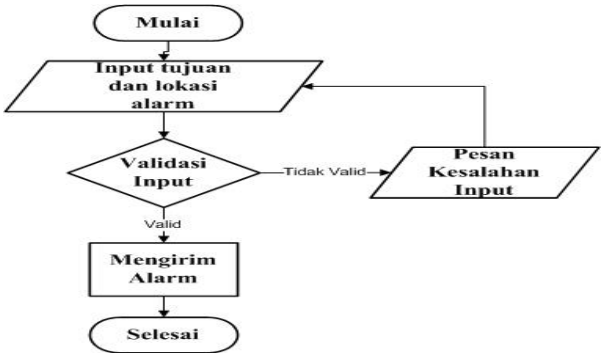

**Gambar 3.38 Flowchart hapus sent alarm**

**4. Perancangan prosedural login**



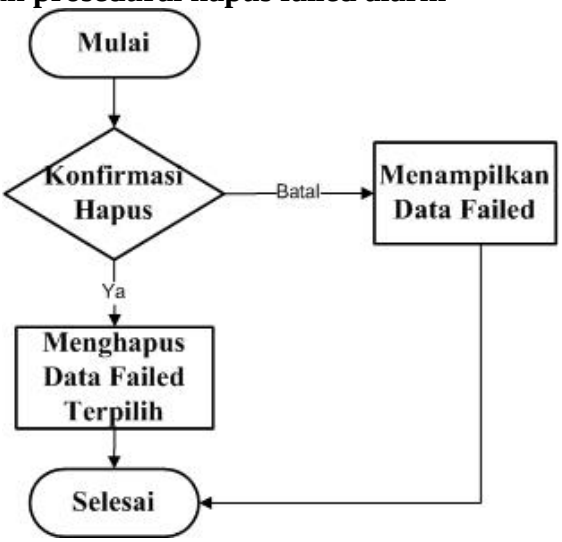

**Gambar 3.39 Flowchart login**

**5. Perancangan prosedural send alarm**



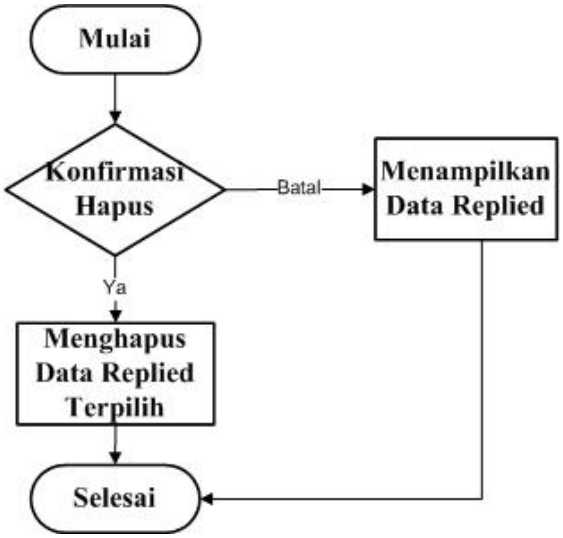

**Gambar 3.40 Flowchart send alarm**

**6. Perancangan prosedural hapus failed alarm**



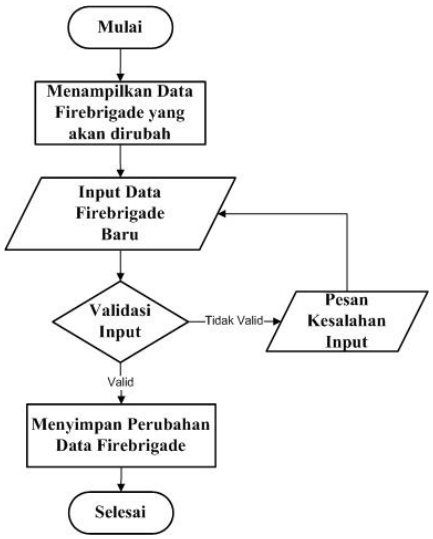

**Gambar 3.41 Flowchart hapus failed alarm**

**7. Perancangan prosedural hapus replied alarm**



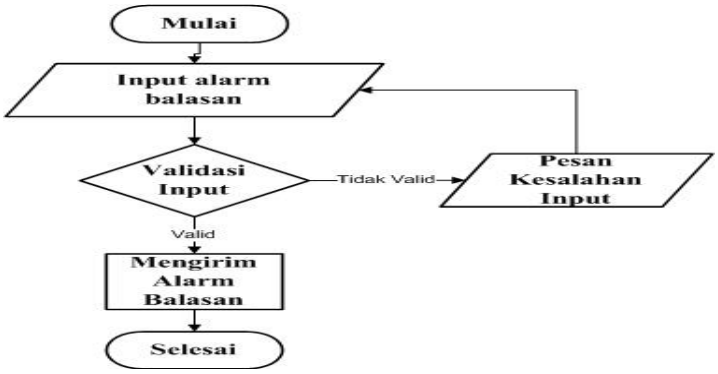

**Gambar 3.42 Flowchart hapus replied alarm**

**8. Perancangan prosedural perubahan data firebrigade**




**Gambar 3.43Flowchart edit firebrigade**

**9. Perancangan prosedural balas replied alarm**




**Gambar 3.44 Flowchart reply alarm**

## 3.4. Implementasi

Setelah sistem dianalisis dan didesain secara rinci maka selanjutnya akan menuju tahap implementasi. Tujuan implementasi adalah untuk menjelaskan tentang manual modul kepada semua user yang akan menggunakan sistem. Sehingga user tersebut dapat merespon apa yang ditampilkan di sistem dan memberikan masukan kepada pembuat sistem untuk dilakukan perbaikan agar sistem lebih baik lagi

### 3.4.1. Kebutuhan Perangkat Keras Pembangun

Spesifikasi perangkat keras yang digunakan untuk membangun aplikasi ini adalah :

**Tabel 3.14 Spesifikasi Perangkat Keras (Pembangunan)**

| No | Perangkat Keras | Spesifikasi |
|---|---|---|
| 1 | Processor | Intel Core 2 Duo |
| 2 | Monitor | Monitor LCD 14.1” |
| 3 | VGA | Intel(R) Chipset 1GB |
| 4 | Memori | 4 GB |
| 5 | Keyboard | Standar |
| 6 | Mouse | Standar |
| 7 | Modem | Wavecom MB130B-Q2406B |

### 3.4.2. Kebutuhan Perangkat Lunak Pembangun

Beberapa perangkat lunak pendukung yang dapat mengoptimalkan program yang akan dibangun, diantaranya adalah sebagai berikut.

**Tabel 3.15 Spesifikasi Perangkat Lunak untuk Implementasi**

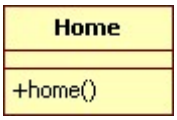
| No | Jenis | Perangkat Lunak |
|---|---|---|
| 1 | Bahasa Pemrograman | HTML |
| | | PHP |
| | | CSS |
| | | SQL |
| | | Delphi |
| 2 | DBMS | MySQL |
| | | Sybase SQL Anywhere |
| 3 | Development Tools | Notepad++ |
| | | StarUML |
| | | Microsoft Office Visio 2007 |
| | | CodeIgniter |
| | | XAMPP |

| | | |
|---|---|---|
| | | Apache |
| 4 | SMS Gateway | Gammu |

### 3.4.3. Implementasi Kelas

#### 3.4.3.1. Kelas Controller Home

Kelas Home merupakan kelas yang digunakan untuk menampilkan tampilan awal sistem




**Gambar 3.45 Kelas Controller Home**

**A. Deskripsi Atribut**

Tidak memiliki atribut

**B. Deskripsi Layanan**

Operasi – operasi yang ada pada kelas ini adalah :

1) Home(), *method* untuk menampilkan tampilan awal sistem.

#### 3.4.3.2. Kelas Controller Alarm




**Gambar 3.46 Kelas Controller Alarm**

**A. Deskripsi Atribut**

Tidak memiliki atribut

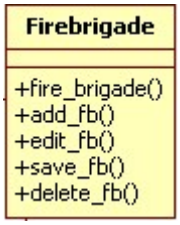
**B. Deskripsi Layanan**

Operasi – operasi yang ada pada kelas ini adalah :

1) Send_alarm(), *method* untuk menampilkan form send alarm
2) Replied_alarm(), *method* untuk menampilkan data replied_alarm
3) Sending_alarm(), *method* untuk mengirimkan notifikasi alarm ke firebrigade
4) Reply_alarm(), *method* untuk membalas replied_alarm
5) Delete_replied(), method untuk menghapus data replied_alarm

6) Sent_alarm(), method untuk menampilkan data sent_alarm
7) Delete_sent(), method untuk menghapus data sent_alarm
8) Failed(), method untuk menampilkan data failed_alarm
9) Delete_failed(), method untuk menghapus data failed_alarm

### 3.4.3.3. Kelas Controller Firebrigade




**Gambar 3.47 Kelas Controller Firebrigade**

**A. Deskripsi Atribut**

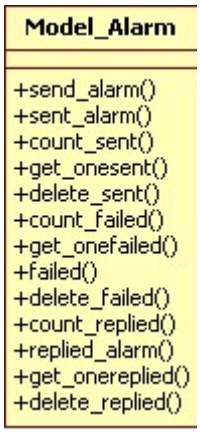
Tidak memiliki atribut

**B. Deskripsi Layanan**

Operasi – operasi yang ada pada kelas ini adalah :

1) Fire_brigade(), method untuk menampilkan data firebrigade
2) Add_fb(), method untuk menambah data friebrigade
3) Edit_fb(), method untuk merubah data firebrigade
4) Save_fb(), method untuk menyimpan data firebrigade
5) Delete_fb(), method untuk menghapus data firebrigade

### 3.4.3.4. Kelas Model_Alarm




**Gambar 3.48 Kelas Model_Alarm**

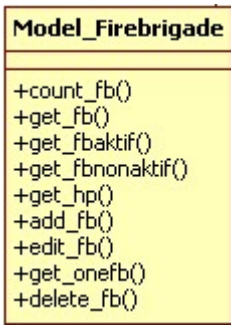
**A. Deskripsi Atribut**

Tidak memiliki atribut

**B. Deskripsi Layanan**

1) Send_alarm(), method untuk mengirim notifikasi alarm
2) Sent_alarm(), method untuk menampilkan data sent_alarm
3) Count_sent(), method untuk menghitung jumlah data sent_alarm
4) Get_onesent(), method untuk mengambil satu data sent_alarm terpilih
5) Delete_sent(), method untuk menghapus data sent_alarm
6) Count_failed(), method untuk menghitung jumlah data failed_alarm
7) Get_onefailed(), method untuk mengambil satu data failed_terpilih
8) Failed(), method untuk menampilkan data failed_alarm
9) Delete_failed(), method untuk menghapus data failed_alarm
10) Count_replied(), method untuk menghitung jumlah data replied_alarm
11) Replied_alarm(), method untuk menampilkan
12) Get_onereplied(), method untuk mengambil satu data replied_alarm
13) Delete_replied(), method untuk menghapus data replied_alarm

### 3.4.3.5. Kelas Model_Firebrigade




**Gambar 3.49 Kelas Model_Firebrigade**

**A. Deskripsi Atribut**

Tidak memiliki atribut

**B. Deskripsi Layanan**

Operasi – operasi yang ada pada kelas ini adalah :

1) Count_fb(), method untuk menghitung jumlah data firebrigade
2) Get_fb(), method untuk menampilkan data semua firebrigade
3) Get_fbaktif(), method untuk menampilkan data firebrigade yang aktif
4) Get_fbnonaktif(), method untuk menampilkan data firebrigade yang tidak aktif
5) Get_hp(), method untuk mengambil no handphone firebrigade

6) Add_fb(), method untuk menambah data firebrigade
7) Edit_fb(), method untuk mengubah data firebrigade
8) Get_onefb(), method untuk mendapatkan satu data firebrigade
9) Delete_fb(), method untuk menghapus data firebrigade

### 3.4.4. Implementasi Basis Data

Database Management Sistem (DBMS) yang digunakan untuk mengimplementasikan data pada sistem ini adalah MySQL 5.5.20. implementasi database dalam aplikasi yang telah dirancang adalah sebagai berikut :

1) Tabel inbox

```
CREATE TABLE IF NOT EXISTS `inbox` (
	`UpdatedInDB` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
	`ReceivingDateTime` timestamp NOT NULL DEFAULT '0000-
00-00 00:00:00',
	`Text` text NOT NULL,
	`SenderNumber` varchar(20) NOT NULL DEFAULT '',
	`Coding`
enum('Default_No_Compression','Unicode_No_Compression','8bit',
'Default_Compression','Unicode_Compression') NOT NULL DEFAULT
'Default_No_Compression',
	`UDH` text NOT NULL,
	`SMSCNumber` varchar(20) NOT NULL DEFAULT '',
	`Class` int(11) NOT NULL DEFAULT '-1',
	`TextDecoded` varchar(160) NOT NULL DEFAULT '',
	`ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
	`RecipientID` text NOT NULL,
	`Processed` enum('false','true') NOT NULL DEFAULT
'false',
	PRIMARY KEY (`ID`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=18 ;
```

2) Tabel fire_brigade

```
CREATE TABLE IF NOT EXISTS `fire_brigade` (
	`nik` int(4) NOT NULL,
	`nama` varchar(100) NOT NULL,
```

```
       `dept` varchar(100) NOT NULL,
       `hp` varchar(12) NOT NULL,
       `aktif` tinyint(1) NOT NULL,
       PRIMARY KEY (`nik`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

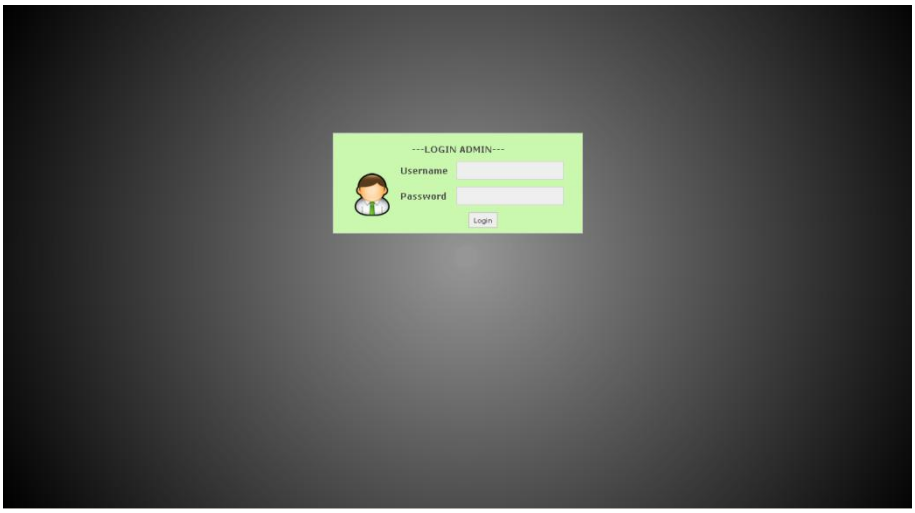
3) Tabel sentitems

```
     CREATE TABLE IF NOT EXISTS `sentitems` (
       `UpdatedInDB`     timestamp     NOT     NULL     DEFAULT
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
       `InsertIntoDB` timestamp NOT NULL DEFAULT '0000-00-00
00:00:00',
       `SendingDateTime` timestamp NOT NULL DEFAULT '0000-00-
00 00:00:00',
       `DeliveryDateTime` timestamp NULL DEFAULT NULL,
       `Text` text NOT NULL,
       `DestinationNumber` varchar(20) NOT NULL DEFAULT '',
       `Coding`
enum('Default_No_Compression','Unicode_No_Compression','8bit',
'Default_Compression','Unicode_Compression') NOT NULL DEFAULT
'Default_No_Compression',
       `UDH` text NOT NULL,
       `SMSCNumber` varchar(20) NOT NULL DEFAULT '',
       `Class` int(11) NOT NULL DEFAULT '-1',
       `TextDecoded` varchar(160) NOT NULL DEFAULT '',
       `ID` int(10) unsigned NOT NULL DEFAULT '0',
       `SenderID` varchar(255) NOT NULL,
       `SequencePosition` int(11) NOT NULL DEFAULT '1',
       `Status`
enum('SendingOK','SendingOKNoReport','SendingError','DeliveryO
K','DeliveryFailed','DeliveryPending','DeliveryUnknown','Error
') NOT NULL DEFAULT 'SendingOK',
       `StatusError` int(11) NOT NULL DEFAULT '-1',
       `TPMR` int(11) NOT NULL DEFAULT '-1',
       `RelativeValidity` int(11) NOT NULL DEFAULT '-1',
       `CreatorID` text NOT NULL,
       PRIMARY KEY (`ID`,`SequencePosition`),
       KEY `sentitems_date` (`DeliveryDateTime`),
```

```
        KEY `sentitems_tpmr` (`TPMR`),
        KEY `sentitems_dest` (`DestinationNumber`),
        KEY `sentitems_sender` (`SenderID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

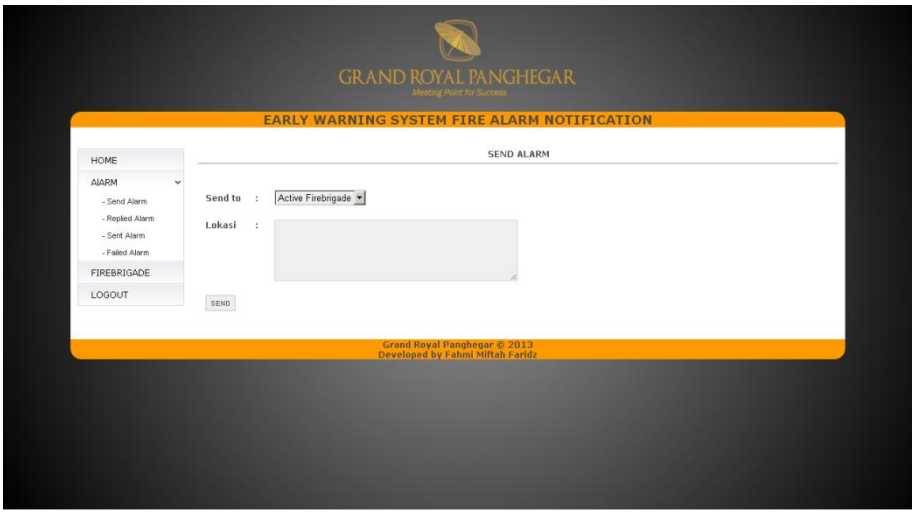
### 3.4.5. Implementasi Antarmuka Form

#### 1) Implementasi antarmuka form login




**Gambar 3.50 Implementasi Antarmuka Form Login**

#### 2) Implementasi antarmuka form send alarm



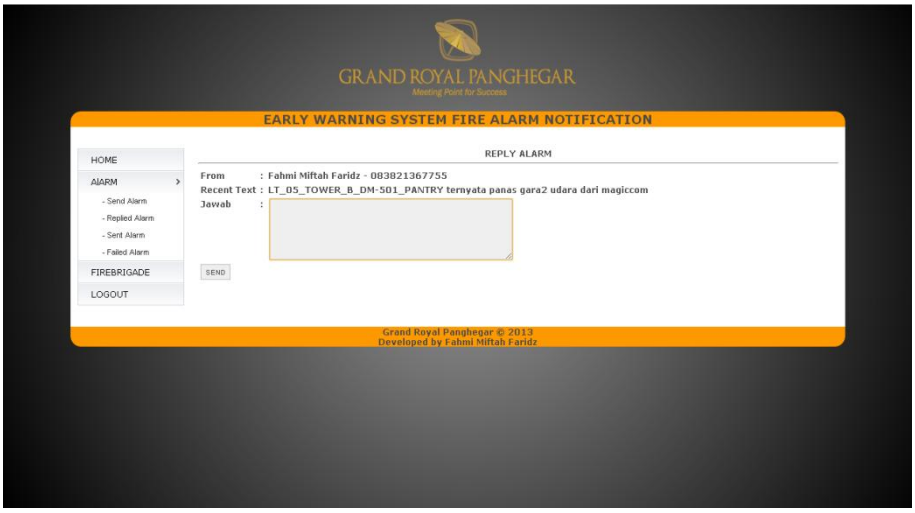

**Gambar 3.51 Implementasi Antarmuka Form Send Alarm**

### 3) Implementasi antarmuka form replied alarm




**Gambar 3.52 Implementasi Antarmuka Form Replied Alarm**

### 4) Implementasi antarmuka form jawab balasan alarm




**Gambar 3.53 Implementasi Antarmuka Form Jawab Balasan Alarm**

## 5) Implementasi antarmuka form sent alarm




**Gambar 3.54 Implementasi Antarmuka Form Sent Alarm**

## 6) Implementasi antarmuka form failed alarm



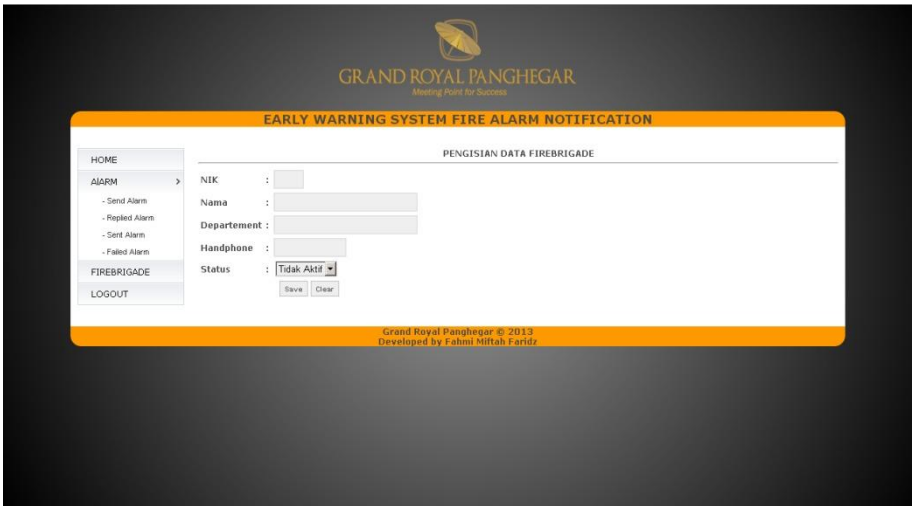

**Gambar 3.55 Implementasi Antarmuka Form Failed Alarm**

### 7) Implementasi antarmuka form pengolahan data firebrigade



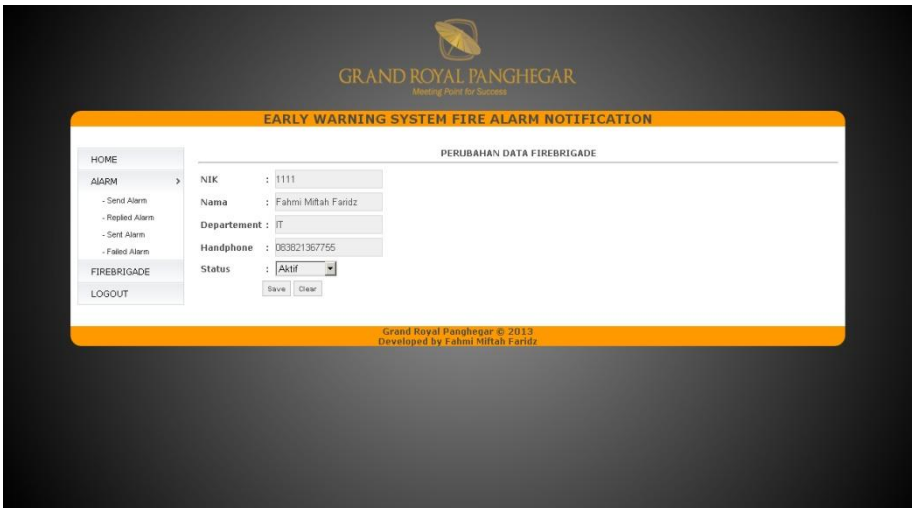

**Gambar 3.56 Implementasi Antarmuka Form Pengolahan Data Firebrigade**

### 8) Implementasi antarmuka form tambah firebrigade



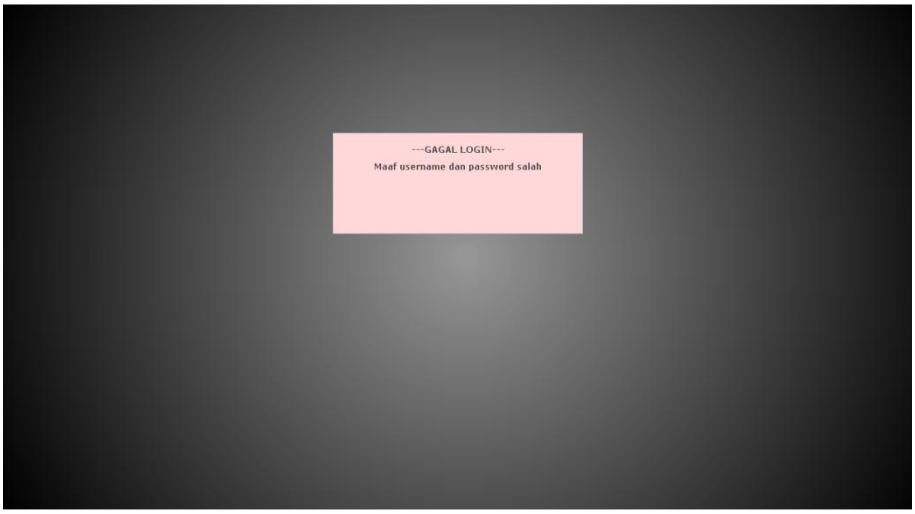

**Gambar 3.57 Implementasi Antarmuka Form Tambah Firebrigade**

### 9) Implementasi antarmuka form edit firebrigade

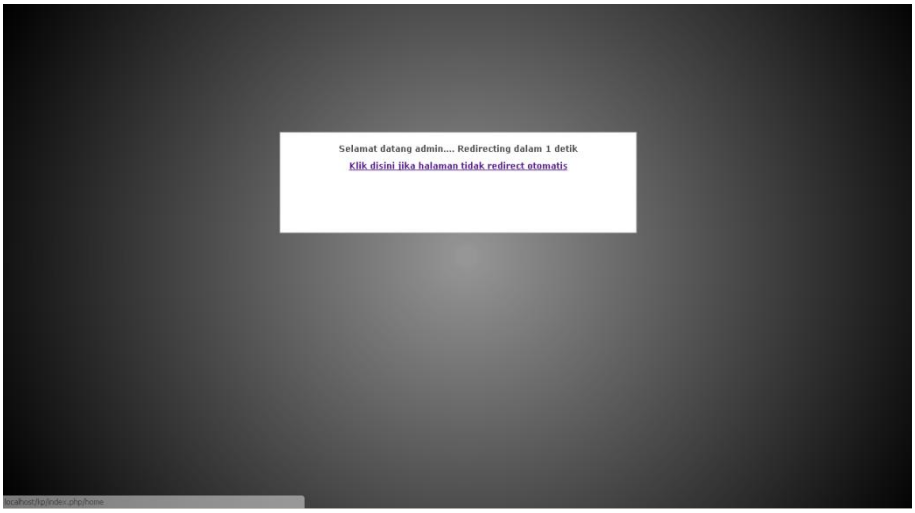



**Gambar 3.58 Implementasi Antarmuka Form Edit Firebrigade**

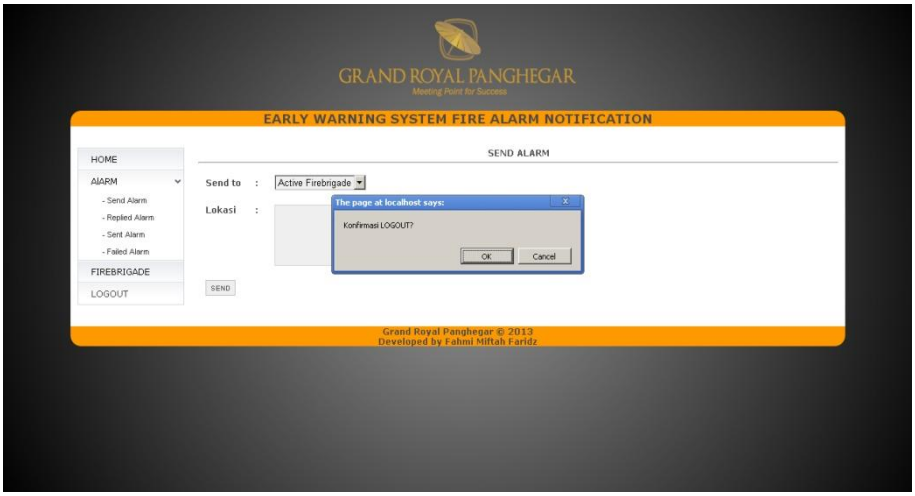
## 3.4.6. Implementasi Antarmuka Pesan



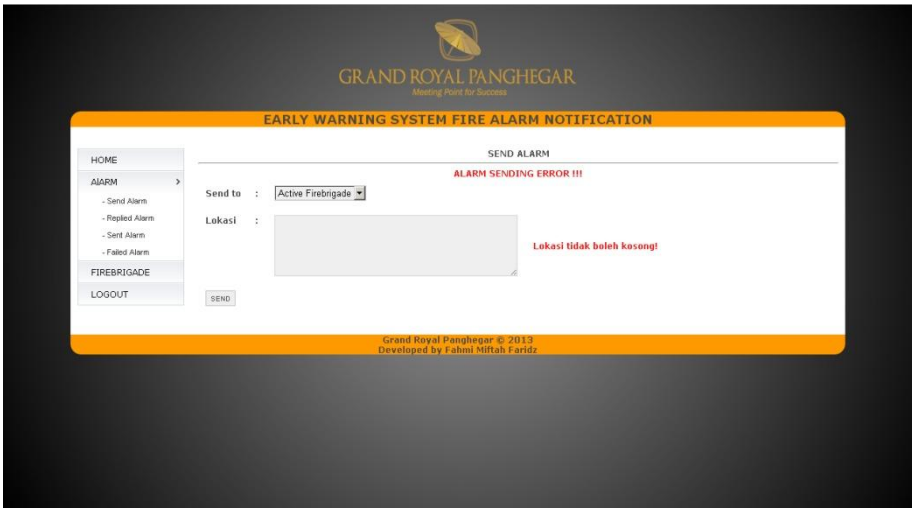

**Gambar 3.59 Implementasi Antarmuka Pesan P01**



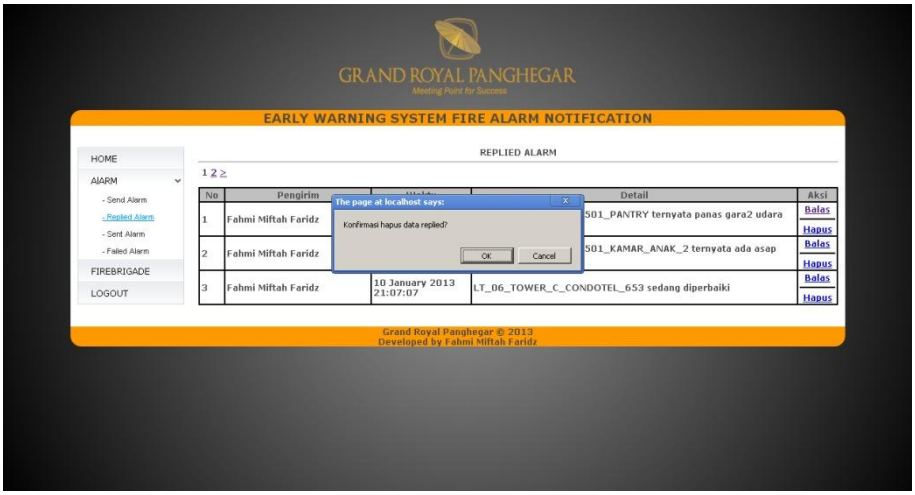

**Gambar 3.60 Implementasi Antarmuka Pesan P02**



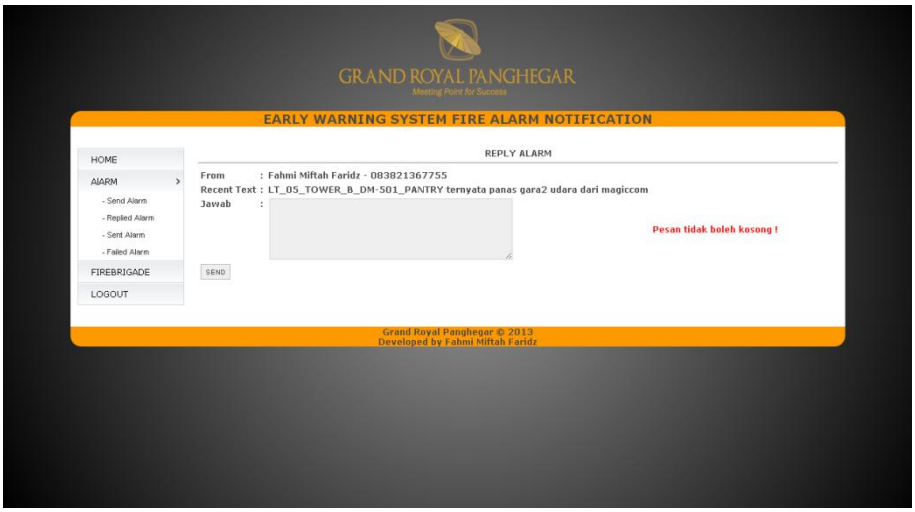

**Gambar 3.61 Implementasi Antarmuka Pesan P03**




**Gambar 3.62 Implementasi Antarmuka Pesan P04**



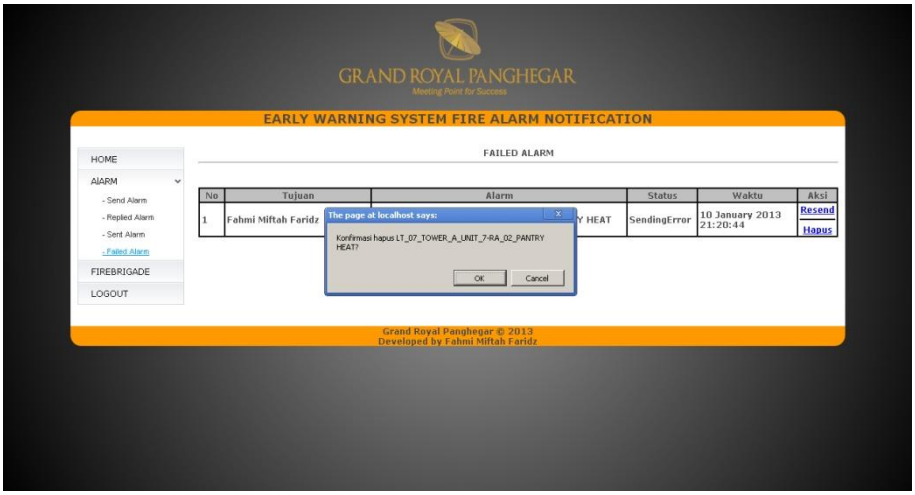

**Gambar 3.63 Implementasi Antarmuka Pesan P05**



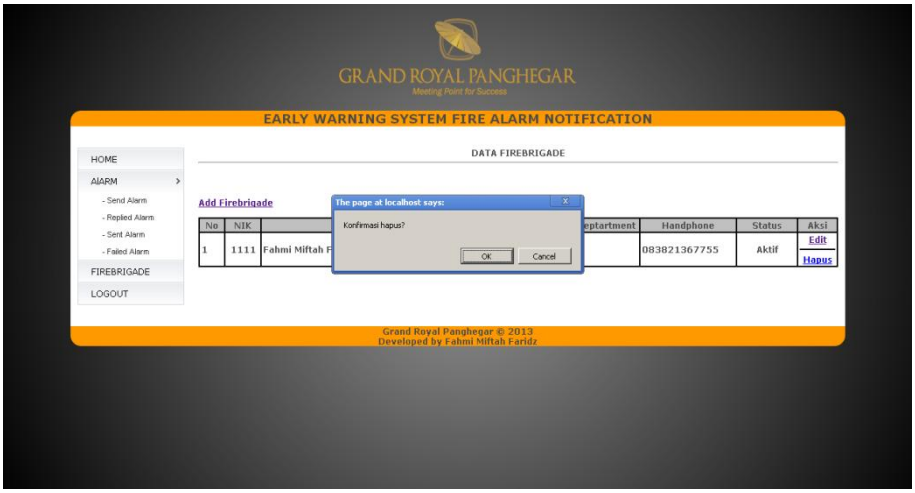

**Gambar 3.64 Implementasi Antarmuka Pesan P06**



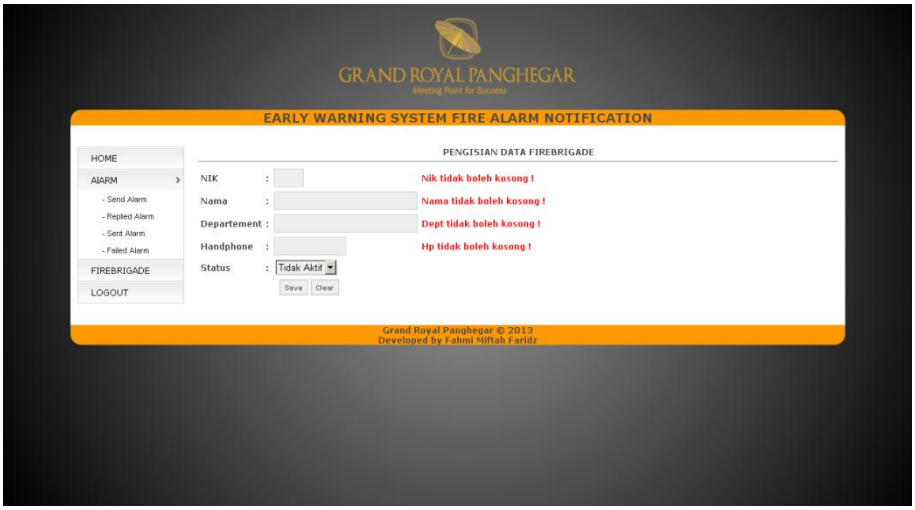

**Gambar 3.65 Implementasi Antarmuka Pesan P07**



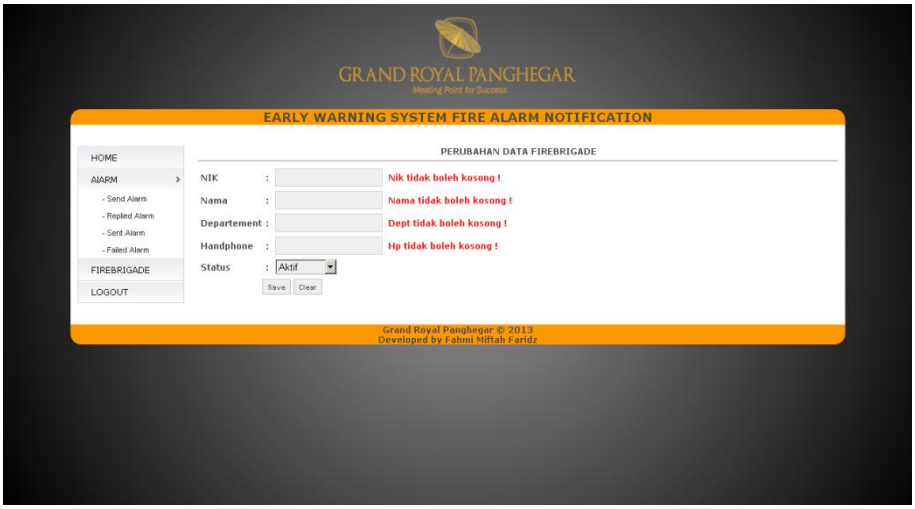

**Gambar 3.66 Implementasi Antarmuka Pesan P08**




**Gambar 3.67 Implementasi Antarmuka Pesan P09**




**Gambar 3.68 Implementasi Antarmuka Pesan P10**




**Gambar 3.69 Implementasi Antarmuka Pesan P11**

## 3.5. Pengujian

Dalam pengujian perangkat lunak ada elemen kritis dari jaminan kualitas perangkat lunak yang mempresentasikan kajian pokok dari spesifikasi, desain dan pengkodean. Pengujian yang digunakan untuk menguji sistem ini menggunakan metode pengujian *black-box*, dimana pengujian ini berfokus pada persyaratan fungsional perangkat lunak.

### 3.5.1. Pengujian Black Box

Proses pengujian *black box* adalah pengujian yang dilakukan dengan cara menguji aplikasi dengan memasukkan data ke dalam form-form yang telah disediakan. Pada tahap ini merupakan kelanjutan dari tahap implementasi. Pengujian yang akan dilakukan dengan cara *alpha* yaitu dengan metode pengujian *black box* yang berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini memungkinkan perekayasa sistem mendapat serangkaian kondisi input yang sepenuhnya semua persyaratan fungsional untuk suatu program.

#### 3.5.1.1. Rencana Pengujian

Rencana pengujian yang akan dilakukan dengan menguji secara *alpha* menggunakan metode *black box* dan wawancara.

**Tabel 3.16 Tabel Rencana Pengujian**

| **Pengujian Grand Royal Panghegar Hotel Early Warning System Fire Alarm Notification** | | |
|---|---|---|
| **Item Pengujian** | **Detail Pengujian** | **Jenis Pengujian** |
| Login Admin | Username | Black box |
| | Password | Black box |
| Send Alarm | Lokasi | Black box |
| Reply Alarm | Pesan jawab | Black box |
| Add Firebrigade | Nik | Black box |
| | Nama | Black box |
| | Departemen | Black box |
| | Handphone | Black box |
| | Status | Black box |
| Edit Firebrigade | Nik | Black box |
| | Nama | Black box |
| | Departemen | Black box |
| | Handphone | Black box |
| | Status | Black box |

### 3.5.1.2. Kasus dan Hasil Pengujian Alpha

Proses pengujian *alpha* dulakukan untuk mengetahui apakah data yang dimasukkan sudah sesuai dengan yang diharapkan. Berdasarkan rencana pengujian, maka dapat dilakukan pengujian alpha pada sistem ini.

**1. Form Login**

**Tabel 3.17 Tabel Pengujian Login Admin**

| **Kasus Data dan hasil Uji (Data Normal)** | |
|---|---|
| **Data Masukan** | Username : admin, Password:admin |
| **Yang Diharapkan** | Data yang dimasukkan benar lalu menampilkan pesan selamat datang, kemudian redirect ke home |
| **Pengamatan** | Menampilkan pesan selamat datang dan redirect ke halaman home |
| **Kesimpulan** | diterima |
| **Kasus Data dan hasil Uji (Data Salah)** | |
| **Data Masukan** | Username: user, Password:admin |
| **Yang Diharapkan** | Muncul pesan kesalahan gagal login |
| **Pengamatan** | Muncul pesalahan gagal login |
| **Kesimpulan** | Diterima |

## 2. Form Send Alarm

**Tabel 3.18 Tabel Pengujian Send Alarm**

| **Kasus Data dan hasil Uji (Data Normal)** | |
|---|---|
| **Data Masukan** | Lokasi kejadian alarm |
| **Yang Diharapkan** | Menampilkan pesan ALARM SENT |
| **Pengamatan** | Menampilkan pesan ALARM SENT |
| **Kesimpulan** | Diterima |
| **Kasus Data dan hasil Uji (Data Salah)** | |
| **Data Masukan** | Tidak memasukan lokasi alarm |
| **Yang Diharapkan** | Menampilkan pesan ALARM SENDING ERROR |
| **Pengamatan** | Menampilkan pesan ALARM SENDING ERROR |
| **Kesimpulan** | Diterima |

## 3. Form Reply Alarm

**Tabel 3.19Tabel Pengujian Reply Alarm**

| **Kasus Data dan hasil Uji (Data Normal)** | |
|---|---|
| **Data Masukan** | Jawaban pesan |
| **Yang Diharapkan** | Mengirim jawaban kemudian menampilkan halaman replied alarm |
| **Pengamatan** | Menampilkan halaman replied alarm |
| **Kesimpulan** | Diterima |
| **Kasus Data dan hasil Uji (Data Salah)** | |
| **Data Masukan** | Tidak memasukan pesan |
| **Yang Diharapkan** | Menampilkan pesan kesalahan “pesan tidak boleh kosong” |
| **Pengamatan** | Menampilkan “pesan tidak boleh kosong” |
| **Kesimpulan** | Diterima |

## 4. Form Add Firebrigade

**Tabel 3.20 Tabel Pengujian Add Firebrigade**

| **Kasus Data dan hasil Uji (Data Normal)** | |
|---|---|
| **Data Masukan** | Nik:1111, nama:fahmi miftah faridz, department:IT, handphone:083821367755, status:aktif |
| **Yang Diharapkan** | Menyimpan data firebrigade baru kemudian menampilkan halaman data firebrigade |
| **Pengamatan** | menampilkan halaman data firebrigade |
| **Kesimpulan** | Diterima |
| **Kasus Data dan hasil Uji (Data Salah)** | |
| **Data Masukan** | Nik, nama, departemen, handphone kosong. Nik:1111, handphone:083821367755 |
| **Yang Diharapkan** | Menampilkan pesan kesalahan tidak boleh kosong dan tidak boleh sama |
| **Pengamatan** | Menampilkan pesan kesalahan tidak boleh kosong dan |

| | |
|---|---|
| | tidak boleh sama |
| **Kesimpulan** | Diterima |

**5. Form Edit Firebrigade**

**Tabel 3.21 Tabel Pengujian Edit Firebrigade**

| **Kasus Data dan hasil Uji (Data Normal)** | |
|---|---|
| **Data Masukan** | Nik:1111, nama:fahmi miftah faridz, department:IT, handphone:083821367755, status:tidak aktif |
| **Yang Diharapkan** | Menyimpan perubahan data firebrigade baru kemudian menampilkan halaman data firebrigade |
| **Pengamatan** | menampilkan halaman data firebrigade |
| **Kesimpulan** | Diterima |
| **Kasus Data dan hasil Uji (Data Salah)** | |
| **Data Masukan** | Nik, nama, departemen, handphone kosong. |
| **Yang Diharapkan** | Menampilkan pesan kesalahan tidak boleh kosong. |
| **Pengamatan** | Menampilkan pesan kesalahan tidak boleh kosong |
| **Kesimpulan** | Diterima |

### 3.5.1.3. Kesimpulan Hasil Pengujian Alpha

Berdasarkan hasil pengujian *alpha* yang telah dilakukan dapat ditarik kesimpulan bahwa aplikasi sudah berjalan cukup maksimal, tetapi tidak menutup kemungkinan dapat terjadi kesalahan pada suatu saat aplikasi digunakan. sehingga membutuhkan proses *maintenance* untuk lebih mengetahui kekurangan dari aplikasi.

### 3.5.1.4. Kasus dan Hasil Pengujian Betha

Pengujian *betha* merupakan pengujian yang dilakukan secara objektif dimana aplikasi diuji secara langsung ke lapangan, yaitu dengan mewawancarai operator yang menggunakan sistem ini.

Hasil pengujian *betha* adalah sebagai berikut:

1. Apakah aplikasi ini mempermudah komunikasi antara operator dengan firebrigade ? **ya**
2. Apakah respon aplikasi ini cepat ? **ya**
3. Apakah dengan adanya aplikasi ini anggaran biaya untuk pendanaan handy talky berkurang ? **ya**
4. Apakah aplikasi ini mempermudah pencarian data firebrigade ? **tidak**

5. Apakah tampilan aplikasi ini mudah dipahami ? **ya**
6. Apakah masih ada yang perlu ditambahkan pada aplikasi ini ? **ya**

#### 3.5.1.5. Kesimpulan Hasil Pengujian Betha

Berdasarkan hasil wawancara diatas didapatkan dari pengujian *betha,* yang mewawancarai kepada operator yang menggunakan aplikasi ini. Bahwa aplikasi mempermudah komunikasi dengan firebrigade, namun belum membantu pencarian data firebrigade.