BAB III

PEMBAHASAN

3.1 Jadwal Kerja Praktek

Penulis melakukan kerja praktek di PT. Hariff Daya Tunggal Engineering yang berlokasi di Jl. Soekarno Hatta No. 450 Bandung. Kerja praktek ini dilaksanakan selama 1 (satu) bulan, yaitu mulai tanggal 16 September 2013 sampai dengan 14 Oktober 2013.

Tabel 3.1 Jadwal Kerja Praktek

No	Waktu & Kegiatan	Sept 2013					Sept 2013					Okt 2013					Okt 2013				
		Minggu III					Minggu IV					Minggu I					Minggu II				
		S	S	R	K	J	S	S	R	K	J	S	S	R	K	J	S	S	R	K	J
1	Pengumpulan Data																				
2	Analisis																				
3	Desain																				
4	Pengembangan																				
5	Pengujian																				

3.2 Data Kerja Praktek

Kegiatan kerja praktek yang dilakukan di PT. Hariff Daya Tunggal Engineering dilaksanakan selama satu bulan. Adapun jadwal masuk kerja yang harus dipenuhi dimulai dari hari Senin sampai Jumat pukul 08:00 sampai 17:00. Tugas yang diberikan oleh PT. Hariff adalah membangun Aplikasi Alarm Handler untuk keperluan Trouble Ticket Perangkat Hariff Power System.

3.3 Analisis

Analisis dilakukan dengan tujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem serta menentukan kebutuhan dari sistem yang akan dibangun. Analisis tersebut meliputi analisis masalah, analisis arsitektur sistem, analisis komponen sistem, analisis kebutuhan non fungsional dan fungsional.

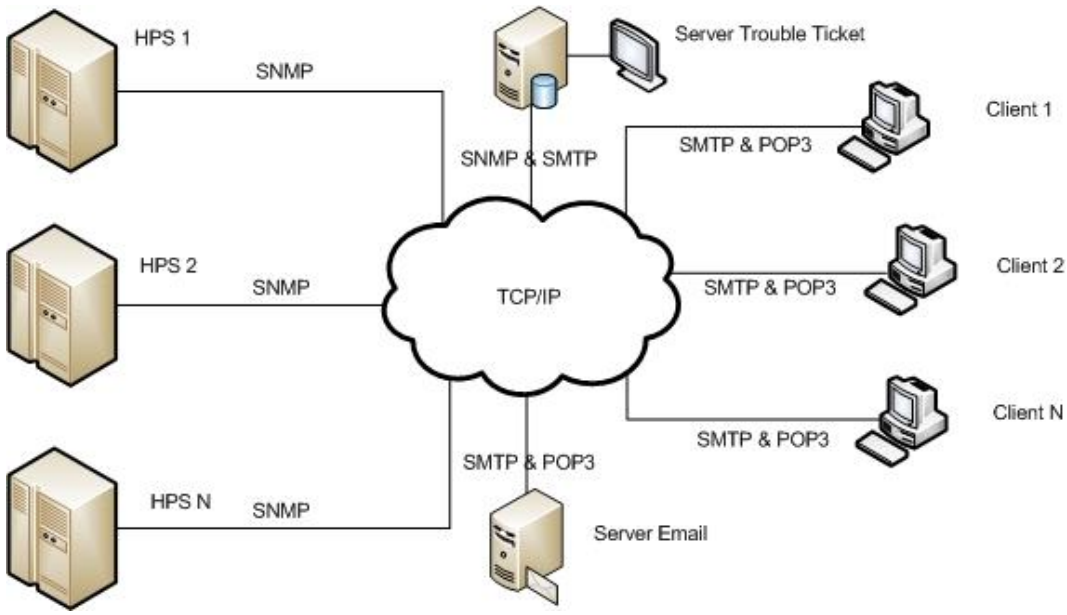
3.3.1 Analisis Masalah

Identifikasi masalah adalah langkah awal dari analisis sistem. Langkah ini diperlukan untuk mengetahui permasalahan apa saja yang ada pada sistem. Oleh karena itu langkah pertama adalah menganalisis masalah yang muncul. Identifikasi masalah yang dimaksud di sini adalah bagaimana cara membangun sistem monitoring gangguan perangkat HPS secara efektif dan efisien dengan tanpa harus mendatangi ke lokasi secara langsung dan setiap gangguan yang terjadi dapat diketahui oleh teknisi lokal secara realtime, sehingga para teknisi lokal dapat segera melakukan tindakan perbaikan atau pemeliharaan. Hal tersebut dapat memberikan kualitas pelayanan kepada pelanggan menjadi lebih baik. Selain itu, bagaimana informasi gangguan yang terjadi di lapangan dapat ditampilkan dalam sebuah layar aplikasi dan dapat disimpan dalam suatu server basis data sebagai laporan untuk melakukan perbaikan dan pemeliharaan secara berkala.

3.3.2 Analisis Arsitektur Sistem

Dalam tahap analisis sistem akan ditentukan *requirement* (kebutuhan-kebutuhan) dari sistem yang akan dibangun, mengidentifikasi dan mengevaluasi

permasalahan-permasalahan yang terjadi untuk membangun aplikasi alarm handler untuk keperluan trouble ticket power sistem. Adapun arsitektur sistem yang akan dibangun dapat dilihat pada gambar di bawah ini :



Gambar 3.1 Gambaran Sistem Yang Akan Dibangun

3.3.3 Analisis Komponen Sistem

Sesuai dengan gambar 3.1 di atas, komponen sistem yang dibangun adalah sebagai berikut :

1. Perangkat HPS

   Merupakan perangkat yang dimonitor oleh server aplikasi trouble ticket. Perangkat ini dipasang di lokasi shelter transmisi komunikasi milik pelanggan PT. Hariff Daya Tunggal Engineering yang tersebar di berbagai wilayah di seluruh Indonesia.

2. Server Aplikasi Trouble Ticket

   Aplikasi yang berfungsi melakukan monitoring perangkat HPS yang dipasang di lokasi. Aplikasi inilah yang memberikan pesan notifikasi

email kepada para teknisi lokal jika suatu saat terjadi gangguan pada perangkat HPS. Aplikasi ini dipasang di kantor pusat PT. Hariff Daya Tunggal Engineering yang berlokasi di Bandung yang dilengkapi dengan aplikasi server basis data untuk menyimpan dan mengolah data.

3. Server Email

   Merupakan server yang digunakan oleh aplikasi trouble ticket dalam melakukan pengiriman pesan notifikasi email kepada para teknisi lokal sebagai informasi peringatan dini atas kejadian yang dialami oleh perangkat HPS yang berada di lokasi.

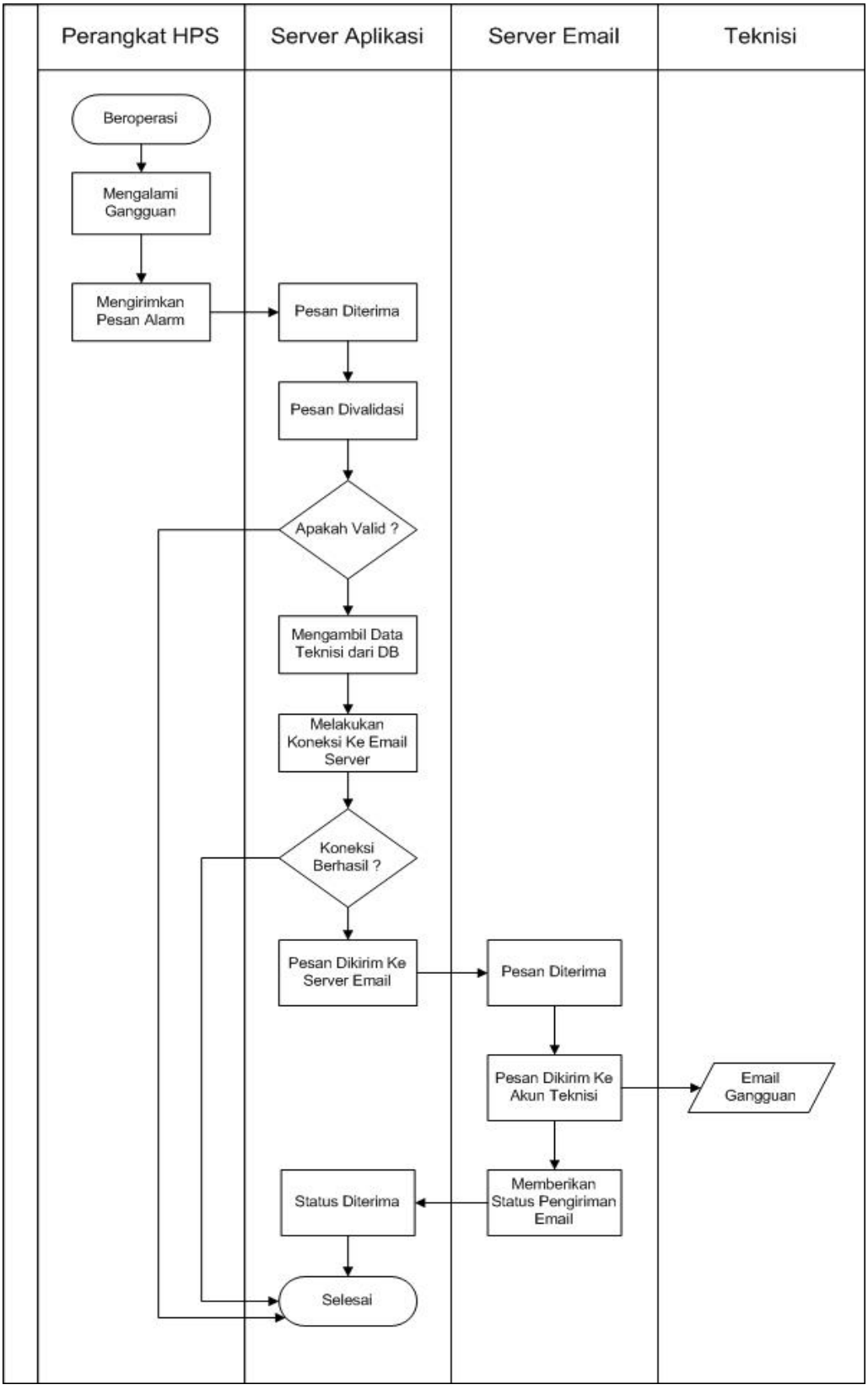
4. Client / Workstation

   Merupakan aplikasi yang digunakan oleh para teknisi untuk membaca email-email notifikasi yang dikirim oleh server trouble ticket

3.3.4 Analisis Cara Kerja Sistem

Jika suatu ketika perangkat HPS mengalami gangguan, perangkat tersebut mengirimkan pesan gangguan ke server aplikasi menggunakan protokol SNMP. Pesan SNMP tersebut oleh server aplikasi divalidasi. Pesan gangguan tersebut terdiri dari nama gangguan dan lokasi kejadian. Setelah melakukan validasi pesan, aplikasi mengambil data teknisi sesuai dengan lokasi perangkat HPS yang mengalami gangguan dari server basis data, kemudian melakukan pengiriman pesan email ke para teknisi melalui server email menggunakan protokol SMTP.

Untuk lebih memahami cara kerja sistem, dapat di dilihat pada gambar 3.2 di bawah ini.



Gambar 3.2 Cara Kerja Sistem

3.3.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan melibatkan analisis perangkat keras (hardware), analisis perangkat lunak (software) dan analisis pengguna (user).

3.3.5.1 Analisis Perangkat Keras (Hardware)

Analisis kebutuhan non fungsional dari segi perangkat keras yaitu perangkat keras yang digunakan dalam pembangunan sistem aplikasi, yang terdiri dari dua bagian, yaitu :

1. Analisis Perangkat Keras Server Aplikasi

   Berikut adalah daftar spesifikasi perangkat keras yang digunakan oleh server aplikasi.

Tabel 3.2 Spesifikasi Perangkat Keras Server Aplikasi

No	Perangkat Keras	Spesifikasi
1	Processor	Core i5 2.4 GHz
2	Memory	4GB DDR3
3	HDD	500GB
4	VGA	64MB
5	Monitor	LCD 17” resolusi 1366x768 pixel
6	LAN Card	100/1000 MB Fast Ethernet
7	Keyboard+Mouse	Standard USB

2. Analisis Perangkat Keras Client/Workstation

   Berikut adalah daftar spesifikasi perangkat keras yang digunakan oleh komputer client/workstation.

Tabel 3.3 Spesifikasi Perangkat Keras Client Aplikasi

No	Perangkat Keras	Spesifikasi
1	Processor	Core 2 Duo 2.0 GHz
2	Memory	2GB DDR2
3	HDD	250GB
4	VGA	32MB
5	Monitor	LCD 17” resolusi 1366x768 pixel
6	LAN Card	100/1000 MB Fast Ethernet
7	Keyboard+Mouse	Standard USB

3.3.5.2 Analisis Perangkat Lunak (Software)

Analisis kebutuhan non fungsional dari segi perangkat lunak yaitu perangkat lunak yang digunakan dalam pembangunan sistem aplikasi, yang terdiri dari dua bagian, yaitu :

1. Analisis Perangkat Lunak Server Aplikasi

   Berikut adalah daftar spesifikasi perangkat lunak yang digunakan oleh server aplikasi.

Tabel 3.4 Spesifikasi Perangkat Lunak Server Aplikasi

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 7 Professional 32 bit
2	Java	JRE 6.0 update 45 Win32
3	Database	MySQL Server 5.1.33
4	SNMP Library	Adventnet 4.0.7
5	Email Library	JavaMail 1.4.2

2. Analisis Perangkat Lunak Client/Workstation

   Berikut adalah daftar spesifikasi perangkat lunak yang digunakan oleh client/workstation.

Tabel 3.5 Spesifikasi Perangkat Lunak Client/Workstation

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 7 Professional 32 bit
2	Email Client	Microsoft Outlook 2007
3	Web Browser (optional)	Mozilla Firefox versi 23

3.3.5.3 Analisis Pengguna

Aplikasi yang penulis kembangkan memiliki dua jenis pengguna, yaitu :

1. Teknisi

   Tipe pengguna yang hanya akan menerima pesan notifikasi email jika suatu ketika perangkat HPS yang berada di wilayah kerjanya mengalami gangguan. Tipe pengguna ini berada di lokasi/wilayah kerja yang tersebar di beberapa daerah di Indonesia. Email notifikasi dapat dibaca dengan menggunakan aplikasi email client, baik yang berbasis web maupun berbasis aplikasi desktop, baik menggunakan komputer, laptop atau menggunakan perangkat mobile.

2. Koordinator Teknisi

   Tipe pengguna yang dapat menjalankan dan mematikan aplikasi trouble ticket, mengelola data-data yang digunakan oleh aplikasi seperti data lokasi, perangkat HPS, gangguan, tingkat kefatalan suatu gangguan, dan mengelola data teknisi. Tipe pengguna ini berada di kantor pusat, tempat dimana aplikasi ini dipasang. Jika ada informasi gangguan yang diterima di kantor pusat, pengguna inilah yang langsung menghubungi teknisi setempat untuk melakukan pemeliharaan atau perbaikan.

3.3.6 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses yang akan diterapkan dalam sistem yang akan dibangun sesuai dengan kebutuhan bisnis proses yang ada di lingkungan pengguna sistem, yang dapat dimodelkan baik dengan cara terstruktur menggunakan *Data Flow Diagram* (DFD) atau dengan cara berorientasi objek menggunakan *Unified Modeling Language* (UML).

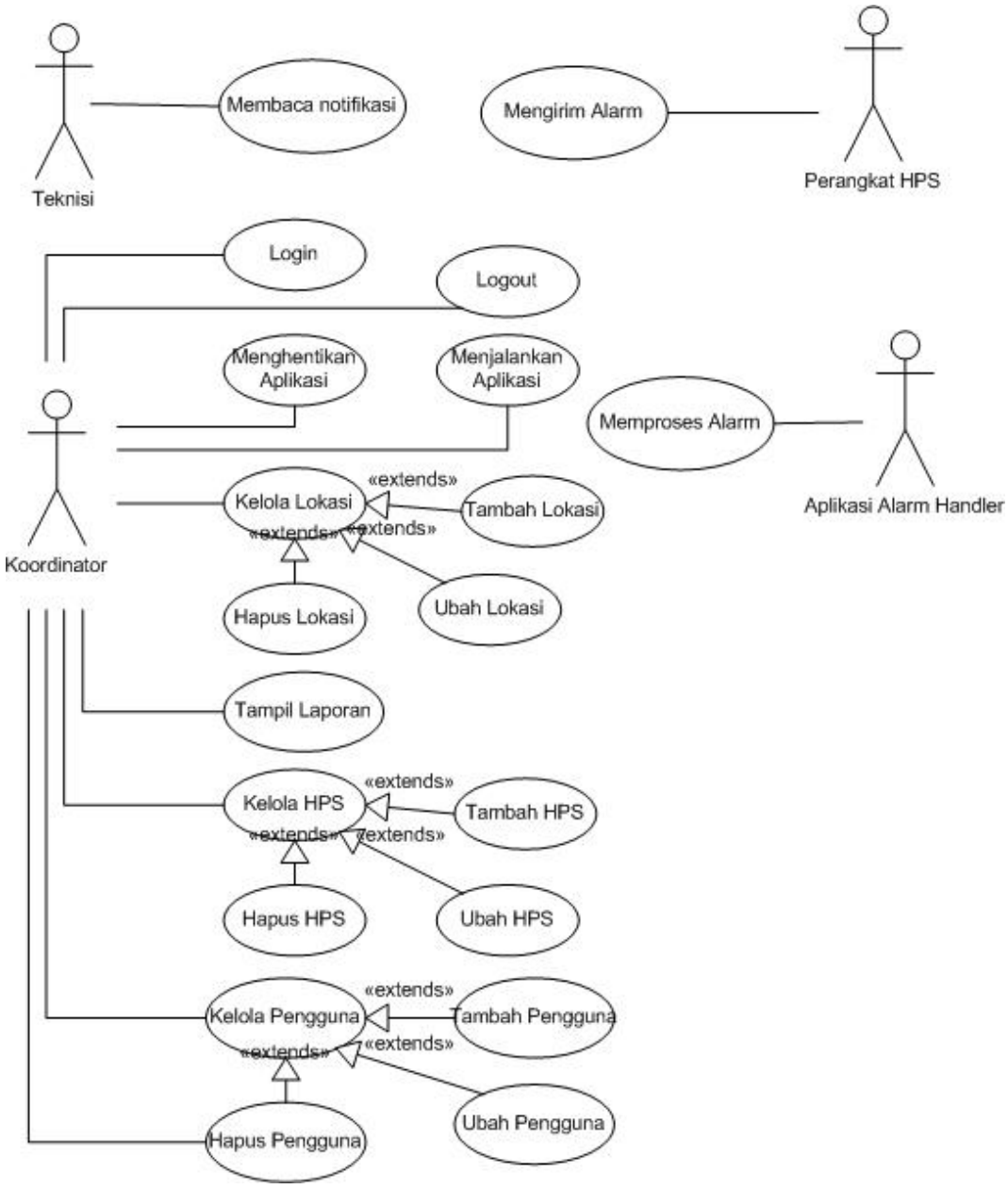
Analisis kebutuhan fungsional pada penulisan laporan kerja praktek ini dimodelkan dengan menggunakan UML. Tahap-tahap pemodelan dalam analisis menggunakan UML antara lain *Usecase Diagram*, *Usecase Scenario, Activity Diagram*, *Sequence Diagram* dan *Class Diagram*.

3.3.6.1 Use Case Diagram

Use Case Diagram merupakan suatu diagram yang menggambarkan tentang fungsionalitas dari sistem yang akan dibangun yang terdiri dari fungsi-fungsi atau prosedur-prosedur, sehingga dengan melihat diagram tersebut kita akan mengetahui kemampuan fungsi yang dimiliki oleh suatu sistem yang sedang dikembangkan.

Analisis aplikasi alarm handler untuk keperluan trouble ticket perangkat HPS terdiri dari prosedur menjalankan aplikasi alarm handler, prosedur menghentikan aplikasi alarm handler, prosedur memproses pesan alarm berikut pengiriman pesan notifikasi email, prosedur login, prosedur logout, prosesur kelola data lokasi, prosedur kelola data perangkat HPS dan prosedur kelola pengguna dan prosedur menampilkan laporan gangguan.

Untuk memperjelas prosedur yang terdapat di dalam sistem yang dikembangkan, dapat di lihat pada diagram *use case* di bawah ini.



Gambar 3.3 *Use Case Diagram* Aplikasi Alarm Handler

Sesuai dengan gambar 3.3 diatas, terdapat empat aktor yang terdapat di dalam sistem, yaitu :

1. Perangkat HPS

   Aktor ini dapat mengirimkan pesan gangguan ke aplikasi alarm handler jika terdapat gangguan di lokasi.

2. Aplikasi Alarm Handler

   Aktor yang bertugas memproses pesan alarm yang dikirim oleh perangkat HPS.

3. Teknisi

   Aktor yang akan diberikan pesan notifikasi email oleh apalikasi alarm handler atas gangguan yang dialami oleh perangkat HPS.

4. Koordinator

   Aktor yang melakukan monitoring perangkat HPS menggunakan aplikasi di kantor pusat, dapat melakukan pengelolaan data teknisi dan data-data yang berhubungan dengan perangkat HPS. Dengan kata lain, koordinator merupakan operator aplikasi alarm handler.

3.3.6.2 *Use Case* Skenario

Use Case Scenario merupakan deskripsi urutan langkah-langkah dalam proses bisnis baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. *Use Case Scenario* yang akan dibahas pada penulisan laporan kerja praktek ini adalah untuk aktor aplikasi alarm handler dan aktor koordinator. Berdasarkan diagram *Use Case* Aplikasi Alarm Handler pada gambar 3.3, maka *Use Case Scenario* adalah sebagai berikut :

1. *Use Case Scenario* Menjalankan Aplikasi Alarm Handler

   Skenario ini menjelaskan tentang cara menjalankan aplikasi alarm handler, mulai dari aplikasi dalam keadaan mati (*stopped*), sampai aplikasi tersebut bekerja dengan normal (*running*).

Tabel 3.6 *Use Case Scenario* Menjalankan Aplikasi Alarm Handler

IDENTIFIKASI	
Nomor	UCS-01
Nama	*Use Case* Menjalankan Aplikasi Alarm Handler
Deskripsi	Menjelaskan tentang cara menjalankan alarm handler
Aktor	Koordinator
Kondisi Awal	Aktor telah membuka aplikasi alarm handler
SKENARIO	
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol "start"	
	2. Memproses *request* dari aktor
	3. Alarm handler aktif
Kondisi Akhir : Status alarm handler dalam kondisi "started"	

2. *Use Case Scenario* Menghentikan Aplikasi Alarm Handler

   Skenario ini menjelaskan tentang cara mematikan aplikasi alarm handler, mulai dari aplikasi alarm handler dalam kondisi hidup hingga kondisi mati.

Tabel 3.7 *Use Case Scenario* Menghentikan Aplikasi Alarm Handler

IDENTIFIKASI	
Nomor	UCS-02
Nama	*Use Case* Mematikan Alarm Handler
Deskripsi	Menjelaskan tentang cara mematikan alarm handler
Aktor	Koordinator
Kondisi Awal	Aktor telah menjalankan aplikasi alarm handler
SKENARIO	
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol "stop"	

	2. Memproses *request* dari aktor
	3. Alarm handler berhenti
Kondisi Akhir : Status alarm handler dalam kondisi "stopped"	

3. *Use Case Scenario* Login

Login diperlukan untuk melakukan pengelolaan data seperti penambahan, pengubahan dan penghapusan data yang digunakan di dalam aplikasi. Skenario di bawah ini menjelaskan tentang cara memproses login yang dilakukan oleh koordinator.

Tabel 3.8 *Use Case Scenario* Login

IDENTIFIKASI	
Nomor	UCS-03
Nama	*Use Case* Login
Deskripsi	Menjelaskan tentang cara memproses login koordinator pada aplikasi alarm handler
Aktor	Koordinator
Kondisi Awal	Halaman login sudah ditampilkan
SKENARIO UTAMA	
Aksi Aktor	**Reaksi Sistem**
1. Memasukan username dan password	
2. Menekan tombol "login"	
	3. Memvalidasi username dan password
	4. Menampilkan pesan "login berhasil"
SKENARIO ALTERNATIF – LOGIN TIDAK *VALID*	
Aksi Aktor	**Reaksi Sistem**
1. Memasukan username dan password	
2. Menekan tombol "login"	
	3. Memvalidasi username dan password
	4. Menampilkan pesan "login gagal"
5. Memasukan username dan password yang benar	
6. Menekan tombol "login"	

	7. Memvalidasi username dan password
	8. Menampilkan pesan "login berhasil"
Kondisi Akhir : Mengaktifkan menu pengelolaan data	

4. *Use Case Scenario* Logout

Skenario di bawah ini menjelaskan tentang cara melakukan logout atau keluar dari aplikasi alarm handler.

Tabel 3.9 *Use Case Scenario* Logout

IDENTIFIKASI	
Nomor	UCS-04
Nama	*Use Case* Logout
Deskripsi	Menjelaskan tentang cara logout dari aplikasi
Aktor	Koordinator
Kondisi Awal	Aktor telah melakukan login
SKENARIO	
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol "logout"	
	2. Menampilkan pesan konfirmasi "apakah akan ke luar ?" serta tombol "ok" dan "cancel"
3. Menekan tombol "ok"	
	4. Memproses *request* dari aktor
Kondisi Akhir : Menu pengelolaan data tidak diaktifkan	

5. *Use Case Scenario* Tambah Lokasi

Skenario di bawah ini menjelaskan tentang cara menambah data lokasi, mulai dari menampilkan daftar lokasi hingga data lokasi tersimpan kedalam database.

Tabel 3.10 Use Case Scenario Tambah Lokasi

IDENTIFIKASI	
Nomor	UCS-05
Nama	*Use Case* Menambah Lokasi

Deskripsi	Menjelaskan tentang cara menambah lokasi
Aktor	Koordinator
Kondisi Awal	Pengguna sedang menampilkan halaman daftar lokasi pada menu “Management”
SKENARIO	
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol “New”	
	2. Menampilkan kotak isian/form lokasi
3. Memasukan nama lokasi	
4. Menekan tombol “Save”	
	5. Memproses request dari aktor
Kondisi Akhir : kotak isian lokasi ditutup dan data lokasi yang dimasukan ditampilkan.	

6. *Use Case Scenario* Ubah Lokasi

Skenario di bawah ini menjelaskan tentang cara mengubah data lokasi.

Tabel 3.11 *Use Case Scenario* Ubah Lokasi

IDENTIFIKASI	
Nomor	UCS-06
Nama	*Use Case* Mengubah Lokasi
Deskripsi	Menjelaskan tentang cara mengubah lokasi
Aktor	Koordinator
Kondisi Awal	Pengguna sedang menampilkan halaman daftar lokasi pada menu “Management”
SKENARIO	
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih *record* lokasi yang akan diubah	
	2. *Record* yang dipilih aktor diberi warna.
3. Menekan tombol “Edit”	
	4. Menampilkan kotak isian lokasi
5. Memasukan data lokasi yang baru	
6. Menekan tombol “Save”	
	7. Memproses *request* dari aktor
Kondisi Akhir : kotak lokasi ditutup dan data lokasi baru ditampilkan	

7. *Use Case Scenario* Hapus Lokasi

Skenario di bawah ini menjelaskan tentang cara menghapus data lokasi.

Tabel 3.12 *Use Case Scenario* Menghapus Lokasi

IDENTIFIKASI	
Nomor	UCS-07
Nama	*Use Case* Menghapus Lokasi
Deskripsi	Menjelaskan tentang cara menghapus lokasi
Aktor	Koordinator
Kondisi Awal	Pengguna sedang menampilkan halaman daftar lokasi pada menu "Management"
SKENARIO	
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih *record* lokasi yang akan diubah	
	2. *Record* yang dipilih aktor diberi warna.
3. Menekan tombol "Delete"	
	4. Menampilkan kotak dialog konfirmasi
5. Menekan tombol "Ok"	
	6. Memproses *request* dari aktor
Kondisi Akhir : kotak isian lokasi ditutup dan data lokasi yang terpilih dihapus.	

8. *Use Case Scenario* Tambah HPS

Skenario di bawah ini menjelaskan tentang cara menambah data HPS.

Tabel 3.13 *Use Case Scenario* Tambah HPS

IDENTIFIKASI	
Nomor	UCS-08
Nama	*Use Case* Menambah HPS
Deskripsi	Menjelaskan tentang cara menambah HPS
Aktor	Koordinator
Kondisi Awal	Pengguna sedang menampilkan halaman daftar HPS pada menu "Management"
SKENARIO	
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol "New"	

	2. Menampilkan kotak isian/form HPS
3. Memasukan data HPS	
4. Menekan tombol “Save”	
	5. Memproses *request* dari aktor
Kondisi Akhir : kotak isian HPS ditutup dan data HPS yang dimasukan ditampilkan.	

9. *Use Case Scenario* Ubah HPS

Skenario di bawah ini menjelaskan tentang cara melakukan perubahan data HPS.

Tabel 3.14 *Use Case Scenario* Ubah HPS

IDENTIFIKASI	
Nomor	UCS-09
Nama	*Use Case* Mengubah HPS
Deskripsi	Menjelaskan tentang cara mengubah HPS
Aktor	Koordinator
Kondisi Awal	Pengguna sedang menampilkan halaman daftar HPS pada menu “Management”
SKENARIO	
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih *record* HPS yang akan diubah	
	2. *Record* yang dipilih aktor diberi warna.
3. Menekan tombol “Edit”	
	4. Menampilkan kotak isian HPS
5. Memasukan data HPS yang baru	
6. Menekan tombol “Save”	
	7. Memproses *request* dari aktor
Kondisi Akhir : kotak isian HPS ditutup dan data HPS yang baru ditampilkan	

10. *Use Case Scenario* Hapus HPS

Skenario di bawah ini menjelaskan tentang cara melakukan penghapusan data HPS.

Tabel 3.15 *Use Case Scenario* Hapus HPS

IDENTIFIKASI	
Nomor	UCS-10
Nama	*Use Case* Menghapus Lokasi
Deskripsi	Menjelaskan tentang cara menghapus lokasi
Aktor	Koordinator
Kondisi Awal	Pengguna sedang menampilkan halaman daftar lokasi pada menu "Management"
SKENARIO	
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih *record* lokasi yang akan diubah	
	2. *Record* yang dipilih aktor diberi warna.
3. Menekan tombol "Delete"	
	4. Menampilkan kotak dialog konfirmasi
5. Menekan tombol "Ok"	
	6. Memproses *request* dari aktor
Kondisi Akhir : kotak isian lokasi ditutup dan data lokasi yang terpilih dihapus.	

11. *Use Case Scenario* Tambah Pengguna

Skenario di bawah ini menjelaskan tentang cara memasukan atau menambahkan data pengguna.

Tabel 3.16 *Use Case Scenario* Tambah Pengguna

IDENTIFIKASI	
Nomor	UCS-11
Nama	*Use Case* Menambah Pengguna
Deskripsi	Menjelaskan tentang cara menambah pengguna
Aktor	Koordinator
Kondisi Awal	Pengguna sedang menampilkan halaman daftar pengguna pada menu "Management"

SKENARIO	
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol "New"	
	2. Menampilkan kotak isian/form pengguna
3. Memasukan data pengguna	
4. Menekan tombol "Save"	
	5. Memproses *request* dari aktor
Kondisi Akhir : kotak isian pengguna ditutup dan data pengguna yang dimasukan ditampilkan.	

12. *Use Case Scenario* Ubah Pengguna

Skenario di bawah ini menjelaskan tentang cara melakukan perubahan data pengguna.

Tabel 3.17 *Use Case Scenario* Ubah Pengguna

IDENTIFIKASI	
Nomor	UCS-12
Nama	*Use Case* Mengubah Pengguna
Deskripsi	Menjelaskan tentang cara mengubah pengguna
Aktor	Koordinator
Kondisi Awal	Pengguna sedang menampilkan halaman daftar pengguna pada menu "Management"
SKENARIO	
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih *record* pengguna yang akan diubah	
	2. *Record* yang dipilih aktor diberi warna.
3. Menekan tombol "Edit"	
	4. Menampilkan kotak isian pengguna
5. Memasukan data pengguna yang baru	
6. Menekan tombol "Save"	
	7. Memproses *request* dari aktor
Kondisi Akhir : kotak isian pengguna ditutup dan data pengguna yang baru ditampilkan	

13. *Use Case Scenario* Hapus Pengguna

Skenario di bawah ini menjelaskan tentang cara melakukan penghapusan data pengguna.

Tabel 3.18 *Use Case Scenario* Hapus Pengguna

IDENTIFIKASI	
Nomor	UCS-13
Nama	*Use Case* Menghapus Pengguna
Deskripsi	Menjelaskan tentang cara menghapus pengguna
Aktor	Koordinator
Kondisi Awal	Pengguna sedang menampilkan halaman daftar pengguna pada menu "Management"
SKENARIO	
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih *record* pengguna yang akan diubah	
	2. *Record* yang dipilih aktor diberi warna.
3. Menekan tombol "Delete"	
	4. Menampilkan kotak dialog konfirmasi
5. Menekan tombol "Ok"	
	6. Memproses *request* dari aktor
Kondisi Akhir : kotak isian pengguna ditutup dan data pengguna yang terpilih dihapus.	

14. *Use Case Scenario* Memproses Alarm

Skenario di bawah ini menjelaskan tentang cara memproses alarm yang dikirim oleh perangkat HPS.

Tabel 3.19 *Use Case Scenario* Memproses Alarm

IDENTIFIKASI	
Nomor	UCS-14
Nama	*Use Case* Memproses Alarm
Deskripsi	Menjelaskan tentang cara memproses alarm
Aktor	Perangkat HPS
Kondisi Awal	Perangkat HPS dalam kondisi bekerja secara normal
SKENARIO	

Aksi Aktor	Reaksi Sistem
1. Aktor mengirimkan pesan gangguan ke server aplikasi berupa pesan SNMP trap	
	2. Pesan gangguan diparsing
	3. Pesan gangguan divalidasi, jika valid maka disimpan ke database
	4. Mengambil data teknisi sesuai dengan lokasi dari perangkat yang mengirimkan pesan gangguan
	5. Mengirimkan pesan notifikasi email kepada teknisi yang bertugas di lapangan
Kondisi Akhir : memberikan status pengiriman pesan notifikasi.	

15. *Use Case Scenario* Menampilkan Laporan Gangguan

Skenario di bawah ini menjelaskan tentang cara menampilkan laporan gangguan.

Tabel 3.20 *Use Case Scenario* Menampilkan Laporan Gangguan

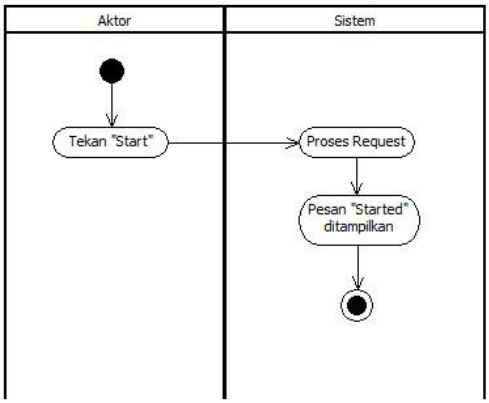
IDENTIFIKASI	
Nomor	UCS-15
Nama	*Use Case* Menampilkan Laporan Gangguan
Deskripsi	Menjelaskan tentang cara menampilkan laporan
Aktor	Koordinator
Kondisi Awal	Pengguna sudah membuka aplikasi alarm handler
SKENARIO	
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan tombol “Alarm” pada toolbar atau menu “Report Alarm”	
	2. Menampilkan form laporan gangguan
3. Memasukan data lokasi dan periode tanggal	
4. Klik tombol “View”	
	5. Menampilkan daftar gangguan
Kondisi Akhir : semua daftar gangguan yang mengacu pada lokasi dan periode tanggal yang dipilih ditampilkan.	

3.3.6.3 *Activity Diagram*

Activity diagram merupakan diagram yang memodelkan aliran kerja dari urutan aktifitas suatu proses yang mengacu pada *use case* diagram. *Activity diagram* yang terdapat pada aplikasi ini adalah sebagai berikut :

1. *Activity Diagram* Menjalankan Aplikasi Alarm Handler

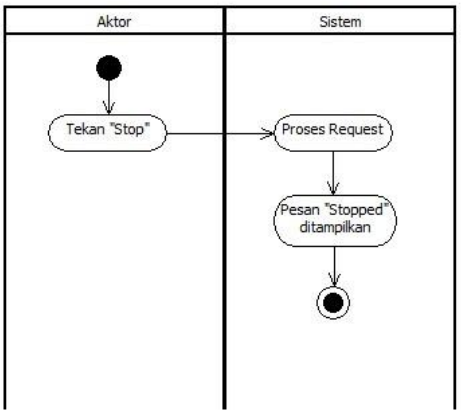
   Activity Diagram di bawah ini menjelaskan tentang cara menjalankan aplikasi alarm handler.



Gambar 3.4 *Activity Diagram* Menjelankan Aplikasi Alarm Handler

2. *Activity Diagram* Menghentikan Aplikasi Alarm Handler

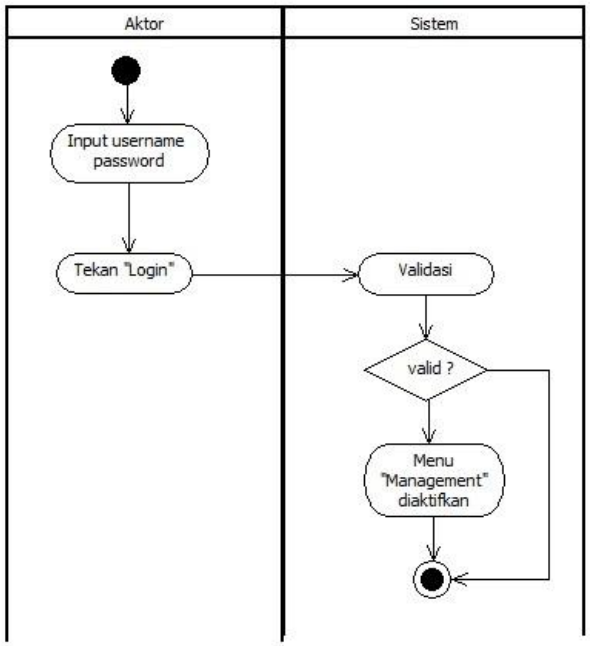
   Activity Diagram di bawah ini menjelaskan tentang cara menghentikan aplikasi alarm handler.



Gambar 3.5 *Activity Diagram* Menghentikan Aplikasi Alarm Handler

3. *Activity Diagram* Login

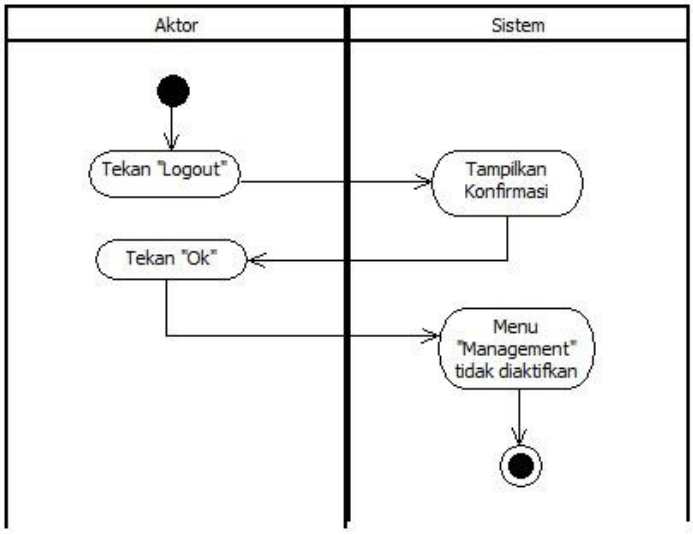
   Activity Diagram di bawah ini menjelaskan tentang cara melakukan login.



Gambar 3.6 *Activity Diagram* Login

4. *Activity Diagram* Logout

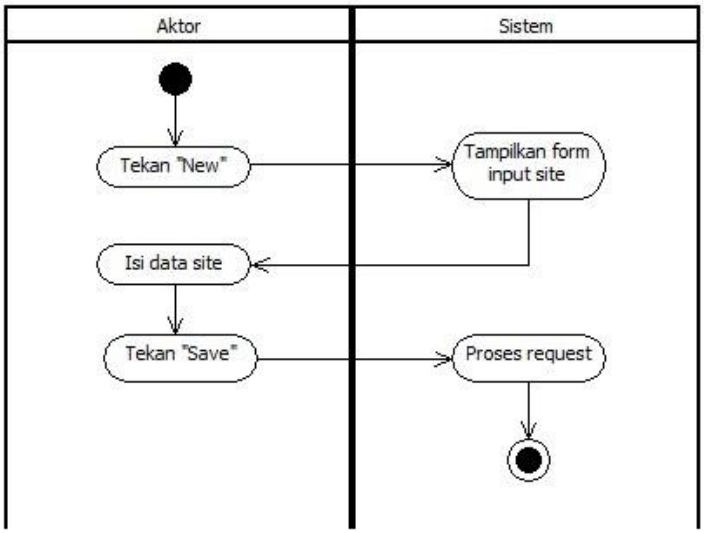
   Activity Diagram di bawah ini menjelaskan tentang cara melakukan logout.



Gambar 3.7 *Activity Diagram* Logout

5. *Activity Diagram* Tambah Lokasi

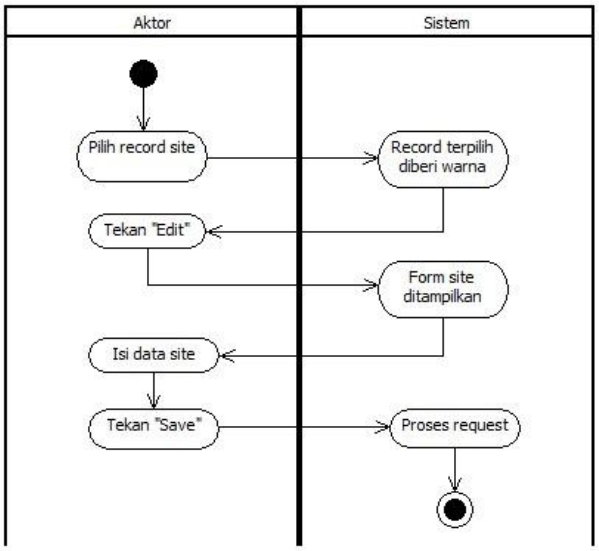
   Activity Diagram di bawah ini menjelaskan tentang cara menambah data lokasi.



Gambar 3.8 *Activity Diagram* Tambah Lokasi

6. *Activity Diagram* Ubah Lokasi

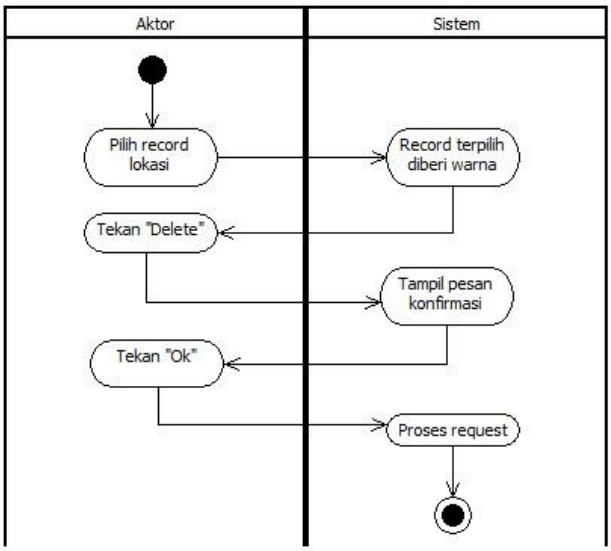
   Activity Diagram di bawah ini menjelaskan tentang cara melakukan perubahan data lokasi.



Gambar 3.9 *Activity Diagram* Ubah Lokasi

7. *Activity Diagram* Hapus Lokasi

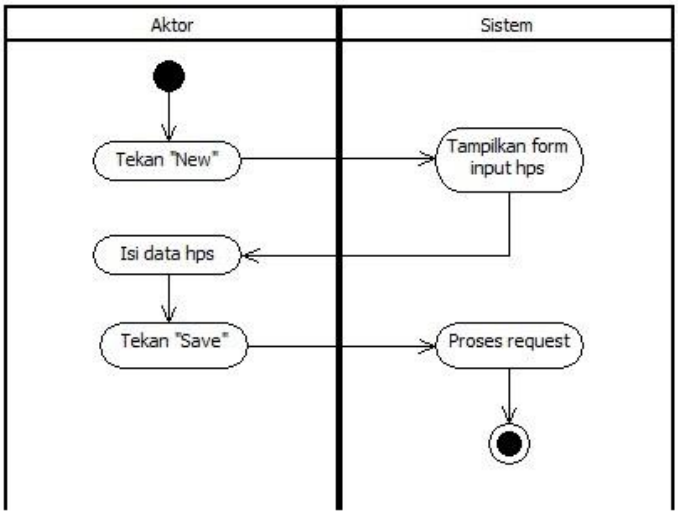
   Activity Diagram di bawah ini menjelaskan tentang cara menghapus lokasi.



Gambar 3.10 *Activity Diagram* Hapus Lokasi

8. *Activity Diagram* Tambah HPS

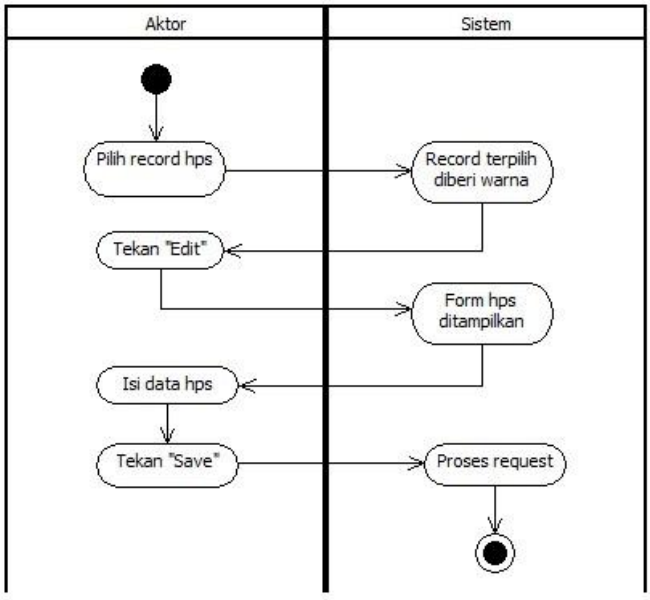
   Activity Diagram di bawah ini menjelaskan tentang cara menambah data perangkat HPS.



Gambar 3.11 *Activity Diagram* Tambah HPS

9. *Activity Diagram* Ubah HPS

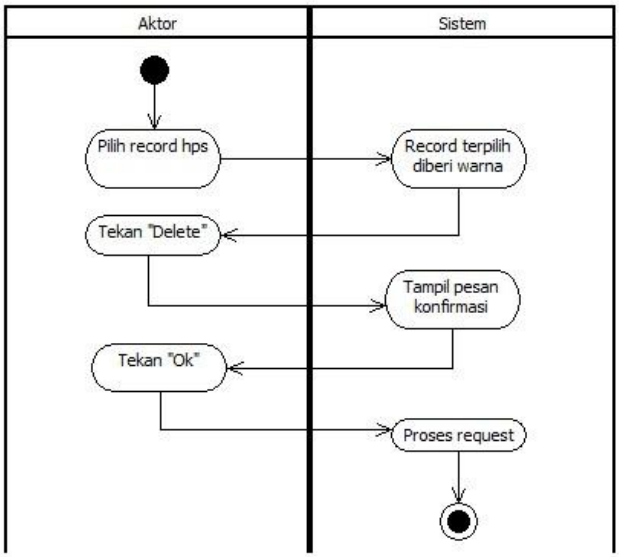
Activity Diagram di bawah ini menjelaskan tentang cara melakukan perubahan data perangkat HPS.



Gambar 3.12 *Activity Diagram* Ubah HPS

10. *Activity Diagram* Hapus HPS

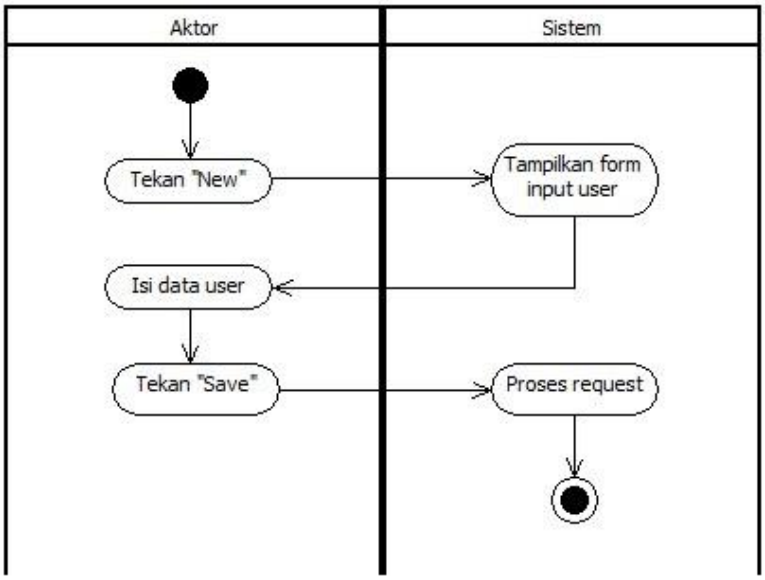
Activity Diagram di bawah ini menjelaskan tentang cara menghapus data perangkat HPS.



Gambar 3.13 *Activity Diagram* Hapus HPS

11. *Activity Diagram* Tambah Pengguna

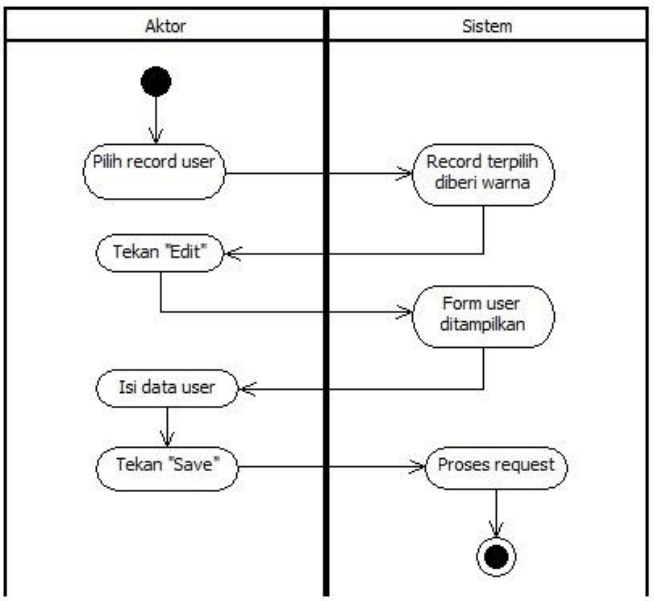
Activity Diagram di bawah ini menjelaskan tentang cara membuat user.



Gambar 3.14 *Activity Diagram* Tambah Pengguna

12. *Activity Diagram* Ubah Pengguna

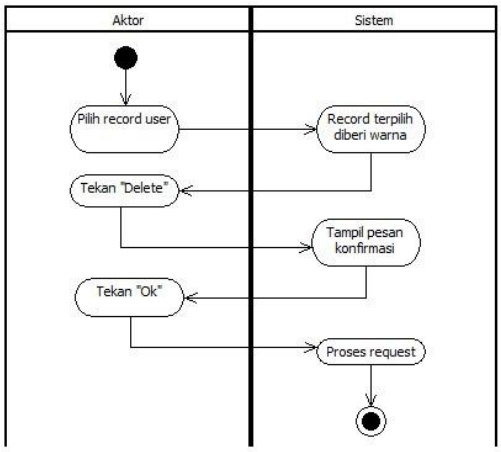
Activity Diagram di bawah ini menjelaskan tentang cara mengubah user.



Gambar 3.15 *Activity Diagram* Ubah Pengguna

13. *Activity Diagram* Hapus Pengguna

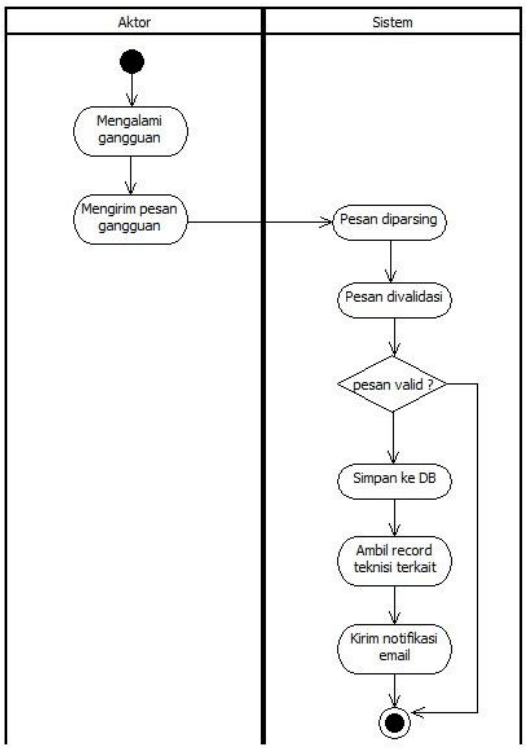
Activity Diagram di bawah ini menjelaskan tentang cara menghapus user.



Gambar 3.16 *Activity Diagram* Hapus Pengguna

14. *Activity Diagram* Memproses Alarm

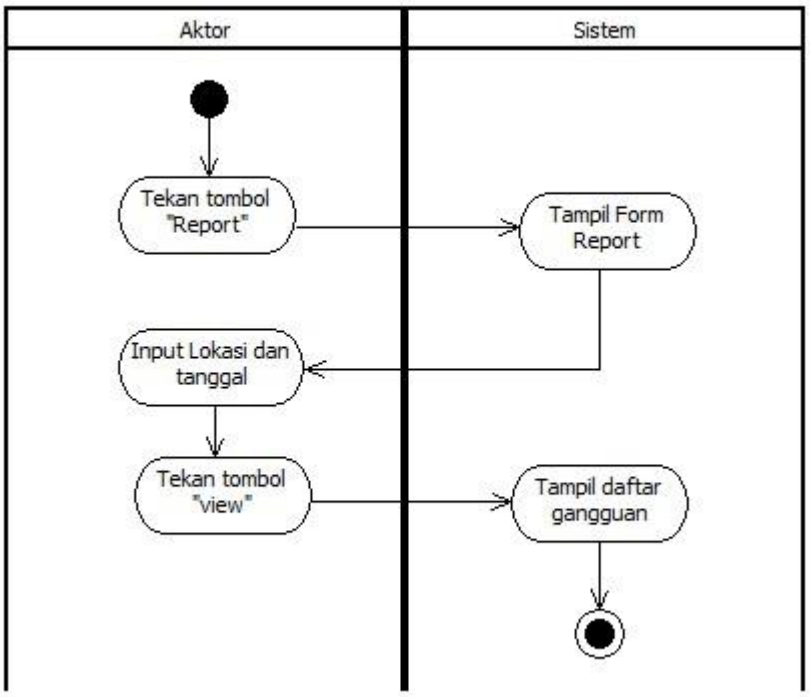
Activity Diagram di bawah ini menjelaskan tentang cara memproses pesan alarm yang dikirim oleh perangkat HPS ke server aplikasi.



Gambar 3.17 *Activity Diagram* Memproses Alarm

15. *Activity Diagram* Menampilkan Laporan Gangguan

Activity Diagram di bawah ini menjelaskan tentang cara menampilkan laporan gangguan.



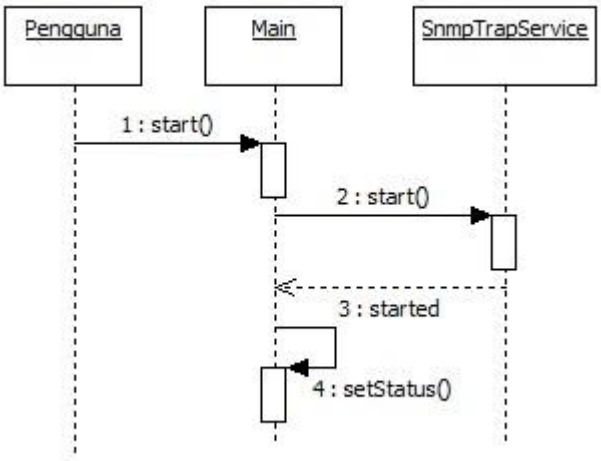
Gambar 3.18 *Activity Diagram* Menampilkan Laporan Gangguan

3.3.6.4 *Sequence Diagram*

Sequence diagram menjelaskan gambaran interaksi antar objek dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek-objek yang saling berinteraksi. *Sequence diagram* pada aplikasi ini adalah sebagai berikut :

1. *Sequence Diagram* Menjalankan Aplikasi Alarm Handler

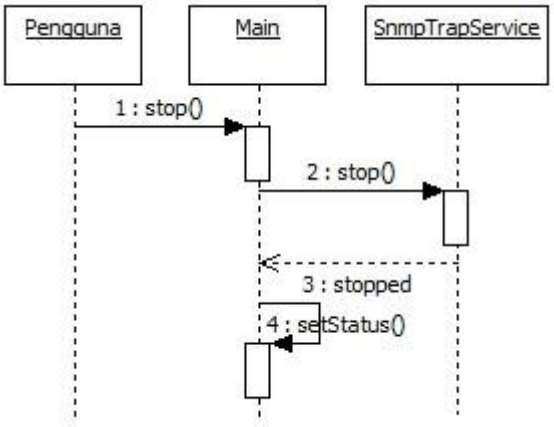
   Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses menjalankan aplikasi alarm handler.



Gambar 3.19 *Sequence Diagram* Menjalankan Aplikasi Alarm Handler

2. *Sequence Diagram* Menghentikan Aplikasi Alarm Handler

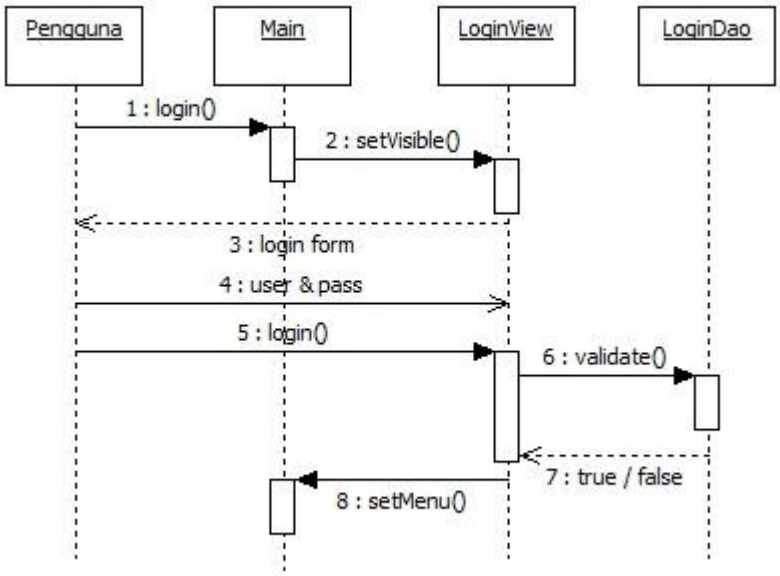
   Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses menghentikan aplikasi alarm handler.



Gambar 3.20 *Sequence Diagram* Menghentikan Aplikasi Alarm Handler

3. *Sequence Diagram* Login

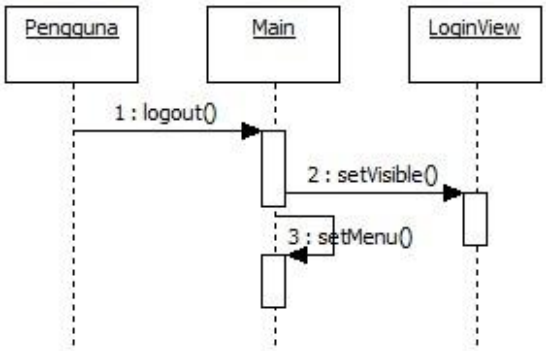
   Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses login.



Gambar 3.21 *Sequence Diagram* Login

4. *Sequence Diagram* Logout

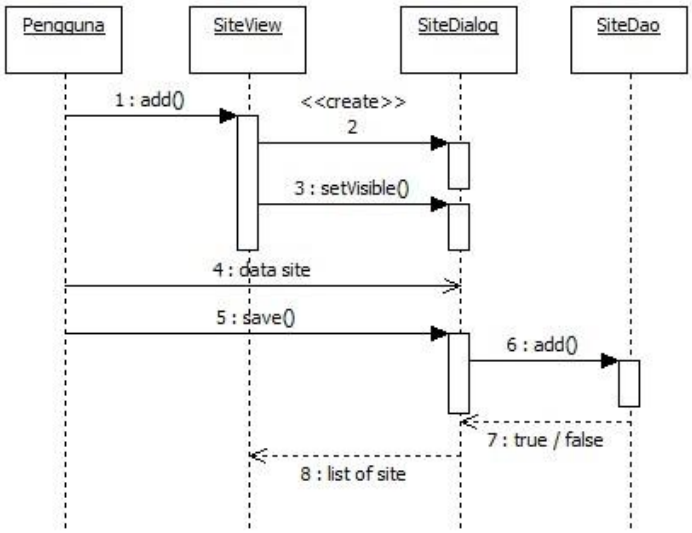
   Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses logout.



Gambar 3.22 *Sequence Diagram* Logout

5. *Sequence Diagram* Tambah Lokasi

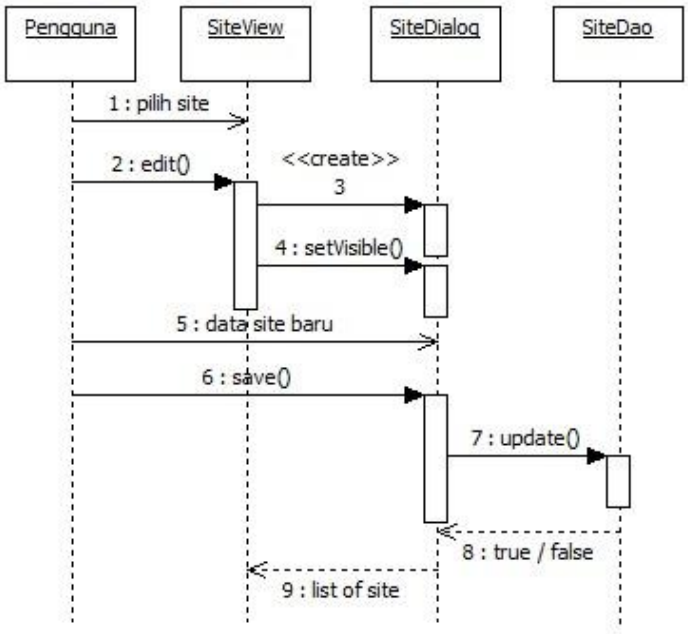
   Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses penambahan data lokasi.



Gambar 3.23 *Sequence Diagram* Tambah Lokasi

6. *Sequence Diagram* Ubah Lokasi

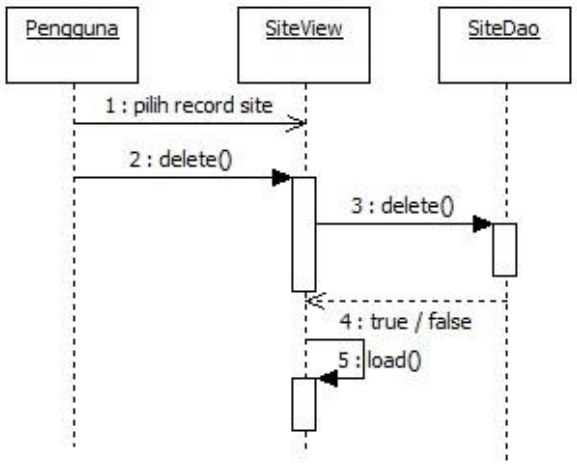
   Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses perubahan data lokasi.



Gambar 3.24 *Sequence Diagram* Ubah Lokasi

7. *Sequence Diagram* Hapus Lokasi

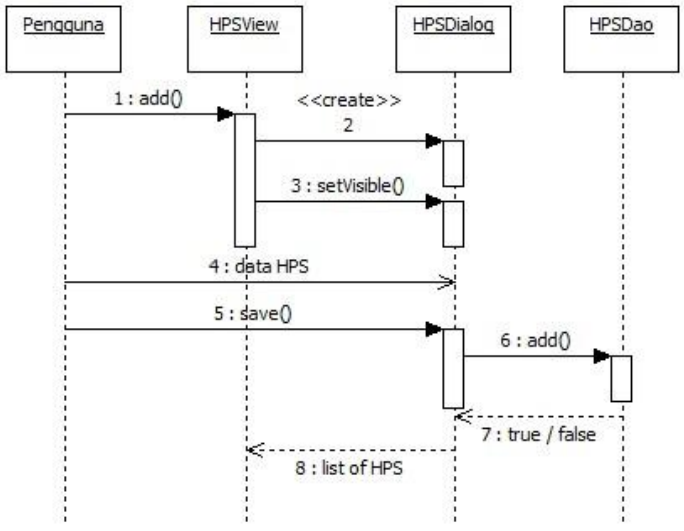
   Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses penghapusan data lokasi.



Gambar 3.25 *Sequence Diagram* Hapus Lokasi

8. *Sequence Diagram* Tambah HPS

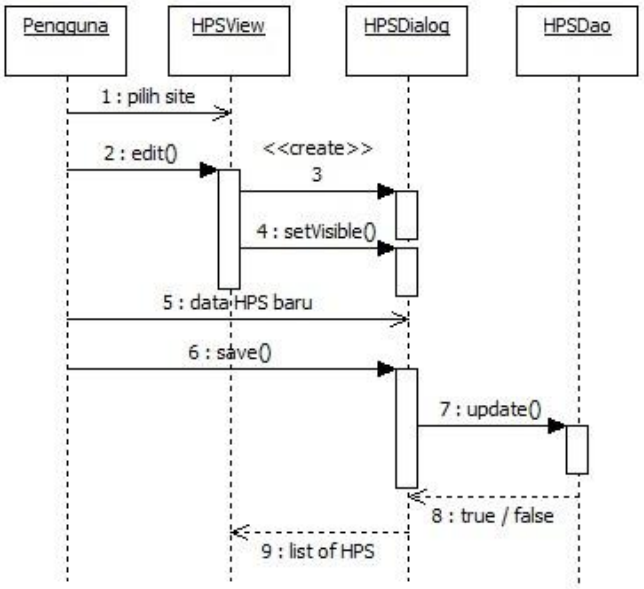
   Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses penambahan data HPS.



Gambar 3.26 *Sequence Diagram* Tambah HPS

9. *Sequence Diagram* Ubah HPS

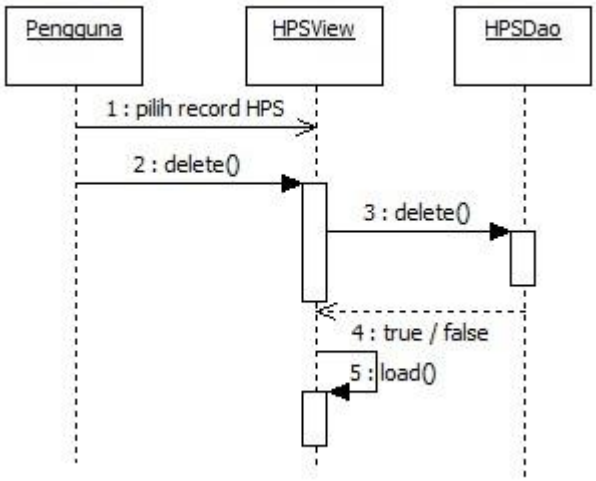
Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses pengubahan data HPS.



Gambar 3.27 *Sequence Diagram* Ubah HPS

10. *Sequence Diagram* Hapus HPS

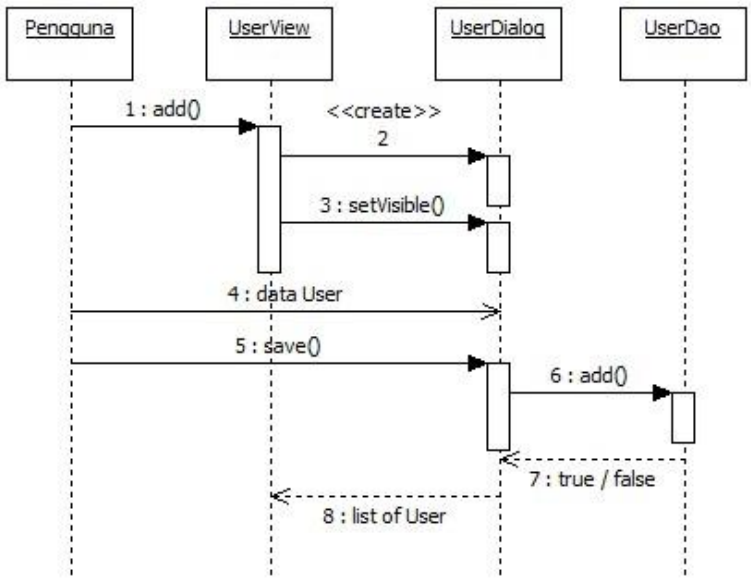
Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses penghapusan data HPS.



Gambar 3.28 *Sequence Diagram* Hapus HPS

11. *Sequence Diagram* Tambah Pengguna

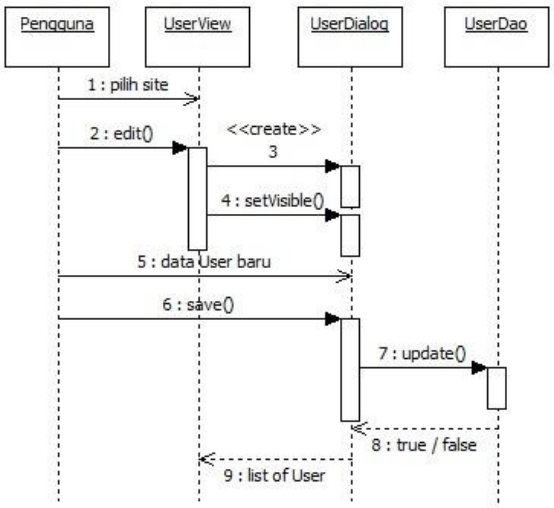
Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses pembuatan user.



Gambar 3.29 *Sequence Diagram* Tambah Pengguna

12. *Sequence Diagram* Ubah Pengguna

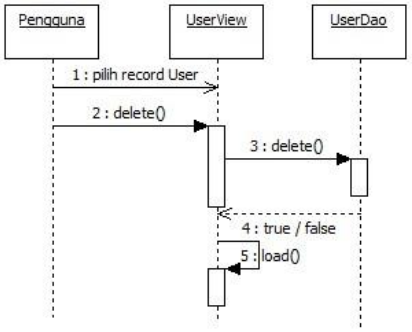
Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses pengubahan user.



Gambar 3.30 *Sequence Diagram* Ubah Pengguna

13. *Sequence Diagram* Hapus Pengguna

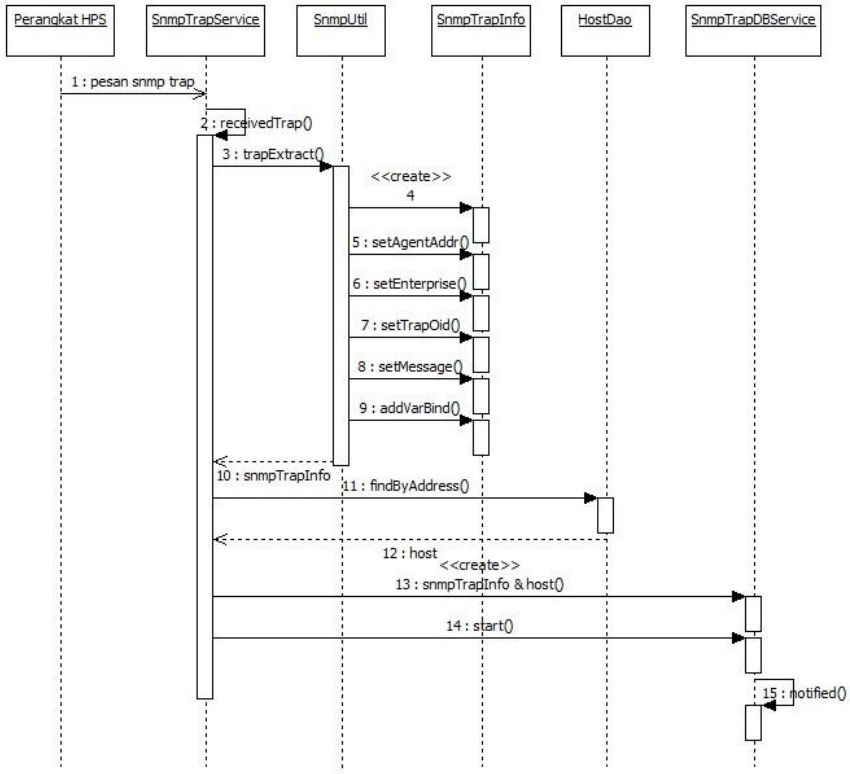
Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses penghapusan user.



Gambar 3.31 *Sequence Diagram* Hapus Pengguna

14. *Sequence Diagram* Memproses Alarm

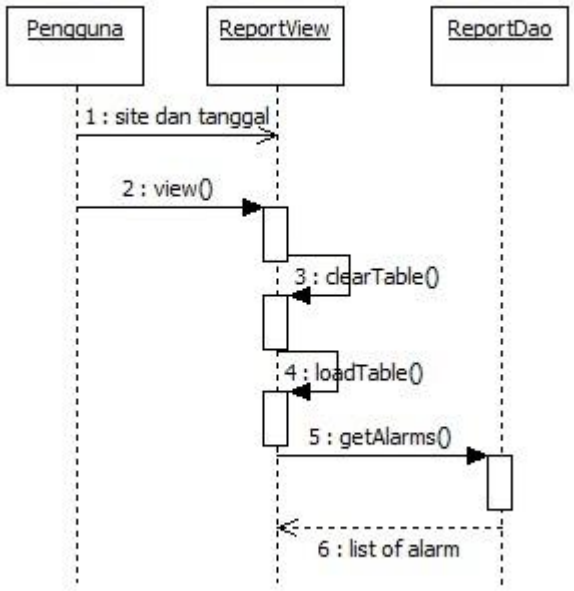
Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses pengolahan pesan alarm yang diterima oleh server aplikasi dari perangkat HPS.



Gambar 3.32 *Sequence Diagram* Memproses Alarm

15. *Sequence Diagram* Menampilkan Laporan Gangguan

Sequence diagram di bawah ini menunjukan rangkaian interaksi pengiriman pesan antar objek pada proses menampilkan laporan gangguan.

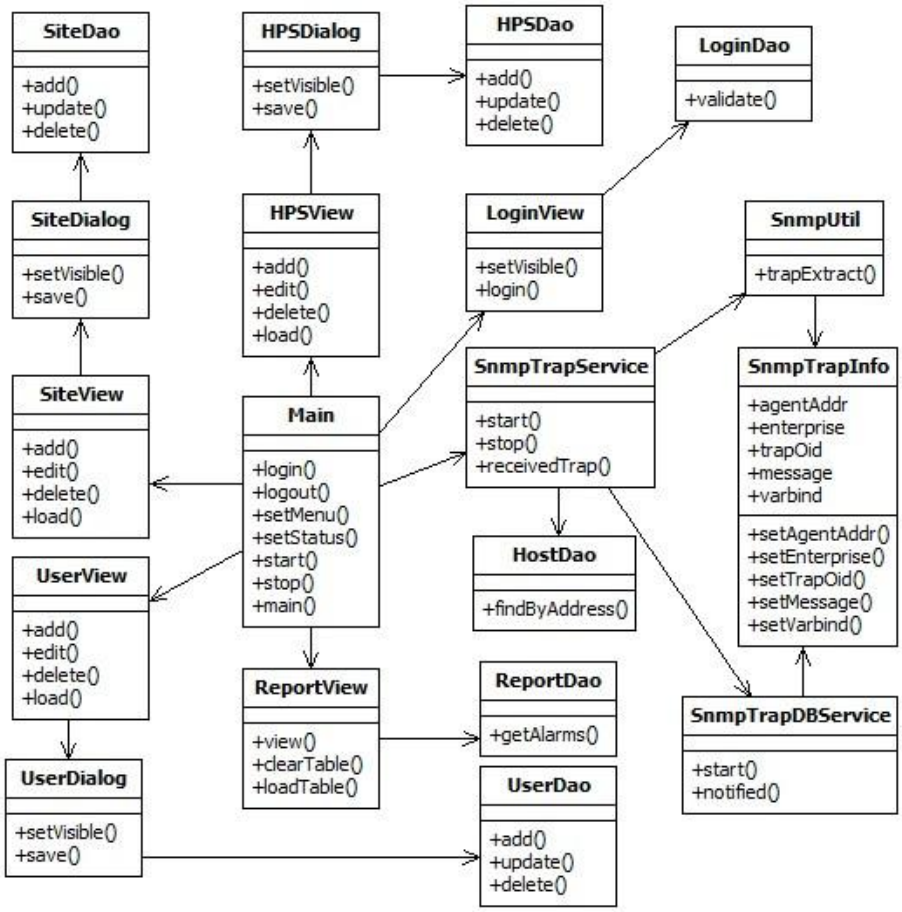


Gambar 3.33 *Sequence Diagram* Menampilkan Laporan Gangguan

3.3.6.5 *Class Diagram*

Class diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas yang ada dalam perangkat lunak yang sedang dikembangkan. *Class diagram* menggambarkan struktur dan deskripsi kelas beserta hubungan kelas yang satu dengan kelas yang lain.

Class diagram yang terdapat dalam aplikasi yang penulis kembangkan adalah sebagai berikut :



Gambar 3.34 *Class Diagram*

Penjelasan untuk tiap-tiap kelas yang terdapat dalam class diagram di atas, dapat dilihat pada tabel di bawah ini.

Tabel 3.21 Deskripsi *Class Diagram* Aplikasi Alarm Handler

Class	**Jenis *Class***	**Deskripsi**
Main	Control	*Class* utama untuk menjalankan aplikasi alarm handler
SiteView	View	*Class* yang berfungsi untuk menampilkan daftar lokasi yang ada di database
SiteDialog	View	*Class* yang berfungsi untuk memasukan data lokasi.
SiteDao	Model	*Class* yang berfungsi untuk melakukan manipulasi data lokasi yang ada di database, seperti penambahan,

		pengubahan dan penghapusan
HPSView	View	*Class* yang berfungsi untuk menampilkan daftar perangkat HPS yang terdaftar di dalam sistem
HPSDialog	View	*Class* yang berfungsi untuk memasukan data perangkat HPS
HPSDao	Model	*Class* yang berfungsi untuk melakukan manipulasi data HPS yang ada di database, seperti penambahan, pengubahan dan penghapusan
UserView	View	*Class* yang berfungsi untuk menampilkan daftar pengguna aplikasi, baik koordinator maupun teknisi lapangan yang bertugas di lokasi
UserDialog	View	*Class* yang berfungsi untuk memasukan data pengguna
UserDao	Model	*Class* yang berfungsi melakukan manipulasi data pengguna yang ada di database, seperti penambahan, pengubahan dan penghapusan data
LoginView	View	*Class* yang berfungsi sebagai antar muka untuk melakukan login
LoginDao	Model	*Class* yang berfungsi untuk melakukan validasi user yang melakukan login
HostDao	Model	*Class* yang berfungsi untuk melakukan validasi perangkat HPS berdasarkan IP address
ReportView	View	*Class* yang berfungsi untuk menampilkan laporan gangguan
ReportDao	Model	*Class* yang berfungsi untuk mengambil data gangguan dari database
SnmpTrapService	Control	*Class* yang berfungsi menerima dan memproses pesan SNMP trap yang dikirim oleh perangkat HPS
SnmpUtil	Control	*Class* yang berfungsi untuk melakukan ekstraksi pesan SNMP trap menjadi struktur data buatan untuk

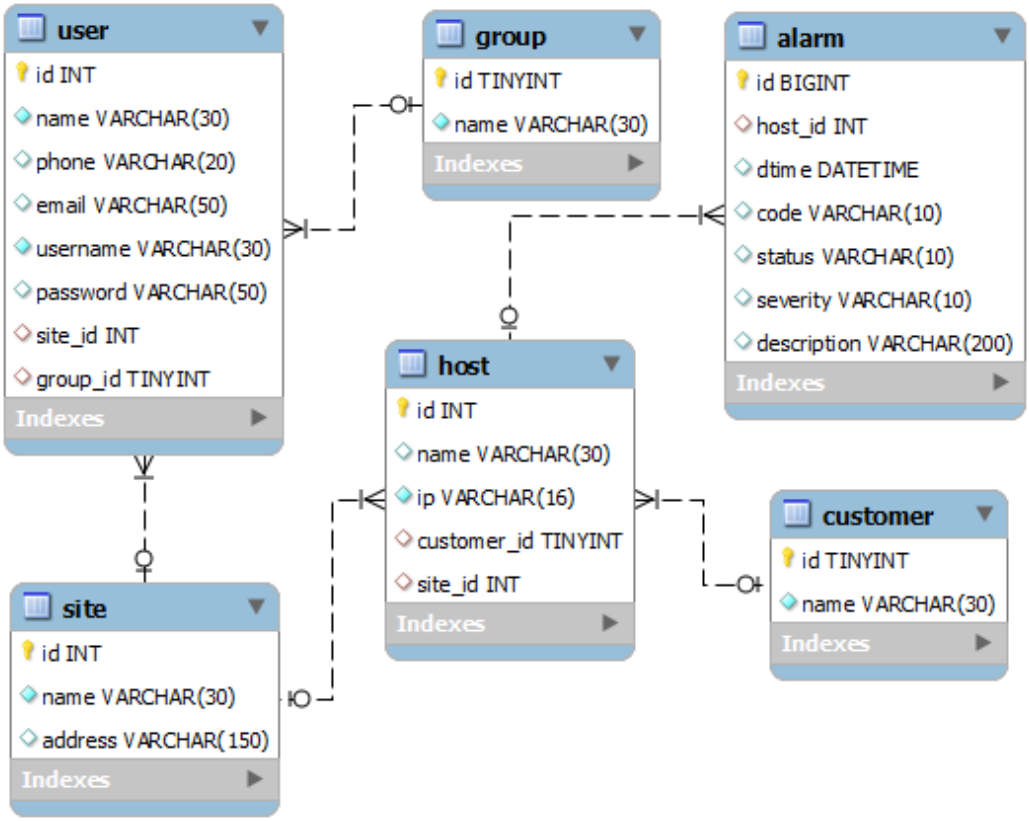
		mempermudah dalam pengambilan informasi
SnmpTrapInfo	Entity	*Class* yang memuat informasi tentang pesan SNMP trap sebagai hasil ekstraksi oleh class SnmpUtil
SnmpTrapDBService	Control	*Class* yang berfungsi untuk menyimpan pesan trap/alarm yang diterima kedalam database dan melakukan notifikasi email ke para teknisi lapangan

3.4 Perancangan Basis Data

Pada sub bab berikut ini akan membahas tentang perancangan basis data untuk pengolahan data pada sistem yang akan dibangun, yang terdiri dari perancangan skema relasi yang menggambarkan tentang keterhubungan antara data yang satu dengan data yang lain (*relationship*) dan struktur tabel yang menggambarkan atribut-atribut untuk suatu objek tertentu berikut dengan tipe datanya. Perancangan basis data dilakukan agar data dapat dibuat dalam struktur yang rapi, sehingga dapat mempermudah dalam melakukan pengelolaan data.

3.4.1 Skema Relasi

Relasi antar tabel merupakan gabungan antar file yang mempunyai kunci utama yang sama, sehingga file-file tersebut menjadi satu kesatuan yang dihubungkan oleh field kunci (*primary key*). Pada proses ini elemen-elemen data dikelompokan menjadi satu file database beserta entitas dan hubungannya. Skema relasi dari sistem yang akan dibangun adalah sebagai berikut :



Gambar 3.35 Skema Relasi

3.4.2 Struktur Tabel

Struktur tabel menggambarkan tentang struktur data dari suatu objek yang dinyatakan dengan atribut-atribut. Struktur tabel pada sistem yang akan dibangun adalah sebagai berikut :

1. Tabel Site

   Tabel yang digunakan untuk menyimpan data lokasi perangkat HPS.

Tabel 3.22 Struktur Tabel Site

Field	Type	Null	Key	Default	Extra	Ref
id	int(11)	No	PK	Null	Auto	
name	varchar(30)	No	UNI	Null		
address	varchar(150)	Yes		Null		

2. Tabel Customer

Tabel yang digunakan untuk menyimpan data pelanggan yang menggunakan perangkat HPS.

Tabel 3.23 Struktur Tabel Customer

Field	Type	Null	Key	Default	Extra	Ref
id	tinyint(4)	No	PK	Null	Auto	
name	varchar(30)	No	UNI	Null		

3. Tabel Host

Tabel yang digunakan untuk menyimpan data perangkat HPS yang terpasang di lokasi.

Tabel 3.24 Struktur Tabel Host

Field	Type	Null	Key	Default	Extra	Ref
id	int(11)	No	PK	Null	Auto	
name	varchar(30)	Yes		Null		
ip	varchar(16)	No	UNI	Null		
customer_id	tinyint(4)	Yes	FK	Null		customer
site_id	int(11)	Yes	FK	Null		site

4. Tabel Group

Tabel yang digunakan untuk menyimpan data jenis pengguna, yang terdiri dari Koordinator dan Teknisi.

Tabel 3.25 Struktur Tabel Group

Field	Type	Null	Key	Default	Extra	Ref
id	tinyint(4)	No	PK	Null	Auto	
name	varchar(30)	No	UNI	Null		

5. Tabel User

Tabel yang digunakan untuk menyimpan data pengguna, baik data Koordinator maupun data Teknisi lapangan.

Tabel 3.26 Struktur Tabel User

Field	Type	Null	Key	Default	Extra	Ref
id	int(11)	No	PK	Null	Auto	
name	varchar(30)	Yes		Null		
phone	varchar(20)	Yes		Null		
email	varchar(50)	No		Null		
username	varchar(30)	No	UNI	Null		
password	varchar(50)	Yes		Null		
site_id	int(11)	Yes	FK	Null		site
group_id	tinyint(4)	Yes	FK	Null		group

6. Tabel Alarm

Tabel yang digunakan untuk menyimpan data gangguan yang dikirimkan oleh perangkat HPS dari lokasi.

Tabel 3.27 Struktur Tabel Alarm

Field	Type	Null	Key	Default	Extra	Ref
id	bigint(20)	No	PK	Null	Auto	
host_id	int(11)	Yes	FK	Null		host
dtime	datetime	Yes		Null		
code	varchar(10)	Yes		Null		
status	varchar(10)	Yes		Null		
severity	varchar(10)	Yes		Null		
description	varchar(200)	Yes		Null		

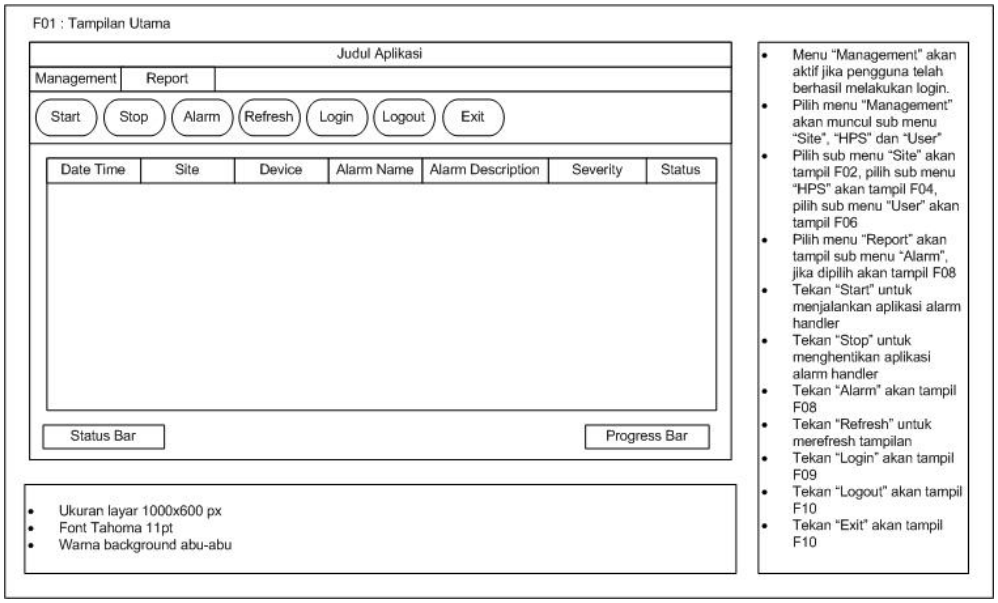
3.5 Perancangan Antarmuka

Interface atau antarmuka merupakan tampilan dari suatu program aplikasi yang berperan sebagai media komunikasi atau sarana dialog antara program dengan pengguna. Antarmuka yang baik adalah antarmuka yang mudah dipahami dan digunakan oleh pengguna.

Perancangan antarmuka mendeskripsikan rencana tampilan dari setiap form yang akan digunakan pada aplikasi yang akan dibangun. Rancangan antarmuka pada aplikasi yang penulis kembangkan adalah sebagai berikut :

1. Perancangan Halaman Utama

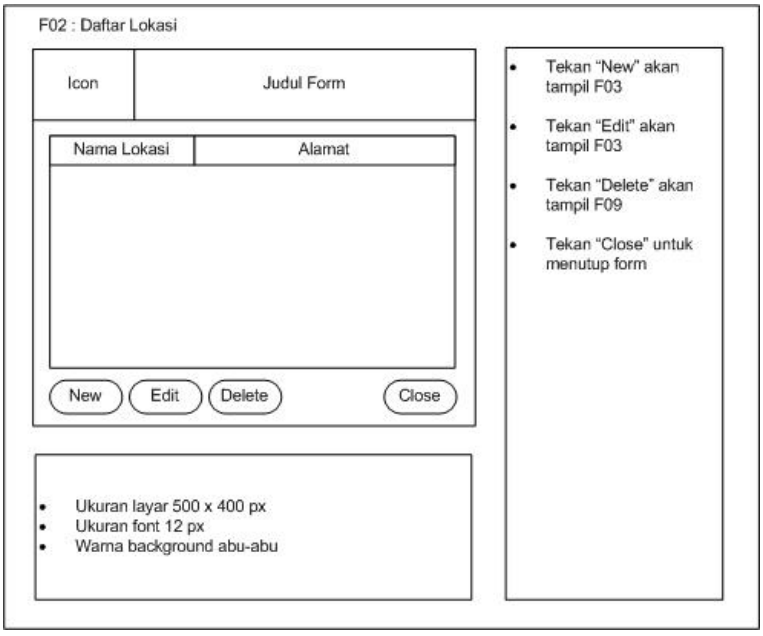
Gambar di bawah ini menggambarkan tampilan utama aplikasi alarm handler.



Gambar 3.36 Perancangan Halaman Utama

2. Perancangan Daftar Lokasi

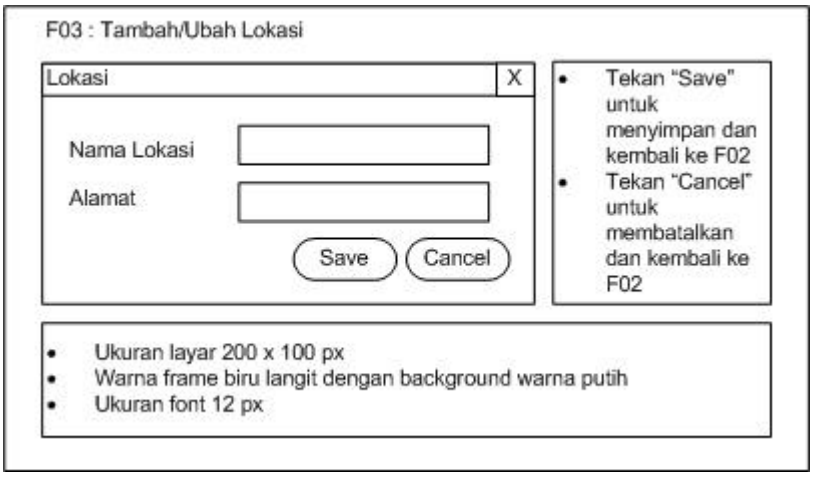
Gambar di bawah ini menggambarkan rancangan tampilan daftar lokasi yang tersimpan di database.



Gambar 3.37 Perancangan Daftar Lokasi

3. Perancangan Input Lokasi

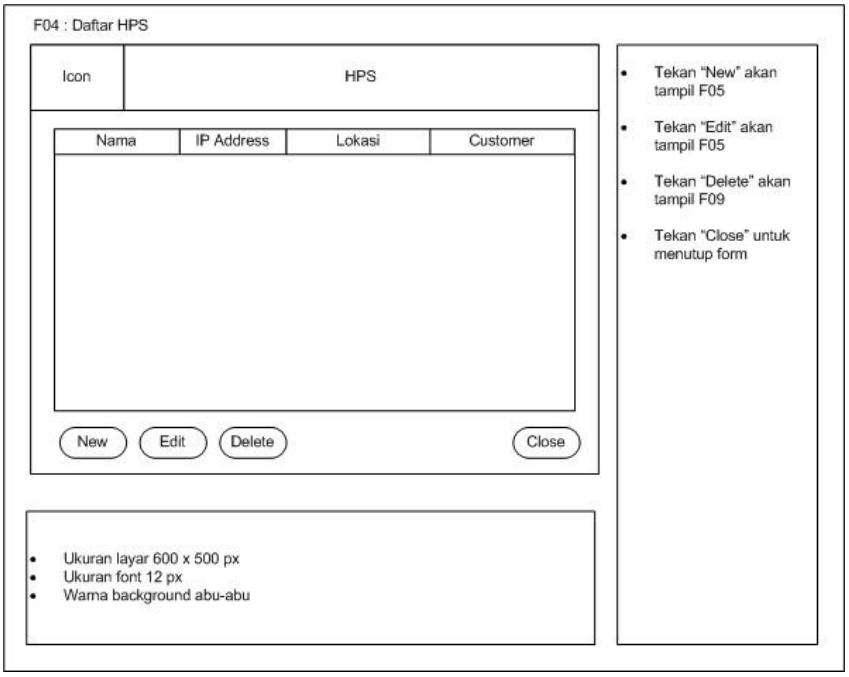
Gambar di bawah ini menggambarkan rancangan tampilan untuk melakukan input data lokasi.



Gambar 3.38 Perancangan Input Lokasi

4. Perancangan Daftar HPS

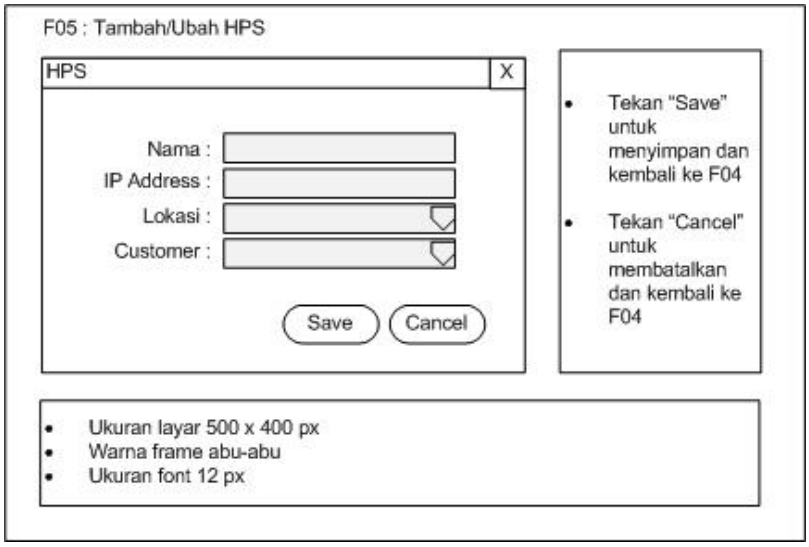
Gambar di bawah ini menggambarkan rancangan tampilan untuk menampilkan daftar perangkat HPS yang tersimpan di database.



Gambar 3.39 Perancangan Daftar HPS

5. Perancangan Input HPS

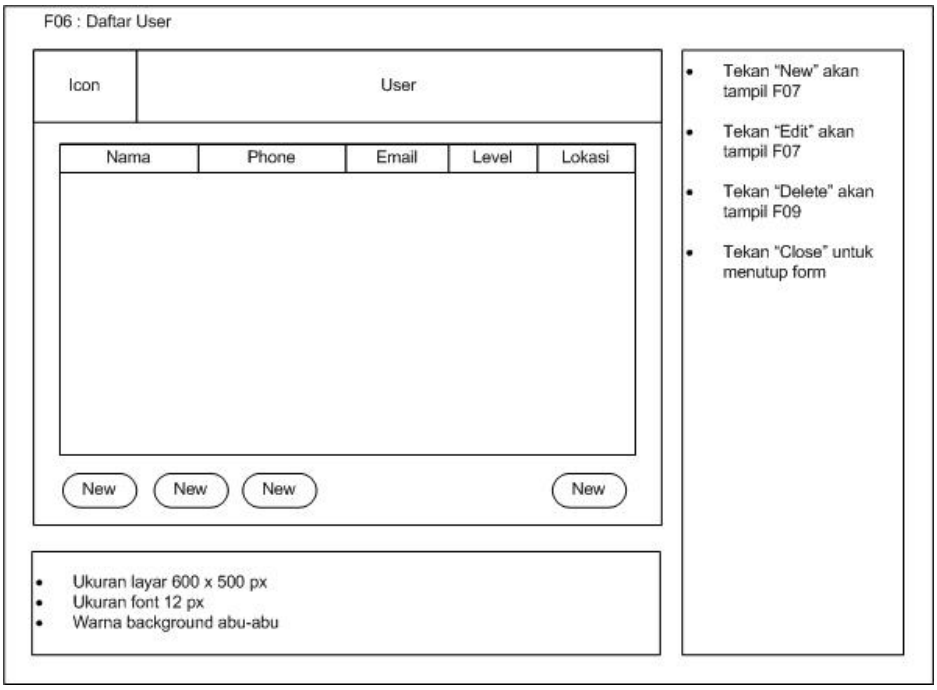
Gambar di bawah ini menggambarkan rancangan tampilan untuk melakukan input data perangkat HPS.



Gambar 3.40 Perancangan Input HPS

6. Perancangan Daftar User

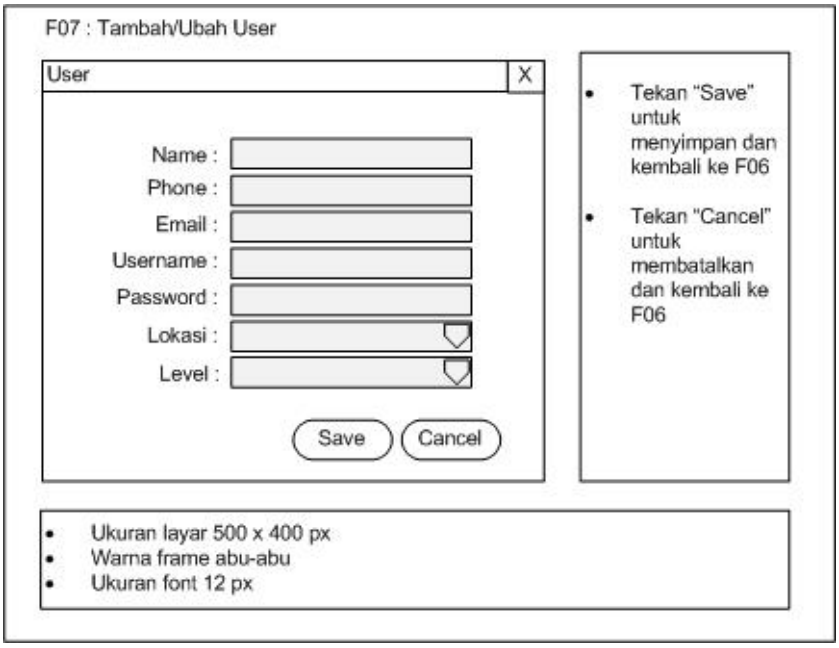
Gambar di bawah ini mengambarkan rancangan tampilan untuk menampilkan data user yang tersimpan di database.



Gambar 3.41 Perancangan Daftar User

7. Perancangan Input User

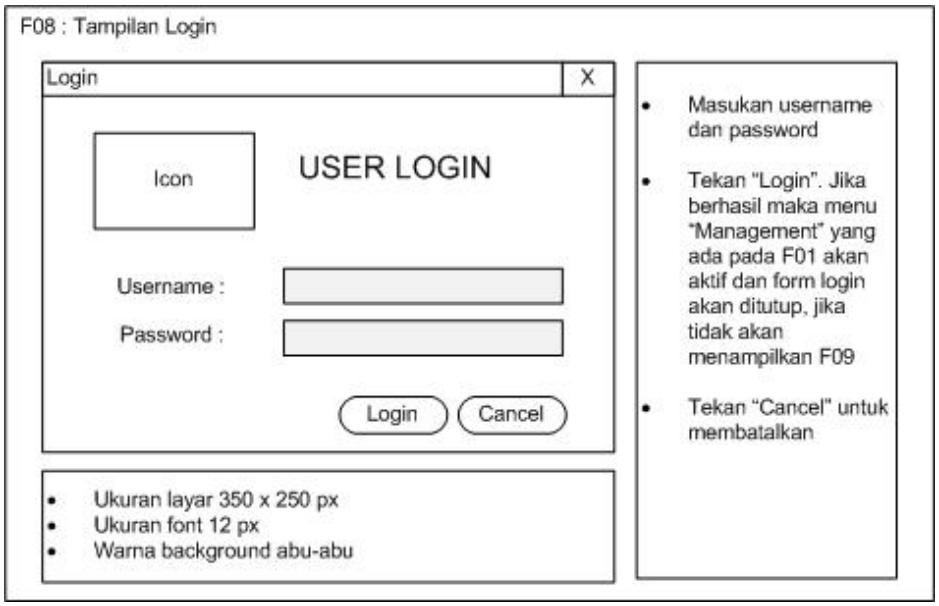
Gambar di bawah ini menggambarkan rancangan tampilan untuk melakukan pembuatan user.



Gambar 3.42 Perancangan Input User

8. Perancangan Login

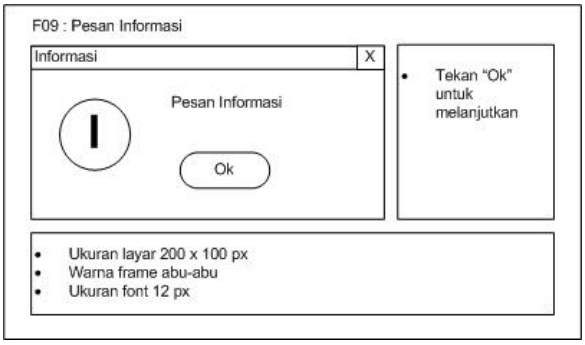
Gambar di bawah ini menggambarkan rancangan tampilan login.



Gambar 3.43 Perancangan Login

9. Perancangan Pesan Informasi

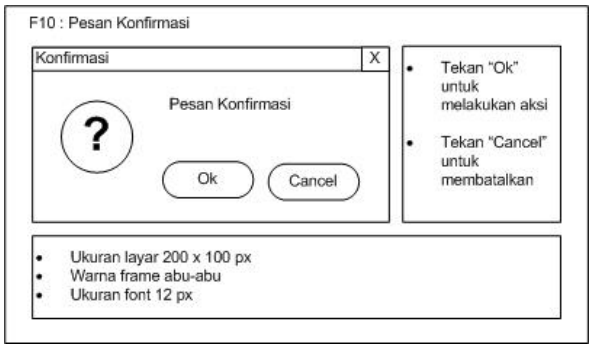
Gambar di bawah ini menggambarkan rancangan tampilan pesan informasi.



Gambar 3.44 Perancangan Pesan Informasi

10. Perancangan Pesan Konfirmasi

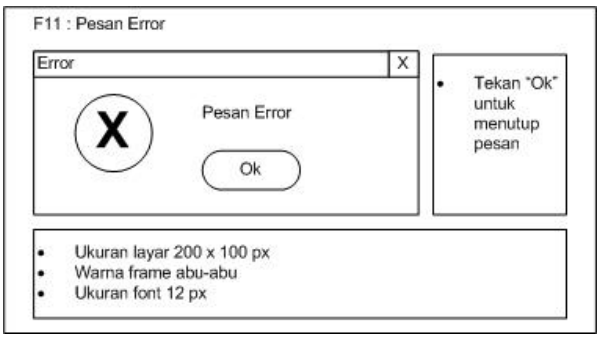
Gambar di bawah ini menggambarkan rancangan tampilan pesan konfirmasi.



Gambar 3.45 Perancangan Pesan Konfirmasi

11. Perancangan Pesan Error

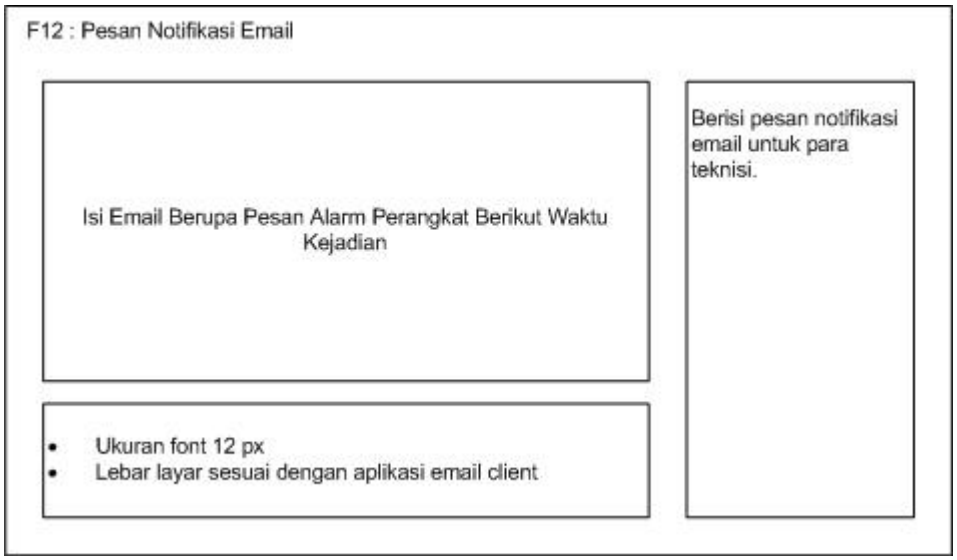
Gambar di bawah ini menggambarkan rancangan tampilan pesan error.



Gambar 3.46 Perancangan Pesan Error

12. Perancangan Pesan Notifikasi Email

Gambar di bawah ini menggambarkan rancangan tampilan pesan notifikasi email.



Gambar 3.47 Perancangan Pesan Notifikasi Email

3.6 Implementasi

Implementasi merupakan suatu tahapan pengaplikasian atas hasil dari analisis dan perancangan sistem yang akan dibangun pada bab sebelumnya. Implementasi yang dimaksud terdiri dari implementasi perangkat keras, perangkat lunak, basis data dan antarmuka aplikasi.

3.6.1 Implementasi Perangkat Keras

Implementasi perangkat keras menjelaskan perangkat keras yang digunakan untuk menjalankan aplikasi dengan spesifikasi sebagai berikut :

Tabel 3.28 Implementasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Processor	Core i5 2.4 GHz
2	Memory	4GB DDR3

3	HDD	500GB
4	VGA	64MB
5	Monitor	LCD 17” resolusi 1366x768 pixel
6	LAN Card	100/1000 MB Fast Ethernet
7	Keyboard+Mouse	Standard USB

3.6.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak menjelaskan perangkat lunak yang digunakan untuk menjalankan aplikasi dengan spesifikasi sebagai berikut :

Tabel 3.29 Implementasi Perangkat Lunak

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 7 Professional 32 Bit
2	Java	JRE 1.6.0 update 33
3	Database	MySQL versi 5.1.33 community
4	Email Client	Zimbra Web Application
5	SNMP Tools	Ireasoning MIB Browser 5.0

3.6.3 Implementasi Basis Data

Implementasi basis data merupakan kumpulan perintah/sintak *Data Definition Language* (DDL) SQL yang digunakan untuk membuat tabel-tabel yang akan digunakan oleh aplikasi alarm handler. Kumpulan perintah SQL untuk masing-masing tabel adalah sebagai berikut :

1. Perintah Membuat Tabel Lokasi (*Site*)

```
CREATE TABLE site
(
    id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(30) NOT NULL,
    address VARCHAR(150),
    PRIMARY KEY(id),
    UNIQUE(name)
) ENGINE=INNODB;
```

2. Perintah Membuat Tabel Pelanggan (*Customer*)

```
CREATE TABLE customer
(
    id TINYINT NOT NULL AUTO_INCREMENT,
    name VARCHAR(30) NOT NULL,
    PRIMARY KEY(id),
    UNIQUE(name)
) ENGINE=INNODB;
```

3. Perintah Membuat Tabel Perangkat HPS (*Host*)

```
CREATE TABLE host
(
    id INT NOT NULL AUTO INCREMENT,
    name VARCHAR(30),
    ip VARCHAR(16) NOT NULL,
    customer_id TINYINT,
    site_id INT,
    PRIMARY KEY(id),
    UNIQUE(ip),
    CONSTRAINT fk_site FOREIGN KEY(site_id)
        REFERENCES site(id),
    CONSTRAINT fk_cust FOREIGN KEY(customer_id)
        REFERENCES customer(id)
) ENGINE=INNODB;
```

4. Perintah Membuat Tabel Jenis Pengguna (*Group*)

```
CREATE TABLE `group`
(
    id TINYINT NOT NULL AUTO INCREMENT,
    name VARCHAR(30) NOT NULL,
    PRIMARY KEY(id),
    UNIQUE(name)
) ENGINE=INNODB;
```

5. Perintah Membuat Tabel Pengguna (*User*)

```
CREATE TABLE `user`
(
    id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(30) NOT NULL,
    phone VARCHAR(20),
    email VARCHAR(50),
    username VARCHAR(30) NOT NULL,
    password VARCHAR(50),
    site_id INT,
    group id TINYINT,
    PRIMARY KEY(id),
    UNIQUE(username),
    FOREIGN KEY(site_id) REFERENCES site(id),
    FOREIGN KEY(group_id) REFERENCES `group`(id)
) ENGINE=INNODB;
```

6. Perintah Membuat Tabel Gangguan (*Alarm*)

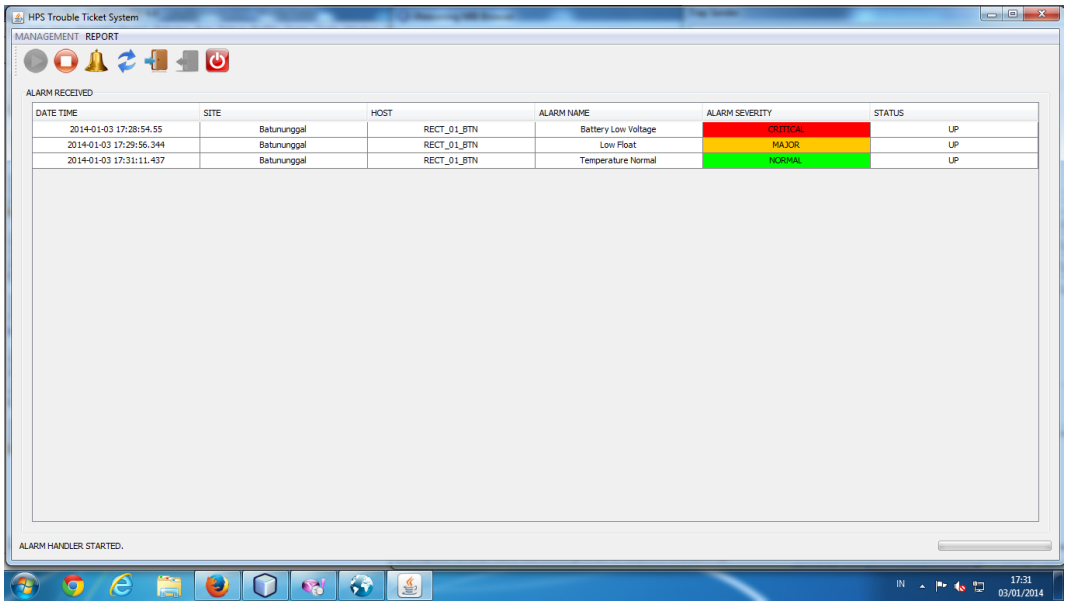
```
CREATE TABLE alarm
(
    id BIGINT NOT NULL AUTO_INCREMENT,
    host_id INT,
    dtime DATETIME,
    code VARCHAR(10),
    status VARCHAR(10),
    severity VARCHAR(10),
    description VARCHAR(200),
    PRIMARY KEY(id),
    FOREIGN KEY(host_id) REFERENCES host(id)
) ENGINE=INNODB;
```

3.6.4 Implementasi Antarmuka

Implementasi antarmuka merupakan penerapan konsep desain tampilan program sebagai media interaksi antara pengguna dengan program aplikasi. Implementasi antarmuka pada aplikasi alarm handler adalah sebagai berikut :

1. Implementasi Halaman Utama

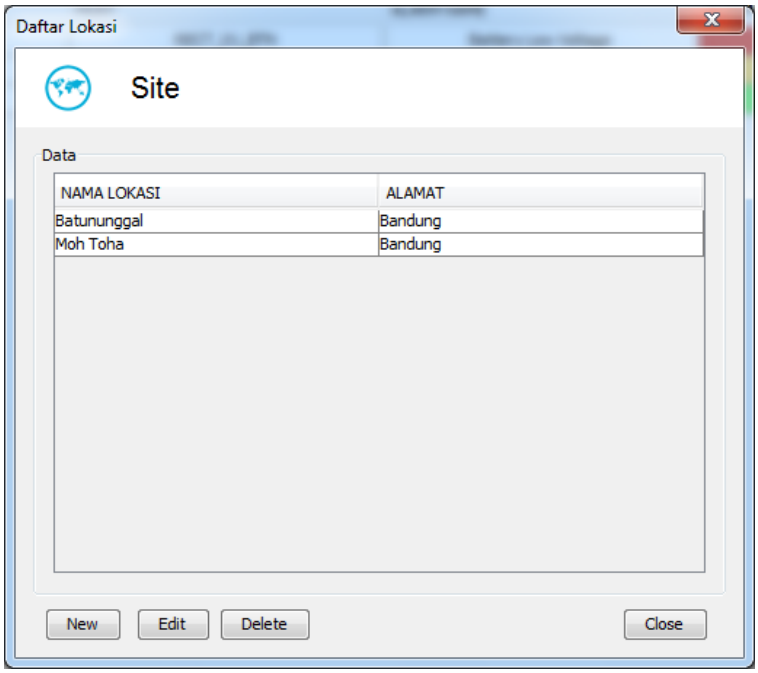
   Gambar 3.48 di bawah ini merupakan implementasi tampilan halaman utama.



Gambar 3.48 Implementasi Halaman Utama

2. Implementasi Daftar Lokasi

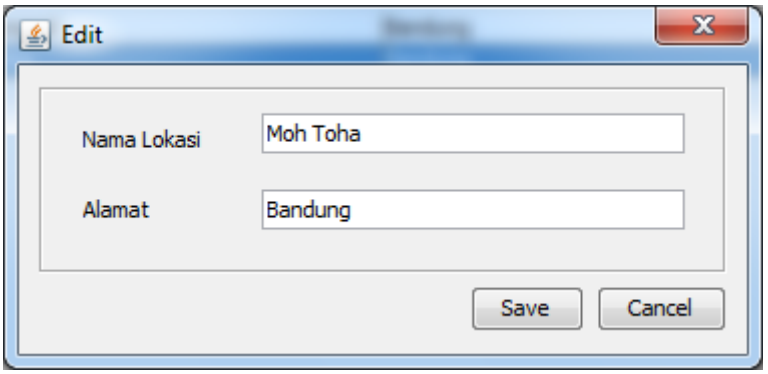
Gambar 3.49 di bawah ini merupakan implementasi tampilan daftar lokasi.



Gambar 3.49 Implementasi Daftar Lokasi

3. Implementasi Input Lokasi

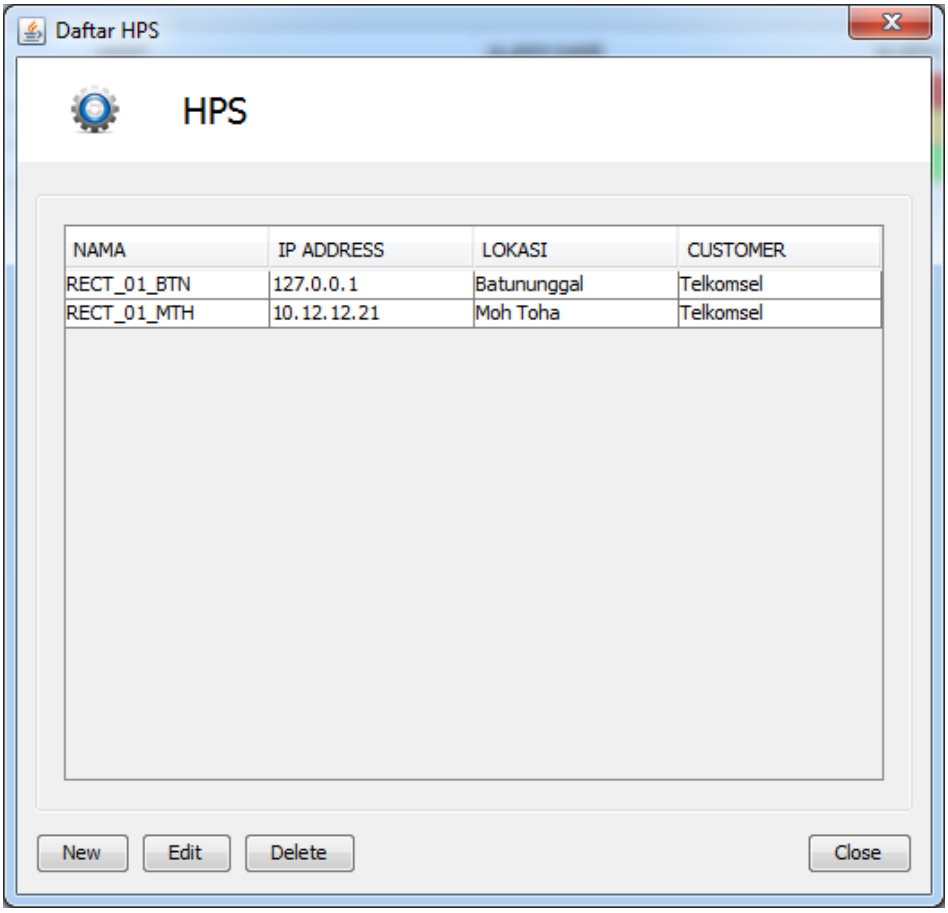
Gambar 3.50 di bawah ini merupakan implementasi tampilan input lokasi.



Gambar 3.50 Implementasi Input Lokasi

4. Implementasi Daftar HPS

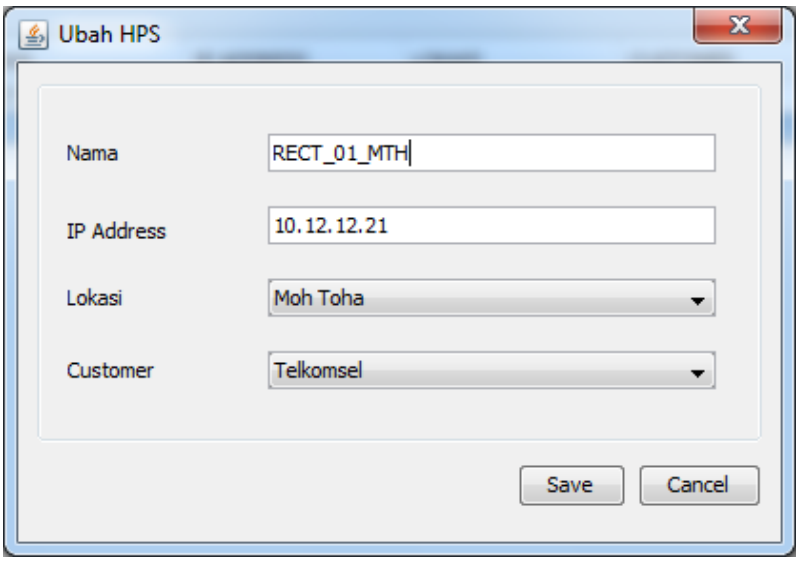
Gambar 3.51 di bawah ini merupakan implementasi tampilan daftar HPS.



Gambar 3.51 Implementasi Daftar HPS

5. Implementasi Input HPS

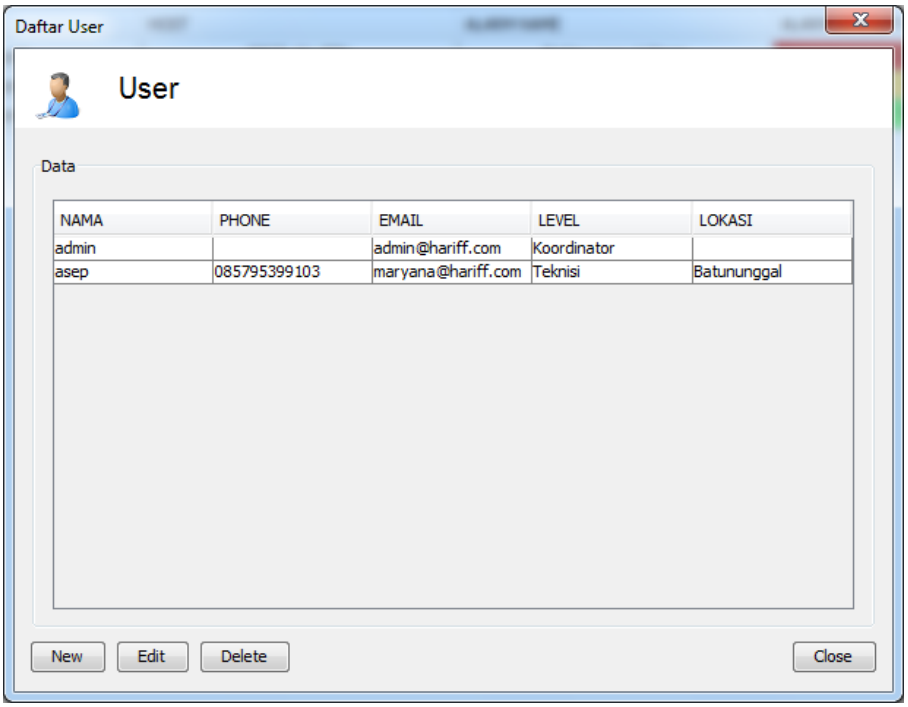
Gambar 3.52 di bawah ini merupakan implementasi tampilan input HPS.



Gambar 3.52 Implementasi Input HPS

6. Implementasi Daftar User

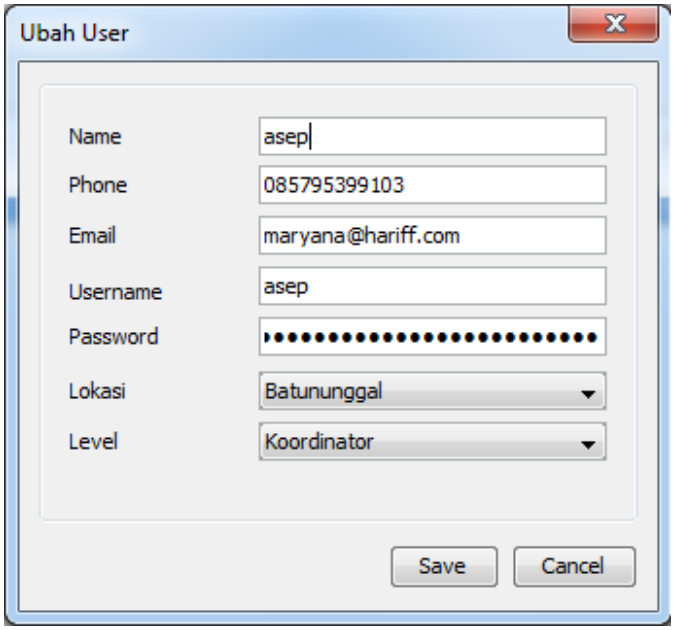
Gambar 3.53 di bawah ini merupakan implementasi tampilan daftar user.



Gambar 3.53 Implementasi Daftar User

7. Implementasi Input User

Gambar 3.54 di bawah ini merupakan implementasi tampilan input user.



Gambar 3.54 Implementasi Input User

8. Implementasi Login

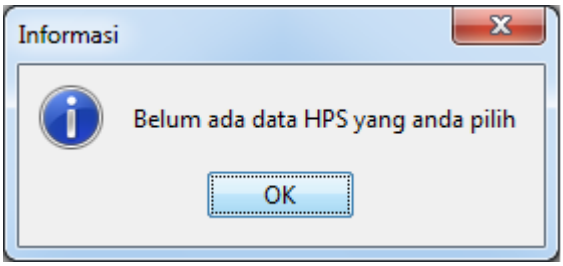
Gambar 3.55 di bawah ini merupakan implementasi tampilan login.



Gambar 3.55 Implementasi Login

9. Implementasi Pesan Informasi

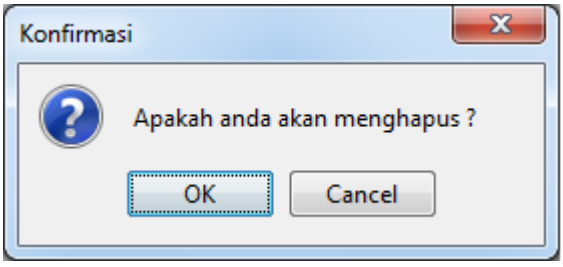
Gambar 3.56 di bawah ini merupakan contoh tampilan pesan informasi saat akan melakukan ubah data HPS, tetapi belum ada data HPS yang dipilih.



Gambar 3.56 Implementasi Pesan Informasi

10. Implementasi Pesan Konfirmasi

Gambar 3.57 di bawah ini merupakan contoh tampilan pesan konfirmasi saat akan melakukan penghapusan data.



Gambar 3.57 Implementasi Pesan Konfirmasi

11. Implementasi Pesan Error

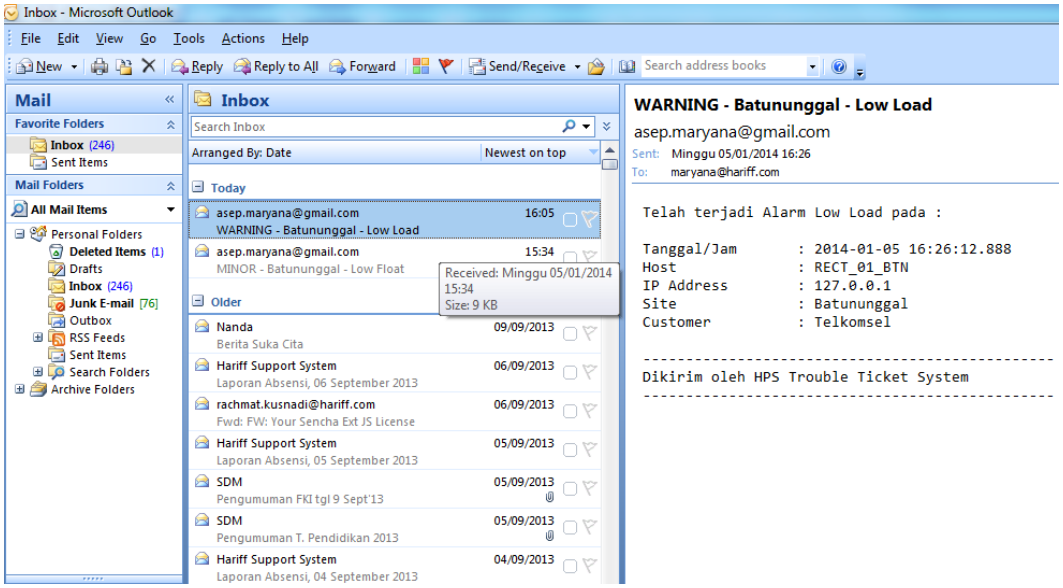
Gambar 3.58 di bawah ini merupakan contoh tampilan pesan error saat melakukan login ke sistem.



Gambar 3.58 Implementasi Pesan Error

12. Implementasi Pesan Notifikasi Email

Gambar 5.59 di bawah ini merupakan contoh tampilan pesan notifikasi email yang diterima oleh teknisi.



Gambar 3.59 Implementasi Pesan Email Notifikasi

3.7 Pengujian

Pengujian sistem merupakan hal terpenting yang dilakukan untuk menemukan kekurangan atau kesalahan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak.

Adapun metode pengujian yang digunakan pada perangkat lunak ini adalah metode *white box* dan *black box*. Pengujian *white box* adalah pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada dan menganalisis apakah ada kesalahan atau tidak. Sedangkan metode pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Metode pengujian *black box* sendiri terdiri dari dua tahapan pengujian yaitu tahap pengujian alpha dan tahap pengujian beta.

3.7.1 Pengujian *White Box*

Pengujian *white box* merupakan metode perancangan kasus uji yang menggunakan struktur kontrol dari perancangan prosedural dalam mendapatkan kasus uji. Adapun metode yang digunakan dalam pengujian *white box* ini adalah metode *basis path*. Metode *basis path* mengijinkan pendesain kasus uji untuk mendapatkan perkiraan lojik yang kompleks dari desain prosedural dan menggunakan perkiraan ini untuk mendefinisikan aliran eksekusi.

Pengujian menggunakan metode white box dalam penyusunan laporan kerja praktek ini adalah pengujian untuk pemrosesan pesan gangguan yang dikirim oleh perangkat HPS ke server aplikasi, dengan proses-proses sebagai berikut :

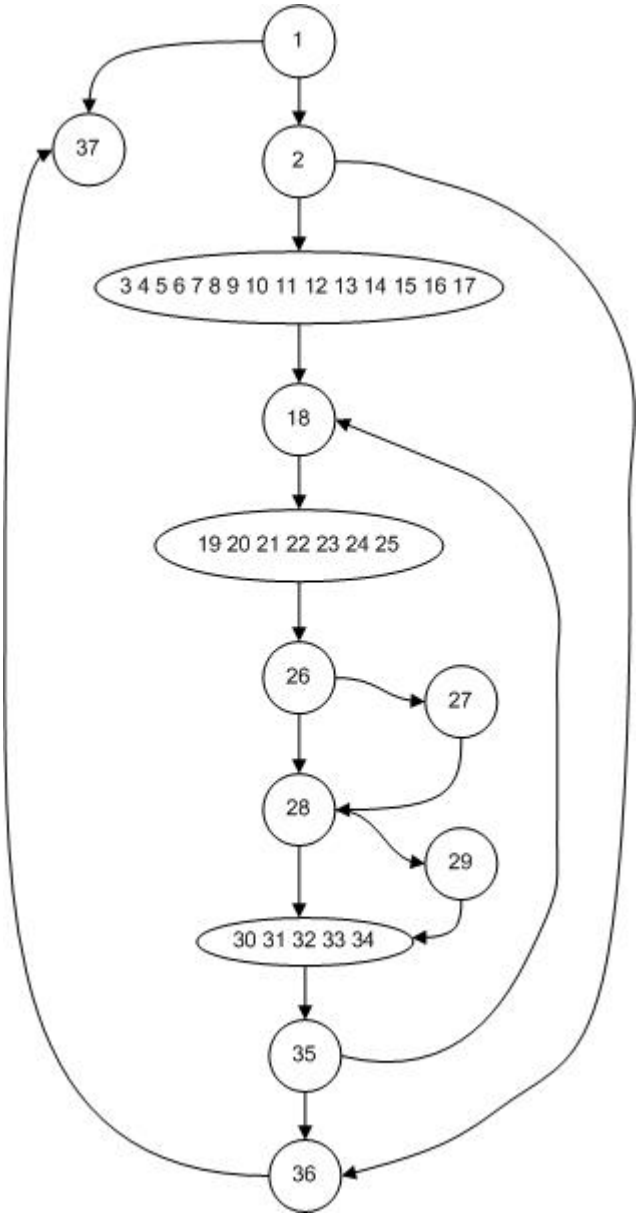
1. Pengujian *White Box Extract SNMP Trap*

   Berikut kode untuk melakukan *extract* pesan *SNMP Trap*.

```
1.   com.adventnet.snmp.mibs.MibNode notification =
     trap.getNotificationDefinition();
2.   if(notification != null){
3.       snmpInfo = new SnmpTrapInfo();
4.       snmpInfo.setRequestID(trap.getReqid());
5.       snmpInfo.setVersion(2);
6.       snmpInfo.setDtime(new
```

```
     java.sql.Timestamp(System.currentTimeMillis()));
7.       snmpInfo.setMessage
     (notification.getDescription().replaceAll("\"",
     ""));
8.
     snmpInfo.setAgentAddr(trap.getAgentAddress());
9.       snmpInfo.setFromAddr(trap.getRemoteHost());
10.      snmpInfo.setCommunity(trap.getCommunity());
11.      snmpInfo.setEnterprise
     (notification.getParent().getLabel());
12.      snmpInfo.setUptime(trap.getUpTime());
13.      snmpInfo.setTrapOid(notification.getLabel());
14.      Vector vbs =
     trap.getTrapPDU().getVariableBindings();
15.      int n = vbs.size();
16.      String idx = "";
17.      int v = 1;
18.      for(int i=2; i<n; i++){
19.          SnmpVarBind vb  = (SnmpVarBind)
     vbs.get(i);
20.          SnmpOID oid     = vb.getObjectID();
21.          String oidStr   = oid.toString();
22.          SnmpVar var     = vb.getVariable();
23.          MibNode mibNode = mibOps.getMibNode(oid);
24.          MibModule mibModule =
     mibOps.getMibModule(mibNode);
25.          String nodeName =
     mibModule.translateToNames(oidStr);
26.          if(mibNode.isIndex())
27.              idx += "." + var.toValue();
28.          if(mibNode.isTableColumn())
29.              nodeName += idx;
30.          SnmpVarBindInfo hvb = new
     SnmpVarBindInfo("var_name" + v,
     getNameObjectID(nodeName));
31.          snmpInfo.addVarBind(hvb);
32.          hvb = new SnmpVarBindInfo("var_val" + v,
     var.toValue());
33.          snmpInfo.addVarBind(hvb);
34.          v++;
35.      }
36.  }
```

Berdasarkan kode *extract SNMP Trap* maka diagram alirnya adalah sebagai berikut :



Gambar 3.60 Diagram Alir *Extract SNMP Trap*

Dari diagram alir *extract SNMP trap*, dapat dihitung *cyclomatic complexity*, yakni :

V(G) = Jumlah Busur – Jumlah Simpul + 2

V(G) = 17 – 13 + 2 = 6

V(G) = Predicate node + 1

V(G) = 5 + 1 = 6

Dari hasil perhitungan *cyclomatic complexity* terdapat *independent path*, yaitu :

Path 1 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-28-30-31-32-33-34-35-36-37

Path 2 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-30-31-32-33-34-35-36

Path 3 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37

Path 4 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-28-29-30-31-32-33-34-35-36-37

Path 5 = 1-2-36-37

Path 6 = 1-37

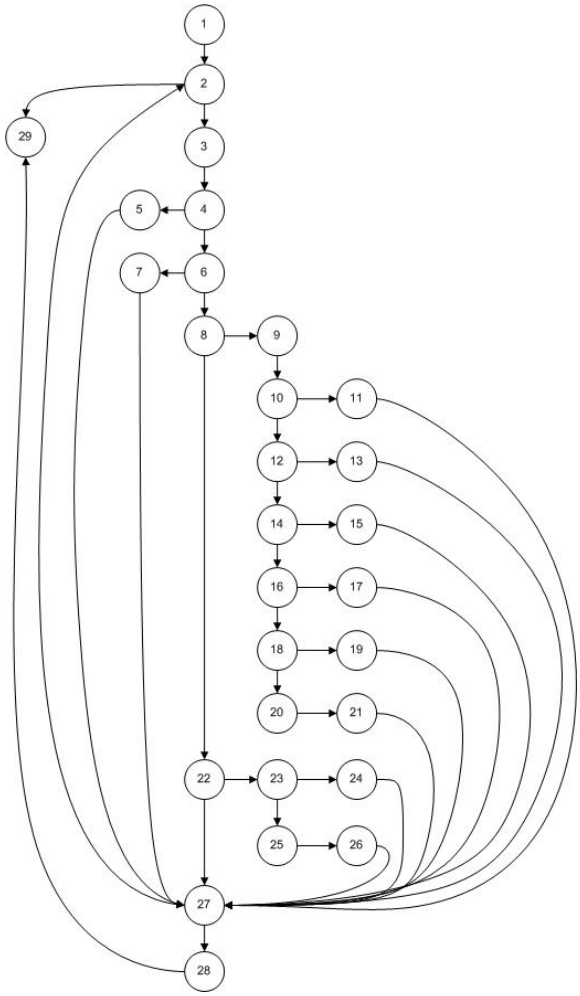
2. Pengujian *White Box Traslate SNMP Trap To Alarm*

   Berikut merupakan kode untuk melakukan translasi pesan SNMP trap menjadi pesan alarm.

```
1.   Iterator<SnmpVarBindInfo> iter =
     trapInfo.getVarBinds().iterator();
2.   while(iter.hasNext()) {
3.       SnmpVarBindInfo el = (SnmpVarBindInfo)
     iter.next();
4.       if(el.getVarName().equals("var_val6"))
5.           alarm.setCode((String) el.getVarValue());
6.       else if(el.getVarName().equals("var_val5"))
7.           alarm.setDescription((String)
     el.getVarValue());
8.       else if(el.getVarName().equals("var_val4")) {
9.           int severity =
     ((Integer)el.getVarValue()).intValue();
10.          if(severity == 2)
11.              alarm.setSeverity("NORMAL");
12.          else if(severity == 3)
13.              alarm.setSeverity("WARNING");
14.          else if(severity == 4)
```

```
15.              alarm.setSeverity("MINOR");
16.          else if(severity == 5)
17.              alarm.setSeverity("MAJOR");
18.          else if(severity == 6)
19.              alarm.setSeverity("CRITICAL");
20.          else
21.              alarm.setSeverity("UNKNOWN");
22.      } else if(el.getVarName().equals("var_val3"))
     {
23.
     if(el.getVarValue().toString().equals("1"))
24.              alarm.setStatus("ACTIVE");
25.          else
26.              alarm.setStatus("CLEAR");
27.      }
28.  }
```

Berdasarkan kode translasi SNMP Trap To Alarm, maka diagram alirnya adalah sebagai berikut :



Gambar 3.61 Diagram Alir *Traslate SNMP Trap To Alarm*

Dari diagram alir *Traslate SNMP Trap To Alarm*, dapat dihitung *cyclomatic complexity*, yakni :

V(G) = Jumlah Busur – Jumlah Simpul + 2

V(G) = 39 – 29 + 2 = 12

V(G) = Predicate node + 1

V(G) = 11 + 1 = 12

Dari hasil perhitungan *cyclomatic complexity* terdapat *independent path*, yaitu :

Path 1 = 1-2-3-4-6-8-22-27-28-29

Path 2 = 1-2-3-4-5-27-28-29

Path 3 = 1-2-3-4-6-7-27-28-29

Path 4 = 1-2-3-4-6-8-9-10-11-27-28-29

Path 5 = 1-2-3-4-6-8-9-10-12-13-27-28-29

Path 6 = 1-2-3-4-6-8-9-10-12-14-15-27-28-29

Path 7 = 1-2-3-4-6-8-9-10-12-14-16-17-27-28-29

Path 8 = 1-2-3-4-6-8-9-10-12-14-16-18-19-27-28-29

Path 9 = 1-2-3-4-6-8-9-10-12-14-16-18-20-21-27-28-29

Path 10 = 1-2-3-4-6-8-22-23-24-27-28-29

Path 11 = 1-2-3-4-6-8-22-23-25-26-27-28-29

Path 12 = 1-2-28-29

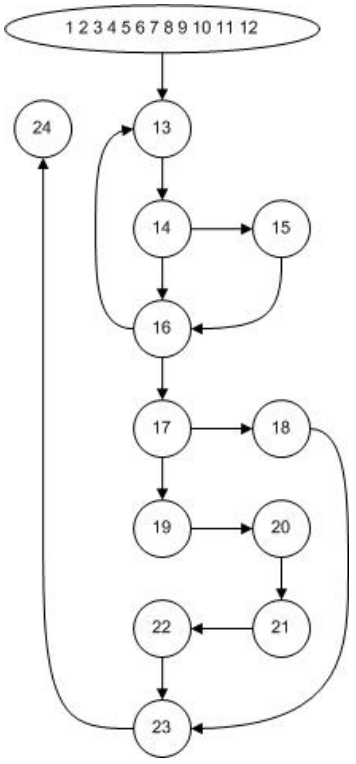
3. Pengujian White Box Email Notification

Berikut merupakan kode untuk melakukan pengiriman pesan notifikasi email.

```
1.    subject = alarm.getSeverity() + " - " +
      alarm.getHost().getSite() + " - " +
      alarm.getDescription();
2.    message = "Telah terjadi Alarm
      "+alarm.getDescription()+" pada :\n\n";
3.    message+= "Tanggal/Jam\t\t: " +
      alarm.getdTime().toString()+"\n";
4.    message+= "Host\t\t\t: " + alarm.getHost()+"\n";
5.    message+= "IP Address\t\t: " +
      alarm.getHost().getIp()+"\n";
6.    message+= "Site\t\t\t: " +
      alarm.getHost().getSite()+"\n";
```

```
7.    message+= "Customer\t\t: " +
      alarm.getHost().getCustomer()+"\n\n";
8.    message+= "-----------------------------------------
      -----\n";
9.    message+= "Dikirim oleh HPS Trouble Ticket System\n";
10.   message+= "-----------------------------------------
      -----\n";
11.   to = "";
12.   List<User> users =
      userDao.findBySiteId(alarm.getHost().getSite().getId()
      );
13.   for(User user : users) {
14.       if(user.getEmail() != null)
15.           to += user.getEmail() + ",";
16.   }
17.   if(to.equals(""))
18.       logger.warn("User is not exists.");
19.   else {
20.       to = to.substring(0, to.length()-1);
21.       mailer.sendMail(to, subject, message);
22.   }
23.   reportDao.save(alarm);
24.   subject = alarm.getSeverity() + " - " +
      alarm.getHost().getSite() + " - " +
      alarm.getDescription();
```

Berdasarkan kode *email notification*, maka diagram alirnya adalah sebagai berikut :



Gambar 3.62 Diagram Alir Notifikasi Email

Dari diagram alir notifikasi email, dapat dihitung *cyclomatic complexity*, yakni :

V(G) = Jumlah Busur – Jumlah Simpul + 2

V(G) = 15 – 13 + 2 = 4

V(G) = Predicate node + 1

V(G) = 3 + 1 = 4

Dari hasil perhitungan *cyclomatic complexity* terdapat *independent path*, yaitu :

Path 1 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-16-17-19-20-21-22-23-24

Path 2 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-16-17-18-23-24

Path 3 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-19-20-21-22-23-24

Path 4 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-23-24

3.7.2 Pengujian *Black Box*

Sub bab berikut ini akan membahas tentang pengujian aplikasi dengan menggunakan metode *black box* yang terdiri dari pengujian *alpha* dan beta.

3.7.2.1 Pengujian *Alpha*

Pengujian *alpha* merupakan pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Pihak pembangun mendampingi serta mencatat kesalahan-kesalahan maupun permasalahan yang dirasakan oleh pengguna. Pengujian *alpha* yang dilakukan pada aplikasi alarm handler untuk keperluan trouble ticket power system dibagi tiga bagian, yaitu rencana pengujian, kasus dan hasil pengujian dan kesimpulan pengujian.

3.7.2.2 Rencana Pengujian

Rencana pengujian yang akan dilakukan pada aplikasi ini selengkapnya terlihat pada tabel 3.30 di bawah ini.

Tabel 3.30 Rencana Pengujian

Kelas Uji	Detail Pengujian	Jenis Pengujian
Menjalankan Aplikasi Alarm Handler	Menjalankan service aplikasi SNMP trap receiver	*Black box*
Memproses Pesan Alarm	Menampilkan pesan alarm yang diterima ke layar	*Black box*
	Mengirimkan pesan notifikasi email ke teknisi lokal	*Black box*
	Menyimpan pesan alarm ke database	*Black box*
Login	Melakukan login ke aplikasi	*Black box*
Logout	Melakukan logout dari aplikasi	*Black box*
Tambah Lokasi	Memasukan data lokasi baru	*Black box*
Ubah Lokasi	Mengubah data lokasi	*Black box*
Hapus Lokasi	Menghapus data lokasi	*Black box*
Tambah HPS	Memasukan data HPS baru	*Black box*
Ubah HPS	Mengubah data HPS	*Black box*
Hapus HPS	Menghapus data HPS	*Black box*
Tambah Pengguna	Membuat pengguna baru	*Black box*
Ubah Pengguna	Mengubah data pengguna	*Black box*
Hapus Pengguna	Menghapus data pengguna	*Black box*
Menampilkan Laporan	Menampilkan laporan gangguan	*Black box*
Menghentikan Aplikasi Alarm Handler	Mematikan service SNMP trap	*Black box*

3.7.2.3 Kasus dan Hasil Pengujian Alpha

Kasus dan hasil pengujian berisi pemaparan dari rencana pengujian yang telah disusun pada skenario pengujian. Pengujian ini dilakukan secara *black box* dengan hanya memperhatikan masukan ke dalam sistem dan keluaran dari masukan tersebut.

1. Menjalankan Aplikasi Alarm Handler

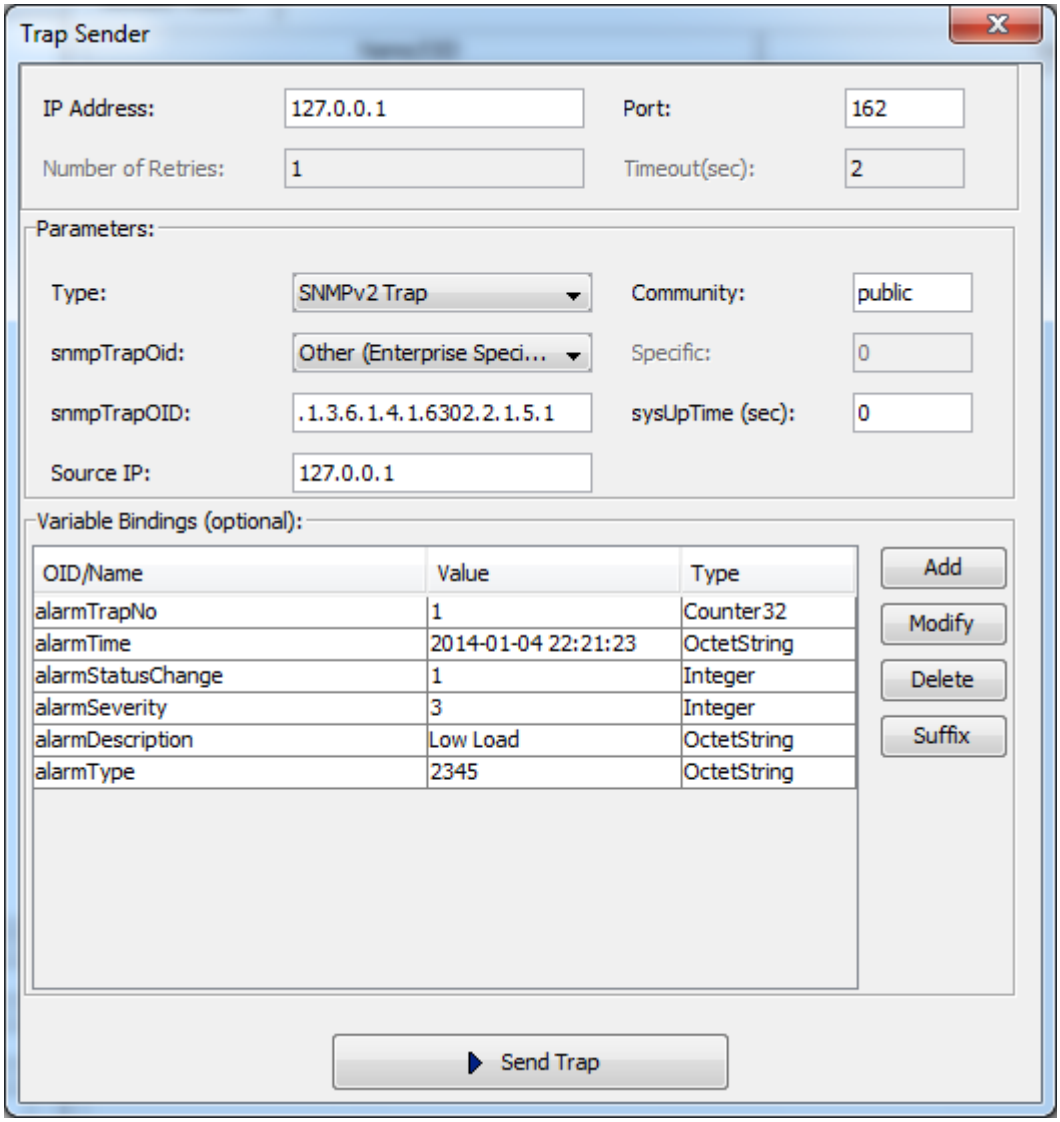
   Pengujian ini dilakukan untuk menjalankan aplikasi service *SNMP trap receiver* yang telah dibuat agar pesan-pesan *SNMP trap* yang dikirim oleh perangkat HPS dapat diproses.

   Tabel 3.31 Kasus dan Hasil Uji Menjalankan Aplikasi Alarm Handler

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tombol "Start"	Ketika menekan tombol "Start" aplikasi menjalankan service SNMP trap dan melakukan koneksi database sampai terjadi perubahan status, semula "Alarm Handler Stopped" menjadi "Alarm Handler Started"	Pesan "Alarm Handler Started" ditampilkan.	Diterima

2. Memproses Pesan Alarm

   Pengujian ini dilakukan untuk mengirimkan pesan alarm berupa pesan SNMP trap dari perangkat HPS ke server aplikasi, sehingga pesan tersebut dapat ditampilkan ke layar aplikasi dan dikirimkan sebagai pesan notifikasi email ke para teknisi serta disimpan ke database. Untuk kasus pengujian pengiriman pesan SNMP menggunakan tools bernama iReasoning MIB browser, dengan mengisi parameter-parameter masukan sebagai informasi bagi aplikasi alarm handler. Untuk lebih jelasnya dapat dilihat pada gambar 3.63 di bawah ini



Gambar 3.63 Simulasi Pengiriman *SNMP Trap*

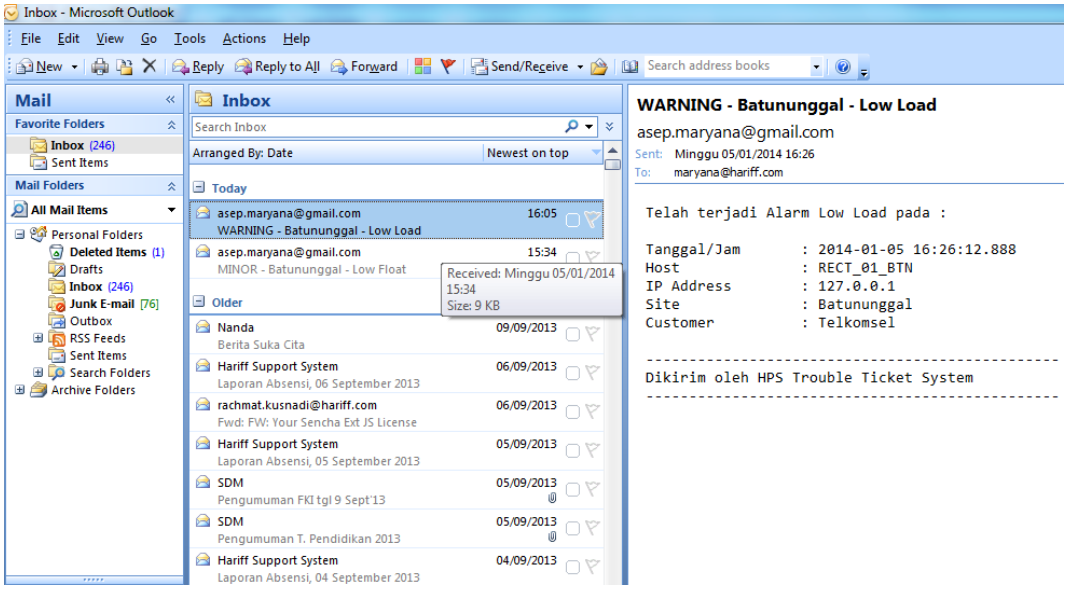
Berdasarkan gambar 3.63 diatas, maka kasus dan hasil uji pengiriman pesan trap adalah sebagai berikut :

Tabel 3.32 Kasus dan Hasil Uji Memproses Pesan Alarm

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
IP address: 127.0.0.1, port:162, Type: SNMPv2 trap, snmpTrapOID:	Pesan dapat diterima oleh aplikasi alarm handler, dapat	Teknisi mendapat pesan notifikasi	Diterima

.1.3.6.1.4.1.6302.2.1.5.1, Source IP: 127.0.0.1, Community: public, alarmTrapNo: 1, alarmTime: 2014-01-05 16:26:12, alarmStatusChange: 1, alarmSeverity: 3, alarmDescription: Low Load, alarmType: 2345	ditampilkan ke layar, dapat diteruskan ke teknisi sebagai pesan notifikasi serta dapat disimpan ke database	email.	

Pesan notifikasi yang diterima oleh teknisi dapat di lihat pada gambar 3.33 di bawah ini.



Gambar 3.64 Pesan Notifikasi Email

3. Login

Pengujian ini dilakukan untuk melakukan login ke aplikasi, sehingga pengguna dapat melakukan pengelolaan data-data yang digunakan dalam aplikasi.

Tabel 3.33 Kasus dan Hasil Uji Login

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Username:	Pengguna dapat	Dapat	Diterima

admin	memasukan username	memasukan username pada textbox	
Password: admin	Pengguna dapat memasukan password	Dapat memasukan password pada textbox	Diterima
Klik “Login”	User dapat melakukan login dengan benar ke aplikasi	Kotak login ditutup dan menu “management” menjadi aktif	Diterima

4. Logout

Pengujian ini dilakukan untuk mengakhiri sesi login ke aplikasi.

Tabel 3.34 Kasus dan Hasil Uji Logout

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tombol “Logout”	Pengguna dapat logout dari aplikasi	Menu “management” tidak diaktifkan	Diterima

5. Tambah Lokasi

Pengujian ini dilakukan untuk memasukan data lokasi agar dapat disimpan ke database.

Tabel 3.35 Kasus dan Hasil Uji Tambah Lokasi

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama lokasi: Moh Toha	Pengguna dapat memasukan nama lokasi	Dapat memasukan nama lokasi ke textbox	Diterima
Alamat: Bandung	Pengguna dapat memasukan alamat lokasi	Dapat memasukan alamat ke textbox	Diterima
Klik “Save”	Data dapat	Data berhasil	Diterima

	dimasukan ke database	disimpan ke database	

6. Ubah Lokasi

Pengujian ini dilakukan untuk melakukan perubahan data lokasi yang sebelumnya sudah tersimpan di database.

Tabel 3.36 Kasus dan Hasil Uji Ubah Lokasi

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama lokasi: Batununggal	Pengguna dapat memasukan nama lokasi	Dapat memasukan nama lokasi ke textbox	Diterima
Alamat: Bandung	Pengguna dapat memasukan alamat lokasi	Dapat memasukan alamat ke textbox	Diterima
Klik “Save”	Data dapat dimasukan ke database	Perubahan data berhasil disimpan ke database	Diterima

7. Hapus Lokasi

Pengujian ini dilakukan untuk melakukan penghapusan data lokasi yang tersimpan di database.

Tabel 3.37 Kasus dan Hasil Uji Hapus Lokasi

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama lokasi: Batununggal	Pengguna dapat memilih record lokasi	Record lokasi yang dipilih diberi warna	Diterima
Klik “Delete”	Pengguna dapat mengahapus data lokasi yang dipilih	Lokasi yang dipilih tidak muncul di database	Diterima

8. Tambah HPS

Pengujian ini dilakukan untuk memasukan data perangkat HPS ke database.

Tabel 3.38 Kasus dan Hasil uji Tambah HPS

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama: RECT_01_MTH	Pengguna dapat memasukan nama perangkat HPS	Dapat memasukan nama perangkat ke textbox	Diterima
IP Address: 10.12.12.21	Pengguna dapat memasukan IP address	Dapat memasukan IP address ke textbox	Diterima
Lokasi: Moh Toha	Pengguna dapat memilih lokasi	Dapat memilih lokasi	Diterima
Customer: Telkomsel	Pengguna dapat memilih pelanggan	Dapat memilih customer	Diterima
Klik “Save”	Data dapat dimasukan ke database	Data berhasil disimpan ke database	Diterima

9. Ubah HPS

Pengujian ini dilakukan untuk melakukan perubahan data perangkat HPS yang sebelumnya sudah ada di database.

Tabel 3.39 Kasus dan Hasil Uji Ubah HPS

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama: RECT_01_BTN	Pengguna dapat memasukan nama perangkat HPS	Dapat memasukan nama perangkat ke textbox	Diterima
IP Address: 10.12.12.22	Pengguna dapat memasukan IP address	Dapat memasukan IP address ke textbox	Diterima
Lokasi: Moh	Pengguna dapat	Dapat memilih	Diterima

Toha	memilih lokasi	lokasi	
Customer: Telkomsel	Pengguna dapat memilih pelanggan	Dapat memilih customer	Diterima
Klik “Save”	Data perubahan dapat disimpan ke database	Data yang dipilih berhasil diubah	Diterima

10. Hapus HPS

Pengujian ini dilakukan untuk melakukan penghapusan data perangkat HPS.

Tabel 3.40 Kasus dan Hasil Uji Hapus HPS

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama HPS: RECT_01_BTN	Pengguna dapat memilih record HPS	Record HPS yang dipilih diberi warna	Diterima
Klik “Delete”	Pengguna dapat mengahapus data HPS yang dipilih	HPS yang dipilih tidak muncul di database	Diterima

11. Tambah Pengguna

Pengujian ini dilakukan untuk membuat pengguna aplikasi yang baru.

Tabel 3.41 Kasus dan Hasil Uji Tambah Pengguna

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Name: asep	Pengguna dapat memasukan nama pengguna	Dapat memasukan nama pengguna ke textbox	Diterima
Phone: 085795399103	Pengguna dapat memasukan nomor handphone	Dapat memasukan nomor handphone ke textbox	Diterima
Email:	Pengguna dapat	Dapat	Diterima

maryana@hariff.com	memasukan alamat email ke textbox	memasukan alamat email ke textbox	
Username: asep	Pengguna dapat memasukan username ke textbox	Dapat memasukan nama pengguna ke textbox	Diterima
Password: hdte450	Pengguna dapat memasukan password ke textbox	Dapat memasukan password ke textbox	Diterima
Lokasi: batununggal	Pengguna dapat memilih lokasi pada combobox	Dapat memilih lokasi pada combobox	Diterima
Level: koordinator	Pengguna dapat memilih jenis user pada combobox	Dapat memilih level pada combobox	Diterima
Klik “Save”	Data dapat dimasukan ke database	Data berhasil disimpan ke database	Diterima

12. Ubah Pengguna

Pengujian ini dilakukan untuk melakukan perubahan data pengguna.

Tabel 3.42 Kasus dan Hasil Uji Ubah Pengguna

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Name: asep maryana	Pengguna dapat memasukan nama pengguna	Dapat memasukan nama pengguna ke textbox	Diterima
Phone: 085795399103	Pengguna dapat memasukan nomor handphone	Dapat memasukan nomor handphone ke textbox	Diterima
Email: maryana@hariff.com	Pengguna dapat memasukan alamat email ke textbox	Dapat memasukan alamat email ke textbox	Diterima

Username: asep	Pengguna dapat memasukan username ke textbox	Dapat memasukan nama pengguna ke textbox	Diterima
Password: hdte450	Pengguna dapat memasukan password ke textbox	Dapat memasukan password ke textbox	Diterima
Lokasi: batununggal	Pengguna dapat memilih lokasi pada combobox	Dapat memilih lokasi pada combobox	Diterima
Level: koordinator	Pengguna dapat memilih jenis user pada combobox	Dapat memilih level pada combobox	Diterima
Klik “Save”	Data dapat dimasukan ke database	Data berhasil disimpan ke database	Diterima

13. Hapus Pengguna

Pengujian ini dilakukan untuk melakukan penghapusan pengguna.

Tabel 3.43 Kasus dan Hasil Uji Hapus Pengguna

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Name: asep maryana	Pengguna dapat memilih record user	Record user yang dipilih diberi warna	Diterima
Klik “Delete”	Pengguna dapat mengahapus data user yang dipilih	user yang dipilih tidak muncul di database	Diterima

14. Menampilkan Laporan

Pengujian ini dilakukan untuk menampilkan laporan gangguan berdasarkan lokasi tertentu dan periode tanggal tertentu.

Tabel 3.44 Kasus dan Hasil Uji Menampilkan Laporan

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Lokasi: Batununggal	Pengguna dapat memilih combobox lokasi	Combobox lokasi dapat dipilih	Diterima
Dari: 1 Des 2013	Pengguna dapat memasukan tanggal awal laporan	Tanggal awal laporan dimasukan pada textbox	Diterima
Dari: 31 Des 2013	Pengguna dapat memasukan tanggal akhir laporan	Tanggal akhir laporan dimasukan pada textbox	Diterima
Klik "View"	Pengguna dapat menampilkan daftar riwayat kejadian gangguan	Daftar gangguan ditampilkan	Diterima

15. Menghentikan Aplikasi Alarm Handler

Pengujian ini dilakukan untuk menghentikan aplikasi service alarm handler.

Tabel 3.45 Kasus dan Hasil Uji Menghentikan Aplikasi Alarm Handler

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tombol "Stop"	Ketika menekan tombol "Stop" aplikasi menghentikan service SNMP trap sampai terjadi perubahan status, semula "Alarm Handler Started" menjadi "Alarm Handler Stopped"	Pesan "Alarm Handler Stopped" ditampilkan.	Diterima

3.7.2.4 Kesimpulan Pengujian Alpha

Berdasarkan hasil pengujian aplikasi yang dilakukan maka dapat disimpulkan bahwa aplikasi alarm handler untuk keperluan trouble ticket power sistem yang dibangun sudah berjalan sesuai dengan yang diharapkan.

3.7.2.5 Pengujian Beta

Pengujian beta dilakukan di lingkungan pengguna tanpa kehadiran pihak pembangun aplikasi. Pengujian ini merupakan pengujian yang bersifat langsung di lingkungan yang sebenarnya. Pengguna melakukan penilaian terhadap aplikasi dengan menggunakan kuesioner. Dari hasil kuesioner tersebut maka dapat ditarik kesimpulan apakah aplikasi yang dibangun telah sesuai dengan tujuan atau tidak.

Pengujian beta dilakukan terhadap pengguna dengan menggunakan kuesioner. Untuk mengetahui tanggapan dan penilaian pengguna terhadap aplikasi ini, telah disebarkan kuesioner kepada 10 responden. Kuesioner ini disebarkan menggunakan teknik sampling yaitu *Simple Random Sampling* yang disebarkan kepada banyak pengguna. Dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan aplikasi yang dibangun. Berikut adalah pertanyaan dan hasil kuesioner yang telah dibagikan dengan menggunakan rumus :

$$Y = \frac{P}{Q} \text{ x } 100\ \%$$

Keterangan : P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

Y = Nilai presentase

Berikut data yang didapat berdasarkan kuesioner pada bagian pertama :

1. 100% (10 responden) adalah laki-laki.
2. 0% (0 responden) adalah wanita.

Berikut ini adalah hasil presentase masing-masing nilai jawaban kuesioner yang diujikan kepada 10 responden dan telah dihitung dengan rumus.

1. Apakah dengan adanya aplikasi alarm handler dapat membantu dalam melakukan pemantauan perangkat HPS yang berada di lokasi ?

Tabel 3.46 Jawaban Kuesioner untuk Pertanyaan Nomor 1

Jawaban	Responden	Presentase (%)
Sangat Membantu	4	40%
Membantu	5	50%
Ragu-ragu	1	10%
Total Responden	10	100%

2. Apakah aplikasi alarm handler mempermudah dalam melakukan analisa perbaikan dan pemeliharaan ?

Tabel 3.47 Jawaban Kuesioner untuk Pertanyaan Nomor 2

Jawaban	Responden	Presentase (%)
Sangat Mempermudah	3	30%
Mempermudah	5	50%
Ragu-ragu	2	20%
Total Responden	10	100%

3. Apakah informasi gangguan yang dikirim oleh aplikasi alarm handler cukup realtime ?

Tabel 3.48 Jawaban Kuesioner untuk Pertanyaan Nomor 3

Jawaban	Responden	Presentase (%)
Sangat Realtime	5	50%
Realtime	3	30%
Ragu-ragu	1	10%
Total Responden	10	100%

4. Apakah aplikasi alarm handler membebani traffic email internal hariff ?

Tabel 3.49 Jawaban Kuesioner untuk Pertanyaan Nomor 4

Jawaban	Responden	Presentase (%)
Sangat Mengganggu	1	10%
Mengganggu	2	20%
Tidak Mengganggu	7	70%
Total Responden	10	100%

5. Apakah laporan gangguan yang terdapat dalam aplikasi alarm handler cukup bermanfaat ?

Tabel 3.50 Jawaban Kuesioner untuk Pertanyaan Nomor 5

Jawaban	Responden	Presentase (%)
Sangat Bermanfaat	2	20%
Bermanfaat	6	60%
Tidak Bermanfaat	2	20%
Total Responden	10	100%

Dari hasil jawaban responden dapat disimpulkan bahwa aplikasi yang dibuat sangat membantu dan mempermudah para teknisi dalam melakukan pemantauan perangkat HPS yang berada di lokasi, karena informasi gangguan yang diterima oleh para teknisi bersifat *realtime* sehingga proses penanganan gangguan dapat dilakukan dengan segera.