BAB III

PEMBAHASAN

3.1 Analisis Masalah

Analisis adalah penguraian dari suatu masalah atau objek yang akhirnya menghasilkan suatu kesimpulan, hal ini dimaksudkan untuk mengidentifikasi dan mengevaluasi masalah-masalah atau objek.

Tahap analisis dilakukan setelah tahap perencanaan sistem dan sebelum tahap desain sistem. Tahap ini merupakan tahap yang kritis karena kesalahan dalam tahap ini menyebabkan kesalahan pada tahap selanjutnya, misalnya perangkat lunak dihadapkan pada suatu masalah untuk menentukan seberapa jauh sistem tersebut telah mencapai sasarannya, jika sistem mempunyai beberapa kelemahan, perangkat lunak harus dapat menemukannya, sehingga tugas ini disebut sebagai analisis sistem.

Tugas utama dari proses analisis sistem meliputi kegiatan:

1. menentukan lingkup sistem.
2. mengumpulkan fakta.
3. menganalisis fakta.
4. mengomunikasikan temuan-temuan tersebut melalui laporan analisis sistem.

Fakta merupakan bagian dari in*form*asi yang menunjukkan realita, situasi, dan relasi yang menjamin analisis dan pemodelan.

Langkah-langkah dasar yang dilakukan dalam analisis sistem adalah :

1. *Understand*, memahami kerja sistem yang ada

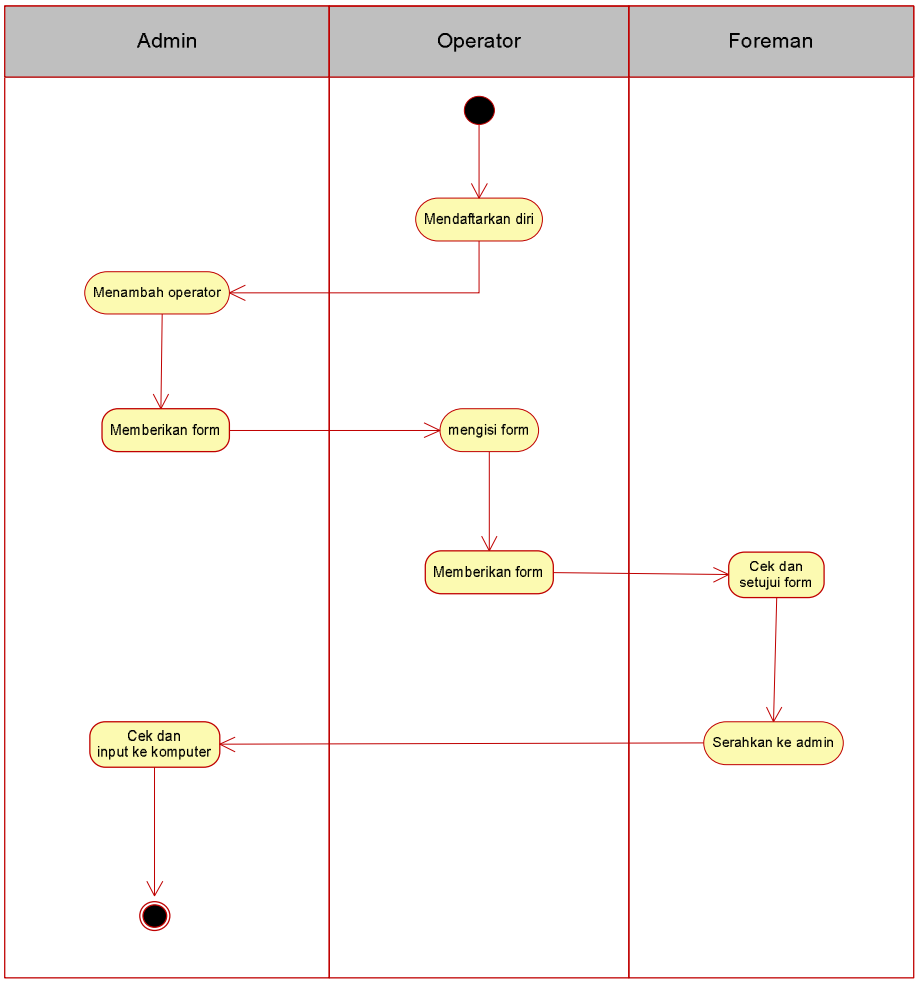
2. *Identify*, mengindentifikasikan masalah
3. *Analyze*, menganalisis sistem
4. *Report*, membuat laporan

3.1.1 Prosedur Sistem yang sedang Berjalan

Selama ini kegiatan *maintenance* di P.T. SKFIndonesia, menggunakan media kertas sebagai *form* pengawasan. *Form* pengawasan berisi semua kegiatan yang dilakukan untuk *maintenance* mesin. *Form* pengawasan diisi operator dan disetujui oleh foreman yang kemudian dikirim ke bagian manajer setiap minggunya. Berikut akan dijelaskan secara rinci prosedur pengawasan kegiatan *maintenance.*

1. *Operator* diberikan *form* pengawasan sesuai dengan *chanel* dan mesin tempat mereka ditugaskan.
2. *Operator* mengisi *form* kegiatan *maintenance* mesin. Sesuai, dengan kegiatan yang telah mereka kerjakan. Setelah selesai *form* tersebut diisi, dikumpulkan ke foreman untuk ditandatangani .
3. *Foreman* kemudian menandatangani dan memberikan *form* tersebut kepada *supervisor.*
4. *Supervisor* kemudian mengecek dan menyimpan *form* tersebut ke dalam komputer.

Analisis prosedur yang berjalan dapat dijabarkan dengan menggunakan *activity diagram* pada gambar 3.1.



Gambar 3.1 Activity diagram sistem yang sedang berjalan

3.1.2 Analisis Dokumen

Berdasarkan *use case* di atas, terdapat beberapa dokumen yang ada pada prosedur kegiatan *maintenance* mesin, diantaranya adalah sebagai berikut :

1. Dokumen user pengguna sistem.
2. Dokumen mesin dan kegiatan *maintenance* yang dilakukan.

3.1.3 Lingkup Masalah

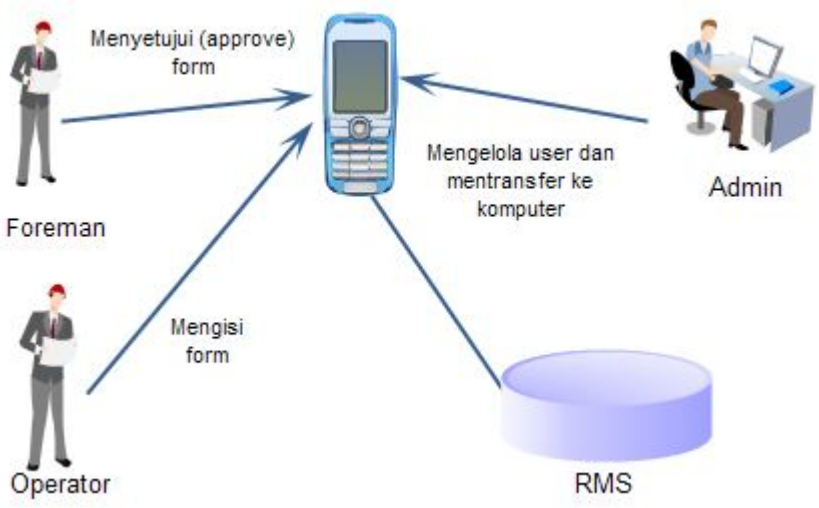
Dari penjabaran sistem diatas, dapat disimpulkan perangkat lunak yang akan dibangun, berfungsi untuk mengelola kegiatan *maintenance* mesin, yang meliputi pendaftaran operator, pemberian *form*, penyetujuan *form* oleh *foreman* dan input *form* ke komputer.

3.2 Deskripsi Global Perancangan Perangkat Lunak

Deskripsi global perangkat lunak membahas secara umum perangkat lunak yang akan dibangun dengan mendeskripsikan perspektif dan fungsi dari perangkat lunak, karakteristik untuk pengguna perangkat lunak, batasan-batasan perangkat lunak yang akan dibangun, serta asumsi dan ketergantungan.

3.2.1 Perspektif Produk

Produk yang akan dibuat merupakan simulasi perangkat lunak yang digunakan untuk diimplementasikan pada media handphone. Perangkat lunak ini, berfungsi sebagai pengganti *form maintenance*. File disimpan di dalam media penyimpanan RMS (Record Management System) untuk nantinya disetujui *foreman*. *Output* yang dihasilkan berupa file yang nantinya dapat ditransfer oleh admin ke dalam komputer. Admin adalah orang yang menjabat sebagai *supervisor*. Sebelum menggunakan aplikasi ini operator harus login terlebih dahulu. Jika belum terdaftar operator wajib mendaftar kepada admin. Gambar 3.3 menjelaskan tentang perspektif produk dari perangkat lunak yang akan dibangun.



Gambar 3.2 Perspektif produk

Seperti terlihat pada gambar 3.2 di atas, Aplikasi yang dibangun, menggunakan media handphone. Semua data disimpan pada media penyimpanan handphone yaitu RMS (*Record Management System*). Admin bertugas untuk mentransfer file yang tersimpan di RMS ke dalam komputer.

3.2.2 Fungsi Produk

Secara umum aplikasi *paperless operator maintenance* ini berfungsi sebagai pengganti *form* operator *maintenance* yang sebelumnya menggunakan media kertas. Namun, pada penerapannya dibutuhkan fungsi-fungsi tambahan yang berguna sebagai penunjang fungsi utama. Secara garis besar fungsi perangkat lunak tersebut yaitu:

1. Autentikasi pengguna
2. Menyediakan *form* kegiatan *maintenance* sesuai dengan mesin tempat operator ditugaskan.
3. Menyimpan data kegiatan *maintenance*
4. Mengelola data pengguna.

3.2.3 Karakteristik Pengguna

Perangkat lunak ini akan digunakan oleh tiga pengguna, yaitu *operator, foreman* dan admin.

Tabel 3.1 Karakteristik pengguna

Pengguna	**Tanggung Jawab**	**Hak Akses**	**Tingkat Pendidikan**	**Tingkat Keterampilan**	**Pengalaman**
Operator	Mengisi kegiatan *maintenance*	Menjalankan program sesuai dengan tugasnya	SMA atau sederajat	Mengerti cara penggunaan *handphone*	Menggunakan handphone yang menggunakan sistem operasi symbian.
Foreman	Menyetujui	Menjalankan	S1	Mengerti cara	Menggunakan

	(*approve*) *form* yang telah disimpan operator	program sesuai dengan tugasnya		penggunaan handphone	handphone yang menggunakan sistem operasi symbian.
Admin	Mengelola user, mentransfer file yang telah disetujui foreman ke dalam komputer	Menjalankan program sesuai dengan tugasnya	S1	Mengerti cara penggunaan *handphone*. Menggunakan aplikasi selectQ,	Menggunakan handphone yang menggunakan sistem operasi symbian danpernah menggunakan aplikasi selectQ.

3.2.4 Perancangan Proses

Perancangan proses adalah suatu tahapan lebih lanjut dari tahapan analisis sistem yang merupakan suatu tahap persiapan dalam merancang sistem yang baru. Di dalam perancangan proses ini, akan dijelaskan lebih detail tentang perubahan-perubahan yang dilakukan terhadap perangkat lunak yang lama, sehingga perangkat lunak yang baru yang diusulkan dapat lebih menjamin ketepatan in*form*asi, dan mengurangi permasalahan yang terjadi pada sistem yang lama.

Pada tahap perancangan proses, secara umum komponen-komponen yang terkait terdiri dari :

1. Perancangan *use case diagram.*
2. Perancangan *activity diagram.*

3. Perancangan *sequence diagram*

4. Perancangan *class diagram*

3.2.5 Perancangan *Form* Perangkat Lunak

Perancangan *form* perangkat lunak ini terdiri dari 6 (enam) *form* utama, yaitu : *form* login, *form* mesin, *form* kegiatan, *form* simpan kegiatan mesin, *form* simpan kegiatan mesin *fix* dan *form* user.

1. *Form Login*

   Form login merupakan *form* awal untuk menjalankan perangkat lunak. Terdapat 5 data yang wajib diinputkan untuk validasi. Yaitu NPK, nama, password, chanel dan status user.

2. *Form* Mesin

   Form mesin adalah *form* yang berisi jenis mesin sesuai dengan *chanel* tempat operator bekerja. Ada dua *textfield* yang wajib diisi oleh operator yaitu nama mesin dan *shift* .

3. *Form* Kegiatan

   Form Kegiatan berisi kegiatan-kegiatan apa saja yang wajib dilakukan operator dalam kegiatan *maintenance* mesin.

4. *Form* Menu Foreman

   Form Menu Foreman berisi menu foreman yang terdiri menu untuk melihat data yang telah disimpan operator.

5. *Form* Simpan Kegiatan Mesin

   Form Simpan Kegiatan Mesin ini digunakan untuk menampilkan data kegiatan mesin yang telah disimpan operator. Jika foreman yang membuka *form* ini, terdapat fitur untuk

menyetujui (*approve*) data kegiatan mesin yang telah disimpan operator. Namun, jika admin yang membuka *form* ini tidak terdapat fitur tersebut.

6. *Form* Hapus User

Form Hapus merupakan *form* untuk menghapus user.

7. *Form add* User

Form add user digunakan untuk menghapus user.

3.2.6 Batasan-batasan

Aplikasi ini tidak menyediakan fitur untuk mentransfer file. Kegiatan transfer file ini dilakukan secara manual yaitu dengan membuka file system handphone menggunakan aplikasi pihak ketiga yaitu SelectQ.

3.2.7 Asumsi dan Ketergantungan

Aplikasi papperles operator maintenance ini diimplementasikan menggunakan handphone. Aplikasi ini digunakan untuk satu chanel mesin. Diasumsikan, proses pengambilan data yang telah disimpan di RMS diambil oleh admin dengan cara membuka file *system* menggunakan aplikasi pihak ketiga yaitu SelectQ.

3.3 Deskripsi Rinci Kebutuhan

Bagian ini berisi semua kebutuhan perangkat lunak yang diuraikan secara rinci untuk keperluan perancangan parangkat lunak. Kebutuhan perangkat lunak

tersebut meliputi kebutuhan antarmuka eksternal, kebutuhan fungsional, non fungsional, kebutuhan data, dan atribut perancangan.

3.3.1 Kebutuhan Antarmuka Eksternal

Bagian ini menjelaskan tentang antarmuka pemakai, antarmuka perangkat lunak, antarmuka perangkat keras, dan antarmuka komunikasi.

a. Antarmuka Pemakai

Pengguna utama aplikasi ini adalah operator, foreman dan admin. Admin merupakan orang yang menjabat sebagai *supervisor*. rancangan antarmuka pemakai harus sederhana dan mudah digunakan. Tampilan aplikasi akan dibangun dalam bentuk *form*. Navigasi menu dan pemasukan data menggunakan tombol handphone dengan menggunakan tombol bilangan 0 – 9 dan A – Z.

b. Antarmuka Perangkat Lunak

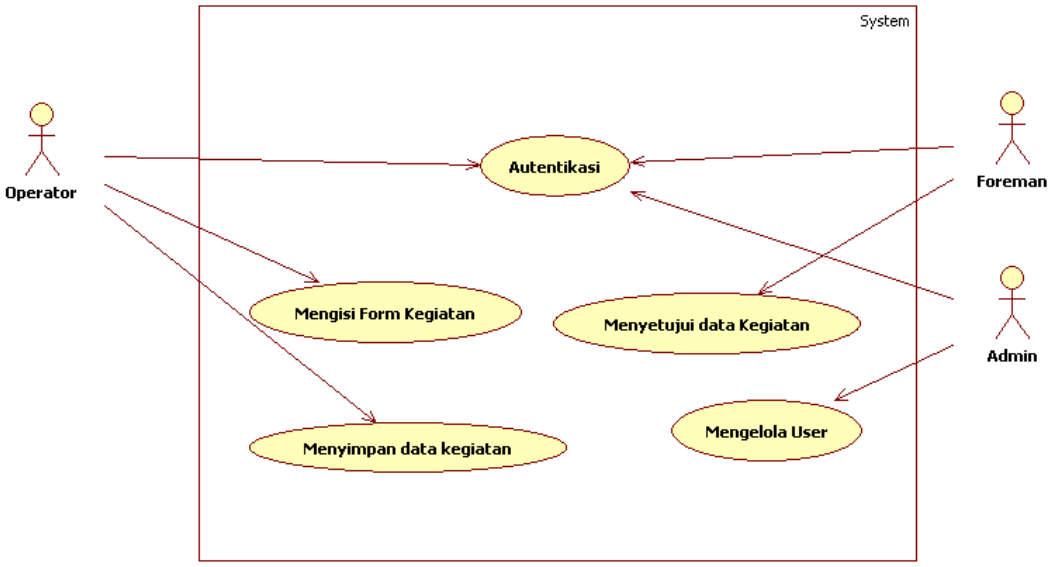
Aplikasi yang dibangun akan langsung berhubungan dengan datebase yang ada di handphone yaitu RMS (Record Management System). Bagian admin dapat mengambil database yang tersimpan secara langsung tanpa antarmuka perangkat lunak ini.

c. Antarmuka Perangkat Keras

Untuk dapat menjalankan perangkat lunak ini, membutuhkan handphone yang bersistem operasi Symbian atau Java.

3.3.2 Kebutuhan Fungsional

Perangkat lunak yang dibuat akan menangani 6 (tujuh) fungsi utama, seperti terlihat dalam *use case diagram* seperti pada gambar 3.3 dibawah ini.



Gambar 3.3 Use case diagram program

Use case diagram di atas menggambarkan apa saja yang dapat dilakukan *user* terhadap perangkat lunak yang dibuat.

1. Operator, dapat melakukan autentikasi, mengisi *form* kegiatan dan menyimpannya ke dalam database.
2. Foreman, dapat menyetujui data kegiatan.

3. Admin, dapat mengelola user dan melihat data yang telah disimpan operator.

3.3.2.1. Autentikasi

Fungsi ini melakukan autentikasi terhadap pengguna aplikasi. Selain demi tujuan keamanan, fungsi ini akan mengecek status user. Agar masing-masing user akan mendapatkan fitur sesuai dengan status yang disandangnya.

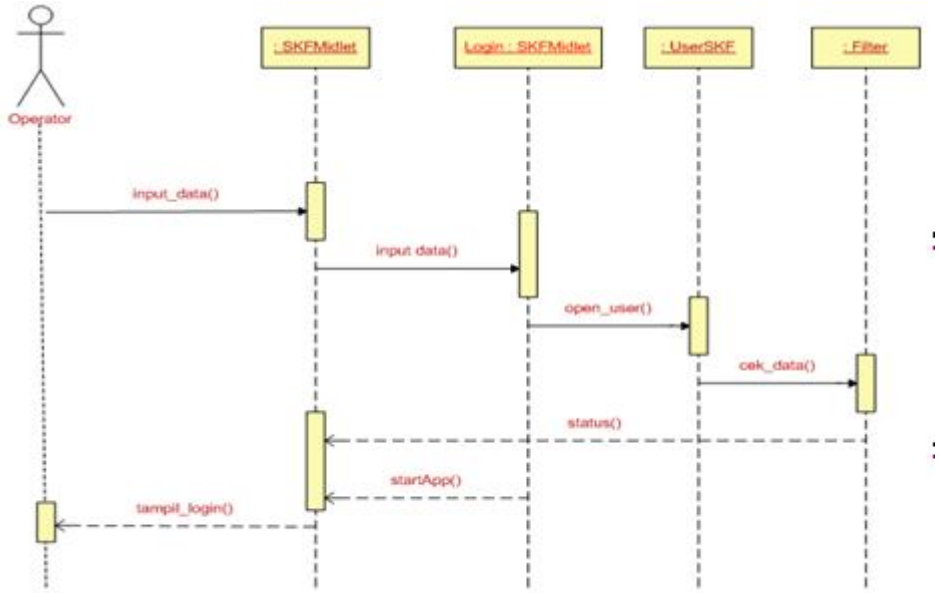
1. Skenario

Tabel 3.2 Skenario autentikasi

Identifikasi	
Nomor	1
Nama	Autentikasi pengguna
Tujuan	Memastikan hak akses masing-masing pengguna
Tipe	
Aktor	Operator, admin, *foreman*
Skenario utama	
Aksi Aktor	**Reaksi Sistem**
Pengguna menginputkan data ke dalam sistem	Sistem mengecek data dan status pengguna
	Sistem menampilkan *form* sesuai dengan

	hak akses pengguna
Skenario Kegagalan	
Aksi Aktor	**Reaksi Sistem**
Menginputkan data yang diperlukan untuk autentikasi	Menampilkan keterangan kegagalan bila salah
Menginputkan kembali data ke dalam sistem	
Kondisi akhir	Sistem menampilkan *form* sesuai dengan hak akses pengguna

2. Diagram sequence



Gambar 3.4 Diagram sequence autentikasi

3.3.2.2 Mengisi *Form* Kegiatan.

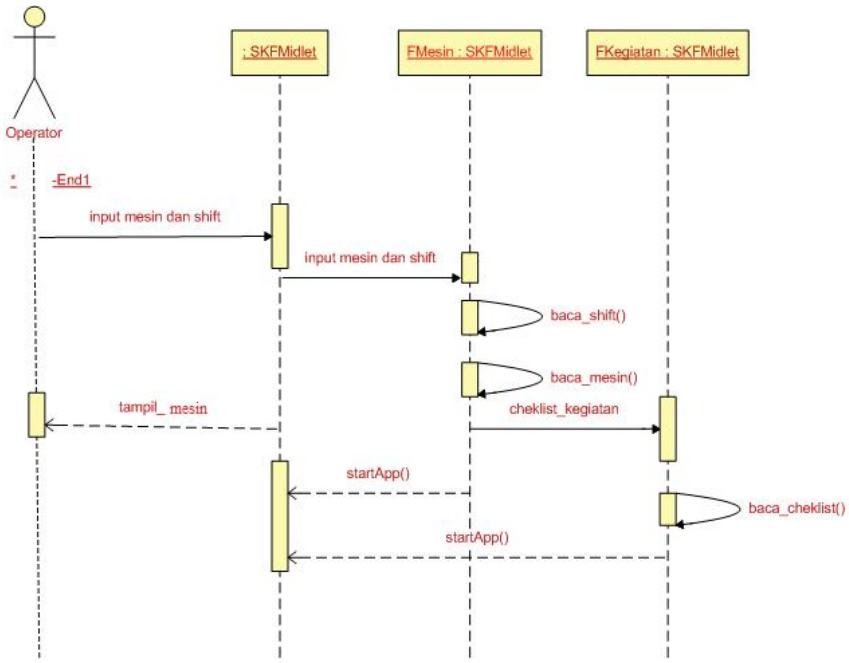
Setelah login operator sudah dapat menginputkan data kegiatan *maintenance* mesin, yang nantinya dapat disimpan ke dalam RMS.

1. Skenario

Tabel 3.3 Skenario mengisi form kegiatan

Identifikasi	
Nomor	2
Nama	Mengisi *form* kegiatan
Tujuan	Mengisi *form* kegiatan maintenance mesin
Tipe	
Aktor	Operator
Skenario Utama	
Kondisi awal	Operator sudah terautentikasi
Aksi Aktor	**Reaksi Sistem**
Aktor memilih mesin dan shift tempat mereka ditugaskan	Sistem menampilkan *form* yang berisi kegiatan maintenance mesin sesuai dengan mesin yang aktor pilih
Aktor menceklist data sesuai dengan kegiatan yang mereka kerjakan	

2. Sequence diagram



Gambar 3.5 Diagram sequence mengisi form kegiatan

3.3.2.3 Meyimpan Data Kegiatan

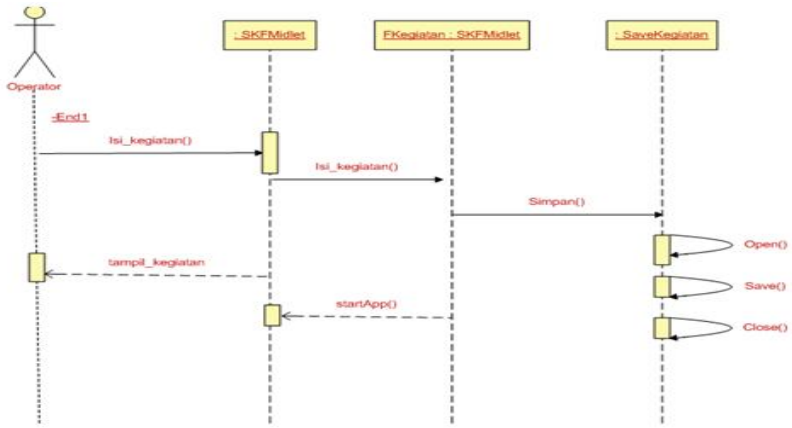
Setelah *form* kegiatan diisi, data tersebut disimpan di dalam RMS (Record Management System).

1. Skenario

Tabel 3.4 Skenario menyimpan *form* kegiatan

Identifikasi masalah	
Nomor	3
Nama	Menyimpan *Form* Kegiatan
Tujuan	Menyimpan data dari *form* yang telah terisi kedalam database
Tipe	
Aktor	Operator
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
Aktor menyimpan data yang telah diisinya	Sistem membuka *database*(RMS)
	Sistem mengambil string dari *form* kemudian menyimpannya ke dalam RMS

2. Sequence diagram



Gambar 3.6 Diagram sequence menyimpan form kegiatan

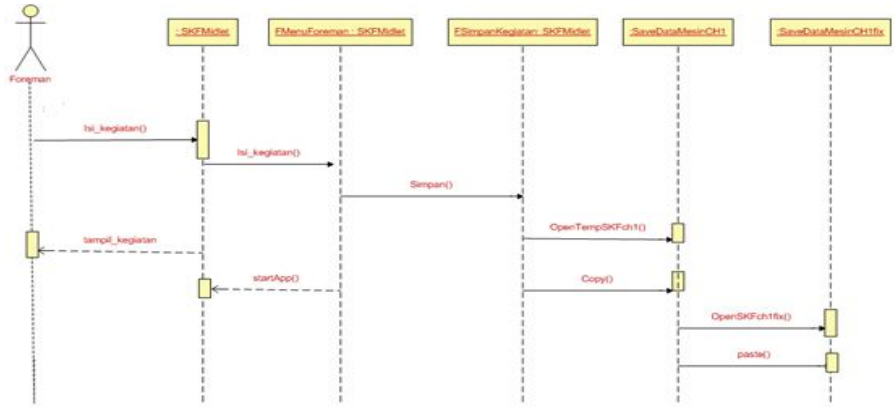
3.3.2.4 Menyetujui Data Kegiatan

1. Skenario

Tabel 3.5 Skenario menyetujui form kegiatan

Identifikasi Masalah	
Nomor	4
Nama	Menyetujui data Kegiatan
Tujuan	Menyetujui data kegiatan yang telah disimpan oleh operator
Tipe	
Aktor	Foreman
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
Memilih menu approve	Tambahkan tanggal dan nama user ke dalam file yang tersimpan.

2. Diagram sequence



Gambar 3.7 Diagram sequence menyetujui form kegiatan

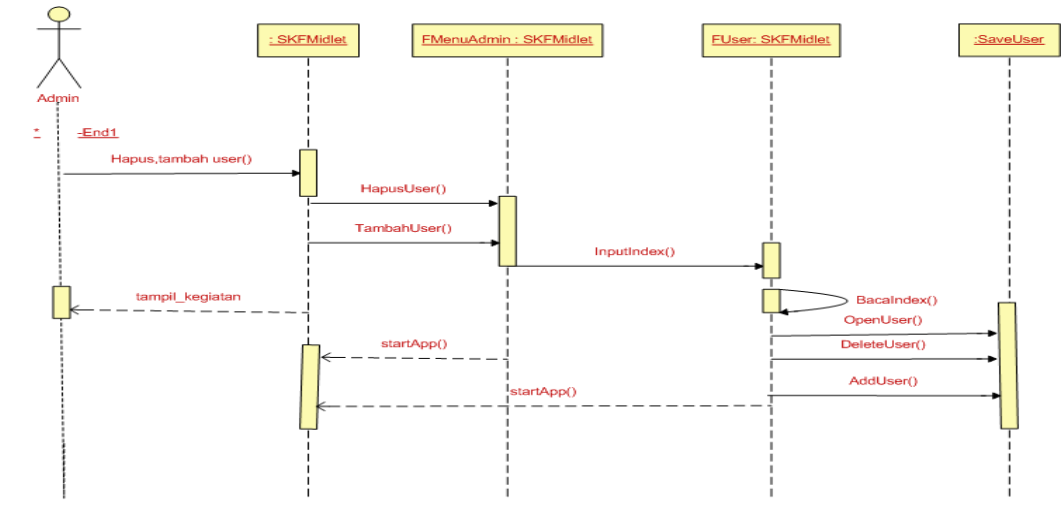
3.3.2.5 Mengelola user

1. Skenario

Tabel 3.6 Skenario mengelola user

Identifikasi Masalah	
Nomor	6
Nama	Mengelola User
Tujuan	Menghapus dan menambah user
Tipe	
Aktor	Admin
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
Menginputkan index	Menghapus user sesuai dengan index.
Mengisi data	Menambahkan data ke dalam *database* (RMS)

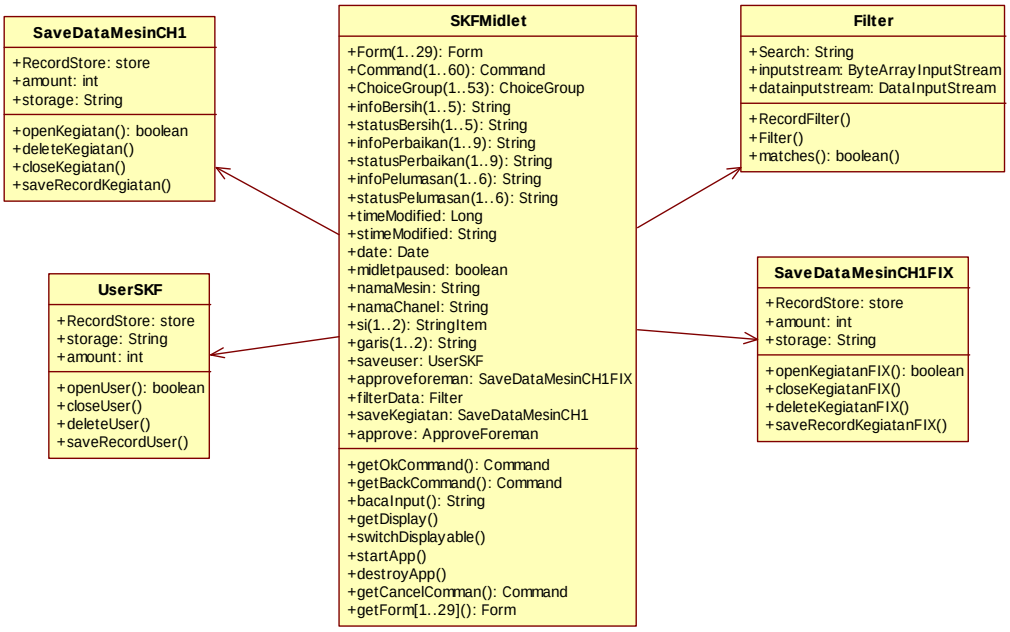
2. Diagram sequence



Gambar 3.8 Diagram sequence mengelola user

3.3.3 Kelas Diagram

Kelas diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan status yang terdapat di antara mereka.



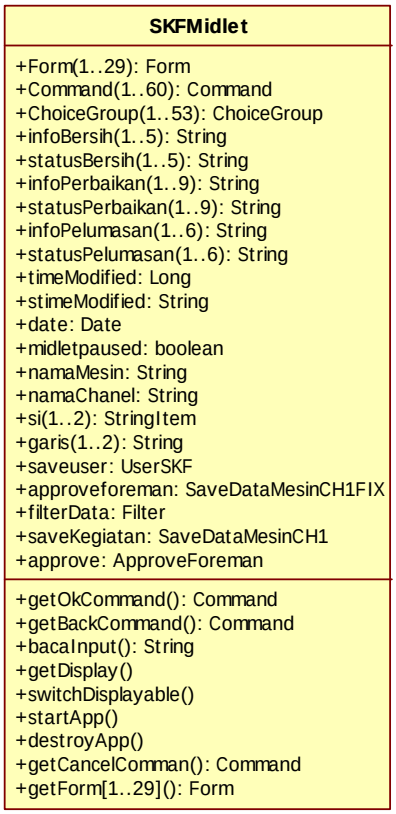
Gambar 3.9 Class diagram

3.3.3.1 Spesifikasi Kelas

Menjabarkan setiap kelas secara detail dengan mendeskripsikan atribut dan operasi yang dilakukan dari masing-masing kelas tersebut.

1. **Kelas SKFMidlet**

   Kelas SKFMidlet adalah kelas yang menjadi pusat dari interface dan fungsi-fungsi yang digunakan dalam aplikasi.



Gambar 3.10 Kelas SKFMidlet

a. Deskripsi Atribut

Mendeskripsikan setiap atribut dari kelas SKFMidlet dengan menentukan nama dan tipe atribut.

Tabel 3.7 Deskripsi atribut kelasSKFMidlet

Nama	Tipe
Form(1..29)	*Form*
Command(1..60)	Command
ChoiceGroup(1..53)	ChoiceGroup
InfoBersih(1..5)	String
StatusBersih(1..5)	String

InfoPelumasan(1..6)	String
StatusPelumasan(1..6)	String
InfoPerbaikan(1..9)	String
StatusPerbaikan(1..9)	String
timeModified	Long
stimeModified	String
date	Date
namaMesin	String
namaChanel	String
midletPaused	boolean
Si(1..2)	StringItem
garis	String
saveUser	UserSKF
approveForeman	SaveDataMesinCH1FIX
filterData	Filter
SaveKegiatan	SaveDataMesinCH1
Approve	ApproveForeman

b. Deskripsi Operasi

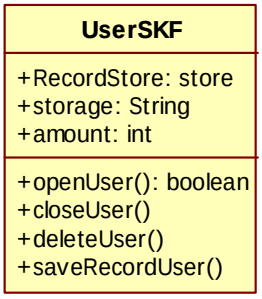
Mendeskripsikan setiap operasi dari kelas SKFMidlet, operasi-operasi yang ada dalam kelas ini adalah:

a. Method switchDisplayable() digunakan untuk membuat link ke *form* lainnya.

b. Method getOKCommand() digunakan untuk mengambil event dari command yang bertipe OK.
c. Method getCancelCommand() digunakan untuk mengambil event dari command yang bertipe Cancel.
d. Method getBackCommand() digunakan untuk mengambil event dari command yang bertipe Back.
e. Method startApp() digunakan untuk mejalankan aplikasi midlet.
f. Method getForm[1..29] digunakan untuk mengambil event dari objek form. Method ini dibuat sebanyak jumlah form yaitu 29 form.
g. Method destroyApp() digunakan untuk menghentikan aplikasi berbasis midlet.

2. Kelas UserSKF

Kelas UserSKF adalah kelas yang digunakan untuk melakukan dan meyimpan data user ke dalam RMS.



Gambar 3.12 Kelas UserSKF

a. Deskripsi Atribut

Mendeskripsikan setiap atribut dari kelas UserSKF dengan menentukan nama dan tipe atribut.

Tabel 3.8 Deskripsi atribut kelas UserSKF

Nama	Tipe
RecordStore	Store
Storage	String
Amount	Int

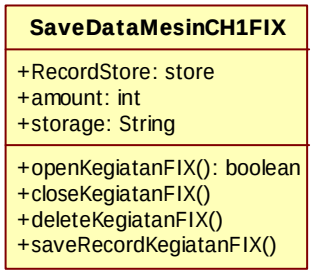
b. Deskripsi Operasi

Mendeskripsikan setiap operasi dari kelas SaveUserCH1, operasi-operasi yang ada dalam kelas ini adalah:

a. Method openUser() digunakan untuk membuka RMS yang menampung data user.
b. Method closeUser() digunakan untuk menutup RMS yang menampung data user.
c. Method deleteUser() digunakan untuk menghapus data yang terdapat di dalam RMS.
d. Method save recordUser() digunakan untuk menyimpan data ke dalam rms.

3. Kelas SaveDataMesinCH1FIX

Kelas SaveDataMesinCH1FIX adalah kelas yang digunakan untuk menghapus dan menambah data kegiatan yang tersimpan di dalam RMS data mesin FIX. Untuk nantinya siap ditransfer ke dalam komputer oleh admin.



Gambar 3.12 Kelas SaveDataMesinCH1FIX

a. Deskripsi Atribut

Mendeskripsikan setiap atribut dari kelas SaveDataMesinCH1 FIX dengan menentukan nama dan tipe atribut.

Tabel 3.9 Deskripsi atribut kelas SaveDataMesinCH1FIX

Nama	Tipe
RecordStore	Store
Storage	String
Amount	Int

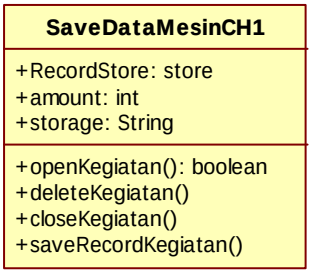
b. Deskripsi Operasi

Mendeskripsikan setiap operasi dari kelas SaveDataMesinCH1, operasi-operasi yang ada dalam kelas ini adalah:

a. Method openKegiatanFIX() digunakan untuk membuka RMS yang menampung data kegiatan.
b. Method closeKegiatanFIX() digunakan untuk menutup RMS yang menampung data kegiatan.
c. Method deleteKegiatanFIX() digunakan untuk menghapus data yang terdapat di dalam RMS.
d. Method saveRecordKegaiatnFIX() digunakan untuk menyimpan data ke dalam rms.

4. Kelas SaveDataMesinCH1

Kelas SaveDataMesinCH1 adalah kelas yang digunakan untuk menghapus dan menambah data kegiatan yang tersimpan di dalam RMS data mesin sementara.



Gambar 3.13 Kelas SaveDataMesinCH1

a. Deskripsi Atribut

Mendeskripsikan setiap atribut dari kelas SaveDataMesinCH1 dengan menentukan nama dan tipe atribut.

Tabel 3.9 Deskripsi atribut kelas SaveDataMesinCH1

Nama	Tipe
RecordStore	Store
Storage	String
Amount	Int

b. Deskripsi Operasi

Mendeskripsikan setiap operasi dari kelas SaveDataMesinCH1, operasi-operasi yang ada dalam kelas ini adalah:

e. Method openKegiatan() digunakan untuk membuka RMS yang menampung data kegiatan.

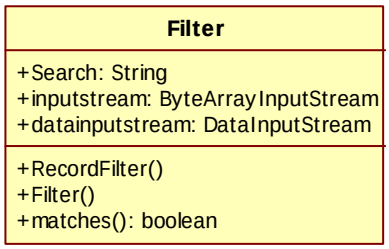
f. Method closeKegiatan() digunakan untuk menutup RMS yang menampung data kegiatan.

g. Method deleteKegiatan() digunakan untuk menghapus data yang terdapat di dalam RMS.

h. Method saveRecordKegiatan() digunakan untuk menyimpan data ke dalam rms.

5. Kelas Filter

Kelas Filter adalah kelas yang digunakan untuk mencocokkan inputan user dengan data yang tersimpan di dalam RMS.



Gambar 3.14 Kelas Filter

1. Deskripsi Atribut

Mendeskripsikan setiap operasi dari kelas Filter, atribut yang ada dalam kelas ini adalah:

Tabel 3.10 Deskripsi atribut kelas Filter

Nama	Tipe
Search	String
inputstream	ByteInputStream
datainputstream	DataInputStream

2. Deskripsi Operasi

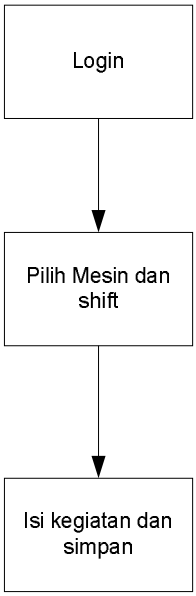
Mendeskripsikan setiap operasi dari kelas Filter, operasi-operasi yang ada dalam kelas ini adalah:

a. Method RecordFilter() berfungsi untuk mengambil record dari inputan user.
b. Method Filter() berfungsi untuk membuka stream.
c. Method Matches() berfungsi untuk membandingkan data yang tersimpan di dalam RMS dengan inputan user.

3.4 Struktur Menu Program

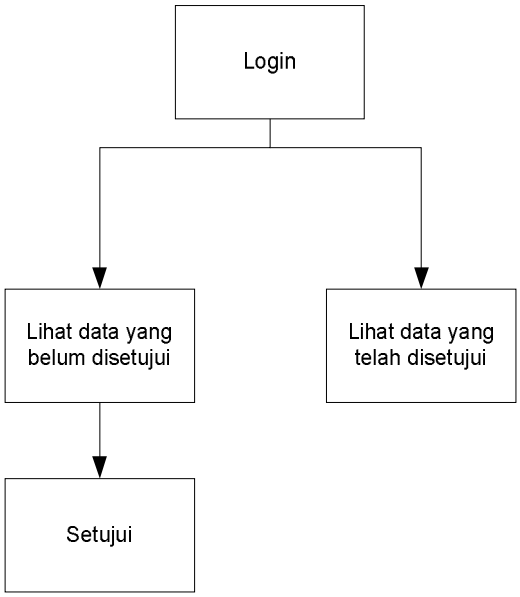
Struktur menu program dari bagian operator, *foreman*, dan admin dapat dilihat pada gambar 3.51.

3.4.1 Struktur Menu Bagian Operator



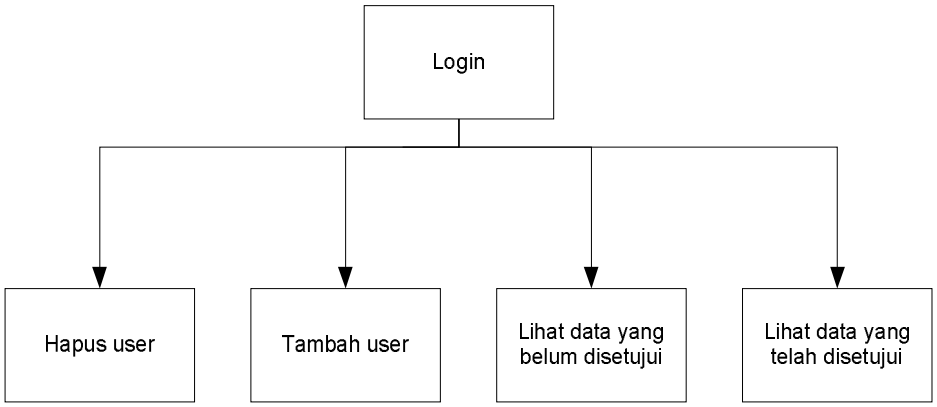
Gambar 3.15 Struktur menu bagian operator

3.4.2 Struktur Menu Bagian Foreman



Gambar 3.16 Struktur menu bagian foreman

3.4.3 Struktur Menu Bagian Admin

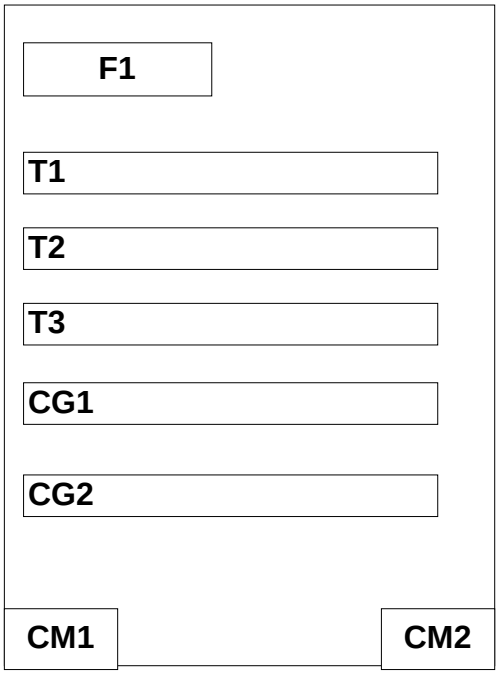


Gambar 3.17 Struktur menu bagian admin

3.5 Perancangan Antarmuka

Perancangan antarmuka merupakan perancangan yang dibuat untuk memberikan gambaran sebelum aplikasi dibangun. Perancangan antar muka *input/ output*, struktur dan tampilan dirancang sedemikian rupa agar mudah dioperasikan oleh pengguna.

3.5.1 Tampilan Halaman Login Pengguna



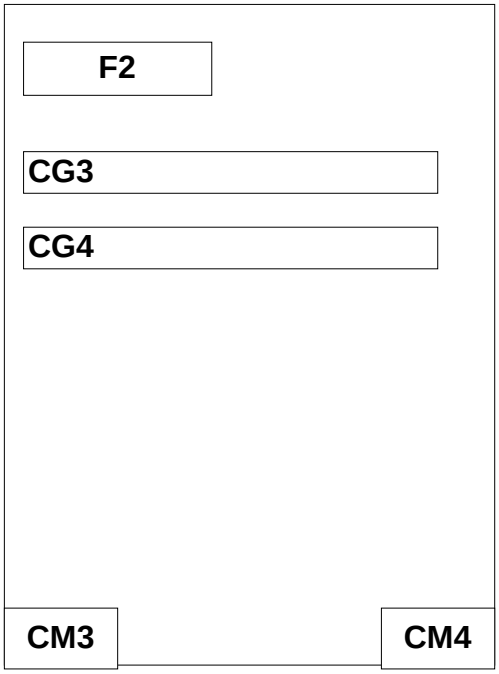
Gambar 3.18 Tampilan halaman login pengguna

Tabel 3.11 Keterangan tampilan halaman login

Nomor Simbol	Keterangan
F1	Tampilan halaman login pengguna
T1	*Textfield* untuk menginputkan NPK
T2	*Textfield* untuk menginputkan Nama
T3	*Textfield* untuk menginputkan Password
CG1	*ChoiceGroup* yang digunakan untuk memilih chanel tempat operator atau foreman bekerja

CG2	*ChoiceGroup* yang digunakan untuk memilih status user
CM1	*Command* yang digunakan bila user ingin keluar dari aplikasi.
CM2	*Command* yang digunakan bila user ingin melanjutkan ke *form* selanjutnya.

3.5.2 Tampilan Halaman Mesin



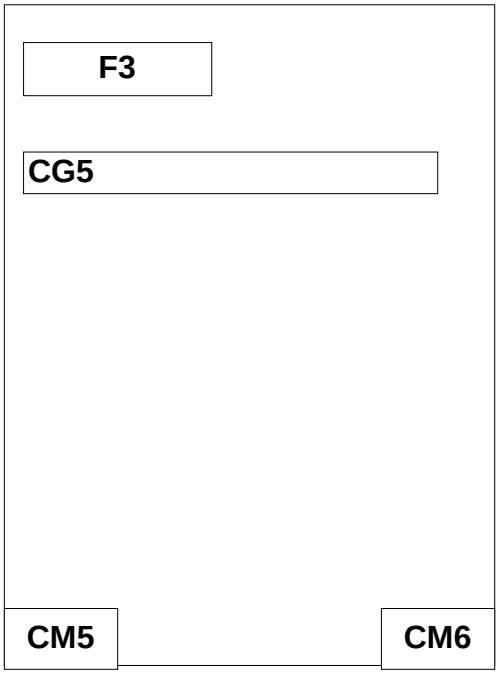
Gambar 3.19 Tampilan halaman mesin

Tabel 3.12 Keterangan tampilan halaman mesin

Nomor Simbol	Keterangan
T02	Tampilan *form* pengisian mesin
CG3	*ChoiceGroup* untuk memilih shift tempat operator bekerja

CG4	*ChoiceGroup* untuk memilih mesin tempat operator bekerja.
CM3	*Command* untuk kembali ke *form* login
CM4	*Command* untuk melanjutkan ke menu *form* selanjutnya

3.5.3 Tampilan *Form* Isi Kegiatan

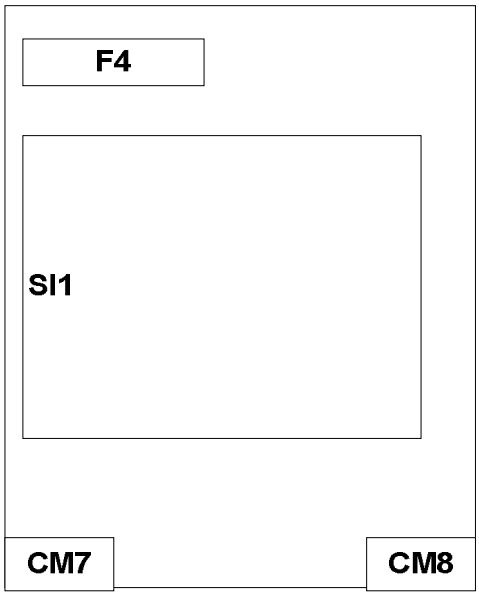


Gambar 3.20 Tampilan form isi kegiatan

Tabel 3.13 Keterangan tampilan *form* isi kegiatan

Nomor Simbol	Keterangan
F3	Tampilan *form* isi kegiatan
CG5	*ChoiceGroup* untuk mengisi kegiatan mesin.
CM5	*Command* untuk kembali ke *form* sebelumnya.
CM6	*Command* untuk menyimpan kegiatan *maintenance*.

3.5.4 Tampilan *Form* Data Kegiatan Belum Disetujui

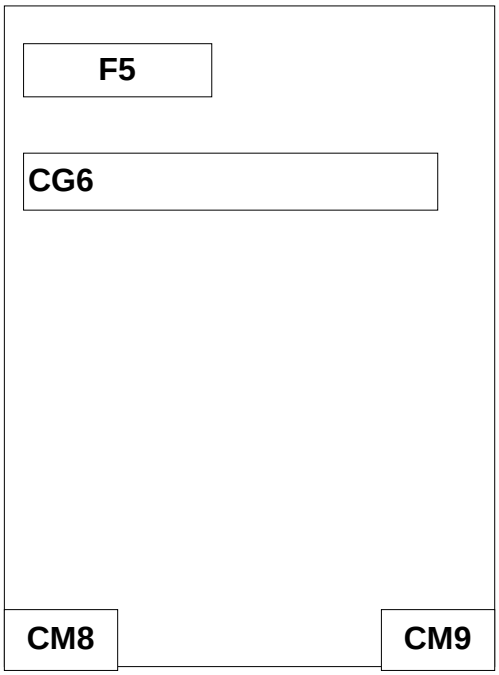


Gambar 3.21 Tampilan form data kegiatan belum disetujui

Tabel 3.14 Keterangan tampilan *form* data kegiatan belum disetujui

Nomor Simbol	Keterangan
F4	Tampilan *form* data kegiatan belum disetujui
SI1	*StringItem* untuk menampilkan data yang belum disetujui
CM7	*Command* untuk kembali ke *form* sebelumnya
CM8	*Command* untuk keluar dari aplikasi

3.5.5 Tampilan *Form* Menu Foreman

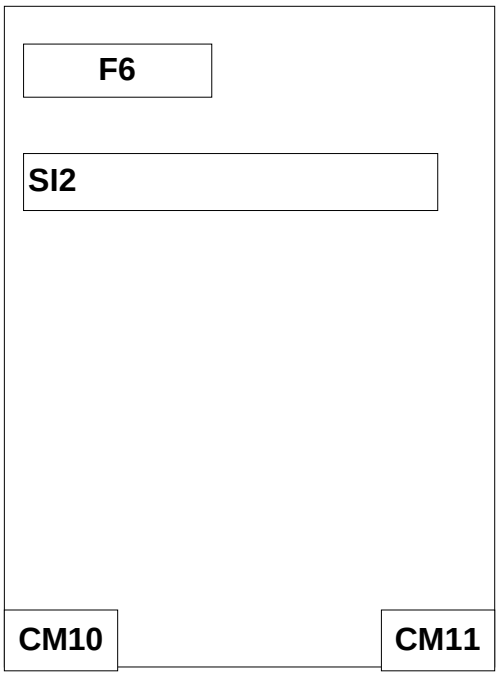


Gambar 3.22 Tampilan form menu foreman

Tabel 3.15 Keterangan tampilan *form* menu foreman

Nomor Simbol	Keterangan
F5	Tampilan halaman *form* menu foreman
CG6	*ChoiceGroup* menu foreman
CM8	*Command* untuk ke menu sebelumnya
CM9	*Command* untuk ke menu selanjutnya.

3.5.6 Tampilan *Form Approve Foreman*

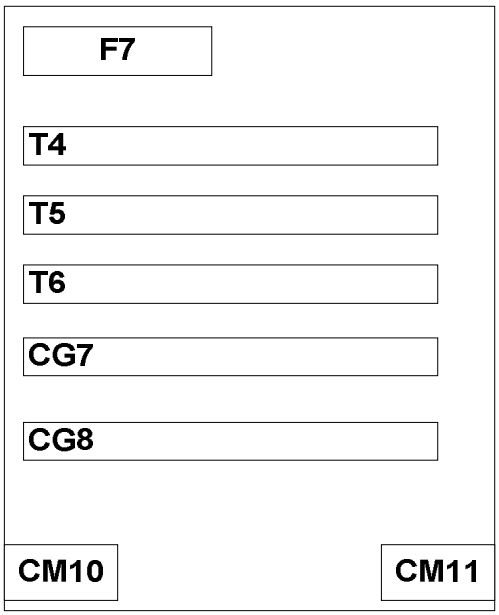


Gambar 3.23 Tampilan halaman approve foreman

Tabel 3.16 Keterangan tampilan halaman approve foreman

Nomor Simbol	Keterangan
F6	Tampilan *Form* Approve Foreman
SI2	String data kegiatan yang belum disetujui
CM10	*Command* untuk kembali kemenu selanjutnya
CM11	*Command* untuk menyetujui data kegiatan

3.5.7 Tampilan *Form* Tambah Data

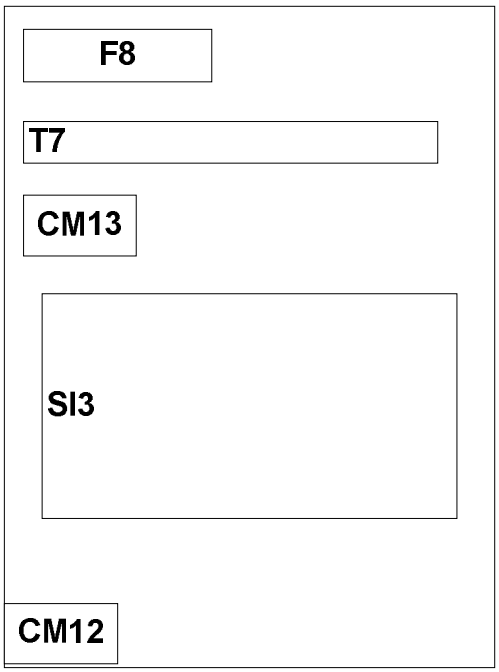


Gambar 3.24 Tampilan form tambah data

Tabel 3.17 Keterangan tampilan *form* tambah data

Nomor Simbol	Keterangan
F7	Tampilan *form* tambah data
T4	*TextField* untuk mengisi NPK
T5	*TextField* untuk mengisi nama
T6	*TextField* untuk mengisi password
CG7	*ChoiceGroup* untuk memilih chanel
CG8	*ChoiceGroup* untuk memilih status
CM10	Command untuk kembali ke menu sebelumnya
CM11	Command untuk menyimpan data

3.5.8 Tampilan *Form* Hapus Data



Gambar 3.25 Tampilan form hapus data

Tabel 3.18 Keterangan tampilan halaman hapus data

Nomor Simbol	Keterangan
F8	Tampilan *form* hapus data
T7	TextField untuk menginput index yang akan dihapus
CM13	Command untuk menghapus data
SI3	StringItem untuk menampilkan data user
CM12	Command untuk keluar dari *form*