

## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Alur Penelitian

Untuk pembangunan sistem, penelitian menggunakan metodologi SDLC (*Software Development Life Cycle*). SDLC adalah siklus hidup pengembangan sistem yang terdapat di dalam rekayasa *software*. SDLC itu sendiri merupakan sebuah pola yang berfungsi untuk melakukan proses pembuatan maupun perubahan sistem dari sebuah aplikasi (Dita, 2012).

Metodologi SDLC yang dipakai dalam penelitian adalah *waterfall model*. *Waterfall model* atau *classic life cycle* merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). *Waterfall model* adalah metode pengembangan *software* yang bersifat sekuensial, tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan (Justinus, 2007).

*Waterfall model* ditunjukkan pada Gambar 3.1.



**Gambar 3.1** Alur Penelitian

Gambar 3.1 menunjukkan *waterfall model* yang digunakan untuk memudahkan penulis dalam melakukan penelitian. Adapun penjelasan alur penelitian skripsi ini adalah sebagai berikut:

#### ANALISIS KEBUTUHAN

Analisis kebutuhan adalah tahap menganalisis semua kebutuhan untuk mengembangkan aplikasi. Pada tahap ini, analisis kebutuhan mengacu kepada *work flow MOC* secara manual.

#### PERANCANGAN MENGGUNAKAN UML

Perancangan menggunakan UML adalah tahap merancang aplikasi dengan *use case diagram, activity diagram, class diagram, entity relationship diagram*, kamus data *database*, rancangan prosedural, dan rancangan *interface*.

#### PENGEMBANGAN APLIKASI

Pengembangan aplikasi adalah tahap mengembangkan aplikasi sesuai dengan perancangan aplikasi yang telah dibuat.

#### PENGUJIAN

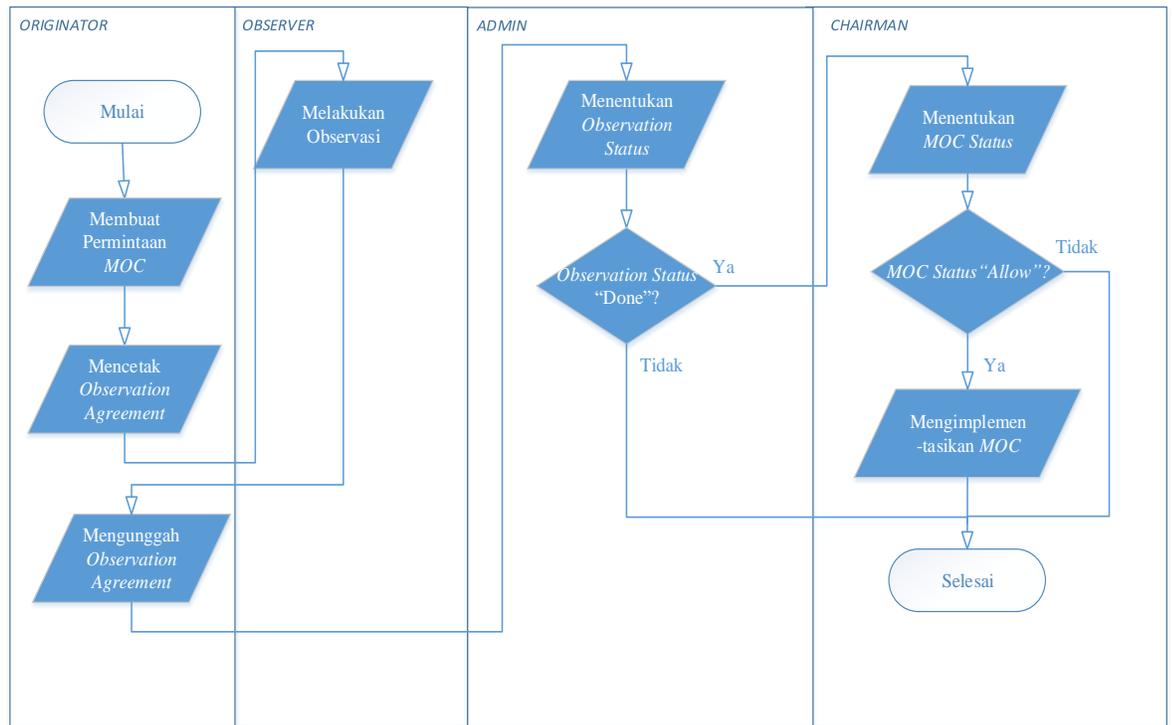
Pengujian adalah tahap menguji aplikasi kepada *user* dengan pengujian secara fungsionalitas aplikasi menggunakan metode *interview*.

#### IMPLEMENTASI

Implementasi adalah tahap mengimplementasikan aplikasi dan disesuaikan dengan saran yang diterima.

### 3.2. *Work Flow MOC*

Gambar 3.2 menunjukkan *work flow MOC*



**Gambar 3.2** *Work Flow MOC*

Penjelasan mengenai *work flow MOC* adalah sebagai berikut:

1. *Originator* membuat permintaan *MOC*.
2. *Originator* mencetak *observation agreement*.
3. *Observer* melakukan observasi.
4. *Originator* mengunggah *observation agreement*.
5. *Admin* menentukan *observation status*.
6. *Chairman* menentukan *MOC status*.
7. *Chairman* mengimplementasikan *MOC*.

### 3.3. Analisis Kebutuhan

Analisis kebutuhan didapatkan dari studi literatur dan melakukan diskusi bersama calon pengguna aplikasi. Berikut merupakan analisis kebutuhan pada aplikasi:

- a. Halaman untuk *login* sebagai *authentication user*, berfungsi agar aplikasi hanya dapat digunakan oleh *user* yang telah terdaftar.
- b. Halaman untuk penyimpanan, pembaharuan, penghapusan, dan pencarian data *activity*, berfungsi agar *user* dapat melakukan penyimpanan, pembaharuan, penghapusan, dan pencarian data.
- c. Halaman untuk penyimpanan, pembaharuan, penghapusan, dan pencarian data *employee*, berfungsi agar *user* dapat melakukan penyimpanan, pembaharuan, penghapusan, dan pencarian data.
- d. Halaman untuk pembaharuan data *MOC* oleh *admin*, berfungsi agar *user* dapat melakukan pembaharuan data. Hal ini mengacu pada salah satu proses *MOC* pada Gambar 3.2, yaitu proses menentukan *Observation Status*.
- e. Halaman untuk penghapusan, dan pencarian data *MOC* oleh *admin*, berfungsi agar *user* dapat melakukan penghapusan, dan pencarian data.
- f. Halaman untuk penyimpanan data *MOC* oleh *originator*, berfungsi agar *user* dapat melakukan penyimpanan data. Hal ini mengacu pada salah satu proses *MOC* pada Gambar 3.2, yaitu proses membuat permintaan *MOC*.
- g. Halaman untuk pencarian data *MOC* oleh *originator*, berfungsi agar *user* dapat melakukan pencarian data.
- h. Halaman untuk pembaharuan data *MOC* oleh *chairman*, berfungsi agar *user* dapat melakukan pembaharuan data. Hal ini mengacu pada salah satu proses *MOC* pada Gambar 3.2, yaitu proses menentukan *MOC Status*.
- i. Halaman untuk pencarian data *MOC* oleh *chairman*, berfungsi agar *user* dapat melakukan pencarian data.

- j. Halaman untuk mencetak *Observation Agreement* oleh *originator*, digunakan *user* untuk mencetak *Observation Agreement*. Hal ini mengacu pada salah satu proses *MOC* pada Gambar 3.2, yaitu proses mencetak *Observation Agreement*.
- k. Halaman untuk mengunggah *Observation Agreement* oleh *originator*, digunakan *user* untuk mengunggah *Observation Agreement*. Hal ini mengacu pada salah satu proses *MOC* pada Gambar 3.2, yaitu proses mengunggah *Observation Agreement*.

### 3.4. Alat dan Bahan

Dalam pembuatan aplikasi digunakan beberapa alat dan bahan untuk mendukung berjalannya perancangan dan pembuatan aplikasi.

#### 3.4.1. Alat

Alat yang digunakan untuk membuat aplikasi adalah berupa *hardware* dan *software*.

##### a. *Hardware*

*Hardware* yang digunakan dapat dilihat pada Tabel 3.1.

**Tabel 3.1** Spesifikasi *Hardware*

<b>Nama Perangkat</b>	<b>Spesifikasi</b>
<i>Processor</i>	<i>Intel Core i5-3337u CPU @ 1.80GHz</i>
<i>RAM</i>	12.00 GB
<i>Operating System</i>	<i>Windows 10 Enterprise 64-bit</i>
<i>Hard Disk</i>	500 GB

b. *Software*

*Software* merupakan hal yang terpenting, karena digunakan untuk memberi perintah-perintah pada *hardware* agar dapat saling berinteraksi. *Software* yang digunakan dapat dilihat pada Tabel 3.2.

**Tabel 3.2** Spesifikasi *Software*

Nama Perangkat	Spesifikasi
<b><i>Tools</i> Pembangun Aplikasi</b>	<i>Microsoft Visual Studio 2015</i>
	<i>Microsoft SQL Server Management Studio 2014</i>

**3.4.2. Bahan**

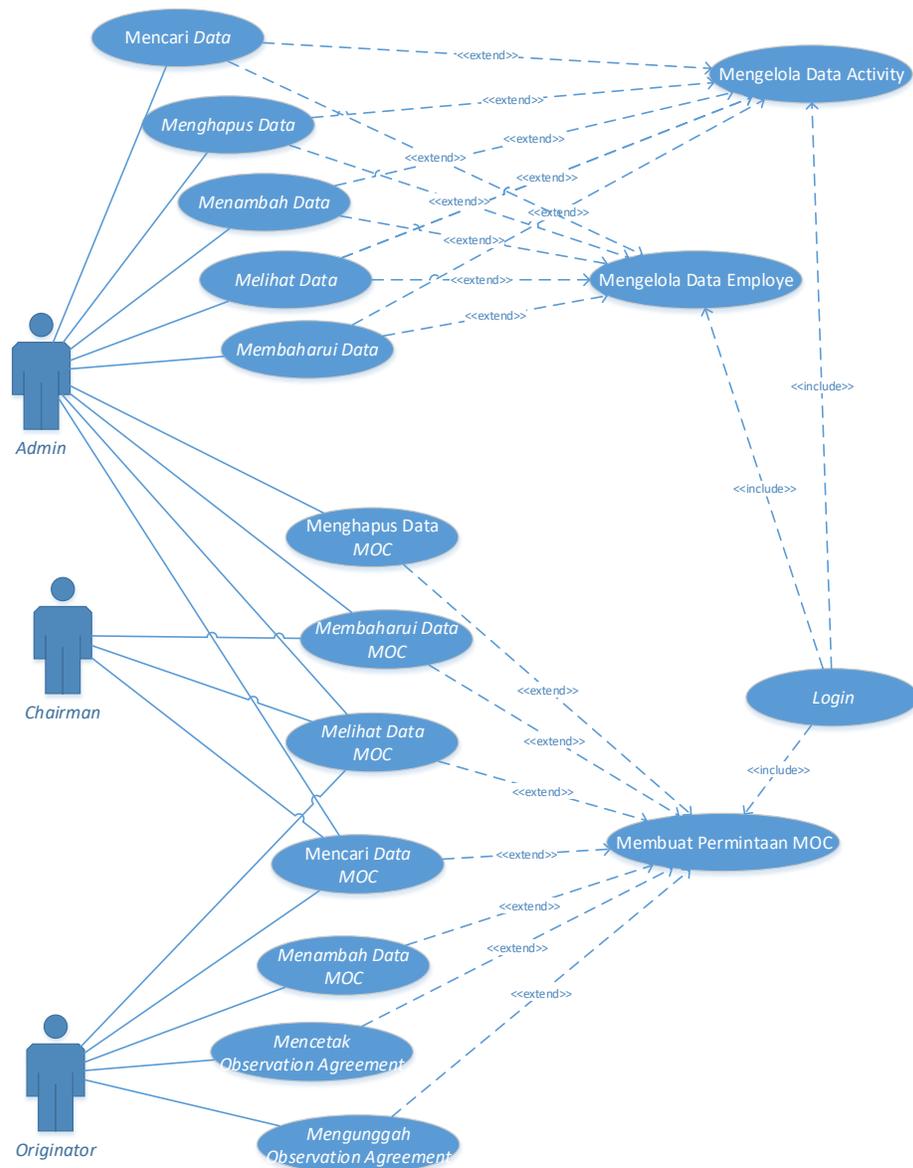
Bahan yang akan digunakan dalam pembuatan aplikasi adalah data yang diperoleh dari:

- a. Studi literatur berdasarkan *Management of Change* di PT Vale Indonesia Tbk.
- b. *Template form CM001 Change Management Request Form* yang telah ada sebelumnya.
- c. *Risk Assessment* yang telah ada sebelumnya.

### 3.5. Rancangan

#### 3.5.1. Use Case Diagram

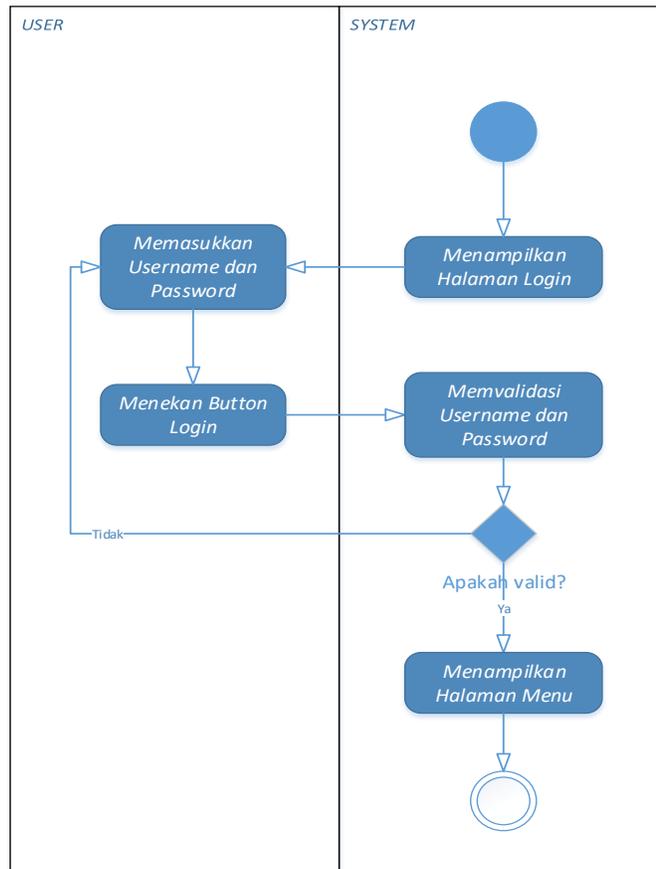
Gambar 3.3 menunjukkan *Use case diagram*. *Use case diagram* ini dibuat berdasarkan dari analisis kebutuhan yang berfungsi untuk membantu perancangan aplikasi. *Use case diagram* ini memiliki 3 aktor dan 16 *use case*



**Gambar 3.3** Use Case Diagram

### 3.5.2. Activity Diagram

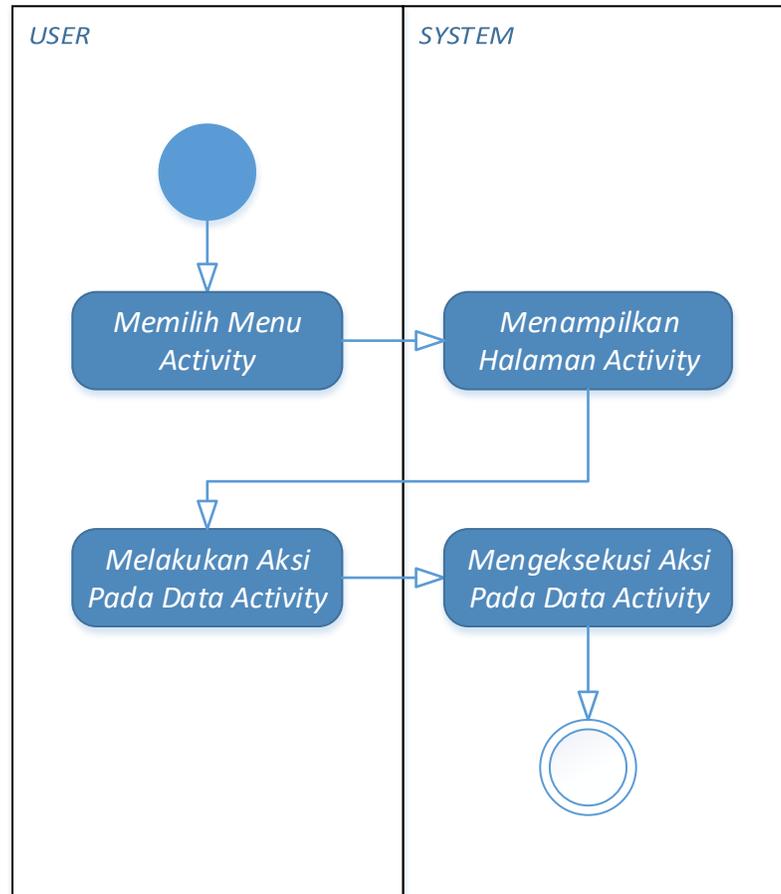
#### a. Halaman Login



**Gambar 3.4** Activity Diagram Halaman Login

Keterangan pada Gambar 3.4 adalah sebagai berikut:

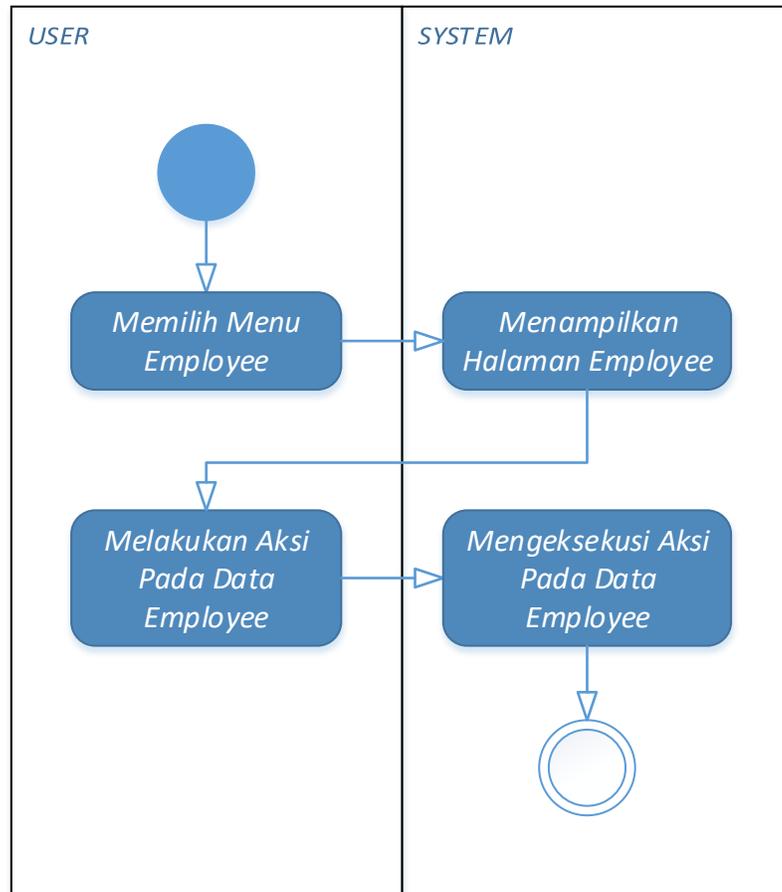
- System* menampilkan halaman *login*.
- User* memasukkan *username* dan *password*.
- User* menekan *button login*. Jika *button login* ditekan, maka *system* melakukan validasi pada *username* dan *password*.
- System* melakukan validasi pada *username* dan *password*. Jika salah, maka *system* meminta *user* untuk memasukkan kembali *username* dan *password*. Jika benar, maka *system* akan menampilkan halaman menu.

b. Halaman *Activity*

**Gambar 3.5** *Activity Diagram* Halaman *Activity*

Keterangan pada Gambar 3.5 adalah sebagai berikut:

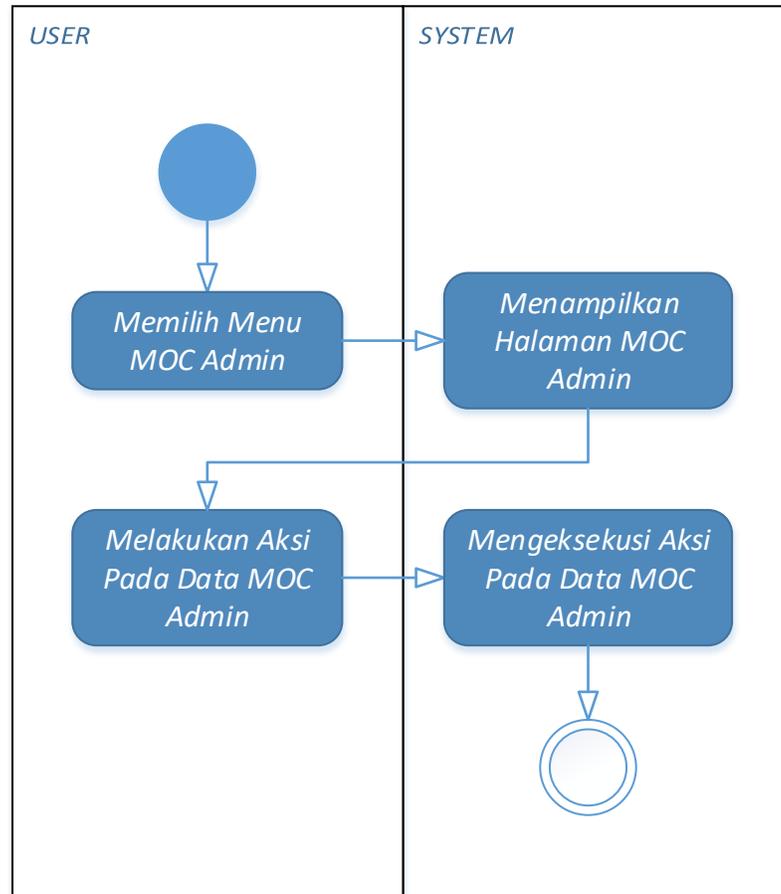
- a. *User* memilih menu *Activity* pada halaman menu.
- b. *System* menampilkan halaman *Activity*.
- c. *User* melakukan aksi pada data *Activity*, seperti menambah, melihat, membaharui, menghapus, dan mencari data *Activity*.
- d. *System* mengeksekusi data sesuai dengan aksi yang telah dilakukan.

c. Halaman *Employee*

**Gambar 3.6** Activity Diagram Halaman *Employee*

Keterangan pada Gambar 3.6 adalah sebagai berikut:

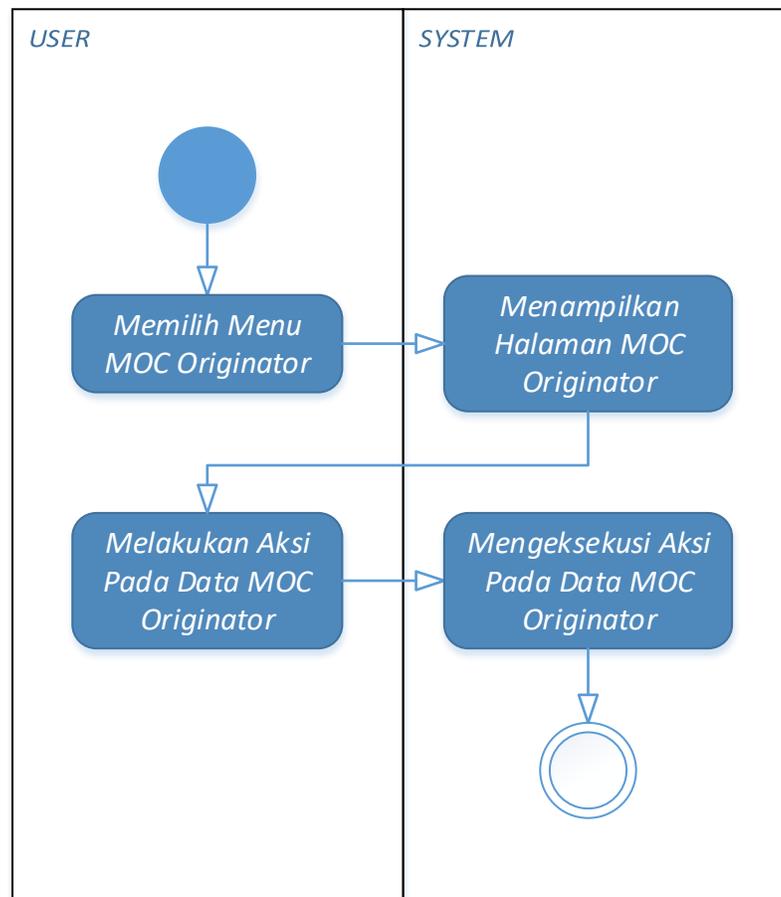
- User* memilih menu *Employee* pada halaman menu.
- System* menampilkan halaman *Employee*.
- User* melakukan aksi pada data *Employee*, seperti menambah, melihat, membaharui, menghapus, dan mencari data *Employee*.
- System* mengeksekusi data sesuai dengan aksi yang telah dilakukan.

d. Halaman *MOC Admin*

**Gambar 3.7** Activity Diagram Halaman *MOC Admin*

Keterangan pada Gambar 3.7 adalah sebagai berikut:

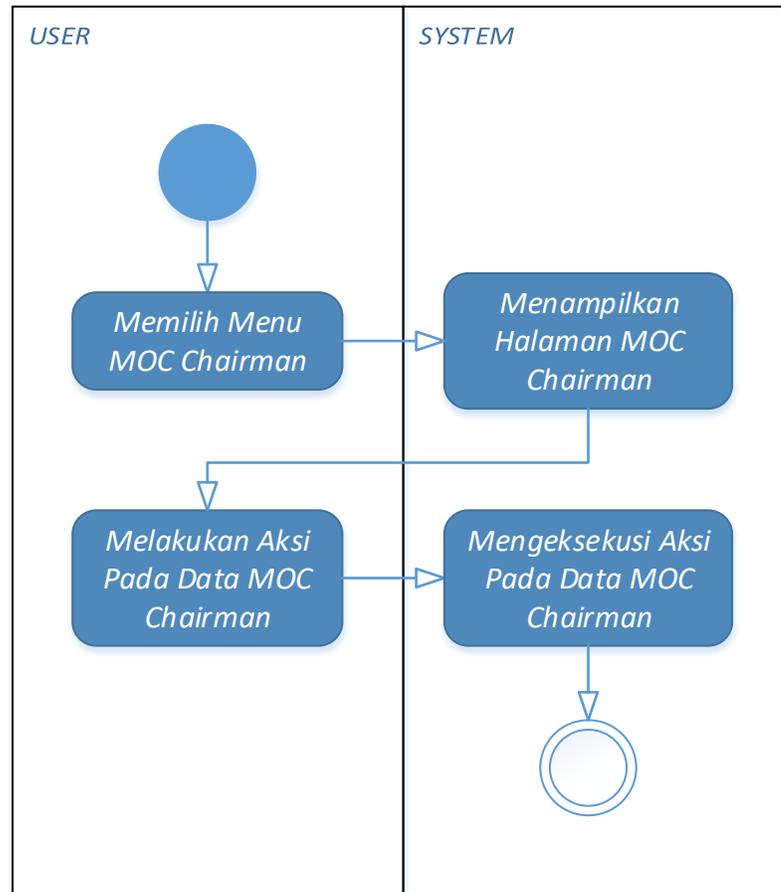
- a. *User* memilih menu *MOC Admin* pada halaman menu.
- b. *System* menampilkan halaman *MOC Admin*.
- c. *User* melakukan aksi pada data *MOC Admin*, seperti melihat, membaharui, menghapus, dan mencari data *MOC Admin*.
- d. *System* mengeksekusi data sesuai dengan aksi yang telah dilakukan.

e. Halaman *MOC Originator*

**Gambar 3.8** Activity Diagram Halaman *MOC Originator*

Keterangan pada Gambar 3.8 adalah sebagai berikut:

- User* memilih menu *MOC Originator* pada halaman menu.
- System* menampilkan halaman *MOC Originator*.
- User* melakukan aksi pada data *MOC Originator*, seperti menambah, melihat, dan mencari data *MOC Originator*.
- System* mengeksekusi data sesuai dengan aksi yang telah dilakukan.

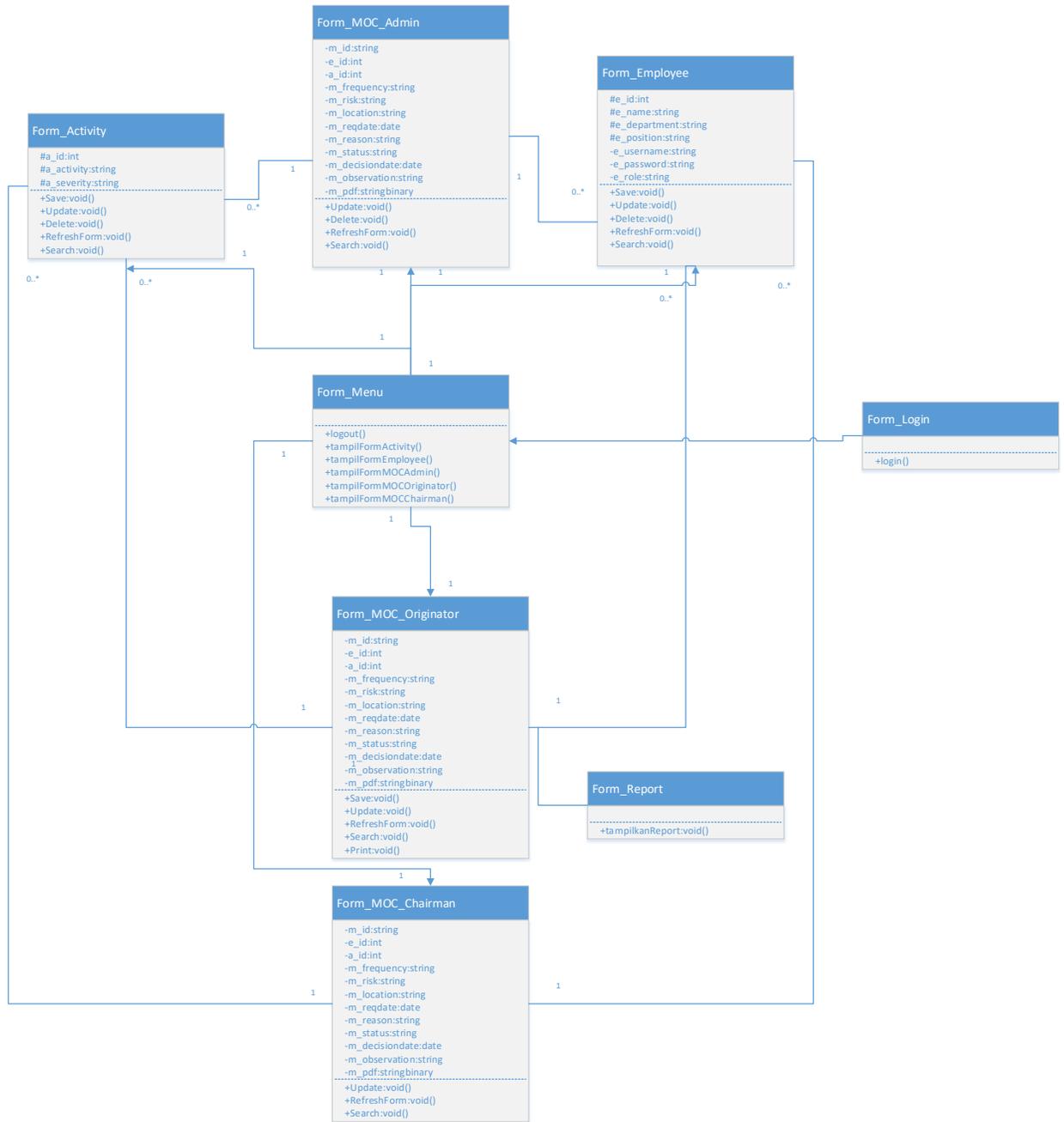
f. Halaman *MOC Chairman*

**Gambar 3.9** Activity Diagram Halaman *MOC Chairman*

Keterangan pada Gambar 3.9 adalah sebagai berikut:

- User* memilih menu *MOC Chairman* pada halaman menu.
- System* menampilkan halaman *MOC Chairman*.
- User* melakukan aksi pada data *MOC Chairman*, seperti melihat, membaharui, dan mencari data *MOC Chairman*.
- System* mengeksekusi data sesuai dengan aksi yang telah dilakukan.

### 3.5.3. Class Diagram



Gambar 3.10 Class Diagram

Berikut ini merupakan penjelasan Gambar 3.10 *Class Diagram* Aplikasi:

a. *Class Form\_Form\_Activity*

Pada *class Form\_Form\_Activity* terdapat fungsi *save*, *select*, *update*, dan *delete*. *Save* berfungsi untuk menambahkan data. *Select* berfungsi untuk menampilkan data. *Update* berfungsi untuk membarui data yang telah ada. Sedangkan *Delete* berfungsi untuk menghapus data.

*Class Form\_Form\_Activity* memiliki asosiasi dengan *class Form\_MOC\_Admin*. *Form\_Form\_Activity* boleh tidak memiliki *Form\_MOC\_Admin* atau memiliki *Form\_MOC\_Admin* lebih dari nol.

*Class Form\_Form\_Activity* memiliki asosiasi dengan *class Form\_MOC\_Originator*. *Form\_Form\_Activity* boleh tidak memiliki *Form\_MOC\_Originator* atau memiliki *Form\_MOC\_Originator* lebih dari nol.

*Class Form\_Form\_Activity* memiliki asosiasi dengan *class Form\_MOC\_Chairman*. *Form\_Form\_Activity* boleh tidak memiliki *Form\_MOC\_Chairman* atau memiliki *Form\_MOC\_Chairman* lebih dari nol.

b. *Class Form\_MOC\_Originator*

Pada *class Form\_MOC\_Originator* terdapat fungsi *save*, *select*, *update*, *print*, dan *upload*. *Save* berfungsi untuk menambahkan data. *Select* berfungsi untuk menampilkan data. *Update* berfungsi untuk membarui data yang telah ada. *Print* berfungsi untuk mencetak data. Sedangkan *Upload* berfungsi untuk mengunggah data.

*Class Form\_MOC\_Originator* memiliki asosiasi dengan *class Form\_Activity*. *Form\_MOC\_Originator* hanya boleh memiliki satu *Form\_Activity* dan *class Form\_MOC\_Originator* memiliki asosiasi dengan *class Form\_Employee*. *Form\_MOC\_Originator* hanya boleh memiliki satu *Form\_Employee*.

c. *Class Form\_MOC\_Admin*

Pada *class Form\_MOC\_Admin* terdapat fungsi *select*, *update*, *delete*. *Select* berfungsi untuk menampilkan data. *Update* berfungsi untuk membarui data yang telah ada. *Delete* berfungsi untuk menghapus data.

*Class Form\_MOC\_Admin* memiliki asosiasi dengan *class Form\_Activity*. *Form\_MOC\_Admin* hanya boleh memiliki satu *Form\_Activity* dan *class Form\_MOC\_Admin* memiliki asosiasi dengan *class Form\_Employee*. *Form\_MOC\_Admin* hanya boleh memiliki satu *Form\_Employee*.

d. *Class Form\_MOC\_Chairman*

Pada *class Form\_MOC\_Chairman* terdapat fungsi *select* dan *update*. *Select* berfungsi untuk menampilkan data. *Update* berfungsi untuk membarui data yang telah ada.

*Class Form\_MOC\_Chairman* memiliki asosiasi dengan *class Form\_Activity*. *Form\_MOC\_Chairman* hanya boleh memiliki satu *Form\_Activity* dan *class Form\_MOC\_Chairman* memiliki asosiasi dengan *class Form\_Employee*. *Form\_MOC\_Chairman* hanya boleh memiliki satu *Form\_Employee*.

e. *Class Form\_Employee*

Pada *class Form\_Employee* terdapat fungsi *insert*, *select*, *update*, dan *delete*. *Insert* berfungsi untuk menambahkan data. *Select* berfungsi untuk menampilkan data. *Update* berfungsi untuk membarui data yang telah ada. Sedangkan *Delete* berfungsi untuk menghapus data.

*Class Form\_Employee* memiliki asosiasi dengan *class Form\_MOC\_Admin*. *Form\_Employee* boleh tidak memiliki *Form\_MOC\_Admin* atau memiliki *Form\_MOC\_Admin* lebih dari nol.

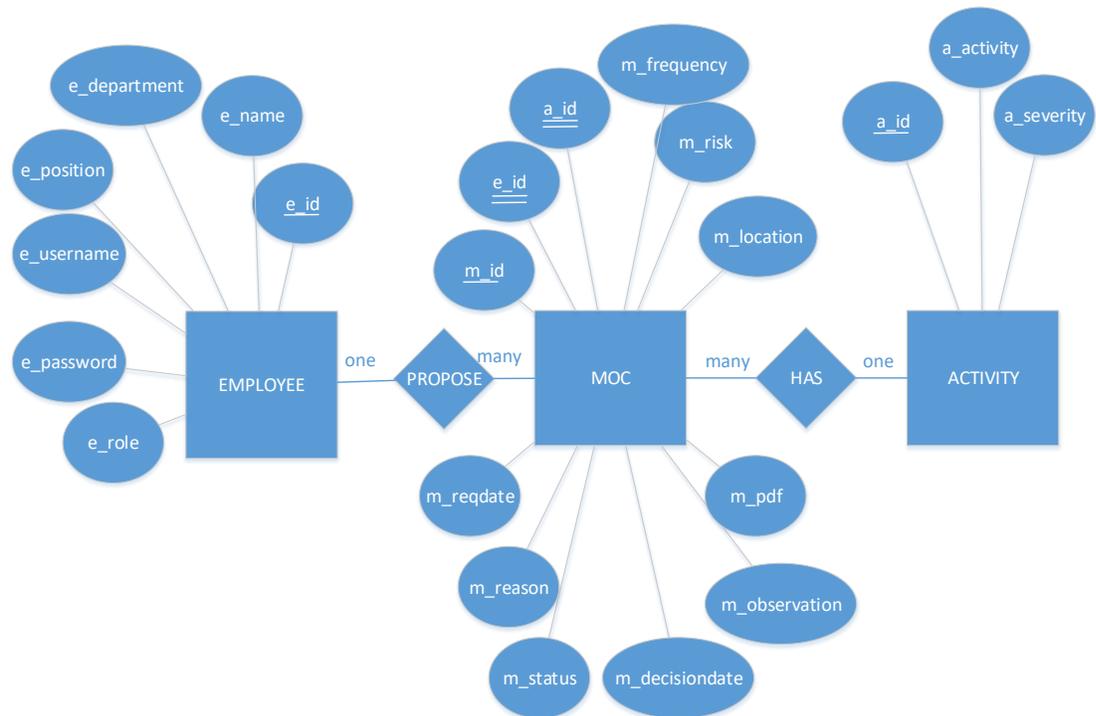
*Class Form\_Employee* memiliki asosiasi dengan *class Form\_MOC\_Originator*. *Form\_Employee* boleh tidak memiliki *Form\_MOC\_Originator* atau memiliki *Form\_MOC\_Originator* lebih dari nol.

*Class Form\_Employee* memiliki asosiasi dengan *class Form\_MOC\_Chairman*. *Form\_Employee* boleh tidak memiliki *Form\_MOC\_Chairman* atau memiliki *Form\_MOC\_Chairman* lebih dari nol.

#### **3.5.4. Entity Relationship Diagram**

*Entity Relationship Diagram* (ERD) merupakan diagram yang digunakan untuk merancang *database* dan memperlihatkan relasi antar entitas atau objek dengan atributnya. Tujuan ERD dalam pengembangan aplikasi adalah dapat memberikan gambaran umum tentang sistem yang dikembangkan sehingga dapat memudahkan dalam merancang *database*. Gambaran ERD yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.11. Pada Gambar 3.11 dapat dilihat bahwa *database* yang dirancang memiliki 3 buah entitas yaitu:

- a. *Activity*
- b. *MOC*
- c. *Employee*



**Gambar 3.11** Entity Relationship Diagram

Pada entitas *Employee* dan *MOC* memiliki relasi *one-to-many* yang memiliki arti bahwa satu data pada entitas *Employee* dapat memiliki banyak data pada entitas *MOC*, tetapi satu data pada entitas *MOC* tidak dapat memiliki banyak data pada entitas *Employee*.

Pada entitas *Activity* dan *MOC* memiliki relasi *one-to-many* yang memiliki arti bahwa satu data pada entitas *Activity* dapat memiliki banyak data pada entitas *MOC*, tetapi satu data pada entitas *MOC* tidak dapat memiliki banyak data pada entitas *Activity*.

### 3.5.5. Kamus Data Database

Kamus data *database* yang ada pada aplikasi *Management of Change* PT Vale Indonesia Tbk ditunjukkan pada Tabel 3.3, Tabel 3.4, Tabel 3.5, dan Tabel 3.6.

**Tabel 3.3** Kamus Data Tabel *Activity*

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Panjang Data</b>	<b>Key</b>	<b>Keterangan</b>
<i>a_id</i>	<i>Int</i>		<i>Primary Key</i>	Identitas <i>activity</i>
<i>a_activity</i>	<i>Varchar</i>	50		Nama <i>activity</i>
<i>a_severity</i>	<i>Varchar</i>	15		Kategori <i>severity</i>

**Tabel 3.4** Kamus Data Tabel *Employee*

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Panjang Data</b>	<b>Key</b>	<b>Keterangan</b>
<i>e_id</i>	<i>Int</i>		<i>Primary Key</i>	Identitas <i>employee</i>
<i>e_name</i>	<i>Varchar</i>	30		Nama <i>employee</i>
<i>e_department</i>	<i>Varchar</i>	50		Nama <i>department</i>
<i>e_position</i>	<i>Varchar</i>	50		Kategori <i>position</i>
<i>e_username</i>	<i>Varchar</i>	15		Nama <i>username</i>
<i>e_password</i>	<i>Varchar</i>	15		Nama <i>password</i>
<i>e_role</i>	<i>Varchar</i>	10		Kategori <i>role</i>

**Tabel 3.5** Kamus Data Tabel *MOC*

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
<i>m_id</i>	<i>Varchar</i>	10	<i>Primary Key</i>	Identitas <i>MOC</i>
<i>e_id</i>	<i>Int</i>		<i>Foreign Key</i>	<i>Primary Key</i> (identitas) dari tabel <i>employee</i>
<i>a_id</i>	<i>Int</i>		<i>Foreign Key</i>	<i>Primary Key</i> (identitas) dari tabel <i>activity</i>

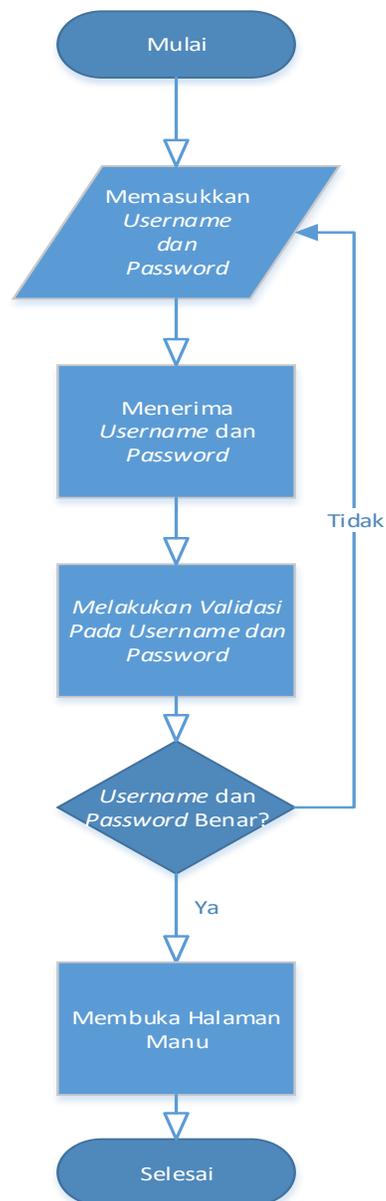
**Tabel 3.6** Kamus Data Tabel *MOC* (lanjutan)

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
<i>m_frequency</i>	<i>Varchar</i>	30		Kategori <i>frequency</i>
<i>m_risk</i>	<i>Varchar</i>	30		Kategori <i>risk</i>
<i>m_location</i>	<i>Varchar</i>	30		Nama <i>location</i>
<i>m_reqdate</i>	<i>Date</i>			Waktu <i>request date</i>
<i>m_reason</i>	<i>Varchar</i>	30		Penjelasan <i>reason</i>
<i>m_status</i>	<i>Varchar</i>	30		Kategori <i>status</i>
<i>m_decisiondate</i>	<i>Date</i>			Waktu <i>decision date</i>
<i>m_observation</i>	<i>Varchar</i>	15		Kategori <i>observation</i>
<i>m_pdf</i>	<i>Varbinary</i>	<i>Max</i>		Nama <i>file</i>

### 3.5.6. Rancangan Prosedural

#### 1. Rancangan Prosedur *Login*

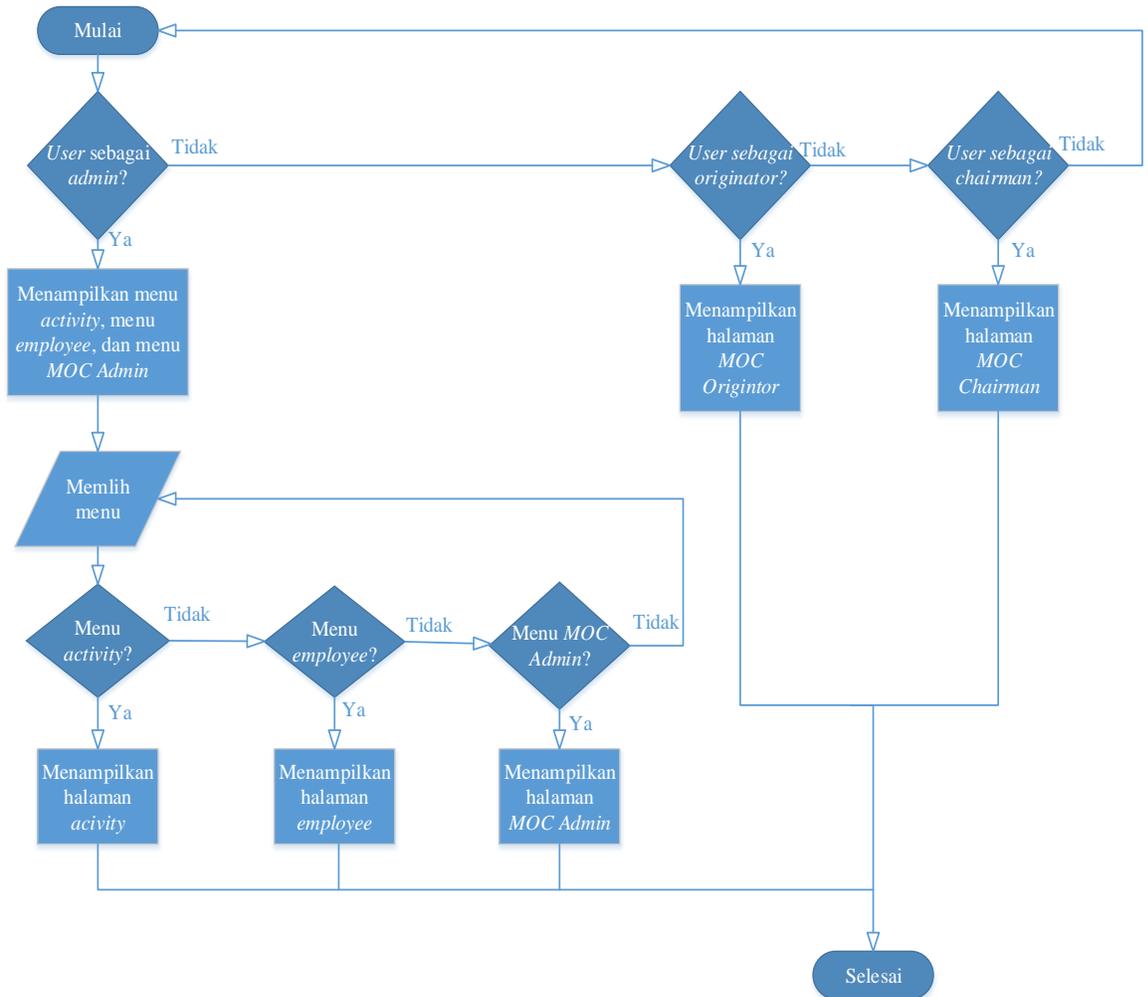
Berikut ini merupakan perancangan prosedur *login*, yang dapat dilihat pada Gambar 3.12.



**Gambar 3.12** Flow Chart Prosedural *Login*

## 2. Rancangan Prosedur Halaman Menu

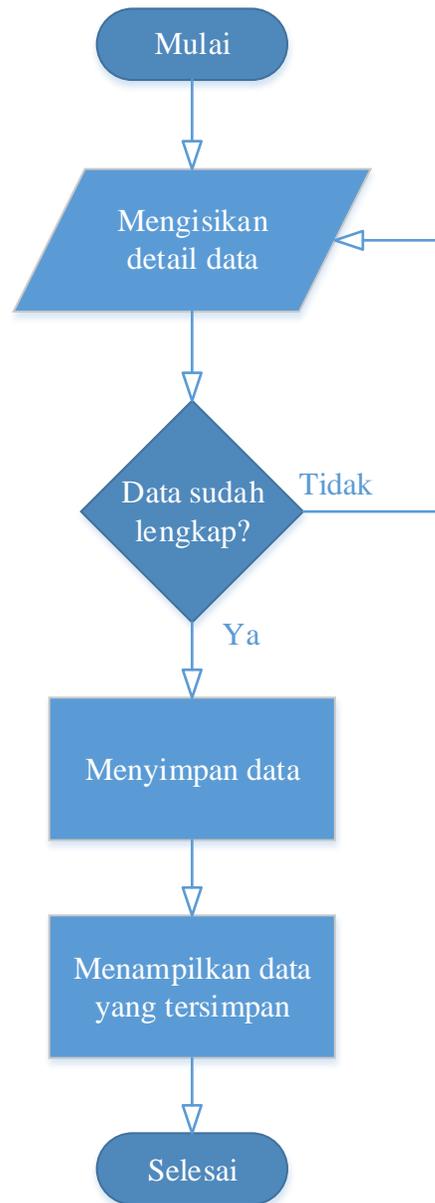
Berikut ini merupakan perancangan prosedur halaman menu, yang dapat dilihat pada Gambar 3.13.



**Gambar 3.13** Flow Chart Prosedural Halaman Menu

### 3. Rancangan Prosedur Penambahan Data

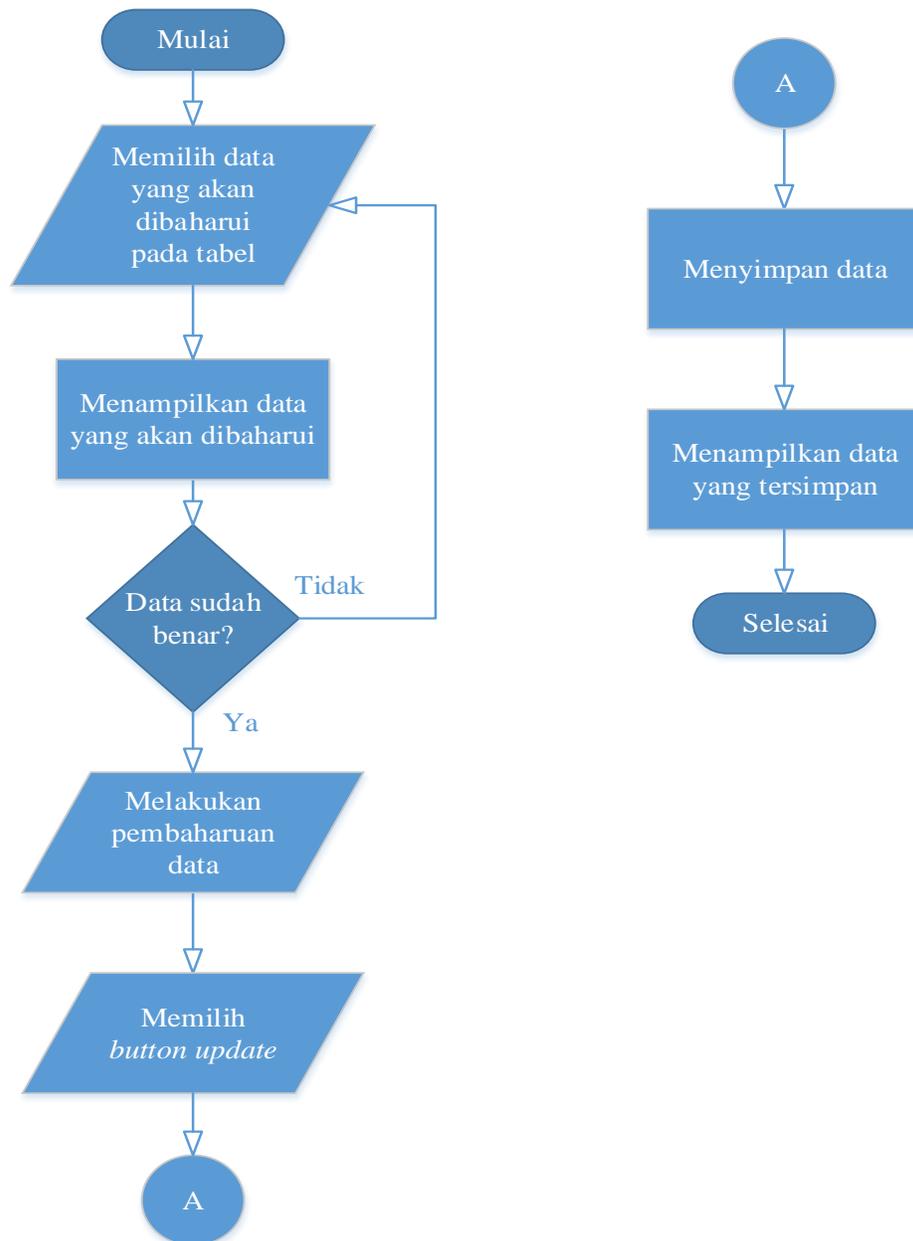
Gambar perancangan prosedur penambahan data dapat dilihat pada Gambar 3.14.



**Gambar 3.14** *Flow Chart* Prosedural Penambahan Data

#### 4. Rancangan Prosedur Pembaharuan Data

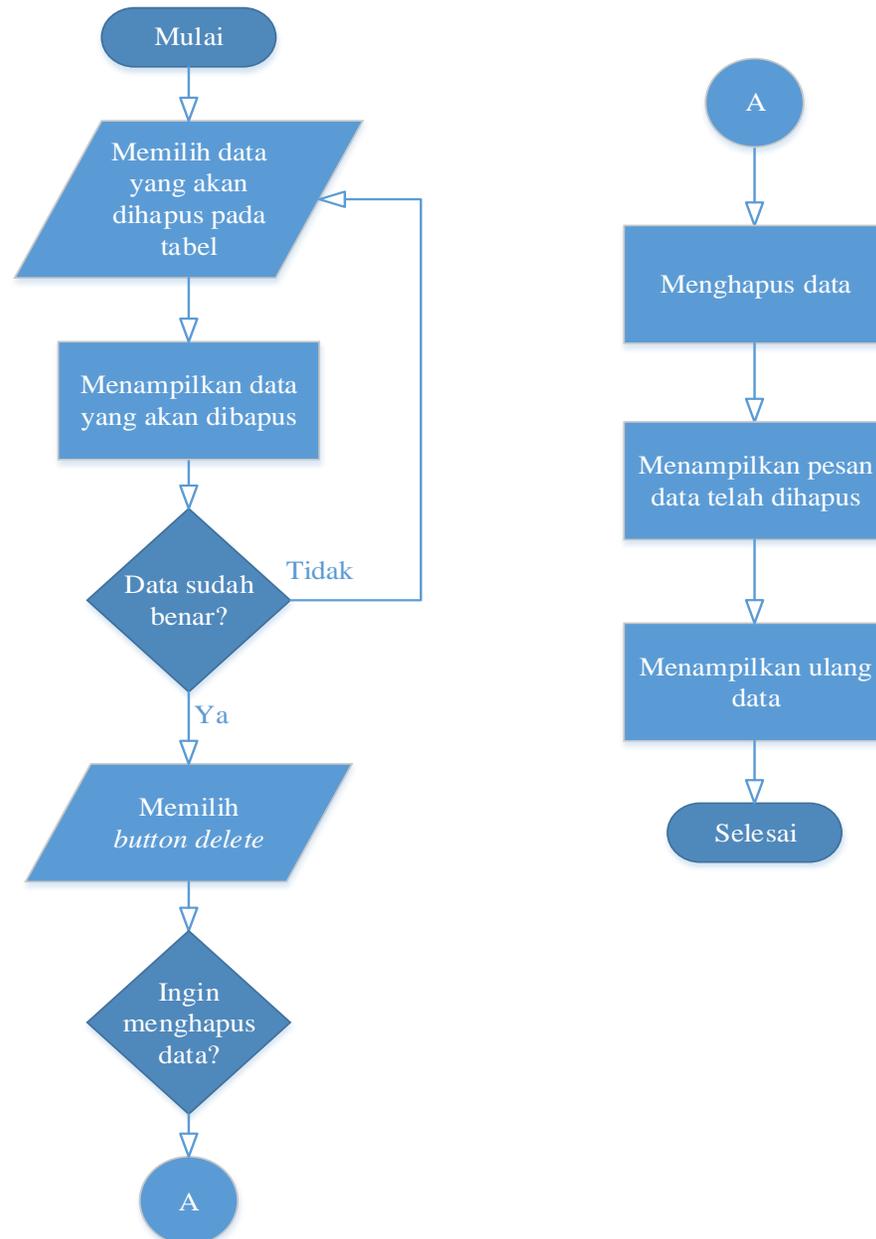
Gambar perancangan prosedur pembaharuan data dapat dilihat pada Gambar 3.15.



**Gambar 3.15** Flow Chart Prosedural Pembaharuan Data

## 5. Rancangan Prosedur Penghapusan Data

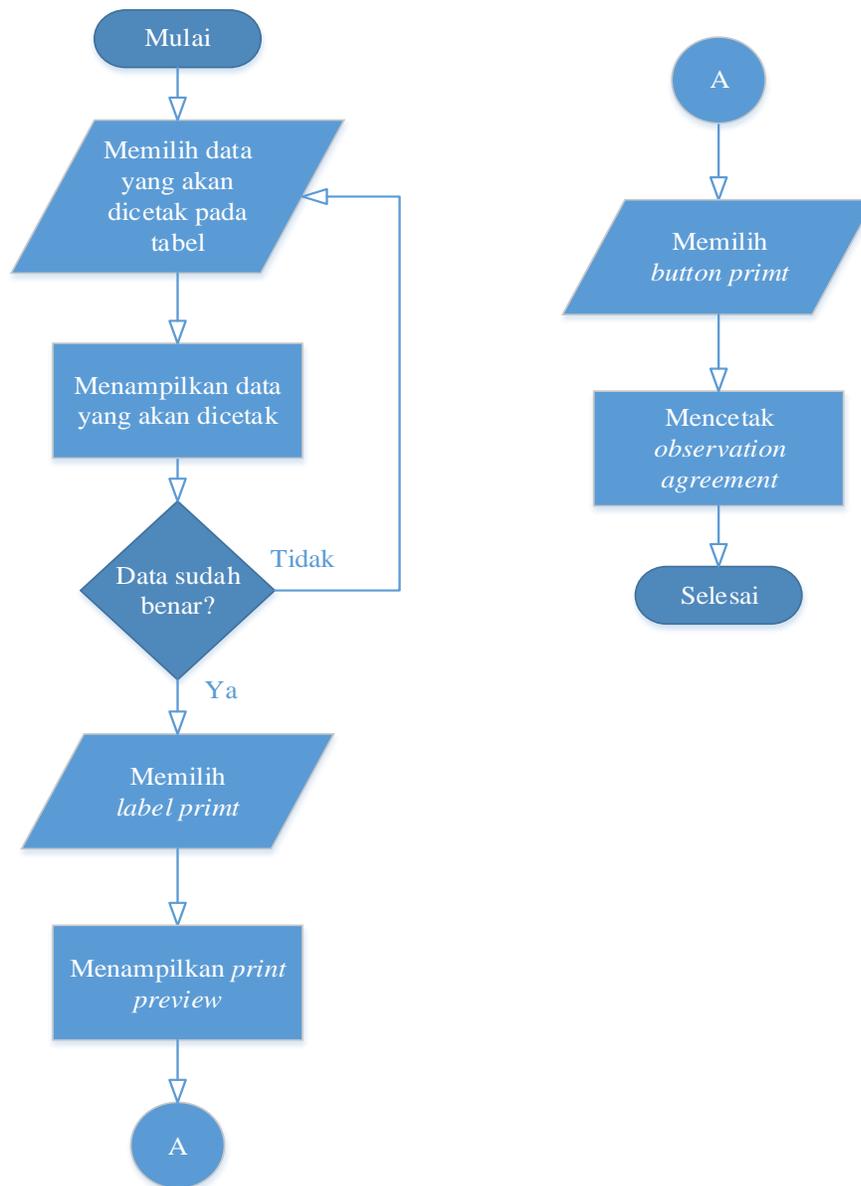
Gambar perancangan prosedur penghapusan data dapat dilihat pada Gambar 3.16.



**Gambar 3.16** Flow Chart Prosedural Penghapusan Data

## 6. Rancangan Prosedur Pencetakan *Observation Agreement*

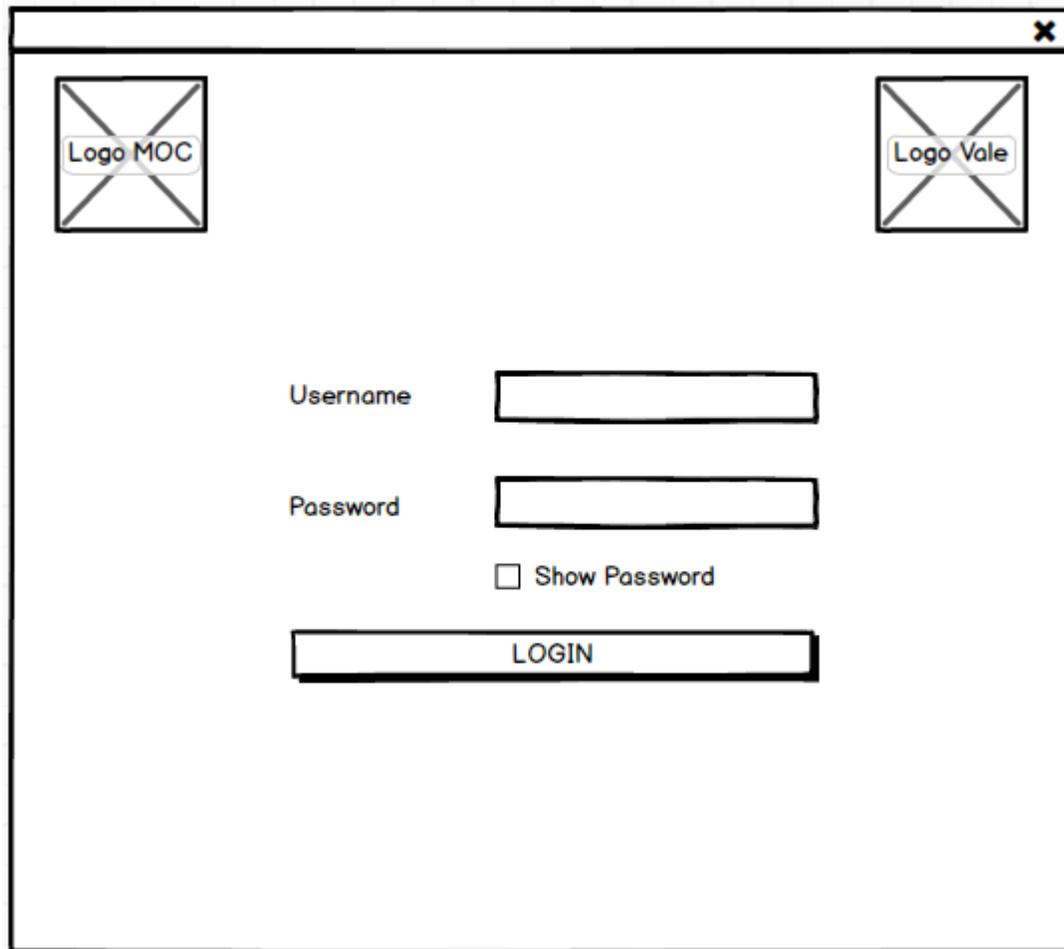
Gambar perancangan prosedur pencetakan *observation agreement* dapat dilihat pada Gambar 3.17.



**Gambar 3.17** Flow Chart Prosedural Pencetakan *Observation Agreement*

### 3.5.7. Rancangan *Interface*

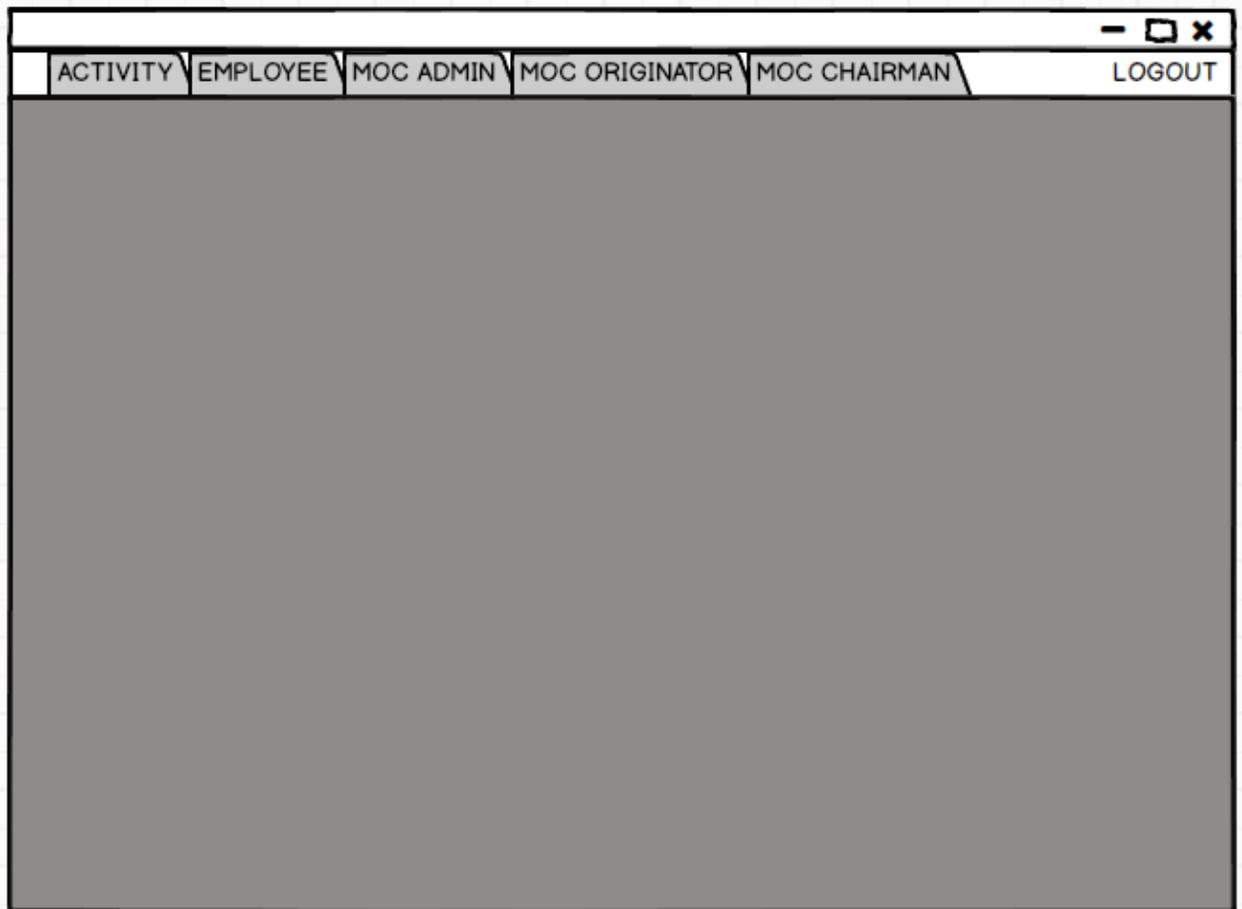
*Interface* merupakan bagian penting dari suatu aplikasi karena merupakan bagian dari *software* yang menjadi sarana komunikasi antar *user* dan *system*.



The image shows a wireframe of a login interface within a window. The window has a title bar with a close button (X) in the top right corner. Inside the window, there are two placeholder boxes for logos: 'Logo MOC' on the left and 'Logo Vale' on the right. Below the logos, there are two text input fields. The first is labeled 'Username' and the second is labeled 'Password'. Below the password field, there is a checkbox labeled 'Show Password'. At the bottom of the form, there is a large button labeled 'LOGIN'.

**Gambar 3.18** Rancangan *Interface* Halaman *Login*

Gambar 3.18 adalah rancangan *interface* untuk halaman *login*. Pada halaman ini terdapat dua *textbox* untuk *input username* dan *password*, dan satu *button* untuk *login*, serta *checkbox* untuk *show password*.



**Gambar 3.19** Rancangan *Interface* Halaman Menu

Gambar 3.19 adalah rancangan *interface* untuk halaman *menu*. Pada halaman ini terdapat lima menu yaitu *Activity*, *Employee*, *MOC Admin*, *MOC Originator*, dan *MOC Chairman*, serta satu *label* untuk *logout*.

The image shows a web interface titled "ACTIVITY". At the top left, there is a search bar with a magnifying glass icon and the text "By ID Activity or Severity". Below the search bar is a data table with four columns and four rows, all of which are currently empty. Below the table is a form with three input fields: "ID Activity" (a text box), "Activity" (a text box), and "Severity" (a dropdown menu). At the bottom of the form are three buttons: "SAVE", "DELETE", and "REFRESH".

Gambar 3.20 Rancangan *Interface* Halaman *Activity*

Gambar 3.20 adalah rancangan *interface* untuk halaman *activity*. Pada halaman ini terdapat dua *textbox* dan satu *combobox* yang berfungsi untuk mengisi data, terdapat tiga *button* yaitu *save*, *delete*, dan *refresh*, terdapat satu *button update* yang akan muncul ketika telah memilih data yang akan dibaharui, dan terdapat satu *textbox* yang berfungsi untuk mencari data, serta terdapat *datagridview* yang berfungsi untuk menampilkan data *activity* yang ada.

## EMPLOYEE

Search

ID Employee	<input type="text"/>
Name	<input type="text"/>
Department	<input type="text"/>
Position	<input type="text"/>
Username	<input type="text"/>
Password	<input type="text"/>
Role	<input type="text"/>

Gambar 3.21 Rancangan *Interface* Halaman *Employee*

Gambar 3.21 adalah rancangan *interface* untuk halaman *employee*. Pada halaman ini terdapat lima *textbox* dan dua *combobox* yang berfungsi untuk mengisi data, terdapat tiga *button* yaitu *save*, *delete*, dan *refresh*, terdapat satu *button update* yang akan muncul ketika telah memilih data yang akan dibaharui, dan terdapat satu *textbox* yang berfungsi untuk mencari data, serta terdapat *datagridview* yang berfungsi untuk menampilkan data *employee* yang ada.

**MANAGEMENT OF CHANGE**

Search


ID MOC	<input type="text"/>	<input type="button" value="UPDATE"/>
ID Employee	<input type="text"/>	<input type="button" value="DELETE"/>
Name	<input type="text"/>	<input type="button" value="REFRESH"/>
Department	<input type="text"/>	
Position	<input type="text"/>	
Activity	<input type="text"/>	
Severity	<input type="text"/>	
Frequency	<input type="text"/>	
Location	<input type="text"/>	
Request Date	<input type="text" value="/ /"/> 	
Reason	<input type="text"/>	
Observation Status	<input type="text"/>	

Observation Agreement 

**Gambar 3.22** Rancangan *Interface* Halaman *MOC Admin*

Gambar 3.22 adalah rancangan *interface* untuk halaman *MOC admin*. Pada halaman ini terdapat enam *textbox*, lima *combobox*, dan satu *datetimepicker* yang berfungsi untuk mengisi data, terdapat tiga *button* yaitu *update*, *delete*, dan *refresh*, terdapat satu *picturebox* untuk menampilkan gambar, dan terdapat satu *textbox* yang berfungsi untuk mencari data, serta terdapat *datagridview* yang berfungsi untuk menampilkan data *MOC* yang ada.

**MANAGEMENT OF CHANGE**

Search


ID MOC

ID Employee

Name

Department

Position

Activity

Severity

Frequency

Location

Request Date

Reason

[Click here to print](#)

[Click here to upload](#)

Step 1: Choose File

Step 2: Click the button below

**Gambar 3.23** Rancangan *Interface* Halaman *MOC Originator*

Gambar 3.23 adalah rancangan *interface* untuk halaman *MOC originator*. Pada halaman ini terdapat enam *textbox*, empat *combobox*, dan satu *datetimepicker* yang berfungsi untuk mengisi data, terdapat dua *button* yaitu *save* dan *refresh*, terdapat dua *label* yang berfungsi untuk aksi *print* dan *upload*, ketika memilih *label upload*, maka akan muncul satu *textbox browse* dan dua *button* untuk fungsi *browse* dan *upload file*, dan terdapat satu *textbox* yang berfungsi untuk mencari data, serta terdapat *datagridview* yang berfungsi untuk menampilkan data *MOC* yang ada.

The image shows a web interface titled "MANAGEMENT OF CHANGE". At the top, there is a search bar with the text "Search" and a magnifying glass icon, followed by "By ID MOC or MOC Status". Below the search bar is a data grid with four columns and three rows. Underneath the grid is a form with the following fields:

- ID MOC:
- ID Employee:
- Name:  (dropdown arrow)
- Department:
- Position:
- Activity:
- Severity:  (dropdown arrow)
- Frequency:  (dropdown arrow)
- Location:  (dropdown arrow)
- Request Date:  (calendar icon)
- Reason:
- MOC Status:  (dropdown arrow)
- Decision Date:  (calendar icon)

To the right of the form are two buttons: "UPDATE" and "REFRESH".

**Gambar 3.24** Rancangan *Interface* Halaman *MOC Chairman*

Gambar 3.24 adalah rancangan *interface* untuk halaman *MOC chairman*. Pada halaman ini terdapat enam *textbox*, lima *combobox*, dan dua *datetimepicker* yang berfungsi untuk mengisi data, terdapat dua *button* yaitu *update* dan *refresh*, dan terdapat satu *textbox* yang berfungsi untuk mencari data, serta terdapat *datagridview* yang berfungsi untuk menampilkan data *MOC* yang ada.

OBSERVATION AGREEMENT  
MANAGEMENT OF CHANGE  
PT VALE INDONESIA TBK

--	--	--	--

Sign

Observer Name

**Gambar 3.25** Rancangan *Interface* Halaman Report

Gambar 3.25 adalah rancangan *interface* untuk halaman *report* untuk menampilkan *preview* data *observation agreement* sebelum dicetak.

### 3.6. Pengujian

Metode pengujian yang dipakai dalam pengembangan aplikasi adalah pengujian fungsionalitas dan *interview*. Pengujian fungsionalitas yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsionalitas dari *software*. Hal-hal yang menjadi perhatian dalam pengujian aplikasi “*Management of Change*” adalah sebagai berikut:

1. Halaman *Login* Aplikasi
  - a. Saat *user login*, aplikasi dapat memberi petunjuk atau pesan jika terjadi kesalahan *username* atau *password*.
  - b. Aplikasi akan mengarahkan *user* ke dalam halaman menu, jika *user* berhasil *login* atau jika *username* dan *password* benar.
2. Halaman *Activity*
  - a. Aplikasi dapat menyimpan data *activity* yang diisi oleh *user*.
  - b. Aplikasi dapat membaharui data *activity* yang telah ada.
  - c. Aplikasi dapat menghapus data *activity* yang telah ada.
  - d. Aplikasi dapat menampilkan data *activity* yang telah tersimpan.
3. Halaman *Employee*
  - a. Aplikasi dapat menyimpan data *employee* yang diisi oleh *user*.
  - b. Aplikasi dapat membaharui data *employee* yang telah ada.
  - c. Aplikasi dapat menghapus data *employee* yang telah ada.
  - d. Aplikasi dapat menampilkan data *employee* yang telah tersimpan.

4. Halaman *MOC Admin*
  - a. Aplikasi dapat membaharui data *MOC Admin* yang telah ada.
  - b. Aplikasi dapat menghapus data *MOC Admin* yang telah ada.
  - c. Aplikasi dapat menampilkan data yang telah tersimpan.
5. Halaman *MOC Originator*
  - a. Aplikasi dapat menyimpan data *MOC Originator* yang diisi oleh *user*.
  - b. Aplikasi dapat menampilkan data yang telah tersimpan.
  - c. Aplikasi dapat mencetak *observation agreement* dari data *MOC Originator*.
  - d. Aplikasi dapat mengunggah *observation agreement* pada data *MOC Originator*.
6. Halaman *MOC Admin*
  - a. Aplikasi dapat membaharui data *MOC Originator* yang telah ada.
  - b. Aplikasi dapat menampilkan data yang telah tersimpan.