

## **BAB 3**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1. Analisis Sistem**

Analisis sistem merupakan bagian yang menjelaskan konsep sistem secara utuh baik dari mengidentifikasi masalah, kebutuhan sistem dan alur dari sistem yang akan dibangun.

##### **3.1.1. Analisis Masalah**

Analisis masalah yang terdapat di divisi CodeLabs saat ini adalah sebagai berikut :

1. Belum adanya proses penilaian terkait kehadiran harian anggota divisi CodeLabs.
2. Anggota Codelabs yang lain tidak memiliki akses keruangan CodeLabs ketika datang lebih dulu dari yang memiliki kunci.

##### **3.1.2. Analisis Sistem yang Sedang Berjalan**

Analisis sistem yang sedang berjalan merupakan tahapan yang bertujuan untuk memberikan gambaran lebih detail bagaimana cara kerja sistem yang sedang berjalan saat ini.

###### **3.1.2.1. Prosedur kehadiran anggota CodeLabs**

Merupakan prosedur untuk menentukan apakah anggota divisi CodeLabs hadir atau tidak. Prosesnya adalah sebagai berikut:

1. Anggota CodeLabs datang ke ruangan CodeLabs jam 9.
2. Anggota CodeLabs masuk ke ruangan dan melakukan kegiatan yang termasuk kedalam komponen utama KPI antara lain: Riset, Kompetisi, Operasional. Pusbok, dan Bisnis.
3. Anggota Codelabs yang melakukan kegiatan terkait KPI maka akan dianggap hadir.
4. Jika anggota CodeLabs tidak hadir maksimal 3 maka akan mendapatkan teguran dari ketua divisi CodeLabs.

### 3.1.2.2. Prosedur Akses Ruangan CodeLabs

Merupakan prosedur yang dimulai dari awal anggota CodeLabs datang sampai masuk ke ruangan CodeLabs. Prosesnya adalah sebagai berikut :

1. Anggota CodeLabs datang ke ruangan CodeLabs.
2. Sampai di ruangan anggota mengecek apakah ruangan sudah terbuka.
3. Jika ruangan belum terbuka maka harus meminta kunci ke office boy/girl atau menunggu anggota yang memegang kunci datang dan membuka ruangan.
4. Jika ruangan sudah terbuka maka bisa masuk dan memulai kegiatan yang termasuk kedalam komponen utama KPI.

### 3.1.2.3. Batasan terkait kehadiran dan akses ruangan CodeLabs

Adapun batasan-batasan yang terdapat adalah sebagai berikut :

1. Anggota CodeLabs datang sesuai jam kerja CodeLabs yaitu dari jam 09-00WIB sampai 17.00WIB.
2. Anggota CodeLabs melakukan kegiatan terkait sub komponen KPI.
3. Anggota CodeLabs yang bisa membuka ruangan hanya yang biasa datang pagi atau pulang terakhir karena memiliki kunci ruangan.
4. Ruangan CodeLabs hanya untuk anggota CodeLabs.

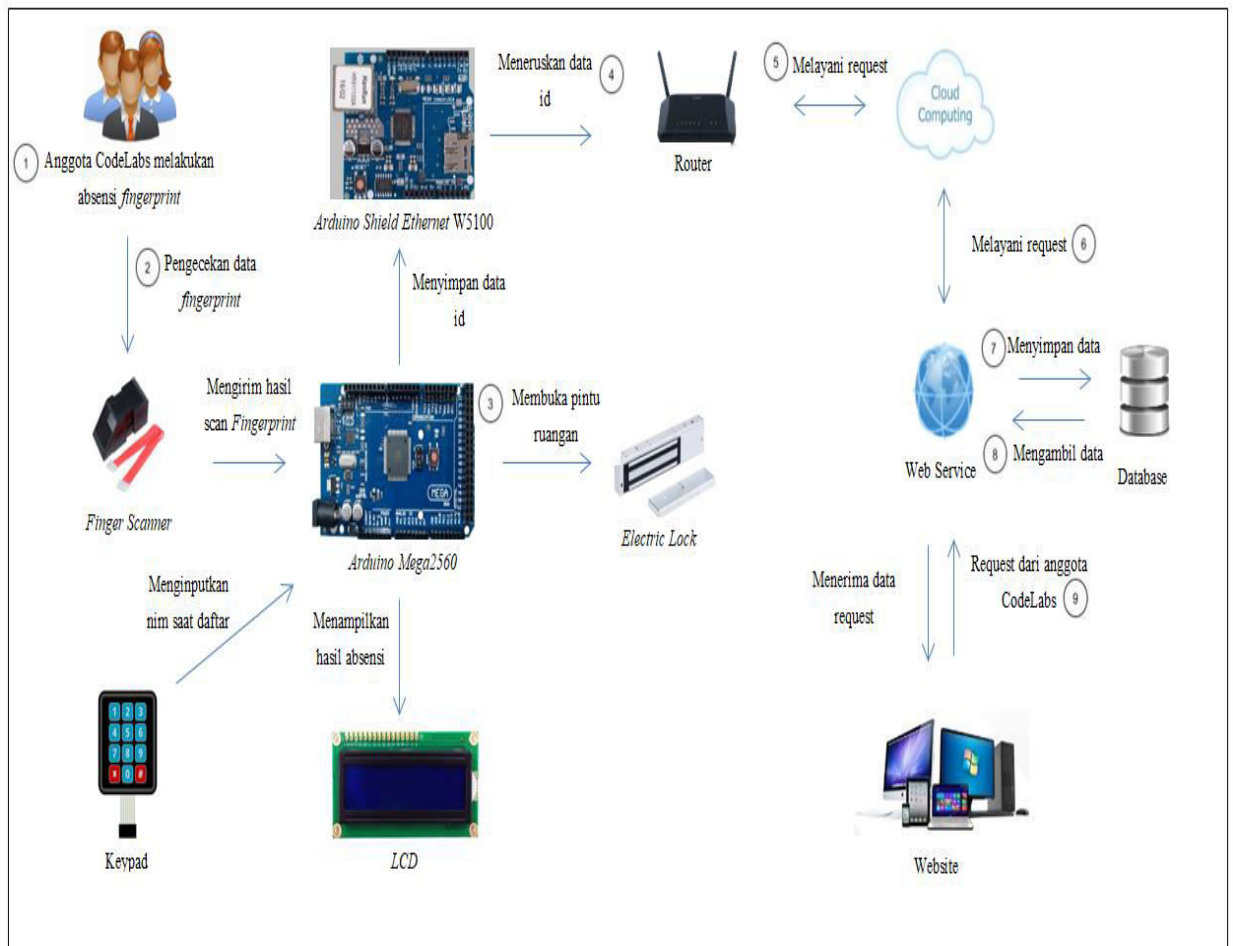
### 3.1.2.4. Evaluasi Sistem yang Sedang Berjalan

**Tabel 3.1 Tabel Evaluasi Sistem yang Sedang Berjalan**

No	Evaluasi Prosedur Akses Ruangan dan Kehadira CodeLabs
1	Anggota CodeLabs harus datang keruangan CodeLabs dari jam 09:00-17:00 WIB
2	Anggota CodeLabs harus melaksanakan kegiatan terkait sub komponen KPI
3	Anggota CodeLabs akan dianggap hadir jika melaksanakan kegiatan terkait sub komponen KPI
4	Anggota CodeLabs yang lain tidak bisa membuka ruangan CodeLabs jika ruangan terkunci
5	Hanya anggota CodeLabs yang memiliki kunci ruangan yang bisa membuka ruangan CodeLabs
6	Anggota CodeLabs maksimal tidak hadir hanya 3 kali

### 3.1.3. Analisis Arsitektur Sistem

Pada bagian analisis arsitektur ini penulis akan menjelaskan bagaimana alur sistem yang akan dibangun, dan pada bagian ini penulis juga akan menjelaskan alat-alat yang digunakan dalam membangun sistem ini.



Gambar 3.1 Arsitektur Sistem

Berikut adalah penjelasan dari Gambar 3.2 sebagai berikut :

1. Anggota divisi CodeLabs yang datang akan melakukan kehadiran dengan scan *fingerprint*.
2. *FingerScanner* akan secara otomatis membaca dan mengidentifikasi *fingerprint* tersebut dengan data yang disimpan, kemudian mengirimkan data berupa id sidik jari ke *Arduino Mega2560*.

3. Setelah *Arduino Mega2560* mendapatkan id sidik jari *Arduino Mega2560* akan menentukan nim yang memiliki id sidik jari ini dan meneruskannya ke *Arduino Shield Ethernet W5100* agar bisa disimpan.
4. Nim di inputkan ketika mendaftarkan *fingerprint*.
5. Kemudian *Arduino Mega2560* akan memberi perintah membuka akses ruangan CodeLabs, menampilkan pesan ke lcd dan dikirim *web service* melalui *Arduino Shield Ethernet W5100*.
6. Operasional web masuk ke sistem berbasis web melalui browser yang tersedia yang terhubung ke internet.
7. Web service akan memproses request dengan mengakses data yang ada di database, dan kemudian data akan ditampilkan ke browser.

#### 3.1.4. Analisis Alat

Analisis alat merupakan bagian yang menjelaskan mengenai alat-alat yang digunakan dalam pembangunan sistem ini.

##### 3.1.4.1. Perangkat Masukan

1. FPM10A

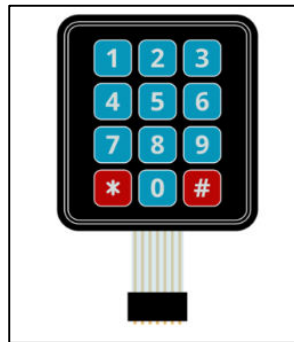
Fingerprint Modul FPM10A merupakan perangkat masukan pada sistem ini, FPM10A ini digunakan untuk mendeteksi, membaca, menyimpan dan mencocokkan pola sidik jari anggota CodeLabs kemudian memberikan id untuk setiap sidik jari dan mengirimnya ke *Arduino Mega2560*.



Gambar 3.2 *Fingerprint Modul FPM10A*

## 2. Keypad

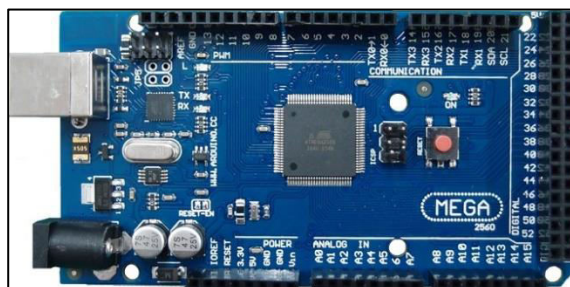
Keypad ini digunakan untuk memberi masukan berupa nim dan id, perangkat ini digunakan saat mendaftarkan fingerprint, dan juga untuk menentukan kepemilikan id sidik jari yang diberikan oleh *fingerscanner*.



Gambar 3.3 Keypad

### 3.1.4.2. Perangkat Proses

Arduino Mega2560 merupakan perangkat yang digunakan untuk pemrosesan data yang diterima *fingerscanner* yaitu berupa id sidik jari. *Arduino Mega2560* akan menentukan apakah id sidik jari yang diterima valid atau tidak, jika id sidik jari tidak valid maka *Arduino Mega2560* akan memberikan keterangan berupa warna lampu dan menampilkan pesan ke lcd “Absensi Gagal Ulangi!“, sedangkan jika id sidik jari valid maka *Arduino Mega2560* akan memberikan keterangan lampu, mengirimnya ke *Shield Ethernet W5100* agar bisa disimpan dan di kirimkan ke web service, kemudian menampilkan pesan ke lcd “ Absensi Berhasil”, ”ID : nomor id fingerprint yang di scan“, dan “Nama : nama anggota CodeLabs yang di scan sidikjarinya”, dan terakhir membuka akses ruangan CodeLabs.



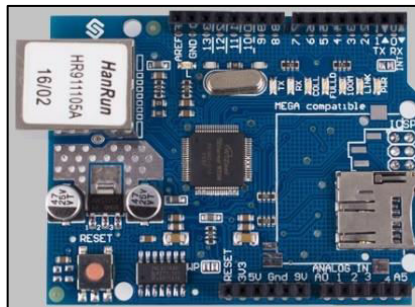
Gambar 3.4 Arduino Mega2560

### 3.1.4.3. Perangkat Output

Perangkat output merupakan perangkat yang digunakan untuk menerima data yang telah di olah dan diproses pada perangkat proses.

#### 1. Arduino Shield Ethernet W51000

Merupakan perangkat output yang digunakan untuk mengirimkan data yang didapat dari perangkat proses ke web service dengan menggunakan LAN, dan juga jika terjadi masalah koneksi data akan disimpan di micro sd.



**Gambar 3.5** *Arduino Shield Ethernet W51000*

#### 2. Elektrik Lock

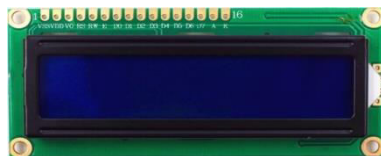
Merupakan perangkat yang digunakan untuk menjadi akses kontrol pintu di CodeLabs, perangkat akan menerima perintah langsung dari *Arduino Mega2560*.



**Gambar 3.6** *Electric lock*

### 3. LCD

Merupakan perangkat output yang digunakan untuk menampilkan pesan hasil scan fingerprint yang diterima oleh *Arduino Mega2560* apakah data yang diterima valid atau tidak valid.



**Gambar 3.7 LCD**

#### 3.1.5. Analisis Fingerprint

Merupakan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan didalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. Dengan kata lain flowchart ini merupakan deskripsi secara grafik dari urutan prosedur-prosedur yang terkombinasi membentuk suatu sistem.

##### 1. Mendaftarkan Anggota CodeLabs

Merupakan alur kerja atau prosedur-prosedur saat mendaftarkan anggota Codelabs. Berikut adalah penjelasan dari setiap prosedur-prosedur saat mendaftarkan anggota CodeLabs:

###### 1) Melakukan Scan Fingerprint

Yaitu mengscan *fingerprint* anggota baru agar terdaftar didalam sistem, tetapi sebelumnya scan terlebih dahulu *fingerprint* admin setelah itu letakkan *fingerprint* anggota pada *fingerscanner*.

###### 2) Get Image

*Fingerscanner* akan mengambil gambar *fingerprint* yang hasilnya berupa image, kemudian image ini akan di konvert ke image to tanzama image.

###### 3) Get ID

Kemudian *fingerscanner* akan mengenerate id untuk fingerprint yang discan saat mendaftar.

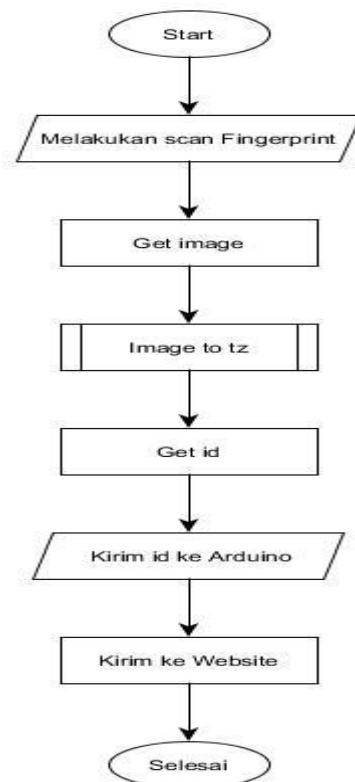
## 4) Kirim ID Ke Arduino

ID yang di generate oleh *fingerscanner* akan dikirim ke arduino untuk pengimputan nim id sidik jari tersebut.

## 5) Kirim Ke Website

ID yang telah diolah oleh arduino akan dikirim ke website melalui jaringan internet dan kemudian operasional CodeLabs akan memasukkan identitas id sidik jari tersebut.

Berikut adalah flowchar mendaftarkan anggota CodeLabs.



**Gambar 3.8 Flowchart Mendaftarkan Anggota CodeLabs**

## 2. Absensi Anggota CodeLabs Dan Akses Ruang CodeLabs

Merupakan alur kerja atau prosedur-prosedur saat melakukan absensi harian anggota CodeLabs dan untuk akses ruangan CodeLabs. Berikut adalah penjelasan prosedur-prosedur nya:



1) Melakukan Scan Fingerprint

Yaitu mengscan *fingerprint* anggota CodeLabs saat datang ke ruangan CodeLabs tetapi data *fingerprint* sudah tersimpan didalam sistem.

2) Get Image

*Fingerscanner* akan mengambil gambar *fingerprint* yang hasilnya berupa image, kemudian image ini akan di konvert ke image to tanzama image.

3) Get ID

Kemudian *fingerscanner* akan mengenerate id *fingerprint* yang discan.

4) Pengecekan Dan Pengiriman Ke Arduino

Setelah itu akan dilakukan pengecekan id apakah id tersebut terdaftar atau tidak dan hasil dari pengecekan itu akan dikirim ke arduino beserta id sidik jarinya.

5) Tampil Pesan Ke LCD

LCD akan menampilkan pesan dari pengecekan id yang dilakukan pada tahap sebelumnya jika id valid mana akan muncul pesa “Absensi Berhasil”, “ID: sesuai data “, dan “Nama: sesuai data “

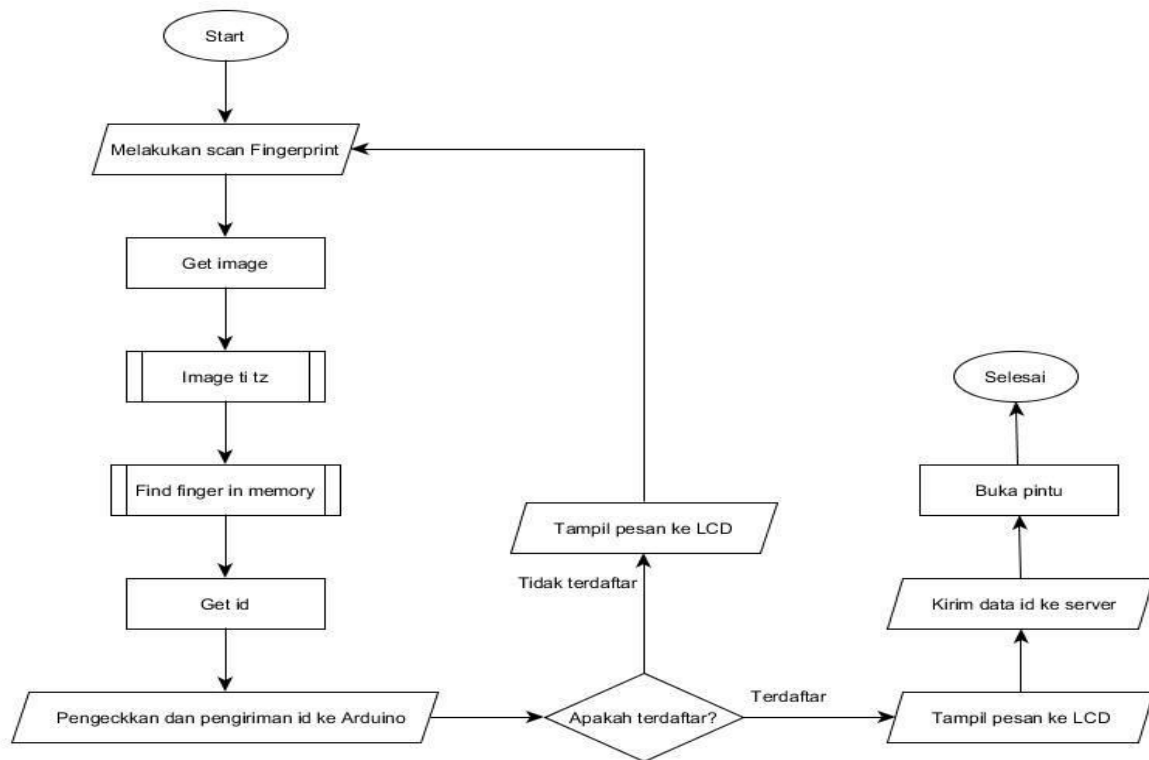
6) Kirim Data Ke Server

Data yang didapatkan arduino akan dikirim ke server melalui koneksi internet melalui LAN yang terpasang pada Ethernet Shield.

7) Buka Pintu

Dan yang terakhir sistem akan membuka akses ruangan CodeLabs sesuai dengan perintah dari arduino.

Berikut adalah flowchart absensi dan akses ruangan CodeLabs.



Gambar 3.9 Flowchart Absensi Harian Anggota CodeLabs

### 3.1.6. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan ini menentukan kebutuhan masukan yang diperlukan sistem, keluaran sistem, dan proses yang dibutuhkan untuk mengolah suatu masukan sehingga menghasilkan keluaran yang diinginkan. Adapun beberapa bagian termasuk kebutuhan non fungsional yaitu, analisis kebutuhan perangkat keras, analisis kebutuhan perangkat lunak dan analisis pengguna.

#### 3.1.6.1. Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras dilakukan untuk mengetahui spesifikasi perangkat keras yang diusulkan untuk menjalankan sistem. Spesifikasi perangkat keras yang digunakan dalam sistem ini adalah sebagai berikut :

## Perangkat Keras Yang Dimiliki Programmer

**Tabel 3.2 Perangkat Keras yang Dibutuhkan**

No	Perangkat Keras	Spesifikasi
1	Processor	Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz (4 CPUs), ~2.5GHz
2	Resolusi Monitor	Resolusi monitor 1366 x 768
3	RAM	4 GB
4	Hardisk	500GB
5	Koneksi Internet	Ya

### 3.1.6.2. Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak dilakukan adalah untuk mengetahui apa saja perangkat lunak yang digunakan untuk menghubungkan sistem dengan perangkat keras. Perangkat lunak yang digunakan adalah sebagai berikut :

**Tabel 3.3 Tabel Analisis Perangkat Lunak**

No	Perangkat Lunak	Kegunaan
1	XAMPP	Aplikasi untuk mensimulasikan web server guna menguji coba source code yang berkaitan dengan pembangunan web secara lokal.
2	Sublime Text	Text Editor untuk mengedit sintaks pemrograman web.
3	Mozilla Firefox	Aplikasi yang digunakan programmer untuk menampilkan hasil pemrograman web.

### 3.1.6.3. Analisis Pengguna

Analisis pengguna bertujuan untuk menentukan mengetahui pengguna yang akan berinteraksi dengan sistem, pengguna yang akan menggunakan website dan hak akses yang didapatkan oleh pengguna aplikasi.

**Tabel 3.4 Tabel Analisis Pengguna**

No	Pengguna	Tugas	Hak Akses
1	Anggota CodeLabs	- Melakukan kehadiran sesuai dengan jadwal yang telah ditentukan	- Membuka ruangan CodeLabs
2	Operasional CodeLabs	- Mengexport laporan absensi dan menginput data anggota CodeLabs	- Mengolah data yang ada di <i>website</i>

### 3.1.7. Analisis Spesifikasi Kebutuhan Perangkat Lunak

Pada tahapan analisis spesifikasi kebutuhan perangkat lunak terdapat dua spesifikasi kebutuhan, yaitu spesifikasi kebutuhan perangkat lunak fungsional dan kebutuhan perangkat lunak non fungsional. Spesifikasi kebutuhan fungsional merupakan spesifikasi kebutuhan mengenai fasilitas untuk sistem yang harus disediakan. Spesifikasi dapat dilihat pada tabel berikut ini.

**Tabel 3.5 Spesifikasi Kebutuhan Fungsional**

Kode SKPL	Spesifikasi Kebutuhan Fungsional
SKPL-F-01	Sistem <i>website</i> menyediakan fasilitas login
SKPL-F-02	Sistem <i>website</i> menyediakan fasilitas untuk melihat dan menambahkan daftar anggota CodeLabs
SKPL-F-03	Sistem <i>website</i> menyediakan fasilitas tambah dan edit data anggota CodeLabs
SKPL-F-04	Sistem <i>website</i> menyediakan fasilitas untuk melihat kehadiran anggota CodeLabs dan mengexport nya

Adapun spesifikasi kebutuhan non fungsional yaitu spesifikasi yang menyangkut mengenai batasan-batasan dari sistem seperti batasan waktu, batasan pengembangan proses standarisasi dan lain-lain. Spesifikasi kebutuhan non fungsional dapat dilihat pada tabel berikut ini.

**Tabel 3.6 Spesifikasi Kebutuhan Non Fungsional**

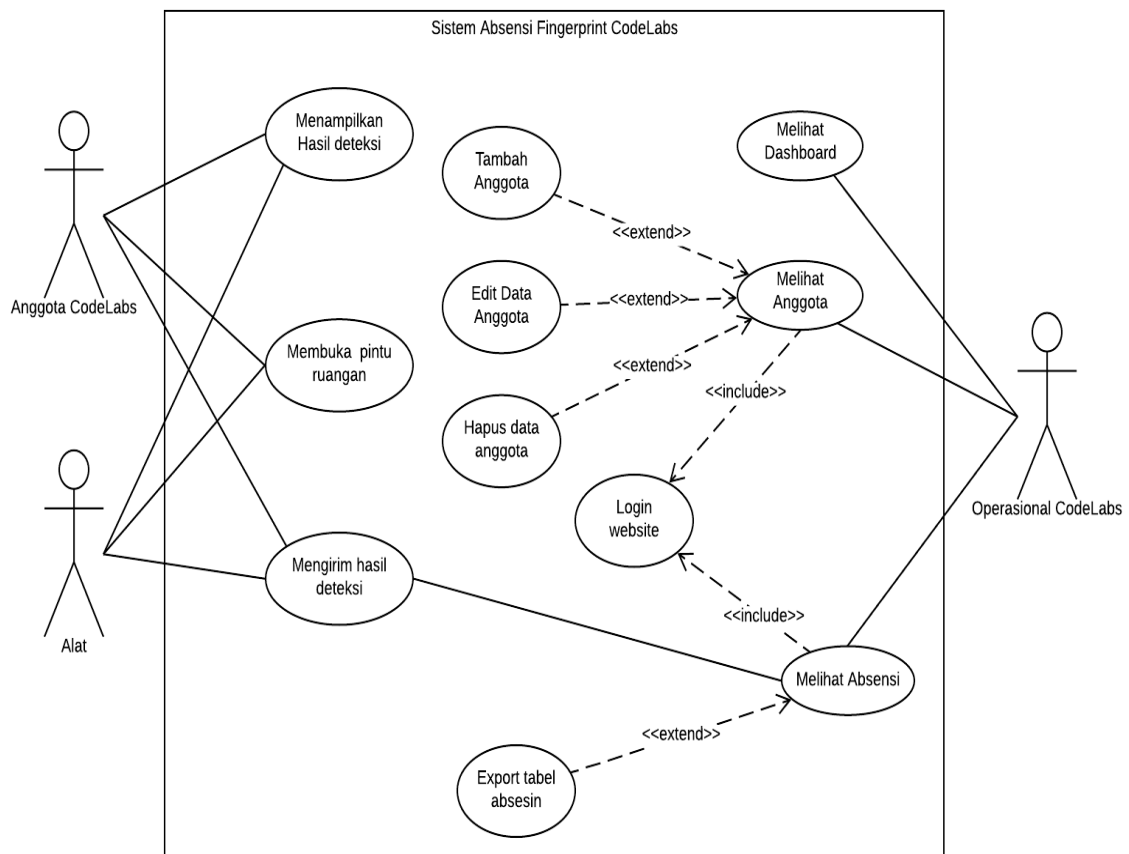
Kode SKPL	Spesifikasi Kebutuhan Fungsional
SKPL-NF-01	Sistem mampu mendeteksi fingerprint anggota CodeLabs
SKPL-NF-02	Sistem mampu mengakses ruangan CodeLabs
SKPL-NF-03	Sistem mampu menampilkan hasil kehadiran
SKPL-NF-04	Sistem mampu mengirimkan data melalui jaringan internet

### 3.1.8. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sistem dan menjelaskan kebutuhan yang diperlukan agar sistem dapat belajar dengan baik sesuai dengan kebutuhan sistem.

### 3.1.8.1. Use Case Diagram

*Use case diagram* adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar dan menjelaskan sistem secara fungsional yang terlihat pengguna.



**Gambar 3.10 Use Case Diagram Absensi CodeLabs Berbasis IoT**

### 3.1.8.2. Use Case Scenario

*Use Case Scenario* bertujuan untuk mendeskripsikan langkah-langkah dalam proses bisnis baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor, dijelaskan sebagai berikut.

1. *Use Case Scenario* Menampilkan Hasil Deteksi *Fingerprint*

**Tabel 3.7 Use Case Menampilkan Hasil Deteksi Fingerprint**

<i>Use Case Name</i>	Menampilkan Hasil Deteksi
<i>Related Requirement</i>	SKPL-NF-03
<i>Goal in Content</i>	Untuk menampilkan hasil deteksi fingerprint

<i>Precondition</i>	Anggota CodeLabs belum melakukan deteksi fingerprint	
<i>Successful End Condition</i>	Sistem menampilkan hasil deteksi fingerprint	
<i>Failed End Condition</i>	Sistem tidak menampilkan hasil deteksi fingerprint	
<i>Primary Actor</i>	Anggota CodeLabs	
<i>Secondary Actor</i>	Sistem	
<i>Trigger</i>	Anggota CodeLabs datang ke ruangan CodeLabs	
<i>Included Cases</i>	-	
<i>Main Flow</i>	Step	Action
	1	Anggota CodeLabs meletakkan jari di fingerscanner
	2	Sistem mencocokkan fingerprint yang dibaca dengan data yang ada didatabase
	3	Sistem menghidupkan lampu merah atau hijau
	4	Sistem menampilkan pesan hasil deteksi fingerprint ke LCD

## 2. Use Case Scenario Membuka Pintu Ruangan

**Tabel 3.8 Use Case Membuka Pintu Ruangan**

<i>Use Case Name</i>	Membuka Pintu Ruangan	
<i>Related Requirement</i>	SKPL-NF-02	
<i>Goal in Content</i>	Untuk membuka akses ruangan CodeLabs	
<i>Precondition</i>	Anggota CodeLabs belum melakukan deteksi fingerprint	
<i>Successful End Condition</i>	Anggota CodeLabs berhasil membuka ruangan	
<i>Failed End Condition</i>	Anggota CodeLabs gagal membuka ruangan	
<i>Primary Actor</i>	Anggota CodeLabs	
<i>Secondary Actor</i>	Sistem	
<i>Trigger</i>	Anggota CodeLabs datang ke ruangan CodeLabs	
<i>Included Cases</i>	-	
<i>Main Flow</i>	Step	Action
	1	Anggota CodeLabs meletakkan jari di fingerscanner
	2	Sistem mencocokkan fingerprint yang dibaca dengan data yang ada didatabase
	3	Sistem menghidupkan lampu hijau dan membuka akses ruangan CodeLabs
	4	Anggota CodeLabs masuk ke ruangan

### 3. Use Case Scenario Mengirim Hasil Deteksi Fingerprint

**Tabel 3.9 Use Case Mengirim Hasil Deteksi Fingerprint**

<i>Use Case Name</i>	Mengirim Hasil deteksi Fingerprint	
<i>Related Requirement</i>	SKPL-NF-04	
<i>Goal in Content</i>	Untuk menyimpan dan mengirim hasil deteksi fingerprint	
<i>Precondition</i>	Anggota CodeLabs belum melakukan deteksi fingerprint	
<i>Successful End Condition</i>	Sistem berhasil menyimpan dan mengirim hasil deteksi	
<i>Failed End Condition</i>	Sistem gagal menyimpan dan mengirim hasil deteksi	
<i>Primary Actor</i>	Anggota CodeLabs	
<i>Secondary Actor</i>	Sistem	
<i>Trigger</i>	Anggota CodeLabs datang ke ruangan CodeLabs	
<i>Included Cases</i>	-	
<i>Main Flow</i>	Step	Action
	1	Anggota CodeLabs meletakkan jari di fingerscanner
	2	Sistem mencocokkan fingerprint yang dibaca dengan data yang ada didatabase
	3	Sistem menghidupkan lampu hijau
	4	Sistem menyimpan dan mengirim hasil deteksi fingerprint ke web service

### 4. Use Case Scenario Login

Interaksi antara operasional CodeLabs dengan sistem dijelaskan pada *use case skenario login* berikut ini.

**Tabel 3.10 Use Case Scenario Login**

<i>Use Case Name</i>	Login	
<i>Related Requirement</i>	SKPL-F-01	
<i>Goal in Content</i>	Untuk memasuki halaman home website	
<i>Precondition</i>	Operasional CodeLabs belum melakukan login	
<i>Successful End Condition</i>	Operasional CodeLabs berhasil melakukan login	
<i>Failed End Condition</i>	Operasional CodeLabs gagal melakukan Login	
<i>Primary Actor</i>	Operasional CodeLabs	
<i>Secondary Actor</i>	Sistem	
<i>Trigger</i>	User mengakses halaman website	
<i>Included Cases</i>	-	
<i>Main Flow</i>	Step	Action
	1	Operasional CodeLabs memasukkan username dan password

	2	Operasional CodeLabs menekan tombol login
	3	Sistem melakukan verifikasi
	4	Sistem menampilkan halaman dashboard website
	Step	Branching action
<i>Extension</i>	3.1	Sisten menampilkan pesan kesalahan username dan password

### 5. Use Case Scenario Dashboard

Interaksi antara operasional CodeLabs dengan sistem dijelaskan pada *Use Case Scenario* dashboard.

**Tabel 3.11 Use Case Scenario Dashboard**

<i>Use Case Name</i>	Dashboard	
<i>Related Requirement</i>	SKPL-F-02	
<i>Goal in Content</i>	Untuk melihat halaman dashboard	
<i>Precondition</i>	Operasional CodeLabs memasuki halaman dashboard	
<i>Successful End Condition</i>	Operasional CodeLabs melihat tampilan halama dashboard	
<i>Failed End Condition</i>	Operasional CodeLabs tidak dapat melihat tampilan halama dashboard	
<i>Primary Actor</i>	Operasional CodeLabs	
<i>Secondary Actor</i>	Sistem	
<i>Trigger</i>	User mengklik menu dashboard	
<i>Included Cases</i>	Login	
<i>Main Flow</i>	Step	Action
	1	Operasional CodeLabs mengklik menu dashboard
	2	Sistem mengambil data ke database
	3	Sistem menampilkan halaman dashboard

### 6. Use Case Scenario Anggota

Interaksi antara operasional CodeLabs dengan sistem dijelaskan pada *Use Case Scenario* anggota.

**Tabel 3.12 Use Case Scenario Anggota**

<i>Use Case Name</i>	Anggota
<i>Related Requirement</i>	SKPL-F-02
<i>Goal in Content</i>	Untuk melihat daftar anggota CodeLabs
<i>Precondition</i>	Operasional CodeLabs memasuki halaman anggota
<i>Successful End Condition</i>	Operasional CodeLabs melihat tabel daftar anggota CodeLabs



<i>Failed End Condition</i>	Operasional CodeLabs tidak dapat melihat tabel daftar anggota CodeLabs	
<i>Primary Actor</i>	Operasional CodeLabs	
<i>Secondary Actor</i>	Sistem	
<i>Trigger</i>	User mengklik menu anggota	
<i>Included Cases</i>	login	
<i>Main Flow</i>	Step	Action
	1	Operasional CodeLabs mengklik menu anggota
	2	Sistem mengambil data ke database
	3	Sistem menampilkan halaman anggota

## 7. Use Case Scenario Tambah Data Anggota

Interaksi antara operasional CodeLabs dengan sistem dijelaskan pada Use Case Scenario tambah data anggota

**Tabel 3.13 Use Case Scenario Tambah Data Anggota**

<i>Use Case Name</i>	Tambah Data Anggota	
<i>Related Requirement</i>	SKPL-F-02	
<i>Goal in Content</i>	Untuk menambah daftar anggota CodeLabs	
<i>Precondition</i>	Operasional CodeLabs sudah melakukan login	
<i>Successful End Condition</i>	Operasional CodeLabs berhasil menambahkan daftar anggota CodeLabs	
<i>Failed End Condition</i>	Operasional CodeLabs tidak dapat menambahkan daftar anggota CodeLabs	
<i>Primary Actor</i>	Operasional CodeLabs	
<i>Secondary Actor</i>	Sistem	
<i>Trigger</i>	Operasional CodeLabs mengklik tombol tambah anggota	
<i>Included Cases</i>	Anggota	
<i>Main Flow</i>	Step	Action
	1	Operasional CodeLabs mengklik tombol tambah anggota
	2	Sistem menampilkan tampilan tambah anggota
	3	Operasional CodeLabs mengisi nim, nama, dan bidang keilmuan anggota dan mengklik tombol simpan
	4	Sistem menampilkan halaman daftar anggota yang sudah ditambahkan
<i>Extension</i>	Step	Branching action
	3.1	Sistem menampilkan pesan data berhasil disimpan

### 8. Use Case Scenario Edit Data Anggota

Interaksi antara operasional CodeLabs dengan sistem dijelaskan pada *Use Case Scenario* Edit Biodata.

**Tabel 3.14 Use Case Scenario Edit Data Anggota**

<i>Use Case Name</i>	Edit Data Anggota	
<i>Related Requirement</i>	SKPL-F-03	
<i>Goal in Content</i>	Untuk mengedit data anggota	
<i>Precondition</i>	Operasional CodeLabs memasuki halaman anggota	
<i>Successful End Condition</i>	Operasional CodeLabs berhasil mengedit data anggota	
<i>Failed End Condition</i>	Operasional CodeLabs gagal mengedit data anggota	
<i>Primary Actor</i>	Operasional CodeLabs	
<i>Secondary Actor</i>	Sistem	
<i>Trigger</i>	User menekan tombol icon edit	
<i>Included Cases</i>	Anggota	
<i>Main Flow</i>	Step	Action
	1	Operasional CodeLabs menekan tombol edit
	2	Sistem mengambil data dari database
	3	Sistem menampilkan tampilan edit data
	4	User mengedit data anggota dan menekan tombol simpan
	5	Sistem menyimpan data ke database
<i>Extension</i>	Step	Branching action
	5.1	Sistem menampilkan pesan data berhasil diupdate

### 9. Use Case Scenario Hapus Data Anggota

Interaksi antara operasional CodeLabs dengan sistem dijelaskan pada *Use Case Scenario* Hapus Data Anggota.

**Tabel 3.15 Use Case Scenario Hapus Data Anggota**

<i>Use Case Name</i>	Hapus Data Anggota
<i>Related Requirement</i>	SKPL-F-03
<i>Goal in Content</i>	Untuk menghapus data anggota
<i>Precondition</i>	Operasional CodeLabs memasuki halaman anggota
<i>Successful End Condition</i>	Operasional CodeLabs berhasil menghapus data anggota

<i>Failed End Condition</i>	Operasional CodeLabs gagal menghapus data anggota	
<i>Primary Actor</i>	Operasional CodeLabs	
<i>Secondary Actor</i>	Sistem	
<i>Trigger</i>	User menekan tombol icon hapus	
<i>Included Cases</i>	Anggota	
<i>Main Flow</i>	Step	Action
	1	Operasional CodeLabs menekan tombol hapust
	2	Sistem menghapus data amggota
	3	Sistem menampilkan halaman daftar anggota yang sudah dihapus
<i>Extension</i>	Step	Branching action
	3.1	Sistem menampilkan pesan data berhasil dihapus

#### 10. Use Case Scenario Absensi

Interaksi antara operasional CodeLabs dengan sistem dijelaskan pada *Use Case Scenario Absensi*.

**Tabel 3.16 Use Case Scenario Absensi**

<i>Use Case Name</i>	Absensi	
<i>Related Requirement</i>	SKPL-F-04	
<i>Goal in Content</i>	Untuk melihat halaman absensi	
<i>Precondition</i>	Operasional CodeLabs belum masuk ke halaman absensi	
<i>Successful End Condition</i>	Operasional CodeLabs melihat tabel absensi	
<i>Failed End Condition</i>	Operasional CodeLabs tidak dapat melihat tabel absensi	
<i>Primary Actor</i>	Operasional CodeLabs	
<i>Secondary Actor</i>	Sistem	
<i>Trigger</i>	Operasional CodeLabs mengklik menu absensi	
<i>Included Cases</i>	login	
<i>Main Flow</i>	Step	Action
	1	Operasional CodeLabs mengklik menu absensi
	2	Sistem mengambil data dari database
	3	Sistem menampilkan halaman absensi

#### 11. Use Case Scenario Export

Interaksi antara operasional CodeLabs dengan sistem dijelaskan pada *Use Case Scenario export*.

**Tabel 3.17 Use Case Scenario Export**

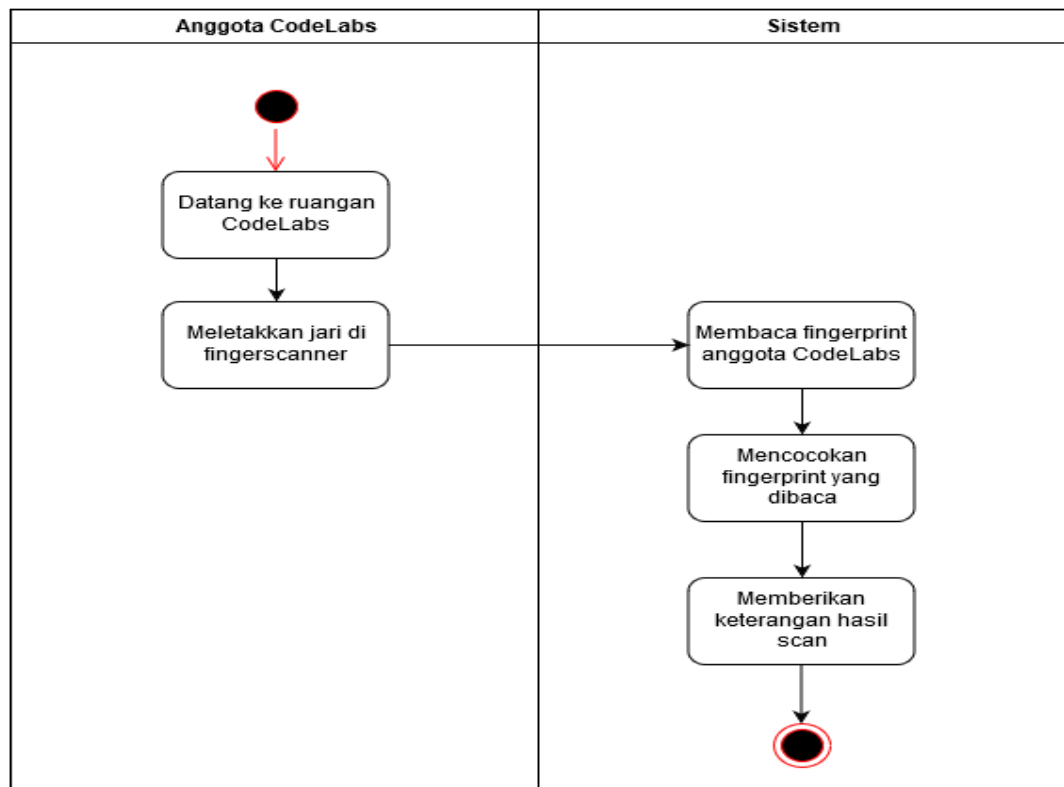
<i>Use Case Name</i>	Export tabel absensi	
<i>Related Requirement</i>	SKPL-F-04	
<i>Goal in Content</i>	Untuk mengexport tabel absensi	
<i>Precondition</i>	Operasional CodeLabs sudah masuk ke halaman absensi	
<i>Successful End Condition</i>	Operasional CodeLabs berhasil mengexport tabel absensi	
<i>Failed End Condition</i>	Operasional CodeLabs tidak berhasil mengexport tabel absensi	
<i>Primary Actor</i>	Operasional CodeLabs	
<i>Secondary Actor</i>	Sistem	
<i>Trigger</i>	Operasional CodeLabs mengklik tombol export	
<i>Included Cases</i>	login	
<i>Main Flow</i>	Step	Action
	1	Operasional CodeLabs mengklik tombol export
	2	Sistem mengambil data dari database
	3	Sistem menampilkan export data

### 3.1.8.3. Activity Diagram

*Activity diagram* memodelkan aliran kerja atau *workflow* urutan dari aktifitas dalam suatu proses yang mengacu pada *Use Case Diagram* yang ada.

#### 1. *Activity Diagram* Melakukan Deteksi *Fingerprint*

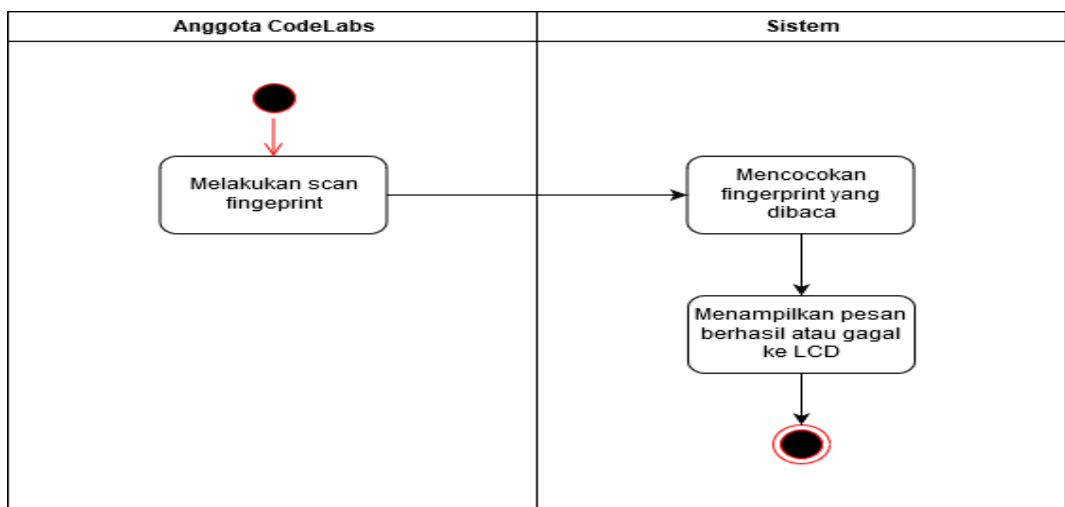
*Activity Diagram* melakukan deteksi *fingerprint* menjelaskan bagaimana proses melakukan deteksi *fingerprint*.



**Gambar 3.11 Activity Diagram Melakukan Deteksi Fingerprint**

2. *Activity Diagram Menampilkan Hasil Deteksi Fingerprint*

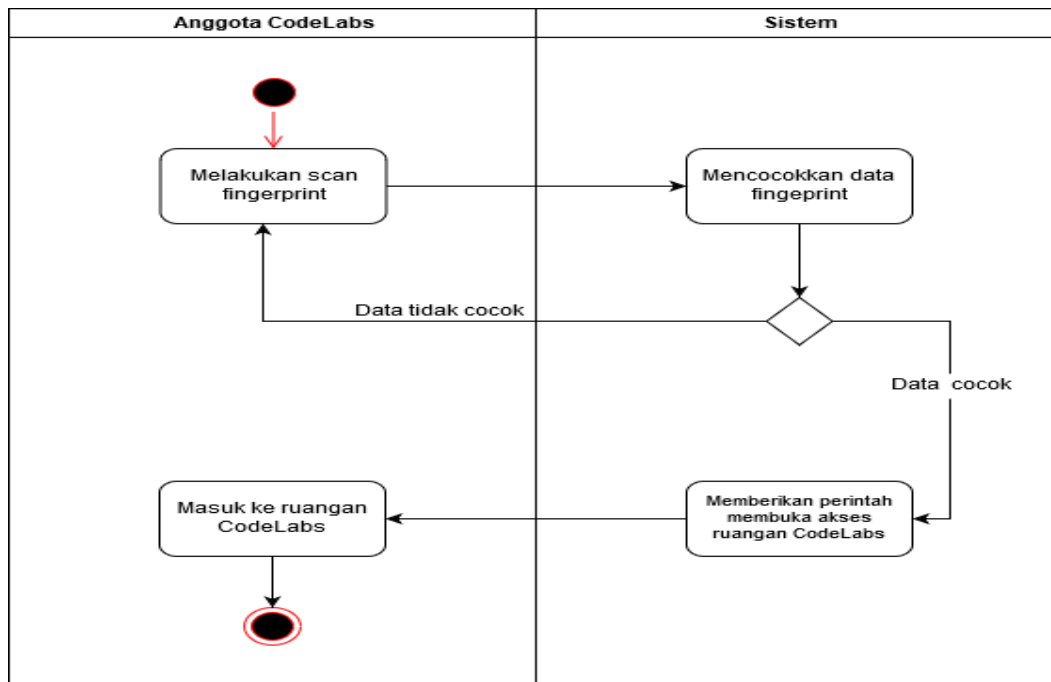
*Activity Diagram* menampilkan hasil deteksi *Fingerprint* menjelaskan bagaimana proses menampilkan deteksi ke LCD.



**Gambar 3.12 Activity Diagram Menampilkan Hasil Deteksi Fingerprint**

### 3. *Activity Diagram* Membuka Pintu Ruang

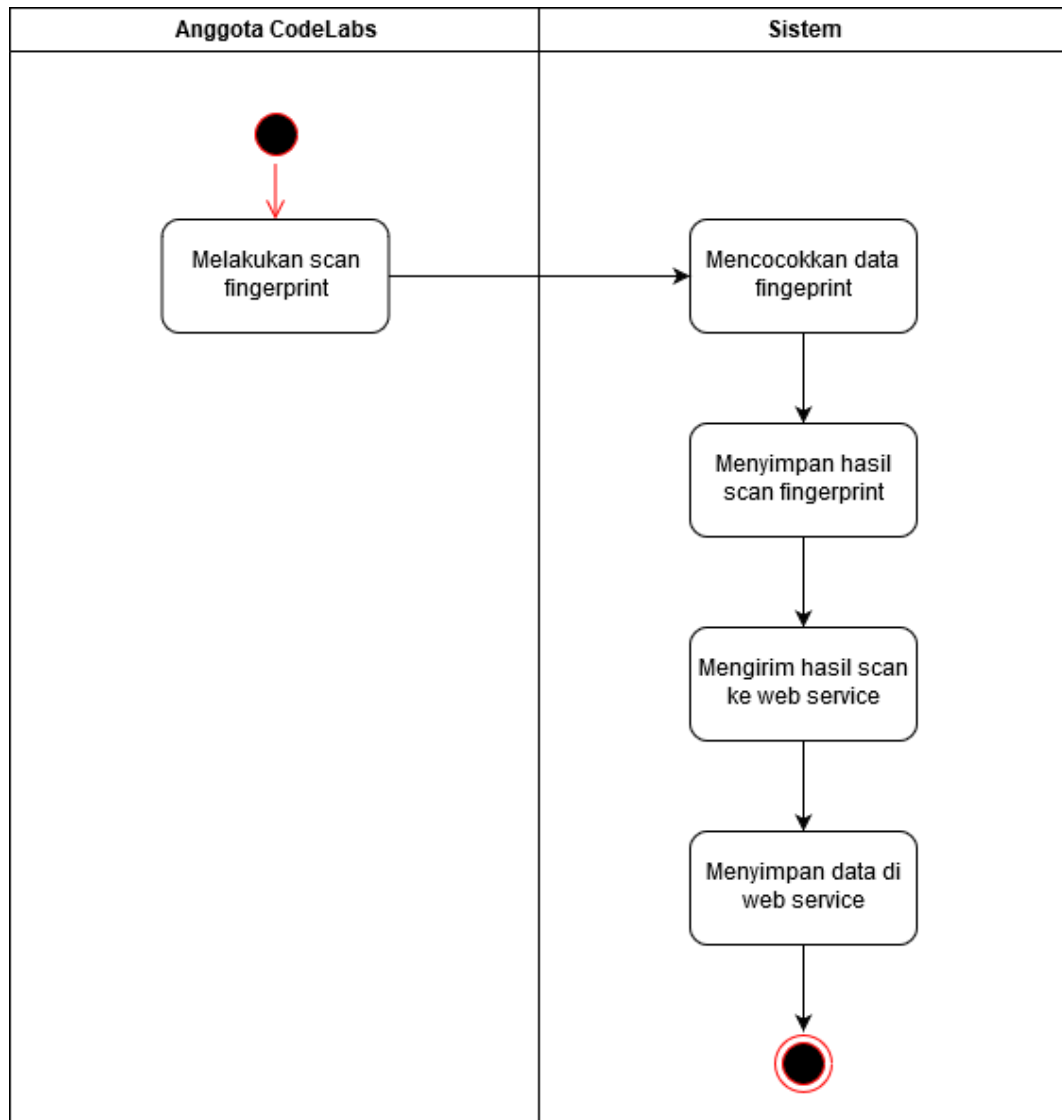
*Activity Diagram* membuka ruangan CodeLabs menjelaskan bagaimana proses akses ruangan CodeLabs.



**Gambar 3.13** Activity Diagram Membuka Pintu Ruang

### 4. *Activity Diagram* Mengirim Hasil Deteksi *Fingerprint*

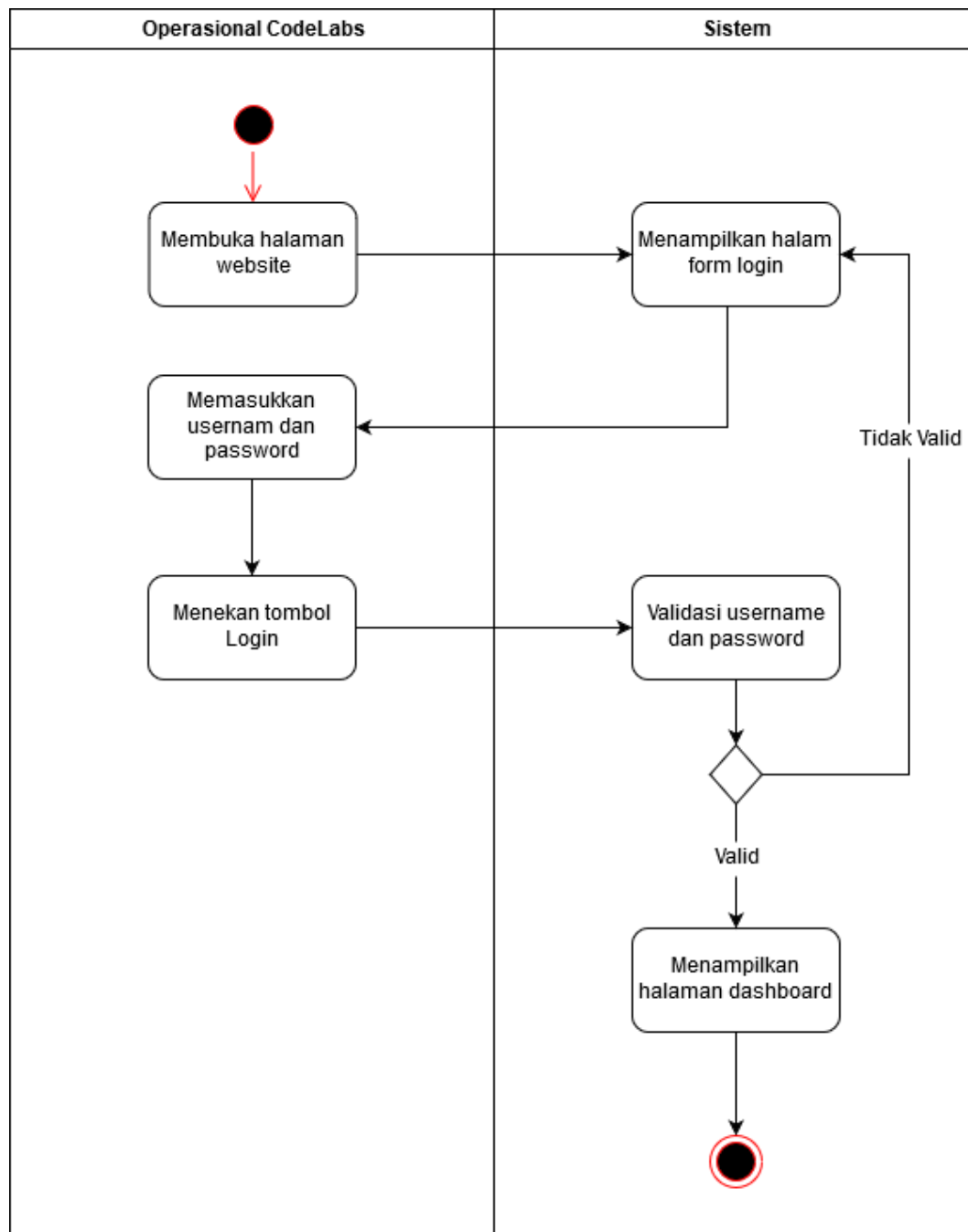
*Activity Diagram* mengirim hasil deteksi *Fingerprint* menjelaskan bagaimana proses menyimpan dan mengirim hasil deteksi ke web service.



**Gambar 3.14 Activity Diagram Mengirim Hasil Deteksi Fingerprint**

##### 5. *Activity Diagram* Login

*Activity Diagram* login menjelaskan bagaimana proses login dari awal sampai memasuki halaman dashboard website.

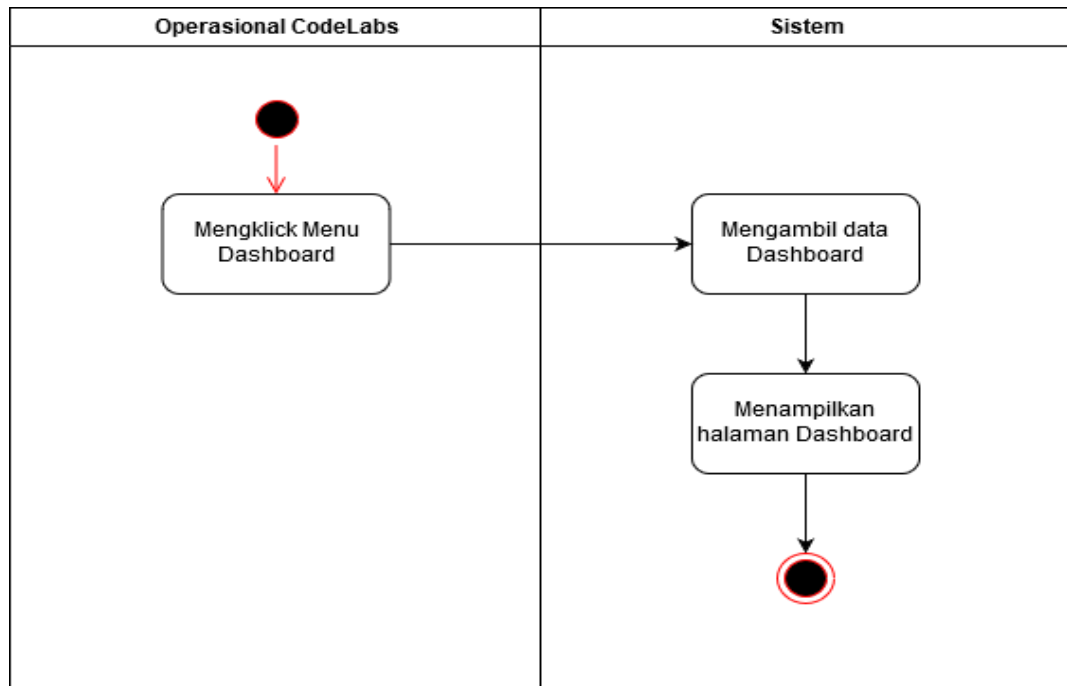


Gambar 3.15 Activity Diagram Login

6. Activity Diagram Dashboard.

Activity Diagram dashboard menjelaskan bagaimana proses melihat halaman Dashboard.

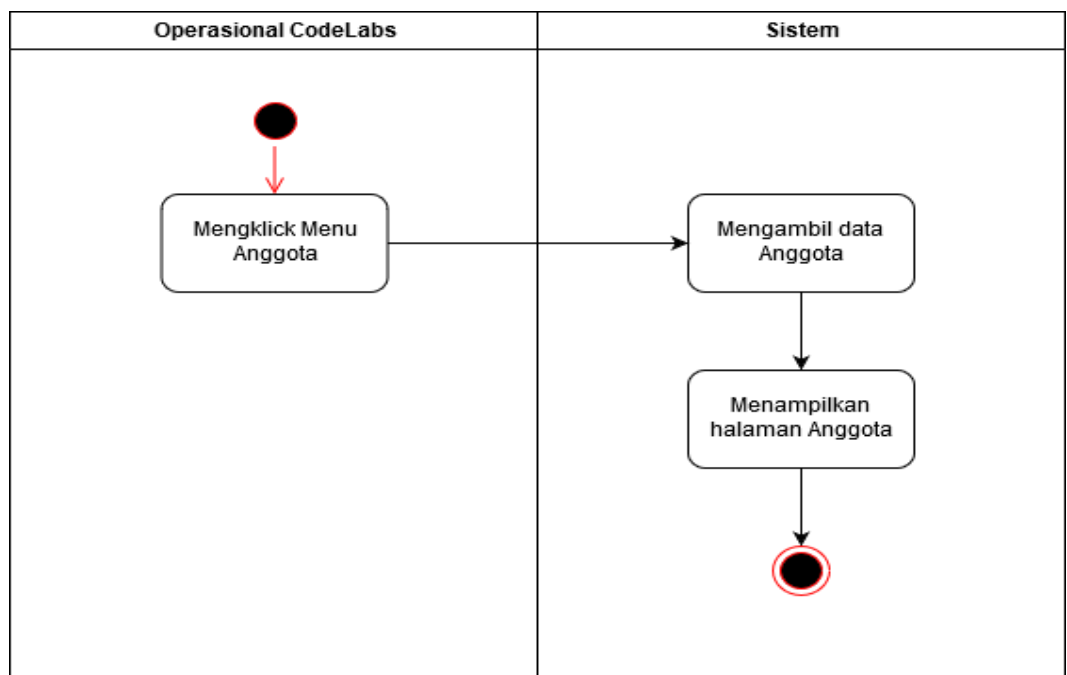




Gambar 3.16 Activity Diagram Dashboard

#### 7. Activity Diagram Anggota

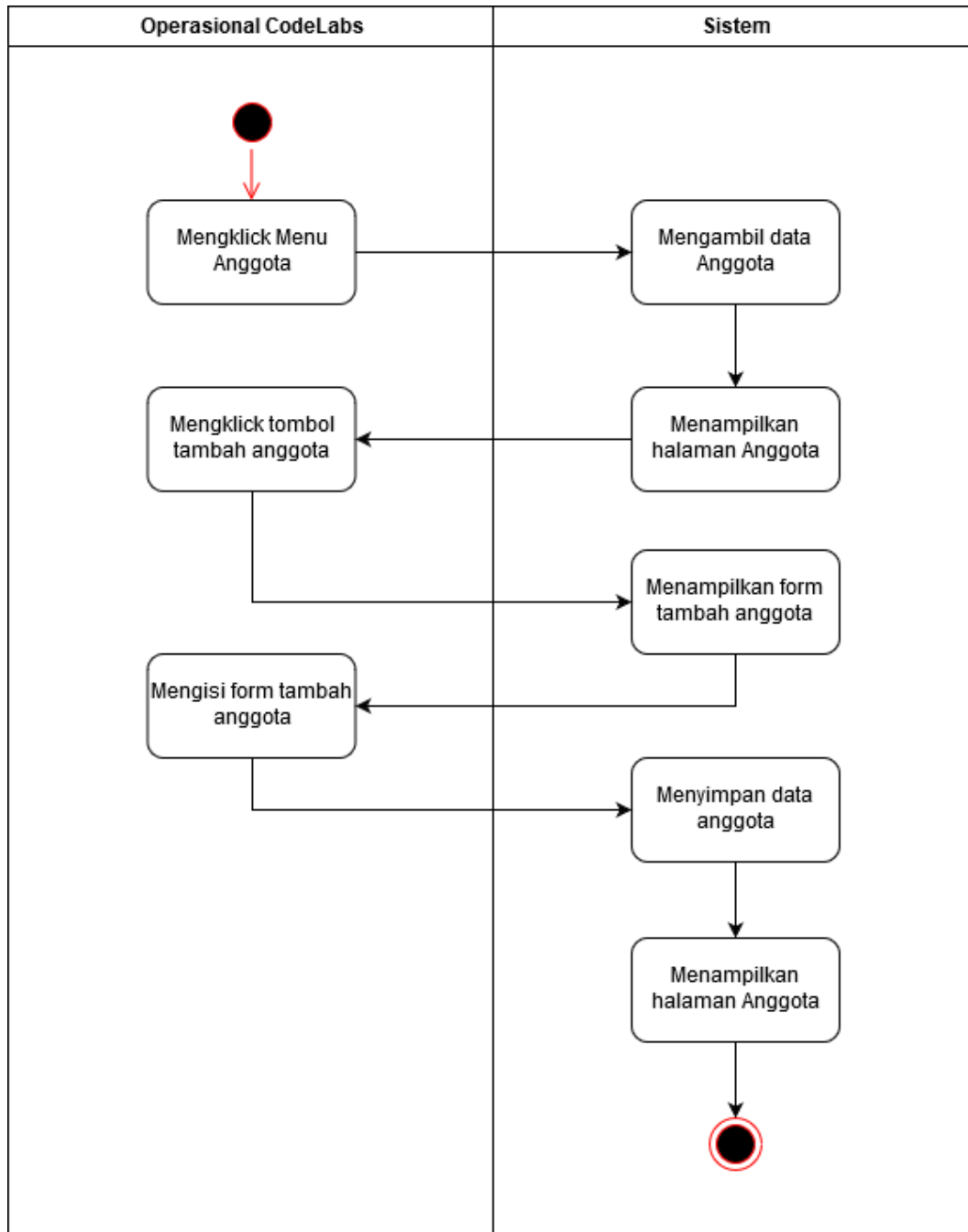
Activity Diagram anggota menjelaskan bagaimana proses menampilkan halaman anggota



Gambar 3.17 Activity Diagram Anggota

8. *Activity Diagram* Tambah Anggota

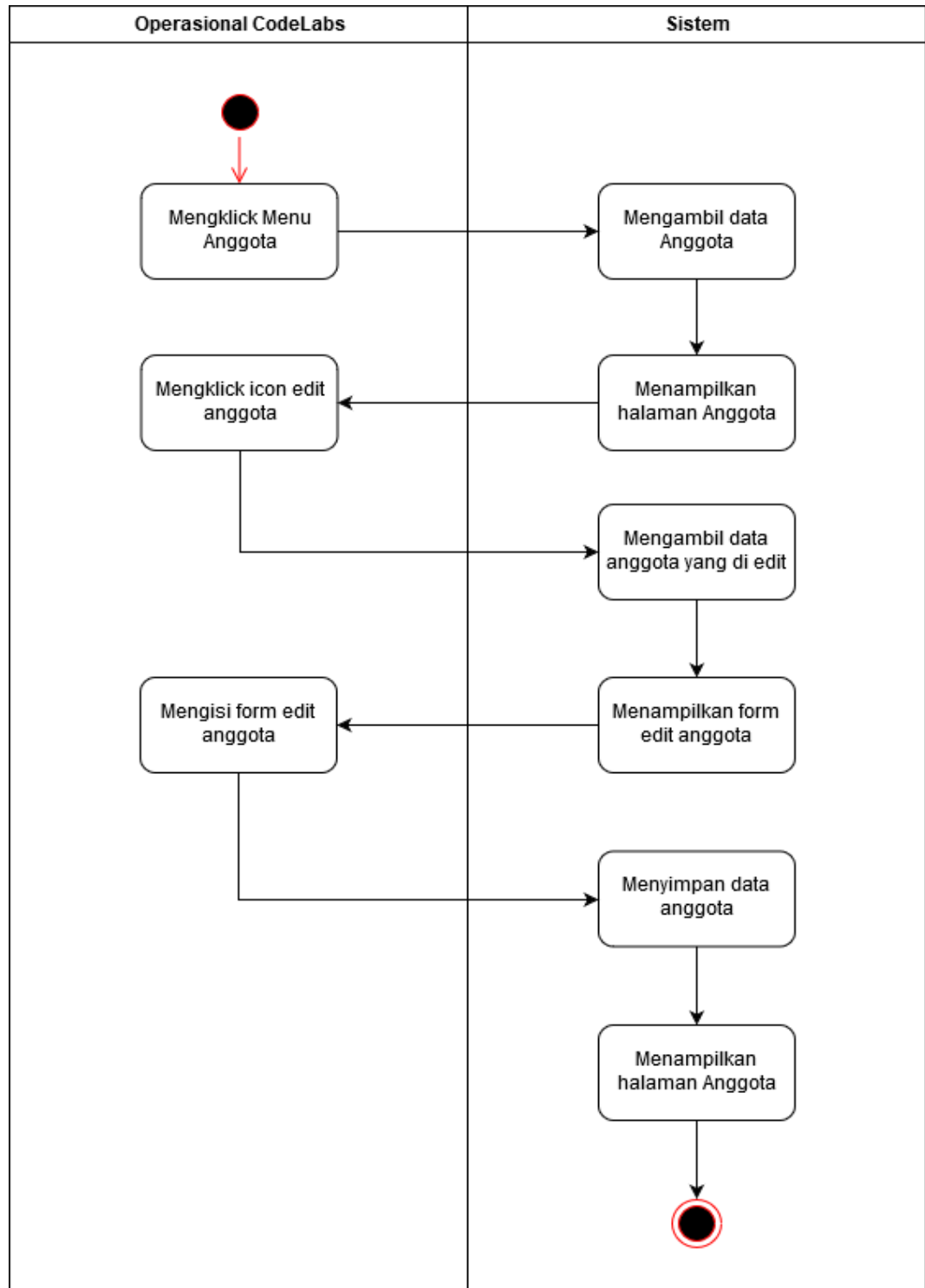
*Activity Diagram* tambah anggota menjelaskan bagaimana proses tambah anggota.



**Gambar 3.18** *Activity Diagram* Tambah Anggota

### 9. Activity Diagram Edit Data Anggota

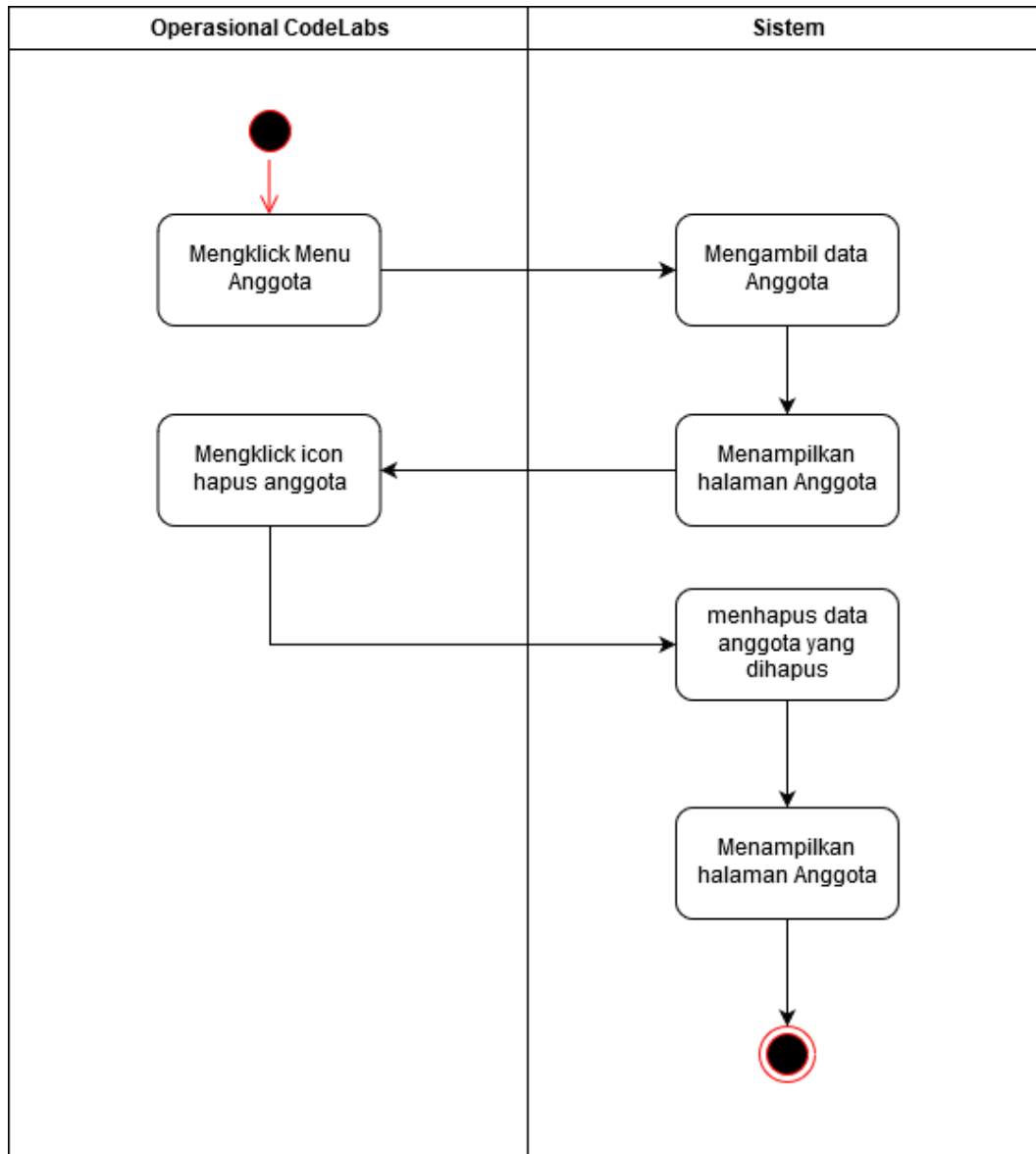
Activity Diagram edit data anggota menjelaskan bagaimana proses mengedit data anggota.



Gambar 3.19 Activity Diagram Edit Data Anggota

### 10. Activity Diagram Hapus Data Anggota

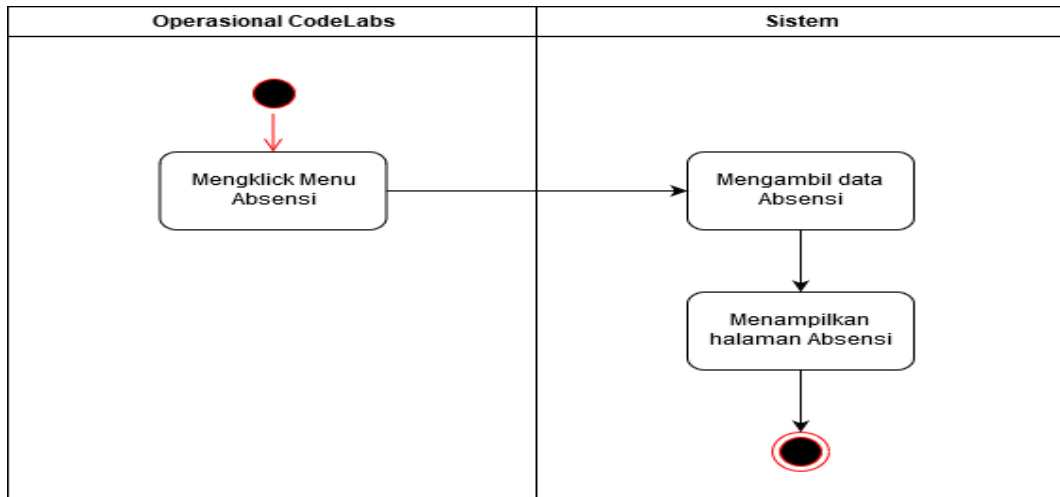
Activity Diagram hapus data anggota menjelaskan bagaimana proses menghapus data anggota CodeLabs.



Gambar 3.20 Activity Diagram Hapus Data Anggota

### 11. Activity Diagram Absensi

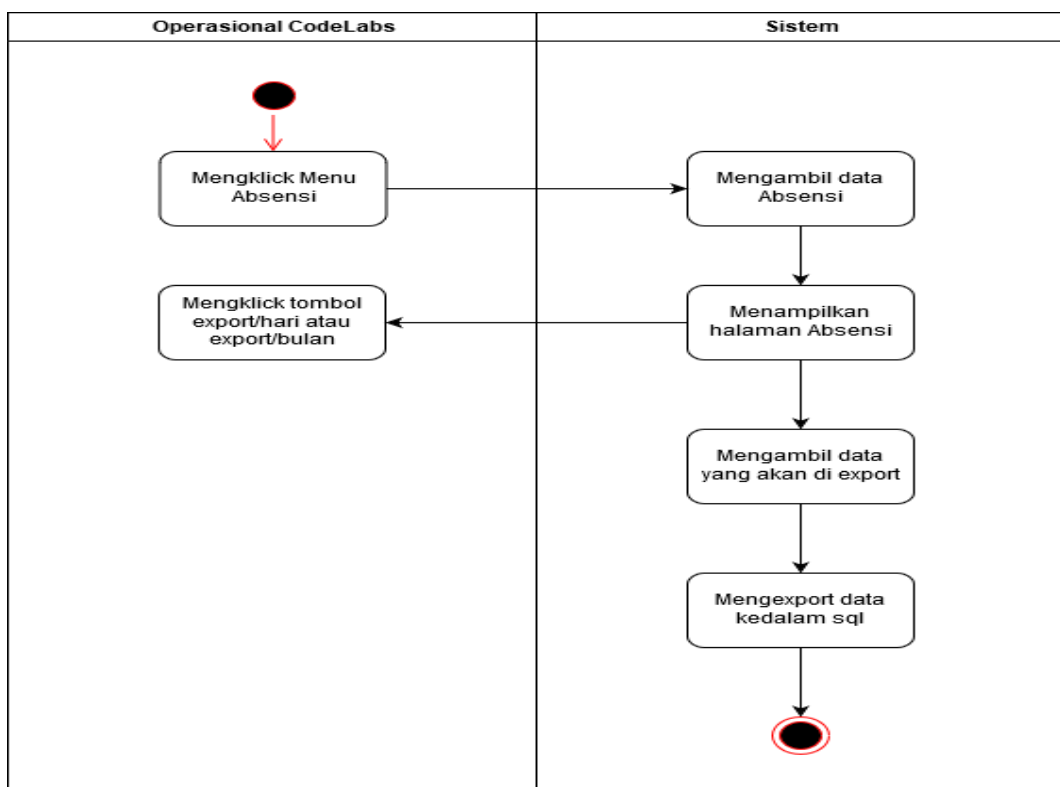
*Activity Diagram* Absensi menjelaskan bagaimana proses tampil tabel absensi



**Gambar 3.21 Activity Diagram Absensi**

### 12. Activity Diagram Export

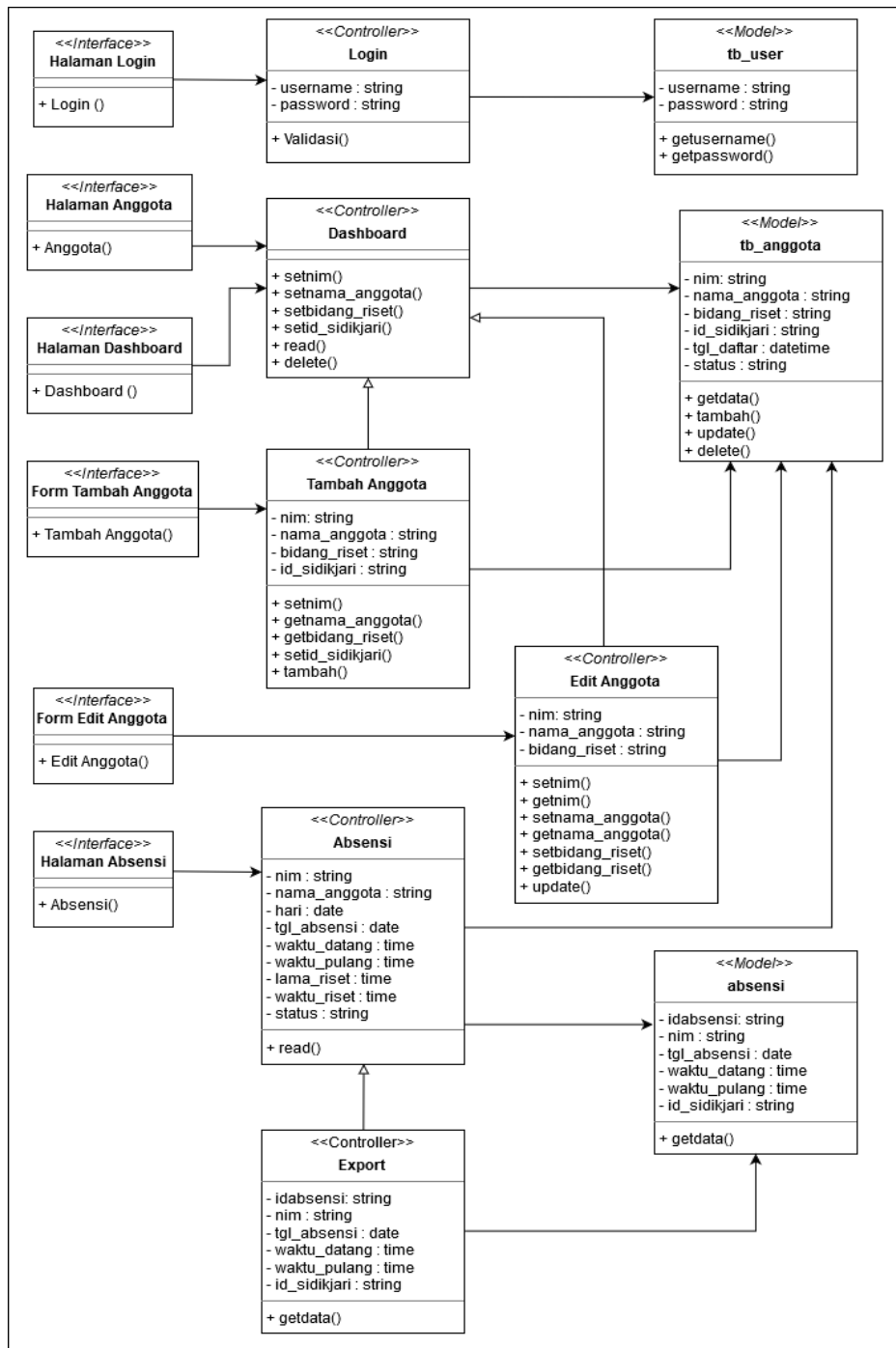
*Activity Diagram* Export menjelaskan bagaimana proses export.



**Gambar 3.22 Activity Diagram Export**

### 3.1.8.4. Class Diagram

*Class Diagram* merupakan diagram yang memodelkan sekumpulan kelas, interface, kolaborasi dan relasi. Class diagram digambarkan dengan kotak yang pada dasarnya terbagi tiga bagian, yaitu : nama kelas, atribut, dan operasi.



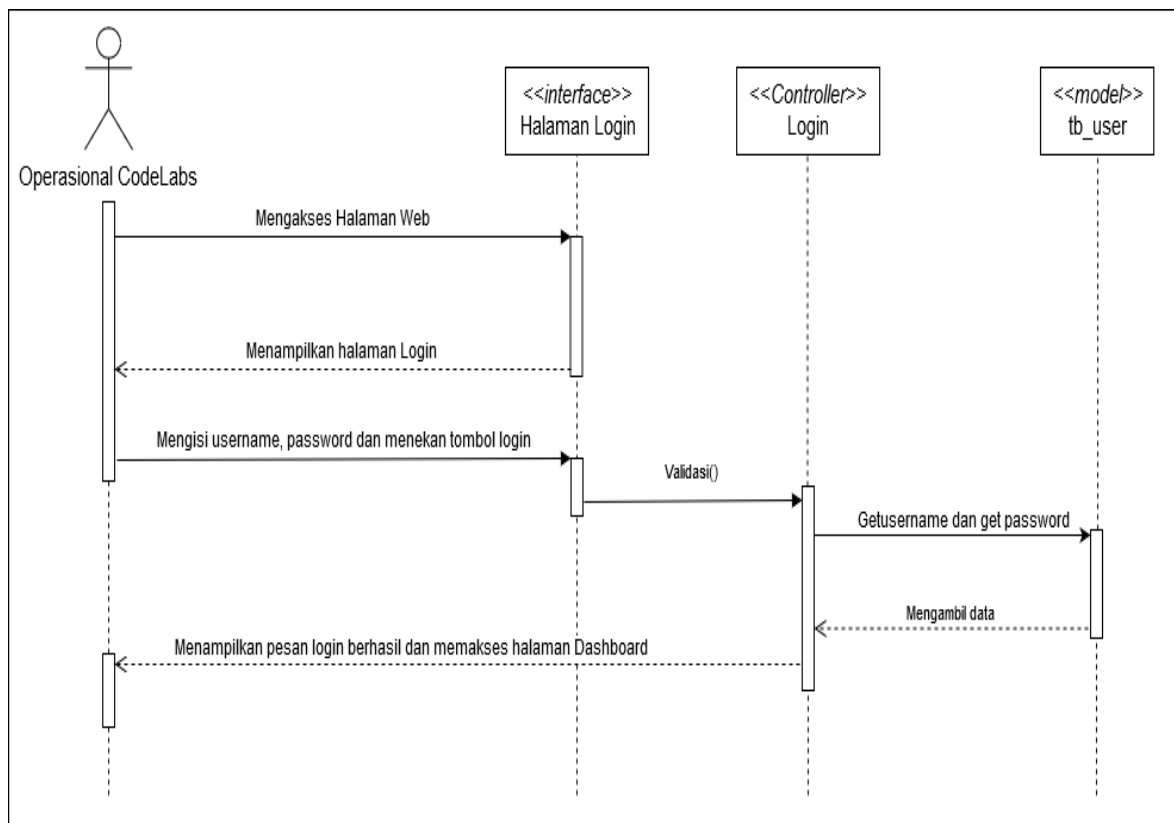
Gambar 3.23 Class Diagram

### 3.1.8.5. Sequence Diagram

*Sequence Diagram* adalah suatu diagram yang menggambarkan interaksi antara objek dan mengidikasikan komunikasi antara objek-objek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu. Objek-objek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi berada di paling kiri dari diagram.

#### 1. *Sequence Diagram Login*

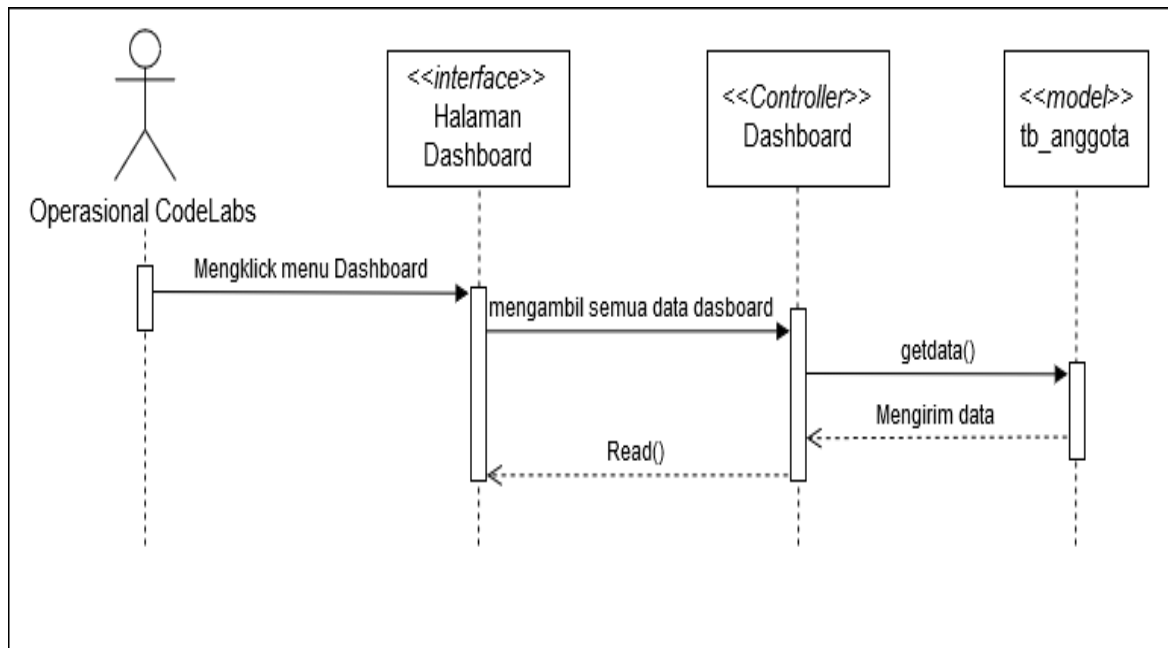
*Sequence Diagram Login* dapat dilihat pada gambar berikut.



**Gambar 3.24** *Sequence Diagram Login*

#### 2. *Sequence Diagram Dashboard*

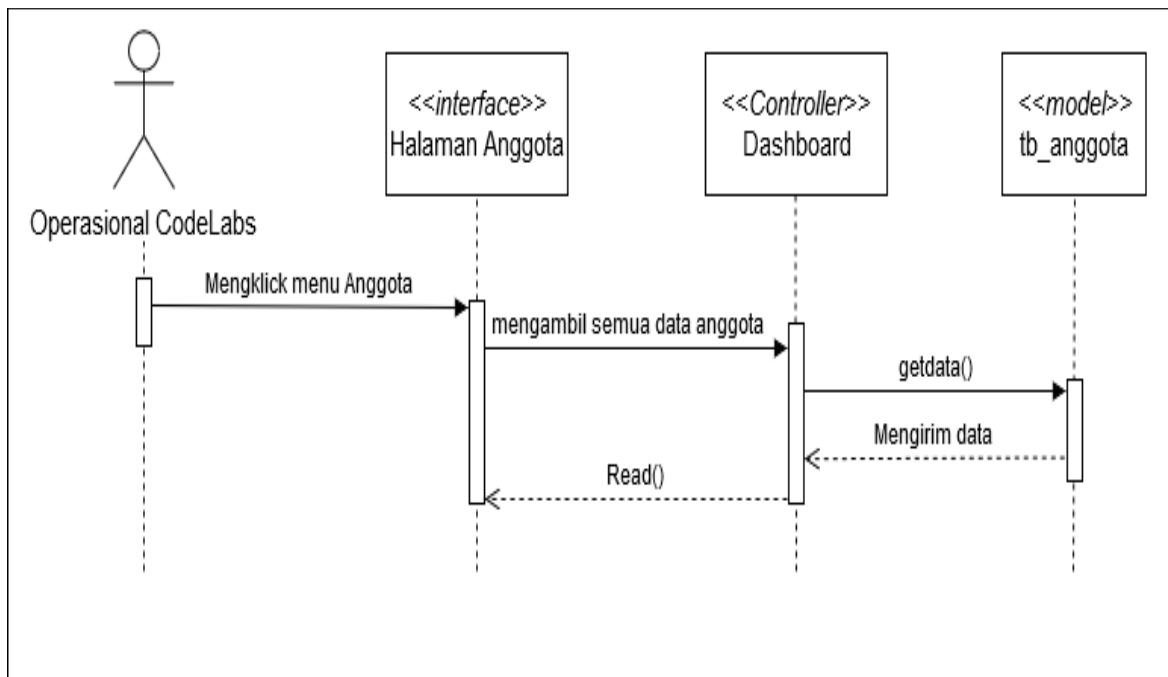
*Sequence Diagram Dashboard* dapat dilihat pada gambar berikut.



**Gambar 3.25 Sequence Diagram Dashboard**

### 3. Sequence Diagram Anggota

Sequence Diagram Anggota dapat dilihat pada gambar berikut.

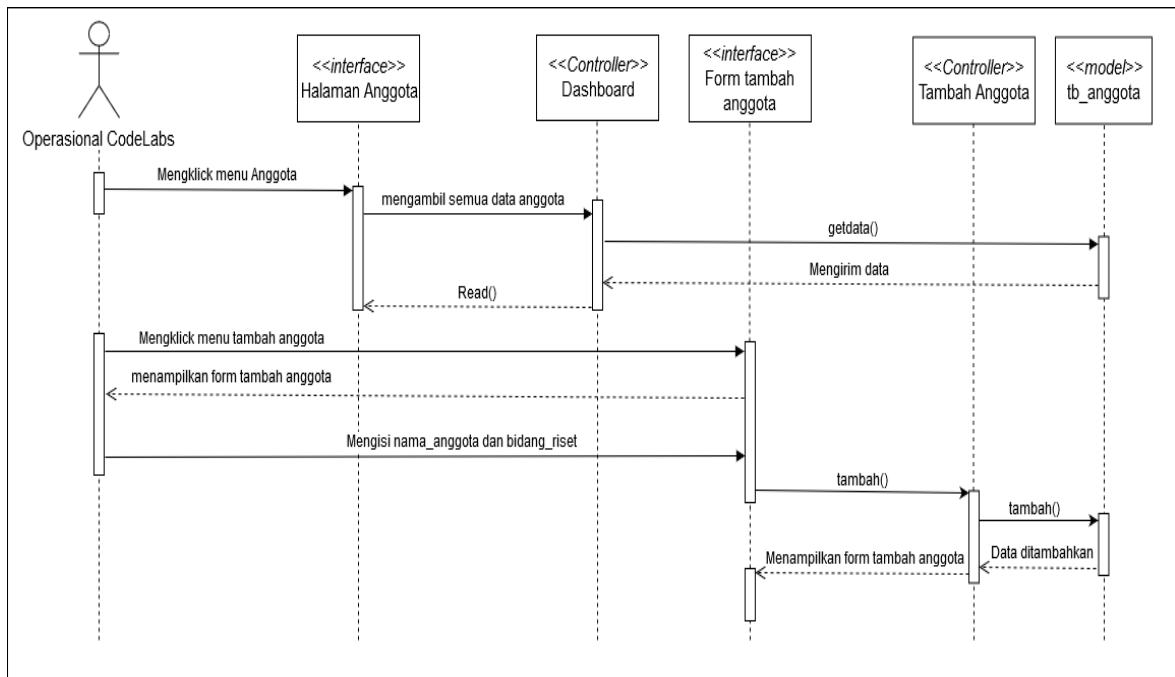


**Gambar 3.26 Sequence Diagram Anggota**

### 4. Sequence Diagram Tambah Data Anggota

Sequence Diagram Tambah Data Anggota dapat dilihat pada gambar berikut.

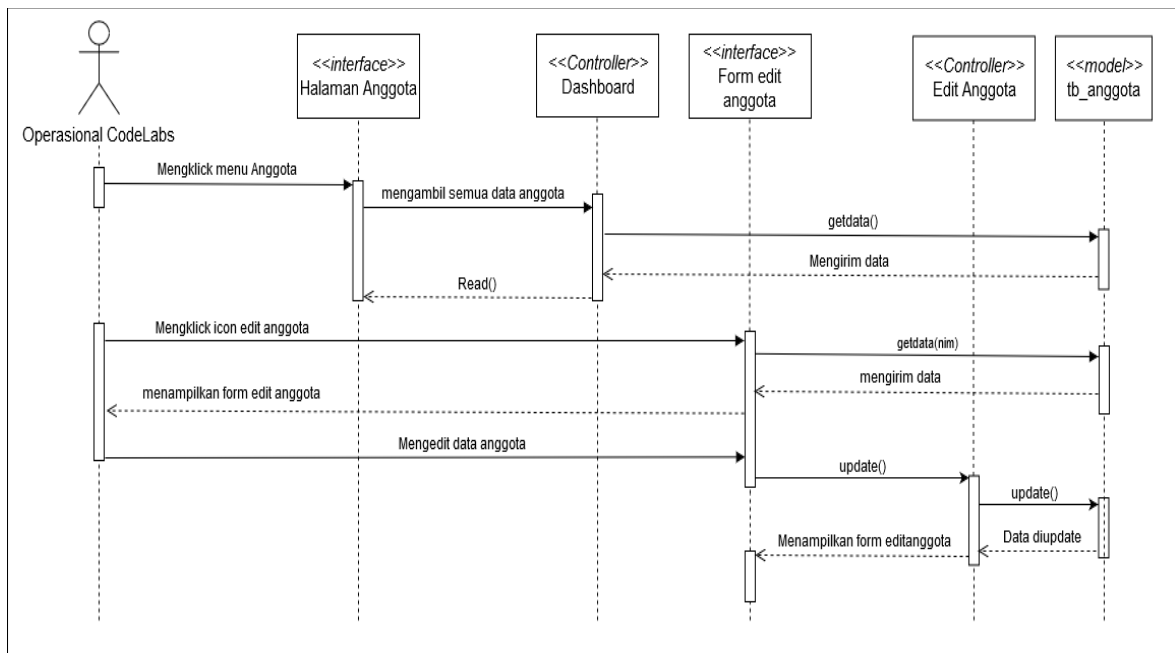




Gambar 3.27 Sequence Diagram Tambah Data Anggota

5. Sequence Diagram Edit Data Anggota

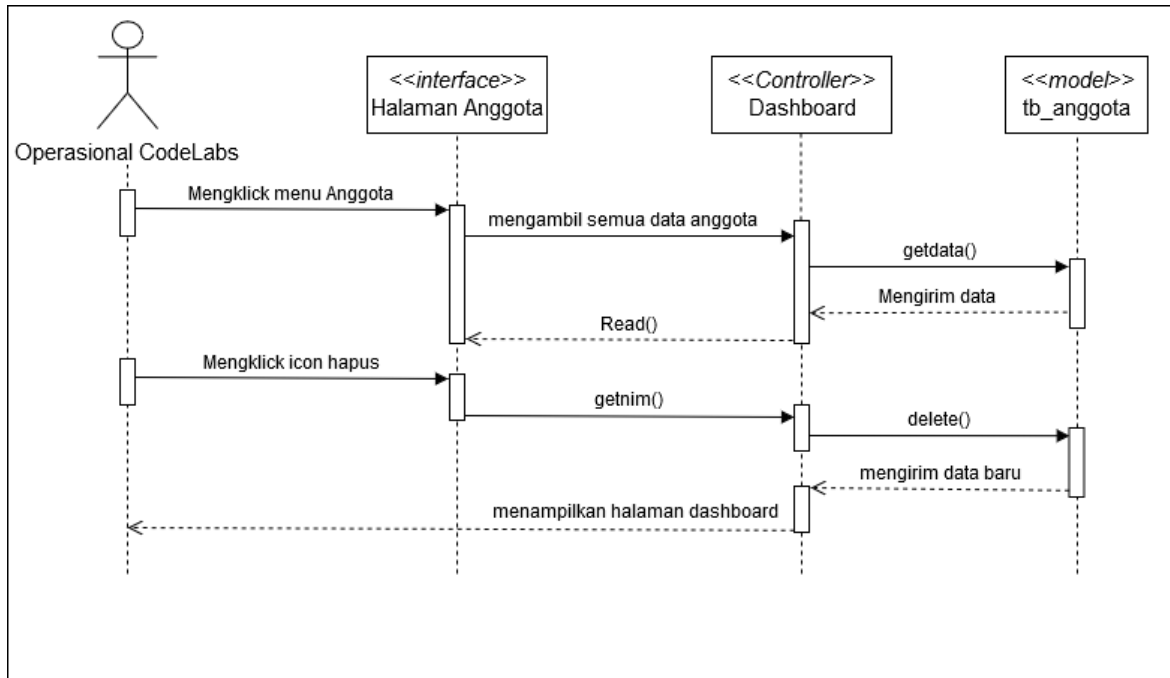
Sequence Diagram Edit Data Anggota dapat dilihat pada gambar berikut.



Gambar 3.28 Sequence Diagram Edit Data Anggota

6. Sequence Diagram Hapus Data Anggota

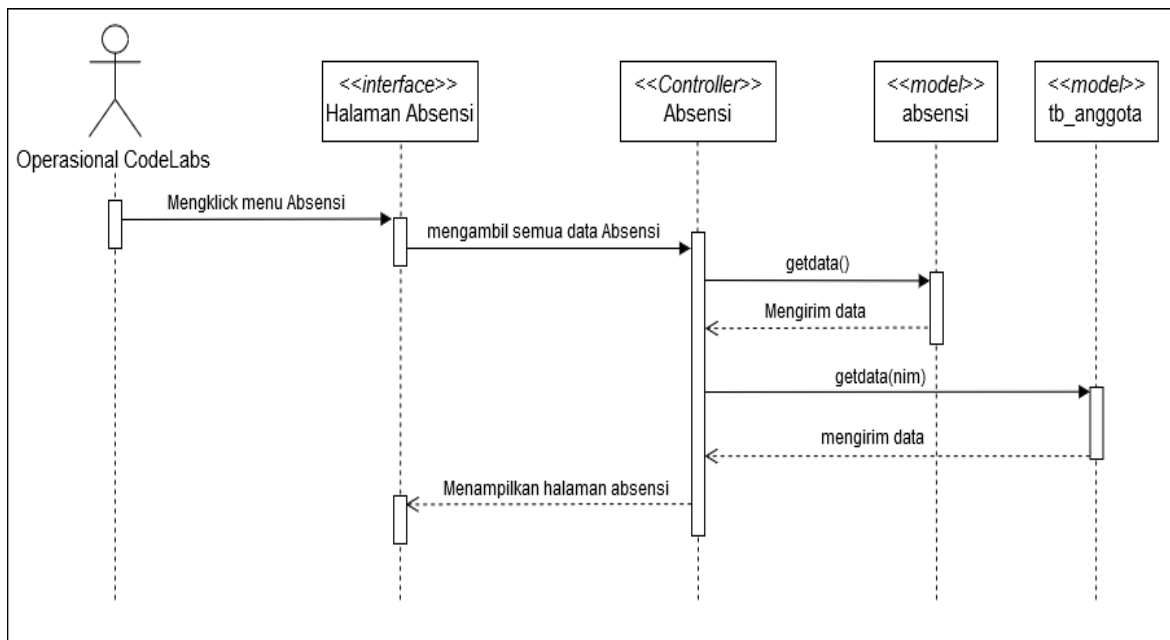
Sequence Diagram Hapus Data Anggota dapat dilihat pada gambar berikut.



**Gambar 3.29** *Sequecne Diagram Hapus Data Anggota*

## 7. Sequence Diagram Absensi

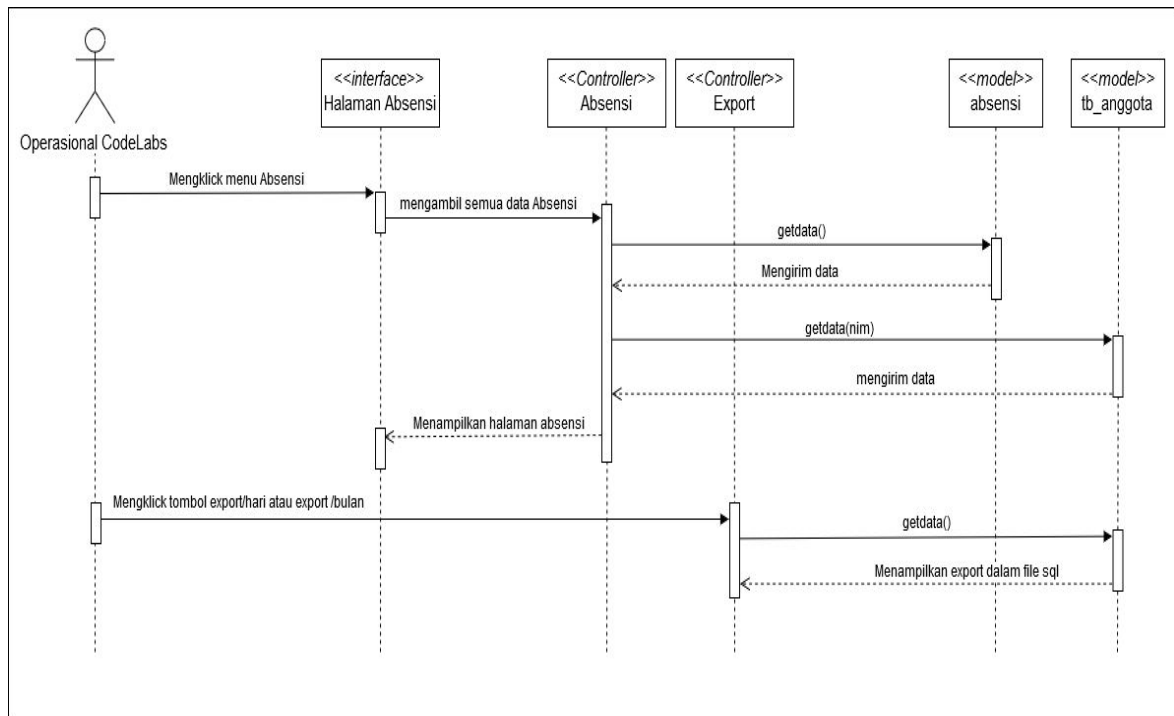
*Sequence Diagram Absensi* dapat dilihat pada gambar berikut.



**Gambar 3.30** *Sequence Diagram Absensi*

## 8. Sequence Diagram Export

*Sequence Diagram Export* dapat dilihat pada gambar berikut



**Gambar 3.31** *Sequence Diagram Export*

## 3.2. Perancangan Sistem

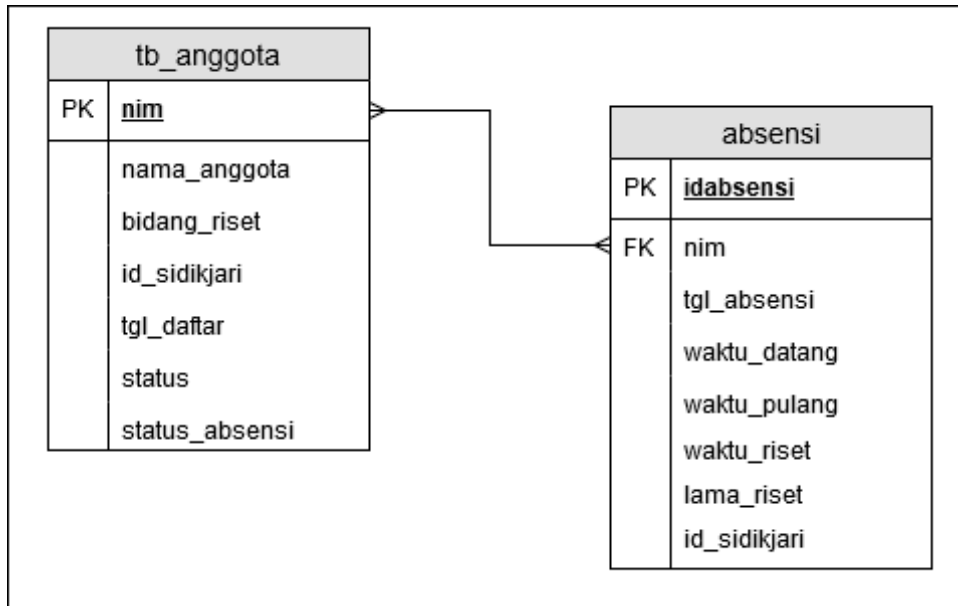
Perancangan sistem merupakan bagian yang menjelaskan mengenai perancangan sistem secara utuh baik dari perancangan data, perancangan antarmuka dan jaringan semantik.

### 3.2.1. Perancangan Data

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

#### 3.2.1.1. Skema Relasi

Skema relasi adalah rangkaian hubungan antara beberapa tabel pada sistem basis data. Penjelasan rangkaian basis data pada sistem ini dapat dilihat pada gambar dibawah ini.



Gambar 3.32 Skema Relasi

### 3.2.1.2. Struktur Tabel

Struktur tabel merupakan penjelasan dari tabel-tabel yang ada pada basis data. Struktur tabel dari sistem yang dibuat dapat dilihat sebagai berikut.

#### 1. Tabel tb\_anggota

Merupakan rancangan tabel tb\_anggota yang akan digunakan pada sistem yang akan dibangun.

Tabel 3.18 Struktur Tabel tb\_anggota

No	Field	Type	Size	Kunci	Keterangan
1	nim	Char	8	Primary key	Not null
2	nama_anggota	Varchar	40		Null
3	bidang_keilmuan	Varchar	50		Null
4	id_sidikjari	Varchar	3		Not null
5	tgl_daftar	Timestamp			
6	status	Enum			True/false
7	status_absensi	enum			Hadir/ Tidak hadir

#### 2. Tabel absensi

Merupakan rancangan tabel absensi yang akan digunakan pada sistem yang akan dibangun.

Tabel 3.19 Struktur Tabel absensi

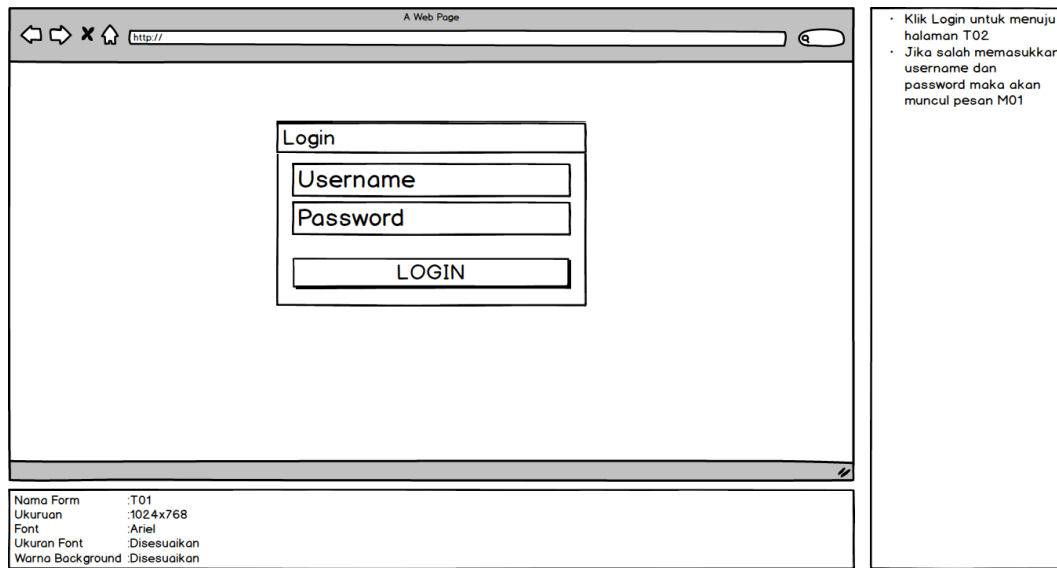
No	Field	Type	Size	Kunci	Keterangan
1	id_absensi	Int	5	Primary key	Not null, Auto increment
2	nim	Char	8	Foreign key	Not null, Reference tb_anggota(nim)
3	tgl_absensi	Date			Not null
4	waktu_datang	Time			Not null
5	waktu_pulang	Time			Default 00:00:00
6	waktu_riset	Time			Default 00:00:00
7	lama_riset	Time			Null
8	id_sidikjari	Varchar	3		Not null

### 3.2.2. Perancangan Antarmuka

Perancangan antarmuka dibuat untuk menggambarkan tampilan program yang digunakan oleh pengguna untuk berinteraksi dengan sistem yang akan dibangun. Perancangan dibuat berdasarkan tampilan antarmuka baik *input* atau *output* yang akan dihasilkan saat aplikasi diimplementasikan.

#### 1. Antarmuka Login

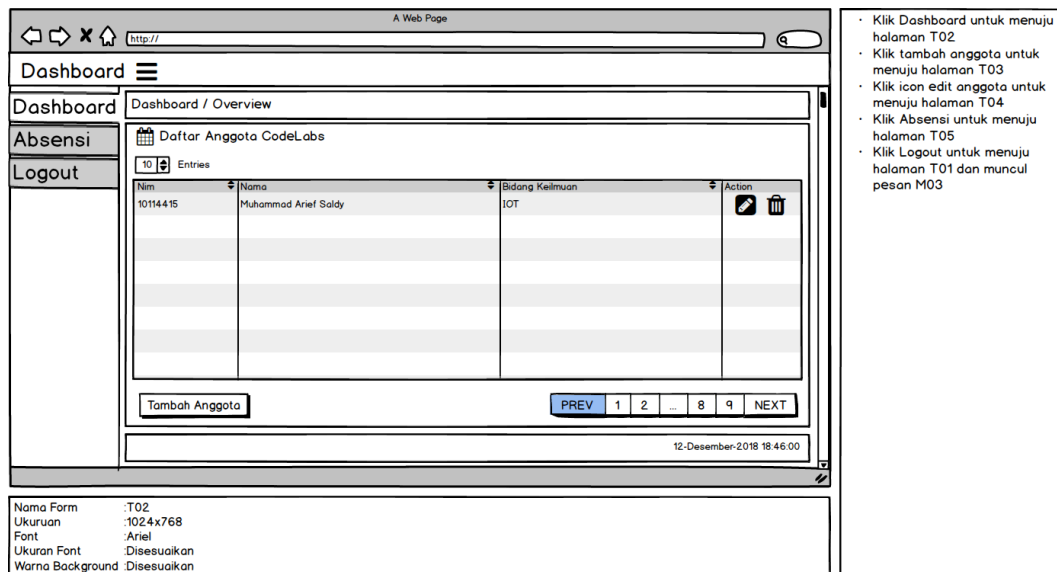
Antarmuka login merupakan tampilan untuk masuk kedalam sistem yang hanya dapat di akses oleh yang memiliki hak akses. Perancangan antarmuka login dapat dilihat pada gambar berikut.



**Gambar 3.33 Antarmuka Login**

## 2. Antarmuka Dashboard

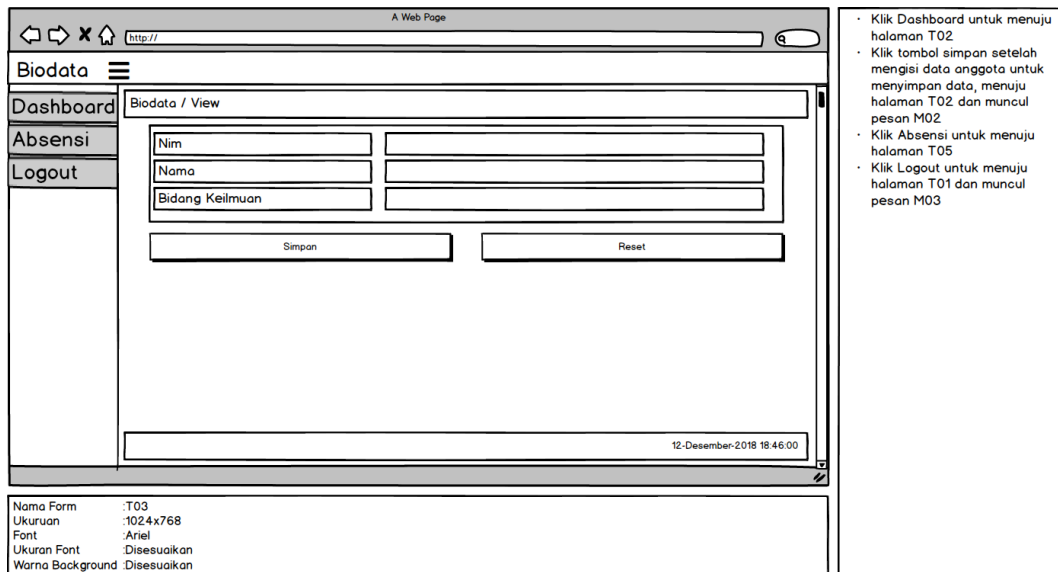
Antarmuka dashboard merupakan rancangan tampilan utama website setelah login, tampilan ini akan menampilkan data anggota CodeLabs.



**Gambar 3.34 Antarmuka Dashboard**

## 3. Antarmuka Tambah Anggota

Antarmuka tambah anggota merupakan rancangan tampilan untuk menambahkan anggota CodeLabs. Rancangan antarmuka tambah anggota dapat dilihat pada gambar berikut.

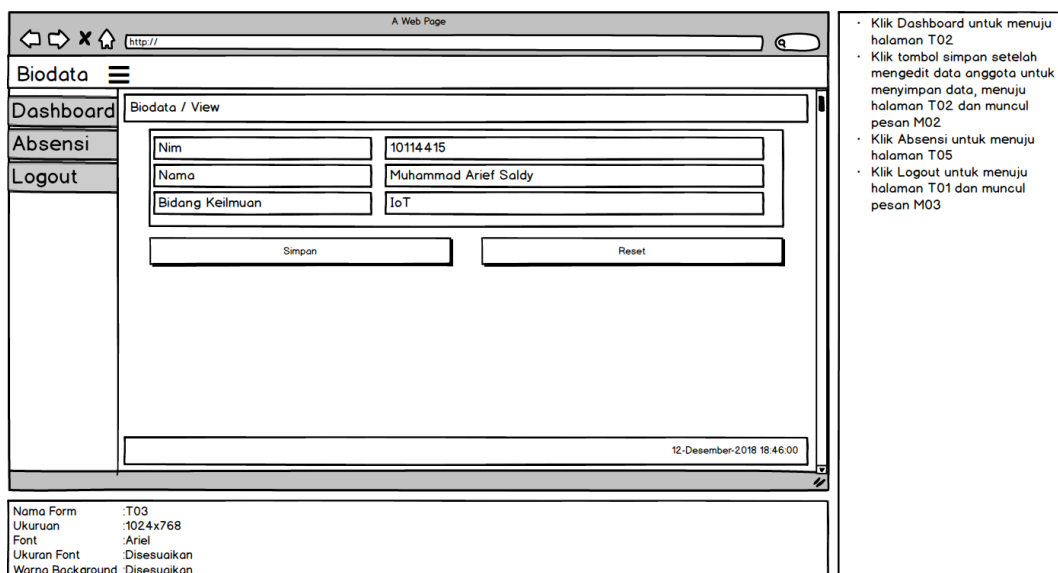


- Klik Dashboard untuk menuju halaman T02
- Klik tombol simpan setelah mengisi data anggota untuk menyimpan data, menuju halaman T02 dan muncul pesan M02
- Klik Absensi untuk menuju halaman T05
- Klik Logout untuk menuju halaman T01 dan muncul pesan M03

**Gambar 3.35 Antarmuka Tambah Anggota**

#### 4. Antarmuka Edit Data Anggota

Antarmuka edit data anggota merupakan rancangan tampilan untuk mengedit data anggota dan menyimpannya ke database. Rancangan antarmuka edit data anggota dapat dilihat pada gambar berikut.

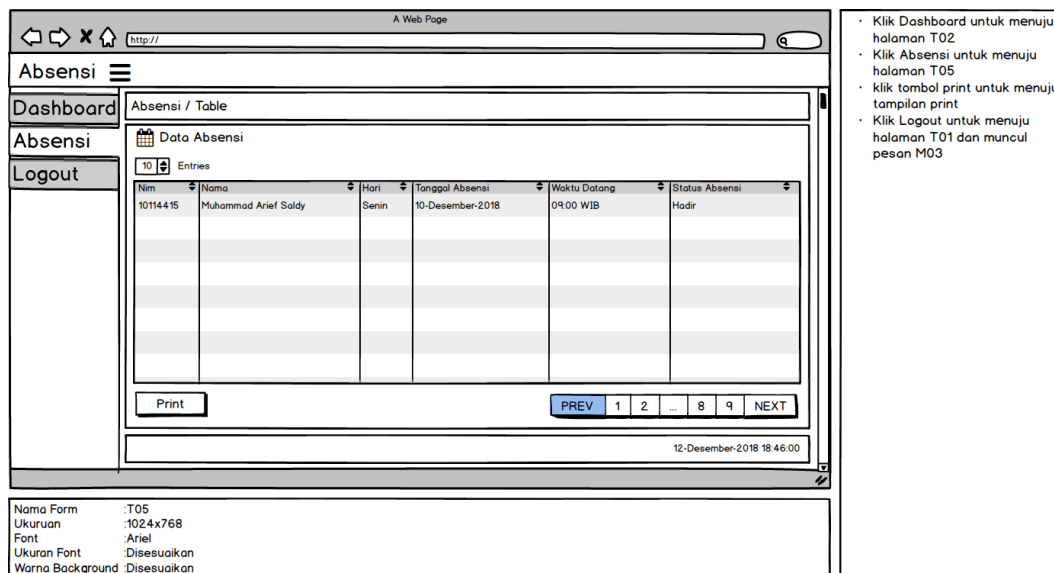


- Klik Dashboard untuk menuju halaman T02
- Klik tombol simpan setelah mengedit data anggota untuk menyimpan data, menuju halaman T02 dan muncul pesan M02
- Klik Absensi untuk menuju halaman T05
- Klik Logout untuk menuju halaman T01 dan muncul pesan M03

**Gambar 3.36 Antarmuka Edit Data Anggota**

## 5. Antarmuka Absensi

Antarmuka Absensi merupakan rancangan tampilan untuk melihat table absensi dan mengexportnya. Rancangan antar muka absensi dapat dilihat pada gambar berikut.



Gambar 3.37 Antarmuka Absensi

### 3.2.3. Perancangan Pesan

Perancangan pesan memberikan penjelasan pesan-pesan yang berada didalam aplikasi, jika masukkan dari operasional CodeLabs tidak sesuai dengan keinginan sistem. Adapun perancangan pesan sebagai berikut.

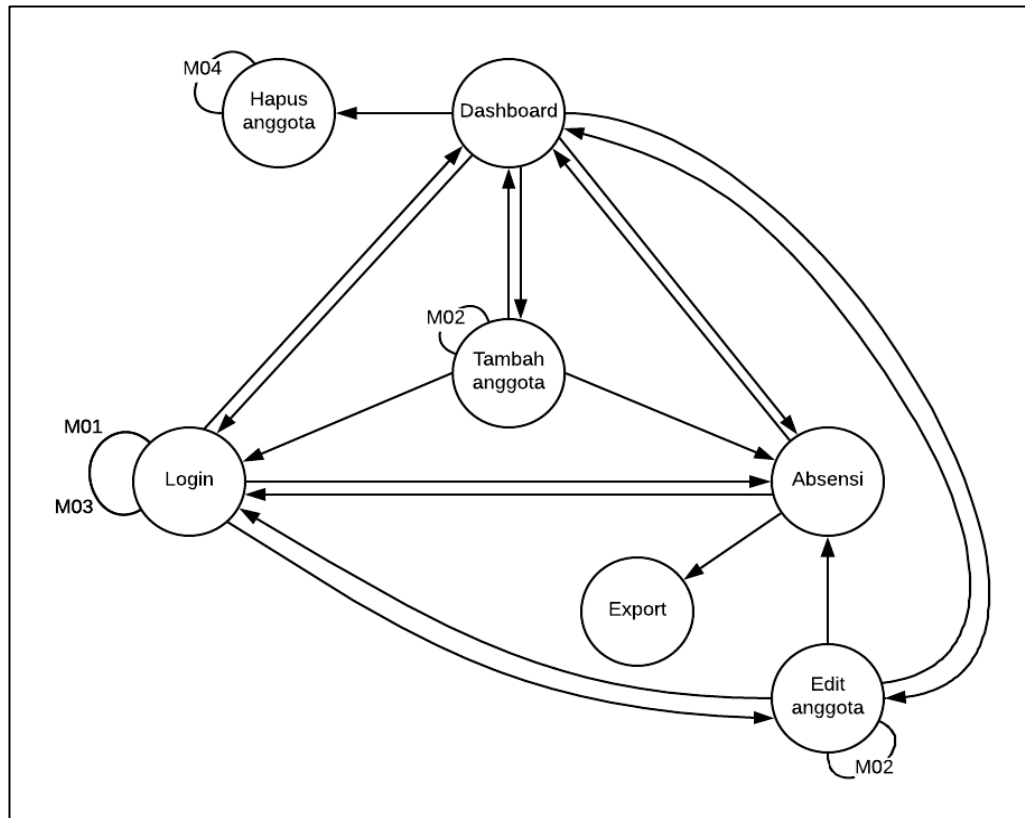
Kode Pesan	Pesan
M01	Username atau password anda salah
M02	Data anggota berhasil disimpan
M03	Anda berhasil Logout
M04	Data anggota berhasil dihapus

Tabel 3.20 Perancangan Pesan



### 3.2.4. Jaringan semantik

Jaringan semantik merupakan gambaran mengenai hubungan navigasi menu dari suatu halaman ke halaman lainnya.



Gambar 3.38 Jaringan Semantik

