**BAB IV**

**ANALISIS DAN PERANCANGAN SISTEM**

* 1. **Analisis Sistem**

Analisis sistem merupakan suatu tahapan yang bertujuan untuk mengetahui dan mengamati apa saja yang terlibat dalam suatu sistem. Pembahasan yang ada pada analisis sistem ini yaitu analisis masalah, analisis kebutuhan fungsional dan analisis kebutuhan *non*fungsional.

* 1. **Analisis Kebutuhan Sistem**

Analisis kebutuhan sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem dimana aplikasi dibangun, meliputi perangkat lunak, dan hasil analisis terhadap sistem serta elemen-elemen sistem.

**4.2.1. Kebutuhan Fungsional**

Analisis kebutuhan fungsional adalah segala bentuk data yang dibutuhkan oleh sistem agar sistem dapat berjalan sesuai dengan prosedur yang dibangun. Setelah melalui tahapan analisis, maka telah ditetapkan kebutuhan-kebutuhan untuk membangun sistem meliputi *input*, proses, dan *output.*

**4.2.1.1. Analisis Kebutuhan *Input***

*Input* dari aplikasi yang dibangun terdiri dari :

1. Informasi data mahasiswa jurusan Teknik Informatika UHO.
2. Informasi data dosen Teknik Informatika UHO.
3. Informasi data mata kuliah jurusan Teknik Informatika UHO.

**4.2.1.2. Analisis Kebutuhan Proses**

Kebutuhan proses dalam aplikasi ini antara lain :

1. Proses pembuatan aplikasi menggunakan Bahasa pemrograman *PHP* di sisi *webservice* dan menggunakan Bahasa pemrograman *Dart* di sisi aplikasi *user*.
2. Proses enkripsi dan dekripsi data *QR Code* menggunakan algoritma *Rivest Cipher 4 (RC4)*.

**4.2.1.3. Analisis Kebutuhan *Output***

*Output* yang diperoleh dalam sistem ini adalah *QR Code Card* yang datanya telah terenkripsi dengan algoritma *RC4 (Rivest Cipher 4)* dan aplikasi berbasis *android* untuk menjalankan fungsi dekripsi dari algoritma tersebut dan mendata absensi mahasiswa jurusan Teknik Informatika UHO.

**4.2.2. Kebutuhan Nonfungsional**

Analisis kebutuhan nonfungsional adalah sebuah langkah dimana seorang pembangun aplikasi menganalisis sumber daya yang dibutuhkan untuk membangun aplikasi yang akan dibangun. Analisis kebutuhan nonfungsional yang dilakukan dibagi dalam dua tahap, yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak.

**4.2.2.1. Kebutuhan Perangkat Keras**

Perangkat keras yang digunakan pada pembangunan sistem ini, sebagai berikut:

**Tabel 4.1 Spesifikasi Perangkat Keras**

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Perangkat** | **Spesifikasi** |
| 1. | *PC* | *Acer Aspire E 14 E5-421-61-C1* |
| 2. | *Processor* | *AMD A6 Quad Core* |
| 3. | *RAM* | *4 GB* |
| 4. | *Harddisk* | *240 GB SSD* |
| 5. | *Monitor* | *Monitor 14 Inch* |

**4.2.2.2. Kebutuhan Perangkat Lunak**

Perangkat lunak yang digunakan pada pembangunan sistem ini, sebagai berikut:

**Tabel 4.2 Spesifikasi Perangkat Lunak**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Perangkat** | **Fungsi** | **Spesifikasi** |
| 1. | *Windows* | *Operating System* | *Windows 10 Enterprise* |
| 2. | *Xampp* | Universal development environment | *Xampp v3.2.2* |
| 3. | *Apache* | *Web Server* | *Apache 2.4 win64-VC14* |
| 4. | *Mysql* | *Database Management* | *Mysql 5.7* |
| 5. | *PHP* | *Web development* | *PHP 7.3* |
| 6. | *Dart* | *Android Development* | *Dart 3.4.1* |
| 7. | *Chrome* | *Web browser* | *Chrome version 76.0* |
| 8. | *Android Studio* | *SDK Development Tool* | *Android Studio 3.3* |
| 9. | *Visual Studio Code* | *Code Editor* | *VS Code v1.38* |
| 10. | *Flutter* | *Framework Android* | *Flutter 3.4.1* |

* 1. **Perancangan *Flowchart***

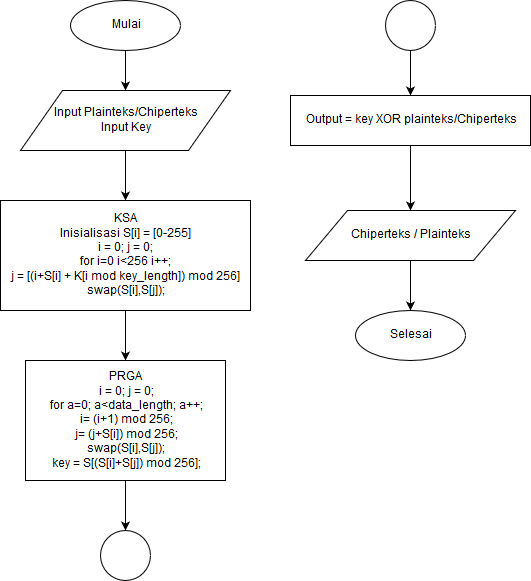
Flowchart adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung. Perancangan flowchart yang akan dibangun terdiri atas flowchart sistem, flowchart metode *Rivest Cipher* *4* (*RC4*).

* + 1. ***Flowchart* Sistem**

Setelah menganalisis sistem, maka didapatkan flowchart diagram untuk aplikasi absensi, dan flowchart metode *Rivest Cipher* *4* (*RC4*).

* + - 1. ***Flowchart* metode *Rivest Cipher 4 (RC4)***

Adapun flowchart enkripsi dan dekripsi algoritma RC4, yang ditunjukkan oleh Gambar 4.1. Adapun alur kerja flowchart enkripsi dan dekripsi algoritma RC4:



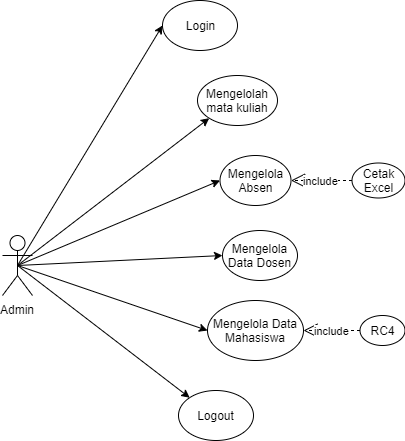
Gambar 4. 1 Flowchart Enkripsi dan Dekripsi Algoritma RC4

* + 1. ***Unified Modeling Language* (UML)**

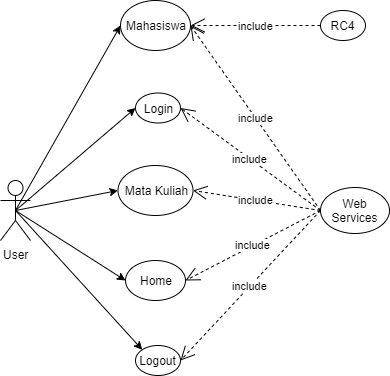
Aplikasi dibangun dengan menggunakan Unified Modeling Language (UML). UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram yang terdiri dari *Use Case Diagram, Activity Diagram, Class Diagram dan Sequence Diagram.*

1. ***Use Case Diagram***

*Use Case* *Diagram* adalah sebuah diagram yang dapat merepresentasikan interaksi yang terjadi antara user dengan sistem. *Use Case Diagram* ini mendeskripsikan siapa saja yang menggunakan sistem dan bagaimana cara mereka berinteraksi dengan sistem. *Use Case* *Diagram* dari sistem yang akan dibangun adalah sebagai berikut.



Gambar 4. 2 Use Case Diagram Admin



Gambar 4. 3 *Use Case Diagram User*

**Tabel 4.3 Keterangan *Use Case Diagram***

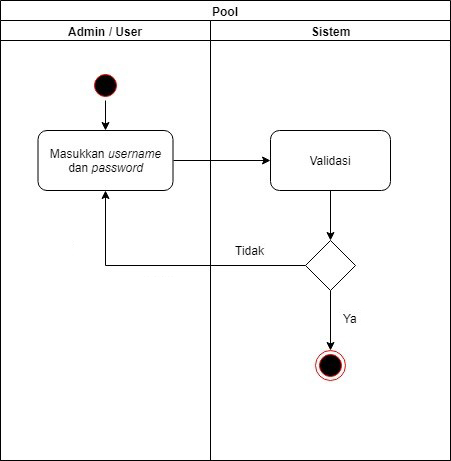
|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| *Admin* memilih menu absen | Sistem akan menampilkan data absensi yang diterima dari *web service* dan menyediakan pilihan cetak absen |
| *Admin* memilih menu mahasiswa | Sistem akan menyediakan menu untuk edit data, hapus data, tampilkan data dan tambah data mahasiswa yang kemudian akan dienkripsi dan tersimpan dalam *web service* |
| *Admin* dan *User* memilih menu *login* | Sistem akan menampilkan *form login* |
| *User* memilih menu mata kuliah | Sistem akan menampilkan daftar mata kuliah |
| *User* memilih menu *scan QR Code* | Sistem akan mengaktifkan kamera dan mencari *QR Code* dan melakukan *scanning* dan deskripsi pada data *QR Code* |
| *User* memilih menu data kehadiran mahasiswa | Sistem akan menampilkan data kehadiran masing-masing mahasiswa |

1. ***Activity Diagram***

*Activity* diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut ini adalah *activity* diagram yang akan menggambarkan alir aktivitas sistem.

1. ***Activity Diagram* *Login***

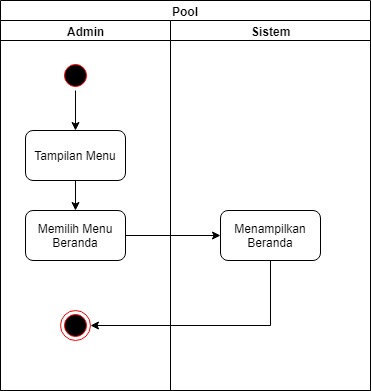
Pada Gambar 4.4 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* dan *user* ketika memasukkan *username* dan *password*, lalu sistem akan menampilkan melakukan validasi.



Gambar 4. 4 Activity Diagram Login (Admin dan User)

1. ***Activity Diagram* Beranda/*Dashboard***

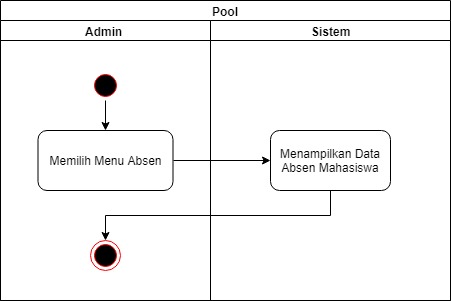
Pada Gambar 4.5 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* ketika memilih menu beranda, lalu sistem akan menampilkan menu beranda yang berisikan tampilan menu beranda.



Gambar 4. 5 Activity Diagram Beranda

1. ***Activity Diagram* Tampil Data Absensi**

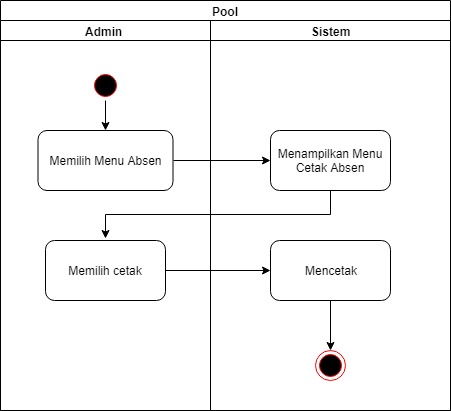
Pada Gambar 4.6 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu absen lalu sistem menampilkan halaman data absensi mahasiswa.



Gambar 4. 6 ActivityDiagram Tampil Data Absensi

1. ***Activity Diagram* Cetak Absen**

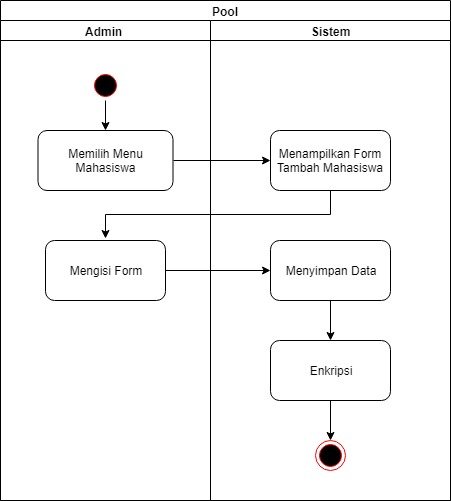
Pada Gambar 4.7 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu absen lalu sistem menampilkan halaman cetak absen.



Gambar 4.7 ActivityDiagram Cetak Absen

1. ***Activity Diagram* Tambah Data Mahasiswa**

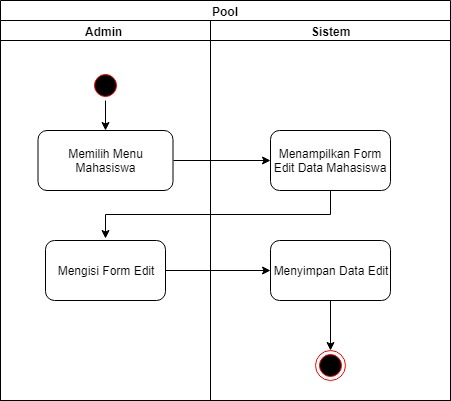
Pada Gambar 4.8 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mahasiswa lalu sistem menampilkan *form* tambah data mahasiswa yang kemudian akan disimpan dan dienkripsi.



Gambar 4.8 ActivityDiagram Tambah Data Mahasiswa

1. ***Activity Diagram* *Edit* Data Mahasiswa**

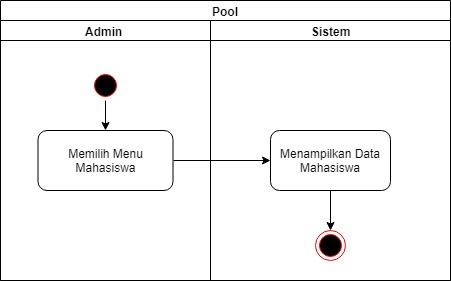
Pada Gambar 4.9 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mahasiswa lalu sistem menampilkan form edit data mahasiswa.



Gambar 4.9ActivityDiagram Edit Data Mahasiswa

1. ***Activity Diagram* Tampil Data Mahasiswa**

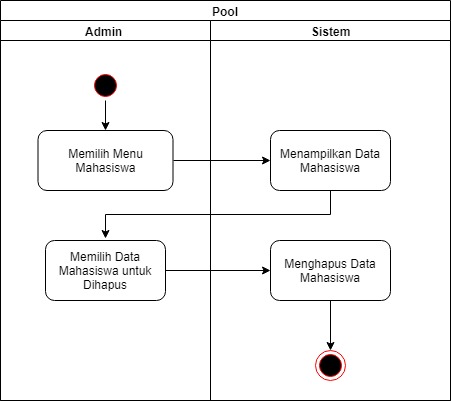
Pada Gambar 4.10 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mahasiswa lalu sistem menampilkan data mahasiswa.



Gambar 4.10 ActivityDiagram Tampil Data Mahasiswa

1. ***Activity Diagram* Hapus Data Mahasiswa**

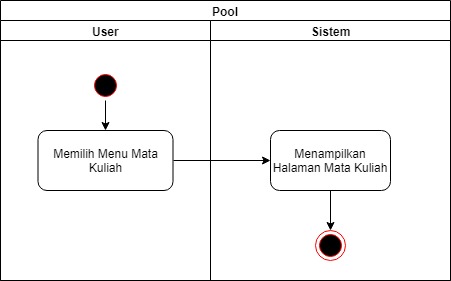
Pada Gambar 4.11 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mahasiswa lalu sistem menampilkan data mahasiswa lalu *admin* memilih hapus kemudian sistem menghapus data mahasiswa tersebut.



Gambar 4.11 ActivityDiagram Hapus Data Mahasiswa

1. ***Activity Diagram* Mata Kuliah**

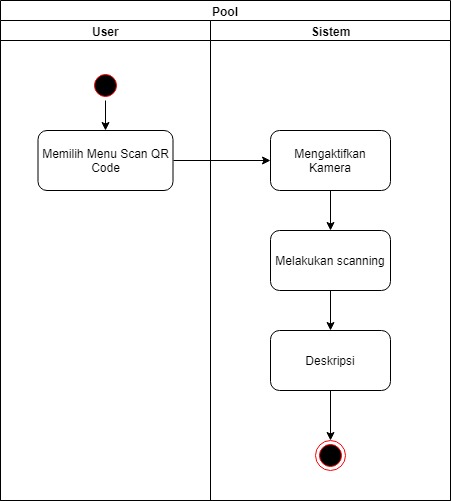
Pada Gambar 4.12 merupakan diagram aktivitas yang menunjukkan aktivitas *user* memilih menu mata kuliah lalu sistem menampilkan halaman mata kuliah.



Gambar 4.12 ActivityDiagram Mata Kuliah

1. ***Activity Diagram* Deskripsi**

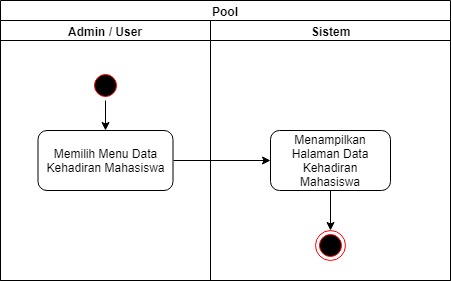
Pada Gambar 4.13 merupakan diagram aktivitas yang menunjukkan aktivitas *user* memilih menu *scan QR Code* lalu sistem mengaktifkan kamera dan melakukan scanning pada *QR Code* lalu melakukan deskripsi terhadap data *QR Code* tersebut.



Gambar 4.13 ActivityDiagram Deskripsi

1. ***Activity Diagram* Data Kehadiran Mahasiswa**

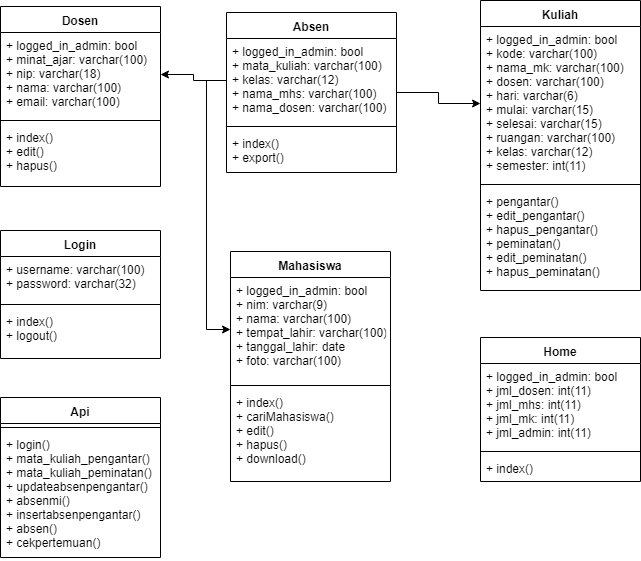
Pada Gambar 4.14 merupakan diagram aktivitas yang menunjukkan aktivitas user memilih menu data kehadiran mahasiswa lalu sistem menampilkan halaman data kehadiran mahasiswa.



Gambar 4.14 ActivityDiagram Data Kehadiran Mahasiswa

1. ***Class Diagram***

*Class diagram* merupakan diagram untuk menjelaskan pemodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berinteraksi untuk mencapai suatu tujuan. Berikut ini adalah *class diagram system*.

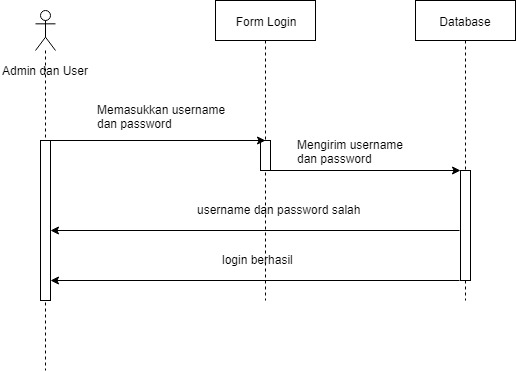
**

Gambar 4.15 Class Diagram

1. ***Sequence Diagram***

*Sequence Diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem yang digambarkan terhadap waktu. Berikut ini adalah *Sequence Diagram* yang akan menggambarkan interkasi antar objek dan sistem.

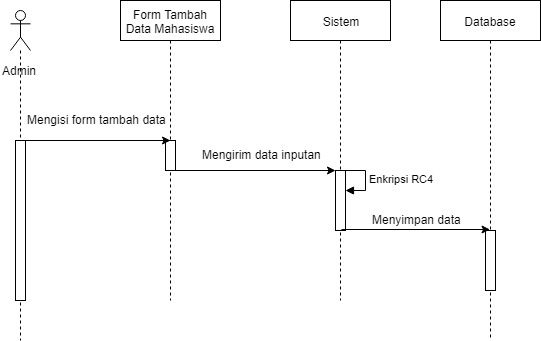
1. ***Sequence Diagram Login***



Gambar 4. 16 Sequence Diagram Login

Gambar 4.16 adalah *sequence* yang menunjukkan proses login bagi admin dan user dimana *actor* memasukkan *username* dan *password* lalu divalidasi oleh *database*.

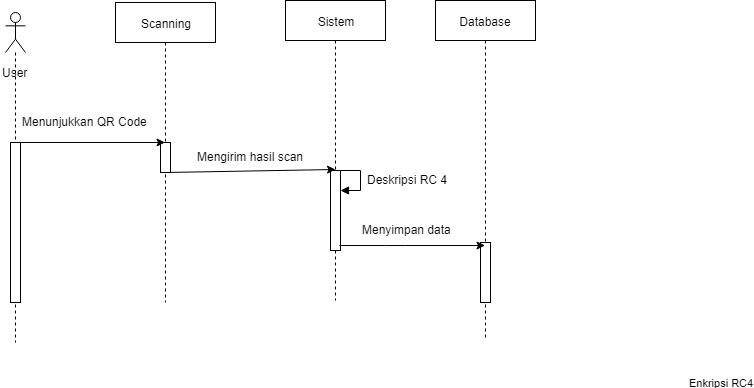
1. ***Sequence Diagram* Enkripsi**



Gambar 4.17 Sequence Diagram Enkripsi

Gambar 4.17 adalah *sequence* yang menunjukkan proses enkripsi dimana *admin* mengisi data mahasiswa yang baru lalu data akan diproses oleh sistem dan melakukan enkripsi pada kode khusus yang juga dibuat oleh sistem baru kemudian disimpan ke *database.*

1. ***Sequence Diagram* Deskripsi**



Gambar 4.18 Sequence Diagram Deskripsi

Gambar 4.18 adalah *sequence* yang menunjukkan proses deskripsi dimana *user* menunjukkan *QR Code* untuk kemudian di-*scan* lalu kemudian hasil *scan* akan dikirim ke sistem lalu didekripsi baru kemudian disimpan ke *database.*

* 1. **Perancangan Antarmuka (*Interface*)**

Rancangan antarmuka pengguna atau *design user interface* merupakan penggambaran tampilan yang digunakan secara langsung oleh pengguna terhadap sistem. Rancangan *user* *interface* ini dibuat sederhana agar mudah dimengerti pengguna dan tidak ada kerumitan dalam menjalankannya sehingga mencapai tujuan perangkat lunak yang *user friendly*.

* + 1. **Menu *Dashboard***

Menu Dashboard menampilkan data aplikasi secara umum. Mulai dari jumlah dosen, mahasiswa, mata kuliah dan *user*. Selain itu pada halaman ini, sistem juga menampilkan data admin serta *form* untuk mengubah *password admin*. Bentuk dari tampilan menu *dashboard* dapat dilihat pada gambar berikut.

****

Gambar 4.19 Menu Dashboard

* + 1. **Menu Absen**

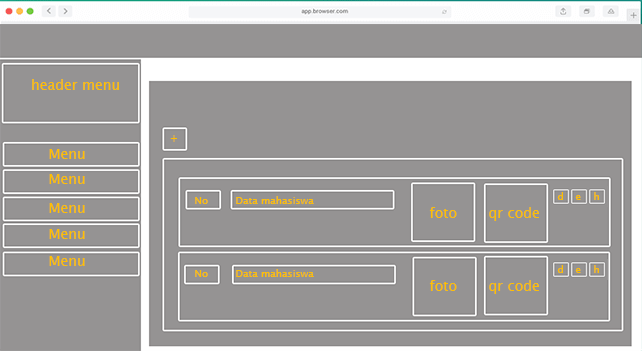
Menu absen menampilkan data absensi berdasarkan nama mata kuliah dan kelas. Terdapat tabel yang berisikan nama dan nim masing-masing mahasiswa serta daftar kehadirannya dari pertemuan 1 sampai pertemuan ke 16. Serta informasi persentase kehadiran mahasiswa tersebut. Selain itu, juga terdapat tombol cetak untuk melakukan *print out* data absensi tersebut. Bentuk dari tampilan menuabsendapat dilihat pada gambar berikut.

****

Gambar 4. 20 Menu Absen

* + 1. **Menu Mahasiswa**

Menu mahasiswa berisikan data pribadi setiap mahasiswa, foto dan gambar *QR Code* yang secara otomatis dibuat oleh sistem berdasarkan data *inputan user.* Bentuk dari tampilan menumahasiswa dapat dilihat pada gambar berikut.

****

Gambar 4.21 Menu Mahasiswa

* + 1. **Menu Dosen**

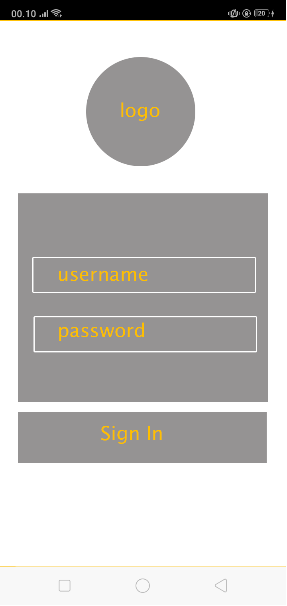
Menu dosen berisi data setiap dosen yang ada pada jurusan Teknik Informatika UHO. Terdapat pula foto dari masing data dosen agar lebih memudahkan dalam mencari data atau informasi tentang dosen. Bentuk dari tampilan menudosendapat dilihat pada gambar berikut.

****

Gambar 4.22 Menu Dosen

* + 1. **Halaman *Login Mobile***

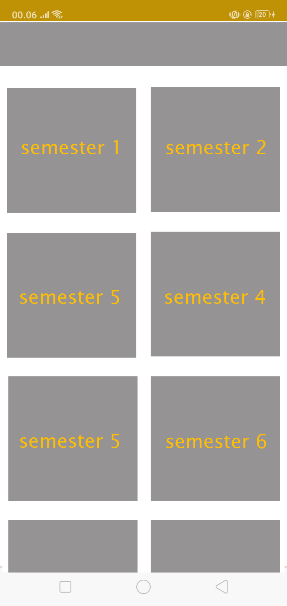
Halaman login pada aplikasi *mobile* menampilkan *form login* yang terdiri dari *username* dan *password*. Setiap *field* harus diisi agar dapat melakukan *login.* Selain itu, *form* ini juga telah ditambahkan fungsi validasi agar dapat menyaring data *user* yang benar. Bentuk dari tampilan *login mobile* dapat dilihat pada gambar berikut.

****

Gambar 4.23 Halaman Login Mobile

* + 1. **Halaman *Home Mobile***

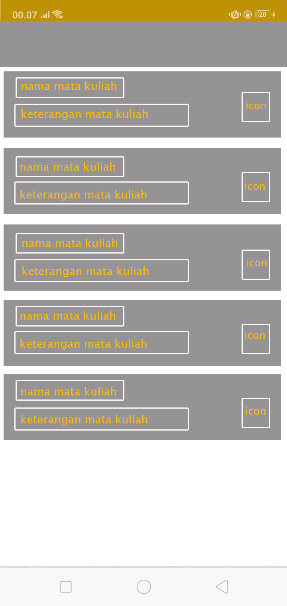
Halaman *home mobile* menampilkan daftar semester 1 sampai 8. Bentuk dari tampilan *home mobile* dapat dilihat pada gambar berikut.

******

Gambar 4.24 Halaman Home Mobile

* + 1. **Halaman Mata Kuliah Mobile**

Halaman mata kuliah mobile menampilkan daftar mata kuliah pada semester dan kelas tertentu. Keterangan mata kuliah juga dicantumkan seperti jadwal dan nama ruangan perkuliahan. Bentuk dari tampilan mata kuliah *mobile* dapat dilihat pada gambar berikut.

****

Gambar 4.25 Halaman Mata Kuliah Mobile

* + 1. **Halaman Daftar Mahasiswa Per-Mata Kuliah**

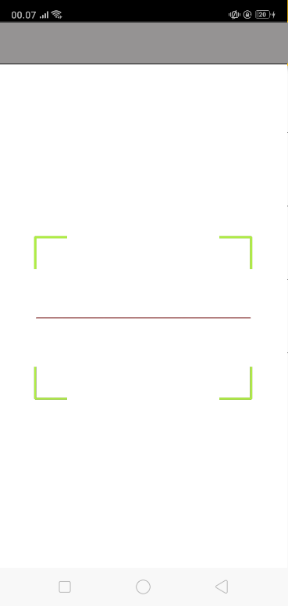
Halaman ini menampilkan data mahasiswa yang terdaftar dalam satu mata kuliah tertentu. Disertai pula *icon* informasi yang akan menampilkan status kehadiran mahasiswa tersebut. Adapun bentuk dari tampilan daftar mahasiswa permata kuliahdapat dilihat pada gambar berikut.

****

Gambar 4.26 Halaman Daftar Mahasiswa Per Mata Kuliah

* + 1. **Halaman** ***Scanning QR Code***

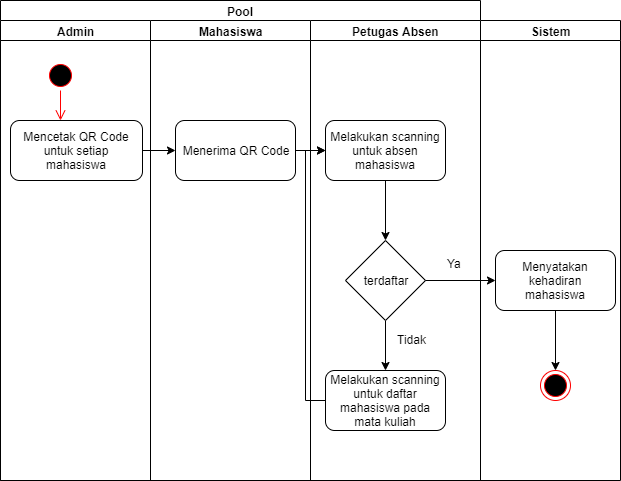
Halaman *scanner* menampilkan alat *scanner* berupa kamera yang siap melakukan *scanning* terhadap QR Code yang ditemukan. Adapun bentuk dari tampilan *Scanning QR Code* dapat dilihat pada gambar berikut.

****

Gambar 4. 27 Halaman Scanning QR Code

* 1. **Prosedur Penggunaan Sistem**
     1. **Aturan**

1. Ada 1 atau lebih petugas yang melakukan absensi.
2. Petugas melakukan absensi 10 menit sebelum dimulainya mata kuliah dan 10 menit waktu keterlambatan.
3. Jika mahasiswa sakit atau izin, harus melapor ke petugas absensi dengan membawa bukti keterangan dan *QR Code* miliknya.
   * 1. **Langkah-langkah penggunaan sistem.**



Gambar 4.28 Prosedur Penggunaan Sistem