

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Di era modern seperti sekarang ini, kemajuan teknologi berkembang semakin pesat. Berbagai sistem mulai diciptakan menggunakan teknologi dengan tujuan untuk mempermudah pekerjaan manusia. Mulai dari bidang pemerintahan, perusahaan, politik dan pendidikan seperti sekolah dan universitas. Adapun di universitas sendiri ada salah satu sistem pengelolaan yang masih di kerjakan secara manual yaitu daftar hadir. Sistem daftar hadir merupakan salah satu faktor yang sangat penting dalam dunia universitas. Selain digunakan sebagai alat kontrol mahasiswa, daftar hadir juga digunakan sebagai salah satu penunjang penilaian dosen dan syarat dokumentasi borang setiap 5 tahun sekali.

Universitas Halu Oleo merupakan salah satu universitas yang masih menggunakan sistem manual dalam penanganan daftar hadir di masing-masing jurusan. Tanpa terkecuali jurusan Teknik Informatika yang setiap semester harus mencetak lembaran daftar hadir dan merekap data tersebut di akhir semester. Dari 300 mahasiswa aktif dengan lebih dari 20 mata kuliah disetiap semester, tentu menjadi tugas yang tidak mudah bagi pegawai staff di jurusan untuk mengontrol data tersebut. Maka dengan kemajuan teknologi, tidak menutup kemungkinan untuk mengembangkan sebuah aplikasi agar dapat membantu dan memudahkan sistem daftar hadir tersebut.

Telpon seluler (*Smartphone*) merupakan tools yang tepat dalam membangun sistem tersebut. Dengan kemampuan kamera di setiap *smartphone* yang dapat mengambil, menyimpan dan menampilkan gambar dengan format *JPEG*. Memunculkan ide untuk memanfaatkan *QR Code* dan *smartphone* Android dalam mengembangkan sistem daftar hadir tersebut. Dimana dalam *QR Code* ini nantinya akan menyimpan data setiap mahasiswa dan dapat dicetak.

Informasi yang dikodekan dalam *QR Code* dapat berupa *URL*, nomor telepon, pesan *SMS*, *V-Card* atau teks apa pun. Disebut *QR Code* karena teknologi memungkinkan sistem untuk memecahkan kode konten dengan kecepatan tinggi. Ditambah lagi dengan sistem operasi *Android* pada *smartphone* yang sangat populer dan banyak digunakan, membuat sistem ini akan menjadi lebih cepat,

dan praktis. Namun, karena *QR Code* ini dapat dibaca dengan aplikasi *QR Scanner* apapun, maka penting menambahkan algoritma pada aplikasi ini yang berguna untuk mengamankan data pada *QR Code* tersebut (Ashford, 2016).

Salah satu metode yang banyak digunakan dalam sistem pengamanan data adalah Algoritma *RC4 (Rivers Code 4)*. Algoritma ini bekerja dengan kunci enkripsi yang didapat dari 256 bit *state array* yang diinisialisasi dengan sebuah *key* tersendiri dengan panjang 1-256 bit. Setelah itu, *state array* yang didapatkan diacak kembali dan diproses untuk menghasilkan sebuah kunci enkripsi yang akan di-XOR dengan *plaintext* ataupun *ciphertext* sehingga didapatkan hasil dari enkripsi ataupun dekripsi (Sholeh, 2019).

Dengan modifikasi yang digabungkan dengan algoritma ini akan membuat data dalam *QR Code* ini sulit untuk dipecahkan. Adapun pesan yang keluar tidak akan mudah dimengerti karena telah dienkripsi sehingga *QR Code* tersebut hanya dapat digunakan pada aplikasi tertentu. Selain itu, proses algoritma ini menggunakan kunci simetris yang dapat membuat proses enkripsi dan dekripsi *QR Code* menjadi lebih cepat (Jumrin, Sutardi, & Subardin, 2016).

## **1.2. Rumusan Masalah**

Rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan algoritma *RC4* sebagai metode pengamanan data *QR Code* pada aplikasi daftar hadir Teknik Informatika?
2. Bagaimana mengembangkan sistem daftar hadir agar dapat terdigitalisasi dan dapat memudahkan staff Jurusan Teknik Informatika Universitas Halu Oleo?
3. Bagaimana mengembangkan aplikasi daftar hadir yang praktis, efektif dan efisien?

## **1.3. Batasan Masalah**

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Aplikasi ini hanya menerapkan algoritma *RC4 (Rivers Code 4)* pada data *QR Code*.
2. Tidak ada sistem dalam aplikasi ini yang dibuat untuk mahasiswa.

3. Ponsel *Android* pengguna aplikasi harus terkoneksi jaringan internet dengan baik.
4. Aplikasi ini hanya digunakan dalam lingkup jurusan Teknik Informatika UHO.

#### **1.4. Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Membangun aplikasi daftar hadir yang praktis dan cepat serta dokumentasi yang mudah di jurusan Teknik Informatika Universitas Halu Oleo.
2. Menerapkan algoritma *RC4 (Rivers Code 4)* untuk mengamankan data *QR Code* pada aplikasi daftar hadir Teknik Informatika.
3. Meningkatkan keamanan pada aplikasi daftar hadir mahasiswa Teknik Informatika UHO.

#### **1.5. Manfaat Penelitian**

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Penulis dapat mengimplementasikan algoritma *RC4 (Rivers Code 4)* sebagai metode pengamanan data *QR Code* aplikasi daftar hadir Teknik Informatika.
2. Mengubah sistem daftar hadir mahasiswa yang semula manual menjadi sistem yang terdigitalisasi dan memberikan kemudahan kepada staf jurusan Teknik Informatika UHO dalam dokumentasi daftar hadir.
3. Mengembangkan aplikasi daftar hadir yang praktis dan cepat.

#### **1.6. Sistematika Penulisan**

Sistematika penulisan proposal Tugas Akhir ini adalah sebagai berikut:

#### **BAB I PENDAHULUAN**

Bab ini menjelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan tinjauan pustaka.

#### **BAB II LANDASAN TEORI**

Bab ini memuat pengertian-pengertian dan teori-teori yang menjadi acuan dalam pembuatan analisa dan pemecahan dari permasalahan yang dibahas meliputi QR Code, Autentikasi, RC4, Android, PHP, DBMS, MySQL, dan pendukung lain.

### BAB III METODOLOGI PENELITIAN

Bab ini berisi metode penelitian yang digunakan. Langkah – langkah pengumpulan data, prosedur pengembangan perangkat lunak dan perangkat keras yang dilakukan dalam penelitian.

### BAB IV ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi tentang gambaran umum sistem, desain perangkat keras dan perancangan sistem yang telah dibuat.

### BAB V IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini membahas mengenai implementasi dan pengujian sistem terhadap sistem yang telah dibuat.

### BAB VI PENUTUP

Bab ini berisi kesimpulan yang diambil dari hasil pembuatan sistem serta saran-saran untuk pengembangan dari penulis.

#### 1.7. Tinjauan Pustaka

Pada tahun 2010, Calvin Irwan melakukan penelitian dengan judul Enkripsi Pada *QR Code* Tiket Dengan *RSA*. Berdasarkan pengalaman saat menjadi panitia sebuah konser di Jakarta, peneliti memanfaatkan *QR Code* sebagai tiket masuk. Namun karena tidak dienkripsi, banyak penonton yang berhasil masuk dengan menggunakan tiket palsu. Hal tersebut yang membuat peneliti mulai mengimplementasikan Metode *RSA* untuk ditanamkan ke dalam *QR Code* sehingga lebih aman saat digunakan sebagai tiket pada konser-konser berikutnya.

Pada tahun 2015, Bambang Sugintoro dan Fuad Hasan melakukan penelitian yang berjudul Pengembangan *QR Code Scanner* Berbasis *Android* Untuk Sistem Informasi Museum Sonobudoyo Yogyakarta. Penelitian tersebut memanfaatkan *QR Code* sebagai objek yang dapat discan dan memberikan informasi seputar museum Sonobudoyo Yogyakarta kepada pengunjung menggunakan aplikasi khusus yang telah dibuat.

Pada tahun 2016, Moh. Lukman Sholeh dan Lutfi Ali Muharom melakukan penelitian dengan judul *Smart Presensi Menggunakan QR Code Dengan Enkripsi Vigenere Cipher*.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Data**

Menurut berbagai kamus Bahasa Inggris-Indonesia data dapat diartikan sebagai istilah yang berasal dari kata datum yang berarti bahan-bahan keterangan. Sedangkan menurut Gordon B. Davis dalam bukunya yang berjudul *Management Information System: Conceptual Foundations, Structures, and Development* menyebut data sebagai lambing-lambang tidak acak yang menunjukkan jumlah atau tindakan. Dari beberapa gambaran diatas dapat kita simpulkan bahwa data merupakan bahan baku informasi, yang didefinisikan sebagai kelompok teratur dari symbol-simbol yang mewakili kuantitas, tindakan, benda, dan sebagainya. Data terbentuk dari karakter, dapat berupa symbol khusus seperti \*, \$ dan /. Data disusun untuk diolah dalam bentuk struktur data, struktur file, dan basis data (Zakaria, 2018).

#### **2.2. Autentikasi**

Autentikasi (*Authentication*) adalah suatu proses yang merupakan sebuah tindakan pembuktian (validasi) terhadap identitas seorang pengguna pada saat akan memasuki (mengakses) sebuah sistem. Dimana proses validasi tersebut biasanya menggunakan nama dan password dari seorang pengguna yang nantinya akan dijadikan sebagai penanda (verifikasi) apakah seseorang itu adalah orang yang berhak untuk masuk kedalam sistem tersebut.

Adapun metode-metode autentikasi adalah sebagai berikut

1. *Something you know*

Ini adalah metode autentikasi yang paling umum. Cara ini mengandalkan kerahasiaan informasi, contohnya adalah password dan PIN. Cara ini berasumsi bahwa tidak ada seorangpun yang mengetahui rahasia itu kecuali anda seorang.

2. *Something you have*

Cara ini biasanya merupakan faktor tambahan untuk membuat autentikasi menjadi lebih aman. Cara ini mengandalkan barang yang sifatnya unik, contohnya adalah kartu *magnetic/smartcard*, *hardware token*, *USB token* dan sebagainya. Cara ini berasumsi bahwa tidak ada seorangpun yang memiliki barang tersebut kecuali anda seorang.

3. *Something you are*

Ini adalah metode yang paling jarang dipakai karena faktor teknologi dan manusia juga. Cara ini mengandalkan keunikan bagian-bagian tubuh anda yang tidak mungkin ada pada orang lain seperti sidik jari, suara atau sidik retina. Cara ini berasumsi bahwa bagian tubuh anda seperti sidik jari dan sidik retina, tidak mungkin sama dengan orang lain.

4. *Something you do*

Melibatkan bahwa setiap user dalam melakukan sesuatu dengan cara yang berbeda. Contoh: Penggunaan analisis suara (*voice recognition*), dan analisis tulisan tangan (Jakfar, 2018).

### 2.3. *QR Code (Quick Response Code)*

*QR Code (Quick Response Code)* adalah barcode dua dimensi yang dapat menyimpan data. *QR Code* dikembangkan oleh Denso Corporation, Jepang dan dapat digunakan secara gratis, bahkan untuk keperluan komersial. Berikut adalah contoh dari *QR Code*:



Gambar 2. 1 Contoh *QR Code (Quick Response Code)*

Besaran data yang dapat disimpan bervariasi, tergantung pada versi *QR Code*, ukuran *QR Code* dan tingkat *Error Correction Capability*-nya.

Berikut tabel yang menunjukkan variasi besaran data yang dapat dimuat *QR Code*:

Versi QR Code	Modul	Tingkat Error Correction Capability	Jumlah Data (bits)
1	21x21	L	152
		M	128
		Q	104
		H	72
2	25x25	L	272
		M	224
		Q	176
		H	128
3	29x29	L	440
		M	352
		Q	272
		H	208

Gambar 2. 2 Besaran Data *QR Code*

*Error Correction Capability (ECC)* menunjukkan batasan *QR code* masih dapat terbaca ketika terjadi kerusakan pada *QR code* tersebut. *QR Code* dapat rusak ketika dicetak dan diletakkan pada kemasan produk dapat rusak.

### 2.3.1. Keuntungan dan Kerugian *QR Code*

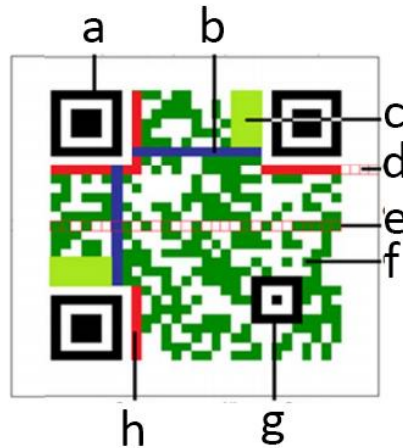
Berikut adalah pertimbangan dalam implementasi *QR Code*. Keuntungan penggunaan *QR Code* antara lain:

1. Gratis dalam pembuatan dan penggunaannya
2. Tersedia *QR Code* scanner gratis
3. Menghemat kertas
4. Ukuran kecil
5. Tidak perlu membeli perangkat khusus scan *QR Code*
6. Sistem dapat cepat memberikan respon terkait hasil scan.

Sedangkan kekurangan *QR code* antara lain:

1. *QR code* hanya mudah diakses oleh pengguna smartphone.
2. Pengguna smartphone harus download aplikasi *QR Code Scanner* terlebih dahulu.

### 2.3.2. Anatomi QR Code



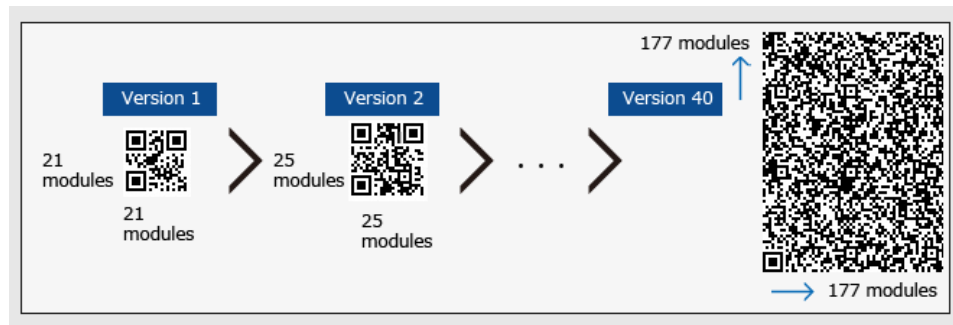
Gambar 2. 3 Anatomi QR Code (Sumber: qrcode.com)

Gambar di atas menyajikan struktur dari sebuah QR Code dan berikut istilah-istilah yang berkenaan dengan QR Code menurut Ariadi (2011):

- a. Finding Pattern : Pola untuk mendeteksi posisi QR Code.
- b. Timing Pattern : Pola yang digunakan untuk identifikasi koordinat pusat dari QR Code. Dibuat dalam bentuk modul hitam putih bergantian.
- c. Version Information : Versi dari sebuah QR Code. Versi terkecil adalah 1 (21 x 21 modul) dan versi terbesar adalah 40 (177 x 177 modul).
- d. Quiet Zone : Daerah kosong dibagian terluar QR Code yang mempermudah mengenali pengenalan QR Code oleh sensor CCD.
- e. QR Code Version : Versi QR Code. Pada contoh gambar versi yang digunakan adalah versi 3 (29 x 29 modul).
- f. Data : Daerah tempat data tersimpan atau data dikodekan.
- g. Alignment Pattern : Pola yang digunakan untuk memperbaiki penyimpanan QR Code terutama distorsi non linear.
- h. Format Information : Informasi tentang *error correction level* dan *mark pattern*.



### 2.3.3. Versi QR Code



Gambar 2. 4 Versi QR Code (Sumber: qrcode.com)

Versi simbol QR Code terdiri dari versi 1 sampai versi 40. Setiap versi memiliki konfigurasi dan jumlah modul yang berbeda-beda. (Modul ini mengacu pada titik-titik hitam dan putih yang membentuk QR Code). Konfigurasi modul mengacu pada jumlah modul yang terkandung dalam simbol dari Versi 1 ( $21 \times 21$  modul) hingga Versi 40 ( $177 \times 177$  modul). Setiap nomor versi yang lebih tinggi berisi 4 modul tambahan di setiap sisi.

### 2.3.4. Macam-Macam QR Code

Adapun macam-macam QR Code yaitu sebagai berikut:

- a. QR Code model 1 dan model 2



QR Code Model 1

Gambar 2. 5 QR Code Model 1 (Sumber: qrcode.com)

Model 1 adalah *Qr Code* asli, dapat menampung 1.167 angka dengan versi maksimum 14 ( $73 \times 73$  modul) (qrcode.com, 2013).

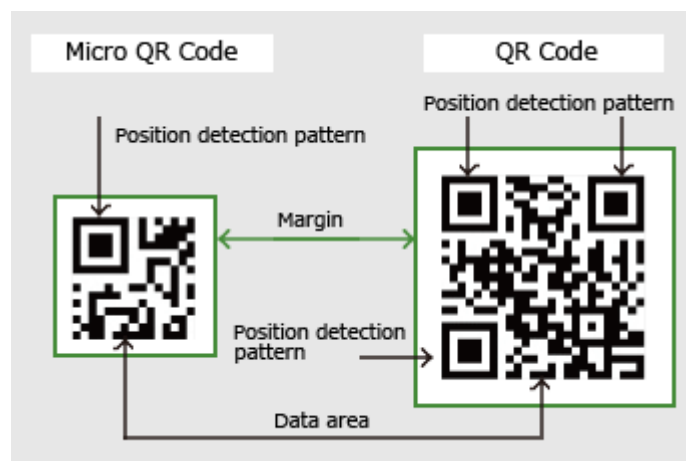


QR Code Model 2

*Gambar 2. 6 QR Code Model 2 (Sumber: qrcode.com)*

Model 2 adalah penyempurnaan dari model 1 dengan versi terbesar 40 (177 x 177 modules), yang mampu menyimpan sampai 7.089 angka (qrcode.com, 2013).

b. *Micro QR Code*



*Gambar 2. 7 Micro QR Code (Sumber: qrcode.com)*

Versi terbesar dari kode ini adalah M4 (17x17 modul) yang dapat menyimpan hingga 35 angka. Fitur utama dari Micro Qr Code adalah hanya memiliki satu pola deteksi posisi, dibandingkan dengan regular QR Code yang memerlukan sejumlah tempat karena pola deteksi posisi yang terletak di tiga sudut simbol. Qr Code biasa membutuhkan setidaknya empat modul yang lebar di sekitar simbol, sedangkan Micro QR Code hanya membutuhkan cukup dua modul margin. Konfigurasi Micro Qr Code memungkinkan pencetakan di tempat lebih kecil dari Qr Code (qrcode.com,2013).

c. *iQr Code*



*Gambar 2. 8 iQR Code (Sumber: qrcode.com)*

Kode yang dapat dihasilkan dari salah satu modul, persegi atau persegi panjang. Dan dapat di cetak sebagai kode inversi hitam putih atau kode pola dot (bagian penanda). Versi terbesar dari kode ini dapat mencapai 61 (422x422 modul), yang dapat menyimpan 40.000 angka (qrcode.com, 2013).

d. *SQRC*



*Gambar 2. 9 SQRC (Sumber: qrcode.com)*

Jenis Qr Code ini dilengkapi dengan membaca fungsi pembatas. Ini dapat digunakan untuk menyimpan informasi pribadi untuk mengelola informasi internal perusahaan dan sejenisnya (qrcode.com, 2013).

e. *Frame QR / LogoQ*



*Gambar 2. 10 Frame QR (Sumber: qrcode.com)*

Jenis Qr Code yang dapat menggabungkan fitur desain tingkat tinggi seperti ilustrasi, huruf dan logo. Qr Code ini menggunakan Logika Since proprietary (qrcode.com,2013).

## **2.4. Kriptografi**

Kriptografi adalah ilmu mengenai teknik enkripsi dimana “naskah asli” (plaintext) diacak menggunakan suatu kunci enkripsi menjadi “naskah acak yang sulit dibaca” (ciphertext) oleh seseorang yang tidak memiliki kunci dekripsi. Dekripsi menggunakan kunci dekripsi bisa mendapatkan kembali data asli. Probabilitas mendapat kembali naskah asli oleh seseorang yang tidak mempunyai kunci dekripsi dalam waktu yang tidak terlalu lama adalah sangat kecil.

Teknik enkripsi yang digunakan dalam kriptografi klasik adalah enkripsi simetris dimana kunci dekripsi sama dengan kunci enkripsi. Untuk public key cryptography, diperlukan teknik enkripsi asimetris dimana kunci dekripsi tidak sama dengan kunci enkripsi. Enkripsi, dekripsi dan pembuatan kunci untuk teknik enkripsi asimetris memerlukan komputasi yang lebih intensif dibandingkan enkripsi simetris, karena enkripsi asimetris menggunakan bilangan – bilangan yang sangat besar. (Kromodimoeljo, 2010).

### **2.4.1. Sejarah Kriptografi**

Kriptografi sudah digunakan jauh sejak zaman dahulu. Penerapan kriptografi yang pertama kali (yang sudah ditemukan) adalah *hieroglyphics* yang

diterapkan oleh bangsa mesir kuno sejak 3000 tahun sebelum masehi. Selanjutnya pada 400 SM bangsa spartan di Yunani juga menerapkan kriptografi di bidang militer. Mereka menggunakan alat yang disebut dengan *scytale*, yakni pita panjang terbuat dari bahan *papyrus*, cara membaca pesannya yaitu dengan menggulungkan pita tadi pada batang silinder. Bangsa Cina dan Jepang mulai mengenal kriptografi pada abad ke 15 M

Peradaban islam juga menggunakan ilmu kriptografi ini. Ilmuwan muslim yang bernama lengkap Abu Yusuf Ya'qub ibn 'Ishaq as-Shabbah al Kindi ini memaparkan tentang kriptanalisis dengan bukunya yang berjudul *Risalah fi Istikhraj al-Mu'amma* (Manuskrip untuk memecahkan pesan-pesan Kriptografi). Beliau terinspirasi dari keindahan Al-Qur'an sehingga beliaulah yang menemukan teknik analisis frekuensi dalam ilmu kriptografi, yakni teknik untuk memecahkan ciphertext berdasarkan frekuensi kemunculan karakter pada sebuah pesan.

#### **2.4.2. Tujuan Kriptografi**

Ada empat tujuan dasar dalam menggunakan kriptografi yaitu:

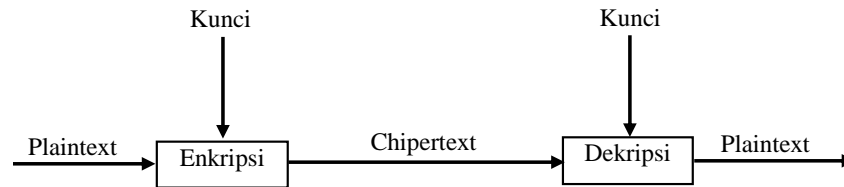
1. Menjaga kerahasiaan agar informasi yang tersedia tidak dapat di akses atau tidak dapat diketahui oleh pihak yang tidak memiliki otoritas untuk mengetahui informasi yang telah di isikan sandi.
2. Menjaga keutuhan informasi agar pada saat informasi di transmisikan tidak terjadi perubahan oleh pihak yang tidak memiliki otoritas untuk mengubahnya. Contohnya untuk menghindari pinyisipan, penghapusan dan pensubsitusian datalain kedalam data aslinya.
3. Memasitikan identitas antar dua pihak yang saling berkomunikasi dan memastikan keaslian dari isi data, waktu pengirimanm, dll.
4. Mencegah penyangkalan bahwa data yang dikirimkan memang berasal dari pengirim yang benar yang telah sesuai dengan perjanjian.

#### **2.4.3. Pembagian Algoritma Kriptografi**

Berdasarkan kunci pemecahannya algoritma kriptografi juga dibagi dalam dua jenis

- a. Kriptografi Simetri

Algoritma kriptografi ini menggunakan kunci pemecahan yang sama saat enkripsi maupun dekripsi. Sehingga algoritma ini sering disebut dengan algoritma kunci tunggal.

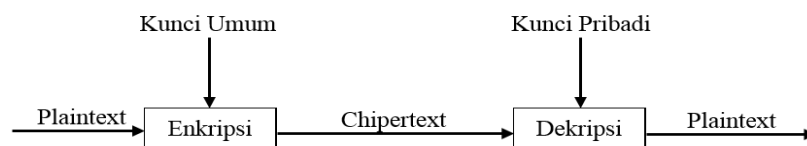


Gambar 2. 11. Proses Enkripsi Deskripsi Menggunakan Kriptografi Simetri

Kunci simetri meliputi enkripsi yang menggunakan algoritma Tiny Encryption Algorithm (TEA), OTP, Data Encryption Standard DES, Rivest Code 4 (RC4), Twofish, Rijndael, AES dan Blowfish.

#### b. Kriptografi Asimetri

Algoritma ini menerapkan kunci yang berbeda saat proses enkripsi maupun dekripsi. Misal saat proses enkripsi menggunakan kunci A, namun saat proses dekripsinya kita diharuskan untuk menggunakan kunci B. Kunci A dalam algoritma ini sering disebut dengan kunci publik. Sedangkan kunci B disebut kunci rahasia (private key).



Gambar 2. 12. Proses Enkripsi Deskripsi Menggunakan Kriptografi Asimetri

Algoritma kriptografi asimetri meliputi algoritma ECC, LUC, RSA, El Gamal, DH, DSA dan lain sebagainya.

### 2.5. Algoritma RC4

RC4 merupakan jenis aliran kode yang berarti operasi enkripsinya dilakukan per karakter 1byte untuk sekali operasi. Algoritma kriptografi *Rivest Code 4 (RC4)* merupakan salah satu algoritma kunci simetris dibuat oleh RSA DataSecurity Inc (RSADSI) yang berbentuk stream cipher. Algoritma ini ditemukan pada tahun 1987

oleh Ronald Rivest dan menjadi simbol keamanan RSA (merupakan singkatan dari tiga nama penemu yaitu Rivest, Shamir, dan Adleman) (Ariyus, 2008).

Secara garis besar algoritma dari metode RC4 StreamCipher ini terbagi menjadi dua bagian, yaitu Key Setup atau Key Scheduling Algorithm (KSA) dan Stream Generation atau Pseudo Random Generation Algorithm (PRGA) dan proses XOR dengan stream data.

RC4 menghasilkan *pseudorandom stream bit*. Seperti halnya stream cipher lainnya, algoritma RC4 ini dapat digunakan untuk mengenkripsi dengan mengombinasikannya dengan plainteks dengan menggunakan *bit-wise XOR* (Exclusive-or). Proses dekripsinya dilakukan dengan cara yang sama (karena XOR merupakan fungsi simetrik). Untuk menghasilkan *key-stream*, cipher menggunakan *state internal* yang meliputi dua bagian :

1. Sebuah permutasi dari 256 kemungkinan *byte*.
2. 2 Indeks-pointer 8-bit.

Permutasi diinisialisasi dengan sebuah variabel panjang kunci, biasanya antara 40 sampai 256 bit dengan menggunakan algoritma *key-scheduling* (KSA). Setelah proses ini selesai, *stream* yang terdiri dari sekumpulan bit tersebut terbentuk dengan menggunakan Pseudo-Random Generation Algorithm (PRGA). Berikut ini akan dijelaskan tentang kedua algoritma tersebut.

### 1. *Key-Scheduling Algorithm (KSA)*

*Algoritma key scheduling* digunakan untuk menginisialisasi permutasi di array “S”. panjang kunci didefinisikan sebagai jumlah *byte* di kunci dan mempunyai rentang panjang kunci dari 1 sampai 256, khususnya antara 5-16 tergantung dari panjang kunci 40-128 bit. Pertama-tama array “S” diinisialisasi untuk identitas permutasi. S kemudian diproses ke 256 iterasi dengan cara yang

```

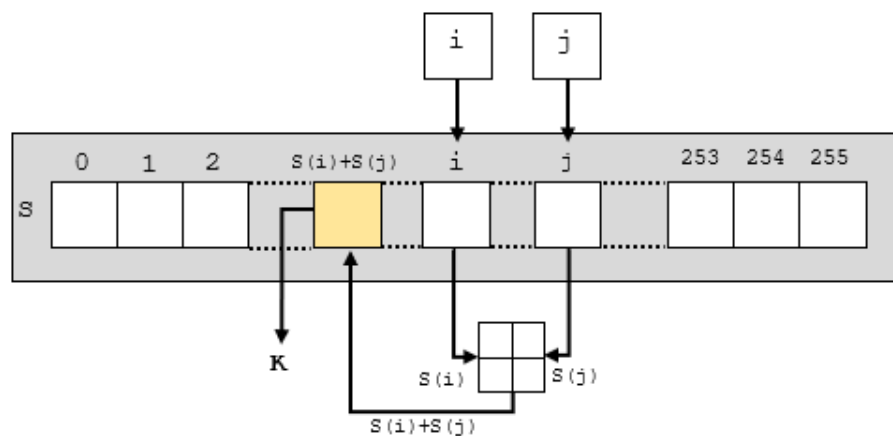
for i from 0 to 255
    S[i] := i
endfor
j := 0
for i from 0 to 255
    j := (j + S[i] + key[i mod keylength]) mod 256
    swap values of S[i] and S[j]
endfor

```

sama dengan PRGA utama, tapi juga dikombinasikan dalam byte dari kunci dalam waktu yang bersamaan. Berikut adalah algoritma KSA :

## 2. Pseudo-Random Generation Algorithm (PRGA)

PRGA (Pseudo-Random Generation Algoritihm) memodifikasi *state* dan *output* sebuah *byte* dari *key-stream*. Hal ini penting karena banyaknya dibutuhkan iterasi. Dalam setiap iterasi, PRGA menginkremen  $i$ , menambahkan nilai  $S$  yang ditunjuk oleh  $i$  sampai  $j$ , kemudian menukar nilai  $S[i]$  dan  $S[j]$ , lalu mengembalikan elemen dari  $S$  di lokasi  $S[i] + S[j]$  (modulo 256). Setiap elemen  $S$  ditukar dengan elemen lainnya paling tidak satu kali setiap 256 iterasi.



Gambar 2. 13. Proses *Pseudo Random* Pada Algoritma RC4

Realisasi dari algoritma PRGA yaitu :

```

i := 0
j := 0
while GeneratingOutput:
  i := (i + 1) mod 256
  j := (j + S[i]) mod 256
  swap values of S[i] and S[j]
  K := S[(S[i] + S[j]) mod 256]
  output K
endwhile

```



### 2.5.1 Cara Kerja Algoritma RC4 Stream Chiper

Cara kerja algoritma RC4 yaitu inisialisasi *S-Box* pertama,  $S[0]$ ,  $S[1]$ , ...,  $S[255]$ , dengan bilangan 0 sampai 255. Pertama isi secara berurutan  $S[0]=0$ ,  $S[1]=1$ , ...,  $S[255]=255$ . Kemudian inisialisasi *array* lain (*S-Box* lain), misal *array* K dengan panjang 256. Isi *array* K dengan kunci diulangi sampai seluruh *array*  $K[0]$ ,  $K[1]$ , ...,  $K[255]$  terisi seluruhnya. Setelah itu menyimpan key dalam *Key Byte Array*, Permutasi pada *S-Box*. Pada *Stream Generation* akan menghasilkan nilai *pseudorandom* yang akan dikenakan operasi XOR untuk menghasilkan *ciphertext* ataupun sebaliknya yaitu untuk menghasilkan *plaintext*. Berikut cara kerja dari algoritma RC4 dengan menggunakan 4-bit kunci.

Array S : 0 1 2 3

Array K : 2 5 7 3

Inisiasi  $i$  dan  $j$  dengan 0, kemudian dilakukan KSA agar tercipta *state-array* yang acak. Penjelasan lebih lanjut adalah sebagai berikut.

#### Iterasi 1

$i = 0$

$j = (0 + S[0] + K[0 \bmod 4]) \bmod 4 = (0 + 0 + 2) \bmod 4 = 2$

*swap* ( $S[0]$ ,  $S[2]$ )

hasil *array* S : 2 1 0 3

#### Iterasi 2

$i = 1$

$j = (2 + S[1] + K[1 \bmod 4]) \bmod 4 = (2 + 1 + 5) \bmod 4 = 0$

*swap* ( $S[1]$ ,  $S[0]$ )

hasil *array* S : 1 2 0 3

**Iterasi 3**

$$i = 2$$

$$j = (0 + S[2] + K [2 \bmod 4]) \bmod 4 = (0 + 0 + 7) \bmod 4 = 3$$

*swap* (S[2], S[3])

hasil *array* S : 1 2 3 0

**Iterasi 4**

$$i = 3$$

$$j = (3 + S[3] + K [3 \bmod 4]) \bmod 4 = (3 + 0 + 3) \bmod 4 = 2$$

*swap* (S[3], S[2])

hasil *array* S : 1 2 0 3

Setelah melakukan KSA, akan dilakukan PRGA. PRGA akan dilakukan sebanyak 4 kali dikarenakan plainteks yang akan dienkripsi berjumlah 4 karakter. Hal ini disebabkan karena dibutuhkan 1 kunci dan 1 kali pengoperasian XOR untuk tiap karakter pada plainteks. Berikut adalah tahapan penghasilan kunci enkripsi dengan PRGA.

*Araay* S : 1 2 0 3

Inisialisasi

$$i = 0$$

$$j = 0$$

**Iterasi 1**

$$i = (0 + 1) \bmod 4 = 1$$

$$j = (0 + S[1]) \bmod 4 = (0 + 2) \bmod 4 = 2$$

*swap* (S[1], S[2])

1 0 2 3

$$K1 = S[(S[1] + S[2]) \bmod 4] = S[2 \bmod 4] = 2$$

$$K1 = 00000010$$

### **Iterasi 2**

$$i = (1 + 1) \bmod 4 = 2$$

$$j = (2 + S[2]) \bmod 4 = (2 + 2) \bmod 4 = 0$$

*swap* (S[2], S[0])

2 0 1 3

$$K2 = S[(S[2] + S[0]) \bmod 4] = S[3 \bmod 4] = 3$$

$$K2 = 00000011$$

### **Iterasi 3**

$$i = (2 + 1) \bmod 4 = 3$$

$$j = (0 + S[3]) \bmod 4 = (0 + 3) \bmod 4 = 3$$

*swap* (S[3], S[3])

1 0 2 3

$$K3 = S[(S[3] + S[3]) \bmod 4] = S[6 \bmod 4] = 2$$

$$K3 = 00000010$$

### **Iterasi 4**

$$i = (3 + 1) \bmod 4 = 0$$

$$j = (3 + S[0]) \bmod 4 = (3 + 1) \bmod 4 = 0$$

*swap* (S[0], S[0])

1 0 2 3

$$K1 = S[(S[0] + S[0]) \bmod 4] = S[2 \bmod 4] = 2$$

$$K1 = 00000010$$

Berikut adalah tahapan penghasiian kunci enkripsi dengan PRGA.

Array S : 1 2 0 3

Inisialisasi

$$i = 0$$

$$j = 0$$

### Iterasi 1

$$i = (0 + 1) \bmod 4 = 1$$

$$j = (0 + S[1]) \bmod 4 = (0 + 2) \bmod 4 = 2$$

*swap* (S[1], S[2])

1 0 2 3

$$K1 = S[(S[1] + S[2]) \bmod 4] = S[2 \bmod 4] = 2$$

$$K1 = 00000010$$

### Iterasi 2

$$i = (1 + 1) \bmod 4 = 2$$

$$j = (2 + S[2]) \bmod 4 = (2 + 2) \bmod 4 = 0$$

*swap* (S[2], S[0])

2 0 1 3

$$K2 = S[(S[2] + S[0]) \bmod 4] = S[3 \bmod 4] = 3$$

$$K2 = 00000011$$

### Iterasi 3

$$i = (2 + 1) \bmod 4 = 3$$

$$j = (0 + S[3]) \bmod 4 = (0 + 3) \bmod 4 = 3$$

*swap* (S[3], S[3])

2 0 1 3

$$K3 = S[(S[3] + S[3]) \bmod 4] = S[6 \bmod 4] = 1$$

$$K3 = 00000001$$

### Iterasi 4

$$i = (3 + 1) \bmod 4 = 0$$

$$j = (3 + S[0]) \bmod 4 = (3 + 2) \bmod 4 = 1$$

*swap* (S[0], S[1])

0 2 1 3

$$K4 = S[(S[0] + S[1]) \bmod 4] = S[2 \bmod 4] = 1$$

$$K4 = 00000001$$

Proses XOR kunci enkripsi dengan plainteks

H A L O : 01001000 01000001 01001100 01001111

Key : 00000010 00000011 00000001 00000001

Chiperteks : 01001010 01000010 01001101 01001110

(J) (B) (M) (N)

## 2.6. PHP

PHP Kepanjangan dari PHP adalah "*Hypertext Preprocessor*" (ini merupakan singkatan rekursif). PHP adalah bahasa scripting web *HTML-embedded*. Ini berarti kode PHP dapat disisipkan ke dalam HTML halaman Web. Ketika sebuah halaman PHP diakses, kode PHP dibaca atau "diurai" oleh *server*. *Output* dari fungsi PHP pada halaman biasanya dikembalikan sebagai kode HTML, yang dapat dibaca oleh *browser*. Karena kode PHP diubah menjadi HTML sebelum halaman dibuka, pengguna tidak dapat melihat kode PHP pada halaman. Ini membuat halaman PHP cukup aman untuk mengakses database dan informasi aman lainnya. (Ferdianto, 2013)

Banyak sintaks PHP yang hasil adaptasi dari bahasa lain seperti bahasa C, Java dan Perl. Namun, PHP memiliki sejumlah fitur unik dan fungsi tertentu juga. Tujuan dari bahasa pemrograman PHP adalah untuk memungkinkan pengembangan web untuk menulis halaman yang dihasilkan secara dinamis dengan cepat dan mudah. PHP juga bagus untuk menciptakan situs Web *database-driven*. Jika Anda ingin mempelajari lebih lanjut tentang PHP, situs resminya yaitu PHP.net. (Ferdianto, 2013)

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya *milis - milis* dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah system. (Ferdianto, 2013).

## 2.7. *Hypertext Markup Language (HTML)*

HTML adalah sebuah bahasa pemrograman yang berbentuk skrip-skrip yang berguna untuk membuat sebuah halaman *web*. HTML dapat dibaca oleh berbagai *platform* seperti: Windows, Linux, Macintosh. Kata *Markup Language* pada HTML menunjukkan fasilitas yang berupa tanda tertentu dalam skrip HTML dimana pengguna bisa mengatur judul, garis, tabel, gambar, dan lain-lain dengan perintah yang telah ditentukan pada elemen HTML. HTML sendiri dikeluarkan oleh W3C (*World Wide Web Consortium*), setiap terjadi perkembangan level HTML harus dievaluasi ketat dan disetujui oleh W3C. Contoh tag HTML antara lain: `<head>`, `<body>` dan `<table>` (Kadir, 2002).

## 2.8. *Cascading Style Sheet (CSS)*

*Cascading Style Sheets (CSS)* adalah salah satu bahasa pemrograman desain *web (style sheet language)* yang mengontrol format tampilan sebuah halaman *web* yang ditulis dengan menggunakan bahasa penanda (*markup language*). Biasanya CSS digunakan untuk mendesain sebuah halaman HTML dan XHTML, tetapi sekarang bahasa pemrograman CSS bisa diaplikasikan untuk segala dokumen XML, termasuk SVG dan XUL. CSS dibuat untuk memisahkan konteks utama (biasanya dibuat dengan menggunakan bahasa HTML dan sejenisnya) dengan tampilan dokumen yang meliputi *layout*, warna dan *font*. Pemisahan ini dapat meningkatkan daya akses konten pada *web*, menyediakan lebih banyak fleksibilitas dan control dalam spesifikasi dari sebuah karakteristik dari sebuah tampilan, memungkinkan untuk membagi banyak halaman untuk sebuah formatting dan mengurangi kerumitan dalam penulisan kode dan struktur dari konten, contohnya teknik tables pada layout desain *web (layout tanpa tabel)*

*Style Sheet* adalah sebuah *text file* yang sederhana (dimana berekstensi \*.css), ditulis menurut aturan bahasa yang dipaparkan pada rekomendasi CSS1 atau CSS2. Cara kerja CSS dengan menggunakan dua buah elemen penting untuk pemformatan tampilannya, diantaranya selektor dan deklarator. Dua buah elemen ini berfungsi sebagai penentu format tampilan dan lainnya menempatkan format tampilan tersebut. Deklarator berisi beberapa perintah-perintah CSS untuk menentukan format dari sebuah elemen pada halaman *web*. Sedangkan selektor adalah sebuah

perintah lanjut dari deklarator dan berfungsi menempatkan format tampilan dari deklarator.

Dalam *Cascading Style Sheets* ada dua cara menghubungkan sebuah dokumen HTML dengan CSS

1. *Selector class*
2. *Selector ID*
3. *Selector Descendant* (turunan)
4. *Selector Link Pseudo Class*
5. *Selector Pseudo elemen*
6. *Selector dynamic pseudo class*
7. *Selector languange*
8. *Selector child*
9. *Selector first-child*
10. *Selector adjacent* (berdekatan)

## **2.9. Android**

*Android* adalah sistem operasi yang dikeluarkan oleh Google. Android dibuat khusus untuk smartphone dan tablet. Berbagai macam produsen telah menggunakan Android sebagai sistem operasi untuk peranti (device) yang mereka produksi. Android juga mempunyai store dengan lebih dari 2 miliar pengguna aktif, per Januari 2018 ketika tulisan ini dibuat.

Sejak dirilis tahun 2008, Google telah mengeluarkan beberapa versi, dengan "Pie" sebagai versi yang terbaru.

Pada tahun 2013, Android menjadi operation system (OS) terlaris pada tablet dan smartphone. Tercatat pada tahun 2016, store Android memiliki lebih dari 2.8 juta aplikasi.

Android menarik bagi perusahaan teknologi yang membutuhkan barang siap jadi, biaya rendah dan kustomisasi OS untuk perangkat teknologi tinggi mereka. Hal ini menjadi daya tarik bagi banyak perusahaan, sehingga mereka memilih Android.



Source code dari Android bersifat open source. Ini adalah hal menarik bagi komunitas developer, karena lisensi open source sangat mendukung untuk mengembangkan produknya dengan aman.

Versi-versi android antara lain :

1. Android versi 1.1
2. Android versi 1.5 (Cupcake)
3. Android versi 1.6 (Donut)
4. Android versi 2.0 / 2.1 (Eclair)
5. Android versi 2.2 Froyo (Frozen Yoghurt)
6. Android versi 2.3 (Gingerbread)
7. Android versi 3.0/3.1 (Honeycomb)
8. Android versi 4.0 ICS (Ice Cream Sandwich)
9. Android versi 4.1 – 4.3 (Jelly Bean)
10. Android versi 4.4 (Kitkat)
11. Android versi 5.0 – 5.1 (Lollipop)
12. Android versi 6.0 (Marshmallow)
13. Android versi 7.0 (Nougat)
14. Android versi 8.0 / 8.1 (Oreo)
15. Android versi 9.0 (Pie)

## 2.10. MySQL

*MySQL* adalah aplikasi *SQL database server* yang multi *user*. Oleh karena itu, *MySQL* digunakan dalam pembuatan aplikasi ini yang digunakan sebagai *database server* untuk menyimpan data lokasi yang dikirim oleh masing-masing telepon seluler. Dalam *MYSQL* ada beberapa operasi dasar yang lebih dikenal dengan operasi CRUD yaitu :

### 1. Create

*Create* adalah operasi penambahan data baru ke dalam table. Terdapat 2 *Query* untuk menambah data, yang pertama adalah penambahan yang tidak menspesifikasikan nama kolom yang akan ditambahkan dan hanya memberikan isi dari tabelnya. *Query*nya adalah sebagai berikut:

```
INSERT INTO table_name
VALUES (value1, value2, value3,...);
```

Kedua adalah menginputkan data dengan mencantumkan nama kolom yang akan diisi dengan isi di dalamnya. *Query*nya adalah sebagai berikut:

```
INSERT INTO table_name (column1,column2,column3,...)
VALUES (value1, value2, value3,...);
```

## 2. Read

*Read* adalah operasi untuk menampilkan semua atau sebagian data yang berada di dalam *database*. *Query*nya adalah sebagai berikut:

```
SELECT column_name, column_name
FROM table_name;
```

## 3. Update

*Update* adalah operasi untuk mengubah data yang ada di dalam *database*. *Query*nya adalah sebagai berikut:

```
UPDATE table_name
SET column1=value1, column2=value2,...
WHERE some_column=some_value;
```

## 4. Delete

*Delete* adalah operasi untuk menghapus data yang ada di dalam table. *Query*nya adalah sebagai berikut:

```
DELETE FROM table_name
WHERE some_column = some_value;
```

**Tabel 2.1 Tipe Data Database MySQL**

No.	Type	Keterangan
1.	<i>Varchar</i>	Tipe data karakter yang panjangnya tidak tetap
2.	TINYINT	Adalah tipe data bilangan bulat yang rentangnya -128 sampai 127 atau menggunakan atribut unsigned dari 0 hingga 255
3.	<i>Text</i>	Tipe data yang dapat menampung semua tipe data.
4.	<i>Date</i>	Tipe data yang digunakan untuk mendiskripsikan tanggal.

5.	<i>Smallint</i>	Adalah tipe data bilangan bulat yang rentangnya -32768 sampai 32767. The unsigned range is 0 to 65535
6.	<i>Mediumint</i>	Adalah tipe data bilangan bulat yang rentangnya -8388608 to 8388607. unsigned range-nya 0 sampai 16777215
7.	<i>Int</i>	Tipe data yang bernilai integer/bilangan bulat.
8.	<i>Time</i>	Tipe data waktu. Jangkauannya adalah '-838:59:59' hingga '838:59:59'. MySQL menampilkan TIME dalam format 'HH:MM:SS'.
9.	<i>Char</i>	Tipe data untuk menampung data yang bertipe karakter
10.	<i>Primary Key</i>	Kunci primer adalah suatu atribut atau satu set minimal atribut yang tidak hanya mendefinisikan secara unik suatu kejadian spesifik tetapi juga dapat mewakili setiap kejadian dari suatu kejadian

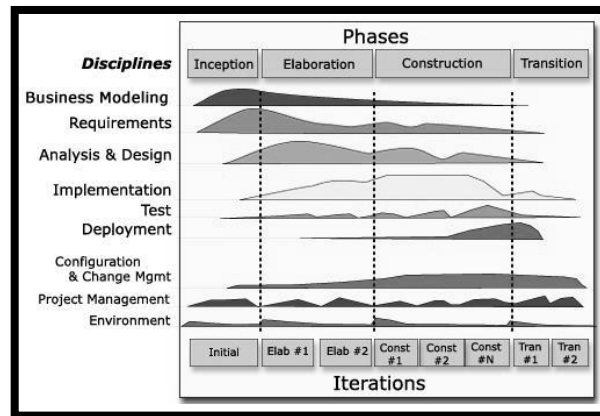
### 2.11. Rational Unified Process (RUP)

*Rational Unified Process* (RUP) merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan *use-case driven* dan pendekatan iteratif untuk siklus pengembangan perangkat lunak. Gambar dibawah menunjukkan secara keseluruhan arsitektur yang dimiliki RUP.

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Melalui Gambar 2.5 dapat dilihat bahwa RUP memiliki 2 dimensi yaitu:

1. Dimensi pertama digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari fase selanjutnya. Setiap fase dapat berdiri dari satu beberapa iterasi. Dimensi ini terdiri atas *Inception*, *Elaboration*, *Construction*, dan *Transition*.
2. Dimensi kedua digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan

kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing*, *what*, *how* dan *when*. Dimensi ini terdiri atas *Business Modeling*, *Requirement*, *Analysis and Design*, *Implementation*, *Test*, *Deployment*, *Configuration* dan *Change Manegement*, *Project Management*, *Environtment*.



Gambar 2. 14. Arsitektur Rational Unified Process

Dalam metode ini, terdapat empat tahap pengembangan perangkat lunak yaitu:

#### 1. *Inception*

Pada tahap ini pengembang mendefinisikan batasan kegiatan, melakukan analisis kebutuhan user, dan melakukan perancangan awal perangkat lunak (perancangan arsitektural dan *use case*). Pada akhir fase ini, prototipe perangkat lunak versi *Alpha* harus sudah dirilis

#### 2. *Elaboration*

Pada tahap ini dilakukan perancangan perangkat lunak mulai dari menspesifikasikan fitur perangkat lunak hingga perilsan prototipe versi *Betha* dari perangkat lunak.

#### 3. *Construction*

Pengimplementasian rancangan perangkat lunak yang telah dibuat dilakukan pada tahap ini. Pada akhir tahap ini, perangkat lunak versi akhir yang sudah disetujui administrator dirilis beserta dokumentasi perangkat lunak.

#### 4. *Transition*

Instalasi *deployment* dan sosialisasi perangkat lunak dilakukan pada tahap ini. Pada tahapan ini dilakukan tahap evaluasi dari sistem yang telah



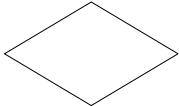



direncanakan, dianalisa dan dibangun. Dari tahap ini bisa dilihat kekurangan dari sistem yang telah dibangun.

### 2.12. Flowchart

*Flowchart* adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung. Jadi, setiap simbol *flowchart* melambangkan pekerjaan dan instruksinya. Simbol-simbol *flowchart* adalah standar yang ditentukan oleh *Amerika National Standard Institute Inc.*

Simbol-simbol yang digunakan dalam *flowchart* ditampilkan pada Tabel 2.2

**Tabel 2.2. Simbol-simbol Flowchart**

NO.	SIMBOL	NAMA	KETERANGAN
1.		Proses	Mempresentasikan operasi.
2.		<i>Input / Output</i>	Mempresentasikan <i>Input</i> data atau <i>Output</i> data yang diproses atau informasi.
3.		Keputusan	Keputusan dalam program.
4.		Dokumen	Dokument I / O dalam format cetak.
5.		<i>Terminal points</i>	Awal / akhir <i>flowchart</i> .
6.		<i>Preparation</i>	Pemberian harga awal.

### 2.13. Unified Modeling Language (UML)

*Unified Modeling Language* (UML) merupakan pengganti dari metode analisis berorientasi objek dan desain berorientasi objek (OOA&D) yang dimunculkan sepenggunar akhir tahun 80-an dan awal tahun 90-an. UML

merupakan gabungan dari metode Booch, Rumbaugh (OMT) dan Jacobson. Tetapi UML ini akan mencakup lebih luas daripada OOA&D. Pada pertengahan pengembangan UML dilakukan standarisasi proses dengan OMG (*Object Management Group*) dengan harapan UML akan menjadi bahasa standar pemodelan pada masa yang akan datang.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat. Bahasa pemodelan merupakan bagian terpenting dari metode. Ini merupakan bagian kunci tertentu untuk komunikasi. UML merupakan bahasa standar untuk penulisan *blue print software* yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak (Shalahudin, 2011).


#### 2.14. Diagram Dalam UML








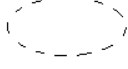
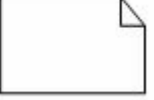
Dalam UML terdapat diagram-diagram yang bisa menggambarkan bagian atau aspek tertentu dari sebuah sistem. Sehingga dapat terlihat jelas alur dan gambaran umum dari perangkat lunak yang dibangun. Ada beberapa jenis diagram dalam UML yaitu:

##### 1. *Use case Diagram*

Menggambarakan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagrams*. *Use case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem (Shalahudin, 2011).

**Tabel 2.3 Simbol *Use Case Diagram***

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>


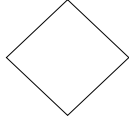
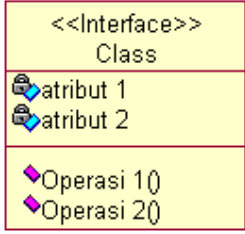

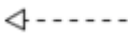

2.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (dependent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent)
3.		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> )
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan
6.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas
8.		<i>Use case</i>	Deskripsi dari uraian aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu actor
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi)
10.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

## 2. Class Diagram

Menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), *specialized* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (grup bersama

sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram* (Rosa dan Shalahudin, 2011).

**Tabel 2.4 Simbol *Class Diagram***

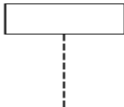

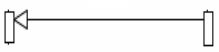
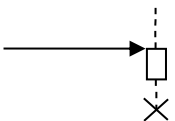
NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
2.		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek
3.		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu actor
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek
6.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri

### 3. *Sequence Diagram*

Menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaanya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem (Shalahudin, 2011).



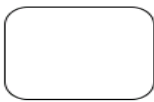


Tabel 2.5 Simbol *Sequence Diagram*



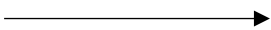
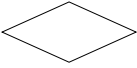
NO	SIMBOL	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
4		<i>Message</i>	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

#### 4. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi (Shalahudin, 2011).

Tabel 2.6 Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actifity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.

4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
6		<i>Line Connector</i>	Digunakan untuk menghubungkan satu simbol dengan simbol yang lain
7		<i>Decision</i>	Menunjukkan suatu keputusan yang mempunyai satu atau lebih transisi sesuai dengan kondisi

## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Waktu dan Tempat Penelitian

##### 3.1.1. Waktu

Waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari bulan Februari 2019 sampai dengan Juli 2019. Rincian kegiatan dapat dilihat pada Tabel 3.1 berikut:

**Tabel 3.1 Gantt Chart Waktu Penelitian**

No	Uraian	Waktu															
		April				Mei				Juni				Juli			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	<i>Inception</i>																
2.	<i>Elaboration</i>																
3.	<i>Construction</i>																
4.	<i>Transition</i>																

##### 3.1.2. Tempat Penelitian

Adapun tempat penelitian tugas akhir yang akan dilakukan di Gedung Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo.

#### 3.2. Metode Pengumpulan Data

Dalam penelitian ini metode pengumpulan data yang digunakan pada perancangan aplikasi tersebut adalah studi literatur, mulai dari buku-buku, jurnal, data dari jurusan Teknik Informatika Fakultas Teknik Universitas Halu Oleo maupun artikel, dan sumber-sumber lain. Metode ini dilaksanakan dengan melakukan studi kepustakaan yang relevan. Studi kepustakaan ini dilakukan untuk mencari sumber pelengkap yang berhubungan dengan sistem yang akan dibangun, yaitu dengan mencari referensi yang membahas tentang penggunaan metode *Rivest Cipher 4 (RC4)*, sehingga metode *RC4* dapat diimplementasikan dalam aplikasi Absensi Mahasiswa Jurusan Teknik Informatika.

### 3.3. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam sistem ini adalah metode Rational Unified Process (RUP). Dalam metode ini, terdapat empat tahap pengembangan perangkat lunak, yaitu:

#### 3.3.1. Permulaan (*Inception*)

Pada fase ini dilakukan proses pengidentifikasian sistem, dilakukan dengan analisis kebutuhan akan aplikasi, melakukan kajian terhadap penelitian yang terkait dengan metode *Rivest Cipher 4 (RC4)*.

#### 3.3.2. Perluasaan / Perencanaan (*Elaboration*)

Setelah menentukan ruang lingkup penelitian, tahap ini akan dilakukan perancangan dan analisis sistem menggunakan flowchart meliputi flowchart Aplikasi Absensi, dan flowchart metode *Rivest Cipher 4 (RC4)*. Pada perancangan ini, digunakan juga *UML (Unified Modelling Language)* yang meliputi use case diagram, activity diagram, class diagram dan sequence diagram.

#### 3.3.3. Konstruksi (*Construction*)

Proses yang dilakukan pada tahap ini yaitu membangun aplikasi dengan perancangan yang telah dilakukan sebelumnya, mulai dari tampilan interface sampai implementasi rancangan UML. Proses yang juga dilakukan pada tahap ini yaitu penerapan coding metode *Rivest Cipher 4 (RC4)* pada sistem.

#### 3.3.4. Transisi (*Transition*)

Pada tahap Transition difokuskan untuk melakukan proses pengujian terhadap aplikasi. Dalam penelitian ini, dilakukan pengujian menggunakan black box terhadap aplikasi yang meliputi pengujian enkripsi data *QR Code* dan deskripsi data dari *QR Code* untuk dikirim ke *web server*.

### **3.4. Analisis Perancangan Sistem**

Perancangan sistem yang akan dibangun terdiri atas perancangan flowchart dan perancangan UML serta perancangan user interface.

## **BAB IV**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **4.1 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 *nonfungsional*.

#### **4.2 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.	<i>PC</i>	<i>Acer Aspire E 14 E5-421-61-C1</i>
2.	<i>Processor</i>	<i>AMD A6 Quad Core</i>
3.	<i>RAM</i>	<i>4 GB</i>
4.	<i>Harddisk</i>	<i>240 GB SSD</i>
5.	<i>Monitor</i>	<i>Monitor 14 Inch</i>

##### 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.	<i>Windows</i>	<i>Operating System</i>	<i>Windows 10 Enterprise</i>

2.	<i>Xampp</i>	<i>Universal development environment</i>	<i>Xampp v3.2.2</i>
3.	<i>Apache</i>	<i>Web Server</i>	<i>Apache 2.4 win64-VC14</i>
4.	<i>Mysql</i>	<i>Database Management</i>	<i>Mysql 5.7</i>
5.	<i>PHP</i>	<i>Web development</i>	<i>PHP 7.3</i>
6.	<i>Dart</i>	<i>Android Development</i>	<i>Dart 3.4.1</i>
7.	<i>Chrome</i>	<i>Web browser</i>	<i>Chrome version 76.0</i>
8.	<i>Android Studio</i>	<i>SDK Development Tool</i>	<i>Android Studio 3.3</i>
9.	<i>Visual Studio Code</i>	<i>Code Editor</i>	<i>VS Code v1.38</i>
10.	<i>Flutter</i>	<i>Framework Android</i>	<i>Flutter 3.4.1</i>

### 4.3. 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)*.

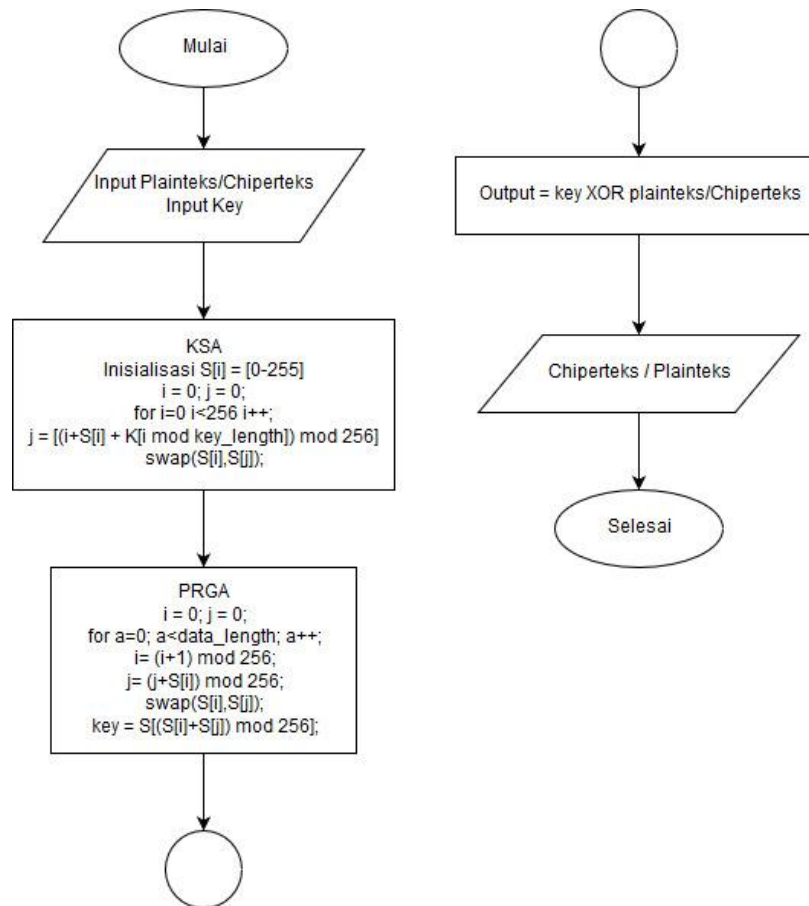
#### 4.3.1. *Flowchart* Sistem

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

##### 4.3.1.1. *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:





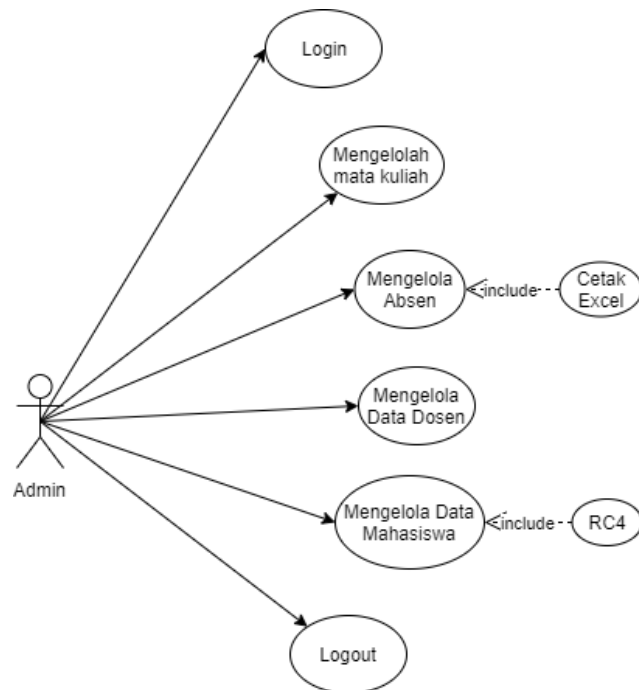
Gambar 4. 1 *Flowchart* Enkripsi dan Dekripsi Algoritma RC4

#### 4.3.2. Unified Modeling Language (UML)

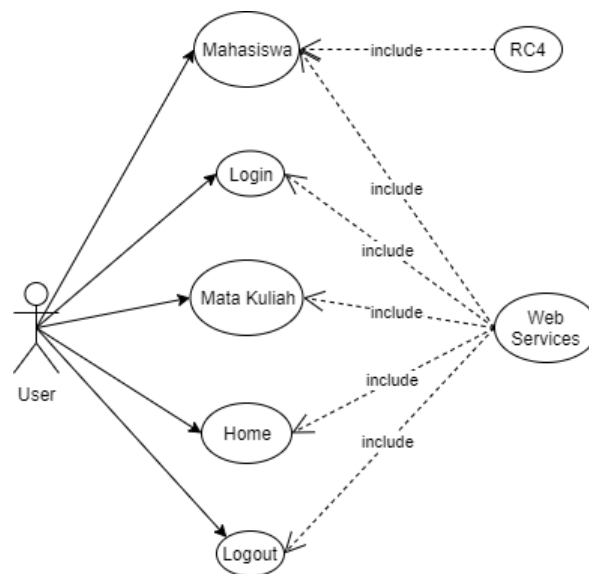
Aplikasi dibangun dengan menggunakan Unified Modeling Language (UML). UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram yang terdiri dari *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*.

##### 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
<i>Admin</i> memilih menu absen	Sistem akan menampilkan data absensi yang diterima dari <i>web service</i> dan menyediakan pilihan cetak absen

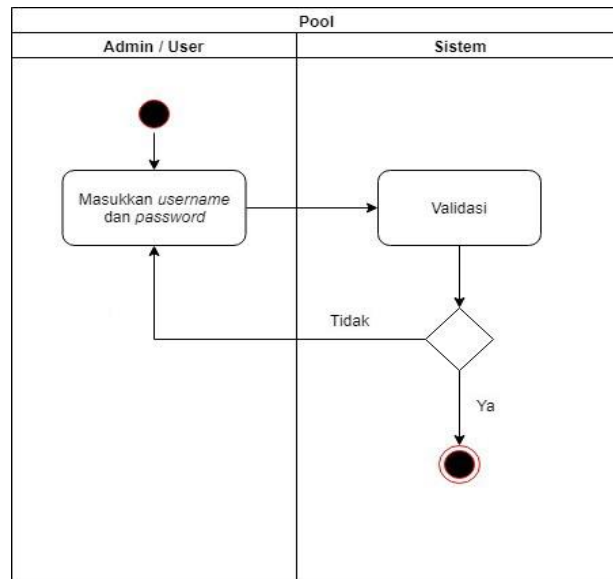
<i>Admin</i> memilih menu mahasiswa	Sistem akan menyediakan menu untuk edit data, hapus data, tampilkan data dan tambah data mahasiswa yang kemudian akan dienkripsi dan tersimpan dalam <i>web service</i>
<i>Admin</i> dan <i>User</i> memilih menu <i>login</i>	Sistem akan menampilkan <i>form login</i>
<i>User</i> memilih menu mata kuliah	Sistem akan menampilkan daftar mata kuliah
<i>User</i> memilih menu <i>scan QR Code</i>	Sistem akan mengaktifkan kamera dan mencari <i>QR Code</i> dan melakukan <i>scanning</i> dan deskripsi pada data <i>QR Code</i>
<i>User</i> memilih menu data kehadiran mahasiswa	Sistem akan menampilkan data kehadiran masing-masing mahasiswa

## 2. 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.

### a. Activity Diagram Login

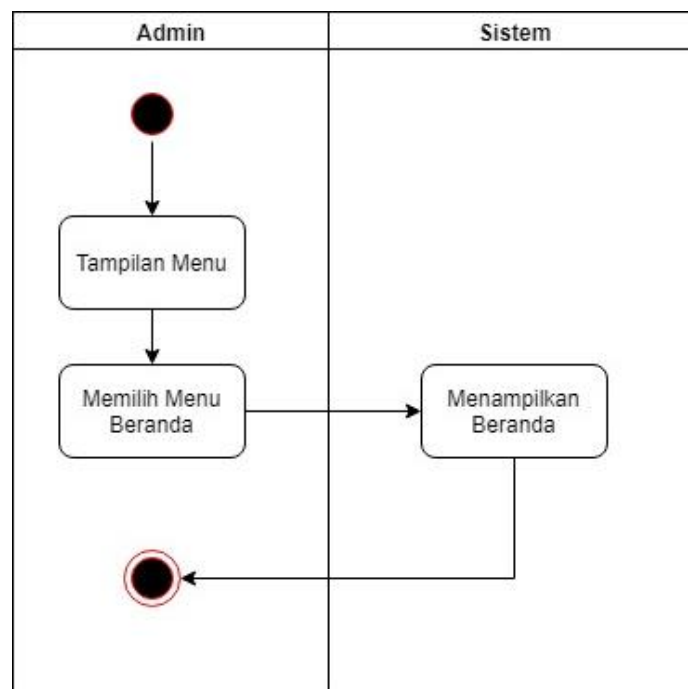
Pada Gambar 4.4 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* dan *user* ketika memasukkan *username* dan *password*, lalu sistem akan menampilkan melakukan validasi.



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

**b. Activity Diagram Beranda/Dashboard**

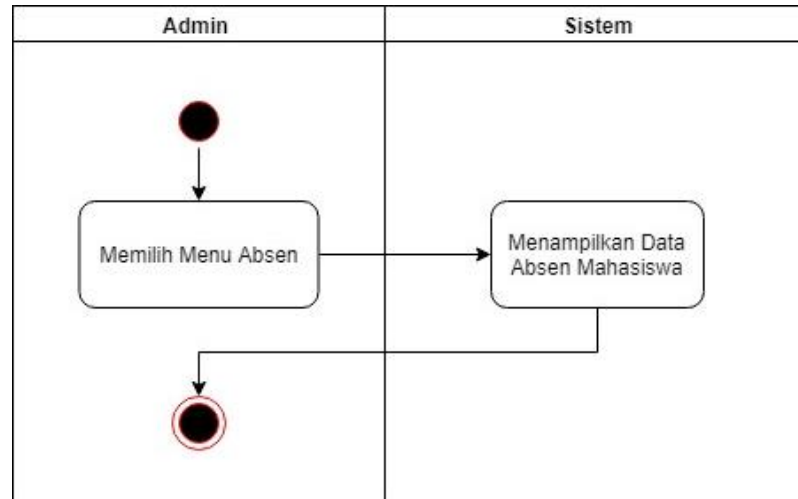
Pada Gambar 4.5 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* ketika memilih menu beranda, lalu sistem akan menampilkan menu beranda yang berisikan tampilan menu beranda.



Gambar 4. 5 Activity Diagram Beranda

**c. Activity Diagram Tampil Data Absensi**

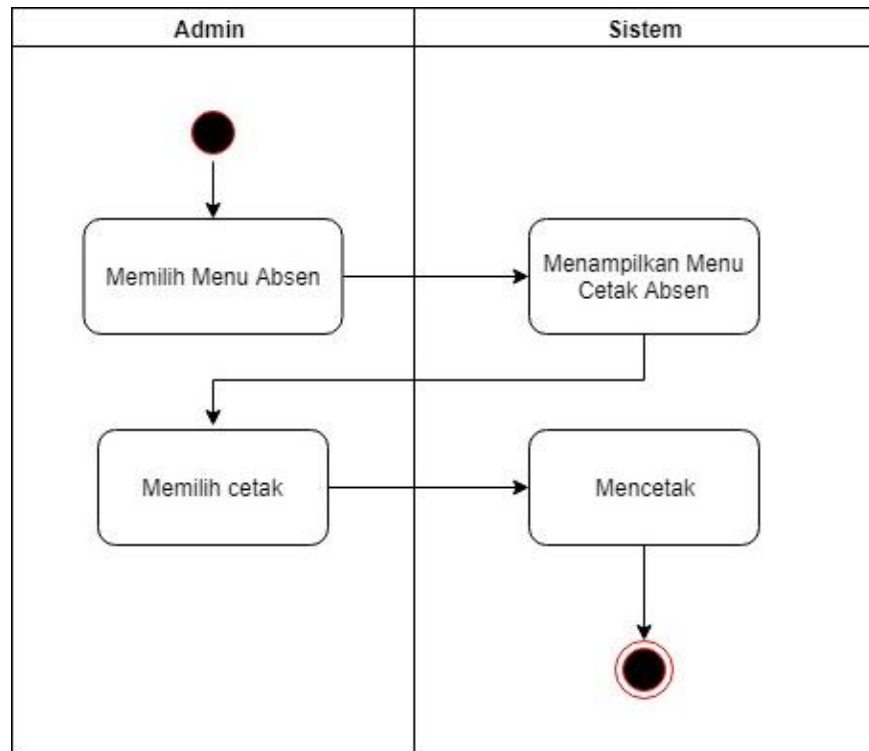
Pada Gambar 4.6 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu absen lalu sistem menampilkan halaman data absensi mahasiswa.



Gambar 4. 6 Activity Diagram Tampil Data Absensi

**d. Activity Diagram Cetak Absen**

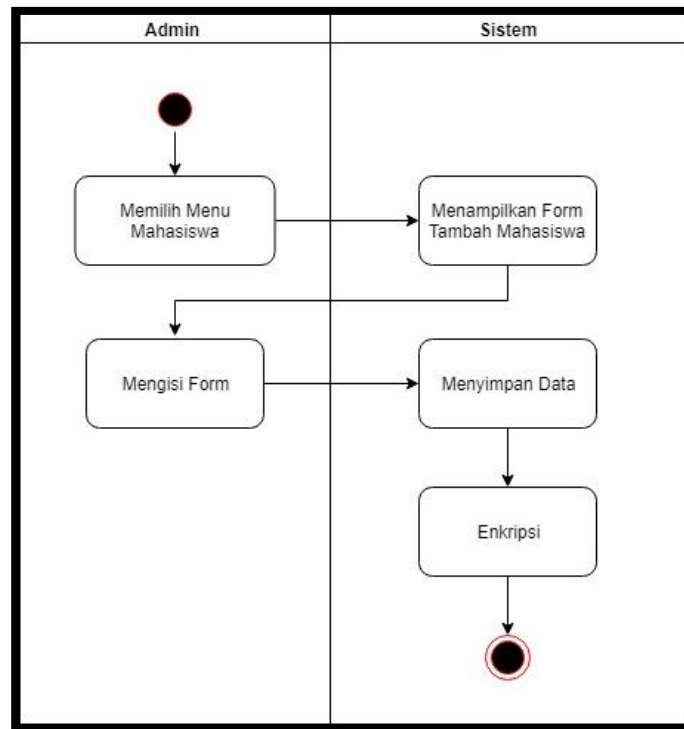
Pada Gambar 4.7 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu absen lalu sistem menampilkan halaman cetak absen.



Gambar 4.7 *Activity Diagram* Cetak Absen

**e. *Activity Diagram* Tambah Data Mahasiswa**

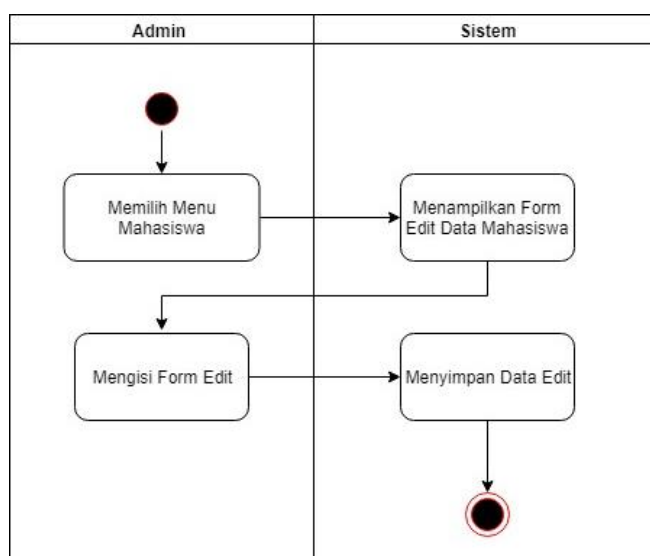
Pada Gambar 4.8 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mahasiswa lalu sistem menampilkan *form* tambah data mahasiswa yang kemudian akan disimpan dan dienkripsi.



Gambar 4.8 *Activity Diagram* Tambah Data Mahasiswa

**f. *Activity Diagram* Edit Data Mahasiswa**

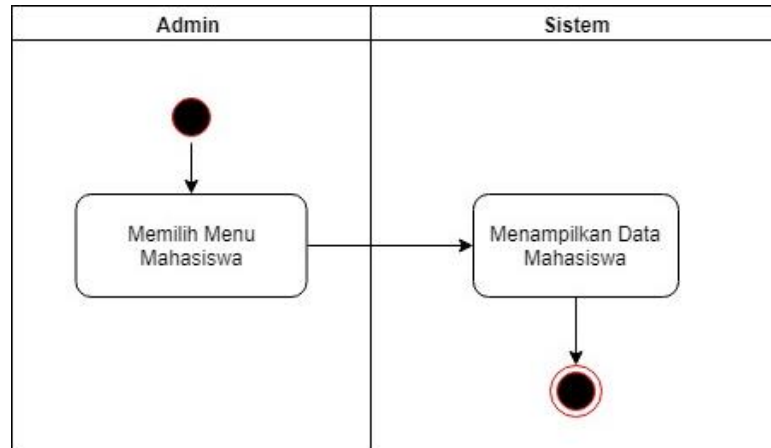
Pada Gambar 4.9 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mahasiswa lalu sistem menampilkan form edit data mahasiswa.



Gambar 4.9 *Activity Diagram* Edit Data Mahasiswa

**g. Activity Diagram Tampil Data Mahasiswa**

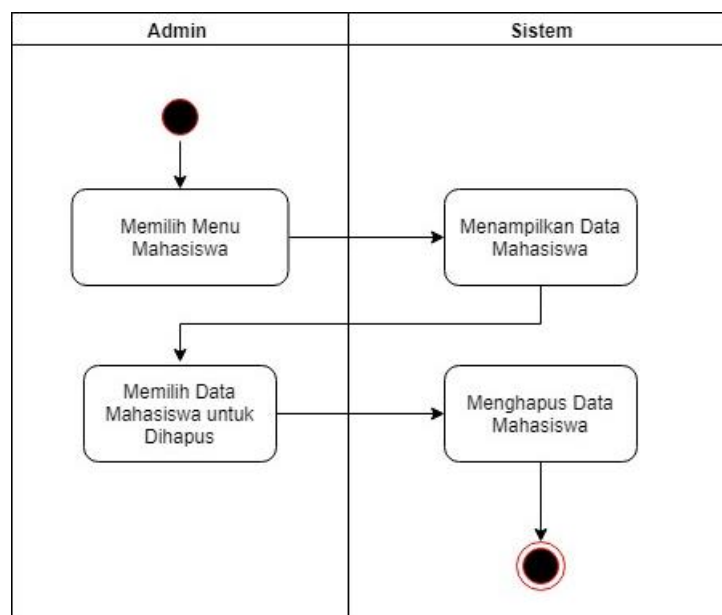
Pada Gambar 4.10 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mahasiswa lalu sistem menampilkan data mahasiswa.



Gambar 4.10 Activity Diagram Tampil Data Mahasiswa

**h. 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.

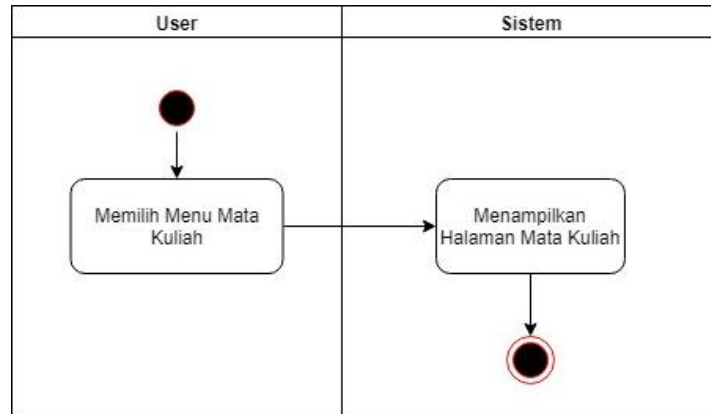


Gambar 4.11 Activity Diagram Hapus Data Mahasiswa



**i. Activity Diagram Mata Kuliah**

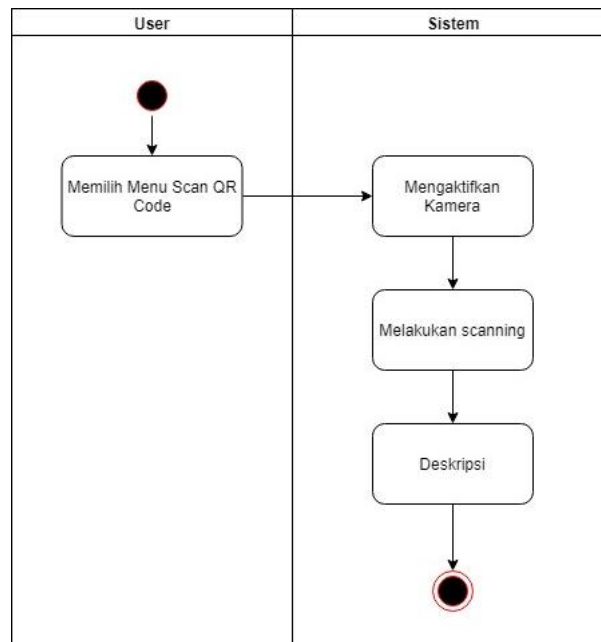
Pada Gambar 4.12 merupakan diagram aktivitas yang menunjukkan aktivitas *user* memilih menu mata kuliah lalu sistem menampilkan halaman mata kuliah.



Gambar 4.12 Activity Diagram Mata Kuliah

**j. Activity Diagram Deskripsi**

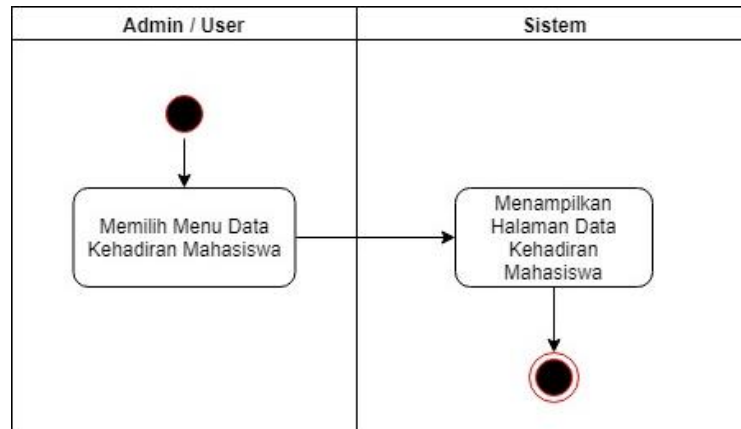
Pada Gambar 4.13 merupakan diagram aktivitas yang menunjukkan aktivitas *user* memilih menu *scan QR Code* lalu sistem mengaktifkan kamera dan melakukan scanning pada *QR Code* lalu melakukan deskripsi terhadap data *QR Code* tersebut.



Gambar 4.13 Activity Diagram Deskripsi

### k. Activity Diagram Data Kehadiran Mahasiswa

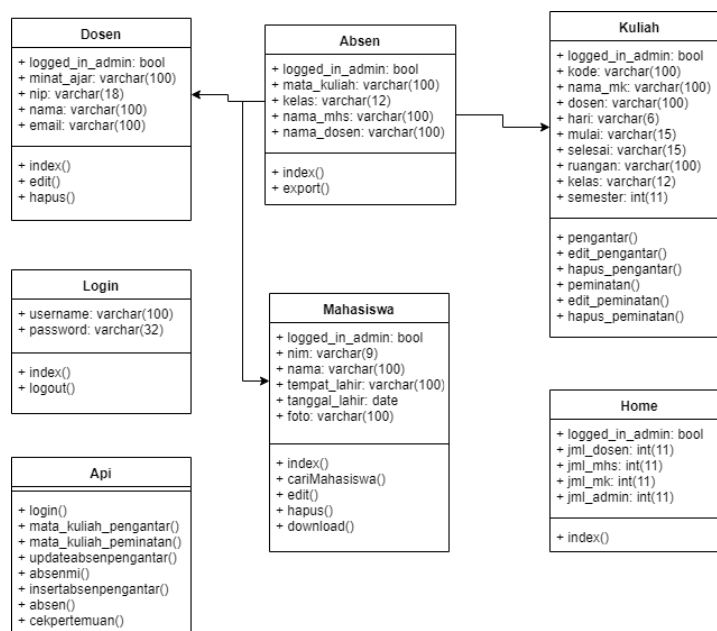
Pada Gambar 4.14 merupakan diagram aktivitas yang menunjukkan aktivitas user memilih menu data kehadiran mahasiswa lalu sistem menampilkan halaman data kehadiran mahasiswa.



Gambar 4.14 Activity Diagram Data Kehadiran Mahasiswa

### 3. 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*.

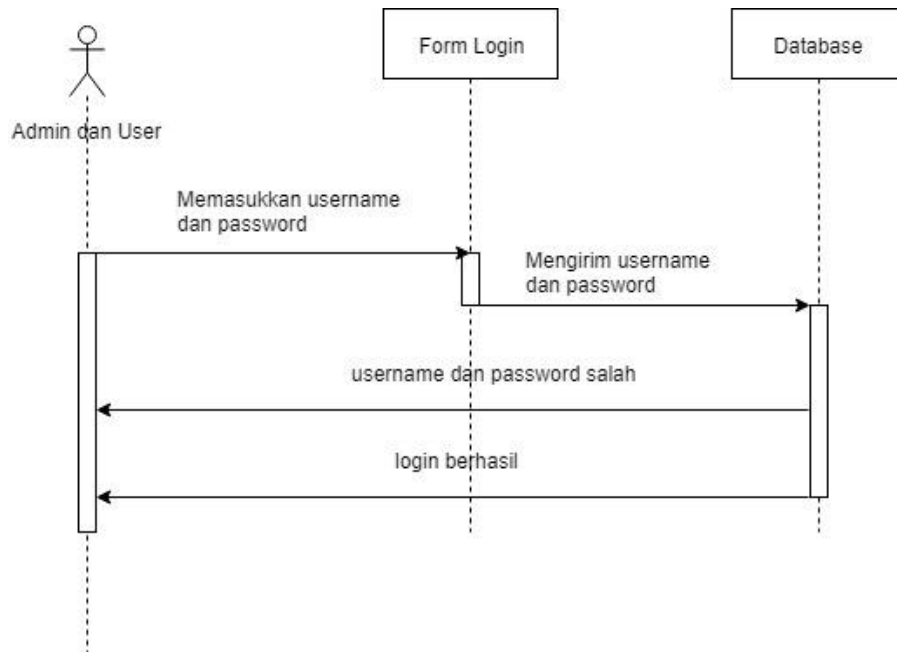


Gambar 4.15 Class Diagram

#### 4. 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 interaksi antar objek dan sistem.

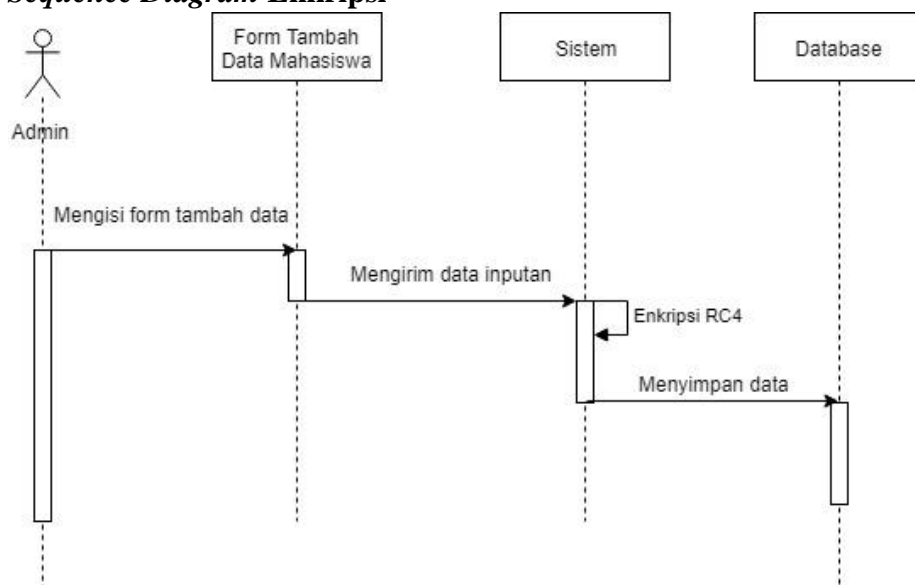
##### a. Sequence Diagram Login



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*.

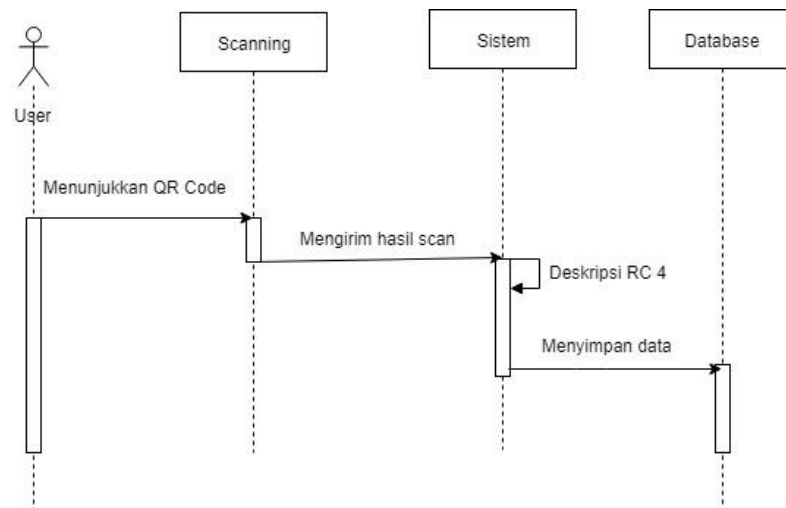
##### b. Sequence Diagram Enkripsi



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*.

#### c. *Sequence Diagram* Deskripsi



Gambar 4.18 *Sequence Diagram* Deskripsi

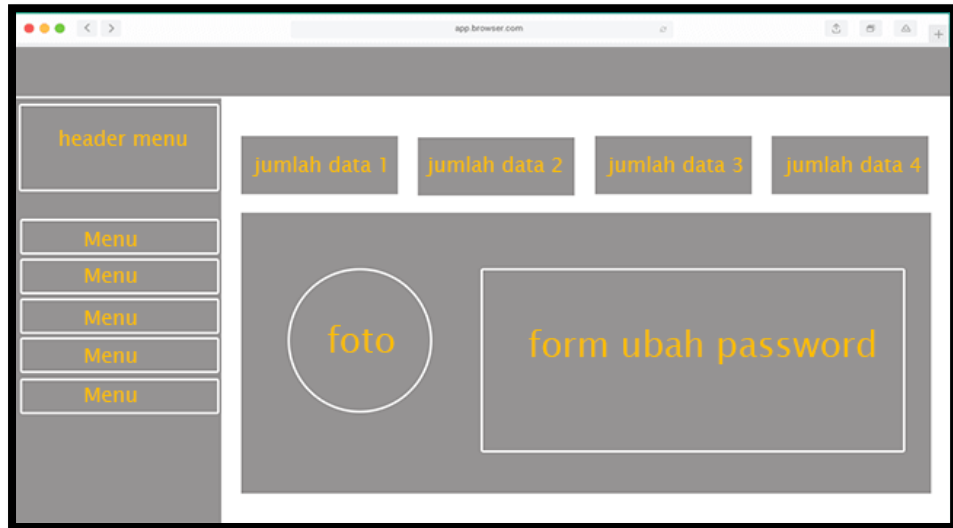
Gambar 4.18 adalah *sequence* yang menunjukkan proses deskripsi dimana *user* menunjukkan *QR Code* untuk kemudian di-*scan* lalu kemudian hasil *scan* akan dikirim ke sistem lalu didekripsi baru kemudian disimpan ke *database*.

#### 4.4. 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*.

#### 4.4.1. Menu Dashboard

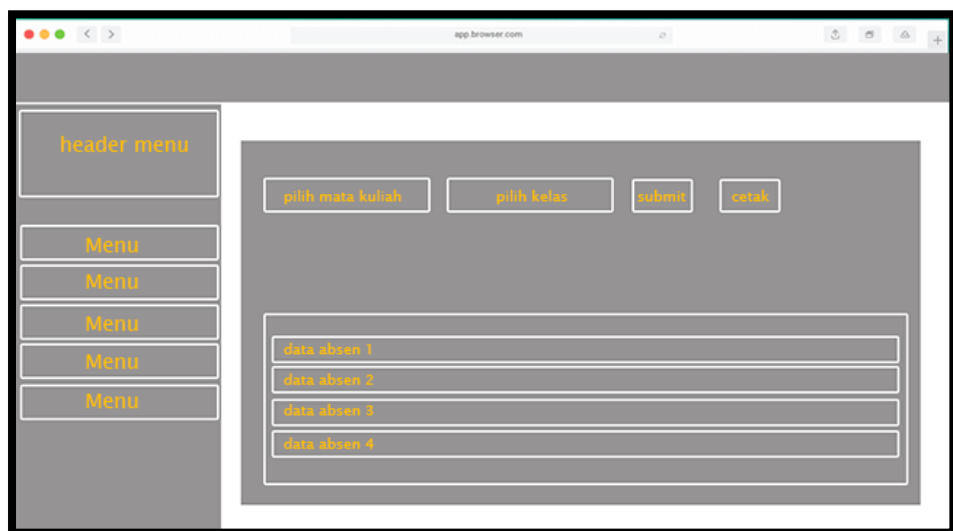
Menu Dashboard menampilkan data aplikasi secara umum. Mulai dari jumlah dosen, mahasiswa, mata kuliah dan *user*. Selain itu pada halaman ini, sistem juga menampilkan data admin serta *form* untuk mengubah *password* admin. Bentuk dari tampilan menu *dashboard* dapat dilihat pada gambar berikut.



Gambar 4.19 Menu Dashboard

#### 4.4.2. Menu Absen

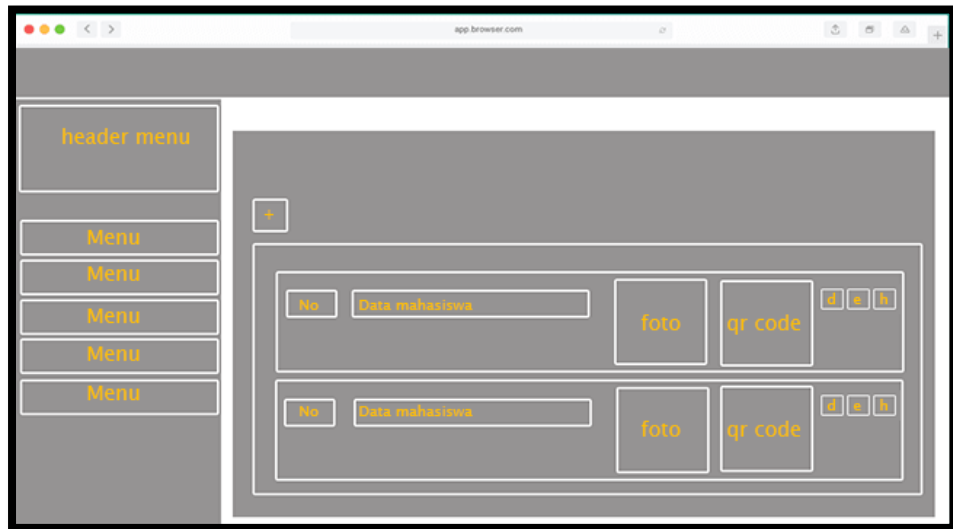
Menu absen menampilkan data absensi berdasarkan nama mata kuliah dan kelas. Terdapat tabel yang berisikan nama dan nim masing-masing mahasiswa serta daftar kehadirannya dari pertemuan 1 sampai pertemuan ke 16. Serta informasi persentase kehadiran mahasiswa tersebut. Selain itu, juga terdapat tombol cetak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



Gambar 4. 20 Menu Absen

#### 4.4.3. Menu Mahasiswa

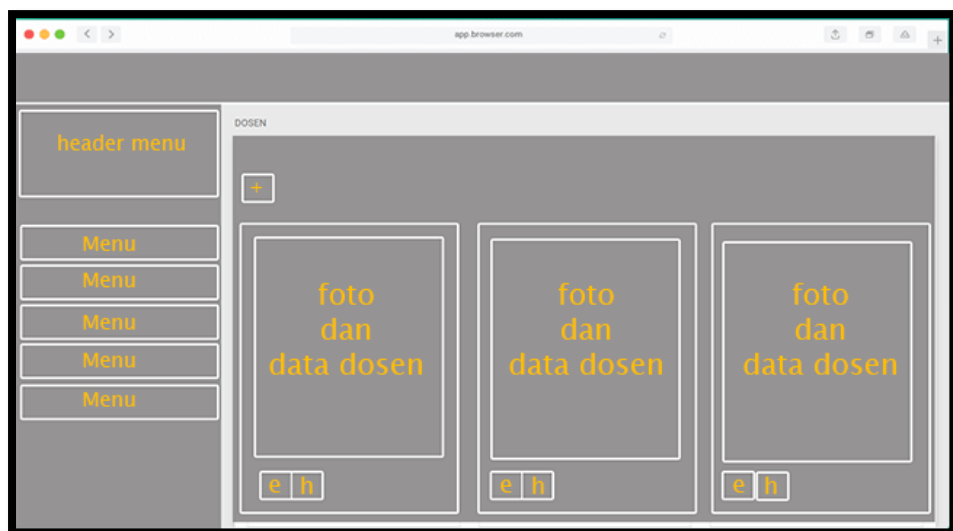
Menu mahasiswa berisikan data pribadi setiap mahasiswa, foto dan gambar *QR Code* yang secara otomatis dibuat oleh sistem berdasarkan data *inputan user*. Bentuk dari tampilan menu mahasiswa dapat dilihat pada gambar berikut.



Gambar 4.21 Menu Mahasiswa

#### 4.4.4. Menu Dosen

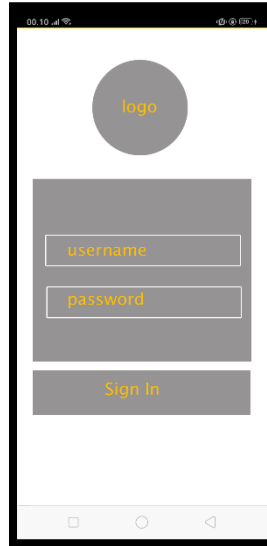
Menu dosen berisi data setiap dosen yang ada pada jurusan Teknik Informatika UHO. Terdapat pula foto dari masing data dosen agar lebih memudahkan dalam mencari data atau informasi tentang dosen. Bentuk dari tampilan menu dosen dapat dilihat pada gambar berikut.



Gambar 4.22 Menu Dosen

#### 4.4.5. Halaman *Login Mobile*

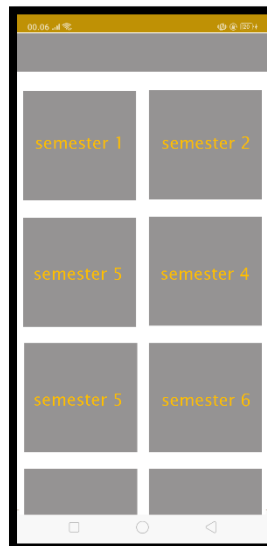
Halaman login pada aplikasi *mobile* menampilkan *form login* yang terdiri dari *username* dan *password*. Setiap *field* harus diisi agar dapat melakukan *login*. Selain itu, *form* ini juga telah ditambahkan fungsi validasi agar dapat menyaring data *user* yang benar. Bentuk dari tampilan *login mobile*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



Gambar 4.23 Halaman *Login Mobile*

#### 4.4.6. 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*

#### 4.4.7. 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*

#### 4.4.8. 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 per mata kuliah dapat dilihat pada gambar berikut.

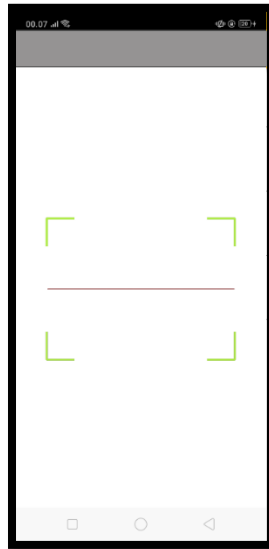


Gambar 4.26 Halaman Daftar Mahasiswa Per Mata Kuliah



#### 4.4.9. 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.



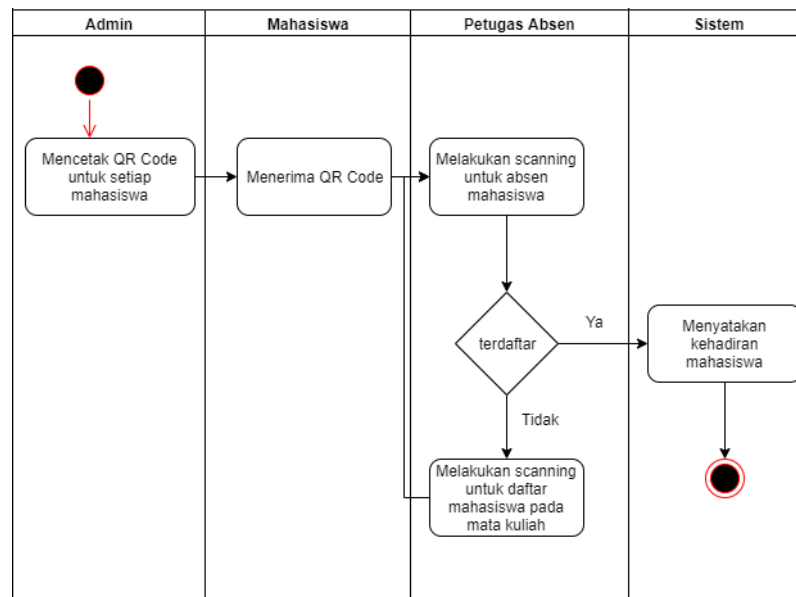
Gambar 4. 27 Halaman *Scanning QR Code*

## 4.5. Prosedur Penggunaan Sistem

### 4.5.1. 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.

### 4.5.2. Langkah-langkah penggunaan sistem.



Gambar 4.28 Prosedur Penggunaan Sistem

## BAB V

### IMPLEMENTASI DAN PENGUJIAN SISTEM

#### 5.1. Implementasi Sistem

Implementasi merupakan tahap di mana sistem telah siap untuk digunakan. Hasil analisis dan perancangan sistem yang telah dilakukan sebelumnya diimplementasikan dalam bentuk aplikasi berupa aplikasi *web services* dan aplikasi berbasis android serta *QR Code* yang data di dalamnya telah dienkripsi menggunakan algoritma *RC 4 (Rivest Cipher 4)*. Kedua aplikasi ini dibangun dengan bahasa pemrograman PHP pada server side dan Dart pada aplikasi androidnya.

##### 5.1.1. Kebutuhan Sistem

Untuk menjalankan sistem diperlukan perangkat lunak sebagai berikut:

1. *MySQL* sebagai *database management system*.
2. *Xampp* sebagai *local web server*
3. *Web Browser* untuk menampilkan aplikasi *web services*.
4. *VS Code* untuk membuild aplikasi *mobile*.

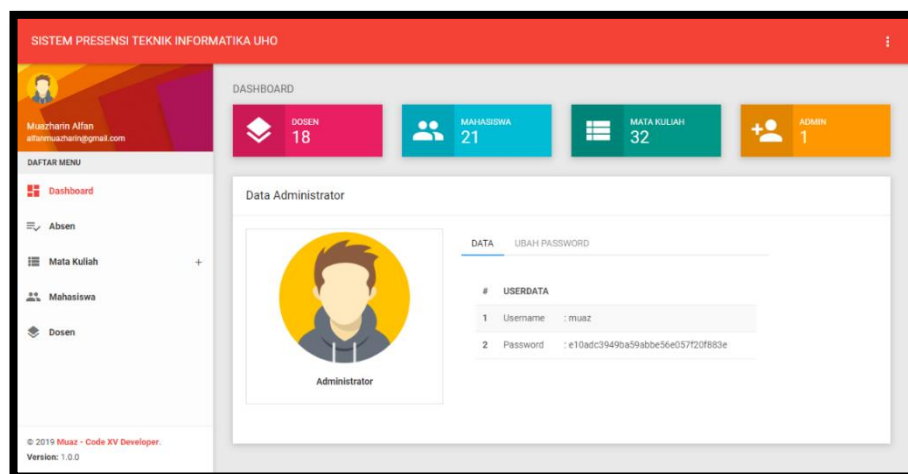
Aplikasi ini dapat beroperasi pada *PC* dengan spesifikasi sebagai berikut:

1. Sistem Operasi Windows 10 Enterprise 64-bit.
2. *Processor AMD A6 Quad Core*.
3. *SSD 240 GB*.
4. *RAM 4 GB*.

##### 5.1.2. Implementasi Tampilan Antarmuka Sistem

Berikut adalah tampilan antarmuka sistem di masing-masing menu:

###### 1. Menu *Dashboard*

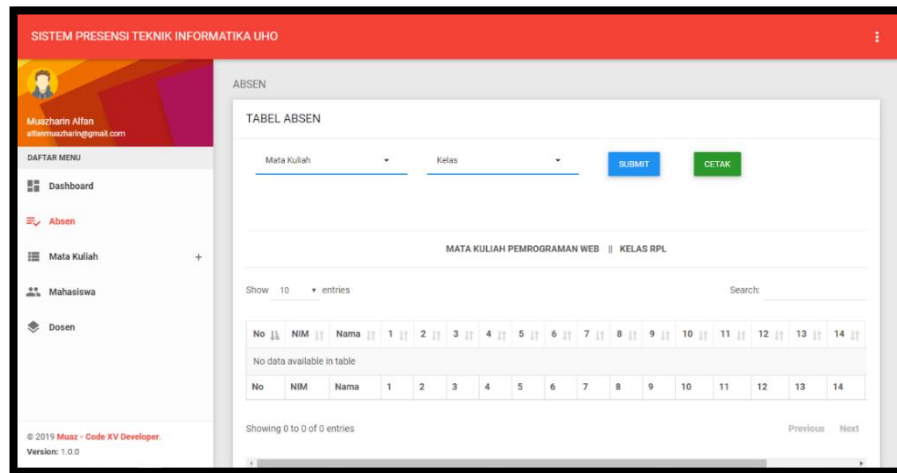


Gambar 5.1 Menu *Dashboard*

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*.

## 2. Menu Absen

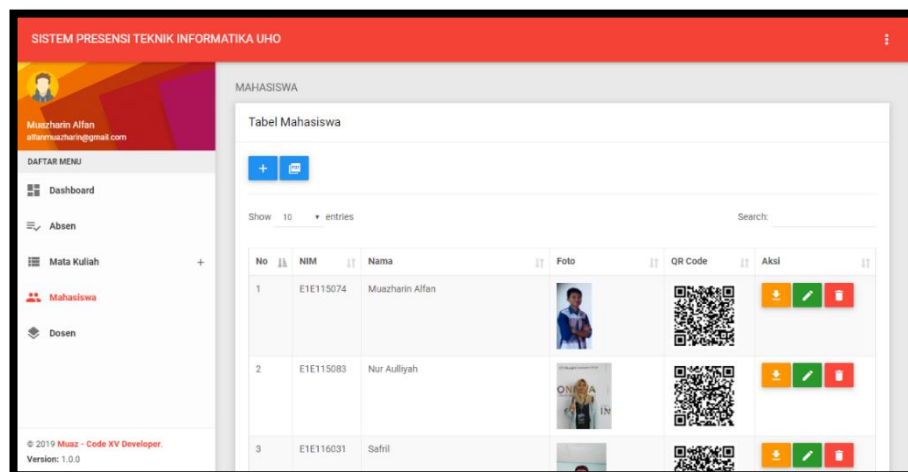
Menu absen menampilkan data absensi berdasarkan nama mata kuliah dan kelas. Terdapat tabel yang berisikan nama dan nim masing-masing mahasiswa serta daftar kehadirannya dari pertemuan 1 sampai pertemuan ke 16. Serta informasi persentase kehadiran mahasiswa tersebut. Selain itu, juga terdapat tombol cetak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



Gambar 5.2 Menu Absen

## 3. Menu Mahasiswa

Menu mahasiswa berisikan data pribadi setiap mahasiswa, foto dan gambar QR Code yang secara otomatis dibuat oleh sistem berdasarkan data inputan user.



Gambar 5.3 Menu Mahasiswa

#### 4. Menu Mata Kuliah

Menu mata kuliah berisi daftar mata kuliah yang terdapat pada jurusan Teknik Informatika UHO mulai dari semester 1 sampai semester 8.

No	Kode	Nama Mata Kuliah	Dosen1	Dosen2	Hari	Mulai	Selesai	Ruang	Kelas	Semester
1	TIF61019	PRAKTIKUM APLIKASI KOMPUTER	JUMADIL NANGIL S.Kom, MT		Senin	10:01	10:50	LAB. SI & PROGRAMMING	genap	1
2	TIF61014	KALKULUS I	NATALIS RANSI,		Senin	13:01	15:30	IT-3	genap	1
3	TIF61015	MATEMATIKA DISKRIT	I A SURIMI, S.Si, M.Cs		Selasa	9:30	12:00	IT-2	genap	1

Gambar 5.4 Menu Mata Kuliah

#### 5. Menu Dosen

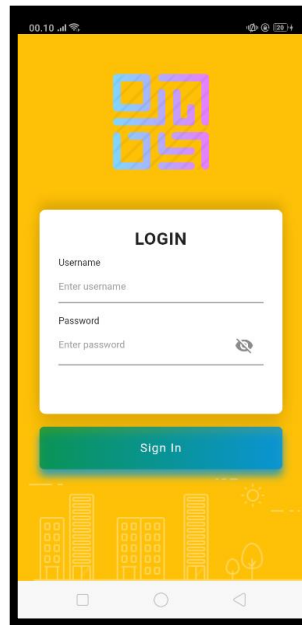
Menu dosen berisi data setiap dosen yang ada pada jurusan Teknik Informatika UHO. Terdapat pula foto dari masing data dosen agar lebih memudahkan dalam mencari data atau informasi tentang dosen.

No	Foto	Nama Dosen	Email	Alamat	Telepon
1		Adha Mashur Sajjah, S.T., M.Eng.	- adha.m.sajjah@gmail.com	- Komputasi Cerdas dan Visualisasi	- Tenaga Pengajar
2		Bambang Pranono, S.Si, M.T.	- bambangpranono9@gmail.com	- Rekayasa Perangkat Lunak	- Kepala Laboratorium...
3		Dr. Ir. H. Muhammad Ihsan Sarita,...	- ihsan@gmail.com	- Komputasi Cerdas dan Visualisasi	- Lektor

Gambar 5.5 Menu Dosen

## 6. *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.



Gambar 5.6 *Login Mobile*

## 7. *Home Mobile*

Halaman *home mobile* menampilkan daftar semester 1 sampai 8.



Gambar 5.7 *Home Mobile*

### 8. Mata Kuliah *Mobile*

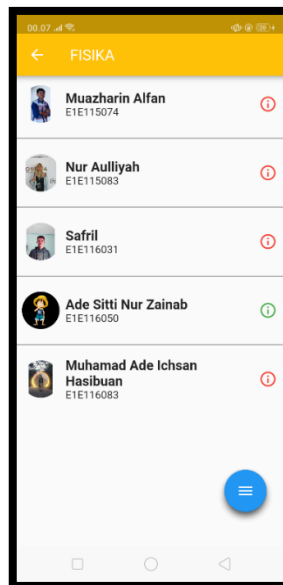
Halaman mata kuliah mobile menampilkan daftar mata kuliah pada semester dan kelas tertentu. Keterangan mata kuliah juga dicantumkan seperti jadwal dan nama ruangan perkuliahan.



Gambar 5.8 Mata Kuliah Mobile

### 9. Mahasiswa *Mobile*

Berdasarkan gambar di atas, terlihat daftar mahasiswa yang mengambil mata kuliah FISIKA. Tercantum nama dan NIM mahasiswa tersebut dengan simbol informasi pada sisi kanan berwarna merah/hijau. Simbol ini menandakan tingkat persentase kehadiran mahasiswa. Jika berwarna merah, maka persentase kehadiran mahasiswa tersebut tidak mencapai 80% dan sebaliknya jika berwarna hijau maka persentase kehadiran mahasiswa tersebut mencapai 80% atau lebih.



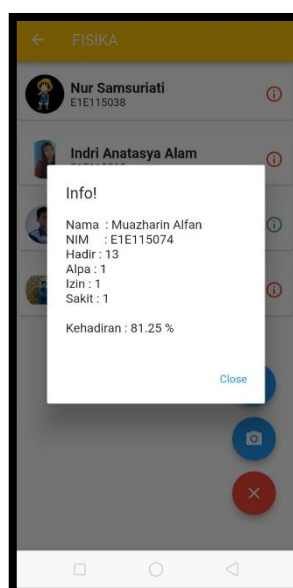
Gambar 5.9 Mahasiswa *Mobile*

Persentase kehadiran tersebut didapatkan berdasarkan rumus persentase dengan mengambil parameter (jumlah kehadiran mahasiswa / total kehadiran untuk 1 mata kuliah) x 100%. Implementasi rumus dapat dilihat pada gambar berikut.

```
final res = listmhs[i];
double p = double.parse(res.persentase);
double sen = (p / 16) * 100;
```

Gambar 5. 10 Implementasi rumus persentase mahasiswa

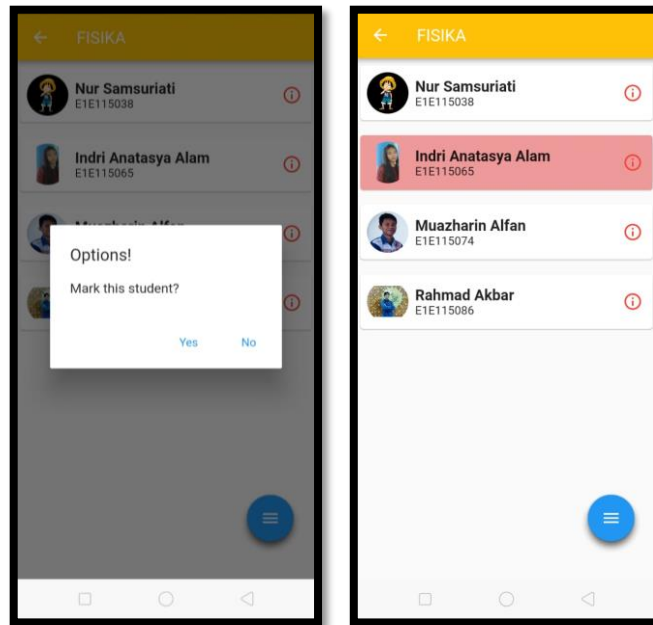
Sementara bentuk dari tampilan persentase mahasiswa tersebut dapat dilihat pada Gambar 5.11 di bawah ini.



Gambar 5.11 Tampilan Persentase Kehadiran Mahasiswa

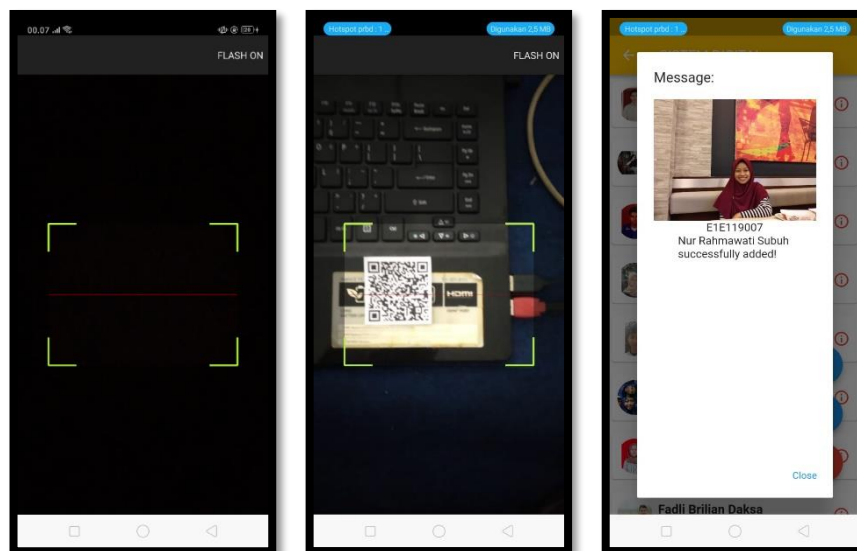


Selain menampilkan data persentase, pada halaman ini juga dilengkapi dengan fitur *marker* dimana fitur ini berfungsi untuk menandai mahasiswa jika terdapat kesalahan pada proses absensi. Kemudian petugas absen dapat melaporkan kesalahan tersebut pada admin.



Gambar 5.12 *Marker* mahasiswa

## 10. Scanner



Gambar 5.13 Tampilan *Scanner*

Halaman *scanner* menampilkan alat *scanner* berupa kamera yang siap melakukan *scanning* terhadap QR Code yang ditemukan.

## 5.2. Pengujian Sistem

Pengujian merupakan tahap yang utama dalam pembuatan suatu aplikasi. Hasil dari pengujian yang didapat akan dijadikan sebagai tolak ukur dalam proses pengembangan selanjutnya.

### 5.2.1. Pengujian Black Box

Pengujian *black box* dilakukan dengan menguji perangkat lunak dari segi fungsionalitas perangkat lunak. Fungsionalitas perangkat lunak yang diuji sesuai dengan skenario pada tahap desain.

**Tabel 5.1 Pengujian Black Box**

Input/Event	Output	Hasil Uji
Memilih menu dashboard	Menampilkan menu dashboard	Berhasil
Memilih menu absen	Menampilkan menu absen	Berhasil
Memilih menu mata kuliah	Menampilkan menu mata kuliah	Berhasil
Memilih menu dosen	Menampilkan menu dosen	Berhasil
Masuk ke halaman <i>home mobile</i>	Menampilkan halaman <i>home mobile</i>	Berhasil
Masuk ke halaman mata kuliah <i>mobile</i>	Menampilkan halaman mata kuliah <i>mobile</i>	Berhasil
Masuk ke halaman mahasiswa <i>mobile</i>	Menampilkan halaman mahasiswa <i>mobile</i>	Berhasil
Mengaktifkan scanner	Mengaktifkan kamera dan menampilkan alat scanner	Berhasil

### 5.2.2 Pengujian Metode *Rivest Cipher 4 (RC4)*

Pengujian ini bertujuan untuk menampilkan data *string* yang akan dienkripsi menggunakan metode *RC4*. Lalu ditampilkan hasil enkripsi dalam bentuk karakter dan simbol ASCII.

**Tabel 5.2 Tabel Pengujian Metode *Rivest Cipher 4 (RC4)***

No	Sebelum Enkripsi	Kunci	Sesudah Enkripsi
1.	E1E11996-11-0715074Raha	15074	F6? ? ? ? ? ? 5 ? ? ? ? ? m ? ?

2.	E1E11997-10-2815001Kendari	15001	}~Pn g_c
3.	E1E11998-05-1815022Dili	15022	m;*cR%,0□
4.	E1E11997-11-0715026Kendari	15026	*J>~v>J^4a ?
5.	E1E11997-09-1815030Boloimbu	15030	qdeDWI"u cشTe
6.	E1E11997-02-2815035Kendari	15035	Zu(DU-! ?
7.	E1E11997-04-0516010Tampo	16010	nnU RB v_c
8.	E1E11998-10-2816024Pondidaha	16024	3lXJ8z-θ
9.	E1E11997-05-0516026Lambubalano	16026	b0]jgVTGPqY□[j
10.	E1E12000-06-1917027kendari	17027	vXF/S{w6

### 5.2.3 Pengujian Aplikasi Pada Kelas

Pengujian ini bertujuan untuk menampilkan penggunaan waktu setiap mahasiswa pada saat melakukan *scanning*. Adapun hasilnya dapat dilihat pada tabel berikut.

**Tabel 5.3 Pengujian Aplikasi Pada Beberapa Kelas Sekaligus**

No.	Jumlah Kelas	Jumlah Mahasiswa		Waktu		Rata-rata Kecepatan Scanning	
1.	1 K	20		2 m 5 d		6.25 d	
2.	2 K	K1	K2	K1	K2	K1	K2
		20	23	2 m 13 d	3 m 5 d	6.65 d	8.04 d
Rata-rata Kecepatan Scanning Setiap Mahasiswa						(6.25 d+ 6.65 d+ 8.04 d) / 3 = (20.94 d) / 3 = 6.93 d	

Keterangan:

K = kelas

m = menit

d = detik

Adapun data di atas diperoleh berdasarkan hasil *scanning* pada beberapa kelas dengan rekam data yang tersimpan pada *database* seperti berikut.

nim	nama_mhs	nama_mata_kuliah	kelas	per_catu	tgl_catu	per_dus	tgl_dus
E1E119048	Andri Razaidin	PRAKTIKUM APLIKASI KOMPUTER	ganji		0000-00-00 00:00:00	h	2019-10-08 08:10:35
E1E119049	Annisa Dwi Santoso	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:04:53	h	2019-10-08 08:10:28
E1E119056	Fabriona	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:05:07	h	2019-10-08 08:10:23
E1E119041	Agatha Mega Putri	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:05:14	h	2019-10-08 08:10:17
E1E119034	Nadisa Quratul Ain	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:05:21	h	2019-10-08 08:10:11
E1E119077	RIZKY NHAIRUN NISA	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:05:28	h	2019-10-08 08:09:57
E1E119072	Nurul Nisa	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:05:34	h	2019-10-08 08:09:40
E1E119035	Nurhalisa Masdukiyahan	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:05:40	h	2019-10-08 08:09:32
E1E119058	IFANI FADILLAH PUTRI Shafiq	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:05:46	h	2019-10-08 08:09:28
E1E119079	Khoerunnisa Ardi	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:05:55	h	2019-10-08 08:09:22
E1E119038	Susana Aprilia Maharani	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:06:02	h	2019-10-08 08:09:17
E1E119071	Niken Indriyani	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:06:08	h	2019-10-08 08:09:11
E1E119080	Turbidil Umas	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:06:12	h	2019-10-08 08:08:58
E1E119003	Dini Fadilah	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:06:19	h	2019-10-08 08:08:50
E1E119017	And Gamma Brilliant Arinda	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:06:25	h	2019-10-08 08:08:41
E1E119069	Muhammad Yasin Akbar T	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:06:32	h	2019-10-08 08:08:36
E1E119043	Ahmad Dhan Al	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:06:38	h	2019-10-08 08:08:30
E1E119068	Muhammad Syaban Adhiam Rabbil	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:06:46	h	2019-10-08 08:08:22
E1E119037	Sukarnas Mard Imnan	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:06:50	h	2019-10-08 08:10:25
E1E119064	Muhammad Yasin	PRAKTIKUM APLIKASI KOMPUTER	ganji	h	2019-10-08 08:06:58	h	2019-10-08 08:11:32

Gambar 5.15 Catatan waktu *scanning* pada *database*

## **BAB VI**

### **PENUTUP**

#### **6.1 Kesimpulan**

Berdasarkan hasil penelitian yang telah dilakukan mengenai enkripsi dan dekripsi data QR Code menggunakan Algoritma *Rivest Cipher 4 (RC4)*, maka diperoleh kesimpulan yaitu:

1. Enkripsi data *QR Code* menggunakan algoritma *RC4* berjalan dengan baik dan dapat didekripsi kembali dalam sistem sehingga dapat mengenali pemilik data *QR Code* dengan mudah dan cepat.
2. Kecepatan proses absen setiap mahasiswa berjalan cukup cepat yaitu dengan kisaran waktu 6 sampai 8 detik saja.
3. Kelebihan algoritma *RC4* yaitu memiliki kecepatan yang lebih baik karena berbasis *stream cipher* yang melakukan enkripsi *one byte at a time*, serta karena *RC4* adalah algoritma dengan kunci simetris.
4. Kunci setiap data mahasiswa dibuat otomatis oleh sistem dengan panjang karakter yang sama namun tetap unik.

#### **6.2 Saran**

Beberapa saran yang perlu diperhatikan untuk pengembangan penelitian selanjutnya yaitu sebagai berikut.

1. Enkripsi dan dekripsi data *QR Code* dapat dilakukan dengan menggunakan algoritma lain yang dianggap lebih aman dan sesuai dengan perkembangan teknologi.
2. Pada penelitian selanjutnya sistem ini dapat dibuat lebih kompleks lagi seperti adanya sistem tersedia untuk mahasiswa sehingga masing-masing dari mereka dapat melihat atau mengolah data.