

PROPOSAL

IMPLEMENTASI ALGORITMA RC4 PADA PENGGUNAAN QR-CODE APLIKASI PERMOHONAN PENGAJUAN HAK ATAS TANAH MILIK INSTANSI PEMERINTAH DI KABUPATEN KONAWE SELATAN

Diajukan Untuk Memenuhi

Salah Satu Syarat Memperoleh Gelar Sarjana Teknik



ANNISA FITRI HAMID

E1E1 17 005

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HALU OLEO

KENDARI

2020

LEMBAR PENGESAHAN

Seminar Proposal

IMPLEMENTASI ALGORITMA RC4 PADA PENERAPAN QR CODE APLIKASI PERMOHONAN PENGAJUAN HAK ATAS TANAH MILIK INSTANSI PEMERINTAH DI KABUPATEN KONAWE SELATAN

dibuat dan diserahkan sebelumnya baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Universitas Halu Oleo ataupun institusi pendidikan lainnya.

Kendari, Desember 2020



ANNISA FITRI HAMID

NIM. E1E1 17 005

Pembimbing I



BAMBANG PRAMONO, S.Si., MT

NIP. 19710425 200801 1 010

Pembimbing II



JUMADIL NANGIS, S.K., MT

NIP. 19870206 201504 1 003

DAFTAR ISI

HALAMAN JUDUL	1
HALAMAN PENGESAHAN	ii
DAFTAR ISI.....	iii
DAFTAR TABEL	v
DAFTAR GAMBAR.....	vi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan	4
1.7 Tinjauan Pustaka	5
BAB II LANDASAN TEORI	7
2.1 Kriptografi	7
2.1.1 Pembagian Algoritma Kriptografi	8
2.2 Algoritma RC4.....	9
2.2.1 Inisialisasi <i>State Array</i>	11
2.2.2 Penghasilan Kunci Enkripsi dan Pengenkripsian.....	12
2.2.3 Contoh Penerapan Algoritma RC4	12
2.3 <i>QR-Code</i>	16
2.3.1 Keuntungan dan Kerugian <i>QR-Code</i>	16
2.3.2 Anatomi <i>QR-Code</i>	17
2.3.3 Versi <i>QR-Code</i>	18
2.4 <i>Hyper Text Markup Language (HTML)</i>	18
2.5 PHP	18
2.6 MySQL	19
2.7 <i>Database (Basis Data)</i>	20
2.8 <i>Code Igniter</i>	21
2.9 <i>Flowchart</i>	21
2.10 <i>Unified Modeling Language (UML)</i>	22
2.11 Metode Pengembangan Sistem.....	26
BAB III METODOLOGI PENELITIAN	29

3.1	Waktu dan Tempat Penelitian	29
3.1.1	Waktu	29
3.1.2	Tempat Penelitian	29
3.2	Metode Pengumpulan Data	29
3.3	Metode Pengembangan Sistem.....	30
3.3.1	Permulaan (<i>Inception</i>)	30
3.3.2	Perluasaan/Perencanaan (<i>Elaboration</i>).....	30
3.3.3	Konstruksi (<i>Construction</i>)	30
3.3.4	Transisi (<i>Transition</i>).....	30
3.4	Analisis Perancangan Sistem.....	30
3.4.1	<i>Flowchart</i>.....	31
3.4.2	<i>Unified Modeling Language (UML)</i>.....	33
3.5	Perancangan Antarmuka (<i>Interface</i>).....	39
3.5.1	<i>Form Login</i>.....	39
3.5.2	Laman Menu Utama	40
3.5.3	Laman Menu Pengajuan Baru.....	40
3.5.4	Laman Riwayat Pengajuan	41
3.6	Metode Pengujian.....	42
3.6.1	Rencana Pengujian	43
DAFTAR PUSTAKA		44

DAFTAR TABEL

	Hal
Tabel 2.1 Simbol-simbol flowchart.....	21
Tabel 2.2 Simbol-simbol Use Case Diagram	23
Tabel 2.3 Simbol Activity Diagram	24
Tabel 2.4 Simbol Sequence Diagram (Lanjutan)	25
Tabel 2.5 Simbol Class Diagram	26
Tabel 3.1 Gantt Chart Waktu Penelitian	29
Tabel 3.2 Keterangan Use Case Diagram	34
Tabel 3.3 Rencana Pengujian	43

DAFTAR GAMBAR

	Hal
Gambar 2.1 Proses Enkripsi Deskripsi Menggunakan Kriptografi Simetri.....	8
Gambar 2.2 Proses Enkripsi Deskripsi Menggunakan Kriptografi Asimetri..	9
Gambar 2.1. Proses Pseudo Random Pada Algoritma RC4.....	11
Gambar 2.4 Anatomi <i>QR code</i>	17
Gambar 2.5 Versi <i>QR code</i>	18
Gambar 3.1 Flowchart Sistem Aplikasi Pengajuan.....	31
Gambar 3.2 Flowchart Algoritma RC4 (Rivest Cipher 4)	33
Gambar 3.3 Use Case Diagram Sistem	34
Gambar 3.4 Activity Diagram Login	35
Gambar 3.5 Activity Diagram Profile	36
Gambar 3.6 Activity Diagram Pengajuan Baru	37
Gambar 3.7 Activity Diagram Riwayat Pengajuan	37
Gambar 3.8 Class Diagram Aplikasi pengajuan	38
Gambar 3.9 Sequence Diagram Login	38
Gambar 3.10 Sequence Diagram Enkripsi Data	39
Gambar 3.11 Form Login.....	40
Gambar 3.12 Laman Menu Utama.....	40
Gambar 3.13 Laman Menu Pengajuan.....	41
Gambar 3.14 Laman Menu Pengajuan.....	41
Gambar 3.15 Laman Riwayat Pengajuan.....	42
Gambar 3.16 Laman Riwayat Pengajuan.....	42

BAB I

PENDAHULUAN

1.1 Latar Belakang

Badan Pertanahan Nasional (BPN) pada awalnya adalah lembaga pemerintah non-kementerian di Indonesia yang mempunyai kewajiban melaksanakan tugas pemerintahan di bidang Pertanahan. Namun pada tahun 2015 Badan Pertanahan Nasional berubah menjadi Kementerian Agraria dan Tata Ruang / Badan Pertanahan Nasional sesuai ketetapan Presiden Nomor 17 Tahun 2015 tentang Kementerian Agraria yang berfungsi Tata Ruang dan Peraturan Presiden Nomor 20 Tahun 2015 tentang Badan Pertanahan Nasional yang ditetapkan pada 21 Januari 2015 (*Kantor Pertanahan Kab Bogor, n.d.*).

Pandemi Covid-19 sangat mempengaruhi kinerja Badan Pertanahan Nasional (BPN) dalam melaksanakan fungsinya sebagai perumusan dan pelaksanaan kebijakan dibidang penetapan hak tanah, pendaftaran tanah, pemberdayaan masyarakat, dan layanan pertanahan lainnya. Salah satu yang sangat berdampak adalah pelayanan pertanahan yang berhubungan loket layanan pada kantor-kantor pertanahan. Badan Pertanahan Nasional (BPN) berupaya untuk mendukung pemerintah dalam pencegahan penyebaran virus Covid-19, dengan menerapkan protokol kesehatan dan upaya upaya lainnya yang disinyalir menjadi bagian dari proses penyebaran virus Covid-19 mengingat bahwa area perkantoran merupakan *cluster* baru dalam penyebaran virus Covid-19. Untuk memaksimalkan upaya dalam pencegahan penularan virus Covid-19 diperlukan penyederhanaan interaksi pelayanan yang saat ini berbasis loket layanan, menjadi interaksi berbasis *website*.

Salah satu layanan dari Badan Pertanahan Nasional (BPN) adalah melakukan pensertifikatan tanah aset pemerintah yang dalam prosesnya memiliki jalur yang berhubungan dengan loket layanan. Dalam pemrosesan permohonan pensertifikatan, diawali dengan pengisian blangko permohonan yang disertai dengan kelengkapan berkas-berkas pendukung oleh pemohon dalam hal ini instansi pemerintah. Setelah melakukan pengisian blanko permohonan, maka berkas akan diserahkan ke loket pelayanan Kantor Pertanahan untuk diperiksa, apabila terdapat

data pengisian yang tidak sesuai, maka pemohon akan dikembalikan untuk ditinjau ulang.

Untuk menyederhanakan interaksi pemohon pada layanan Pengajuan Permohonan Hak Atas Tanah Oleh Instansi Pemerintah, diperlukan adanya komunikasi maya dalam bentuk *website* sehingga dapat di akses secara fleksible. Pembuatan *website* ini nantinya akan menjadi loket firtual dalam mengajukan Permohonan Hak Atas Tanah Oleh Instansi Pemerintah, yang pada dasarnya memiliki proses yang sama dengan pengajuan langsung ke loket kantor pertanahan. Meski demikian, pada akhir pengajuan yang dilakukan nantinya berhujung pada penyerahan dokumen asli kekantor secara langsung oleh pemohon agar pengajuan yang telah diterima dapat diproses ke tahap selanjutnya. Namun untuk membedakan dokumen pemohon yang telah melakukan permohonan melalui *website*, diperlukan *code* khusus pada dokumen yang menandakan ke-validan dokumen yang di ajukan.

Quick Rensponse Code (QR-Code) adalah bar-code dua dimensi yang dapat menyimpan data. Keunggulan *QR-Code* ialah dapat menyimpan informasi secara vertical maupun horizontal sehingga memungkinkan untuk menampung data yang lebih banyak dibandingkan dengan barcode satu dimensi (Dedy irawan & Adriantantri, 2019). Penggunaan *QR-Code* telah banyak diterapkan dalam berbagai bentuk aplikasi *QR code* Reader dan *QR code* Generator, sehingga memudahkan seseorang dalam membuat informasi dalam bentuk *QR code* dan memperoleh informasi yang diinginkan, hanya dengan melakukan proses pembacaan dan pemindaian data menggunakan media dari kamera handphone (Ani et al., 2011).

Penggunaan *QR-Code* dapat menjadi *alternative* dalam penggunaan code khusus yang memuat informasi ke-validan dokumen pengajuan yang dibawa oleh pemohon ke kantor pertanahan. 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, 2010). Untuk mengamankan data pada QR-Code dapat digunakan algoritma yang dapat menyembunyikan data, salah satunya adalah *Rivers Code 4 (RC4)*. Algoritma ini bekerja dengan kunci enkripsi yang didapat dari 256 bit *sate array* yang diinisialisasi dengan sebuah *key* berupa *strings* dengan panjang 1-256 bit, setelah itu *state array* yang didapatkan diacak kembali dan diproses untuk menghasilkan sebuah kunci

enkripsi yang akan di-*XOR* (*exclusive or*) dengan *plain text* ataupun *cipher text* sehingga didapatkan hasil dari enkripsi ataupun dekripsi (Muharom & Sholeh, 2016).

Dari pemaparan di atas, peneliti akan menerapkan QR-Code ter-enkripsi sebagai bukti validasi pada dokumen pengajuan, agar dapat meningkatkan keamanan dan keaslian dokumen yang akan diserahkan ke kantor pertanahan. Karena hal tersebut, maka peneliti mengangkat judul **“Penerapan Algoritma RC4 pada penerapan QR-Code pada Aplikasi Permohonan Pengajuan Hak Atas Tanah Milik Pemerintah di Kabupaten Konawe Selatan”**. Aplikasi ini akan diterapkan pada Kantor Wilayah BPN Kabupaten Konawe Selatan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka rumusan masalah dalam penelitian ini adalah Bagaimana proses pengamanan *QR code* pada Aplikasi Permohonan Pengajuan Hak Atas Tanah milik instansi pemerintah menggunakan Algoritma RC4.

1.3 Batasan Masalah

Adapun hal-hal yang menjadi batasan masalah pada penelitian ini adalah sebagai berikut :

1. Aplikasi ini hanya dibuat untuk Kantor Wilayah BPN di Kabupaten Konawe Selatan.
2. Aplikasi ini hanya menerapkan Algoritma RC4 untuk proses enkripsi dan deskripsi data *QR code*.
3. Aplikasi ini hanya dapat digunakan oleh pengguna yang telah terdaftar di dalam system.
4. Aplikasi ini hanya digunakan pada tahap awal pendaftaran.

1.4 Tujuan Penelitian

Tujuan penelitian dalam penelitian Tugas Akhir ini adalah sebagai berikut:

1. Menerapkan Algoritma RC4 untuk mengamankan data *QR code* pada Aplikasi Permohonan Pengajuan Hak Atas Tanah milik instansi pemerintah.
2. Mendigitalisasi proses pelayanan pengajuan permohonan hak atas tanah milik instansi pemerintah.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut :

1. Penulis dapat menerapkan algoritma *RC4 (Rivers Code 4)* sebagai metode pengamanan data *QR code* pada Aplikasi Permohonan Pengajuan Hak Atas Tanah milik instansi pemerintah.
2. Memberikan kemudahan kepada masyarakat untuk melakukan proses pengajuan permohonan hak atas tanah milik pemerintah.

1.6 Sistematika Penulisan

Sistematika penulisan tugas akhir digunakan untuk menjelaskan penulisan perbab sebagai berikut:

BAB I PENDAHULUAN

Bab ini menguraikan tentang latar belakang masalah, rumusan masalah, batasan masalah yang menjelaskan tentang batasan-batasan dari sistem yang dibuat agar tidak menyimpang dari ketentuan yang telah ditetapkan. Selain itu, juga terdapat tujuan dan manfaat penelitian Penerapan Algoritma RC4 pada penerapan QR-Code pada Aplikasi Permohonan Pengajuan Hak Atas Tanah Milik Pemerintah di Kabupaten Konawe Selatan.

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.

1.7 Tinjauan Pustaka

(Syahdan et al., 2017) melakukan penelitian yang berjudul Penggunaan *QR code* dengan Enkripsi Vigenere Cipher dalam Pengamanan Data. Peneliti menyimpulkan bahwa *QR code* memiliki kemampuan untuk menampung cukup banyak informasi dalam bentuk yang berukuran kecil. Selain itu *QR code* juga memiliki kemampuan untuk mengoreksi kesalahan. Sedangkan Vigenere Cipher merupakan persandian yang relatif sederhana. Sehingga berdasarkan uraian di atas, dapat disimpulkan bahwa *QR code* yang dimodifikasi dengan enkripsi Vigenere Cipher dapat mempermudah seseorang dalam mengamankan data. Adapun saran dari penulis untuk penelitian lebih lanjut, yaitu Pemanfaatan *QR code* dengan enkripsi Vigenere Cipher dapat lebih difokuskan pada tujuan tertentu seperti pengamanan dokumen penting, dan sebagainya. Selain itu, disarankan juga menggabungkan *QR code* dengan enkripsi data yang lebih baru sehingga dapat menghasilkan sistem pengamanan data yang lebih baik.

(Muzaqqi & Rahayu, 2017) melakukan penelitian yang berjudul Implementasi Kriptografi Algoritma Exclusive Or Kombinasi Algoritma Rc4 Sebagai Pengamanan Dokumen Dengan Memanfaatkan QR-Code. Peneliti memiliki kesimpulan bahwa Algoritma XOR yang dikombinasikan dengan RC4 dapat diimplementasikan pada QR-Code dan dapat didekripsi serta Kriptografi Algoritma Exclusive OR yang dikombinasikan dengan Algoritma RC4 dan diimplementasikan ke dalam bentuk QR-Code pada dokumen dapat mengurangi kecurangan/pemalsuan terhadap dokumen.

(Suryani, 2017) Dalam penelitian tersebut, kesimpulan yang dapat diambil dari studi pemakaian metode RC4 sebagai metode enkripsi adalah, RC4 adalah sebuah stream cipher yang sinkron yang dapat dijalankan dengan panjang kunci variabel dan mengenkripsi suatu plainteks secara digit per digit dengan kunci simetris. RC4 merupakan metode enkripsi tercepat dibandingkan dengan DES, Triple DES, Blowfish-256, AES-128, dan AES-256.

BAB II

LANDASAN TEORI

2.1 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. (Andika, 2015).

Dalam Kriptografi terdapat beberapa aspek keamanan, antara lain sebagai berikut:

1. Kerahasiaan (*confidentiality*), menjamin bahwa data-data tersebut hanya bisa diakses oleh pihak-pihak tertentu saja. Kerahasiaan bertujuan untuk melindungi suatu informasi dari semua pihak yang tidak berhak atas informasi tersebut.
2. Otentikasi (*authentication*), merupakan identifikasi yang dilakukan oleh masing – masing pihak yang saling berkomunikasi, maksudnya beberapa pihak yang berkomunikasi harus mengidentifikasi satu sama lainnya. Informasi yang didapat oleh suatu pihak dari pihak lain harus diidentifikasi untuk memastikan keaslian dari informasi yang diterima.
3. Integritas (*integrity*), menjamin setiap pesan yang dikirim pasti sampai pada penerimanya tanpa ada bagian dari pesan tersebut yang diganti, diduplikasi, dirusak, diubah urutannya, dan ditambahkan. Integritas data bertujuan untuk

mencegah terjadinya pengubahan informasi oleh pihak-pihak yang tidak berhak atas informasi tersebut. Untuk menjamin integritas data ini pengguna harus mempunyai kemampuan untuk mendeteksi terjadinya manipulasi data oleh pihak-pihak yang tidak berkepentingan. Manipulasi data yang dimaksud di sini meliputi penyisipan, penghapusan, maupun penggantian data.

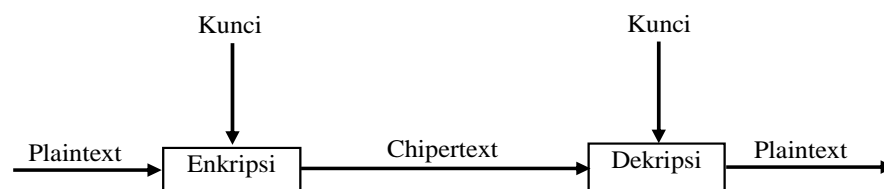
4. Nirpenyangkalan (*Nonrepudiation*), mencegah pengirim maupun penerima mengingkari bahwa mereka telah mengirimkan atau menerima suatu pesan. Jika sebuah pesan dikirim, penerima dapat membuktikan bahwa pesan tersebut memang dikirim oleh pengirim yang tertera. Sebaliknya, jika sebuah pesan diterima, pengirim dapat membuktikan bahwa pesannya telah diterima oleh pihak yang ditujunya. (Andika, 2015).

2.1.1 Pembagian Algoritma Kriptografi

Berdasarkan kunci pemecahannya algoritma kriptografi juga dibagi dalam dua jenis, yaitu :

a. Kriptografi Simetri

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

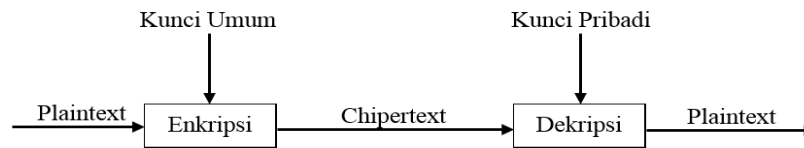


Gambar 2.1 Proses Enkripsi Deskripsi Menggunakan Kriptografi Simetri

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 2.2 Proses Enkripsi Deskripsi Menggunakan Kriptografi Asimetri

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

2.2 Algoritma RC4

Algoritma RC4 adalah algoritma kriptografi simetrik. Disebut algoritma kriptografi simetrik karena menggunakan kunci yang sama untuk mengenkripsi ataupun mendekripsi suatu pesan, data, ataupun informasi. Kunci enkripsi didapat dari sebuah 256 bit state-array yang diinisialisasi dengan sebuah *key* tersendiri dengan panjang 1-256 bit. Setelah itu, state-array tersebut akan diacak kembali dan diproses untuk menghasilkan sebuah kunci enkripsi yang akan di-XOR-kan dengan plainteks ataupun cipherteks.

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

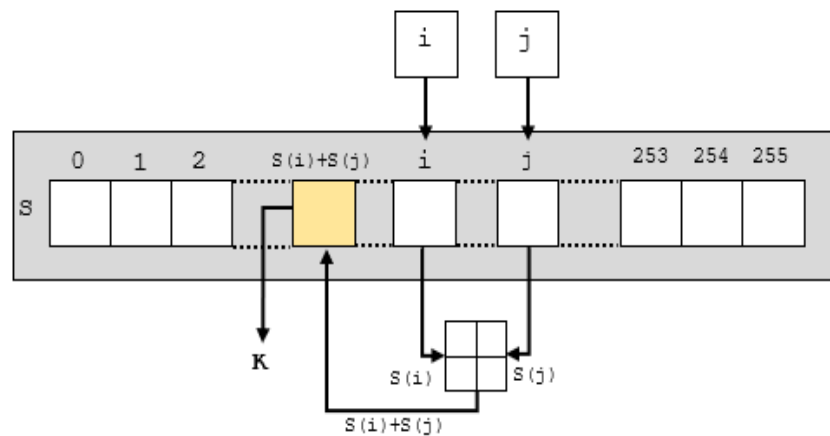
```

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

```

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

PRGA (*Pseudo-Random Generation Algorithm*) 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 2.3 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

```

2.2.1 Inisialisasi State Array

Dalam penginisialisasian state-array, terdapat 2 statearray yang harus diinisialisasi, S dan K . Array S sebesar 256 bit diinisialisasi dengan angka dari 0 sampai dengan 255. Sedangkan array K sebesar 256 bit diisi dengan *key* dengan panjang 1-256 bit secara berulang sampai seluruh array K terisi penuh. Setelah itu, dilakukan *Key Scheduling Algorithm* untuk menghasilkan permutasi dari array S berdasarkan *key* yang tersedia. for i from 0 to 255 $S[i] := i$ endfor j := 0 for i from 0 to 255 $j := (j + S[i] + key[i \bmod keylength]) \bmod 256$ swap(& $S[i]$, & $S[j]$) endfor Proses swap di atas adalah proses menukar nilai antara $S[i]$ dengan $S[j]$ (Suryani, 2017).

2.2.2 Penghasilan Kunci Enkripsi dan Pengenkripsian

Setelah memiliki state array yang telah teracak, maka kita akan menginisialisasi kembali i dan j dengan 0. Setelah itu, kita lakukan pseudo-random generation algorithm atau PRGA untuk menghasilkan kunci enkripsi (dalam hal ini cipher_byte) yang akan di-XOR-kan dengan plainteks. Untuk menghasilkan kunci enkripsi, PRGA meng-incremen i , menambahkan nilai $S[i]$ dan $S[j]$ menukar nilai keduanya, dan nilai kunci yang dihasilkan adalah S dengan indeks yang sama dengan jumlah $S[i]$ dan $S[j]$ di-modulo dengan 256 (Suryani, 2017).

2.2.3 Contoh Penerapan Algoritma RC4

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 penghasiian 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 penghasilan 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

N I S A : 01001110 01001001 01010011 01000001

Key : 00000010 00000011 00000001 00000001

Chiperteks : 01001100 01001010 01010010 01000000

2.3 *QR-Code*

QR-code adalah singkatan dari *quick response code*. Kode ini adalah barcode dua dimensi yang bisa memberikan beragam jenis informasi secara langsung. Untuk membukanya dibutuhkan *Scan* atau pemidai seperti kamera *handphone*. *QR code* biasanya mampu menyimpan 2089 digit atau 4289 karakter, termasuk tanda baca dan karakter spesial. Hal ini membuat *QR code* mampu menampilkan teks pada pengguna, membuka URL, menyimpan kontak ke buku telepon, dan masih banyak lagi.

QR-code dinilai lebih praktis dibanding *barcode* karena mampu menyimpan lebih banyak data. *QR code* terdiri dari titik-titik hitam dan spasi putih yang disusun dalam bentuk kotak, dan setiap elemennya memiliki makna tersendiri. Hal tersebut membuatnya mampu di-*scan* oleh *smartphone* dan menampilkan data atau informasi yang dimuatnya.

2.3.1 **Keuntungan dan Kerugian *QR-Code***

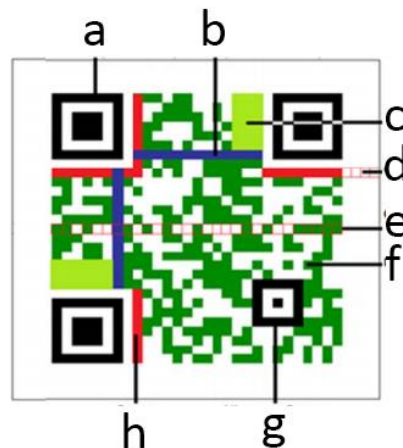
Berikut adalah pertimbangan dalam implementasi *QR code*. Keuntungan penggunaan *QR code* antara lain:

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

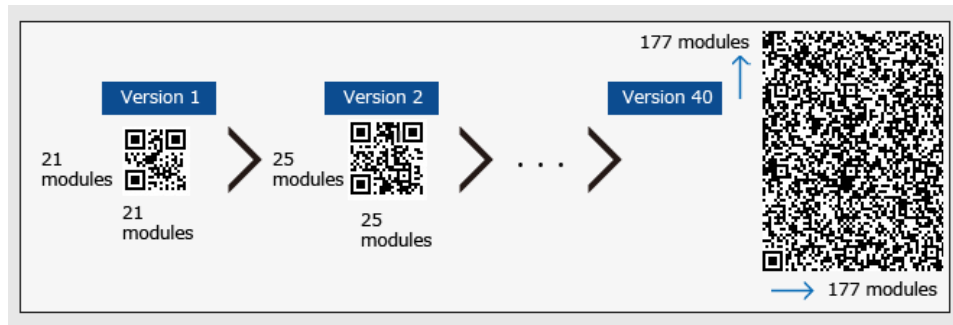


Gambar 2.4 Anatomi QR code (Sumber: qrcode.com)

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 2.5 Versi QR code (Sumber www.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 (21×21 modul) hingga Versi 40 (177×177 modul). Setiap nomor versi yang lebih tinggi berisi 4 modul tambahan di setiap sisi.

2.4 *Hyper Text Markup Language (HTML)*

HTML yang merupakan singkatan dari *Hyper Text Markup Language* adalah serangkaian kode program yang merupakan dasar dari representasi visual sebuah halaman *Web*. Didalamnya berisi kumpulan informasi yang disimpan dalam *tag-tag* tertentu, dimana *tag-tag* tersebut digunakan untuk melakukan format terhadap informasi yang dimaksud.

Berbagai pengembangan telah dilakukan terhadap kode *HTML* dan telah melahirkan teknologi-teknologi baru di dalam dunia pemrograman *web*. Kendati demikian, sampai sekarang *HTML* tetap berdiri kokoh sebagai dasar dari bahasa *web* seperti *PHP*, *ASP*, *JSP* dan lainnya. Bahkan secara umum, mayoritas situs *web* yang ada di Internet pun masih tetap menggunakan *HTML* sebagai teknologi utama (Constantianus & Suteja, 2005).

2.5 **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)

2.6 MySQL

MySQL (dibaca: mi-se-kyu-el) merupakan *software* yang tergolong sebagai *DBMS* (*Database Management System*) yang bersifat *open source*. *Open source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (code yang dipakai

untuk membuat *MySQL*). Selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi. Dan bisa diperoleh secara gratis dengan men-*download* di internet (Elib.unikom.ac.id, 2005).

SQL (Structured Query Language) merupakan sebuah bahasa relational yang berisi pernyataan yang digunakan untuk memasukkan, mengubah, menghapus, memilih dan melindungi data (Prihatna, 2005). *SQL* bukan *database* aplikasi, tetapi lebih berarti dengan suatu bahasa yang digunakan untuk mengajukan pertanyaan ke dalam *database* berupa pengguna *SQL*. *Database* sistem yang memiliki konsep sama dengan *SQL*, adalah Postgres dan *MySQL*, dimana *database* tersebut bisa didapatkan gratis atau dengan harga yang murah. *MySQL* adalah *server multithreaded*, sehingga sangat memungkinkan daemon untuk menghandle permintaan layanan secara stimultan. Model koneksi dengan protocol TCP-IP membuat akses ke server *database* lebih cepat jika dibandingkan dengan menggunakan *mapping drive*.

2.7 Database (Basis Data)

Basis data atau *Database* merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya yang diaorganisasikan sesuai struktur tertentu dan disimpan dengan baik. Untuk mendapatkan informasi yang berguna dari kumpulan data maka diperlukan suatu perangkat lunak (*software*) untuk memanipulasi data sehingga mendapatkan informasi yang berguna. (Echa & Buana, 2020). *Database* terbentuk dari beberapa komponen, yaitu :

1. Table

Table atau Tabel adalah sekumpulan data dengan struktur yang sedemikian rupa, terbentuk dari *record* dan *field*. Istilah Tabel disini berbeda dengan istilah Tabel pada HTML, walaupun secara visual hampir sama.

2. Record

Record adalah sekumpulan *field* yang membentuk suatu objek tertentu.

3. *Field*

Field adalah atribut dari objek yang memiliki tipe data tertentu.

2.8 *Code Igniter*



Code Igniter adalah sebuah web application network yang bersifat open source yang digunakan untuk membangun aplikasi php dinamis. Code Igniter menjadi sebuah *framework* PHP dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, Code Igniter juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih Code Igniter sebagai *framework* pilihannya. Karena kelebihan-kelebihan yang dimiliki oleh CodeIgniter, pembuat PHP Rasmus Lerdorf memuji Code Igniter di frOSCon (Kelen, 2018) dengan mengatakan bahwa dia menyukai CodeIgniter karena “*it is faster, lighter and the least like a framework.*”

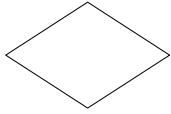

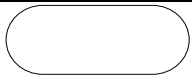
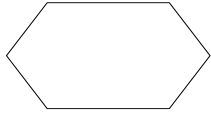
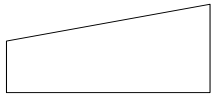

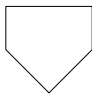
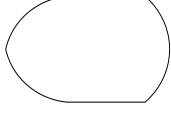
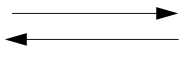
Code Igniter pertamakali dikembangkan pada tahun 2006 oleh Rick Ellis. Dengan logo api yang menyala, Code Igniter dengan cepat “membakar” semangat para *web developer* untuk mengembangkan web dinamis dengan cepat dan mudah menggunakan *framework* PHP yang satu ini.

2.9 *Flowchart*

Flowchart adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung sehingga setiap simbol *flowchart* melambangkan pekerjaan dan instruksinya (Ewolf, 2011). Simbol–simbol yang digunakan dalam *flowchart* adalah sebagai berikut:

Tabel 2.1 Simbol-simbol flowchart

NO.	GAMBAR	NAMA	KETERANGAN
1.		Proses	Mempresentasikan operasi.
2.		<i>Input / Output</i>	Mempresentasikan <i>Input</i> 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.
7.		Manual <i>input</i>	<i>Input</i> yang dimasukkan secara manual dari <i>keyboard</i> .
8.		Penghubung	Keluar atau masuk dari bagian lain <i>flowchart</i> khususnya.
9.		Penghubung	Keluar atau masuknya dari bagian lain <i>flowchart</i> khususnya halaman lain.
10.		<i>Display</i>	<i>Output</i> yang ditampilkan pada terminal.
11.		Anak panah	Mempresentasikan alur kerja.

Sumber: Ewolf, 2011.






2.10 Unified Modeling Language (UML)





Unified Modeling Language (UML) merupakan bahasa dalam mendesain perangkat lunak secara visual. Dengan UML, desainer dapat melihat konsep global suatu desain. Desain kemudian dapat dijadikan panduan dalam proses pengembangan dan rekayasa perangkat lunak. Selain itu, UML dapat menjadi media komunikasi gagasan antara pengembang perangkat lunak dengan pengguna (Shalahudin, 2011). Ada beberapa jenis diagram dalam UML yaitu:

1. Use Case Diagram

Use case diagram menggambarkan 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 2.2 Simbol-simbol Use Case Diagram

NO	GAMBAR	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 (<i>dependent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
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 <i>actor</i> .
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).


Sumber: Shalahuddin, M., 2011.

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 alir berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Tabel 2.3 Simbol Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Activity</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.

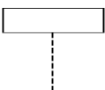


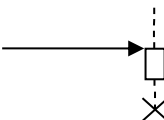
4.		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan diakhiri.
----	---	----------------------------	--

Sumber: Shalahuddin, M., 2011.

3. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu objek dan pesan yang dikirimkan dan diterima antar objek. Jumlah *Sequence diagram* yang akan digambar adalah minimal sama dengan jumlah *use case* yang didefinisikan (Sukamto, 2013).

Tabel 2.4 Simbol *Sequence Diagram* (Lanjutan)

NO	GAMBAR	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> .

Sumber: Sukamto dan Shalahuddin, 2013.

4. Class Diagram

Class diagram adalah kumpulan objek-objek yang mempunyai struktur umum, *behavior* umum, relasi umum, dan *semantic* atau kata yang umum. *Class-class* ditentukan atau ditemukan dengan cara memeriksa objek-objek

dalam *sequence diagram* dan *collaboration diagram*. Sebuah *class* digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan.

Tabel 2.5 Simbol Class Diagram

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

Sumber: Sukanto dan Shalahuddin, 2013.

2.11 Metode Pengembangan Sistem

Metode Pengembangan sistem yang digunakan pada tugas ini adalah metode *Rational Unified Process* (RUP). RUP merupakan salah satu proses rekayasa perangkat lunak yang menyediakan pendekatan untuk menentukan tugas

dan tanggung jawab dalam pengembangan suatu organisasi, tujuannya adalah untuk memastikan produksi kualitas tinggi, software memenuhi dengan kebutuhan user sesuai dengan jadwal dan biaya yang telah dirancang. Dalam metode RUP ini, terdiri dari 4 tahap, yaitu:

a. Inception

Tahap ini membangun *business case* untuk sistem dan membatasi ruang lingkungannya, untuk melakukan hal ini diharuskan untuk mengidentifikasi semua entitas eksternal yang akan berinteraksi dengan sistem, dan mendefinisikan interaksi pada level tertentu. Ini juga termasuk mengidentifikasi semua *use cases* dan menjelaskan beberapa yang signifikan. *Business case* termasuk kriteria keberhasilan, perkiraan resiko, dan mengestimasi sumber daya yang dibutuhkan.

b. Elaboration

Tujuan dari tahap *elaboration* adalah menganalisis domain masalah, membuat sebuah dasar arsitektur, membangun rencana proyek, dan mengeliminasi resiko terbesar dari proyek. Untuk menjalankan objek-objek tersebut diperlukan melihat lebih luas dan lebih dalam terhadap sistem. Pada tahap ini merupakan tahap paling sulit karena pada tahap ini memastikan bahwa arsitektur, kebutuhan, dan perencanaan cukup stabil sehingga waktu dan biaya tidak berubah.

c. Construction

Dalam tahap ini semua komponen dan fitur aplikasi yang dibuat dan diintegrasikan kedalam *software*. Dalam tahapan ini juga dituntut untuk mengoptimalkan sumber daya, biaya, jadwal dan kualitas. Pada tahapan ini meliputi bagaimana suatu aplikasi biasa diimplementasikan dan diuji coba.

1. Implementasi

Penjelasan mengenai perangkat keras dan perangkat lunak apa saja yang dibutuhkan untuk mengimplementasikan sistem.

2. Coding

Proses pengkodean dilakukan dengan menggunakan bahasa pemrograman. Pengkodean sendiri berisi tahapan-tahapan perhitungan metode.

3. *Testing*

Pada tahap ini dilakukan pengujian terhadap aplikasi yang telah dibangun untuk mengetahui tingkat akurasi dan kualitas dari aplikasi tersebut, apakah sudah sesuai dengan yang diharapkan atau tidak. *Testing* dilakukan dengan menguji semua tombol-tombol yang terdapat pada aplikasi apakah sudah berjalan sesuai dengan fungsi nya atau tidak.

d. *Transition*

Pada tahap ini dilakukan *testing* akhir pada sistem yang telah jadi, kemudian dilakukan sosialisasi penggunaan perangkat lunak yang telah dibangun ke *administrator*.

BAB III METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

3.1.1 Waktu

Waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari bulan Desember 2020 sampai dengan Februari 2021. Rincian kegiatan dapat dilihat pada Tabel 3.1 berikut:

Tabel 3.1 Gantt Chart Waktu Penelitian

No	Uraian	(2020)				(2021)							
		Desember				Januari				Februari			
		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 Kantor Pertanahan Kabupaten Konawe Selatan.

3.2 Metode Pengumpulan Data

Dalam penelitian ini metode pengumpulan data yang digunakan pada perancangan aplikasi tersebut adalah studi *literature*, mulai dari buku-buku, jurnal 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 algoritma RC4 (*Rivest Cipher 4*), sehingga Algoritma RC4 (*Rivest Cipher 4*) dapat diterapkan dalam Aplikasi Permohonan Pengajuan Hak Atas Tanah Milik Pemerintah di Kabupaten Konawe Selatan.

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 Algoritma RC4 (*Rivest Cipher 4*).

3.3.2 Perluasaan/Perencanaan (*Elaboration*)

Setelah menentukan ruang lingkup penelitian, tahap ini akan dilakukan perancangan dan analisis sistem menggunakan flowchart meliputi flowchart Aplikasi Pengajuan, dan flowchart Algoritma RC4 (*Rivest Cipher 4*). 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* Algoritma RC4 (*Rivest Cipher 4*) 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* pada form pengajuan.

3.4 Analisis Perancangan Sistem

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

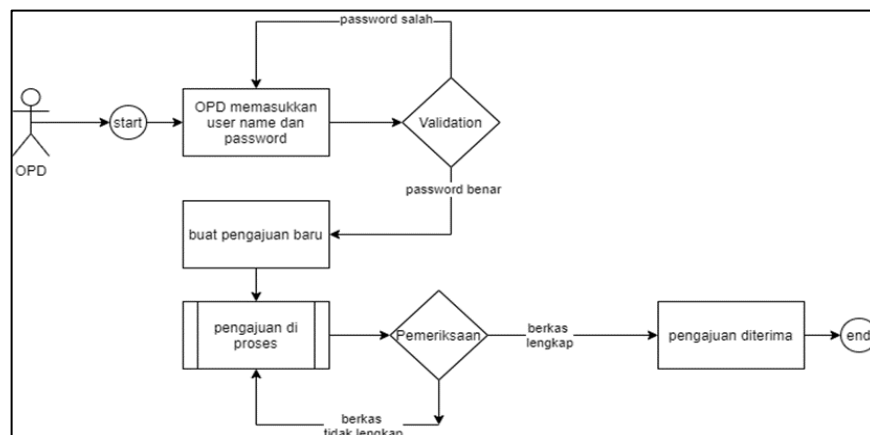
3.4.1 Flowchart

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

3.4.1.1 Flowchart Sistem

Setelah menganalisis sistem, maka didapatkan *flowchart diagram* untuk Aplikasi Pengajuan, dan *flowchart* Algoritma RC4 (*Rivest Cipher 4*). Adapun alur kerja *flowchart diagram* sistem ditunjukkan oleh Gambar 3.1 sebagai berikut:

- User mengunjungi laman *login* situs Aplikasi Pengajuan.
- User memasukkan *username* dan *password*.
- User masuk ke Menu Utama.
- User memilih menu “Pengajuan Baru”.
- User menekan tombol *download* untuk mengunduh blanko pengajuan.
- User melengkapi pengisian formulir.
- Pengajuan diproses, apabila berkas lengkap dan sesuai maka pengajuan diterima, apabila berkas tidak lengkap maka akan dilakukan pengajuan ulang.



Gambar 3.1 Flowchart Sistem Aplikasi Pengajuan

3.4.1.2 Flowchart Algoritma RC4 (Rivest Cipher 4)

Setelah menganalisis sistem, maka didapatkan *flowchart diagram* Algoritma RC4 (*Rivest Cipher 4*), yang ditunjukkan oleh Gambar 3.2. Adapun alur kerja *flowchart diagram* Algoritma RC4 (*Rivest Cipher 4*), yaitu :

- Pertama pengambilan data sebagai masukan plainteks/cipherteks dan masukan *key* yang akan digunakan.

- b. Kemudian, inisialisasi i dan j dengan 0 kemudian dilakukan KSA (*Key State Array*) agar tercipta state-array yang acak. Menggunakan rumus sebagai berikut :

```

for  $i$  from 0 to 255
     $S[i] := i$ 
endfor
 $j := 0$ 
for  $i$  from 0 to 255
     $j := (j + S[i] + key[i \bmod keylength]) \bmod 256$ 
     $swap(&S[i], &S[j])$ 
endfor

```

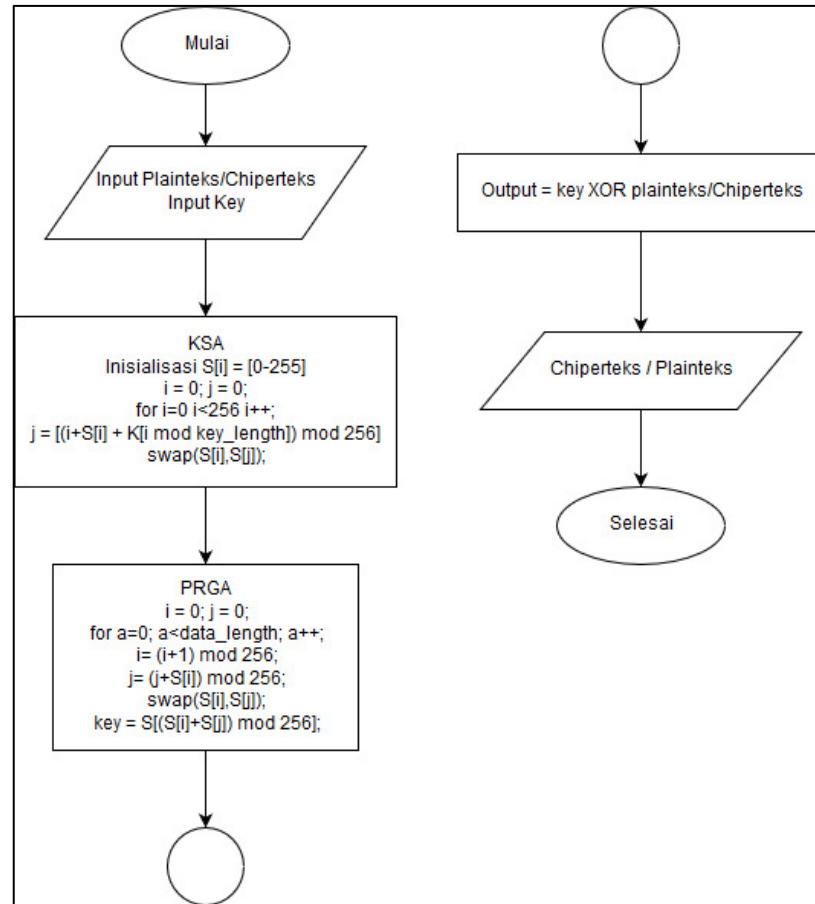
- c. Setelah melakukan KSA (*Key State Array*), akan dilakukan PRGA (*Pseudo-Random Generation Algorithm*) untuk menghasilkan kunci enkripsi yang akan di-XOR-kan dengan plainteks. Untuk menghasilkan kunci enkripsi, PRGA meng-*increment* i , menambahkan nilai $S[i]$ dan $S[j]$ menukar nilai keduanya, dan nilai kunci yang dihasilkan adalah S dengan indeks yang sama dengan jumlah $S[i]$ dan $S[j]$ di-modulo dengan 256.

```

 $i := 0$ 
 $j := 0$ 
while GeneratingOutput:
     $i := (i + 1) \bmod 256$ 
     $j := (j + S[i]) \bmod 256$ 
     $swap(&S[i], &S[j])$ 
     $key := S[(S[i] + S[j]) \bmod 256]$ 
endwhile

```

- d. Setelah menemukan kunci untuk tiap karakter, maka dilakukan operasi XOR antara karakter pada plaintext dengan kunci yang dihasilkan. Berikut adalah tabel ASCII untuk tiap-tiap karakter pada plaintks yang digunakan.
- e. Setelah terkirim, pesan yang telah dienkrpsi akan didekripsikan. Proses pendekripsian dilakukan dengan proses XOR antara kunci dekripsi yang sama dengan kunci dekripsi dengan cipherteks.



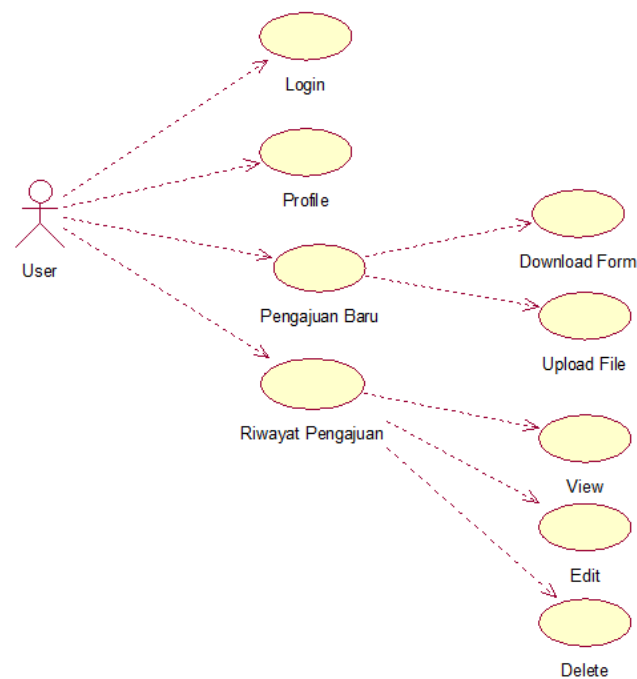
Gambar 3.2 Flowchart Algoritma RC4 (Rivest Cipher 4)

3.4.2 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*.

3.4.2.1 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 dapat ditunjukkan pada Gambar 3.3.



Gambar 3.3 Use Case Diagram Sistem

Adapun penjelasan mengenai actor dan system pada Use Case Diagram diatas adalah sebagai berikut:

Tabel 3.2 Keterangan Use Case Diagram

Aktor	Sistem
<i>User</i> (Pengaju/OPD) Melakukan <i>login</i>	Sistem akan memcocokkan email dan password yang user masukkan, jika email benar dan terdaftar maka <i>user</i> akan masuk ke halaman utama.
<i>User</i> (Pengaju) memilih menu Profile	Sistem akan menampilkan data profile pengaju yang terdaftar.
<i>User</i> (Pengaju) memilih menu Pengajuan Baru	Sistem akan menampilkan form pengajuan untuk pengaju mengisikan dan mengunggah (<i>Upload</i>) beberapa data yang diperlukan,

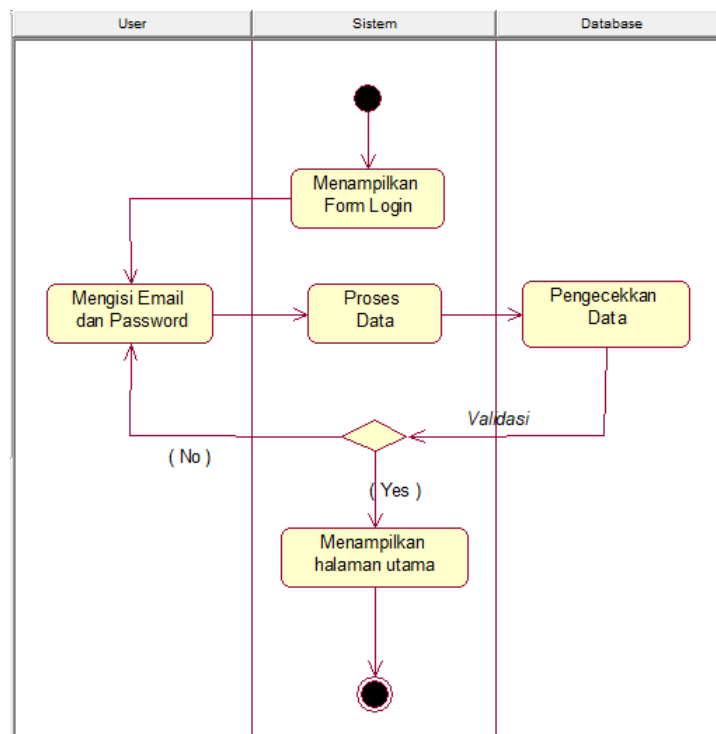
	serta pengaju dapat men- <i>download</i> blanko pengajuan.
<i>User</i> (Pengaju) memilih menu Riwayat pengajuan	Sistem akan menampilkan daftar pengajuan yang telah terkirim, diterima, ataupun ditolak.

3.4.2.2 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.

1. Activity Diagram Login

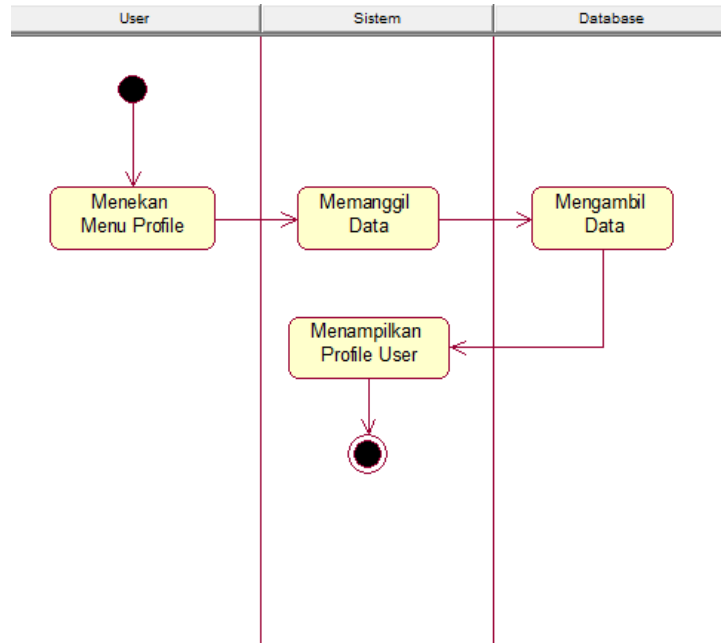
Gambar 3.4 merupakan diagram aktivitas yang menunjukkan aktivitas *user* melakukan login, kemudian sistem akan melakukan pencocokan email dan password.



Gambar 3.4 Activity Diagram Login

2. Activity Diagram Profile

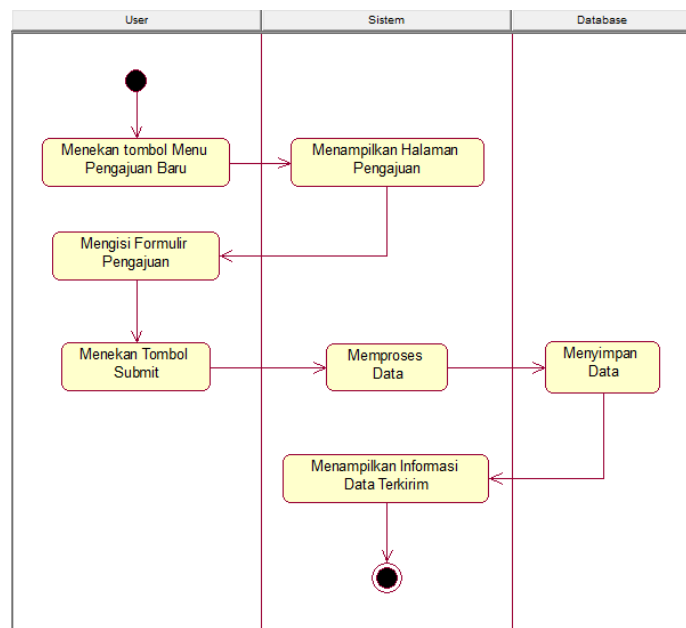
Gambar 3.5 tersebut merupakan diagram aktivitas yang menunjukkan aktivitas *user* ketika telah menekan tombol menu *Profile*, maka system akan menampilkan data diri/ *profile* pengaju yang sudah terdaftar dalam sistem.



Gambar 3.5 Activity Diagram Profile

3. Activity Diagram Pengajuan Baru

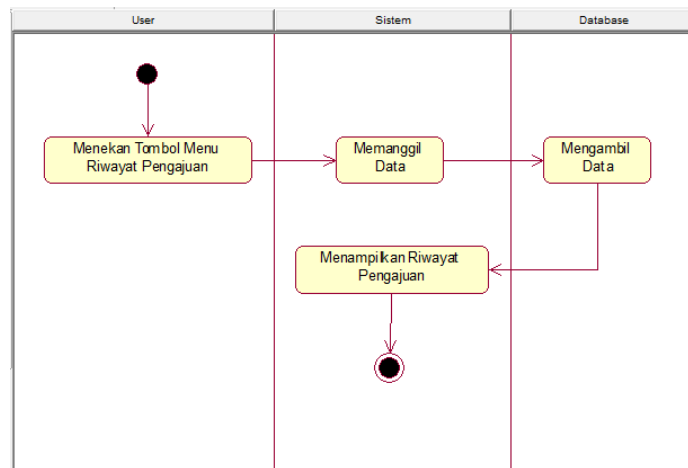
Gambar 3.6 merupakan gambar *Activity Diagram* yang menunjukkan aktivitas *user* memilih menu pengajuan baru, dimana sistem akan menyajikan form pengisian data pengajuan, setelah itu data *user* akan tersimpan kedalam *database* dan system akan memberikan keterangan jika data telah terikirim.



Gambar 3.6 Activity Diagram Pengajuan Baru

4. Activity Diagram Riwayat Pengajuan

Gambar 3.7 merupakan gambar *Activity Diagram* yang menunjukkan aktivitas *user* memilih menu Riwayat Pengajuan, system akan menampilkan daftar pengajuan yang telah dikirim oleh *user*, sedang diperiksa, ataupun berkas yang tertolak dari pihak instansi.

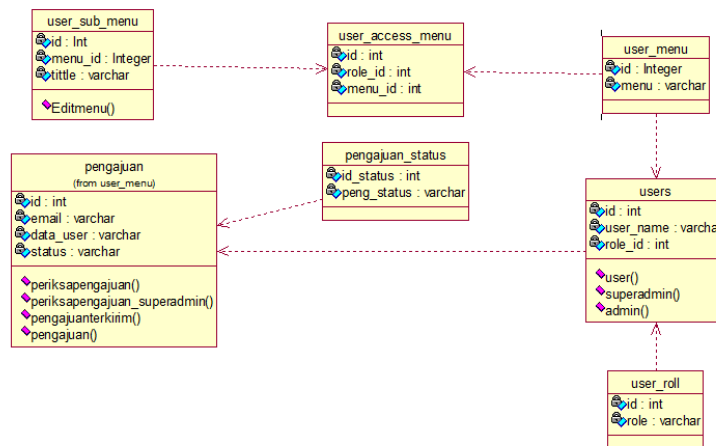


Gambar 3.7 Activity Diagram Riwayat Pengajuan

3.4.2.3 Class Diagram

Class diagram merupakan diagram yang selalu ada dipemodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* dalam sistem

yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Berikut ini adalah *class diagram* aplikasi pengajuan hak atas tanah milik instansi pemerintah.

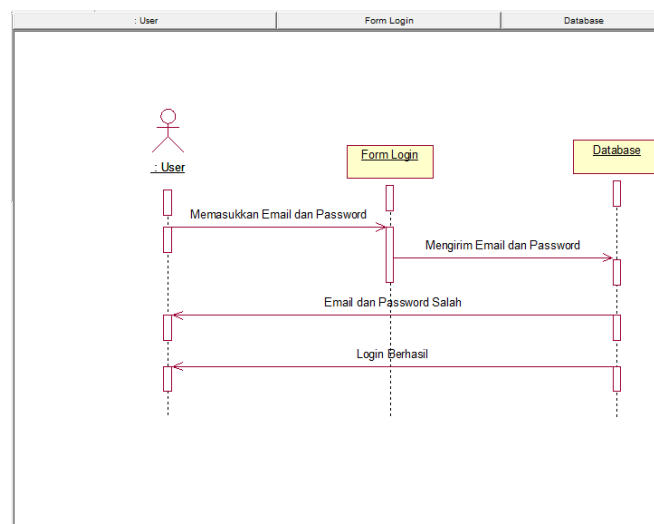


Gambar 3.8 Class Diagram Aplikasi Pengajuan

3.4.2.4 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.

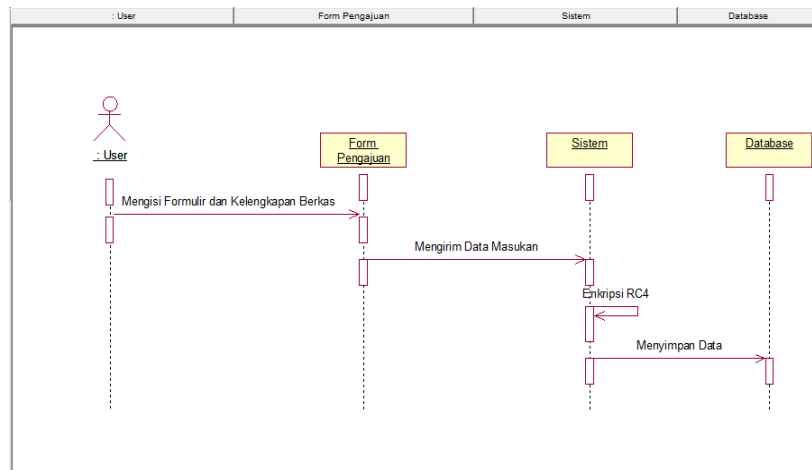
1. Sequence Diagram Login



Gambar 3.9 Sequence Diagram Login

Gambar 3.9 adalah *sequence* yang menampilkan proses login bagi pengaju, dengan memasukkan email dan password yang akan divalidasi oleh *database*.

2. Sequence Diagram Enkripsi



Gambar 3.10 Sequence Diagram Enkripsi Data

Gambar 3.10 adalah *sequence* yang menunjukkan proses enkripsi dimana data pengaju akan diproses oleh sistem dan melakukan enkripsi pada kode khusus yang juga dibuat oleh sistem baru kemudian disimpan ke *database*.

3.5 Perancangan Antarmuka (*Interface*)

Perancangan *user interface* adalah tahapan pembuatan antarmuka yang akan digunakan pada pembangunan aplikasi kamus yang dibagi menjadi beberapa bagian antara lain perancangan laman *login*, laman menu utama, laman pengajuan baru, laman riwayat pengajuan.

3.5.1 *Form Login*

Form login adalah tampilan pertama yang akan ditampilkan oleh sistem e-pintah. *Form* ini berfungsi untuk membatasi dan membagi tingkatan pengguna. *Email* dan *password* dibutuhkan untuk mengakses layanan e-pintah ini. Berikut tampilan *form login* :

The screenshot shows a web browser window titled "A Web Page" with the address bar displaying "http://epintah.co.id". The main content area features the title "APLIKASI E-PINTAH" in bold. Below the title is a login form with the heading "Masukkan User Name dan Password". The form contains two input fields: "user name" and "password", each followed by a text input box. A "Login" button is positioned below the password field.

Gambar 3.11 Form Login

3.5.2 Laman Menu Utama

Pada laman utama terdapat beberapa menu, yaitu menu ujian. Pada laman menu utama (*dashboard*), hanya berfungsi sebagai tampilan antarmuka pertama yang dijumpai oleh user saat berhasil *login*.

The screenshot shows the main menu dashboard of the E-PINTAH application. The browser window title is "A Web Page" and the address bar shows "http://epintah.co.id". The dashboard has a header bar with "E-PINTA" on the left and a user profile icon with an "EXIT" button on the right. On the left side, there is a vertical menu with four items: "provile", "pengajuan baru", "riwayat ajuan", and "proses ajuan". The main content area displays the message "SELAMAT DATANG DI APLIKASI E-PINTAH" in bold.

Gambar 3.12 Laman Menu Utama

3.5.3 Laman Menu Pengajuan Baru

Pada laman pengajuan baru, *user* dapat menginputkan beberapa data dan juga mengunggah *file* yang diperlukan sebagai pelengkap berkas pengajuan.

The screenshot shows a web browser window titled 'A Web Page' with the URL 'http://epintah.co.id/pengajuan_baru'. The page header includes 'E-PINTAH' and an 'EXIT' button. On the left, there is a sidebar menu with four items: 'proville', 'pengajuan baru', 'riwayat ajuan', and 'dalam proses ajuan'. The main content area is titled 'MASUKKAN IDENTITAS SUBJEK' and contains three text input fields: the first two are labeled 'type text/number/date' and the third is labeled 'type text'. Below these fields is a button labeled 'tambahkan file' with a plus icon. A vertical scrollbar is visible on the right side of the form area.

Gambar 3.13 Laman Menu Pengajuan

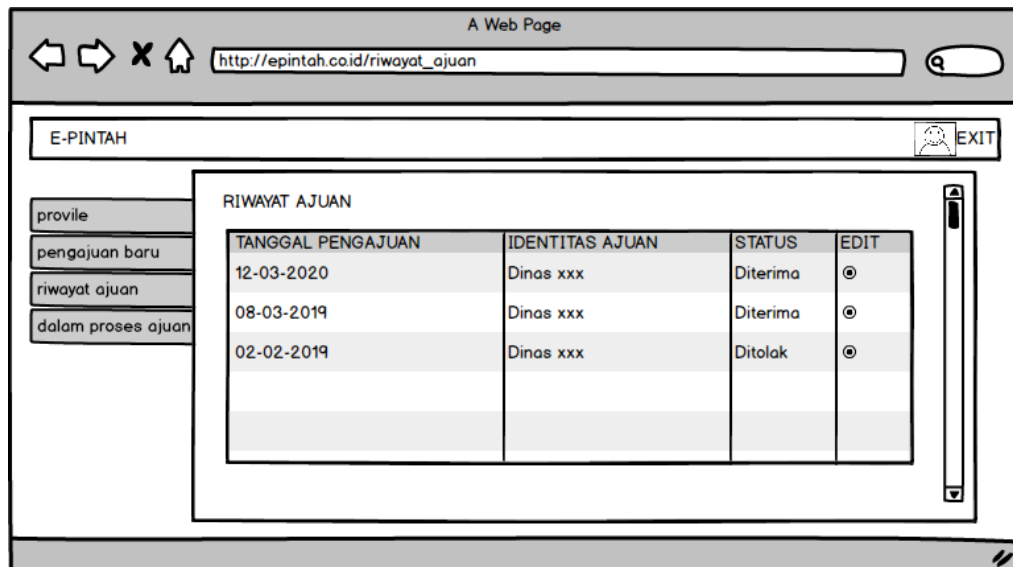
Setelah melengkapi semua data, maka user akan menekan tombol kirim, sebelum itu User harus memastikan data yang dimasukkan telah benar, dengan memberi centang pada kolom yang tersedia.

The screenshot shows the same web browser window, but the main content area is now titled 'MASUKKAN IDENTITAS OBJEK'. It contains three text input fields: the first two are labeled 'type text/number/date' and the third is labeled 'type text'. Below these fields is a button labeled 'tambahkan file' with a plus icon. At the bottom of the form, there is a checkbox with the text 'dengan mencentang kolom ini maka anda telah menyatakan data yang anda masukkan telah disetujui oeh lembaga/instansi terkait serta data yang anda masukkan adalah falid dan dapat dipertanggung jawabkan secara hukum'. To the right of the checkbox is a button labeled 'KIRIM'. A vertical scrollbar is visible on the right side of the form area.

Gambar 3.14 Laman Menu Pengajuan

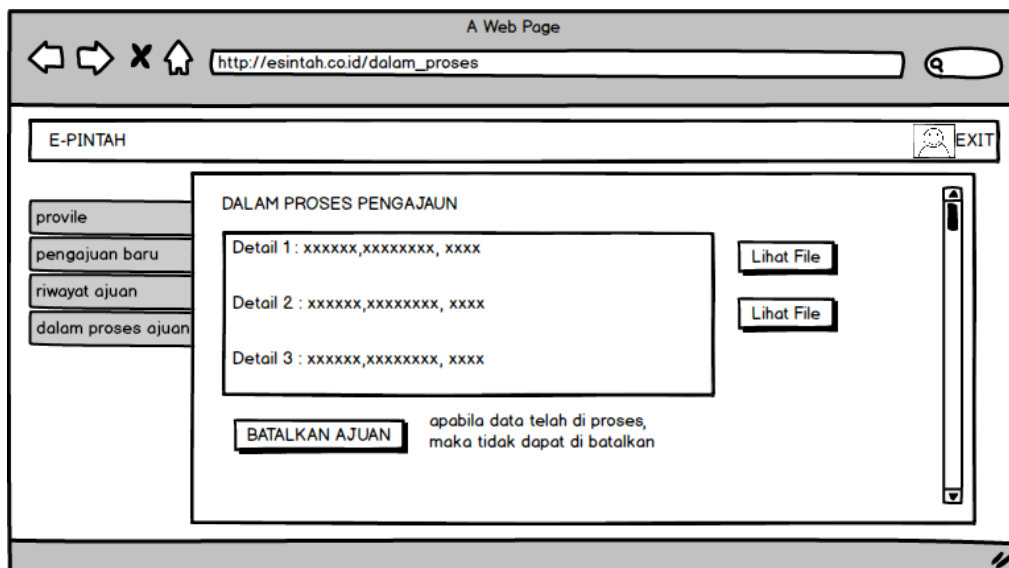
3.5.4 Laman Riwayat Pengajuan

Pada saat *user* mengirim data, data akan masuk dan tersimpan. Setelah itu, maka *user* dapat memantau data telah diterima atau belum oleh pihak instansi melalui laman riwayat pengajuan.



Gambar 3.15 Laman Riwayat Pengajuan

Selain Riwayat Pengajuan, User dapat membatalkan pengajuan yang telah terkirim. Dengan syarat pengajuan tersebut belum diterima/diproses.



Gambar 3.16 Laman Riwayat Pengajuan

3.6 Metode Pengujian

Pengujian merupakan metode yang dilakukan untuk menjelaskan mengenai pengoperasian perangkat lunak yang terdiri dari perangkat pengujian, metode pengujian dan pelaksanaan pengujian. Pengujian program ini menggunakan metode *black box*. Pengujian *black box* merupakan pengujian program berdasarkan fungsi dari program. Tujuan dari metode *black box* ini adalah untuk menemukan kesalahan fungsi pada program.

Pengujian dengan metode ini dilakukan dengan cara memberikan sejumlah *input* pada program aplikasi yang kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah program aplikasi menghasilkan keluaran yang diinginkan dan sesuai dengan fungsi dari program tersebut. Apabila dari masukan yang diberikan proses menghasilkan keluaran yang sesuai dengan kebutuhan fungsionalnya, maka program aplikasi yang bersangkutan telah benar, tetapi jika keluaran yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, maka masih terdapat kesalahan pada program aplikasi.

Pengujian dilakukan dengan mencoba semua kemungkinan yang terjadi dan pengujian dilakukan berulang-ulang. Jika dalam pengujian ditemukan kesalahan, maka akan dilakukan penelusuran dan perbaikan untuk memperbaiki kesalahan yang terjadi. Jika telah selesai melakukan perbaikan, maka akan dilakukan pengujian kembali. Pengujian dan perbaikan dilakukan secara terus menerus hingga diperoleh hasil yang terbaik.

3.6.1 Rencana Pengujian

Pengujian perangkat lunak berikut menggunakan data uji berdasarkan data dari masing-masing data. Rencana selengkapnya dapat dilihat pada tabel berikut :

Tabel 3.3 Rencana Pengujian

Kelas Uji	Detail Uji	Jenis Pengujian
Pengujian Menu	Login	<i>Black Box</i>
	Pengajuan Baru	<i>Black Box</i>
	<i>Profile</i>	<i>Black Box</i>
	Riwayat Pengajuan	<i>Black Box</i>
Pengujian Algoritma RC4 (<i>Rivest Cipher 4</i>)	Hasil Enkripsi data	<i>Black Box</i>
	Pembacaan <i>QR-Code</i> (Dekripsi)	<i>Black Box</i>

DAFTAR PUSTAKA

- Andika, D. (2015). Pengertian dan Sejarah Kriptografi. In *IT Jurnal*. <http://www.it-jurnal.com/2015/05/pengertian-dan-sejarah-kriptografi.html>
- Ani, N., Deby, R., Nugraha, M. P., & Munir, R. (2011). Pengembangan Aplikasi *QR code* Generator dan *QR code* Reader dari Data Berbentuk Image. *Konferensi Nasional Informatika – KNIF 2011*, 148–155.
- Anonymous. (2012). Bab ii landasan teori 2. 1. In *Www.Google.Com* (pp. 4–15). http://www.fali.unsri.ac.id/userfiles/Bentuk_dan_Jenis_Antena.pdf.
- Ashford, R. (2010). *Digital Commons @ George Fox University QR codes and academic libraries: Reaching mobile users*. https://digitalcommons.georgefox.edu/libraries_fac
- Constantianus, F., & Suteja, B. (2005). Analisa dan Desain Sistem Bimbingan Tugas Akhir Berbasis Web dengan Studi Kasus Fakultas Teknologi Informasi. *Jurnal Informatika*, 1(2), 93–106.
- Dedy irawan, J., & Adriantantri, E. (2019). Pemanfaatan Qr-Code Segabai Media Promosi Toko. *Jurnal Mnemonic*, 1(2), 56–61. <https://doi.org/10.36040/mnemonic.v1i2.39>
- Denso Wave. (2016). History of *QR code*. In *Denso Wave Incorporated* (pp. 1–5). <http://www.qrcode.com/en/history/>
- Echa, S. S., & Buana, U. M. (2020). ARTIKEL TUGAS SISTEM INFORMASI MANAJEMEN " SISTEM MANAJEMEN BASIS DATA ". *October*.
- Elib.unikom.ac.id. (2005). *Pengertian Xampp Version 1.7*. 8–27. <http://elib.unikom.ac.id/files/disk1/491/jbptunikompp-gdl-muhammadri-24505-4-babiil-i.pdf>
- Kantor Pertanahan Kab Bogor. (n.d.).
- Kelen, L. (2018). Implementasi Model-View-Controller (Mvc) Pada Ujian Online Melalui Penerapan Framework Codeigniter. *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*, 1(1), 10–16. <https://doi.org/10.37792/jukanti.v1i1.5>
- Muharom, L. A., & Sholeh, M. L. (2016). SMART PRESENSI MENGGUNAKAN QR-Code DENGAN ENKRIPSI VIGENERE CIPHER. *Limits: Journal of Mathematics and Its Applications*, 13(2), 31. <https://doi.org/10.12962/j1829605x.v13i2.1933>
- Muzaqqi, N., & Rahayu, Y. D. (2017). Implementasi Kriptografi Algoritma Exclusive or Kombinasi Algoritma Rc4 Sebagai Pengamanan Dokumen Dengan Memanfaatkan Qr-Code. *Journal of Undergraduate Thesis, Universitas Muhammadiyah Jember*, 1210651119.
- Suryani, K. N. (2017). Algoritma Rc4 Sebagai Metode Enkripsi. *Enkripsi*, 4, 5.
- Syahdan, R., Anitasari, E., Pendidikan, P., Program Pascasarjana, M., &

Yogyakarta, U. N. (2017). *Penggunaan QR code dengan Enkripsi Vigenere Cipher dalam Pengamanan Data.*