FILE TO IMAGE ENCRYPTION (FTIE) MENGGUNAKAN ALGORITMA RANDOMIZED TEXT DAN ARNOLD CAT MAP (ACM)

Disusun Oleh:

N a m a : Johdhy Prasojo
NIM : 12523217

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2018

ii

HALAMAN PENGESAHAN DOSEN PEMBIMBING

FILE TO IMAGE ENCRYPTION (FTIE) MENGGUNAKAN ALGORITMA RANDOMIZED TEXT DAN ARNOLD CAT MAP (ACM)

TUGAS AKHIR

Disusun Oleh:

N a m a : Johdhy Prasojo

NIM : 12523217

Yogyakarta, 16 Agustus 2018

Pembimbing,

(Yudi Prayudi, S.Si., M.Kom)

HALAMAN PENGESAHAN DOSEN PENGUJI

FILE TO IMAGE ENCRYPTION (FTIE) MENGGUNAKAN ALGORITMA RANDOMIZED TEXT DAN ARNOLD CAT MAP (ACM)

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Teknik Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 16 Agustus 2018

Tim Penguji

Yudi Prayudi, S.Si., M.Kom

Anggota 1

Fietyata Yudha, S.Kom., M.Kom

Anggota 2

Syarif Hidayat, S.Kom., M.I.T

Mengetahui,

Ketua Program Studi Teknik Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

Dr. Raden Teduh Dirgahayu, S.T., M.Sc.

iv

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Johdhy Prasojo

NIM : 12523217

Tugas akhir dengan judul:

FILE TO IMAGE ENCRYPTION (FTIE) MENGGUNAKAN ALGORITMA RANDOMIZED TEXT DAN ARNOLD CAT MAP (ACM)

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 7 Agustus 2018

(Johdhy Prasojo)

HALAMAN PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Syukur Alhamdulillah kehadirat Allah Subhanallahu wa Ta'ala atas rahmat-Nya sehingga karya sederhana ini dapat terselesaikan

Karya sederhana ini dipersembahkan untuk:

Papaku Bapak Waluyo Abu Saputro dan Mamaku ibu Ida Rusdiati

Bapak Ibu berdua adalah pahlawan sejati dalam hidupku yang selalu membimbing dan memberikan do'a serta semangat buat saya dengan tak pernah lelah mendidik saya untuk selalu mencari ilmu, belajar, ibadah, dan berdo'a. Terima kasih juga untuk semua pelajaran berharga yang telah kalian ajarkan selama ini hingga membuat saya menjadi anak yang selalu tidak mudah menyerah dalam menjalani hidup ini.

Kakakku

Mba Dyah Puspita Dewi

Terimakasih, atas segala doa dan dukungannya selama ini, walaupun usia kita terpaut jauh, kalian tetap menjadi panutanku dari kecil hingga sekarang. Semoga kita selalu bisa membahagiakan papa dan mama.

HALAMAN MOTO

*ارْغَبْفَ رَبِّكَ وَإِلَى * فَانصَبْ فَرَغْتَ فَإِذَا * يُسْرًا الْعُسْرِ مَعَ إِنَّ * يُسْرًا الْعُسْرِ مَعَ فَإِنَّ

Karena sesungguhnya sesudah kesulitan itu ada kemudahan
Maka apabila kamu telah selesai (dari sesuatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain
Dan hanya kepada Tuhanmulah hendaknya kamu berharap (QS. Al-Insyirah: 5-8)

"Pendidikan merupakan perlengkapan paling baik untuk hari tua." (Aristoteles)

"Musuh yang paling berbahaya di atas dunia ini adalah penakut dan bimbang. Teman yang paling setia, hanyalah keberanian dan keyakinan yang teguh." (Andrew Jackson)

"Semua orang tidak perlu menjadi malu karena pernah berbuat kesalahan, selama ia menjadi lebih bijaksana daripada sebelumnya." (Alexander Pope)

“Wisuda setelah 12 semester adalah kesuksesan yang tertunda.”

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah Subhanahu Wa Ta'ala, yang telah memberikan rahmat dan karunia-Nya sehingga penulisan laporan tugas akhir ini dapat diselesaikan dengan lancar.

Dalam laporan tugas akhir ini penulis mengambil judul "*File To Image Encryption* (FTIE) menggunakan Algoritma *Randomized Text* dan *Arnold Cat Map (ACM)* ".

Adapun tujuan dari penulisan laporan tugas akhir ini adalah untuk melengkapi program perkuliahan S1 pada Fakultas Teknologi Industri jurusan Teknik Informatika Universitas Islam Indonesia.

Pada Kesempatan yang sama, penulis juga ingin menyampaikan ucapan terima kasih kepada yang terhormat :

1. Dekan Fakultas Teknologi Industri UII.
2. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua jurusan Teknik Informatika.
3. Bapak Yudi Prayudi S.Si., M.Kom. selaku Dosen pembimbing yang mengarahkan penyusunan laporan tugas akhir ini.
4. Kedua orang tua kami serta saudara-saudara tercinta yang telah memberikan dorongan material dan spiritual.
5. Dan semua pihak yang telah mendukung penyusun yang tidak dapat penyusun sebutkan satu persatu.

Harapan penulis semoga laporan ini dapat bermanfaat bagi penulis sendiri dan juga bagi semua pihak yang membacanya dalam menambah wawasan serta pengetahuan.

Akhir kata kami menyadari kekurangan-kekurangan, baik mengenai materi maupun penyusunan-nya dikarenakan batas kemampuan dan waktu penulis. Oleh karena itu kami dengan senang hati akan menerima segala kritik dan saran yang di tujukan untuk perbaikan agar menjadikan laporan ini menjadi sempurna.

Yogyakarta, 7 agustus 20178

(Johdhy Prasojo)

SARI

File To Image Encryption (FTIE) adalah teknik enkripsi data dengan cara mengubah sebuah file menjadi sebuah gambar. FTIE adalah teknik yang dikembangkan dari teknik Text To Image Encryption (TTIE), yang dimana TTIE melakukan enkripsi dari sebuah teks menjadi gambar, sedangkan FTIE melakukan enkripsi dari sebuah file menjadi sebuah gambar. Pada penelitian sebelumnya sudah dibuat teknik FTIE dengan algoritma Randomized Text dan Arnold Cat Map (ACM) yang sama, tetapi masih ada kekurangan pada proses aplikasi yaitu aplikasi sebelumnya belum bisa diakses secara online. atas permasalahan tersebut pada penelitian ini didapatkan solusi bagaimana merancang aplikasi FTIE dengan menggunakan algoritma Randomized Text dan ACM agar dapat menghasilkan sebuah aplikasi yang dapat di akses secara online. Tujuan dibuatnya aplikasi ini untuk mengamankan sebuah informasi data. Setelah melakukan penelitian dan pembuatan aplikasi FTIE, diperoleh hasil enkripsi dengan menggunakan algoritma Arnold Cat Map (ACM) dan Randomized Text memiliki ukuran 2 kali lebih besar dari ukuran asli dan hasil analisis entropy untuk uji keamanan menghasilkan nilai rata-rata sebesar 7,82690875, yang menyatakan bahwa jika nilai hasil enkrpisi mendekati 8 maka hasil enkripsi tersebut dapat dinyatakan aman dari serangan atau sulit ditebak kriptanalisis.

Kata kunci: FTIE, TTIE, Arnold Cat Map(ACM), Randomized Text,

GLOSARIUM

Plaintext	pesan yang dapat dibaca
Chiper text	pesan acak yang tidak bisa dibaca
Website	halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet.
File	adalah kumpulan berbagai informasi yang berhubungan dan juga tersimpan di dalam secondary storage, secara konsep file memiliki beberapa tipe ada yang bertipe Data terdiri dari numeric, character dan binary.
Download	proses dalam pengambilan file-file tertentu yang terdapat di internet baik melalui web server, FTP server, mail server, server ataupun sistem lain yang identik.
Database	kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.
Byte	istilah yang biasa digunakan sebagai satuan dari penyimpanan data dalam komputer.
Password	kumpulan karakter atau string yang digunakan oleh pengguna jaringan atau sebuah sistem operasi yang mendukung banyak pengguna (multiuser) untuk memverifikasi identitas dirinya kepada sistem keamanan yang dimiliki oleh jaringan atau sistem tersebut.
Modulo (M)	Misalkan dua bilangan a dan b, a modulo b (disingkat a mod b) adalah bilangan bulat sisa pembagian a oleh b. Misalnya, "1 mod 3", "4 mod 3", dan "7 mod3" memiliki hasil 1, karena ketiga bilangan tersebut memiliki sisa 1 jika dibagi oleh 3, sedangkan "9 mod3" sama dengan 0.
Form	Form merupakan salah satu bentuk halaman web yang digunakan untuk menerima masukan dari pengguna, untuk selanjutnya masukan dari pengguna tersebut diolah menggunakan bahasa pemrograman web, baik secara server side scripting(misalkan PHP, JSP) ataupun client-side scripting (javascript).

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING **Error! Bookmark not defined.**
HALAMAN PENGESAHAN DOSEN PENGUJI ..... **Error! Bookmark not defined.**
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR..... **Error! Bookmark not defined.**
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI viii
GLOSARIUM ix
DAFTAR ISI x
DAFTAR TABEL xiii
DAFTAR GAMBAR xiv
BAB I PENDAHULUAN 1
8.1 Latar Belakang 1
8.2 Rumusan Masalah 2
8.3 Batasan Masalah 3
8.4 Tujuan Penelitian 3
8.5 Manfaat Penelitian 4
8.6 Metodologi Penelitian 4
8.7 Sistematika Penulisan 5
BAB II LANDASAN TEORI 6
2.1 Literature *review* 6
2.2 Kriptografi 8
2.2.1 Definisi kriptografi 8
2.2.2 Tujuan kriptografi 9
2.2.3 Terminologi kriptografi 9
2.2.4 Kebutuhan akan kriptografi 10
2.2.5 Jenis-jenis serangan algoritma kriptografi 15
2.3 Sistem *ASCII* 16
2.4 Text to *image encryption* (TTIE) 17
2.5 Randomized Text 19
2.6 Arnold Cat Map 20
2.7 Pemrograman *Hypertext Preprocessor* (PHP) 20
BAB III METODOLOGI PENELITIAN 22
3.1 Bagan Alir (*Flowchart*) Metode Penelitian 22
3.2 Perancangan Sistem 24
3.2.1 Flowchart *File* To *Image Encryption* (FTIE) dengan menggunakan algoritma Randomized Text dan Arnold Cat Map 24
3.2.2 Perancangan Model Enkripsi 26
3.2.3 Perancangan Model Dekripsi 27
3.3 Kebutuhan Sistem 29
3.3.1 Kebutuhan Perangkat Lunak 29
3.4 Perancangan Aplikasi 30
3.4.1 Use Case Diagram 30
3.4.2 Perancangan Diagram Activity (*Login* administrator) 31
3.4.3 Perancangan Diagram Activity (*Login User*) 32

3.4.4 Perancangan Diagram Activity (*User*) 34
3.4.5 Perancangan Diagram Activity *User* (Halaman Enkripsi) 35
3.4.6 Perancangan Diagram Activity *User* (Halaman Dekripsi) 36
3.5 Perancangan *Basisdata* 37
3.5.1 Entity Relationship Diagram (ERD) 37
3.5.2 Tabel Data Administrator 37
3.5.3 Tabel Data *User* 38
3.5.4 Tabel Data Enkripsi 38
3.5.5 Tabel Data Dekripsi 39
3.6 Perancangan Antarmuka 40
3.6.1 Perancangan Antarmuka *Login* Admin 40
3.6.2 Perancangan Antarmuka *Login User* 40
3.6.3 Perancangan Antarmuka Daftar *User* 41
3.6.4 Perancangan Antarmuka *Home* Admin 42
3.6.5 Perancangan Antarmuka *Home User* 42
3.6.6 Perancangan Antarmuka Data *User* 43
3.6.7 Perancangan Antarmuka *Profile User* 43
3.6.8 Perancangan Antarmuka Enkripsi 44
3.6.9 Perancangan Antarmuka dekripsi 44
3.7 Analisis dan Pengujian 45
3.7.1 Analisis Entropy 45
3.7.2 Analisis Ruang Kunci 45
3.7.3 Analisis Waktu Enkripsi dan Dekripsi 46
3.7.4 Analisis Besar Ukurusan File Hasil Enkripsi 46
3.7.5 Pengujian Hasil Enkripsi 46
3.7.6 Pengujian Hasil Dekripsi 47
BAB IV HASIL DAN PEMBAHASAN 49
4.1 Implementasi Perangkat Lunak 49
4.2 Implementasi Algoritma 49
4.2.1 Algoritma Randomized Text 49
4.2.2 Algoritma Arnold Cat Map 51
4.3 Implementasi Antarmuka 54
4.3.1 Antarmuka *Login* Admin 54
4.3.2 Antarmuka *Home* Admin 55
4.3.3 Antarmuka Data *User* 55
4.3.4 Antarmuka *Logout* 57
4.3.5 Antarmuka *Login User* 58
4.3.6 Antarmuka Daftar *User* 59
4.3.7 Antarmuka *Home User* 60
4.3.8 Antarmuka *Profile User* 60
4.3.9 Antarmuka Enkripsi 62
4.3.10 Antarmuka Dekripsi 63
4.3.11 Antarmuka *Logout* 64
4.4 Skenario Aplikasi FTIE 64
4.5 Analisis Keamanan 66
4.5.1 Analisis Entropy 66
4.5.2 Analisis Ruang Kunci 67
4.5.3 Analisis Waktu Enkripsi 68
4.5.4 Analisis Besar Ukuran File Hasil Enkripsi 68
4.5.5 Pengujian Hasil Enkripsi 69

4.5.6 Pengujian Hasil Dekripsi 71
4.5.7 Pengujian Kunci Dekripsi Salah 73
4.5.8 Perbedaan Penelitian ..... **Error! Bookmark not defined.**
BAB V KESIMPULAN DAN SARAN 76
5.1 Kesimpulan 76
5.2 Saran 76
DAFTAR PUSTAKA 77
LAMPIRAN 79

DAFTAR TABEL

Table 2.1 Review Penelitian 6
Table 3.1 data administrator.....38
Table 3.2 data *user*.....38
Table 3.3 data enkripsi.....39
Table 3.4 data dekripsi.....39
Table 3.5 format analisis *entropy*.....45
Table 3.6 analisis waktu.....46
Table 3.7 analisis *size* hasil enkripsi.....46
Table 3.8 bahas pengujian.....47
Table 3.9 hasil pengujian enkripsi.....47
Table 3.10 bahan pengujian *chiper image*.....47
Table 3.11 hasil dekripsi.....48
Table 4.1 percobaan pertama.....50
Table 4.2 percobaan kedua.....50
Table 4.3 perhitungan MX.....52
Table 4.4 perhitungan MY.....52
Table 4.5 perhitungan MS.....52
Table 4.6 eliminasi MS.....53
Table 4.7 analisis *entropy*.....68
Table 4.8 analisis waktu.....69
Table 4.9 analisis *size* hasil enkripsi.....70
Table 4.10 bahan pengujian.....70
Table 4.11 hasil pengujian enkripsi.....70
Table 4.12 bahan pengujian *chiper image*.....72
Table 4.13 hasil dekripsi.....74
Table 4.14 perbedaan penelitian.....76

DAFTAR GAMBAR

Gambar 2.1 Tabel ASCII.....16
Gamabr 2.2 konsep *Text To Image Encryption*.....18
Gambar 2.3 *skema transmisi Text To Image Encryption*.....18
Gambar 2.4 *Randomized Text encryption flowchart*.....19
Gambar 3.1 *flowchart* metode peneltian.....23
Gambar 3.2 *flowchart* tahapan FTIE dengan menggunakan algoritma *Randomized Text* dan *Arnold Cat Map*.....25
Gambar 3.3 *flowchart* perancangan model enkripsi FTIE.....26
Gambar 3.4 *flowchart* perancangan model dekripsi FTIE.....28
Gambar 3.5 *use case diagra*.....30
Gambar 3.6 diagram activity *login* admin.....31
Gambar 3.7 diagram activity *login user*.....32
Gambar 3.8 diagram activity administrator.....33
Gambar 3.9 diagram activity *user*.....34
Gambar 3.10 diagram activity *user* (halaman enkripsi).....35
Gambar 3.11 diagram activity *user* (halaman dekripsi).....36
Gambar 3.12 *Entity Relationship Diagram* (ERD).....37
Gambar 3.13 perancangan *login* admin.....40
Gambar 3.14 perancangan *login user* 41
Gambar 3.15 perancangan daftar *user*.....41
Gambar 3.16 perancangan antarmuka *home* admin.....42
Gambar 3.17 perancangan antarmuka *home user*.....42
Gambar 3.18 perancangan antarmuka daftar *user*.....43
Gambar 3.19 perancangan antarmuka *profile user*.....43
Gambar 3.20 perancangan antarmuka *enkripsi*.....44
Gambar 3.21 perancangan antarmuka dekripsi.....44
Gambar 4.1 skema pengecekan *Anold Cat Map*.....53
Gambar 4.2 antarmuka *login*.....54
Gambar 4.3 antarmuka input sukses.....55
Gambar 4.4 antarmuka *home* admin.....55
Gambar 4.5 antarmuka data *user*.....55

Gambar 4.6 hapus data *user*.....56
Gambar 4.7 halaman edit *user*.....56
Gambar 4.8 data diganti.....57
Gambar 4.9 lihat file *user* enkripsi.....57
Gambar 4.10 lihat file *user* asli.....57
Gambar 4.11 *logout*.....58
Gambar 4.12 *login user*.....58
Gambar 4.13 *login* sukses.....59
Gambar 4.14 *login* gagal.....59
Gambar 4.15 akun *expired*.....59
Gambar 4.16 daftar *user*.....60
Gambar 4.17 data tersimpan.....60
Gambar 4.18 *home user*.....61
Gambar 4.19 *profile user*.....61
Gambar 4.20 edit email *user*.....62
Gambar 4.21 data diganti.....62
Gambar 4.22 ganti *password*.....62
Gambar 4.23 *password* salah.....63
Gambar 4.24 konfirmasi *password* salah.....63
Gambar 4.25 *password* berhasil diganti.....63
Gambar 4.26 halaman enkripsi.....64
Gambar 4.27 *download file chiper image*.....64
Gambar 4.28 halaman dekripsi.....64
Gambar 4.29 *download* hasil dekripsi.....65
Gambar 4.30 berhasil *logout*.....65
Gambar 4.31 skenarion aplikasi FTIE.....66
Gambar 4.32 pengujian *file.txt*.....75
Gambar 4.33 hasil *file* enkripsi *file.txt*.....75
Gambar 4.34 hasil dekripsi *file.txt* dengan kunci berbeda.....75

BAB I
PENDAHULUAN

5.1 Latar Belakang

Seiring dengan kemajuan teknologi komputer dan dunia digital yang berkembang pesat telah menjadi kebutuhan primer untuk munculnya inovasi aplikasi digital di era yang modern saat ini. Inovasi yang terus bermunculan tersebut akan berdampak negatif pada sistem keamanan dalam pertukaran informasi yang menyebabkan penyadapan data. Dengan adanya penyadapan data keamanan dalam pertukaran informasi menjadi penting, karena terdapat jalur transmisi yang menghubungkan perpindahan suatu informasi data jarak jauh yang berpotensi untuk terjadinya penyadapan dari orang yang tidak berkepentingan. Untuk mengatasi hal tersebut akan di rancang suatu sistem kemanan yang memiliki fungsi melindungi informasi data dari berbagai ancaman.

Keamanan data sangat dibutuhkan untuk menjaga dan melindungi kerahasiaan data. Sudah banyak terjadi pembobolan dan penyadapan informasi data pada aplikasi berbasis web, karena kurangnya keamanan data pada sistem yang telah dibangun. Keamanan informasi data ini harus benar-benar diperhatikan oleh para pengembang aplikasi web agar mempunyai sistem keamanan data yang lebih baik dan tidak gampang diretas oleh kriptanalisis atau pihak yang tidak berkepentingan.

Salah satu teknik pengamanan data adalah menggunakan teknik enkripsi dan dekripsi. Enkripsi dan dekripsi merupakan bidang ilmu kriptografi. (Nurdin Nurdin and Prayitno, 2017) menjelaskan bahwa algoritma kriptografi merupakan suatu bidang pengetahuan yang menggunakan persamaan matematis untuk melakukan proses enkripsi dan dekripsi dengan mengkonversi data ke bentuk kode-kode tertentu sehingga informasi tidak dapat terbaca oleh pihak yang tidak berkepentingan. Selain itu, (Hidayat and Afrianto, 2017) juga menyatakan bahwa algoritma Kriptografi merupakan suatu teknik matematika yang erat kaitanya dengan keamanan data seperti kerahasiaan dan keutuhan data, karena algoritma kriptografi bukan hanya penyembunyian pesan namun sekumpulan teknik yang menyediakan keamanan data.

Perkembangan algoritma kriptografi pada jaman sekarang sudah tidak bisa di anggap aman lagi, oleh karena itu para peneliti sudah mulai bersaing untuk menemukan algoritma-algoritma baru yang lebih aman jika dibandingkan dengan algoritma-algoritma kriptografi sebelumnya. (Shanthi and Palanisamy, 2014) telah menemukan teknik enkripsi yang baru yaitu

Text To Image Encryption (TTIE). TTIE adalah teknik enkripsi yang mentranformasi sebuah teks menjadi sebuah gambar. Secara teknis algoritma enkripsi ini memiliki keamanan terhadap plaintext itu sendiri dan keamanan terhadap *text* yang dihasilkan dalam adalah bentuk gambar. Hal tersebut akan membuat seorang kriptanalisis bingung untuk menebak pesan *plaintext*. Jika kriptanalisis tersebut menjadikan informasi asli dari gambar tersebut adalah gambar juga, maka kriptanalisis tersebut tidak dapat memecahkan dan menemukan informasi asli dari gambar tersebut.

Selain itu, untuk menghasilkan informasi yang lebih aman dalam pembuatan aplikasi ini akan digunakan penggabungan algoritma antara algoritma *Randomized Text* dengan *Arnold Cat Map* (ACM). Tujuan memilih algoritma *Randomized Text* karena algoritma ini merupakan salah satu dari jenis *randomized encryption*. (Maurer, 1992) menyatakan bahwa *chipertext* dapat disempurnakan jika memliki kunci rahasia yang sama besar dengan *plaintextnya*. Akan tetapi penggunaan algoritma *Randomized encryption* masih memiliki celah keamanan yaitu *chipertext* yang dihasilkan masih memiliki pola, oleh karena itu dibutuhkan salah satu algoritma transformasi untuk menghilangkan pola tersebut. Tujuan memilih algoritma *Arnold Cat Map* (ACM) dikarenakan algoritma ACM menurut (Ronsen, Halim and Syahputra, 2014) memiliki tingkat keamanan yang rendah dan transformasi yang sederhana, tetapi sangat untuk mengacak posisi *chipertext* dari *randomized text*.

Oleh karena itu, dibutuhkan sebuah aplikasi keamanan informasi yang tidak dapat dibaca oleh kriptanalisis. Salah satunya adalah dengan teknik enkripsi *File To Image Encryption* (FTIE) menggunakan algoritma *Randomized Text* dan algoritma *Arnold Cat Map* (ACM). Dari uraian diatas maka akan dibuat sebuah aplikasi *File To Image Encryption* (FTIE) dengan menggunakan algoritma *Randomized Text* dan *Arnold Cat Map* (ACM) berbasis website untuk keamanan data digital yang nantinya aplikasi berbasis web tersebut akan digunakan sebagai pengamanan sebuah informasi data.

5.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka dapat disimpulakan rumusan masalah tugas akhir ini yaitu,

1. Bagaimana membangun aplikasi *File To Image Encryption* (FTIE) menggunakan algoritma *Randomized Tex*t dan *Arnold Cat Map* (ACM) berbasis website untuk keamanan data digital untuk mengamankan sebuah informasi data?

2. Bagaimana membangun teknik enkripsi *File To Image Encryption* (FTIE) dengan cara melakukan penggabungan antara algoritma *Randomized Text* dan *Arnold Cat Map* (ACM) untuk menghasilkan *chipertext* yang memiliki tingkat keamanan yang tinggi pada aplikasi berbasis web ini?

5.3 Batasan Masalah

Agar penulisan laporan tugas akhir ini tidak menyimpang dan mengambang dari tujuan yang semula direncanakan sehingga memper mudah mendapatkan data dan informasi yang diperlukan, maka akan dilakukan menetapkan batasan-batasan sebagai berikut:

1. Pembahasan hanya mencakup aplikasi berbasis website dengan teknik enkripsi *File To Image Encryption* (FTIE) menggunakan algoritma *Randomized Text* dan *Arnold Cat Map* (ACM).
2. Pembangunan aplikasi berbasis web ini menggunakan enkripsi *File To Image Encryption* (FTIE) dengan algoritma *Randomized Text* dan *Arnold Cat Map* (ACM).
3. Implementasi hanya menggunakan enkripsi *File To Image Encryption* (FTIE) menggunakan algoritma *Randomized Text* dan *Arnold Cat Map* (ACM).
4. Type data yang digunakan untuk uji coba enkripsi ini adalah tipe data dokumen digital (*docx, pdf, mp4, rar, jpg, pptx*).
5. Implementasi pemrograman menggunakan bahasa *Hypertext Prepocessor* (PHP), *Html* dan *Javascript*.

5.4 Tujuan Penelitian

Sejalan dengan rumusan masalah di atas, laporan tugas akhir ini disusun dengan tujuan sebagai berikut:

1. Tujuan penelitian ini adalah menghasilkan aplikasi *File To Image Encryption* (FTIE) menggunakan algoritma *Randomized Tex*t dan *Arnold Cat Map* (ACM) berbasis Website untuk keamanan data digital.
2. Melakukan pengujian terhadap teknik FTIE terhadap berbagai analisis yang meliputi analisis entropy, analisis ruang kunci, analisis waktu enkripsi, analisis besar ukurusan hasil enkripsi, pengujian hasil enkripsi dan pengujian hasil dekripsi.

5.5 Manfaat Penelitian

Adapun beberapa manfaat yang di harapakan terutama bagi mahasiswa yang telah membuat laporan tugas akhir ini.

1. Dapat menerapkan ilmu kripografi enkripsi terutama dengan algoritma *Randomized Text* dan *Arnold Cat Map* (ACM).
2. Melindungi kerahasiaan suatu informasi data dari pihak yang tidak diharapkan.
3. Menghasilkan aplikasi berbasis web dengan teknik enkripsi *File To Image Encryption* (FTIE) menggunakan algoritma *Randomized Text* dan *Arnold Cat Map* (ACM).
4. Sebagai aplikasi penunjang untuk keamanan informasi data dan *file*.
5. Untuk menjadi bahan acuan jika melakukan pengembangan aplikasi berbasis website.
6. Menambah wawasan dan kualitas keilmuan baik dalam hal teori maupun praktek.

5.6 Metodologi Penelitian

Adapun beberapa tahap metode penelitian yang akan digunakan, untuk menyelesaikan laporan tugas akhir ini diantaranya:

1. Studi *Literature* keamanan yang di ambil dari berbagai sumber seperti buku, jurnal, serta website-website yang berkaitan dengan teknik-teknik algoritma kriptografi khususnya untuk menentukan teknik dan algoritma yang cocok untuk konsep keamanan pada saat transmisi data.
2. Membangun rancangan dari teknik FTIE dengan menggunakan algoritma *Randomized Text* dan *Arnold Cat Map* (ACM).
3. Implementasi sistem dengan menggunakan bahasa pemrograman PHP, Html dan Javascript yang nantinya akan menghasilkan suatu aplikasi enkripsi FTIE berbasis web.
4. Kebutuhan terhadap perangkat keras dan perangkat lunak untuk mendukung sistem pada aplikasi berbasis web yang akan dibuat.
5. penyusunan laporan tugas akhir hasi dari penelitian yang telah dilakukan.

5.7 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini disusun berdasarkan format laporan standar baku dari "Panduan Tugas Akhir" Jurusan Teknik In*f*ormatika Fakultas Teknologi Industri Universitas Islam Indonesia yang terdiri dari lima (5) bagian pokok yaitu sebagai berikut :

BAB I PENDAHULUAN

Bab I ini merupakan pengantar terhadap permasalahan yang akan dibahas. Di dalamnya terdapat uraian-uraian tentang gambaran penelitian yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab II ini merupakan penjelasan teori yang akan digunakan untuk memecahkan masalah dalam peneltian ini.

BAB III METODOLOGI

Pada bab III ini akan di bahas tentang langkah-langkah penlitian. Didalamnya terdapat beberapa uraian yang akan digunakan dalam penelitian meliputi kebutuhan sistem, perancangan antar muka, dan perancangan sistem beserta *flowchart*.

BAB IV HASIL DAN PEMBAHASAN

Pada bab IV merupakan hasil akhir dari aplikasi yang telah dibuat, berisi tentang implementasi dari algoritma yang digunakan ke dalam aplikasi serta pengujian hasil aplikasi yang telah dibuat.

BAB V SARAN DAN KESIMPULAN

Pada bab V ini memuat kesimpulan-kesimpulan dari hasil penelitian dan saran-saran yang perlu diperhatikan berdasar keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama melakukan penelitian dan juga rekomendasi yang dibuat untuk pengembangan penelitian selanjutnya.

DAFTAR PUSTAKA

LAMPIRAN

BAB II
LANDASAN TEORI

2.1 Literature *review*

Table 2.1 Review Penelitian

NO	JUDUL/METODE	PENELITI	PERSAMAAN/PERBEDAAN
1.	Enkripsi Citra Digital menggunakan *Arnold Cat Map* dan *Nonlinear Chaotic Algorithm*	Ronsen Purba, Arwin Halim, Indra Syahputra, 2014	Sama-sama menggunakan algoritma *Arnold Cat Map*, pada penelitian ini menggunakan penggabungan algoritma ACM dan NCA Sedangkan FTIE melakukan modifikasi penggabungan ACM dan *Randomized Text.*
2.	Penyisipan pesan pada Gambar menggunakan Algoritma *Arnold Cat Map* (ACM), *Least Significant Bit* (LSB), dan *Scale Invariant Feature Transform* (SIFT)	Cahya Nurani Indah	Sama-sama memanfaatkan *Arnold Cat Map*, pada penelitian ini hanya menyisipkan gambar menggunakan algortima *Arnold Cat Map*.
3.	Melakukan Enkripsi Teks ke dalam bentuk Gambar	Ahmad Abusukhon, 2012	TTIE dapat merubah teks ke dalam bentuk gambar, sedangkan FTIE dapat merubah kedalam bentuk gambar.
4.	Melakukan Enkripsi terhadap Teks menggunakan Algoritma *Randomized Text* dimana setiap *Plaintext* akan menghasilkan 2 kali lipat	Jamshed Memon, 201	*Randomized Text* merupakan teknik enkripsi hanya untuk teks, sedangkan FTIE dapat Melakukan enkripsi terhadap semua jenis *file*.

NO	JUDUL/METODE	PENELITI	PERSAMAAN/PERBEDAAN
5.	Melakukan Enkripsi Teks dalam bentuk Gambar dan menggabungkan dengan Algoritma *AES Rijendael*	Shanthi, 2014	algoritma *AES Rijndael* untuk menghasilkan *chiper image* yang lebih kuat, FTIE memanfaatkan algoritma *Randomized Text* dan *Arnold Cat Map* untuk menghasilkan *chiper image* yang lebih kuat.
6.	Penggabungan Algoritma *Chaos* dan *Rivers Shamir Adleman* (RSA) untuk peningkatan keamanan Citra	Pahrul Irfan, 2015	Sama-sama memanfaatkan *Arnold Cat Map* untuk transformasi. *Arnold Cat Map* pada penelitian ini digabungkan dengan algoritma RSA untuk meningkatkan keamanan citra.
7.	Algoritma Enkripsi Selektif Citra Digital dalam Ranah Frekuensi berbasis Permutasi *Chaos*	Rinaldi Munir, 2012	Sama-sama menggunakan *Arnold Cat Map*, pada penilitan ini hanya mempresentasikan algoritma enkripsi citra digital dalam ranah frekuensi.
8.	Aplikasi Steganografi untuk Penyisipan Data Teks ke dalam Citra Digital	Temmy Maradilla,	Pada peneltian ini menggunakan algoritma Steganografi sedangkan yang telah di teliti saat ini menggunakan algoritma Kriptografi.
9.	Pengembangan Aplikasi Pengamanan Dokumen Digital memanfaatkan Algoritma *Advanve Encryption* Standar, RSA *Digital Signature* dan *Invisible Watermarking*	Aji setiyo Sukarno	Sama-sama membuat aplikasi untuk keamanan digital, pada penelitian ini menggunakan algoritma AES dan RSA sedangakan yang telah di teliti saat ini menggunakan algoritma ACM dan *Randomized Text*.
10.	Enkripsi Gambar menggunakan 2 Algoritma *Chaos* yaitu ACM dan *Henon Map*	Agyan Kumar Prust, 2013	Sama-sama memanfaatkan *Arnold Cat Map*, pada penelitian ini menggabungkan *Arnold Cat Map* dengan algoritma *chaos* yang lain

NO	JUDUL/METODE	PENELITI	PERSAMAAN/PERBEDAAN
			yaitu *Henon Map*. Sedangkan yang telah di teliti penggabungan antar ACM dan *Randomized Text*.
11.	Usulan Peneltian	**aplikasi *File To Image Encryption* (FTIE) menggunakan algoritma *Randomized Tex*t dan *Arnold Cat Map* (ACM) berbasis Website untuk keamanan data digital**	Melakukan enkripsi terhadap semua jenis ke dalam bentuk gambar menggunakan teknik FTIE dengan kombinasi algoritma *Randomized Text* dan *Arnold Cat Map* untuk keamanan data digital, hasil dari penelitian ini berupa sebuah aplikasi enkripsi FTIE berbasis web.

2.2 Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani, memiliki arti dari dua suku kata yaitu *kripto* dan *graphia*. *Kripto* artinya menyembunyikan, sedangkan *graphia* artinya tulisan. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, integritas data, keabsahan data, serta autentikasi data .Tetapi tidak semua aspek keamanan informasi dapat diselesaikan dengan kriptografi.

2.2.1 Definisi kriptografi

Menurut (Abidin, Hardianti and Setiani, 2016) Dalam penerapannya, kriptografi merupakan suatu metode enkripsi atau penyandian data yang hanya diketahui oleh suatu kelompok pengguna tertentu. Metoda ini telah dikenal pada masa kekaisaran Romawi Kuno. Pada waktu itu Julius Caesar tidak menginginkan berita atau pesan yang dibawa oleh kurir-kurirnya jatuh kepada pihak lawan. Oleh karena itu, beliau menggunakan sistem substitusi sederhana, yang kini disebut dengan *Caesar Cipher*. Pada prinsipnya kriptografi memiliki 4 komponen utama, yaitu: *Plaintext* (pesan yang dapat dibaca, *Ciphertext* (pesan acak yang tidak dapat dibaca), *Key* (kunci untuk melakukan teknik kriptografi), *Algorithm* (metode untuk melakukan enkrispi dan dekripsi).

2.2.2 Tujuan kriptografi

(Sari, 2016) menjelaskan Tujuan kriptografi adalah untuk mengamankan suatu informasi data dan menjaga kebutuhan data. Salah satu algoritma yang dapat mengamankan informasi data adalah algoritma *Caesar Chiper*.

(Harahap, 2016) kriptografi memiliki 4 tujuan yang termasuk ke dalam aspek keamanan in*f*ormasi data, yaitu:

1. Kerahasian Data (*Confidentiality*) adalah layanan yang di gunakan untuk Menjaga informasi data agar tetap rahasia dari pihak-pihak tidak berwenang yang mungkin mencoba membaca data tersebut. Ada beberapa pendekatan untuk menjaga suatu informasi data, dari pengamana secara fisik dan penggunaan algoritma matematika agar in*form*asi-in*form*asi data tidak dapat di pahami.
2. Integritas Data (*Integrity*) adalah layanan penjagaan untuk memastikan pengiriman informasi data masih tetap asli dengan informasi data yang diterima tanpa ada perubahan atau modifikasi terhadap data tersebut.
3. Autentikasi (*Authentication*) adalah layanan yang berhubungan dengan identifikasi dan untuk Memastikan bahwa pengirim dan penerima benar-benar terjamin keasliannya. Dua pihak yang berkomunikasi harus saling mengetahui satu dengan lainnya.
4. Non-Repudiasi (*Non-Repudiation*) adalah layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan, yaitu pengirim pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan.

2.2.3 Terminologi kriptografi

a. Pesan, *plaintext*, dan *chipertext*

Pesan (*message*) adalah suatu pemberitahuan, kata, atau komunikasi baik secara lisan atau tertulis, yang dapat dikirimkan dari satu orang ke orang lain. Pesan mrupakan inti dari proses-proses komunikasi yang terjalin. Nama lain untuk pesan adalah (*plaintext*). *Chiphertext* merupakan sebuah pesan yang tersandi, dengan adanya *chipertext* akan membuat sebuah pesan tidak dapat di pahami dan harus dapat ditransformasikan lagi menjadi *plaintext* agar pesan yang di terima dapat dibaca.

b. Enkripsi dan dekripsi

Enkripsi merupakan proses yang di lakukan untuk mengamankan sebuah pesan

(*plaintext*) menjadi pesan yang tersembunyi (*chipertext*). Dekripsi merupakan proses mengubah *chipertext* menjadi.

c. Pengirim dan penerima

Pengirim adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima adalah entitas yang menerima pesan.

Entitas di sini dapat berupa orang, mesin (komputer), dan sebagainya.

d. *Chiper* dan kunci

Algoritma kriptografi disebut juga *cipher* yaitu aturan untuk *enciphering* (enkripsi) dan *deciphering* (dekripsi), atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yaitu himpunan yang berisi elemen-elemen *plaintext* dan himpunan yang berisi *ciphertext*. Enkripsi dan dekripsi adalah fungsi yang memetakan elemen-elemen antara kedua himpunan tersebut.

e. Penyadap

Penyadap adalah orang yang berusaha untuk menangkap pesan selama ditransmisikan. Tujuan penyadap adalah untuk mendapatkan informasi sebanyak-banyaknya mengenai sistem kriptografi yang akan digunakan dalam berkomunikasi dengan maksud untuk memecahkan *chipertext*.

f. Sistem kriptografi

Sistem kriptografi adalah kumpulan yang terdiri dari algoritma kriptografi, semua *plaintext* dan *chipertext* yang mungkin dan kunci. Didalam kriptografi *chiper* hanyalah satu komponen saja.

g. Kriptanalisis dan kriptologi

Dengan adanya perkembangan kriptografi yang sangat pesat, akan melahirkan bidang yang berlawanan yaitu kriptanalis. Kriptanalisis (*cryptanalysis*) adalah ilmu dan seni untuk memecahkan *chipertext* menjadi *plaintext* tanpa mengetahui kunci yang digunakan pelakunya disebut kriptanalis. Kriptologi adalah studi mengenai kriptografi dan kriptanalisis.

2.2.4 Kebutuhan akan kriptografi

“Kriptografi diperlukan karena pada dasarnya informasi sangat peting bagi segala aspek, tuntutan kemanan informasi berubah dari waktu ke waktu. Perubahan tuntutan ini terjadi karena transformasi atau penggunaan perlengkapan kebutuhan utama untuk pertukaran informasi, dari mulai cara tradisional (fisik) yang membutuhkan mekansime pengarsipan atau administrasi

secara fisik dan membutuhkan ruang yang lebih besar, menggunakan otomatisasi komputer personal, sampai transfer informasi melalui penggunaan jaringan komputer, baik intranet maupun internet yang sekarang menjadi tren dan kebutuhan" (Santi, 2010) **.**Berikut ini adalah beberapa kondisi dimana kriptografi itu di butuhkan menurut (Kromodimoeljo, 2009)**:**

A. In*form*asi sensitif

Setiap orang, perusahaan, institusi dan instansi pemerintahan mempunyai informasi sensitif yang sebaiknya tidak jatuh ketangan orang yang tidak berhak untuk mendapatkannya. Contoh dari informasi sensitif seseorang secara pribadi antara lain:

1. Informasi kesehatan pribadi.
2. Informasi keuangan pribadi dan pola belanja.

Dari sisi kepentingan umum, memang ada informasi kesehatan pribadi yang dibutuhkan oleh instansi tertentu, misalnya untuk pencegahan penularan pe nyakit. Namun dari sisi pribadi, informasi kesehatan adalah sesuatu yang sensitif. Tentunya ada informasi yang perlu diketahui oleh dokter dan rumah sakit, misalnya kondisi jantung atau masalah alergi terhadap obat-obat tertentu. Tetapi jika informasi kesehatan pribadi disebar-luaskan, ada kemungkinan timbul tindakan atau reaksi diskriminatif terhadap yang bersangkutan. Informasi kesehatan pribadi juga bisa menjadi bahan untuk gosip. Demikian juga dengan informasi keuangan, instansi tertentu seperti instansi perpajakan mungkin perlu mengetahui informasi keuangan pribadi. Namun elemen-elemen kriminal dapat menggunakan informasi keuangan dan pola belanja seseorang untuk melakukan perbuatan kriminal.

Suatu perusahaan juga mempunyai informasi sensitif atau rahasia yang sebaiknya tidak disebar-luaskan. Contoh informasi sensitif yang perlu dirahasiakan oleh suatu perusahaan termasuk:

1. Cara pembuatan suatu produk.
2. Rencana strategis yang rinci.

Meskipun dari sisi pemegang saham, perusahaan diinginkan agar transparan, tentunya ada rahasia perusahaan seperti resep pembuatan suatu produk, yang jika jatuh ke perusahaan lain, akan menguntungkan perusahaan lain dan merugikan perusahaan pemilik resep. Kadang, rencana strategis yang rinci juga perlu dirahasiakan. Untuk suatu institusi, contoh informasi sensitif termasuk:

1. Alamat, nomor telepon dan email anggota institusi.
2. Nilai akademis.

Kerap seorang anggota institusi tidak ingin informasi pribadinya diberikan kepada pihak ketiga. Informasi pribadinya bisa berupa alamat, nomor telpon dan email. Suatu institusi pendidikan seperti universitas juga menyimpan informasi sensitif. Sebagai contoh, nilai akademis mahasiswa sebaiknya tidak bisa begitu saja diberikan ke pihak ketiga tanpa persetujuan mahasiswa. Suatu instansi pemerintahan juga memiliki informasi yang sensitif, sebagai contoh antara lain:

1. Informasi pribadi pembayar pajak.
2. Data mengenai persenjataan militer.

Meskipun publik menginginkan instansi pemerintahan yang transparan, ada informasi tertentu yang sensitif dan perlu dirahasiakan. Sebagai contoh, instansi perpajakan sebaiknya merahasiakan informasi pribadi seorang pembayar pajak, kecuali jika informasi tersebut diperlukan untuk kepentingan hukum. Contoh lain adalah instansi militer dimana informasi tertentu mengenai persenjataan atau pergerakan pasukan dimasa perang perlu dirahasiakan.

Seseorang atau suatu organisasi tentunya bertanggung jawab atas kerahasiaan informasi sensitif yang dimilikinya, dan kriptografi dapat membantu menjaga kerahasiaan informasi sensitif yang disimpan secara elektronik. Berbagai langkah menggunakan kriptografi dapat diambil untuk menjaga kerahasiaan informasi sensitif yang disimpan secara elektronik termasuk:

1. *Access* control terhadap informasi.
2. Enkripsi data dalam transit.
3. Enkripsi data dalam media penyimpanan.

Di jaman sekarang dimana komputer saling terhubung, *access* control bukan semata kontrol secara fisik, namun juga harus meliputi kontrol *access* secara online. Ini biasanya dilakukan menggunakan *password* atau *passphrase* dan diamankan menggunakan kriptografi sebagai berikut:

1. Menggunakan *secure hashing* untuk penyimpanan di server, dan
2. Menggunakan enkripsi untuk transmisi.

Data sensitif dalam transit juga perlu diamankan menggunakan enkripsi. Definisi data dalam transit juga mungkin perlu diperluas, bukan saja data yang sedang ditransmisi melalui jalur komunikasi, tetapi meliputi juga data dalam *notebook* computer dan *flash disk* yang keduanya dapat saja dicuri atau hilang. Dalam prakteknya, pengamanan data dalam transit dapat dilakukan dengan:

1. Menggunakan *secure session* seperti SSL/TLS, SSH dan *IPsec*.

2. Mengenkripsi atau *system* dalam *flash disk* dan *hard drive notebook*.

Untuk tingkat pengamanan yang lebih tinggi lagi, bukan hanya data sensitif yang disimpan dalam *flashdisk* dan *hard drive notebook* saja yang perlu dienkripsi, tetapi juga data sensitif yang disimpan di media penyimpanan lain seperti *hard drive* untuk desktop komputer. Ini terutama jika media dapat diakses oleh orang yang tidak diinginkan mengakses data sensitif tersebut.

B. Mencegah penyamaran

Di era online dimana komunikasi dilakukan melalui jaringan internet, identitas orang atau komputer yang hendak berkomunikasi dengan kita kadang perlu dipastikan. Sebagai contoh, jika akses ke suatu sistem informasi hanya diperbolehkan untuk pengguna yang telah terdaftar, maka sistem harus memastikan bahwa seorang yang ingin mengakses sistem adalah pengguna yang telah terdaftar. Berbagai contoh situasi dimana identitas perlu dipastikan antara lain:

1. Seorang yang mengaku pengguna dan ingin mengakses sistem memang benar pengguna yang telah terdaftar.
2. *Website* yang sedang kita kunjungi dan kepada siapa kita hendak mengirim nomor kartu kredit memang benar *website* yang kita kehendaki.
3. Perangkat yang sedang mencoba untuk bergabung dalam jaringan komunikasi lokal *nirkabel* memang perangkat yang diperbolehkan untuk bergabung.

Memastikan identitas pengguna adalah bagian dari *access control*. Memastikan identitas kerap disebut *authentication*. Trend saat ini adalah menggunakan *multiplefactor authentication* yaitu menggunakan beberapa atribut unik pengguna sistem untuk identifikasi. Sifat atribut dapat berupa:

1. Atribut langsung pengguna (*what you are*) misalnya sidik jari.
2. Benda yang dimiliki pengguna (*what you have*) misalnya suatu token.

Apa yang diketahui pengguna (*what you know*) misalnya *password* atau kunci privat.

Enkripsi berperan dalam mengamankan komunikasi mengenai apa yang diketahui pengguna, baik *password* maupun *authentication* menggunakan *public key cryptography*. Tentunya *access* control bukan hanya memastikan identitas, tetapi juga meliputi kontrol terhadap apa yang dapat diakses oleh pengguna sistem. Kerap apa yang dapat diakses oleh satu pengguna berbeda dengan apa yang dapat diakses pengguna lain. Kontrol terhadap apa yang

dapat diakses berbagai pengguna sistem dapat diamankan menggunakan kriptografi misalnya dengan mengenkripsi *file*.

C. Mencegah penyadapan

Jika mendengar kata "penyadapan" maka yang terbayang di pikiran pembaca mungkin penyadapan oleh agen asing atau penyadapan oleh penegak hukum. Namun dewasa ini penyadapan komunikasi dapat dilakukan oleh siapa saja, termasu kelemen kriminal, dengan peralatan yang relatif murah. Secara umum, komunikasi *nirkabel* rentan terhadap penyadapan karena penyadap tidak perlu akses fisik ke kabel komunikasi. Berikut adalah beberapa macam penyadapan yang sebagian diantaranya dapat dicegah menggunakan enkripsi:

1. Penyadapan komunikasi nirkabel.
2. Penyadapan komunikasi dengan kabel (tembaga maupun optik).
3. Penyadapan radiasi elektromagnetik.
4. Penyadapan akustik.

Dua standard komunikasi lokal nirkabel yang populer adalah *Wi-Fi* (IEEE 802.11) dan *Bluetooth*. Kedua protokol sebenarnya sudah menyediakan enkripsi, *Wi-Fi* melalui *WEP* dan *WPA*, dan *Bluetooth* melalui *security* mode 2, 3 dan 4 dan *encryption* mode 2 dan 3. Untuk *WiFi*, sebaiknya WPA digunakan jika ada karena WEP terlalu mudah untuk dipecahkan. Akan tetapi ini tidak cukup jika kunci WPA atau WEP yang digunakan adalah kunci bersama, jadi sebaiknya gunakan enkripsi tambahan untuk data yang sensitif, contohnya menggunakan SSH. Jika pembaca ingin rekomendasi yang lebih rinci mengenai pengamanan Wi-Fi yang sudah mendukung WPA.

Untuk *Wi-Fi* yang belum mendukung WPA, silahkan membaca. *Bluetooth* lebih rentan terhadap penyadapan karena limitasi perangkat, baik limitasi fitur keamanan yang ada dalam perangkat, maupun limitasi yang diakibatkan kesalahan implementasi fitur keamanan dalam perangkat. Jadi untuk *Bluetooth*, sebaiknya gunakan enkripsi tambahan untuk data sensitif. Untuk komunikasi data sensitif melalui jaringan selular, jelas enkripsi tambahan diperlukan. Hampir semua komunikasi melalui kabel menggunakan protokol internet yaitu TCP/IP. Sebetulnya, pada layer IP sudah tersedia pengamanan menggunakan enkripsi yaitu IPsec. Namun untuk orang awam, *deployment* IPsec agak lebih sulit dibandingkan pengamanan sesi pada layer atas seperti SSL/TLS dan SSH. Untuk mencegah informasi bocor lewat radiasi elektromagnetik dari perangkat, biasanya perangkat dilindungi dengan *Faraday's cage*, yaitu "sangkar" dari bahan logam.

Tentunya enkripsi sebaiknya tetap digunakan untuk komunikasi data sensitif dari/ke perangkat. Penyadapan akustik bukan hanya penyadapan percakapan. Suara dari penggunaan keyboard dapat disadap, dan dengan analisis statistik frekuensi penggunaan setiap kunci di keyboard, apa yang akan diketik di keyboard dapat diketahui.

Kemungkinan penyadapan akustik mungkin tidak terlalu besar dibandingkan kemungkinan adanya *trojan* yang melakukan *keyboard logging*. Namun jika pembaca paranoid atau sedang menginput suatu rahasia negara yang sangat sensitif, maka pembaca sebaiknya menggunakan *keyboard* dalam ruangan kedap suara. Enkripsi secara tradisional tidak akan membantu untuk masalah ini. Yang mungkin dapat dilakukan adalah memancarkan juga secara acak bunyi berbagai kunci bersamaan dengan penggunaan *keyboard*.

Jadi bunyi kunci yang ditekan bercampur dengan bunyi kunci secara acak. Konsep ini mirip dengan *steganography*, yaitu penyembunyian pesan dalam sesuatu yang lebih besar (contohnya dalam gambar atau audio). Bedanya, dalam *steganography*, orang yang kepada siapa pesan ditujukan, diharapkan dapat membaca pesan yang terpendam.

2.2.5 Jenis-jenis serangan algoritma kriptografi

Yang dimaksud dengan serangan kriptografi adalah setiap usaha atau percobaan yang di lakukan oleh kriptanalisi untuk menemukan kunci dan mengungkap *plaintext*. Di bawah ini akan dijelaskan macam-macam Serangan terhadap pesan yang sudah dienkripsi, berdasarkan ketersediaan data yang ada, dan tingkat kesulitannya bagi penyerang, dimulai dari yang paling sulit adalah Menurut (Naila Fithria) :

1. *Ciphertext only attack*, penyerang hanya mendapatkan *ciphertext* dari sejumlah pesan yang seluruhnya telah dienkripsi menggunakan algoritma yang sama. Sehingga, metode yang digunakan untuk memecahkannya adalah *exhaustive key search*, yaitu mencoba semua kemungkinan yang ada untuk menemukan kunci.
2. *Known plaintext attack*, dimana penyerang selain mendapatkan sandi, juga mendapatkan pesan asli. Terkadang disebut pula *clear-text attack*.
3. Choosen *plaintext attack*, sama dengan *known plaintext attack*, namun penyerang bahkan dapat memilih penggalan mana dari pesan asli yang akan disandikan. Serangan jenis ini lebih hebat daripada *known-plaintext attack*, karena kriptoanalis dapat memilih plainteks tertentu untuk dienkripsikan, yaitu plainteks-plainteks yang lebih mengarahkan penemuan kunci.

4. *Chosen-ciphertext attack*. Pada tipe ini, kriptoanalis dapat memilih cipherteks yang berbeda untuk didekripsi dan memiliki akses atas *plaintext* yang didekripsi.
5. *Chosen-key attack*. Kriptoanalis pada tipe penyerangan ini memiliki pengetahuan tentang hubungan antara kunci-kunci yang berbeda dan memilih kunci yang tepat untuk mendekripsi pesan.
6. *Rubber-hose cryptanalysis*. Pada tipe penyerangan ini, kriptoanalis mengancam, menyiksa, memeras, memaksa, atau bahkan menyogok seseorang hingga mereka memberikan kuncinya. Ini adalah cara yang paling ampuh untuk mendapatkan kunci.
7. *Adaptive – chosen – plaintext attack*. Penyerangan tipe ini merupakan suatu kasus khusus *chosen-plaintext attack*. Kriptoanalis tidak hanya dapat memilih plainteks yang dienkripsi, ia pun memiliki kemampuan untuk memodifikasi pilihan berdasarkan hasil enkripsi sebelumnya. Dalam *chosen-plaintext attack*, kriptoanalis mungkin hanya dapat memiliki plainteks dalam suatu blok besar untuk dienkripsi; dalam *adaptive-chosen-plaintext attack* ini iadapat memilih blok plainteks yang lebih kecil dan kemudian memilih yang lain berdasarkan hasil yang pertama, proses ini dapat dilakukannya terus menerus hingga ia dapat memperoleh seluruh in*form*asi.

2.3 Sistem *ASCII*

(sumber : www.asciitable.com)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Gambar 2.1 Tabel ASCII

Kode ASCII (*American Standard Codes for International Interchange*) adalah kumpulan kode-kode yang dipergunakan untuk mempermudah interaksi antara *user* dan komputer.

Kode Standar Amerika untuk Pertukaran In*form*asi atau ASCII (*American Standard Code for Information Interchange*) merupakan suatu standar internasional dalam kode huruf dan simbol seperti *Hex* dan *Unicode* tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks.

Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 8 bit. Dimulai dari 00000000 hingga 11111111. Total kombinasi yang dihasilkan sebanyak 256, dimulai dari kode 0 hingga 255 dalam sistem bilangan Desimal.

ASCII Character Set adalah Sebuah standard kode 7 bit yang menggambarkan karakter dari ASCII dengan menggunakan nilai biner. Jangkauan nilai kode ini adalah dari 0-127. Kebanyakan dari Komputer Pribadi (PC) menggunakan perluasan dari kode ASCII berbasis 8 bit, sehingga didapatkan 128 karakter ekstra, yang digunakan sebagai simbol khusus, karakter khusus, dan simbol grafis.

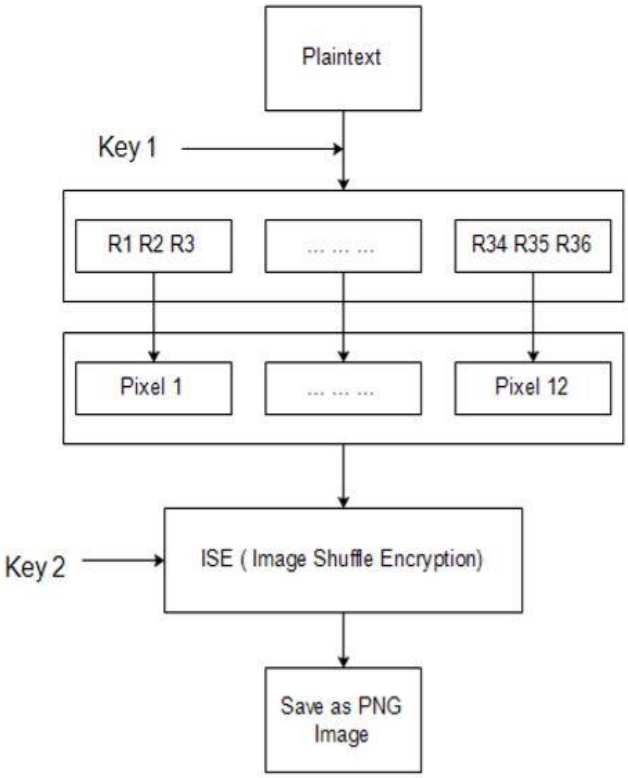
2.4 Text to *image encryption* (TTIE)

(Suprianto, Prayudi and Sugiantoro, 2017) menjelaskan Ahmad Abusukhon pertama kali menemukan teknik enkripsi yang disebut dengan *Text to image encryption*, teknik ini adalah teknik yang merubah sebuah *plaintext* menjadi sebuah gambar, ada 2 tahap yang dilakukan pada TTIE yaitu tahap TTIE itu sendiri dan tahap *Image Shuffle Encryption* (ISE). Pada tahap TTIE teks biasa ditran*form*asikan (dienkripsi) kedalam.

sebuah gambar. Pada tahap ini masing-masing karakter dari *plaintext* disimpan kedalam sebuah *array*, satu karakter dari *array* ini akan mewakili 1 *pixel* dari gambar, dari satu *pixel* gambar ada 3 bilang bulat dengan rentan nilai 0 sampai 255, yang dimana dalam salah satu bilangan tersebut terdapat 1 bilangan yang merupakan karakter dari *plaintext*.

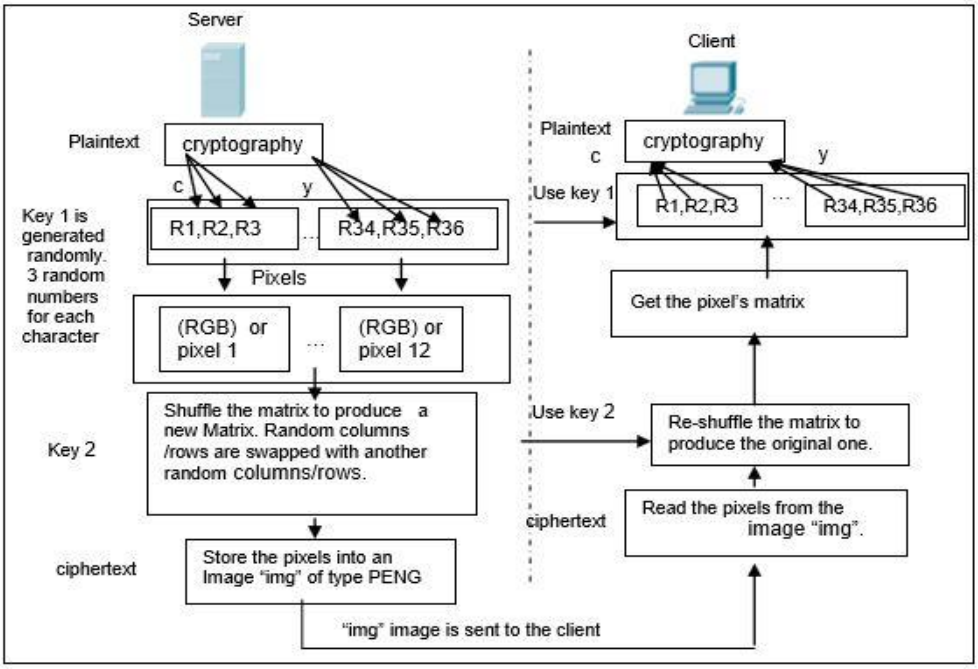
Hal tersebut akan mempersulit seorang kriptanalis untuk menentukan mana yang merupakan karakter asli dari *plaintext*. Pada tahap ISE, array yang dihasilkan pada tahap TTIE tadi kemudian dilakukan pengacakan posisi yang meliputi pengacakan baris dan pengacakan pada kolom.

Hal tersebut akan mempersulit seorang kriptanalis untuk menentukan posisi asli dari sebuah *pixel*. Konsep dari TTIE ditunjukkan pada Gambar 2.5 dibawah ini:



Gambar 2.2 konsep *Text To Image Encryption*

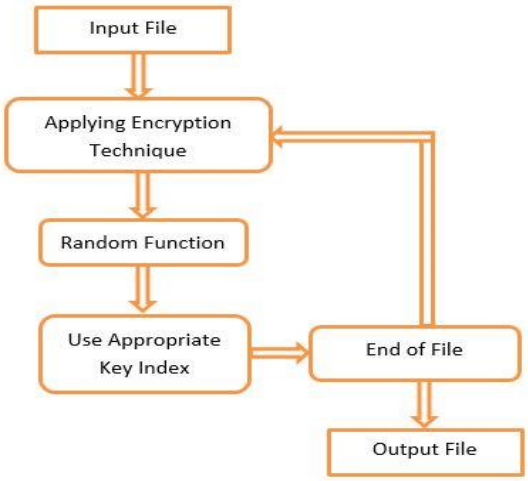
Kemudian untuk skema transmisi data dari server ke klien menggunakan TTIE ditunjukkan oleh Gambar 2.6 berikut ini:



Gambar 2.3 *skema transmisi Text To Image Encryption*

2.5 Randomized Text

Randomized text adalah salah satu teknik enkripsi yang algoritmanya bersifat dinamis artinya algoritma ini selalu menghasilkan pengacakan walaupun dari *plaintext* yang sama dengan kunci yang sama pula. *Randomized text* adalah algoritma kriptografi yang di temukan oleh (Munir, 2012b).



Gambar 2.4 *Randomized Text encryption flowchart*

Pada gambar di atas adalah flowchat dari teknik enkripsi *randomized text*. *Flowchart* tersebut terdiri dari input *file*, *applying encryption technique*, random *function*, use *appropriate key index*, *end of file*, dan *output file*. *Randomized* memliki persamaan enkripsi yang sederhana yaitu:

$$C1 = K + 2P + R$$

$$C2 = 2K + P + R \quad (2.1)$$

Dimana :

K = Kunci

P = *Plaintext*

R = Nilai Random

C = *Chipertext*

Randomized Text melakukan enkripsi per-karakter dengan persamaan diatas, setiap satu karakter dari *plaintext* akan menghasilkan 2 karakter . Sehingga tiap kali melakukan enkripsi, yang dihasilkan pasti akan menghasilkan dua kali lipat dari ukuran *plaintext*. Sedangkan persamaan dekripsinya adalah :

$$P = (C1-K) – (C2-2K) \quad (2.2)$$

Pada saat proses dekripsi ukuran plaintext akan menjadi setengah dari ukuran.

2.6 Arnold Cat Map

Menurut (Munir, 2012b) *Arnold Cat Map* (ACM) merupakan fungsi *chaos* dwimatra dan bersifat *reversible*. Fungsi chaos ini ditemukan oleh Vladimir Arnold pada tahun 1960, dan kata "*cat*" muncul karena dia menggunakan citra seekor kucing dalam eksperimennya. ACM mentransformasikan koordinat (x, y) di dalam citra yang berukuran N x N ke koordinat baru (x', y'). Persamaan iterasinya adalah

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \bmod(N) \qquad (6.3)$$

yang dalam hal ini (xi, yi) adalah posisi pixel di dalam citra, (xi+1, yi+1) posisi *pixel* yang baru setelah iterasi ke-i; b dan c adalah integer positif sembarang. Determinan matriks $\begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix}$ harus sama dengan 1 agar hasil trans*form*asinya bersifat *area-preserving*, yaitu tetap berada di dalam area citra yang sama. ACM termasuk pemetaan yang bersifat satu-ke-satu karena setiap posisi *pixel* selalu ditransformasikan ke posisi lain secara unik. ACM diiterasikan sebanyak m kali dan setiap iterasi menghasilkan citra yang acak. Nilai b, c, dan jumlah iterasi m dapat dianggap sebagai kunci rahasia.

2.7 Pemrograman *Hypertext Preprocessor* (PHP)

PHP singkatan dari *Hypertext Processor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan web yang disisiplkan pada dokumen *Html*. Pengunaan PHP memungkinkan web dapat dibuat dinamis sehingga *maintenance* situs web tersebut menjadi lebih mudah dan efisien. PHP merupakan *software Open-Source* yang disebarkan dan dilisensikan secara gratis serta dapat di*download* secara bebas dari situs resminya *http://www.php.net* . PHP ditulis dengan menggunakan bahasa C (Suhartanto, 2012).

A. Sejarah singkat Pemrograman *Hypertext Preprocessor* PHP

PHP ditulis (diciptakan) oleh Rasmus Lerdorf, seorang software engineer asal Greenland sekitar tahun 1995. Pada awalnya PHP digunakan Rasmus hanya sebagai pencatat jumlah pengunjung pada website pribadi beliau. Karena itu bahasa tersebut dinamakan Personal *Home Page* (PHP) *Tools*. Tetapi karena perkembangan nya yang cukup disukai oleh komunitas nya, maka beliau pun merilis bahasa PHP tersebut ke publik dengan lisensi *open-source*. Saat ini, PHP adalah *server-side scripting* yang paling banyak digunakan di website-website di seluruh dunia, dengan versi sudah mencapai versi 5 dan statistiknya terus bertambah (Yuliano, 2007).

B. Sintaks PHP

Menurut (Suhartanto, 2012) Sintaks Program/*Script* ditulis dalam apitan tanda khusus PHP. Ada empat macam pasangan tag PHP yang dapat digunakan untuk menandai blok *script* PHP yaitu :

1. <?php…….?>
2. <script language=”PHP”>……..</script>
3. <?.....?>
4. <%.....%>.

BAB III
METODOLOGI PENELITIAN

3.1 Bagan Alir (*Flowchart*) Metode Penelitian

Menurut (Chrystanti and Wardati, 2011) Bagan alir (*flowchart*) adalah bagan yang menggambarkan urut-urutan instruksi proses dan hubungan satu proses dengan proses lainnya menggunakan simbol-simbol tertentu. *Flowchart* digunakan sebagai alat bantu komunikasi dan dokumentasi. *Flowchart* merupakan bagan yang menunjukkan pekerjaan secara keseluruhan dari sistem.

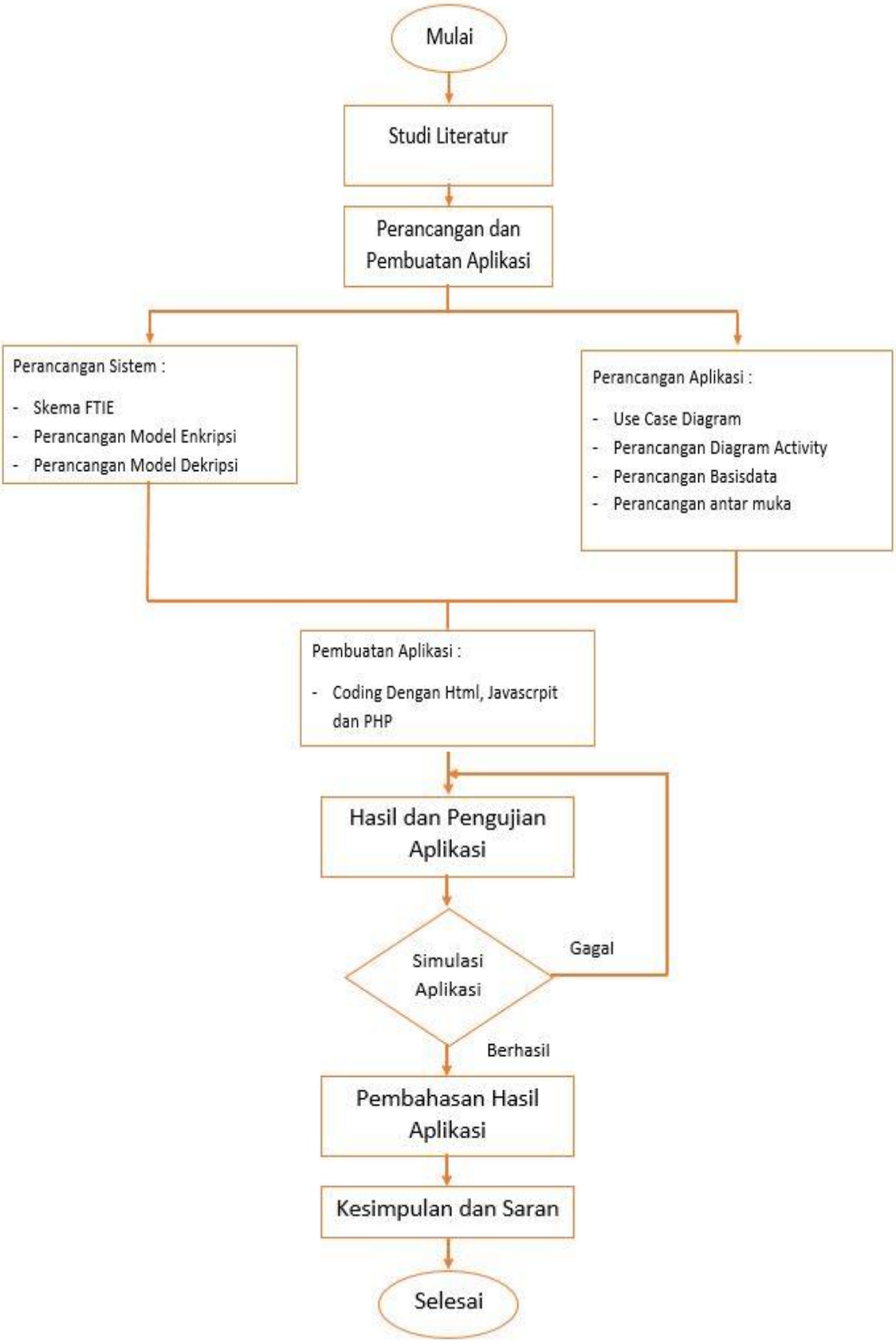
Bagan ini menjelaskan urut-urutan dari prosedur yang ada di dalam suatu sistem dan menunjukkan apa saja yang dikerjakan pada sistem. *Flowchart* dokumen (*document flowchart*) atau f*lowchart* formulir (*form* flowchart) merupakan bagan alir yang menunjukkan arus dari laporan dan formulir.

Flowchart program (program *flowchart*) adalah suatu bagan yang menggambarkan urutan proses secara mendetail dan hubungan atara proses yang satu dengan proses lainnya dalam suatu program.

Pada gambar 3.1 merupakan *flowchart* untuk proses pembuatan aplikasi FTIE enkripsi dan dekripsi, terdapat beberapa proses dalam *flowchart* tersebut meliputi isian pada bab I, landasan teori, perancangan dan pembuatan aplikasi, hasil dan pengujian aplikasi, pembahasan aplikasi, dan kesimpulan. Juga terdapat decision simulasi aplikasi jika terjadi gagal maka akan balik lagi ke pengujian aplikasi dan jika berhasil akan dilanjutkan menuju pembahasan hasil aplikasi.

Pada bagian perancangan dan pembuatan aplikasi terdapat 3 perancangan dan 4 perancangan aplikasi. Perancangan sistem meliputi, skema FTIE, perancangan model enkripsi dan perancangan model dekripsi. Sedangkan perancangan aplikasi meliputi, Use Case Diagram, perancangan diagram activity, perancangan basisdata dan perancangan antar muka yang nantinya masing masing dari perancangan akan dibahas pada bab III.

Bagan alir (*flowchart*) metode penelitian proses pembuatan aplikasi "*File To Image Encryption* (FTIE) dengan menggunakan algoritma *Randomized Text* dan *Arnold Cat Map* (ACM) untuk keamanan data digital pada sistem operasi android" dapat diilustrasikan pada gambar 3.1.



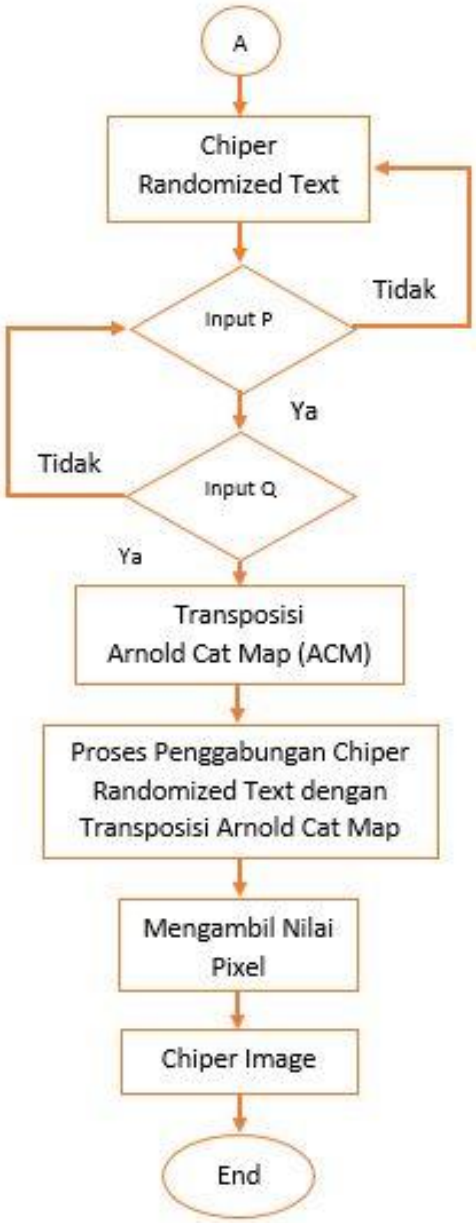
Gambar 3.1 *flowchart* metode penelitian

3.2 Perancangan Sistem

Perancangan sistem ini dilakukan dengan menggunakan skema FTIE menggunakan algoritma *Randomized Text* dan *Arnold Cat Map*, perancangan model enkripsi dan dekripsi, dengan adanya skema pada rancangan ini cukup membantu dalam membuat rancangan sehingga perancangan sistem menjadi terstruktur.

3.2.1 Flowchart *File* To *Image Encryption* (FTIE) dengan menggunakan algoritma Randomized Text dan Arnold Cat Map

Pada rancangan ini akan dibuat penggabungan algoritma *Randomized Text* dan *Arnold Cat Map* yang nantinya akan terbentuk skema dari *File To Image Encryption* (FTIE). Gambar 3.1 adalah perancangan dari skema FTIE dengan menggunakan algoritma *Randomized Text* dan *Arnold Cat Map*.

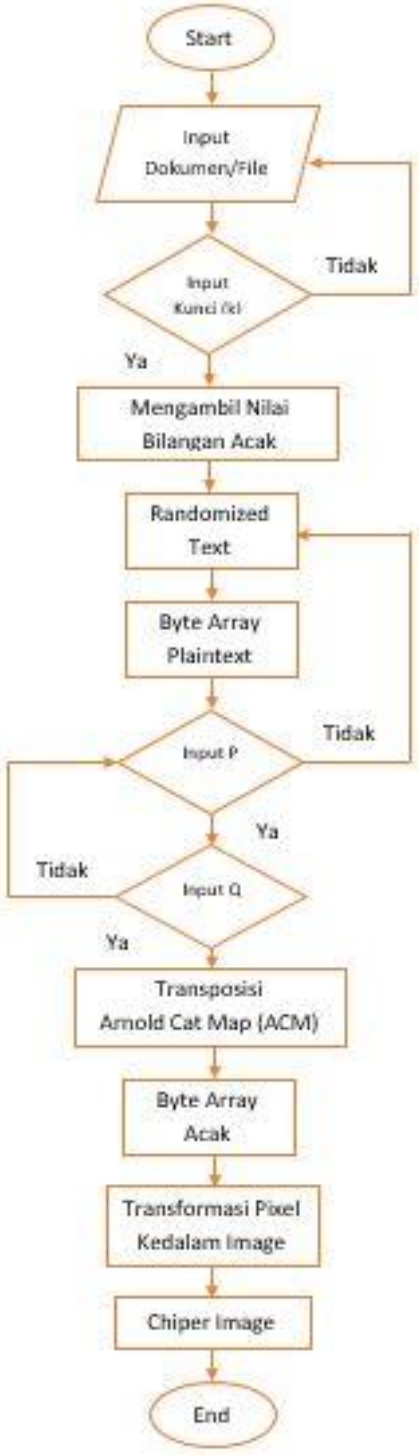


Gambar 3.2 *flowchart* tahapan FTIE dengan menggunakan algoritma *Randomized Text* dan *Arnold Cat Map*

Pada gambar di atas adalah rancangan alur enkripsi dari sebuah *file* menjadi sebuah gambar. Dimulai dari menentukan *file* yang akan dienkripsi dengan menggunakan algoritma *Randomized Text*, kemudian *byte* yang hasilkan akan transposisi dengan menggunakan algoritma ACM, dan mendapatkan *byte* yang sudah teracak, pada tahap akhir *byte* yang sudah teracak akan ditransformasi ke dalam bentuk gambar, dan pada akhirnya hasil dari enkripsi adalah sebuah gambar. Untuk perancangan model enkripsi dapat dilihat pada gambar dibawah ini.

3.2.2 Perancangan Model Enkripsi

Enkripsi merupakan proses untuk mengamankan suatu informasi agar informasi tersebut tidak dapat diketahui oleh orang lain. Pada tahapan ini akan dibuat *flowchart* perancangan model enkripsi dari sebuah *file* dan hasil akhirnya akan terbentuk sebuah gambar.



Gambar 3.3 *flowchart* perancangan model enkripsi FTIE

Pada gambar di atas merupakan *flowchart* rancangan dari model enkripsi teknik FTIE pada dokumen digital, dengan prosedur sebagi berikut :

1. Menentukan dokumen digital/*file* yang akan di enkripsi.
2. Mengambil nilai *byte* dari dokumen digital/*file* tersebut kemudian dimasukkan kedalam *byte*.
3. Menentukan nilai kunci (K).
4. Mengambil nilai dari pembangkit bilangan acak.
5. Enkripsi dari masing-masing nilai *byte* yang ada pada *byte* dengan menggunakan persamaan enkripsi *Randomized Text* kemudian hasilnya menjadi *byte* baru.
6. Menentukan nilai P dan Q.
7. Melakukan pengacakan posisi menggunakan algoritma ACM 1-D sesuai dengan nilai P dan Q kemudian akan menghasilkan *byte* baru.
8. Mengelompokkan *byte* baru menjadi nilai R,G,B yang masing-masing R,G,B mewalikili 1 *pixel*.
9. Gabung *pixel-pixel* menjadi satu kesatuan gambar.

3.2.3 Perancangan Model Dekripsi

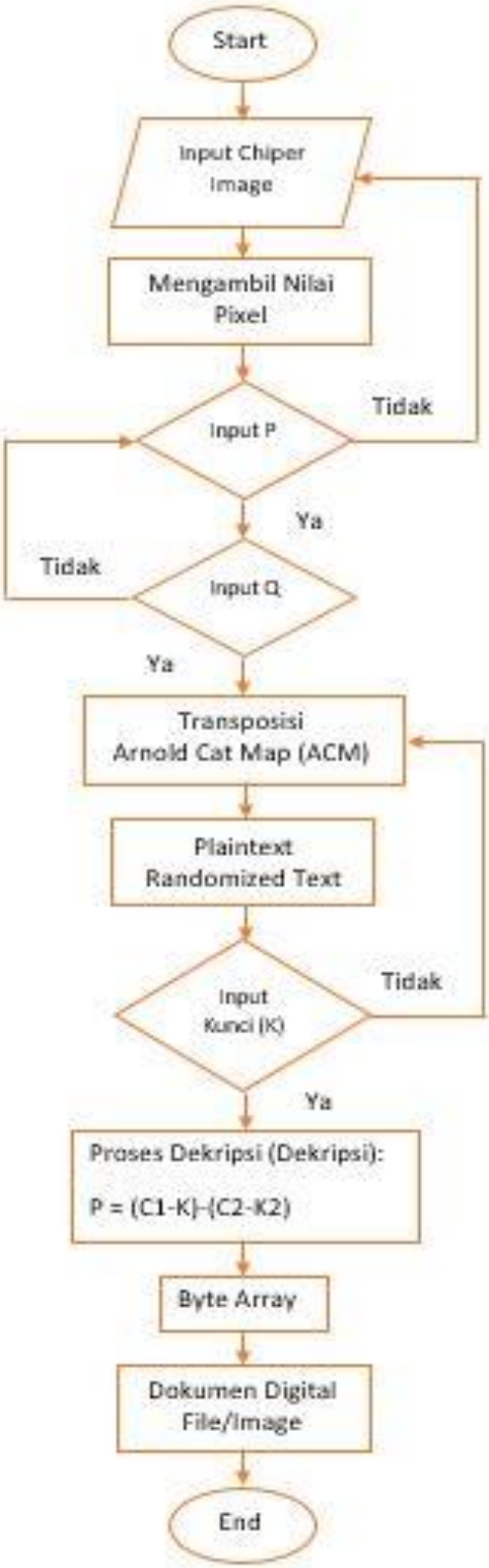
Dekripsi merupakan kebalikan dari enkripsi yaitu upaya untuk mengolah suatu data menjadi sesuatu yang dapat diutarakan dengan jelas yang bertujuan agar dapat dimengerti oleh orang yang tidak langsung mengalaminya sendiri.

Flowchart atau diagram alir adalah sebuah jenis diagram yang mewakili algoritme, alir kerja atau proses, yang menampilkan langkah-langkah dalam bentuk simbol-simbol grafis, dan urutannya dihubungkan dengan panah.

Diagram ini mewakili ilustrasi atau penggambaran penyelesaian masalah. Diagram alir digunakan untuk menganalisa, mendesain, mendokumentasi atau memanajemen sebuah proses atau program di berbagai bidang, Tujuan Membuat Flowchat:

1. Menggambarkan suatu tahapan penyelesaian masalah.
2. Secara sederhana, terurai, rapi dan jelas.
3. Menggunakan simbol-simbol standar.

Pada tahapan ini akan di buat flowchart perancangan model dekripsi dimulai dari proses mengambil file dan mengambil nilai pixel serta tahapan-tahapan transposisi ACM dari sebuah gambar dan hasil akhirnya berupa dokumen digital/file.



Gambar 3.4 *flowchart* perancangan model dekripsi FTIE

Pada gambar di atas merupakan *flowchart* rancangan dari model dekripsi teknik FTIE pada dokumen digital, dengan prosedur sebagi berikut :

1. Menentukan gambar yang akan di dekripsi.
2. Mengambil nilai *pixel*, dimana satu *pixel* akan memiliki nilai R,G,B. Selanjutnya akan digabung menjad tiap *byte*.
3. Menetukan nilai P dan Q.
4. Melakukan pengembalian posisi *byte* ke kondisi awal menggunakan persamaan ACM 1-D sesuai dengan nilai P dan Q kemudian akan menghasilkan *byte* baru.
5. Menentukan nilai kunci (K).
6. Dekripsi masing-masing nilai *byte* yang ada pada *byte* menggunakan persamaan dekripsi *Randomized Text* kemudian hasilnya menjadi *byte* baru.
7. Menggabung *byte* menjadi dokumen digital.

3.3 Kebutuhan Sistem

3.3.1 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang di butuhkan dalam penelitian ini adalah sebagai berikut:

1. Sistem operasi Microsoft Windows 10

   Penggunaan sistem operasi ini lebih dikarenakan sitem operasi ini memiliki *compatible* yang baik dengan aplikasi-aplikasi yang digunakan.
2. XAMPP

   Sebagai media untuk pembuatan dan penyimpanan *database* dari data-data yang akan digunakan.
3. Sublime Text 3

   Aplikasi *open source* ini digunakan untuk membuat website aplikasi dengan menggunakan bahasa *php, html, javascript* dan lain-lain.
4. Microsoft Visio

   Digunakan untuk membuat diagram *activity* dan *use case* diagram.

3.3.2 Kebutuhan Perangkat Keras

Kebutuhan perangkat lunak yang di butuhkan dalam penelitian ini adalah sebagai berikut:

1. Komputer *Dual Core*

   Berguna agar dapat memproses data jadi lebih cepat.
2. RAM 8GB atau yang lebih tinggi

   RAM penting karena meningkatkan kinerja komputer ketika dijalankan

3. *Hardisk* 500 GB

   Hardisk berguna untuk ruang penyimpan utama di komputer.

4. *Intel Core I5*

   Berguna agar proses pemrograman tidak lambat.

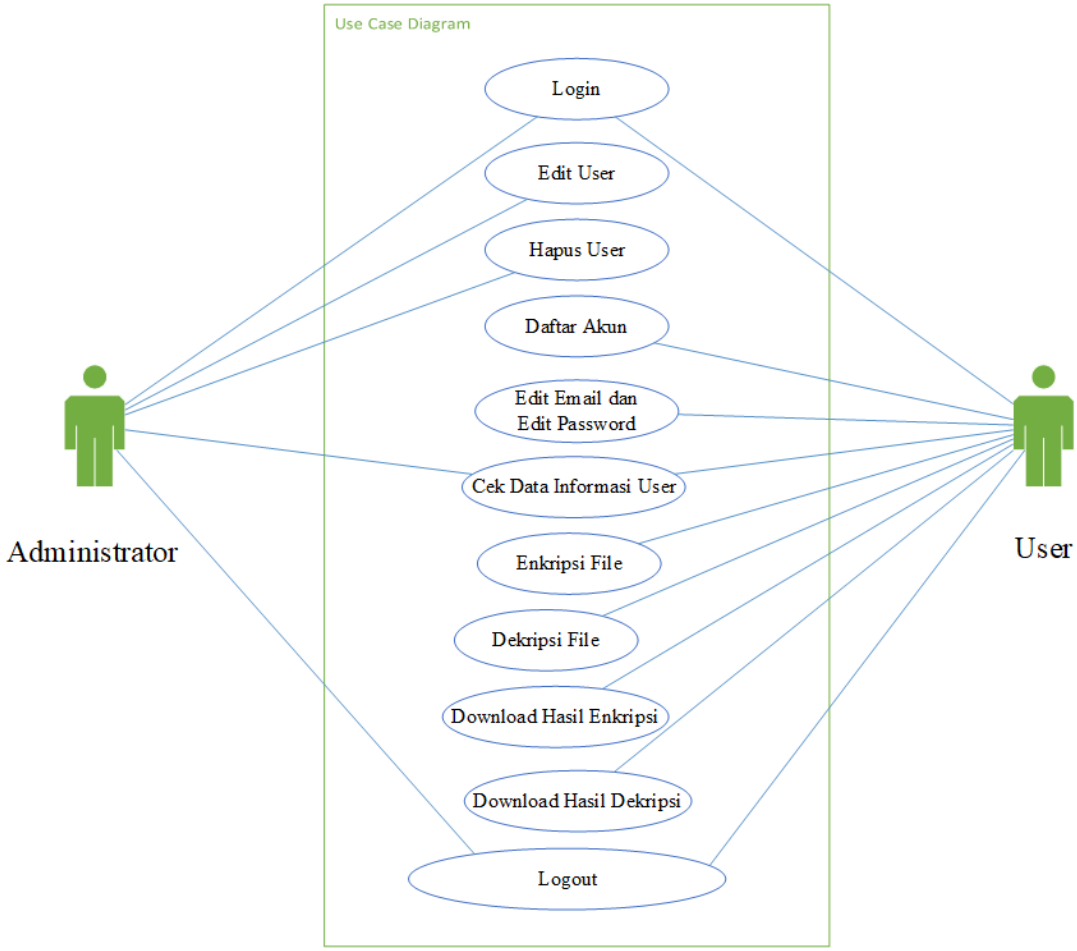
5. *Nvidia Geforce 740m*

   Berguna untuk mempercepat proses program.

3.4 Perancangan Aplikasi

Perancangan aplikasi ini dilakukan dengan menggunakan, *use case* diagram dan diagram *activity*, diagram-diagram ini cukup membantu dalam membuat rancangan sehingga perancangan aplikasi ini menjadi terstruktur.

3.4.1 Use Case Diagram



Gambar 3.5 *use case* diagram

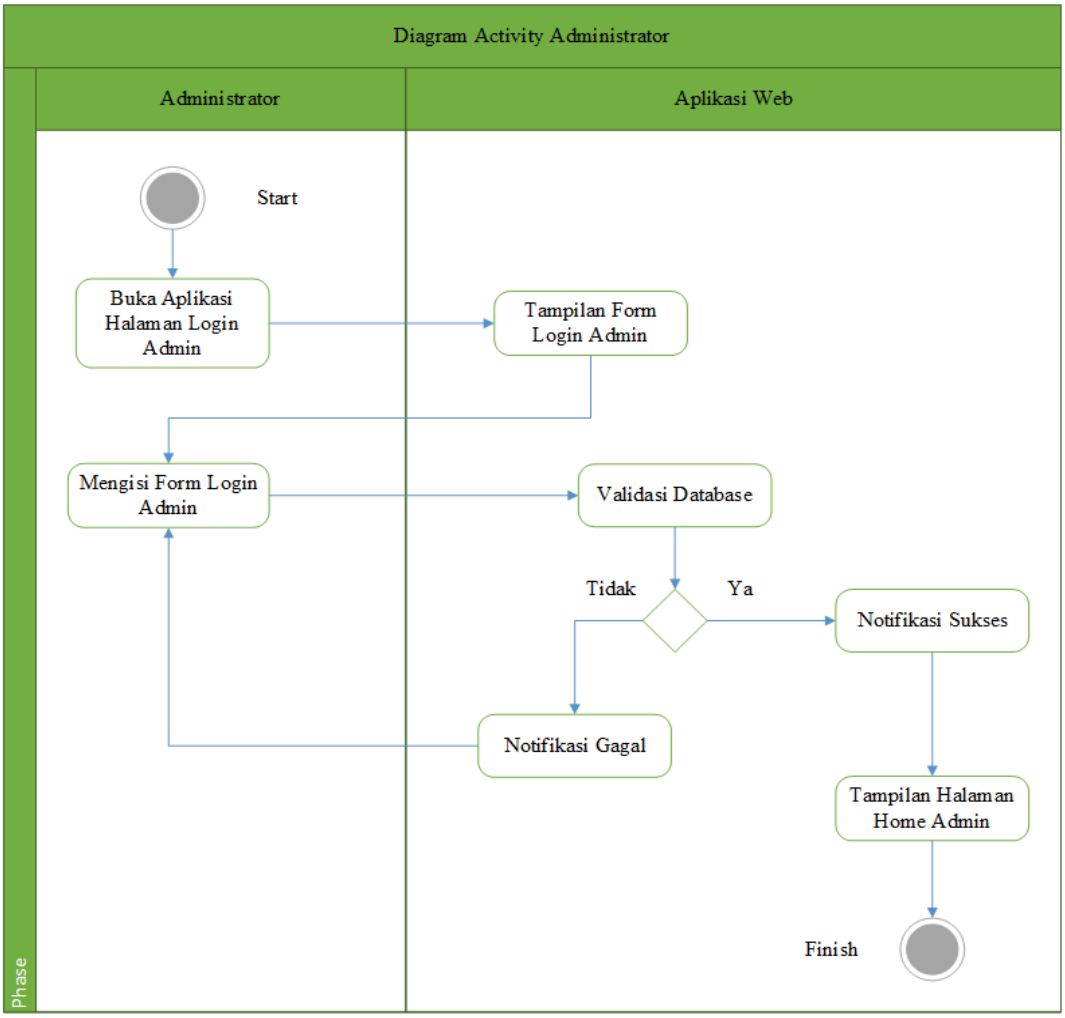
Pada *Use Case* Diagram di atas, maka dapat mendiskripsikan hal-hal sebagai berikut:

1. Administrator dan *user* merupakan *Actor*.

2. Administrator dan *user* dapat melakukan *login*, cek data informasi *user* dan *logout*.
3. Administrator fungsionalitas adalah: *login*, *edit user*, hapus *user*, cek data informasi *user* dan *logout*.
4. *User* fungsionalitas adalah: *login*, daftar, cek data informasi *user*, enkripsi, dekripsi, *download* hasil enkripsi, *download* hasil dekripsi, *logout*.

3.4.2 Perancangan Diagram Activity (*Login* administrator)

Gambar 3.6 menunjukan aktivitas antara *login* admin pada aplikasi web. Dimana admin dapat mengakses aplikasi ketika data yang dimasukan admin saat *login* valid dengan data admin yang tersimpan di *database*. Setelah sistem aplikasi ini melakukan validasi makan admin akan di arahkan ke dalam *home* admin.

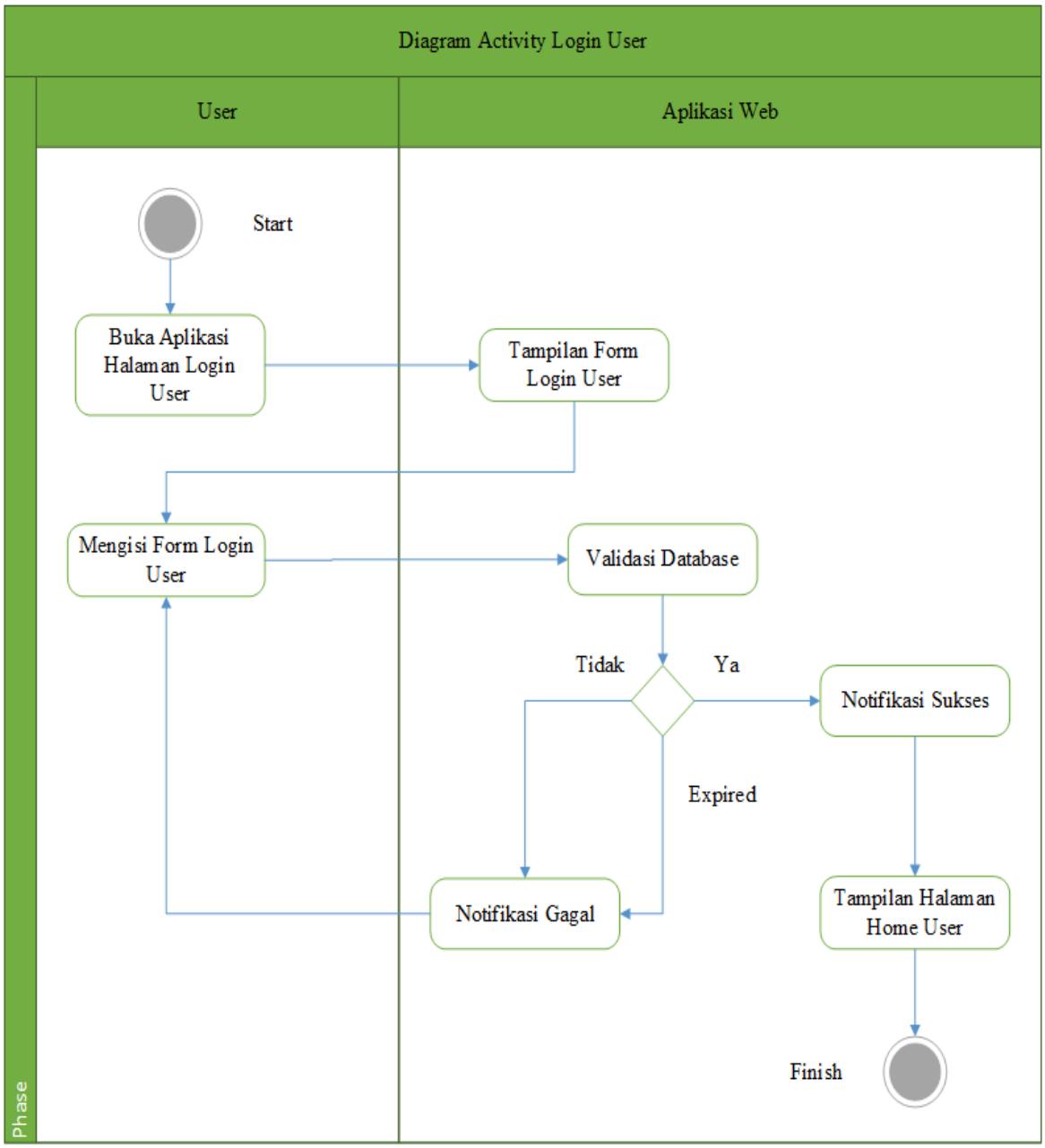


Gambar 3.6 diagram activity *login* admin

3.4.3 Perancangan Diagram Activity (*Login User*)

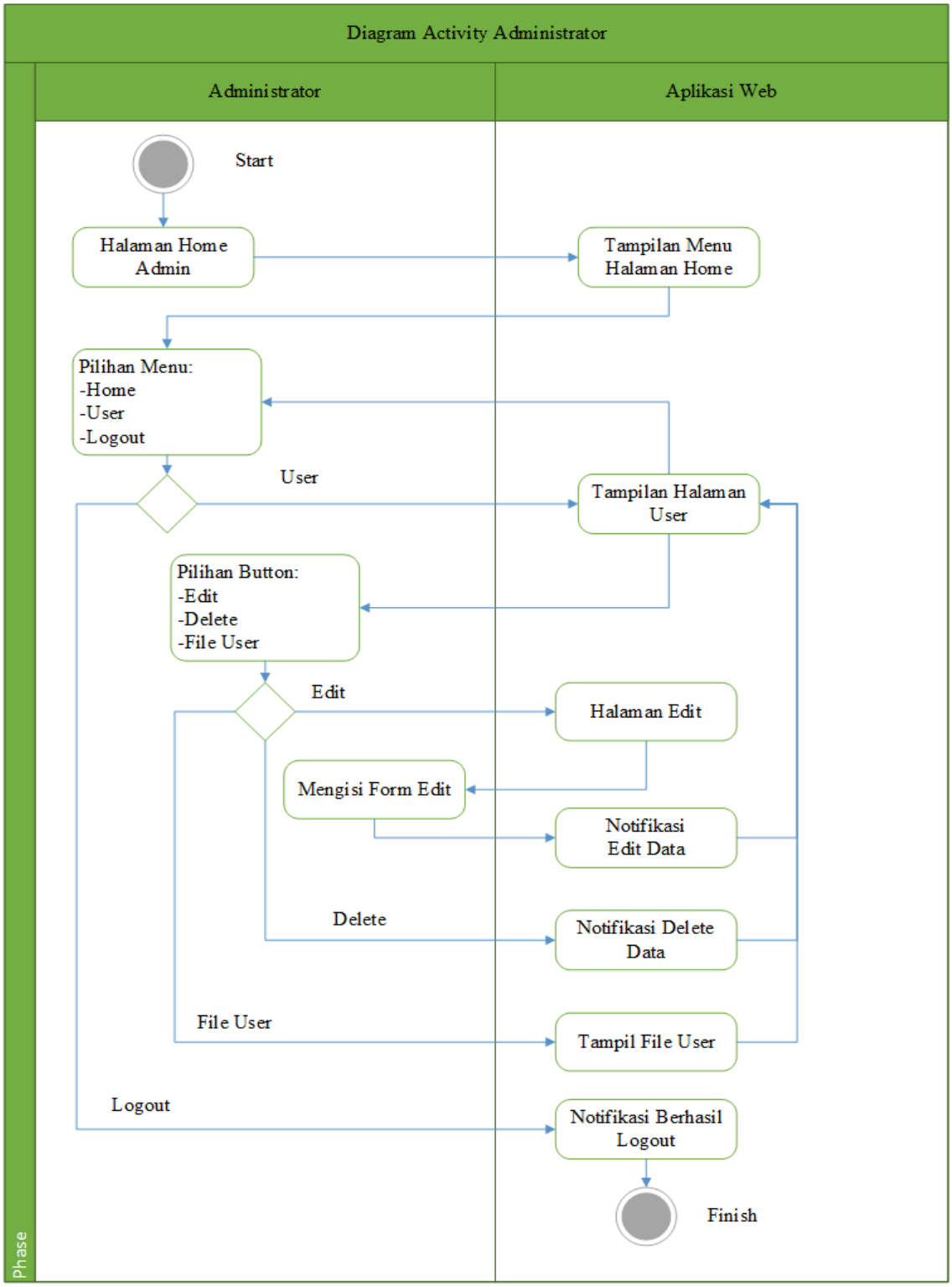
Gambar 3.7 menunjukan aktivitas antara *login user* pada aplikasi web. Dimana *user* dapat mengakses aplikasi ketika data yang dimasukan *user* saat *login* valid dengan data *user* yang tersimpan di *database*. Setelah sistem pada aplikasi ini melakukan validasi *database* maka akan di arahakan menuju halam *home user*.

Pada aktivitas di bawah ini juga menunjukan *user* akan *expired* Jika sudah menggunakan aplikasi ini selama 30 hari maka akan muncul notifikasi *expired* dan *user* diwajibkan untuk melakukan registrasi ulang agar Perancangan Diagram Activity (Administrator)



Gambar 3.7 diagram activity *login user*

Gambar 3.8 menunjukan aktivitas interaksi antar admininstrator dan aplikasi web, dimulai dari admin masuk ke dalam halaman *home* admin beserta hak akses admin terhadap aplikasi web.

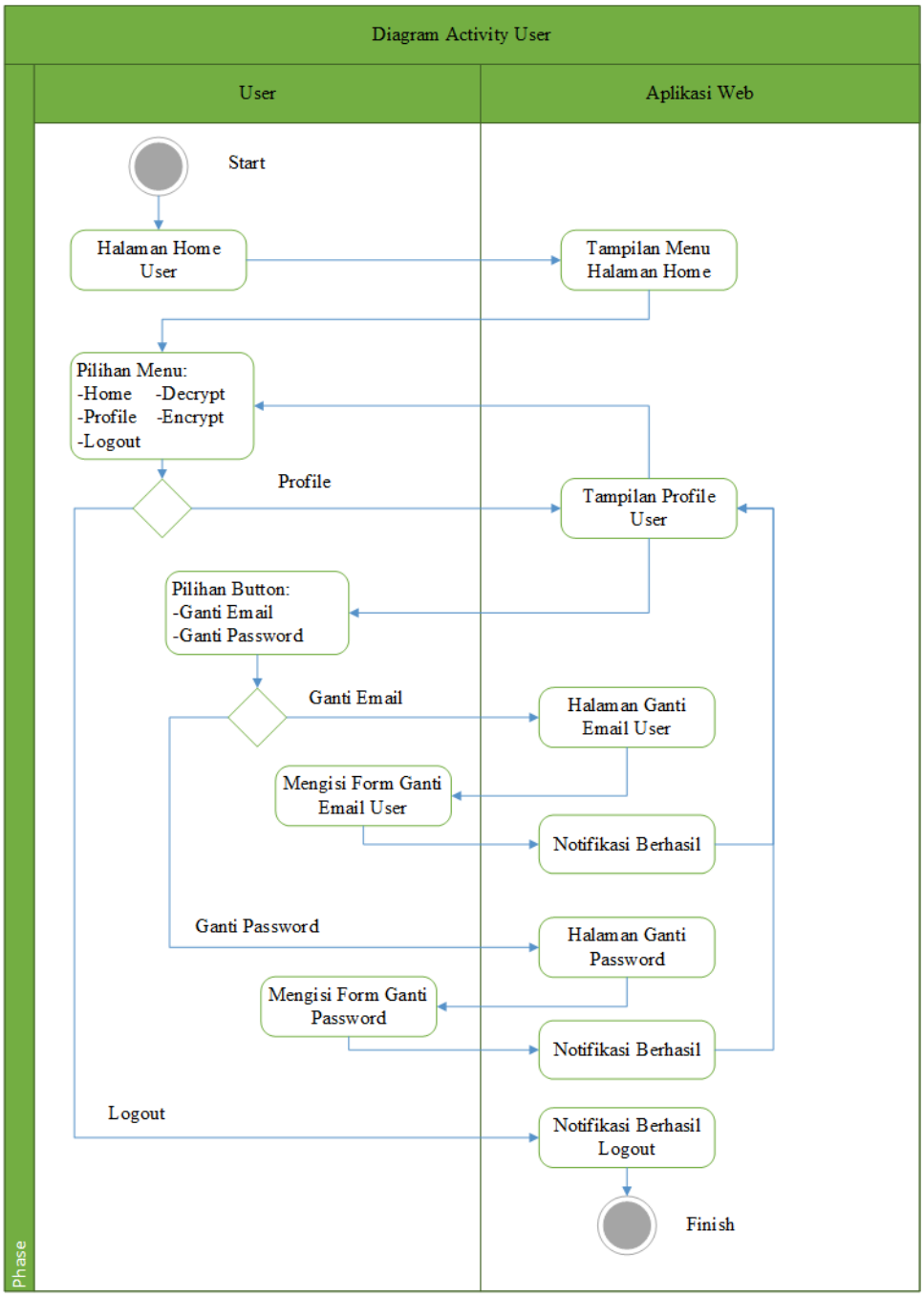


Gambar 3.8 diagram activity administrator

3.4.4 Perancangan Diagram Activity (*User*)

Activity diagram adalah sesuatu yang menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

Gambar 3.9 menunjukan aktivitas interaksi antara *user* dan aplikasi web, dimulai dari *user* masuk ke dalam halaman *home user* beserta hak akses *user* terhadap aplikasi ini.

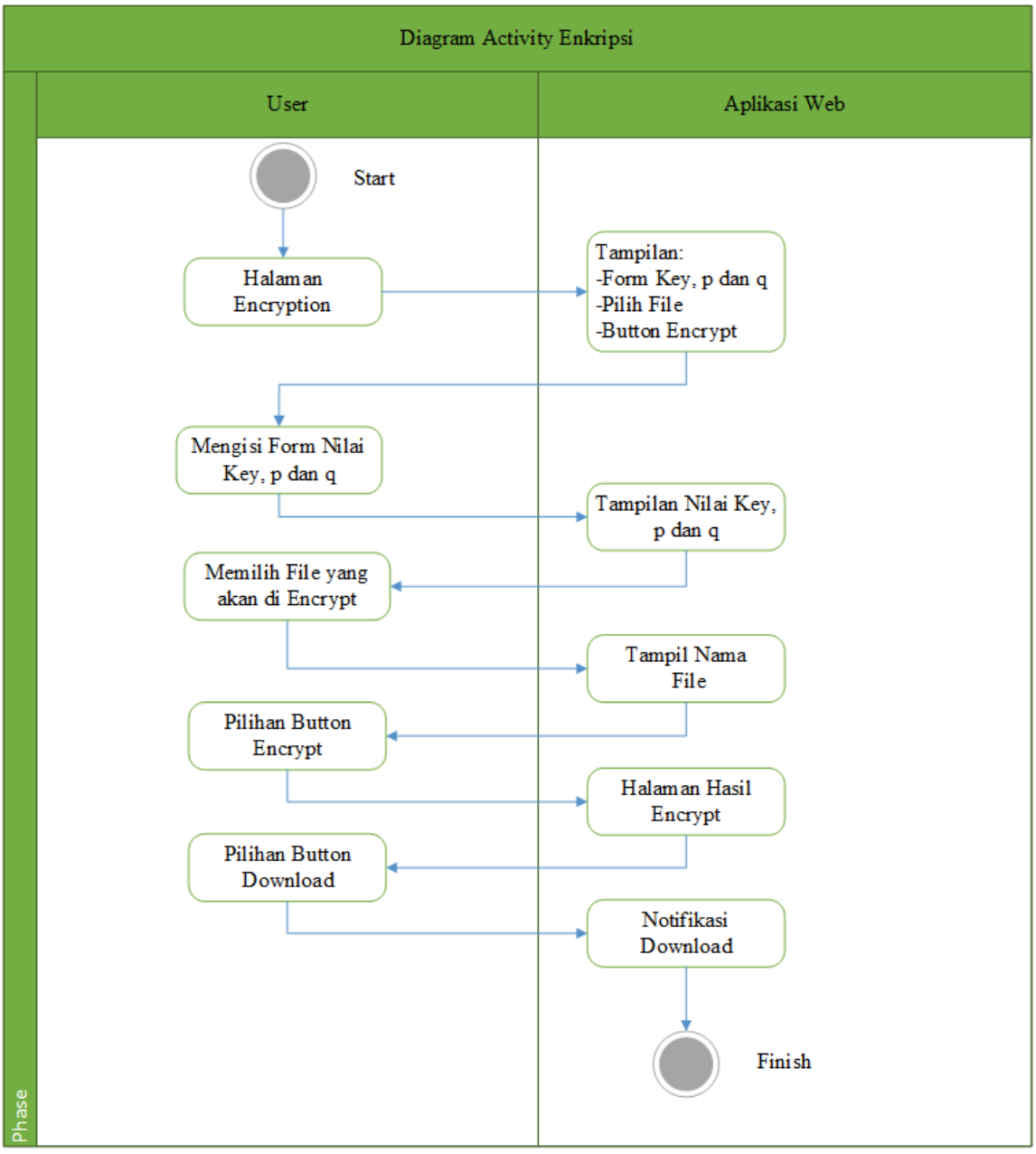


Gambar 3.9 diagram activity *user*

3.4.5 Perancangan Diagram Activity *User* (Halaman Enkripsi)

Gambar 3.10 menunjukan aktivitas interaksi antara *user* dan halaman enkripsi, dimulai dari *user* masuk ke dalam halaman enkripsi beserta hak akses *user* terhadap halaman enkripsi ini.

Pada halaman ini *user* akan diminta untuk mengisi *form*, memilih *file* dan melakukan proses enkripsi, setelah melakukan proses enkripsi maka *user* akan mendapatkan hasil enkripsi berupa sebuah gambar. Hasil gambar tersebut dapat di*download* agar, *user* dapat menyimpan *file* hasil enkripsi tersebut.

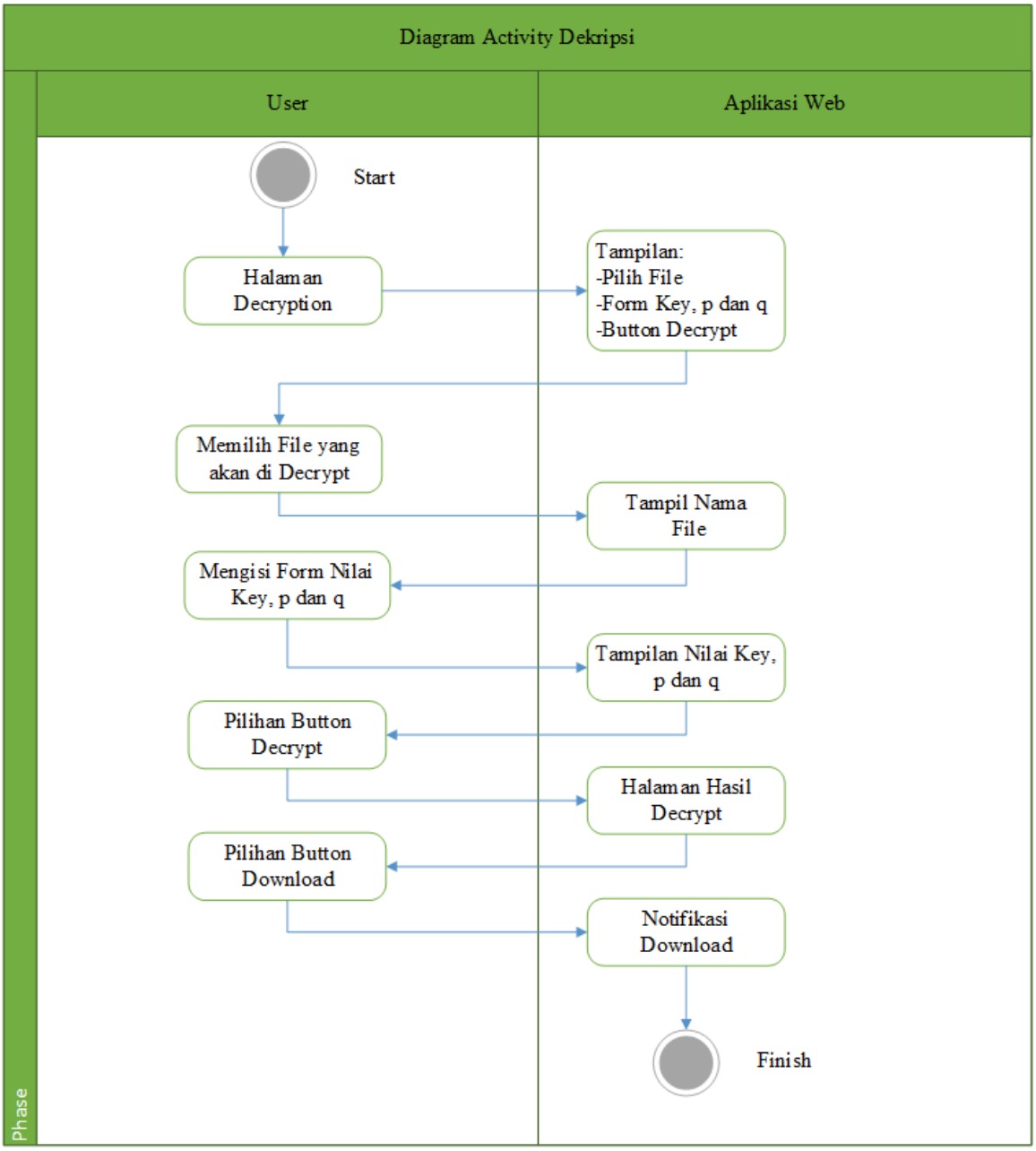


Gambar 3.10 diagram activity *user* (halaman enkripsi)

3.4.6 Perancangan Diagram Activity *User* (Halaman Dekripsi)

Gambar 3.11 menunjukan aktivitas interaksi antara *user* dan halaman dekripsi, dimulai dari *user* masuk ke dalam halaman dekripsi beserta hak akses *user* terhadap halaman dekripsi ini.

Pada halaman ini *user* akan diminta untuk memilih *file* gambar enkripsi yang akan di dekripsi. Setelah memilih *file user* akan diminta untuk mengisi *form* dan melakukan proses enkripsi, setelah melakukan proses dekripsi maka *user* akan mendaptkan hasil dekripsi berupa sebuah *file* asli atau *file* sebelum di enkripsi. Hasil *file* asli tersebut dapat di*download*, agar *user* dapat menyimpan *file* hasil dekripsi tersebut.



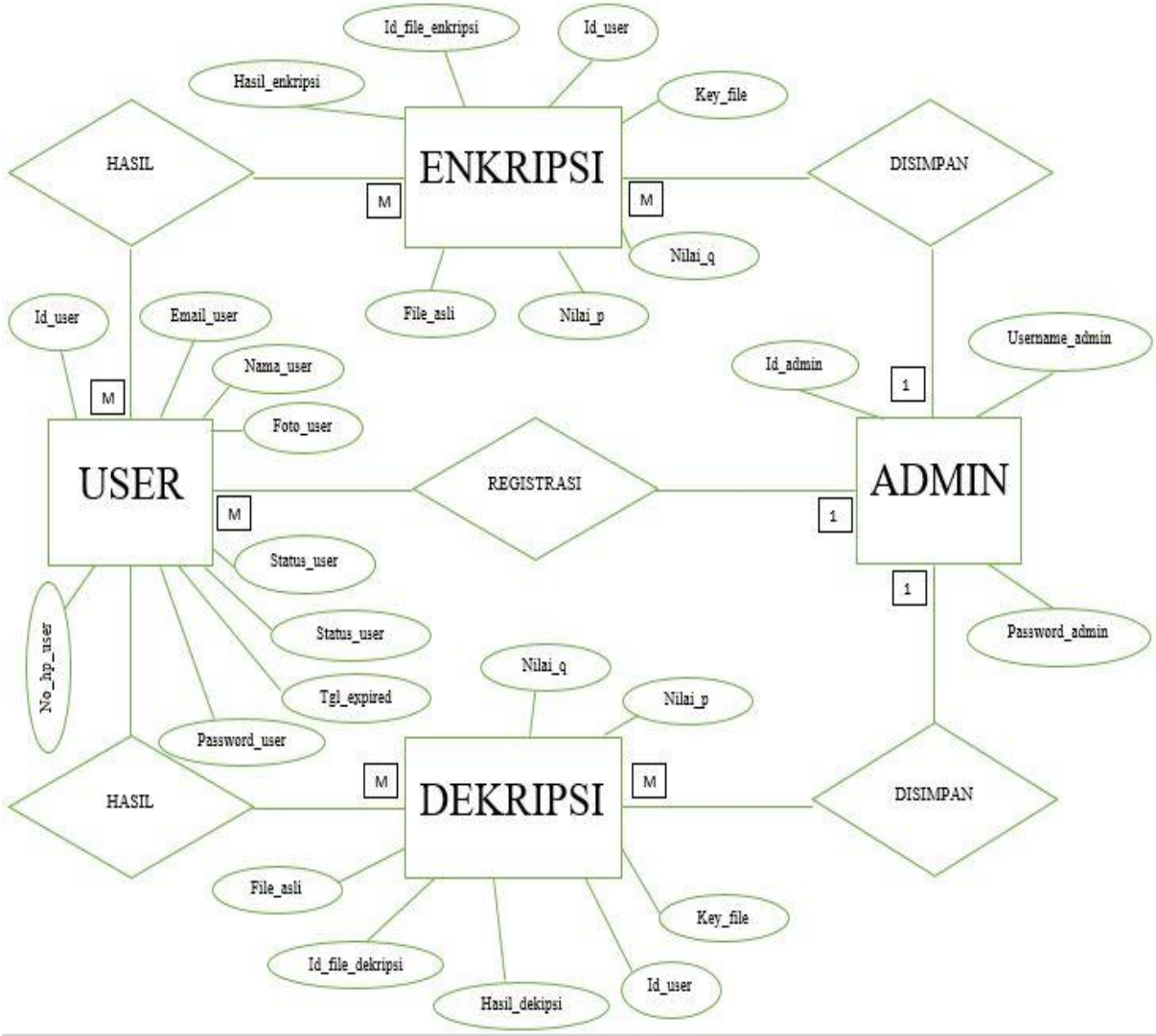
Gambar 3.11 diagram *activity user* (halaman dekripsi)

3.5 Perancangan *Basisdata*

Basisdata digunakan untuk menampung semua data yang diperlukan sistem. Dalam tahapan ini dibuat struktur basisdata yang disesuaikan dengan kebutuhan sistem.

3.5.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) digunakan untuk menunjukkan hubungan antara keempat tabel, yaitu tabel data administrator, tabel data user, tabel data enkripsi dan tabel data dekripsi.



Gambar 3.12 Entity Relationship Diagram (ERD)

3.5.2 Tabel Data Administrator

Tabel administrator berfungsi sebagai penyimpanan terhadap data admin berupa *username* admin dan *password* admin.

Tabel 3.1 data administrator

No	Name	Type	Collation	Null	Default	Extra
1	Id_admin	Int (11)		No	None	AUTO_INCREMENT
2	username_Admin	varchar (255)	Latin1_swedish_ci	No	None	
3	password_Admin	varchar (255)	Latin1_swedish_ci	No	None	

3.5.3 Tabel Data *User*

Tabel *user* berfungsi sebagai penyimpanan terhadap data *user* berupa nama *user*, *email user*, *password user*, no hp *user*, status *user* dan tgl *expired user*. Tabel *user* juga berfungsi sebagai hak akses *user* kedalam aplikasi web.

Tabel 3.2 data *user*

No	Name	Type	Collation	Null	Default	Extra
1	id_user	Int (11)		No	None	AUTO_INCREMENT
2	nama_user	varchar (255)	Latin1_swedish_ci	No	None	
3	email_user	varchar (255)	Latin1_swedish_ci	No	None	
4	password_user	varchar (255)	Latin1_swedish_ci	No	None	
5	no_hp_user	varchar (255)	Latin1_swedish_ci	No	None	
6	status_user	varchar (255)	Latin1_swedish_ci	No	None	
7	tgl_expired	date		No	None	

3.5.4 Tabel Data Enkripsi

Tabel enkripsi berfungsi sebagai penyimpanan data enkripsi yang di hasilkan oleh *user* berupa nilai dari *key*, nilai p, nilai q, *file* asli dan hasil enkripsi.

Tabel 3.3 data enkripsi

No	Name	Type	Collation	Null	Default	Extra
1	id_file_enkripsi	Int (11)		No	None	AUTO_INCREMENT
2	id_user	Int (11)		No	None	
3	key_file	varchar (255)	Latin1_swedish_ci	No	None	
4	nilai_p	varchar (255)	Latin1_swedish_ci	No	None	
5	nilai_p	varchar (255)	Latin1_swedish_ci	No	None	
6	file_asli	varchar (255)	Latin1_swedish_ci	No	None	
7	hasil_enkripsi	varchar (255)	Latin1_swedish_ci	No	None	

3.5.5 Tabel Data Dekripsi

Tabel dekripsi berfungsi sebagai penyimpanan data dekripsi yang di hasilkan oleh *user* berupa nilai dari *key*, nilai p, nilai q, hasil dekripsi dan *chiper image*.

Tabel 3.4 data dekripsi

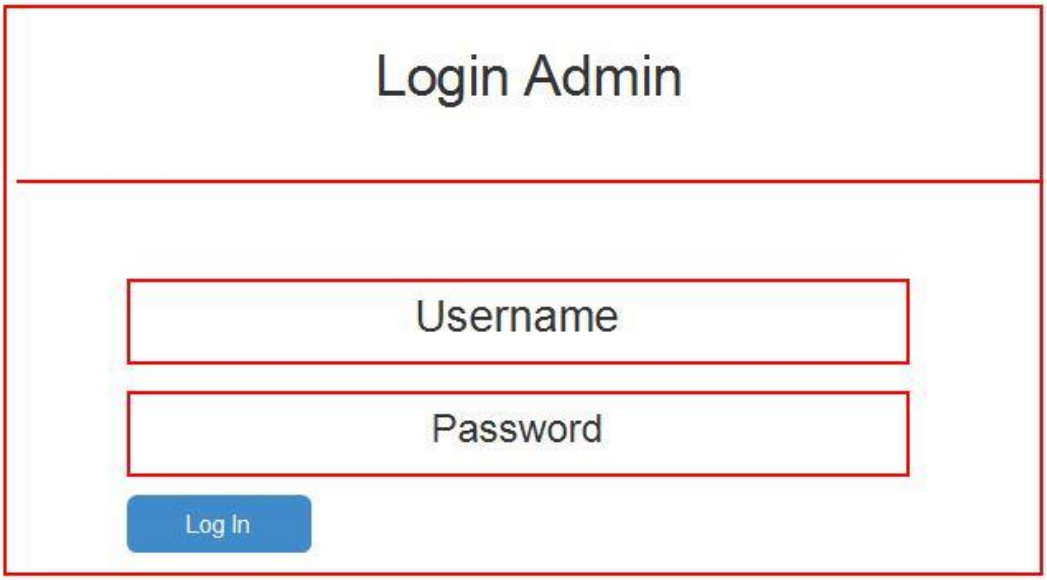
No	Name	Type	Collation	Null	Default	Extra
1	id_file_dekripsi	Int (11)		No	None	AUTO_INCREMENT
2	id_user	Int (11)		No	None	
3	key_file	varchar (255)	Latin1_swedish_ci	No	None	
4	nilai_p	varchar (255)	Latin1_swedish_ci	No	None	
5	nilai_p	varchar (255)	Latin1_swedish_ci	No	None	
6	hasil_dekripsi	varchar (255)	Latin1_swedish_ci	No	None	
7	chiper_image	varchar (255)	Latin1_swedish_ci	No	None	

3.6 Perancangan Antarmuka

Secara umum ada dua fungsi yang tersedia pada aplikasi android yaitu untuk melakukan proses enkripsi dan dekripsi. Untuk perancangan *interface* dapat dilihat pada penjelasan dibawah ini.

3.6.1 Perancangan Antarmuka *Login* Admin

Rancangan halaman *login* admin yang terdapat pada gambar 3.13 merupakan halaman *login* administrator untuk mendapatkan hak akses kepentingan-kepentingan administrator didalam aplikasi *FTIE* berbasis website ini. Dimana terdapat kotak *form* untuk memasukan *username* dan *password* sebelum dapat mengakses halaman administrator ini.

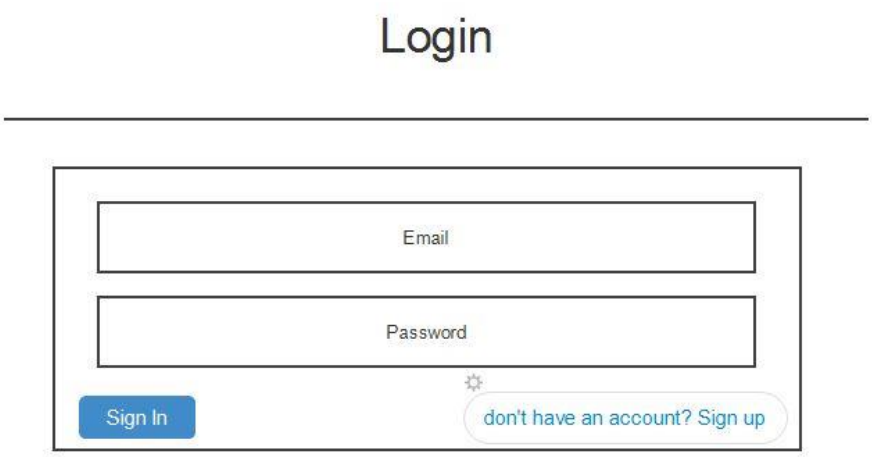


Gambar 3.13 perancangan *login* admin

3.6.2 Perancangan Antarmuka *Login User*

Rancangan halaman *login user* yang terdapat pada gambar 3.14 merupakan halaman *login user* untuk mendapatkan hak akses kedalam halaman *user*, karena *user* diwajibkan untuk *login* agar dapat mengakses aplikasi *FTIE* berbasis website ini.

Dimana terdapat kotak *form* untuk memasukan *username* dan *password* sebelum dapat mengakses aplikasi ini. Dibagian perancangan *login user* ini juga terdapat *button sign-up* yang tujuannya sebagai pendaftaran *user*.



Gambar 3.14 perancangan *login user*

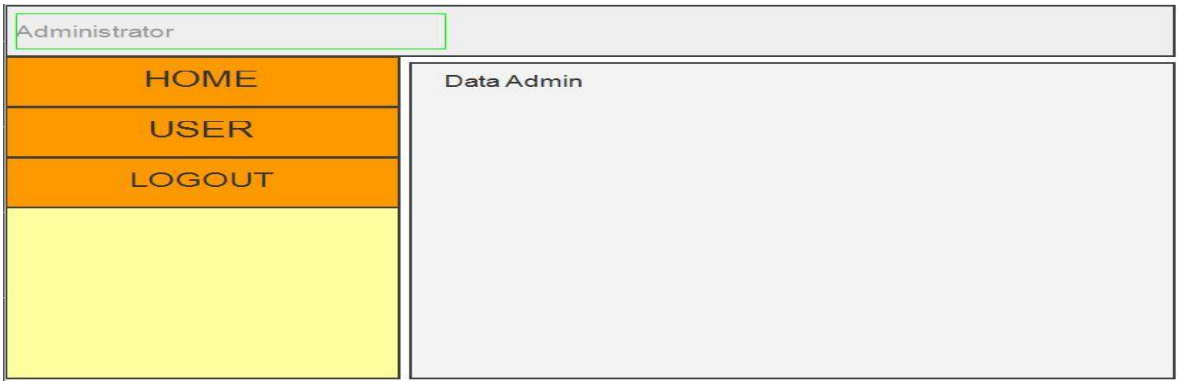
3.6.3 Perancangan Antarmuka Daftar *User*

Rancangan antarmuka halaman daftar *user* ini terdapat pada gambar 3.15 merupakan halaman untuk *user* mendaftarkan diri agar dapat mengakses aplikasi web ini. Jika *user* tidak melakukan pendaftaran maka *user* tidak bisa mengakses aplikasi web ini. Pada rancangan ini terdapat rancangan *form* yang meliputi nama, *email*, *password* dan ponsel. Disebalah kanan *button sign up* juga terdapat *button login* yang nantinya jika di klik maka *user* akan kembali kehalaman *login*.

Gambar 3.15 perancangan daftar *user*

3.6.4 Perancangan Antarmuka *Home* Admin

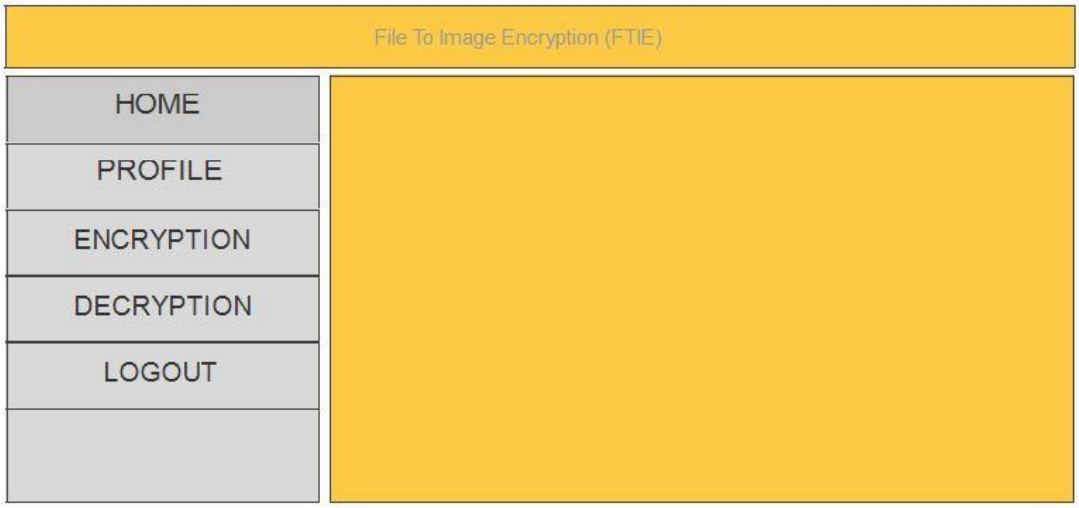
Rancangan antarmuka halaman *home* admin terdapat pada gambar 3.16 merupakan tempat interaksi antara admin dengan sistem. Dimana admin memiliki hak akses terhadap pengelolan data *user* yang meliputi, *edit user*, hapus *user* dan melihat *file user*. pada rancangan menu terdapat di sisi kiri layar, menu ini berisikan halaman *home*, halaman data *user* dan *log out*.



Gambar 3.16 perancangan antarmuka *home* admin

3.6.5 Perancangan Antarmuka *Home User*

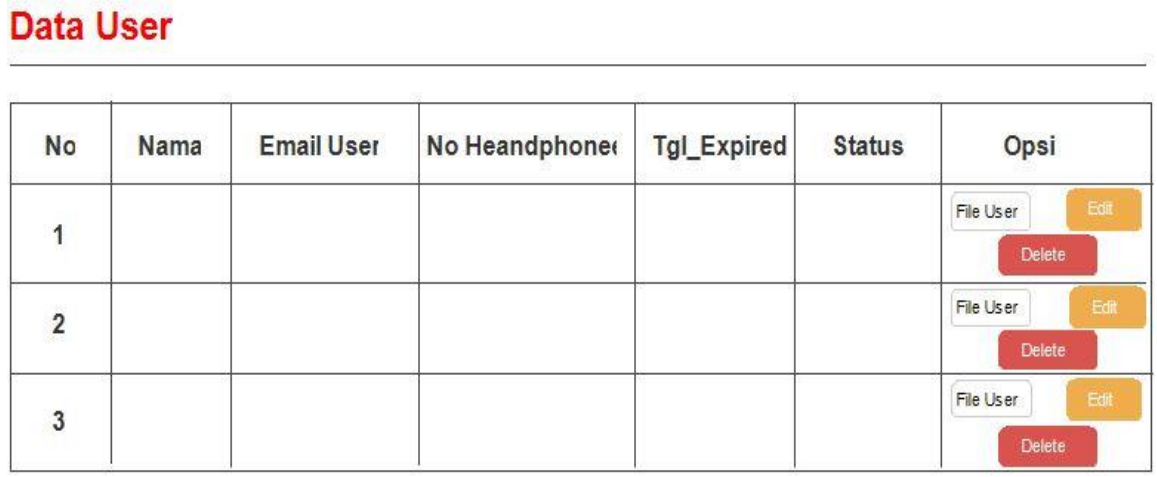
Rancangan antarmuka halaman *home user* terdapat pada gambar 3.17 merupakan tempat interaksi antara *user* dengan aplikasi web. Dimana *user* memiliki hak akses terhadap aplikasi web yang meliputi, *edit email* dan *password user*, enkripsi *file*, dekripsi *file* dan *logout*. pada rancangan menu terdapat di sisi kiri layar, menu ini berisikan halaman *home*, halaman *profile*, halaman *encryption*, halaman *decryption* dan *log out*.



Gambar 3.17 perancangan antarmuka *home user*

3.6.6 Perancangan Antarmuka Data *User*

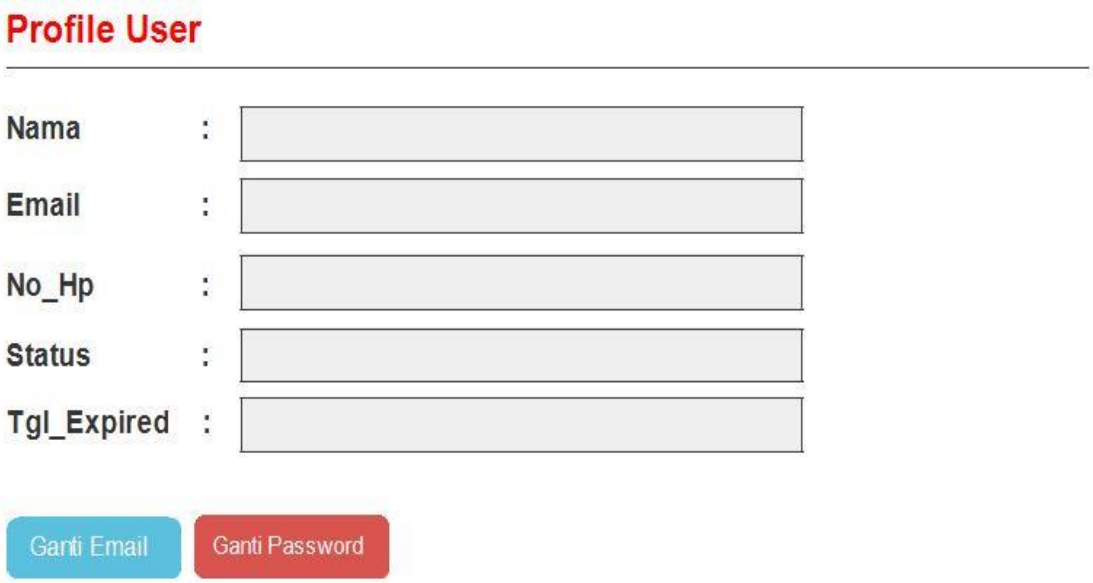
Rancangan antarmuka halaman data *user* terdapat pada gambar 3.17 merupakan tempat interaksi antar admin dengan *user*. Rancangan ini adalah rancangan dimana admin dapat menghapus *user*, mengedit *user* dan melihat *file user*. Pada rancangan ini terdapat tombol-tombol *button delete,edit* dan *file user*.



Gambar 3.18 perancangan antarmuka data *user*

3.6.7 Perancangan Antarmuka *Profile User*

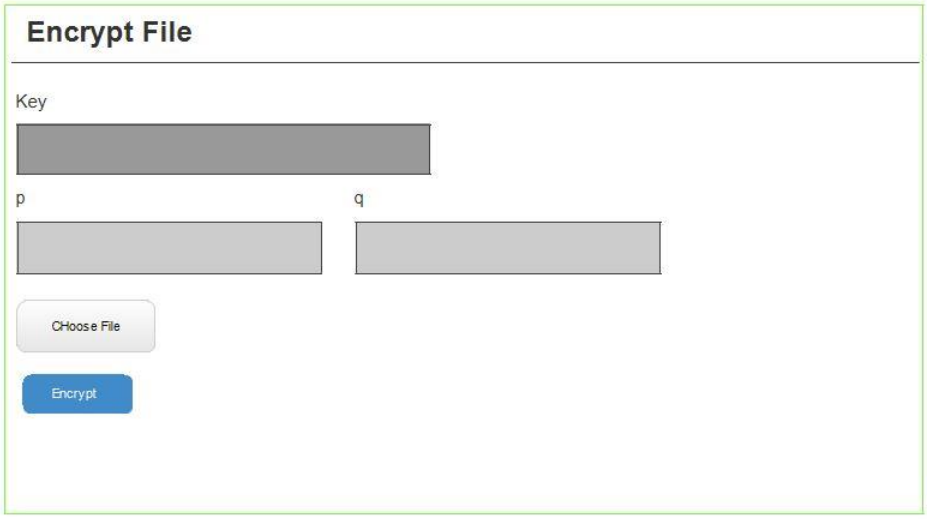
Rancangan antarmuka halaman *profile user* ini terdapat pada gambar 3.19 tujuan dari rancangan agar *user* dapat melihat *profile user* dan status *active* atau *expired*. Pada rancangan ini juga terdapat pilihan *button* ganti *email* dan ganti *password user*.



Gambar 3.19 perancangan antarmuka *profile user*

3.6.8 Perancangan Antarmuka Enkripsi

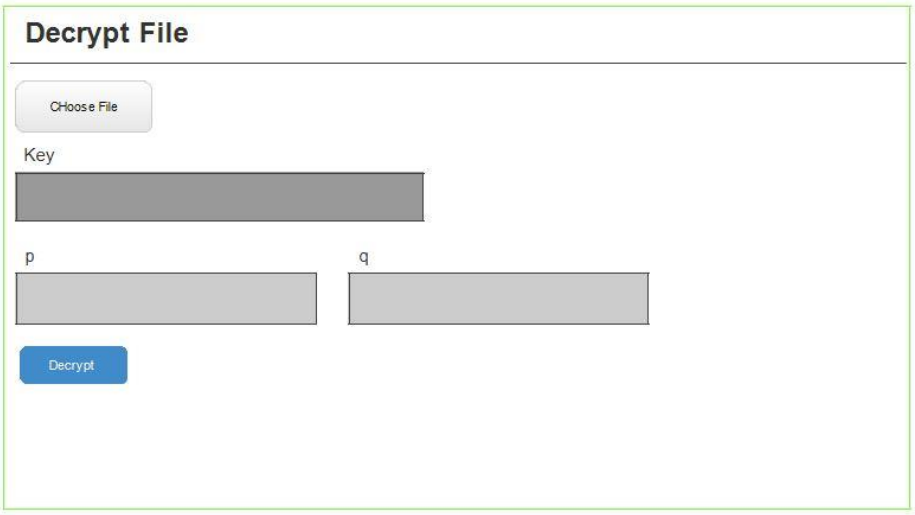
Rancangan antarmuka halaman enkripsi ini terdapat pada gambar 3.20 pada rancangan ini terdapat tiga *form* yang meliputi input nilai *key*, nilai p dan nila q. Rancangan ini juga terdapat pilih *file* yang akan di enkripsi, dan jika semuanya sudah terisi maka *user* akan mengeklik tombol *button encrypt* agar *file* dapat di enkripsi.



Gambar 3.20 perancangan antarmuka enkripsi

3.6.9 Perancangan Antarmuka dekripsi

Rancangan antarmuka halaman dekripsi ini terdapat pada gambar 3.21. pada rancangan ini juga terdapat pilih *file* enkripsi yang akan di dekripsi. Rancangan ini juga terdapat tiga *form* yang meliputi input nilai *key*, nilai p dan nila q. Rancangan, dan jika semuanya sudah terisi maka *user* akan mengeklik tombol *button decrypt* agar *file* dapat di enkripsi.



Gambar 3.21 perancangan antarmuka dekripsi

3.7 Analisis dan Pengujian

Analisis dan pengujian dilakukan untuk mengetahui tingkat kekuatan dan efektifitas algoritma. Adapun beberapa faktor yang dinilai adalah sebagai berikut:

3.7.1 Analisis Entropy

Analisis bertujuan untuk menganalisis tingkat kecakan sebuah informasi yang sudah terenkripsi, Nilai entropy ideal jika sebuah informasi dienkripsi dan dalam kondisi teracak adalah 7,99902 (~8). Jika nilai entropy lebih kecil dari 8, dapat dikatakan sistem enkripsi masih dapat ditebak (Suprianto, Prayudi and Sugiantoro, 2017). Entropy dari pesan dapat dihitung dengan rumus (Younes and Jantan, 2008):

$$H_e = -\sum_{k=0}^{G-1} P(k) \log_2 (P(k)) \quad 3.1$$

Keterangan:

He : Nilai entropy

G : Pesan atau nilai keabuan dari citra (0..255)

P(k) : Peluang kemunculan simbol ke-k

yang dalam hal ini P(k) menyatakan peluang simbol mi di di dalam pesan dan entropi dinyatakan dalam satuan bit. Pesan acak seharusnya memiliki entropi yang ideal atau hampir mendekati 8. Jika entropi kurang dari delapan, maka terdapat derajat mampu-prediksi (predictability) yang merupakan ancaman bagi keamanan (Munir, 2012).

Hasil dari analisis menggunakan metode akan dimasukkan kedalam Tabel 3.5 dengan format kolom seperti terlihat pada Tabel dibawah ini.

Tabel 3.5 format analisis entropy

No	Nama	Entropy		
		File Asli	Enkripsi	Dekripsi
1	Perancangan.docx	5,5555555	7,7777777	5,5555555
2	Perancangan.pdf	4,4444444	6,6666666	4,4444444

3.7.2 Analisis Ruang Kunci

a. Brute Force Attack

Brute force attack adalah metode untuk mengalahkan skema kriptografi dengan mencoba semua kemungkinan password atau kunci. Brute force attack memungkinkan dapat menyerang

kunci privat di hampir semua skema kriptografi, tipe serangan bergantung pada ukuran kunci dan mekanisme pada enripsi yang digunakan (Wicaksono, 2013).

b. Exhaustive Attack

Pada jenis serangan ini, kriptanalis tidak mencoba-coba semua kemungkinan kunci tetapi menganalisis kelemahan algoritma kriptografi untuk mengurangi kemungkinan kunci yang tidak mungkin ada. Analisis dilakukan dengan dengan memecahkan persamaan-persamaan matematika (yang diperoleh dari definisi suatu algoritma kriptografi) yang mengandung peubah-peubah yang merepresentasikan plainteks atau kunci (Fithria, 2018).

3.7.3 Analisis Waktu Enkripsi dan Dekripsi

Agar dapat mengetahui performa dari algoritma yang telah digunakan, maka pada penelitian ini akan dilakukan analisis lama waktu (detik) eksekusi yang diperlukan untuk melakukan enkripsi dan dekripsi berdasarkan ukuran dari sebuah file dan akan ditampilkan dalam sebuah tabel seperti pada Tabel 3.6 dibawah ini.

Tabel 3.6 analisis waktu

Nama File	Ukuran File (Kb)	Waktu Enkripsi	Waktu Dekripsi
Perancangan.docx	1,111 KB	11.111	11.111
Perancangan.pdf	2,222 KB	22.222	2.2222

3.7.4 Analisis Besar Ukurusan File Hasil Enkripsi

Untuk mengetahui hasil ukuran dari file setelah di enkripsi, pada pada penelitian ini akan di lakukan analisis size file yang telah di enkripsi. Yang diperlukan untuk melakukan analisis enkripsi dan dekripsi berdasrakan ukuran dari file asli akan ditampilkan pada tabel dibawah ini.

Tabel 3.7 analasis size hasil enkripsi

Nama File	Ukuran File Asli (Kb)	File PNG Hasil Enkripsi
Perancangan.docx	1,111 KB	4,444,444 byte
Perancangan.pdf	2,222 KB	6,666,666 byte

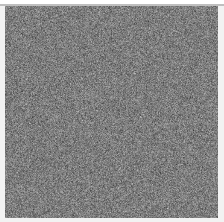
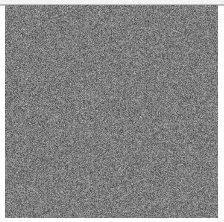
3.7.5 Pengujian Hasil Enkripsi

Untuk mengetahui apakah hasil pengujian enkripsi berlangsung dengan baik, maka pada penelitian ini akan dibuat sebuah tabel uji coba yang akan ditampilkan pada tabel dibawah ini.

Tabel 3.8 bahan pengujian

Nama *File*	Ukuran (*bytes*)	Tipe *File*
Perancangan.docx	1,111 KB	World *document*
Perancangan.pdf	2,222 KB	Pdf *document*

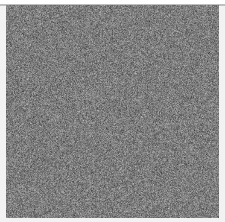
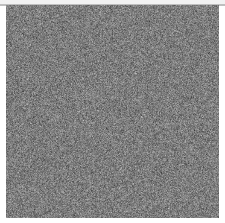
Tabel 3.9 hasil pengujian enkripsi

Nama *File*	Kunci	*Chiper Image*
Perancangan.docx	K = 111 p = 222 q = 333	
Perancangan.pdf	K = 444 p = 555 q = 666	

3.7.6 Pengujian Hasil Dekripsi

Untuk mengetahui apakah hasil pengujian Dekripsi berlangsung dengan baik, maka pada penelitian ini akan dibuat sebuah tabel uji coba yang akan ditampilkan pada tabel dibawah ini.

Tabel 3.10 bahan pengujian (chiper image)

Nama *File*	Kunci	*Chiper Image*
Perancangan.docx	K = 111 p = 222 q = 333	
Perancangan.pdf	K = 444 p = 555 q = 666	

Tabel 3.11 hasil dekripsi

Nama *File*	File dekripsi
21313143423333_ Perancangan	Hasil Dekripsi dari chiper image 20180804020843_percobaan_docx.png 
34324234234234_ Perancangan	Hasil Dekripsi dari chiper image 20180804021045_percobaan_pdf.png 

BAB IV
HASIL DAN PEMBAHASAN

4.1 Implementasi Perangkat Lunak

Tahap implemetasi perangkat lunak adalah tahap pengimplementasian aplikasi web yang sudah dibuat dapat dioperasikan dengan sempurna dan mengetahui apakah aplikasi web ini sudah sesuai dengan tujuan yang diharapkan atau perlu ada perbaikan terhadap aplikasi web ini.

Dalam bab ini juga akan dibahas mengenai implementasi algoritma dan ujicoba hasil. Hal ini dilakukan untuk mengetahui ciri, efektifitas dan tingkat ketahanan algoritma yang sudah dirancang terhadap berbagai macam serangan.

Pengujian hasil program aplikasi web ini dilakukan oleh *system* operasi *windows* dan dilakukan pada *computer localhost* (*intranet*). Pada aplikasi website ini terdapat dua sistem bagian yaitu bagian administrator dan *user*. Administrator di aplikasi website ini memiliki fungsi sebagi pengelola data *user*, menu aplikasi website untuk administrator dibagi menjadi tiga bagian yaitu menu *home*, menu *user* dan menu *logout*. *User* di aplikasi webisite ini memiliki fungsi untuk melakukan eksekusi enkripsi dan dekripsi, menu aplikasi website untuk *user* dibagi menjadi empat bagian yaitu menu *home*, menu *profile*, menu *encryption*, menu *decryption* dan menu *logout*.

4.2 Implementasi Algoritma

Implementasi algoritma ini mengacu pada perancangan sistem yang sudah dibahas pada bab sebelumnya maka pada bab ini akan dilakukan implementasi terhadap contoh kasus, sehingga dapat dilihat proses algoritma, mulai dari enkripsi sampai dekripsi secara keseluruhan.

4.2.1 Algoritma Randomized Text

Dibawah ini akan dijelaskan persamaan enkripsi dan dekripsi dari algoritma *Randomized Text* adalah sebagai berikut:

Persamaan enkripsi:

$C1 = K + 2P + R$

$C2 = 2K + P + R$ (4.1)

Persamaan dekripsinya:

$P = (C1-K) – (C2-2K)$ (4.2)

Dimana :

K = Kunci R = Nilai Random

P = *Plaintext* C = *Chipertext*

Dibawah ini akan dijelakan implementasi dari rumus *Randomized Text* berdasarkan contoh kasus berikut:

Plaintext = {A,P,L,I,K,A,S,I}

Dalam *ASCII* = {65,80,76,73,75,65,83,73}

Kunci = (5,10,15)

Tabel 4.1 percobaan pertama

P	**K**	**R**	**C1 = K + 2P + R Mod 256**	**C2 = 2K + P + R Mod 256**
65	5	14	149	89
80	10	24	194	124
76	15	31	198	137
73	5	16	167	99
75	10	32	192	127
65	15	11	156	106
83	5	9	180	102
73	10	53	209	146

Tabel 4.2 percobaan kedua

P	**K**	**R**	**C1 = K + 2P + R Mod 256**	**C2 = 2K + P + R Mod 256**
65	5	47	182	122
80	10	77	247	147
76	15	34	201	140
73	5	11	162	94
75	10	89	249	184
65	15	55	216	150
83	5	61	232	154
73	10	88	244	181

Pada percobaan tabel 4.1 dan 4.2, menghasilkan angka yang berbeda walaupun menggunakan nilai *plaintext* dan kunci yang sama. Antara percobaan tabel 4.1 dan 4.2 dapat dilihat tidak ada keterkaitan jika dilihat secara sekilas, akan tetapi kesamaan dari kedua percobaa di atas dapat dilihat dengan cara membandingkan C1 dan C2 pada table 4.1 dengan C1 dan C2 pada tabel 4.2. Kesamaan akan didapat dengan rumus |C1-C2| untuk mencari selisih antara C1 dan C2.

Tabel 4.1 = |C1-C2| = |149-89| = 60

Tabel 4.2 = |C1-C2| = |182-122| = 60

Selisih perbedaan antara tabel 4.1 dan tabel 4.2 memiliki nilai yang sama yaitu 60. Dari hasil tersebut dapat disimpulkan bahwa setiap percobaan akan menghasilkan nilai yang berbeda, akan tetapi pada setiap percobaan tersebut masih memiliki pola yang bisa mengidentifikasikan nilai yang dihasilkan dari plaintext dan kunci yang sama.

4.2.2 Algoritma Arnold Cat Map

Dibawah ini akan dijelaskan persamaan iterasi algoritma *Arnold Cat Map* adalah sebagai berikut:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \bmod(N) \tag{4.3}$$

Persamaa rumus diatas khusus digunakan untuk transformasi array 2 dimensi, dalam penelitian ini diolah dan diproses dalam bentuk array 1 dimensi, sehingga ada modifikasi dari persamaan rumus diatas agar bisa digunakan untuk transformasi array 1 dimensi. Langkah-langkah untuk mendapatkan nilai acak dari array 1 dimensi adalah sebagai berikut:

1. Masukkan beberapa inputan berupa sebuah *byte* simpan sebagai C
2. Hitung panjang dari *byte* tersebut simpan sebagai L
3. Pilih dua buah integer positif dan simpan sebagai P dan Q
4. Hitung akar dari L dan bulatkan hasilnya kebawah kemudian tambahkan dengan 1 simpan sebagai N
5. Hitung (1 * x) + (P * y) mod N simpan sebagai MX
6. Hitung (Q * x)+(((P * Q) + 1) * y) mod N simpan sebagai MY
7. Hitung (N * MX) + MY simpan sebagai MS
8. Buang matriks yang memiliki nilai MS<0 dan MS>L
9. Susun nilai yang tersisa, dan didapatlah nilai acak dari 1 sampai L
10. Ganti posisi dengan nilai acak yang sudah didapatkan simpan sebagai C

Contoh kasus:

C = {80, 121, 37, 14, 73, 47, 76, 52, 55, 69, 77}

L = 11

P = 188

Q = 268

N = $\sqrt{11}$ = 3,31 = 3 + 1 = 4

MX = (1 * x) + (P * y) mod N

Tabel 4.3 perhitungan MX

Nilai y →

Nilai x ↓

	1	2	3	4
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	0	0	0	0

MY = (Q * x)+(((P * Q) + 1) * y) mod N

Tabel 4.4 perhitungan MY

Nilai y →

Nilai x ↓

	1	2	3	4
1	1	2	3	0
2	1	2	3	0
3	1	2	3	0
4	1	2	3	0

MS = (N * MX) + MY

Tabel 4.5 perhitungan MS

Nilai y →

Nilai x ↓

	1	2	3	4
1	5	6	7	4
2	9	10	11	6
3	13	15	16	12
4	1	2	3	0

Eliminiasi nilai MS = MS<0 dan MS>L

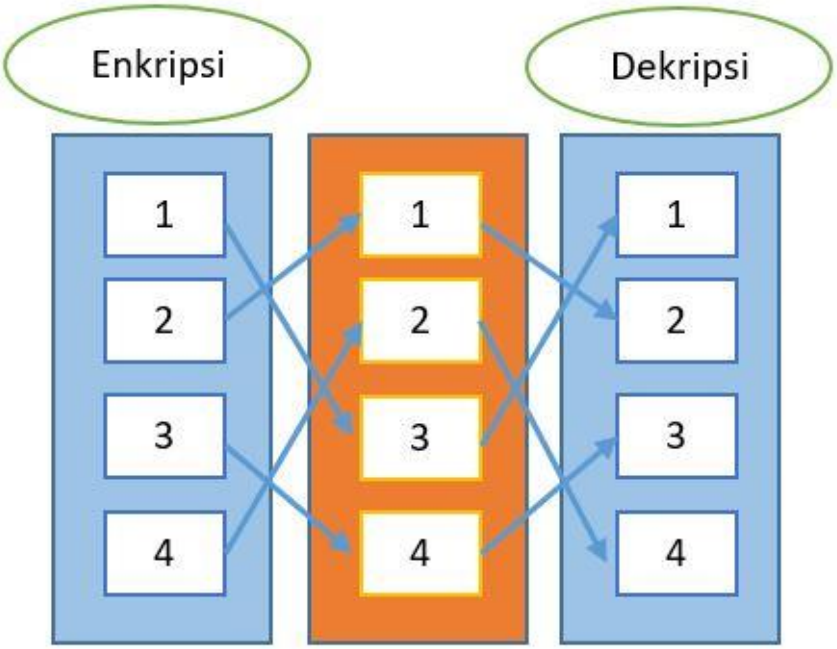
Tabel 4.6 eliminasi MS

Nilai y →

Nilai x ↓

	1	2	3	4
1	5	6	7	4
2	9	10	11	6
3				
4	1	2	3	

Susunan nilai acaknya berturut-turut adalah {5, 6, 7, 4, 9, 10, 11, 6, 1, 2, 3}. Setelah mendaptakan nilai acak dari proses perhitungan ACM tersebut, selanjutnya akan dilakukan



Gambar 4.1 skema pengacakan Arnold Cat Map

perpindahan index. Skema pengacakan index ACM ini bisa dilihat pada Gambar 4.1 berikut ini:

Ketika proses enkripsi dilakukan, index ke-1 dari awal akan dipindah menjadi index ke-3 dari hasil. Kemudian sebaliknya ketika proses dekripsi dilakukan, index ke-3 dari hasil akan dipindah menjadi index ke-1 dari awal. Pengacakan ACM dijelaskan pada contoh dibawah ini:

Jika C = {80, 121, *37*, 14, 73, *47*, *76*, 52, 55, 69, 77}

Dan posisi acaknya = {5, 6, 7, 4, 9, 10, 11, 6, 1, 2, 3}

Maka C’ = {73, 47, 76, 14, 69, 55, 77, 80, 52, 121, 37,}

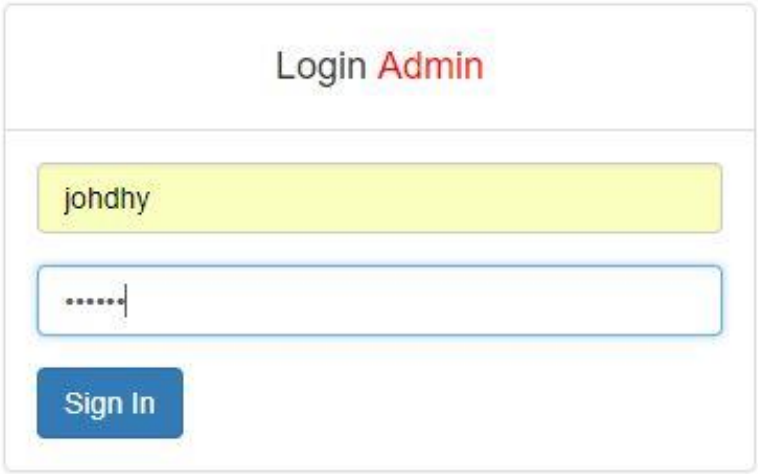
4.3 Implementasi Antarmuka

Implementasi antarmuka ini mengacu pada rancangan yang sudah ada di BAB III. Pengimplementasian antarmuka aplikasi ini dibagi menjadi dua yaitu antarmuka administrator dan antarmuka *user*.

Implementasi antarmuka administrator meliputi, antarmuka *login* admin, amtarmuka *home* admin, antarmuka *user* (delete *user*, edit *user*, lihat *file user*) dan antarmuka *logout*. Implementasi antarmuka *user* meliputi, antarmuka *login user*, antarmuka daftar *user*, antarmuka *home user*, antarmuka *profile user* (edit *email user*, edit *password user*), antarmuka enkripsi, antarmuka dekripsi dan antarmuka *logout*. Untuk lebih jelasnya implementasi dapat dilihat dibawah ini:

4.3.1 Antarmuka *Login* Admin

Pada halaman *login* admin ini berisi *username* dan *password* yang sudah di cocokan oleh *database* admin. Setelah proses mengisi *form login* admin ini selesai, maka akan di arahkan untuk mengklik tombol *button Sign In* agar dapat diproses untuk masuk kedalam halaman berikutnya.

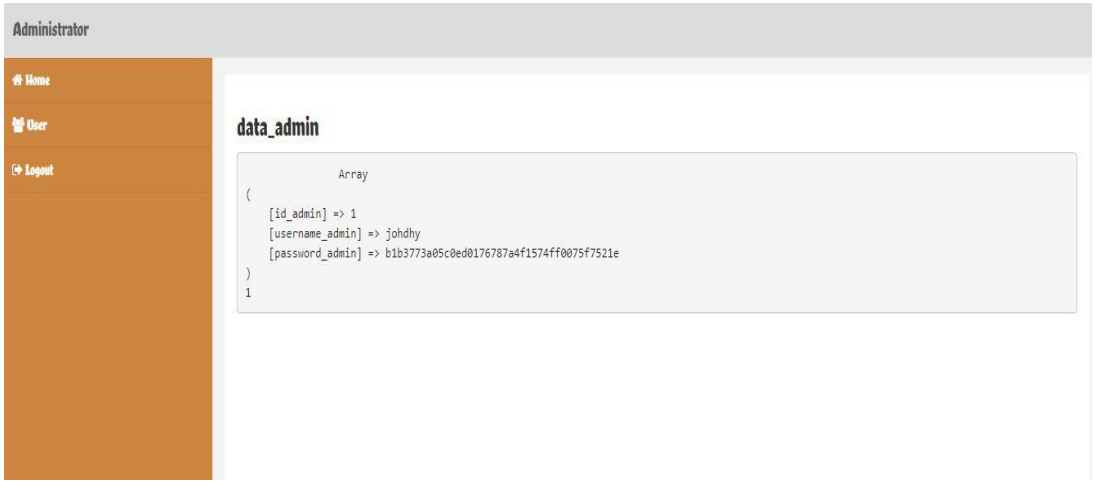


Gambar 4.2 antarmuka *login* admin

Gambar 4.3 pesan input sukses

Gambar 4.3 menunjukan pesan input berhasil ketika data *login* dari admin sesuai dengan yang ada di *database* admin.

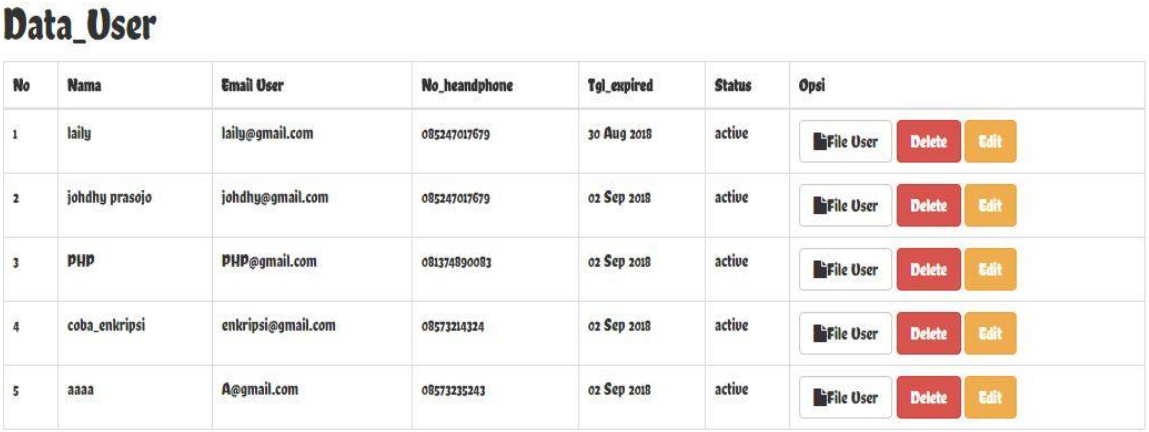
4.3.2 Antarmuka *Home* Admin



Gambar 4.4 antarmuka *home* admin

Pada halaman *home* admin ini terdapat tiga tampilan menu dan tampilan data admin. Tampilan menu tersebut meliputi, menu *home*, menu *user* dan menu *logout*.

4.3.3 Antarmuka Data *User*



Gambar 4.5 antarmuka data *user*

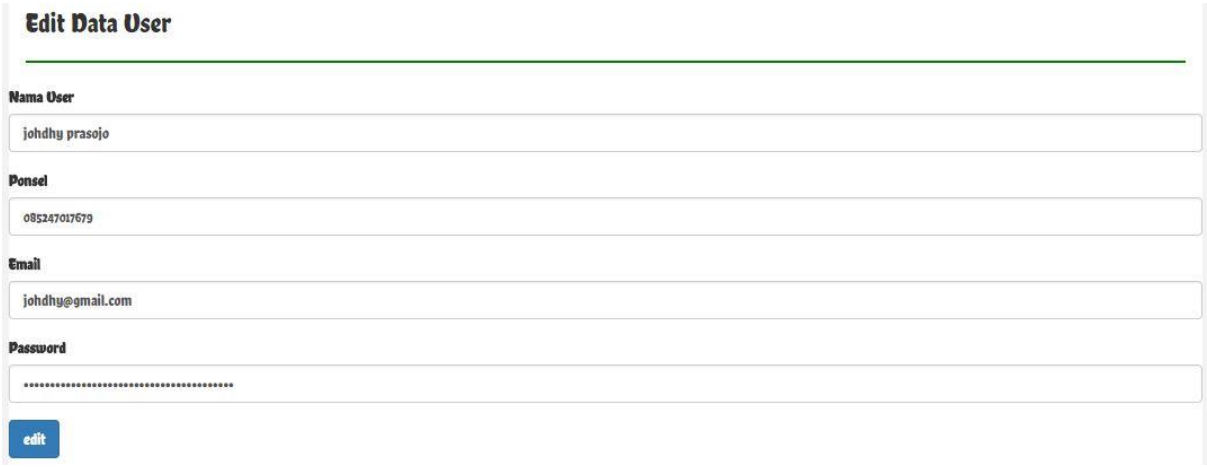
Gambar 4.5 merupakan tampilan data *user* yang meliputi nama, email *user*, no *heandphone* tgl *expired*,status dan tiga pilhan opsi (*file user*, *delete*, edit). Pada gambar di atas admin mempunyai hak akses untuk edit data *user*, hapus *user* dan melihat *file user*.

A. Menghapus *User*

Gambar 4.6 hapus data *user*

Gambar 4.6 merupakan pesan menghapus data *user*. Data *user* dapat terhapus ketika admin mengklik tombol *button* delete yang ada pada tampilan data *user*, ketika data sudah terhapus maka akan di arahakan kembali ke hamalaman data *user*.

B. Edit *User*



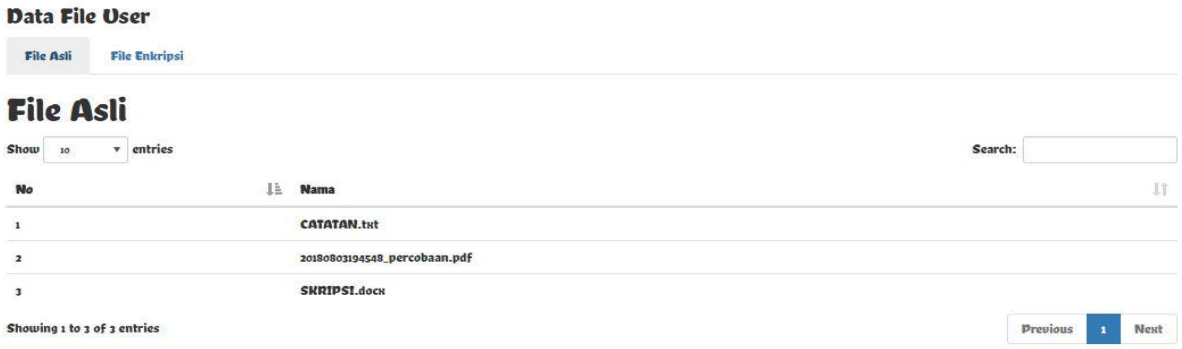
Gambar 4.7 halaman edit *user*

Pada halaman ini edit *user* ini admin dapat mengubah data *user* dengan cara mengisi *form* yang telah tersedia. Setelah admin sudah mengisi *form* edit *user*, akan diteruskan proses edit dengan mengklik *button* edit.

Gambar 4.8 data diganti

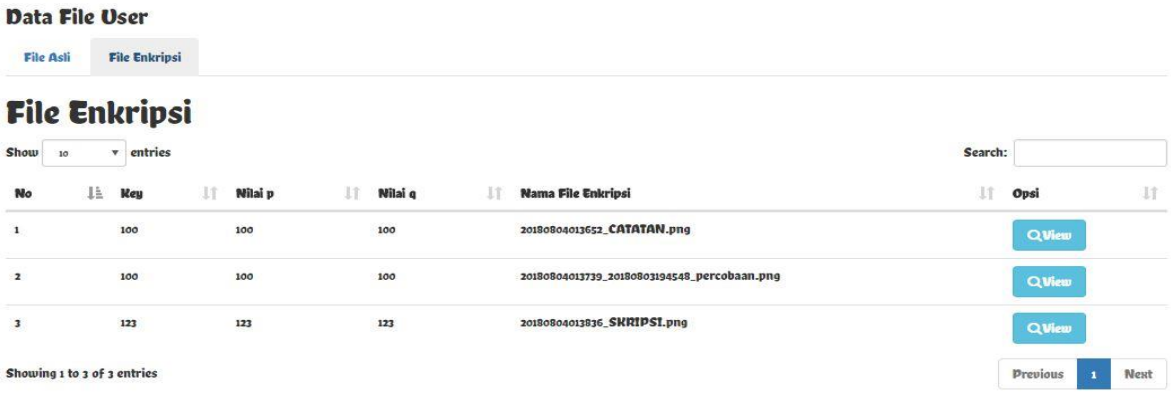
Gambar 4.8 merupakan pesan bahwa data telah terganti, setelah data terganti dan mengklik OK maka akan diteruskan menuju halaman data *user*.

C. *File User*



Gambar 4.9 lihat *file user* enkripsi

Pada halaman ini lihat *file user* ini terbagi menjadi dua bagian data *file*, yaitu *file* asli dan *file* enkripsi. Untuk *file* enkripsi terdapat tombol *button view* yang nantinya admin akan dapat melihat data *file* enkripsi.



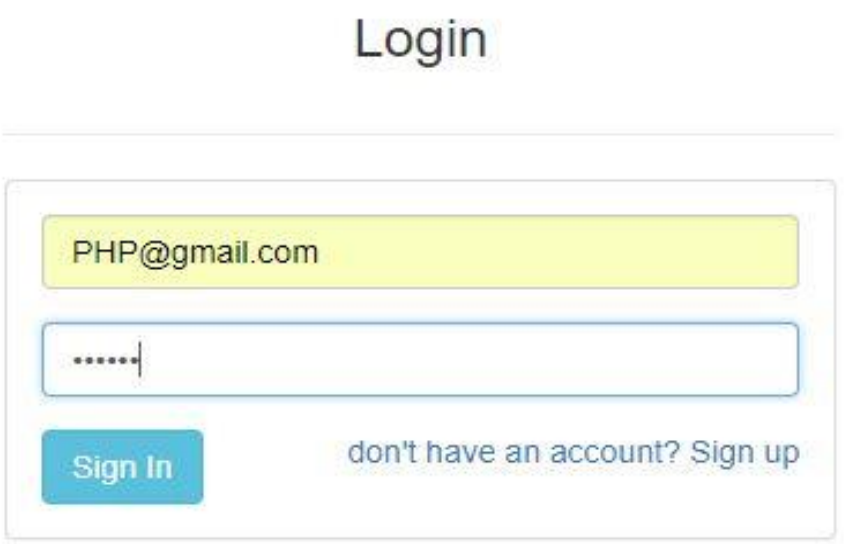
Gambar 4.10 lihat *file user* asli

4.3.4 Antarmuka *Logout*

Gambar 4.11 *logout*

Setelah admin selesai mengeksekusi hak aksesnya, admin akan di arahkan menuju menu *logout*. Gambar 4.11 merupakan pesan bahwa admin berhasil *logout* dan akan di arahkan menuju halaman *login* kembali.

4.3.5 Antarmuka *Login User*



Gambar 4.12 *login user*

Pada halaman *login user* ini berisi *username* dan *password* yang sudah di cocokan oleh *database user*. Setelah proses mengisi *form login user* ini selesai, maka akan di arahkan untuk mengklik tombol *button* Sign In agar dapat diproses untuk masuk kedalam halaman berikutnya.

Pada gambar 4.12 terdapat pilihan *button sign up*, jika *user* belum mempunyai akun data aplikasi web ini maka *user* diwajibkan untuk mendaftar terlebih dahulu agar dapat masuk kehalaman aplikasi web ini.

Gambar 4.13 *login* sukses

Gambar 4.13 menunjukan pesan sukses jika data *login* dari *user* sesuai dengan yang ada di *database user*.

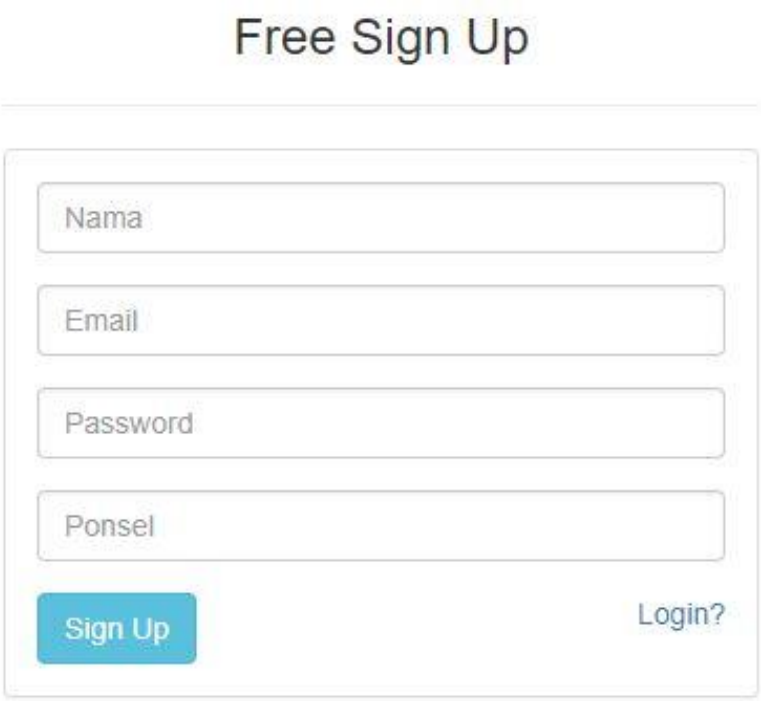
Gambar 4.14 *login* gagal

Gambar 4.14 menunjukan pesan gagal jika data *login* dari *user* tidak sesuai dengan yang ada di *database user*. Setalah proses *login* gagal, *user* akan di arahkan kembali menuju halaman *login*.

Gambar 4.15 akun *expired*

Gambar 4.15 menunjukan pesan *expired* jika data *user* sudah melebihi 30 hari pemakaian aplikasi web ini, setelah *user* mendapatkan pesan tersebut *user* diwajibkan untuk mendaftar kembali.

4.3.6 Antarmuka Daftar *User*



Gambar 4.16 daftar *user*

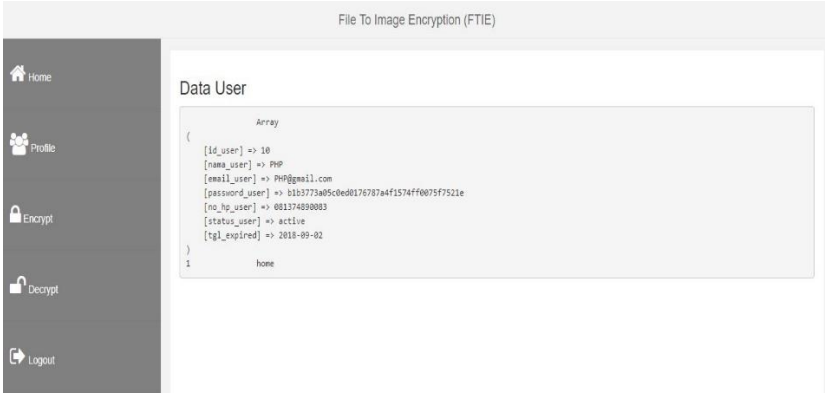
Pada halaman daftar *user* terdapat *form* yang akan diisi *user* untuk mendaftarkan akun sebagai pengguna aplikasi web ini. *User* akan diberi waktu selama 30 hari terhitung dihari saat

user mendaftarkan akun. Setelah *user* mengisi *form*, *user* akan di arahkan untuk mengklik tombol *button Sign up* untuk proses penyimpanan data kedalam *database*.

Gambar 4.17 data tersimpan

Pada gambar 4.17 menunjukan pesan bahwa *user* telah berhasil menyimpan data akun kedalam *database*. Setelah *user* mengklik *button* OK, *user* akan diproses menuju halaman *login*.

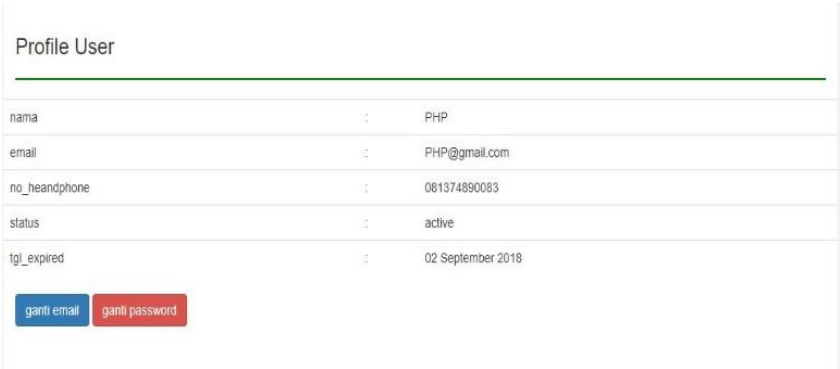
4.3.7 Antarmuka *Home User*



Gambar 4.18 *home user*

Pada halaman *home user* ini terdapat lima tampilan menu dan tampilan cara pemakaian aplikasi enkripsi dan dekripsi. Tampilan menu tersebut meliputi, menu *home*, menu *profile*, menu *encryption*, menu *decryption* dan menu *logout*.

4.3.8 Antarmuka *Profile User*



Gambar 4.19 *profile user*

Pada gambar 4.19 merupakan tampilan *profile user* yang telah menggunakan aplikasi web ini. Didalam halaman *profile user* ini terdapat dua pilihan *button* yaitu, *button* ganti email dan ganti *password*. Salah satu tujuan dibuatnya tampilan *user profile* ini, agar *user* tau tgl *expired* yang telah ditentukan oleh aplikasi web ini.

A. Edit Email *User*

Gambar 4.20 edit email *user*

Gambar 4.21 merupakan tampilan edit email *user*. *User* dapat mengisi *form* edit untuk mengganti email lama menjadi email baru yang diinginkan oleh *user*.

Gambar 4.21 data diganti

Gambar 4.21 merupakan pesan bahwa data telah terganti, setelah data terganti dan mengklik OK maka akan diteruskan menuju halaman *profile user*.

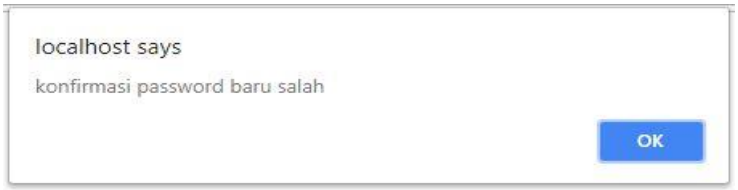
B. Edit *Password User*

Gambar 4.22 ganti *password*

Pada halaman ganti *password* terdapat tiga isian *form* untuk mengganti *password* lama *user*. *User* diwajibkan untuk mengisi *form password* yang lama untuk dapat proses menuju *password* baru, jika *user* tidak bisa mengisi *form password* lama dengan benar akan muncul pesan seperti pada gambar 4.22.

Gambar 4.23 *password* salah

Jika *password* lama sudah benar terisi, akan dilanjutkan dengan pengisian *form password* baru. Sebelum proses berhasil *user* diwajibkan untuk mengisi *form* konfirmasi *password* baru, jika konfirmasi *password* tidak sesuai akan muncul pesan seperti pada gambar 4.23.



Gambar 4.24 konfirmasi *password* salah

Jika semua inputan *form password user* benar maka proses akan berhasil dan muncul pesan seperti pada gambar 2.24. setelah proses berhasil tersimpan di *database*, *user* akan diteruskan menuju halaman *profile user*.

Gambar 4.25 *password* berhasil diganti

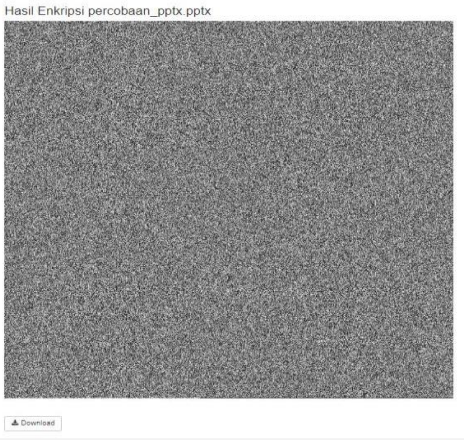
4.3.9 Antarmuka Enkripsi

Gambar 4.26 halaman enkripsi

Pada tampilan enkripsi ini merupakan halaman bagi *user* untuk mengeksekusi *file* yang ingin di enkripsi. *User* diwajibkan untuk mengisi *form* berupa angka dan dillanjutkan dengan memilih *file* yang ingin di enkripsi. Setalah *user* melakukan pengisian *form* dan memlilih *file*, *user* akan diarahkan untuk mengklik tombol *button encrypt* agar proses enkripsi dapat dilakukan. File yang telah di enkripsi akan terganti nama file sesuai dengan tanggal dan waktu saat user melakukan proses enkripsi.

A. *Download* Hasil Enkripsi

Pada tahap *download* ini, user akan diberikan arahan tombol *button download* dibawah hasil *chiper image* yang telah di enkripsi ini. Gambar 4.27 merupakan contoh hasil dari *chiper image* yang bisa di*download*.



Gambar 4.27 *download file chiper image*

4.3.10 Antarmuka Dekripsi

Gambar 4.28 halaman Dekripsi

Pada tampilan dekripsi ini merupakan halaman bagi *user* untuk mengeksekusi *file chiper image* yang ingin di dekripsi. *User* diwajibkan untuk memilih *file chiper image* dan dilanjutkan dengan mengisi *form* berupa angka. Setalah *user* melakukan memilih *file* dan mengisi *form*, *user* akan diarahkan untuk mengklik tombol *button encrypt* agar proses enkripsi dapat dilakukan. *File* yang telah di dekripsi akan terganti nama *file* sesuai dengan tanggal dan waktu saat *user* melakukan proses dekripsi.

A. *Download* Hasil Dekripsi

Pada tahap *download* ini, user akan diberikan arahan tombol *button download* dibawah hasil *file* asli yang telah di dekripsi ini. Gambar 4.29 merupakan contoh hasil dari *file* asli yang bisa di *download*.

Gambar 4.29 *download* hasil dekripsi

4.3.11 Antarmuka *Logout*

Gambar 4.30 berhasil *logout*

Setelah *User* selesai mengeksekusi aplikasi web ini, *user* akan di arahkan menuju menu *logout*. Gambar 4.30 merupakan pesan bahwa *user* berhasil *logout* dan akan di arahkan menuju halaman *login* kembali.

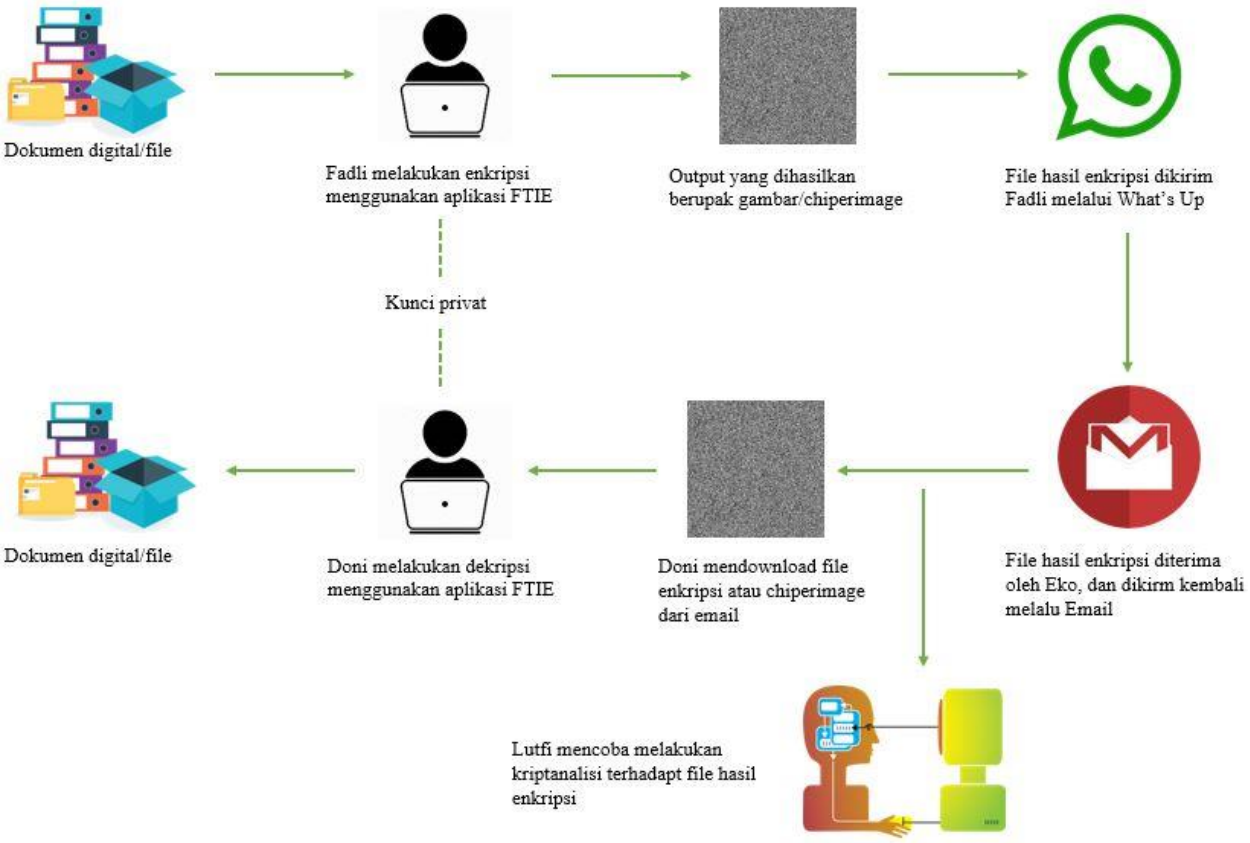
4.4 Skenario Aplikasi FTIE

Teknologi membuat pekerjaan dapat dikerjakan dengan cepat, akurat dan efisien sehingga banyak dokumen yang penting disimpan dalam bentuk digital. Keamanan dokumen digital tersebut harus dijaga baik dalam penyimpanan maupun dalam pengiriman, sehingga tidak bisa

disalahgunakan oleh pihak lain yang tidak berhak atas dokumen tersebut yang akan menyebabkan kerugian. Untuk menghindari penyalahgunaan oleh pihak lain maka setidaknya informasi yang ditransmisikan harus dienkripsi, agar informasi hanya bisa dibaca oleh pihak yang berhak atas informasi itu saja. Akan tetapi terkadang informasi yang sudah terenkripsi sekalipun masih bisa dipecahkan dikarenakan kurangnya keamanan sebuah teknik atau algoritma yang digunakan untuk melakukan enkripsi.

Aplikasi yang dibangun diharapkan dapat menjadikan dokumen digital/file yang dikirim melalui jaringan publik aman dari tindakan pencurian dan pengaksesan oleh pihak yang tidak berhak. Informasi berupa dokumen digital/file akan dienkripsi menjadi sebuah gambar dengan tujuan selain untuk mengamankan data atau informasi, perubahan ekstensi dapat mengecoh kriptanalis sehingga menganggap bahwa itu hasil enkripsi dari sebuah gambar, bukan hasil enkripsi dari sebuah dokumen digital.

Skenario pengiriman dokumen digital/file yang sudah terenkripsi dengan teknik FTIE menggunakan algoritma Randomized Text dan Arnold Cat Map (ACM) ditunjukkan oleh Gambar 4.31 berikut ini:



Gambar 4.31 skenario aplikasi FTIE

Pada gambar 4.31 merupakan gambaran skenario aplikasi FTIE dimulai dengan Fadli ingin mengirimkan chiper image hasil enkripsi yang memiliki informasi penting kepada Doni melalui jaringan sosial media yang rawan terhadap penyadapan atau kriptanalisis. Untuk menjaga keamanan dan kerahasiaan pada dokumen digital/file tersebut yang dikirim maka Fadli sebelumnya melakukan enkripsi dengan teknik FTIE sehingga dokumen/file tersebut akan menjadi sebuah gambar. Diasumsikan Fadli tidak memiliki sosial media Doni dan hanya mempunyai no telpn Doni, maka Fadli akan meminta bantuan kepada Eko dengan mengirimkan file chiper image memalui What's Up (WA) dan Eko mengirimkan kembali kepada Doni dengan menggunakan via Email. Selama pengiriman gambar berlangsung terlihat Lutfi mendapatkan gambar pada proses pengiriman via Email, dan Lufti melakukan proses kriptanalisis untuk memecahkan informasi yang terkandung pada gambar, Lutfi tidak mengatahui apakah gambar tersebut merupakan hasil dari enkripsi sebuah gambar atau hanya cuma sekedar gambar biasa saja yang tidak memiliki arti ataukah gambar tersebut hasil dari enkripsi dari sebuah gambar maka Luffi tidak dapat memecahkan informasi data dari gambar tersebut. Setelah chiper image tersebut sudah diterima oleh Doni, doni akan membuka aplikasi FTIE dan melakukan proses dekripsi yang nantinya hasil chiper image akan menjadi dokumen/file asli.

4.5 Analisis Keamanan

Setelah dilakukan proses enkripsi dan dekripsi dari aplikasi, selanjutnya untuk mengetahui tingkat keamanan dari hasil tersebut maka akan dilakukan beberapa analisis dan pengujian. Ada 3 variabel analisis dan 3 variabel pengujian yang di ambil untuk mengetahui tingkat keamanan dari teknik FTIE menggunakan algoritma Randomized Text dan ACM yaitu:

a. Analisis Entropy.
b. Analisis Ruang Kunci.
c. Analisis Besar Ukuran Hasil Enkripsi.
d. Pengujian Hasil Enkripsi.
e. Pengujian Hasil Dekripsi.
f. Pengujian Kunci Dekripsi Salah.

4.5.1 Analisis Entropy

Analisis menunjukkan tingkat keacakan chipertext yang dihasilkan dari proses enkripsi. Makin besar nilai maka makin bagus kualitas keacakan chipertext tersebut. Keacakan chpierte

xt keacakan chipertext tersebut. Hasil analisis nilai ditunjukkan pada tabel 4.7 dibawah ini:

Tabel 4.7 analisis entropy

No	Nama	Entropy		
		File Asli	Enkripsi	Dekripsi
1	Percobaan.docx	6,0595529	7,52980087	6,0595529
2	Percobaan.jpg	7,8069376	7,91997447	7,8069376
3	Percobaan.mp4	7,9629075	7,93251324	7,9629075
4	Percobaan.pdf	7,2923661	7,61805211	7,2923661
5	Percobaan.pptx	5,2302836	7,97832931	5,2302836
6	Percobaan.rar	9,3205685	7,98278253	9,3205685

Tabel 4.7 menunjukkan bahwa nilai rata-rata entropy dari seluruh adalah 7,82690875. Berdasarkan hasil penelitian Jolfaei dan Mirghadri (2011) menyatakan bahwa, jika sebuah informasi dienkripsi dan dalam kondisi teracak sempurna, maka nilai entropy yang ideal adalah ≈ 8. Berdasarkan teori tersebut maka algoritma FTIE yang dirancang ini aman dari serangan atau sulit ditebak oleh kriptanalis karena nilai informasi yang terdapat didalam gambar telah berada dalam keadaan teracak.

4.5.2 Analisis Ruang Kunci

Pada penelitian ini, parameter kunci yang digunakan untuk melakukan enkripsi adalah kunci, nilai p, dan nilai q. Kunci menggunakan 256 karakter, nilai p dan nilai q kuncinya berupa angka antara 0 sampai 9. Diasumsikan minimal panjang karakter kunci (k) adalah 6 karakter, nilai p minimal 3 karakter, dan nilai q 3 karakter. Maka jumlah kemungkinan kunci adalah:

$= 256^6 * 10^3 * 10^3$

= 281.474.976.710.656 * 1000 * 1000

= 281.474.976.710.656.000.000

Jadi kemungkinan panjang kunci minimal untuk melakukan serangan bruteforce pada teknik FTIE adalah 281.474.976.710.656.000.000 kemungkinan. Jika diasumsikan sebuah komputer mampu melakukan perhitungan sebanyak 1.000 milyar per detik, maka waktu yang dibutuhkan untuk dapat membuka ciphertext dapat dihitung dengan melakukan pembagian antara jumlah kombinasi kunci dengan jumlah kemampuan komputer dalam melakukan komputasi:

Estimasi Waktu = $\frac{281.474.976.710.656.000.000}{1.000.000.000.000}$

= 281.475 detik

= 4.691 jam

= 195 hari

= 6.5 bulan

= $^{1}/_{2}$ tahun

Dari data percobaan perhitungan diatas, terlihat waktu yang dibutuhkan untuk melakukan percobaan dekripsi menggunakan metode bruterforce terhadap minimal kemungkinan kunci yang ada sangat lama atau tidak mungkin untuk dilakukan. Sehingga dapat diambil kesimpulan bahwa penerapan teknik FTIE dapat menghasilkan yang aman dari serangan bruteforce.

4.5.3 Analisis Waktu Enkripsi

Analisis ini menunjukkan jumlah waktu yang dibutuhkan untuk melakukan proses enkripsi dan dekripsi berdasarkan ukuran. Analisis dilakukan menggunakan temporary dengan ukuran dari 100KB sampai 5MB kemudian ada 2 pembagian waktu yaitu waktu enkripsi dan dekripsi, hasil dari ukuran waktu adalah detik. Pada analisis ini akan menggunakan nilai key, p, dan q sebesar 100, Hasil analisis dapat dilihat pada Tabel 4.8 dibawah ini.

Tabel 4.8 analisi waktu

Nama File	Ukuran File (Kb)	Waktu Enkripsi	Waktu Dekripsi
Percobaan.docx	2.260 KB	24.177	16.360
Percobaan.jpg	1.576 KB	17.850	10.681
Percobaan.mp4	4.533 KB	49.329	30.288
Percobaan.pdf	2.223 KB	23.328	14.086
Percobaan.pptx	327 KB	05.010	03.038
Percobaan.rar	3.212 KB	36.171	19.855

4.5.4 Analisis Besar Ukuran File Hasil Enkripsi

Analisis ini menunjukkan hasil ukuran size file setelah di enkripsi berdasarkan ukuran byte. Hasil enkripsi berupa sebuah gambar yang memiliki ukuran lebih besar dari ukuran file asli. Hasil analisis dapat dilihat pada Tabel 4.9 dibawah ini:

Tabel 4.9 analasis size hasil enkripsi

Nama File	Ukuran File Asli (Kb)	File PNG Hasil Enkripsi
Percobaan.docx	2.260 KB	11,837,793 byte

Nama File	Ukuran File Asli (Kb)	File PNG Hasil Enkripsi
Percobaan.jpg	1.576 KB	8,254,410 byte
Percobaan.mp4	4.533 KB	23,746,796 byte
Percobaan.pdf	2.223 KB	11,644,634 byte
Percobaan.pptx	327 KB	1,711,985 byte
Percobaan.rar	3.212 KB	16,824,204 byte

4.5.5 Pengujian Hasil Enkripsi

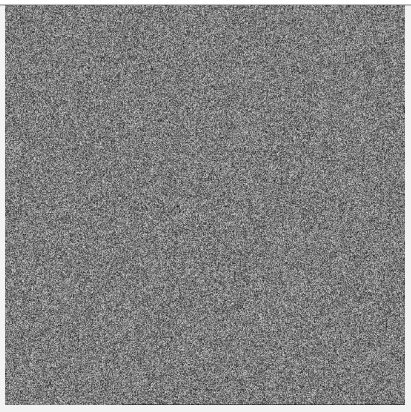
Adapun dokumen digital yang akan digunakan untuk bahan pengujian dapat dilihat pada Tabel 4.7 dibawah ini.

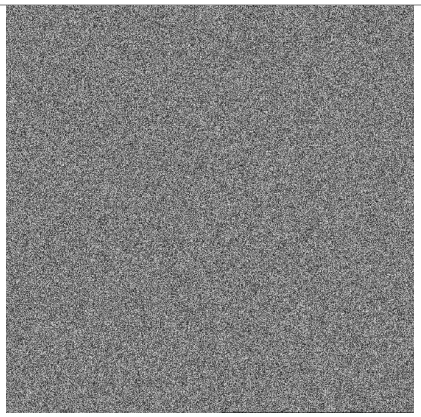
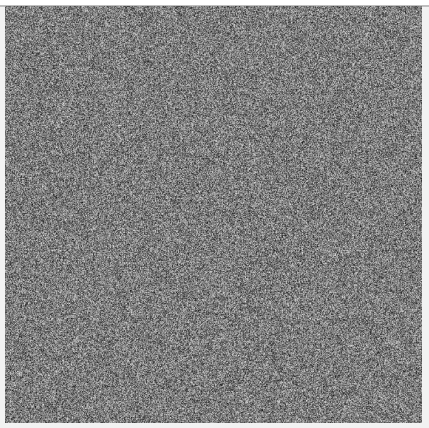
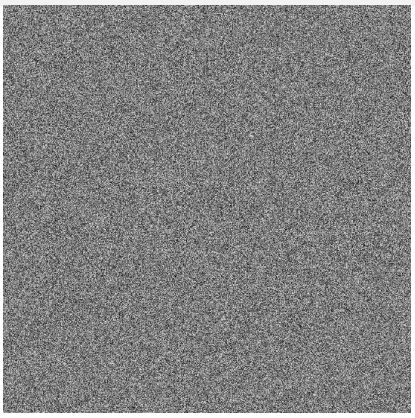
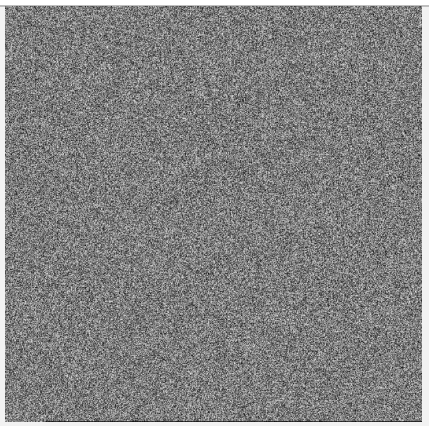
Tabel 4.10 bahan pengujian

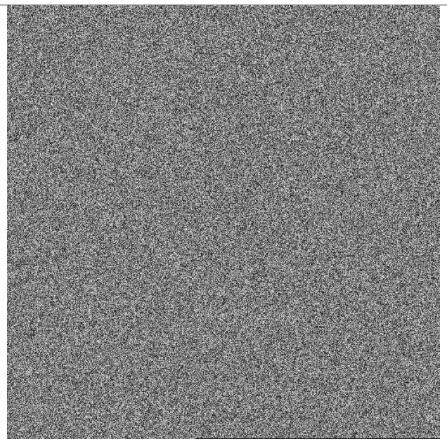
Nama *File*	Ukuran (*bytes*)	Tipe *File*
Percobaan docx	2.314.240	*World document*
Percobaan pdf	2.276.352	Pdf *document*
Percobaan mp4	4.641.792	MP4 *File*
Percobaan mp3	3.365.888	MP3 *format sounds*
Percobaan jpg	1.613.824	Jpg *File*
Percobaan pptx	334.848	Pptx *document*

Pada tabel dibawah ini merupakan hasil enkripsi dari tabel 4.8 untuk hasilnya sebagai berikut:

Tabel 4.11 hasil pengujian

Nama *File*	Kunci	*Chiper Image*
Percobaan docx	k = 100 p = 353 q = 123	

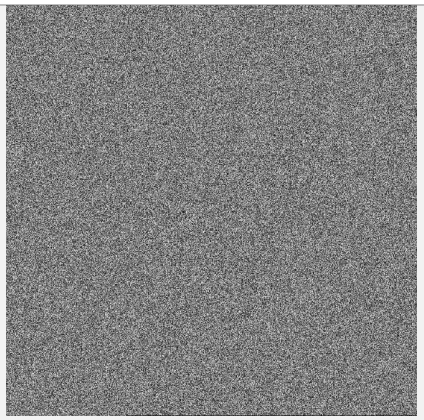
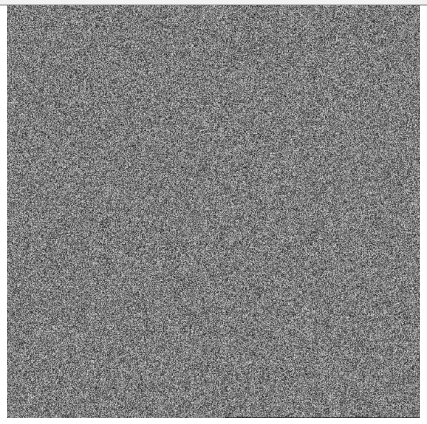
Nama *File*	**Kunci**	***Chiper Image***
Percobaan pdf	k = 40 p = 602 q = 432	
Percobaan mp4	k = 30 p = 121 q = 44	
Percobaan rar	k = 333 p = 222 q = 111	
Percobaan jpg	k = 100 p = 100 q = 100	

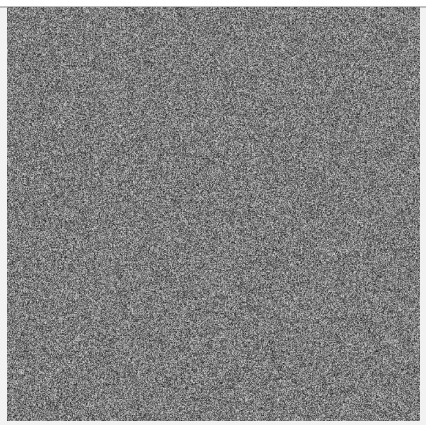
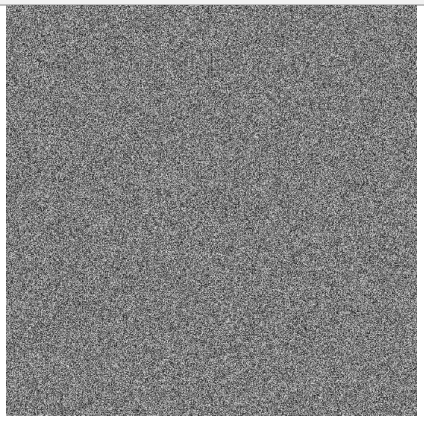
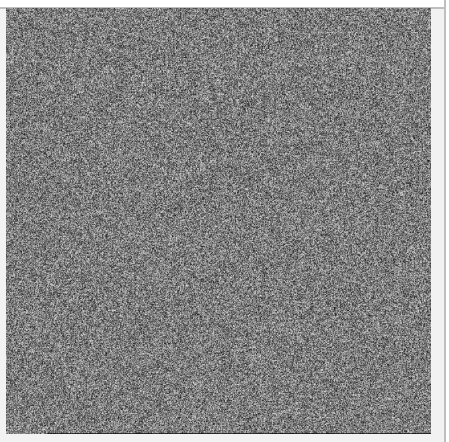
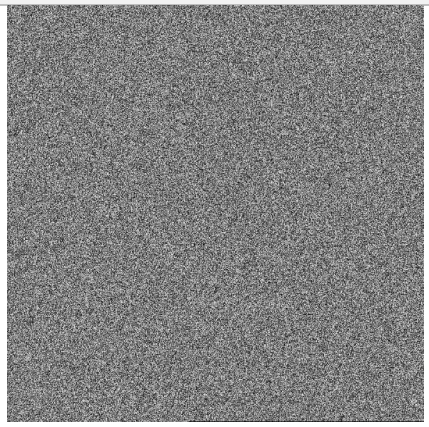
Nama *File*	Kunci	*Chiper Image*
Percobaan pptx	k = 321 p = 876 q = 123	

4.5.6 Pengujian Hasil Dekripsi

Adapun *file chiper image* yang akan digunakan untuk bahan pengujian dapat dilihat pada Tabel 4.9 dibawah ini:

Tabel 4.12 bahan pengujian (*chiper image*)

Nama *File*	Kunci	*Chiper Image*
20180804020843_percobaan_docx	k = 100 p = 353 q = 123	
20180804021045_percobaan_pdf	k = 40 p = 602 q = 432	

Nama *File*	Kunci	*Chiper Image*
20180804021211_percobaan_mp4	k = 30 p = 121 q = 44	
20180804021930_percobaan_rar	k = 333 p = 222 q = 111	
20180804021558_percobaan_jpg	k = 100 p = 100 q = 100	
20180804021502_percobaan_pptx	k = 321 p = 876 q = 123	

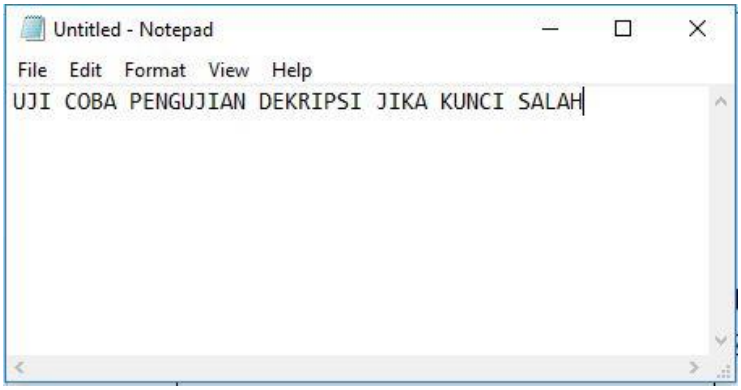
Pada tabel dibawah ini merupakan hasil dekripsi dari tabel 4.10 untuk hasilnya sebagai berikut:

Tabel 4.13 hasil dekripsi

Nama *File*	File dekripsi
20180804023729_percobaan	
20180804023905_percobaan	
20180804024503_percobaan	
20180804024740_percobaan	
20180804024822_percobaan	
20180804025001_percobaan	

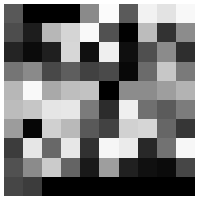
4.5.7 Pengujian Kunci Dekripsi Salah

P ada tahap ini akan dibuat pengujian jika kunci untuk proses dekripsi tidak sesuai dengan kunci pada saat proses enkripsi. Pengujian ini menggunakan contoh pada file .txt. gambar dibawah ini merupakan file untuk pengujian.



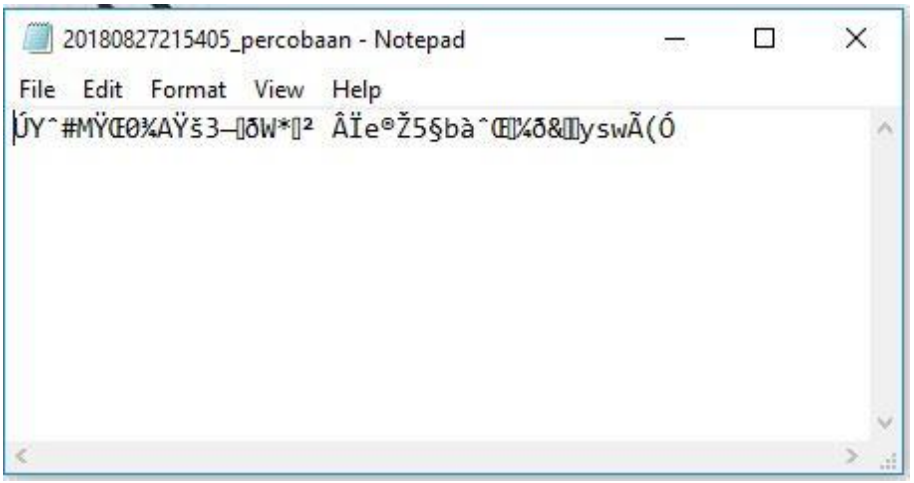
Gambar 4.32 pengujian file .txt

Ketika di enkripsi dengan nilai key (123), nilai p (321), dan nilai q (213) akan menghasilkan sebuah gambar chipertext seperti pada gambar dibawah ini:



Gambar 4.33 hasil enkripsi file .txt

Selanjutnya akan didekripsi kembali dengan menggunakan kunci yang berbeda dengan nilai key (200), nilai p (100), nilai q (300), maka akan menghasilkan file seperti gambar dibawah ini:



Gambar 4.34 hasil dekripsi file .txt dengan kunci berbeda

Kesimpulannya, semua file yang didekripsi dengan nilai berbeda tetap dapat melakukan proses dekripsi dan menjadi sebuah file, tetapi file tersebut tidak akan menjadi file asli sebelum dienkripsi. Pada proses perbedaan kunci ini berlaku untuk semua file.

4.6 Perbedaan Penelitian

Pada tabel 4.11 akan dijelaskan beberapa perbedaan penelitian yang telah dilakukan.

Tabel 4.14 perbedaan penelitian

Aplikasi *File To Image Encryption* (FTIE) menggunakan algoritma *Randomized Tex*t dan *Arnold Cat Map* (ACM) berbasis Website untuk keamanan data digital.	Aplikasi *File To Image Encryption* (FTIE) menggunakan algoritma *Randomized Tex*t dan *Arnold Cat Map* (ACM) untuk keamanan data digital.
Aplikasi ini berbasis website dengan menggunakan bahasa pemrograman PHP, *Html* dan *Javascript*.	Aplikasi ini berbasis desktop dengan menggunakan bahasa pemrograman C#.
Output yang dihasilakan merupakan suatu aplikasi website yang tujuannya dapat digunakan secara online.	*Output* aplikasi ini hanya bisa di gunakan pada persatuan komputer digital saja dan tidak bisa di onlinekan.
Lebih banyak mengulas implementasi program.	Lebih banyak melakukan analisis program.
Chiper image yang dihasilkan berupa pixel R, G, B hitam dan putih.	*Chiper image* yang dihasilkan berupa pixel R, G, B berwarna.
Tidak hanya melakukan proses enkripsi dan dekripsi, lebih banyak fitur didalam aplikasi website ini. Contohnya seperti penggunaan hak akses admin dan hak akses *user*.	Hanya dapat melakukan enkripsi dan dekripsi.
Semua proses enkripsi dan dekripsi tersimpan di *database*.	Semua proses enkripsi dan dekripsi langsung dari aplikasi.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil yang didapat dalam pembuatan tugas akhir aplikasi berbasis website ini, maka dapat diambil kesimpulan sebagai berikut:

1. Aplikasi web FTIE yang dibangun akan menghasilkan gambar *chiper image* dari proses enkripsi dan *file* asli dari proses dekripsi.
2. Hasil enkripsi dengan menggunakan algoritma Arnold Cat Map (ACM) dan Randomized Text memiliki ukuran 2 kali lebih besar dari ukuran asli.
3. Nilai entropy untuk uji keamanan menghasilkan nilai rata-rata sebesar 7,82690875, yang menyatakan bahwa jika nilai hasil enkrpisi mendekati 8 maka hasil enkripsi tersebut dapat dinyatakan aman dari serangan atau sulit ditebak kriptanalisis.
4. Semakin besar size file atau dokumen tersebut, akan memakan waktu yang cukup lama untuk melakukan proses enkripsi dan dekripsi.
5. Proses enkripsi dan dekripsi berjalan baik, proses enkripsi menggunakan teknik FTIE memiliki ketahanan dan keamanan yang kuat terhadap kriptanalisis dan proses dekripsi menggunakan teknik FTIE memiliki integritas yang dapat dipertanggungjawabkan.

5.2 Saran

Dalam pembuatan aplikasi FTIE berbasis web yang dibangun masih terdapat beberapa kekurangan. Sehingga dapat ditarik saran untuk pengembangan selanjutnya adalah sebagai berikut:

1. Dapat menggunakan algoritma untuk mengacak besar size setelah dienkripsi, karena size file asli dan size file yang telah dienkripsi rata rata memiliki perbedaan yang sama.
2. Penelitian selanjutnya dapat menggunakan algoritma yang dapat melakukan kompresi karena hasil enkripsi dengan menggunakan algoritma *Randomized Text* dan *Arnold Cat Map* memiliki ukuran 2 kali lebih besar dari ukuran asli.
3. Proses waktu pada saat melakukan enkripsi dan dekripsi masih terlalu lama, dibutuhkan proses waktu yang lebih cepat agar hasil lebih maksimal.

DAFTAR PUSTAKA

Abidin, A. M., Hardianti, F. and Setiani, I. N. (2016) 'Analisa Dan Implementasi Proses Kriptografi Encryption-Decryption Dengan Algoritma Advanced Encryption Standard (Aes-128)', *Jurnal Sarjana Teknik Informatika, Keamanan Komputer*, p. `1-20.

Chrystanti, Y. C. and Wardati, I. U. (2011) 'Sistem Pengolahan Data Simpan Pinjam khusus Perempuan (SPP) Pada Unit Pengelola Kegiatan (UPK) Mitra Usaha Mandiri Program Nasional Pemberdayan Masyarakat Mandiri Perdesaan (PNPM-MPd) Kecamatan Pringkuku Kabupaten Pacitan', *Journal Speed – Sentra Penelitian Engineering dan Edukasi*, 3(1), pp. 44–61. Available at: https://anzdoc.com/journal-speed-sentra-penelitian-engineering-dan-edukasi-volu8d7f8685b8b7fd6cbf53ac00eaa9d33c71098.html.

Fithria, N. (2018) 'Jenis-Jenis Serangan terhadap Kriptografi', *Jurnal ATM*, (13506036).

Harahap, M. K. (2016) 'Analisis Perbandingan Algoritma Kriptografi Klasik Vigenere Cipher Dan One Time Pad', *Jurnal Nasional Informatika dan Teknologi Jaringan*, 1(1), pp. 61–64.

Hidayat, A. D. and Afrianto, I. (2017) 'Sistem Kriptografi Citra Digital pada Jaringan Intranet Menggunakan Metode Kombinasi Chaos Map dan Teknik Selektif', *Ultimatics*, 9(1), pp. 59–66.

Kromodimoeljo, S. (2009) *Teori & Aplikasi Kriptografi*.

Maurer, U. M. (1992) 'Conditionally-perfect secrecy and a provably-secure randomized cipher', *Journal of Cryptology*, 5(1), pp. 53–66. doi: 10.1007/BF00191321.

Munir, R. (2012a) 'Digital Menggunakan Kombinasi Dua Chaos Map Dan Penerapan Teknik Selektif', *Juti*, 10(2), pp. 89–95.

Munir, R. (2012b) 'Robustness Analysis of Selective Image Encryption Algorithm Based on Arnold Cat Map Permutation', *Proceedings of 3rd Makassar International Conference on Electrical Engineering and Informatics*, (December), pp. 1–5.

Nurdin Nurdin and Prayitno, A. (2017) 'ANALISA DAN IMPLEMENTASI KRIPTOGRAFI PADA PESAN RAHASIA MENGGUNAKAN ALGORITMA CIPHER TRANSPOSITION', *Jesik*, 3(1), pp. 1–11. Available at: nnurdin69@gmail.com.

Ronsen, P., Halim, A. and Syahputra, I. (2014) 'Enkripsi Citra Digital Menggunakan Arnold's Cat Map dan Nonlinear Chaotic Algorithm', *JSM STMIK Mikroskil*, 15(2), pp. 61–71.

Santi, R. C. N. (2010) 'Implementasi Algoritma Enkripsi Playfair pada File Teks', *Teknologi Informatika DINAMIK*, XV(1), pp. 27–33.

Sari, D. I. (2016) 'Perancangan Aplikasi Kompresi Citra Dengan Metode Run Length

Encoding Untuk Keamanan File Citra Menggunakan Caesar Chiper', *INFOTEK*, 1(2), pp. 43–47.

Shanthi and Palanisamy, D. V. (2014) 'A Novel Text to Image Encryption Technique by AES Rijndael Algorithm with Color Code Conversion', *IJETT*, 13(5), pp. 237–241.

Suhartanto, M. (2012) 'Pembuatan Website Sekolah Menengah Pertama Negeri 3 Delanggu Dengan Menggunakan Php Dan Mysql', *Journal Speed – Sentra Penelitian Engineering dan Edukasi*, 4(1), pp. 1–8. Available at: http://speed.web.id/ejournal/index.php/Speed/article/view/226.

Suprianto, A., Prayudi, Y. and Sugiantoro, B. (2017) 'File To Image Encryption (FTIE) Menggunakan Algoritma Randomized Text Dan Arnold Cat Map (ACM) Untuk Keamanan Transmisi Data Digital', *HADFEX (Hacking and Digital Forensics Exposed)*, (August), pp. 19–26.

Wicaksono, L. (2013) 'No Title', *Ketahanan Algoritma RSA Terhadap Brute Force Attack*.

Younes, M. A. B. and Jantan, A. (2008) 'Image Encryption Using Block-Based Transformation Algorithm', *International Journal of Computer Science*, 35(1), pp. 407–415.

Yuliano, T. (2007) 'Pengenalan PHP', *Ilmiu Komputer*, pp. 1–9.

LAMPIRAN

- **Source Code**
- **CDR**