APLIKASI PENGAMAN DATA MENGGUNAKAN ALGORITMA RIVEST CHIPHER 4 (RC4) DAN ALGORITMA RIVEST CHIPHER 6 (RC6) BERBASIS VISUAL BASIC

Disusun dan Diajukan Untuk Memenuhi Persyaratan Ujian Akhir Memperoleh
Gelar Sarjana Komputer Pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH:

NAMA : **AHMAD FAUZI POHAN**

NPM : **1624371061**

PROGARAM STUDI : **SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019

ABSTRAK

AHMAD FAUZI POUAN

APLIKASI PENGAM DATA MENGGUNAKAN ALGORITMA RIVEST CHIPER 4 (RC4) DAN RIVEST CHIPER (RC6) BERBASI VISUAL BASIC

2019

Keamanan data merupakan hal yang sangat penting dalam menjaga kerahasiaan informasi terutama yang berisi informasi sensitif yang hanya boleh diketahui isinya oleh pihak yang berhak saja. Perkembangan teknologi yang semakin pesat saat ini tidak hanya berdampak baik dalam memudahkan bertukar data dan mendapatkan informasi saja, namun hal ini juga bisa menyebabkan kerugian bagi pihak yang melakukan komunikasi karena semakin banyak cara yang bisa dilakukan oleh pihak-pihak yang tidak bertanggung jawab. Semakin banyak serangan yang mungkin terjadi dalam proses pertukaran data maupun mendapatkan informasi. Maka pada penelitian ini keamanan informasi yang digunakan yaitu dengan teknik kriptografi. Adapun algoritma kriptografi yang digunakan yakni kriptografi algoritma RC4 dan RC6. Kemudian informasi yang akan dienkripsi yakni berupa file teks. Dan objek penelitian ini diterapkan pada data karyawan di suatu perusahaan. Kemudian sistem tersebut diimplementasikan menggunakan bahasa pemrograman Visual Basic 2010. Dan hasil dari penerapan algoritma RC4 dan RC6 dengan Visual Basic 2010 yaitu algoritma tersebut dapat mengamankan data karyawan dengan enkripsi yang dihasil dari penerapan algoritma RC4 dan RC6.

Kata kunci: **Keamanan Data**, **Kriptografi, Algoritma RC4**, **Algoritma RC6**, **Visual Basic 2010, Data Karyawan.**

DAFTAR ISI

BAB 3: METODE PENELITIAN

BAB 4: HASIL PENELITIAN DAN PEMBAHASAN

BAB 5: PENUTUP

BAB I

PENDAHULUAN

1.1 Latar Belakang

Keamanan data merupakan hal yang sangat penting dalam menjaga kerahasiaan informasi terutama yang berisi informasi sensitif yang hanya boleh diketahui isinya oleh pihak yang berhak saja. Perkembangan teknologi yang semakin pesat saat ini tidak hanya berdampak baik dalam memudahkan bertukar data dan mendapatkan informasi saja, namun hal ini juga bisa menyebabkan kerugian bagi pihak yang melakukan komunikasi karena semakin banyak cara yang bisa dilakukan oleh pihak-pihak yang tidak bertanggung jawab.

Semakin banyak serangan yang mungkin terjadi dalam proses pertukaran data maupun mendapatkan informasi, perlu adanya suatu teknik atau metode agar meningkatkan keamanan informasi yaitu dengan teknik kriptografi. Kriptografi yaitu informasi yang ingin disampaikan dienkripsi terlebih dahulu menggunakan suatu kunci agar tidak dapat langsung diketahui maknanya. Namun, sekarang ini sudah banyak digunakan algoritma kriptografi untuk menyembunyikan suatu informasi.

Oleh karena itu, penulis membuat sebuah implementasi algoritma kriptografi yang sudah ada dan algoritma tersebut akan dimodifikasi bagian kuncinya dan mempertimbangkan solusi *encrypted end to end* pada perangkat dengan fitur keamanan sebagai fitur tambahan. Keamanan ditinjau dari ancaman keamanan pada

teknologi *secure message*, maka terdapat aspek aspek yang perlu diperhatikan, yaitu: *Privacy/ Confidentiality* Jaminan kerahasiaan data yang ditransmisikan. Ancaman terhadap confidentiality ini antara lain: *sniffer* (penyusup), keylogger (penyadap kunci), dll. Dapat diproteksi dengan enkripsi integrity jaminan keutuhan data sampai ke *end user*.

Adapun algoritma kriptografi yang digunakan yakni kriptografi algoritma RC4 dan RC6. Dilakukan modifikasi kunci pada algoritma RC4 dan RC6 agar pihak yang tidak berhak tetap kesulitan dalam membaca informasi yang akan dikirim ke pihak yang berhak walaupun sudah mengetahui algoritma yang dipakai. Dan informasi yang akan dienkripsi yakni berupa file teks.

1.2 Rumusan Masalah

Berdasarkan uraian dari latar belakang yang telah dipaparkan sebelumnya, Adapun rumusan masalah pada analisa ini adalah:

1. Apakah algoritma RC4 dan RC6 merupakan algoritma yang baik dalam menjaga keamanan data?
2. Bagaimana aplikasi ini dapat bermanfaat bagi pengguna untuk pengamanan data yang mempunyai rahasia tingkat tinggi?

1.3 Batasan Masalah

Untuk menghindari pembahasan diluar permasalahan maka perlu dilakukannya pembatasan masalah. Adapun batasan masalah pada penelitian ini, yaitu:

1. Implementasi program pada *VB.Net 2010*

2. Proses enkripsi dan dekripsi data hanya digunakan untuk data *text* berupa huruf angka dan *symbol*

1.4 Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut yaitu mengimplementasikan enkripsi dan dekripsi data pada data karyawan dengan Algoritma RC4 dan RC6 Berbasis Visual Basic.

1.5 Manfaat Penelitian

Berikut ini beberapa manfaat penelitian yang akan dibahasa dalam penelitian ini yaitu sebagai berikutt:

1. Aplikasi ini dapat di instal / dipasang di Laptop dan di PC sehingga lebih mudah untuk menampilkan project aplikasi ini.
2. Penelitian ini berguna untuk menambah pengetahuan serta memberikan pemahaman kepada penulis dan pembaca yang berkentingan tentang keamanan data dan informasi
3. Diharapkan pengguna dapat menggunakan hasil penelitian ini sebagai sarana untuk memberi keamanan terhadap informasi data karyawan berbasis visual basic.

1.6 Metode Penelitian

Dalam merancang aplikasi keamanan data menggunakan algoritma RC4 dan RC6 berbasis Visual Basic, terdiri dari beberapa tahap yaitu :

1. Rekayasa sistem, merupakan kegiatan untuk menentukan informasi / kebutuhan apa yang dibutuhkan oleh sistem yang akan dibuat.

2. Analisis sistem, dilakukan untuk memperoleh informasi tentang sistem, menganalisis data-data yang ada dalam sistem dan juga menganalisis point–point masalahh pada sistem terutama mengenai kelebihan dan kekurangan sistem. Adapun alat dan bahan yang digunakan adalah :
   a. VB.Net (Visual Basic.Net)
   b. Windows 10.
   c. Perangkat lunak pendukung seperti *GUI* dan *OOP*
   d. Pengembangan system dilakukan dengan menggunakan komputer yang berspesifikasi RAM 6.0 Gb, Intel Inside CORE i3 64-bit.
3. Perancangan (desain), Perancangan desain dilakukan sebagai pematangan lebih lanjut setelah proses analisis data. Desain dibuat dengan Adobe Photoshop CS6 dan di*import* dalam bentuk PNG. Hal ini sangat membantu perjalanan implementasi sistem.
4. Penulisan Program (*Coding*), mebuat program didalam aplikasi yang dibuat agar dapat berjalan sesuai dengan yang difungsikan, dengan menggunakan bahasa pemrograman java.
5. Pengujian Sistem (*Testing*), kegiatan untuk melakukan pengetasan system yang sudah dibuat, apakah sistem / program yang dibuat sudah bejalan sesuai dengan yang dibutuhkan, apabila sistem / program sudah berjalan maka program dapat digunakan.
6. Pemeliharaan (*maintenance*), kegiatan untuk memelihara program aplikasi yang telah dibuat, baik dalam updating data,menjaga sistem agar tidak terserang virus, *error*, dan data *corrupt*.

1.7 Sistematika Penulisan

Agar dapat memberikan gambaran yang jelas pada penulisan tugas akhir ini, maka penulis membaginya dalam beberapa bab sebagai berikut :

BAB I : PENDAHULUAN

Menjelaskan mengenai latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian dan sistematika penulisan Aplikasi Pengaman Data Dengan Menggunakan Rivest Code 6 Berbasis Visual Basic.

BAB II : LANDASAN TEORI

Bab ini menguraikan tentang teori-teori dasar dalam pembangunan aplikasi berbasis visual basic. Mulai dari pengertian apa itu visual basic hingga apa–apa saja yang dibutuhkan untuk memulai pemograman visual basic pada Aplikasi Pengamanan Data Dengan Menggunakan Rivest Code 6 Berbasis Visual Basic

BAB III : METODE PENELITIAN

Pada bab ini penulis menguraikan tentang Analisa masalah dan rancangan program, meliputi perancangan *layout* dan *gambaran* aplikasi yang akan di buat sebagaimana aplikasi itu dijalankan.

BAB IV : HASIL PENELITIAN DAN PEMBAHASAN

Bab ini menguraikan tentang bagaimana proses lanjutan dari perancangan sistem. Menguraikan secara detail bagian implementasi sistem apakah implementasi sesuai dengan perancangan sistem pada Aplikasi Penerapan Algoritma RC4 dan RC6 Berbasis Visual Basic

BAB V : PENUTUP

Dalam bab ini berisi kesimpulan dari penelitian dan saran-saran pada Aplikasi Penerapan Algoritma RC 4 dan RC 6 Berbasis Visual Basic

BAB II

LANDASAN TEORI

2.1 Kriptografi

Menurut Onno (2006: 1), Kriptografi adalah bidang ilmu pengetahuan yang mempelajari pemakaian persamaan matematika untuk melakukan proses penyandian data. Kriptografi mempunyai tujuan yaitu mengamankan isi data atau menjaga kerahasian informasi dari orang yang tidak berhak untuk mengetahui isi data tersebut. Agar isi data aman maka diperlukan teknik atau algoritma untuk mengamankannya seperti proses enkripsi dan dekripsi. Untuk dapat melakukan proses tersebut maka pengirim dan penerima harus mengetahui algoritma yang digunakan dan memiliki kunci yang sesuai.

Tingkat keamanan dari data sandi terhadap upaya proses deskirpsi secara paksa oleh orang yang tidak berhak ditentukan oleh kekuatan algoritma yang digunakan dan kerahasian kunci. Kekuatan algoritma yang digunakan untuk proses enkripsi dan deskipsi berhubungan erat dengan penggunaan persamaan matematika. Semakin banyak dan rumit perhitungan dari persamaan matematika yang digunakan maka data sandi semakin aman, menurut Alfred (1997: 2). Kerahasian kunci adalah bagaimana cara kunci tersebut disimpan dan didistribusikan kepada pihak yang berhak menerima data, karena kunci ini akan digunakan maka jika semakin rapi kunci

disimpan dan didistribusikan maka kunci sandi semakin aman, Berikut ini istilah yang berhubungan erat dengan kriptografi:

1. *Plaintext* adalah data asli atau informasi bersifat terbuka yangisinya dapat dibaca dan dipahami secara langsung.
2. *Chiphertext* adalah data sandi hasil dari proses dekripsi.
3. *Chiphe*r adalah algoritma mengubah plaintext menjadi ciphertext menggunakan persamaan matematika.
4. *Substitution Chipher* adalah algoritma mengubah *plaintext* menjadi *ciphertext* dengan cara mengganti menggunakan persamaan matematika tertentu.
5. *Transporsition Cipher* adalah algoritma mengubah *plaintext* menjadi *ciphertext* dengan cara menggeser menggunakan persamaan matematika tertentu.
6. *Block Cipher* adalah algoritma mengubah *plaintext* menjadi *ciphertext*untuk setiap blok data, jumlah data atau besarnya block adalah tertentu.
7. Kunci (*key*) adalah data atau nilai yang sangat spesifik yang diketahui oleh pengirim dan penerima yang berhak.
8. Enkripsi adalah proses yang digunakan untuk menyembunyikan *plaintext.*
9. Dekripsi adalah proses mengembalikan *ciphertext* menjadi *plaintext.*
10. Kriptosistem adalah system kriptografi yang didalamnya terdiri dari algoritma kriptografi, plaintext, *cipheryrxt*, *key*, dan unsur lain yang berpengaruh dalam system kriptografi.

11. *Code Breaking* adalah kegiatan untuk mengubah *ciphertext* menjadi pesan asli tanpa mengetahui kunci yang sesuai dengan cara mencoba-coba secara sistimatis.
12. *Cryptology* adalah ilmu matematika yang mendasari *crypthography* dan *code breaking*.

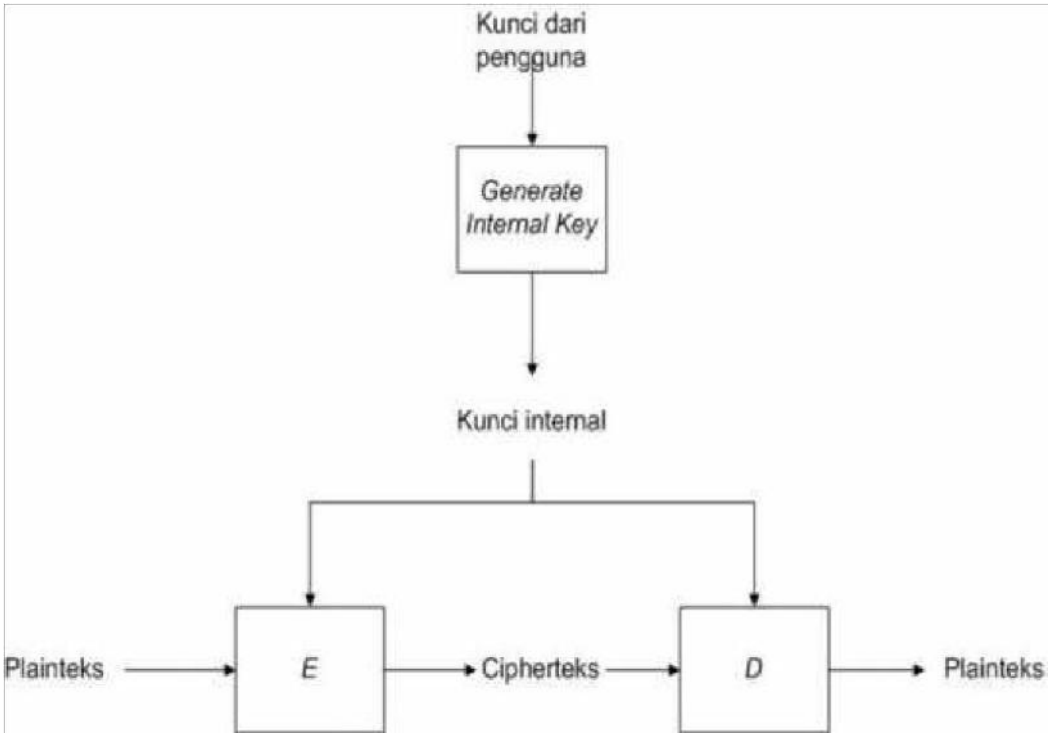
Keamanan informasi setelah dilakukan proses pengiriman dan penerimaan informasi maka dapat dilakukan tindakan – tindakan berikut ini:

1. Membuktikan keaslian adalah proses yang memungkinkan penerima informasi untuk mengetahui asal atau pengirim informasi yang sebenarnya.
2. Menjaga integritas data yaitu proses yang menjamin penerima informasi dapat memeriksa, apakah infomasi telah berubah sebelum diterima.
3. Pembuktian seseorang telah mengirim pesan adalah proses untuk menjamin pengirim informasi tidak dapat menyangkal bahwa dia telah mengirim informasi tersebut.
4. Menjaga kerahasian yaitu proses untuk menjamin informasi yang dikirim tidak dapat dipahami isinya oleh orang yang tidak berhak.

2.2 Block Cipher

Block cipher adalah suatu tipe algoritma kriptografi kunci simetris yang mengubah *plainteks* yang dibagi dalam blok-blok dengan panjang yang sama menjadi *cipherteks* yang memiliki panjang blok yang sama. Ukuran panjang blok dapat beragam bergantung kepada algoritma yang digunakan, ukuran yang sering digunakan adalah 64 bit dan menuju 128 bit. Seperti semua algoritma kunci

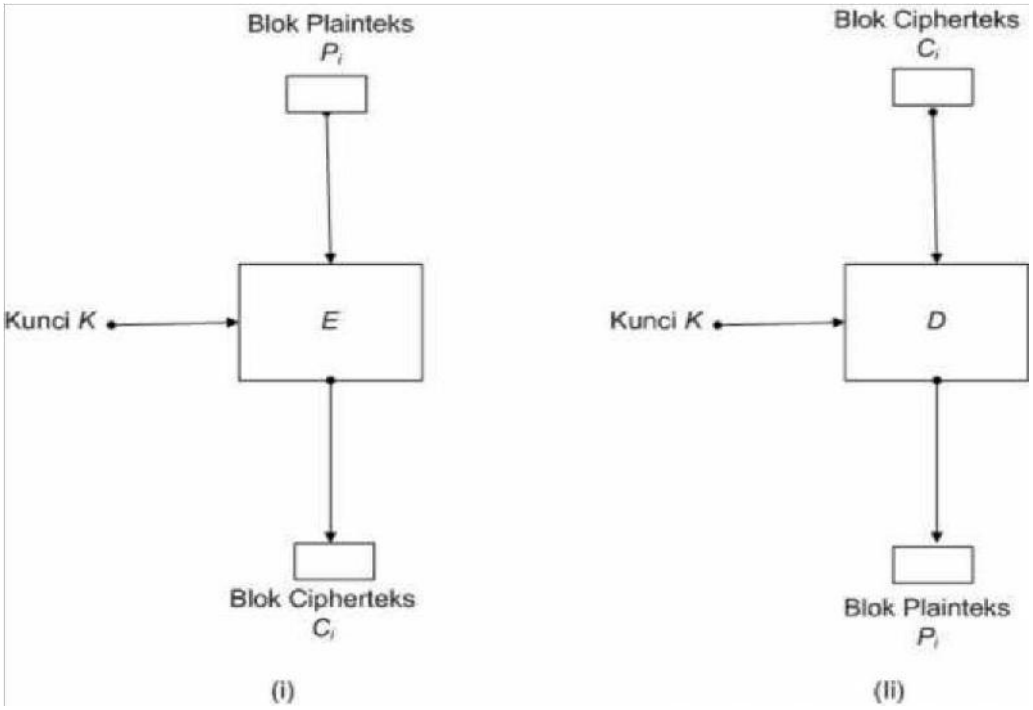
simetri, proses enkripsi yang dilakukan akan menggunakan suatu input dari user yang disebut sebagai kunci rahasia. Kunci rahasia ini juga akan dipakai ketika melakukan proses dekripsi. Cara kerja secara umum dari *block cipher* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Skema Cara Kerja Block Cipher

Dalam penggunaannya *block cipher* dikombinasikan dengan suatu teknik yang dinamakan mode operasi dari *block cipher*. Mode operasi yang sederhana dan sering digunakan adalah mode *Electronic Code Book* (ECB). Pada mode ECB setiap blok pada *plainteks* dienkripsi satu persatu secara independen. Hasil enkripsi masing-masing blok tidak mempengaruhi blok yang lain. Proses enkripsi pada mode ini sangat sederhana, setiap blok *plainteks* dienkripsi dengan fungsi enkrispsi secara terpisah. Seperti halnya dalam proses enkripsi, dalam proses dekripsi, masing-masing blok-blok *cipherteks* dikenakan dengan fungsi dekripsi secara

independen. Proses enkripsi dan dekripsi dari mode ECB dapat dilihat pada Gambar 2.2.



Gambar 2.2 (i) Proses enkripsi ECB dan

(ii) Proses dekripsi ECB pada sebuah blok

Dalam melakukan perancangan *block cipher*, beberapa prinsip harus dipertimbangkan. Prinsip-prinsip tersebut yaitu:

1. Prinsip *Confusion* dan *Diffusion.*

   Menurut Munir (2006: 4), Tujuan dari prinsip *confusion* adalah untuk menyembunyikan hubungan apapun yang ada antara *plainteks, cipherteks*, dan kunci, sehingga dapat membuat kripanalisis kesulitan dalam menemukan pola-pola pada *cipherteks*. Tujuan dari prinsip *diffusion* adalah menyebarkan pengaruh satu bit plainteks atau kunci ke sebanyak mungkin *cipherteks*,

sehingga dengan berubahnya satu bit *plainteks* dapat mengubah *cipherteks* yang sulit untuk diprediksi.

2. *Iterated Cipher*

   Untuk menambah keamanan, pada algoritma-algoritma *block cipher* dilakukan iterasi pada pemrosesan setiap blok, pada setiap rotasi dari iterasi tersebutdigunakan fungsi transformasi yang sama namun memakai kunci yang berbeda yang disebut dengan kunci internal. Kunci internal pada umumnya merupakan hasil dari kunci yang dimasukan oleh pengguna yang dikomputasi menggunakan suatu fungsi tertentu. Dengan adanya iterasi tersebut keamanan akan semakin terjamin, namun performansi akan berkurang karena adanya waktu lebih yang dibutuhkan untuk melakukan iterasi. *Block cipher* yang menerapkan konsep iterasi ini disebut juga dengan *iterated block cipher*.

3. Kunci Lemah

   Suatu hal yang perlu dihindari dalam melakukan perancangan algoritma kriptografi adalah kunci yang dapat menghasilkan *cipherteks* yang mirip atau serupa dengan *plainteks.*

2.3 Algoritma Rivest Code 4 (RC 4)

Algoritma kriptografi Rivest Code 4 (RC4) merupakan salah satu algoritma kunci simetris dibuat oleh RSA Data Security Inc (RSADSI) yang berbentuk *Stream Chipper*. Algoritma ini ditemukan pada tahun 1987 oleh Ronald Rivest dan menjadi simbol keamanan RSA (merupakan singkatan dari tiga nama penemu: Rivest, Shamir, Adleman). RC4 merupakan enkripsi *stream*

*simetrik proprietary*yang dibuat oleh *RSA Data Security Inc* (RSADSI). Penyebarannya diawali dari sebuah source code yang diyakini sebagai RC4 dan dipublikasikan secara '*anonymously*' pada tahun 1994. Algoritma yang dipublikasikan ini sangat identik dengan implementasi RC4 pada produk resmi. RC4 digunakan secara luas pada beberapa aplikasi dan umumnya dinyatakan sangat aman.

Sampai saat ini diketahui tidak ada yang dapat memecahkan/membongkarnya, hanya saja versi ekspor 40 bitnya dapat dibongkar dengan cara "*brute force*" (mencoba semua kunci yang mungkin). RC4 tidak dipatenkan oleh RSADSI, hanya saja tidak diperdagangkan secara bebas (*trade secret*). RC4 adalah salah satu bentuk stream cipher yang banyak digunakan pada protokol-protokol enkripsi, antara lain WEP, WPA, dan SSL/TSL.RC4 merupakan salah satu jenis *stream cipher*, yaitu memproses unit atau input data, pesan atau informasi pada satu saat. Unit atau data pada umumnya sebuah *byte* atau bahkan kadang kadang bit (*byte* dalam hal RC4).

Dengan cara ini enkripsi atau dekripsi dapat dilaksanakan pada panjang yang variabel. Algoritma ini tidak harus menunggu sejumlah input data, pesan atau informasi tertentu sebelum diproses, atau menambahkan byte tambahan untuk mengenkrip. Contoh *stream cipher*adalah RC4, Seal, A5, Oryx, dan lain-lain. Tipe lainnya adalah *block cipher* yang memproses sekaligus sejumlah tertentu data (biasanya 64 bit atau 128 bit blok), contohnya: Blowfish, DES, Gost, Idea, RC5, Safer, Square, Twofish, RC6, Loki97, dan lain-lain.

2.4 Algoritma Rivest Code 6 (RC 6)

Menurut Maman Abdurrohman (2002: 2), Algoritma RC6 merupakan salah satu kandidat *Advanced Encryption Standard* (AES) yang diajukan oleh *RSA Laboratoriest* kepada NIST. Dirancang oleh Ronald L Rivest, M.J.B.Robshaw, R. Sidney dan Y.L. Yin, algoritma inimerupakan pengembangan dari algoritma sebelumnya yaitu RC5 dan telah memenuhi semua kriteria yang diajukan oleh NIST. Algoritma RC6 adalah versi yang dilengkapi dengan beberapa parameter, sehingga dituliskan sebagai RC6-w/r/b, dimana parameter w merupakanukuran kata dalam satuan bit, r adalah bilangan bulat bukan negatif yang menunjukan banyaknya iterasi selam proses enkripsi, dan b menunjukanukuran kunci enkripsi dalam *byte*. Ketika algoritma ini masuk sebagai kandidat AES. maka ditetapkannilai parameter w = 32, r = 20 dan b bervariasiantara 16, 24, dan 32 *byte*.

RC6-w/r/b memecah *block* 128 bit menjadi 4buah *block* 32 bit, dan mengikuti enam aturanoperasi dasar sebagai berikut :

1. A + B Operasi penjumlahan bilangan integer.
2. A – B Operasi pengurangan bilangan integer.
3. A B Operasi *exclusive*-OR (XOR)
4. A x B Operasi perkalian bilangan integer.
5. A<<<B A dirotasikan ke kiri sebanyak variabel kedua (B)
6. A>>>B A dirotasikan ke kanan sebanyak variabel kedua (B)

2.4.1 Enkripsi Algoritma RC6

Karena RC6 memecah *Block* 128 bit menjadi 4 buah *block* 32 bit, maka algoritma ini bekerja dengan 4 buah register 32-bit A, B, C, D. *Byte* yang pertama dari *plaintext* atau *ciphertext* ditempatkan pada *byte* A. Sedangkan *byte* yang terakhirnya ditempatkan pada *byte* D. Dalam prosesnya akan didapatkan (A, B, C, D) = (B, C, D, A) yang diartikan bahwa nilai yang terletak pada sisi kanan berasal dari register disisi kiri. (Maman Abdurohman, 2002). Berikut ini adalah algoritma enkripsi RC6:

```
        B = B + S[ 0 ] D = D + S[ 1 ] for i = 1 to 20 do
{
        t = ( B x ( 2B + 1 ) ) <<< 5 u = ( D x ( 2D + 1 ) )
        <<< 5 A = ( ( A Å t ) <<< u ) + S[ 2i ]
        C = ( ( C Å u ) <<< t ) + S[ 2i+ 1 ]
        (A, B, C, D) = (B, C, D, A)
}

                A = A + S[ 42 ]
                C = C + S[ 43 ]
```

Algoritma RC6 menggunakan 44 buah sub kunci yang dibangkitkan dari kunci dan dinamakan dengan S[0] hingga S[43]. Masing-masing sub kunci panjangnya 32 bit. Proses enkripsi pada algoritma RC6 dimulai dan diakhiri dengan proses *whitening* yang bertujuan untuk menyamarkan iterasi yang pertama dan yang terakhir dari proses enkripsi dan dekripsi.Pada proses *whitening* awal, nilai B akan dijumlahkan dengan S[0], dan nilai D dijumlahkan dengan S[i]. Pada masing-masing iterasi pada RC6 menggunakan 2 buah sub kunci. Sub kunci pada iterasi

yang pertama menggunakan S[2] dan S[3], sedangkan iterasi-iterasi berikutnya menggunakan sub-sub kunci lanjutannya. Setelah iterasi ke-20 selesai, dilakukan proses *whitening* akhir dimana nilai A dijumlahkan dengan S[42], dan nilai C dijumlahkan dengan S[43].

Setiap iterasi pada algoritma RC6 mengikutiaturan sebagai berikut, nilai B dimasukan ke dalamfungsi f, yang didefinisikan sebagai f(x) =x(2x+1) ,kemudian diputar kekiri sejauh lg-w atau 5 bit.Hasil yang didapat pada proses ini dimisalkansebagai u. Nilai u kemudian di XOR dengan C danhasilnya menjadi nilai C. Nilai t juga digunakansebagai acuan bagi C untuk memutar nilainyakekiri. Begitu pula dengan nilai u, juga digunakan sebagai acuan bagi nilai A untuk melakukan prosespemutaran kekiri. Kemudian sub kunci S[2i] padaiterasi dijumlahkan dengan A, dan sub kunciS[2i+1] dijumlahkan dengan C. keempat bagiandari *block* kemudian akan dipertukarkan denganmengikuti aturan, bahwa nilai A ditempakan padaD, nilai B ditempatkan pada A, nilai C ditempatkanpada B, dan nilai (asli) D ditempatkan pad C. Demikian iterasi tersebut akan terus berlangsung hingga 20 kali.

2.4.2 Dekripsi Algoritma RC6

Proses dekripsi *ciphertext* pada algoritma RC6 merupakan pembalikan dari proses enkripsi.Pada proses *whitening*, bila proses enkripsimenggunakan operasi penjumlahan, maka padaproses dekripsi menggunakan operasi pengurangan.Sub kunci yang digunakan pada proses *whitening*setelah iterasi terakhir diterapkan sebelum iterasipertama, begitu juga sebaliknya sub kunci yangditerapkan pada proses *whitening* sebelum iterasipertama digunakan pada *whitening* setelah

iterasiterkahir. Akibatnya, untuk melakukan dekripsi, halyang harus dilakukan sematamata hanyalahmenerapkan algoritma yang sama dengan enkripsi, dengan tiap iterasi menggunakan sub kunci yang sama dengan yang digunakan pada saat enkripsi, hanya saja urutan sub kunci yang digunakan terbalik. Berikut ini adalah algoritma deskripsi RC6:

```
        C = C - S[ 43 ] A = A - S[ 42 ] for i = 20 downto 1 do
{
        (A, B, C, D) = (D, A, B, C) u = ( D x ( 2D + 1 ) )
        <<< 5 t = ( B x ( 2B + 1 ) )<<< 5 C = ( ( C - S[ 2i + 1 ] )
        >>> t ) ⊕ u
        A = ( ( A - S[ 2i ] ) >>> u) ⊕ t
}
                        D = D - S[ 1 ]
                        B = B - S[ 0 ]
```

2.5 Entity Relationship Diagram (ERD)

Menurut Brady dan Loonam (2010:3), *Entity Relationship Diagram* (*ERD*) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh *System Analys* dalam tahap analisis persyaratan proyek pengembangan sistem. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain *database* relasional yang mendasari sistem informasi yang dikembangkan. *ERD* bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk *database*.

Dapat disimpulkan bahwa *ERD* secara umum merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek

dasar data yang mempunyai hubungan antar relasi. *ERD* memodelkan struktur data dan hubungan antar data.

2.5.1 Komponen ERD

Dalam pembentukan *ERD* terdapat tiga komponen yang akan dibentuk, antara lain sebagai berikut :

1. Entitas

   Pengertian *entity* (entitas) yaitu suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Pengertian lainnya menurut Brady dan Loonam (2010: 4), entitas adalah objek yang menarik di bidang organisasi yang dimodelkan.

   Contoh : Mahasiswa, Kartu Anggota Perpustakaan (*KAP*) dan Buku.

2. Hubungan atau Relasi (*Relationship*)

   Suatu hubungan adalah hubungan antara dua jenis entitas dan direpresentasikan sebagai garis lurus yang menghubungkan dua entitas.

   Contoh: Mahasiswa mendaftar sebagai anggota perpustakaan (*KAP*), relasinya adalah mendaftar.

3. Atribut

   Atribut memberikan informasi lebih rinci tentang jenis entitas. Atribut memiliki struktur internal berupa tipe data.

2.5.2 Derajat relasi

Derajat relasi atau kardinalitas rasio merupakan penjelaskan mengenai jumlah maksimum hubungan antara satu entitas dengan entitas lainnya.

1. *One to One* (1:1)

   Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitu pula sebaliknya.

2. *One to many* (1:M)

   Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya.

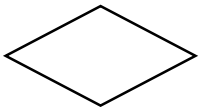
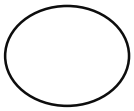
3. *Many to one* (M:1)

   Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B, tetapi tidak sebaliknya.

4. *Many to many* (M:M)

   Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya.

Tabel 2.1 Simbol-simbol Entity Relationship Diagram

Nama Simbol	Simbol	Keterangan
Entitas		Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai
Relasi		Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda
Atribut		Atribut berfungsi mendeskripsikan karakter entitas (atribut yang memiliki *key* diberi garis bawah)

Garis	_____	Garis sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut

2.6 Basis Data

Basis data (*database*) merupakan kumpulan dari beberapa data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi bagi para pemakai. Penerapan *database* dalam sistem informasi disebut dengan *database system*. Sistem basis data (*database system*) adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam didalam suatu organisasi (Jogiyanto, 2004). Sampai dengan membentuk suatu *database,* data mempunyai jenjang mulai dari karakter-karakter (*character*) item data (*data item* atau *field*), *record, file* dan kemudian *database.* Karakter merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf atau karakter-karakter khusus (*special character*) yang membentuk suatu item data.

1. Field

   Suatu *field* menggambarkan suatu atribut dari *record* yang menunjukkan suatu item dari data, seperti misalnya nama, alamat, dan lain sebagainya. Kumpulan dari *field* membentuk suatu *record*.

2. Record

   Kumpulan dari *field* yang membentuk suatu *record*. *Record* menggambarkan suatu unit data individu yang tertentu. Kumpulan dari *record* membentuk suatu *file*. Misalnya *file* personalia, tiap-tiap *record* dapat mewakili data tiap-tiap karyawan.

3. *File*

   File terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis. Misalnya *file* matakuliah berisi data tentang semua matakuliah yang ada.

4. *Database*

   Kumpulan dari *file* yang membentuk suatu *database*.

2.7 Alat Bantu Pengembangan Sistem

Pengembangan sistem membutuhkan suatu alat bantu dalam analisa data. Salah satu teknik analisa data secara terstruktur yang digunakan adalah *flowchart* dan diagram *use case.,* di mana penganalisis sistem dapat merepresentasikan proses-proses data di dalam organisasi.

1.7.1 Flowchart

Flowchart (bagan alir) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.

Indrajani (2011), *flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur program yang biasanya mempermudah penyelesaian masalah.

Berikut ini akan dijelaskan jenis-jenis atau macam-macam dari flowchart antara lain yaitu:

1. *System Flowchart*

   Bagan alir sistem (*system flowchart*) dapat didefinisikan sebagai bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan pada sistem.

2. *Document Flowchart*

   Bagan alir dokumen (*document flowchart*) atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.

3. *Schematic Flowchart*

   Bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaannya adalah, bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem, juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbolsimbol bagan alir. Penggunaan gambar-

gambar ini memudahkan untuk dipahami, tetapi sulit dan lama dalam proses menggambarnya.

4. *Program Flowchart*

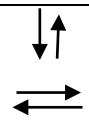
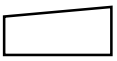
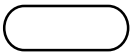
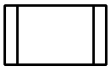
   Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir sistem. Bagan alir program dapat terdiri dari dua macam, yaitu bagan alir logika program (*program logic flowchart*) dan bagan alir program komputer terinci (*detailed computer program flowchart*).

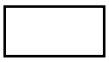
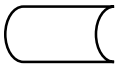
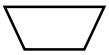
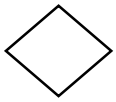
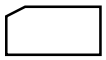
   a. Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika. Bagan alir logika program ini dipersiapkan oleh analis sistem.
   b. Bagan alir program komputer terinci digunakan untuk menggambarkan instruksi-instruksi program komputer secara terinci. Bagan alir ini dipersiapkan oleh pemrogram.

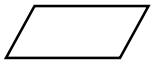
5. *Process Flowchart*

   Bagan alir proses (*process flowchart*) merupakan bagan alir yang banyak digunakan di teknik industri. Bagan alir ini juga berguna bagi analis sistem untuk menggambarkan proses dalam suatu prosedur.

Tabel 2.2 Simbol-simbol Flowchart

Simbol	Keterangan	Simbol	Keterangan
	*Flow Direction Symbol*, menghubungkan antara simbol yang satu dengan yang lainnya. Simbol ini disebut juga *conecction line*		Simbol *manual input* untuk pemasukan data secara manual *online keyboard*
	*Terminator Symbol*, digunakan untuk permulaan (*start*) atau akhir (*stop*) dari suatu kegiatan		Simbol *Preparation*, untuk mempersiapkan penyimpanan yang akan digunakan untuk tempat penyimpanan didalam *storage*
	*Conecctor Symbol*, digunakan untuk keluar-masuk ataupun menyambungkan proses dalam lembar/halaman yang sama		*Predefine Proses*, untuk pelaksana suatu bagian/ sub program/*prosedure*
	*Conecctor Symbol*, digunakan untuk keluar-masuk ataupun menyambungkan proses dalam lembar/halaman yang berbeda		Simbol *Display*, untuk menyatakan peralatan *output* yang digunakan

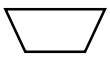
			yaitu layar, plotter, printer dan sebagainya
	*Processing Symbol*, simbol untuk menunjukkan proses pengolahan yang dilakukan oleh komputer		Simbol *Disk and On-Line Storage*, digunakan untuk menyatakan *input* yang berasal dari *disk* atau di simpan ke *disk*
	Simbol *Manual Operation*, digunakan untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer		Simbol *Magnetik Tape Unit*, untuk menyatakan *input* berasal dari pita magnetik atau *output* di simpan kepita magnetik
	Simbol *Decision*, digunakan untuk pemilihan proses berdasarkan kondisi yang ada		Simbol *Punch Card*, menyatakan bahwa *input* berasal dari kartu atau *output* ditulis ke kartu

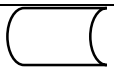
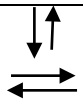
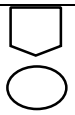
	Simbol *Input-Output* yang menyatakan proses *input* dan *output* tanpa tergantung dengan jenis peralatannya		Simbol Dokumen untuk menyatakan bahwa *input* berasal dari dokumen dalam bentuk kertas atau *output* dicetak ke kertas

1.7.2 Flowmap

Pada tahun 1990 *flowmap* mulai dikembangkan dengan program sederhana untuk aliran barang maupun aliran manusia didalam menggambarkan peta. Program *flowmap* dikembangkan oleh Tom De Jong (Universitas Utrecht Belanda). *Flowmap* adalah gabungan peta dan *flowchart*, yang menunjukan pergerakan benda dari satu lokasi ke lokasi lain. *Flowmap* menolong analisis dan programmer untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoprasian.

Tabel 2.3 Simbol-simbol Flowmap

Simbol	Keterangan	Simbol	Keterangan
	Dokumen, menunjukkan *I/O* baik untuk proses manual, mekanik dan komputer		Manual, menunjukkan dimana pekerjaan dilakukan secara manual

	Kartu *punch*, menunjukkan *I/O* yang menggunakan kartu *punch*		Proses, menunjukkan kegiatan proses dari operasi program komputer
	Pita mekanik, menunjukkan *I/O* yang menggunakan pita mekanik		Disket, menunjukkan *I/O* menggunakan disket
	Pita kertas berlubang, yaitu menunjukkan proses *I/O* yang menggunakan pita berlubang		*Keyboard*, menunjukkan *input* yang menggunakan *online keyboard*
	Garis alir, menunjukkan aliran proses		Penghubung, menunjukkan penghubung ke halaman yang sama atau halaman yang lain
	Operasi luar, menunjukkan operasi yang dilakukan diluar operasi komputer		*Disk*, menunjukkan *I/O* yang menggunakan *harddisk*
	*Sort offline*, menunjukkan proses pengurutan data diluar proses komputer		Drum magnetk, menunjukkan *I/O* yang menggunakan drum magnetik
	Simpanan *offline*, *file* non komputer yang diarsip urut angka		Simpanan *offline*, *file* non komputer yang diarsip urut huruf

Hal yang dapat dilakukan untuk pembuatan diagram *flowmap* yang baik adalah dengan cara seperti berikut :

1. Pembuatan *flowmap* untuk pertama kalinya adalah membagi-bagi diagram ke dalam kolom-kolom.
2. Setiap kolom diberi nama entitas yang terlibat.
3. Diagram dibaca dari atas ke bawah dan dari kiri ke kanan.
4. Setiap kolom terdapat siklus pengolahan data *input*-proses-*output.*
5. Ketika menyeberangi garis yang memisahkan antara satu kolom dengan kolom lain, gunakan simbol konektor.
6. Cara mengakses *file* komputer adalah melalui simbol proses komputer.
7. Prosedur kerja yang kejadiannya tidak bersamaan dapat digambarkan melalui *flowmap* yang terpisah.

1.7.3 Data Flow Diagram

Menurut Andri Kristanto *Data Flow Diagram* (*DFD*) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan ke mana tujuan data yang keluar dari sistem, di mana data tersimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data tersimpan dan proses yang dikenakan pada data tersebut.

Menurut Tata Sutabri, *DFD* adalah suatu *network* yang menggambarkan suatu sistem otomatis atau komputerisasi, manualisasi atau gabungan dari keduanya, yang penggambarannya disusun di dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan.

2.7.3.1 Level DFD

Dalam proses perancangan *DFD* terdapat tiga level yang akan dibentuk, antara lain sebagai berikut:

1. Diagram Konteks

   Menggambarkan satu lingkaran besar yang dapat mewakili seluruh proses yang terdapat di dalam suatu sistem. Merupakan tingkatan tertinggi di dalam *DFD* dan biasanya diberi nomor 0 (nol). Semua entitas eksternal yang ditunjukkan pada diagram konteks yaitu berupa aliran-aliran data utama menuju dan dari sistem. Diagram konteks ini sama sekali tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan.

2. Diagram Nol (Diagram level-1)

   Merupakan satu lingkaran besar yang mewakili lingkaran-lingkaran kecil yang ada di dalamnya. Merupakan pemecahan dari diagram konteks ke diagram nol. Di dalam diagram nol ini memuat penyimpanan data.

3. Diagram Rinci

4. Merupakan diagram yang menguraikan proses apa yang ada di dalam diagram nol.

2.7.3.2 Fungsi DFD

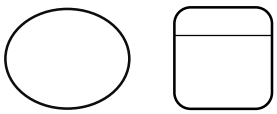
Berikut ini akan dijelaskan mengenai fungsi-fungsi dari *data flow diagram* antara lain sebagai berikut:

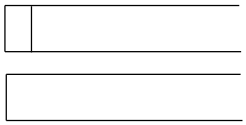
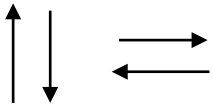
1. Sebagai alat pembuatan model yang memungkinkan professional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional

yang dihubungkan satu sama lain dengan alur data, baik itu secara manual maupun komputerisasi.

2. Sebagai salah satu dari alat pembuatan model yang sering dipergunakan, khususnya jika fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, *DFD* adalah alat pembuatan model yang memberikan penekanan pada fungsi sistem.
3. Sebagai alat perancangan sistem yang berorientasi pada alur data dengan menggunakan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

Tabel 2.4 Simbol-simbol Data Flow Diagram

Nama Simbol	Simbol	Keterangan
External Entity		*External Entity* merupakan di lingkungan luar sistem yang bisa berupa orang, organisasi atau sistem lain
Process		*Process* merupakan proses seperti aritmatik penulisan suatu formula atau pembuatan laporan

Data Storage		*Data Storage* (simpan data) adalah berupa suatu *file* atau *database* pada sistem komputer atau catatan manual
Data Flow		*Data Flow* (arus data), arus data ini mengalir diantara proses, simpan data dan kesatuan luar

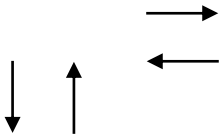
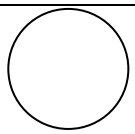
1.7.4 Diagram Konteks

Menurut Ladjmudin (2013), pengertian diagram konteks adalah: “Diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. “ Diagram konteks merupakan DFD level yang paling atas yang hanya terdiri dari satu proses yang menggambarkan sistem atau program secara keseluruhan.

Tabel 2.5 Diagram Konteks

Sumber: Ladjmudin (2013)

Simbol	Pengertian	Keterangan
	Eksternal *Entity*	Menunjukan bagian luar sistem atau sumber input dan output

	Garis aliran	Menunjukan arus sata antar simbol/proses
	Sistem	Menunjukan sistem
Ktp, uang dan lain-lain	Atribut	Data-data yang diolah

1.8 Visual Basic 2010

Menurut Adelia dan Setiawan (2011: 114): "*Microsoft Visual Basic* (sering disingkat sebagai VB saja) merupakan sebuah bahasa pemrograman yang bersifat *event driven* dan menawarkan *Integrated Development Environment (IDE) visual* untuk membuat program aplikasi berbasis sistem operasi *Microsoft Windows* dengan menggunakan model pemrograman *Common Object Model (COM)"*.

Visual Basic merupakan turunan bahasa BASIC dan menawarkan pengembangan aplikasi komputer berbasis grafik dengan cepat, akses ke basis data menggunakan *Data Access Objects* (DAO), *Remote Data Objects* (RDO), atau *ActiveX Data Object* (ADO), serta menawarkan pembuatan kontrol *ActiveX* dan objek *ActiveX*.

Visual Basic merupakan turunan bahasa *BASIC* dan menawarkan pengembangan aplikasi komputer berbasis grafik dengan cepat, akses ke basis data menggunakan *Data Access Objects* (DAO), *Remote Data Objects* (RDO), atau *ActiveX Data Object* (ADO), serta menawarkan pembuatan kontrol *ActiveX* dan objek *ActiveX*. Beberapa bahasa skrip seperti Visual Basic for Applications (VBA)

dan *Visual Basic Scripting Edition* (*VBScript*), mirip seperti halnya *Visual Basic*, tetapi cara kerjanya yang berbeda. Para *programmer* dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh *Microsoft Visual Basic* Program-program yang ditulis dengan *Visual Basic* juga dapat menggunakan *Windows API*, tapi membutuhkan deklarasi fungsi eksternal tambahan.

Visual Basic. Net merupakan sebuah program yang lengkap untuk membangun aplikasi *Web ASP, Service XML Web*, Aplikasi *Desktop* dan *Mobile*. *Microsoft Visual Basic .NET* merupakan sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem *.NET Framework*, dengan menggunakan bahasa *BASIC*. Dengan menggunakan alat ini, para *programmer* dapat membangun aplikasi *Windows Forms*, Aplikasi *web* berbasis ASP.NET, dan juga aplikasi *command-line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti *Microsoft Visual C++*, *Visual C#*, atau *Visual J#)*, atau juga dapat diperoleh secara terpadu dalam *Microsoft Visual Studio .NET*. Bahasa *Visual Basic .NET* sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari *Microsoft Visual Basic* versi sebelumnya yang diimplementasikan diatas .NET Framework. Peluncurannya mengundang kontroversi, mengingat banyak sekali perubahan yang dilakukan oleh Microsoft, dan versi baru ini tidak kompatibel dengan versi terdahulu.

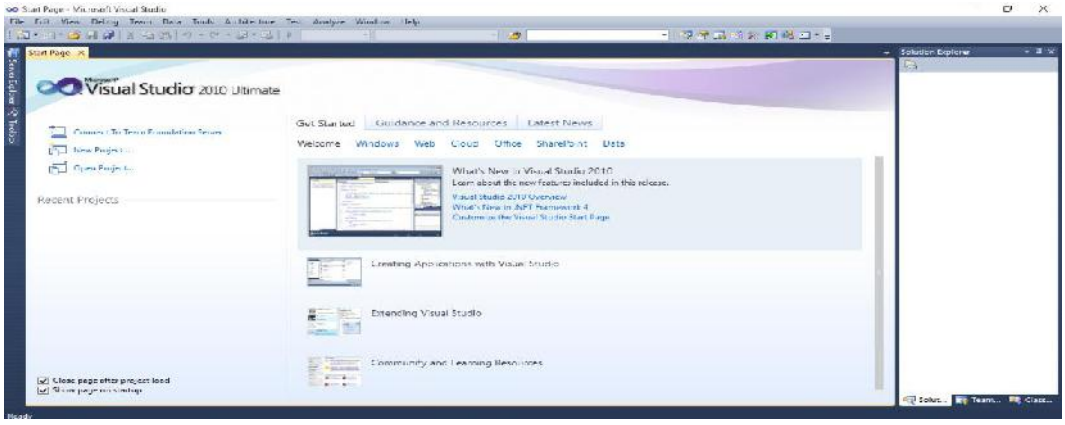
Visual Basic.NET 2003 tersedia dalam beberapa jenis cita rasa: *Professional, Enterprise Architect* dan *Academic Edition*. Khusus untuk *Visual Basic.NET 2003 Academic Edition*, versi tersebut didistribusikan secara gratis

untuk beberapa sekolah didalam setiap negara; versi *Professional* dan *Enterprise Architect* merupakan produk komersial.

Setelah itu, *Microsoft* pun berkonsentrasi dalam mengembangkan *Microsoft .NET Framework 2.0*, dan tentunya alat bantu untuk membangun program diatasnya. Hingga pada tahun 2005, mereka pun merilis versi terbaru dari *Visual Basic.NET*, yang kali ini disebut dengan *Visual Basic 2005* (dengan membuang kata "*.NET*"), bersama-sama dengan beberapa aplikasi pengembangan lainnya. Sedangkan Tampilan awal *Visual basic 2010* dapat dilihat pada Gambar 2.3.

Gambar 2.3 Tampilan Awal Visual Basic 2010

Sumber: Andi Putra (2008)



1.9 MySQL

My Structured Query Language (*MySQL*) adalah sebuah implementasi dari sistem manajemen basis data relasional (*RDBMS*) yang didistribusikan secara gratis di bawah lisensi *General Public License* (*GPL*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang

telah ada sebelumnya *Structured Query Language* (*SQL*). *SQL* adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

1. Kelebihan dan Kekurangan MySQL

   Hal paling mendasar yang menjadikan *MySQL* pilihan utama sebagai *database* yang digunakan adalah karena *MySQL* menggunakan Lisensi *GPL* dan multiplatform, sehingga tidak membutuhkan biaya besar dalam membuat aplikasi. Tapi alasan tersebut tidaklah cukup untuk menjadikan *MySQL* sebagai *RDBMS* yang akan digunakan.

   Berikut ini adalah beberapa kelebihan dari *MySQL*, antara lain yaitu :

   a. Berlisensi *GPL* dan Multi Platform.
   b. Dapat di integrasikan dengan beberapa bahasa Pemrograman seperti .Net, Java, Python, Perl yang merupakan bahasa pemrograman yang paling dominan di kalangan programmer.
   c. Mendukung *ODBC* untuk sistem operasi Windows sehingga bisa digunakan aplikasi yang berjalan di windows
   d. Bisa dijalankan pada spesifikasi *hardware* yang rendah karena lebih hemat *resource memory* dibandingkan *database* lain sehingga mudah digunakan untuk bahan pembelajaran
   e. *MySQL* dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari 20 bahasa meskipun Bahasa Indonesia belum termasuk didalamnya.

Berikut ini adalah beberapa kekurangan dari *MySQL*, antara lain yaitu:

a. Banyak mengklaim kurang *support* terhadap pemrograman Visual/Desktop, sehingga sedikit yang menggunakan untuk aplikasi visual.
b. Karena berlisensi *GPL* sehingga sulit mendapatkan *update* untuk *problem* yang *urgent*, sehingga perusahaan skala menengah keatas lebih memilih *RDBMS* berlisensi dan disupport seperti *Oracle* dan *MS SQL Server*
c. Sangat diragukan dalam menangani data skala besar, karena ada beberapa opini yang pro dan kontra terhadap kemampuan *MySQL* terhadap pengolahan data yang besar.

1.10 XAMPP

XAMPP merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. *XAMPP* adalah perangkat yang menggabungkan tiga aplikasi kedalam satu paket, yaitu Apache, *MySQL*, dan PHPMyAdmin. *XAMPP* merupakan salah satu paket installasi Apache, *PHP* dan *MySQL instant* yang dapat kita gunakan untuk membantu proses installasi ketiga produk tersebut. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan web server yang mudah untuk digunakan.

Mengenal bagian penting dari *XAMPP* yang biasa digunakan pada umumnya, antara lain yaitu:

1. Htdoc adalah folder tempat meletakkan berkas-berkas yang akan dijalankan, seperti berkas *PHP, HTML* dan skrip lain.
2. PHPMyAdmin merupakan bagian untuk mengelola basis data *MySQL* yang ada dikomputer. Untuk membuka halaman PHPMyAdmin, buka browser lalu ketikkan alamat http://localhost/PHPMyAdmin, maka akan muncul halaman PHPMyAdmin.

3. Kontrol Panel yang berfungsi untuk mengelola layanan (*service*) *XAMPP*. Seperti menghentikan (*stop*) layanan, ataupun memulai (*start*).

1.10.1 PHPMyAdmin

PHPMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman *PHP* yang digunakan untuk menangani administrasi *MySQL* melalui *WWW (World Wide Web)*. PHPMyAdmin mendukung berbagai operasi *MySQL*, diantaranya mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perizinan (*permissions*), dan lain sebagainya dengan mudah, tanpa harus menghafal baris perintahnya. Pada dasarnya, mengelola basis data dengan *MySQL* harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (*command line*) untuk setiap maksud tertentu.

BAB III

METODE PENELITIAN

3.1 Analisis Sistem

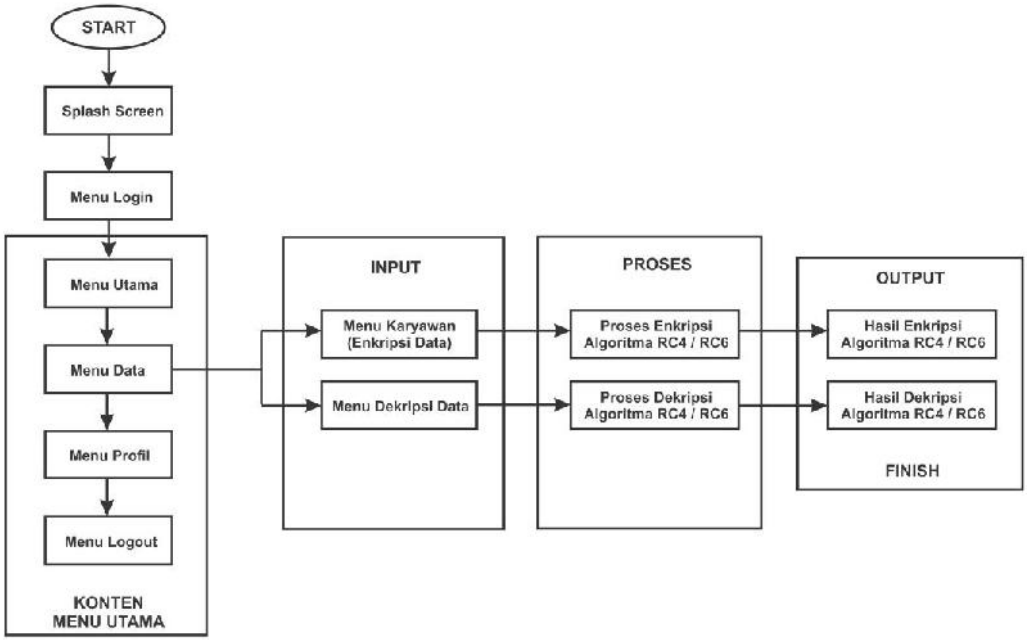
Analisis sistem adalah suatu tahapan sistem yang dilakukan untuk membantu memahami suatu masalah dan apa yang akan menjadi kebutuhan sistem. Tujuannya adalah untuk membantu mengetahui masalah yang ada dan merancang model suatu sistem yang akan dibangun sehingga menjadi tepat guna.

3.2 Analisis Masalah

Analisis masalah yang dilakukan pada penelitian ini untuk keamanan data yang merupakan hal yang sangat penting yang berisi informasi sensitif yang hanya boleh diketahui isinya oleh pihak yang berhak saja. Sistem pengamanan data menggunakan algoritma RC 4 dan RC 6 dalam aplikasi ini dapat mencegah hal-hal yang menyebabkan kerugian bagi pihak yang melakukan karena semakin banyak cara yang bisa dilakukan oleh pihak yang tidak bertanggung jawab yang ingin mengetahui bahkan menghilangkan informasi tersebut.

3.3 Gambaran Umum Sistem

Adapun gambaran umum dari sistem ini digambarkan sebagai berikut:



Gambar 3.1 Gambaran Umum Sistem

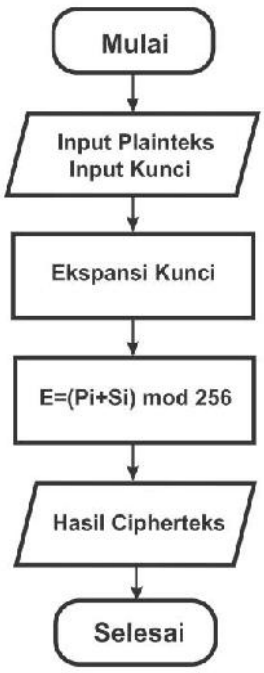
Penjelasan dari gambaran umum sistem yaitu sebagai berikut:

3.3.1 Input

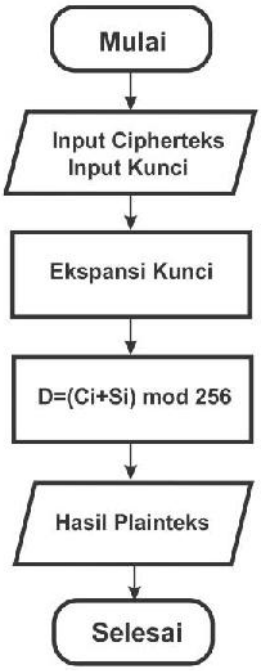
Input dari sistem ini yaitu pada menu karyawan (enkripsi data). Data karyawan yang tertera pada menu tersebut yang akan dienkripsi menggunakan algoritma RC4 dan RC6. Dan juga untuk data yang akan didekripsi yaitu juga dengan data yang sama berdasarkan hasil enkripsi data sebelumnya ke bentuk data semula sebelum dienkripsi.

3.3.2 Proses

Untuk proses dari enkripsi dan dekripsi data akan dijelaskan dan digambarkan dalam bentuk *flowchart* (diagram alir) sebagai berikut:



Gambar 3.2 Proses Enkripsi Data



Gambar 3.3 Proses Dekripsi Data

Penjelasannya sebagai berikut:

Langkah pertama yaitu *input plainteks*, kemudian input kuncinya. Selanjutnya, prosedur *Ekspansi kunci* dilakukan, proses selanjutnya adalah mengenkripsi dan dekripsi data, dengan menggunakan tabel *ASCII* yang memiliki 256 karakter, dengan menggunakan Kunci *S* yang telah dihasilkan dari proses ekspansi kunci, sehingga untuk mengenkripsi dan dekripsi dengan algoritma *Vigenere Cipher* digunakan rumus sebagai berikut:

$$C_i = E\ (\ P_i + S_i\ )\ mod\ 256 \quad \text{.....} \quad (3.1)$$

$$P_i = D\ (\ C_i - S_i\ )\ mod\ 256 \quad \text{.....} \quad (3.2)$$

Plainteks di enkripsi menggunakan kunci *S* setiap 1 *byte*, demikian juga dengan cipherteks yang di dekripsi menggunakan kunci *S.* Proses enkripsi dilakukan dengan menambahkan nilai *ASCII* dari tiap-tiap plainteks dengan kunci dan dimoduluskan dengan 256, demikian sebaliknya, cipherteks di dekripsi dengan cara pengurangan modulo dari nilai *ASCII* tiap 1 *byte* dengan tiap 1 *byte* nilai *ASCII* kunci *S*.

Pada Gambar 3.2. dan 3.3 menjelaskan proses enkripsi dan dekripsi, untuk enkripsi mulamula plainteks dan kunci dimasukkan oleh pengguna, kemudian dilakukan proses ekspansi kunci algoritma RC 4 dan RC6 seperti yang dijelaskan pada sebelumnya pada penelitian ini, selanjutnya dilakukan proses enkripsi untuk mendapatkan *cipherteks*. Demikian juga untuk proses dekripsi, dilakukan proses yang sama, yakni dimasukkan *cipherteks* dan kunci, kemudian lakukan proses ekspansi kunci yang sama seperti diatas lalu proses dekripsi dilakukan untuk menghasilkan *plainteks*.

Berikut adalah contoh perhitungan implementasi algoritma RC4 dengan mode 4 *byte* (untuk lebih menyederhanakan dalam perhitungan manual) serta untuk kebutuhan sistem yang sangat terbatas. S-Box dengan panjang 4 byte, dengan S[0]=0, S[1]=1, S[2]=2 dan S[3]=3 sehingga array S menjadi: 0 1 2 3

Inisialisasi 4 byte kunci array, K. Misalkan kunci Ulang kunci sampai memenuhi seluruh adalah 2 5 7 3, sehingga array K berisi 2 5 7 3 dan mencoba untuk mengenkripsikan kata HALO.

Inisialisasi i dan j dengan 0 kemudian dilakukan KSA agar tercipta state-array yang acak. Penjelasan iterasi lebih lanjut dapat dijelaskan sebagai berikut:

Iterasi 1

i = 0

j = (0 + S[0] + K [0 mod 4]) mod 4

= (0 + 0 + 2) mod 4 = 2

Swap (S[0],S[2])

Hasil Array S = 2 1 0 3

Iterasi 2

i = 1

j = (2 + S[1] + K [1 mod 4]) mod 4

= (2 + 1 + 5) mod 4 =0

Swap (S[1],S[0])

Hasil Array S = 1 2 0 3

Iterasi 3

i = 2

j = (0 + S[2] + K [2 mod 4]) mod 4

= (0 + 0 + 7) mod 4 = 3

Swap (S[2],S[3])

Hasil = 1 2 3 0

Iterasi 4

i = 3

j = (3 + S[3] + K [3 mod 4]) mod 4

= (3 + 0 + 3) mod 4 = 2

Swap (S[3],S[2])

Hasil Array S = 1 2 0 3

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

Array S = 1 2 0 3

Inisialisasi

i = 0

j = 0

Iterasi 1

i = (0 + 1) mod 4 = 1

j = (0 + S[1]) mod 4 = (0+ 2) mod 4 = 2

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

1 0 2 3

K1 = S[(S[1]+S[2]) mod 4] = S[2

mod 4] = 2

K1 = 00000010

Iterasi 2

i = (1 + 1) mod 4 = 2

j = (2 + S[2]) mod 4 = (2+ 2) mod

4 = 0

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

2 0 1 3

K2 = S[(S[2]+S[0]) mod 4] = S[3

mod 4] = 3

K2 = 00000011

Iterasi 3

i = (2 + 1) mod 4 = 3

j = (0 + S[3]) mod 4 = (0+ 3) mod

4 = 3

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

1 0 2 3

K3 = S[(S[3]+S[3]) mod 4] = S[6

mod 4] = 2

K3 = 00000010

Iterasi 4

i = (3 + 1) mod 4 = 0

j = (3 + S[0]) mod 4 = (3+ 1) mod

4 = 0

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

1 0 2 3

K1 = S[(S[0]+S[0]) mod 4] = S[2 mod 4] = 2

K4 = 00000010

Setelah menemukan kunci untuk tiap karakter, makadilakukan operasi XOR antara karakter pada plaintext dengan kunci yang dihasilkan. Berikut adalah tabel ASCII untuk tiap-tiap karakter pada plaintks yang digunakan.

Huruf	Kode ASCII (Binary 8 bit)
H	01001000
A	01000001
L	01001100
O	01001111

Berikut adalah proses pengXORan dari plainteks dengan key yang telah didapat:

H A L O : 01001000 01000001 01001100 01001111
Key : 00000010 00000011 00000010 00000010
Cipherteks : 01001010 01000010 01001110 01001101

Contoh Dekripsi:

Proses dekripsi *ciphertext* menggunakan algoritma RC4 ini sama untuk proses key-schedule-nya. Untuk mendapatkan plaintext, ciphertext yang diperoleh di XORkan dengan pseudo random byte yang didapat sebelumnya. Maka hasilnya adalah plainteks atau teks asli.

Pesan dikirim dalam bentuk cipherteks sehingga setelah sampai di penerima pesan dapat kembali diubah menjadi plainteks dengan meng-XOR-kan dengan kunci yang sama. Pemrosesan pesan setelah sampai pada penerima dapat dilihat pada dibawah ini.

Proses XOR pseudo random byte dengan cipherteks pada dekripsi yaitu:

Cipherteks : 01001010 01000010 01001110 01001101
Pseudo Random Byte : 00000010 00000011 00000010 00000010
Plainteks : 01001000 01000001 01001100 01001111
H A L O

3.3.3 Output

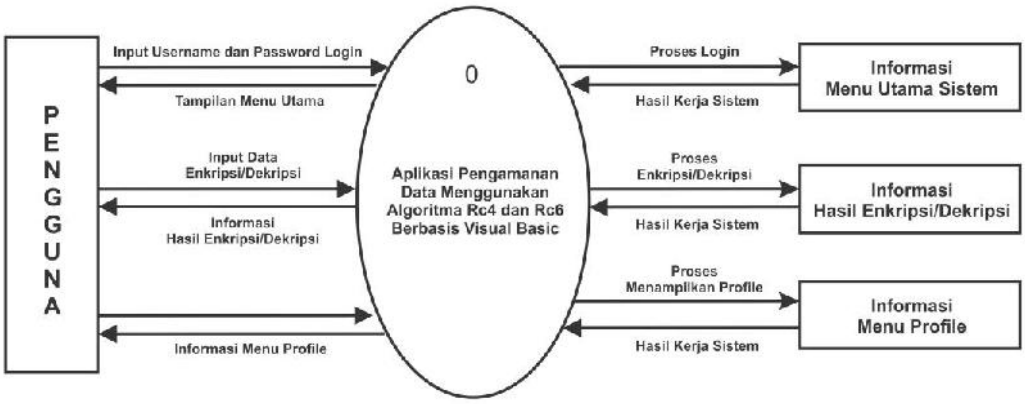
Output dari proses sistem ini yaitu hasil data yang telah dienkripsi ke dalam bentuk symbol enkripsi data dan hasil dari dekripsi data yaitu hasil yang berdasarkan enkripsi sebelumnya diubah ke bentuk data semula sebelum proses enkripsi data.

3.4 Perancangan Sistem

Perancangan sistem ini digunakan untuk menggambarkan rancangan sistem yang akan dibuat secara keseluruhan. Perancangan sistem ini mencakup *activity diagram, flowchart*, perancangan *database* dan *user interface.*

3.5 Perancangan DFD

Data Flow Diagram (DFD) adalah alat yang biasa dipakai untuk mendokumentasi proses dalam sistem atau sebuah teknis grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari *input* menjadi *output*. DFD yang menggambarkan keseluruhan sistem secara utuh disebut DFD level 0 atau biasa dikenal dengan Diagram Konteks. Berikut ini dapat dilihat Diagram Konteks atau DFD level 0 yang digunakan dalam sistem seperti gambar 3.4 berikut:



Gambar 3.4 Diagram Konteks Sistem

3.6 Activity Diagram

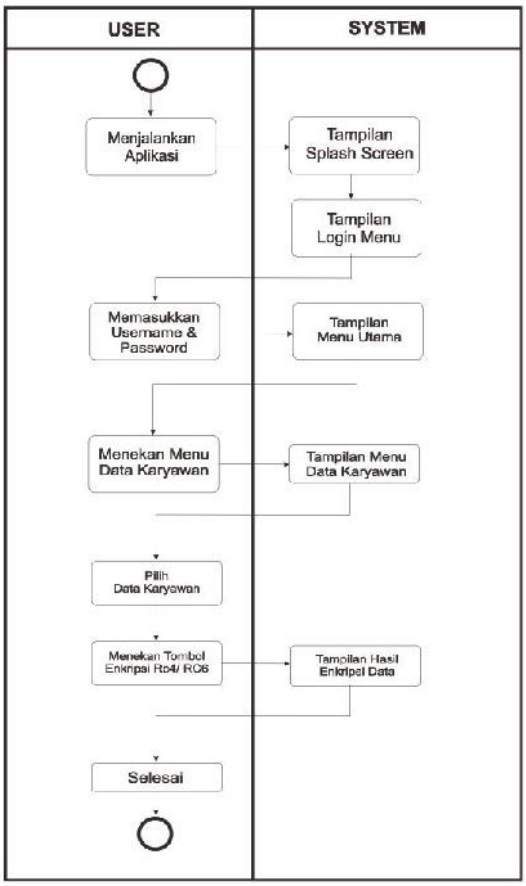
Activity diagram adalah diagram yang menggambarkan aktivitas yang terjadi selama sistem dijalankan. Diagram aktivitas mendeskripsikan bagaimana awal proses dimulai, keputusan tindakan yang dilakukan dan bagaimana akhir atau hasil dari proses tersebut.

Diagram aktivitas pada sistem user merupakan diagram yang menggambarkan aktivitas yang terjadi selama sistem dijalankan pada sistem user di platform android. Diagram aktivitas mendeskripsikan bagaimana awal proses dimulai, keputusan tindakan yang dilakukan dan bagaimana akhir atau hasil dari proses tersebut.

3.6.1 Activity Diagram Enkripsi Data

Aktivitas yang terjadi pada menu enkripsi data yaitu dimulai pada proses memulai aplikasi. Kemudian setelah aplikasi dapat berjalan maka sistem akan menampilkan *splash screen* yang didalam prosesnya terdapat proses *loading data.*

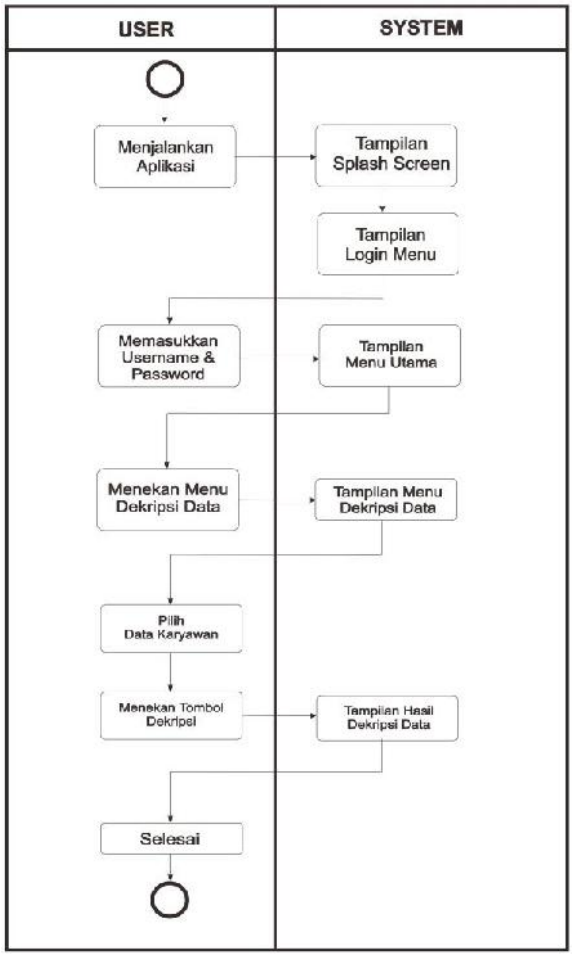
Kemudian setelah itu sistem akan masuk ke menu login, pengguna ataupun *administrator* diminta untuk memasukkan *username* dan *password.* Kemudian menekan Button Login. Setelah itu sistem akan masuk ke menu utama. Kemudian masuk ke menu Data Karyawan untuk melakukan enkripsi data. Setelah masuk ke Menu Data Karyawan, maka akan dipilihlah data mana yang akan di enkripsi. Kemudian setelah data dipilih, tekan buton Enkripsi RC4 atau RC6, maka setelah itu hasil enkripsi data akan ditampilkan pada *TextBox* Hasil Enkripsi Data. Adapun aktivitasnya digambarkan pada gambar 3.5 berikut:



Gambar 3.5 Activity Diagram Enkripsi Data

3.6.2 Activity Diagram Dekripsi Data

Aktivitas yang terjadi pada menu dekripsi data yaitu dimulai pada proses memulai aplikasi. Kemudian setelah aplikasi dapat berjalan maka sistem akan menampilkan *splash screen* yang didalam prosesnya terdapat proses *loading data.* Kemudian setelah itu sistem akan masuk ke menu login, pengguna ataupun *administrator* diminta untuk memasukkan *username* dan *password.* Kemudian menekan Button Login. Setelah itu sistem akan masuk ke menu utama. Kemudian masuk ke menu Dekripsi Data untuk melakukan dekripsi data. Setelah masuk ke Menu Dekripsi Data, maka akan dipilihlah data mana yang akan di dekripsi. Kemudian setelah data dipilih, tekan *buton Dekripsi*, maka setelah itu hasil dekripsi data akan ditampilkan pada *TextBox* Hasil Dekripsi Data. Adapun aktivitasnya digambarkan pada gambar 3.6 berikut:

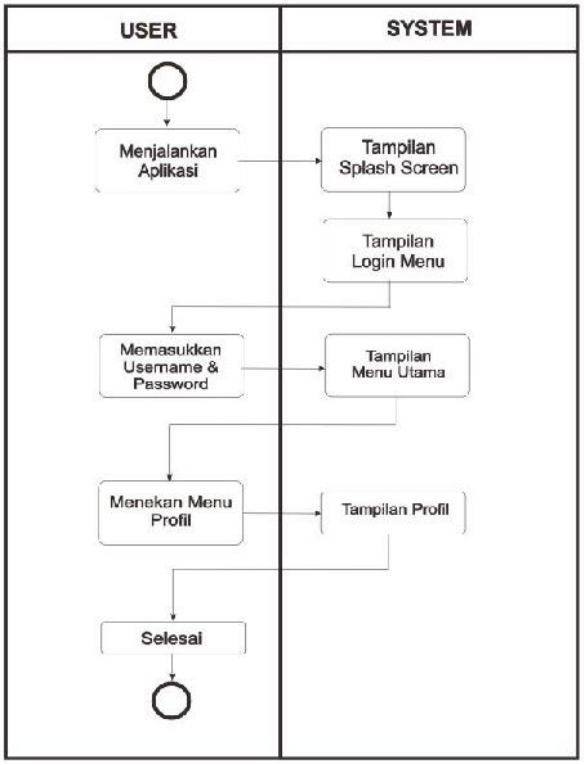


Gambar 3.6 Activity Diagram Dekripsi Data

3.6.3 Activity Diagram Menu Profil

Aktivitas yang terjadi pada menu profil yaitu dimulai pada proses memulai aplikasi. Kemudian setelah aplikasi dapat berjalan maka sistem akan menampilkan *splash screen* yang didalam prosesnya terdapat proses *loading data.* Kemudian setelah itu sistem akan masuk ke menu login, pengguna ataupun *administrator* diminta untuk memasukkan *username* dan *password.* Kemudian menekan Button Login. Setelah itu sistem akan masuk ke menu utama. Kemudian pilih menu Profil.

Kemudian informasi profil akan ditampilkan oleh sistem. Adapun aktivitasnya digambarkan pada gambar 3.7 berikut:

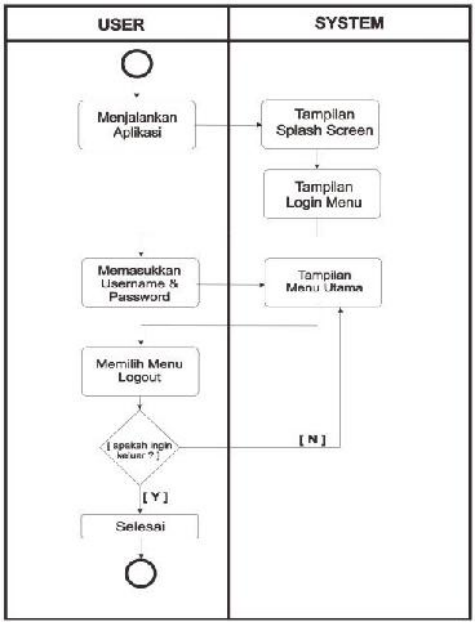


Gambar 3.7 Activity Diagram Menu Profil

3.6.4 Activity Diagram Keluar Aplikasi

Aktivitas yang terjadi pada menu *logout* yaitu dimulai pada proses memulai aplikasi. Kemudian setelah aplikasi dapat berjalan maka sistem akan menampilkan *splash screen* yang didalam prosesnya terdapat proses *loading data.* Kemudian setelah itu sistem akan masuk ke menu login, pengguna ataupun *administrator* diminta untuk memasukkan *username* dan *password.* Kemudian menekan Button Login. Setelah itu sistem akan masuk ke menu utama. Kemudian masuk ke Menu

Logout. Kemudian apabila berhasil, maka sistem akan berhenti. Dan apabila sistem tidak berhasil keluar, maka sistem akan tetap di menu utama. Adapun aktivitasnya digambarkan pada gambar 3.8 berikut:



Gambar 3.8 Activity Diagram Menu Logout

3.7 Perancangan *Database*

Database merupakan tempat penyimpanan data. *Database* digunakan untuk operasi pengolahan data dengan memperhatikan kecepatan proses pengolahan data, waktu minimum yang digunakan dalam penelusuran data, kemampuan penyimpanan data, serta kemudahan dalam *update* data. Dalam hal ini *database editor* yang digunakan adalah MySQL

Adapun tabel-tabel yang penulis rancang untuk menyimpan data dan informasi dalam sistem yaitu sebagai berikut:

1. Tabel karyawan

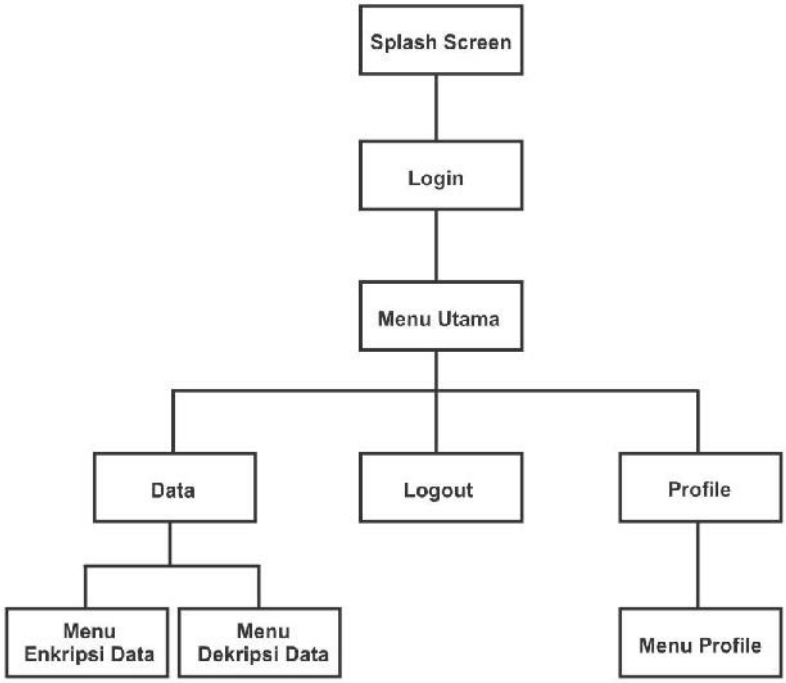
No	Nama	Type	Null	Extra
1	Id	int (255)	No	AUTO_INCREMENT
2	nama_karyawan	text	No	-
3	Nik	int (255)	No	-
4	Nip	int(255)	No	-
5	Alamat	text	No	-
6	pendidikan	text	No	-
7	Notelfn	text	No	-
8	Ipk	int (255)	No	-

2. Tabel tb_login

No	Nama	Type	Null	Extra
1	Id	int (255)	No	AUTO_INCREMENT
2	username	varchar(1000)	No	-
3	Pass	varchar(1000)	No	-

3.8 Perancangan Antarmuka (*Interface*) Sistem

Antar muka program merupakan bagian di mana terjadi komunikasi antara pengguna dengan sistem. Kemudahan bagi pengguna di dalam memahami cara penggunaan sistem ini dapat dijadikan indikasi keberhasilan antara muka melakukan komunikasi dengan pengguna. Adapun sistem yang akan dibuat dengan tampilan *windows*, adalah sebagai berikut:



Gambar 3.9 Rancangan Struktur Menu

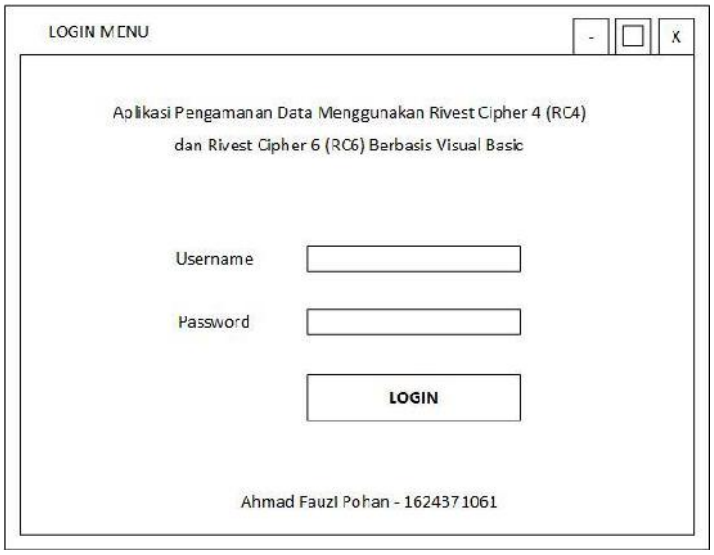
3.8.1 Rancangan Halaman *Splash Screen*

Rancangan halaman *splash screen* hanya tampil dalam beberap detik saja ditandai dengan proses loading masuk ke sistem. Rancangan halamnnya dapat dilihat pada Gambar 3.10 berikut:

Gambar 3.10 Rancangan Halaman Splash Screen

3.8.2 Rancangan Halaman Menu Login

Rancangan halaman menu login menampilkan teks judul skripsi, *TextBox Username* dan *Password* untuk *login user*, kemudian ada *Button Login* untuk masuk ke menu utama serta ada teks nama penulis dan nim penulis. Untuk tampilannya dapat di lihat pada Gambar 3.11 berikut:



Gambar 3.11 Rancangan Halaman Menu Login

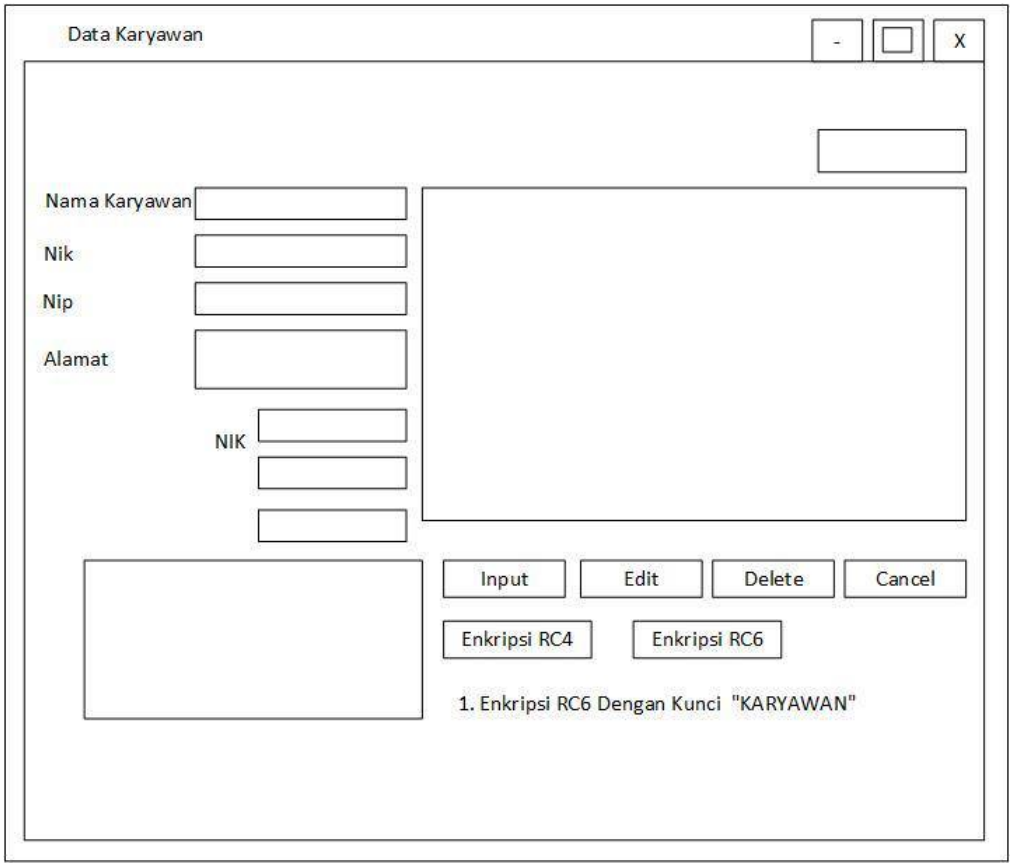
3.8.3 Rancangan Halaman Menu Utama

Pada Rancangan Tampilan *Menu Utama* berisi *Sub Menu* yaitu *Menu Data* yang terdiri dari Menu Data Karyawan (Enkripsi Data) dan Menu Dekripsi Data. Kemudian Menu *Logout* dan *Profile.* Untuk tampilannya dapat di lihat pada gambar 3.12 berikut:

Gambar 3.12 Rancangan Halaman Menu Utama

3.8.4 Rancangan Halaman Menu Enkripsi Data

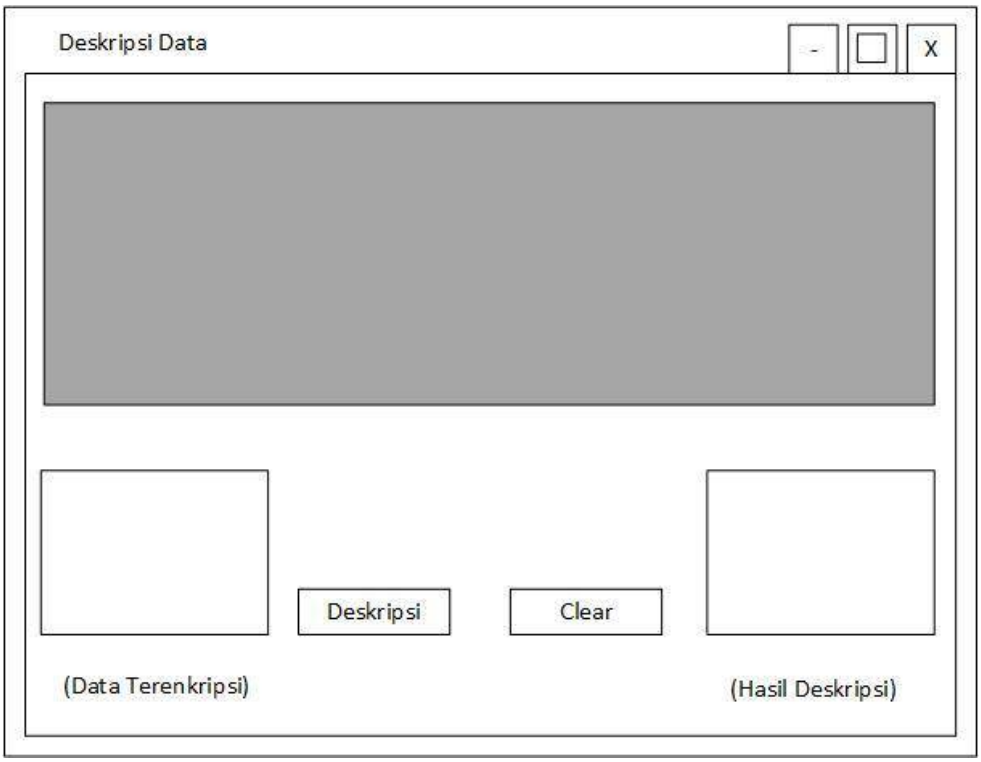
Pada rancangan menu enkripsi data, menampilkan sekumpulan data karyawan. Pada menu ini data karyawan dapat di *input, edit, delete.* Kemudian pada menu ini data karyawan dapat di enkripsi dengan algoritma RC4 dan algoritma RC6 dengan menekan *Button* Enkripsi RC4 dan atau / Enkripsi RC6. Dan kemudian hasil enkripsi akan ditampilkan pada *TextBox* hasil enkripsi. Adapun tampilan menunya pada gambar 3.13 berikut:



Gambar 3.13 Rancangan Halaman Menu Enkripsi Data

3.8.5 Rancangan Halaman Menu Dekripsi Data

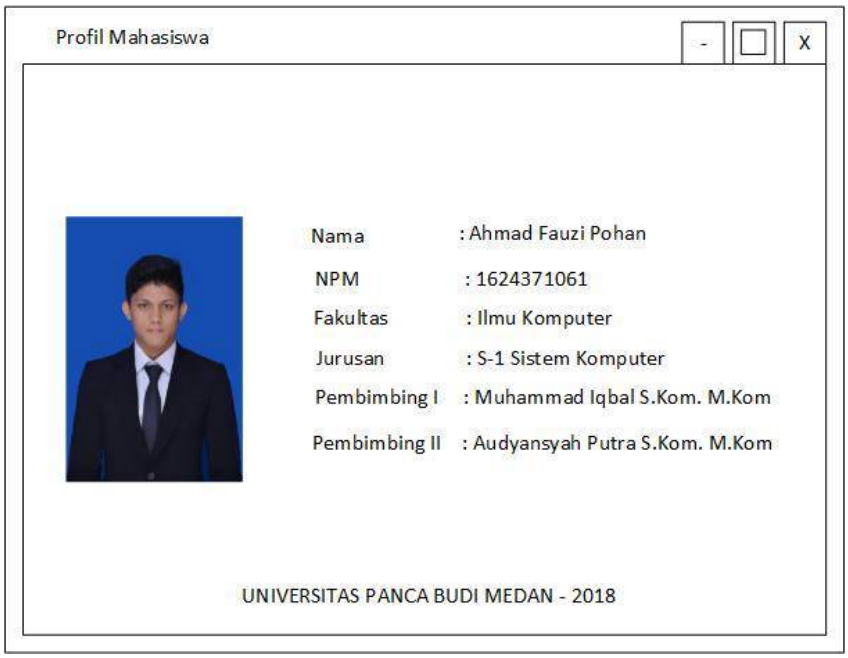
Pada Rancangan Halaman Menu Dekripsi Data, menampilkan *datagrid* kumpulan data karyawan yang telah terenkripsi sebelumnya. Kemudian ada *TextBox* yang menampilkan hasil data enkripsi, kemudian ada *Button* Dekripsi yang berfungsi untuk memproses dekripsi data dan ada *TextBox Hasil Dekripsi* yang berfungsi untuk menampilkan hasil dekripsi data. Adapun tampilannya pada gambar 3.14 sebagai berikut:



Gambar 3.14 Rancangan Halaman Dekripsi Data

3.8.6 Rancangan Halaman Menu Profil

Pada rancangan menu *Profil* menampilkan informasi Judul Skripsi, Nama Penulis, NPM Penulis, Fakultas, Jurusan dari Penulis, Nama Pembimbing I dan II serta Nama Universitas dan Tahun Penelitian Penulis. Adapun tampilannya menunya pada gambar 3.15 berikut:



Gambar 3.15 Rancangan Halaman Menu Profil

BAB IV

HASIL DAN PEMBAHASAN

1.1 Pengertian Implementasi Sistem

Implementasi sistem adalah proses yang dilakukan dalam menyelesaikan desain sistem yang telah disetujui untuk dibuat, diuji, diinstal, dan melalui sistem baru maupun sistem yang diperbaiki untuk menggantikan sistem yang lama.

1.2 Tujuan Implementasi Sistem

Tujuan dari implementasi sistem yaitu mengkaji rangkaian sistem baik dari segi *software* maupun *hardware* sebagai sarana pengolahan data dan penyajian data, menyelesaikan rancangan sistem yang ada didalam dokumentasi sistem yang baru atau yang telah disetujui, memastikan bahwa pemakai dapat mengoperasikan dengan mudah sistem yang telah dibuat, memastikan bahwa sistem telah berjalan dengan lancer dengan mengontrol dan melakukan instalasi secara benar, dan memperhitungkan bahwa sistem telah memenuhi permintaan pengguna yaitu dengan menguji sistem secara menyeluruh.

1.3 Komponen Implementasi Sistem

Pada dasarnya didalam suatu aplilkasi yang dirancang memerlukan beberapa perangkat agar aplikasi dapat berjalan lancar. Berikut beberapa perangkat yang dibutuhkan dalam pembuatan aplikasi ini:

4.3.1 Spesifikasi Perangkat Keras (*Hardware)*

Adapun spesifikasi perangkat keras (hardware) yang digunakan untuk membangun sistem pada penelitian ini adalah sebagai berikut:

1. Processor Intel Core i3 – 403U, 1,9 GHz
2. Memori RAM 4 GB
3. Harddisk 500 GB
4. Resolusi layar 1366 x 768 pixel (14.0 “)

4.3.2 Spesifikasi Perangkat Lunak (*Software)*

Adapun spesifikasi perangkat lunak (software) yang digunakan pada sistem adalah sebagai berikut:

1. Sistem Operasi Windows 10 Professional 64 bit.
2. IDE Microsoft Visual Studio 2010.
3. XAMPP versi 5.6.24
4. Apache versi 2.4.12
5. PHP Versi 5.6.8
6. Bahasa pemrograman Visual Basic (VB)

1.4 Brainware

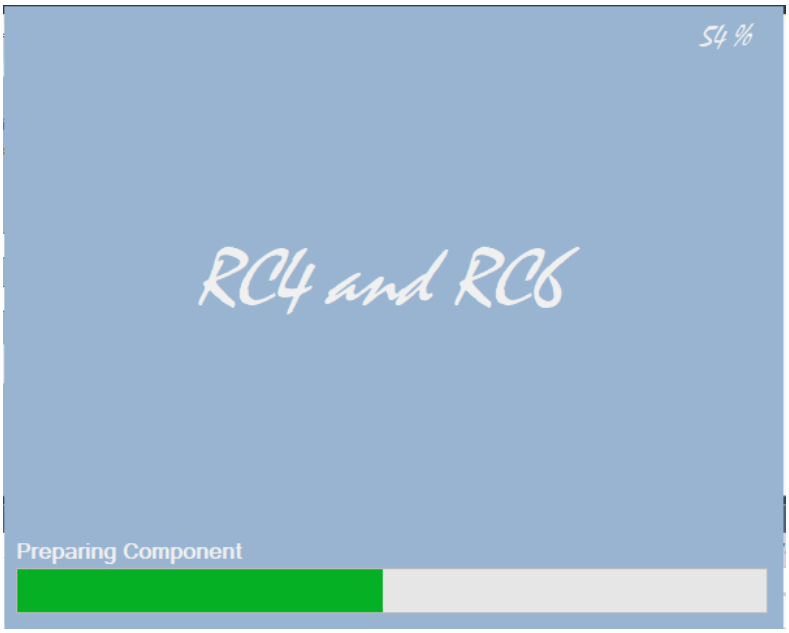
Brainware atau disebut juga pengguna adalah manusia yang terlibat dalam mengoperasikan serta mengatur sistem di dalam komputer. Diartikan juga sebagai perangkat intelektual yang mengoperasikan dan mengeksplorasi kemampuan dari perangkat keras maupun perangkat lunak. *Brainware* termasuk bagian penting dari sebuah sistem komputer. Karena sebuah sistem tidak akan berjalan apabila tidak ada peran *brainware*. Dalam sistem yang dibangun ini. *Hardware* tidak dapat bekerja tanpa adanya *software*, sedangkan *software* dan *hardware* tidak dapat bekerja tanpa adanya *brainware*.

1.5 Tampilan Antarmuka Sistem

Pada tahap ini, dilakukan implementasi tampilan sesuai dengan perancangan antarmuka pada bab 3. Setiap tampilan pada aplikasi akan dibahas bagaimana proses tampilan dan penggunaannya.

1.5.1 Tampilan Halaman *Splash Screen*

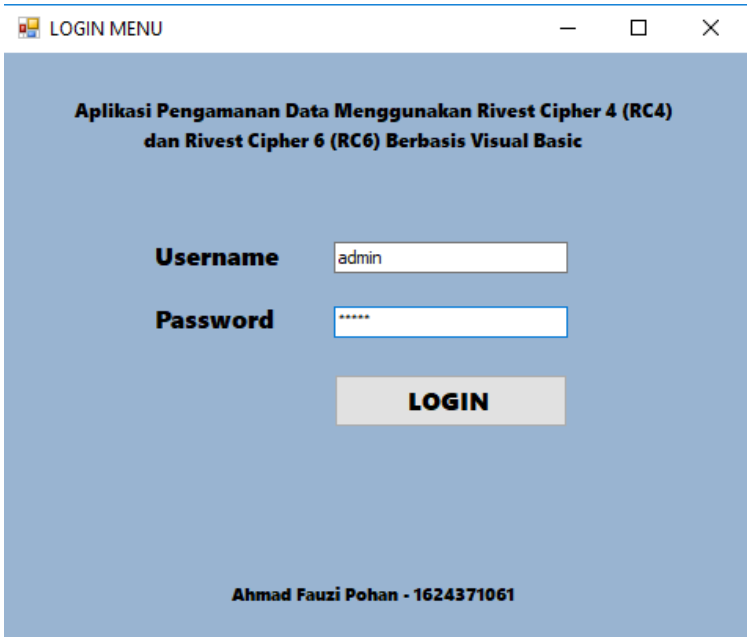
Pada tampilan *splash screen*, terdapat *background*, dan teks. Tampilan *splash screen* dapat dilhat pada gambar 4.1 berikut ini.



Gambar 4.1 Tampilan Halaman *Splash Screen*

1.5.2 Tampilan Halaman Menu Login

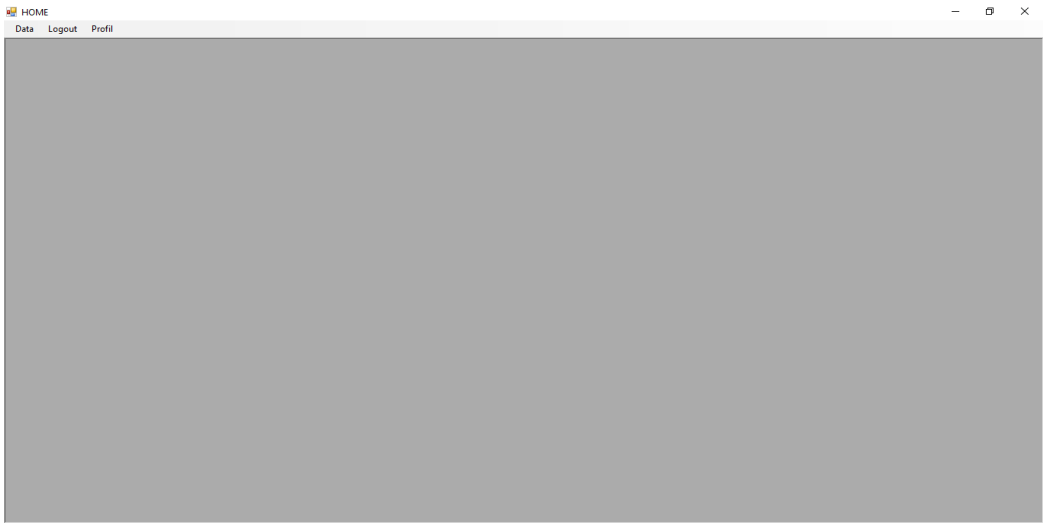
Tampilan *Menu Login*, menampilkan teks judul skripsi, *TextBox Username* dan *Password* untuk *login user*, kemudian ada *Button Login* untuk masuk ke menu utama serta ada teks nama penulis dan nim penulis. Untuk tampilannya dapat di lihat pada gambar 4.2 berikut.



Gambar 4.2 Tampilan Halaman Menu Login

1.5.3 Tampilan Halaman Menu Utama

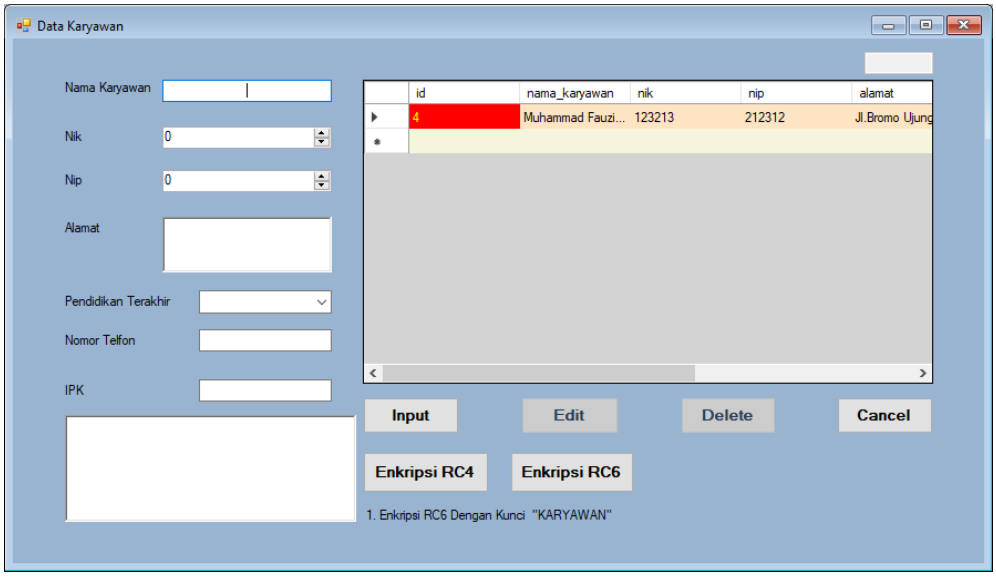
Setelah berhasil login, maka sistem akan masuk ke menu utama. Pada Tampilan *Menu Utama* berisi *Sub Menu* yaitu *Menu Data* yang terdiri dari Menu Data Karyawan (Enkripsi Data) dan Menu Dekripsi Data. Kemudian Menu *Logout* dan *Profile.* Untuk tampilannya dapat di lihat pada gambar 4.3 berikut:



Gambar 4.3 Tampilan Halaman Menu Utama

1.5.4 Tampilan Menu Enkripsi Data

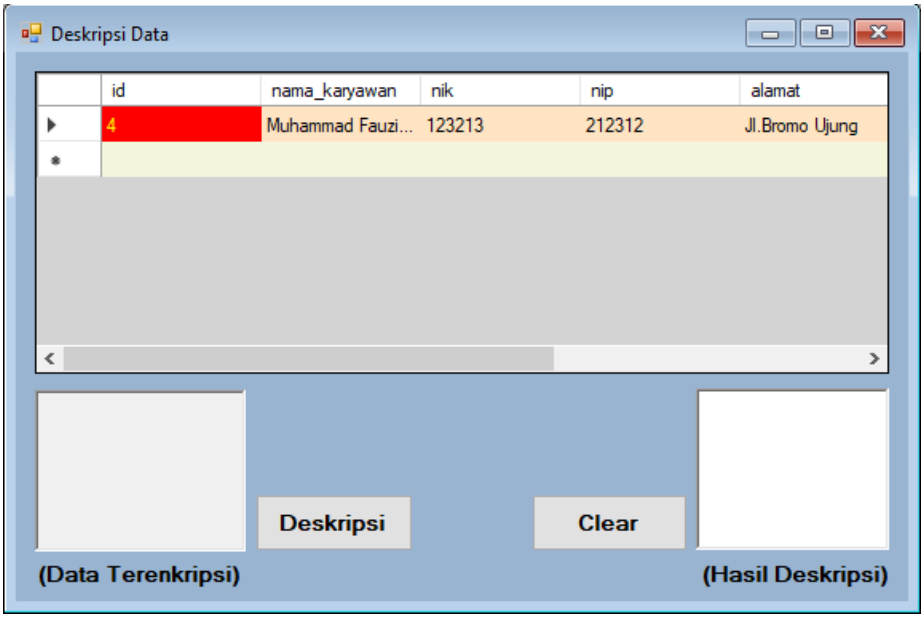
Pada menu enkripsi data, menampilkan sekumpulan data karyawan. Pada menu ini data karyawan dapat di *input, edit, delete.* Kemudian pada menu ini data karyawan dapat di enkripsi dengan algoritma RC4 dan algoritma RC6 dengan menekan *Button* Enkripsi RC4 dan atau / Enkripsi RC6. Dan kemudian hasil enkripsi akan ditampilkan pada *TextBox* hasil enkripsi. Adapun tampilan menunya pada gambar 4.4 berikut:



Gambar 4.4 Tampilan Halaman Menu Enkripsi Data

1.5.5 Tampilan Menu Dekripsi Data

Pada Menu Dekripsi Data, menampilkan *datagrid* kumpulan data karyawan yang telah terenkripsi sebelumnya. Kemudian ada *TextBox* yang menampilkan hasil data enkripsi, kemudian ada *Button* Dekripsi yang berfungsi untuk memproses dekripsi data dan ada *TextBox Hasil Dekripsi* yang berfungsi untuk menampilkan hasil dekripsi data. Adapun tampilannya pada gambar 4.5 sebagai berikut:



Gambar 4.5 Tampilan Halaman Menu Dekripsi Data

1.5.6 Tampilan Profil

Pada menu *Profil* menampilkan informasi Judul Skripsi, Nama Penulis, NPM Penulis, Fakultas, Jurusan dari Penulis, Nama Pembimbing I dan II serta Nama Universitas dan Tahun Penelitian Penulis. Adapun tampilannya menunya pada gambar 4.6 berikut:

Gambar 4.6 Tampilan Halaman Profil

1.6 Pengujian Sistem

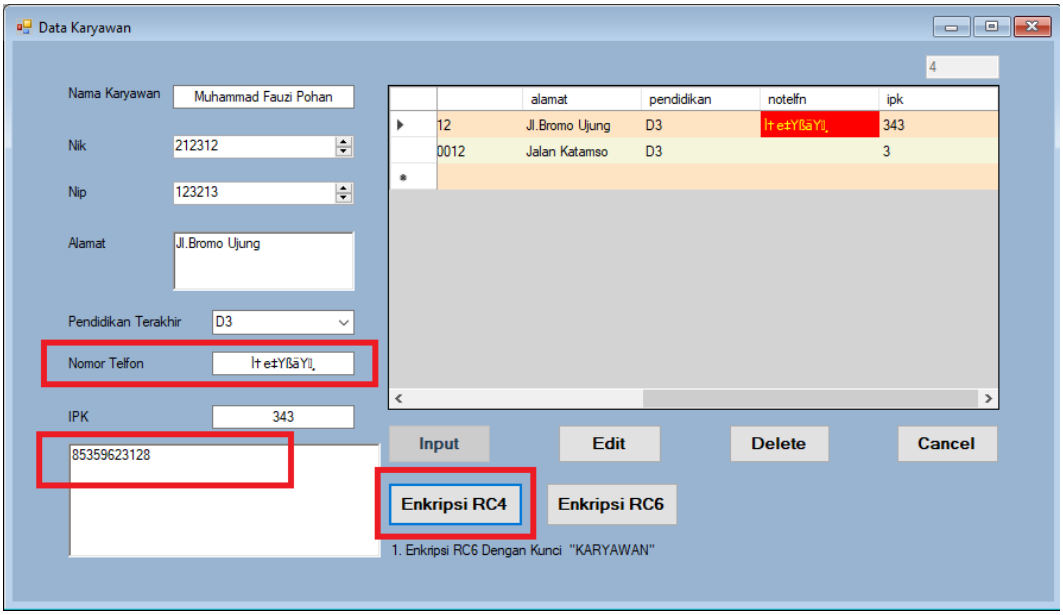
Pada tahap ini akan dilakukan pengujian sistem untuk mengetahui apakah sistem dapat berjalan sesuai dengan analisis dan perancangan sistem. Dan tujuan dari pengujian sistem ini berguna untuk mengetahui apa-apa saja yang akan dikembangkan pada penelitian selanjutnya dan sudah sejauh mana aplikasi dapat memfungsikan fitur-fitur sistem yang telah dirancang.

1.6.1 Pengujian Enkripsi Data

Pada tahap ini akan menguji sistem dalam proses enkripsi data. Adapun langkah-langkah pengujian enkripsi data sebagai berikut:

Untuk proses enkripsi data dengan algoritma RC4 sebagai berikut:

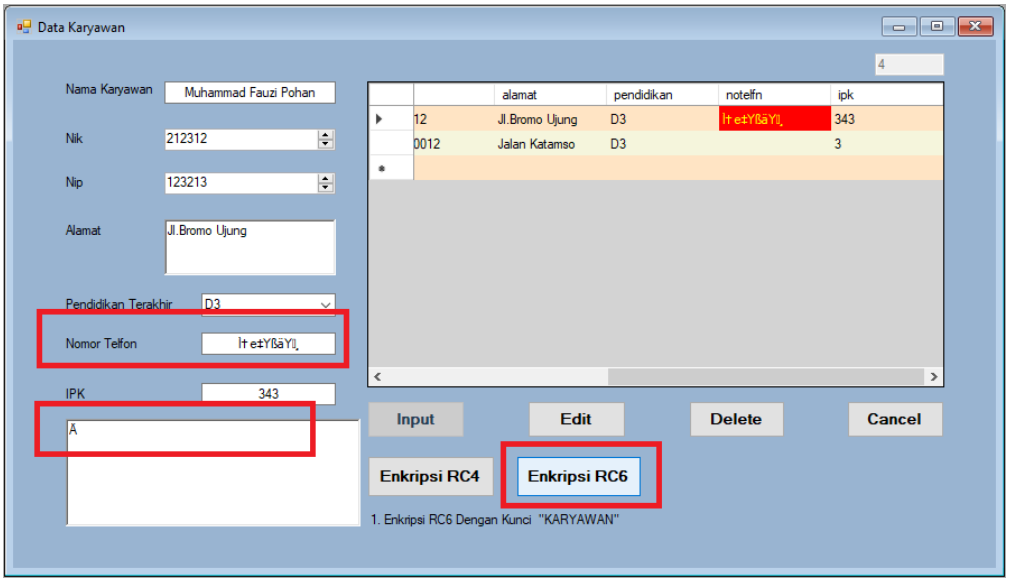
1. Langkah pertama yaitu, pengguna akan memilih data karyawan mana yang akan dienkripsi. Data yang akan dienkripsi dari data karywan tersebut yaitu data nomor telepon.
2. Kemudian setelah pengguna memilih nomor telepon karyawan, maka pengguna harus menekan *button Enkripsi RC4.*
3. Kemudian sistem akan menampilkan hasil enkripsi nomor telepon karyawan pada *TextBox Hasil Enkripsi Data.* Adapun tampilan proses dan hasil enkripsi data dengan Enkripsi RC4 pada gambar 4.7 berikut:



Gambar 4.7 Prosedur dan Hasil Enkripsi Data Dengan Algoritma RC4

Dan untuk proses enkripsi data dengan algoritma RC6 sebagai berikut:

1. Langkah pertama yaitu, pengguna akan memilih data karyawan mana yang akan dienkripsi. Data yang akan dienkripsi dari data karywan tersebut yaitu data nomor telepon.
2. Kemudian setelah pengguna memilih nomor telepon karyawan, maka pengguna harus menekan *button Enkripsi RC6.*
3. Kemudian sistem akan menampilkan hasil enkripsi nomor telepon karyawan pada *TextBox Hasil Enkripsi Data.* Adapun tampilan proses dan hasil enkripsi data dengan Enkripsi RC6 pada gambar 4.8 berikut:



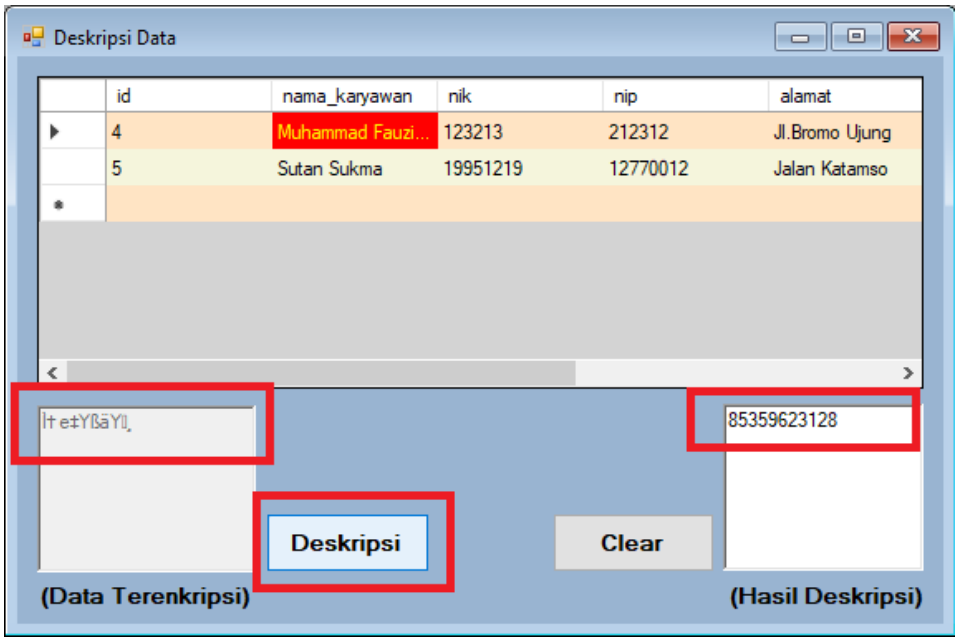
Gambar 4.8 Prosedur dan Hasil Enkripsi Data Dengan Algoritma RC6

1.6.2 Pengujian Dekripsi Data

Pada tahap ini akan menguji sistem dalam proses dekripsi data. Adapun langkah-langkah pengujian dekripsi data sebagai berikut:

1. Langkah pertama yaitu, pengguna akan masuk pada menu dekripsi data. Kemudian pengguna akan memilih data karyawan mana yang akan di dekripsi sesuai dengan hasil enkripsi data karyawan pada menu sebelumnya. Data yang akan di dekripsi dari data karyawan tersebut yaitu data nomor telepon karyawan yang sebelumnya telah dienkripsi di menu enkripsi data.
2. Kemudian setelah pengguna memilih nomor telepon karyawan, maka sistem akan menampilkan hasil enkripsi data sebelumnya pada *TextBox Data Terenkripsi*
3. Kemudian setelah data hasil enkripsi ditampilkan, maka pengguna harus menekan *button Dekripsi*.

4. Kemudian sistem akan menampilkan hasil dekripsi nomor telepon karyawan pada *TextBox Hasil Dekripsi Data.* Adapun tampilan proses dan hasil enkripsi data dengan Enkripsi RC4 pada gambar 4.7 berikut:



Gambar 4.9 Prosedur dan Hasil Dekripsi Data

1.7 Pengujian Dengan Metode *Black Box*

Pengujian dengan metode *blackbox* berfokus pada persyaratan fungsional dari sistem yang dibangun. Berikut ini adalah hasil pengujian sistem e-voting menggunakan metode *BlackBox* pada tabel 4.1 berikut :

Tabel 4.1 Hasil Pengujian Metode Blackbox

No	Komponen yang diuji	Butir Uji	Hasil Pengujian
1	*Splash Screen*	Menampilkan Splash Screen	Berhasil
2	Menu Login	Fungsi TextBox Username	Berhasil
		Fungsi TextBox Password	Berhasil

		Fungsi Button Login	Berhasil
3	Menu Utama	Menu Data	Berhasil
		Menu Logout	Berhasil
		Menu Profile	Berhasil
4	Menu Data	Data Karyawan (Enkripsi Data)	Berhasil
		Dekripsi Data	Berhasil
5	Menu Data Karyawan (Enkripsi Data)	TextBox Nama Karyawan	Berhasil
		TextBox Nik	Berhasil
		TextBox Nip	Berhasil
		TextBox Alamat	Berhasil
		TextBox Pendidikan Terakhir	Berhasil
		TextBox Nomor Telepon	Berhasil
		TextBox IPK	Berhasil
		TextBox Hasil Enkripsi	Berhasil
		Datagrid Data Karyawan	Berhasil
		Fungsi Button Input	Berhasil
		Fungsi Button Edit	Berhasil
		Fungsi Button Delete	Berhasil
		Fungsi Button Cancel	Berhasil
		Fungsi Button Enkripsi RC4	Berhasil
		Fungsi Button Enkripsi RC6	Berhasil
6	Menu Dekripsi Data	Menampilkan Datagrid Karyawan	Berhasil
		Tampilan TextBox Hasil Dekripsi Data	Berhasil
		Tampilan TextBox Hasil Enkripsi Data	Berhasil
		Fungsi Button Dekripsi Fungsi Button Clear	Berhasil
7	Menu Profil	Menampilkan Informasi Profil	Berhasil
8	Menu Logout	Keluar Aplikasi	Berhasil

1.8 Kesimpulan Hasil Pengujian Sistem

Setelah dilakukannya pengujian sistem baik itu dari pengujian sistem enkripsi data ataupun dekripsi data, maka dapat ditarik kesimpulan bahwa secara fungsional, sistem dapat berjalan dengan baik dan menghasilkan *output* sesuai dengan yang diharapkan serta sesuai dengan analisis dan perancangan siste

BAB V

PENUTUP

5.1 Kesimpulan

Setelah menganalisis, merancang, mengimplementasikan, dan melakukan pengujian sistem, maka diperoleh kesimpulan dari penelitian ini sebagai berikut:

1. Perancangan dan pembuatan aplikasi ini membutuhkan pengetahuan mengenai hal pemograman berorientasi objek dalam hal ini menggunakan pemograman *Visual Basic* yang masih dalam lingkungan berbasis *desktop.*
2. Algoritma RC4 dan RC6 yang dirancang untuk sistem dalam enkripsi dan dekripsi data mampu dan berhasil diimplementasikan pada pengamanan data.
3. Prosedur aplikasi dan alur sistem jelas dan tidak terlalu rumit sehingga pengguna tidak kesulitan dalam mengoperasikan sistem.
4. Sistem masih menggunakan IDE Microsoft Visual Studio 2010 sehingga sistem tidak terlalu berat ketika dioperasikan dan sistem tidak terlalu memakan kapasitas memory.
5. Aplikasi ini tergolong *User Friendly* walaupun tampilan masih terbilang sederhana.

5.2 Saran

Dengan selesainya pembuatan aplikasi ini penulis ingin memberikan saran dengan harapan yang nantinya dapat mendukung perngembangan sistem ini lebih lanjut:

1. Diharapkan untuk penelitian selanjutnya sistem diimplementasikan menggunakan algoritma yang lebih akurat hasilnya.
2. Diharapkan untuk penelitian selanjutnya sistem dapat diujikan dengan data yang lebih banyak agar akurasi ketetapan lebih dihasilkan.
3. Diharapkan untuk penelitian selanjutnya sistem dapat diimplementasikan dengan berbasis android.

DAFTAR PUSTAKA

Ariyus, D. 2008. Pengantar Ilmu Kriptografi: Teori, Analisis dan Implementasi. Andi Offeset: Yogyakarta.

Barus, S., Sitorus, V. M., Napitupulu, D., Mesran, M., & Supiyandi, S. (2018). Sistem Pendukung Keputusan Pengangkatan Guru Tetap Menerapkan Metode Weight Aggregated Sum Product Assesment (WASPAS). Jurnal Media Informatika Budidarma, 2(2).

Dharwiyanti, 2003. Sri. Pengantar Unified Modeling Language (UML): Jakarta.

Dony Ariyus, 2006. Keamaman Kriftografi: Bandung.
Fathansyah, Ir, 2009. Basis Data, Informatika,
Bandung.

Fiansyah, E. 2008. Implementasi algoritma dasar RC4 stream cipher dan pengacakan plaintext dengan teknik dynamic blocking pada aplikasi sistem informasi kegiatan skripsi di departemen teknik elektro. Skripsi Universitas Indonesia: 1-94. (Online) http://lib.ui.ac.id/file?file=digital/126532-R030856.pdf (Diakses 12 November 2018).

Hartono, Budi. Ruang Lingkup Kriptografi untuk mengamankan data. Edisi Mei 2004, volume ix, no. 2.

Hartanto, S. (2017). Implementasi fuzzy rule based system untuk klasifikasi buah mangga. *TECHSI-Jurnal Teknik Informatika*, *9*(2), 103-122.

Hendrayudi, 2010, Dasar-Dasar Pemograman Microsoft Visual Basic 2008, Bandung, Satu Nusa.

Imron Rozidi, Romzi. 2004. Membuat Sendiri Sms Gateway Berbasis Protocol Smpp. Andi.

Khairul, k., ilhamiarsyah, u., wijaya, r. F., & utomo, r. B. (2018, september). Implementasi augmented reality sebagai media promosi penjualan rumah. In *seminar nasional royal (senar)* (vol. 1, no. 1, pp. 429-434).

Lusmiarwan, Driana, 2006. ”Perancangan Prototype Single Identity Number (SIN) Untuk Menunjang E-Goverment”, Bandung.

Muhammad Fairuzabadi, Jurnal Dinamika Informatika, Volume 4, 2010:66

Munir, Rinaldi. 2006. Kriptografi. Informatika. Bandung.

Prayudi, Yudi, Idham Halik. 2005. Studi Analisis Algoritma Rivest Code 6 (RC6) Dalam Enkripsi/ Dekripsi Data. Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNATI 2005), Yogyakarta.

Putri, R. E., & Siahaan, A. (2017). Examination of document similarity using Rabin-Karp algorithm. *International Journal of Recent Trends in Engineering & Research*, *3*(8), 196-201.

Rahim, R., Supiyandi, S., Siahaan, A. P. U., Listyorini, T., Utomo, A. P., Triyanto, W. A., & Khairunnisa, K. (2018, June). TOPSIS Method Application for Decision Support System in Internal Control for Selecting Best Employees. In Journal of Physics: Conference Series (Vol. 1028, No. 1, p. 012052). IOP Publishing.

Rahmanto, Arif, 2012. Enkripsi Sms (Short Message Service) Dengan Menggunakan Algoritma Rc6 Pada Sistem Operasi Android.

Rifiki, Sadikin. 2012. Kriptografi Untuk Keamanan Jaringan Dan Implementasi Dalam Bahasa Java. Andi.

Rika, Safrina. 2006. Studi Dan Perbandingan Sistem Penyandian Pesan Dengan Algoritma Rc2, Rc4, Rc5, Dan Rc6.

Rizal, Ansar, Suharto. 2011. Implementasi Algoritma RC4 Untuk Keamanan Login Pada Sistem Pembayaran Uang Sekolah. Dielektrika, ISSN 2086-9487 Vol. 2 No.2

Safaat, Nazruddin. 2011. Pemograman Aplikasi Mobile Smartphone Dan Tablet Pc Berbasis Android.Informatika. Bandung.

Sari, R. D., Supiyandi, A. P. U., Siahaan, M. M., & Ginting, R. B. (2017). A Review of IP and MAC Address Filtering in Wireless Network Security. Int. J. Sci. Res. Sci. Technol, 3(6), 470-473.

Setiadi, Herald, 2006. ”Database Kependudukan Nasional Sebagai Prasyarat Untuk Pelaksanaan Good Governance”, Bandung.

Siahaan, MD Lesmana, Melva Sari Panjaitan, and Andysah Putera Utama Siahaan. "MikroTik bandwidth management to gain the users prosperity prevalent." *Int. J. Eng. Trends Technol* 42.5 (2016): 218-222.

Siahaan, A. P. U., Aryza, S., Nasution, M. D. T. P., Napitupulu, D., Wijaya, R. F., & Arisandi, D. (2018). Effect of matrix size in affecting noise reduction level of filtering.

Sidik, A. P. (2018). Algoritma RSA dan Elgamal sebagai Algoritma Tambahan untuk Mengatasi Kelemahan Algoritma One Time Pad pada Skema Three Pass Protocol.

Suharno. 2005 "Menuju Terciptanya Single Identification Number di Indonesia", Jakarta.

Sukardi, Yanto. Setiawan Ridwan. 2011. Studi Perbandingan Metode Hash Md5, Huffman Dan Rc6 Untuk Pengenkripsian Dan Kompresi Data Teks Sms.

Suprianto, Dodit.Agustina Rin, 2012. Pemograman Aplikasi Android. Mediakom.

Tasril, V. (2018). Sistem Pendukung Keputusan Pemilihan Penerimaan Beasiswa Berprestasi Menggunakan Metode Elimination Et Choix Traduisant La Realite. INTECOMS: Journal of Information Technology and Computer Science, 1(1), 100-109.

Weerasinghe, T.D.B. 2013. An Affective RC4 Stream Cipher. Industrial and Information System (ICIIS), IEEE International 8: 69-74

Sidik, A. P. (2018). Algoritma RSA dan Elgamal sebagai Algoritma Tambahan untuk Mengatasi Kelemahan Algoritma One Time Pad pada Skema Three Pass Protocol.

Suharno. 2005 "Menuju Terciptanya Single Identification Number di Indonesia", Jakarta.

Sukardi, Yanto. Setiawan Ridwan. 2011. Studi Perbandingan Metode Hash Md5, Huffman Dan Rc6 Untuk Pengenkripsian Dan Kompresi Data Teks Sms.

Suprianto, Dodit.Agustina Rin, 2012. Pemograman Aplikasi Android. Mediakom.

Tasril, V. (2018). Sistem Pendukung Keputusan Pemilihan Penerimaan Beasiswa Berprestasi Menggunakan Metode Elimination Et Choix Traduisant La Realite. INTECOMS: Journal of Information Technology and Computer Science, 1(1), 100-109.

Weerasinghe, T.D.B. 2013. An Affective RC4 Stream Cipher. Industrial and Information System (ICIIS), IEEE International 8: 69-74.

Widodo, Reza. Brianca. 2011. Proposal Makalah if3058. Kriptografi Studi Dan Perbandingan Algoritma Rc6 Dan Blowfish.

Zimmermann, P. 1998. An Introduction to Cryptography. Network Associates, Inc: Santa Clara.