ENKRIPSI DAN DEKRIPSI DENGAN MENGGUNAKAN METODE KRIPTOGRAFI VERNAM CIPHER (XOR)

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	:	**YUDHA IRNANDA**
NPM	:	**1414370274**
PROGRAM STUDI	:	**SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019

ABSTRAK

YUDHA IRNANDA

Enkripsi dan Dekripsi Dengan Menggunakan Metode Kriptografi *Vernam Cipher* (XOR)

2019

Kriptografi sebagai salah satu cabang ilmu yang dapat digunakan untuk mengamankan data hingga saat ini terus dikembangkan melalui berbagai algoritma. Beberapa penelitian terkait mengenai kriptografi masih menggunakan media berupa teks saja, *image* saja, maupun *file* tertentu saja. Pada penelitian ini akan digunakan media berupa teks saja sebagai media *inputan*. Adapun algoritma yang digunakan yaitu *Vernam Cipher*. Algoritma ini dikenal cepat, mudah dan aman untuk digunakan. Percobaan yang dilakukan menggunakan *file notepad* serta telah diuji melalui aplikasi yang dibangun dengan *Visual Studio* 2010 telah menghasilkan proses enkripsi dan dekripsi data yang berjalan dengan baik. *File* hasil enkripsi dapat dibuka dengan kunci yang telah ditetapkan dan tidak mengalami kerusakan dan sebaliknya untuk proses dekripsi data juga demikian.

Kata Kunci : *File*, Kriptografi, *Vernam Cipher.*

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

KATA PENGANTAR

Assallamuallaikum Wr. Wb.

Puji syukur kehadirat Allah SWT, yang telah melimpahkan rahmat dan hidayahNya sehingga penulis dapat menyelesaikan skripsi ini dengan judul " Enkripsi dan Dekripsi Menggunakan Metode Kriptografi *Vernam Cipher* (XOR)*"*

Skripsi ini disusun serta disajikan untuk memenuhi persyaratan ujian akhir dengan memperoleh gelar sarjana pada Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.

Penulis menyadari bahwa skripsi ini jauh dari kesempurnaan, oleh karena itu segala kritik dan saran yang sifatnya membangun sangat penulis harapkan dari pembaca demi kesempurnaan skripsi ini. Selama dalam penyusunan skripsi ini, penulis banyak menerima bimbingan, bantuan, masukan dan dorongan yang sangat berarti. Penulis pada kesempatan ini menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Kepada ALLAH SWT yang selalu memberikan saya kesehatan dan semangat sehingga dapat menyelesaikan skripsi ini.
2. Kepada kedua orang tua yang telah menjaga dan mengasihi saya dari kecil hingga dewasa.
3. Kepada saudara-saudaraku yang telah memberikan support baik berupa bimbingan dan materi, merekalah pengganti kedua orang tua.
4. Bapak Dr. H.M. Isa Indrawan, SE.,MM., selaku Rektor Universitas Pembangunanan Panca Budi Medan.
5. Ibu Sri Shindi Indira S.T., M.SC., selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
6. Bapak Eko Hariyanto, S.Kom., M.Kom., selaku Ketua Program Studi Sistem Komputer Universitas Pembangunan Panca Budi Medan.
7. Bapak Andysah Putera Utama Siahaan, S.Kom., M.Kom., Ph.D., selaku Dosen Pembimbing I Universitas Pembangunan Panca Budi Medan.
8. Bapak Eko Hariyanto, S.Kom., M.Kom., selaku Dosen Pembimbing II Universitas Pembangunan Panca Budi Medan.
9. Dosen-dosen pada Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
10. Seluruh staf dan karyawan pada Universitas Pembangunan Panca Budi Medan.
11. Para sahabat dan teman yang selalu mendampingi saya disaat susah dan senang dan teman-teman yang telah memberikan semangat dan motivasi dalam penyelesaian skripsi ini.

Akhir kata penulis sampaikan rasa terima kasih bagi semua pihak yang secara langsung terlibat dalam penyelesaian skripsi ini yang tidak dapat penulis sebutkan satu persatu. Semoga skripsi ini memberikan manfaat bagi penulis khususnya dan bagi kita semua umumnya.

Medan, September 2019
Penulis,

YUDHA IRNANDA
1414370274

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kriptografi bagi kebanyakan orang adalah sesuatu yang sangat sulit dan kita sebagai pemula cenderung malas untuk mempelajarinya. Namun ada sebuah metode kriptografi yang agak mudah untuk dipelajari dan para ahli pun telah menyatakan bahwa metode ini merupakan metode kriptografi yang cukup aman untuk digunakan. Metode tersebut biasa dikenal dengan nama *One Time Pad* (OTP) atau yang lebih sering dikenal dengan sebutan *Vernam Cipher*. *Vernam Cipher* diciptakan oleh Mayor J. Maugborne dan G. Vernam pada tahun 1917.

Dalam proses enkripsi, algoritma ini menggunakan cara *stream cipher* yang berasal dari hasil XOR antara *bit plaintext* dan *bit key*. Pada metode ini *plaintext* diubah ke dalam kode ASCII dan kemudian dikenakan operasi XOR terhadap kunci yang sudah diubah ke dalam kode ASCII.

Algoritma dari enkripsi adalah fungsi-fungsi yang digunakan untuk melakukan fungsi enkripsi dan dekripsi. Algoritma yang digunakan menentukan kekuatan dari enkripsi, dan ini biasanya dibuktikan dengan basis matematika. Berdasarkan cara memproses teks (*plaintext*), *cipher* dapat dikategorikan menjadi dua jenis : *block cipher* dan *stream cipher*. *Block cipher* bekerja dengan memproses data secara blok, dimana beberapa karakter/data digabungkan menjadi satu blok. Setiap proses satu blok menghasilkan keluaran satu blok juga.

Sementara itu *stream cipher* bekerja memproses masukan (karakter atau data) secara terus-menerus dan menghasilkan data pada saat yang bersamaan.

Suatu *One Time Pad* diciptakan dengan men-*generate* suatu *string* yang terdiri dari karakter-karakter atau angka-angka yang panjangnya harus minimal sama dengan kata terpanjang dalam pesan yang akan dienkripsikan. *String* ini di-*generate* secara acak atau *random*, misalnya dengan menggunakan *random number generator* pada komputer. *String* tersebut kemudian dituliskan pada suatu *pad*. *Pad-pad* tersebut kemudian diberikan kepada siapapun yang ingin menggunakannya untuk mengirim ataupun menerima pesan.

Aplikasi yang akan dibuat oleh penulis adalah dengan menggunakan *visual studio* 2010 dengan menggunakan kriptografi *Vernam Cipher* agar dapat mengenkripsi dan dekripsi data teks yang akan digunakan secara rahasia dan lebih mudah digunakan oleh *user* nantinya. Berdasarkan latar belakang di atas maka penulis tertarik untuk memilih judul **"Enkripsi Dan Dekripsi Dengan Menggunakan Metode Kriptografi *Vernam Cipher* (XOR)"**.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas maka rumusan masalah adalah sebagai berikut :

a. Bagaimana merancang sebuah *software* pengamanan informasi teks dengan menggunakan *Visual Studio* 2010 ?
b. Bagaimana membuat enkripsi dan dekripsi informasi dengan menggunakan *Vernam Cipher* ?

1.3 Batasan Masalah

Dalam perancangan aplikasi kriptografi *Vernam Cipher* ini penulis membatasi masalah sebagai berikut :

a. Metode yang digunakan pada perancangan aplikasi pengamanan informasi ini menggunakan metode *Vernam Cipher* untuk enkripsi dan dekripsi teks.
b. Bahasa pemrograman yang digunakan dalam perancangan aplikasi kriptografi *Vernam Cipher* ini adalah *Visual Studio*.NET 2010 dan *database* MySQL.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai penulis dalam perancangan aplikasi kriptografi *Vernam Cipher* ini adalah :

a. Untuk mengubah pengiriman data.
b. Membuat suatu aplikasi kriptografi yang mengimplementasikan algoritma *Vernam Cipher* sehingga dapat mengatasi masalah keamanan informasi serta menjaga kerahasiaan data.

1.5 Manfaat Penelitian

Perancangan aplikasi kriptografi *Vernam Cipher* ini bermanfaat bagi masyarakat luas antara lain :

a. Mempermudah bagi pengguna untuk mengenkripsi data teks yang akan digunakan secara rahasia.

b. Aplikasi kriptografi *Vernam Cipher* ini dapat digunakan oleh semua kalangan masyarakat luas agar dapat membuat teks yang dapat dikunci dengan kata khusus.

1.6 Metodologi Penelitian

Metode pengumpulan data yang digunakan dalam penelitian ini adalah metode deskriptif. Adapun teknik pengumpulan data dilakukan dengan cara sebagai berikut :

a. Studi *Literature*

   Pengumpulan data dengan cara mengumpulkan *literature,* jurnal, *paper* dan bacaan-bacaan yang ada kaitannya dengan judul penelitian.

b. Studi Pustaka

   Pengumpulan data dengan menggunakan atau mengumpulkan sumber-sumber tertulis, dengan cara membaca, mempelajari dan mencatat hal-hal penting yang berhubungan dengan masalah yang sedang dibahas guna memperoleh gambaran secara teoritis.

1.7 Sistematika Penulisan

Adapun struktur penulisan pada masing-masing bab dalam laporan tugas akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Membahas Latar Belakang Masalah, Rumusan Masalah, Batasan Masalah, Tujuan dan Manfaat Penelitian, Metodelogi Penelitian dan

Sistematika Penulisan.

BAB II LANDASAN TEORI

Memaparkan teori-teori yang didapat dari sumber-sumber yang relevan untuk digunakan sebagai panduan dalam penelitian serta penyusunan skripsi.

BAB III METODE PENELITIAN

Menjelaskan tentang gambaran sistem serta deskripsi dari hasil analisis sistem yang akan dijadikan sebagai petunjuk untuk perancangan sistem selanjutnya.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menguraikan langkah-langkah dalam implementasi sistem, disertai dengan komponen-komponen kebutuhan sistem.

BAB V PENUTUP

Mengemukakan kesimpulan yang diambil dari hasil penelitian dan perancangan sistem, serta saran-saran untuk pengembangan selanjutnya, agar dapat dilakukan perbaikan-perbaikan dimasa yang akan datang.

BAB II

LANDASAN TEORI

2.1 Kriptografi

Kriptografi adalah metode melindungi informasi dan komunikasi melalui penggunaan kode sehingga hanya mereka yang dituju informasi yang dapat membaca dan memprosesnya. *Pre-fix* "*crypt*" berarti "*hidden*" atau "*vault*" dan "*graph*" *suffix* adalah singkatan dari "*writing*". Dalam ilmu komputer, kriptografi mengacu pada keamanan informasi dan teknik komunikasi yang berasal dari konsep matematika dan seperangkat perhitungan berbasis aturan yang disebut algoritma untuk mengubah pesan dengan cara yang sulit untuk diuraikan. Algoritma deterministik ini digunakan untuk pembuatan kunci kriptografis dan penandatanganan digital dan verifikasi untuk melindungi privasi data, penelusuran *web* di internet, dan komunikasi rahasia seperti transaksi kartu kredit dan email. (Zelvina, 57 : 2014)

Jika transformasinya dapat dikembalikan, kriptografi juga dapat diartikan sebagai proses mengubah kembali data yang terenkripsi menjadi bentuk yang mudah dipahami. Sehingga, kriptografi juga dapat diartikan sebagai proses untuk melindungi data dalam arti yang luas. Pengertian kriptografi dalam kamus bahasa Inggris *Oxford* adalah sebuah teknik rahasia dalam penulisan, dengan karakter khusus, dengan menggunakan huruf dan karakter diluar bentuk aslinya atau dengan metode-metode lain yang hanya dapat dipahami oleh pihak-pihak yang

memproses kunci, juga semua hal yang ditulis dengan cara seperti ini. Jadi, secara umum kriptografi diartikan sebagai seni menulis atau memecahkan *cipher*.

Dalam perkembangannya, kriptografi juga digunakan untuk mengidentifikasi pengiriman pesan dan tanda tangan digital dan keaslian pesan dengan sidik jari digital. (Dony Ariyus, 2013)

Di dalam kriptografi kita akan sering menemukan berbagai istilah atau *terminology*. Beberapa istilah yang harus diketahui yaitu :

a. Pesan (*message*)

   Plaintext adalah teks atau pesan yang tidak dienkripsi dalam bentuk aslinya yang dapat dibaca manusia. Teks biasa adalah *input* dari proses enkripsi dan *output* dari proses dekripsi. Disebut juga teks jernih, ini adalah kebalikan dari teks sandi. *Ciphertext* adalah teks terenkripsi. *Plaintext* adalah apa yang anda miliki sebelum enkripsi, dan *ciphertext* adalah hasil terenkripsi. Istilah *cipher* kadang-kadang digunakan sebagai sinonim untuk *ciphertext*, tetapi lebih tepat berarti metode enkripsi daripada hasilnya.

b. Pengirim dan Penerima

   Dalam proses komunikasi, "penerima" adalah pendengar, pembaca, atau pengamat — yaitu, individu (atau kelompok individu) yang kepadanya pesan diarahkan. Penerima juga disebut "audiens" atau "*decoder"*. Orang yang memprakarsai pesan dalam proses komunikasi disebut "pengirim." Pengirim adalah pencetus pesan pada acara tertentu; penerima adalah audiens mereka pada kegiatan tersebut.

c. Enkripsi dan Dekripsi

Dalam kriptografi, enkripsi adalah proses penyandian pesan atau informasi sedemikian rupa sehingga hanya pihak yang berwenang yang dapat mengaksesnya dan mereka yang tidak berwenang tidak dapat melakukannya. Enkripsi itu sendiri tidak mencegah gangguan tetapi menyangkal konten yang dapat dipahami oleh calon pencegat. Dekripsi pada umumnya adalah proses kebalikan dari enkripsi. Ini adalah proses mendekode data yang telah dienkripsi ke dalam format rahasia. Pengguna yang diotorisasi hanya dapat mendekripsi data karena dekripsi memerlukan kunci atau kata sandi rahasia.

d. *Cipher* dan Kunci

Dalam kriptografi, *cipher* adalah algoritma untuk melakukan enkripsi atau dekripsi — serangkaian langkah yang didefinisikan dengan baik yang dapat diikuti sebagai prosedur. Alternatif, istilah yang kurang umum adalah enkripsi. Untuk mengenkripsi atau menyandikan berarti mengubah informasi menjadi sandi atau kode. Dalam bahasa umum, "sandi" identik dengan "kode", karena keduanya adalah serangkaian langkah yang mengenkripsi pesan; Namun, konsepnya berbeda dalam kriptografi, terutama kriptografi klasik. Kode umumnya menggantikan *string* panjang karakter yang berbeda dalam *output*, sementara *cipher* umumnya mengganti jumlah karakter yang sama seperti *input*. Ada pengecualian dan beberapa sistem sandi dapat menggunakan sedikit lebih banyak, atau lebih sedikit, karakter ketika *output versus* jumlah yang dimasukkan.

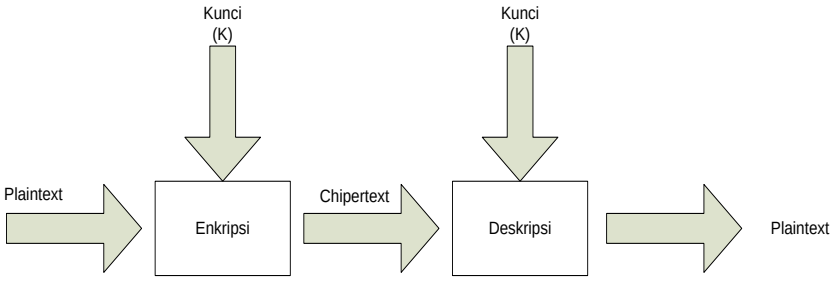
Kode dioperasikan dengan mengganti menurut buku kode besar yang menautkan *string* acak karakter atau angka ke kata atau frasa. Misalnya, "UQJHSE" bisa menjadi kode untuk "Lanjutkan ke koordinat berikut." Saat menggunakan *cipher*, informasi asli dikenal sebagai *plaintext* dan bentuk terenkripsi sebagai *ciphertext*. Pesan *ciphertext* berisi semua informasi dari pesan *plaintext* tetapi tidak dalam format yang dapat dibaca oleh manusia atau komputer tanpa mekanisme yang tepat untuk mendekripsi. Operasi *cipher* biasanya tergantung pada sepotong informasi tambahan, yang disebut kunci (atau, dalam bahasa NSA tradisional, variabel kripto). Prosedur enkripsi bervariasi tergantung pada kunci, yang mengubah operasi rinci dari algoritma. Kunci harus dipilih sebelum menggunakan sandi untuk mengenkripsi pesan. Tanpa mengetahui kunci, seharusnya sangat sulit, jika bukan tidak mungkin, untuk mendekripsi *ciphertext* yang dihasilkan menjadi *plaintext* yang dapat dibaca. Dengan menggunakan K, fungsi enkripsi dan dekripsi dapat ditulis sebagai berikut:

$$E\ K\ (P) = C\ D\ K\ (C) = \mathrm{P}$$

Keterangan :

P	=	*plaintext*
C	=	*ciphertext*
K	=	kunci
EK	=	proses enkripsi menggunakan kunci *K*
DK	=	proses dekripsi menggunakan kunci *K*

Skema enkripsi dengan menggunakan kunci diperlihatkan pada gambar dibawah ini :



Gambar 2.1. Skema enkripsi dan dekripsi

Sumber : (Dony Ariyus, 2013)

e. Sistem Kriptografi

Cryptosystem adalah seperangkat algoritma kriptografi yang diperlukan untuk mengimplementasikan layanan keamanan tertentu, paling umum untuk mencapai kerahasiaan (enkripsi). Biasanya, *cryptosystem* terdiri dari tiga algoritma: satu untuk pembuatan kunci, satu untuk enkripsi, dan satu untuk dekripsi.

f. Penyadap (*eavesdropper*)

Penyadap adalah orang yang mencoba menangkap pesan selama ditransmisikan. Penyadap adalah tindakan diam-diam atau diam-diam mendengarkan percakapan pribadi atau komunikasi orang lain tanpa persetujuan mereka. Praktik ini secara luas dianggap tidak etis, dan di banyak yurisdiksi adalah ilegal.

g. Kriptanalisis dan Kriptologi

Kriptanalisis adalah dekripsi dan analisis kode, sandi atau teks terenkripsi. *Cryptanalysis* menggunakan rumus matematika untuk mencari kerentanan algoritma dan membobol sistem keamanan informasi atau kriptografi.

2.1.1 Algoritma kriptografi

Cryptosystems menggunakan seperangkat prosedur yang dikenal sebagai algoritma kriptografi, atau *ciphertext*, untuk mengenkripsi dan mendekripsi pesan untuk mengamankan komunikasi antara sistem komputer, perangkat seperti *smartphone*, dan aplikasi. Sebuah *ciphertext* menggunakan satu algoritma untuk enkripsi, algoritma lain untuk otentikasi pesan dan lainnya untuk pertukaran kunci. Proses ini, tertanam dalam protokol dan ditulis dalam perangkat lunak yang berjalan pada sistem operasi dan sistem komputer jaringan, melibatkan generasi kunci publik dan swasta untuk enkripsi / dekripsi data, penandatanganan digital dan verifikasi untuk otentikasi pesan, dan pertukaran kunci.

2.1.2 Teknik Kriptografi

Kriptografi terkait erat dengan disiplin ilmu kriptologi dan kriptanalisis. Ini mencakup teknik seperti mikrodot, menggabungkan kata-kata dengan gambar, dan cara-cara lain untuk menyembunyikan informasi dalam penyimpanan atau transit. Namun, di dunia komputer-sentris saat ini, kriptografi paling sering dikaitkan dengan pengacakan *plaintext* (teks biasa, kadang-kadang disebut sebagai *cleartext*) menjadi *ciphertext* (proses yang disebut enkripsi), kemudian

kembali lagi (dikenal sebagai dekripsi). Individu yang berlatih bidang ini dikenal sebagai *cryptographers*.

Kriptografi modern berkaitan dengan empat tujuan berikut:

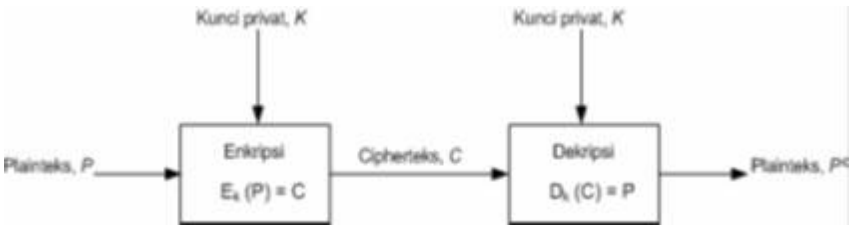
a. Kerahasiaan: informasi tidak dapat dipahami oleh siapa pun yang tidak disengaja
b. Integritas: informasi tidak dapat diubah dalam penyimpanan atau transit antara pengirim dan penerima yang dituju tanpa perubahan yang terdeteksi
c. Non-repudiation: pencipta / pengirim informasi tidak dapat menyangkal pada tahap selanjutnya niatnya dalam pembuatan atau transmisi informasi
d. Otentikasi: pengirim dan penerima dapat mengkonfirmasi identitas satu sama lain dan asal / tujuan informasi
e. Prosedur dan protokol yang memenuhi beberapa atau semua kriteria di atas dikenal sebagai *cryptosystems*. *Cryptosystems* sering dianggap hanya merujuk pada prosedur matematika dan program komputer; namun, mereka juga memasukkan pengaturan perilaku manusia, seperti memilih kata sandi yang sulit ditebak, keluar dari sistem yang tidak digunakan, dan tidak membahas prosedur sensitif dengan orang luar.

2.1.3 Jenis-jenis Kriptografi

Berdasarkan kunci yang digunakan untuk enkripsi dan dekripsi, kriptografi dapat dibedakan menjadi 2 macam, yaitu kriptografi simetri (*symetric cryptography*) dan kriptografi asimetri (*asymetric cryptography*).

1. Kriptografi Simetri (*Symetric Cryptography*)

Pada sistem kriptografi simetri, kunci untuk proses enkripsi sama dengan kunci pada proses dekripsi. Keamanan sistem kriptografi simetri terletak pada kerahasiaan kunci. Istilah lain untuk kriptografi simetri adalah kriptografi kunci privat (*privat key cryptography*) atau kriptografi konvensional (*conventional cryptography*).



Gambar 2.2. Kriptografi Simetri (*Symetric Cryptography*)

Sumber : (Zelvina, 58 : 2014)

Algoritma kriptografi simetri dapat dikelompokkan menjadi dua kategori antara lain :

a. *Cipher* aliran (*stream cipher*)

   Algoritma kriptografi beroperasi pada *plaintext*/*ciphertext* dalam bentuk *bit* tunggal yang dalam hal ini rangkaian *bit* dienkripsikan/didekripsikan *bit* per *bit*. *Cipher* aliran mengenkripsi satu *bit* setiap kali.

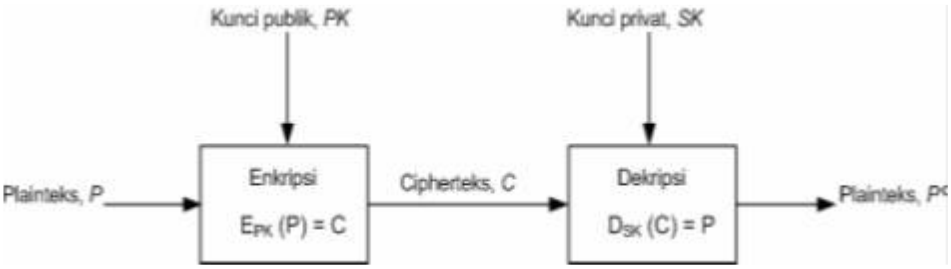
b. *Cipher* blok (*block cipher*)

   Algoritma kriptografi beroperasi pada *plaintext*/*ciphertext* dalam bentuk blok *bit*, yang dalam hal ini rangkaian *bit* dibagi menjadi

blok-blok *bit* yang panjangnya sudah ditentukan sebelumnya. *Cipher* blok mengenkripsi satu blok *bit* setiap kali.

2. Kriptografi Asimetri (*Asymetric Cryptography*)

   Pada sistem kriptografi asimetri, kunci untuk proses enkripsi tidak sama dengan kunci untuk proses dekripsi. Istilah lain untuk kriptografi asimetri adalah kriptografi kunci public (*public key cryptography*), sebab kunci untuk enkripsi tidak rahasia dan dapat diketahui oleh siapapun, sementara kunci untuk dekripsi hanya diketahui oleh penerima pesan.



Gambar 2.3. Kriptografi Asimetri (*Asymetric Cryptography*)

Sumber : (Zelvina, 78 : 2014)

2.2 Vernam Cipher

Vernam Cipher didasarkan pada prinsip bahwa setiap karakter *plaintext* dari sebuah pesan 'dicampur' dengan satu karakter dari *keystream*. Jika *keystream* yang benar-benar acak digunakan, hasilnya akan menjadi *ciphertext* yang benar-benar 'acak' yang tidak ada hubungannya dengan *plaintext* asli. Dalam hal ini, *cipher* mirip dengan *One-Time Pad* (OTP) yang tidak dapat dipecahkan. Seperti

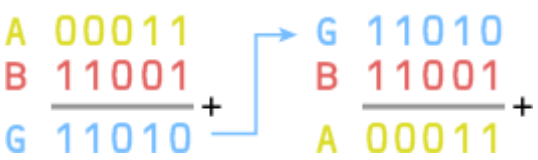
yang umumnya digunakan dengan *teleprinter* dan 5-*level tape*, sistem ini juga dikenal sebagai *One-Time Tape* atau OTT.

Jika *ciphertext* yang dihasilkan dalam sistem OTT yang diuraikan di atas benar-benar acak, ia dapat dengan aman dikirim melalui udara, tanpa risiko diuraikan oleh *eavesdropper*. Yang harus dilakukan penerima adalah mencampur *ciphertext* dengan OTT yang sama untuk mengungkapkan teks asli. Seseorang hanya harus menjamin bahwa OTT benar-benar acak, bahwa hanya ada dua salinannya, bahwa kedua salinan itu dihancurkan segera setelah digunakan dan bahwa mereka hanya digunakan satu kali.

Ciphertext dihasilkan dengan menerapkan operasi XOR logis (eksklusif-atau) ke *bit* individu *plaintext* dan *keystream*. Keuntungan menggunakan operasi XOR untuk ini, adalah dapat dikembalikan, cukup dengan melakukan operasi yang sama lagi. Dengan kata lain:

$$plaintext + key = ciphertext \Rightarrow ciphertext + key = plaintext$$

Dalam matematika, operasi XOR dikenal sebagai penambahan modulo-2. Dalam kasus ini, *bit* individual dari *plaintext* adalah XOR-ed dengan *bit* individu dari kunci. *Bit* yang dihasilkan hanya akan menjadi '1' jika dua *bit input* berbeda. Jika keduanya sama (keduanya 1 atau keduanya 0), hasilnya adalah '0'.



Ambil huruf 'A', yang diwakili oleh 00011, dan tambahkan ke huruf 'B', diwakili oleh 11001. Operasi XOR yang sedikit bijaksana menghasilkan 11010 yang, dalam tabel ITA2, adalah huruf 'G'. Faktanya, setiap *bit* dari kunci memberi tahu kita apakah *bit* yang sesuai dari *plaintext* harus dibalik. Dengan membalik *bit* kunci ini lagi, seperti yang ditunjukkan di atas, karakter asli terungkap.

2.3 Metode XOR

Operasi XOR merupakan operasi logika *bitwise* yang bekerja dengan membandingkan dua buah *bit* yang apabila pada salah satu *bit*nya bernilai benar, maka hasil akhir operasi XOR tersebut adalah benar. Namun, bila kedua *bit* yang akan dibandingkan bernilai salah atau keduanya benar maka hasil akhir operasi XOR tersebut adalah salah.

XOR enkripsi, meskipun bukan kunci publik seperti RSA, hampir bisa dipecahkan melalui metode *brute force*. Hal ini rentan terhadap pola, tetapi kelemahan ini dapat dihindari melalui, pertama mengompresi *file* (sehingga untuk menghilangkan pola). Enkripsi eksklusif atau membutuhkan baik *encryptor* dan *decryptor* memiliki akses ke kunci enkripsi, tetapi algoritma enkripsi, sementara sangat sederhana, hampir bisa dipecahkan. Karya XOR enkripsi dengan menggunakan fungsi aljabar *Boolean* XOR.

Code:

```
X    Y    X^Y
--------------
1    1    0
1    0    1
0    1    1
0    0    0
```

Namun bagaimana jika kita melakukan dua kali operasi XOR dua kali terhadap suatu *bit* dengan *operand* yang sama, maka hasilnya akan kembali seperti semula.

Seperti contoh gambar berikut.

Code:

```
X    Y    X^Y    (X^Y)^Y
-----------------------
1    1    0      1
1    0    1      1
0    1    1      0
0    0    0      0
```

Dapat dilihat dari kedua gambar di atas, pada gambar pertama terlihat nilai pada variabel X yang di XOR kan dengan variabel Y dan menghasilkan nilai yang ada pada variabel X^Y. Namun, jika kita lihat pada gambar kedua, variabel X^Y di XOR kan lagi dengan variabel Y dan kemudian menghasilkan nilai yang sama dengan nilai yang ada pada variabel X. Sifat seperti ini yang dapat kita gunakan untuk membuat enkipsi sederhana.

2.4 Pengertian Informasi

Informasi dapat dianggap sebagai resolusi ketidakpastian; itu adalah yang menjawab pertanyaan "apa itu entitas" dan dengan demikian mendefinisikan esensi dan sifat karakteristiknya. Ini terkait dengan data, karena data mewakili nilai yang dikaitkan dengan parameter, dan informasi adalah data dalam konteks dan dengan makna yang dilampirkan. Informasi juga berkaitan dengan pengetahuan, karena pengetahuan menandakan pemahaman konsep abstrak atau konkret. Dalam hal komunikasi, informasi dinyatakan baik sebagai isi pesan atau melalui pengamatan langsung atau tidak langsung. Apa yang dirasakan dapat ditafsirkan sebagai pesan dalam dirinya sendiri, dan dalam pengertian itu, informasi selalu disampaikan sebagai isi pesan.

Informasi dapat dikodekan ke dalam berbagai bentuk untuk transmisi dan interpretasi (misalnya, informasi dapat dikodekan ke dalam urutan tanda, atau ditransmisikan melalui sinyal). Itu juga dapat dienkripsi untuk penyimpanan dan komunikasi yang aman. Ketidakpastian suatu peristiwa diukur dengan probabilitas kejadiannya dan berbanding terbalik dengan itu. Semakin tidak pasti suatu peristiwa, semakin banyak informasi yang dibutuhkan untuk menyelesaikan ketidakpastian peristiwa itu. *Bit* adalah unit informasi yang khas, tetapi unit lain seperti nat dapat digunakan. Sebagai contoh, informasi yang dikodekan dalam satu *flip* koin "adil" adalah log2 (2/1) = 1 bit, dan dalam dua *flip* koin adil adalah log2 (4/1) = 2 *bit*.

Konsep informasi memiliki makna yang berbeda dalam konteks yang berbeda. Dengan demikian konsep menjadi terkait dengan pengertian kendala, komunikasi, kontrol, data, bentuk, pendidikan, pengetahuan, makna, pemahaman, rangsangan mental, pola, persepsi, representasi, dan entropi.

2.5 Visual Basic.Net

Visual Basic.Net merupakan salah satu *tool development Microsoft* yang dapat digunakan untuk membuat aplikasi di lingkungan kerja berbasis sistem operasi *Windows*. Visual Basic.Net menyediakan *tools* bagi para *developer* untuk membangun aplikasi yang berjalan di .*Net Framework* (Safik, 2016 : 2)

Visual basic merupakan turunan bahasa pemrograman BASIC yang menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat. Dengan menggunakan bahasa pemrograman VB, para *programmer* dapat membangun aplikasi dengan menggunakan komponen-komponen yang di sediakan VB.

Microsoft Visual Basic (sering disingkat sebagai VB saja) merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment* (*IDE*) *visual* untuk membuat program perangkat lunak berbasis sistem operasi *Microsoft Windows* dengan menggunakan model pemrograman (*COM*), *Visual Basic* merupakan turunan bahasa pemrograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat, beberapa bahasa skrip seperti *Visual Basic for Applications* (*VBA*) dan

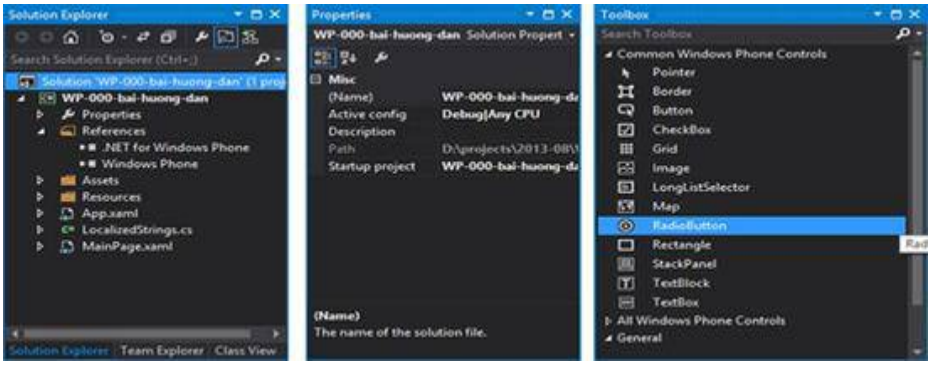
Visual Basic Scripting Edition (*VBScript*), mirip seperti halnya *Visual Basic*, tetapi cara kerjanya yang berbeda.

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 luar tambahan.

Dalam pemrograman untuk bisnis, *Visual Basic* memiliki pangsa pasar yang sangat luas. Dalam sebuah survey yang dilakukan pada tahun 2005, 62% pengembang perangkat lunak dilaporkan menggunakan berbagai bentuk *Visual Basic* yang diikuti oleh C++, JavaScript, dan Java.

Beberapa komponen kerja program *visual basic* 2010 telah ditampilkan sebagai tampilan standard. Masih banyak lagi komponen yang masih tersembunyi sehingga memerlukan perintah tertentu untuk menampilkannya. Kita dapat mengatur komponen di dalam program *visual basic* 2010 sesuai dengan yang kita butuhkan. Berikut ini adalah beberapa komponen kerja dari *visual basic* 2010 adalah.

Toolbox adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membuat suatu desain mulai dari tombol *label*, *pointer*, *button*, dan lain-lain. Berikut ini adalah gambaran *toolbox* pada *visual basic* 2010.



Gambar 2.4. Tampilan *Toolbox*

Sumber : (Safik : 2016 : 2)

Berikut ini adalah *table* yang berisi nama tombol yang terdapat didalam *toolbox* beserta fungsinya.

Tabel 2.1. Toolbox

Nama tombol	Fungsi
Pointer	Memilih, mengatur ukuran dan memindahkan posisi yang terpasang di bagian *form*.
Bindingsources	Untuk mengkoneksikan program ke *database.*
Label	Menampilkan teks, dimana pengguna program tidak bisa mengubah teks tersebut.
Groupbox	Untuk mengelompokkan *item* yang ada di *form.*
Checkbox	Membuat kotak periksa, dimana pengguna program dapat memilih sekaligus.
Listbox	Membuat daftar pilihan.
Timer	Membuat kontrol waktu dan interval yang diperlukan.
Image	Menampilkan gambar pada *form* dalam format *bitmap, icone*, atau *metafile*.

Picturebox	Menampilkan gambar dari sebuah *file*.
Textbox	Membuat teks, dimana teks tersebut dapat diubah oleh pembuat program.
Button	Membuat tombol perintah.
Combobox	Menambahkan kontrol kotak *combo* yang merupakan kontrol gabungan antara *textbox* dan *listbox*.

Sumber : (Safik : 2016 : 2)

2.6 *Unified Modeling Language* (UML)

2.6.1 Pengenalan UML

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang memungkinkan pengembang menentukan, memvisualisasikan, membuat, dan mendokumentasikan artefak sistem perangkat lunak. Dengan demikian, UML membuat artefak ini *scalable*, aman dan kuat dalam eksekusi. UML adalah aspek penting yang terlibat dalam pengembangan perangkat lunak berorientasi objek.

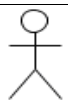
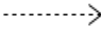
Unified Modeling Language (UML) adalah bahasa pemodelan tujuan-umum, pengembangan, di bidang rekayasa perangkat lunak yang dimaksudkan untuk memberikan cara standar untuk memvisualisasikan desain sistem. Penciptaan UML awalnya dimotivasi oleh keinginan untuk membakukan sistem notasi yang berbeda dan pendekatan untuk desain perangkat lunak. Ini dikembangkan oleh Grady Booch, Ivar Jacobson dan James Rumbaugh di Rational Software pada tahun 1994-1995, dengan pengembangan lebih lanjut yang dipimpin oleh mereka sampai tahun 1996. Pada tahun 1997, UML diadopsi sebagai standar oleh *Object Management Group* (OMG) dan telah dikelola oleh

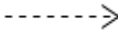
organisasi ini sejak itu. Pada tahun 2005, UML juga diterbitkan oleh Organisasi Internasional untuk Standarisasi (ISO) sebagai standar ISO yang disetujui. Sejak itu standar tersebut telah direvisi secara berkala untuk mencakup revisi terbaru UML.

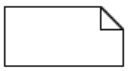
2.6.2 *Use Case Diagram*

Diagram *use case* adalah penggambaran grafis dari interaksi antara elemen-elemen sistem. *Use case* adalah metodologi yang digunakan dalam analisis sistem untuk mengidentifikasi, mengklarifikasi, dan mengatur persyaratan sistem. Dalam konteks ini, istilah "sistem" mengacu pada sesuatu yang sedang dikembangkan atau dioperasikan, seperti situs penjualan produk dan layanan *e-mail*. Diagram *use case* digunakan dalam UML (*Unified Modeling Language*), sebuah notasi standar untuk pemodelan objek dan sistem dunia nyata. (Haviluddin : 2013 : 4).

Tabel 2.2. Simbol *Use Case Diagram*

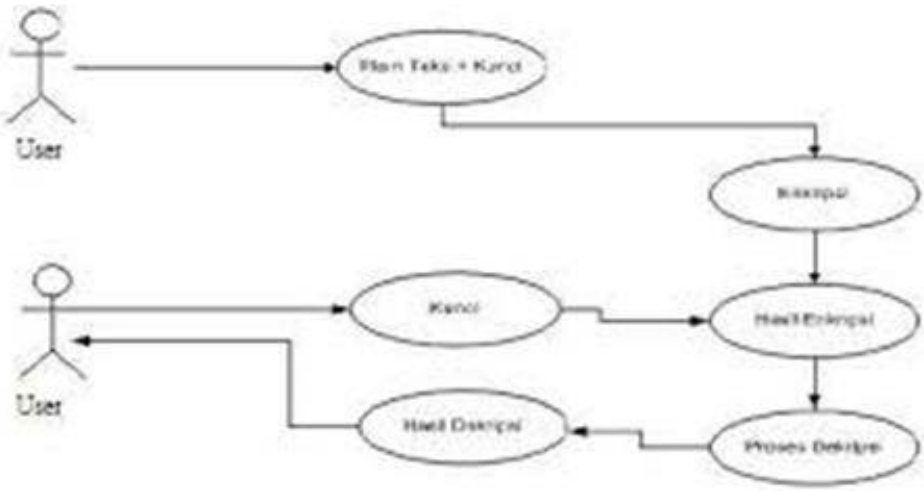
NO.	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi

			elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Sumber : (Gellysa Urva, 94 : 2015)

Contoh *Use Case Diagram* :



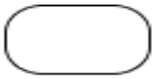
Gambar 2.5. Contoh *Use Case Diagram*

Sumber : (Haviluddin : 2014 : 4)

2.6.3 *Activity Diagram*

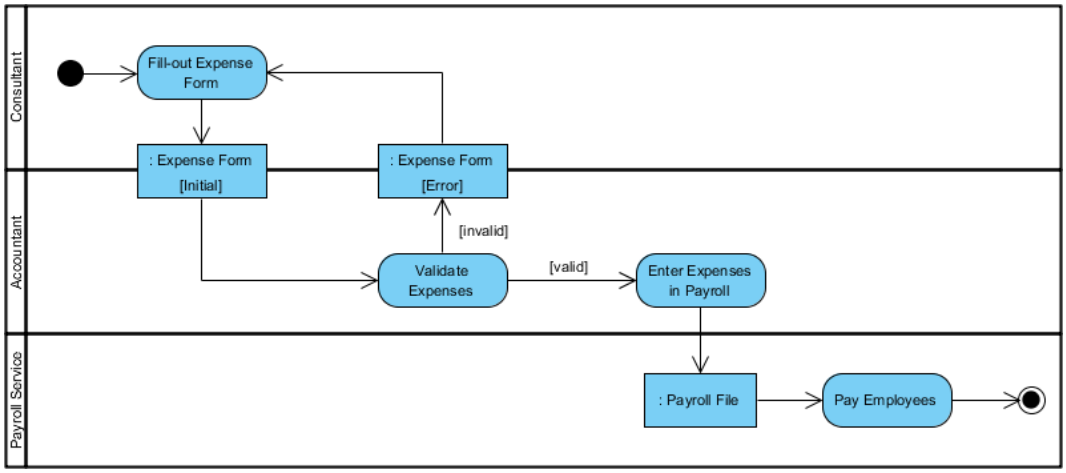
Activity diagram adalah diagram penting lainnya dalam UML untuk menggambarkan aspek dinamis sistem. *Activity* diagram pada dasarnya adalah diagram alur untuk mewakili aliran dari satu aktivitas ke aktivitas lain. Aktivitas tersebut dapat digambarkan sebagai operasi sistem. Aliran kontrol diambil dari satu operasi ke operasi lainnya.

Tabel 2.3. Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		*Action*	*State* dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan.
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

Sumber : (Gellysa Urva, 94 : 2015)

Contoh *Activity Diagram*:



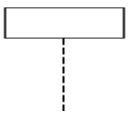
Gambar 2.6. Contoh *Activity Diagram*

Sumber : (Gellysa Urva, 94 : 2015)

2.6.4 *Sequence Diagram*

Diagram urutan, dalam konteks UML, mewakili kolaborasi objek dan digunakan untuk menentukan urutan kejadian antara objek untuk hasil tertentu. Diagram urutan adalah komponen penting yang digunakan dalam proses yang berkaitan dengan analisis, desain, dan dokumentasi. Diagram urutan juga dikenal sebagai diagram waktu, diagram acara dan skenario acara.

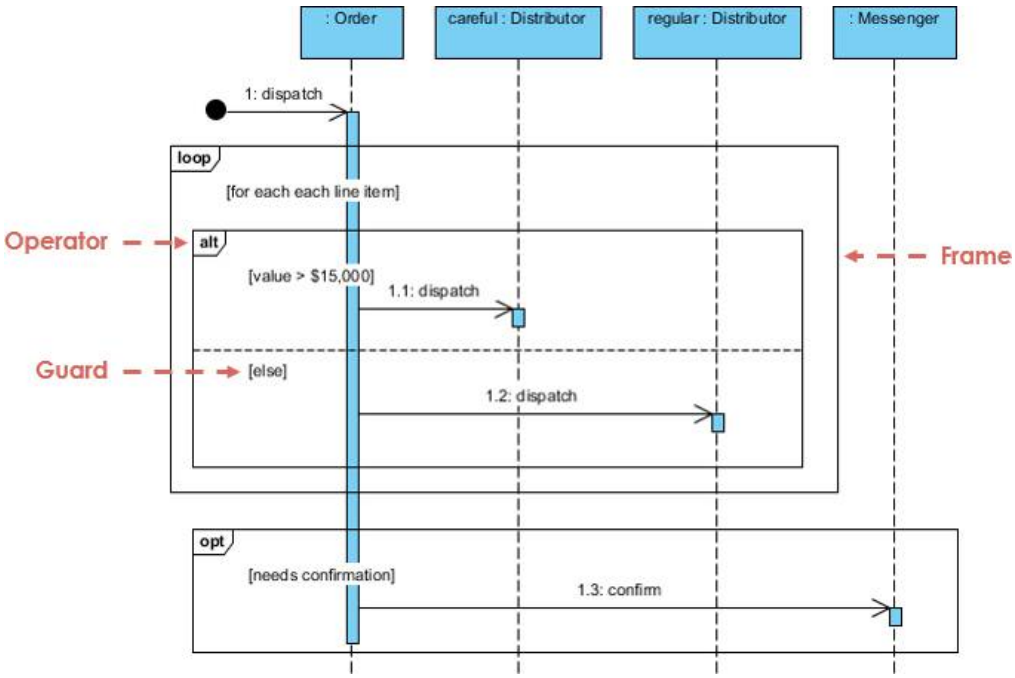
Tabel 2.4. Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.

2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

Sumber : (Gellysa Urva, 95 : 2015)

Contoh *Sequence Diagram*:



Gambar 2.7. Contoh *Sequence Diagram*

Sumber : (Gellysa Urva, 95 : 2015)

2.6.5 *Class Diagram*

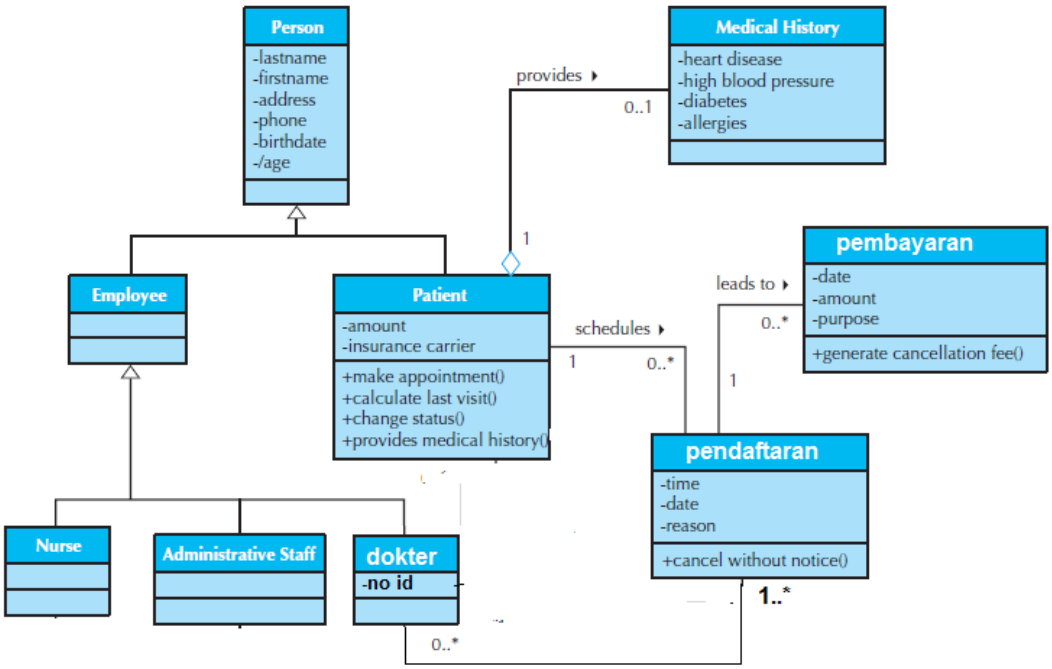
Diagram kelas adalah ilustrasi hubungan dan dependensi kode sumber antara kelas-kelas dalam *Unified Modeling Language* (UML). Dalam konteks ini, kelas mendefinisikan metode dan variabel dalam suatu objek, yang merupakan entitas spesifik dalam program atau unit kode yang mewakili entitas itu.

Tabel 2.5. Simbol *Class Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya.
3		*Extend*	Menspesifikasikan bahwa use case target memperluas prilaku dari use case sumber pada suatu titik yang diberikan.

Sumber : (Gellysa Urva, 95 : 2015)

Contoh *Class Diagram*:



Gambar 2.8. Contoh *Class Diagram*

Sumber : (Gellysa Urva, 95 : 2015)

BAB III

METODE PENELITIAN

3.1 Analisa Sistem Yang Berjalan

Dalam materi perkuliahan keamanan komputer terdapat bab mengenai enkripsi. Salah satu bentuk enkripsi adalah menggunakan metode *ciphertext*. Untuk mendapatkan hasil teks yang diubah (*ciphertext*), menggunakan angka dan tabel untuk konversi. Penggunaan angka jauh lebih sulit dibandingkan dengan menggunakan tabel. Algoritma *Vernam Cipher* nantinya akan menganalisa langkah-langkah kerja algoritma kriptografi *Vernam Cipher* tersebut, sehingga nantinya algoritma kriptografi yang penulis bangun akan memiliki tingkat kesulitan yang lebih tinggi untuk dipecahkan dibandingkan algoritma kriptografi *Vernam Cipher*.

3.2 Proses Enkripsi Vernam Cipher

Adapun algoritma enkripsi *Vernam Cipher* dalam bentuk *pseudocode* dibawah ini :

Langkah 1 : *Input plaintext*

Langkah 2 : *Input* kunci

Langkah 3 : Ubah setiap karakter pada *plaintext* kedalam bentuk ASCII *Code*

I = 0

Jum = LEN(*Plaintext*)

For i = 1 *to* jum

P(i) = Asc(substr(*Plaintext*, 1,i))

Next i

Langkah 4 : Ubah setiap karakter pada kunci ke dalam bentuk ASCII *Code*

I = 0

Jum = LEN(Kunci)

For i = 1 *to* jum

K(i) = Asc(substr(Kunci, 1,i))

Next i

Langkah 5 : Lakukan enkripsi dengan rumus

I = 0

Jum = LEN(*Plaintext*)

For i = 1 *to* jum

C(i) = P(i) XOR (K(i)

Next i

Langkah 6 : Ubah Ascii *ciphertext* ke dalam bentuk karakter

I = 0

Jum = LEN(*Ciphertext*)

For i = 1 *to* jum

Karakter_C(i) = chr(substr(C(i), 1,i))

Next i

3.3 Proses Dekripsi Vernam Cipher

Adapun algoritma dekripsi *Vernam Cipher* dalam bentuk *pseudocode* dibawah ini:

Langkah 1 : *Input ciphertext*

Langkah 2 : *Input* kunci

Langkah 3 : Ubah setiap katakter pada *ciphertext* kedalam bentuk Ascii *Code*

```
I = 0
Jum = LEN(Ciphertext)
For i = 1 to jum
        C(i) = Asc(substr(Ciphertext, 1,i))
Next i
```

Langkah 4 : Ubah setiap karakter pada kunci kedalam bentuk ASCII *Code*

```
I = 0
Jum = LEN(Kunci)
For i = 1 to jum
        K(i) = Asc(substr(Kunci, 1,i))
Next i
```

Langkah 5 : Lakukan enkripsi dengan rumus

```
I = 0
Jum = LEN(Ciphertext)
For i = 1 to jum
        P(i) = C(i) XOR (K(i)
Next i
```

Langkah 6 : Ubah ASCII *plaintext* kedalam bentuk karakter

I = 0

Jum = LEN(*Plaintext*)

For i = 1 *to* jum

Karakter_P(i) = chr(substr(*Plaintext*, 1,i))

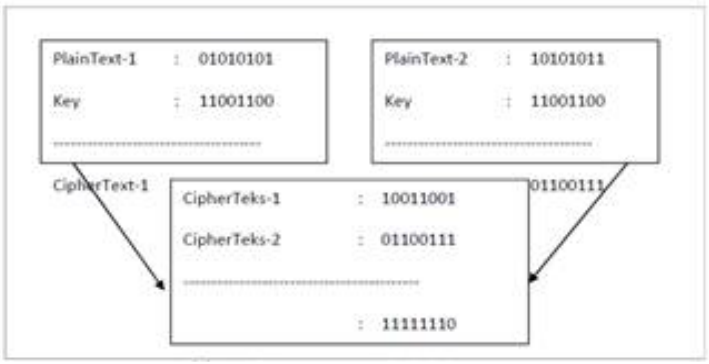
Next i

3.4 Analisis Kelemahan Algoritma Vernam Cipher

Adapun kelemahan terlihat pada kotak yang ada di *pseudocode* diatas, kelemahan algoritma *Vernam Cipher* ini terletak pada pemakaian XOR dalam melakukan enkripsi dan dekripsi antara *plaintext* dan kunci. (Agustanti, 2013)

$$P(i) = C(i) \text{ XOR } K(i)$$

Dimana jika diasumsikan A berhasil menyadap 2 buah *ciphertext* berbeda dengan kunci yang sama, A kemudian meng XOR kan kedua *ciphertext* tersebut, jika berhasil mengetahui *plaintext* dan *ciphertext* tersebut maka akan dengan mudah menemukan *plaintext* yang lain tanpa perlu mengetahui rangkaian kuncinya seperti contoh dibawah ini :



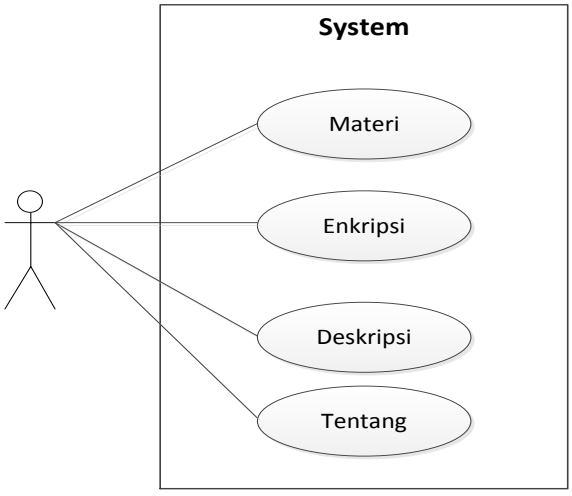
Gambar 3.1. Contoh hasil XOR sehingga mendapatkan *Plaintext*

3.5 Perancangan Berorientasi Objek

Tujuan dari perancangan berorientasi objek ini memungkinkan adanya komunikasi yang lebih berkualitas antara pengguna, pengembang penganalisis, *tester*, manajer dan siapapun yang terlibat dalam proyek pengembangan sistem informasi.

3.5.1 *Use Case Diagram*

Berikut adalah *use case diagram* yang menggambarkan kegiatan.



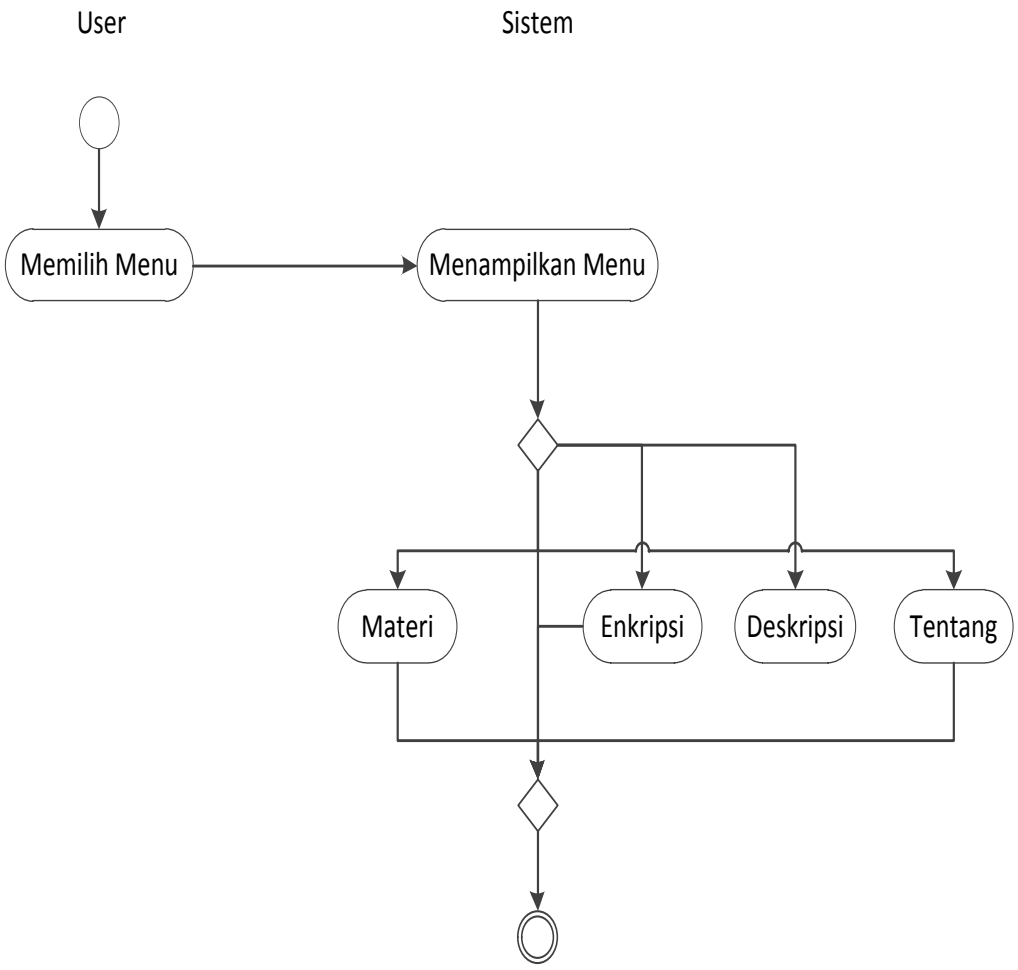
Gambar 3.2. *Use Case Diagram*

Keterangan :

Dalam *use case diagram* di atas, *user*/pengguna sebagai *actor* yang mempunyai *use case* Data teks, Enkripsi, Dekripsi dan Tentang.

3.5.2 Pembuatan *Activity Diagram*

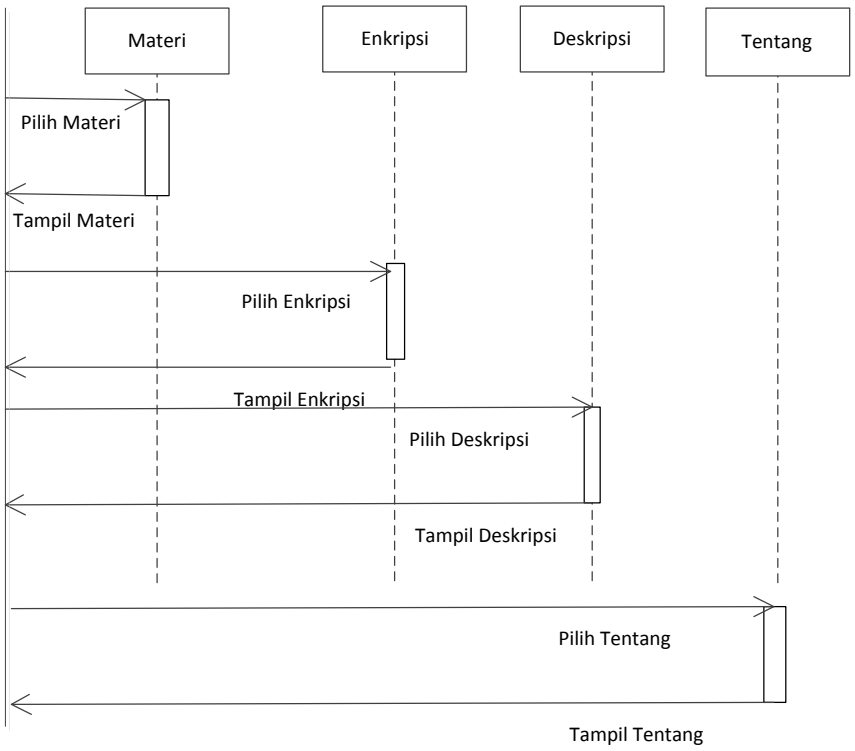
Activity diagram menggambarkan aktivitas-aktivitas yang terjadi dalam aplikasi dari aktivitas dimulai sampai aktivitas berhenti.



Gambar 3.3. *Activity Diagram*

3.5.3 *Sequence Diagram*

Berikut ini adalah gambar dari *sequence diagram* dari algoritma enkripsi dan dekripsi Vernam cipher.



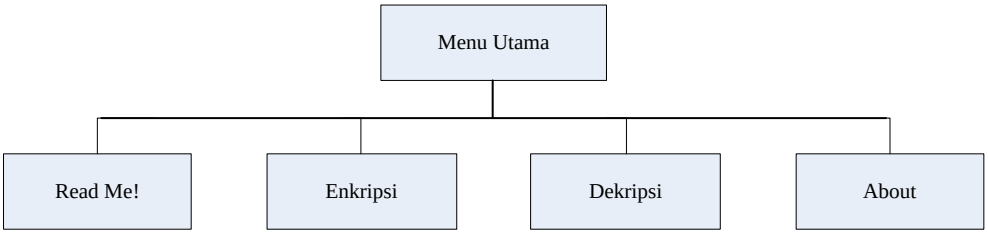
Gambar 3.4. *Sequence Diagram*

Keterangan gambar :

1. Dalam diagram di atas menjelaskan bahwa *user* memilih data teks kemudian sistem menampilkan data teks tersebut.
2. *User* me*request* Enkripsi kemudian sistem menampilkan *menu* Enkripsi.
3. *User* me*request* Dekripsi kemudian sistem menampilkan *menu* Dekripsi.
4. *User* me*request Menu* Tentang kemudian sistem menampilkan *Form* Tentang.

3.6 Struktur Program

Struktur program mempresentasikan organisasi komponen program (modul) serta mengimplementasikan suatu hirarki kontrol. Hirarki kontrol tidak mengimplementasikan aspek prosedural lunak seperti urutan proses, kejadian atau urutan dari keputusan atau perulangan operasi.

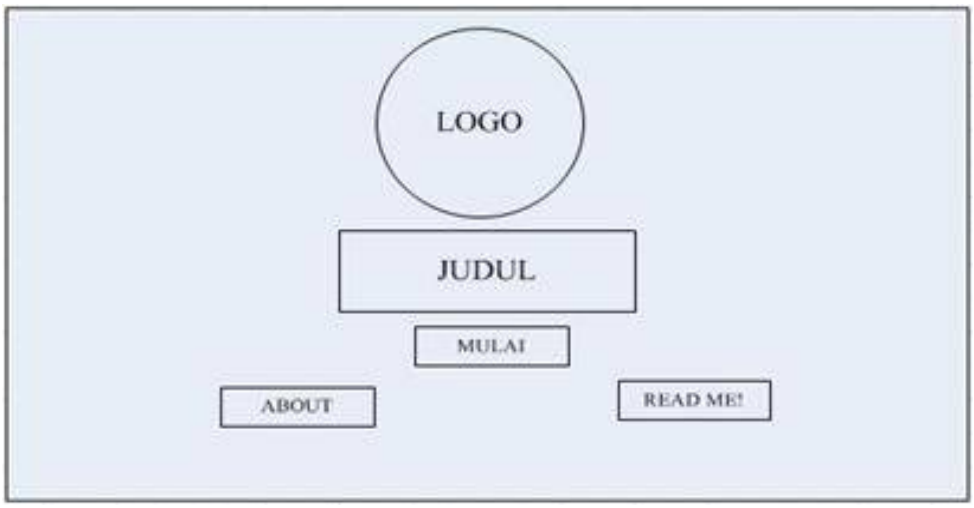


Gambar 3.5. Struktur navigasi enkripsi

3.7 Perancangan Antarmuka

3.7.1 Rancangan Halaman Menu Utama

Form ini berisi tombol-tombol seperti menu Mulai, *About* dan *Read Me*!.



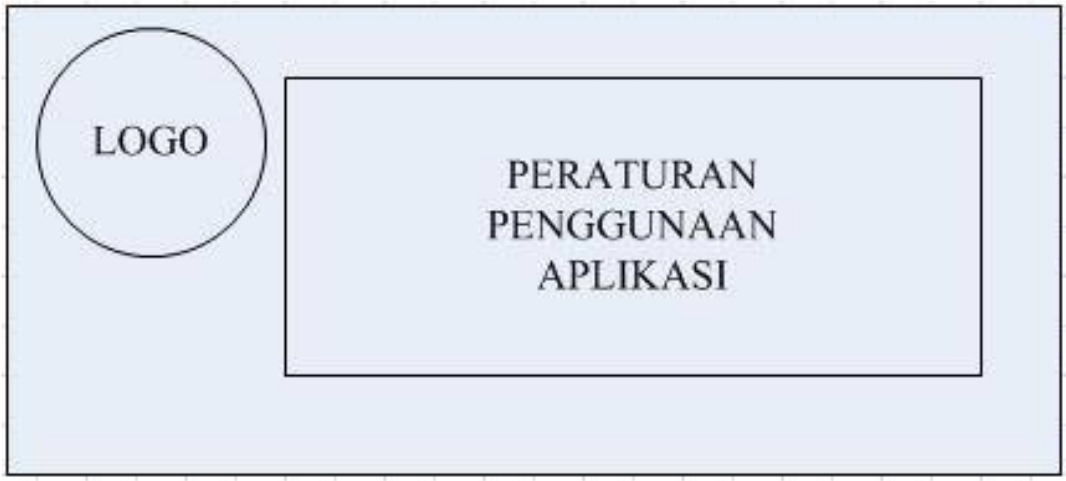
Gambar 3.6. Rancangan halaman menu utama

Pada tampilan di atas terdapat 3 tombol yaitu Mulai, *About* dan *Read Me!*.

- Tombol Mulai berfungsi untuk menampilkan pengguna ke *form* utama enkripsi.
- Tombol *About* berfungsi untuk menampilkan *form About*.
- Tombol *Read Me!* berfungsi untuk menempilkan *form* peraturan penggunaan aplikasi.

3.7.2 Rancangan Halaman Peraturan

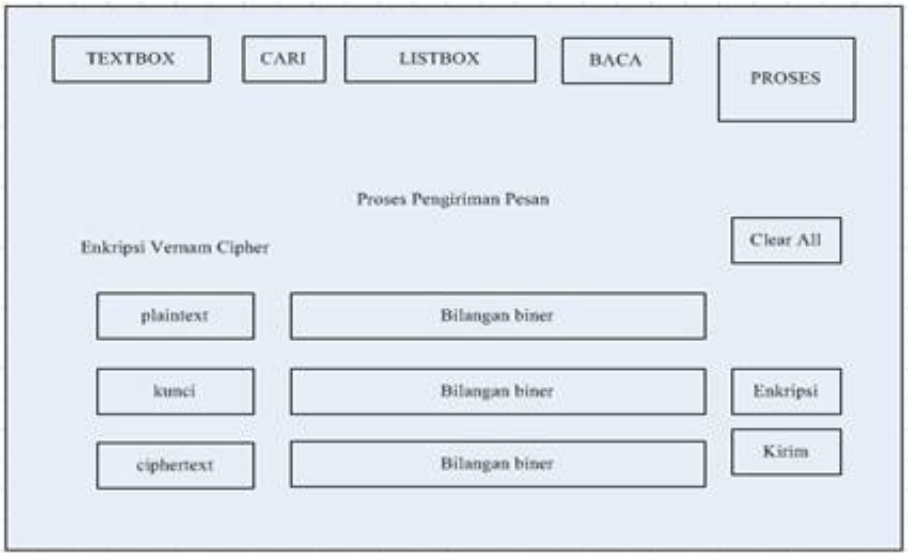
Form ini digunakan untuk menjelaskan cara kerja atau cara penggunaan aplikasi baik itu dalam proses pengiriman maupun penerimaan pesan.



Gambar 3.7. Rancangan halaman peraturan

3.7.3 Rancangan Halaman Enkripsi

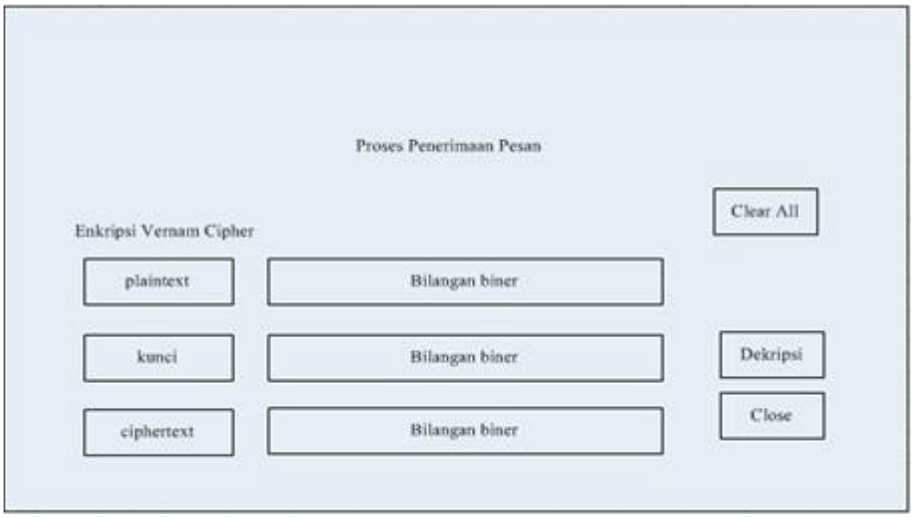
Halaman ini berisi penjelasan mengenai enkripsi. Pengguna memasukkan tulisan alsi atau *plaintext* ke dalam tombol masukkan *plaintext* kemudian dimasukkan juga kunci. Setelah itu, ditekan tombol Proses Enkripsi yang kemudian akan menampilkan *ciphertext* atau tulisan yang telah disandikan.



Gambar 3.8. Rancangan halaman enkripsi

3.7.4 Rancangan Halaman Dekripsi

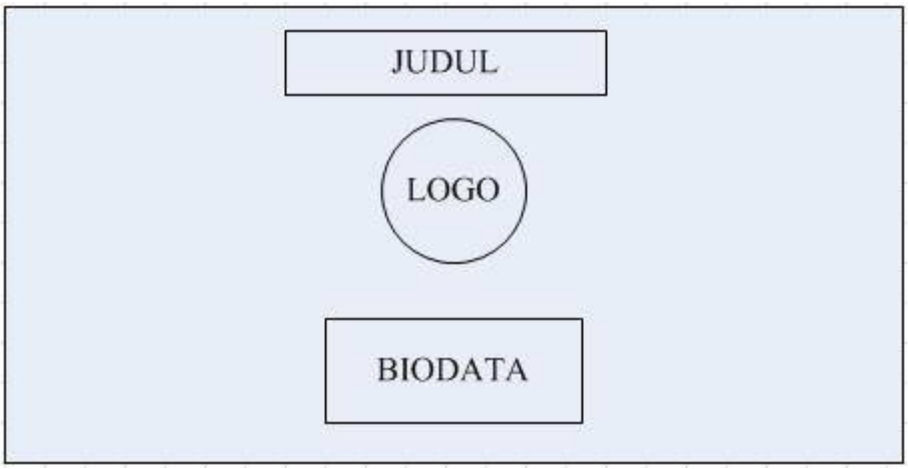
Halaman ini berisi penjelasan mengenai dekripsi. Pengguna memasukkan tulisan yang disandikan atau *ciphertext* ke dalam tombol Proses Dekripsi yang kemudian akan menampilkan *plaintext* atau tulisan asli.



Gambar 3.9. Rancangan halaman dekripsi

3.7.5 Rancangan Halaman About

Halaman ini berisi mengenai judul skripsi yang ditulis dan biodata singkat penulis.



Gambar 3.10. Menu About

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahap dimana aplikasi yang telah dirancang dijalankan. Tahap ini menunjukkan apakah setiap proses dapat berjalan dengan baik dan mampu memberikan hasil yang diharapkan. Proses perancangan aplikasi menggunakan *Visual Basic NET* 2010 di tampilkan dalam bentuk *form-form* yang menjadi sarana bagi pengguna untuk melakukan proses implementasi.

4.2 Pengujian Sistem

Pengujian sistem dilakukan untuk menunjukkan apakah sistem yang telah dirancang dapat berjalan sesuai harapan. Selain itu tujuan pengujian adalah untuk dapat menemukan kesalahan fungsi pada aplikasi yang dibangun dan memperbaikinya.

Pengujian dilakukan dengan memasukkan karakter atau huruf dari *file* berformat .txt, selanjutnya diproses oleh aplikasi apakah aplikasi tersebut dapat memberikan hasil yang sesuai. Proses yang akan dilakukan pengujian dalam aplikasi ini adalah simulasi pengiriman pesan dengan menggunakan metode algoritma *Vernam Cipher* antara pengirim dan penerima dengan kunci tunggal/kunci yang telah ditentukan sehingga pada akhirnya pesan asli yang dikirimkan oleh pengirim dapat dibaca oleh penerima.

4.2.1 Tampilan Awal/*Home*

Tampilan pada gambar 4.1 merupakan tampilan awal ketika aplikasi dijalankan. Pada *form* ini pengguna dapat memilih untuk membuka beberapa *form* lainnya seperti tombol *About* yang akan mengarahkan pengguna menuju *form* yang menjelaskan profil aplikasi ini, tombol *Read Me* yang akan mengarahkan pengguna ke *form* yang menjelaskan tata cara penggunaan dari aplikasi ini.

Gambar 4.1. Tampilan awal/home

Keterangan :

1. Mulai. Proses untuk melanjutkan ke *form* selanjutnya yaitu *form* Menu Utama.
2. *Read Me*!. Berfungsi untuk menampilkan dan menjelaskan proses algoritma Vernam.
3. *About*. Berfungsi untuk menampilkan tentang pembuat aplikasi ini.

4.2.2 Tampilan Aturan Penggunaan Aplikasi

Tampilan aturan penggunaan aplikasi merupakan tampilan halaman atau *form* yang berisi tentang tata cara penggunaan aplikasi yang dijalankan. Pada halaman tersebut di jelaskan apa-apa saja yang menjadi kewajiban bagi pengirim dan penerima saat menjalankan simulasi algoritma *Vernam Cipher*.

Gambar 4.2. Tampilan aturan penggunaan aplikasi

4.2.3 Tampilan Halaman Enkripsi Algoritma Vernam Cipher

Tampilan berikut merupakan tampilan utama pada aplikasi ini. Algoritma Vernam Cipher merupakan algoritma yang menggunakan kunci yang telah ditentukan antara pihak-pihak yang melakukan enkripsi dan dekripsi. Kedua belah pihak menggunakan kunci yang sama yang telah mereka sepakati sebelumnya untuk melakukan enkripsi dan dekripsi.

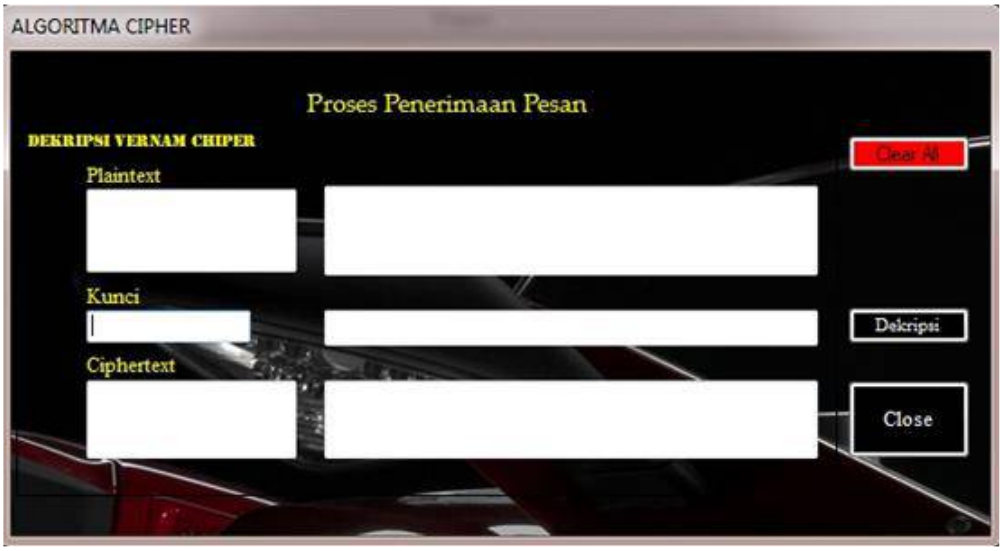
Gambar 4.3. Tampilan halaman enkripsi algoritma Vernam Cipher

Keterangan :

1. Cari. Berfungsi untuk membuka *file* pesan berformat .txt.
2. *Listbox1*. Berfungsi untuk memilih teks atau pesan yang akan diproses menggunakan algoritma Vernam Cipher.
3. Baca File. Berfungsi untuk memindahkan teks atau pesan yang ada pada *listbox* ke *plaintext*.
4. Kunci. Berfungsi untuk memberikan sandi dan mengubah pesan asli menjadi acak.
5. *Ciphertext*: Berisikan pesan yang akan dikirim oleh penerima pesan.
6. Enkripsi. Proses acak dengan menggunakan metode XOR meggunakan algoritma Vernam Cipher.
7. Kirim. Berfungsi untuk mengirimkan pesa *ciphertext* ke penerima pesan.

4.2.4 Tampilan Halaman Dekripsi Algoritma Vernam Cipher

Berikut ini adalah tampilan halaman dari dekripsi algoritma Vernam cipher.



Gambar 4.4Tampilan halaman dekripsi algoritma vernam cipher

Keterangan :

1. *Plaintext*. Pesan yang diterima dari pengirim pesan.
2. Kunci. Berfungsi untuk membuka pesan asli dari *plaintext* yang dikirimkan oleh pengirim pesan.
3. *Ciphertext*. Berfungsi untuk menampilkan pesan asli yang dikirimkan oleh si pengirim pesan.
4. *Clear All*. Berfungsi untuk mengosongkan seluruh *textbox* dan *listbox*.
5. *Close*. Berfungsi untuk menutup *Form* Dekripsi pesan.

6. Enkripsi. Proses acak dengan menggunakan metode XOR yang berguna untuk membuka pesan.

4.2.5 Tampilan Menu *About*

Tampilan menu *About* merupakan tampilan mengenai judul skripsi dan profil singkat penulis.

Gambar 4.5. Tampilan menu *About*

4.3 Hasil Pengujian

Perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, perancangan, dan pengkodean. Pengujian yang digunakan untuk menguji sistem ini adalah metode pengujian *black-box*. Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak.

4.3.1 Rencana Pengujian

Pengujian fungsi enkripsi dan dekripsi dengan menggunakan metode kriptografi *vernam cipher* (XOR) dilakukan dengan menggunakan metode *black-box*. Pengujian dilakukan pada fungsi-fungsi sistem untuk menentukan apakah fungsi tersebut telah berjalan sesuai dengan yang diharapkan.

1) Cari Enkripsi *File*

Tabel 4.1. Rencana pengujian tombol cari

Menu yang diuji	**Detail pengujian**	**Kesimpulan**
Klik Tombol Cari	Mencari *File* *.txt Muncul *form* penambahan pengguna	Diterima

2) Proses Enkripsi

Tabel 4.2. Rencana pengujian pengguna (*user*) proses enkripsi

Menu yang diuji	**Detai pengujian**	**Jenis uji**
Proses	Melakukan proses enkripsi	Diterima
Kirim	Proses pengiriman *file* enkripsi	Diterima
Clear All	Menghapus seluruh *text* yang ada pada *textbox*	Diterima

3) Proses Dekripsi

Tabel 4.3. Rencana pengujian pengguna (*user*) peroses dekripsi

Menu yang diuji	**Detai pengujian**	**Jenis uji**
Dekripsi	Melakukan proses deskripsi atau pengembalian pesan asli	Diterima
Close	Menutup semua program	Diterima
Clear All	Menghapus seluruh *text* yang ada pada *textbox*	Diterima

4.3.2 Pengujian Proses

Pengujian proses yang telah disusun, maka dapat dilakukan pengujian sebagai berikut :

Tabel 4.4. Proses pengujian enkripsi dan dekripsi (*user*)

Data Pengujian Hasil					**Hasil**
No	**Isi Pesan**	**Kunci**	**Enkripsi**	**Dekripsi**	
1	A	1	r	A	Berhasil
	01000001	00110001	01110010	01000001	
2	Saya	54y4	ˆ•ò•	Saya	Berhasil
	0101001101100 0010111100101 100001	001101010011 010001111001 00110100	10110001101000 00001000101111 00101000000010 0010	0101001101100 0010111100101 100001	

3	sedang apa? 0111001101100 1010110010001 1000010110111 0011001110010 0000011000010 1110000011000 0100111111	123key 001100010011 001000110011 011010110110 010101111001	¤—— ÌÓàQ“£Ì¤ 10100100100000 00010100100000 00010100110011 00110100111110 00000101000110 00000001110010 10001111001100 10100100	sedang apa? 0111001101100 1010110010001 1000010110111 0011001110010 0000011000010 1110000011000 0100111111	Berhasil
4	Sekarang lagi musim hujan 0101001101100 1010110101101 1000010111001 0011000010110 1110011001110 0100000011011 0001100001011 0011101101001 0010000001101 1010111010101 1100110110100 1011011010010 0000011010000	karakter123 ++ 011010110110 000101110010 011000010110 101101110100 011001010111 001000110001 001100100011 001100101011 00101011	¾ÆÝÂÝÕÓÙQž””’‹ÎçÔÔá…Ú¦œ”™ 10111110110001 10110111011100 00101101110111 01010111010011 11011001010100 01101111110100 00000011101100 00000011001100 00000011101100 00000111001110 01110111001111 10101001101010 01110000110000	Sekarang lagi musim hujan 0101001101100 1010110101101 1000010111001 0011000010110 1110011001110 0100000011011 0001100001011 0011101101001 0010000001101 1010111010101 1100110110100 1011011010010 0000011010000	Berhasil

	1110101011010 1001100001011 01110		00010011011011 01010100110101 01001110000000 01110110000100 100010	1110101011010 1001100001011 01110	
5	Nama saya adalah YUDHA IRNANDA, mahasiswa Pancabudi	12345fastek	□“ •UÙÂìÕ… Ì•“Ÿ•□†ºÈ¸- ¬Q{…,v´¥´ … Ø’š”§žÙØÔ”µ ÌŸ•”–ªÊÊ	Nama saya adalah YUDHA IRNANDA, mahasiswa Pancabudi	Berhasil
	0100111001100 0010110110101 1000010010000 0011100110110 0001011110010 1100001001000 0001100001011 0010001100001 0110110001100 0010110100000 1000000101100 1010101010100 0100010010000 1000001001000 0001001001010	001100010011 001000110011 001101000011 010101100110 011000010111 001101110100 011001010110 1011	01111111100000 00011100101000 00100000001000 10010101011101 10011100001011 10110011010101 10000000100110 11001100100000 00100010100000 00011100101111 00010000000100 01010011101100 00000100000101 11010110010001 01110001010110	0100111001100 0010110110101 1000010010000 0011100110110 0001011110010 1100001001000 0001100001011 0010001100001 0110110001100 0010110100000 1000000101100 1010101010100 0100010010000 1000001001000 0001001001010	

	1001001001110 0100000101001 1100100010001 0000010010110 0001000000110 1101011000010 1101000011000 0101110011011 0100101110011 0111011101100 0010010000001 0100000110000 1011011100110 0011011000010 1100010011101 0101100100011 01001		11010110001010 00101111011100 00000100110100 00000011010011 10110101101001 01001011011010 01010000010000 00010011011011 00010000000011 00110110000110 00000001110110 10011110111111 01101100111011 00011010100100 00000011101101 10101110011001 01111000100000 00100010100000 00011101100000 00010011101010 10110010101100 1010	1001001001110 0100000101001 1100100010001 0000010010110 0001000000110 1101011000010 1101000011000 0101110011011 0100101110011 0111011101100 0010010000001 0100000110000 1011011100110 0011011000010 1100010011101 0101100100011 01001	

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pembahasan dalam perancangan penerapan algoritma *Vernam Cipher* dalam pengamanan data, maka dapat diambil kesimpulan sebagai berikut :

1. Perangkat lunak ini dirancang untuk menampilkan simulasi pengiriman pesan berekstensi *txt antara pengirim dan penerima pesan dan penggunaan algoritma Vernam Cipher (XOR) sangat baik digunakan untuk proses pengamanan data.
2. Penggunaan kunci sulit ditebak dikarenakan menggunakan *text to binary*, kemungkinan bocornya kunci saat proses pertukaran informasi kunci tunggal dapat dihindari.

5.2 Saran

Adapun saran-saran yang dapat dilakukan penelitian ataupun pengembangan selanjutnya adalah sebagai berikut :

1. Perangkat lunak ini dapat dikembangkan dengan menggunakan kombinasi metode-metode lain.
2. Perangkat lunak ini dapat dikembangkan dan terhubung ke jaringan sehingga dapat dijalankan di lebih dari satu komputer.
3. Perangkat lunak ini dapat dikembangkan menggunakan algoritma-algoritma lain yang lebih kompleks.

DAFTAR PUSTAKA

A. Cheddad, J. Condell, K. Curran and P. Mc Kevitt , "Digital Image Steganography: Survey and Analysis of Current Methods," International Journal of Signal Processing, vol. 90, no. 3, pp. 727-752, 2014.

E. H. Rachmawanto and C. A. Sari, "Keamanan File Menggunakan Teknik Kriptografi Shift Cipher," Jurnal Techno. Com, vol. 14, no. 4, pp. 329-335, 2015.

C. A. Sari, E. H. Rachmawanto, Y. P. Astuti and L. Umaroh, "Optimasi Penyandian File Kriptografi Shift Cipher," in Prosiding Sendi_U 2016, Semarang, 2016.

D. Ariyus, Pengantar Ilmu Kriptografi: Teori, Analisi dan Implementasi, Yogyakarta: Andi, 2013.
R. Sadikin, Kriptografi Untuk Keamanan Jaringan, Yogyakarta: Andi, 2013.

E. H. Rachmawanto, C. A. Sari, Y. P. Astuti and L. Umaroh, "Kriptografi Dengan Algoritma Vernam cipher Untuk Keamanan Data," in Prosiding Sendi_U ke 2 Tahun 2016, Semarang, 2016.

Erika, Winda, Heni Rachmawati, and Ibnu Surya. "Enkripsi Teks Surat Elektronik (E-Mail) Berbasis Algoritma Rivest Shamir Adleman (RSA)." *Jurnal Aksara Komputer Terapan* 1.2 (2012).

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

Harumy, T. H. F., & Sulistianingsih, I. (2016). Sistem penunjang keputusan penentuan jabatan manager menggunakan metode mfep pada cv. Sapo durin. In *Seminar Nasional Teknologi Informasi dan Multimedia* (pp. 6-7).

Herdianto, H. (2018). Perancangan Smart Home dengan Konsep Internet of Things (IoT) Berbasis Smartphone. Jurnal Ilmiah Core IT: Community Research Information Technology, 6(2).

Khairul, K., Haryati, S., & Yusman, Y. (2018). Aplikasi Kamus Bahasa Jawa Indonesia dengan Algoritma Raita Berbasis Android. Jurnal Teknologi Informasi dan Pendidikan, 11(1), 1-6.

Muttaqin, muhammad. "analisa pemanfaatan sistem informasi e-office pada universitas pembangunan panca budi medan dengan menggunakan metode utaut." *jurnal teknik dan informatika* 5.1 (2018): 40-43.

Perwitasari, I. D. (2018). Teknik Marker Based Tracking Augmented Reality untuk Visualisasi Anatomi Organ Tubuh Manusia Berbasis Android. *INTECOMS: Journal of Information Technology and Computer Science*, *1*(1), 8-18.

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.

Ramadhani, S., Suherman, S., Melvasari, M., & Herdianto, H. (2018). Perancangan Teks Berjalan Online Sebagai Media Informasi Nelayan. *Jurnal Ilmiah Core IT: Community Research Information Technology*, *6*(2).

Rizal, Chairul. "Pengaruh Varietas dan Pupuk Petroganik Terhadap Pertumbuhan, Produksi dan Viabilitas Benih Jagung (Zea mays L.)." ETD Unsyiah (2013).

S. Kromodimoeljo, Teori dan Aplikasi Kriptografi, Jakarta: SPK IT Consulting, 2014.