Sistem Keamanan Administrasi Menggunakan Pesan Teks Dengan Metode Transposisi Kolom

Disusun dan Diajukan Untuk Memenuhi Persyaratan Ujian Akhir
Memperoleh Gelar Sarjana Strata-1Pada Jurusan Sistem Komputer
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

NAMA	**: RAHMAT M YUSUP**
N.P.M	**: 1514370162**
PROGRAM STUDI	**: SISTEM KOMPUTER**
KONSENTRASI	**: KEAMANAN JARINGAN KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN

2019

DAFTAR ISI

Halaman

ABSTRAK

RAHMAT M YUSUP

Sistem Keamanan Administrasi Menggunakan Pesan Teks
Dengan Metode Transposisi Kolom

Visual basic pada dasarnya adalah bahasa pemrograman komputer. Bahasa pemrogramannya adalah perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Selain sebagai bahasa pemrograman, Visual Basic juga sering disebut sebagai alat untuk menghasilkan Program berbasis Windows. Visual Basic mampu mengakomodasi berbagai jenis program, dengan Visual Basic kita dapat merancang program berdasarkan sains, telekomunikasi, database, multimedia dan sebagainya. Program enkripsi kata sandi menggunakan metode transposisi kolom menggunakan Visual Basic sebagai bahasa pemrograman. Visual Basic menyediakan fasilitas untuk mengenkripsi kata sandi keduanya dalam bentuk teks atau dalam bentuk file dengan ekstensi * .TXT, RTF, sedangkan untuk jenis file yang lain harus disimpan dulu ke rtf form kemudian untuk enkripsi kata sandi untuk file itu mengandung 100 karakter lebih snagat membutuhkan waktu lama untuk diproses dan kelemahan lain dalam menentukan kata sandi, di mana meskipun kata sandi yang dikumpulkan oleh pengguna berbeda dari kata kata sandi utama, tetapi posisi kolom kata sandi cocok, maka pesan enkripsi masih bisa dilakukan hasil yang sama, sehingga masih perlu pengembangan lebih lanjut.

Kata Kunci: *Kriptografi, Transposition coloum.*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam keamanan mengirim pesan teks, kadang-kadang ada begitu banyak masalah saat mengirim, kadang-kadang pesan yang dikirim tidak lagi dalam bentuk aslinya dengan pihak ketiga mencoba masuk atau mengubah pesan aslinya. Oleh karena itu timbul ilmu yang mempelajari cara menyimpan pesan atau data yang aman dikirim dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga yang biasa disebut dengan kriptografi. Kriptografi adalah ilmu yang hanya digunakan untuk menjaga kerahasiaan data, menggunakan metode tertentu sehingga data hanya dapat dibaca oleh orang yang berhak atas data tersebut.

Kriptografi adalah bidang pengetahuan yang menggunakan persamaan matematika untuk melakukan proses enkripsi dan dekripsi. Teknik ini untuk mengubah data menjadi kode-kode tertentu sehingga informasi tidak dapat dibaca oleh siapa pun kecuali yang berhak. Salah satu metode kriptografi yang umum digunakan adalah transposisi kolom yang menggunakan kunci yang sama ketika mengenkripsi dan mendekripsi, sehingga informasi sulit dipahami.

Kriptografi memiliki 2 (dua) bagian penting, yaitu enkripsi dan dekripsi. Enkripsi adalah proses penyandian pesan asli menjadi pesan yang tidak dapat diartikan sebagai aslinya. Sedangkan deskripsinya adalah mengubah pesan yang disandikan menjadi pesan aslinya. Pesan asli biasanya disebut *plaintext*, sedangkan pesan yang telah disandikan disebut *ciphertext*. Pada kesempatan ini,

kita akan membahas secara khusus tentang membuat program enkripsi transposisi kolom menggunakan bahasa pemrograman visual basic. Ini juga akan menjelaskan bagan alur program, membuat program dan bagaimana program berjalan.

Sehubungan dengan uraian diatas, maka diangkatlah judul skripsi sebagai berikut "***Sistem Keamanan Administrasi Menggunakan Pesan Teks Dengan Metode Transposisi kolom***". Dimana akan dibuat sebuah media aplikasi yang bertujuan sebagai keamanan pada sebuah pesan teks.

1.2 Rumusan Masalah

Berkaitan dengan perancangan program kriptografi menggunakan metode transposisi kolom, rumusan masalah pada penelitian ini adalah sebagai berikut:

1) Adanya beberapa masalah yang menghambat proses pengiriman data yang telah di sepakati.
2) Bagaimana mengimplementasikan enkripsi dan deskripsi dengan menggunakan metode transposisi kolom dengan menggunakan bahasa pemograman Visual basic 2010.

1.3 Batasan Masalah

Karena keterbatasan dan waktu maka penulis akan membatasi pokok permasalahan yang akan dibahas yaitu:

1) Data yang di input berupa teks atau tulisan, bukan suara ataupun gambar ke dalam pesan teks.

2) Teks yang akan di enkripsi berupa angka, huruf atau symbol.
3) Memiliki fitur utama yaitu membuat, menyimpan dan merubahnya ke dalam bentuk enkripsi, kemudian untuk dapat membukanya digunakan deskripsi dengan *key* yang hanya diketahui si pengirim.

1.4 Tujuan Penulisan

Adapun tujuan dari penulisan ini sebagai berikut:

1) Untuk mengetahui enkripsi dan deskripsi pesan teks dengan metode transposisi kolom.
2) Untuk mengetahui fungsi dari kriptografi dalam mengamankan pesan teks dari pihak yang tidak berkepentingan.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

1) Memberikan keamanan data agar tidak mudah dimanfaatkan oleh pihak – pihak yang tidak berkompeten.
2) Melindungi data agar tidak disalah gunakan oleh pihak yang tidak bertanggung jawab.

BAB II

LANDASAN TEORI

2.1 Kriptografi

2.1.1 Pengertian Kriptografi

Kriptografi terdiri dari dua kata yang berasal dari bahasa *Yunani*, yaitu: "*kryptos*" dan "*graphia*". Arti kata "*kryptos*" adalah sesuatu yang tersembunyi, tidak diketahui, tersembunyi, rahasia atau misterius. Sedangkan "*graphia*" berarti menulis. Jadi, kriptografi dapat dijelaskan secara harfiah sebagai penulisan rahasia atau kadang-kadang disebut sebagai seni penulisan rahasia dan sains. Menurut sebuah buku berjudul "Kriptografi Terapan" yang ditulis oleh *Bruce Schneider (John Wiley & Sons, 1996)*, kriptografi adalah seni atau ilmu untuk menjaga kerahasiaan suatu artikel agar tetap aman, tanpa sepengetahuan pihak yang tidak berwenang. Ahli kriptografi dikenal sebagai kriptografi. Selain kriptografi, ada kriptanalisis yang merupakan kebalikan dari proses kriptografi dalam kriptologi. Kriptologi adalah salah satu cabang algoritma dalam matematika. *Cryptologist* dikenal sebagai *cryptologist*. Dalam kriptanalisis, penganalisa dan pemecah kode menjadi teks biasa tanpa melalui proses deskripsi yang masuk akal yang disebut kripanalis. Algoritma kriptografi dan semua kemungkinan *ciphertext, plaintext*, dan kunci (kunci lain) disebut *cryptosystems*. *Plaintext* adalah pesan / data asli yang dapat dibaca. *Ciphertext* adalah pesan / data acak, yang sulit ditafsirkan. Kunci adalah nilai yang digunakan untuk mengonversi *plaintext* ke *ciphertext*.

Dalam ilmu kriptografi terdapat aspek-aspek keamanan, meliputi : *authority* (pemalsuan), data *integrity* (keutuhan data), *authentication* (autentikasi), *non- repudiation* (tidak ada penyangkalan). Sebenarnya kriptografi sudah digunakan sejak zaman Romawi oleh *Julius Caesar* dalam keperluan militernya. Pada saat perang dunia ke-II, Jerman dan Jepang juga menggunakan algoritma kriptografi dalam berkomunikasi untuk kebutuhan militernya, namun kunci dari *Enigma* (produk kriptografi Jerman) dan *Purple* (produk kriptografi Jepang) dapat dipecahkan oleh sekutu, sehingga dengan mudah sekutu dapat mengetahui langkah-langkah pertahanan dan perlawanan mereka, dan segera menyusun cara mengantisipasi. Kriptografi dahulu hanya menjadi bidang khusus yang diperlajari didalam kemiliteran.

2.1.2 *Cryptanalysis* (Kripanalisis)

Ilmu ini digunakan untuk mendapatkan *plaintext* tanpa harus mengetahui kunci secara wajar (proses deskripsi). Pendapat mengenai kripanalisis ini pertama kali dinyatakan sekitar abad ke-19 oleh Dutchman A *Kerckhoffs* bahwa kerahasiaannya berada pada kunci, dan analisis sandi memiliki rincian lengkap mengenai algoritma kriptografi dan implementasinya. Menurut *Lars Knudsen*, ada beberapa jenis penggolongan pemecahan algoritmanya, yaitu :

1). *Total break* (pemecahan total) yang berhasil menemukan *key* (kunci) yang digunakan untuk melindungi data dalam rumus : $\mathbf{D_k(C) = P}$

D=deskripsi C=*ciphertext*

K=*key* P=*plaintext*

2). *Global deduction* (deduksi global) dengan mendapatkan algoritma alternatif yang ekuivalen dengan rumus diatas, tanpa harus mengetahui kunci

3). *Instance/ Local deduction* (deduksi local) dengan mendapatkan *plaintext* atau *ciphertext* yang disadap

4). *Information deduction* (deduksi informasi) memperoleh informasi tentang kunci atau *plaintext t*nya.

Cara-cara untuk mengukur kompleksitas serangan :

1). *Data complexity* (kompleksitas data), jika jumlah data yang digunakan untuk serangan hanya sedikit, kualitas algoritma yang digunakan kurang baik

2). *Processing complexity* (kompleksitas proses), semakin cepat waktu yang tersedia untuk melakukan serangan yang sering disebut faktor kerja, semakin buruk kualitas algoritma yang digunakan

3). *Storage requirements* (kebutuhan penyimpanan) mengukur jumlah memori yang dibutuhkan untuk melakukan serangan.

2.2 Fungsi Kriptografi

Fungsi kriptografi dalam teknologi informasi, terus menerus dikembangkan cara untuk menangkal berbagai bentuk serangan seperti penyadapan dan pengubahan data yang dikirimkan. Salah satu cara yang ditempuh mengatasi masalah ini adalah dengan menggunakan kriptografi yang menggunakan transformasi data sehingga data yang dihasilkan tidak dapat

dimengerti oleh pihak yang tidak berhak mengakses. Transformasi ini memberikansolusi pada dua macam masalah keamanan data, yaitu masalah privasi (*privacy*) dan keautentikan *(authenticatioan*). Privasi mengandung arti bahwa data yang dikirimkan hanya dapat dimengerti informasinya oleh penerima yang sah atau berhak. Sedangkan keotentikan mencegah pihak ketiga untuk mengirimkan data yang salah atau mengubah data yang dikirimkan. (Doni, 2006)

2.2.1 Enkripsi

Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode dari yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti (tidak terbaca). Enkripsi dapat diartikan sebagai kode atau *chiper*. Isu- isu yang terkait dengan keamanan dan kerahasiaan data adalah *privacy (*kerahasiaan), *integrity* (keutuhan), *authenticity* (keaslian), *non-repudiation* (pembuktian yang tak tersangkal). Di pertengahan tahun 1970-an, enkripsi kuat dimanfaatkan untuk pengamanan oleh sekretariat agen pemerintah Amerika Serikat pada domain publik, dan saat ini enkripsi telah digunakan pada sistem secara luas, seperti Internet *e-commerce*, jaringan Telepon bergerak dan ATM pada bank. Enkripsi dapat digunakan untuk tujuan keamanan. Ilmu yang mempelajari teknik enkripsi disebut kriptografi. Gambaran sederhana tentang enkripsi, misalnya mengganti huruf a dengan n, b dengan m dan seterusnya. Pembahasan enkripsi akan terfokus pada enkripsi password dan enkripsi komunikasi data. Terdapat tiga kategori enkripsi yaitu :

1) Kunci enkripsi rahasia, dalam hal ini terdapat sebuah kunci yang digunakan untuk mengikrisi dan juga sekaligus mendeskripsikan informasi.
2) Kunci enkripsi *public*, dalam hal ini terdapat dua kunci yang digunakan, satu untuk proses enkripsi, satu lagi untuk proses deskripsi.
3) Fungsi *one-way*, dimana informasi dienkripsi untuk menciptakan *"signature"* dari informasi asli yang bisa digunakan untuk keperluan *autentifikasi.*

Dalam enkripsi akan dilakukan operasi-operasi berikut untuk setiap ronde:

1). Transformasi *SubBytes*() merupakan operasi substitusi non-linier pada tiap-tiap *byte* dalam *state* dengan menggunakan tabel substitusi yang dinamakan *S-box* (kotak S).
2). Transformasi *ShiftRows*() menggeser dengan cara memutar *byte-byte* pada baris
3). Transformasi *MixColumn*() adalah perkalian terhadap matriks konstan yang dioperasikan pada kolom-kolom dalam state.
4). Transformasi *AddRoundKey*() dengan cara menambahkan ronde ke *state* dalam operasi XOR. Pada perputaran terakhir, transformasi *MixColumn*() tidak digunakan.

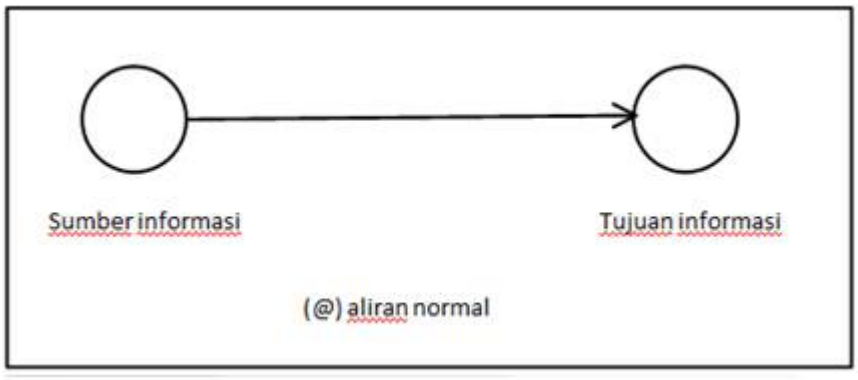
Apabila kunci yang diperlukan melebihi kapasitas jumlah yang tersedia, dalam hal ini hanya 128 bit sampai dengan 256 bit, dapat dilakukan ekspansi kunci untuk memenuhi kebutuhan *subkey* hingga ribuan bit melalui sebuah proses yang dinamakan *key schedule*.

2.2.2 Deskripsi

Deskripsi adalah satu kaedah upaya pengolahan data menjadi sesuatu yang dapat diutarakan secara jelas dan tepat dengan tujuan agar dapat dimengerti oleh orang yang tidak langsung mengalaminya sendiri. Dalam keilmuan, deskripsi diperlukan agar peneliti tidak melupakan pengalamannya dan agar pengalaman tersebut dapat dibandingkan dengan pengalaman peneliti lain, sehingga mudah untuk dilakukan pemeriksaan dan kontrol terhadap deskripsi tersebut. Pada umumnya deskripsi menegaskan sesuatu, seperti apa sesuatu itu kelihatannya, bagaimana bunyinya, bagaimana rasanya, dan sebagainya. Deskripsi yang detail diciptakan dan dipakai dalam disiplin ilmu sebagai istilah teknik. Tulisan deskripsi adalah tulisan yang bertujuan untuk menjelaskan sebuag objek secara terperinci tanpa adanya pengaruh pendapat pendapat pengarang di dalam deskripsi tersebut. (Arif prayitno, 2017)

2.3 Serangan Keamanan Kriptografi

Menurut Laksana (2007) ada beberapa jenis serangan terhadap keamanan yang ada dalam suatu sistem komputer,dan dapat dikelompokkan berdasarkan fungsi dari sistem komputer. Jika dilihat secara garis besar, ada informasi yang berasal dari suatu sumber yang kemudian akan menuju ke suatu titik tujuan.



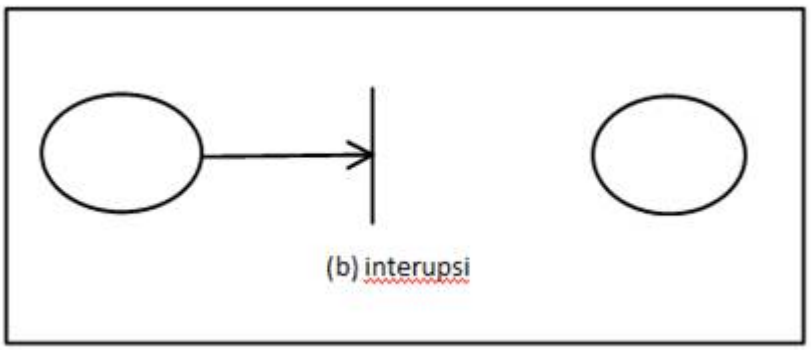
Sumber : Munir, 2006

Gambar 2.1 Aliran Normal

Gambar diatas menunjukkan aliran normal dari pengiriman pesan, jika tidak terjadi serangan diantaranya pengirim dan penerima.

Secara umum, serangan keamanan dibagi menjadi empat kategori, yakni :

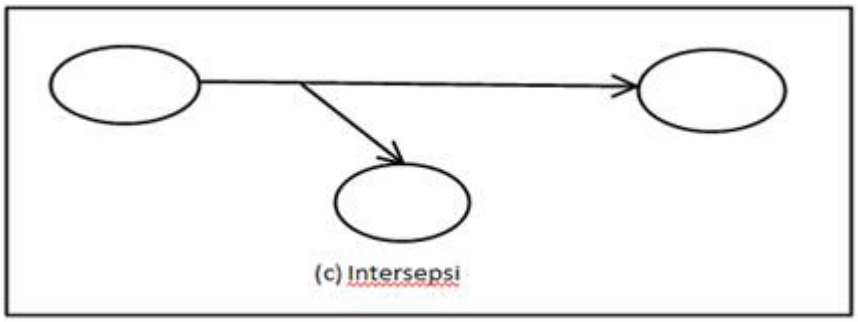
1) Interupsi merupakan aset yang dapat menghancurkan atau merusak, sehingga sudah tidak dapat digunakan kembali. Interupsi ini dapat menyerang bagian *availability* (kemampuan). Contohnya kerusakan pada hard disk atau terjadinya kelumpuhan sistem manajemen file.



Sumber : Munir, 2006

Gambar 2.2 Interupsi

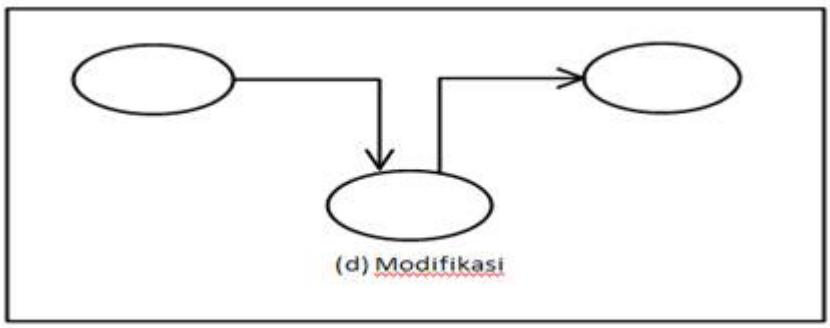
2) Intersepsi, yaitu pihak yang sebenarnya tidak berhak memiliki hak akses. Serangan ini dapat dilakukan oleh komputer maupun manusia dan dapat mengacaukan kerahasiaan suatu aset., misalnya *hacker*.I



Sumber: Munir, 2006

Gambar 2.3 Intersepsi

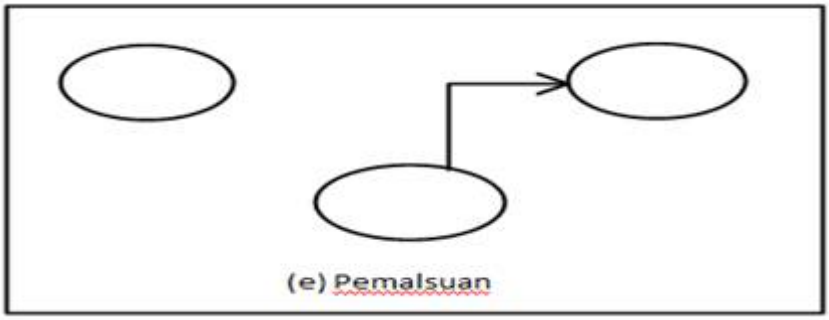
3) Modifikasi adalah pihak ketiga yang tidak mempunyai hak untuk mengakses suatu data, tetapi juga merusak data tersebut. Ini merupakan serangan pada keutuhan data, seperti mengubah isi file atau data, sehingga akan menyebabkan tampilan yang berbeda dan memodifikasi isi atau pesan yang dikirimkan melalui jaringan.



Sumber : Munir, 2006

Gambar 2.4 Modifikasi

4) Pemalsuan dengan penyisipan objek tiruan ke dalam sistem yang telah dibangun. Serangan ini berpengaruh pada autentikasi, seperti pemalsuan pesan dalam sebuah jaringan dengan cara menambah *record-record* ke dalam file.



Sumber : Munir, 2006

Gambar 2.5 Pemalsuan

Beberapa macam serangan yang dapat dilakukan oleh kripanalis/pemecah kode dengan asumsi algoritma enkripsinya telah dikenal secara luas :

1) *Ciphertext only attack*, yang didapatkan dengan serangan jenis ini hanyalah beberapa pesan *chipertext* yang semuanya dienkrip dengan algoritma yang sama. Pemecah kode dapat mencari kuncinya atau mendapatkan *plaintext*. Dapat dirumuskan dengan :

   Diketahui : $C_1 = E_k(P_1)$, $C_2 = E_k(P_2)$, $C_3 = E_k(P_3)$, ...

   Dicari : K dan P_1, P_2, P_3, ...

2) *Known-plaintext attack* berguna untuk mendapatkan beberapa *plaintext* dan *ciphertext* nya. Jika pemecah kode dapat menemukan kunci, maka *ciphertext* yang telah dienkrip dengan algoritma yang sama,

dapat diketahui *plaintext* nya.

Diketahui: P_1, P_2, P_3, ... dan C_1, C_2, C_3, ...

Dicari : K atau P lainnya

3) *Chosen-plaintext attack,* menggunakan serangan ini, kripanalis dapat mengetahui beberapa *plaintext* dan *ciphertext*, dan bebas memilih blok *plaintext* yang akan dienkrip dengan algoritma dan kunci yang sama.

   Diketahui: P_1, P_2, P_3, ... dan C_1, C_2, C_3, ... dan kripanalis dapat memilih P_1, P_2, P_3

   Dicari : K atau P lainnya

4) *Adaptive-chosen-plaintext attack*, serangan ini hampir sama dengan jenis *chosen- plaintext attack,* tetapi dengan menggunakan *adaptive-chosen-plaintext attack* kripanalis dapat memilih blok *plaintext* yang lebih kecil untuk dienkrip, memodifikasi pilihannya berdasarkan hasil enkripsi sebelumnya.

5) *Chosen- ciphertext attack* membuat kripanalis untuk dapat memilih ciphertext yang berbeda untuk didekrip dan mempunyai akses terhadap plaintext yang dienkrip.

   Diketahui : C_1,$P_1 = D_k(C_1)$, C_2,$P_2 = D_k(C_2)$, ... C_i,$P_i = D_k(C_i)$

   Dicari : K

6) *Chosen- text attack* merupakan gabungan dari *chosen-plaintext attack* dan *chosen- ciphertext attack*. Dapat digunakan terhadap algoritma kunci publik, namun lebih sering digunakan untuk algoritma simetri

2.3.1 Serangan Aktif

Serangan ini dapat melibatkan perubahan dari isi pesan yang disampaikan. Ada empat kategori yang termasuk serangan aktif, yakni :

1) *masquerade* (penyamaran) terjadi ketika satu *entity* berpura-pura menjadi *entity* lain, yang memiliki hak akses. Biasanya dilakukan rangkaian autentikasi yang sah oleh pihak yang berhak, yang kemudian ditangkap, dipelajari, dan dipakai kembali oleh penyamar.
2) *replay* (pengiriman ulang) memerlukan keterlibatan dari pengangkapan pasif suatu data dan transmisi yang mengakibatkan dampak yang tidak terotorisasi.
3) *modification of message* (modifikasi pesan) dapat berupa penggantian sebagian isi pesan, penundaan penerimaan pesan, ataupun perubahan pada susunan urutan pesan.
4) *denial of service* (penolakan pelayanan) meliputi perintangan terhadap penggunaan atau manajemen fasilitas komunikasi, maupun gangguan dalam jaringan, dengan mendisfungsikan jaringan atau memenuhi kapasitas ukuran pesan sampai berlebihan, sehingga menurunkan kinerja dari jaringan tersebut.

Untuk dapat mendeteksi, memulihkan kerusakan, dan mengatasi serangan aktif, diperlukan perlindungan fisik terhadap seluruh fasilitas komunikasi dan jalur- jalurnya setiap waktu.

2.3.2 Serangan Pasif

Pencurian dengar atau pemantauan transmisi termasuk dalam serangan pasif. Serangan pasif terdiri dari dua jenis, yaitu: analisa jalur dan penyingkapan isi pesan. Tujuan dari serangan ini adalah perolehan informasi yang ditransmisikan. Penyingkapan isi pesan meliputi percakapan melalui telepon, pengiriman pesan melalui email, pertukaran file. Oleh karena itu, diperlukan suatu sistem yang dapat mencegah dan mengurangi resiko, agar hanya pihak yang berhak yang dapat mengetahui isi dari pesan yang disampaikan. Jika dibandingkan dengan serangan aktif, serangan pasif lebih sulit untuk dideteksi, tetapi masih ada cara yang dapat dilakukan untuk menggagalkannya dan apabila diserang, tidak banyak pengaruh pada isi pesan.

2.4 Algoritma Kriptografi Klasik dan Modern

Algoritma kriptografi terdiri dari dua jenis, yaitu algoritma kriptografi klasik dan modern. Algoritma kriptografi modern berhasil merahasiakan *key* (kunci) lebih baik daripada algoritma terbatas, dengan hanya menyembunyikan kunci, tanpa merahasiakan algoritma yang digunakan, yang sering disebut dengan *password*. Keamanan enkripsi hanya terletak pada kunci dan tidak tergantung pada kemungkinan algoritmanya dikenali. Sistem seperti inilah yang ada dalam algoritma DES dan RSA. Interval kemungkinan nilai kunci yang ada disebut *keyspace*.

2.4.1 Sistem Kriptografi Klasik

Sistem kriptografi klasik umumnya telah digunakan jauh sebelum era komputer. Kriptografi klasik juga dibagi menjadi dua jenis cipher yaitu cipher transposisi yang mengubah susunan huruf - huruf di dalam pesan dan cipher substitusi yang mengganti setiap huruf atau kelompok huruf dengan sebuah huruf atau kelompok huruf lain. Kriptografi klasik, teknik enkripsi yang digunakan adalah enkripsi simetris dimana kunci dekripsi sama dengan kunci enkripsi. Penyandian ini berorientasi pada karakter. Terdapat 5 bagian dalam sistem kriptografi klasik yaitu : (Sadikin, 2012)

1*). Plaintext*

Pesan atau data dalam bentuk aslinya yang dapat dibaca dan masukan bagi algoritma enkripsi.

2). *Secret Key*

Masukan bagi algoritma enkripsi merupakan nilai yang bebas terhadap teks asli dan menentukan hasil keluaran algoritma enkripsi.

3). *Ciphertext*

Hasil dari proses algoritma enkripsi dan teks asli dianggap telah tersembunyi.

4). Algoritma Enkripsi

Algoritma enkripsi memiliki 2 masukan yaitu teks asli dan kunci rahasia, kedua masukan tersebut akan diproses sehingga menghasilkan teks sandi.

5). Algoritma Dekripsi

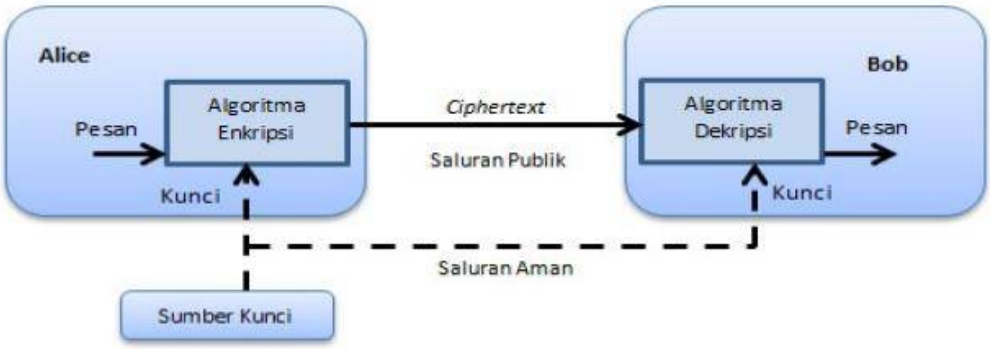
Algoritma dekripsi memiliki 2 masukan yaitu teks sandi dan kunci rahasia,

keduanya akan diproses sehingga menghasilkan teks asli.

2.4.2 Sistem Kriptografi Modern

Sistem kriptografi modern umumnya berorientasi pada bit. Untuk *public key cryptography*, diperlukan teknik enkripsi asimetris dimana kunci dekripsi tidak sama dengan kunci enkripsi. Enkripsi, dekripsi dan pembuatan kunci untuk teknik enkripsi asimetris memerlukan komputasi yang lebih intensif dibandingkan enkripsi simetris, karena enkripsi asimetris menggunakan bilangan - bilangan yang sangat besar. Beberapa mekanisme yang berkembang pada kriptografi modern (Sadikin, 2012)

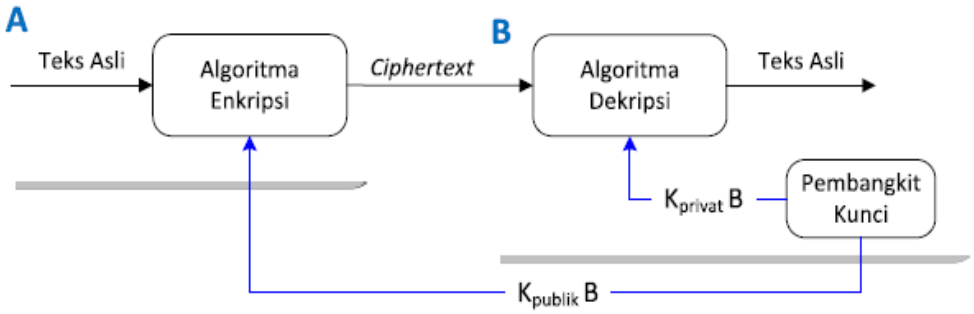
1. Penyandian dengan kunci simetrik *(symmetric key encipherment*). Penyandian dengan kunci simetrik adalah penyandian yang kunci enkripsi dan kunci dekripsi bernilai sama. Penyandian ini masih digunakan pada kriptografi modern. Skema penyandian ini dapat digambarkan pada Gambar 2.1



Sumber : Sadikin, 2012

Gambar 2.6 Sistem Kriptografi Simetrik

2. Penyandian dengan kunci asimetrik (asymmetric key encipherment) Penyandian dengan kunci asimetrik yang disebut juga dengan kunci publik adalah penyandian yang kunci enkripsi dan kunci dekripsi bernilai berbeda. Penyandian ini yang banyak dikembangkan. Skema penyandian ini dapat digambarkan pada Gambar 2.2.



Sumber : Fauzana, 2013

Gambar 2.7 Sistem Kriptografi Asimetrik

2.5 *Three-pass Protocol*

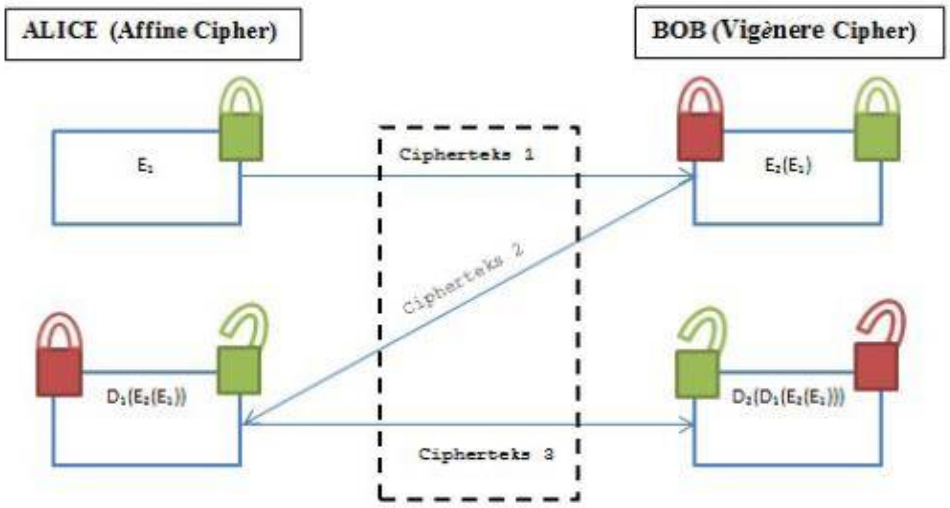
Dalam kriptografi *Three-pass protocol* adalah konsep yang memungkinkan satu pihak bisa dengan aman mengirim pesan kepada pihak kedua tanpa harus bertukar atau mendistribusikan kunci enkripsi. Protokol ini pertama kali dikembangkan oleh Adi Shamir seorang ahli kriptografi pada tahun 1980. Protokol ini dimodifikasi oleh *James Massey* dan *Jim K Omura* yang disebut dengan *Massey-omura*. Keduanya adalah pakar teori informasi pada tahun 1982 (Pramana, 2013).

Three-pass protocol memiliki beberapa tahapan untuk dapat menyampaikan pesan itu dari pengirim kepada penerima. Berikut tahapannya (Pramana, 2013)

1). Pengirim memilih kunci enkripsi eA. Pengirim mengenkripsi pesan dengan kunci dan mengirimkan pesan terenkripsi kepada penerima.

2). Penerima memiliki kunci enkripsi eB. Penerima mengenkripsi pesan C1 (eA,m) dengan kunci dan mengirim pesan terenkripsi lagi C2 (eB,C1(eA,m)) kepada pengirim.

3). Pengirim mendekripsi pesan C2 (eB,C1(eA,m)) dengan menggunakan dA dan mengirim lagi pesan C3 (eB,m) yang mana pesan ini dienkripsi oleh kunci penerima. Pengirim kembali mengirim pesan tersebut ke penerima dan kemudian penerima akan mendekripsi pesan tersebut dengan dB untuk bisa melihat pesan. Tahapan - tahapan ini dapat diilustrasikan pada Gambar 2.3



Sumber : Pramana, 2013

Gambar 2.8 Skema cara kerja Three-pass protoco

2.6 Metode Tranposisi Kolom

Pada metode transposisi kolom, huruf-huruf di dalam *plaintext* tetap, hanya saja urutannya diubah. Dengan kata lain algoritma ini melakukan transpose terhadap rangkaian karakter di dalam teks. Nama lain untuk metode ini adalah permutasi atau pengacakan (*scrambling*) karena transpose setiap karakter di dalam teks sama dengan mempermutasikan karakter-karkater tersebut. (Juwita Artanti Kusumaningtyas, 2018)

Pada metode transposisi kolom, *plaintext* tetap sama, tetapi urutannya diubah. Dengan kata lain, algoritma ini melakukan transpose terhadap rangkaian karakter di dalam teks. Nama lain untuk metode ini adalah permutasi, karena *transpose* setiap karakter di dalam teks sama dengan mempermutasikan karakter-karakter tersebut. Dalam transposisi kolom, pesan ditulis dalam deretan panjang tetap, dan kemudian membaca lagi kolom dengan kolom, dan kolom yang dipilih disesuaikan dengan rangka yang sudah ditetapkan. Kedua lebar baris dan permutasi dari kolom biasanya ditentukan oleh kata kunci.

Metode transposisi kolom cukup sederhana, yaitu dengan membagi *plaintext* menjadi blok-blok dengan panjang kunci (k) tertentu yang kemudian blok-blok tersebut disusun dalam bentuk baris dan kolom. Terdapat dua metode yang digunakan apabila panjang *plaintext* (n) tidak habis dibagi oleh kunci (k). Pertama adalah *irregular case*, yaitu melakukan enkripsi tanpa merubah *plaintext* dan yang kedua adalah *regular case* yaitu melakukan enkripsi setelah menambahkan karakter-karakter *dummy (pad)* sebanyak d dengan $0<d<n$ sehingga panjang plainteks habis dibagi kunci. Hasil enkripsi adalah dengan membaca

secara vertikal (tiap kolom) sesuai urutan kolom. Sebagai contoh, kata "zebras" adalah panjang 6 (sehingga baris yang panjang 6), dan permutasi ditentukan oleh urutan abjad dari huruf-huruf dalam kata kunci. Dalam hal ini, order akan "6 3 2 4 1 5". Dalam transposisi kolom biasa, spasi kadang dipenuhi dengan nulls; ruang yang dibiarkan kosong. Akhirnya, pesan tersebut dibacakan dalam kolom, dalam urutan yang ditentukan oleh kata kunci.

2.7 Algoritma Transposisi Kolom

Algoritma transposisi columnar merupakan algoritma klasik yang penggunaanya cukup sederhana. Pada tahap enkripsi transposition cipher tidak mengganti huruf *plaintext* untuk menghasilkan ciphertext layaknya subtitution cipher. Hasil enkripsi *transposition* cipher didapatkan dari menyusun ulang karakter *plaintext* dengan posisi yang berbeda.

Plaintext akan ditulis dalam matrik dengan panjang kolom sesuai dengan panjang karakter kunci yang digunakan. Penulisan *plaintext* ditulis dari baris per baris dimulai dengan baris pertama. *Ciphertext* transposisi columnar cipher dihasilkan dari penyusunan ulang plaintext. Kolom yang disusun pertama adalah kolom yang berhubungan dengan karakter sesuai urutan abjad.b Contoh enkripsi menggunakan pesan "MEET ME AT NEXT MID NIGHT" dan kunci "FANCY"Penyusunan dimulai dari kolom yang berhubungan dengan karakter urutan pertama pada abjad yaitu "A", kemudian "C", "F", "N" dan "Y". Hasil dari enkripsi tersebut adalah "EATITNIHMEXNETMGMEDT". Model matematis proses enkripsi transposisi columnar cipher menggunakan persamaan:

$$Ct\ of\ P = \begin{pmatrix} Y_o & \ldots\ldots & Y_l \\ X_{po_1} & \ldots\ldots & X_{pl_1} \\ X_{po_2} & \ldots\ldots & X_{pl_2} \\ \vdots & & \vdots \\ X_{po_m} & \ldots\ldots & X_{pl_m} \end{pmatrix}$$

Keterangan:

$Ct\ of\ P$ = *Columnar Transposition* dari pesan

Yo = Karakter pertama dari kunci

Yl = Karakter terakhir dari kunci

$X\ po1$ = Karakter pertama dari pesan yang berelasi dengan 0

$X\ pl1$ = Karakter pertama dari pesan yang berelasi dengan $Y\ l$

$X\ pom$ = Karakter terakhir dari pesan yang berelasi dengan $Y0$

$X\ plm$ = Karakter terakhir dari pesan yang berelasi dengan Y e

Jika pada persamaan (1) $Ct\ of\ P$ didefinisikan sebagai $CtPi$ dengan i adalah kolom pada persamaan (1). Maka cipher text dari proses enkripsi tersebut dapatdimodelkan sebagai :

Cp = {CtP1 + Ct P2 + Ct P3 +….+ CtPm} (2)

Keterangan:

Cp : Hasil enkripsi (*Ciphertext*)

m : Kolom terakhir dari persamaan (1)

Contoh :

Pada *plainteks*, misalnya terdapat 27 karakter yang dimasukkan seperti :

Plainteks : UNIVERSITASPEMBANGUNANMEDAN

Kunci : 3

Cara penyelesaiannya adalah dengan membagi setiap karakter dengan jumlah kolom, jumlah kolom ditentukan oleh kunci yang dapat dilihat pada table berikut ini.

Proses *Plaintext*

U	N	I
V	E	R
S	I	T
A	S	P
E	M	B
A	N	G
U	N	A
N	M	E
D	A	N

2.8 Penyandian Transposisi Kolom

Penyandian Transposisi Kolom dituliskan secara baris (biasa) dengan panjang yang telah ditentukan sebagai kunci-nya. Teks sandi-nya dibaca secara

kolom demi kolom dengan pengacakan melalui permutasian angka kunci nya. Panjang baris dan permutasian kolom nya disebut sebagai “kata kunci”.

Dalam prosesnya, kata kunci tersebut di definisikan dahulu dengan angka sesuai urutan abjad. Sedangkan proses untuk mengembalikan ke teks sandi ke teks aslinya dilakukan langkah kebalikan darinya. Lebih mudahnya dapat dilihat dalam contoh berikut :

Teks pesan asli :

UNIVERSITAS PEMBANGUNAN MEDAN

Kata kunci : TIGA yang berarti 3 kolom

U N I

V E R

S I T

A S P

E M B

A N G

U N A

N M E

D A N

Proses *Ciphertext*

U	V	S
N	E	I
I	R	T
A	E	A
S	M	N
P	B	G
U	N	D
N	M	A
A	E	N

Hasil penyandian (teks sandi) :

UVS NEI IRT AEA SMN PBG UND NMA AEN

2.9 Visual Basic Net 2010

Merupakan sebuah bahasa pemograman dan sebagai sarana (*tool*) untuk menghasilkan program-program aplikasi berbasiskan windows. Beberapa kemampuan atau manfaat dari Visual Basic diantaranya (Kholissodin. 2015)

1). Untuk membuat program aplikasi berbasiskan windows.

2). Untuk membuat obyek-obyek pembantu program, seperti *Control Active* X, File Help, Aplikasi Internet dan sebagainya.

3). Menguji program (*debugging*) dan menghasilkan program akhir berakhiran ”EXE” yang bersifat *executable* atau dapat langsung dijalankan. Keistimewaan utama dari *Visual Basic* adalah:

4). Menggunakan *platform* pembuatan program yang diberi nama *developer studio*, yang memiliki tampilan seperti C++ dan visual J++.

5). Memiliki kompiler handal yang dapat menghasilkan File *Executable* yang lebih cepat dan efisien.

6). Memiliki tambahan saran *wizard* yang baru. Tambahan kontrol-kontrol baru dan lebih canggih serta peningkatan kaidah struktur bahasa *Visual Basic*.

7). Kemampuan membuat *Active* X dan fasilitas internet yang lebih banyak.

8). Sarana akses yang lebih cepat dan andal untuk membuat aplikasi database yang berkemampuan tinggi.

9). *Visual Basic.net* memiliki beberapa versi baru edisi yang disesuaikan dengan kebutuhan pemakainya.

Dalam pemograman berbasiskan OOP (*Object Oriented Programming*), sebuah program dibagi menjadi bagian-bagian kecil yang disebut dengan obyek. Setiap obyek memiliki entiti terpisah dengan entiti-entiti lain dalam lingkungannya. Obyek-obyek yang terpisah ini dapat diolah sendiri-sendiri, dan setiap obyek memiliki sekumpulan sifat dan metode yang melakukan fungsi tertentu sesuai dengan yang telah diprogramkan kepadanya.

Adapun obyek-obyek yang dipergunakan dalam program ini adalah:

1). Project

Project adalah sekumpulan modul. Jadi project merupakan aplikasi itu sendiri. *Project* disimpan dalam file yang berakhiran VBP. Jika kita akan melaksanakan pembuatan program aplikasi, akan terdapat jendela *project*

yang berisi semua file yang dibutuhkan menjalankan program aplikasi *Visual Basic.net* pada saat pembuatan program aplikasi baru maka jendela project otomatis akan berisi object form1. Pada jendela *project* terdapat tiga icon yaitu *View Code, View Object*, dan *Toggle Folders*. Icon *View* Code dipakai untuk menampilkan jendela editor kode program. Icon *View Object* dipakai untuk menampilkan bentuk formulir (form) dan icon *Toggle Folders* digunakan untuk menampilkan folder

2). *Form*

Form adalah jendela yang dipakai untuk membuat user interface/tampilan. Secara otomatis akan tersedia form yang baru jika membuat suatu program aplikasi yang baru, dengan nama Form1. pada umumnya dalam suatu form terdapat garis titik-titik yang disebut dengan Grid. Untuk lebih memahami form ini maka di bawah ini terdapat gambar jendela form.

3). Toolbox

Toolbox adalah kumpulan dari obyek yang digunakan untuk membuat user interface (tampilan) serta control bagi program aplikasi. Untuk menempatkan control pada suatu form dapat dilakukan dengan klik ganda control dalam toolbox, kemudian mengubah besar dan ukurannya serta memindahkannya dengan metode Drag and Drop atau dengan cara mengklik kontrol toolbox, kemudian pindahkan pointer mouse jendela form. Kursor berubah menjadi Crosshair lalu tempatkan pada sudut kiri atas dimana kita inginkan kontrol tersebut diletakkan, tekan tombol mouse kiri dan tahan ketika menyeret kursor ke arah sudut kanan bawah.

4). *Properties*

Properties berisikan daftar struktur setting properti yang digunakan pada sebuah *object* terpilih. Kotak drop-down pada bagian atas jendela berisi daftar semua object pada form yang aktif. Ada tab tampilan, yaitu *alphabetic* (urut abjad) dan *categorized* (urut berdasarkan kelompok).

5). Kode Program

Kode program adalah serangkaian tulisan perintah yang akan dilaksanakan jika suatu obyek dijalankan. Kode program ini mengkontrol dan menentukan jalannya suatu obyek.

6). *Event*

Event adalah peristiwa atau kejadian yang diterima suatu obyek, misalnya klik, seret, tunjuk, dan lain sebagainya.

7). Metode (*Methods*)

Metode adalah serangkaian perintah yang sudah tersedia pada suatu obyek yang dapat diminta untuk mengerjakan tugas khusus.

8). Module

Module dapat disejajarkan dengan form, tetapi module tidak mengandung obyek. Module berisikan prosedur umum, deklarasi variabel dan definisi konstanta yang digunakan oleh aplikasi.

2.10 Metodologi Berorientasi Objek

Metodologi akan membantu dalam pengaturan dari keseluruhan proyek dengan memecah proses pengembangannya menjadi bagian-bagian yang lebih

kecil, untuk memberikan spesifikasi pada keinginan dan tidak adanya ketergantungan dari setiap *task* dengan cara merencanakannya terlebih dulu, dilanjutkan dengan penjadwalan dan pengawasan untuk setiap laporan selama pembangunan sistem. Metodologi yang dapat digunakan terbagi menjadi dua,yaitu : *structured* (terstruktur) dan *object oriented* (berorientasi objek). Ciri-ciri metodologi berorientasi objek adalah (Ariyus: 2006)

1). Adanya *encapsulation* (pemaketan data-data dan operasi-operasi ke dalam sebuah objek), tidak ada data yang bersifat global.

2). Adanya *inheritance* (mekanisme untuk mendefinisikan sebuah kelas baru dalam sebuah kelas yang masih berjalan), adanya *polymorphism* (kemampuan untuk menyembunyikan implementasi lainnya di belakang antarmuka biasa).

3). Lebih ditekankan pada tahapan awal dengan komunikasi lebih efektif

4). Model yang sudah dirancang dapat dikembangkan dan diperbaiki

Adapun kelebihan dari metodologi berorientasi objek bila dibandingkan yang terstruktur:

1) Dapat digunakan kembali,sehingga akan lebih efisien daripada harus menulis ulang
2) Perawatan yang mudah
3) Lebih mudah untuk dites, karena setiap modul bersifat tidak tergantung pada modul lain, dapat dimasukkan ke dalam sistem besar secara terpisah dan dapat didefinisikan secara jelas.
4) Sistem yang dibuat dapat berupa sebuah sistem yang besar dan komplek

2.11 *Unified Modelling Language (UML)*

Diagram UML merupakan salah satu dari alat terpenting dari beberapa analisis kebutuhan dan perancangan dari sistem *software* (perangkat lunak) yang berbasiskan pada orientasi objek. Dalam diagram akan terlihat kelas-kelas, atribut-atribut dan operasi-operasi sebagaimana berhubungan sesuai dengan macam-macam tipe yang ada dalam kelas-kelas. UML adalah sebuah bahasa grafik standar dalam permodelan *software* yang berdasarkan pada orientasi objek. UML mulai dikembangkan sejak pertengahan tahun 1990-an sebagai sebuah usaha bersama antara *James Rumbaugh, Grady Booch*, dan *Ivar Jacobson*, dengan pengembangan notasi masing-masing di awal tahun 1990-an. Lambang 'U' yang berasal dari UML diartikan untuk *"unified"* (penggabungan), yang menunjukkan tiga pengembang yang mengkombinasikan fitur- fitur terbaik bahasa-bahasa yang telah dikembangkan sebelumnya. UML terbagi menjadi beberapa tipe diagram, yakni : (Fadhlan. 2015)

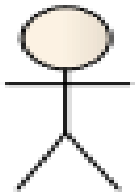
1) *Class diagram* (diagram kelas) yang mendeskripsikan kelas-kelas yang ada dan sekaligus hubungan antar kelas tersebut.
2) *Interaction diagram* (diagram interaksi) yang terdiri dari dua jenis,yaitu: *sequence diagram* dan *collaboration diagram.* Diagram ini menitik beratkan pada perilaku sistem dalam jangka waktu, dimana setiap objek melakukan interaksi masing- masing.
3) *State diagram and activity diagram* yang menunjukkan bagaimana sistem berjalan internal.
4) *Component diagram and deployment diagram* yang akan menonjolkan

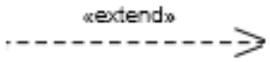
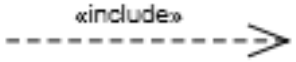
bagaimana bermacam-macam komponen yang ada di dalam sistem diurutkan secara logika dan fisik.

UML itu sendiri tidak hanya berupa notasi-notasi yang digambarkan dalam diagram, tetapi juga memiliki fitur tambahan yang menarik :

1) Rincian semantik yang mendeskripsikan pengertian dari notasi yang berbeda-beda.
2) Perpanjangan mekanisme yang memungkinkan perancang *software* dalam merepresentasikan konsep yang bukan merupakan bagian dari inti UML
3) Sebuah bahasa tekstual yang telah terasosiasi yang dikenal dengan nama *OCL (Object Constraint Language)* yang memungkinkan berbagai keadaan mengenai elemen-elemen yang ada di dalam diagram.

Tabel 2.1 Tentang Simbol UML *(Unified Modelling Language)*

	*ACTOR* Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama *actor*.
	*USE CASE* Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesar antar unit atau actor biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case.

	ASOSIASI/*ASSOCIATION* Komunikasi antara *actor* dan use case yang berpartisipasi pada use case atau use case memiliki interraksi dengan *actor*.
	EKSTENSI/*EXTEND* Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan memiliki nama depan yang sama dengan use case yang di tambahkan.
	GENERALISASI/*GENERALIZATION* Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
	MENGGUNAKAN/*INCLUDE* Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsional atau sebagai syarat dijalankan use case ini.

2.12 Interaction diagram

Interaction diagram (diagam interaksi) digunakan sebagai model dari aspek dinamika sebuah sistem *software*. Dari diagram tersebut dapat dilihat bagaimana sekumpulan aktor dan objek berkomunikasi dengan tiap-tiap langkah dari *use case* atau beberapa fungsionalitas. Diagam interaksi menunjukkan beberapa tipe komunikasi yang berbeda. Hal ini meliputi pesan yang melintas dalam hubungan (jaringan), panggilan prosedur sederhana, dan semuanya itu disebut *messages*. Elemen-elemen yang dapat ditemukan dalam *interaction diagram* antara lain: *instance* (hasil penurunan sifat) dari aktor-aktor dan kelas-

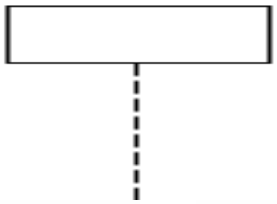
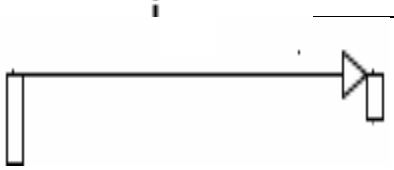
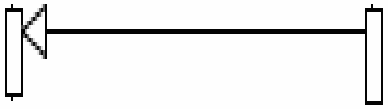
kelas yang ada digambarkan dengan kotak-kotak yang berisi kelas dan identifikasi objek yang ditandai dengan garis bawah, *messages* (pesan-pesan) yang digambarkan dengan tanda panah dari aktor ke objek atau dari objek ke objek. (Lusiana.2010)

Dua bentuk diagram interaksi adalah *sequence diagram* dan *collaboration diagram.* Dalam bentuk sederhana, keduanya mengandung informasi yang sama mengenai interaksi dan dapat dikonversikan ke dalam berbagai tipe. *Sequence diagram* lebih menitik beratkan pada urutan kegiatan pada sebuah *time line.* Sedangkan *collaboration diagram* (diagram kolaborasi) isinya lebih padat, termasuk hubungan- hubungan yang ada diantara objek dan aktor.

2.12.1 *Sequence diagram*

Menggambarkan urutan dari pesan-pesan yang dilakukan oleh sekumpulan objek ataupun aktor internal. Objek akan bergerak dari kiri menuju ke kanan melintasi diagram, biasanya digambarkan dengan aktor yang berada di posisi paling kiri. Dimensi vertikal merepresentasikan waktu. Poin awal terletak di bagian teratas dari diagram dan waktu progres berada dibawahnya, turun terus hingga mencapai bagian terbawah dalam gambar. Garis vertikal yang disebut *lifeline*, menghubungkan setiap objek atau aktor. *Lifeline* yang berbentuk sebuah kotak disebut *activation box* (kotak aktivasi), dalam jangka waktu dimana objek tesebut menunjukkan komputasi, akan dikatakan bahwa objek itu memiliki *life activation*. (Syafari Anjar.2007)

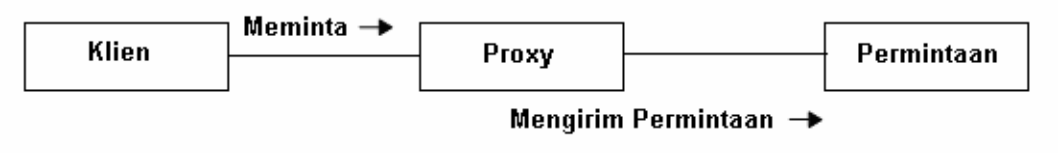
Tabel 2.2 Tentang Simbol *Sequence diagram*

GAMBAR	NAMA	KETERANGAN
	*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

2.12.2 *Collaboration diagram*

Menggambarkan beberapa objek yang bekerja bersama, ditonjolkan sebagai sebuah grafik dengan sekumpulan objek dan aktor secara vertikal. Diagram kolaborasi ini sangat mirip dengan sebuah *instance diagram,* kecuali pada penggambaran hubungan asosiasi. Juga memiliki persamaan dengan *sequence diagram*, namun tidak memiliki kotak aktivasi dan *lifeline*. Hanya terdapat hubungan komunikasi antara setiap pasangan objek dengan pengiriman pesan.

Contoh *collaboration diagram*



Sumber : Veronika, 2010

Gambar 2.9 Collaboration Diagram

2. 13 Activity diagram

Sebuah *activity diagram* (diagram aktivitas) menyerupai sebuah *state diagram*, selain memiliki simbol-simbol tambahan dan digunakan dalam konteks yang berbeda. Jika pada *state diagram* kebanyakan transisi disebabkan oleh kegiatan *external* (yang berada/berasal dari luar), maka pada *activity diagram* kebanyakan transisi disebabkan oleh kegiatan *internal* (yang berada/berasal dari dalam). Diagram ativitas digunakan untuk memahami aliran dari pekerjaan yang dilakukan oleh sebuah objek atau komponen yang ditampilkan. Kelebihan dari diagram aktivitas adalah representasi dari aktivitas-aktivitas yang mungkin dilakukan. Kemungkinan itu digambarkan dalam tiga bentuk, yakni : *fork*, *join*, dan *rendezvous*.

Fork berasal dari satu transisi yang kemudian melebar menjadi beberapa transisi. Sedangakan *join* merupakan kebalikan dari *fork*, berasal dari beberapa transisi yang selanjutnya berkumpul dan menjadi satu transisi. *Rendezvous* adalah gabungan beberapa transisi yang kemudian akan menyebar kembali menjadi lebih dari satu transisi. (putra, 2015)

: *fork*

:join

:rendezvous

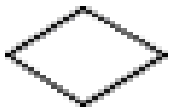
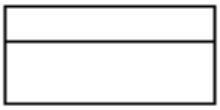
Gambar 2.10 Simbol Fork, Join, Rendezvous

2.14 *Class diagram*

Intisari dari UML *class diagram* adalah mendeskripsikan data yang ditemukan di dalam sebuah sistem *software*. Simbol-simbol utama yang tampak di dalam *class diagram*, antara lain :

1) Kelas-kelas yang merepresentasikan sebagai sebuah kotak dengan nama kelas itu di dalamnya.
2). Asosiasi yang menunjukkan penghubung yang berada di kelas yang satu dengan kelas lainnya.
3). Atribut yang merupakan data sederhana yang ditemukan di dalam kelas-kelas dan kelas turunan yang telah mendapatkan penurunan sifat induknya
4). Operasi yang merepresentasikan fungsi-fungsi yang ada di setiap kelas dan kelas turunannya.
5). Generalisasi dimana kelompok dari kelas-kelas menjadi hirarki masing-masing.

Tabel 2.3 Tentang Simbol *Class diagram*

GAMBAR	NAMA	KETERANGAN
	Generalization	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
	*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
	*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	Association	Apa yang menghubungkan antara objek satu dengan objek lainnya

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

1). Studi Pustaka

Pada tahap ini dilakukan studi literature yang bertujuan mengumpulkan, mempelajari dan memilih bahan dan sumber.

2). Tahapan Perencanaan

Kegiatan yang dilakukan pada tahap ini termasuk mengidentifikasi masalah, merumuskan masalah, menentukan batas masalah dalam penelitian, dan menentukan tujuan penelitian

3). Tahapan Perancangan

Perancangan sistem merupakan tahap pengembangan setelah perencanaan penelitian dilakukan. Beberapa proses dalam perancangan program tersebut seperti, perancangan *input* dan *output flowchart* keamanan data menggunakan algoritma kriptografi Transposisi.

4). Uji Coba Program

Setelah program selesai, program akan diuji untuk mengetahui apakah program telah bekerja dengan benar dan sesuai dengan sistem.

5). Pembuatan Kesimpulan

Pada tahap akhir ini adalah pembuatan kesimpulan atau ringkasan dari skripsi ini dan kesimpulan tentang program yang telah dibuat

3.2 Metode Pengumpulan Data

Dalam mengumpulkan data, penulis melakukan beberapa cara, yaitu *literatur*, jurnal, dan berbagai bacaan terkait dengan judul penelitian. Dalam penelitian, beberapa metode dilakukan, yaitu:

1) Mempelajari bermacam sumber *literatur*. Yaitu dari beberapa sumber buku, dan *internet* khususnya yang berhubungan dengan Transposisi Kolom.
2) Konsultasi dengan dosen pembimbing dan pihak – pihak lain yang bisa membantu.
3) Pencarian data melalui referensi skripsi alumni yang terdapat di perpustakaan UNPAB.
4) Mencoba memasukkan berbagai jenis file dengan berbagai ukuran untuk mengetahui seberapa efektif program tersebut.

3.3 Analisa Sistem Yang Berjalan

Dalam analisis ini dekomposisi dan investigasi masalah ini adalah untuk mendapatkan pemahaman, pengertian dan makna sebenarnya dari masalah ini. Dalam keamanan komputer memiliki istilah enkripsi, yang merupakan salah satu jenis enkripsi yang menggunakan metode transposisi kolom. Untuk mendapatkan hasil *ciphertext*, gunakan kolom dan baris untuk mengkonversi. Algoritma kriptografi adalah metode keamanan informasi dengan menambahkan *plaintext* dengan kunci untuk menghasilkan *ciphertext* kongruen

Input yang diproses dalam aplikasi dirancang dalam bentuk karakter alfanumerik yang akan diproses dengan algoritma transposisi kolom. Jadi dalam hal mengirim dan menerima pesan dapat mengenkripsi dan mendekripsi pesan teks.

3.4 Rancangan Penelitian

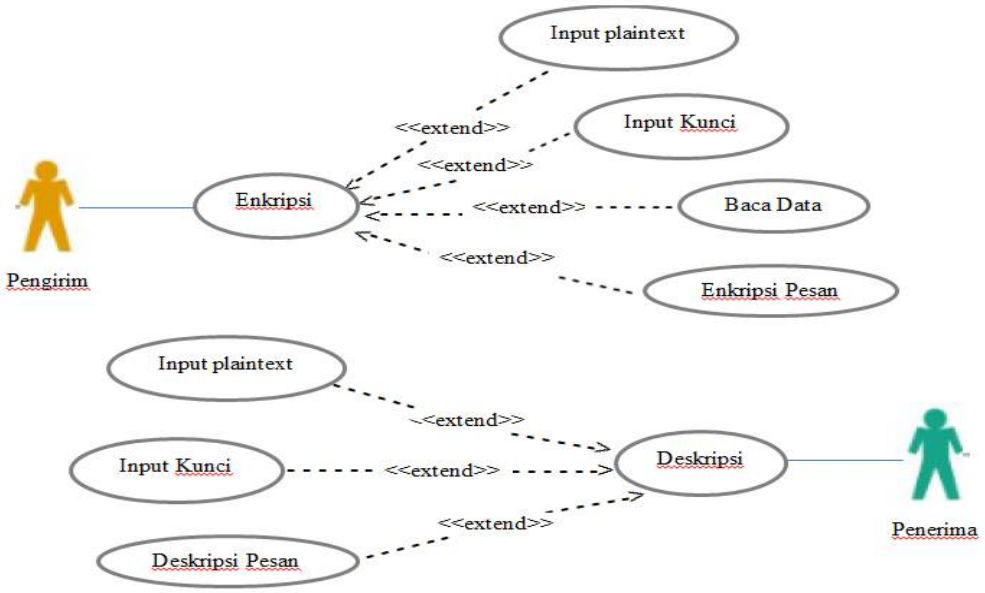
Mengenkripsi pesan teks atau informasi dalam teori di atas dapat dilakukan dengan menggunakan algoritma Transposisi Kolom dan membutuhkan *Padding*, di mana *Padding* berguna untuk pertukaran posisi pesan yang telah di sandi. nilai dalam *plaintext* yang akan diinput dengan nilai K yang merupakan kuncinya. Program ini menggunakan kunci yang dapat melindungi dan membuat informasi lebih aman dan mengamankan kerahasiaan dan keasliannya sehingga sulit dideteksi oleh pihak yang tidak berwenang karena penyandian tidak hanya dapat menyandikan huruf tetapi juga dapat digunakan untuk angka, simbol, tanda baca dan lain-lain. Dalam algoritma ini pemilihan kunci dilakukan secara acak dengan beberapa peluang untuk menemukan kunci yang sesuai dengan sifat dari algoritma Transposisi Kolom.

3.4.1 Pemodelan Sistem

Permodelan Sistem ini menggunakan diagram UML (*Unified Modelling Language*) untuk mengambarkan bagaimana sistem akan bekerja khususnya sistem yang berorientasi objek. Diagram UML yang digunakan adalah *Use Case Diagram, Activity Diagram* dan *Sequence Diagram.*

1. ***Use Case Diagram***

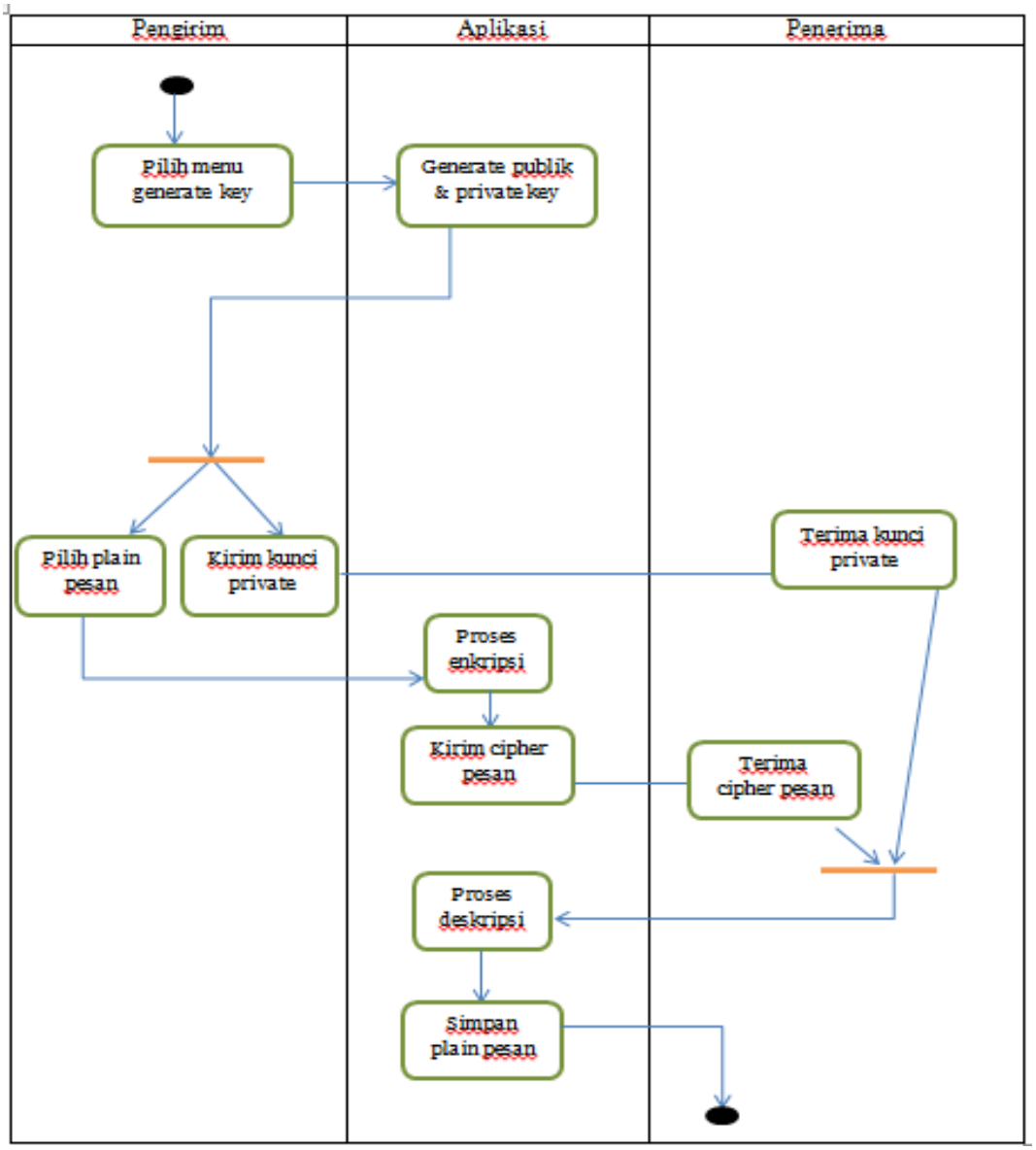
Diagram *Use Case* mendekripsikan sebuah interaksi antara satu atau lebih pengguna dengan sistem yang akan dibuat. Diagram ini menggambarkan kebutuhan fungsional yang telah dirincikan sebelumnya. Diagram *use case* untuk kebutuhan fungsional dapat dilihat pada gambar 3.1



Gambar 3.1 Diagram *Use Case*

2. ***Activity Diagram***

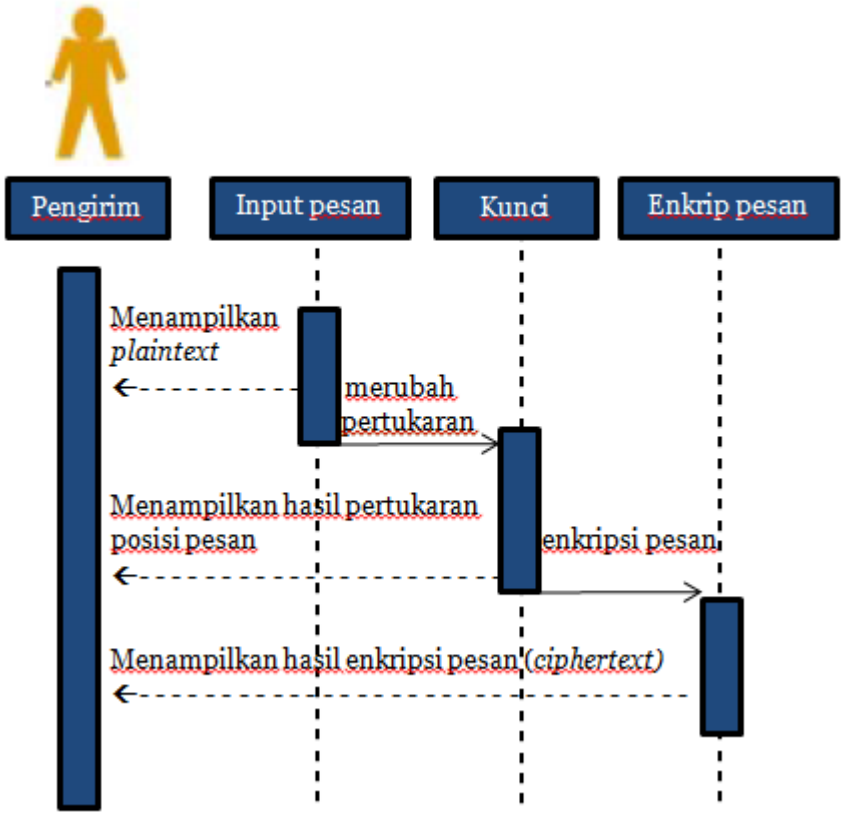
Activity Diagram menggambarkan proses-proses yang terjadi mulai aktifitas dimulai sampai aktifitas berhenti, untuk kebutuhan proses dalam sistem yang akan dibangun, digambarkan pada gambar 3.2



Gambar 3.2 *Activity Diagram*

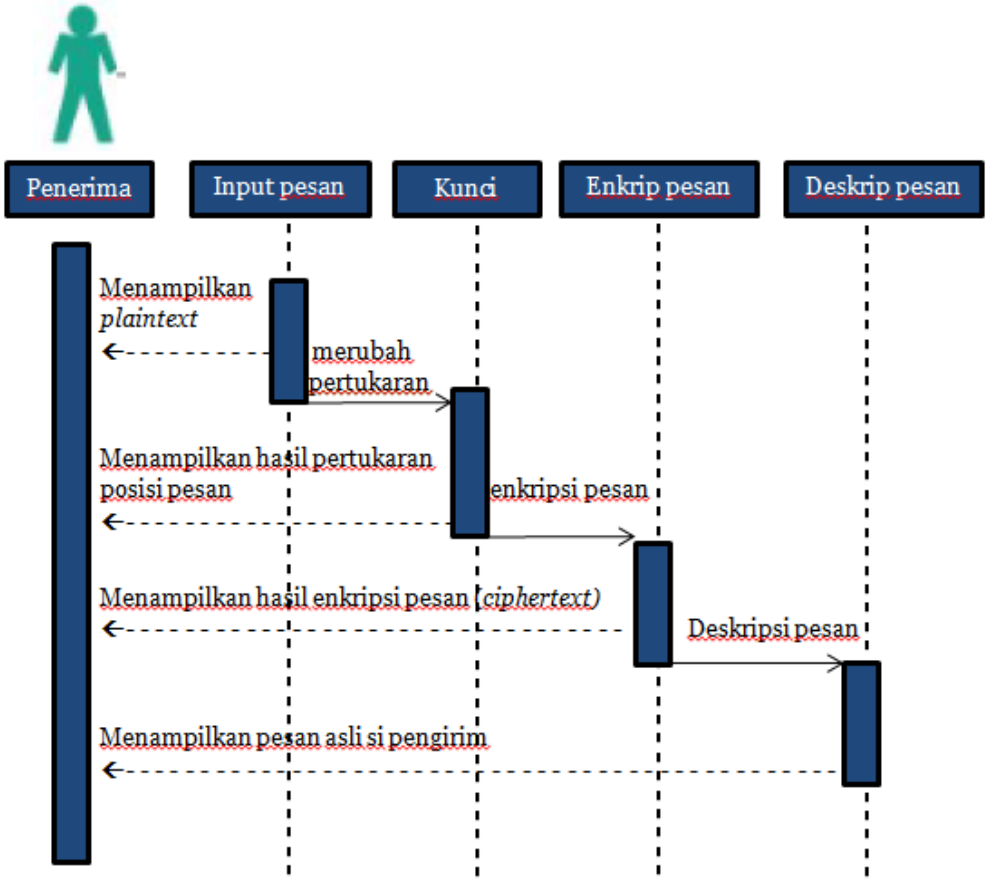
3. *Sequence Diagram*

Sequence diagram merupakan diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Digambarkan pada gambar 3.3



Gambar 3.3 *Sequence Diagram* untuk proses enkripsi

Pada gambar 3.3 dapat dilihat interaksi antara sistem dengan pengirim pesan secara berurutan. Aksi pengirim pesan terhadap sistem ditunjukkan dengan tanda panah garis penuh, sedangkan respon sistem terhadap pengirim pesan ditunjukkan dengan tanda panah garis putus-putus.



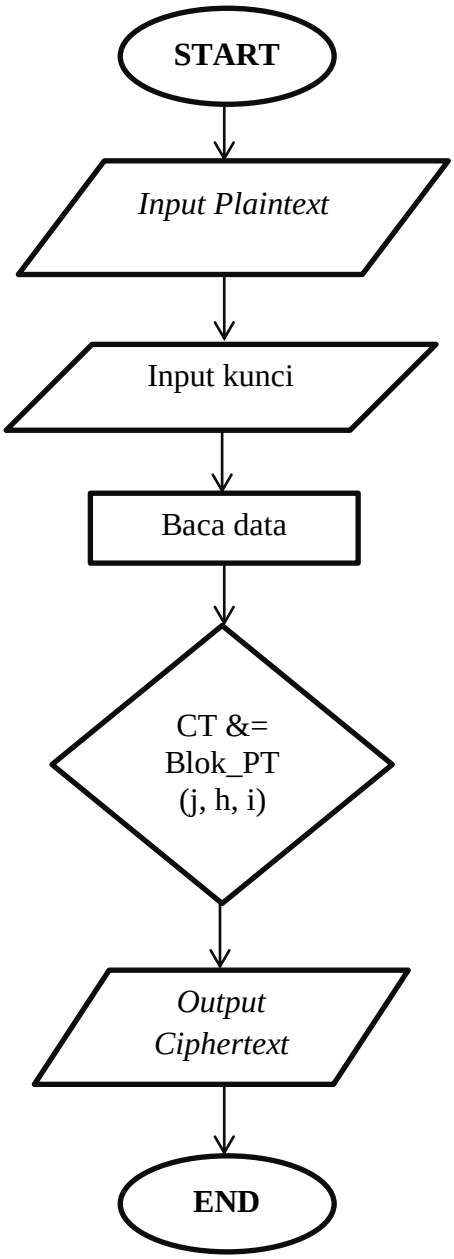
Gambar 3.4 *Sequence Diagram* proses deskripsi

Pada gambar 3.4 dapat dilihat interaksi antara sistem dengan pengirim pesan secara berurutan. Aksi pengirim pesan terhadap sistem ditunjukkan dengan tanda panah garis penuh, sedangkan respon sistem terhadap pengirim pesan ditunjukkan dengan tanda panah garis putus-putus.

3.5 *Flowchart* Transposisi Kolom

Flowchart Transposisi Kolom merupakan aliran data untuk proses penyadian Transposisi Kolom dapat dilihat pada gambar

1. **Proses Enkripsi**



Gambar 3.5 *Flowchart* Proses Enkripsi pesan

Dalam proses Enkripsi, pada awalnya memasukkan kata atau pesan dan memasukkan kunci sehingga setelah dienkripsi muncul nilai dalam *padding* yang terkandung dalam *plaintext, key,* dan *chipertext* maka hasil *plaintext* terenkripsi akan ditukar posisi menggunakan *padding* dengan nilai kunci, yang kemudian akan menghasilkan pesan atau teks seperti pada awalnya.

Dalam proses enkripsi terdapat beberapa hitungan yang akan menghasilkan *plaintext* yang telah di enkripsi. Misalnya saya mengambil 27 karakter dengan jumlah kolom dan baris terdiri dari 3 kolom dan 3 baris.

Panjang teks = 27 karakter

Kunci = Kolom x Baris = 3

Jumlah Blok = = $\frac{27}{3\ x\ 3} = \frac{27}{9} = 3$

Padding = (3 x 3 x 3) – 27 = 0

Pada tahap enkripsi ini *plaintext* akan dirubah menjadi *ciphertext* menggunakan algoritma Transposisi Kolom.

Plaintext = UNIVERSITASPEMBANGUNANMEDAN

Kunci = 3

Padding = UNIVERSITASPEMBANGUNANMEDAN

Ciphertext = UVSNEIIRTAEASMNPBGUNDNMAAEN

Padding itu berfungsi sebagai proses pertukaran kata atau huruf yang telah di enkripsi. Hasil yang telah di enkripsi dengan kata di atas telah menghasilkan *Ciphertext* yang telah di enkripsi. Dibawah ini terdapat kolom dan baris yang telah

di buat. Kunci yang telah di masukkan akan menghasilkan kolom dan baris yang sama.

Panjang Teks = 27

Kunci = 3

Jumlah Blok = 3

Jumlah *Padding* = 0

Blok *Plaintext* Blok *Ciphertext*

U	N	I
V	E	R
S	I	T

U	V	S
N	E	I
I	R	T

A	S	P
E	M	B
A	N	G

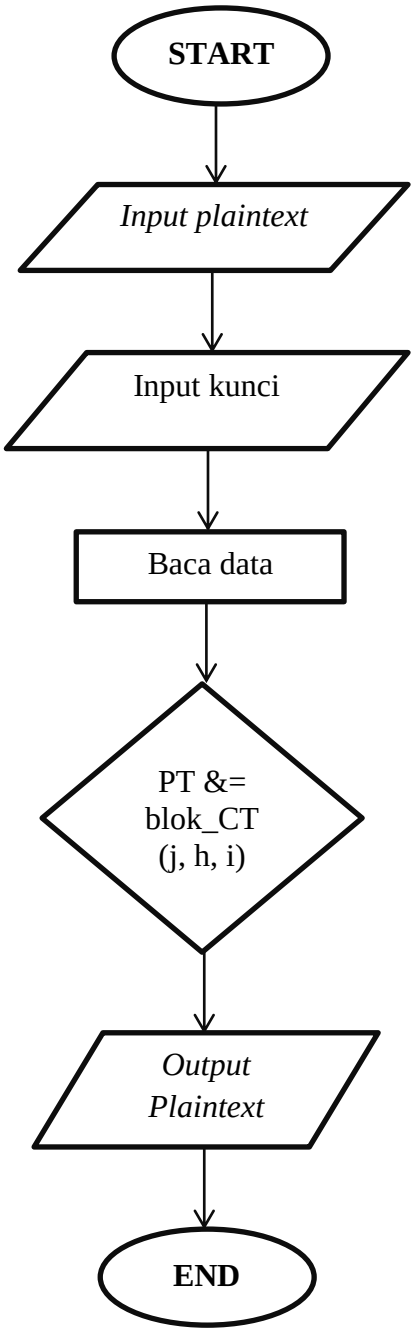
A	E	A
S	M	N
P	B	G

U	N	A
N	M	E
D	A	N

U	N	D
N	M	A
A	E	N

Hasil Penyandian teks sandi menggunakan proses enkripsi

2. Proses Deskripsi



Gambar 3.6 *Flowchart* Proses Deskripsi

Dalam proses dekripsi, pada awalnya kata atau pesan yang telah di enkripsi itu masih ter acak acak dengan kunci yang telah di berikan. Maka dari itu pesan yang di enkripsi harus di deskripsi kan kembali, agar supaya pesan asli tersebut di terima dengan kata awal pas waktu pengiriman.

C = 27 karakter

Kunci = Kolom x Baris = 3

Jumlah Blok = = $\frac{27}{3\ x\ 3}$ = $\frac{27}{9}$ = 3

Padding = (3 x 3 x 3) – 27 = 0

Setelah tahap proses deskripsi ini. *Plaintext* telah berubah menjadi teks aslinya *Plaintext* menggunakan algoritma Transposisi Kolom.

Ciphertext = UVSNEIIRTAEASMNPBGUNDNMAAEN

Kunci = 3

Plaintext = UNIVERSITASPEMBANGUNANMEDAN

Didalam Proses *Ciphertext* padding telah di hilangkan. *Padding* itu hanya berfungsi untuk proses enkripsi pesan.

Panjang Teks = 27

Kunci = 3

Jumlah Blok = 3

Blok *Ciphertext*

U	V	S
N	E	I
I	R	T

A	E	A
S	M	N
P	B	G

U	N	D
N	M	A
A	E	N

Blok *Plaintext*

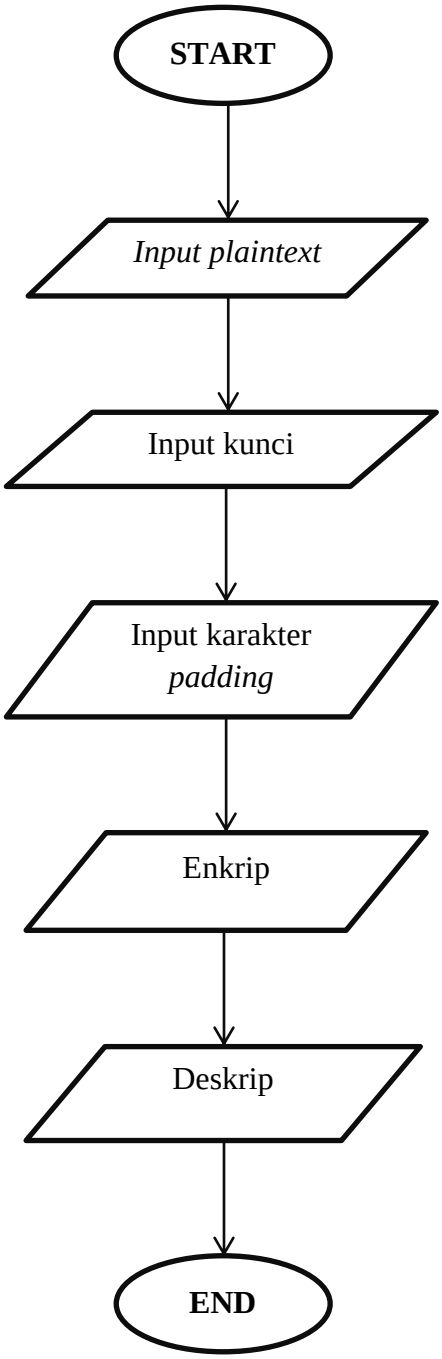
U	N	I
V	E	R
S	I	T

A	S	P
E	M	B
A	N	G

U	N	A
N	M	E
D	A	N

Hasil Penyandian teks sandi menggunakan proses Deskripsi

3.6 Pembangkit Kunci



Gambar 3.7 *Flowchart* Pembangkitan Kunci

Dalam algoritma Transposisi Kolom, hanya ada satu kunci rahasia untuk mengubah teks menjadi teks sandi, menghasilkan kunci dan kemudian mengenkripsi itu. Kunci dapat dihasilkan jika kunci yang digunakan memenuhi persyaratan pertukaran posisi, yaitu:

```
PAD = PT
For I = To JPad - 1
```

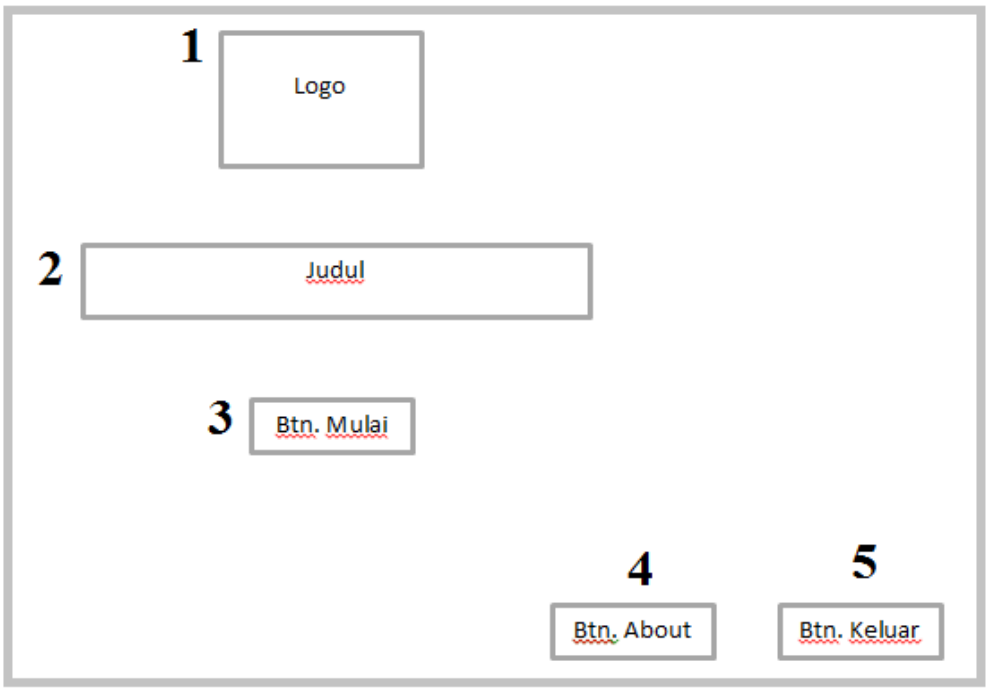
Di mana pertukaran posisi dilakukan secara acak, pada jumlah pertukaran posisi, jumlah pertukaran posisi akan bersama dengan kunci. Pilihan kunci tidak hanya dalam bentuk huruf, tetapi bisa juga dengan angka dan simbol sehingga semakin sulit kuncinya, semakin sulit bagi seorang *cryptanalyst* untuk menebak kuncinya.

3.7 Perancangan Sistem

Pada perancangan pengamanan pesan menggunakan algoritma transposisi, akan dijelaskan mengenai rancangan dan hal – hal yang dikerjakan serta fitur – fitur yang akan dipakai pada aplikasi tersebut. Hal ini bertujuan untuk menjelaskan tahapan – tahapan yang dikerjakan, prosedur penggunaan, *design* tampilan, serta spesifikasi sistem dari segi perangkat lunak maupun perangkat keras yang di gunakan dalam proses perancangan

37.1 Halaman Awal

Pada halaman awal ini kita dapat melihat judul utama pada sistem, logo universitas, judul, dan sebuah tombol untuk masuk dan memulai sistem dan tombol akhir untuk menghentikan sistem tersebut. Halaman awal ditunjukkan pada gambar 333333



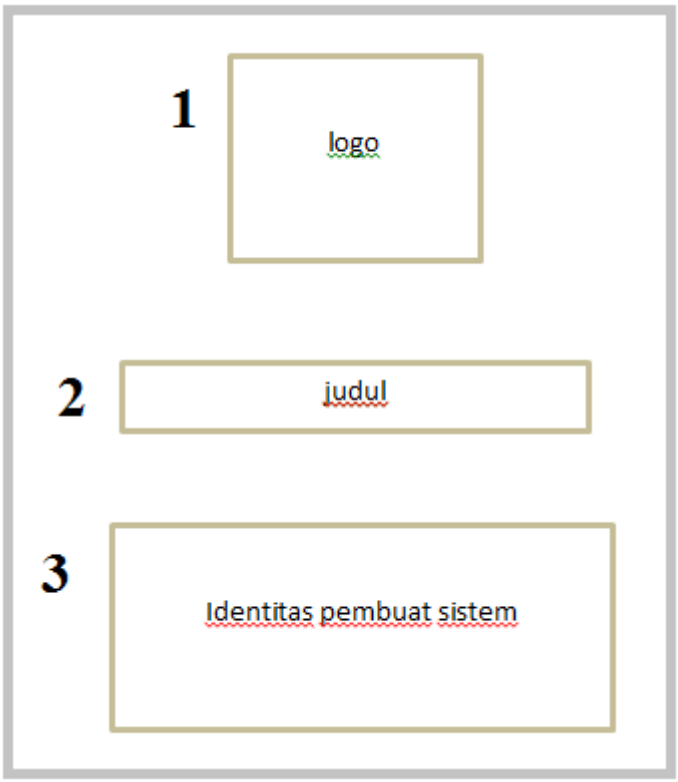
Gambar 3.8 Perancangan Halaman Awal

Keterangan gambar:

1. *Picturebox* : digunakan untuk logo Universitas Panca Budi
2. *Label* : digunakan untuk judul sistem
3. *Button* Mulai : Berfungsi untuk masuk dan memulai sistem
4. *Button About* : Berfungsi untuk menampilkan identitas pembuat sistem
5. *Button* Keluar : Berfungsi untuk mengakhiri sistem yang berjalan

37.2 Tampilan About

Pada Tampilan About ini kita dapat melihat judul utama pada sistem, logo universitas, judul, dan identitas pembuat sistem. Tampilan About ditunjukkan pada gambar 333333

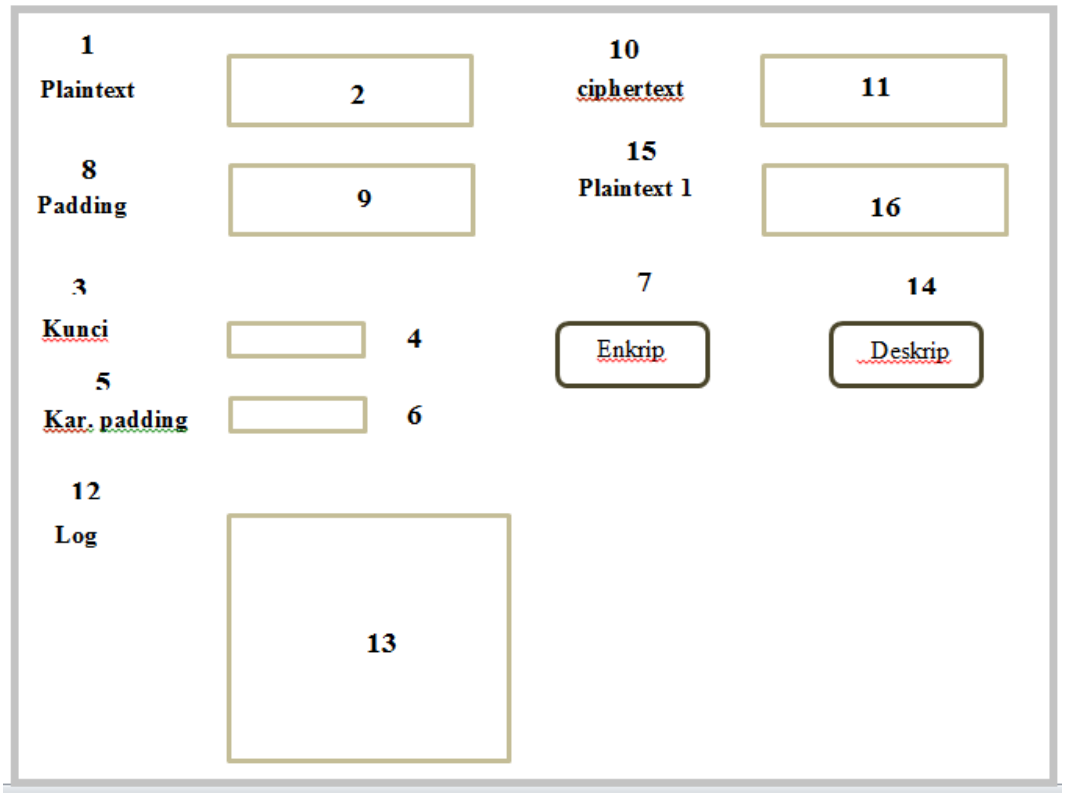


Gambar 3.9 Perancangan Tampilan About untuk menampilkan identitas pembuat sistem

Keterangan gambar:

1. *Picturebox* : digunakan untuk logo Universitas Panca Budi
2. *Label* : digunakan untuk judul sistem
3. *Label* : digunakan untuk identitas pembuat sistem

3.7.3 Tampilan Transposisi Kolom



Gambar 3.10 Perancangan Tampilan Transposisi Kolom

Keterangan gambar

1. *Label* : digunakan untuk tulisan *plaintext*
2. *Rich Text Box* : digunakan untuk menginput *plaintext* atau membaca isi pesan yang akan di enkripsi
3. *Label* : digunakan untuk tulisan kunci
4. *Textbox* : digunakan untuk pertukaran posisi pesan yang telah di enkripsi
5. *Label* : digunakan untuk tulisan karakter *padding*
6. *Rich Textbox* : digunakan untuk mengisi kolom yang tidak terisi kata

7. *Button* : berfungsi untuk menampilkan pesan yang telah di sandi
8. *Label* : digunakan untuk tulisan *padding*
9. *Rich Textbox* : digunakan untuk pertukaran posisi pesan
10. *Label* : digunakan untuk tulisan *ciphertext*
11. *Rich Textbox* : digunakan untuk hasil pesan yang telah di enkrip dan di sandi
12. *Label* : digunakan untuk tulisan Log
13. *Rich Textbox* : digunakan untuk semua hasil pertukaran yang dilakukan
14. *Button* : berfungsi untuk menampilkan pesan asli
15. *Label* : digunakan untuk tulisan *plaintext* 1
16. *Rich Textbox* : digunakan untuk menmpilkan pesan asli dari si pengirim

3.8 Spesifikasi Sistem

Penyandian Transposisi Kolom diterapkan pada proses pengiriman pesan sebagai keamanan data kerahasian pesan. Dalam perancangan dibutuhkan perangkat pendukung *software* dan *hardware.* Adapun perangkat tersebut sebagai berikut :

1. Perangkat Lunak

Software yang digunakan adalah sistem operasi *windows 10* dan *Microsoft Visual Basic 2010*

2. Perangkat Keras

Hardware yang penulis gunakan untuk pendukung peranancangan adalah 1 unit laptop

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Pada tahap implementasi sistem ini adalah merupakan sebuah tahap aplikasi yang telah dirancang dan dijalankan. Tahap ini menunjukkan setiap proses yang sedang berjalan dan mampu bekerja seperti yang diharapkan. Proses desain menggunakan *visual basic net 2010* yang ditampilkan dalam bentuk form - form sebagai sarana bagi pengguna untuk melakukan proses implementasi.

Microsoft Visual Studio 2010 adalah varian terbaru dari *software Visual Studio*. *Software* ini terbagi menjadi 3 versi yaitu *Ultimate, Premium* dan *Professional*. Perbedaan ketiganya anda bisa lihat deskripsinya

Sistem Operasi :

- Windows 7
- Windows Server 2003 R2 (32-Bit x64 dan Service Pack 2
- Windows Server 2008 R2 dan Service Pack 2
- Windows Vista Service Park 2
- Windows XP Service Park 3

Spesifikasi Hardware yang dibutuhkan

- Prosesor dengan kecepatan minimum 1.6GHz
- Memori 1 GB (1.5 GB jika menggunakan virtual machine)
- Hardisk free space 3 GB
- DirectX 9
- Display minimum 1024 x 768

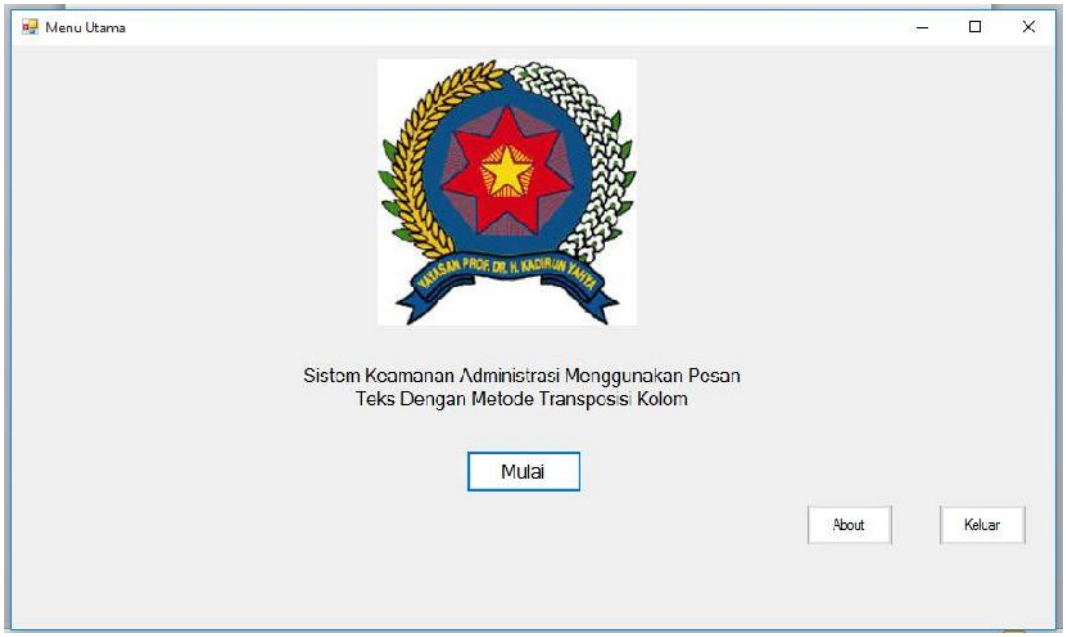
4.2 Pengujian Sistem

Dalam pengujian suatu sistem memiliki tujuan untuk dapat menemukan fungsi kesalahan dalam aplikasi yang sedang dibangun dan memperbaikinya, selain itu pengujian sistem dilakukan untuk mengetahui apakah sistem tersebut dapat berjalan seperti yang diharapkan.

Pengujian ini dapat dilakukan dengan teks dan kemudian diproses oleh aplikasi apakah aplikasi dapat memberikan hasil yang sesuai. Proses yang akan diuji dalam aplikasi ini adalah simulasi pengiriman pesan dalam bentuk teks menggunakan metode algoritma transposisi kolom dengan menggunakan kunci sehingga keaslian pesan tetap terjaga.

4.2.1 Tampilan Menu Utama

Pada tampilan gambar di bawah ini adalah tampilan menu utama saat aplikasi dijalankan. Dalam form ini pengguna dapat memilih beberapa form dengan fungsinya masing-masing, selain itu ada beberapa tombol yaitu: Mulai, About, Keluar, yang masing-masing juga memiliki fungsi yang berbeda.



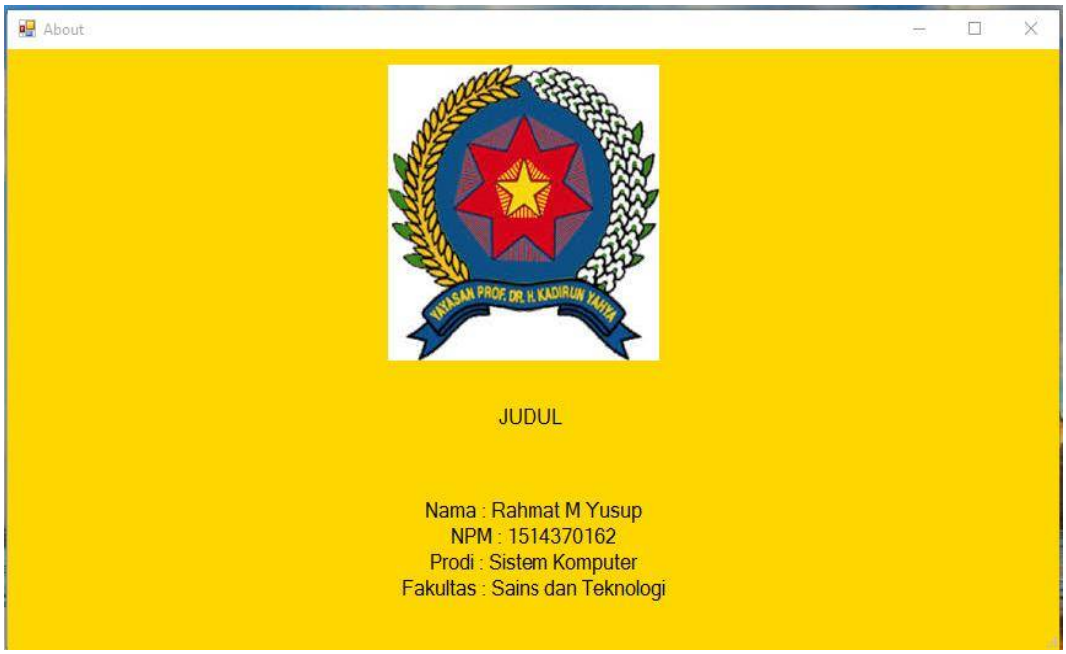
Gambar 4.1 Tampilan Menu Utama

Keterangan :

1. Mulai : Proses Untuk melanjutkan ke form selanjutnya, yaitu form Transposisi Kolom
2. About : Berfungsi untuk menampilkan tentang pembuatan aplikasi
3. Keluar : Berfungsi untuk menghentikan semua program yang berjalan

4.2.2 Tampilan About

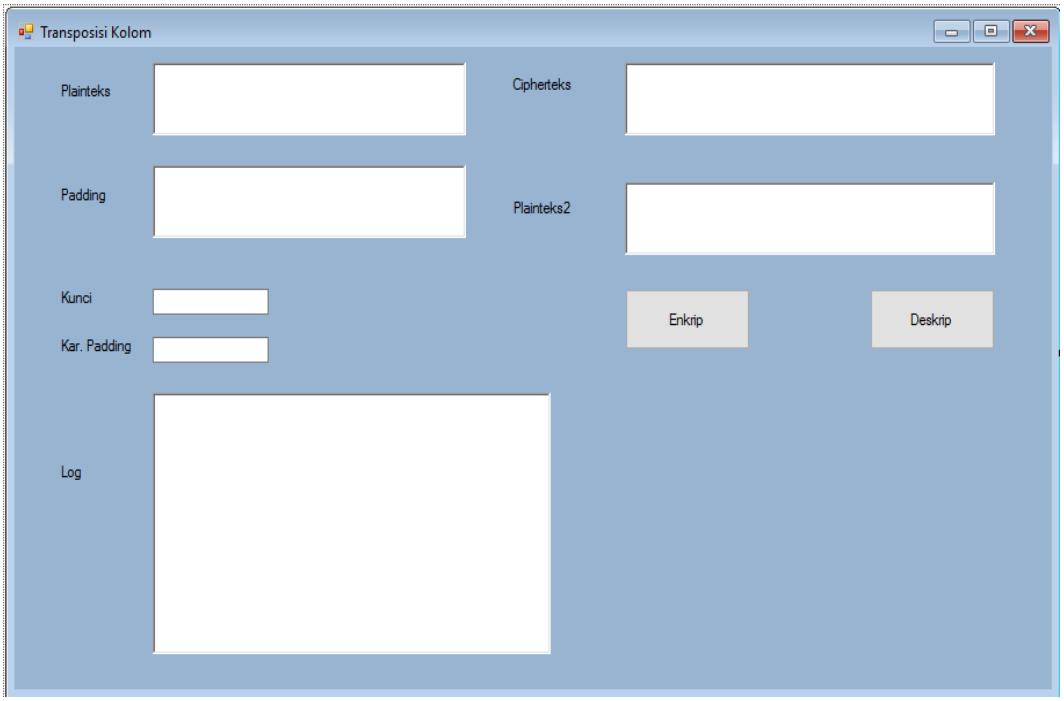
Tampilan About berisi logo, judul, dan biodata penulis



Gambar 4.2 Tampilan About

4.2.3 Tampilan Form Transposisi Kolom

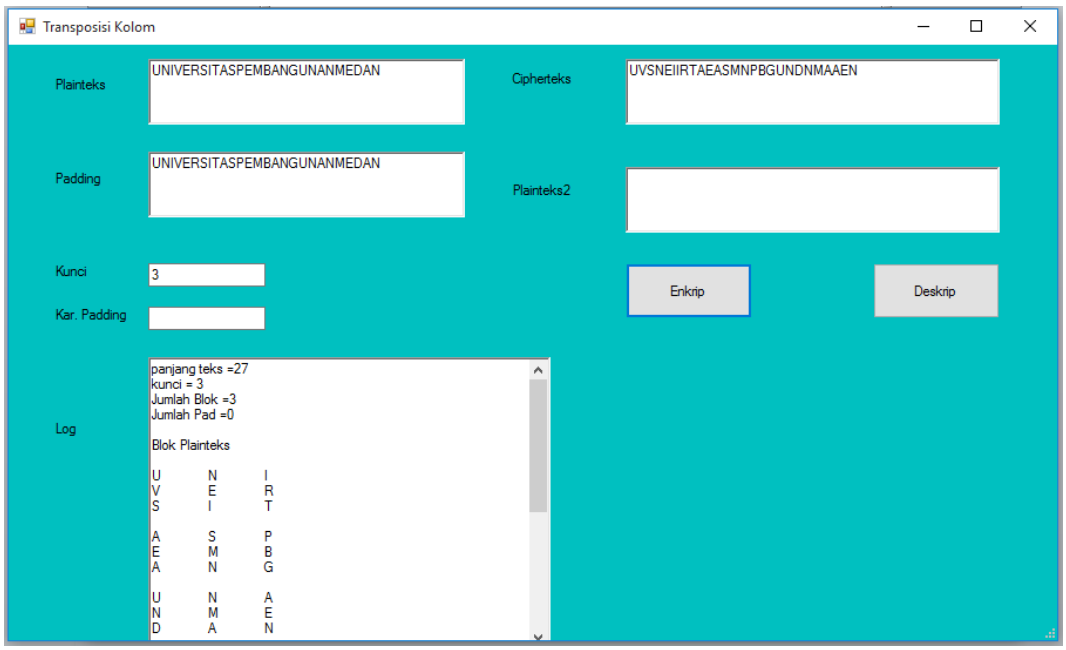
Pada tampilan gambar dibawah merupakan tampilan awal ketika aplikasi dijalankan. Pada form ini terdapat beberapa form dengan fungsi masing-masing, selain itu terdapat beberapa tombol yaitu : dekripsi, enkripsi, permutasi yang masing-masing juga memiliki fungsi yang berbeda.



Gambar 4.3 Tampilan Form Transposisi Kolom

4.2.4 Tampilan Enkripsi

Tampilan enkripsi ini dilakukan dengan memasukan teks pada *plaintext* dan kunci, lalu tekan tombol enkripsi. Pada tahap ini *plaintext* akan dirubah menjadi *ciphertext* menggunakan algoritma Transposisi Kolom. Seperti contoh gambar dibawah ini



Gambar 4.4 Tampilan Enkripsi

4.2.5 Tampilan Deskripsi

Tampilan dekripsi ini dilakukan agar mengetahui maksud dari pesan teks yang telah dirubah yang sulit diartikan, pada tahap ini proses dilakukan setelah melakukan enkripsi.

Gambar 4.5 Tampilan Deskripsi

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan beberapa kesimpulan yang dapat penulis ambil dari penulisan skripsi dan perancangan sistem ini adalah sebagai berikut :

1. Sistem ini dirasakan dapat mengamankan data berupa pesan teks yang dapat menyelesaikan permasalahan keamanan pesan, sehingga si pengirim aman untuk melakukan pengiriman kepada si penerima pesan tersebut.
2. Pada aplikasih tersebut pengamanan pesan teks dengan metode transposisi kolom di buat secara tukar posisi.
3. Integritas pesan akan terjaga yaitu dengan keamanan pesan yang membuat isi pesan tidak terbaca atau diartikan oleh pihak lain.
4. Di dalam aplikasih ini hanya terdapat 1 kunci.

5.2 Saran

Ada pun saran yang ingin disampaikan si penulis ialah sebagai berikut :

1. Sistem aplikasih ini diharapkan dapat dikembangkan dengan metode kriptografi lain nya yang mempunyai spesifikasi keamanan yang lebih tinggi tanpa merusak integritas pesan atau data.
2. Perangkat lunak ini bisa di kembangkan melalui smart phone dan bisa dijalankan di lebih dari satu computer.

DAFTAR PUSTAKA

Andi. (2003), Memahami Model Enskripsi dan Security Data, Yogyakarta.

Andrian, Yudhi, and Purwa Hasan Putra. "Analisis Penambahan Momentum Pada Proses Prediksi Curah Hujan Kota Medan Menggunakan Metode Backpropagation Neural Network." Seminar Nasional Informatika (SNIf). Vol. 1. No. 1. 2017.

Ariyus, Dony. 2006. Kriptografi Keamanan Data dan Komunikasi. Yogyakarta :

Aryza, S., Irwanto, M., Lubis, Z., Siahaan, A. P. U., Rahim, R., & Furqan, M. (2018). A Novelty Design Of Minimization Of Electrical Losses In A Vector Controlled Induction Machine Drive. In IOP Conference Series: Materials Science and Engineering (Vol. 300, No. 1, p. 012067). IOP Publishing.

Ginting, G., Fadlina, M., Siahaan, A. P. U., & Rahim, R. (2017). Technical approach of TOPSIS in decision making. Int. J. Recent Trends Eng. Res, 3(8), 58-64.

Graha Ilmu.

Hafni, Layla, And Rismawati Rismawati. "Analisis Faktor-Faktor Internal Yang Mempengaruhi Nilai Perusahaan Pada Perusahaan Manufaktur Yang Terdaftar Di Bei 2011-2015." Bilancia: Jurnal Ilmiah Akuntansi 1.3 (2017): 371-382.

Hamdi, Muhammad Nurul, Evi Nurjanah, And Latifah Safitri Handayani. "Community Development Based Onibnu Khaldun Thought, Sebuah Interpretasi Program Pemberdayaan Umkm Di Bank Zakat El-Zawa." El Muhasaba: Jurnal Akuntansi (E-Journal) 5.2 (2014): 158-180.

Indra Permana, Aminuddin "Sistem Pakar Mendeteksi Hama Dan Penyakit Tanaman Kelapa Sawit Pada Pt. Moeis Kebun Sipare-Pare Kabupaten Batubara." (2013).

Kholissodin, Imam. 2015. Penggunaan Kriptosistem Kurva Elliptik untuk Enkripsi dan Dekripsi Data. Surabaya : Universitas Airlangga.

Laba, L, G. (2008). Awal Sejarah Kriptografi di Dunia. Yogyakarta, STMIK AMIKOM.

Laksana, B., Prawira, A., & Dave, J. (2007). encryption and decryption using caesa cipher. International Jurnal of Computer Science, 20(5), 10–15.

Lusiana,, Veronica. 2010. Kriptografi Kunci Publik. Tugas Akhir.Universitas Stikubank Semarang

M.Zaki Riyanto, Teknik Pembangkitan Kunci dan Pembangkitan Bilangan Acak Semu. zaki@mail.ugm.ac.id

Mayasari, Nova. "Comparison of Support Vector Machine and Decision Tree in Predicting On-Time Graduation (Case Study: Universitas Pembangunan Panca Budi)." Int. J. Recent Trends Eng. Res 2.12 (2016): 140-151.

Munir, R. (2006). Analisa Algoritma Ciphers Transposition. Jurnal Online Informatika, vol 2 No 2, 102–109.

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.

Permana, A. I., and Z. Tulus. "Combination of One Time Pad Cryptography Algorithm with Generate Random Keys and Vigenere Cipher with EM2B KEY." (2020).

Permana, Aminuddin Indra. "Kombinasi Algoritma Kriptografi One Time Pad dengan Generate Random Keys dan Vigenere Cipher dengan Kunci EM2B." (2019).

Pramana, J. A. K. (2013). Analisis Algoritma One Time Pad Dengan Algoritma Cipher Transposisi Sebagai Pengamanan Pesan Teks. Jurnal & Penelitian Teknik Informatika, Vol 1, No, 58–62.

Prayitno, Arif, N. (2017). analisa dan implementasi kriptografi pesan teks. Jurnal Elektronik Sistem Informasi Dan Komputer, Vol 3 No 1, 1–10.

Puspita, Khairani, and Purwa Hasan Putra. "Penerapan Metode Simple Additive Weighting (SAW) Dalam Menentukan Pendirian Lokasi Gramedia Di Sumatera Utara." Seminar Nasional Teknologi Informasi Dan Multimedia, ISSN. 2015.

Putera, A., Siahaan, U., & Rahim, R. (2016). Dynamic key matrix of hill cipher using genetic algorithm. Int. J. Secur. Its Appl, 10(8), 173-180.

Putra, Fadhlan. 2015. Perbandingan Dan Analisis Performansi Enkripsidekripsi Teks Menggunakan Algoritma Aes Dan Aes Yang Termodifikasi Berbasis Android. Bandung: Universitas Telkom.

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

Sadikin, F. (2012). Penerapan Kriptografi Pada Teks Pesan dengan Menggunakan Metode Transposisi Kolom. Jurnal AL-AZHAR INDONESIA SERI SAINS DAN TEKNOLOGI, Vol. 4, No, 110–115.

Sons, J. W. &. (1996). “Applied Cryptography.” Jerman: Bruce Schneider. Syafari Anjar, 2007,Sekilas Tentang Enkripsi Blowfish, IlmuKomputer.com Weiss, Mark Allen. 2007. Data Structures and Algorithm Analysis in C. Penerbit

Syahputra, Rizki, And Hafni Hafni. "Analisis Kinerja Jaringan Switching Clos Tanpa Buffer." Journal Of Science And Social Research 1.2 (2018): 109-115. Wahana Komputer.

Wahyuni, Sri. "Implementasi Rapidminer Dalam Menganalisa Data Mahasiswa Drop Out." Jurnal Abdi Ilmu 10.2 (2018): 1899-1902.