PERANCANGAN SIMULASI ENKRIPSI DAN DEKRIPSI MENGGUNAKAN ALGORITMA ONE TIME PAD PADA PESAN BERBASIS TEKS

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	:	**YOGI UTOMO**
NPM	:	**1514370846**
PROGRAM STUDI	:	**SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020

ABSTRAK

YOGI UTOMO
Perancangan Simulasi Enkripsi dan Dekripsi Menggunakan Algoritma One TimePadpada Pesan Berbasi Teks
2020

Pertukaran informasi melalui media internet secara bebas membuat pemilik informasi perlu waspada. Bukan hanya informasi umum, namun informasi khusus yang bersifat rahasia. Perlu adanya kendali untuk mengatur keamanan dalam informasi tersebut. Dalam hal ini, peran teknik penyandian data yang dikenal dengan nama kriptografi sangat penting. Kriptografi merupakan teknik untuk menyandikan data melalui proses enkripsi dan dekripsi dengan kunci tertentu sehingga menghasilkan data tersandikan yang tidak diketahui oleh orang lain. Dalam skripsi ini akan digunakan algoritma *one time pad*. Algoritma ini termasuk algoritma kunci simetrik yaitu adanya kesamaan kunci antara enkripsi dan dekripsi. Keunggulan *one time pad* dibanding cipher yang lain yaitu menggunakan kunci yang sama panjang dengan fungsi X-NOR. Kunci acak pada *one time pad* berfungsi untuk menyulitkan kriptanalis dalam menemukan plainteks asli. *One time pad* telah diuji coba melalui aplikasi kriptografi dengan media semua ekstensi file dan membuktikan bahwa algoritma tersebut handal. Hal ini dibuktikan dengan proses dekripsi setiap file yang diproses dapat kembali seperti semula dan tidak mengalami kerusakan. waktu eksekusi untuk semua proses enkripsi dan dekripsi tidak lebih dari 0.25 detik.

Kata Kunci: dekripsi, enkripsi, kriptografi, symmetric, OTP, Vernam

KATA PENGANTAR

Puji dan syukur kita panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya kepada kita semua sehingga penulis dapat menyelesaikan penulisan Skripsi dengan judul "**PERANCANGAN SIMULASI ENKRIPSI DAN DEKRIPSI MENGGUNAKAN ALGORITMA *ONE TIME PAD* PADA PESAN BERBASIS TEKS".**

Skripsi ini merupakan salah satu syarat yang harus dipenuhi untuk menyelesaikan pendidikan S-I pada Program Studi Sistem Komputer Fakultas Sains dan TeknologiUniversitas Pembangunan Panca Budi Medan. Dalam hal ini penulis menyadari masih adanya keterbatasan kemampuan dan pengalaman penulis yang terbatas. Untuk itu penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan dariSkripsi ini.

Selesainya laporan ini tidak terlepas dari bantuan dan bimbingan dari berbagai pihak, untuk itu pada kesempatan ini penulis dengan tulus dan ikhlas menyampaikan ucapan terima kasih sebesar-besarnya kepada :

1. Kepada kedua orang tua tercinta, Bapak Swardi dan Ibu Kasiani yang telah memberikan motivasi, nasihat dan doa sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak Dr. H. Muhammad Isa Indrawan, S.E., M.M., selaku Rektor Universitas Pembangunan Panca Budi Medan.
3. Bapak Hamdani, S.T.,M.T.,selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
4. Bapak Eko Hariyanto S.Kom.,M.Kom.,selaku Ketua Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Medan
5. Bapak Andysah Putera Utama Siahaan S.Kom.,M.Kom., selaku Dosen Pembimbing 1 yang telah membimbing dan mengarahkan penulis dalam menyelesaikan laporan skripsi ini.
6. Ibu Ranti Eka Putri, S.Kom., M.Kom., selaku Dosen Pembimbing II yang telah membimbing dan mengarahkan penulis dalam menyelesaikan Laporan Skripsi ini.
7. Bapak dan Ibu Dosen, selaku Pengajar pada Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi.
8. Seluruh teman penulis dari Fakultas Sains dan Teknologi yang telah banyak membantu dan memotivasi dalam menyelesaikan laporan ini.

Akhir kata penulis mengucapkan terima kasih kepada semua pihak yang telah banyak membantu dan semoga Allah SWT melimpahkan karunianya dalam setiap amal kebaikan kita dan diberikan balasan.

Medan, 16 Februari 2020
Penulis

Yogi Utomo
1514370846

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin pesat telah mempengaruhi semua aspek kehidupan dan memberikan banyak sekali keuntungan. Selain itu juga aspek-aspek dari sisi negatif dari kemajuan sistem informasi tersebut. Pada dasarnya melakukan pengiriman data tanpa melakukan pengamanan pada konten dari data yang dikirim, dapat menyebabkan adanya penyadapan pada jalur pengirimannya.

Banyak pengguna internet yang saat ini khawatir terhadap pengiriman informasi diketahui oleh pihak lain. Untuk persoalan ini sebuah metode untuk keamanan informasi yang dikenal dengan kriptografi. Dalam kriptografi, dikenal dengan dua konsep yakni enkripsi dan dekripsi. Untuk itulah peranan teknologi keamanan informasi benar-benar dibutuhkan, salah satu cara yang bisa digunakan adalah menyandikan informasi atau data rahasia yang akan dikirim, sehingga pihak yang tidak berkepentingan tidak dapat membaca informasi tersebut.

Salah satu teknik untuk pengamanan data adalah dengan menggunakan algoritma penyandian data. Algoritma penyandian data saat ini semakin banyak jumlahnya, sejalan dengan berkembangnya ilmu yang mempelajari penyandian data tersebut (Setyawan Hari E. 2018). Algoritma yang digunakan dalam tugas akhir ini yaitu algoritma *one time pad*. Algoritma *onetime pad* merupakan

algoritma sederhana yang saat ini dinyatakan aman karena masih belum ada serangan yang benar-benar mematahkan algoritma ini,

Berdasarkan latar belakang tersebut, penulis termotivasi untuk merancang tugas akhir dengan judul"**PERANCANGAN SIMULASI ENKRIPSI DAN DEKRIPSI MENGGUNAKAN ALGORITMA *ONE TIME PAD* PADA PESAN BERBASIS TEKS".**

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, penulis menyimpulkan beberapa rumusan masalah, diantaranya sebagai berikut:

1. Bagaimana merancang simulasi enkripsi dan dekripsi menggunakan algoritma *one time pad* pada pesan berbasis teks?
2. Bagaimana merancang kunci OTP untuk proses enkripsi dan dekripsi?
3. Bagaimana merancang sistem kerja OTP dengan algoritma VernamCipher?

1.3 Batasan Masalah

Untuk mendapatkan pembahasan yang maksimal dan mudah dipahami serta untuk menghindari pembahasan yang terlalu meluas, maka batasan masalah yang dibahas pada skripsi ini adalah:

1. Aplikasi ini hanya untuk pengelolaan data teks menjadi pesan rahasia berbentuk kriptografi menggunakan algoritma *one time pad*.
2. Karakter yang digunakan ASCII.
3. Bahasa yang di gunakan adalah *Visual basic*.
4. Algoritma one time pad yang digunakan adalah VernamCipher.

1.4 Tujuan Penelitian

Berikut ini beberapa tujuan penelitian yang akan dibahas dalam penelitian ini yaitu sebagai berikut:

1. Untuk merancang simulasi enkripsi dan dekripsi menggunakan algoritma one time pad pada pesan berbasis teks.
2. Untuk merancang kunci OTP untuk proses enkripsi dan dekripsi.
3. Untuk merancang sistem kerja OTP dengan algoritma VernamCipher.

1.5 ManfaatPenelitian

Berikut ini beberapa manfaat penelitian yang akan dibahas dalam penelitian ini yaitu sebagai berikut:

1. Bagi penulis yaitu untuk menambah pengalaman dan ilmu pengetahuan serta menerapkan teori-teori yang penulis peroleh pada saat proses perkuliahan.
2. Bagi Universitas Panca Budi adalah sebagai contoh penelitian bagi mahasiswa/i untuk generasi yang akan datang jika ingin melakukan penelitian tentang perancangan aplikasi yang lebih baik.
3. Untuk menjamin keamanan sebuah informasi yang kita miliki.
4. Dengan pembuatan aplikasi ini diharapkan penggunaan menggunakan algoritma *one time pad* dalam berbagai bidang dapat diterapkan dengan baik.

BAB II

LANDASAN TEORI

2.1 Kriptografi

Kriptografi merupakan sebuah ilmu yang digunakan untuk penyandian data. Kriptografi telah dikenal dan dipakai cukup lama sejak kurang lebih tahun 1900 sebelum masehi pada prasasti-prasasti kuburan. Ada tiga istilah yang berkaitan dengan proteksi data yaitu kriptografi, kriptologi, dan kriptanalisis. Arti ketiganya kurang lebih sama. Secara teknis, kriptologi adalah ilmu yang mempelajari tentang komunikasi pada jalur yang tidak aman beserta masalah-masalah yang berhubungan dengan itu(Rohmanu, 2017).

Kriptografi berasal dari kata "*Crypto*" yang berarti rahasia dan "*graphy*" yang berarti tulisan. Jadi, dapat dikatakan bahwa kriptografi adalah tulisan yang tersembunyi. Dengan adanya tulisan yang tersembunyi ini, orang-orang tidak mengetahui bagaimana tulisan tersebut disembunyikan dan tidak mengetahui bagaimana cara membaca maupun menerjemahkan tulisan tersebut.

Kriptografi berbasis pada algoritma pengkodean data informasi yang mendukung kebutuhan dari dua aspek keamanan informasi, yaitu *secrecy* (perlindungan terhadap kerahasiaan data informasi) dan *authenticity* (perlindungan terhadap pemalsuan dan pengubahan informasi yang tidak diinginkan). Kriptografi menjadi dasar bagi keamanan komputer dan jaringan karena merupakan sarana bagi distribusi data dan informasi. Sehingga data dan informasi tersebut harus diamankan agar hanya orang-orang yang berhak

mengaksesnya yang dapat mengetahui maupun menggunakan data tersebut. Salah satu cara yang paling banyak digunakan dalam mengamankan data adalah dengan kriptografi. Data-data tersebut diamankan dengan sedemikian rupa oleh pengirim sehingga orang lain tidak dapat mengenali data tersebut(Muhammad, 2014).

Dalam kriptografi terdapat aspek-aspek keamanan data yaitu:

1. *Confidentiality*, merupakan usaha untuk kerahasiaan data. Serangan dalam aspek ini antara lain dilakukan dengan penyadapan, misalnya *sniffer* atau *logger*.
2. *Integrity*, memastikan bahwa informasi yang dikirim tidak mengalami modifikasi oleh pihak yang tidak berhak. Serangan dapat berupa pengubahan data oleh orang yang tidak berhak.
3. *Availability*, informasi harus tersedia ketika dibutuhkan. Serangan dapat berupa menghilangkan atau menghapus data.
4. *Authentication*, meyakinkan keaslian data, sumber data, orang yang mengakses data, dan server yang digunakan.
5. *Access Control*, aspek ini berhubungan dengan mekanisme pengaturan akses ke informasi, untuk mengatur siapa yang boleh melakukan apa.

Dalam kriptografi sering ditemukan istilah penting untuk kita ketahui, yaitu:

1. Pesan (*massage*), adalah data atau informasi yang dapat dibaca atau dimengerti maknanya.
2. Pengirim (*sender*), adalah entitas yang melakukan pengiriman pesan kepada entitas lain.

3. Kunci (*key*), adalah aturan atau fungsi yang dilakukan untuk melakukan prosen enkripsi dan dekripsi pada plainteks dan cipherteks.
4. Enkripsi adalah mekanisme yang dilakukan untuk merubah plainteks menjadi cipherteks.
5. Dekripsi adalah mekanisme yang dilakukan untuk merubah cipherteks menjadi plainteks.
6. Penerima (*recipient*), adalah entitas yang penerima berhak menerima pesan dari pengirim.

2.1.1 Tujuan kriptografi

Seperti juga perkembangan ilmu kriptografi, kriptografi bertujuan untuk memberikan layanan keamanan yaitu:

1. Kerahasiaan (*Confidentiality*), informasi dirahasiakan dari semua pihak yang tidak berwenang.
2. Keutuhan Data (*Integrity*)
3. Pesan tidak berubah dalam proses pengiriman hingga pesan diterima oleh si penerima.
4. Autentikasi (*Message Autentication*), kepastian terhadap identitas yang terlibat dan keaslian sumber data.
5. Nirpenyangkalan (*Nonrepudiation*)
6. Setiap entitas yang berkomunikasi tidak dapat menolak atau menyangkal atas data yang telah dikirim atau diterima.

2.1.2. Kriptografi Kunci Simetris dan Asimetris

Berdasarkan kunci yang digunakan, kriptografi dapat dibedakan menjadi dua bagian, yaitu:

1. Algoritma Simetris (*Symmetric Algorithms*)

   Algoritma simetris disebut juga sebagai algorima konvensional. Algoritma simetris menggunakan suatu kunci yang sama untuk proses enkripsi dan dekripsi. Penggunaan kunci yang sama menjadikan kekuatan algoritma simetris menjadi sangat bergantung pada satu kunci yang digunakan.selain itu proses dekripsi pada algoritma simetris juga menjadi kebalikan dari proses enkripsi, apabila pengirim kunci dapat dilakukan secara aman.

2. Algoritma Asimetris (*Asymetric Algorithms*)

   Kunci publik (*public key*) yang merupakan nama lain dari algoritma asimetris. Enkripsi dan dekripsi pada algoritma asimetris menggunakan kunci yang berbeda.Kunci dalam algoritma asimetris dibagi menjadi dua yaitu kunci umum (*public key*) dan kunci pribadi (*private key*).Pada kunci umum, kunci tersebut dapat diketahui oleh semua orang (*public*).Sedangkan pada kunci pribadi hanya dapat diketahui oleh orang yang bersangkutan. Pengetahuan kunci umum memungkinkan seseorang untuk dapat mengenkripsi suatu pesan tetapi tidak dapat mendekripsi kan pesan tersebut. Hanya orang yang memiliki kunci pribadi yang dapat mendekripsi kan pesan yang telah dienkripsi sehingga kedua kunci tersebut (kunci umum dan kunci pribadi) harus saling berhubungan satu dengan yang lainnya.

2.2 Algoritma *One Time Pad* (OTP)

Algoritma *One Time Pad* (OTP) adalah *stream cipher* yang melakukan enkripsi dan dekripsi satu karakter setiap kali. Algoritma ini merupakan perbaikan dari *Vernam cipher* untuk menghasilkan keamanan yang sempurna. Cipher ini termasuk ke dalam kelompok algoritma kriptografi simetri. *One Time Pad* (pad = kertas bloknot) berisi barisan karakter-karakter kunci yang dibangkitkan secara acak. Aslinya, satu buah *One Time Pad* adalah sebuah pita (*tape*) yang berisi barisan karakter-karakter kunci. Satu pad hanya digunakan sekali (*one time*) saja untuk mengenkripsi pesan, setelah itu pad yang telah digunakan dihancurkan supaya tidak dipakai kembali untuk mengenkripsi pesan yang lain(Khairina & Harahap, 2017). Penggunaan *One Time Pad* berikut yang digunakan penulis menggunakan tabel ASCII, berikut merupakan table ASCII.

Tabel 2.1 Contoh karakter ASCII

KARAKTER	ASCII CODE
A	65
B	66
C	67
D	68
E	69
F	70
G	71
H	72

Sumber: (Setiawan, Raharjo, & Susanto, 2016)

Tabel 2.2 Contoh karakter ASCII (Lanjutan)

KARAKTER	ASCII CODE
I	73
J	74
K	75
L	76
L	76
M	77
N	78
O	79
P	80
Q	81
R	82
S	83
T	84
U	85
V	86
W	87
X	88
Y	89
Z	90

Sumber: (Setiawan, Raharjo, & Susanto, 2016)

Rumus dari enkripsi *one time pad* yaitu:

$$Ci = (Pi + Ki - 2 \times 64) \bmod 26 + 64$$

Rumus dari dekripsi *one time pad* yaitu:

$$Pi = (Ci – Ki + 26) \bmod 26 + 64$$

Keterangan :

Ci = Cipherteks

Pi = Plainteks

Ki = Kunci

2.3 Vernam Cipher

*Vernam cipher*adalah jenis algoritma enkripsi simetri.*Vernam Cipher*dapat dibuat sangat cepat sekali, jauhlebih cepat dibandingkan dengan algoritma *blockcipher*yang manapun. Algoritma *block cipher* secara umumdigunakan untuk unit *plaintext* yang besar sedangkan stream cipher digunakan untuk blok data yang lebih kecil,biasanya ukuran bit. Proses enkripsi terhadap *plaintext* tertentu dengan algoritma *block cipher* akanmenghasilkan*ciphertext*yang sama jika kunci yang sama digunakan. Dengan*stream cipher*, transformasi dari unitplaintext yang lebih kecil ini berbeda antara satu dengan lainnya, tergantung pada kapan unit tersebut ditemukanselama proses enkripsi(Siahaan, 2016).

SatuV*ernamCipher*menghasilkanapa yang disebut suatu*key stream*(suatu barisan bit yang digunakan sebagaikunci). Proses enkripsi dicapai dengan menggabungkan keystream dengan *plaintext* biasanya dengan operasi bit*wise XOR*. Pembentukan key stream dapat dibuat independen terhadap ,menghasilkan *synchronous stream cipher*, atau dapat dibuat tergantung pada data dan enkripsinya, dalam hal mana*stream cipher* disebut sebagai *self-synchronizing*. Kebanyakan bentuk *stream stream cipher*(Siahaan, 2017).

Konsentrasi dalam *stream ciphers* pada umumnya berkaitan dengan sifat sifat teoritis yang menarik dari *one-time pad*. Suatu *one-time pad*, kadang-kadang disebut*Vernam cipher*, menggunakan sebuah string dari bit yangdihasilkan murni secararandom. *Key stream*memiliki panjang sama dengan pesan*plaintex*t;*string random* digabungkan dengan menggunakanbit*wiseXOR*dengan*plaintext*untuk menghasilkan*ciphertex*t.Karena*keystream*seluruhnya adalah random, walaupun dengan sumber daya komputasi tak terbatas seseoranghanya dapat menduga *plaintext*jika melihat*ciphertext.* Metodecipherseperti ini disebut memberikan kerahasiaanyang sempurna (*perfectsecrecy*). Algoritma*VernamCipher*yang umum digunakan adalah RC4. Satu hal yang menarikbahwamode operasi tertentu dari suatu*block cipher*dapat mentransformasikan secara efektif hasil operasi tersebutke dalam satu*keystream*generator dan dalam hal ini, *block cipher*apa saja dapat digunakan sebagai suatu*stream cipher*; seperti dalam DES, CFB atau OFB. Akan tetapi,*vernam ciphers*dengan desain khusus biasanya jauh lebih cepat(Sari, Rachmawanto, Utomo, & Sani, 2016).

Cipherteks diperoleh dengan melakukan penjumlahanmodulo2 satu bit *plainteks*dengan satu bit kunci:

C1 = (P1+K1) mod 26

Dimana P1 adalah bit *plainteks*, K1 adalah bit kunci, dan C1 adalah bit *cipherteks*,*plainteks*diperoleh dengan melakukan penjumlahan modulo 2 bit*cipherteks*dengan satu bit kunci.

1. Proses Enkripsi dan Dekripsi Perhitungan di atas merupakan salah satu dasar dalam penerapan algoritma *Vernam* dalam kriptografi

Perhitungan di atas merupakan salah satu dasar dalam penerapan algoritma *Vernam* dalam kriptografi, yaitu suatu string yang diterjemahkan ke dalam biner dapat dienkripsikan dengan suatu kunci tertentu dan dapat pula diperoleh kembali dari pesan sandi dengan menggunakan operator XOR pada kunci yang sama.Dalam algoritma ini, terdapat beberapa langkah untuk proses enkripsi dan dekripsi. Misalnya kita akan mengenkripsi plainteks "*Vernam*" dengan kunci "*Cipher*". Maka langkah-langkahnya adalah :

a. Karakter -karakter yang terdapat pada plainteksdan kunci merupakan karakter ASCII. Maka, ubah plainteks dan kunci menjadi bilangan biner:

   ASCIIBinerVernam: 01010110 01100101 01110010 01101110 01100001 01101101

   Cipher: 01000011 01101000 01101001 01110000 01100101 01110010

b. Lalu, kedua bilangan biner itu kita XOR-kan menurut persamaan a:

   Plainteks: 01010110 01100101 01110010 01101110 01100001 01101101

   Kunci :01000011 01101001 01110000 01101000 01100101 01110010

   Cipherteks:00010101 00001100 00000010 00000110 00000100 00011111

Hasil dari XOR tersebut adalah: “00010101 00001100 00000010 00000110 00000100 00011111”. Rangkaian bilangan bit ini merupakan bit cipherteks dalam bentuk biner. Untuk mengetahui nilai ASCII dari cipherteks tersebut, maka kita perlu konversikan rangkaian biner tersebut ke bilangan ASCII.

Dengan berdasarkan pada tabel tersebut, maka dapat kita lihat bahwa rangkaian bit dari cipherteks tersebut mempunyai nilai ASCII.

§ ♀ ☻ ♠ ♦ ▼

Proses dekripsi dalam algoritma *Vernam Cipher* merupakan kebalikan dari proses enkripsi. Cipherteks dari hasil enkripsi di-XOR-kan dengan kunci yang sama.Misalnya dengan mengambil contoh sebelumnya, dimana cipherteks : “§ ♀ ☻ ♠ ♦ ▼” dan kunci : “Cipher”. Maka langkah pendekripsiannya adalah sebagai berikut:

a. Ubah karakter ASCII dari cipherteks dan kunci ke dalam rangkaian biner:

   ASCIIBiner

   § ♀ ☻ ♠ ♦ ▼ :00010101 00001100 00000010 00000110

   Cipher :01000011 01101000 01101001 01110000 01100101

b. Lalu, kedua rangkaian biner itu kita XOR-kan dengan berdasar pada persamaanb:

   Cipherteks : 00010101 00001100 00000010 00000110 00000100 00011111

Kunci : 01000011 01101001 01110000 01101000 01100101 01110010

Plainteks : 01010110 01100101 01110010 01101110 01100001 01101101

c. Maka didapat rangkaian bit plainteks “01010110 01100101 01110010 01101110 01100001 01101101”. maka kita dapat mengubah rangkaian bit menjadi bilangan ASCII,yaitu menjadi : “*Vernam*”. Terbukti plainteks pada hasil dekripsi adalah sama dengan plainteks pada proses enkripsi.

2.4 Visual Basic

Visual Basic adalah bahasa pemrograman yang digunakan untuk membuat aplikasi Windows yang berbasis grafis. Visual Basic merupakan *event driventprogramming* (pemrograman terkendali kejadian) artinya program menunggu sampai adanya respon dari pemakai berupa event/kejadian tertentu (tombol diklik, menu dipilih, dan lain-lain). Selain itu program ini juga bisa diaplikasikan dengan program yang lain seperti *Microsoft access, Macromedia flash, Microsoft word, Power point*, dan aplikasi-aplikasi yang lain(Wibowo, 2019).

1. Aplikasi-aplikasi dalam visual basic sebagai berikut:
   a. *Form Form* adalah windows atau jendela di mana akan dibuat user interface/tampilan. Pada bagian ini biasanya berisi tentang field-field yang dibuat sebagai tempat pemasukan data.

Form adalah windows atau jendela di mana akan dibuat user interface/tampilan. Pada bagian ini biasanya berisi tentang field-field yang dibuat sebagai tempat pemasukan data.

b. Kontrol (*control*)

Kontrol adalah tampilan berbasis grafis yang dimasukkan pada form untuk membuat interaksi dngan pemakai. Contoh: text box, label, command dan lainnya.

c. Properti (*properties*)

Properti adalah nilai/karakteristik yang dimiliki oleh sebuah objek Visual Basic. Contoh: name, size, caption, text, dan lain-lain.

d. Metode (*Methods*)

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

e. Prosedur Kejadian (*Event Prosedures*)

Prosedur Kejadian adalah kode yang berhubungan dengan suatu objek. Kod ini akan dieksekusi ketika ada respon dari pemakai berupa event tertentu.

f. Prosedur Umum

Prosdur umum merupakan kode yang tak berhubungan dengan suatu objek.

g. Modul

Modul adalah kumpulan dari prosedur umum dan definisi konstanta yang digunakan oleh aplikasi.

2. Kemampuan *Visual Basic*.

Adapun kemampuan atau manfaat dari *visual basic*yaitu :

a. Untuk membuat program aplikasi berbasis *website*

b. Untuk membuat *ActiveX*, aplikasi internet dan sebagainya.

c. Menguji program (*debugging*) dan menghasilkan program akhir berakhiran *EXE* yang bersifat *executable* atau dapat langsung dijalankan.

3. Tampilan Layar Visual Basic sebagai berikut:

a. Main *Windows.*

Main *Windows* terdiri dari *title bar* (baris judul), menu bar, dan toolbar. Baris judul berisi nama proyek, mode operasi *Visual Basic* sekarang, dan form yang aktif. Menu bar merupakan menu drop-down di mana anda dapat mengontrol oeprasi dalam lingkungan *Visual Basic*. Toolbar berisi kumpulan gambar yang mewakili perintah yang ada di menu. Jendela utama juga menampilkan lokasi dari form yang 17 aktif relatif terhadap sudut kiri atas layar (satuan ukurannya twips), juga lebar dan panjang dari form yang aktif.

b. From *Windows*

Form Windows adalah pusat dari pengembangan aplikasi Visual Basic. Di sini tempat untuk “menggambar” aplikasinya.

c. Project *Windows*Berguna untuk menampilkan daftar form dan modul proyek. Proyek merupakan kumpulan dari modul form, modul class, modul standar, dan file. Sumber yang membentuk suatu aplikasi.

d. *Toolbox*

Toolbox adalah kumpulan dari objek yang digunakan untuk membuat user interface serta kontrol bagi pemrogram aplikasi.

e. *Properties Windows*

Berisi daftar struktur setting properti yang digunakan pada sebuah objek terpilih. Kotak drop-down pada bagian atas jendela berisi daftar semua objek pada form yang aktif. Ada dua tab tampilan: Alphabetic (urut abjad) dan Categorized (urut berdasar kelompok). Dibawah bagian kotak terdapat properti dari objek terpilih.

f. *Form LayoutWindows*

Berfungsi menampilkan posisi form relatif terhadap layar monitor.

2.5 Flowchart

Flowchart adalah bagan yang menggambarkan urutan instruksi proses dan hubungan satu proses dengan proses lainnya menggunakan simbol-simbol tertentu. Bagan alir digunakan sebagai alat bantu komunikasi dan dokumentasi.

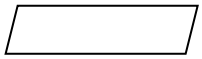
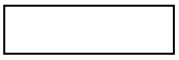
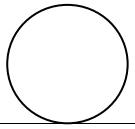
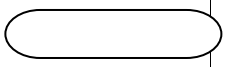
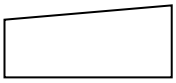
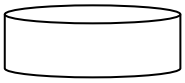
Sistem *flowchart* merupakan bagan yang menunjukkan pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada di dalam sistem dan menunjukkan apa saja yang dikerjakan pada sistem(Jogiyanto, 2016).

Form flowchart merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.

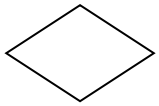
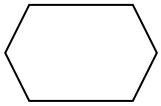
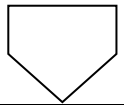
Program *flowchart* adalah suatu bagan yang menggambarkan urutan proses secara mendetail dan hubungan antara proses yang satu dengan proses lainya dalam suatu program.

Beberapa simbol yang digunakan dalam menggambarkan suatu *flowchart* bisa dilihat pada tabel 2.2 berikut:

Tabel 2.3 Simbol Flow chart

SIMBOL	NAMA	FUNGSI
	*Input/Output*	Merepresentasikan *input* data atau *output* data yang diproses atau informasi
	Proses	Mempresentasikan operasi
	Penghubung (*on page connector*)	Keluar ke atau masuk dari bagian lain *flowchat* khususnya halaman yang sama.
	Terminal *point*	Awal/akhir *flowchat*
	*Punched card*	*Input/output* yang menggunakan kartu berlubang
	Dokumen	*I/O* dalam format yang dicetak
	Manual *input*	*Input* yang dimasukkan secara manual dari *keyboard*
	*Database*	Menyimpanke *database*

Tabel 2.4 Simbol Flow chart (Lanjutan)

SIMBOL	NAMA	FUNGSI
	Keputusan	Keputusan dalam program
	*Preparation*	Pemberian harga awa L
	Penghubung (*off page connector*)	Keluar kea tau masuk dari bagian lain *flowchat* halaman yang berbeda.
	Anak panah	Merepresentasikan alur kerja

Sumber: (Nurgoho, 2019)

2.6 Unified Modeling Language

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangunperangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (Object-Oriented). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-

kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software.

Dari berbagai penjelasan yang terdapat di dokumen dan buku-buku *UML*, Sebenarnya konsep dasar *UML* bisa dirangkum dalam table 2.3.

Tabel 2.5 Konsep Dasar UML

Major Area	***View***	***Diagrams***	***Main Concepts***
Structural	*Staticview*	*Class diagram*	*Class, association, generalization, dependency, realization*
	Use caseview	*Use case diagram*	*Use case, actor, association, extend, include, usecasegeneralization.*
	Implementationview	*Component diagram*	*Component, interface, dependency,*
Structural	*Deployment view*	*Deployment diagram*	*Node, component, dependency, location*
Dynamis	*State machine view*	*Statechart diagram*	*State, event, transition, action*
Dynamis	*Activity view*	*Activity diagram*	*State, activity, completion, transition, fork, join*
Dynamis	*Interaction view*	*Collaboration diagram*	*Collaboration, interaction, collaborationrole, message*
Model management	*Model management view*	*Class diagram*	*Package, subsystem, model*

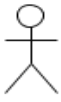
Sumber: (Kurniawan, 2018)

Abstraksi konsep dasar *UML* yang terjadi dari *structural classification*, *dynamic behaviour*, dan model managemen, bisa kita pahami dengan mudah apabila kita melihat tabel diatas dari diagram.*Mainconcepts*bisa kita pandang sebagai *term* yang akan muncul pada saat kita membuat diagram. Dan *view* adalah kategori dari diagram tersebut.Diagram Unified Modelling Language (UML)antara lain sebagai berikut:

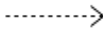
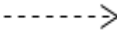
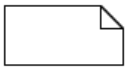
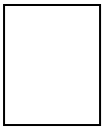
2.6.1 Use Case Diagram

Use case menggambarkan external view dari sistem yang akan kita buat modelnya. Model use case dapat dijabarkan dalam diagram use case, tetapi perlu diingat, diagram tidak indetik dengan model karena model lebih luas dari diagram. (Pooley, 2003:15). Use case harus mampu mengambarkan urutan aktor yang menghasilkan nilai terukur.Aktor merupakan 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 aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.*Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Berikut adalah simbol-simbol yang ada pada diagram *use case*:

Tabel 2.6 Simbol Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.

Tabel 2.7 Simbol Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
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*.
7		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
8		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
9		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
10		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

2.6.2 Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*. Berikut adalah simbol-simbol yang ada pada diagram aktivitas.

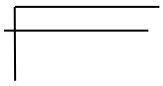
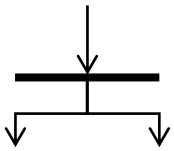
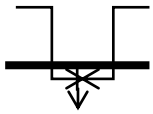
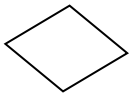
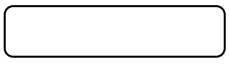
Activity diagram menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram akivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh sistem.

Activity diagram merupakan *stage* diagram khusus, dimana sebagian besar *stage* adalah *action* dan sebagian besar transisi di *trigger* oleh selesainya*stage* sebelum (*unternal processing*). Oleh karena itu activity diagram tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama dengan state, standar *UML* menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas.*Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Tabel 2.8 Simbol Activity Diagram

GAMBAR	KETERANGAN
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	*Endpoint*, akhir aktivitas
	*Swimlane*, pembagian *activity*diagram untuk menunjukkan siapa melakukan apa.
	*Fork*/percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paraelatau untuk menggabungkan dua kegiatan paralel menjadi satu.
	*Join* (penggabungan) atau *rake*, digunakan untuk menunjukkan adanya dekomposisi.
	*Decisionpoint*, menggambarka pilihan untukpengambilan keputusan,*true* atau
	*Aktivities*, menggambarkan suatu proses atau kegiatan bisnis.

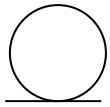
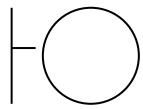
Sumber: (Kurniawan, 2018)

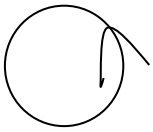
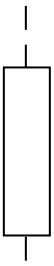
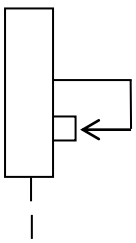
2.6.3 Sequence Diagram

Sequence diagram merupakan sequence diagram yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesanyang dikirimkan dan terima antar objek. *Sequence* diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* yang menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger*aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan.

Masing-masing objek termasuk *actor*, memiliki *lifeline vertikal*.*Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya pada *fase* design berikutnya, *message*akan dipetakan menjadi operasi/metoda dari *class*.*Activationbar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Simbol-simbol yang digunakan dalam *sequence* diagram, yaitu:

Tabel 2.9 Simbol Sequence Diagram

GAMBAR	KETERANGAN
	*Entityclass*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundaryclass*, berisi kumpulan kelas yang menjadi *interfaces* atau interaksi antara satu atau lebih *actor*dengan sistem, seperti tampilan *formentry* dan *form* cetak.

GAMBAR	KETERANGAN
	*Control class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas,contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagi objek.
	*Message*, simbol mengirim pesan antar *class*.
	*Activation*, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat *activation*.
	*Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

Sumber: (Nurgoho, 2019)

2.7 Antarmuka Visual Basic.NET

Antarmuka mendefinisikan properti, metode, dan peristiwa yang dapat diimplementasikan oleh kelas. Antarmuka memungkinkan untuk mendefinisikan fitur sebagai kelompok kecil dari properti, metode dan acara yang berkaitan erat, ini mengurangi masalah kompatibilitas karena dapat mengembangkan implementasi yang ditingkatkan untuk antarmuka tanpa membahayakan kode yang ada. Fitur baru dapat ditambahkan dengan mengembangkan antarmuka dan implementasi tambahan. Ada beberapa alasan lain mengapa antarmuka Visual Basic lebih baik adalah sebagai berikut:

1. Visual Basic lebih cocok untuk situasi dimana aplikasi membutuhkan banyak jenis objek yang mungkin tidak terkait untuk menyediakan fungsionalitas tertentu.
2. Antarmuka Visual Basic lebih simpel daripada yang lainnya, karena Visual Basic dapat menentukan implementasi tunggal yang dapat mengimplementasikan banyak antarmuka.
3. Antarmuka Visual Basic lebih baik dari bahasa pemrograman yang lain.

Antar muka visual basic sangat berguna ketika menggunakan warisan kelas sebagai contoh, struktur tidak dapat mewarisi dari kelas, tetapi mereka dapat mengimplementasikan.

BAB III

METODE PENELITIAN

3.1 Analisa Pemasalahan

Dalam pembahasan ini adalah bagaimana cara mengatasi penyandian data. Analisa yang dimaksud adalah banyaknya pelaku peretasan pesan ataupun keamanan lainnya yang tidak bertanggung jawab. Saat ini banyak keamanan pada pesan yang sangat tidak diperhatikan keamanannya sehingga mengakibatkan banyak pesan dapat dengan mudah ditembus oleh pihak yang tidak bertanggung jawab. Dalam hal ini sangat diperlukan keamanan pesan tersebut agar tidak muda ditembus oleh pihak lain. Untuk menyelesaikan masalah tersebut, maka penulis membuat suatu aplikasi kriptografi dengan memnggunakan algoritma *one time pad*.

3.2 Analisis Sistem

Analisa Sistem merupakan penjabaran dari sistem informasi yang utuh kedalam berbagai macam bagian komponennya dengan maksud agar kita dapat mengidenfikasi atau mengevaluasi berbagai macam masalah maupun hambatan yang akan timbul pada sistem.

1. Analisis Perangkat Keras (*Hardware*)

   Analisis perangkat keras merupakan suatu proses yang kegunaannya untuk mendapatkan sebuah informasi, model spesifikasi mengenai perangkat keras yang diinginkan.

2. Analisis Perangkat Lunak (*Software*)

   Analisis perangkat lunak merupakan suatu proses yang kegunaannya untuk mendapat informasi, model spesifikasi perangkat lunak (*software*) yang diinginkan yang akan digunakan untuk membangun sebuah aplikasi. Perangkat lunak adalah data-data yang terdapat pada sebuah komputer.

3. Analisis Pengguna (*user*)

   Analisis *user*yang dimaksud disini yaitu aplikasi dari *algoritma one time pad* untuk dipergunakan oleh publik untuk melakukan pengamanan pesan.

Berikut ini adalah gambaran umum sebuah contoh proses enkripsi dan dekripsi yang akan di buat dalam contoh soal.

Contohnya:

Saya memiliki sebuah plainteks yaitu “YOGI” dan memiliki sebuah kunci yaitu “IGOY” perlu di ketahui bahwa panjang kunci harus sama dengan plainteks yang kita buat.

Terlebih dahulu kita harus mendapatkan kode ASCII dari plainteks yang kita buat lalu kemudian kita ubah kedalam bentuk biner.

Karakter	ASCII	Biner
Y	89	10110010

Karakter	ASCII	Biner
O	79	10011110
G	71	10001110
I	73	10010010

Berikutnya hal yang sama harus dilakukan untuk mengubah kedalam bilangan biner pada kunci yang di pilih.

Karakter	ASCII	Biner
I	73	10010010
G	71	10011110
O	79	10011110
Y	89	10110010

Setelah itu masing- masing karakter di Xnor-kan dengan kunci untuk menghasilkan cipherteks dari plainteks tersebut.

Plainteks:

Y =10110010 O =10011110 G =10001110 I =10010010

Kunci:

I =10010010 G =10001110 O =10011110 Y =10110010

Cipherteks:

Y dengan I = 00100000 O dengan G =00010000 G dengan O=00001000 I dengan Y=00000100

Untuk proses dekripsi pesan ini, sama juga dengan melakukan operasi yang sama yaitu dengan Xnor-kan antara cipherteks dengan kunci tersebut untuk menghasilkan plainteks

Cipherteks : 00100000 00010000 00001000 00000100

Kunci : 10010010 10011110 10011110 10110010

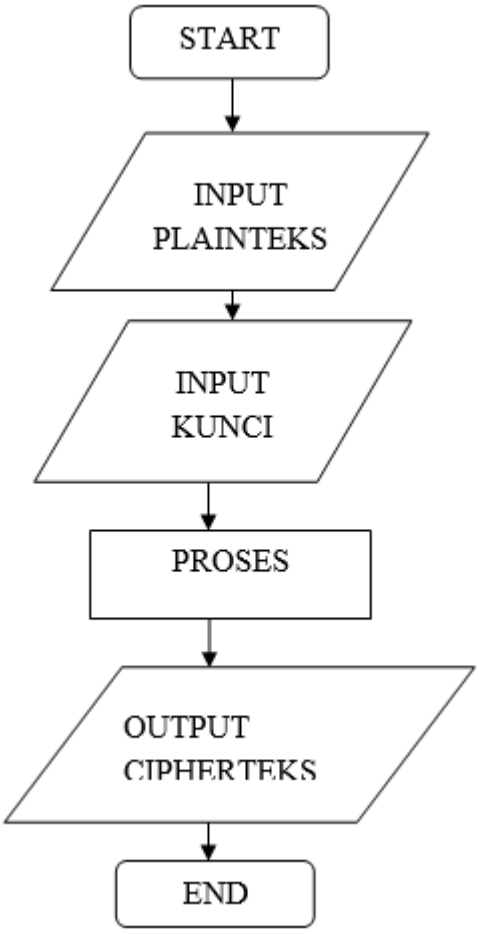
Plainteks : 10110010 10011110 10001110 10010010

Terbukti plainteks pada hasil dekripsi adalah sama dengan plainteks pada proses enkripsi.

3.3 Flowchart Sistem

Flowchart adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. Dalam perancangan flowchart sebenarnya tidak ada rumus atau patokan yang bersifat mutalk (pasti). Hal ini didasari oleh flowchart (bagan alir) adalah sebuah gambaran dari hasil pemikiran dalam menganalisa suatu permasalahan dalam komputer. Karena setiap analisa akan menghasilkan hasil yang bervariasi antara satu dengan lainnya. Kendati begitu secara garis besar setiap perancangan flowchart selalu terdiri dari tiga bagian, yaitu input, proses dan output. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada di dalam sistem dan menunjukkan apa saja yang

dikerjakan pada sistem.Dalam proses kerja enkripsi dapat digambarkan seperti *flowchart* pada gambar berikut ini:

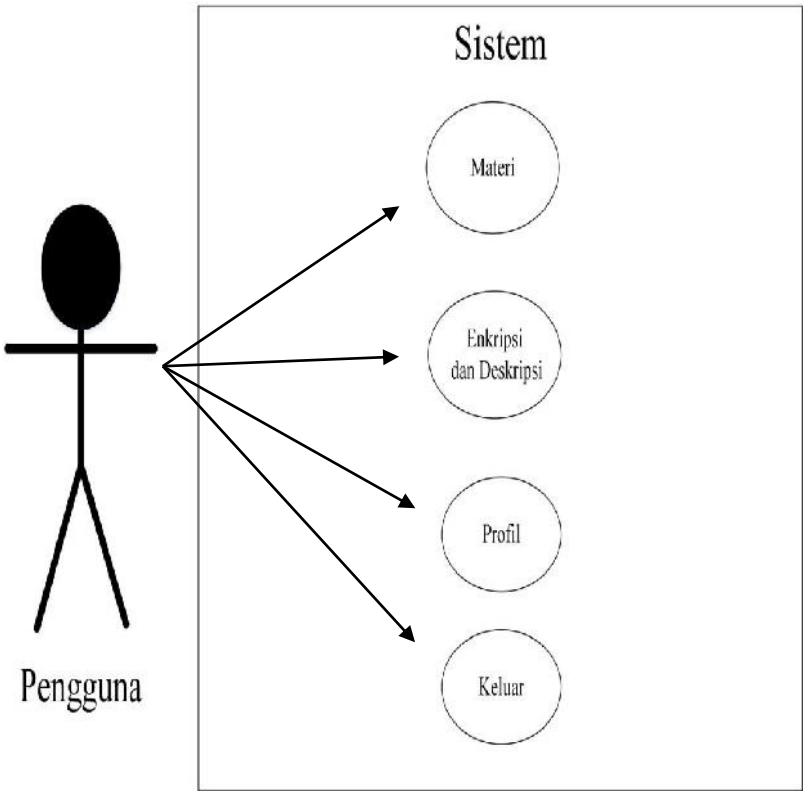


Gambar 3.1 *Flowchart* Sistem

Flowchart diatas menjelaskan proses enkripsi dengan menggunakan algoritma *one time pad*. Dimana pengguna memasukkanplainteks yang ingin dienkripsikan dan juga kunci untuk proses enkripsi. Setelah kedua elemen dimasukkan, maka proses enkripsi dilakukan sehingga menghasilkan cipherteks.

3.4 Use Case Diagram Sistem

Berikut merupakan gambaran use case diagram untuk sistem yang akan dibangun.



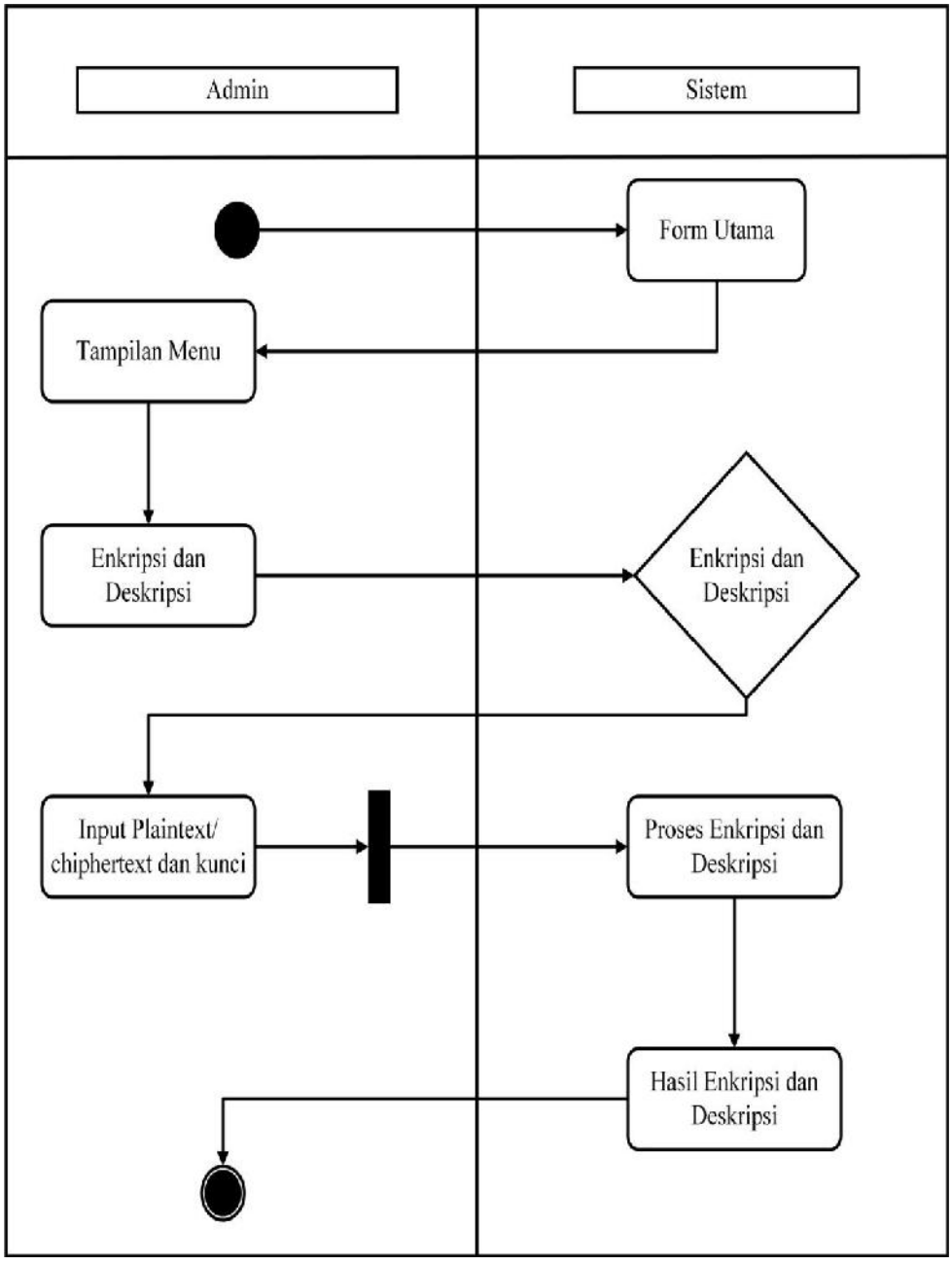
Gambar 3.2 *Use Case Diagram*

Keterangan:

Dalam *use case diagram*diatas, user pengguna sebagai actor yang mempunyai *use case* materi, enkripsi, dekripsi.

3.5 Pembuatan Activity Diagram

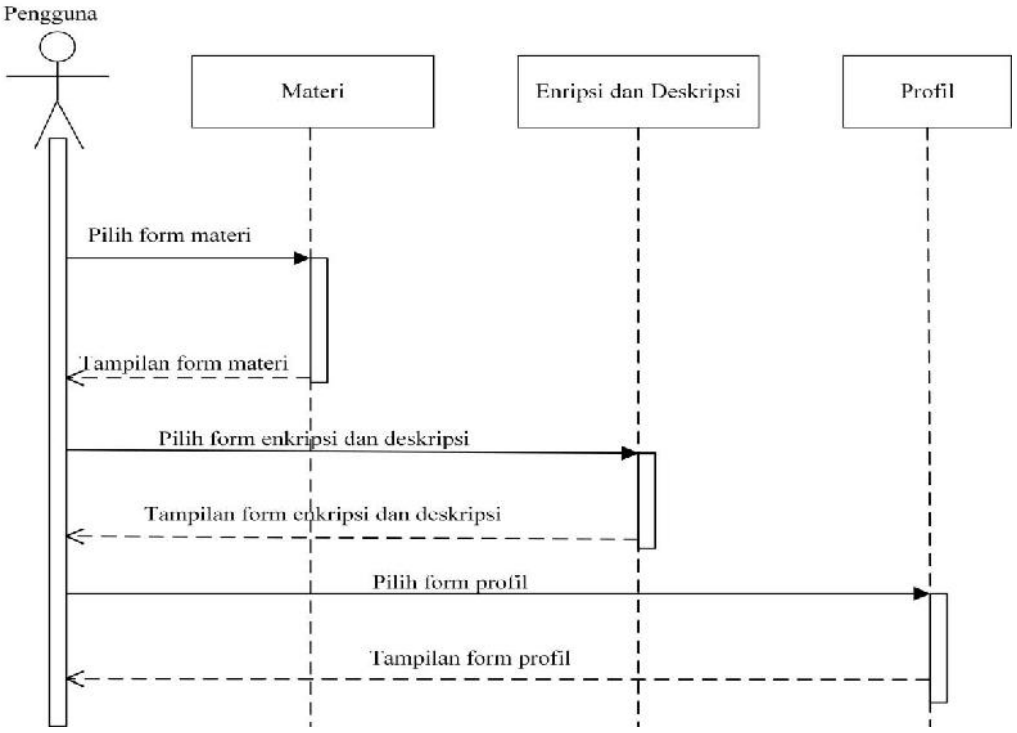
Activity diagram menggambarkan aktifitas-aktifitas yang terjadi dalam aktifitas dimulai sampai aktifitas berhenti.



Gambar 3.3 *Activity*Diagram

3.6 Sequence Diagram

Sequence diagram yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesanyang dikirimkan dan terima antar objek.



Gambar 3.4 *Sequence* Diagram

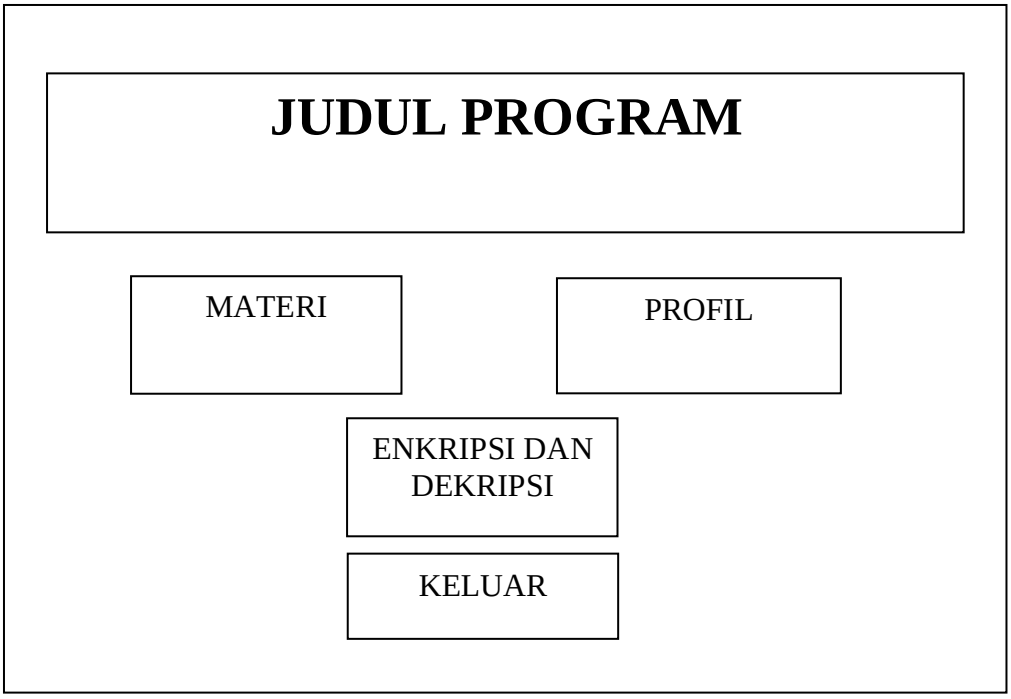
Keterangan:

1. Dalam diagram diatas menjelaskan bahwa user memilih materi kemudian sistem menampilkan materi.
2. User memilih enkripsi kemudian sistem menampilkan menu materi.
3. User memilih dekripsi kemudian sistem menampilkan menu dekripsi.
4. User memilih profil kemudian sistem menampilkan from profil.

3.7 Perancangan Antar Muka

3.7.1 Rancangan Halaman menu utama

From ini berisi tentang tombol-tombol seperti Materi, Enkripsi, Dekripsi, dan Profil.



Gambar 3.5 Rancangan menu utama

Pada tampilan diatas terdapat tombol Materi, Profil, Enkripsi, dan Dekripsi. Dari masing-masing tombol memiliki fungsi sebagai berikut:

1. Tombol materi berfungsi untuk menghubungkan pengguna ketombol from materi.
2. Tombol enkripsi dan dekripsi berfungsi untuk menghubungkan pengguna ke from enkripsi.
3. Tombol profil berfungsi untuk menghubungkan pengguna ke from profil.

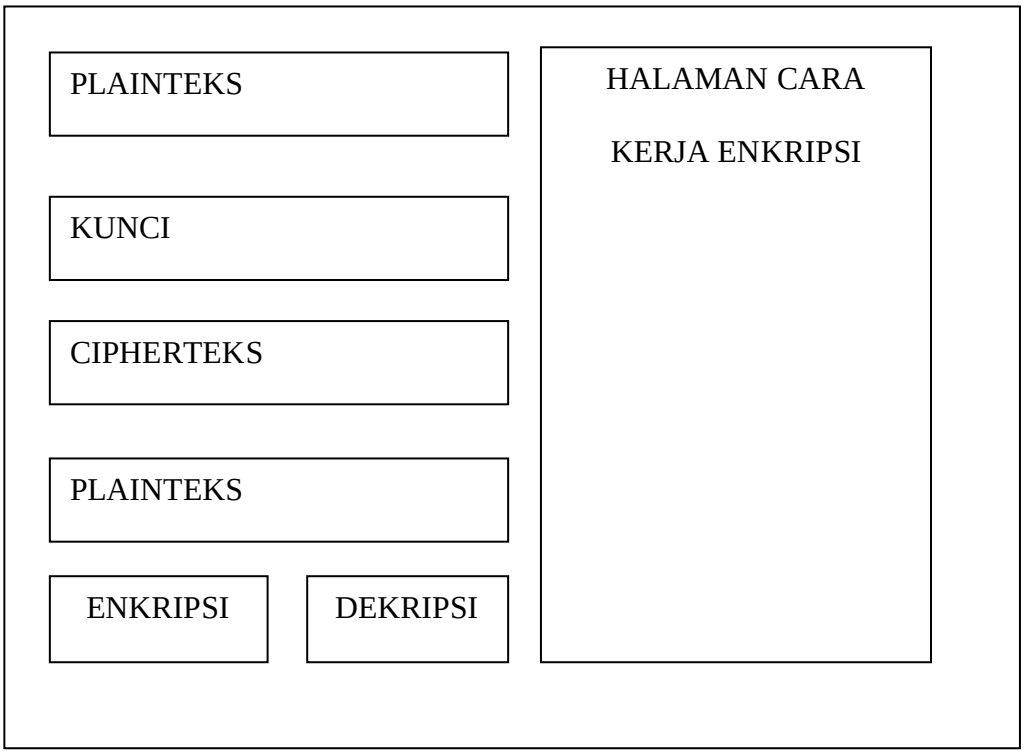
3.7.2 Rancangan Halaman Materi

From materi ini menjelaskan tentang cara kerja penyandian pada pesan, dan sejarah singkat tentang kriptografi.

Gambar 3.6 Rancangan Halaman Materi

3.7.3 Rancangan Halaman Enkripsi

Pada halaman enkripsi ini, berisi penjelasan mengenai enkripsi, pertama pengguna akan memasukan tulisan asli atau plaintekskedalam tombol masukan plainteks lalu kemudian masukan kunci yang akan dibuat, setelah itu, tekan tombol proses enkripsi yang akan kemudian menampilkan hasil cipherteks atau tulisan yang telah disandikan. Berikut contoh rancangan tampilannya:

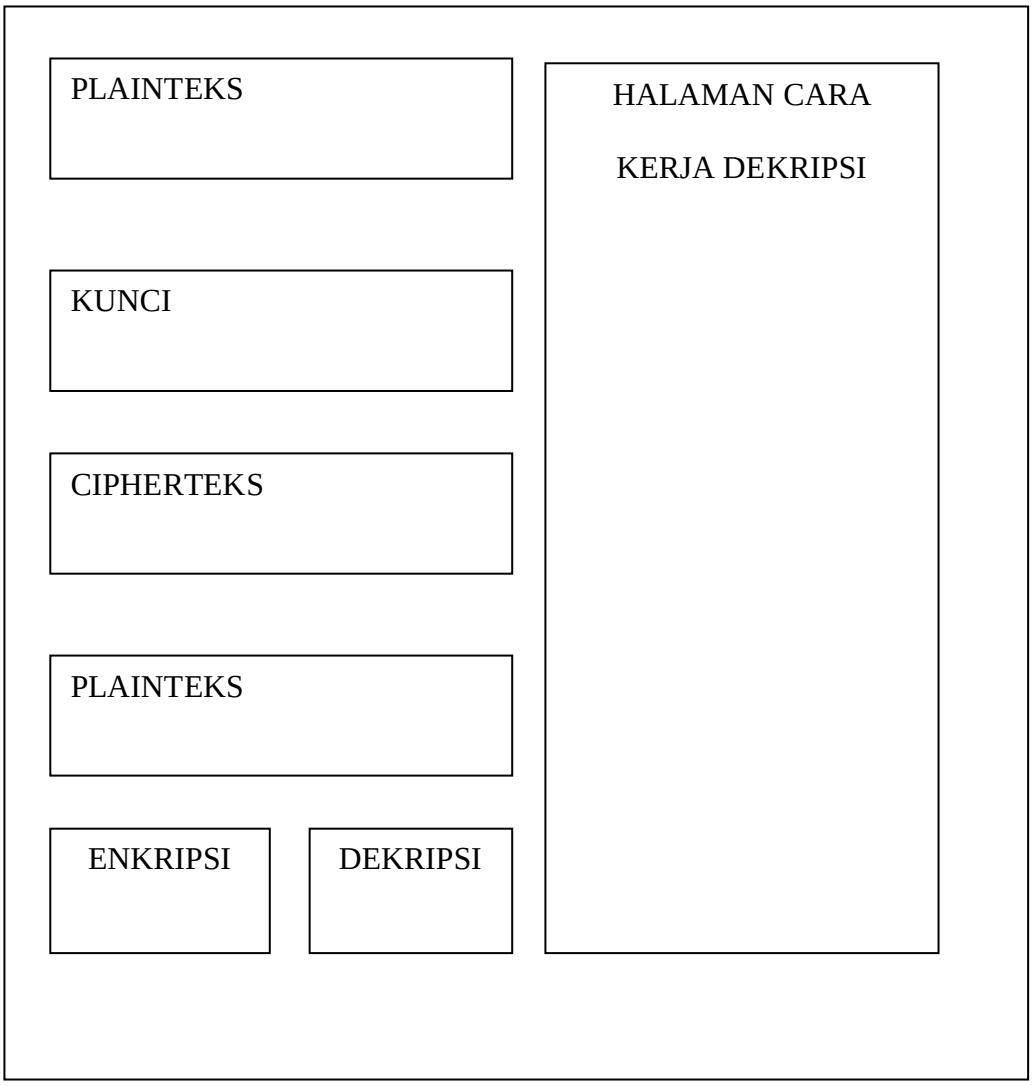


Gambar 3.7 Rancangan Halaman Enkripsi

Penjelasan pada gambar diatas terdapat kotak input enkripsi yang berfungsi untuk memasukkan tulisan yang ingin disandikan. Kemudian terdapat tombol kunci yang harus diisi sesuai keinginan pengguna. Dan kemudian terdapat tombol proses untuk menampilkan hasil dekripsi atau sandi dari pesan yang ingin disandikan tersebut.

3.7.4 Rancangan Halaman Dekripsi

Pada halaman dekripsi ini berisi tentang penjelasan mengenai proses dekripsi dari pesan yang telah disandikan. Dimana disini akan menampilkan tulisan yang telah disandikan untuk kemudian di kembalikan ketulisan yang aslinya, Berikut contoh rancangan dekripsi yang akan dibuat:

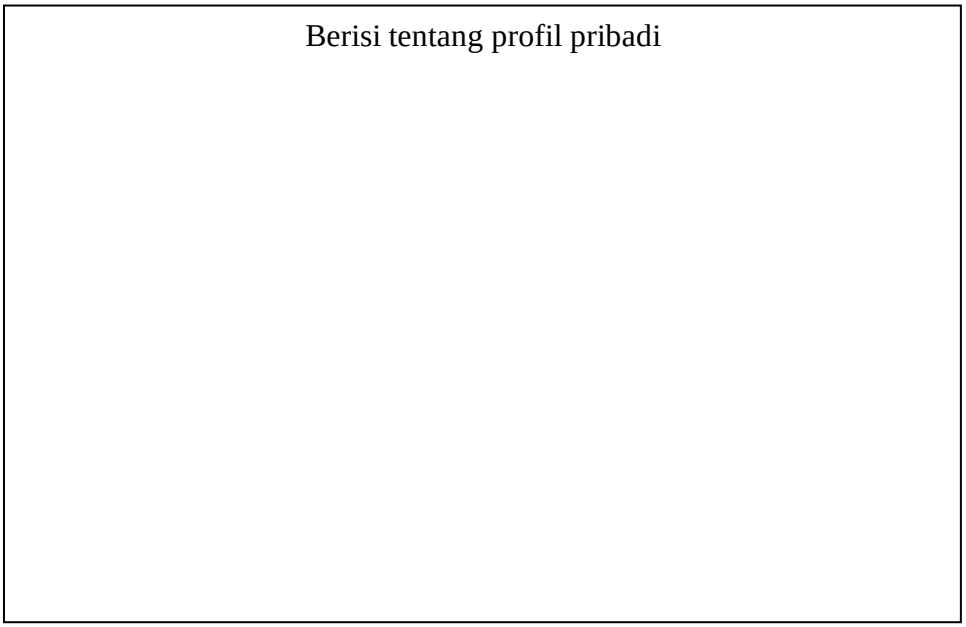


Gambar 3.8 Rancangan Halaman Dekripsi

Pada gambar diatas terdapat kotak input dekripsi yang berfungsi untuk memasukkan tulisan yang telah disandikan, lalu kemudian terdapat tombol proses untuk mengembalikan pesan ketulisan yang asli jika kunci yang dimasukkan sama dengan kunci yang dibuat pengguna pada saat melakukan proses enkripsi.

3.7.5 Rancangan Halaman Profil

Berisi tentang mengenai pembuat program aplikasi tersebut.



Gambar 3.9 Rancangan Halaman Profil

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Perancangan simulasi enkripsi dan dekripsi ini merupakan tahapan penulis melakukan analisis terhadap cara kerja dari algoritma *One Time Pad*, serta mengimplementasikannyakedalam sebuah program menggunakan visual basic tersebut dalam melakukan peroses pengamanan pesan. Lebih rincinya*One TimePad* beroperasi dengan langkah-langkah sebagai berikut.

Contoh untuk melakukan proses enkripsi dan dekripsi, plainteks yang di tulis misalnya sebagai berikut:

UNIVERSITAS PANCA BUDI MEDAN

Kemudian plainteks kita beri kunci sesuai yang pengguna ingin berikan, contoh pengguna memberikan kunci “MEDAN” dan kunci ini akan dilipat gandakan sebanyak atau sesuai plainteks yang kita buat, lalu masing-masing plainteks dan kunci di Xnor-kan, berikut penyelesainya:

Plaintext = UNIVERSITAS PANCA BUDI MEDAN

Blok Kunci = MEDANMEDANMEDANMEDANMEDANMED

Ciphertext[0] = u XNOR M

Ciphertext[0] = 117 XNOR 77

Ciphertext[0] = 01110101

01001101

======== XNOR

11000111

Ciphertext[0] = 199

Ciphertext[0] = Ç

Ciphertext[1] = n XNOR E

Ciphertext[1] = 110 XNOR 69

Ciphertext[1] = 01101110

01000101

======== XNOR

11010100

Ciphertext[1] = 212

Ciphertext[1] = Ô

Ciphertext[2] = i XNOR D

Ciphertext[2] = 105 XNOR 68

Ciphertext[2] = 01101001

01000100

======== XNOR

11010010

Ciphertext[2] = 210

Ciphertext[2] = Ò

Ciphertext[3] = v XNOR A

Ciphertext[3] = 118 XNOR 65

Ciphertext[3] = 01110110

01000001

======== XNOR

11001000

Ciphertext[3] = 200

Ciphertext[3] = È

Ciphertext[4] = e XNOR N

Ciphertext[4] = 101 XNOR 78

Ciphertext[4] = 01100101

01001110

======== XNOR

11010100

Ciphertext[4] = 212

Ciphertext[4] = Ô

Ciphertext[5] = r XNOR M

Ciphertext[5] = 114 XNOR 77

Ciphertext[5] = 01110010

01001101

======== XNOR

11000000

Ciphertext[5] = 192

Ciphertext[5] = À

Ciphertext[6] = s XNOR E

Ciphertext[6] = 115 XNOR 69

Ciphertext[6] = 01110011

01000101

======== XNOR

11001001

Ciphertext[6] = 201

Ciphertext[6] = É

Ciphertext[7] = i XNOR D

Ciphertext[7] = 105 XNOR 68

Ciphertext[7] = 01101001

01000100

======== XNOR

11010010

Ciphertext[7] = 210

Ciphertext[7] = Ò

Ciphertext[8] = t XNOR A

Ciphertext[8] = 116 XNOR 65

Ciphertext[8] = 01110100

01000001

======== XNOR

11001010

Ciphertext[8] = 202

Ciphertext[8] = Ê

Ciphertext[9] = a XNOR N

Ciphertext[9] = 97 XNOR 78

Ciphertext[9] = 01100001

01001110

======== XNOR

11010000

Ciphertext[9] = 208

Ciphertext[9] = Đ

Ciphertext[10] = s XNOR M

Ciphertext[10] = 115 XNOR 77

Ciphertext[10] = 01110011

01001101

======== XNOR

11000001

Ciphertext[10] = 193

Ciphertext[10] = Á

Ciphertext[11] = XNOR E

Ciphertext[11] = 32 XNOR 69

Ciphertext[11] = 00100000

01000101

======== XNOR

10011010

Ciphertext[11] = 154

Ciphertext[11] = š

Ciphertext[12] = p XNOR D

Ciphertext[12] = 112 XNOR 68

Ciphertext[12] = 01110000

01000100

======== XNOR

11001011

Ciphertext[12] = 203

Ciphertext[12] = Ë

Ciphertext[13] = a XNOR A

Ciphertext[13] = 97 XNOR 65

Ciphertext[13] = 01100001

01000001

======== XNOR

11011111

Ciphertext[13] = 223

Ciphertext[13] = ß

Ciphertext[14] = n XNOR N

Ciphertext[14] = 110 XNOR 78

Ciphertext[14] = 01101110

01001110

======== XNOR

11011111

Ciphertext[14] = 223

Ciphertext[14] = ß

Ciphertext[15] = c XNOR M

Ciphertext[15] = 99 XNOR 77

Ciphertext[15] = 01100011

01001101

======== XNOR

11010001

Ciphertext[15] = 209

Ciphertext[15] = Ñ

Ciphertext[16] = a XNOR E

Ciphertext[16] = 97 XNOR 69

Ciphertext[16] = 01100001

01000101

======== XNOR

11011011

Ciphertext[16] = 219

Ciphertext[16] = Û

Ciphertext[17] = XNOR D

Ciphertext[17] = 32 XNOR 68

Ciphertext[17] = 00100000

01000100

======== XNOR

10011011

Ciphertext[17] = 155

Ciphertext[17] = ›

Ciphertext[18] = b XNOR A

Ciphertext[18] = 98 XNOR 65

Ciphertext[18] = 01100010

01000001

======== XNOR

11011100

Ciphertext[18] = 220

Ciphertext[18] = Ü

Ciphertext[19] = u XNOR N

Ciphertext[19] = 117 XNOR 78

Ciphertext[19] = 01110101

01001110

======== XNOR

11000100

Ciphertext[19] = 196

Ciphertext[19] = Ä

Ciphertext[20] = d XNOR M

Ciphertext[20] = 100 XNOR 77

Ciphertext[20] = 01100100

01001101

======== XNOR

11010110

Ciphertext[20] = 214

Ciphertext[20] = Ö

Ciphertext[21] = i XNOR E

Ciphertext[21] = 105 XNOR 69

Ciphertext[21] = 01101001

01000101

======== XNOR

11010011

Ciphertext[21] = 211

Ciphertext[21] = Ó

Ciphertext[22] = XNOR D

Ciphertext[22] = 32 XNOR 68

Ciphertext[22] = 00100000

01000100

======== XNOR

10011011

Ciphertext[22] = 155

Ciphertext[22] = ›

Ciphertext[23] = m XNOR A

Ciphertext[23] = 109 XNOR 65

Ciphertext[23] = 01101101

01000001

======== XNOR

11010011

Ciphertext[23] = 211

Ciphertext[23] = Ó

Ciphertext[24] = e XNOR N

Ciphertext[24] = 101 XNOR 78

Ciphertext[24] = 01100101

01001110

======== XNOR

11010100

Ciphertext[24] = 212

Ciphertext[24] = Ô

Ciphertext[25] = d XNOR M

Ciphertext[25] = 100 XNOR 77

Ciphertext[25] = 01100100

01001101

======== XNOR

11010110

Ciphertext[25] = 214

Ciphertext[25] = Ö

Ciphertext[26] = a XNOR E

Ciphertext[26] = 97 XNOR 69

Ciphertext[26] = 01100001

01000101

======== XNOR

11011011

Ciphertext[26] = 219

Ciphertext[26] = Û

Ciphertext[27] = n XNOR D

Ciphertext[27] = 110 XNOR 68

Ciphertext[27] = 01101110

01000100

======== XNOR

11010101

Ciphertext[27] = 213

Ciphertext[27] = Õ

Cipher Text = ÇÔÒÈÔÀÉÒÊÐÁšËßßÑÛ›ÜÄÖÓ›ÓÔÖÛÕ

Untuk melakukan proses dekripsi dari pesan ini , sama juga dengan melakukan operasi yang sama yaitu dengan Xnor-kan antara cipherteks dengan kunci tersebut untuk menghasilkan plainteks, berikut langkah-langkah cara kerjanya:

Maka hasilnya akan menampilkan Plaintext kembali= universitas panca budi medan

Cipher Text = ÇÔÒÈÔÀÉÒÊÐÁšËßßÑÛ›ÜÄÖÓ›ÓÔÖÛÕ

Blok Kunci = MEDANMEDANMEDANMEDANMEDANMED

Plaintext[0] = Ç XNOR M

Plaintext[0] = 199 XNOR 77

Plaintext[0] = 11000111

01001101

======== XNOR

01110101

Plaintext[0] = 117

Plaintext[0] = u

Plaintext[1] = Ô XNOR E

Plaintext[1] = 212 XNOR 69

Plaintext[1] = 11010100

01000101

======== XNOR

01101110

Plaintext[1] = 110

Plaintext[1] = n

Plaintext[2] = Ò XNOR D

Plaintext[2] = 210 XNOR 68

Plaintext[2] = 11010010

01000100

======== XNOR

01101001

Plaintext[2] = 105

Plaintext[2] = i

Plaintext[3] = È XNOR A

Plaintext[3] = 200 XNOR 65

Plaintext[3] = 11001000

01000001

======== XNOR

01110110

Plaintext[3] = 118

Plaintext[3] = v

Plaintext[4] = Ô XNOR N

Plaintext[4] = 212 XNOR 78

Plaintext[4] = 11010100

01001110

======== XNOR

01100101

Plaintext[4] = 101

Plaintext[4] = e

Plaintext[5] = À XNOR M

Plaintext[5] = 192 XNOR 77

Plaintext[5] = 11000000

01001101

======== XNOR

01110010

Plaintext[5] = 114

Plaintext[5] = r

Plaintext[6] = É XNOR E

Plaintext[6] = 201 XNOR 69

Plaintext[6] = 11001001

01000101

======== XNOR

01110011

Plaintext[6] = 115

Plaintext[6] = s

Plaintext[7] = Ò XNOR D

Plaintext[7] = 210 XNOR 68

Plaintext[7] = 11010010

01000100

======== XNOR

01101001

Plaintext[7] = 105

Plaintext[7] = i

Plaintext[8] = Ê XNOR A

Plaintext[8] = 202 XNOR 65

Plaintext[8] = 11001010

01000001

======== XNOR

01110100

Plaintext[8] = 116

Plaintext[8] = t

Plaintext[9] = Đ XNOR N

Plaintext[9] = 208 XNOR 78

Plaintext[9] = 11010000

01001110

======== XNOR

01100001

Plaintext[9] = 97

Plaintext[9] = a

Plaintext[10] = Á XNOR M

Plaintext[10] = 193 XNOR 77

Plaintext[10] = 11000001

01001101

======== XNOR

01110011

Plaintext[10] = 115

Plaintext[10] = s

Plaintext[11] = š XNOR E

Plaintext[11] = 154 XNOR 69

Plaintext[11] = 10011010

01000101

======== XNOR

00100000

Plaintext[11] = 32

Plaintext[11] =

Plaintext[12] = Ë XNOR D

Plaintext[12] = 203 XNOR 68

Plaintext[12] = 11001011

01000100

======== XNOR

01110000

Plaintext[12] = 112

Plaintext[12] = p

Plaintext[13] = ß XNOR A

Plaintext[13] = 223 XNOR 65

Plaintext[13] = 11011111

01000001

======== XNOR

01100001

Plaintext[13] = 97

Plaintext[13] = a

Plaintext[14] = ß XNOR N

Plaintext[14] = 223 XNOR 78

Plaintext[14] = 11011111

01001110

======== XNOR

01101110

Plaintext[14] = 110

Plaintext[14] = n

Plaintext[15] = Ñ XNOR M

Plaintext[15] = 209 XNOR 77

Plaintext[15] = 11010001

01001101

======== XNOR

01100011

Plaintext[15] = 99

Plaintext[15] = c

Plaintext[16] = Û XNOR E

Plaintext[16] = 219 XNOR 69

Plaintext[16] = 11011011

01000101

======== XNOR

01100001

Plaintext[16] = 97

Plaintext[16] = a

Plaintext[17] = › XNOR D

Plaintext[17] = 155 XNOR 68

Plaintext[17] = 10011011

01000100

======== XNOR

00100000

Plaintext[17] = 32

Plaintext[17] =

Plaintext[18] = Ü XNOR A

Plaintext[18] = 220 XNOR 65

Plaintext[18] = 11011100

01000001

======== XNOR

01100010

Plaintext[18] = 98

Plaintext[18] = b

Plaintext[19] = Ä XNOR N

Plaintext[19] = 196 XNOR 78

Plaintext[19] = 11000100

01001110

======== XNOR

01110101

Plaintext[19] = 117

Plaintext[19] = u

Plaintext[20] = Ö XNOR M

Plaintext[20] = 214 XNOR 77

Plaintext[20] = 11010110

01001101

======== XNOR

01100100

Plaintext[20] = 100

Plaintext[20] = d

Plaintext[21] = Ó XNOR E

Plaintext[21] = 211 XNOR 69

Plaintext[21] = 11010011

01000101

======== XNOR

01101001

Plaintext[24] = 11010100

01001110

======== XNOR

01100101

Plaintext[24] = 101

Plaintext[24] = e

Plaintext[25] = Ö XNOR M

Plaintext[25] = 214 XNOR 77

Plaintext[25] = 11010110

01001101

======== XNOR

01100100

Plaintext[25] = 100

Plaintext[25] = d

Plaintext[26] = Û XNOR E

Plaintext[26] = 219 XNOR 69

Plaintext[26] = 11011011

01000101

======== XNOR

01100001

Plaintext[26] = 97

Plaintext[21] = 105

Plaintext[21] = i

Plaintext[22] = › XNOR D

Plaintext[22] = 155 XNOR 68

Plaintext[22] = 10011011

01000100

======== XNOR

00100000

Plaintext[22] = 32

Plaintext[22] =

Plaintext[23] = Ó XNOR A

Plaintext[23] = 211 XNOR 65

Plaintext[23] = 11010011

01000001

======== XNOR

01101101

Plaintext[23] = 109

Plaintext[23] = m

Plaintext[24] = Ô XNOR N

Plaintext[24] = 212 XNOR 78

Plaintext[26] = a

Plaintext[27] = Õ XNOR D

Plaintext[27] = 213 XNOR 68

Plaintext[27] = 11010101

01000100

======== XNOR

01101110

Plaintext[27] = 110

Plaintext[27] = n

Plain Text = universitas panca budi medan

4.2 Tampilan Perancangan Aplikasi

Tampilan perancangan aplikasi merupakan penjelasan tentang tampilan aplikasi yang sudah di buat berikut tampilan- tampilan perancangan aplikasi tersebut:

4.2.1 Tampilan Menu Utama

Tampilan menu utama merupakan tampilan yang pertama kali muncul saat pertama kali program di buka atau dijalankan, didalam menu utama terdapat menu seperti materi, enkripsi dan dekripsi, profil dan keluar.

Gambar 4.1 Tampilan Menu Utama

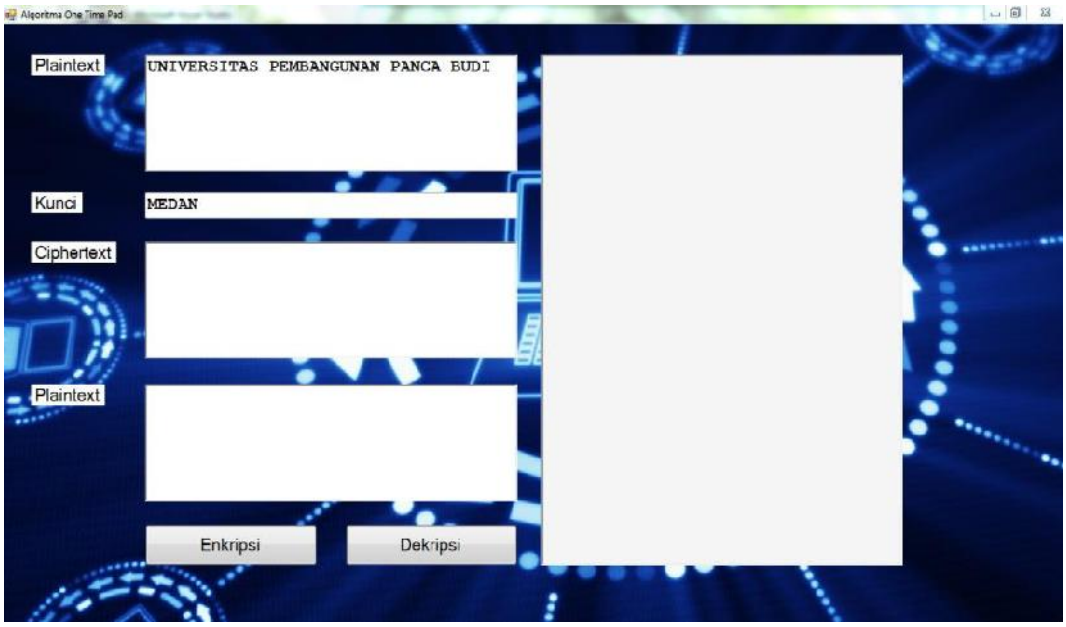
4.2.2 Tampilan Materi

Tampilan materi menampilkan sejarah singkat tentang kriptografi dengan algoritma Vernam Cipher.

Gambar 4.2 Tampilan Materi

4.2.3 Tampilan Enkripsi Dan Dekripsi

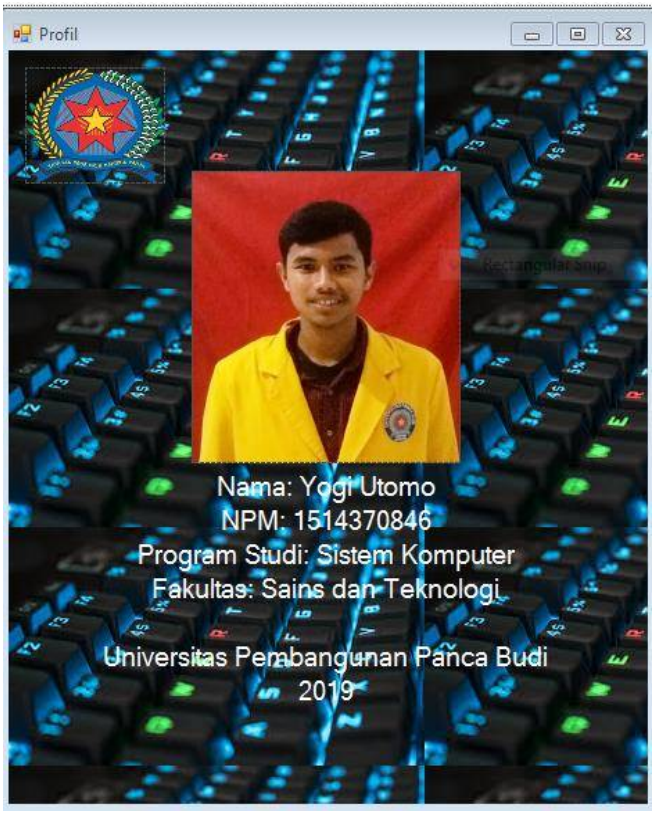
Tampilan enkripsi dan dekripsi ini berfungsi untuk menggantikan tulisan asli menjadi tulisan yang disandikan dan mengembalikannya ketulisan yang asli kembali dengan menggunakan algoritma *One Time Pad*. Untuk mengkonversikan tulisan terserbut dibutuhkan kunci agar tidak muda di buka.



Gambar 4.3 Tampilan Enkripsi dan Dekripsi

4.2.4 Tampilan Profil

Tampilan profil menampilkan Nama, NPM, Prodi, dan Fakultas dari penulis.



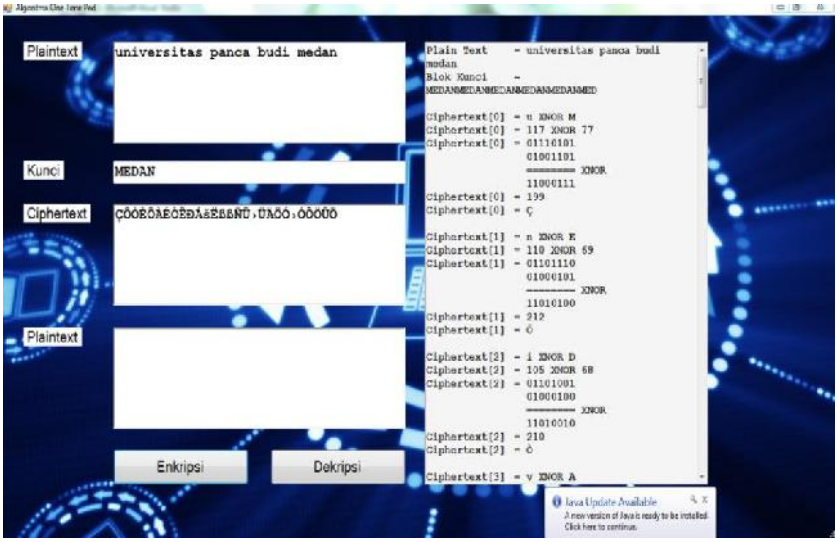
Gambar 4.4 Tampilan Profil

4.3 Pengujian Sistem

Dalam melakukan pengujian sistem, penulis menggunakan sistem operasi windows 7 dengan pemrograman visual basic, sebelum diuji terlebih dahulu penulis melakukan hitungan manual yang bertujuan membandingkan jawaban antara manual dengan yang dibuat menggunakan sistem.

4.3.1 Proses Pengujian Sistem Cara Kerja Enkripsi

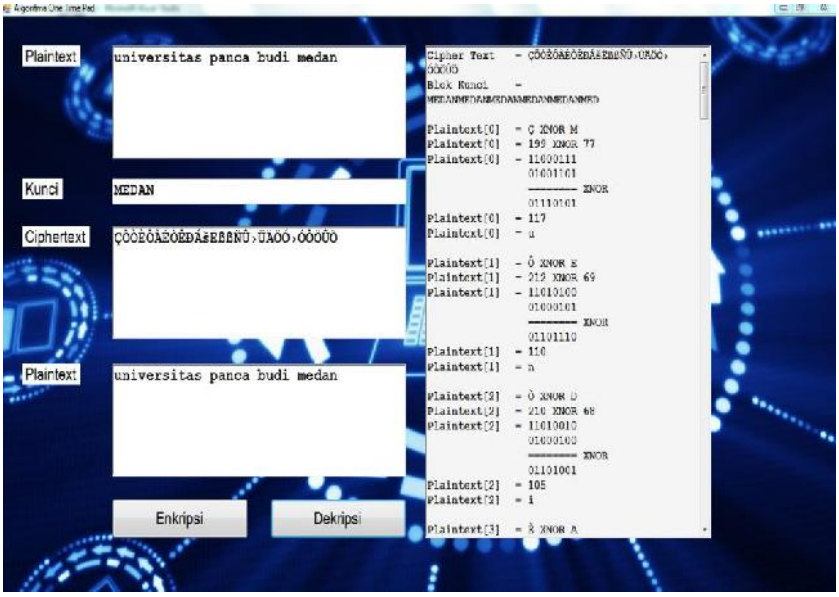
Berikut ini adalah gambar yang menunjukkan sistem cara kerja enkripsi.



Gambar 4.5 Pengujian Sistem Cara Kerja Enkripsi

4.3.2 Proses Pengujian Sistem Cara Kerja Dekripsi

Berikut ini adalah gambar yang menunjukkan sistem cara kerja dekripsi.



Gambar 4.6 Pengujian Sistem Cara Kerja Dekripsi

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan aplikasi simulasi enkripsi dan dekripsi tersebut, maka kesimpulan yang didapat dalam penulisan skripsi ini adalah:

1. Dalam melakukan enkripsi dan dekripsi berhasil dilakukan plainteks dapat dienkripsi menjadi data yang disandikan dan dapat didekripsikan menjadi data yang asli kembali.
2. Dalam melakukan proses enkripsi dan dekripsi kunci harus sama.
1. Dapat membantu pengguna dalam mengenkripsikan serta dekripsi pesan menggunakan algoritma *one time pad*.

5.2 Saran

Penelitian juga memiliki beberapa kelemahan dalam prosesnya. Ada beberapa saran yang dapat penulis paparkan untuk meningkatkan kualitas penelitian ini. Beberapa saran tersebut adalah antara lain:

1. Diharapkan program ini dapat dikembangkan dengan penambahan algoritma lainnya.
2. Untuk menjalankan aplikasi ini, sebaiknya menggunakan komputer yang telah memiliki spesifikasi yang cukup.
3. Diharapkan dengan adanya pengembangan algoritma sehingga kualitas penulis skripsi mendatang lebih baik

DAFTAR PUSTAKA

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

Jogiyanto, H. M. (2016). *Analisis Dan Desain Sistem Informasi, Pendekatan Terstruktur Teori Dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Offset.

Khairina, N., & Harahap, M. K. (2017). Analisis Algoritma One Time Pad Dengan Algoritma Cipher Transposisi Sebagai Pengamanan Pesan Teks. *SinkrOn*, *1*(2), 58. https://doi.org/10.33395/sinkron.v1i2.42

Khairul, K., IlhamiArsyah, U., Wijaya, R. F., & Utomo, R. B. (2018, September). IMPLEMENTASI AUGMENTED REALITY SEBAGAI MEDIA PROMOSI PENJUALAN RUMAH. In Seminar Nasional Royal (SENAR) (Vol. 1, No. 1, pp. 429-434).

Kurniawan, H. (2018). Pengenalan Struktur Baru untuk Web Mining dan Personalisasi Halaman Web. Jurnal Teknik dan Informatika, 5(2), 13-19.

Kurniawan, T. A. (2018). Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *5*(1), 77. https://doi.org/10.25126/jtiik.201851610

Muhammad, S. (2014). Belajar Kriptografi dengan Teknik Subtitusi & Transposisi.

Nurgoho, A. (2019). *Rekayasa Perangkat Lunak Menggunakan UML dan JAVA*. Yogyakarta: Andi Offset.

Putra, Randi Rian, and Cendra Wadisman. "Implementasi Data Mining Pemilihan Pelanggan Potensial Menggunakan Algoritma K Means." INTECOMS: Journal of Information Technology and Computer Science 1.1 (2018): 72-77.

Putra, Randi Rian. "Sistem Informasi Web Pariwisata Hutan Mangrove di Kelurahan Belawan Sicanang Kecamatan Medan Belawan Sebagai Media Promosi." Jurnal Ilmiah Core IT: Community Research Information Technology 7.2 (2019).

Putra, Randi Rian, et al. "Decision Support System In Selecting Additional Employees Using Multi-Factor Evaluation Process Method." (2019).

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

Rohmanu, A. (2017). Implementasi Kriptografi dan Steganografi Dengan Metode Algoritma Des dan Metode End Of File. *Jurnal Informatika SIMANTIK*, *2*(1), 1–11.

Sari, C. A., Rachmawanto, E. H., Utomo, D. W., & Sani, R. . (2016). Penyembunyian Data Untuk Seluruh Ekstensi File Menggunakan Kriptografi Vernam Cipher dan Bit Shiffting. *Journal of Applied Intelligent System*, *1*(3), 179–190.

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

Setiawan, I., Raharjo, W. S., & Susanto, B. (2016). Implementasi Permainan Flow Pada Pembangunan Sistem Captcha. *Jurnal Informatika*, *11*(2), 117–126. https://doi.org/10.21460/inf.2015.112.448

Siahaan, A. P. U. (2016). Securing Short Message ServiceUsing Vernam Cipher in Android Operating System. *IOSR*. https://doi.org/10.9790/0050-03041116

Siahaan, A. P. U. (2017). Vernam Conjugated Manipulation of Bit-plane Complexity Segmentation. *International Journal of Security and Its Applications*, *11*(9), 1–12. https://doi.org/10.14257/ijsia.2017.11.9.01

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

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

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

Sidik, A. P., Efendi, S., & Suherman, S. (2019, June). Improving One-Time Pad Algorithm on Shamir's Three-Pass Protocol Scheme by Using RSA and ElGamal Algorithms. In Journal of Physics: Conference Series (Vol. 1235, No. 1, p. 012007). IOP Publishing.

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

Tasril, V., Wijaya, R. F., & Widya, R. (2019). APLIKASI PINTAR BELAJAR BIMBINGAN DAN KONSELING UNTUK SISWA SMA BERBASIS MACROMEDIA FLASH. Jurnal Informasi Komputer Logika, 1(3).

Wibowo, H. R. (2019). *Visual Basic Database*. Yogyakarta: Jubilee Enterprise.

Wijaya, Rian Farta, et al. "Aplikasi Petani Pintar Dalam Monitoring Dan Pembelajaran Budidaya Padi Berbasis Android." Rang Teknik Journal 2.1 (2019).