RANCANG BANGUN PROSES ENKRIPSI DAN DESKRIPSI STREAM CHIPPER DENGAN TEKNIK XNOR

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	:	**GUSTI PRAMONO ATMOJO**
NPM	:	**1414370482**
PROGRAM STUDI	:	**SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020

ABSTRAK

GUSTI PRAMONO ATMOJO
Rancang Bangun Proses Enkripsi dan Dekripsi Stream Cipher dengan Teknik XNOR
2019

Sistem informasi pada zaman sekarang ini sangat erat kaitannya dengan data. Data adalah hal yang harus dijaga keamanannya. Pertukaran data pada jaringan tidak dapat dihindari karena informasi harus diupdate melalui jaringan tersebut. Pencurian data tidak mungkin dapat dihindari, tetapi ada hal yang dapat dilakukan untuk mencegah penyalahgunaan data. Hal tersebut adalah pengamanan data. Dengan kriptografi, data tersebut dapat diamankan sehingga terhindar dari gangguan orang yang tidak bertanggung jawab. Algoritma kriptografi dengan teknik XNOR sangat baik dalam melakukan proses enkripsi. Algoritma ini bekerja dengan sangat cepat untuk melakukan perubahan plaintext ke ciphertext melalui operasi bit. Dengan menerapkan algoritma ini, keamanan data akan lebih dapat ditingkatkan.

Kata Kunci: dekripsi, enkripsi, kriptografi, XNOR, stream

DAFTAR PUSTAKA

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR ISI

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT karena dengan anugrah dan hidyahNya penulis masih diberikan kesempatan untuk menyelesaikan skripsi ini dapat diselesaikan dengan baik dan sebagaimana mestinya. Skripsi ini berjudul ”RANCANG BANGUN ENKRIPSI DAN DESKRIPSI STREAM CIPHER DENGAN TEKNIK XNOR ”. Penulis mengucapkan banyak terima kasih kepada banyak pihak yang telah membantu dalam penyelesaian penyusunan skripsi ini. Penulis ingin mengucapkan terima kasih kepada:

1. Saya ucapkan terima kasih untuk kedua orang tua saya yang telah mendukung saya untuk menyelesaikan skripsi ini.
2. Bapak Dr. H. Muhammad Isa Indrawan, S.E, M.M selaku Rektor Universitas Pembangunanan Panca Budi Medan.
3. Bapak Ir. Bhakti Alamsyah, M.T, Ph.D., selaku Rektor I Universitas Pembangunan Panca Budi Medan.
4. Bapak Hamdani, ST.,M.T selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
5. Bapak Eko Hariyanto, S.Kom., M.Kom, selaku Ketua Program Studi Sistem Komputer Universitas Pembangunan Panca Budi Medan.
6. Bapak Andysah Putera Utama Siahaan, S.Kom., M.Kom., Ph.D., selaku Dosen Pembimbing I yang telah memberikan arahan dan membimbing dalam peyelesaian skripsi ini.

7. Bapak Heri Kurniawan, S.Kom., M.Kom, selaku Dosen Pembimbing II yang telah memberikan korek terhadap tata tulis untuk penyelesaian skripsi ini.
8. Dosen-dosen pada Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
9. Seluruh staff dan karyawan pada Universitas Pembangunan Panca Budi Medan.
10. Teman-teman penulis dari program studi Sistem Komputer Fakultas Ilmu Komputer Universitas Pembangunan Panca Budi Medan

Penulis juga menyadari bahwa penyusunan skripsi ini belum sempurna baik dalam penulisan maupun isi disebabkan keterbatasan kemampuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang sifatnya membangun dari pembaca untuk kesempurnaan isi skripsi ini.

Medan, 25 Nopember 2019
Penulis

Gusti Pramono Atmojo
1414370482

BAB I

PENDAHULUAN

1.1 Latar Belakang

Keamanan dan kerahasiaan data merupakan suatu aspek yang sangat penting dalam proses pertukaran pesan atau informasi. Suatu pesan yang sifatnya rahasia membutuhkan suatu sistem penyimpanan dan pengiriman data atau *file* agar tidak mudah terbaca dan diketahui semua orang. Ada berbagai macam cara untuk mengamankan data atau *file*, salah satunya adalah menggunakan metode kriptografi.

Saat ini kriptografi terbagi menjadi dua yaitu kriptografi klasik dan kriptografi modern. Pada kriptografi klasik terdapat algoritma *Stream Chiper*.*Stream Chiper* ini mempunyai 26 kemungkinan karena menggunakan alfabet. *Stream Chiper* merupakan algoritma klasik untuk menyandikan sebuah *plaintext* dengan cara substitusi sehingga dalam memecahkan pesan tersebut akan terasa susah. Penelitian ini menggunakan pemrograman *Visual Basic*.*Net* 2010. (Halim Agung; 2015)

Algoritma Stream Chiper merupakan salah satu metode kriptografi berbasis protokol. Protokol adalah aturan yang berisi tentang langkah-langkah yang melibatkan dua kunci yang dibuat untuk menyelesaikan suatu kegiatan. Dalam kriptografi, protokol digunakan oleh orang-orang yang terlibat, seperti untuk proses otentifikasi, pengaktifan bilangan acak, bahkan untuk berbagi dan bertukar informasi yang bersifat rahasia. Pengirim dan penerima pesan melakukan

penukaran sebanyak tiga tahap untuk mengekripsikan pesan tersebut. Pada dasarnya, *Algoritma Stream Chiper* di implementasikan dengan menggunakan satu algoritma enkripsi dan dekripsi yang telah disepakati oleh kedua belah pihak.

Berdasarkan latar belakang yang telah penulis uraikan di atas, maka penulis tertarik untuk memilih judul **"*rancang bangun enkripsi dan deskripsi stream chipper dengan teknik xnor*".**

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas dapat penulis simpulkan bahwa yang menjadi pokok permasalahan dalam pembahasan ini adalah sebagai berikut:

1. Bagaimana merancang bangun aplikasi pesan biasa menjadi pesan rahasia?
2. Bagaimana menerapkan metode algoritma *Stream Chiper* pada pesan rahasia?
3. Bagaimana melakukan proses enkripsi dan dekripsi dengan teknix xnor?

1.3 Batasan Masalah

Berdasarkan perumusan masalah diatas maka penulis melakukan pembatasan masalah yang akan dibahas sebagai berikut:

1. Implementasi enkripsi dan dekripsi hanya berupa teks.
2. Program yang dibahas menggunakan pemrograman Visual Basic.Net 2010.
3. Menggunakan Tabel Konversi *Stream* Cipher ke Angka.
4. Algoritma Srteam Chipper Yang di gunakan adalah kriptograpi XNOR.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini dengan menggunakan algoritma *Stream Chiper* ini yang ingin dicapai adalah sebagai berikut:

1. Merancang pesan rahasia dengan keamanan data teks dengan algoritma *Stream Chiper*.
2. Untuk menentukan proses XNOR pada stream chipper
3. Untuk melakukan proses enkripsi dan deskripsi dengan algoritma stream chiper

1.5 Manfaat Penelitian

Adapun manfaat dalam penelitian ini yang diperoleh dari penerapan algoritma *Stream Chiper* adalah sebagai berikut:

1. Dapat merahasiahkan data informasi setelah adanya proses enkripsi.
2. Informasi yang dikirimkan tidak gampang di curi terhadap orang yang tidak bertanggung jawab.

BAB II

LANDASAN TEORI

2.1 Keamanan Data

Pada zaman teknologi informasi sekarang, data atau informasi merupakan suatu asset yang sangat berharga dan harus dilindungi. Hal ini juga diikuti oleh kemajuan teknologi komputer. Kemajuan teknologi komputer membantu semua aspek kehidupan manusia. Dengan adanya kemajuan dalam teknologi informasi, komunikasi dan komputer maka kemudian muncul masalah baru, yaitu masalah keamanan akan data dan informasi dan dalam hal ini akan membuka peluang bagi orang-orang yang tidak bertanggung jawab untuk menggunakannya sebagai tindak kejahatan. Dan tentunya akan merugikan pihak tertentu. Dalam keamanan data ada beberapa aspek yang berkaitan dengan persyaratan kemanan yaitu(Pabokory, 2015:2).

1. *Secrecy*. Berhubungan dengan akses membaca data dan informasi. Data dan informasi di dalam suatu sistem komputer hanya dapat diakses dan dibaca oleh orang yang berhak.
2. *Integrity*. Berhubungan dengan akses merubah data dan informasi. Data dan informasi di dalam suatu sistem komputer hanya dapat diubah oleh orang yang berhak.

3. *Availability*. Berhubungan dengan ketersediaan data dan informasi. Data dan informasi yang berada dalam suatu sistem komputer tersedia dan dapat dimanfaatkan oleh orang yang berhak. (Pabokory, 2015:2).
4. Lebih lanjut menurut (Pabokory, 2015:2), terdapat lima langkah kemanan komputer yang baik untuk diperhitungkan yaitu; aset, analisis resiko, perlindungan, alat dan prioritas.

2.2 *Kriptografi*

Kriptografi merupakan kata dari bahasa Yunani yaitu cryptography, terdiri dari dua suku kata yaitu kripto dan graphia. Kripto artinya menyembunyikan, sedangkan graphia artinya tulisan. Sehingga, bila digabungkan akan menjadi kata yang berarti menyembunyikan/merahasiakan tulisan. *Kriptografi* adalah suatu ilmu ataupun seni mengamankan pesan dan dilakukan oleh *cryptographer* (Anonim, 2014).

Menurut (Rhee, 2013). *kriptografi* digunakan untuk memastikan privasi dan autentifikasi data dalam komunikasi antar sistem komputer. Terdapat dua proses dasar dalam *kriptografi* yaitu:

1. *Enkripsi*, adalah sebuah proses yang melakukan perubahan sebuah kode dari yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti (tidak terbaca). (Pabokory, 2015).
2. *Deskripsi*, adalah kebalikan dari *Enkripsi* yaitu mengubah kembali bentuk tersamar tersebut menjadi informasi awal. (Pabokory, 2015).

Sebuah pesan atau data yang masih asli dan belum mengalami penyandian dikenal dengan istilah plaintext. Kemudian setelah disamarkan dengan suatu cara penyandian, maka plaintextini disebut sebagai chipertext. Proses penyamaran dari plaintext ke *cipher*text disebut *Enkripsi* (encryption), dan proses pengembalian dari *cipher*textmenjadi plaintext kembali disebut dekripsi (decryption). (Pabokory, 2015). *File* yang dapat di *Enkripsi* dapat berupa teks, gambar maupun audio dan video.

2.3 Macam-Macam Kriptografi

Kriptografi dibedakan menjadi 3 bagian yaitu kriptografi simetris, kriptografi asimetris dan fungsi hash satu arah. Kriptografi simetris disebut juga kriptografi kunci rahasia merupakan jenis kriptografi paling intuitif. Ini termasuk penggunaan kunci rahasia yang dikenal hanya pada pengguna komunikasi yang aman. Kriptografi asimetris sendiri berbeda dengan kriptografi simetris, dimana kriptografi asimetris ini menggunakan dua kunci yang berbeda, yaitu kunci publik dan kunci rahasia atau kunc pribadi. Kunci-kunci tersebut berhubungan secara matematis, tetapi tidak mungkin secara perhitungan untuk menarik kesimpulan satu dengan yang lain.

Fungsi *hash* satu arah, juga dikenal sebagai rangkuman pesan atau fungsi kompresi adalah fungsi matematis yang mengambil input panjang variabel dan mengubahnya ke dalam urutan biner dengan panjang yang tetap.

2.4 Teknik Kriptografi

Kriptografi terkait erat dengan disiplin ilmu kriptologi dan kriptanalisis. Ini mencakup teknik seperti mikrodot, menggabungkan kata-kata dengan gambar, dan cara-cara lain untuk menyembunyikan informasi dalam penyimpanan atau transit. Namun, di dunia komputer-sentris saat ini, kriptografi paling sering dikaitkan dengan scrambling *plaintext* (teks biasa, kadang-kadang disebut sebagai *cleartext*) menjadi *ciphertext* (proses yang disebut enkripsi), kemudian kembali lagi (dikenal sebagai dekripsi). Individu yang berlatih bidang ini dikenal sebagai cryptographers. Kriptografi modern berkaitan dengan empat tujuan berikut:

1. Kerahasiaan: informasi tidak dapat dipahami oleh siapa pun yang tidak disengaja
2. Integritas: informasi tidak dapat diubah dalam penyimpanan atau transit antara pengirim dan penerima yang dituju tanpa perubahan yang terdeteksi
3. Non-repudiation: pencipta / pengirim informasi tidak dapat menyangkal pada tahap selanjutnya niatnya dalam pembuatan atau transmisi informasi
4. Otentikasi: pengirim dan penerima dapat mengkonfirmasi identitas satu sama lain dan asal / tujuan informasi

Prosedur dan protokol yang memenuhi beberapa atau semua kriteria di atas dikenal sebagai *cryptosystems*. *Cryptosystems* sering dianggap hanya merujuk pada prosedur matematika dan program komputer; namun, mereka juga memasukkan pengaturan perilaku manusia, seperti memilih kata sandi yang sulit

ditebak, keluar dari sistem yang tidak digunakan, dan tidak membahas prosedur sensitif dengan orang luar.

2.5 Kriptografi XNOR

Kriptografi XNOR didasarkan pada prinsip bahwa setiap karakter plaintext dari sebuah pesan 'dicampur' dengan satu karakter dari keystream. Jika keystream yang benar-benar acak digunakan, hasilnya akan menjadi ciphertext yang benar-benar 'acak' yang tidak ada hubungannya dengan plaintext asli. Dalam hal ini, cipher mirip dengan One-Time Pad (OTP) yang tidak dapat dipecahkan. Seperti yang umumnya digunakan dengan teleprinter dan 5-level tape, sistem ini juga dikenal sebagai One-Time Tape atau OTT (US Patent 1,310,719, 1919).

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

Hal di atas menjadi mungkin setelah diperkenalkannya telegrafi digital, juga dikenal sebagai Teletype 1 atau Telex. Dengan teletypewriter, setiap karakter diganti dengan kode 5-bit digital - diwakili oleh 5 lubang dalam pita kertas berlubang - yang biasa digunakan dengan mesin telex. Ini umumnya dikenal sebagai ITA2 atau kode Baudot-Murray. Kode digital juga dapat diwakili oleh

serangkaian '1' dan '0', di mana 1 mewakili keberadaan lubang dan 0 mewakili tidak adanya lubang.

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

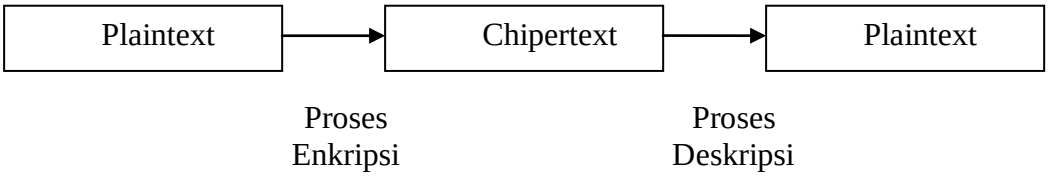
plaintext + key = ciphertext

ciphertext + key = plaintext

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

2.6 *Enkripsi*

Enkripsi merupakan hal yang sangat penting dalam *kriptografi* supaya keamanan data yang dikrimkan bisa terjaga kerahasiaannya. Pesan asli (plaintext) diubah menjadi kode-kode yang tidak dimengerti. *Enkripsi* bisa diartikan dengan chiper atau kode. Sama halnya dengan kita yang tidak mengerti sebuah kata, kita akan dapat melihatnya di dalam kamus atau daftar istilah-istilah. Berbeda halnya dengan *Enkripsi*, untuk mengubah plaintext ke bentuk chipertext, kita harus menggunakan algoritma yang dapat mengkodekan data yang kita inginkan. Berikut adalah penggambaran proses *Enkripsi*.



Gambar 2.1 Proses *Enkripsi* dan *Deskripsi*

(Sumber: Pabokory, 2015)

2.7 Dekripsi

Dekripsi adalah proses mengambil teks yang disandikan atau dienkripsi atau data lain dan mengubahnya kembali menjadi teks yang dapat Anda baca dan pahami oleh komputer. Istilah ini dapat digunakan untuk menggambarkan metode mendekripsi data secara manual atau mendekripsi data menggunakan kode atau kunci yang tepat. Data dapat dienkripsi untuk menyulitkan seseorang untuk mencuri informasi. Beberapa perusahaan juga mengenkripsi data untuk perlindungan umum data perusahaan dan rahasia dagang. Jika data ini perlu dapat dilihat, mungkin memerlukan dekripsi. Jika kode sandi atau kunci dekripsi tidak tersedia, perangkat lunak khusus mungkin diperlukan untuk mendekripsi data menggunakan algoritme untuk memecahkan dekripsi dan membuat data dapat dibaca (Archana & Vashist, 2017).

Dekripsi adalah proses mengubah data yang telah dibuat tidak dapat dibaca melalui enkripsi kembali ke bentuk yang tidak dienkripsi. Dalam dekripsi, sistem mengekstrak dan mengonversi data yang rusak dan mengubahnya menjadi teks dan gambar yang mudah dimengerti tidak hanya oleh pembaca tetapi juga

oleh sistem. Dekripsi dapat dilakukan secara manual atau otomatis. Ini juga dapat dilakukan dengan seperangkat kunci atau kata sandi. Salah satu alasan utama untuk menerapkan sistem enkripsi-dekripsi adalah privasi. Ketika informasi menyebar melalui World Wide Web, informasi tersebut menjadi subyek pengawasan dan akses dari individu atau organisasi yang tidak berwenang. Akibatnya, data dienkripsi untuk mengurangi kehilangan dan pencurian data. Beberapa item umum yang dienkripsi termasuk pesan email, file teks, gambar, data pengguna, dan direktori. Orang yang bertanggung jawab atas dekripsi menerima prompt atau jendela di mana kata sandi dapat dimasukkan untuk mengakses informasi yang dienkripsi.

2.8 *Kriptografi* Klasik

Menurut (Bishop, 2014). *kriptografi* klasik adalah *kriptografi* yang disebut juga sebagai *kriptografi* kunci tunggal atau *kriptografi* simetris yang menggunakan kunci yang sama untuk *Enkripsi* maupun *Deskripsi*. *Kriptografi* klasik merupakan *kriptografi* yang digunakan pada zaman dahulu sebelum komputer ditemukan atau sudah ditemukan namun belum secanggih sekarang. *Kriptografi* ini melakukan pengacakan huruf pada kata terang / plaintext.

2.9 Vernam Cipher

Vernam Cipher didasarkan pada prinsip bahwa setiap karakter *plaintext* dari sebuah pesan 'dicampur' dengan satu karakter dari keystream. Jika *keystream* yang benar-benar acak digunakan, hasilnya akan menjadi *ciphertext* yang benar-

benar 'acak' yang tidak ada hubungannya dengan plaintext asli. Dalam hal ini, *cipher* mirip dengan *One-Time Pad (OTP)* yang tidak dapat dipecahkan. Seperti yang umumnya digunakan dengan teleprinter dan 5-level tape, sistem ini juga dikenal sebagai *One-Time Tape atau OTT* (US Patent 1,310,719, 1919).

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

Hal di atas menjadi mungkin setelah diperkenalkannya telegrafi digital, juga dikenal sebagai *Teletype 1 atau Telex*. Dengan *teletypewriter*, setiap karakter diganti dengan kode 5-bit digital - diwakili oleh 5 lubang dalam pita kertas berlubang - yang biasa digunakan dengan mesin telex. Ini umumnya dikenal sebagai ITA2 atau kode *Baudot-Murray*. Kode digital juga dapat diwakili oleh serangkaian '1' dan '0', di mana 1 mewakili keberadaan lubang dan 0 mewakili tidak adanya lubang.

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

$$plaintext + key = ciphertext$$

ciphertext + key = plaintext

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

2.10 Stream Cipher

Stream cipher adalah cipher kunci simetris di mana digit plaintext digabungkan dengan stream digit cipher *pseudorandom (keystream)*. Dalam *stream cipher*, setiap digit plaintext dienkripsi satu per satu dengan digit *keystream* yang sesuai, untuk memberikan digit stream *ciphertext*. Karena enkripsi setiap digit tergantung pada kondisi sandi saat ini, enkripsi juga dikenal sebagai sandi negara. Dalam praktiknya, digit biasanya sedikit dan operasi penggabungan adalah *eksklusif-or (XOR)* (Wikipedia, 2019a).

Keystream pseudorandom biasanya dihasilkan secara serial dari nilai *seed* acak menggunakan register shift digital. Nilai seed berfungsi sebagai kunci kriptografi untuk mendekripsi aliran *ciphertext*. *Cipher stream* mewakili pendekatan yang berbeda untuk enkripsi simetris dari cipher blok. Cipher blok beroperasi pada blok besar digit dengan transformasi tetap dan tidak berubah. Perbedaan ini tidak selalu jelas: dalam beberapa mode operasi, blok *cipher primitif* digunakan sedemikian rupa sehingga bertindak efektif sebagai *stream cipher*. *Cipher stream* biasanya dieksekusi pada kecepatan yang lebih tinggi

daripada cipher blok dan memiliki kompleksitas perangkat keras yang lebih rendah. Namun, *stream cipher* dapat rentan terhadap masalah keamanan serius jika digunakan secara tidak benar khususnya, kondisi awal yang sama tidak boleh digunakan dua kali (Weerasinghe, 2013).

2.11 *One Time Pad (OTP)*

Algoritma *One Time Pad* (OTP) merupakan algoritma berjenis *Symetric key* yang artinya bahwa kunci yang digunakan untuk melakukan enkripsi dan dekripsi merupakan kunci yang sama. Dalam proses enkripsi, algoritma ini menggunakan cara*stream Cipher* yang berasal dari hasi XOR antara *bitplaintext* dan *bitkey*. Pada metode ini *plain text* diubah kedalam kode ASCII dan kemudian dikenakan operasi XOR terhadap kunci yang sudah diubah ke dalam kode ASCII. (Hamokwarong, 10 :2014)

One-time pad adalah salah satu stream *Cipher* klasik yang secara matematis terbukti sempurna aman. *Cipher* teksnya tidak mungkin dapat dipecahkan. Keamanan algoritma *one-time pad* terletak pada penggunaan barisan bilangan acak sejati (*trully random*) sebagai kunci enkripsi, panjang kunci sama dengan panjang pesan dan tidak ada perulangan kunci sebagaimana pada pada *Vernam Cipher* atau *Vigenere Cipher*. (Munir, 2014)

Sayangnya *one-time pad* tidak dapat diimplementasikan secara praktis sebab pembangkitan bilangan acak sejati tidak dapat diulang kembali di sisi penerima pesan.Oleh karena itu kunci (*pad*) harus dikirim melalui saluran komunikasi yang kedua (misalnya melalui kurir), sayangnya saluran kedua itu

umumnya lambat dan ongkosnya mahal.One-time pad masih dapat diterapkan namun kunci yang berupa barisan bilangan acak diganti dengan barisan bilangan semi-acak (*pseudo-random*) dengan syarat barisan kunci itu tidak boleh berulang. (Munir, 2014)

2.12 Algoritma

Penyelesaian permasalahan dengan menggunakan alat bantu system computer paling tidak akan melibatkan lima tahapan, yaitu:

1. Analisis masalah
2. Merancang algoritma
3. Membuat program computer
4. Menguji hasil program computer
5. Dokumentasi

Poin kedua menerangkan bahwa dalam perancangan sebuah system computer dibutuhkan adanya perancangan algoritma.Sehingga setelahnya dapat dilanjutkan ke tahap-tahap berikutnya hingga dokumentasi.

Algoritma adalah Sistim kerja komputer memiliki brainware, hardware, dan software. Tanpa salah satu dari ketiga sistim tersebut, komputer tidak akan berguna. Kita akan lebih fokus pada software komputer. Software terbangun atas susunan program (silahkan baca mengenai pengertian program) dan syntax (cara penulisan/pembuatan program). Untuk menyusun program atau syntax, diperlukannya langkah-langkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan suatu software.

Maka, Algoritma berperan penting dalam penyusunan program atau syntax tersebut.

Pengertian Algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu. Dalam dunia komputer, Algoritma sangat berperan penting dalam pembangunan suatu software. Dalam dunia sehari-hari, mungkin tanpa kita sadari Algoritma telah masuk dalam kehidupan kita.

Pengertian Algoritma diartikan sebagai susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu.Algoritma adalah kunci dari bidang ilmu komputer, dan pada dasarnya setiap hari kita melakukan aktivitas algoritma. Kata algoritma berasal dari sebutan Algorizm (Abu Abdullah Muhammad Ibn Musa Al Khwarizmi, ahli matematika Uzbeki

a. Algoritma adalah urutan langkah-langkah berhingga untuk memecahkan masalah logika atau matematika
b. Algoritma adalah logika, metode dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan.
c. Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis.
d. Algoritma adalah urutan logis pengambilan keputusan untuk pemecahan masalah.

Pembuatan algoritma harus selalu dikaitkan dengan:

a. Kebenaran algoritma
b. Kompleksitas (lama dan jumlah waktu proses dan penggunaan memori)

Kriteria Algoritma yang baik:

a. Tepat, benar, sederhana, standar dan efektif
b. Logis, terstruktur dan sistematis
c. Semua operasi terdefinisi
d. Semua proses harus berakhir setelah sejumlah langkah dilakukan
e. Ditulis dengan bahasa yang standar dengan format pemrograman agar mudah untuk diimplementasikan dan tidak menimbulkan arti ganda.

2.13 Kekhawatiran Kriptografi

Penyerang dapat menghindari kriptografi, meretas ke dalam komputer yang bertanggung jawab atas enkripsi dan dekripsi data, dan mengeksploitasi implementasi yang lemah, seperti penggunaan kunci default. Namun, kriptografi mempersulit penyerang untuk mengakses pesan dan data yang dilindungi oleh algoritma enkripsi.

Tumbuhnya kekhawatiran tentang kekuatan pemrosesan komputasi kuantum untuk memecah standar enkripsi kriptografi saat ini membuat Institut Standar dan Teknologi Nasional mengeluarkan seruan untuk makalah di antara komunitas matematika dan ilmiah pada 2016 untuk standar kriptografi kunci publik baru. Tidak seperti sistem komputer saat ini, komputasi kuantum menggunakan bit kuantum (*qubit*) yang dapat mewakili 0s dan 1s dan karenanya melakukan dua perhitungan sekaligus. Sementara komputer kuantum skala besar mungkin tidak dibangun dalam dekade berikutnya, infrastruktur yang ada membutuhkan standarisasi algoritma yang dikenal dan dipahami publik yang

menawarkan pendekatan yang aman, menurut *NIST*. Batas waktu pengiriman adalah pada November 2017, analisis proposal diperkirakan akan memakan waktu tiga hingga lima tahun (TechTarget, 2019).

2.14 *Unified Modeling Language* (UML)

2.14.1 Pengenalan UML

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisis dan desain yang berisi sintak dalam memodelkan sistem secara visual (Haviluddin : 2015). Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan oleh banyak pihak adalah *DataFlow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau struktur, kemudian juga ada *State Transition Diagram* (STD) yang digunakan untuk memodelkan *real time* (waktu nyata).

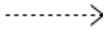
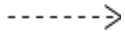
Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML).

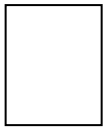
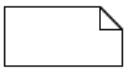
2.14.2 *Use Case Diagram*

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram *use case diagram* (Haviluddin : 2015 : 4).

Tabel 2.1 Simbol *Use Case Diagram*

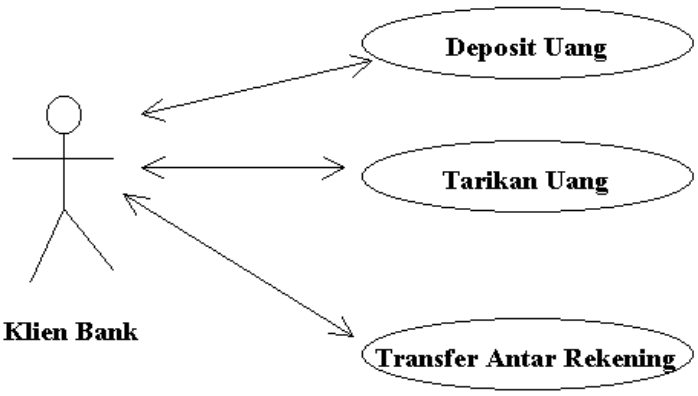
NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Aktor adalah orang proses,atau system lain yang berinteraksi dengan system informasi yang akan di buat.jadi meskipun symbol dari actor ialah gambar orang,tapi actor belum tentu merupakan orang biasanya penamaan akror dinamakan menggunakan kata benda di awal frase nama aktor

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*.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan

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

Sumber : (Gellysa Urva, 94 : 2015)

Contoh Use Case Diagram :



Gambar 2.2 Contoh *Use Case Diagram*
Sumber : (Haviluddin : 2015 : 4)

2.14.3 *Activity Diagram*

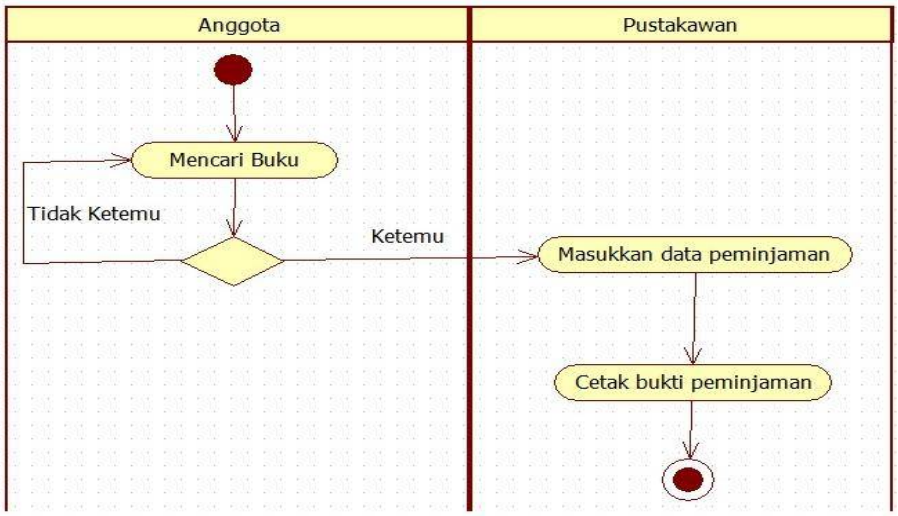
Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau *menu* yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa A.S dan M. Shalahuddin, 2014). (Rinaldi, 2014).

Tabel 2.2 Simbol *ActivityDiagram*

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

Sumber : (Gellysa Urva, 94 : 2015)

Contoh Activity Diagram :



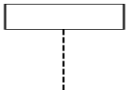
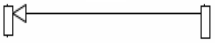
Gambar 2.3 Contoh *Activity Diagram*

Sumber : (Gellysa Urva, 94 : 2015)

2.14.4 *Sequence Diagram*

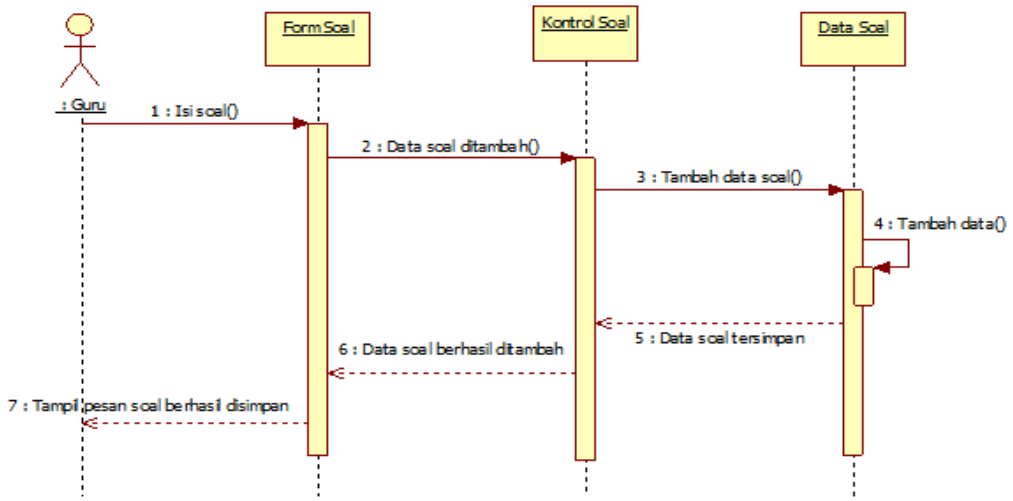
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case.*

Tabel 2.3 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sumber : (Gellysa Urva, 95 : 2015)

Contoh Sequence Diagram :



Gambar 2.4 Contoh *Squence Diagram*
Sumber : (Gellysa Urva, 95 : 2015)

2.14.5 *Class Diagram*

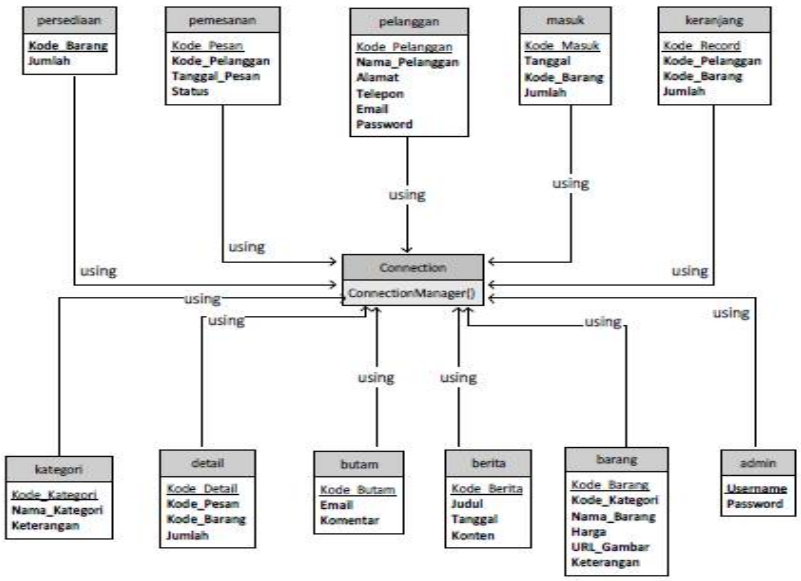
Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas.Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai.Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

Tabel 2.4 Simbol *Class Diagram*

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

Sumber : (Gellysa Urva, 95 : 2015)

Contoh *Class* Diagram :



Gambar 2.5 Contoh *Class Diagram*

Sumber : (Gellysa Urva, 95 : 2015)

2.15 Pengertian Informasi

Secara Etimologi, kata informasi ini berasal dari kata bahasa Perancis kuno *informacion* (tahun 1387) mengambil istilah dari bahasa Latin yaitu *informationem* yang berarti "konsep, ide atau garis besar". Informasi ini merupakan kata benda dari *informare* yang berarti aktivitas dalam "pengetahuan yang dikomunikasikan".

Informasi adalah hasil pemrosesan data yang diperoleh dari setiap elemen sistem menjadi bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan dan berguna (Yulansari, 6 : 2013).

Informasi bisa menjadi fungsi penting dalam membantu mengurangi rasa cemas pada seseorang. Menurut pendapat Notoatmodjo (2018) bahwa semakin

banyak memiliki informasi dapat memengaruhi atau menambah pengetahuan terhadap seseorang dan dengan pengetahuan tersebut bisa menimbulkan kesadaran yang akhirnya seseorang itu akan berperilaku sesuai dengan pengetahuan yang dimilikinya.

Informasi adalah data yang telah diolah melalui proses tertentu menjadi sesuatu yang menambah pengetahuan atau temuan yang mempunyai arti baru bagi pemakainya (Melina, 38 : 2014).

Adapun fungsi-fungsi informasi adalah sebagai berikut:

1. Untuk meningkatkan pengetahuan bagi si pemakai.
2. Untuk mengurangi ketidakpastian dalam proses pengambilan keputusan pemakai.
3. Menggambarkan keadaan yang sebenarnya dari sesuatu hal.Informasi yang berkualitas harus akurat, tepat dan relevan.

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk suara, sinyal, gambar, dan sebagainya.

2.16 Pengertian Visual Studio

Visual Studio .Net merupakan salah satu *tool development Microsoft* yang dapat digunakan untuk membuat aplikasi di lingkungan kerja berbasis sistem operasi *Windows*. *Visual Studio .NET* menyediakan tools bagi para *developer* untuk membangun aplikasi yang berjalan di .*Net Framework* (Safik : 2015 : 2).

Visual Studio (*Beginners All-Purpose Symbolic Instruction Code*) merupakan Bahasa pemrograman *Integrated Development Environment* (IDE), yaitu bahasa pemrograman *visual* yang digunakan untuk membuat program aplikasi atau *software* berbasis sistem operasi *Microsoft Windows*, dengan menggunakan model pemrograman "*Common Object Model* (COM)".

Visual Studio merupakan turunan bahasa pemrograman *STUDIO* yang menawarkan pengembangan perangkat lunak komputer bebasis grafik dengan cepat. Dengan menggunakan bahasa pemrograman VB, para progammer dapat membangun aplikasi dengan menggunakan komponen-komponen yang di sediakan VB.

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

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

Para *programmer* dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh *Microsoft Visual Studio* Program-program yang ditulis dengan *Visual Studio* juga dapat menggunakan *Windows* API, tapi membutuhkan deklarasi fungsi luar tambahan.

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

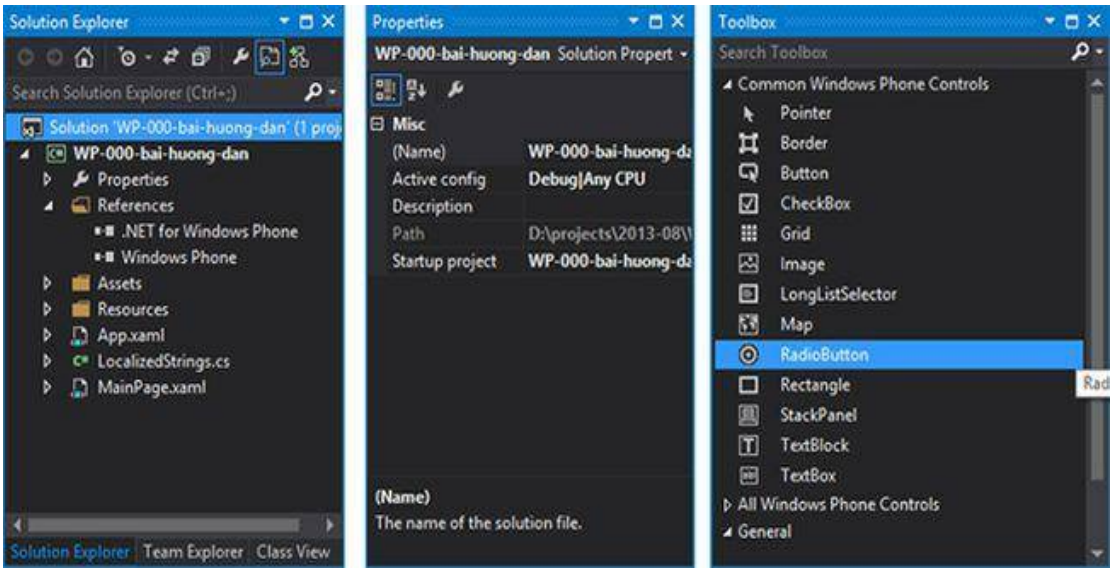
2.16.1 Komponen kerja

Beberapa komponen kerja program *visual Studio 2015* telah ditampilkan sebagai tampilan standard.Masih banyak lagi komponen yang masih tersembunyi sehingga memerlukan perintah tertentu untuk menampilkannya.Kita dapat mengatur komponen di dalam program visual Studio 2015 sesuai dengan yang kita butuhkan. Berikut ini adalah beberapa komponen kerja dari visual Studio 2015adalah :

a. Toolbox

Toolbox adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membuat suatu desain mulai dari tombol *label, pointer, button*, dan lain-lain. Berikut ini adalah gambaran *toolbox* pada *visual Studio 2015* :

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



Gambar 2.6. Tampilan *Toolbox*
Sumber : (Safik : 2015 : 2).

Tabel 2.5 *Toolbox Visual Studio*

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

Nama tombol	Fungsi
Button	Membuat tombol perintah
Combobox	Menambahkan control kotak combo yang merupakan control gabungan antara textbox dan listbox

Sumber : (Safik : 2015 : 2).

2.16.2 Tabel ASCII

ASCII merupakan kepanjangan dari (American Standard Code for Information Interchange), dan pengertian dari ASCII sendiri adalah suatu standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal.

ASCII table

Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex
(nul)	0	0000	0x00	(sp)	32	0040	0x20	@	64	0100	0x40	`	96	0140	0x60
(soh)	1	0001	0x01	!	33	0041	0x21	A	65	0101	0x41	a	97	0141	0x61
(stx)	2	0002	0x02	"	34	0042	0x22	B	66	0102	0x42	b	98	0142	0x62
(etx)	3	0003	0x03	#	35	0043	0x23	C	67	0103	0x43	c	99	0143	0x63
(eot)	4	0004	0x04	$	36	0044	0x24	D	68	0104	0x44	d	100	0144	0x64
(enq)	5	0005	0x05	%	37	0045	0x25	E	69	0105	0x45	e	101	0145	0x65
(ack)	6	0006	0x06	&	38	0046	0x26	F	70	0106	0x46	f	102	0146	0x66
(bel)	7	0007	0x07	'	39	0047	0x27	G	71	0107	0x47	g	103	0147	0x67
(bs)	8	0010	0x08	(	40	0050	0x28	H	72	0110	0x48	h	104	0150	0x68
(ht)	9	0011	0x09	)	41	0051	0x29	I	73	0111	0x49	i	105	0151	0x69
(nl)	10	0012	0x0a	*	42	0052	0x2a	J	74	0112	0x4a	j	106	0152	0x6a
(vt)	11	0013	0x0b	+	43	0053	0x2b	K	75	0113	0x4b	k	107	0153	0x6b
(np)	12	0014	0x0c	,	44	0054	0x2c	L	76	0114	0x4c	l	108	0154	0x6c
(cr)	13	0015	0x0d	-	45	0055	0x2d	M	77	0115	0x4d	m	109	0155	0x6d
(so)	14	0016	0x0e	.	46	0056	0x2e	N	78	0116	0x4e	n	110	0156	0x6e
(si)	15	0017	0x0f	/	47	0057	0x2f	O	79	0117	0x4f	o	111	0157	0x6f
(dle)	16	0020	0x10	0	48	0060	0x30	P	80	0120	0x50	p	112	0160	0x70
(dc1)	17	0021	0x11	1	49	0061	0x31	Q	81	0121	0x51	q	113	0161	0x71
(dc2)	18	0022	0x12	2	50	0062	0x32	R	82	0122	0x52	r	114	0162	0x72
(dc3)	19	0023	0x13	3	51	0063	0x33	S	83	0123	0x53	s	115	0163	0x73
(dc4)	20	0024	0x14	4	52	0064	0x34	T	84	0124	0x54	t	116	0164	0x74
(nak)	21	0025	0x15	5	53	0065	0x35	U	85	0125	0x55	u	117	0165	0x75
(syn)	22	0026	0x16	6	54	0066	0x36	V	86	0126	0x56	v	118	0166	0x76
(etb)	23	0027	0x17	7	55	0067	0x37	W	87	0127	0x57	w	119	0167	0x77
(can)	24	0030	0x18	8	56	0070	0x38	X	88	0130	0x58	x	120	0170	0x78
(em)	25	0031	0x19	9	57	0071	0x39	Y	89	0131	0x59	y	121	0171	0x79
(sub)	26	0032	0x1a	:	58	0072	0x3a	Z	90	0132	0x5a	z	122	0172	0x7a
(esc)	27	0033	0x1b	;	59	0073	0x3b	[	91	0133	0x5b	{	123	0173	0x7b
(fs)	28	0034	0x1c	<	60	0074	0x3c	\	92	0134	0x5c	|	124	0174	0x7c
(gs)	29	0035	0x1d	=	61	0075	0x3d	]	93	0135	0x5d	}	125	0175	0x7d
(rs)	30	0036	0x1e	>	62	0076	0x3e	^	94	0136	0x5e	~	126	0176	0x7e
(us)	31	0037	0x1f	?	63	0077	0x3f	_	95	0137	0x5f	(del)	127	0177	0x7f

Gambar 2.7 Tabel ASCII

Sumber (https://www.asciitable.com)

BAB III

METODE PENELITIAN

3.1 Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan bertujuan untuk mendapatkan hasil berdasarkan perancangan bagaimana algoritma kriptografi dengan teknik XNOR. Beberapa tahapan yang dilakukan dalam melakukan pengumpulan data yaitu:

1. Pengumpulan Data

   Pengumpulan data adalah pencarian terhadap susuatu kerena ada perhatian dan keinginan terhadap hasil suatu aktivitas. Metode pengumpulan data dalam penulisan ini dibagi menjadi 3, yaitu :

   1. Wawancara (*Interview*).

      Wawancara ini dilakukan dengan cara mengadakan komunikasi langsung dengan dosen pengampu mata kuliah kemanan data di Universitas Pembangunan Pancabudi Medan yang dapat memberikan informasi dan data-data yang diperoleh mengenai kemanaan data dan *Stream chiper*.

   2. Pengamatan (*Observation*)

      Pengamatan dilakukan dengan cara melihat hasil yang dikeluarkan oleh program aplikasi dan perhitungan manual. Program aplikasi berfungi untuk melakukan proses enkripsi dan deskripsi dan

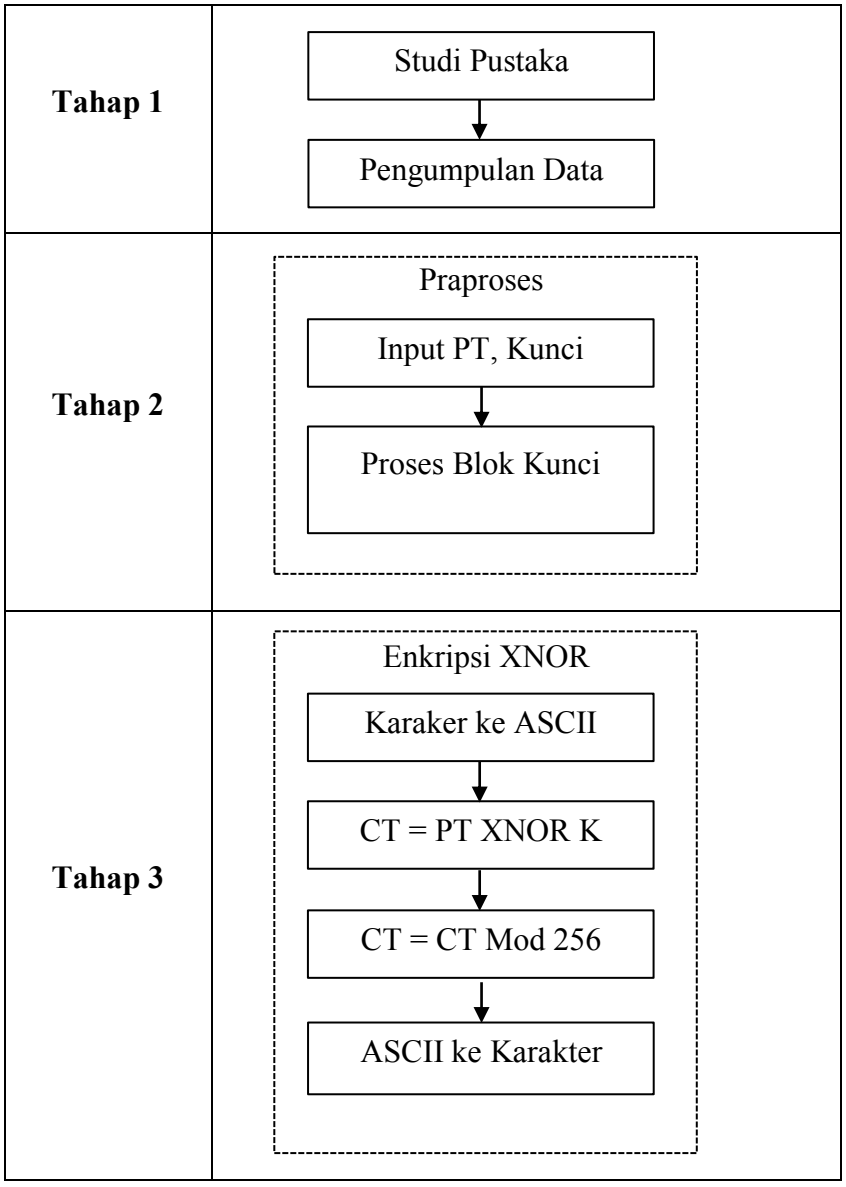
mendapatkan nilai yang benar antara perhitungan manual dan hasil program aplikasi.

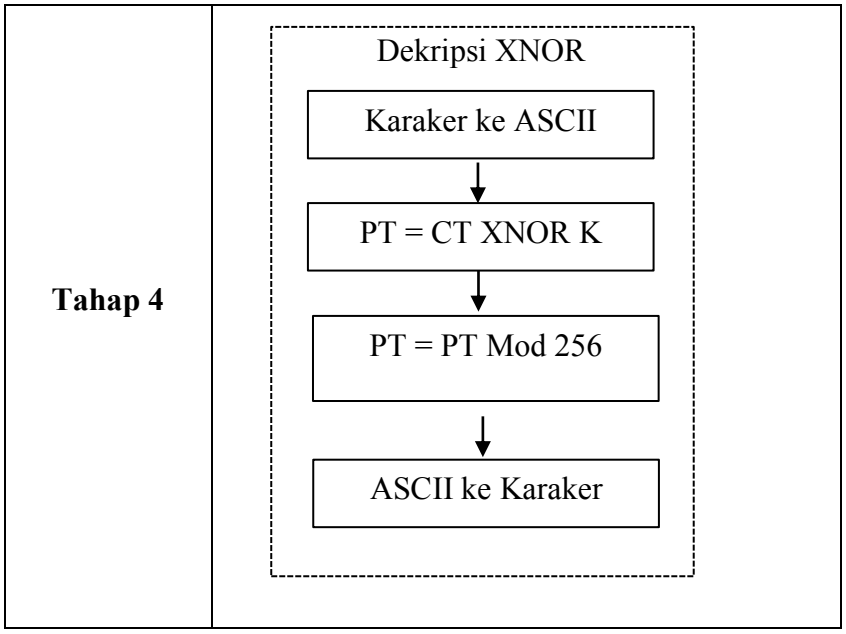
3. Penelitian Kepustakaan (*Library Research*)

   Merupakan cara untuk mencari referensi dengan mengumpulkan bahan-bahan pustaka yang dilakukan di perpustakaan kampus, maupun perpustakaan umum, juga melakukan pencarian lewat internet, dengan mengunjungi situs-situs seperti *google Book online* yang dapat membantu pembahasan materi.

3.2 Tahapan Penelitian

Kegiatan tahap penelitian merupakan suatu proses memperoleh dan mendapatkan suatu pengetahuan atau memecahkan permasalahan yang dihadapi,yang dilakukan secara ilmiah sistematis dan logis, Penelitan terdiri dari beberapa sekema tahapan. Penelitian ini dilakukan transformasi plaintext ke ciphertext dengan teknik XNOR. Algoritma kriptografi XNOR akan dirancang dan diuji coba dengan secara berulang agar hasil yang dihasilkan sesuai dengan yang diingkan dan menghasilkan nilai yang sesuai dengan perhitungan manual. Tahapan yang dilakukan dalam penelitian ini terbagi atas:


Tahap 1
Studi Pustaka
Pengumpulan Data
Tahap 2
Praproses
Input PT, Kunci
Proses Blok Kunci
Tahap 3
Enkripsi XNOR
Karaker ke ASCII
CT = PT XNOR K
CT = CT Mod 256
ASCII ke Karakter



3.3 Rancangan Penelitian

Rancangan penelitian diartikan sebagai kerangka kerja metode dan teknik yang dipilih untuk menyatukan berbagai komponen penelitian dengan cara yang cukup akurat sehingga masalah penelitian ditangani secara efisien. Rancangan ini menyajikan informasi tentang bagaimana metode stream cipher untuk dilakukan pada penelitian ini. Setiap tahap dilakukan untuk mendapatkan kemajuan penelitian. Fungsi rancangan penelitian adalah untuk menyakinkan bahwa bukti yang diperoleh memungkinkan untuk secara efektif mengatasi masalah penelitian sejelas mungkin.

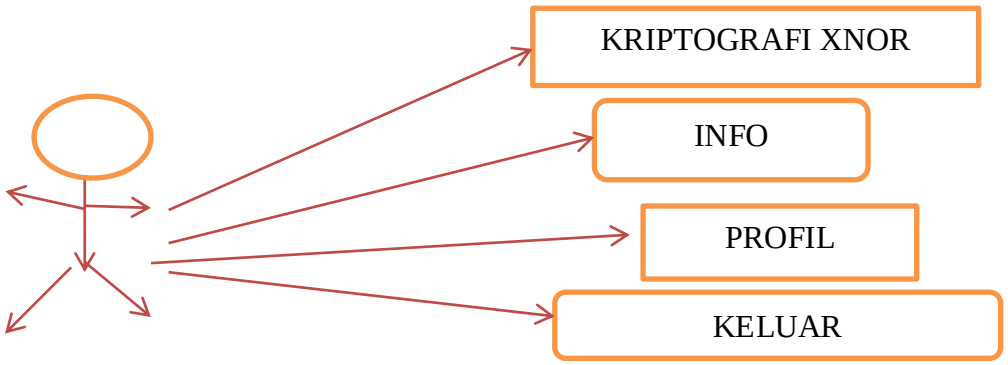
Jenis masalah penelitian yang dihadapi akan menentukan desain penelitian dan bukan sebaliknya. Variabel, alat yang ditunjuk untuk mengumpulkan informasi, bagaimana alat akan digunakan untuk mengumpulkan dan menganalisis data dan faktor-faktor lain diputuskan dalam desain penelitian berdasarkan teknik

penelitian yang diputuskan. Desain penelitian yang berdampak biasanya menciptakan bias minimum dalam data dan meningkatkan kepercayaan pada informasi penelitian yang dikumpulkan dan dianalisis. Desain penelitian yang menghasilkan margin kesalahan terkecil dalam penelitian eksperimental dapat disebut-sebut sebagai yang terbaik.

Pada bagian ini akan dilakukan perancangan penelitian untuk menjelaskan setiap keadaan dan bagian-bagian yang berfungsi untuk melengkapi kegiatan pemakai mengenai gambaran yang jelas tentang perancangan sistem yang akan dibuat serta di bangun.

3.3.1 Use Case Diagram

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *User* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Berikut ini adalah perancangan *Use Case* untuk admin dari algoritma kriptografi XNOR.



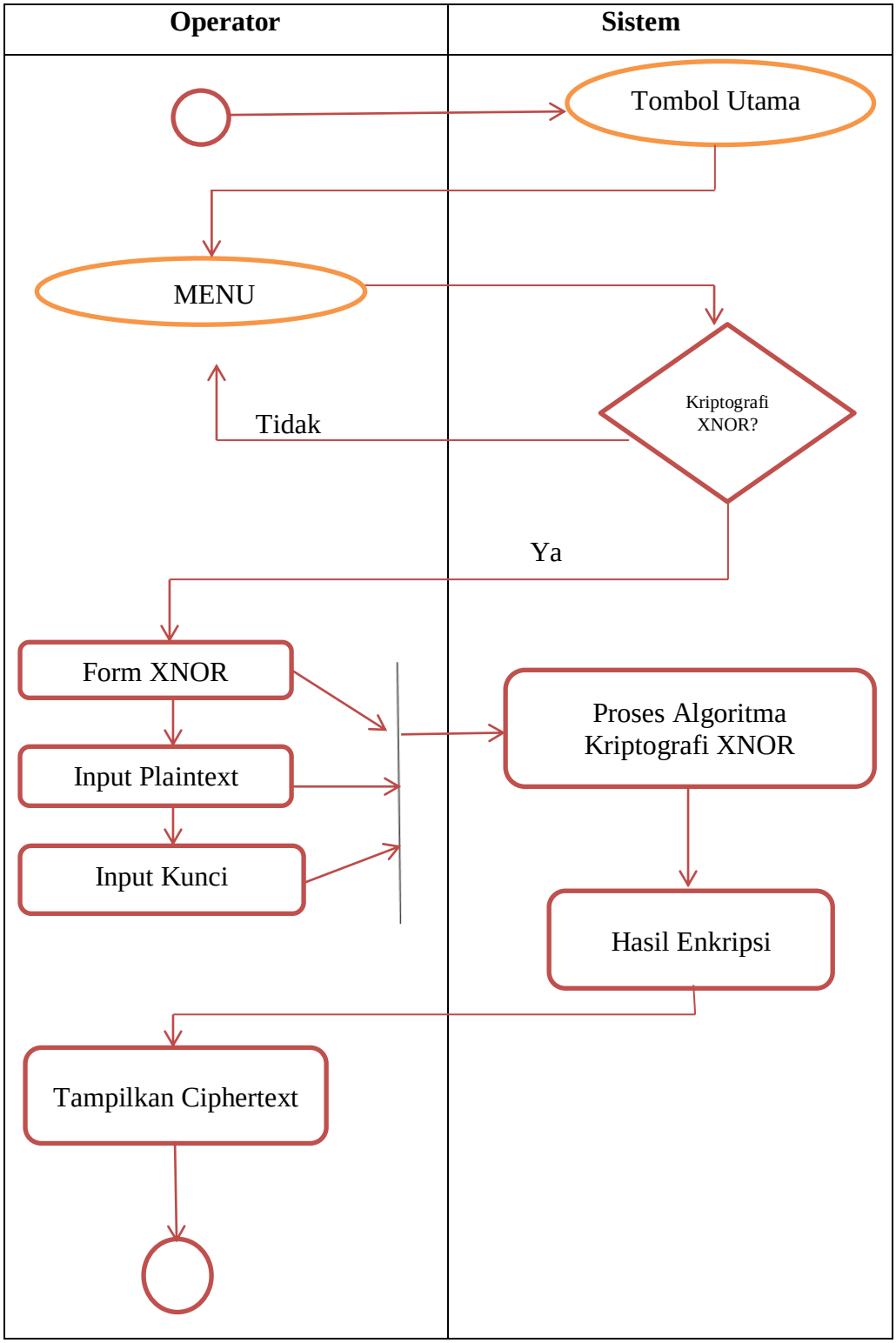
Gambar 3.1 Use Case Diagram

3.3.2 Activity Diagram

Activity diagram atau diagram aktivitas adalah bentuk visual dari alur kerja yang berisi aktivitas dan tindakan ,yang juga dapat berisi pilihan,atau pengulangan.dalam *unified modeling language* (UML), diagram aktivitis dibuat untuk menjelaskan aktivitas computer maupun alur aktivitas dalam organisasi.selain itu diagram aktivitas juga menggambarkan alur control secara garis besar.

Diagram aktivitas memiliki komponen dengan bentuk tertentu,dihubungkan dengan tanda panah.panah tersebut mengarahkan urutan aktivitas yang terjadi ,dari awal sampai akhir.yang perlu di perhatikan yaitu diagram aktivitas bukan menggambarkan aktivitas system yang dilakukan aktor,tetapi menggambarkan aktivitas yang dapat dilakuka oleh system.

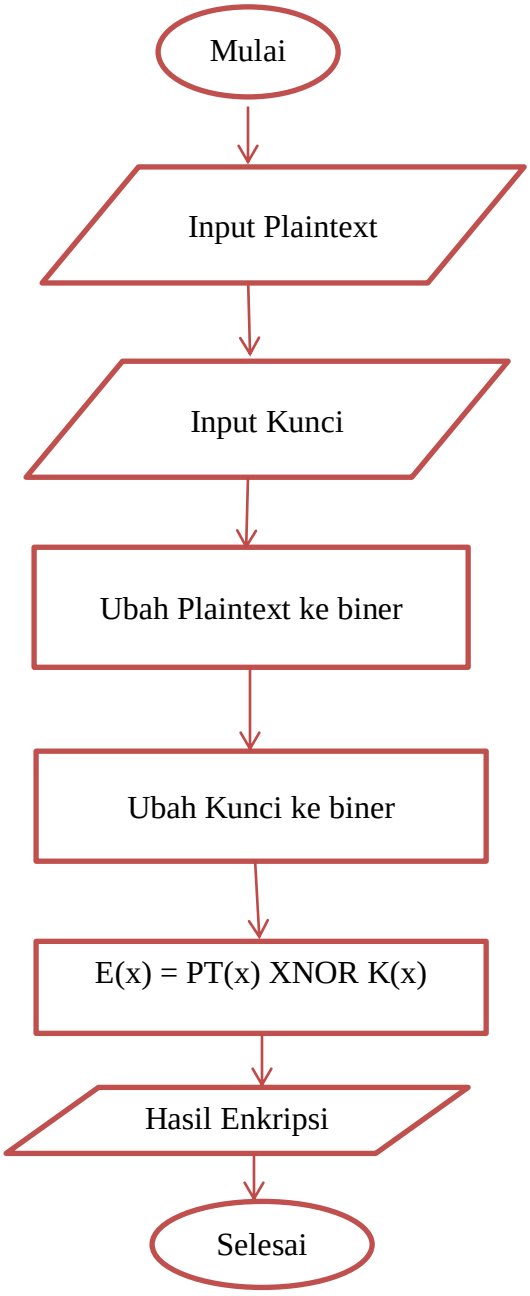
Pada pembahasan ini *activity diagram* akan menggambarkan alur kerja dari sistem, untuk *Activity diagram* dari kriptrografi simetris dengan menggunakan algoritma kriptografi XNOR. Gambar berikut ini akan menjelaskan *activity* gambar diagram tersebut.



Gambar 3.2 Activity Diagram

3.3.3 Flowchart Enkripsi

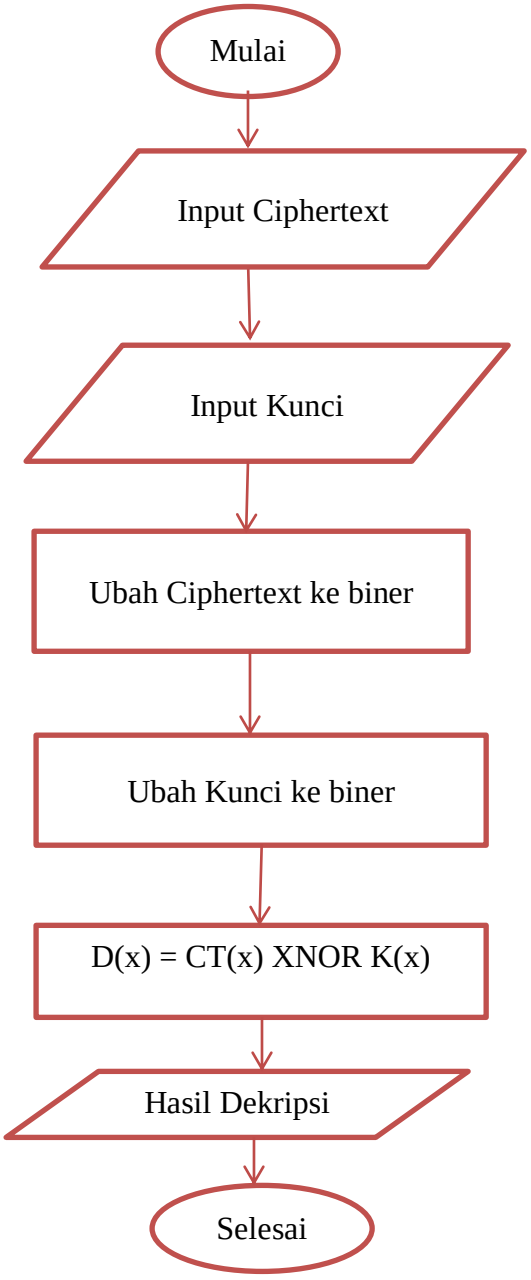
Flowchart akan menjelaskan sebuah alur dari proses enkripsi di dalam metode stream cipher dengan algoritma kriptografi XNOR. Perancangan flowchart enkripsi dapat dilihat pada gambar berikut ini.



Gambar 3.3 Flowchart enkripsi kriptografi XNOR

3.3.4 Flowchart Dekripsi

Flowchart ini akan memberitahukan alur dari proses dekripsi dengan metode stream cipher dengan algoritma kriptografi XNOR. Perancangan flowchart dekripsi bisa dilihat pada gambar berikut ini.



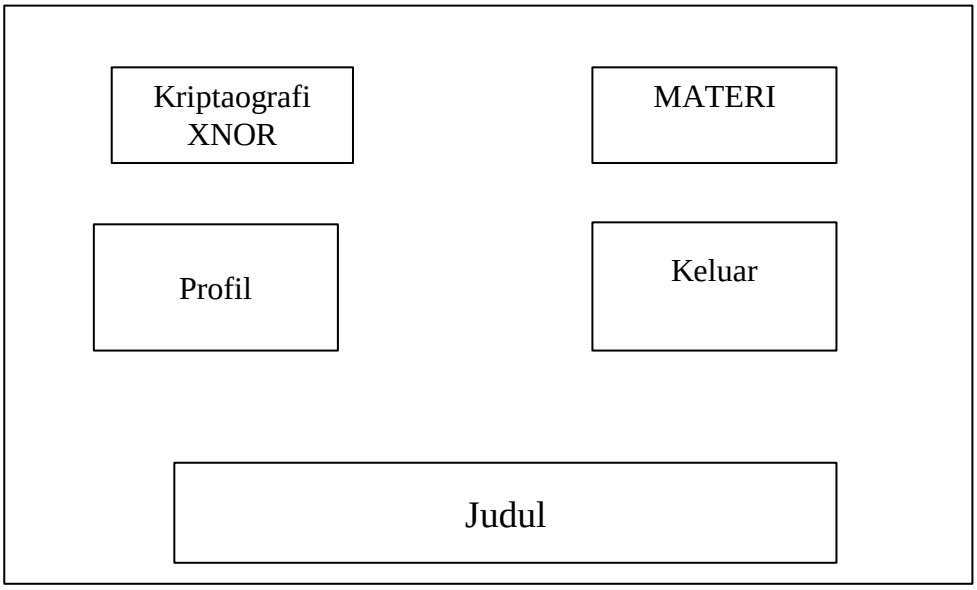
Gambar 3.4 Flowchart Kriptografi XNOR

3.4 Desain Antarmuka

Desain antarmuka adalah untuk membuat interaksi pemrograman sesederhana dan seefisien mungkin. Pemrograman akan dilakukan dengan menggunakan Microsoft Visual Basic.Net 2010. Desain antarmuka ini terbagi menjadi beberapa tahap yang memiliki satu buah menu utama yang berfungsi sebagai pembuka menu yang ada didalamnya. Berikut ini merupakan tahapan dari desain antarmuka program aplikasi algoritma kriptografi XNOR.

3.4.1 Antarmuka Menu Utama

Antarmuka menu utama adalah bagian menu yang pertama sekali ditampilkan pada saat progam aplikasi dijalankan. Gambar berikut ini adalah perancangan menu utama yang terdiri dari lima buah sub-menu.



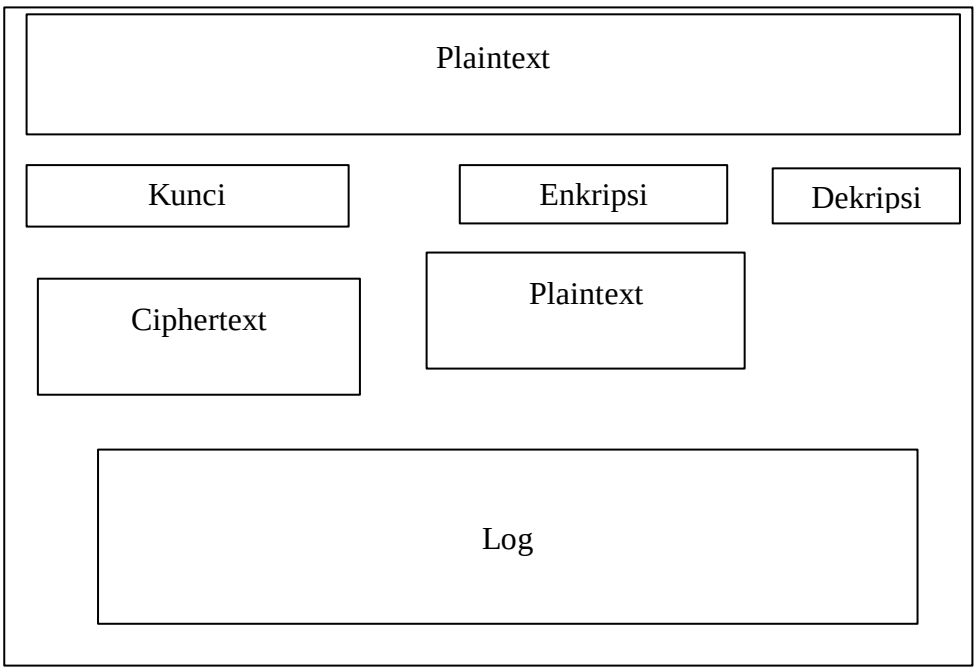
Gambar 3.5 Tampilan Menu Utama

Antarmuka menu ini memiliki berapa sub-menu antara lain:

1. Judul
2. Kriptografi XNOR
3. Info
4. Profil
5. Keluar

3.4.2 Antarmuka Menu Kriptografi XNOR

Antarmuka menu kriptografi XNOR adalah bagian utama dari program aplikasi kriptografi XNOR. Pada menu ini terdapat beberapa bagian yang menjadi input, proses, output dan riwayat perhitungan lengkap dari plaintext ke ciphertext dan juga ciphertext ke plaintext kembali. Gambar 3.6 adalah tampilan menu ini.



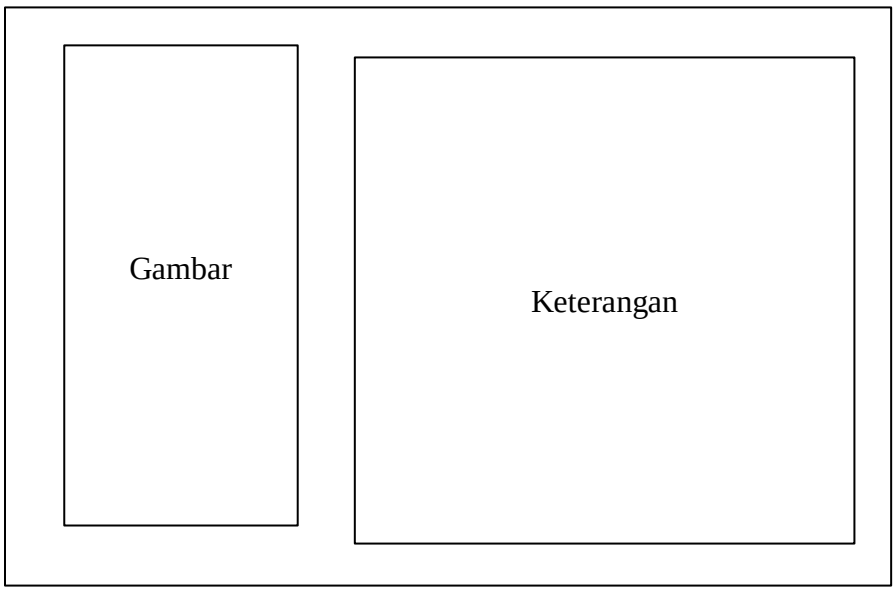
Gambar 3.6 Tampilan Menu Kriptografi XNOR

Antarmuka menu kriptografi XNOR memiliki 6 bagian antara lain:

1. Plaintext
2. Kunci
3. Ciphertext
4. Tombol Enkripsi
5. Tombol Deskripsi
6. Logo

3.4.3 Antarmuka Menu Materi

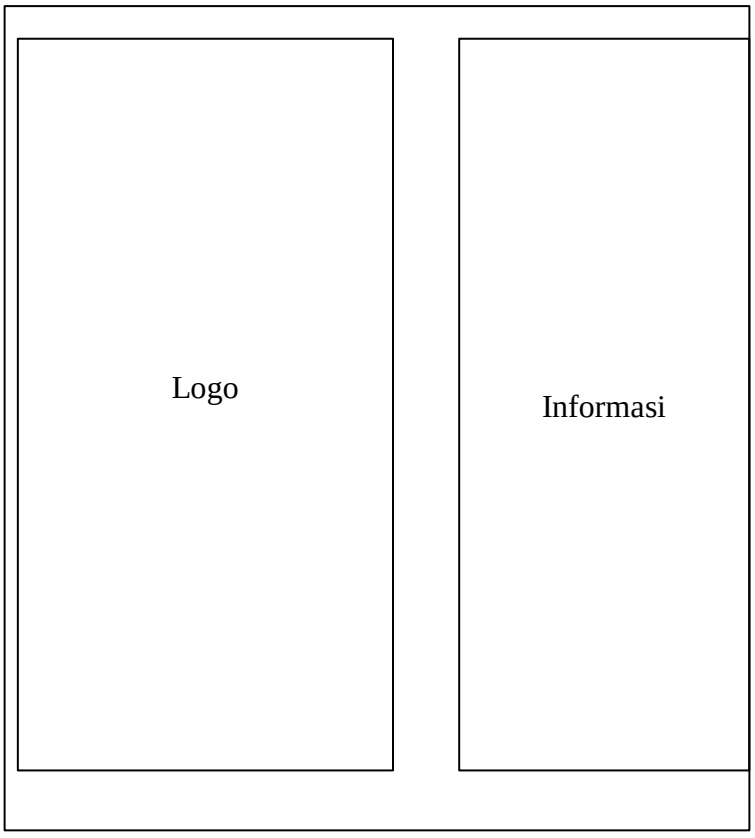
Antarmuka menu materi menampilkan informasi tentang algoritma kriptografi XNOR. Menu ini memiliki dua buah objek, yaitu objek gambar dan keterangan. Gambar berikut ini adalah perancangan menu Info.



Gambar 3.7 Tampilan Menu Materi

3.4.4 Antarmuka Menu Profil

Antarmuka menu profil akan menampilkan informasi tentang penulis. Pada menu ini akan ditampilkan logo dari Universitas Pembangunan Panca Budi. Menu ini terdiri dari dua objek, yaitu logo dan informasi. Gambar berikut ini adalah hasil tampilan dari menu profil.



Gambar 3.8 Tampilan Menu Profil

BAB IV

HASIL DAN PEMBAHASAN

Rancang bangun stream chiper merupakan tahapan menerapkan hasil implementasi dan pembuatan program aplikasi algoritma kriptografi XNOR. bangunan ini juga menjelaskan berbagai macam bagian dari komponen yang terlibat pada topik stream cipher. Pada bagian pembahsan ini dibedakan kepada dua jenis yaitu bangunan algoritma dan bangunan antarmuka.

Penelitian ini menggunakan beberapa jenis parameter yaitu parameter input dan parameter output yang digunakan untuk membangun program aplikasi algoritma kriptografi XNOR tersebut. Beberapa variabel digunakan untuk menunjang hasil proses enkripsi dan dekripsi.

4.1 Spesifikasi Sistem

Penelitian ini merupakan penelitian yang berfokus pada pengembangan ilmu kriptografi substitusi, dalam hal ini adalah algoritma kriptografi XNOR. Selain membutuhkan data penelitian yang bersumber dari internet, dibutuhkan juga perangkat pendukung agar penelitian ini dapat diterapkan dengan baik dan benar. Perangkat pendukung yang dibutuhkan dibagi menjadi dua yaitu perangkat keras dan perangkat lunak. Adapun spesifikasi perangkat keras dan perangkat lunak tersebut dapat dilihat pada bagian selanjutnya.

4.1.1 Spesifikasi Perangkat Keras

Penerapan algoritma kriptografi XNOR pada metode kriptografi stream cipher sudah pasti akan membutuhkan perangkat keras sebagai media fisik sebagai sarana pendukung utama. Berikut ini adalah spesifikasi perangkat keras yang digunakan pada penelitian ini.

Tabel 4.1 Spesifikasi perangkat keras

No.	Nama Komponen	Spesifikasi
1	Processor	Intel Core i5 2.40 GHz
2	RAM	2GB
3	Hardisk	320 GB
4	MONITOR	14 inc

4.1.2 Spesifikasi Perangkat Lunak

Perangkat lunak juga harus wajib ada untuk mendukung kerja dari perangkat keras tersebut. Hal ini digunakan sebagai media antara manusia dan alat dalam mendukung implementasi penelitian ini. Kebutuhan akan perangkat lunak sebagai sarana non-fisik sangat menunjang hasil kerja. Berikut ini adalah spesifikasi perangkat lunak yang digunakan pada penelitian ini.

Tabel 4.2 Spesifikasi perangkat lunak

No.	Nama Komponen	Spesifikasi
1	Sistem Operasi	Windows 7 32 Bit
2	IDE Pemrograman	Microsoft Visual Basic.NET 2010
3	Tangkap Gambar	Snipping Tool
4	Data Editor	Microsoft Excel

4.2 Rancang Bangun Antarmuka

Rancang bangun antarmuka algoritma kriptografi XNOR ini memiliki beberapa bagian yang dapat secara terpisah dan memiliki hasil yang berbeda-beda. Antarmuka ini dibuat menggunakan Microsoft Visual Basic.Net 2010.

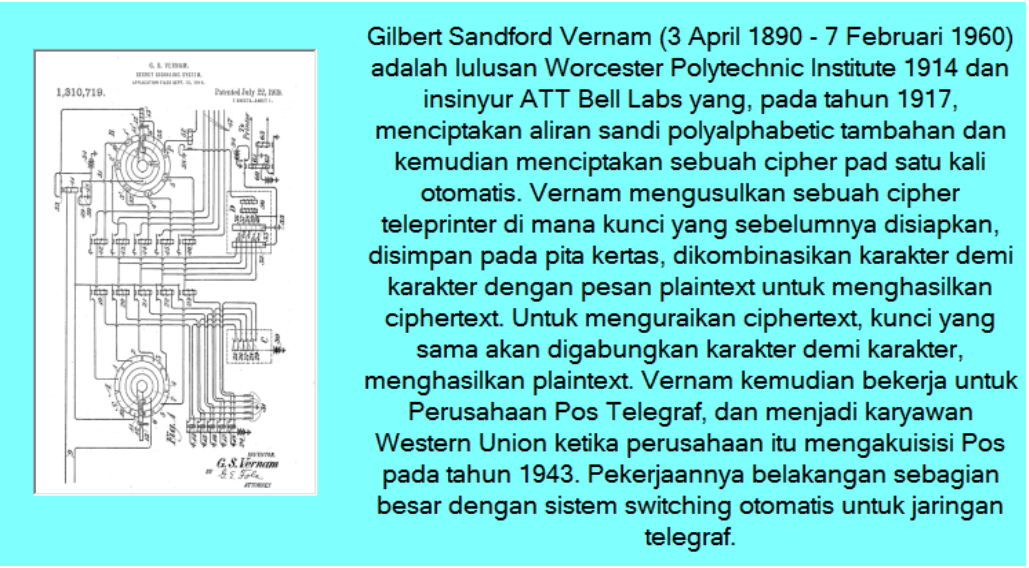
4.2.1 Halaman Menu Utama

Halaman menu utama adalah tampilan yang pertama sekali muncul pada saat program aplikasi dijalankan. Pada tampilan ini, ada beberapa menu yang akan dimunculkan untuk mengizinkan pengguna untuk memilih ke bagian mana pengguna tersebut ingin masuk. Halaman ini terdiri dari tiga buah sub-menu dan satu buah tombol untuk keluar dari aplikasi tersebut. Berikut ini adalah hasil tampilan menu utama.

Gambar 4.1 Halaman Menu Utama

4.2.2 Halaman Materi

Halaman materi adalah menu yang menampilkan penjelasan singkat tentang sejarah algoritma kriptografi XNOR. Halaman ini akan menampilkan sebuah gambar dan sebuah keterangan. Gambar berikut adalah hasil tampilan dari halaman info.



Gambar 4.2 Halaman Materi

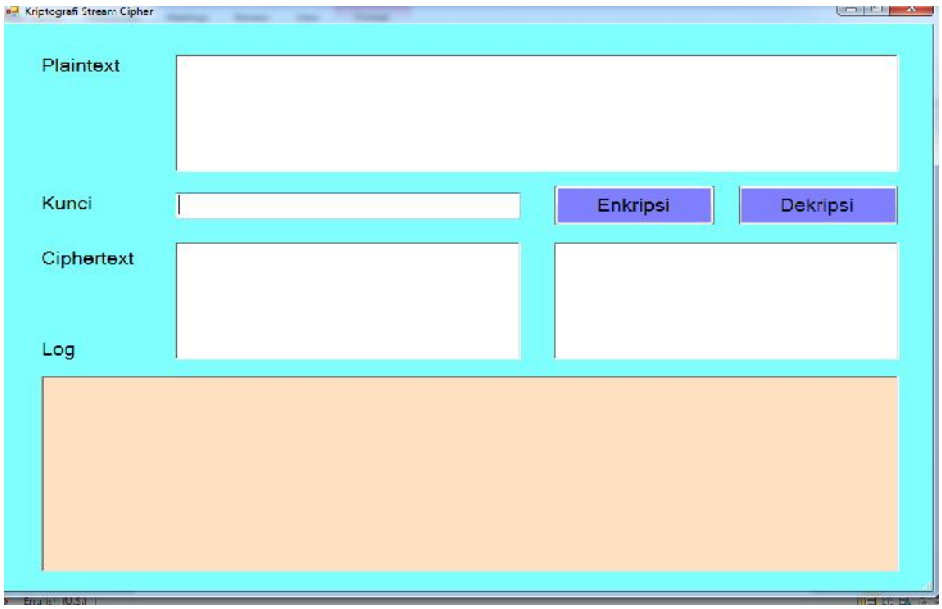
4.2.3 Halaman Profil

Halaman profil adalah tampilan tentang penulis. Halaman ini menampilkan informasi tentang nama, NPM, fakutas dan program studi. Berikut ini adalah tampilan dari halaman About.

Gambar 4.3 Halaman Profil

4.2.4 Halaman Kriptografi XNOR

Halaman ini merupakan proses perhitungan algoritma kriptografi XNOR untuk melakukan proses enkripsi dan dekripsi. Halaman ini memiliki duah buah plaintext, sebuah kunci dan sebuah ciphertext yang dibentuk dari objek textbox. Sementara untuk proses eksekusi enkripsi dan dekripsi, halaman ini memiliki beberapa tombol eksekusi yang dibentuk dari objek button. Gambar berikut ini adalah hasil tampilan dari halaman kriptografi XNOR.



Gambar 4.4 Halaman kriptografi XNOR

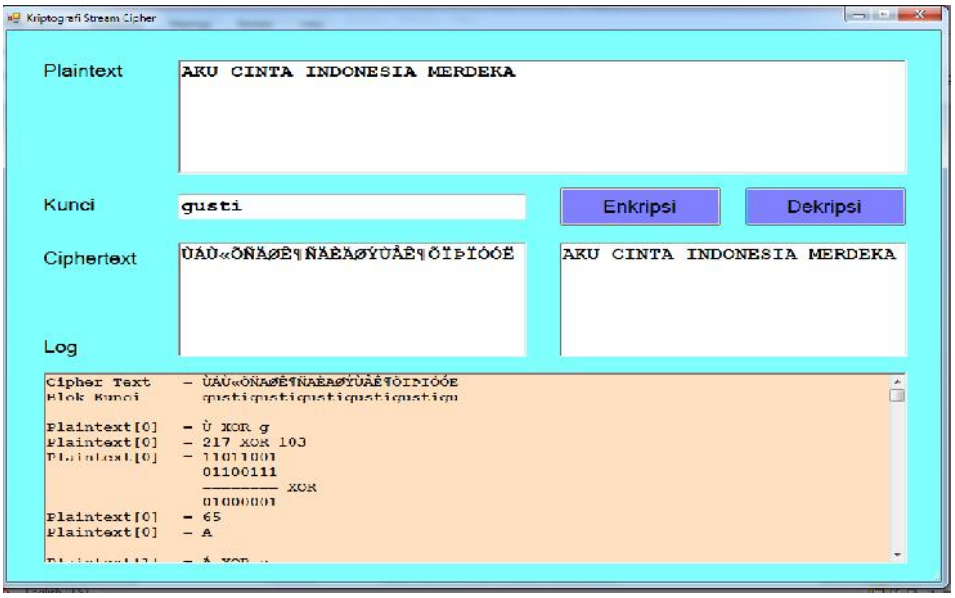
4.2.5 Hasil Perhitungan Algoritma Kriptografi XNOR

Halaman ini berisi tentang hasil tangkapan dari perhitungan yang dilakukan oleh program aplikasi dalam melakukan proses enkripsi dan dekripsi. Ada dua buah input yang harus diisi dalam menentukan hasil yaitu plaintext dan kunci. Kedua textbox ini akan diproses sehingga membentuk blok kunci. Blok kunci akan memiliki panjang sama percis dengan panjang dari plaintext. Setiap karakter pada plaintext akan dilakukan operasi exclusive-or terhadap kunci untuk mendapatkan ciphertext. Proses dekripsi akan melakukan hal yang sama yaitu melakukan proses exclusive-or ciphertext terhadap blok kunci yang sudah ditentukan sebelumnya. Hasil yang benar akan menampilkan bahwa plaintext sebelum proses enkripsi harus sama dengan plaintext yang dihasilkan setelah proses dekirpsi

Gambar berikut ini adalah tampilan dari hasil perhitungan proses enkripsi algoritma kriptografi XNOR.

Gambar 4.5 Halaman enkripsi kriptografi XNOR

Setelah proses enkripsi dilakukan, maka ciphertext akan dilakukan proses dekripsi untuk mengembalikan ke dalam bentuk plaintext seperti pada awalnya. Proses ini akan mengerjakan hal yang sama seperti yang dikerjakan pada proses enkripsi sehingga plaintext diperoleh. Pada textbox plaintext harus menampilkan deretan karakter yang sama percis pada plaintext pertama sekali agar terbukti perhitungan algoritma kriptografi XNOR berjalan dengan baik. Jika ada satu karakter yang tidak memiliki kesamaan dengan plaintext sebelumnya, maka perhitungan ini dinyakatan salah. Gambar berikut ini adalah tampilan dari hasil perhitungan proses dekripsi algoritma kriptografi XNOR.



Gambar 4.6 Halamanan dekripsi kriptografi XNOR

4.3 Hasil Perhitungan

Hasil perhitungan adalah hasil dari uji coba yang menghasilkan perhitungan proses enkripsi dan dekripsi algoritma kriptografi XNOR. Pengujian dilakukan dengan dua cara yaitu manual dan mengguanakan program aplikasi. Hasil dari kedua cara harus menunjukkan keluaran yang sama agar perhitungan dinyatakan benar. Pertama sekali sebelum melakukan perhitungan, ada beberapa tahap yang perlu dilakukan yaitu memberikan nilai pada plaintext dan kunci.

Berikut ini adalah perhitungan lengkap pada proses enkripsi kriptografi XNOR.

Plaintext = AKU CINTA INDONESIA MERDEKA

Kunci = GUSTI

Hasil Enkripsi

```
Plain Text     = AKU CINTA INDONESIA MERDEKA
Blok Kunci     = gustigustigustigustigustigu

Ciphertext[0]  = A XOR g
Ciphertext[0]  = 65 XOR 103
Ciphertext[0]  = 01000001
                 01100111
                 ======== XOR
                 11011001
Ciphertext[0]  = 217
Ciphertext[0]  = Ù

Ciphertext[1]  = K XOR u
Ciphertext[1]  = 75 XOR 117
Ciphertext[1]  = 01001011
                 01110101
                 ======== XOR
                 11000001
Ciphertext[1]  = 193
Ciphertext[1]  = Á

Ciphertext[2]  = U XOR s
Ciphertext[2]  = 85 XOR 115
Ciphertext[2]  = 01010101
                 01110011
                 ======== XOR
                 11011001
Ciphertext[2]  = 217
Ciphertext[2]  = Ù

Ciphertext[3]  =   XOR t
Ciphertext[3]  = 32 XOR 116
Ciphertext[3]  = 00100000
                 01110100
                 ======== XOR
                 10101011
Ciphertext[3]  = 171
```

```
Ciphertext[3]  = «

Ciphertext[4]  = C XOR i
Ciphertext[4]  = 67 XOR 105
Ciphertext[4]  = 01000011
                 01101001
                 ======== XOR
                 11010101
Ciphertext[4]  = 213
Ciphertext[4]  = Õ

Ciphertext[5]  = I XOR g
Ciphertext[5]  = 73 XOR 103
Ciphertext[5]  = 01001001
                 01100111
                 ======== XOR
                 11010001
Ciphertext[5]  = 209
Ciphertext[5]  = Ñ

Ciphertext[6]  = N XOR u
Ciphertext[6]  = 78 XOR 117
Ciphertext[6]  = 01001110
                 01110101
                 ======== XOR
                 11000100
Ciphertext[6]  = 196
Ciphertext[6]  = Ä

Ciphertext[7]  = T XOR s
Ciphertext[7]  = 84 XOR 115
Ciphertext[7]  = 01010100
                 01110011
                 ======== XOR
                 11011000
Ciphertext[7]  = 216
Ciphertext[7]  = Ø

Ciphertext[8]  = A XOR t
Ciphertext[8]  = 65 XOR 116
Ciphertext[8]  = 01000001
                 01110100
                 ======== XOR
                 11001010
Ciphertext[8]  = 202
Ciphertext[8]  = Ê
```

```
Ciphertext[9]  =   XOR i
Ciphertext[9]  = 32 XOR 105
Ciphertext[9]  = 00100000
                 01101001
                 ======== XOR
                 10110110
Ciphertext[9]  = 182
Ciphertext[9]  = ¶

Ciphertext[10]      = I XOR g
Ciphertext[10]      = 73 XOR 103
Ciphertext[10]      = 01001001
                  01100111
                  ======== XOR
                  11010001
Ciphertext[10]      = 209
Ciphertext[10]      = Ñ

Ciphertext[11]      = N XOR u
Ciphertext[11]      = 78 XOR 117
Ciphertext[11]      = 01001110
                  01110101
                  ======== XOR
                  11000100
Ciphertext[11]      = 196
Ciphertext[11]      = Ä

Ciphertext[12]      = D XOR s
Ciphertext[12]      = 68 XOR 115
Ciphertext[12]      = 01000100
                  01110011
                  ======== XOR
                  11001000
Ciphertext[12]      = 200
Ciphertext[12]      = È

Ciphertext[13]      = O XOR t
Ciphertext[13]      = 79 XOR 116
Ciphertext[13]      = 01001111
                  01110100
                  ======== XOR
                  11000100
Ciphertext[13]      = 196
Ciphertext[13]      = Ä

Ciphertext[14]      = N XOR i
Ciphertext[14]      = 78 XOR 105
```

```
Ciphertext[14]      = 01001110
                  01101001
                  ======== XOR
                  11011000
Ciphertext[14]      = 216
Ciphertext[14]      = Ø

Ciphertext[15]      = E XOR g
Ciphertext[15]      = 69 XOR 103
Ciphertext[15]      = 01000101
                  01100111
                  ======== XOR
                  11011101
Ciphertext[15]      = 221
Ciphertext[15]      = Ý

Ciphertext[16]      = S XOR u
Ciphertext[16]      = 83 XOR 117
Ciphertext[16]      = 01010011
                  01110101
                  ======== XOR
                  11011001
Ciphertext[16]      = 217
Ciphertext[16]      = Ù

Ciphertext[17]      = I XOR s
Ciphertext[17]      = 73 XOR 115
Ciphertext[17]      = 01001001
                  01110011
                  ======== XOR
                  11000101
Ciphertext[17]      = 197
Ciphertext[17]      = Å

Ciphertext[18]      = A XOR t
Ciphertext[18]      = 65 XOR 116
Ciphertext[18]      = 01000001
                  01110100
                  ======== XOR
                  11001010
Ciphertext[18]      = 202
Ciphertext[18]      = Ê

Ciphertext[19]      =   XOR i
Ciphertext[19]      = 32 XOR 105
Ciphertext[19]      = 00100000
                  01101001
```

```
                ======== XOR
                10110110
Ciphertext[19]     = 182
Ciphertext[19]     = ¶

Ciphertext[20]     = M XOR g
Ciphertext[20]     = 77 XOR 103
Ciphertext[20]     = 01001101
                01100111
                ======== XOR
                11010101
Ciphertext[20]     = 213
Ciphertext[20]     = Õ

Ciphertext[21]     = E XOR u
Ciphertext[21]     = 69 XOR 117
Ciphertext[21]     = 01000101
                01110101
                ======== XOR
                11001111
Ciphertext[21]     = 207
Ciphertext[21]     = Ï

Ciphertext[22]     = R XOR s
Ciphertext[22]     = 82 XOR 115
Ciphertext[22]     = 01010010
                01110011
                ======== XOR
                11011110
Ciphertext[22]     = 222
Ciphertext[22]     = Þ

Ciphertext[23]     = D XOR t
Ciphertext[23]     = 68 XOR 116
Ciphertext[23]     = 01000100
                01110100
                ======== XOR
                11001111
Ciphertext[23]     = 207
Ciphertext[23]     = Ï

Ciphertext[24]     = E XOR i
Ciphertext[24]     = 69 XOR 105
Ciphertext[24]     = 01000101
                01101001
                ======== XOR
                11010011
```

Ciphertext[24] = 211
Ciphertext[24] = Ó

Ciphertext[25] = K XOR g
Ciphertext[25] = 75 XOR 103
Ciphertext[25] = 01001011
01100111
======== XOR
11010011
Ciphertext[25] = 211
Ciphertext[25] = Ó

Ciphertext[26] = A XOR u
Ciphertext[26] = 65 XOR 117
Ciphertext[26] = 01000001
01110101
======== XOR
11001011
Ciphertext[26] = 203
Ciphertext[26] = Ë

Cipher Text = ÙÁÙ«ÕÑÄØÊ¶ÑÄÈÄØÝÙÅÊ¶ÕÏÞÏÓÓË

Plain Text = AKU CINTA INDONESIA MERDEKA

Berikut ini adalah perhitungan lengkap pada proses dekripsi kriptografi XNOR.

Plaintext = AKU CINTA INDONESIA MERDEKA

Kunci = GUSTI

Hasil Dekripsi

```
Cipher Text    = ÙÁÙ«ÕÑÄØÊ¶ÑÄÈÄØÝÙÅÊ¶ÕÏÞÏÓÓË
Blok Kunci     = gustigustigustigustigustigu

Plaintext[0]   = Ù XOR g
Plaintext[0]   = 217 XOR 103
Plaintext[0]   = 11011001
                 01100111
                 ======== XOR
                 01000001
Plaintext[0]   = 65
Plaintext[0]   = A

Plaintext[1]   = Á XOR u
Plaintext[1]   = 193 XOR 117
Plaintext[1]   = 11000001
                 01110101
                 ======== XOR
                 01001011
Plaintext[1]   = 75
Plaintext[1]   = K

Plaintext[2]   = Ù XOR s
Plaintext[2]   = 217 XOR 115
Plaintext[2]   = 11011001
                 01110011
                 ======== XOR
                 01010101
Plaintext[2]   = 85
Plaintext[2]   = U

Plaintext[3]   = « XOR t
```

```
Plaintext[3]   = 171 XOR 116
Plaintext[3]   = 10101011
                 01110100
                 ======== XOR
                 00100000
Plaintext[3]   = 32
Plaintext[3]   =

Plaintext[4]   = Õ XOR i
Plaintext[4]   = 213 XOR 105
Plaintext[4]   = 11010101
                 01101001
                 ======== XOR
                 01000011
Plaintext[4]   = 67
Plaintext[4]   = C

Plaintext[5]   = Ñ XOR g
Plaintext[5]   = 209 XOR 103
Plaintext[5]   = 11010001
                 01100111
                 ======== XOR
                 01001001
Plaintext[5]   = 73
Plaintext[5]   = I

Plaintext[6]   = Ä XOR u
Plaintext[6]   = 196 XOR 117
Plaintext[6]   = 11000100
                 01110101
                 ======== XOR
                 01001110
Plaintext[6]   = 78
Plaintext[6]   = N

Plaintext[7]   = Ø XOR s
Plaintext[7]   = 216 XOR 115
Plaintext[7]   = 11011000
                 01110011
                 ======== XOR
                 01010100
Plaintext[7]   = 84
Plaintext[7]   = T

Plaintext[8]   = Ê XOR t
Plaintext[8]   = 202 XOR 116
Plaintext[8]   = 11001010
```

```
                  01110100
                  ======== XOR
                  01000001
Plaintext[8]    = 65
Plaintext[8]    = A

Plaintext[9]    = ¶ XOR i
Plaintext[9]    = 182 XOR 105
Plaintext[9]    = 10110110
                  01101001
                  ======== XOR
                  00100000
Plaintext[9]    = 32
Plaintext[9]    =

Plaintext[10]   = Ñ XOR g
Plaintext[10]   = 209 XOR 103
Plaintext[10]   = 11010001
                  01100111
                  ======== XOR
                  01001001
Plaintext[10]   = 73
Plaintext[10]   = I

Plaintext[11]   = Ä XOR u
Plaintext[11]   = 196 XOR 117
Plaintext[11]   = 11000100
                  01110101
                  ======== XOR
                  01001110
Plaintext[11]   = 78
Plaintext[11]   = N

Plaintext[12]   = È XOR s
Plaintext[12]   = 200 XOR 115
Plaintext[12]   = 11001000
                  01110011
                  ======== XOR
                  01000100
Plaintext[12]   = 68
Plaintext[12]   = D

Plaintext[13]   = Ä XOR t
Plaintext[13]   = 196 XOR 116
Plaintext[13]   = 11000100
                  01110100
                  ======== XOR
```

```
                  01001111
Plaintext[13]  = 79
Plaintext[13]  = O

Plaintext[14]  = Ø XOR i
Plaintext[14]  = 216 XOR 105
Plaintext[14]  = 11011000
                  01101001
                  ======== XOR
                  01001110
Plaintext[14]  = 78
Plaintext[14]  = N

Plaintext[15]  = Ý XOR g
Plaintext[15]  = 221 XOR 103
Plaintext[15]  = 11011101
                  01100111
                  ======== XOR
                  01000101
Plaintext[15]  = 69
Plaintext[15]  = E

Plaintext[16]  = Ù XOR u
Plaintext[16]  = 217 XOR 117
Plaintext[16]  = 11011001
                  01110101
                  ======== XOR
                  01010011
Plaintext[16]  = 83
Plaintext[16]  = S

Plaintext[17]  = Å XOR s
Plaintext[17]  = 197 XOR 115
Plaintext[17]  = 11000101
                  01110011
                  ======== XOR
                  01001001
Plaintext[17]  = 73
Plaintext[17]  = I

Plaintext[18]  = Ê XOR t
Plaintext[18]  = 202 XOR 116
Plaintext[18]  = 11001010
                  01110100
                  ======== XOR
                  01000001
Plaintext[18]  = 65
```

```
Plaintext[18]  = A

Plaintext[19]  = ¶ XOR i
Plaintext[19]  = 182 XOR 105
Plaintext[19]  = 10110110
                 01101001
                 ======== XOR
                 00100000
Plaintext[19]  = 32
Plaintext[19]  =

Plaintext[20]  = Õ XOR g
Plaintext[20]  = 213 XOR 103
Plaintext[20]  = 11010101
                 01100111
                 ======== XOR
                 01001101
Plaintext[20]  = 77
Plaintext[20]  = M

Plaintext[21]  = Ï XOR u
Plaintext[21]  = 207 XOR 117
Plaintext[21]  = 11001111
                 01110101
                 ======== XOR
                 01000101
Plaintext[21]  = 69
Plaintext[21]  = E

Plaintext[22]  = Þ XOR s
Plaintext[22]  = 222 XOR 115
Plaintext[22]  = 11011110
                 01110011
                 ======== XOR
                 01010010
Plaintext[22]  = 82
Plaintext[22]  = R

Plaintext[23]  = Ï XOR t
Plaintext[23]  = 207 XOR 116
Plaintext[23]  = 11001111
                 01110100
                 ======== XOR
                 01000100
Plaintext[23]  = 68
Plaintext[23]  = D
```

```
Plaintext[24]  = Ó XOR i
Plaintext[24]  = 211 XOR 105
Plaintext[24]  = 11010011
                 01101001
                 ======== XOR
                 01000101
Plaintext[24]  = 69
Plaintext[24]  = E

Plaintext[25]  = Ó XOR g
Plaintext[25]  = 211 XOR 103
Plaintext[25]  = 11010011
                 01100111
                 ======== XOR
                 01001011
Plaintext[25]  = 75
Plaintext[25]  = K

Plaintext[26]  = Ë XOR u
Plaintext[26]  = 203 XOR 117
Plaintext[26]  = 11001011
                 01110101
                 ======== XOR
                 01000001
Plaintext[26]  = 65
Plaintext[26]  = A

Plain Text     = AKU CINTA INDONESIA MERDEKA
```

Kunci = gusti

BAB V
PENUTUP

5.1 Kesimpulan

Setelah melakukan penelitian, penulis dapat menarik beberapa kesimpulan berdasarkan hasil pengujian yang dilakukan. Beberapa kesimpulan yang diperoleh adalah antara lain:

1. Masalah keamanan pada distribusi kunci dapat lebih baik
2. Dapat bisa menggunakan kunci yang panjang di plaintext.
3. Proses enkripsi algoritma kriptografi XNOR bekerja dengan baik dan dapat di deskripsikan.

5.2 Saran

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

1. Ukuran cipherteks lebih besar dari pada plainteks (bisa dua sampai empat kali ukuran plainteks).
2. Kriptografi XNOR dapat menerapkan skema Three-pass Protocol.
3. Kriptografi XNOR dapat dikombinasikan dengan teknik XOR.

DAFTAR PUSTAKA

Archana, & Vashist, A. (2017). Hill Cipher and Self Repetitive Matrix for Encryption and Decryption. *International Journal of Scientific Research and Education*, *5*(7), 6742–6747.

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).

Edraw. (2019). What is Algorithm - Definition, Types and Application. Retrieved

October 27, 2019, from https://www.edrawsoft.com/algorithm-definition.php

EDUCBA. (2017). What is Cryptography? | Types and Advantages of Cryptography. Retrieved October 23, 2019, from https://www.educba.com/what-is-cryptography/

Gilbert S. Vernam. (1919). *US Patent 1,310,719*.

Hamdani, H., Tharo, Z., & Anisah, S. (2019, May). PERBANDINGAN PERFORMANSI PEMBANGKIT LISTRIK TENAGA SURYA ANTARA DAERAH PEGUNUNGAN DENGAN DAERAH PESISIR. In Seminar Nasional Teknik (SEMNASTEK) UISU (Vol. 2, No. 1, pp. 190-195).

Jogiyanto, H. M. (2006). *Analisis Dan Desain Sistem Informasi, Pendekatan Terstruktur Teori Dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Offset.

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, 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

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

Ladjamudin, A.-B. bin. (2005). *Analisis dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu.

Lee, C. (2014). *Buku Pintar Pemrograman Visual Basic 2010*. Jakarta: Elex Media Komputindo.

Malhotra, M. (2014). A New Encryption Scheme Based on Enhanced RSA and ElGamal. *International Journal of Emerging Technologies in Computational and Applied Sciences*, *8*(2), 138–142.

Nakatsu, R. T. (2009). *Reasoning with Diagrams : Decision-Making and Problem-Solving with Diagrams.* John Wiley & Sons.

Rahim, R., Aryza, S., Wibowo, P., Harahap, A. K. Z., Suleman, A. R., Sihombing, E. E., ... & Agustina, I. (2018). Prototype file transfer protocol application for LAN and Wi-Fi communication. Int. J. Eng. Technol., 7(2.13), 345-347.

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.

Rahmaniar, R. (2019). Model FLASH-NR Pada Analisis Sistem Tenaga Listrik (Doctoral dissertation, Universitas Negeri Padang).

Rossanty, Y., Aryza, S., Nasution, M. D. T. P., & Siahaan, A. P. U. (2018). Design Service of QFC And SPC Methods in the Process Performance Potential Gain and Customers Value in a Company. Int. J. Civ. Eng. Technol, 9(6), 820-829.

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.

Siahaan, A. P. U., Ikhwan, A., & Aryza, S. (2018). A Novelty of Data Mining for Promoting Education based on FP-Growth Algorithm.

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.

Sitorus, Z. (2018). Kebutuhan Web Service untuk Sinkronisasi Data Antar Sistem Informasi dalam Universitas. Jurnal Teknik dan Informatika, 5(2), 87-90.

Stallings, W. (2005). *Cryptography and Network Security Principles and Practices* (4th ed.). Prentice Hall.

Sukmawati, R., & Priyadi, Y. (2019). Perancangan Proses Bisnis Menggunakan UML Berdasarkan Fit/Gap Analysis Pada Modul Inventory Odoo. *INTENSIF: Jurnal Ilmiah Penelitian Dan Penerapan Teknologi Sistem Informasi*, *3*(2), 104. https://doi.org/10.29407/intensif.v3i2.12697

Sukriadi Shafar. (2016). Pengertian Dan Contoh Kriptografi dengan Proses Enkripsi dan Dekripsi. Retrieved fromhttp://ondigitalforensics.weebly.com/cryptography/pengertian-dan-

contoh- kriptografi-dengan-proses-enkripsi-dan-dekripsi#.W7w6mxMzZZ0

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).

TechTarget. (2019). Cryptography. Retrieved October 27, 2019, from https://searchsecurity.techtarget.com/definition/cryptography

Uml-diagrams.org. (2019). Use case diagrams are UML diagrams describing units of useful functionality (use cases) performed by a system in collaboration with external users (actors). Retrieved November 3, 2019, from https://www.uml-diagrams.org/use-case-diagrams.html

UTM. (2019). Concept: Use-Case Model. Retrieved September 19, 2019, from http://www.utm.mx/~caff/doc/OpenUPWeb/openup/guidances/concepts/use_ case_model_CD178AF9.html

Wasserkrug, S., Dalvi, N., Munson, E. V., Gogolla, M., Sirangelo, C., Fischer-Hübner, S., … Snodgrass, R. T. (2009). Unified Modeling Language. In *Encyclopedia of Database Systems* (pp. 3232–3239). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-39940-9_440

Weerasinghe, T. D. B. (2013). An effective RC4 stream cipher. In *2013 IEEE 8th International Conference on Industrial and Information Systems* (pp. 69–74). IEEE. https://doi.org/10.1109/ICIInfS.2013.6731957

Wikipedia. (2019). Stream Cipher. Retrieved October 27, 2019, from https://en.wikipedia.org/wiki/Stream_cipher