PENGAMANAN DATA TEKS DENGAN TEKNIK KRIPTOGRAFI KLASIK MENGGUNAKAN ALGORITMA *REVERSE CIPHER*

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	:	**SERTIANI**
NPM	:	**1614370322**
PROGRAM STUDI	:	**SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020

LEMBAR PENGESAHAN

PENGAMANAN DATA TEKS DENGAN TEKNIK KRIPTOGRAFI KLASIK MENGGUNAKAN ALGORITMA REVERSE CIPHER

Disusun Oleh:

NAMA	:	**SERTIANI**
NPM	:	**1614370322**
PROGRAM STUDI	:	**SISTEM KOMPUTER**

Skripsi Telah Disetujui oleh Dosen Pembimbing Skripsi
Pada Tanggal :

Dosen Pembimbing I	Dosen Pembimbing II
A. P. U. Siahaan, S.Kom., M.Kom.	M. D. L. Siahaan, S.Kom., M.Kom.

Mengetahui:

Dekan Fakultas Sains dan Teknologi	Ketua Program Studi Sistem Komputer
Hamdani, S.T., M.T.	Eko Hariyanto, S.Kom., M.Kom.

SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Sertiani

NPM : 1614370322

Prodi : Sistem Komputer

Konsentrasi : Keamanan Jaringan Komputer

Judul Skripsi : Pengamanan Data Teks Dengan Teknik Kriptografi Klasik Menggunakan Algoritma Reverse Cipher

Dengan ini menyatakan bahwa :

1. Tugas Akhir/Skripsi saya bukan hasil plagiat.
2. Saya tidak akan menuntut perbaikan nilai Indeks Prestasi Komulatif (IPK) setelah ujian Sidang Meja Hijau.
3. Skripsi saya dapat dipublikasikan oleh pihak Lembaga dan saya tidak akan menuntut akibat publikasi tersebut.

Demikian ini pernyataan saya perbuat dengan sebenar-benarnya, terima kasih.

Medan, September 2020

Yang membuat pernyataan

Sertiani

Npm 1614370322

PERNYATAAN ORISINALITAS

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang diajukan untuk memperoleh gelar kesarjanaan di dalam perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang tertulis di acu dalam skripsi ini dan disebutkan dalam daftar pustaka.

Medan, September 2020

Sertiani

Npm 1614370322

ABSTRAK

SERTIANI

Pengamanan Data Teks Dengan Teknik Kriptografi Klasik Menggunakan Algoritma *Reverse Cipher*

2020

Data merupakan hal paling penting untuk dijaga kerahasiaannya. Setiap data memiliki informasi yang sangat vital sehingga memerlukan sistem keamanan yang tinggi. Data teks merupakan sekumpulan informasi yang berbentuk plaintext yang dapat dibuka secara langsung tanpa menggunakan aplikasi khusus. Apabila data teks tersebut jatuh ke tangan orang yang tidak bertanggung jawab, maka data tersebut akan tersebarluas sehingga pemilik data akan mengalami kerugian yang besar. Algoritma *Reverse Cipher* adalah salah satu teknik kriptografi yang dapat melakukan pengamanan pada data teks. Algoritma ini bekerja dengan cara menukar posisi karakter untuk setiap kata secara terbalik dari belakang hingga depan. Hasilnya plaintext akan berubah menjadi ciphertext dan data sudah tidak dapat dibaca lagi secara langsung. Keamanan dan kerahasiaan data akan terjamin dengan menerapkan algoritma *Reverse Cipher*.

Kata Kunci: algoritma, keamanan, Reverse, Cipher, enkripsi, dekripsi

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

KATA PENGANTAR

Puji syukur penulis ucapkan ke hadirat Tuhan YME karena berkat rahmat kesehatan dan hidayah-Nya, sehingga penulis dapat menyelesaikan penulisan skripsi tepat pada waktunya. Dalam penulisan skripsi ini, penulis memilih judul **"PENGAMANAN DATA TEKS DENGAN TEKNIK KRIPTOGRAFI KLASIK MENGGUNAKAN ALGORITMA *REVERSE CIPHER*"**.

Penulisan skripsi ini adalah Salah satu syarat untuk memperoleh gelar sarjana komputer, selama proses penulisan skripsi ini, penulis telah banyak mendapatkan bimbingan dan bantuan baik moral maupun materi dari berbagai pihak. Pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Dr. H. Muhammad Isa Indrawan, S.E., M.M., selaku Rektor Universitas Pembangunan Panca Budi Medan.
2. Hamdani, S.T., M.T., selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
3. Bapak Eko Hariyanto, S.Kom., M.Kom., selaku Ketua Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
4. Bapak Andysah Putera Utama Siahaan, S.Kom., M.Kom., selaku dosen pembimbing I yang telah meluangkan waktunya untuk membimbing dan memberikan arahan kepada penulis sehingga penulisan skripsi ini dapat diselesaikan.
5. Bapak Muhammad Donni Lesmana Siahaan, S.Kom., M.Kom., selaku dosen pembimbing II yang telah meluangkan waktunya untuk membimbing dan memberikan arahan kepada penulis sehingga penulisan skripsi ini dapat diselesaikan.
6. Kedua orang tua penulis yang telah banyak memberikan dukungan kepada penulis, memberikan motivasi dan doa sehingga penulis dapat menyelesaikan skripsi ini.
7. Bapak dan Ibu Dosen selaku Pengajar pada Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.

Penulis menyadari bahwa dalam penulisan skripsi ini masih banyak terdapat kesalahan dan kekurangan. Untuk itu saran dan kritik yang sehat dari semua pihak sangat penulis harap demi pengembangan isi skripsi ini. Akhirnya penulis berharap skripsi ini dapat berguna bagi para pembaca dan bagi penulis khususnya.

Medan, 21 Juni 2020
Penulis

Sertiani
1614370322

BAB I

PENDAHULUAN

1.1 Latar Belakang

Data merupakan kumpulan informasi yang digunakan untuk menyimpan hal-hal yang berkaitan dengan informasi pribadi, pekerjaan atau perusahaan. Sifat data ada yang umum dan rahasia. Data umum dapat digunakan atau dilihat oleh semua kalangan tanpa terkecuali sementara data rahasia adalah informasi yang memiliki kandungan sensitif yang hanya dapat dibuka oleh orang atau pihak yang diberikan wewenang penuh terhadap manajemen data tersebut. Data yang bersifat rahasia harus dijaga benar-benar agar tidak jatuh atau bocor ke tangan atau pihak yang tidak bertanggung jawab. Data ini harus benar-benar terlindungi. Tetapi, kadang-kadang perlindungan terhadap data kurang menjadi perhatian atau tidak terlindungi secara penuh. Ada kalanya data tersebut berhasil diretas oleh orang-orang yang tidak bertanggung jawab akibat kelalaian pemilik data tersebut.

Kebocoran informasi sering terjadi pada saat pengiriman dan pertukaran data. Oleh sebab itu, pengiriman data memerlukan sistem keamanan yang baik agar data tersebut tidak berhasil dicuri atau dibongkar. Teknik yang dapat digunakan dalam melakukan pengamanan data adalah kriptografi. Ilmu kriptografi berfungsi untuk mengubah pesan yang berupa karakter menjadi karakter lain sehingga susunan karakter yang dapat dipahami oleh manusia akan berubah menjadi susunan karakter acak sehingga tidak dapat dibaca dan dipahami oleh manusia yang tidak memiliki kemampuan untuk membongkar data tersebut.

Kriptografi memerlukan algoritma dalam melakukan penyandian data. Ada banyak algoritma yang ditawarkan dalam menjalankan fungsi kriptografi. Penelitian ini menggunakan algoritma *Reverse Cipher* dalam melakukan penyandian terhadap data sebelum data tersebut akan dikirimkan ke penerima. Algoritma ini bekerja dengan cara membalik arah teks atau pesan tersebut dengan posisi belakang ke depan untuk tiap kata yang ada pada pesan tersebut sehingga susunan dari pesan tersebut terbalik.

Penelitian ini dilakukan untuk memberikan keamanan pada pengiriman pesan. Dengan menerapkan algoritma *Reverse Cipher* diharapkan keamanan data akan terjamin. Berdasarkan latar belakang yang telah dijabarkan, penulis mengambil penelitian dengan judul "**PENGAMANAN DATA TEKS DENGAN TEKNIK KRIPTOGRAFI KLASIK MENGGUNAKAN ALGORITMA *REVERSE CIPHER***".

1.2 Rumusan Masalah

Adapun rumusan masalah yang digunakan dalam penulisan skripsi ini adalah sebagai berikut:

1. Bagaimana merancang pengamanan data teks menggunakan algoritma *Reverse Cipher*?
2. Bagaimana mengetahui cara kerja proses enkripsi dan dekripsi algoritma *Reverse Cipher*?
3. Bagaimana menentukan pembalikan karakter pada algoritma *Reverse Cipher*?

1.3 Batasan Masalah

Adapun batasan masalah yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Algoritma *Reverse Cipher* tidak menggunakan kunci dalam proses enkripsi dan dekripsi.
2. Batas proses enkripsi dan dekripsi adalah 1000 karakter.
3. Pesan yang digunakan adalah bertipe teks yang dimasukkan langsung pada textbox.
4. Bahasa pemrograman yang digunakan adalah menggunakan Microsoft Visual Basic.Net 2010.
5. Program aplikasi berbasis *desktop* dan tidak *online*.

1.4 Tujuan Penelitian

Adapun tujuan penelitian yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Untuk merancang pengamanan data teks menggunakan algoritma *Reverse Cipher*.
2. Untuk mengetahui cara kerja proses enkripsi dan dekripsi algoritma *Reverse Cipher*.
3. Untuk menentukan pembalikan karakter pada algoritma *Reverse Cipher*.

1.5 Manfaat Penelitian

Adapun manfaat penelitian yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Data teks yang akan dikirimkan akan aman dari pencurian data.
2. Memberi kenyamanan bagi pengirim dan penerima pesan.
3. Memberikan pengetahuan algoritma *Reverse Cipher*.

BAB II

LANDASAN TEORI

2.1 Pengertian Aplikasi

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus komputer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan (Sopyan et al., 2016).

Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer.

2.2 Program Aplikasi

Program aplikasi adalah program mandiri yang komprehensif yang melakukan fungsi tertentu secara langsung untuk pengguna. Di antara banyak lainnya, program aplikasi meliputi:

1 Surel

2 Browser web

3 permainan

4 Pengolah kata

5 Perangkat lunak perusahaan

6 Perangkat lunak akuntansi

7 Perangkat lunak grafik

8 Pemutar media

9 Manajemen basis data

Karena setiap program memiliki aplikasi khusus untuk pengguna akhir, istilah "aplikasi" digunakan. Misalnya, pengolah kata dapat membantu pengguna membuat artikel, sedangkan aplikasi game dapat digunakan untuk hiburan. Program aplikasi juga dikenal sebagai aplikasi atau perangkat lunak aplikasi.

Perangkat lunak aplikasi dan perangkat lunak sistem adalah dua jenis perangkat lunak utama yang tersedia. Perangkat lunak sistem mengelola operasi internal komputer, terutama melalui sistem operasi (OS). Ia mengelola periferal seperti perangkat penyimpanan, printer, dan monitor juga. Sebaliknya, perangkat lunak aplikasi atau program aplikasi memandu komputer untuk melakukan instruksi yang diberikan oleh pengguna.

Perangkat lunak sistem mencakup program yang berjalan di latar belakang, yang memungkinkan program aplikasi berfungsi. Program perangkat lunak sistem termasuk kompiler, assembler, alat manajemen file serta OS itu

sendiri. Program aplikasi berfungsi di atas perangkat lunak sistem karena perangkat lunak sistem dibangun dari program "tingkat rendah". Perangkat lunak sistem diinstal secara otomatis selama instalasi OS. Namun, pengguna memiliki opsi untuk memilih program aplikasi mana yang diinstal pada sistem mereka. Beberapa contoh program aplikasi meliputi:

1 Application suite: Termasuk berbagai aplikasi yang dikemas bersama
2 Perangkat lunak perusahaan: Mengatasi aliran data dan persyaratan proses suatu organisasi, yang mencakup seluruh departemen
3 Perangkat lunak pekerja informasi: Memungkinkan pengguna untuk membuat dan mengelola informasi
4 Perangkat lunak akses konten: Digunakan terutama untuk mendapatkan akses ke konten tanpa mengedit
5 Perangkat lunak pengembangan media: Membuat media elektronik dan cetak
6 Perangkat lunak pendidikan: Termasuk konten dan / atau fitur yang ditujukan untuk siswa atau pendidik
7 Perangkat lunak rekayasa produk: Mengembangkan perangkat lunak dan produk perangkat keras

2.3 Data

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi (Jogiyanto, 2016). Kegiatan suatu perusahaan, misalnya

transaksi penjualan oleh sejumlah *salesman*, dihasilkan sejumlah faktur-faktur yang merupakan data dari penjualan pada suatu periode tertentu. Faktur-faktur penjualan tersebut masih belum dilaporkan secara terperinci kepada manajemen. Untuk keperluan pengambilan keputusan, maka faktur-faktur tersebut perlu diolah lebih lanjut untuk menjadi suatu informasi (Sun et al., 2014).

2.3.1 Bagaimana Data Disimpan

Komputer mewakili data, termasuk video, gambar, suara dan teks, sebagai nilai biner menggunakan pola hanya dua angka: 1 dan 0. Sedikit adalah unit data terkecil dan hanya mewakili nilai tunggal. Satu byte terdiri dari delapan digit biner. Penyimpanan dan memori diukur dalam megabit dan gigabit.

Unit-unit pengukuran data terus bertambah seiring dengan meningkatnya jumlah data yang dikumpulkan dan disimpan. Istilah "brontobyte" yang relatif baru, misalnya, adalah penyimpanan data yang setara dengan 10 hingga 27 byte. Data dapat disimpan dalam format file, seperti pada sistem mainframe menggunakan ISAM dan VSAM. Format file lain untuk penyimpanan, konversi, dan pemrosesan data termasuk nilai yang dipisah koma. Format ini terus menemukan kegunaan di berbagai jenis mesin, bahkan ketika pendekatan yang lebih berorientasi data terstruktur memperoleh pijakan dalam komputasi perusahaan. Spesialisasi yang lebih besar dikembangkan sebagai basis data, sistem manajemen basis data, dan kemudian teknologi basis data relasional muncul untuk mengatur informasi (Zhang et al., 2009).

2.3.2 Jenis data

Pertumbuhan web dan telepon pintar selama dekade terakhir menyebabkan peningkatan dalam penciptaan data digital. Data sekarang termasuk informasi teks, audio dan video, serta catatan aktivitas log dan web. Banyak dari itu adalah data yang tidak terstruktur.

Istilah big data telah digunakan untuk menggambarkan data dalam kisaran petabyte atau lebih besar. Tulisan singkat menggambarkan data besar dengan 3V - volume, variasi, dan kecepatan. Ketika e-commerce berbasis web telah menyebar, model bisnis berbasis data besar telah berevolusi yang memperlakukan data sebagai aset. Tren semacam itu juga telah menimbulkan keasyikan yang lebih besar dengan penggunaan sosial data dan privasi data.

Data memiliki makna di luar penggunaannya dalam aplikasi komputasi yang berorientasi pada pemrosesan data. Misalnya, dalam interkoneksi komponen elektronik dan komunikasi jaringan, istilah data sering dibedakan dari "informasi kontrol," "bit kontrol," dan istilah serupa untuk mengidentifikasi konten utama dari unit transmisi. Selain itu, dalam sains, istilah data digunakan untuk menggambarkan kumpulan fakta. Itu juga terjadi di bidang-bidang seperti keuangan, pemasaran, demografi dan kesehatan.

2.3.3 Pengelolaan dan Penggunaan Data

Dengan semakin banyaknya data dalam organisasi, penekanan tambahan telah ditempatkan pada memastikan kualitas data dengan mengurangi duplikasi dan menjamin yang paling akurat, catatan saat ini digunakan. Banyak langkah

yang terlibat dengan manajemen data modern termasuk pembersihan data, serta mengekstrak, mengubah dan memuat (ETL) proses untuk mengintegrasikan data. Data untuk diproses telah dilengkapi dengan metadata, kadang-kadang disebut sebagai "data tentang data," yang membantu administrator dan pengguna memahami database dan data lainnya.

Analisis yang menggabungkan data terstruktur dan tidak terstruktur menjadi bermanfaat, karena organisasi berupaya memanfaatkan informasi tersebut. Sistem untuk analitik semacam itu semakin berupaya untuk kinerja waktu-nyata, sehingga mereka dibangun untuk menangani data yang masuk yang dikonsumsi dengan tingkat konsumsi tinggi, dan untuk memproses aliran data untuk penggunaan langsung dalam operasi.

Seiring waktu, gagasan basis data untuk operasi dan transaksi telah diperluas ke basis data untuk pelaporan dan analitik data prediktif. Contoh utama adalah gudang data, yang dioptimalkan untuk memproses pertanyaan tentang operasi untuk analis bisnis dan pemimpin bisnis. Meningkatnya penekanan pada menemukan pola dan memprediksi hasil bisnis telah mengarah pada pengembangan teknik penambangan data (Barone et al., 2017).

2.4 Logika dan Algoritma

Pengertian algoritma sangat lekat dengan kata logika, yaitu kemampuan seorang manusia untuk berfikir dengan akal tentang suatu permasalahan menghasilkan sebuah kebenaran, dibuktikan dan dapat diterima akal, logika

seringkali dihubungkan dengan kecerdasan, seseorang yang mampu berlogika dengan baik sering orang menyebutnya sebagai pribadi yang cerdas.

Logika identik dengan masuk akal dan penalaran. Penalaran adalah salah satu bentuk pemikiran. Pemikiran adalah pengetahuan tak langsung yang didasarkan pada pernyataan langsung pemikiran mungkin benar dan mungkin juga tak benar. Definisi logika sangat sederhana yaitu ilmu yang memberikan prinsip-prinsip yang harus diikuti agar dapat berfikir valid menurut aturan yang berlaku. Pelajaran logika menimbulkan kesadaran untuk menggunakan prinsip-prinsip untuk berfikir secara sistematis. Logika berasal dari dari bahasa Yunani yaitu LOGOS yang berarti ilmu. Logika dapat diartikan ilmu yang mengajarkan cara berpikir untuk melakukan kegiatan dengan tujuan tertentu. Algoritma berasal dari nama seorang Ilmuwan Arab yang bernama Abu Jafar Muhammad Ibnu Musa Al Khuwarizmi penulis buku berjudul Al Jabar Wal Muqabala. Kata Al Khuwarizmi dibaca orang barat menjadi Algorism yang kemudian lambat laun menjadi Algorithm diserap dalam bahasa Indonesia menjadi Algoritma.

Logika identik dengan masuk akal dan penalaran. Penalaran adalah salah satu bentuk pemikiran. Pemikiran adalah pengetahuan tak langsung yang didasarkan pada pernyataan langsung pemikiran mungkin benar dan mungkin juga tak benar. Definisi logika sangat sederhana yaitu ilmu yang memberikan prinsip-prinsip yang harus diikuti.

2.5 Kriptografi

Menurut M. Miftakhul Amin, kriptografi (*Cryptography*) berasal dari bahasa yunani terdiri dari dua suku kata yaitu kripto dan graphia. Kripto artinya menyembunyikan sedangkan graphia artinya tulisan. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data. Tetapi tidak semua aspek keamanan informasi dapat diselesaikan dengan kriptografi (Amin, 2016). Adapun istilah-istilah yang sering digunakan dalam ilmu kriptografi diantara sebagai berikut:

1. *Plaintext*

   Plaintext merupakan pesan asli yang belum disandikan atau informasi yang ingin dikirimkan atau dijaga keamanannya.

2. *Ciphertext*

   Ciphertext merupakan pesan yang telah disandikan (dikodekan) sehingga siap untuk dikirimkan.

3. Enkripsi

   Enkripsi merupakan proses yang dilakukan untuk menyandikan plaintext menjadi ciphertext dengan tujuan pesan tersebut tidak dapat dibaca oleh pihak yang tidak berwenang.

4. Deskripsi

   Deskripsi merupakan proses yang dilakukan untuk memperoleh kembali plaintext dari ciphertext.

5. Kunci

   Kunci yang dimaksud disini adalah kunci yang dipakai untuk melakukan dekripsi dan enkripsi. Kunci terbagi menjadi dua bagian, diantaranya yaitu kunci pribadi (*private key*) dan kunci umum (*public key*).

6. Kriptosistem

   Kriptosistem merupakan sistem yang dirancang untuk mengamankan suatu sistem informasi dengan memanfaatkan kriptografi.

7. Kriptanalisis

   Kriptanalisis merupakan suatu ilmu untuk mendapatkan plaintext tanpa harus mengetahui kunci secara wajar.

Kriptografi dapat pula diartikan sebagai ilmu atau seni untuk menjaga keamanan pesan. Ketika suatu pesan dikirim dari suatu tempat ke tempat lain, isi pesan tersebut mungkin dapat disadap oleh pihak lain yang tidak berhak untuk mengetahui isi pesan tersebut. Untuk menjaga pesan maka pesan tersebut dapat diubah menjadi sebuah kode yang tidak dapat dimengerti pihak lain.

2.5.1 Sejarah Kriptografi

Sejarah kriptografi sebagian besar merupakan sejarah kriptografi klasik, yaitu metode enkripsi yang menggunakan kertas dan pensil atau mungkin dengan bantuan alat mekanik sederhana. Secara umum algoritma kriptografi klasik dikelompokkan menjadi dua kategori, yaitu algoritma transposisi (*transposition cipher*) dan algoritma substitusi (*substitution cipher*). *Cipher* transposisi

mengubah susunan huruf-huruf di dalam pesan, sedangkan *cipher* substitusi mengganti setiap huruf atau kelompok huruf dengan sebuah huruf atau kelompok huruf lain (Pabokory et al., 2015).

2.5.2 Tujuan Kriptografi

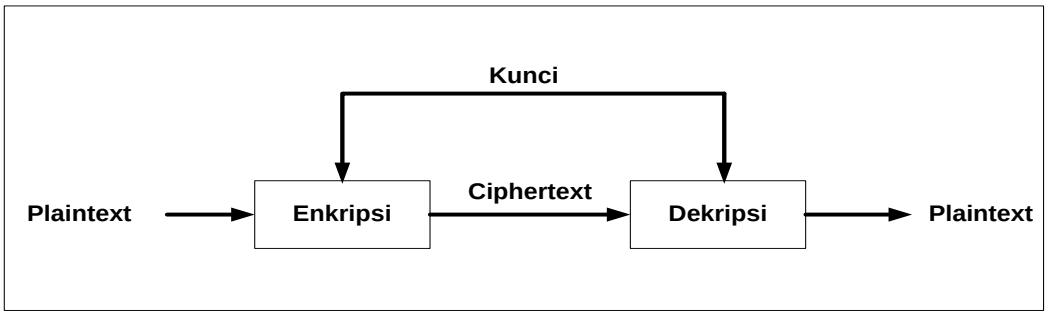
Dari paparan awal dapat dirangkumkan bahwa kriptografi bertujuan untuk member layanan keamanan. Yang dinamakan aspek-aspek keamanan:

1. Kerahasiaan (*confidentiality*) adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.
2. Integritas data (*data integrity*) adalah layanan yang menjamin bahwa pesan masih asli atau belum pernah dimanipulasi selama pengiriman.
3. Otentikasi (*authentication*) adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user authentication*).
4. *Non-repudiation* adalah layanan untuk menjaga entitas yang berkomunikasi melakukan penyangkalan.

2.5.3 Kriptografi Simetris

Kriptografi simetris adalah teknik kriptografi dimana kunci yang digunakan pada proses enkripsi dan dekripsi adalah kunci yang sama. Dalam kriptografi kunci simetris dapat diasumsikan bahwa si penerima dan pengirim pesan telah terlebih dahulu berbagi kunci sebelum pesan dikirimkan. Keamanan dari sistem ini terletak pada kerahasiaan kuncinya.

Pada umumnya yang termasuk ke dalam kriptografi simetris ini beroperasi dalam mode blok (*block cipher*), yaitu setiap kali proses enkripsi atau dekripsi dilakukan terhadap satu blok data (yang berukuran tertentu), atau beroperasi dalam mode aliran (*stream cipher*), yaitu setiap kali enkripsi atau dekripsi dilakukan terhadap satu bit atau satu byte data. Contoh algoritma simetris, yaitu: Trithemius, Double Transposition Cipher, DES (Data Encryption Standard), AES (Advanced Encryption Standard). Gambar 2.1 adalah skema algoritma simetris.



Gambar 2.1 Skema kriptografi simetris
Sumber: (Putri et al., 2018)

Kelebihan kriptografi simetris adalah:

1. Proses enkripsi atau dekripsi kriptografi simetris membutuhkan waktu yang singkat.
2. Ukuran kunci simetris *relative* lebih pendek.
3. Otentikasi pengiriman pesan langsung dari *ciphertext* yang diterima, karena kunci hanya diketahui oleh penerima dan pengirim saja.

Kelemahan kriptografi simetris antara lain:

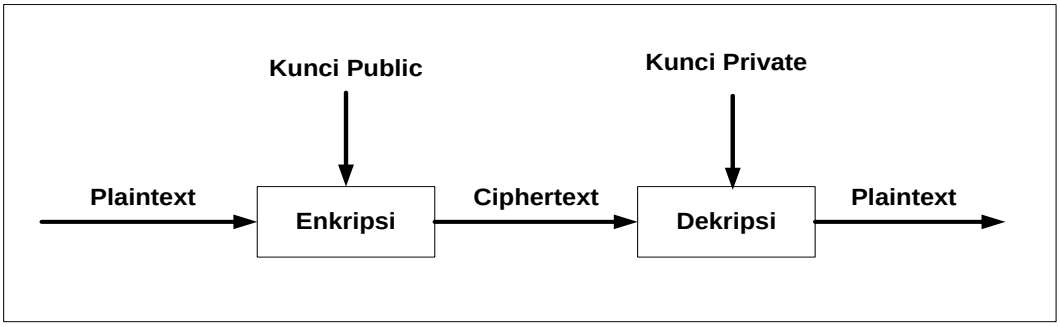
1. Kunci simetris harus dikirim melalui saluran komunikasi yang aman, dan kedua entitas yang berkomunikasi harus menjaga kerahasiaan kunci.
2. Kunci harus sering diubah, setiap kali melaksanakan komunikasi. Apabila kunci tersebut hilang atau lupa, maka pesan tersebut tidak dapat dibuka.

2.5.4 Kriptografi Asimetris

Berbeda dengan kriptografi kunci simetris, kriptografi kunci public memiliki dua buah kunci yang berbeda pada proses enkripsi dan dekripsinya. Dimana kunci yang digunakan untuk proses enkripsi atau sering disebut *public key* dan dekripsi atau sering disebut *private key* menggunakan kunci yang berbeda. Entitas pengirim akan mengenkripsi dengan menggunakan kunci *public*, sedangkan entitas penerima mendekripsi menggunakan kunci *private* (Kamil, 2016).

Contoh algoritma asimetris, yaitu RSA (*Riverst Shamir Adleman)*, Knapsack, Rabin, ElGamal (Ayushi, 2010) (S. et al., 2012). Pada algoritma tak simetri kunci terbagi menjadi dua bagian:

1. Kunci umum (*public key*) adalah kunci yang dapat dan boleh diketahui oleh semua orang.
2. Kunci pribadi (*private key*) adalah kunci yang hanya dapat diketahui penerima dan bersifat rahasia.



Gambar 2.2 Skema kriptografi asimetris
Sumber: (Putri et al., 2018)

Kelebihan kriptografi asimetris adalah:

1. Hanya kunci *private* yang perlu dijaga kerahasiaannya oleh setiap entitas yang berkomunikasi. Tidak ada kebutuhan mengirim kunci *private* sebagaimana kunci simetri.
2. Pasangan kunci *private* dan kunci *public* tidak perlu diubah dalam jangka waktu yang sangat lama.
3. Dapat digunakan dalam pengaman pengiriman kunci simetris.

Kelemahan kriptografi asimetris adalah:

1. Proses enkripsi dan dekripsi umumnya lebih lambat dari algoritma simetri, karena menggunakan bilangan yang besar dan operasi bilangan yang besar.
2. Ukuran *ciphertext* lebih besar dari *plaintext*.
3. Ukuran kunci relatif lebih besar daripada ukuran kunci simetris.

2.6 Enkripsi

Enkripsi adalah proses penyandian *plaintext* menjadi *ciphertext*, atau pengubahan data menjadi bentuk rahasia. Proses *enkripsi algoritma AES* terdiri dari 4 jenis *transformasi bytes*, yaitu *SubBytes, ShiftRows, Mixcolumns*, dan *AddRoundKey*. Pada awal proses *enkripsi*, input yang telah dicopykan ke dalam *state* akan mengalami *transformasi byte AddRoundKey*. Setelah itu, *state* akan mengalami *transformasi SubBytes, ShiftRows, MixColumns,* dan *AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam *algoritma AES* disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada round terakhir, state tidak mengalami transformasi *MixColumns* (Amin, 2016).

Enkripsi digunakan untuk menyandikan data-data atau informasi sehingga tidak dapat dibaca oleh orang lain. Dengan ekkripsi, data kita disandikan (Encrypted) dengan menggunakan sebuah kunci (key). Untuk membuka (mendecrypt) data tersebut, digunakan kunci yang sama ketika mengenkrip. Enkripsi adalah proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan khusus. Dikarenakan enkripsi telah digunakan untuk mengamankan komunikasi di berbagai negara, hanya organisasi-organisasi tertentu dan individu yang memiliki kepentingan yang sangat mendesak akan kerahasiaan yang menggunakan enkripsi. Di pertengahan tahun 1970-an, enkripsi kuat dimanfaatkan untuk pengamanan oleh sekretariat agen pemerintah Amerika Serikat pada domain publik, dan saat ini enkripsi telah

digunakan pada sistem secara luas, seperti Internet e-commerce, jaringan Telepon bergerak dan ATM pada bank.

Keamanan dari enkripsi tergantung beberapa faktor salah satunya yaitu menjaga kerahasiaan kuncinya bukan algoritmanya. Proses enkripsi dapat diterangkan sebagai berikut:

1. Masukkan file dan key
2. Baca isi file
3. Lakukan perhitungan untuk melakukan enkripsi
4. Outputnya adalah ciphertext
5. Pilih Folder Penyimpanan
6. Selesai

Langkah-langkah pada proses enkripsi adalah sebagai berikut:

1. *Plaintext* diubah ke dalam bentuk bilangan. Untuk mengubah plaintext yang berupa huruf menjadi bilangan dapat digunakan kode *ASCII* dalam sistem bilangan desimal.
2. *Plaintext* m dinyatakan menjadi blok-blok m1,m2,m3,...., sedemikian sehingga setiap blok merepresentasikan nilai di dalam selang [0, n-1] , sehingga transformasinya menjadi satu ke satu.
3. Setiap blok mi dienkripsi menjadi blok ci dengan rumus mi = ci e mod n

2.7 Dekripsi

Dekripsi digunakan untuk mengembalikan data-data atau informasi yang sudah dienkripsi ke bentuk awal sehingga dapat dibaca kembali dengan baik. Satu kaidah upaya pengolahan data menjadi sesuatu yang dapat diutarakan secara jelas dan tepat dengan tujuan agar dapat dimengerti oleh orang yang tidak langsung mengalaminya sendiri dalam keilmuan, deskripsi diperlukan agar peneliti tidak melupakan pengalamannya dan agar pengalaman tersebut dapat dibandingkan dengan pengalaman peneliti lain, sehingga mudah untuk dilakukan pemeriksaan dan kontrol terhadap deskripsi tersebut. Pada umumnya deskripsi menegaskan sesuatu, seperti apa sesuatu itu kelihatannya, bagaimana bunyinya, bagaimana rasanya, dan sebagainya (Amin, 2016).

Deskripsi yang detail diciptakan dan dipakai dalam disiplin ilmu sebagai istilah teknik. Saat data yang dikumpulkan, deskripsi, analisis dan kesimpulannya lebih disajikan dalam angka-angka maka hal ini dinamakan penelitian kuantitatif. Sebaliknya, apabila data, deskripsi, dan analisis kesimpulannya disajikan dalam uraian kata-kata maka dinamakan penelitian kualitatif. Proses deskripsi dapat diterangkan sebagai berikut:

1. Pilih folder penyimpanan
2. Masukkan file cipher & key
3. Baca isi file
4. Lakukan perhitungan untuk dekripsi
5. Outputnya adalah plaintext

Dekripsi adalah proses memperoleh kembali *plaintext* menjadi *ciphertext*, atau proses pengubahan kembali data yang berbentuk rahasia menjadi semula. *Transformasi byte* yang digunakan pada invers cipher adalah *InvShiftRows, InvSubBytes, InvMixColumns,* dan *AddRoundKey*. Langkah-langkah pada proses *dekripsi* adalah sebagai berikut:

1. Setiap blok *ciphertext* ci *didekripsi* kembali menjadi blok mi dengan rumus mi = ci d mod n
2. Kemudian blok-blok m1,m2,m3,...., diubah kembali ke bentuk huruf dengan melihat kode *ASCII* hasil *dekripsi*. (Yuza, dkk, 2018)

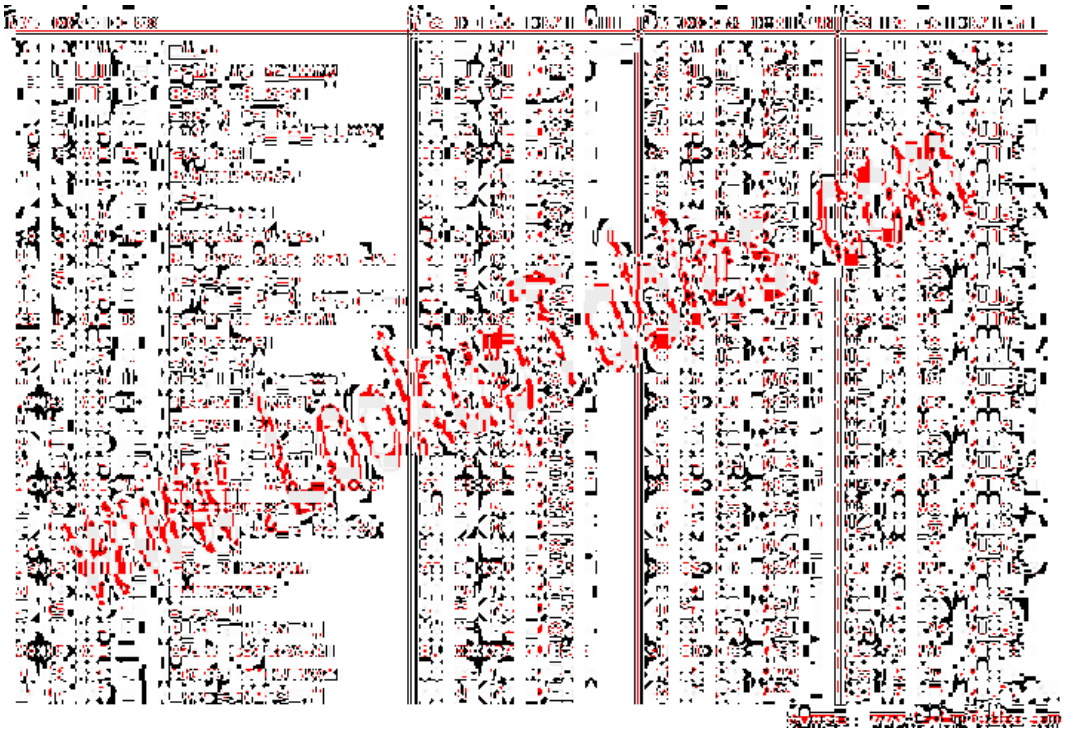
2.8 *Reverse Cipher*

Reverse Cipher mengenkripsi pesan dengan menuliskan kembali deretan karakter dalam urutan terbalik. Jadi, "Hello World!" akan di enkripsi menjadi "!dlroW olleH". Untuk mendekripsi, proses algoritma *Reverse Cipher* dilakukan dengan cara yang sama yaitu cukup membalikkan pesan terbalik untuk mendapatkan pesan asli. Langkah enkripsi dan dekripsi adalah sama. *Reverse Cipher* terbalik adalah cipher yang sangat mudah digunakan. Tetapi algoritma ini memiliki kelemahan yang hanya dengan melihat ciphertext-nya, seseorang dapat mengetahui bahwa itu hanya dalam urutan terbalik. ".yas tahw tuo erugif llits ylbaborp nac uoy, detpyrcne si hellt hguoht neve, elpmaxe roF". Tetapi kode untuk algoritma *Reverse Cipher* mudah untuk digunakan dan cepat dalam melaksanakan proses enkripsi dan dekripsi pada *plaintext* dan *ciphertext.*

2.9 Kode ASCII

ASCII adalah singkatan dari American Standard Code for Information Interchange. Komputer hanya dapat memahami angka, sehingga kode ASCII adalah representasi numerik dari karakter seperti 'a' atau '@' atau semacam aksi. ASCII dikembangkan sejak lama dan sekarang karakter non-cetak jarang digunakan untuk tujuan aslinya. Di bawah ini adalah tabel karakter ASCII dan ini termasuk deskripsi dari 32 karakter non-cetak pertama. ASCII sebenarnya dirancang untuk digunakan dengan teletype dan uraiannya agak kabur.

Tabel 2.1 Kode ASCII



2.10 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak (Mallu, 2015). UML menawarkan sebuah standar untuk merancang model sebuah system. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*) (Isa & Hartawan, 2017).

Unifed Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendekskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (Wasserkrug et al., 2019).

Penggunaan model ini bertujuan untuk mengidentifikasikan bagian-bagian yang termasuk dalam lingkup sistem yang dibahas dan bagaimana hubungan antara sistem dengan subsistem maupun sistem lain diluarnya (Sukmawati & Priyadi, 2019).

2.10.1 Use Case Diagram

Use Case diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*). sehingga pembuatan use case diagram lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan

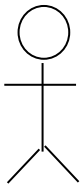
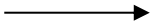
berdasarkan alur atau urutan kejadian. Sebuah use case diagram mempresentasikan sebuah interaksi antara aktor dengan sistem (Isa & Hartawan, 2017).

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. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenarion. Setiap skenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu.

Sedangkan menurut Ade Hendini, *Use Use case diagram* merupakan pemodelan untuk kelakuakn (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Hendini, 2016). Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

Tabel 2.2 Simbol Use Case Diagram

No	Simbol	Deskripsi
1	*Use case* Nama *Use*	Gambaran unit yang saling berkaitan antara aktor dengan sistem yang berjalan

2	Aktor  Nama aktor	Orang, proses atau sistem yang lain yang berinteraksi dengan sistem informasi yang akan dibuat.
3	Asosiasi / *Association*	Komunikasi antara aktor dan *use case*.
4	Ekstensi / *Extend* <<extend>> 	Kelakuan yang hanya berjalan di bawah kondisi tertentu. Seperti jika akun sesuai, atau jika *session* sesuai.
5	Generalisasi 	Elemen yang menjadi spesialisasi elemen lain.
6	*Include* <<include>> 	Kelakuan yang harus terpenuhi agar suatu *event* dapat terjadi.

Sumber: (Hendini, 2016)

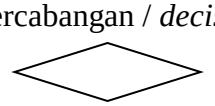
2.10.2 Activity Diagram

Menurut Indra Griha Tofik Isa dan George Pri Hartawan, Activity Digram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya. Diagram ini sangat mirip dengan flowchart karena memodelkan *workflow* dari suatu aktifitas ke aktifitas yang lainnya, atau dari aktifitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu

memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *use case* (Isa & Hartawan, 2017).

Activity Diagram adalah bagian penting dari *UML*, yang menggambarkan aspek dinamis dari sistem. logika prosedural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *activity diagram*. *Activity diagram* mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung prilaku paralel sedangkan *flowchart* tidak bisa (Kurniawan, 2018). *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity Diagram* yaitu:

Tabel 2.3 Simbol Activity Diagram

No	Simbol	Deskripsi
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan / *decision* 	Asosiasi percabangan dimana jika ada aktivitas pilihan lebih dari satu.
4	Penggabungan / Join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

5	Status Akhir 	Tahap akhir dari proses sistem.

Sumber: (Hendini, 2016)

2.11 Bahasa Pemrograman

Bahasa telah menjadi sarana komunikasi dan interaksi utama manusia selama ribuan tahun. Untuk sebuah komunitas, bahasa tersebut mengandung kata-kata yang perlu dikomunikasikan oleh orang-orang, kata-kata itu sendiri abstrak, tetapi mereka menunjukkan artinya, mereka menunjuk ke objek atau tindakan, dll.

Ketika seseorang melihat komputer Anda, seseorang akan menemukan itu tidak jauh berbeda. Ada banyak perangkat keras dan lunak yang perlu saling berkomunikasi. Aplikasi seseorang bereaksi terhadap mouse dan keyboard atau bahkan mik, ia dapat membaca file dari penyimpanan disk seseorang dan sebagainya. Tetapi pada akhirnya, mesin tidak mengerti apa-apa selain bit, 1s, dan 0s, kombinasi yang menciptakan makna.

Komputer paling awal sebenarnya diprogram dengan mengubah yang dan nol secara manual, bergantian sirkuit dan kabel. Tentu saja, itu tidak mudah untuk membuat banyak program karena kebanyakan hanya digunakan untuk aplikasi spesifik, dan mereka berukuran sangat besar sehingga sangat terbatas. Itulah sebabnya penciptaan bahasa pemrograman adalah langkah revolusioner yang membawa bidang ini ke tingkat lain. Tidak seperti bahasa normal, kata kunci dalam bahasa pemrograman terbatas, dan dengan menggabungkan kata kunci ini, pengembang dapat membuat berbagai jenis program. Ada perangkat lunak khusus

yang mengubah kode yang seseorang tulis menjadi bahasa mesin yang dimengerti mesin. Jadi apa itu bahasa pemrograman? Singkatnya, bahasa pemrograman adalah serangkaian instruksi yang digunakan manusia untuk berinteraksi dengan komputer.

2.11.1 Logika Pemrograman

Logika dan algoritma pemrograman merupakan sesuatu yang berbeda tetapi tidak dapat dipisahkan saat digunakan untuk membangun sebuah produk dengan menggunakan bahasa pemrograman. Logika dapat digunakan untuk memecahkan masalah pemrograman yang sedang dihadapi, sedangkan algoritma akan membuat permasalahan tersebut terselesaikan secara runtut sesuai alur yang seharusnya.

Oleh karena itu, seorang pemula tentu harus berlatih kedua hal tersebut dengan tepat, temasuk di dalamnya berlatih untuk mengolah logika pemrograman dasar untuk mengembangkan sebuah program melalui penyelesaian permasalahan yang tepat. Logika pemrograman dasar menjadi kunci logika-logika pengembangan lanjutan.

Berikut ini logika pemrograman dasar yang dapat dipelajari oleh calon pembuat program:

1. Logika Aritmatika

   Seperti kehidupan ini, bahasa pemrograman tidak terlepas dari perhitungan matematika. Dalam mengembangkan sebuah program, aktivitas menghitung tidak dapat kita hindari. Satu atau dua kali pasti kita akan

menggunakan sebuah perhitungan dalam menyelesaikan permasalahan dalam melakukan pemrograman.Oleh karena itu, logika aritmatika masih harus kita pelajari untuk menjadi pembuat program yang handal. Logika ini akan membantu kita dalam memecahkan permasalahan terkait dengan perhitungan, termasuk di dalamnya kasus-kasus yang membutuhkan operasi penjumlahan, pengurangan, perkalian, atau perhitungan lainnya. Secara ringkas, developer pemula dapat mempelajari beberapa operasi matematika. Operasi-operasi tersebut merupakan logika dasar pemrograman yang harus dimiliki oleh seorang developer atau orang yang berkecimpung di dunia pemrograman. Hal ini sangat penting sebagaimana matematika yang menjadi salah satu bagian penting dari bahasa pemrograman.

2. Logika Perbandingan

   Sebagaimana namanya, logika perbandingan merupakan sebuah penalaran yang digunakan untuk membandingkan dua hal yang memiliki nilai. Logika ini dapat digunakan untuk melihat apakah dua hal yang dibandingkan memiliki nilai yang sama, atau berbeda. Dengan menggunakan logika ini, kita akan dapat menalar apakah angka yang pertama memiliki nilai yang sama, lebih besar, lebih kecil, atau tidak sama dengan angka yang kedua. Jenis logika ini dapat kita gunakan untuk membuat sebuah persyaratan untuk tercapainya sebuah kondisi. Misalnya, jika kita ingin membuat kontrol lampu dengan menggunakan kondisi cahaya yang ada, maka kita dapat menggunakan logika perbandingan

“lebih kecil”. Kita dapat menggunakan logika “jika nilai intensitas cahaya lebih kecil dari 50 candela, maka lampu akan menyala”. Logika ini tentu akan sangat bermanfaat jika dipelajari. Seorang pemula harus mempelajari logika ini sebagaimana logika pemrograman dasar lainnya. Dengan mempelajari logika pemrograman dasar ini, seorang pemula dapat mengembangkan logika-logika dasar menjadi logika tingkat lanjutan.

3. Logika Boolean

   Logika boolean merupakan sebuah penalaran yang menghasilkan nilai Benar atau Salah dari dua buah kondisi yang digunakan sebagai syarat. Dalam sebuah program, nilai benar dan salah dari suatu hal dapat dilihat dari masing-masing nilai dari dua kondisi yang menentukannya. Misalnya kita menggunakan logika “dan”, maka nilai dari hal tersebut akan bernilai “benar” jika kedua kondisi syaratnya memiliki nilai “benar”. Akan tetapi, jika nilai salah satu atau bahkan kedua kondisi syarat tersebut bernilai “salah”, maka hal tersebut akan bernilai “salah” juga. Contoh lainya adalah logika “atau”,di mana nilai hasil akan selalu “benar” jika salah satu kondisi syarat memiliki nilai “benar”, dan akan bernilai “salah”, jika dari kedua kondisi syarat tidak ada yang memiliki nilai “benar”. Logika ini juga sangat penting untuk dipelajari, seperti logika pemrograman dasar yang telah dijelaskan sebelumnya. Logika ini merupakan logika yang harus dilatih dan dipelajari oleh seorang pemula setelah menentukan bahasa pemrograman yang tepat untuknya.

Logika-logika yang telah dijelaskan merupakan logika pemrograman dasar yang harus dimiliki oleh seorang developer atau orang yang berkecimpung di dunia pemrograman. Logika-logika dasar ini berlaku untuk semua jenis bahasa pemrograman yang sedang dipelajari.

Dan selain itu, karena ini merupakan logika pemrograman dasar, tentu masih banyak penalaran-penalaran lain yang belum disebutkan. Calon pembuat program harus melatih logika-logika ini, agar terbiasa menggunakannya untuk memecahkan kasus-kasus saat mengembangkan sebuah program dan mendapatkan manfaat lainnya.

2.11.2 Kode Program

Kode ini hampir seperti menulis paragraf instruksi atau membuat daftar tugas untuk komputer. Tidak seperti kita manusia, daftar tugas dan instruksi yang seseorang tulis untuk komputer harus sangat rinci dan ditulis dalam beberapa logika.

Dengan kode dan pemrograman, seseorang dapat membuat komputer menggambar bentuk yang rumit dan membuat grafik komputer yang kaya, dan kemudian membuat program yang memahami mekanika game dan membantu seseorang membuat game yang terasa nyata dengan gravitasi dan tabrakan partikel, dengan program ini yang paling bisa seseorang buat semua permainan intens dan imersif.

Dengan kode dan pemrograman, seseorang dapat membuat dan mengirim konten di seluruh dunia dengan blog dan situs web pribadi seseorang dan gaya

blog seseorang untuk memenuhi gaya Anda. Seseorang dapat membangun solusi bisnis yang digerakkan oleh teknologi dan menjangkau lebih banyak pelanggan serta melayani berbagai kebutuhan yang lebih luas.

Selain itu, dengan kode dan pemrograman, seseorang dapat membuat aplikasi rumah pintar, seperti pengumpan hewan peliharaan otomatis, cermin pintar atau bahkan membuat robot yang dapat membantu menyelesaikan tugas-tugas rumah tangga dan menjadi asisten virtual seseorang untuk berbicara dan memahami Anda. Berbeda dengan apa yang dipikirkan banyak orang, ada banyak seni yang terlibat dalam teknik komputer dan ilmu komputer.

2.11.3 Mempelajari Bahasa pemrograman

Belajar bahasa pemrograman terutama membutuhkan dedikasi dan latihan, Anda tidak dapat menjadi seorang programmer tanpa menulis kode. Orang dapat mulai memahami Ilmu Komputer dengan kursus pengantar yang disukai, rekomendasi adalah CS50 Harvard yang tersedia untuk umum di saluran youtube mereka. Setelah terbiasa dengan konsep pemrograman dan dasar, seseorang dapat mengambil bidang dan bahasa yang disukai dan mencari sumber daya. Orang dapat menemukan ratusan sumber daya bermanfaat, tutorial, dan FAQ di Hackr.io mengenai banyak bahasa dan teknologi. Orang juga akan menemukan peta jalan pemrograman untuk berbagai trek dan artikel blog yang membandingkan dan menjelajahi berbagai bagian pengembangan perangkat lunak.

2.12 Visual Basic.Net 2010

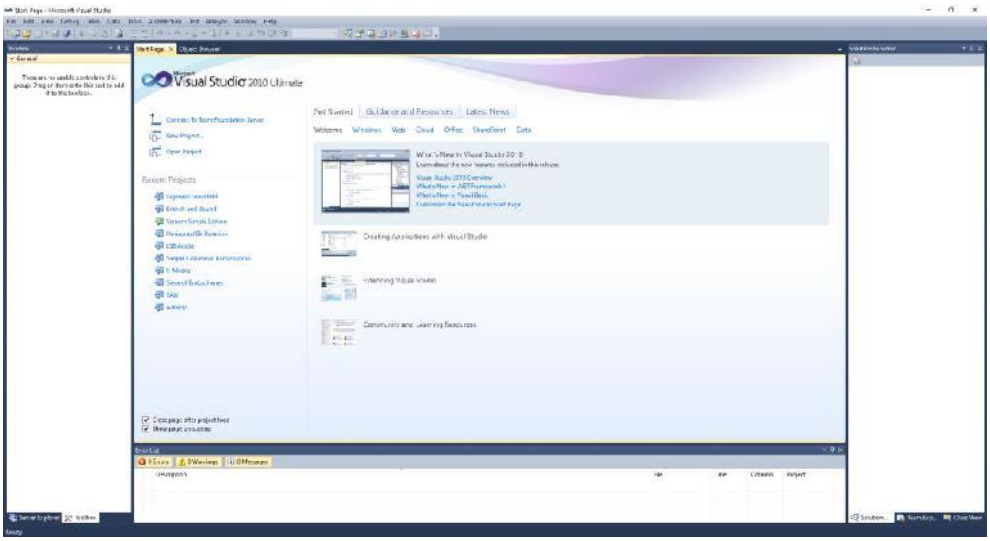
Bahasa Pemrograman *Microsoft Visual Basic .NET* adalah sebuah bahasa pemrograman tingkat tinggi untuk *Microsoft .NET Framework*. Walaupun *VB.NET* ini memang dibuat supaya mudah dipahami dan dipelajari, namun bahasa pemrograman ini juga cukup *powerful* untuk memenuhi kebutuhan dari *programmer* yang berpengalaman. Bahasa pemrograman *Visual Basic .NET* mirip dengan bahasa pemrograman *Visual Basic*, namun keduanya tidak sama".

Bahasa pemrograman *Visual Basic .NET* memiliki struktur penulisan yang mirip dengan bahasa Inggris, di mana hal ini juga menyebabkan kemudahan dalam membaca dan mengerti dari sebuah kode. Di mana dimungkinkan, kata ataupun frasa yang memiliki arti digunakan dan bukannya menggunakan singkatan, akronim ataupun *special characters*".

Pada intinya *Visual Basic.NET* ini adalah sebuah bahasa pemrograman yang berorientasi pada *object*, yang bisa dianggap sebagai evolusi selanjutnya dari bahasa pemrograman *Visual Basic* standar (Wibowo, 2019).

2.12.1 Lingkungan kerja Visual Basic.Net 2010

Pada saat pertama kali dijalankan Visual Basic 2010 Ultimate, akan menampilkan sebuah jendela Splash Visual Studio 2010 Ultimate, setelah jendela Splash Visual Studio 2010 Ultimate muncul kemudian akan keluar sebuah start page Microsoft Visual Studio seperti gambar 2.3.



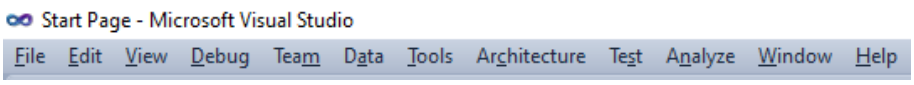
Gambar 2.3 Tampilan Microsoft Visual Studio 2010

2.12.2 Komponen Visual Basic.Net 2010

Pada saat membuka program Visual Basic.Net, ada beberapa komponen yang terlihat. Berikut ini adalah beberapa komponen dari Visual Basic.Net:

1. Menu Bar

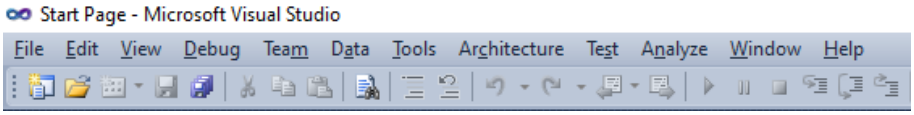
   Menu Bar adalah bagian dari *IDE* yang terdiri atas perintah-perintah untuk mengatur *IDE*, mengedit kode, dan mengeksekusi program. Menu yang terdapat pada menu bar adalah *menu file, edit, view, project, build, debug, data, tools, window* dan *help*. *Menu bar* pada *Visual Sudio* 2010 seperti terlihat pada gambar 2.5.



Gambar 2.4 Tampilan Menu Bar

2. Toolbar

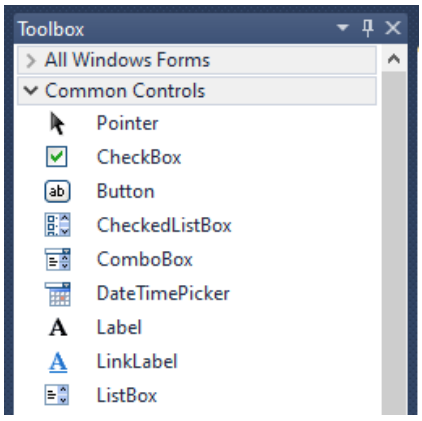
   Fasilitas ini dapat mempercepat pengaksesan perintah-perintah yang ada dalam pemrograman seperti terlihat pada gambar 2.6.



Gambar 2.5 Tampilan Toolbar

3. Toolbox

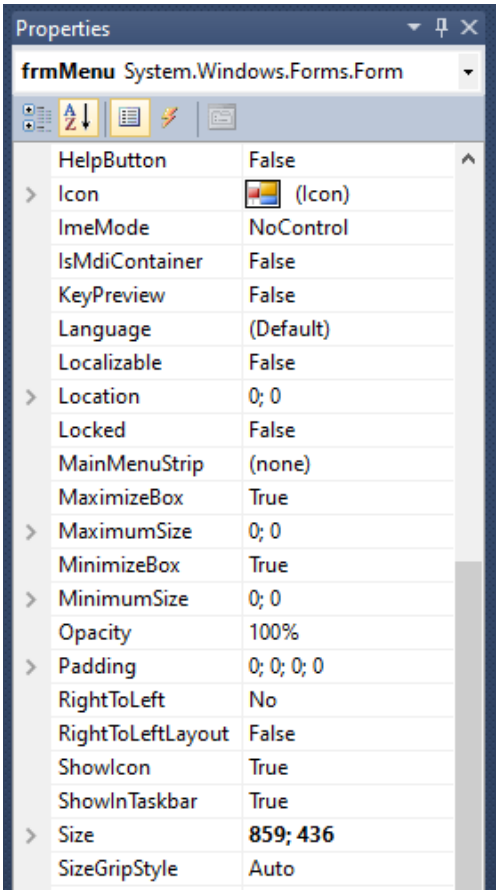
   Sebuah *window* yang berisi tombol-tombol kontrol yang akan seseorang gunakan untuk mendesain atau membangun sebuah *form* atau *report* seperti terlihat pada gambar 2.7.



Gambar 2.6 Tampilan Toolbox

4. Properties Window

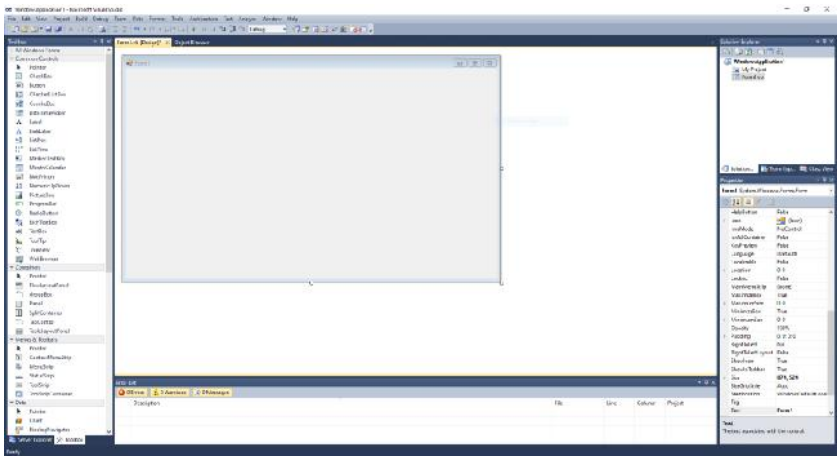
   Properties window adalah tempat menyimpan *property* dari setiap objek control dan komponen.



Gambar 2.7 Tampilan Properties

5. Form

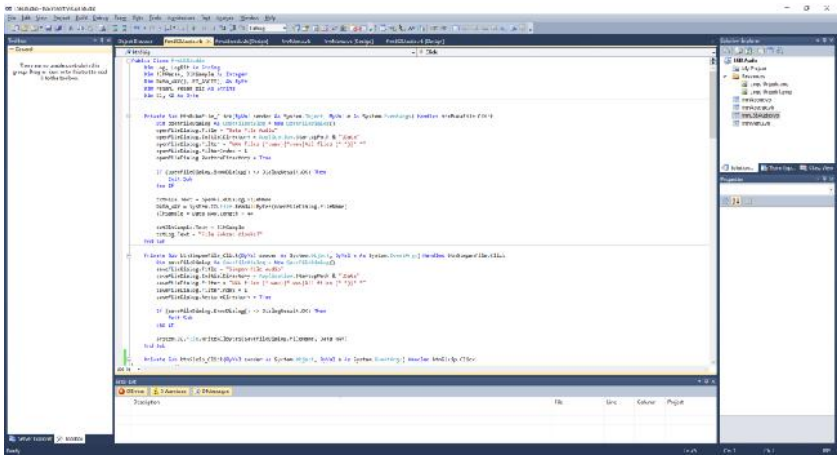
   Form merupakan tempat di mana kontrol-kontrol diletakkan. Form juga berfungsi sebagai tempat pembuatan tampilan atau antarmuka (*user interface)* dari sebuah aplikasi *windows*.



Gambar 2.8 Tampilan Form

6. Code Editor

   Code Editor adalah tempat di mana kita meletakkan atau menuliskan kode program dari program aplikasi kita.



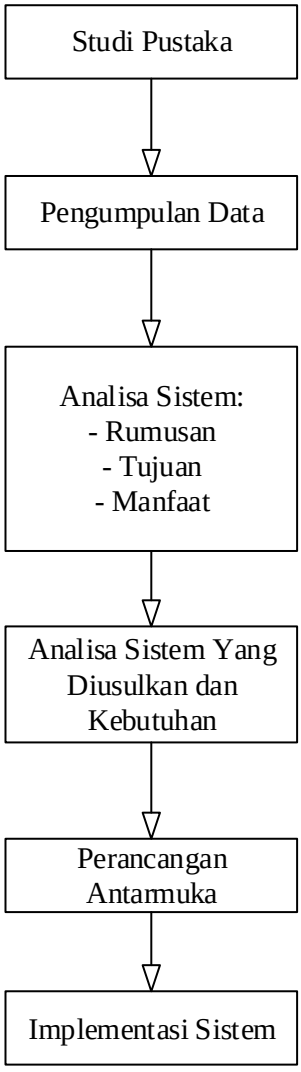
Gambar 2.9 Tampilan Code Editor

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian harus dilakukan untuk menentukan langkah pembuatan program aplikasi. Gambar 3.1 adalah tahapan penelitian dilakukan.



Gambar 3.1 Tahapan Penelitian

Berikut merupakan penjelasan dari gambar tahapan penelitian yang ada di atas:

1. Studi pustaka dilakukan berdasarkan referensi dan buku-buku yang dapat diperoleh dari perpustakaan.
2. Pengumpulan data dilakukan untuk mendapatkan data-data yang berhubungan dengan proses enkripsi dan dekripsi.
3. Analisa sistem dilakukan untuk menentukan arah dan tujuan dari permasalahan yang ada serta menentukan solusi.
4. Analisa sistem yang diusulkan adalah menentukan rancangan dan program aplikasi yang akan menyelesaikan masalah yang ada pada penelitian yang dilakukan.
5. Analisa kebutuhan menentukan perangkat yang dilakukan untuk menyelesaikan masalah yang ada pada penelitian ini.
6. Metode digunakan untuk menentukan algoritma yang digunakan dalam penelitian ini.
7. Desain sistem digunakan untuk menentukan hasil perancangan sistem berdasarkan diagram yang dipergunakan.
8. Pembuatan sistem dilakukan dengan menggunakan bahasa pemrograman *Microsoft Visual Basic.NET 2010*.
9. Implementasi dilakukan untuk menguji kebenaran program aplikasi yang telah dibuat.

3.2 Metode Pengumpulan Data

Dalam mendapatkan data yang akurat, dibutuhkan teknik dalam mengumpulkan data tersebut. Metode ini dilakukan dengan beberapa cara antara lain:

1. Studi Pustaka

   Pengumpulan data-data berupa teori mencari dan mengumpulkan bahan yang berhubungan dengan masalah yang sedang diteliti.

2. Studi Lapangan

   Studi lapangan yaitu kegiatan terjun secara langsung ke lapangan dengan menggunakan teknik pengumpulan data.

3. Observasi

   Observasi merupakan teknik yang digunakan untuk mengumpulkan data dengan cara melakukan pengamatan secara langsung terhadap cara kerja dari enkripsi dan dekripsi pada pengiriman pesan.

3.3 Analisa Sistem

Pengiriman pesan sangat membutuhkan keamanan dan kenyamanan kepada kedua belah pihak, baik pengirim maupun penerima pesan tersebut. Dalam melaksanakan proses pengiriman pesan, sebaiknya pesan tersebut sudah terenkripsi sehingga bebas dari pencurian dan penyalahgunaan data. Informasi yang terkandung dalam pesan tersebut merupakan berita yang harus dilindungi dan dijaga kerahasiaannya.

3.1.1 Analisa Sistem Yang Berjalan

Pengiriman pesan sering tidak menggunakan pengamanan yang baik sehingga dapat terjadi pencurian dan penyalahgunaan data. Dalam mengirimkan pesan, pengguna sering mengabaikan keamanan yang dapat merugikan pihak pengirim dan penerima. Pesan yang dikirim memiliki informasi yang sangat rahasia sehingga apabila informasi tersebut berhasil dicuri, maka akan sangat memberikan dampak negatif kepada pemilik pesan tersebut.

3.1.2 Analisa Sistem Yang Diusulkan

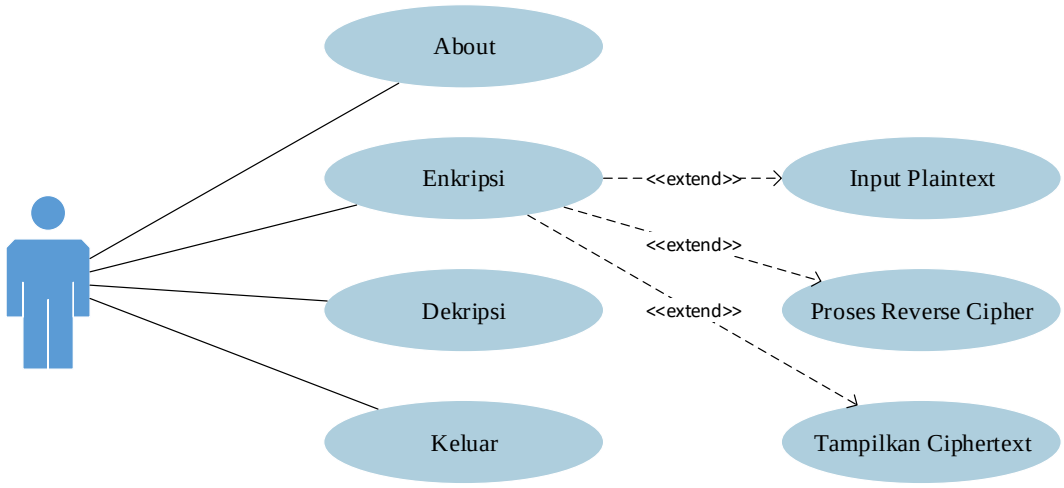
Penelitian diharapakan memiliki fungsi yang baik dalam melakukan pengamanan pesan. Pesan yang dikirim akan dienkripsi terlebih dahulu agar dapat menghindari proses pencurian dan penyalahgunaan data. Apabila pihak yang tidak bertanggung jawab berhasil mencuri pesan dari pemilik pesan, maka pesan tersebut sudah tidak dapat dibaca lagi sehingga berita atau informasi yang terkandung di dalam pesan tersebut akan menjadi aman.

Reverse Cipher merupakan teknik yang digunakan dalam penelitian ini dalam hal mengamankan pesan. Algoritma ini akan mengubah susunan karakter dalam *plaintext* sehingga tidak dapat dibaca dengan baik oleh peretas pesan tersebut. Jenis algoritma ini termasuk kriptografi transposisi dimana susunan dan pola karakter akan diubah letaknya. Posisi karakter akan ditukar dari belakang ke depan dan dari depan ke belakang.

3.2 Rancangan UML

3.2.1 *Use Case Diagram* Enkripsi

Berikut ini adalah *Use Case Diagram* yang digunakan dalam melakukan proses enkripsi pesan.

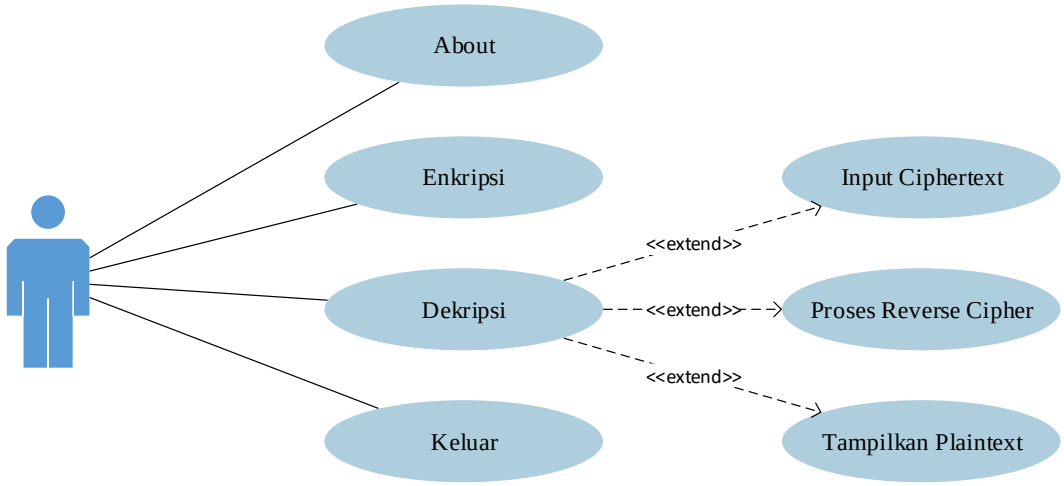


Gambar 3.2 *Use Case Diagram* Enkripsi

Gambar 3.2 merupakan rancangan *Use Case Diagram* enkripsi. Diagram tersebut menjelaskan tahap pertama adalah pengguna masuk ke menu utama yang mengizinkan pengguna memilih menu enkripsi. Pengguna diarahkan untuk memasukkan *plaintext* dan selanjutnya pengguna dapat menekan tombol enkripsi untuk memulai proses enkripsi. Hasil enkripsi dapat dilihat pada *textbox ciphertext.* Selanjutnya, *Ciphertext* hasil enkripsi dapat diduplikatkan untuk disimpan dalam suatu file.

3.2.2 *Use Case Diagram* Dekripsi

Berikut ini adalah *Use Case Diagram* yang digunakan dalam melakukan proses dekripsi pesan.

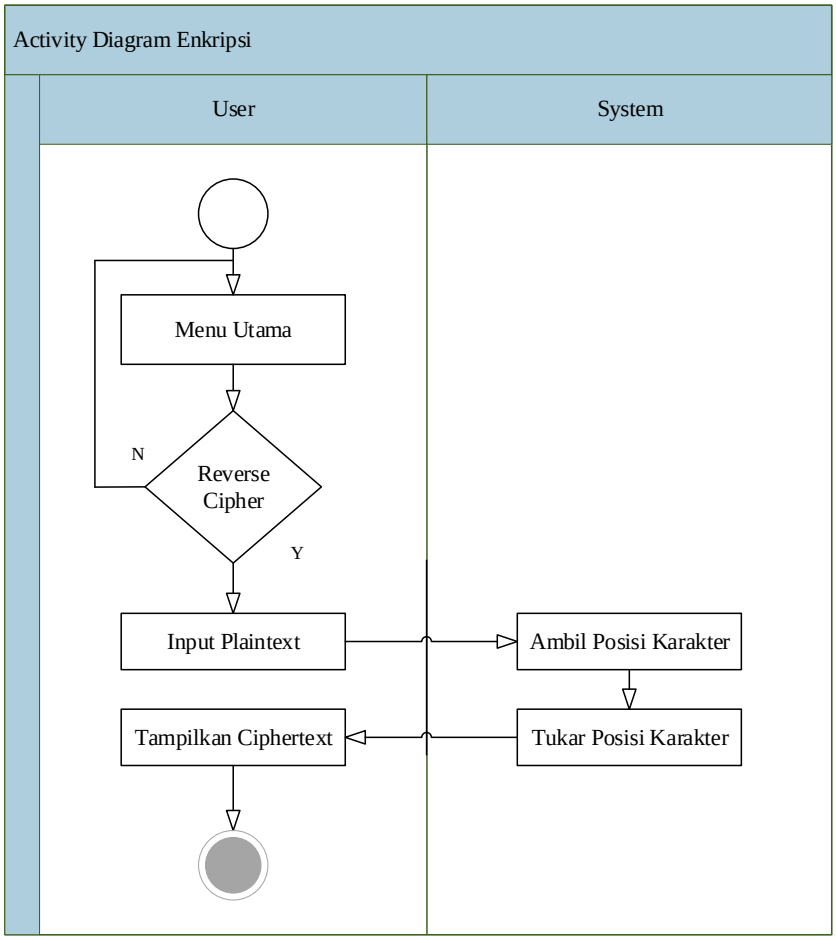


Gambar 3.3 *Use Case Diagram* Dekripsi

Gambar 3.3 merupakan rancangan *Use Case Diagram* dekripsi. Diagram tersebut menjelaskan tahap pertama adalah pengguna masuk ke menu utama yang mengizinkan pengguna memilih menu enkripsi. Pengguna diarahkan untuk memasukkan *ciphertext* dan selanjutnya pengguna dapat menekan tombol dekripsi untuk memulai proses dekripsi. Hasil dekripsi dapat dilihat pada *textbox plaintext.* Selanjutnya, *plaintext* hasil dekripsi dapat diduplikatkan untuk disimpan dalam suatu file.

3.2.3 *Activity Diagram* Enkripsi

Berikut ini adalah *Activity Diagram* proses enkripsi.

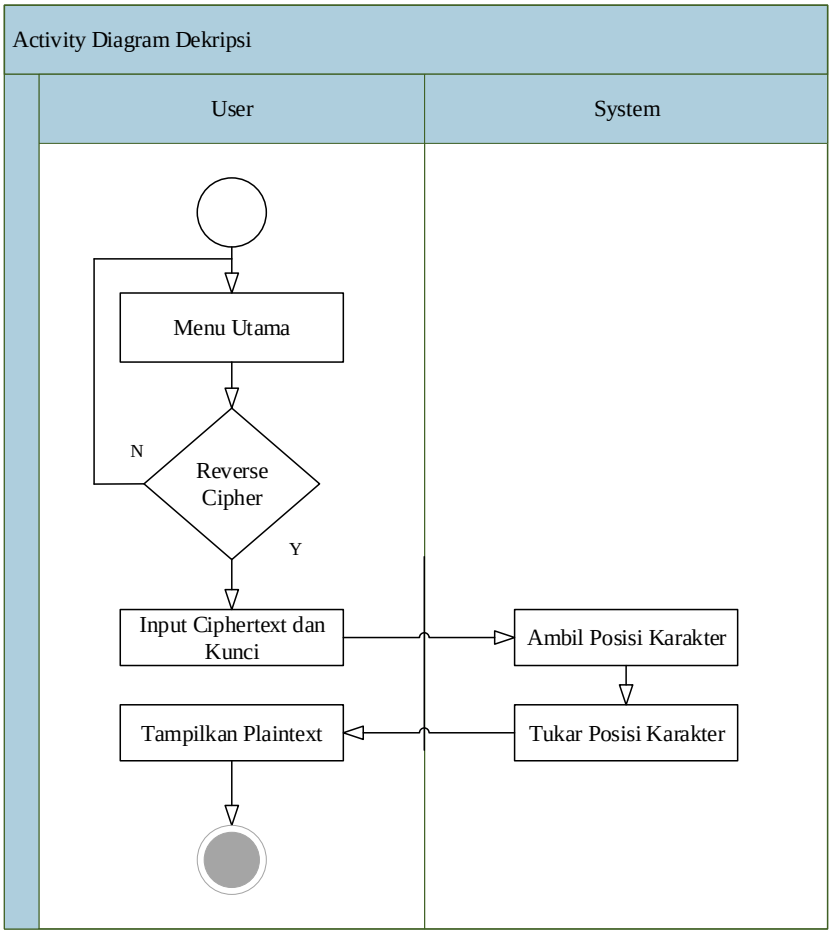


Gambar 3.4 *Activity Diagram* Enkripsi

Gambar 3.4 merupakan rancangan *Activity Diagram* enkripsi pesan dengan menggunakan algoritma *Reverse Cipher*. Pada *Activity Diagram* enkripsi, pengguna akan menginputkan informasi yang digunakan untuk proses enkripsi. Sistem akan memproses *Plaintext* hingga menghasilkan *Ciphertext*.

3.2.4 *Activity Diagram* Dekripsi

Berikut ini adalah *Activity Diagram* proses dekripsi.

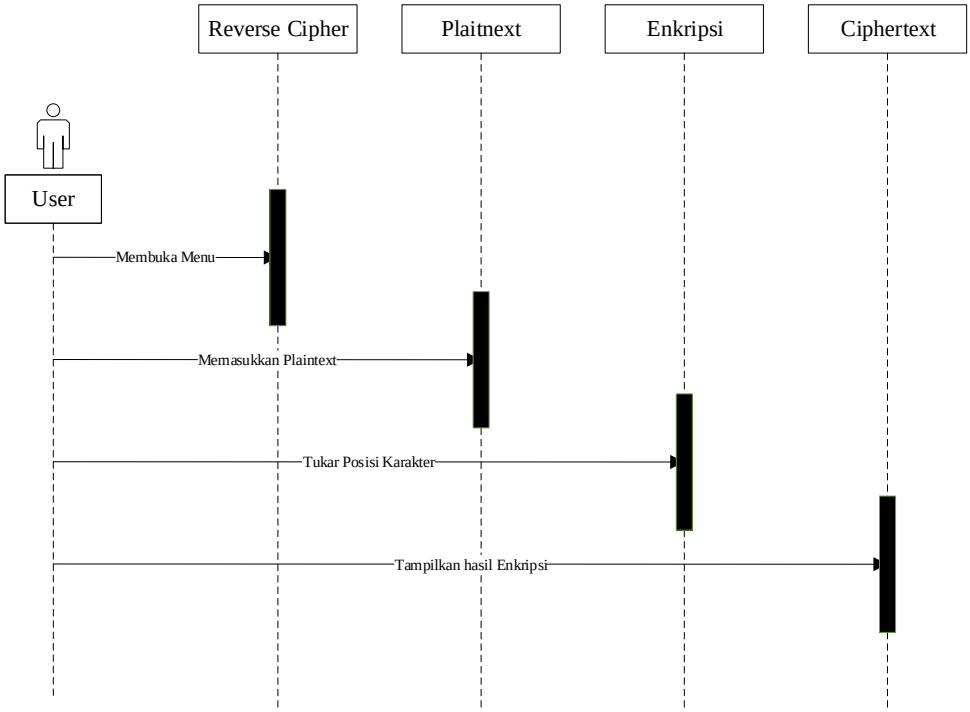


Gambar 3.5 *Activity Diagram* Dekripsi

Gambar 3.5 ini merupakan rancangan *Activity Diagram* dekripsi dengan menggunakan algoritma *Reverse Cipher*. Pada *Activity Diagram* dekripsi, pengguna memasukkan *Ciphertext* dan kunci. Sistem akan memproses dekripsi dan akan menghasilkan *Plaintext*.

3.2.5 *Sequence Diagram* Enkripsi

Berikut ini adalah *Sequence Diagram* proses enkripsi yang menjelaskan alur dari proses enkripsi menggunakan algoritma *Reverse Cipher*.

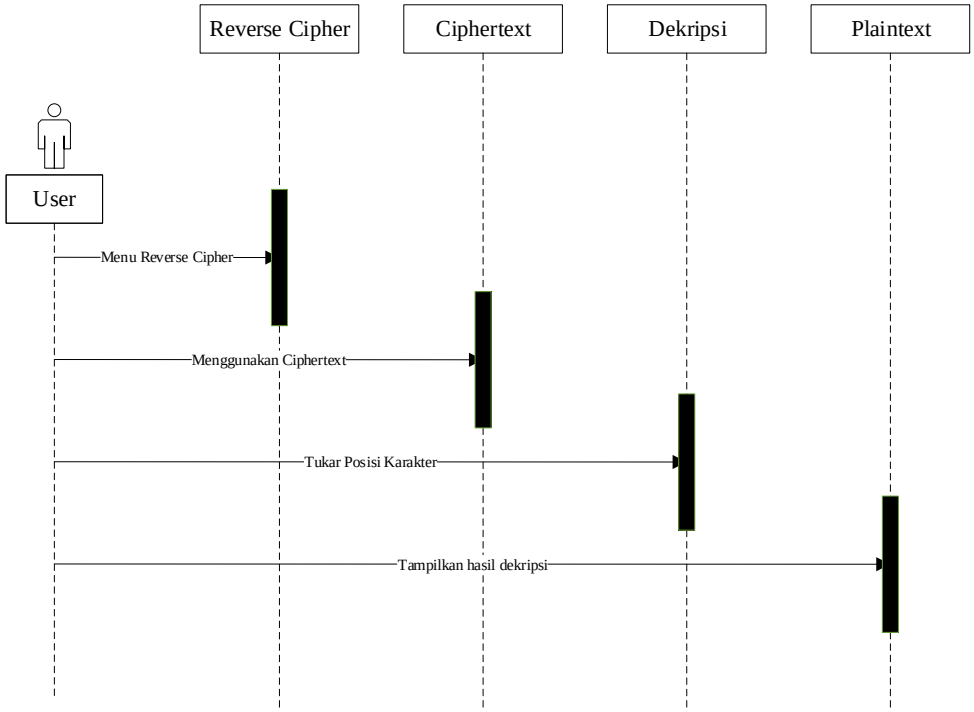


Gambar 3.6 *Sequence Diagram* Enkripsi

Gambar 3.6 merupakan rancangan *Sequence Diagram* enkripsi. Pada *Sequence Diagram* enkripsi, pengguna terlebih dahulu memasuki menu *Reverse Cipher* dan kemudian pengguna dapat memasukkan *plaintext* yang akan digunakan dalam proses enkripsi. Tidak ada penggunaan kunci pada algoritma ini. Proses enkripsi akan menukar karakter berdasarkan letak dan posisinya. Hasil enkripsi akan ditampilkan pada *textbox* yang tersedia.

3.2.6 *Sequence Diagram* Dekripsi

Berikut ini adalah *Sequence Diagram* proses dekripsi yang menjelaskan alur dari proses dekripsi menggunakan algoritma *Reverse Cipher*.



Gambar 3.7 *Sequence Diagram* Dekripsi

Gambar 3.7 merupakan rancangan *Sequence Diagram* dekripsi. Pada *Sequence Diagram* dekripsi, pengguna terlebih dahulu memasuki menu *Reverse Cipher* dan kemudian pengguna dapat menggunakan *ciphertext* yang sudah dihasilkan sebelumnya pada proses enkripsi. Tidak ada penggunaan kunci pada proses dekripsi. Proses dekripsi akan menukar karakter berdasarkan letak dan posisinya. Hasil dekripsi akan ditampilkan pada *textbox* yang tersedia.

3.3 Analisis *Reverse Cipher*

Algoritma *Reverse Cipher* adalah algoritma yang sangat mudah digunakan. Algoritma ini bekerja dengan cara membalik susunan karakter pada *plaitext.* Setelah susunan diubah, maka *ciphertext* dihasilkan. Tidak ada penggunaan kunci pada model algoritma *Reverse Cipher.* Jenis algoritma ini termasuk jenis kriptografi transposisi, yaitu menukar posisi karakter tanpa menggantikan karakter tersebut dengan karakter selain yang ada pada deretan karakter tersebut. Berikut ini diberikan contoh perhitungan dari proses enkripsi algoritma *Reverse Cipher*.

Plaintext									
D	A	N	A	U		T	O	B	A

Ciphertext									
A	B	O	T		U	A	N	A	D

3.4 Perancangan Antarmuka

Antarmuka adalah hal yang paling penting dilakukan untuk menghubungkan pengguna dan program aplikasi. Beberapa tampilan akan dilakukan pada perancangan antarmuka pada penelitian ini.

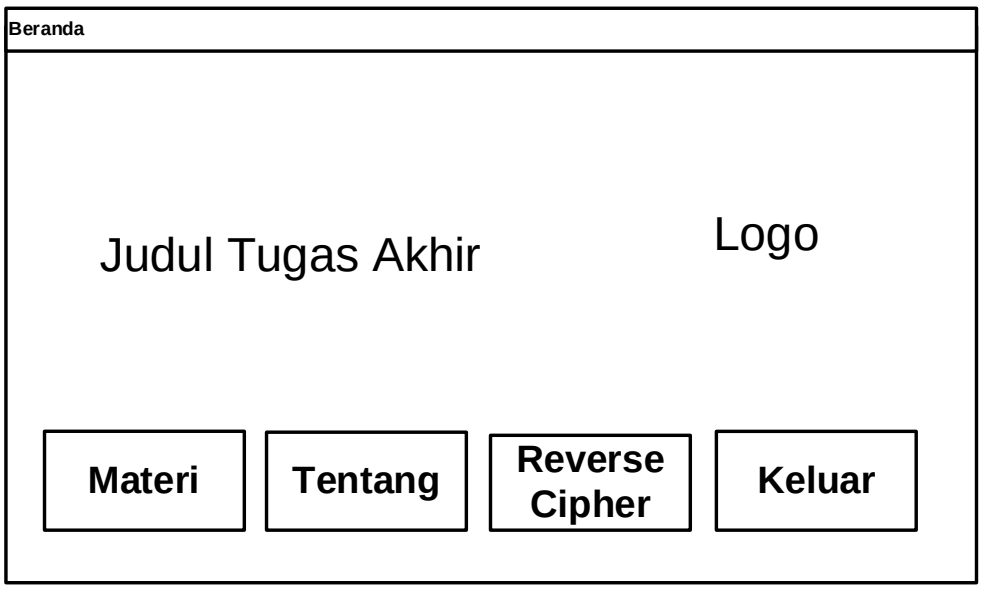
3.4.1 Rancangan Judul

Rancangan judul adalah tampilan yang akan pertama sekali tampil pada saat program aplikasi digunakan. Gambar. 3.8 adalah hasil perancangan judul.

Gambar 3.8 Rancangan Judul

3.4.2 Rancangan Tampilan Beranda

Gambar 3.8 adalah hasil perancangan beranda yang digunakan pada penelitian ini.



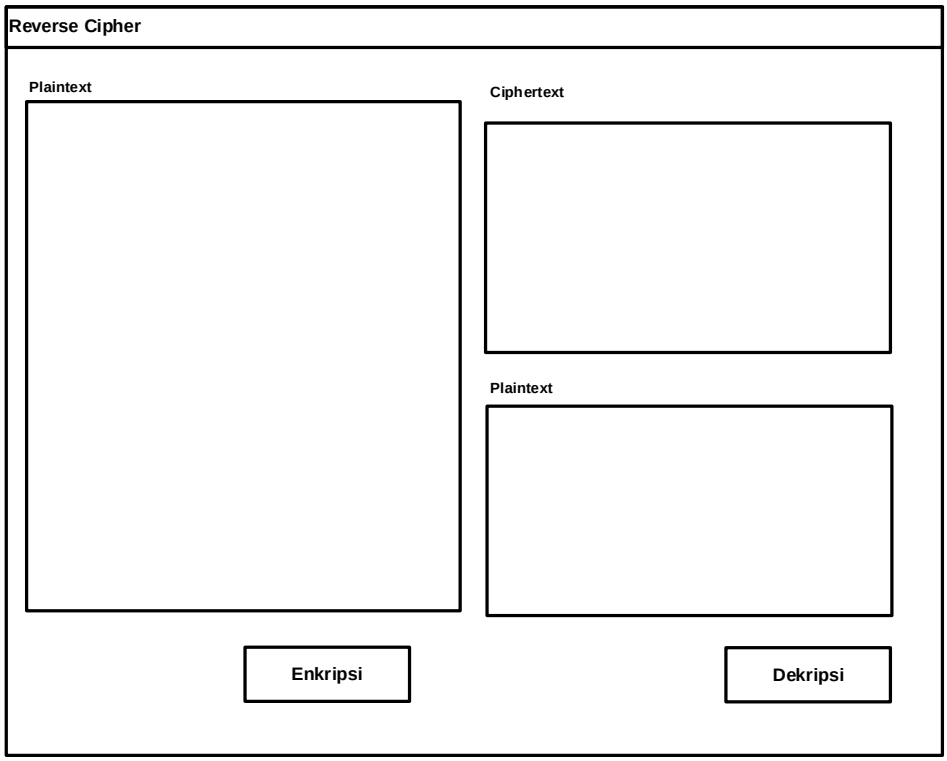
Gambar 3.9 Rancangan Menu Utama

Tampilan ini memiliki berapa sub-menu antara lain:

1. Materi
2. Reverse Cipher
3. Tentang
4. Keluar

3.4.3 Rancangan Tampilan *Reverse Cipher*

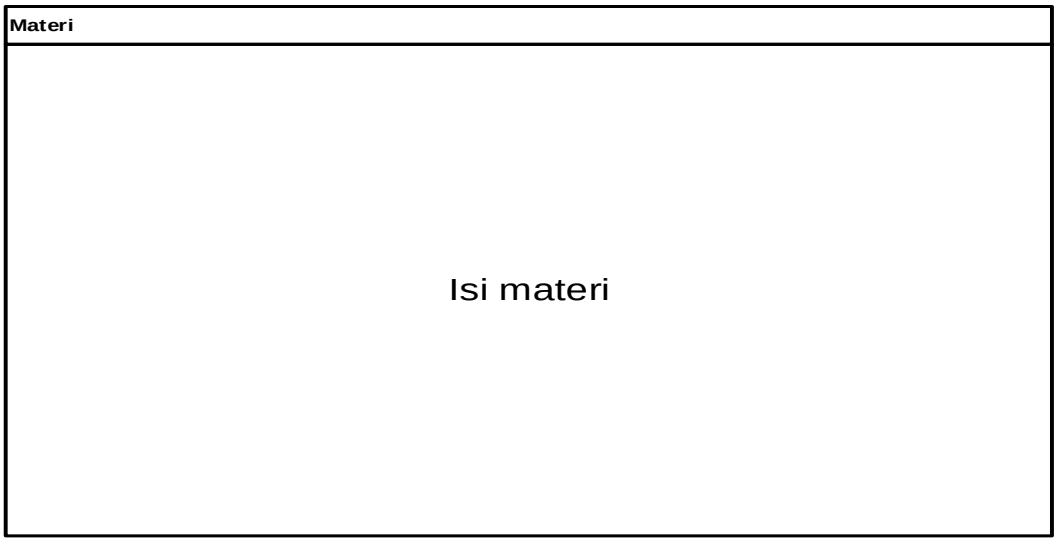
Gambar 3.10 adalah rancangan tampilan dari algoritma *Reverse Cipher* yang ada pada penelitian ini.



Gambar 3.10 Rancangan *Reverse Cipher*

3.4.4 Rancangan Tampilan Materi

Gambar 3.11 merupakan rancangan tampilan materi yang akan memberikan gambaran tentang algoritma *Reverse Cipher*.

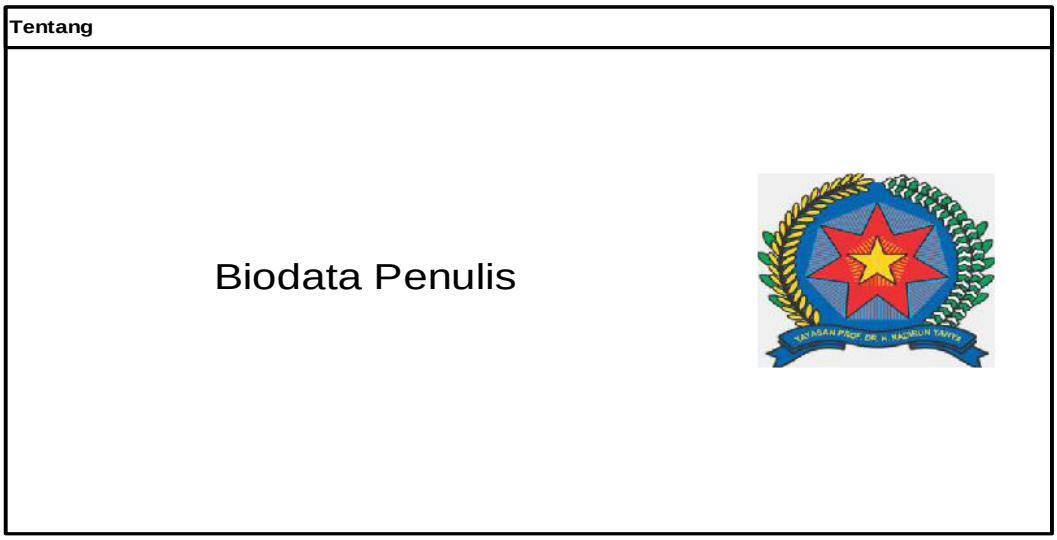


Gambar 3.11 Rancangan Materi

3.4.5 Rancangan Tampilan Tentang

Gambar 3.12 merupakan rancangan tampilan tentang aplikasi. Pada rancangan tampilan ini nantinya pengguna dapat melihat penjelasan singkat biodata penulis. Pada tampilan ini akan menampilan antara lain:

1. Nama
2. NPM
3. Fakultas
4. Universitas



Gambar 3.12 Rancangan Tentang

BAB IV

HASIL DAN PEMBAHASAN

4.1 Kebutuhan Perangkat Keras dan Lunak

Sistem yang telah dirancang membutuhkan perangkat kerass dan perangkat lunak dalam mendukung kinerja program aplikasi tersebut. Berikut ini adalah kebutuhan perangkat tersebut:

1. *Hardware* (Perangkat Keras)

   Untuk menjalankan sistem ini, penulis menggunakan laptop dengan spesifikasi RAM 2GB, Processor Intel Core i3, Hard drive 500GB dan Display 14".

2. Software (Perangkat Lunak)

   Sedangkan pada sisi software, penulis menggunakan beberapa perangkat lunak yaitu:

   a. Windows 7
   b. Microsoft Visual Studio 2010
   c. Microsoft Word 2019
   d. Microsoft Excel 2019
   e. Microsoft Visio 2019
   f. Snipping Tool

4.2 Tampilan Program Aplikasi

Program aplikasi yang sudah dibuat harus dapat diimplementasikan dan dibuktikan kebenarannya. Program aplikasi memiliki beberapa antarmuka yang dapat digunakan oleh pengguna dalam melakukan proses enkripsi dan dekripsi dengan algoritma *Reverse Cipher*. Berikut ini adalah hasil tampilan antarmuka yang diperoleh berdasarkan perancangan aplikasi.

4.2.1 Tampilan Halaman Judul

Halaman judul adalah halaman yang pertama sekali muncul ketika program aplikasi dijalankan. Gambar 4.1 adalah hasil tampilan halaman judul.

Gambar 4.1 Halaman Judul

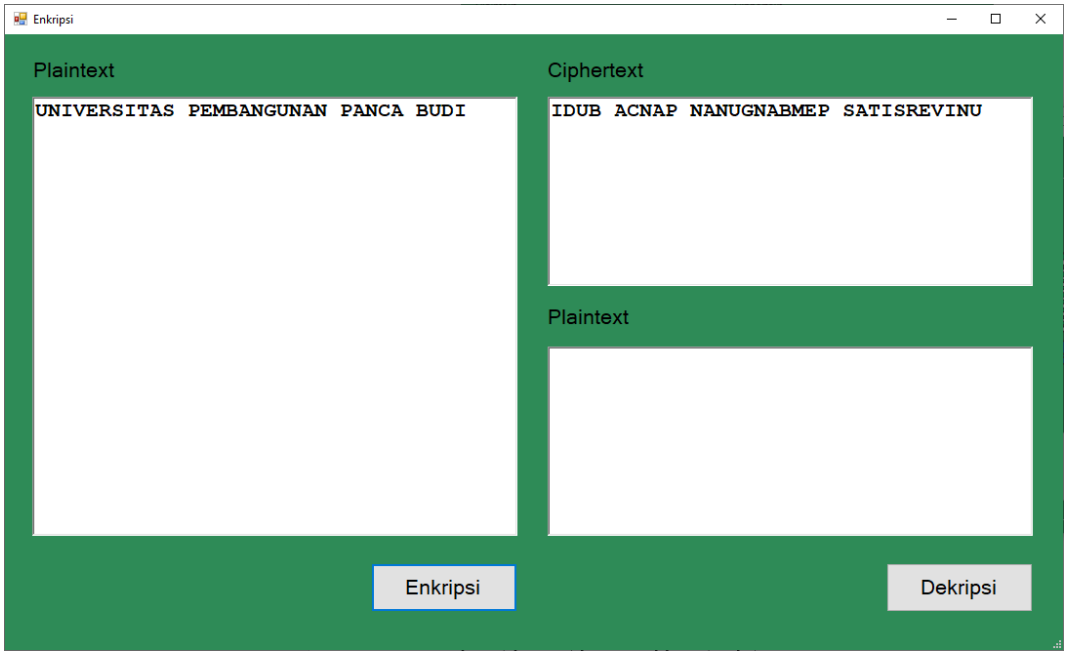
4.2.2 Tampilan Halaman Beranda

Halaman beranda merupakan halaman utama dari program aplikasi algoritma *Reverse Cipher*. Pengguna melakukan pemilihan menu-menu lainnya dari menu ini. Gambar 4.2 adalah hasil tampilan beranda.

Gambar 4.2 Halaman Beranda

4.2.3 Tampilan Halaman Enkripsi

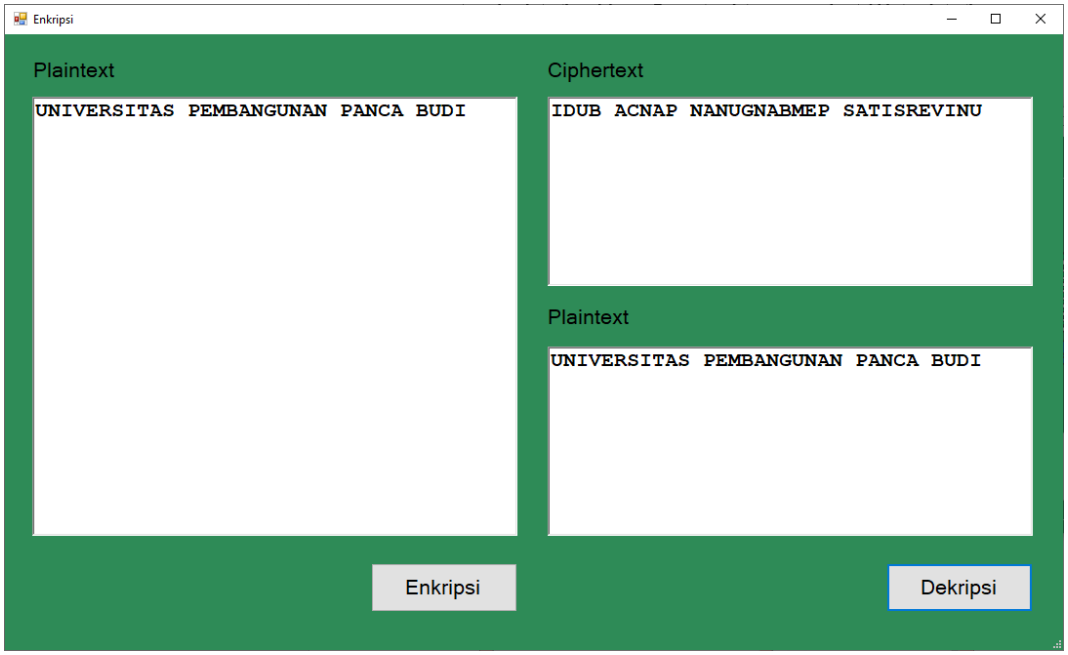
Gambar 4.3 merupakan tampilan dari halaman enkripsi pesan dengan algoritma *Reverse Cipher*. Teks yang akan digunakan akan langsung dimasukkan pada *textbox*. Pengguna dapat melakukan proses enkripsi pesan tersebut dengan cara menekan tombol Enkripsi. Hasil proses enkripsi algoritma *Reverse Cipher* akan ditampilkan pada *textbox ciphertext*.



Gambar 4.3 Halaman Enkripsi

4.2.4 Tampilan Halaman Dekripsi

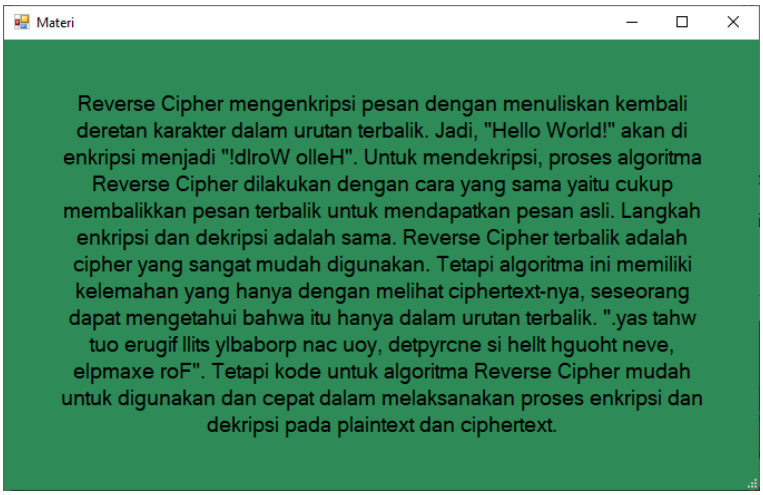
Gambar 4.4 ini merupakan tampilan dari halaman dekripsi algoritma *Reverse Cipher*. *Ciphertext* yang akan digunakan adalah hasil proses enkripsi sebelumnya dan akan berada pada *textbox ciphertext*. Pengguna dapat melakukan proses dekripsi pesan tersebut dengan cara menekan tombol Dekripsi. Hasil dekripsi algoritma *Reverse Cipher* akan ditampilkan pada *textbox plaintext*.



Gambar 4.4 Halaman Dekripsi

4.2.5 Halaman Materi

Gambar 4.5 adalah tampilan dari halaman materi yang menjelaskan secara singkat tentang algoritma *Reverse Cipher*.



Gambar 4.5 Halaman Materi

4.2.6 Halaman Tentang

Gambar 4.6 merupakan tampilan dari halaman biodata penulis. Pada tampilan ini nantinya pengguna dapat melihat informasi tentang pemilik program aplikasi.

Gambar 4.6 Halaman Tentang

4.3 Kode Program

Berikut ini akan ditampilkan kode program aplikasi untuk melakukan proses enkripsi dan dekripsi menggunakan algoritma *Reverse Cipher*.

```
Public Class frmReverse
    Dim Log As String
    Dim PT, CT As String
    Dim Blok, Sisa As Integer

    Private Sub btnEnkrip_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEnkrip.Click
        PT = txtPT.Text

        If PT.Length > 1000 Then
            MessageBox.Show("Panjang karakter tidak boleh melebihi dari
1000 karakter!", "Peringatan")
            Return
```

```
        End If

        CT = ""
        For i = 0 To PT.Length - 1
            CT &= PT(PT.Length - 1 - i)
        Next

        txtCT.Text = CT
    End Sub

    Private Sub btnDekripsi_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDekripsi.Click
        CT = txtCT.Text

        PT = ""
        For i = 0 To CT.Length - 1
            PT &= CT(CT.Length - 1 - i)
        Next

        txtPT2.Text = PT
    End Sub

End Class
```

BAB V

PENUTUP

5.1 Kesimpulan

Berikut merupakan kesimpulan yang penulis buat berdasarkan pembahasan pada implementasi dan penggunaan algoritma *Reverse Cipher*:

1. Perancangan sistem dilakukan dengan menggunakan bahasa pemrograman Microsoft Visual Basic.NET 2010.
2. Enkripsi pada algoritma Reverse Cipher bekerja dengan cara menukar posisi karakter dari depan ke belakang.
3. Dekripsi pada algoritma Reverse Cipher bekerja dengan cara menukar kembali posisi karakter dengan arah sebaliknya.
4. Tidak ada penggunaan kunci pada algoritma Reverse Cipher.

5.2 Saran

Berikut merupakan saran yang penulis paparkan berdasarkan implementasi dan penggunaan program aplikasi algoritma Reverse Cipher:

1. Program aplikasi masih berbasis desktop dan tidak online, sebaiknya dikembangkan sehingga dapat diakses secara online.
2. Program aplikasi hendaknya dapat dikembangkan sehingga dapat dipasang pada perangkat mobile.
3. Hendaknya algoritma Reverse Ciphert dapat menggunakan kunci untuk meningkatkan keamanan.

4. Hendaknya program aplikasi dapat memproses karakter lebih dari 1000 karakter.

DAFTAR PUSTAKA

Amin, M. M. (2016). Implementasi Kriptografi Klasik Pada Komunikasi Berbasis Teks. *Jurnal Pseudocode*, *3*(2).

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

Batubara, Supina. "Analisis perbandingan metode fuzzy mamdani dan fuzzy sugeno untuk penentuan kualitas cor beton instan." IT Journal Research and Development 2.1 (2017): 1-11.

Fitriani, W., Rahim, R., Oktaviana, B., & Siahaan, A. P. U. (2017). Vernam Encypted Text in End of File Hiding Steganography Technique. Int. J. Recent Trends Eng. Res, 3(7), 214-219.

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).

Hariyanto, E., Lubis, S. A., & Sitorus, Z. (2017). Perancangan prototipe helm pengukur kualitas udara. KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer), 1(1).

Iqbal, M., Siahaan, A. P. U., Purba, N. E., & Purwanto, D. (2017). Prim's Algorithm for Optimizing Fiber Optic Trajectory Planning. Int. J. Sci. Res. Sci. Technol, 3(6), 504-509.

Isa, I. G. T., & Hartawan, G. P. (2017). Perancangan Aplikasi Koperasi Simpan Pinjam Berbasis Web (Studi Kasus Koperasi Mitra Setia). *Jurnal Ilmiah Ilmu Ekonomi (Jurnal Akuntansi, Pajak Dan Manajemen)*, *5*(10), 139–151.

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

Mallu, S. (2015). Sistem Pendukung Keputusan Penentuan Karyawan Kontrak Menjadi Karyawan Teatap Menggunakan Metode TOPSIS. *Jurnal Imliah Teknologi Informasi Terapan*, *1*(2), 36–42.

Muttaqin, Muhammad. "Analisa Pemanfaatan Sistem Informasi E-Office Pada Universitas Pembangunan Panca Budi Medan Dengan Menggunakan Metode Utaut." Jurnal Teknik dan Informatika 5.1 (2018): 40-43.

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.

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.

Siagian, P., & Fahreza, F. (2020, February). Rekayasa Penanggulangan Fluktuasi Daya Pembangkit Listrik Tenaga Angin Dengan Vehicle To Grid (V2G). In Seminar Nasional Teknologi Komputer & Sains (SAINTEKS) (Vol. 1, No. 1, Pp. 356-361).

Siagian, P., Syafruddin, H. S., & Tharo, Z. (2020, September). Pengaruh Tekanan Terhadap Inception Partial Discharge Pada Bahan Dielektrik Komposit Dan Non-Komposit. In Seminar Nasional Teknik (SEMNASTEK) UISU (Vol. 3, No. 1, Pp. 134-141).

Siahaan, A. P. U., Ikhwan, A., & Aryza, S. (2018). A Novelty Of Data Mining For Promoting Education Based On FP-Growth Algorithm

Sopyan, Y., Supriyadi, S., & Kurniadi, E. (2016). Implementasi Sistem Pendukung Keputusan Penerimaan Siswa baru Menggunakan Metode Simple Additive Weighting (Studi Kasus : SMK Negeri 3 Kuningan). *Jurnal Nuansa Informatika*, *11*(1).

Tarigan, A. D., & Pulungan, R. (2018). Pengaruh Pemakaian Beban Tidak Seimbang Terhadap Umur Peralatan Listrik. RELE (Rekayasa Elektrikal Dan Energi): Jurnal Teknik Elektro, 1(1), 10-15.

Wibowo, P., Lubis, S. A., & Hamdani, Z. T. (2017). Smart Home Security System Design Sensor Based On Pir And Microcontroller. International Journal Of Global Sustainability, 1(1), 67-73.

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

INTERNET :

Ayushi, M. (2010). A Symmetric Key Cryptographic Algorithm. *International Journal of Computer Applications*, *1*(15), 1–6. https://doi.org/10.5120/331-502

Barone, L., Williams, J., & Micklos, D. (2017). Unmet needs for analyzing biological big data: A survey of 704 NSF principal investigators. *PLOS Computational Biology*, *13*(10), e1005755. https://doi.org/10.1371/journal.pcbi.1005755

Hendini, A. (2016). Pemodelan UML Sistem Informasi Monitoring Penjualan Dan Stok Barang. *Jurnal Khatulistiwa Informatika*, *4*(2), 107–116. https://doi.org/10.31294/jki.v4i2.1262.g1027

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

Pabokory, F. N., Astuti, I. F., & Kridalaksana, A. H. (2015). Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard. *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, *10*, 22. https://doi.org/10.30872/jim.v10i1.23

Putri, G. G., Setyorini, W., & Rahayani, R. D. (2018). Analisis Kriptografi Simetris AES dan Kriptografi Asimetris RSA pada Enkripsi Citra Digital. *ETHOS (Jurnal Penelitian Dan Pengabdian)*, *6*(2), 197–207. https://doi.org/10.29313/ethos.v6i2.2909

S., G., L. Ribeiro, A. R., & David, E. (2012). Asymmetric Encryption in Wireless Sensor Networks. In *Wireless Sensor Networks - Technology and Protocols*. InTech. https://doi.org/10.5772/48464

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

Sun, Y., Zhang, J., Xiong, Y., & Zhu, G. (2014). Data Security and Privacy in Cloud Computing. *International Journal of Distributed Sensor Networks*, *10*(7), 190903. https://doi.org/10.1155/2014/190903

Wasserkrug, S., Dalvi, N., Munson, E. V., Gogolla, M., Sirangelo, C., Fischer- Hübner, S., Ives, Z., Velegrakis, Y., Bevan, N., Jensen, C. S., & Snodgrass, R. T. (2019). Unified Modeling Language. In *Encyclopedia of Database Systems* (pp. 3232–3239). Springer US. https://doi.org/10.1007/978-0-387-39940-9_440

Zhang, D., Tsotras, V. J., Levialdi, S., Grinstein, G., Berry, D. A., Gouet-Brunet, V., Kosch, H., Döller, M., Döller, M., Kosch, H., Maier, P., Bhattacharya, A., Ljosa, V., Nack, F., Bartolini, I., Gouet-Brunet, V., Mei, T., Rui, Y., Crucianu, M., … Pitoura, E. (2009). Indexed Sequential Access Method. In *Encyclopedia of Database Systems* (pp. 1435–1438). Springer US. https://doi.org/10.1007/978-0-387-39940-9_738