PENYANDIAN KRIPTOGRAFI METODE CAESAR CIPHER DENGAN MENGGUNAKAN MODULO 256

Skripsi 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	: DIAN SYAHFITRI
NPM	: 1514370266
PROGRAM STUDI	: SISTEM KOMPUTER

UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS DAN TEKNOLOGI
MEDAN
2020

LEMBAR PENGESAHAN

PENYANDIAN KRIPTOGRAFI METODE CAESAR CIPHER DENGAN MENGGUNAKAN MODULO 256

Disusun Oleh :

NAMA	**: DIAN SYAHFITRI**
NPM	**: 1514370266**
PROGRAM STUDI	**: SISTEM KOMPUTER**

Skripsi Telah Disetujui Oleh Dosen Pembimbing Skripsi
Pada Tanggal :

Dosen Pembimbing I	Dosen Pembimbing II
Andysah Putera Utama Siahaan, S.Kom., M.Kom., Ph.D	**Ranti Eka Putri, S.Kom., M.Kom**

Mengetahui,

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

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama	: DIAN SYAHFITRI
NPM	: 1514370266
Prodi	: SISTEM KOMPUTER
Konsentrasi	: KEAMANAN JARINGAN KOMPUTER
Judul Skripsi	: PENYANDIAN KRIPTOGRAFI METODE CAESAR CIPHER DENGAN MENGGUNAKAN MODULO 256

Dengan ini menyatakan bahwa :

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

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

Medan, 26 November 2019

Yang membuat pernyataan

DIAN SYAHFITRI

UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km 4,5 Medan Fax. 061-8458077 PO.BOX : 1099 MEDAN

PROGRAM STUDI TEKNIK ELEKTRO	(TERAKREDITASI)
PROGRAM STUDI ARSITEKTUR	(TERAKREDITASI)
PROGRAM STUDI SISTEM KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI TEKNIK KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI AGROTEKNOLOGI	(TERAKREDITASI)
PROGRAM STUDI PETERNAKAN	(TERAKREDITASI)

PERMOHONAN JUDUL TESIS / SKRIPSI / TUGAS AKHIR*

Saya yang bertanda tangan di bawah ini :

Nama Lengkap	: DIAN SYAHFITRI
Tempat/Tgl. Lahir	: MEDAN / 18 Desember 1996
Nomor Pokok Mahasiswa	: 1514370266
Program Studi	: Sistem Komputer
Konsentrasi	: Keamanan Jaringan Komputer
Jumlah Kredit yang telah dicapai	: 141 SKS, IPK 3.49
Nomor Hp	: 082304134001

Dengan ini mengajukan judul sesuai bidang ilmu sebagai berikut :

No.	Judul
1.	Penyandian Kriptografi Metode Caesar Cipher Dengan Menggunakan Modulo 256

atan : Diisi Oleh Dosen Jika Ada Perubahan Judul

oret Yang Tidak Perlu

Rektor I,

(Ir. Bhakti Alamsyah, M.T., Ph.D.)

Medan, 01 Agustus ~~16 Desember~~ 2019

Pemohon,

(Dian Syahfitri)

Tanggal : 17-8-2019

Disahkan oleh :
Dekan

(Hamdani, ST., MT)

Tanggal : .....

Disetujui oleh:
Ka. Prodi Sistem Komputer

(Eko Hariyanto, S.Kom., M.Kom)

Tanggal : .....

Disetujui oleh :
Dosen Pembimbing I :

(Andysah Putera Utama Siahaan, S.Kom.,M.Kom.,Ph.D.)

Tanggal : .....

Disetujui oleh:
Dosen Pembimbing II:

(Ranti Eka Putri, S.Kom., M.Kom)

UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI
Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id
Medan - Indonesia

versitas : Universitas Pembangunan Panca Budi
ultas : SAINS & TEKNOLOGI
en Pembimbing I : Andysah putera Utama Siahaan, S.kom., M.kom., PhD.
en Pembimbing II : Ranti Eka putri, S.kom., M.kom
na Mahasiswa : DIAN SYAHFITRI
usan/Program Studi : Sistem Komputer
nor Pokok Mahasiswa : 1514370266
jang Pendidikan : Strata Satu (S1)
ul Tugas Akhir/Skripsi : Penyandian kriptografi metode caesar cipher dengan menggunakan modulo 256

TANGGAL	PEMBAHASAN MATERI	PARAF	KETERANGAN
/8	Revisi Judul		
/8 2019	Acc Seminar Judul		
/10	Revisi Bab I		
/10	Revisi Bab II		
/10	Revisi Bab III		
/11	Revisi Bab IV		
/11	Revisi Bab IV, V		
/11	Acc Seminar Hasil		
/12,	Acc Sidang		
/2020	Acc Jilid		

Medan, 05 Agustus 2020
Diketahui/Disetujui oleh :
Dekan

Hamdani, ST., MT

t yang tidak perlu

UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI
Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id
Medan - Indonesia

versitas : Universitas Pembangunan Panca Budi
ultas : SAINS & TEKNOLOGI
en Pembimbing I : Andysah putera Utama Siahaan, S.kom., M.kom., PhD.
en Pembimbing II : Ranti Eka Putri, S.kom., M.kom
a Mahasiswa : DIAN SYAHFITRI
san/Program Studi : Sistem Komputer
or Pokok Mahasiswa : 1514370266
ang Pendidikan : Strata satu (S1)
l Tugas Akhir/Skripsi : Penyandian Kriptografi Metode Caesar Cipher dengan menggunakan modulo 256

TANGGAL	PEMBAHASAN MATERI	PARAF	KETERANGAN
Agus 2018	ACC sempo		
Oktober 2019	Tambahkan penjelasan mengenai modulo 256 pada bab I dan dilandasan teori, perbaiki format tabel dan cantumkan sumber kutipan		
Oktober 2019	Tambahkan dilatar belakang mengenai modulo 256 yg kamu gunakan, pertimbangkan rumusan dan tujuan penelitian. kamu. Revisi bab I dan Bab II. Siap kan bab III		
Oktober 2019	Tambahkan landasan teori mengenai flow-Chart pd bab II, perbaiki diagram pd bab III, persiapkan draf bab IV dan bab V		
ovember 19	Persiapan draft lengkap. Rapikan Laporan.		
vember 19	ACC semhas		
sember 19	ACC Sidang		
uli 2020	ACC jilid		

Medan, 05 Agustus 2020
Diketahui/Disetujui oleh :
Dekan,

Handani, ST., MT

yang tidak perlu

YAYASAN PROF. DR. H. KADIRUN YAHYA
UNIVERSITAS PEMBANGUNAN PANCA BUDI
LABORATORIUM KOMPUTER
Jl. Jend. Gatot Subroto Km 4,5 Sei Sikambing Telp. 061-8455571
Medan - 20122

KARTU BEBAS PRAKTIKUM

Yang bertanda tangan dibawah ini Ka. Laboratorium Komputer dengan ini menerangkan bahwa :

Nama	:	DIAN SYAIFITRI
N.P.M.	:	1514370266
Tingkat/Semester	:	Akhir
Fakultas	:	SAINS & TEKNOLOGI
Jurusan/Prodi	:	Sistem Komputer

Benar dan telah menyelesaikan urusan administrasi di Laboratorium Komputer Universitas Pembangunan Panca Budi Medan.

Medan, 06 Desember 2019
Ka. Laboratorium

Fachrid Wadly, S. Kom

Telah Diperiksa oleh LPMU
dengan Plagiarisme...42...%
Medan 19 desember 2019
An Ka. LPMU
THARMIZI HAKIM
Cahyo Pramono, SE,MM

FM-BPAA-2012-041

Hal : Permohonan Meja Hijau

Medan, 17 Desember 2019
Kepada Yth : Bapak/Ibu Dekan
Fakultas SAINS & TEKNOLOGI
UNPAB Medan
Di -
Tempat

Telah di terima berkas persyaratan dapat di proses
Medan, 19/12/2019
Ka. BPAA
ar.
TEGUH WAHYONO, SE., MM.

Dengan hormat, saya yang bertanda tangan di bawah ini :

Nama	: DIAN SYAHFITRI
Tempat/Tgl. Lahir	: Medan / 18 Desember 1996
Nama Orang Tua	: RUSLIANDI
N. P. M	: 1514370266
Fakultas	: SAINS & TEKNOLOGI
Program Studi	: Sistem Komputer
No. HP	: 082304134001
Alamat	: Jl. Binjai Km 12

Datang bermohon kepada Bapak/Ibu untuk dapat diterima mengikuti Ujian Meja Hijau dengan judul **Penyandian Kriptografi Metode Caesar Cipher dengan menggunakan Module 256**, Selanjutnya saya menyatakan :

1. Melampirkan KKM yang telah disahkan oleh Ka. Prodi dan Dekan
2. Tidak akan menuntut ujian perbaikan nilai mata kuliah untuk perbaikan indek prestasi (IP), dan mohon diterbitkan ijazahnya setelah lulus ujian meja hijau.
3. Telah tercap keterangan bebas pustaka
4. Terlampir surat keterangan bebas laboratorium
5. Terlampir pas photo untuk ijazah ukuran 4x6 = 5 lembar dan 3x4 = 5 lembar Hitam Putih
6. Terlampir foto copy STTB SLTA dilegalisir 1 (satu) lembar dan bagi mahasiswa yang lanjutan D3 ke S1 lampirkan ijazah dan transkipnya sebanyak 1 lembar.
7. Terlampir pelunasan kwintasi pembayaran uang kuliah berjalan dan wisuda sebanyak 1 lembar
8. Skripsi sudah dijilid lux 2 examplar (1 untuk perpustakaan, 1 untuk mahasiswa) dan jilid kertas jeruk 5 examplar untuk penguji (bentuk dan warna penjilidan diserahkan berdasarkan ketentuan fakultas yang berlaku) dan lembar persetujuan sudah di tandatangani dosen pembimbing, prodi dan dekan
9. Soft Copy Skripsi disimpan di CD sebanyak 2 disc (Sesuai dengan Judul Skripsinya)
10. Terlampir surat keterangan BKKOL (pada saat pengambilan ijazah)
11. Setelah menyelesaikan persyaratan point-point diatas berkas di masukan kedalam MAP
12. Bersedia melunaskan biaya-biaya uang dibebankan untuk memproses pelaksanaan ujian dimaksud, dengan perincian sbb :

1. [102] Ujian Meja Hijau	: Rp.	100.000	20/12 2019
2. [170] Administrasi Wisuda	: Rp.	1,500,000	
3. [202] Bebas Pustaka	: Rp.	100,000	
4. [221] Bebas LAB	: Rp.	5,000	
Total Biaya	: Rp.	~~1.605.000~~ 1.705.000	

5-UK 00% (1 tahun) Rp 3.750.000

Rp 5.455.000

Periode Wisuda Ke : 64

Ukuran Toga : M

27/12
Diketahui/Disetujui oleh :

Ir. Hamdani, ST, MT
Dekan Fakultas SAINS & TEKNOLOGI

Hormat saya

DIAN SYAHFITRI
1514370266

Catatan :

- 1.Surat permohonan ini sah dan berlaku bila ;
  - a. Telah dicap Bukti Pelunasan dari UPT Perpustakaan UNPAB Medan.
  - b. Melampirkan Bukti Pembayaran Uang Kuliah aktif semester berjalan
- 2.Dibuat Rangkap 3 (tiga), untuk - Fakultas - untuk BPAA (asli) - Mhs.ybs.

TANDA BEBAS PUSTAKA
No. 1351 / PERP/ BP/2019
Dinyatakan tidak ada sangkut paut dengan Perpustakaan
Medan 19 DEC 2019
SALNIA S.IP

UKM CENTER
PANCA BUDI

an. Ketua UKMC
19/12-19
RORO RIAN AGUSTIN, S.SOS., MSP

Plagiarism Detector v. 1460 - Originality Report

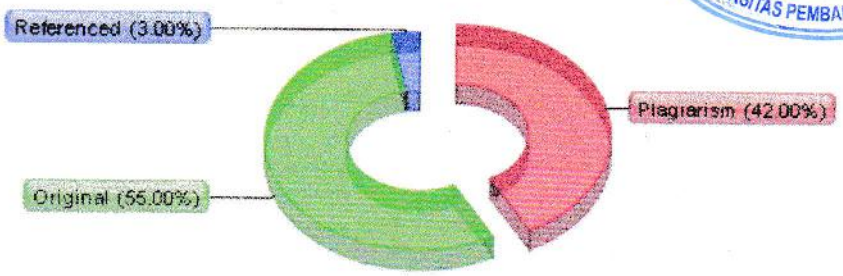
Analyzed document: 12/05/19 14:02:38

"DIAN SYAHFITRI_1514370266_SISTEM KOMPUTER.docx"

Check Type: Internet - via Google and Bing

Licensed to: **Universitas Pembangunan Panca Budi_License03**

Relation chart:



Distribution graph:

Comparison Preset: Rewrite. Detected language: Indonesian

Top sources of plagiarism:

% 12	wrds: 1226	http://etheses.uin-malang.ac.id/6275/1/10610041.pdf
% 12	wrds: 714	https://stianie.wordpress.com/2012/03/29/tugas-x-tkj-ted/
% 9	wrds: 545	http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2006-2007/Makalah/Makala...

ow other Sources:]

Processed resources details:

106 - Ok / 19 - Failed

ow other Sources:]

Important notes:

Wikipedia: Google Books: Ghostwriting services: Anti-cheating:

ABSTRAK

DIAN SYAHFITRI

PENYANDIAN KRIPTOGRAFI METODE CAESAR CIPHER DENGAN MENGGUNAKAN MODULO 256

2020

Skripsi ini bertujuan untuk mengetahui proses enkripsi dan deskripsi kriptografi *caesar cipher* dan membuat program menggunakan *Visual Basic Versi 2010*. Keamanan dan kerahasiaan dalam pesan teks menjadi suatu kebutuhan agar informasi yang dikirim dan diterima tidak disalahgunakan oleh pihak-pihak yang tidak bertanggung jawab. Keamanan dan kerahasiaan informasi juga dapat dijaga dengan memanfaatkan kriptografi. Kriptografi adalah seni untuk menjaga kerahasiaan dan keamanan suatu pesan yang akan dikirim maupun diterima. Kriptografi tidak hanya menyediakan alat untuk keamanan informasi, tetapi juga sekumpulan teknik yang berguna untuk keamanan dan kerahasiaan informasi. Banyak metode yang bisa digunakan untuk melakukan seni ataupun ilmu kriptografi. Pada skripsi ini, akan membahas mengenai penyandian kriptografi metode *caesar cipher* dengan menggunakan modulo 256. Didalam perhitungan ini, penulis menggunakan perhitungan dengan modulo 256, maka pergeseran yang mungkin dilakukan hanya dari 0 sampai 255. Modulo sendiri berarti sisa hasil bagi. Penulis menggunakan tabel *ASCII* (*American Standard Code for Information Interchange*) atau kode standar amerika untuk pertukaran informasi yang merupakan suatu standar internasional yang bersifat *Universal* yang didalamnya sudah berupa simbol, angka, huruf, dsb.

Kata kunci : *Caesar Cipher*, Kriptografi, Modulo 256, *ASCII*

DAFTAR ISI

Halaman

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kerahasiaan dan keamanan data di zaman ini tentunya sangat penting. Karena data dibagi dua jenis yaitu data bersifat rahasia dan yang bersifat tidak rahasia, artinya data yang bersifat rahasia akan sangat dijaga dan diperhatikan sedangkan data yang bersifat tidak rahasia biasanya tidak terlalu diperhatikan dan akan sangat mudah orang untuk menggandakannya. Agar menjaga keamanan data yang bersifat rahasia, diperlukan penyandian yang sedikit sulit untuk dideteksi orang yang tidak berhak membukanya.

Jika kita tidak memperhatikan keamanan data, maka sangat rentan terhadap terjadinya pencurian data dari beberapa pihak yang tidak bertanggung jawab. Untuk menjaga keamanan data atau pesan yang bersifat rahasia, terdapat beberapa cara dan teknik tertentu yang dapat digunakan. Salah satunya dengan kriptografi yang berfungsi untuk menyamarkan pesan menjadi bentuk pesan tersandi. Kriptografi metode *Caesar Cipher* menggunakan modulo 256 mempunyai kecepatan enkripsi yang baik. Ini disebabkan proses enkripsinya cukup sederhana. Dalam perhitungan ini, penulis menggunakan perhitungan dengan menggunakan modulo 256, maka pergeseran yang mungkin dilakukan hanya dari 0 sampai 255. Penulis menggunakan tabel *ASCII* (*American Standard Code of Information Interchange*) yang bersifat universal yang didalamnya sudah berupa simbol, angka, huruf, dan sebagainya. (Albert Ginting, 2015).

Berdasarkan dari latar belakang tersebut maka diangkatlah judul skripsi yang berjudul "**Penyandian Kriptografi Metode *Caesar Cipher* Dengan Menggunakan Modulo 256**".

1.2 Rumusan Masalah

Berdasarkan dari latar belakang tersebut, penulis membuat rumusan masalah sebagai berikut :

1. Bagaimana menerapkan metode *Caesar Cipher* menggunakan modulo 256 dalam penyandian kriptografi ?
2. Bagaimana pergeseran kunci pada proses enkripsi dan dekripsi metode caesar cipher ?

1.3 Batasan Masalah

Dalam penulisan tugas akhir ini, penulis akan membatasi masalah pada beberapa hal berikut ini :

1. Bahasa pemrograman pada penelitian penyandian kriptografi akan dibuat dengan menggunakan *visual basic versi 2010*.
2. Penyandian Kriptografi dalam penelitian ini menggunakan Modulo 256.
3. Data yang dienkripsi dan dideskripsi berupa *text*.

1.4 Tujuan Penelitian

Adapun tujuan yang akan dibahas dalam penulisan ini ialah :

1. Dengan diterapkannya atau digunakannya modulo 256 dalam metode caesar cipher ini memudahkan penulis untuk melindungi sebuah pesan teks menjadi pesan tersandi ataupun pesan tersandi menjadi teks asli.
2. Dengan mengetahui pergeseran kunci dalam metode caesar cipher memudahkan penulis untuk menerapkannya kedalam program.

1.5 Manfaat Penelitian

Peneliti berharap dalam melakukan penelitian ini dapat memberi manfaat antara lain:

1. Bagi peneliti

   Adapun manfaat penelitian ini bagi penulis sebagai berikut :

   a. Dapat menambah wawasan tentang penyandian
   b. Dapat memperkaya sumber pengetahuan tentang kriptografi *Caesar Cipher* meggunakan modulo 256.

2. Bagi pembaca

   Dapat digunakan sebagai bahan perbandingan bagi peneliti selanjutya yang ingin membahas lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Kriptografi

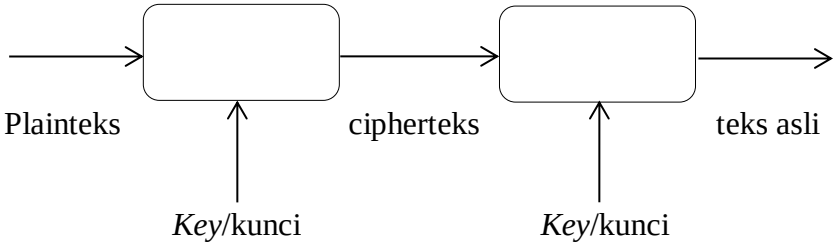
Keamanan informasi meliputi banyak aspek, antara lain pencegahan dari pengaksesan informasi oleh pihak-pihak yang tidak berhak, melindungi kerahasiaan informasi yang bersifat privat, pencegahan dari usaha untuk mengubah informasi, dan lain-lain. Berbicara tentang keamanan informasi tidak lepas dari kriptografi. Kriptografi merupakan salah satu teknik terpenting didalam keamanan informasi. Sejak dulu hingga sekarang manusia sudah mengembangkan banyak teknik kriptografi agar informasi yang dimilikinya tetap aman. Kriptografi sudah diaplikasikan dalam berbagai bidang, baik untuk administrasi pemerintah, perdagangan, telekomunikasi, *e-commerce*, hingga dunia hiburan.

Istilah kriptografi (*cryptography*) berasal dari bahasa yunani yaitu *"kryptos"* yang berarti tersembunyi dan *"gradient"* yang berarti menulis atau jika diterjemahkan secara tersembunyi atau rahasia. Seringkali juga diartikan sebagai praktik penyembunyian data agar data atau pesan tersebut hanya bisa dibaca oleh orang yang berhak (Soetam, 2010). Dengan kriptografi, sebuah informasi dapat diacak atau disandikan menjadi informasi yang sulit atau bahkan tidak dipahami melalui sebuah proses yang dinamakan dengan enkripsi (Murdani, 2017).

Jika sebuah pesan yang telah dikodekan berusaha untuk dipecah, maka proses tersebut disebut sebagai kriptanalisis (*cryptanalysis*), dan orang yang melakukannya disebut sebagai seorang kriptanalis (*cryptanalyst*).

Istilah kriptografi sendiri berbeda dengan istilah kriptologi (*cryptology*), karena kriptologi merupakan cabang ilmu matematika yang menggabungkan antara kriptografi dengan kriptanalisis. Berbeda lagi dengan istilah kriptosistem (*cryptosystem*) yang merupakan algoritma dengan segala kemungkinan teks asal (*plaintext*), dan teks hasil (*ciphertext*) dan semua kemungkinan kunci yang ada.

Secara umum, sebuah proses didalam kriptografi digambarkan dalam skema sebagai berikut :



Gambar 2.1 Skema umum Kriptografi
Sumber : Muhammad Nurtanzis Sutoyo, 2016

Dalam kriptografi terdapat aspek-aspek keamanan data yaitu :

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

5. *Access Control*, aspek ini berhubungan dengan mekanisme pengaturan akses ke informasi, untuk mengatur siapa yang boleh melakukan apa.

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

1. Pesan (*massage*), adalah data atau informasi yang dapat dibaca atau dimengerti maknanya.
2. Pengirim (*sender*), adalah entitas yang melakukan pengiriman pesan kepada entitas lain.
3. Kunci (*key*), adalah aturan atau fungsi yang dilakukan untuk melakukan prosen enkripsi dan dekripsi pada plainteks dan cipherteks.
4. Enkripsi adalah mekanisme yang dilakukan untuk merubah plainteks menjadi cipherteks.
5. Dekripsi adalah mekanisme yang dilakukan untuk merubah cipherteks menjadi plainteks.
6. Penerima (*recipient*), adalah entitas yang penerima berhak menerima pesan dari pengirim.

2.1.1 Kriptografi Kunci Publik

Aplikasi kriptografi kunci-publik dibagi menjadi 3 kategori :

1. Kriptografi kunci-publik dapat digunakan untuk menjaga kerahasiaan data (*provide confidentiality/secrecy*) melalui mekanisme enkripsi dan dekripsi. Contoh untuk algoritma aplikasi ini adalah *RSA*, *Knapsack*, *Rabin*, *ElGamal*, *Elliptic Curve Cryptography (ECC)*.

2. Digital *sigatures* kriptografi kunci-publik dapat digunakan untuk membuktikan otentikasi pesan maupun otentikasi pengirim (*provide authentication*). Contoh algoritma *RSA, DSA, ElGamal, GOST*.
3. Pertukaran kunci (*key exchange*) algoritma kriptografi kunci-publik dapat digunakan untuk pengiriman kunci simetri (*session keys*). Contoh algoritma *RSA, Diffie-Hellman.*

Beberapa algoritma kriptografi kunci-publik cocok digunakan untuk ketiga macam kategori aplikasi (misalnya *RSA*), beberapa algoritma hanya ditunjukan untuk aplikasi spesifik (misalnya *DSA* untuk digital *signature*). Kunci publik (*public key*) yang merupakan nama lain dari algoritma asimetris. Enkripsi dan dekripsi pada algoritma asimetris dibagi menjadi dua yaitu kunci umum (*public key*). Pada kunci umum, kunci tersebut dapat diketahui oleh semua orang (*public*). Sedangkan pada kunci pribadi hanya dapat diketahui oleh orang yang bersangkutan. Pengetahuan kunci umum memungkinkan seseorang untuk dapat mengenkripsi suatu pesan tetapi tidak dapat mendekripsikan pesan tersebut. Hanya orang yang memiliki kunci pribadi yang dapat mendekripsikan pesan yang telah dienkripsi. Sehingga kedua kunci tersebut (kunci umum dan kunci pribadi) harus saling berhubungan satu dengan yang lain.

2.1.2 Kriptografi Kunci Simetris

Algoritma simetris disebut juga sebagai algorima konvensional. Algoritma simetris menggunakan suatu kunci yang sama untuk proses enkripsi dan dekripsi. Penggunaan kunci yang sama menjadikan kekuatan algoritma simetris menjadi sangat bergantung pada satu kunci yang digunakan, selain itu proses dekripsi pada

algoritma simetris juga menjadi kebalikan dari proses enkripsi. Apabila pengiriman kunci dapat dilakukan secara aman, akan menjadi kesempatan *cryptanalyst* untuk mendapat cipherteks dan plainteks semakin kecil.

2.1.3 Perbandingan Kriptografi Kunci-Simetri dengan Kunci-Publik

Baik kriptografi kunci-simetri maupun kriptografi kunci-publik, keduanya mempunyai kelebihan dan kelemahan.

Kelebihan kriptografi kunci-simetri :

1. Algoritma kriptografi simetri dirancang sedemikian rupa sehingga proses enkripsi/dekripsi membutuhkan waktu yang relatif singkat.
2. Ukuran kunci relatif pendek. *DES* memiliki panjang kunci 64 bit, *AES* memiliki kunci sepanjang 128 hingga 256 *bit*.
3. Algoritma kriptografi simetri dapat dirancang sedemikian kompleks untuk menghasilkan *cipher* yang lebih kuat.
4. Otentikasi pengirim pesan langsung diketahui dari cipherteks yang diterima, karena kunci hanya diketahui oleh pengirim dan penerima pesan saja.

Kelemahan kriptografi kunci-simetri :

1. Kunci simetri harus dikirim melalui saluran yang sangat aman atau melalui cara yang tidak dapat disadap. Kedua entitas yang berkomunikasi harus menjaga kerahasiaan kunci ini.
2. Kunci harus sering diubah supaya tidak mudah dicuri, pengubahan kunci mungkin dilakukan pada setiap kali sesi komunikasi.

Kelebihan kriptografi kunci-publik :

1. Hanya kunci privat yang perlu dijaga kerahasiaannya oleh setiap entitas yang berkomunikasi. Tidak ada keperluan mengirim kunci-kunci privat kepada penerima pesan sebagaimana pada sistem simetri.
2. Pasangan kunci *public*/kunci privat tidak perlu diubah, bahkan dalam periode waktu yang panjang sekalipun.
3. Dapat digunakan untuk mengamankan pengiriman kunci rahasia yang digunakan untuk komunikasi pesan dengan algoritma kriptografi kunci-simetri. Kunci rahasia dienkripsi dengan kunci publik penerima pesan, penerima pesan mendekripsi kunci rahasia dengan kunci privatnya. Selanjutnya kunci rahasia digunakan untuk komunikasi menggunakan algoritma kriptografi simetri.
4. Beberapa algoritma kunci-publik dapat digunakan untuk memberi tanda tangan digital pada pesan.

Kelemahan kriptografi kunci-publik :

1. Enkripsi dan dekripsi data umumnya lebih lambat dari pada sistem simetri, karena enkripsi dan dekripsi menggunakan bilangan bulat yang besar dan melibatkan operasi perpangkatan yang banyak.
2. Ukuran cipherteks lebih besar dari pada plainteksnya (bias dua sampai empat kali ukuran plainteks).
3. Ukuran kunci relatif lebih panjang dari pada ukuran kunci simetri.
4. Kunci publik diketahui luas dan dapat digunakan setiap orang, maka cipherteks tidak memberikan informasi mengenai otentikasi pengirim.

5. Tidak ada algoritma kunci-publik yang terbukti aman. Kebanyakan algoritma mendasarkan keamanannya pada asumsi sulitnya memecahkan persoalan-persoalan aritmetik (pemfaktoran, logaritmik, dan sebagainya) yang menjadi dasar pebangkitan kunci. Kriptografi kunci-publik juga tidak aman dari serangan *man-in-the-middle attack*. Orang di "tengah" mengintersepsi komunikasi lalu berpura-pura sebagai salah satu pihak yang berkomunikasi untuk mengetahui informasi rahasia.

2.1.4 Tujuan Kriptografi

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

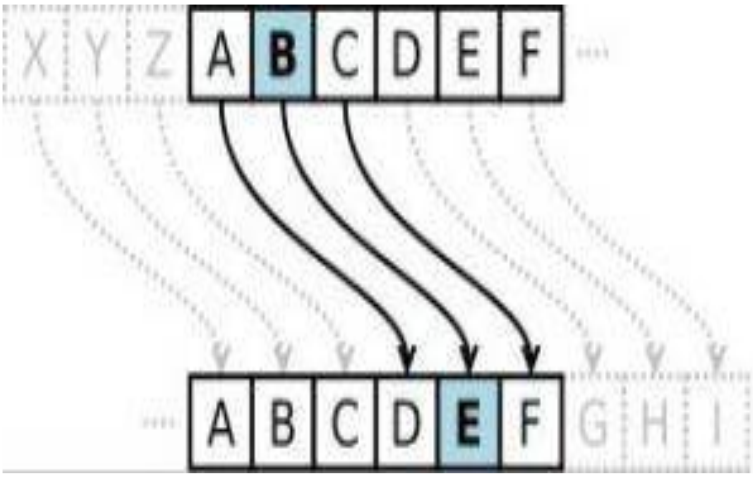
1. Kerahasiaan (*Confidentiality*), informasi dirahasiakan dari semua pihak yang tidak berwenang.
2. Keutuhan Data (*Integrity*)
3. Pesan tidak berubah dalam proses pengiriman hingga pesan diterima oleh si penerima.
4. Autentikasi (*Message Autentication*), kepastian terhadap identitas yang terlibat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

2.2 Algoritma *Caesar Cipher*

Algoritma ialah 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.

Algoritma *Caesar Cipher* merupakan salah satu algoritma *cipher* tertua dan paling diketahui dalam perkembangan ilmu kriptografi. *Caesar Cipher* merupakan salah satu jenis *cipher* substansi yang membentuk *cipher* dengan cara melakukan penukaran karakter pada plainteks menjadi tepat satu karakter pada cipherteks (Atmaja Basuki, 2016).

Konsep *Caesar Cipher* ialah menentukan besarnya pergeseran karakter yang digunakan dalam membentuk cipherteks ke plainteks. Menukarkan karakter pada plainteks menjadi cipherteks dengan berdasarkan pada pergeseran yang telah ditentukan sebelumnya. Misalnya diketahui bahwa pergeseran = 3, maka huruf A akan digantikan oleh huruf D, huruf B menjadi huruf E, huruf C menjadi huruf F, dan seterusnya.



Gambar 2.2 Sandi *Caesar* Dengan Geseran Tiga
Sumber : Atmaja Basuki, 2016

Teknik penyandian ini termasuk sandi tersubtitusi pada setiap huruf pada plainteks digantikan oleh huruf lain yang dimiliki selisih posisi tertentu dalam alphabet.

Cara Kerja *Caesar Cipher*

Cara kerja sandi *Caesar Cipher* diilustrasikan dengan membariskan dua set alphabet. Dimana sandi disusun dengan cara menggeser alphabet biasa ke kanan atau ke kiri dengan angka tertetu (sesuai kunci). Misalnya sandi *Caesar Cipher* dengan kunci 3, yaitu :

Alphabet biasa:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Alphabet sandi :

D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Teks asli :

D I A N

Teks sandi :

G L D Q

Sedangkan untuk memecahkan sandi tersebut dengan cara menggunakan kunci sebaliknya, yaitu p – 3. Proses enkripsi (penyandian) dapat dilakukan secara matematis dengan menggunakan operasi modulo. Dimana dengan mengubah huruf-huruf menjadi angka, A = 0, B = 1, …, Z = 25. Sandi (E_n) dari huruf dengan bergeser n. secara matematis dapat dituliskan dengan rumus :

$$\mathbf{E_n(x) = (x + n) \bmod 256}$$

Sedangkan untuk proses pemecahan sandi (*dekripsi*) dapat dituliskan dengan rumus :

$$\mathbf{D_n(x) = (x - n) \bmod 256}$$

2.3 Modulo 256

Dalam perumusan Algoritma Caesar Cipher ini, sangat dibutuhkan pemahaman tentang modulo, Modulo sendiri berarti sisa hasil bagi. Misalkan a adalah bilangan bulat dan m adalah bilangan bulat dimana a dan m lebih besar dari 0. Maka operasi a mod m (dibaca "a modulo m") memberikan sisa jika a dibagi dengan m. bilangan m disebut modulus atau modulo. Dan hasil modulo m terletak di dalam himpunan (0, 1, 2, ..., m-1).

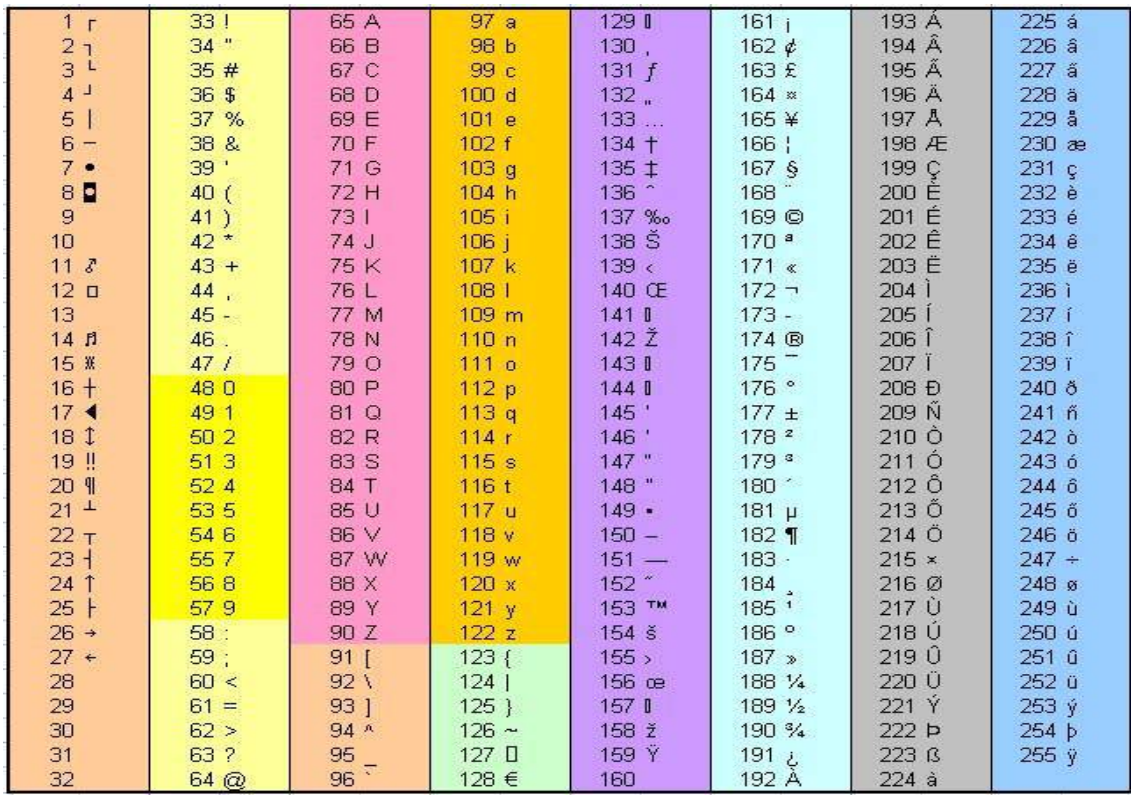
Contoh :

Diambil a = 20 dan m = 6. Karena 20dibagi 6 adalah 3 bersisa 2, maka diperoleh a mod m = 20 mod 6 = 2.

Didalam perhitungan ini, penulis menggunakan perhitungan dengan modulo 256, maka pergeseran yang mungkin dilakukan hanya dari 0 sampai 255. Penulis menggunakan tabel *ASCII* (*American Standard Code for Information Interchange*) atau kode standar Amerika untuk pertukaran informasi merupakan suatu standar internasional yang bersifat *universal* yang didalamnya sudah berupa simbol, angka, huruf dsb. *ASCII* selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks (Albert Ginting, 2015).

Dan dengan digunakannya perumusan ini, memudahkan penulis untuk memecahkan atau menghitung pesan teks menjadi pesan tersandi, pesan tersandi menjadi teks asli.

Berikut ini adalah tampilan gambar kode *ASCII* :



1 ┌	33 !	65 A	97 a	129 ▯	161 ¡	193 Á	225 á
2 ┐	34 "	66 B	98 b	130 ,	162 ¢	194 Â	226 â
3 └	35 #	67 C	99 c	131 ƒ	163 £	195 Ã	227 ã
4 ┘	36 $	68 D	100 d	132 „	164 ¤	196 Ä	228 ä
5 |	37 %	69 E	101 e	133 …	165 ¥	197 Å	229 å
6 ─	38 &	70 F	102 f	134 †	166 ¦	198 Æ	230 æ
7 •	39 '	71 G	103 g	135 ‡	167 §	199 Ç	231 ç
8 ◘	40 (	72 H	104 h	136 ˆ	168 ¨	200 È	232 è
9	41)	73 I	105 i	137 ‰	169 ©	201 É	233 é
10	42 *	74 J	106 j	138 Š	170 ª	202 Ê	234 ê
11 ♂	43 +	75 K	107 k	139 ‹	171 «	203 Ë	235 ë
12 ▫	44 ,	76 L	108 l	140 Œ	172 ¬	204 Ì	236 ì
13	45 -	77 M	109 m	141 ▯	173 -	205 Í	237 í
14 ♫	46 .	78 N	110 n	142 Ž	174 ®	206 Î	238 î
15 ☼	47 /	79 O	111 o	143 ▯	175 ¯	207 Ï	239 ï
16 ►	48 0	80 P	112 p	144 ▯	176 °	208 Ð	240 ð
17 ◄	49 1	81 Q	113 q	145 '	177 ±	209 Ñ	241 ñ
18 ↕	50 2	82 R	114 r	146 '	178 ²	210 Ò	242 ò
19 ‼	51 3	83 S	115 s	147 "	179 ³	211 Ó	243 ó
20 ¶	52 4	84 T	116 t	148 "	180 ´	212 Ô	244 ô
21 §	53 5	85 U	117 u	149 •	181 µ	213 Õ	245 õ
22 ▬	54 6	86 V	118 v	150 –	182 ¶	214 Ö	246 ö
23 ↨	55 7	87 W	119 w	151 —	183 ·	215 ×	247 ÷
24 ↑	56 8	88 X	120 x	152 ˜	184 ¸	216 Ø	248 ø
25 ↓	57 9	89 Y	121 y	153 ™	185 ¹	217 Ù	249 ù
26 →	58 :	90 Z	122 z	154 š	186 º	218 Ú	250 ú
27 ←	59 ;	91 [	123 {	155 ›	187 »	219 Û	251 û
28	60 <	92 \	124 |	156 œ	188 ¼	220 Ü	252 ü
29	61 =	93]	125 }	157 ▯	189 ½	221 Ý	253 ý
30	62 >	94 ^	126 ~	158 ž	190 ¾	222 Þ	254 þ
31	63 ?	95 _	127 ▯	159 Ÿ	191 ¿	223 ß	255 ÿ
32	64 @	96 `	128 €	160	192 À	224 à	

Gambar 2.3 Tabel *ASCII*
Sumber : Endah Handayani, 2017

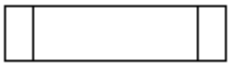
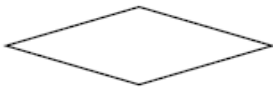
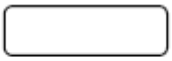
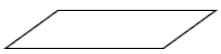
2.4 *Flowchart*

Flowchat ialah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah. Dengan menggunakan *flowchart*, memudahkan pengguna untuk melakukan pengecekan pada bagian-bagian yang terlupakan dalam analisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek.

Flowchart membantu kita memahami urutan-urutan logika yang rumit dan panjang. *Flowchart* membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) akan lebih mudah (Santoso, 2017).

Berikut beberapa simbol yang digunakan dalam menggambarkan suatu *flowchart* :

Tabel 2.1 Simbol-simbol Flowchart

SIMBOL	NAMA	FUNGSI
	*Predefine process*	Permulaan *sub* program.
	*Decision*	Perbandingan, pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya.
	*Terminator*	Penghubung bagian-bagian *flowchart* yang berada pada satu halaman.
	*Connector*	Penghubung bagian-bagian *flowchart* yang berada pada halaman berbeda.
	*Terminal point*	Awal/akhir *flowchart*.
	Arrow	Arah aliran program.
	*Preparation*	Proses inisialisasi/pemberian harga awal.
	*Rectangle*	Proses penghitung/proses pengolahan data.
	*Trapezium*	Proses *input*/*output* data.
	*Document*	Digunakan untuk mewakili *output*.
	*Manual input*	Simbol untuk memasukkan data secara manual melalui *keyboard*

Tabel 2.1 Simbol-simbol Flowchart (lanjutan)

SIMBOL	NAMA	FUNGSI
	*Manual operation*	Simbol yang menunjukkan pengolahan yang tidak dilakukan komputer
	*Display*	Simbol yang menyatakan peralatan *output* yang digunakan seperti layar, *printer*, *plotter*, dan sebagainya.
	*Magnetik disk*	Simbol yang digunakan untuk penyimpanan data ke *database*.
	*Storage data*	Simbol yang menyatakan input yang berasal dari *disk* atau disimpan ke *disk*.
	*Database*	Menyimpan ke *database*.

Sumber : Santoso, 2017

2.5 *UML* (*Unified Modelling Language*)

Unified Modeling Language (*UML*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan dari sebuah sistem pengembangan *software* berbasis *object oriented* (Mamed, 2015).

UML juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*. Tetapi karena *UML* juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa

berorientasi objek seperti *C++*, *java*, *C#*, atau *VB NET*. Walaupun demikian, *UML* tetap dapat digunakan untuk modeling aplikasi *procedural* dalam *VB* atau *C*.

Seperti bahasa lainnya, *UML* mendefinisikan notasi dan *sytax/semantic*. Notasi *UML* merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan *UML syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan.

2.5.1 Konsep Dasar *UML*

Dari berbagai penjelasan yang terdapat di dokumen dan buku-buku *UML*, Sebenarnya konsep dasar *UML* bisa kita rangkum dalam table 2.2 :

Tabel 2.2 Konsep Dasar UML

Major Area	***View***	***Diagrams***	***Main Concepts***
Structural	*Static view*	*Class diagram*	*Class, association, generalization, dependency, realization.*
	Use case view	*Use case diagram*	*Use case, actor, association, extend, include, use case generalization.*
	Implementation view	*Component diagram*	*Component, interface, dependency, realization*
	Deployment view	*Deployment diagram*	*Node, Component, dependency, location*
Dynamic	*State machine view*	*Statechart diagram*	*State, event, transition, action*
	Activity view	*Activity diagram*	*State, activity, completion, transition, fork, join*
	Interaction view	*Sequence diagram*	*Interaction, object, message activation*
		Collaboration diagram	*Collaboration, interaction, collaboration role, message*

Tabel 2.2 Konsep Dasar *UML* (lanjutan)

Major Area	***View***	***Diagrams***	***Main Concepts***
Model management	*Model management view*	*Class diagram*	*Package, subsystem, model*
Extensibility	*All*	*All*	*Constraint, stereotype, tagged values.*

Sumber : Dharwiyanti, 2003

Abstraksi konsep dasar *UML* yang terjadi dari *structural classification*, *dynamic behaviour*, dan model managemen, bisa kita pahami dengan mudah apabila kita melihat tabel diatas dari diagram. *Main concepts* bisa kita pandang sebagai *term* yang akan muncul pada saat kita membuat digram. Dan *view* adalah kategori dari diagram tersebut.

Tercantum juga pada gambar diatas, *UML* mendefinisikan diagram-diagram sebagai berikut :

1. *Use case diagram*
2. *Activity diagram*
3. *Sequence diagram*
4. *Class diagram*
5. *Statechart diagram*
6. *Collaboration diagram*
7. *Component diagram*
8. *Deployment diagram*

2.5.2 *Use Case Diagram*

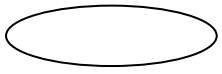
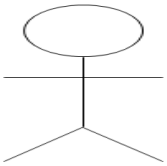
Use case diagram adalah sesuatu atau proses merepresentasikan hal-hal yang dapat dilakukan oleh aktor dalam menyelesaikan sebuah pekerjaan (mamed, 2015). Misalnya *login* ke sistem, membuat sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendekripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

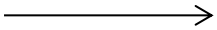
Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan *client*, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Berikut ini adalah simbol-simbol yang ada pada *use case* diagram :

Tabel 2.3 komponen *Use Case* Diagram

SIMBOL	KETERANGAN
	*Use case* menggambarkan fungsionalitas yang di sediakan sistem sabagai unit-unit yang bertukar pesan antar unit dengan aktif, dinyatakan dengan menggunakan kata kerja.
	Aktor (*actor*) adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bias muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki kontrol terhadap *use case*.

Tabel 2.3 komponen *Use Case* Diagram (lanjutan)

SIMBOL	KETERANGAN
	Asosiasi antar aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif.
	Include, merupakan didalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	Extend, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

Sumber : Hendini, 2016

2.5.3 *Activity* Diagram

Activity diagram menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi, bagaimana aksi-aksi tersebut dimulai, keputusan yang mungkin terjadi hingga berakhirnya aksi. *Activity* diagram juga dapat menggambarkan proses lebih dari satu aksi dalam waktu bersamaan.

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

Activity diagram merupakan *stage* diagram khusus, dimana sebagian besar *stage* adalah *action* dan sebagian besar transisi di *trigger* oleh selesainya *stage* sebelum

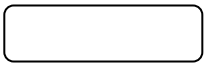
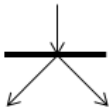
(*unternal processing*). Oleh karena itu *Activity* diagram tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

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

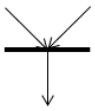
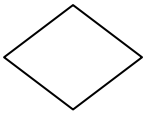
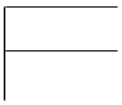
Sama dengan *state*, standar *UML* menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses- proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik,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

Berikut adalah simbol-simbol yang digunakan dalam *activity* diagram :

Tabel 2.4 simbol *Activity* Diagram

SIMBOL	KETERANGAN
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	*End point*, akhir aktivitas
	*Aktivities*, menggambarkan suatu proses atau kegiatan bisnis.
	*Fork*/percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara parael atau untuk menggabungkan dua kegiatan paralel menjadi satu.

Tabel 2.4 simbol *Activity* Diagram (lanjutan)

SIMBOL	KETERANGAN
	*Join* (penggabungan) atau *rake*, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision point*, menggambarka pilihan untuk pengambilan keputusan, *true* atau *false*.
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

Sumber : Hendini, 2016

2.5.4 *Sequence* Diagram

Sequence diagram adalah *tool* yang sangat popular dalam pengembangan sistem informasi secara object *oriented* untuk menampilkan interaksi antar objek (Nofriyadi Nurdam, 2014).

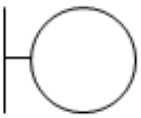
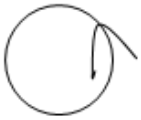
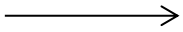
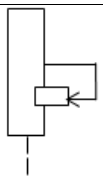
Sequence diagram biasa digunakn untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* yang menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan.

Masing-masing objek termasuk *actor*, memilliki *lifeline vertikal*. *Message* digambarkan sebagai garis berpanah dari suatu objek lainnya pada *fase* design berikutnya, *message*

akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

Berikut adalah komponen-komponen pada *sequence* diagram :

Tabel 2.5 komponen *Sequence* Diagram

SIMBOL	KETERANGAN
	*Entity class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundary class*, berisi kumpulan kelas yang menjadi *interfaces* atau interaksi antara satu atau lebih *actor* dengan sistem, seperti tampilan *form entry* dan *form* cetak.
	*Control class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas,contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagi objek.
	*Message*, simbol mengirim pesan antar *class*.
	*Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation*, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*

Sumber : Hendini, 2016

2.6 Visual Basic

Bahasa pemrograman *visual basic* merupakan salah satu bahasa yang sangat popular hingga saat ini dan merupakan salah satu solusi untuk menciptakan aplikasi pada sistem operasi *windows*, baik *windows* 7, *windows server* 2008, dan *windows mobile* 6.1. hal ini dikarenakan kemudahan yang diberikan aplikasi (Andi, 2012).

2.6.1 Kemampuan Visual Basic

Adapun kemampuan atau manfaat dari *visual basic* yaitu :

1. Untuk membuat program aplikasi berbasis website.
2. Untuk membuat *ActiveX*, aplikasi internet dan sebagainya.
3. Menguji program (*debugging*) dan menghasilkan program akhir berakhiran *EXE* yang bersifat *executable* atau dapat langsung dijalankan.

2.6.2 Jendela-jendela pada *Visual Basic*

1. *Menubar*

Sebelum menulis kode, sebaiknya kita mengenal dahulu *IDE* atau lingkungan kerja yang digunakan, sehingga kita tidak bingung saat bekerja dengan *IDE VISUAL Studio* 2010. Secara umum aplikasi mempunyai dua buah jenis menu, yaitu *menubar* dan *toolbar* (jalan pintas menu). Kecuali pada *Microsoft Office* 2007 dan beberapa aplikasi *windows* terbaru.

2. Menu

Menu adalah sebuah kontrol yang digunakan untuk melakukan proses tertentu. Kebanyakan menu digunakan untuk membuka jendela lain. Menu ini juga dapat terdiri

dari label (*teks*) yang tersusun, maupun sebuah *button* (tombol), kemudian di belakang tombol atau label tersebut terdapat kode tertentu untuk melaksanakan tugas tertentu.

3. Menu konteks (*PopUp*)

Menu konteks (*context menu*) adalah sebuah menu yang akan tampil ketika klik kanan terjadi pada suatu komponen/kontrol. Menu ini dahulu disebut menu *PopUp*, yaitu menu yang muncul seperti deretan label yang mempunyai fungsi tertentu.

4. *Toolbar*

Toolbar adalah sebuah tombol jalan pintas yang terdapat pada *menubar*. Terdapat bermacam-macam jenis *toolbar*, namun yang paling sering digunakan adalah *toolbar* standar. Kita dapat membuka dan menutup *toolbar* melalui menu *view*-jenis *toolbar*.

5. Menu *Window*

Menu *window* menjadi standar pada sebuah aplikasi yang mengijinkan membuka jendela pada satu waktu seperti *word* dan *exel*. Kita dapat berganti jendela yang aktif melalui menu ini.

6. *Toolbox*

Jendela ini berisi kontrol dan komponen yang dapat digunakan sewaktu-waktu dengan menambahkannya ke dalam aplikasi. Terdapat 12 grup komponen sesuai dengan kegunaan masing-masing.

7. *Design*

Jendela ini menampilkan form yang kita buat dan disini pula kita mendesain tampilan dari aplikasi kita. Jendela ini merupakan jendela utama yang paling besar terletak ditengah *IDE*.

8. *Solution Explorer*

Jendela ini menampilkan hierarki dari solution kita. Sebuah *solution* dapat berisi banyak proyek, dimana proyek dapat mengandung banyak *form*, kelas, modul, dan komponen lain untuk menyelesaikan masalah.

9. *Properties*

Jendela ini menampilkan *property* dan objek yang terpilih pada jendela *design*. Dengan jendela *properties* ini kita dapat mengubah property objek terpilih. Selain itu kita juga dapat mengaturnya melalui kode.

10. *Data Sources*

Jendela ini digunakan untuk memanipulasi data *source* yang berhubungan dengan *database*.

11. Jendela *code*

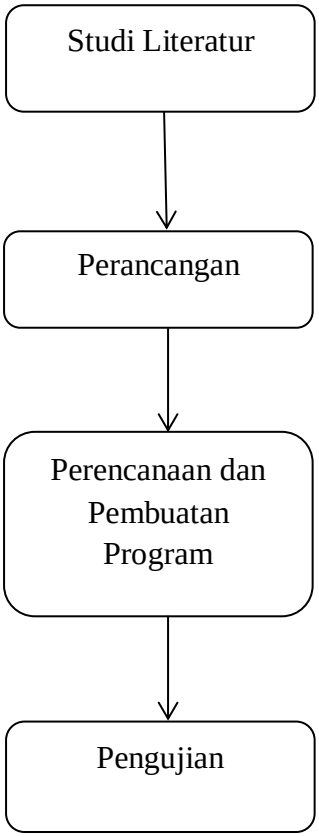
Jendela *code* adalah salah satu jendela yang penting di dalam *microsoft visual basic*. Jendela ini berisi kode-kode program yang merupakan intruksi-intruksi untuk aplikasi *visual basic* dapat ditambahkan dengan kode-kode program untuk melakukan tugas-tugas tertentu seperti menutup aplikasi, membatalkan perintah, dan sebagainya.

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Tahapan yang dilakukan dalam pelaksanaan penelitian ini ialah :



Gambar 3.1 Tahapan Penelitian

Adapun penjelasan tentang tahapan penelitian sebagai berikut :

1. Studi Literatur

   Pada tahap ini, dilakukan studi literatur yang bertujuan mengumpulkan, mempelajari dan menyeleksi bahan-bahan dan sumber-sumber yang tertulis, dengan cara membaca, mempelajari dan mencatat hal yang penting

sehubungan dengan penelitian tersebut. Untuk mendapatkan data yang diperlukan untuk melengkapi tugas akhir ini ialah :

a. Mempelajari bermacam sumber literatur, yaitu dari beberapa sumber buku, *e-book*, dan *internet* khususnya yang berhubungan dengan *Caesar cipher*.
b. Konsultasi dengan dosen pembimbing dan pihak lain yang bisa membantu.
c. Pencarian data melalui referensi skripsi alumni yang terdapat di perpustakaan UNPAB.
d. Mencoba menginputkan berbagai jenis teks dengan beragam agar diketahui seberapa efektifkah program yang telah dibuat.

2. Perancangan

   Pada tahap ini, dilakukan perancangan guna untuk penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Perancangan sistem dapat dirancang dalam bentuk bagan alur sistem (system flowchart), yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukkan urutan-urutan proses dari sistem.

3. Perencanaan dan Pembuatan Program

   Pada tahap ini, pembuatan program akan dilakukan dengan algoritma yang telah dirancang dan dikembangkan ke dalam bahasa pemrograman dengan menggunakan bahasa pemrograman caesar cipher.

4. Uji coba program

   Pengujian program dilakukan setelah program aplikasi selesai dibuat, apakah program ini sesuai dengan apa yang diinginkan dengan cara menguji dan menganalisis program.

3.2 Analisis Sistem yang Diusulkan

Pada analisis sistem ini, penulis menjelaskan tentang perancangan yang bertujuan untuk mengimplementasikan penyandian aloritma *Caesar cipher* pada keamanan. Sistem yang dibangun ialah sebuah pesan teks dengan enkripsi/deskripsi teks dan membuka pesan dengan menggunakan *key* (kunci) kepada penerima agar pesan hanya dapat dibuka oleh orang yang bersangkutan. Dengan diterapkannya sisitem ini, implementasi algoritma caesar cipher dapat membantu dalam keamanan.

Input yang akan diproses dalam aplikasi yang akan dirancang berupa karakter alfanumerik, yang akan diproses dengan algoritma *Caesar cipher*. Sehingga dari segi pengirim dan penerima pesan dapat mengenkripsi maupun mendekripsi sandi pesan teks yang dikirim dan diterima.

Metode penyandian *Caesar cipher* ialah termasuk dalam penyandian klasik, *Caesar cipher* dikenal dengan beberapa nama seperti *shift cipher*, *caesar's code* atau *Caesar shift*. *Caesar cipher* merupakan teknik enkripsi yang paling sederhana dan banyak digunakan. *Cipher* ini berjenis *cipher* substitusi, dimana setiap huruf pada teksnya diganti dengan huruf lain berupa karakter alfanumerik.

Implementasi penyandian *caesar cipher* dapat dijelaskan dengan contoh penyandian sederhana. Transformasi *caesar cipher* dapat dipresentasikan degan menyelaraskan teks

normal dengan teks *cipher*, ke kiri atau ke kanan sebanyak jumlah pergeseran yang diinginkan. Sebagai contoh menerapkan pergeseran pada alphabet dengan jumlah pergeseran sebanyak 3 (tiga) karakter dari susunan alphabet.

Teks normal	: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Teks cipher	: DEFGHIJKLMNOPQRSTUVWXYZABC

Untuk membaca pesan yang dienkripsi, penerima dapat menyelaraskan huruf teks *cipher* yang diterima dengan teks normal yang tepat berada di atasnya.

Contoh dibawah ini adalah penerapan enkripsi dan dekripsi *Caesar cipher*, dapat dilihat pergeseran huruf berdasarkan alphabet.

Teks cipher	: PHGDQ
Teks normal	: MEDAN

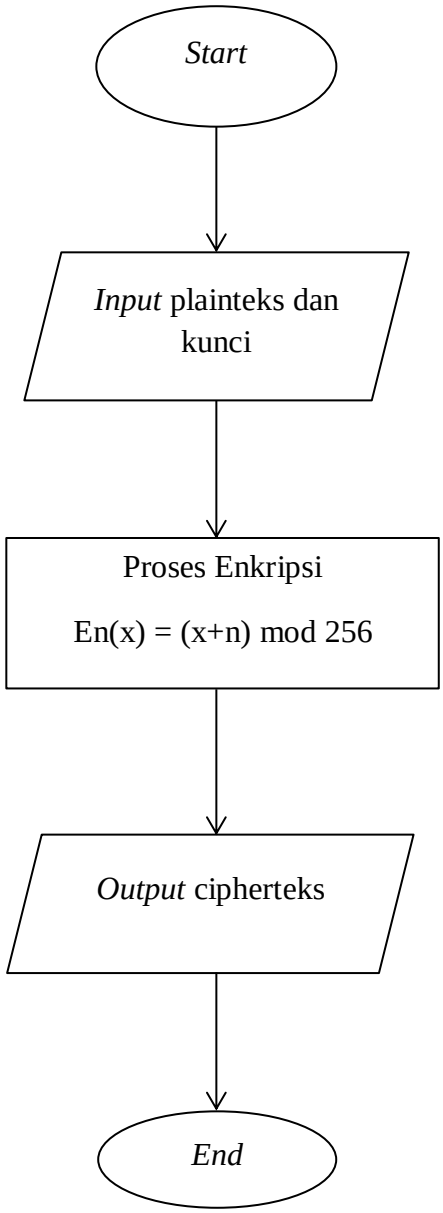
3.3 Rancangan Penelitian

Pada penulisan ini, mengenkripsi pesan teks menggunakan metode *Caesar cipher*. *Caesar cipher* merupakan salah satu jenis *cipher* substansi yang membentuk *cipher* dengan cara melakukan penukaran karakter pada plainteks menjadi tepat satu karakter pada *cipherteks*. Pada program ini menggunakan *mod* 256 dan juga menggunakan kunci (*key*) agar pesan teks lebih aman dan terjaga kerahasiaannya dan keasliannya sehingga sulit terdeteksi oleh pihak-pihak yang tidak berwenang. *Caesar cipher* dapat diilustrasikan dengan membariskan dua *set* alphabet, Dimana sandi disusun dengan menggeser alphabet ke kanan atau ke kiri dengan angka tertentu.

3.3.1 *Flowchart Caesar cipher*

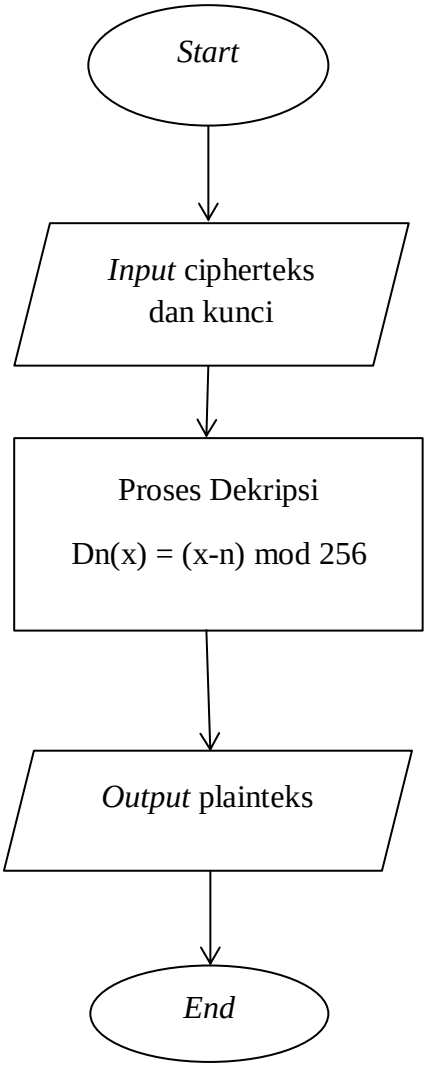
Flowchart merupakan aliran data untuk proses penyandian *Caesar cipher*. Dapat dilihat pada gambar berikut :

a. Proses enkripsi



Gambar 3.2 Proses Enkripsi *Caesar Cipher*

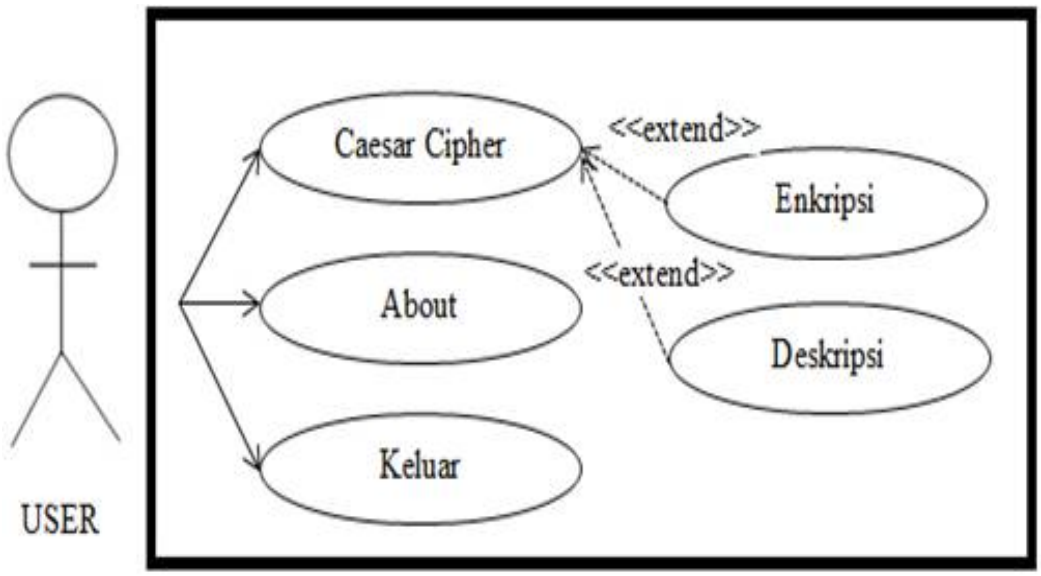
b. Proses Deskripsi



Gambar 3.3 Proses Dekripsi *Caesar Cipher*

3.4 *Use Case* Diagram Algoritma *Caesar Cipher*

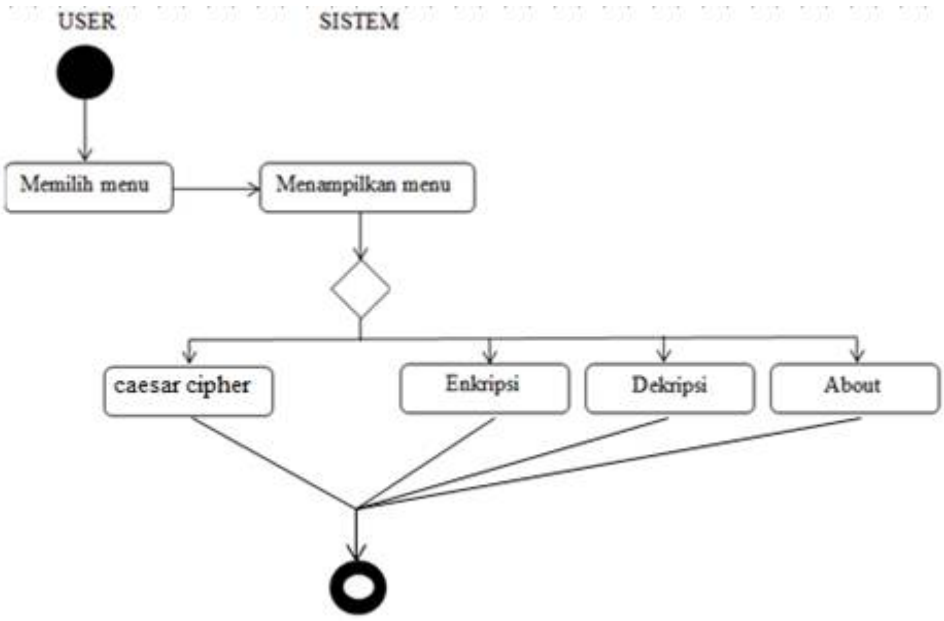
Use case diagram diperlukan untuk menggambarkan fungsional yang diharapkan dari prespektif pengguna. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Use case mempresentasikan sebuah interaksi antara aktor degan sistem. Dalam aplikasi ini use case diagram digambarkan seperti berikut :



Gambar 3.4 *Use Case Diagram*

3.5 *Activity Diagram* Algoritma *Caesar Cipher*

Activity Diagram menggambarkan aktifitas-aktifitas yang terjadi dalam aplikasi dari aktifitas dimulai sampai aktifitas berhenti.



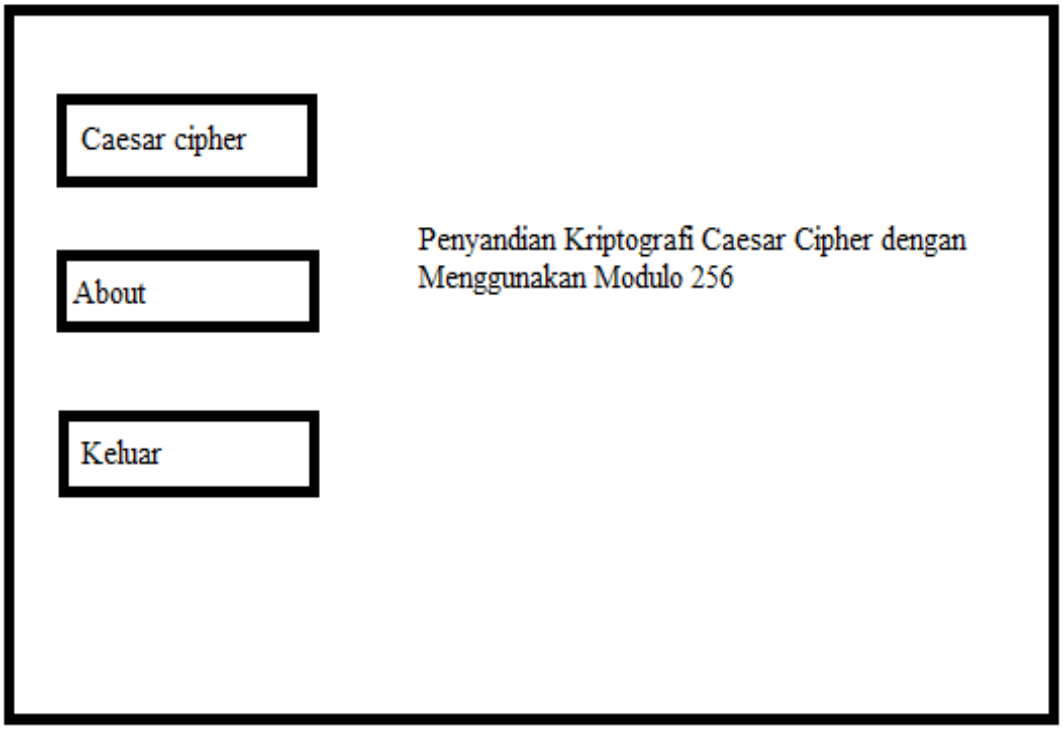
Gambar 3.5 *Activity Diagram*

3.6 Rancangan *Interface*

Pada tahap ini penulis menjelaskan rancangan tampilan halaman yang akan dibangun pada aplikasi yang direncanakan. Adapun tampilan rancangan masing-masing halaman *form* tersebut dapat dijelaskan sebagai berikut.

3.6.1 Rancangan Tampilan Menu Utama

Rancangan ini merupakan tampilan pembuka saat menjalankan aplikasi, dapat dilihat pada gambar dibawah ini :



Gambar 3.6 Rancangan Tampilan Menu Utama

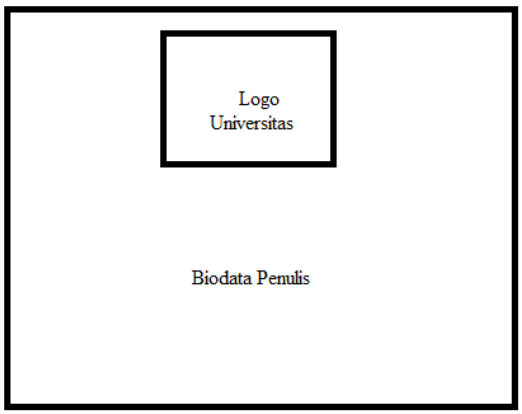
Keterangan :

About : Berfungsi untuk menampilkan tentang pembuat aplikasi ini.

Caesar cipher : Berfungsi untuk menampilkan proses enkripsi dan dekripsi program.

3.6.2 Rancangan Tampilan *About*

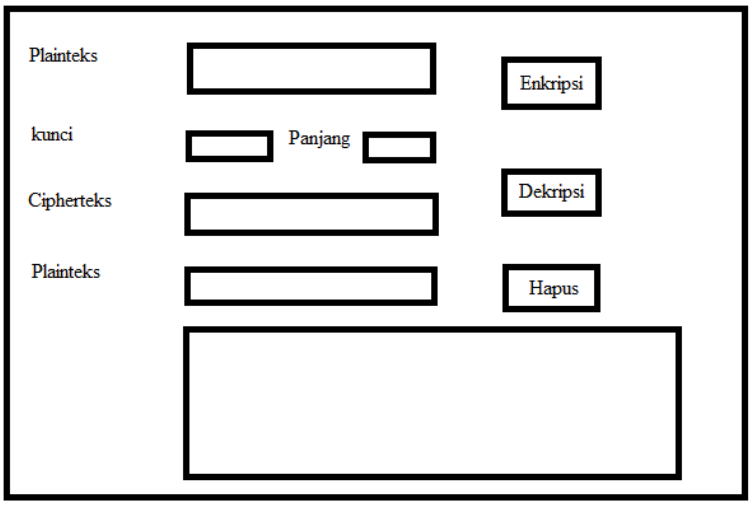
Tampilan *about* ini akan berisikan biodata penulis. Dapat dilihat pada gambar dibawah ini :



Gambar 3.7 Rancangan *Menu About*

3.6.3 Rancangan Tampilan Enkripsi *Caesar Cipher*

Tampilan ini akan berisikan enkripsi *Caesar cipher*, dekripsi, plainteks, kunci, panjang, cipherteks, hapus, dan beserta penjelasannya. Dapat dilihat pada gambar dibawah ini :



Gambar 3.8 Rancangan Tampilan Enkripsi *Caesar Cipher*

Keterangan :

Plainteks	:	Pesan asli yang akan dikirim.
Kunci	:	Berfungsi untuk membuka pesan asli dari plainteks yang dikirim oleh penerima pesan.
Panjang	:	Berfungsi untuk mengetahui seberapa panjang pesan teks.
Cipherteks	:	Bentuk pesan yang tersandi. Cipherteks harus dapat ditransformasi kembali menjadi plainteks.
Enkripsi	:	proses penyandian plainteks menjadi cipherteks.
Dekripsi	:	proses pengembalian cipherteks menjadi plainteks.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

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

4.2 Pengujian Sistem

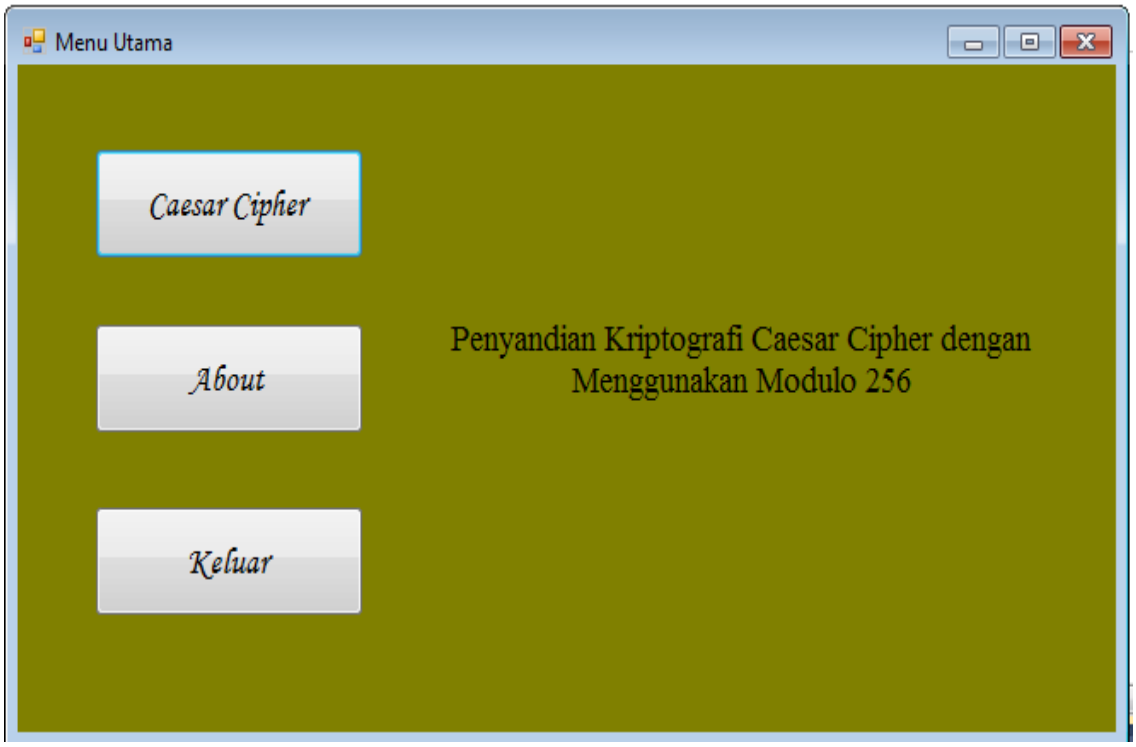
Pengujian sisitem ini dilakukan untuk menunjukkan apakah sistem yang telah dirancang dapat berjalan sesuai harapan. Selain itu tujuan pengujian ini ialah untuk dapat menemukan kesalahan fungsi pada aplikasi yang dibangun dan memperbaikinya.

Pengujian ini dilakukan dengan memasukkan sebuah teks selanjutnya akan diproses oleh aplikasi apakah aplikasi tersebut dapat memberikan hasil yang sesuai. Proses yang akan dilakukan dalam pengujian aplikasi ini adalah simulasi pengiriman pesan teks dengan metode algoritma *caesar cipher* dengan menggunakan kunci sehingga pada akhirnya keaslian pesan tetap terjaga.

4.2.1 Tampilan Awal/*Home*

Tampilan pada gambar dibawah ini merupakan tampilan awal ketika aplikasi dijalankan. Pada *form* ini pengguna dapat memilih untuk membuka beberapa *form*

lainnya seperti tombol *About* yang akan mengarahkan pengguna menuju *form* yang akan menjelaskan *profil* si pembuat apikasi ini, tombol *caesar cipher* yang akan mengarahkan pengguna ke *form* proses jalannya aplikasi ini, tombol keluar yang akan mengarahkan kita unuk keluar/berhenti dari aplikasi ini.



Gambar 4.1 Tampilan Awal/*Home*

Keterangan :

Caesar cipher : Proses untuk melanjutkan jalannya aplikasi ini.

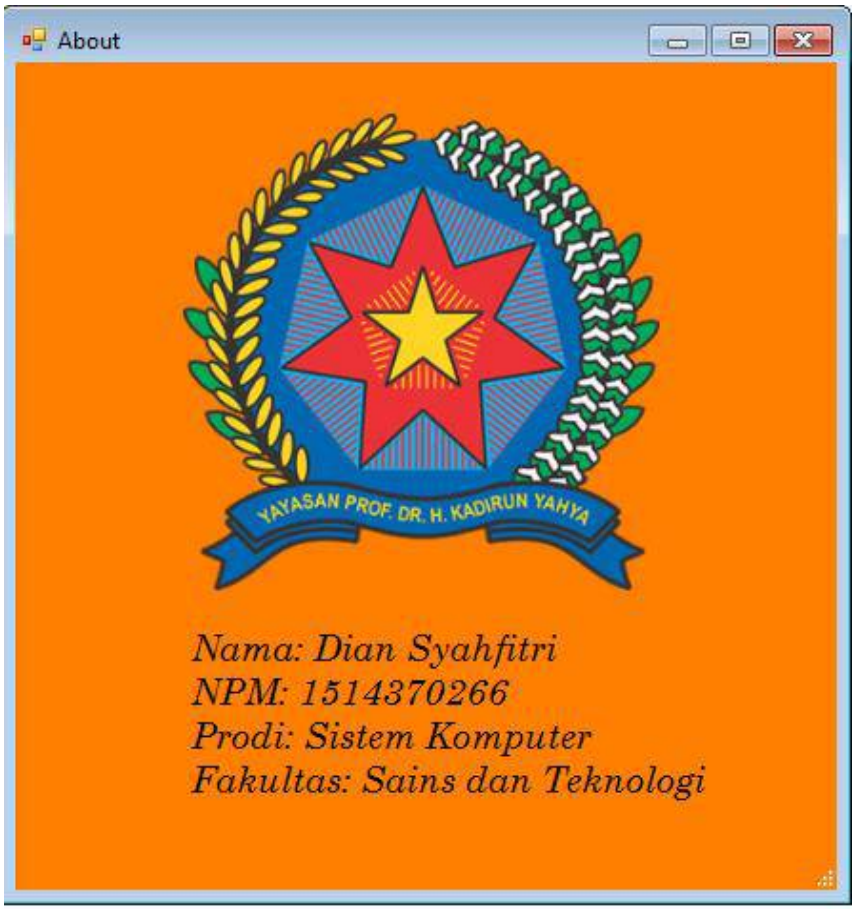
About : Berfungsi untuk menampilkan *profil* si pembuat aplikasi ini.

Keluar : Berfungsi untuk keluar/berhenti dari aplikasi ini.

4.2.2 Tampilan *About*

Tampilan *About* pada aplikasi ini merupakan tampilan halaman atau *form* yang berisi tentang *profil* si pembuat aplikasi ini. Pada *form* ini menjelaskan siapa nama si

pembuat, berapa NPM si pembuat, Program studi si pembuat, dan Fakultas si pembuat. Dapat kita lihat pada gambar dibawah ini :

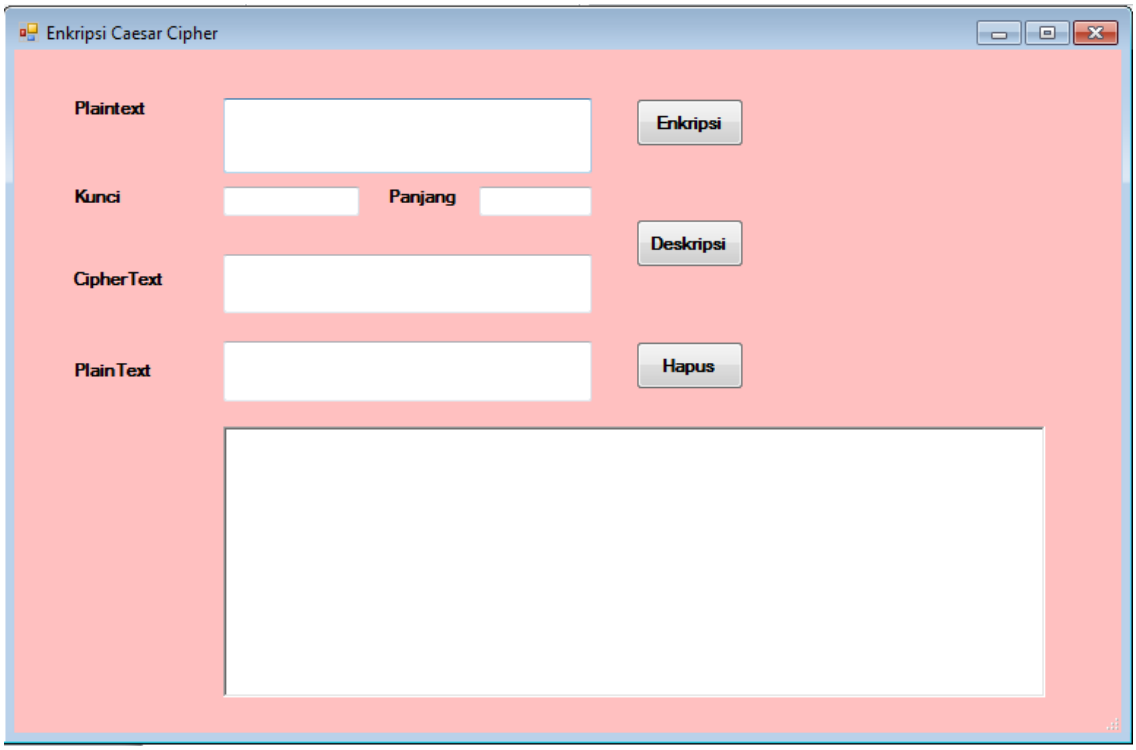


Gambar 4.2 Tampilan *About*

4.2.3 Tampilan Enkripsi *Caesar Cipher*

Tampilan enkripsi ini dilakukan dengan memasukkan teks pada plainteks dan kunci, kemudian tekan tombol enkripsi. Pada tahap ini plainteks akan berubah menjadi cipherteks menggunakan algoritma *caesar cipher*, kemudian akan muncul juga berapa panjang teks. Kemudian cipherteks akan di dekripsi kembali dengan cara memasukkan kembali kunci yang dimasukkan pada awal enkripsi, kemudian munculah hasil plainteks/teks asli yang telah di dekripsi. Tombol hapus berguna untuk menghapus

keseluruhan pesan teks yang dimasukkan. Dapat dilihat pada gambar dibawah ini beserta perhitungannya.



Gambar 4.3 Enkripsi *Caesar Cipher*

Keterangan :

Plainteks : Berfungsi untuk menampilkan pesan asli.

Kunci : Berfungsi untuk membuka pesan tersandi dan pesan asli.

Cipherteks : Berfungsi untuk menampilkan hasil pesan tersandi.

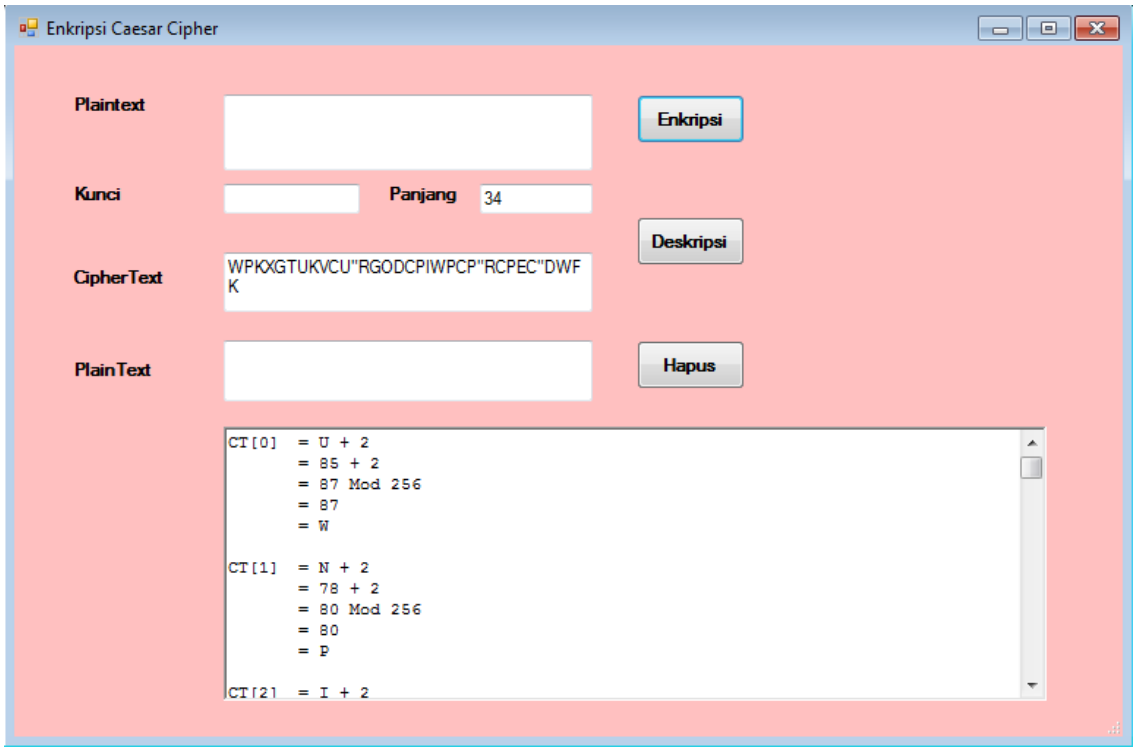
Enkripsi : Mekanisme yang dilakukan untuk merubah plainteks menjadi cipherteks.

Dekripsi : Mekanisme yang dilakukan untuk merubah cipherteks menjadi plainteks.

Hapus : Berfungsi untuk menghapus keseluruhan teks yang ada didalam program.

4.2.4 Perhitungan *Algoritma Caesar*

Berikut ini ialah gambar proses penginputan pesan teks :



Gambar 4.4 Proses Enkripsi Pada Aplikasi

Proses enkripsi diatas dimasukkan pesan teks berupa kata sebagai berikut.

Plainteks : UNIVERSITAS PEMBANGUNAN PANCA BUDI

Kunci : 2

Kemudian diubah menggunakan teknik enkripsi *caesar cipher* menggunakan modulo 256.

Lalu masukkan ke rumus En(x)=(x+n) mod 256.

Perhitungan hasil enkripsi :

CT [0] = U + 2

= 85 + 2

= 87 Mod 256

= 87

= W

CT [1] = N + 2

= 78 + 2

= 80 Mod 256

= 80

= P

CT [2] = I + 2

= 73 + 2

= 75 Mod 256

= 75

= K

CT [3] = V + 2

= 86 + 2

= 88 Mod 256

= 88

= X

CT [4] = E + 2

= 69 + 2

= 71 Mod 256

= 71

= G

CT [5] = R + 2

= 82 + 2

= 84 Mod 256

= 84

= T

CT [6] = S + 2

= 83 + 2

= 85 Mod 256

= 85

= U

CT [7] = I + 2

= 73 + 2

= 75 Mod 256

= 75

= K

CT [8] = T + 2

= 84 + 2

= 86 Mod 256

= 86

= V

CT [9] = A + 2

= 65 + 2

= 67 Mod 256

= 67

= C

CT [10] = S + 2

= 83 + 2

= 85 Mod 256

= 85

= U

CT [11] = + 2

= 32 + 2

= 34 Mod 256

= 34

= “

CT [12] = P + 2

= 80 + 2

= 82 Mod 256

= 82

= R

CT [13] = E + 2

= 69 + 2

= 71 Mod 256

= 71

= G

CT [14] = M + 2

= 77 + 2

= 79 Mod 256

= 79

= O

CT [15] = B + 2

= 66 + 2

= 68 Mod 256

= 68

= D

CT [16] = A + 2

= 65 + 2

= 67 Mod 256

= 67

= C

CT [17] = N + 2

= 78 + 2

= 80 Mod 256

= 80

= P

CT [18] = G + 2

= 71 + 2

= 73 Mod 256

= 73

= I

CT [19] = U + 2

= 85 + 2

= 87 Mod 256

= 87

= W

CT [20] = N + 2

= 78 + 2

= 80 Mod 256

= 80

= P

CT [21] = A + 2

= 65 + 2

= 67 Mod 256

= 67

= C

CT [22] = N + 2

= 78 + 2

= 80 Mod 256

= 80

= P

CT [23] = + 2

= 32 + 2

= 32 Mod 256

= 34

= “

CT [24] = P + 2

= 80 + 2

= 82 Mod 256

= 82

= R

CT [25] = A + 2

= 65 + 2

= 67 Mod 256

= 67

= C

CT [26] = N + 2

= 78 + 2

= 80 Mod 256

= 80

= P

CT [27] = C + 2

= 67 + 2

= 69 Mod 256

= 69

= E

CT [28] = A + 2

= 65 + 2

= 67 Mod 256

= 67

= C

CT [29] = + 2

= 32 + 2

= 34 Mod 256

= 34

= “

CT [30] = B + 2

= 66 + 2

= 68 Mod 256

= 68

= D

CT [31] = U + 2

= 85 + 2

= 87 Mod 256

= 87

= W

CT [32] = D + 2

= 68 + 2

= 70 Mod 256

= 70

= F

CT [33] = I + 2

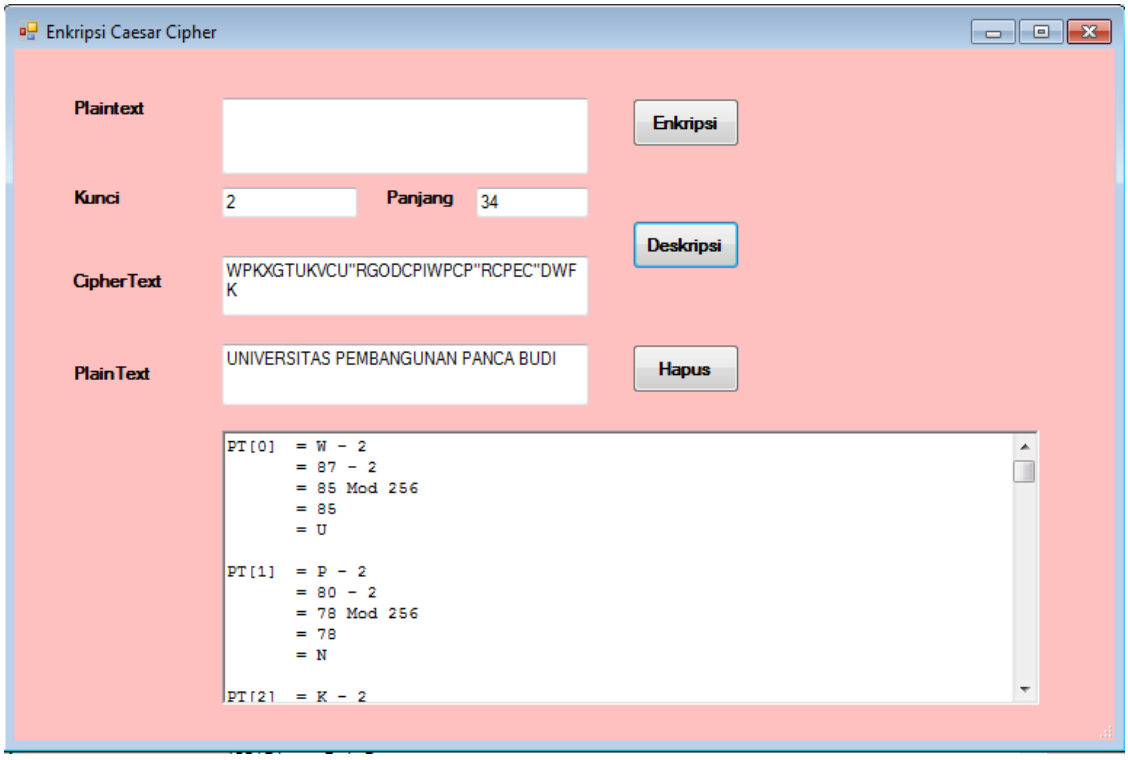
= 73 + 2

= 75 Mod 256

= 75

= K

Jadi, setelah dilakukan proses enkripsi, maka hasil cipherteks dari kata tersebut setelah ialah WPKXGTUKVCU"RGODCPIWPCP"RCPEC"DWFK. Untuk mendekripsikannya masukkan kembali kunci cipherteks ke dalam rumus Dn(x)=(x-n) Mod 256. Terlihat seperti gambar dibawah ini :



Gambar 4.5 Proses Deskripsi Pada Aplikasi

Perhitungan hasil deskripsi :

PT [0] = W – 2

= 87 – 2

= 85 Mod 256

= 85

= U

PT [1] = P – 2

= 80 – 2

= 78 Mod 256

= 78

= N

PT [2] = K – 2

= 75 – 2

= 73 Mod 256

= 73

= I

PT [3] = X – 2

= 88 – 2

= 86 Mod 256

= 86

= V

PT [4] = G – 2

= 71 – 2

= 69 Mod 256

= 69

= E

PT [5] = T – 2

= 84 – 2

= 82 Mod 256

= 82

= R

PT [6] = U – 2

= 85 – 2

= 83 Mod 256

= 83

= S

PT [7] = K – 2

= 75 – 2

= 73 Mod 256

= 73

= I

PT [8] = V – 2

= 86 – 2

= 84 Mod 256

= 84

= T

PT [9] = C – 2

= 67 – 2

= 65 Mod 256

= 65

= A

PT [10] = U – 2

= 85 – 2

= 83 Mod 256

= 83

= S

PT [11] = “ – 2

= 34 – 2

= 32 Mod 256

= 32

=

PT [12] = R – 2

= 82 – 2

= 80 Mod 256

= 80

= P

PT [13] = G – 2

= 71 – 2

= 69 Mod 256

= 69

= E

PT [14] = O – 2

= 79 – 2

= 77 Mod 256

= 77

= M

PT [15] = D – 2

= 68 – 2

= 66 Mod 256

= 66

= B

PT [16] = C – 2

= 67 – 2

= 65 Mod 256

= 65

= A

PT [17] = P – 2

= 80 – 2

= 78 Mod 256

= 78

= N

PT [18] = I – 2

= 73 – 2

= 71 Mod 256

= 71

= G

PT [19] = W – 2

= 87 – 2

= 85 Mod 256

= 85

= U

PT [20] = P – 2

= 80 – 2

= 78 Mod 256

= 78

= N

PT [21] = C – 2

= 67 – 2

= 65 Mod 256

= 65

= A

PT [22] = P – 2

= 80 – 2

= 78 Mod 256

= 78

= N

PT [23] = “ – 2

= 34 – 2

= 32 Mod 256

= 32

=

PT [24] = R – 2

= 82 – 2

= 80 Mod 256

= 80

= P

PT [25] = C – 2

= 67 – 2

= 65 Mod 256

= 65

= A

PT [26] = P – 2

= 80 – 2

= 78 Mod 256

= 78

= N

PT [27] = E – 2

= 69 – 2

= 67 Mod 256

= 67

= C

PT [28] = C – 2

= 67 – 2

= 65 Mod 256

= 65

= A

PT [29] = “ – 2

= 34 – 2

= 32 Mod 256

= 32

=

PT [30] = D – 2

= 68 – 2

= 66 Mod 256

= 66

= B

PT [31] = W – 2

= 87 – 2

= 85 Mod 256

= 85

= U

PT [32] = F – 2

= 70 – 2

= 68 Mod 256

= 68

= D

PT [33] = K – 2

= 75 – 2

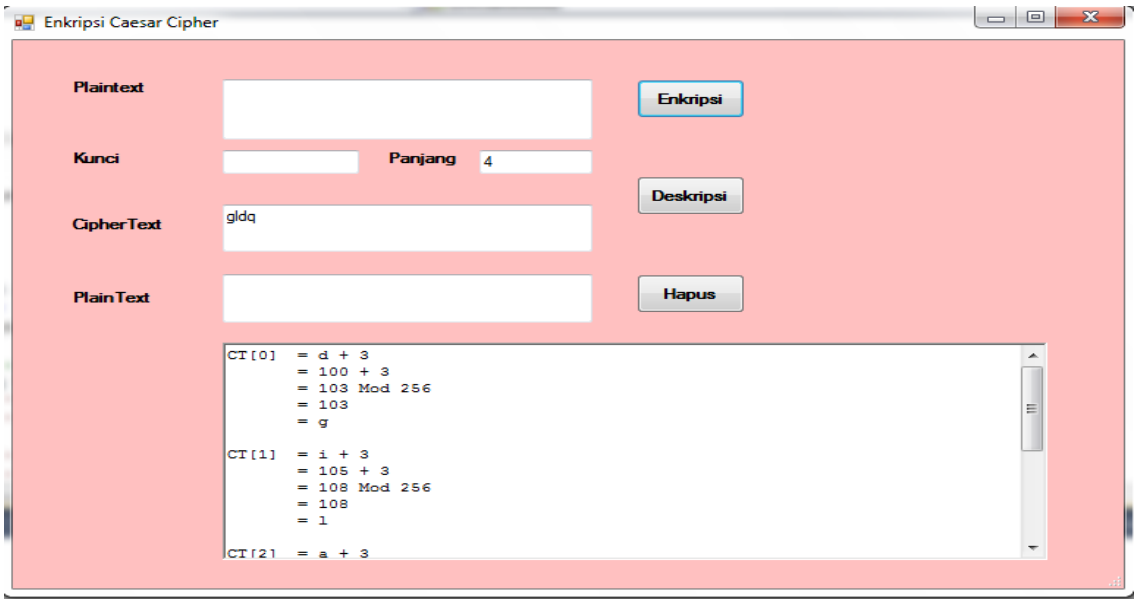
= 73 Mod 256

= 73

= I

Jadi, setelah dilakukan proses dekripsi, maka hasil akhir dari proses deskripsi tersebut ialah UNIVERSITAS PEMBANGUNAN PANCA BUDI.

Contoh ke 2 Penginputan pesan teks :



Gambar 4.6 Proses Enkripsi Pada Aplikasi

Proses enkripsi diatas dimasukkan pesan teks berupa kata sebagai berikut.

Plainteks : dian

Kunci : 3

Kemudian diubah menggunakan teknik enkripsi *caesar cipher* menggunakan modulo 256.

Lalu masukkan ke rumus En(x)=(x+n) mod 256.

Perhitungan hasil enkripsi :

CT [0] = d + 3

= 100 + 3

= 103 Mod 256

= 103

= g

CT [1] = i + 3

= 105 + 3

= 108 Mod 256

= 108

= l

CT [2] = a + 3

= 97 + 3

= 100 Mod 256

= 100

= d

CT [3] = n + 3

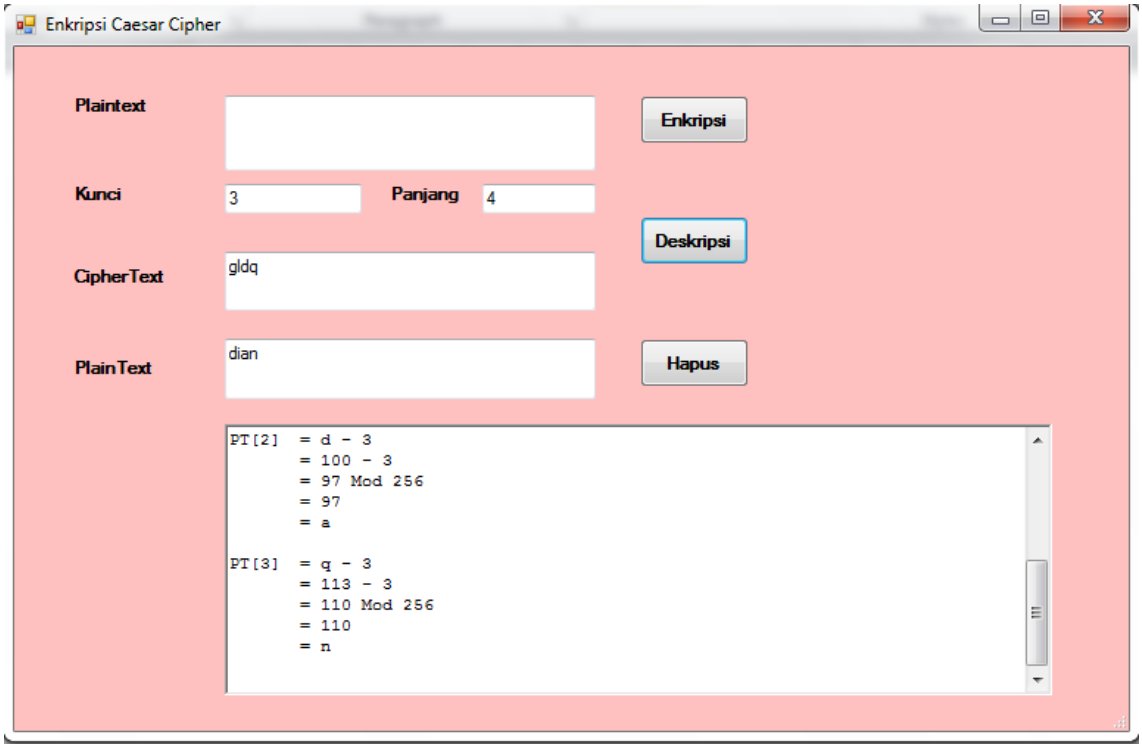
= 110 + 3

= 113 Mod 256

= 113

= q

Jadi, setelah dilakukan proses enkripsi, maka hasil cipherteks dari kata tersebut setelah ialah gldq. Untuk mendekripsikannya masukkan kembali kunci cipherteks ke dalam rumus Dn(x)=(x-n) Mod 256. Terlihat seperti gambar dibawah ini :



Gambar 4.7 Proses Deskripsi Pada Aplikasi

Perhitungan hasil deskripsi :

PT [0] = g – 3

= 103 – 3

= 103 Mod 256

= 100

= d

PT [1] = l – 3

= 108 – 3

= 105 Mod 256

= 105

= i

PT [2] = d – 3

= 100 – 3

= 97 Mod 256

= 97

= a

PT [3] = q – 3

= 113 – 3

= 110 Mod 256

= 110

= n

Jadi, setelah dilakukan proses dekripsi, maka hasil akhir dari proses deskripsi tersebut ialah dian.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan dari hasil pembahasan penyandian kriptografi metode *caesar cipher*, maka dapat diambil kesimpulan sebagai berikut :

1. Perangkat lunak ini dirancang agar pengguna mengetahui keamanan pesan teks kriptografi dengan menggunakan metode *caesar cipher*.
2. Pada aplikasi ini proses keamanan pesan teks dilakukan dengan cara pergeseran.
3. Dalam aplikasi ini hanya menggunakan satu kunci.
4. Kemungkinan bocornya kunci saat proses keamanan pesan teks dapat dihindari.

5.2 Saran

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

1. Perangkat lunak ini dapat dikembangkan dengan menggunakan kombinasi-kombinasi metode lain.
2. Perangkat lunak ini dapat dikembangkan sehingga dapat dijalankan lebih dari satu komputer.

Daftar Pustaka

Ade hendini. (2016). Pemodelan uml sistem informasi monitoring penjualan dan stok barang (studi kasus: distro zhezha pontianak). *Jurnal khatulistiwa informatika*, vol. Iv, no. 2.

Albert ginting. (2015). Implementasi algoritma kriptografi rsa untuk enkripsi dan Dekripsi email. *Jurnal teknologi dan sistem komputer*. Vol. 3, no. 2.

Andi. (2012). *Visual basic 2010 programming*, yogyakarta: andi offset.

Anjar pradipta. (2016). Implementasi metode caesar cipher alphabet majemuk dalam kriptograf untuk pengamanan informasi. *Indonesianjournal on networking and security*. Volume 5, no 3.

Atmaja basuki, u, r. (2016). Perancangan aplikasi kriptografi berlapis menggunakan algoritma caesar, transposisi, vigenere, dan blok cipher berbasis mobile. *Seminar nasional teknologi informasi dan multimedia*. Issn: 2302-3805.

Dwi nurani, (2018). Perancangan aplikasi email menggunakan algoritma caesar Cipher dan base64. *Jiska (jurnal informatika sunan kalijaga).* Vol. 2, no.3.

Endah handayani, (2017). Perancangan aplikasi kriptografi berbasis web dengan algoritma double caesar cipher menggunakan tabel ascii. *Seminar nasional teknologi informasi dan multimedia 2017*. Issn: 2302-3805.

Fachri, barany, agus perdana windarto, and ikhsan parinduri. "penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik." jepin (jurnal edukasi dan penelitian informatika) 5.2 (2019): 202-208.

Fachri, b., windarto, a. P., & parinduri, i. (2019). Penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik. Jepin (jurnal edukasi dan penelitian informatika), 5(2), 202-208.

Fachri, barany; windarto, agus perdana; parinduri, ikhsan. Penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik. Jepin (jurnal edukasi dan penelitian informatika), 2019, 5.2: 202-208.

Fisai anistasari sinaga, m. (2017). Implementasi algoritma rot13 dan algoritma caesar cipher dalam penyandian teks. *Pelita informatikan budi darma, xvi, nomor,38-41*.

Hamdi, nurul. "model penyiraman otomatis pada tanaman cabe rawit berbasis programmable logic control." jurnal ilmiah core it: community research information technology 7.2 (2019).

Indra gunawan. (2018). Kombinasi algoritma caesar cipher dan algoritma rsa untuk pengamanan file dokumen dan pesan teks. *Jurnal nasional dan teknologi jaringan*. Vol 2, no 2.

Mamed rofendy manalu. (2015). *Implementasi sistem informasi penyewaan mobil pada cv. Btn padang bulan dengan metode waterfall*. Issn: 2088-3943.

Muhammad fadlan, h. (2017). Rekayasa aplikasi kriptografi dengan penerapan kombinasi algoritma knapsack merkle hellman dan affine cipher. *Jurnal teknologi dan ilmu komputer (jtiik)*. Vol 4, no.4.

Muhammad nurtanzis sutoyo, m. (2016). Kombinasi algoritma kriptografi caesar cipher dan vigenere cipher untuk keamanan data. *Jurnal mekanova*. Vol 2, no. 1.

Munir, rinaldi. (2019). *Kriptografi*. Bandung: informatika bandung.

Murdani. (2017). Perancangan aplikasi keamanan data teks menggunakan algoritma merkle hellman knapsack. *Jurnal pelita informatika*. Volume 16, nomor 3, 284-302.

Nurdam, nofiyandi. (2014). *Sequence diagram sebagai perkakas perancangan antarmuka pemakai*. Issn: 2085-4552.

Permana, aminuddin indra. "kombinasi algoritma kriptografi one time pad dengan generate random keys dan vigenere cipher dengan kunci em2b." (2019).

Putra, randi rian. "sistem informasi web pariwisata hutan mangrove di kelurahan belawan sicanang kecamatan medan belawan sebagai media promosi." jurnal ilmiah core it: community research information technology 7.2 (2019).

Putra, randi rian, et al. "decision support system in selecting additional employees using multi-factor evaluation process method." (2019).

Putra, randi rian. "implementasi metode backpropagation jaringan saraf tiruan dalam memprediksi pola pengunjung terhadap transaksi." jurti (jurnal teknologi informasi) 3.1 (2019): 16-20.

Priyono. (2016). Penerapan algoritma caesar cipher dan algoritma vigenere cipher dalam pengamanan pesan teks. *Jurnal riset komputer (jurikom)*. Volume: 3, nomor: 5.

Risky, soetam. (2010). *Learning by sample visual basic 2008*. Jakarta: prestasi pustakaraya.

Robbi rahim. (2016). Penyisipan pesan dengan algoritma pixel value differencing dengan algoritma caesar cipher pada proses steganografi. *Jurnal times*. Vol. V, no 1.

Saputra, muhammad juanda, and nurul hamdi. "rancang bangun aplikasi sejarah kebudayaan aceh berbasis android studi kasus dinas kebudayaan dan pariwisata aceh." journal of informatics and computer science 5.2 (2019): 147-157

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.

Sitepu, n. B., zarlis, m., efendi, s., & dhany, h. W. (2019, august). Analysis of decision tree and smooth support vector machine methods on data mining. In journal of physics: conference series (vol. 1255, no. 1, p. 012067). Iop publishing.

Sri dharwiyanti. (2003). Pengantar unified modeling language (uml). 1-13.

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).

Tri a kurniawan. (2018). Pemodelan use case (uml): evaluasi terhadap beberapa kesalahan dalam praktik. *Jurnal teknologi informasi dan ilmu komputer (jtiik)*. Vol. 5, no. 1.