BAB 2

ANALISIS DAN PERANCANGAN

2.1 Kriptografi

Kriptografi berasal dari bahasa Yunani, menurut bahasa di bagi menjadi "*kripto*" dan "*graphia*", *kripto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Menurut terminologinya kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lainnya.

2.2 Algoritma Kriptografi

Algoritma kriptografi terdiri dari tiga fungsi dasar, yaitu :

- **Enkripsi**

  Enkripsi merupakan hal yang sangat penting dalam kriptografi yang merupakan pengamanan data yang dikirimkan terjaga rahasianya. Pesan asli disebut *plaintext* yang dirubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan dengan cipher atau kode. Sama halnya dengan kita tidak mengerti akan sebuah kata, maka kita akan melihatnya didalam kamus atau daftar istilah-istilah.

  Beda halnya degan enkripsi, untuk merubah *plaintext* ke bentuk *chipertext* kita menggunakan algoritma yang dapat mengkodekan data yang kita inginkan.

- **Dekripsi**

  Dekripsi merupakan kebalikan dari enkripsi, pesan yang telah dienkripsi di kembalikan kebentuk aslinya (*plaintext*) disebut dengan dekripsi pesan. Algoritma yang digunakan untuk dekripsi tentu berbeda dengan yang digunakan untuk enkripsi.

- **Kunci**

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

Keamanan dari algoritma kriptografi tergantung dari bagaimana suatu algoritma itu bekerja, maka algoritma semacam ini disebut dengan algoritma terbatas, yang merupakan suatu algoritma yang dipakai sekelompok orang untuk merahasiakan pesan yang dikirimnya. Jika salah satu dari anggota kelompok itu keluar dari kelompoknya maka, algoritma yang dipakai diganti dengan yang baru, jika tidak hal ini bisa menjadi masalah.

2.2.1 Caesar Cipher

Substitusi *cipher* yang pertama dalam dunia penyandian pada waktu pemerintahan Yulius Caesar yang dikenal dengan *Caesar Cipher,* dengan mengganti posisi huruf awal dari alphabet sebagai contoh :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
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

Menjadi

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
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	0	1	2

Contoh dari algoritma *caesar cipher*, untuk *plaintext* diberikan simbol “P” *ciphertext* diberikan simbol “C” dan kunci “K”. jadi rumusannya dapat dibuat sebagai berikut : (Ariyus, 2006, hal 18)

$$C = E(P) = (P+K) \bmod (26)$$

Dan rumus untuk dekripsi sebagai berikut :

$$P = D(C) = (C-K) \bmod (26)$$

Jika diberikan *plaintext*, sebagai berikut :

“KENAIKAN HARGA BBM MEMBUAT RAKYAT KECIL MENDERITA”

Dengan menggunakan kunci “3" maka akan di dapat *ciphertext* sebagai berikut :

“NHQDLNDQKDUJDEEPPHPEXDWUDNBDWNHFLOPHQGHULWD”

Rumus di atas adalah rumus dasar metode *caesar cipher* yang hanya dapat mengenal pengkodean huruf saja. Untuk itu programer mengembangkan metode *caesar cipher* untuk dapat mengkodekan huruf, simbol dan angka menggunakan rumus enkripsi sebagai berikut :

$$C = P \text{ (Desimal dari } \textit{plaintext}) + K$$

Sedangkan untuk rumus dekripsi, sebagi berikut :

$$P = C \text{ (Desimal dari } \textit{ciphertext}) - K$$

Jika di berikan *plaintext* sebagai berikut :

“Diskon 50%”

Dengan menggunakan kunci “3” maka akan di dapat *ciphertext* sebagai berikut :

“Glvnrq#83(“

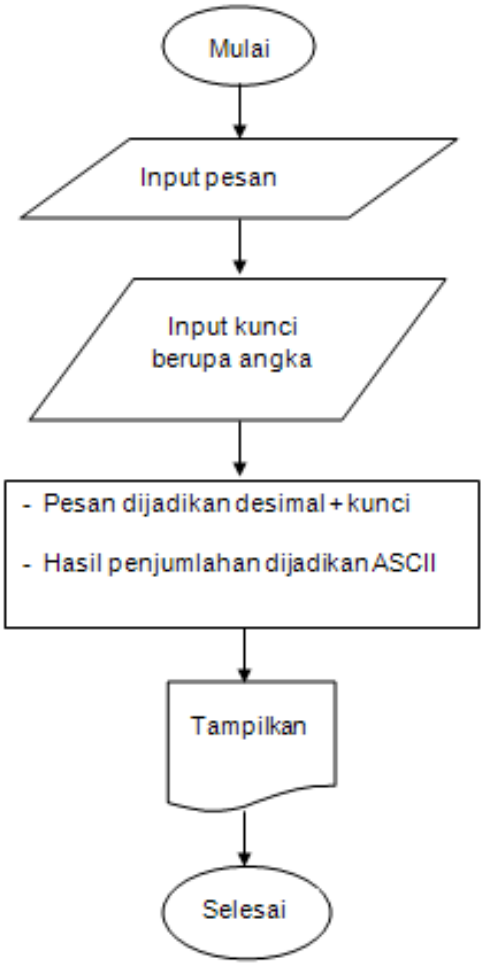
Penyelesaian :

D	i	s	k	o	n		5	0	%
68	105	115	107	111	110	32	53	48	37
3	3	3	3	3	3	3	3	3	3
G	l	v	n	r	q	#	8	3	(

Berikut adalah algoritma enkripsi dari pengembangan metode *caesar cipher* :

1. Masukkan pesan.
2. Masukkan kunci berupa angka.
3. Desimal dari pesan teks akan ditambah dengan kunci.
4. Hasil dari proses penjumlahan akan dijadikan ASCII.
5. Data yang dijadikan ASCII akan ditampilkan.
6. Selesai.

Berikut adalah algoritma enkripsi dari *caesar cipher* dalam bentuk *flowchart*.

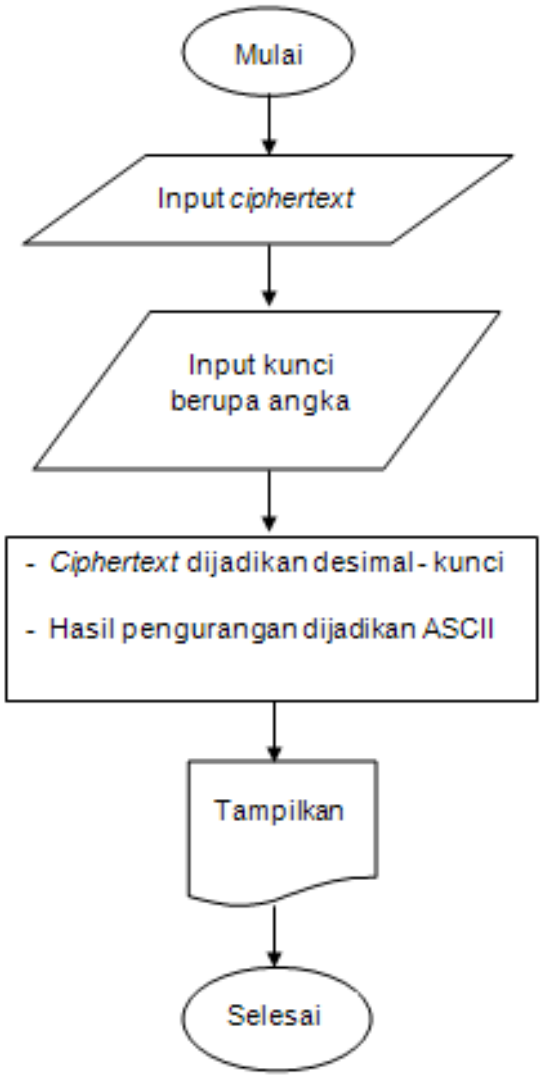


Gambar 2.1 *Flowchart* Enkripsi *Caesar Cipher*

Berikut adalah algoritma dekripsi dari metode *caesar cipher*:

1. Masukkan *ciphertext*.
2. Masukkan kunci berupa angka.
3. Desimal dari *ciphertext* akan dikurangi dengan kunci.
4. Hasil dari proses pengurangan akan dijadikan ASCII.
5. Data yang dijadikan ASCII akan ditampilkan.
6. Selesai.

Berikut adalah algoritma dekrisi dari *caesar cipher* dalam bentuk *flowchart*.



Gambar 2.2 *Flowchart* Dekripsi *Caesar Cipher*

2.2.2 Vigenere Cipher

Bila pada teknik diatas setiap *ciphertext* selalu menggantikan nilai dari setiap *plaintext* tertentu (tidak peduli apakah jumlah dari *ciphertext* yang ekivalen dengan *plaintext* tertentu satu atau lebih) pada teknik substitusi *vigenere* setiap *ciphertext* bisa memiliki banyak kemungkinan *plaintext*.

Teknik dari substitusi *vigenere* bisa dilakukan dengan dua cara :

- Angka
- Huruf

a. Angka

Teknik substitusi *vigenere* dengan menggunakan angka dengan menukarkan huruf dengan angka.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Kita memiliki kunci dengan 6 huruf CIPHER jika kita tukar dengan angka maka akan menjadi K = (2, 8, 15, 7, 4, 17), dan plaintextnya “This Cryptosystem is Not Secure”

T	H	I	S	C	R	Y	P	T	O	S	Y	S	T
19	7	8	18	2	17	24	15	19	14	18	24	18	19
2	8	15	7	4	17	2	8	15	7	4	17	2	8
21	15	23	25	6	8	0	23	8	21	22	15	20	1

E	M	I	S	N	O	T	S	E	C	U	R	E
4	12	8	18	13	14	19	18	4	2	20	17	4
15	7	4	17	2	8	15	7	4	17	2	8	15
19	19	12	9	15	22	8	0	25	19	22	25	19

Plaintext : This cryptosystem is not secure

Kunci : (2, 8, 15, 7, 4, 17)

Ciphertext : VPXZGIAXIVWPUBTTMJPWIZITWZT

Untuk melakukan dekripsi, kita juga bisa menggunakan kunci yang sama dengan modulo 26.

b. Huruf

Teknik *vigenere* dengan huruf bisa menggunakan cara di bawah ini.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

P = STMIK AKAKOM

K = KAMPUS

Penyelesainan

S	T	M	I	K	A	K	A	K	O	M
18	19	12	8	10	0	10	0	10	14	12
K	A	M	P	U	S	K	A	M	P	U
10	0	12	15	20	18	10	0	12	15	20
Jumlahkan										
1	19	24	23	3	18	20	0	22	2	6

Hasilnya										
B	T	Y	X	D	S	U	A	W	C	G

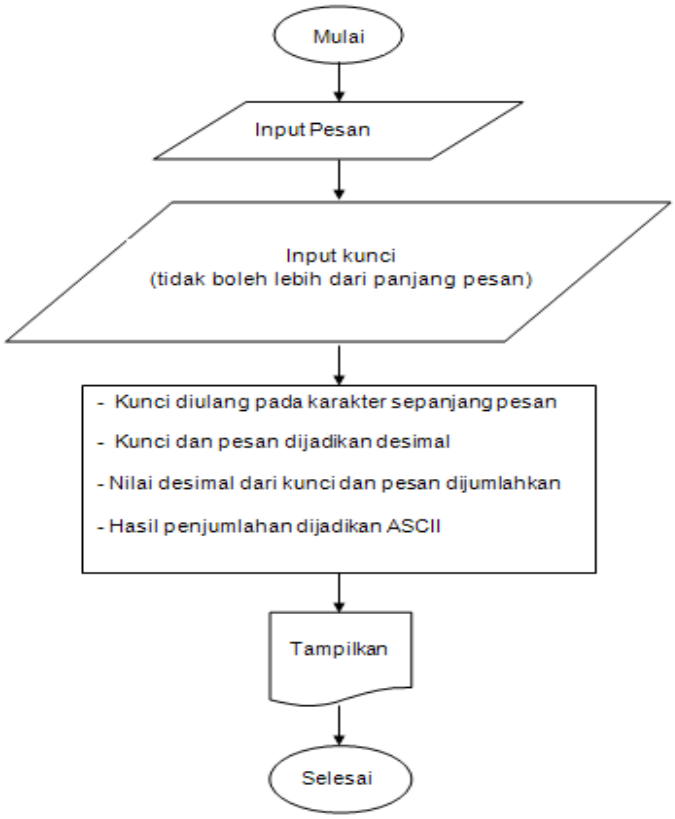
Jadi hasilnya adalah BTYXDSUAWCG

C = BTYXDSUAWCG

Berikut adalah algoritma enkripsi dari metode *vigenere cipher* :

1. Masukkan pesan teks.
2. Masukkan kunci (tidak boleh lebih dari panjang pesan).
3. kunci diulang pada karakter sepanjang pesan.
4. Kunci dan pesan dijadikan desimal.
5. Nilai desimal dari Kunci dan pesan dijumlahkan.
6. Hasil penjumlahan dijadikan ASCII.
7. Data yang dijadikan ASCII ditampilakn.
8. Selesai.

Berikut adalah algoritma enkripsi dari *vegenere cipher* dalam bentuk *flowchart*.

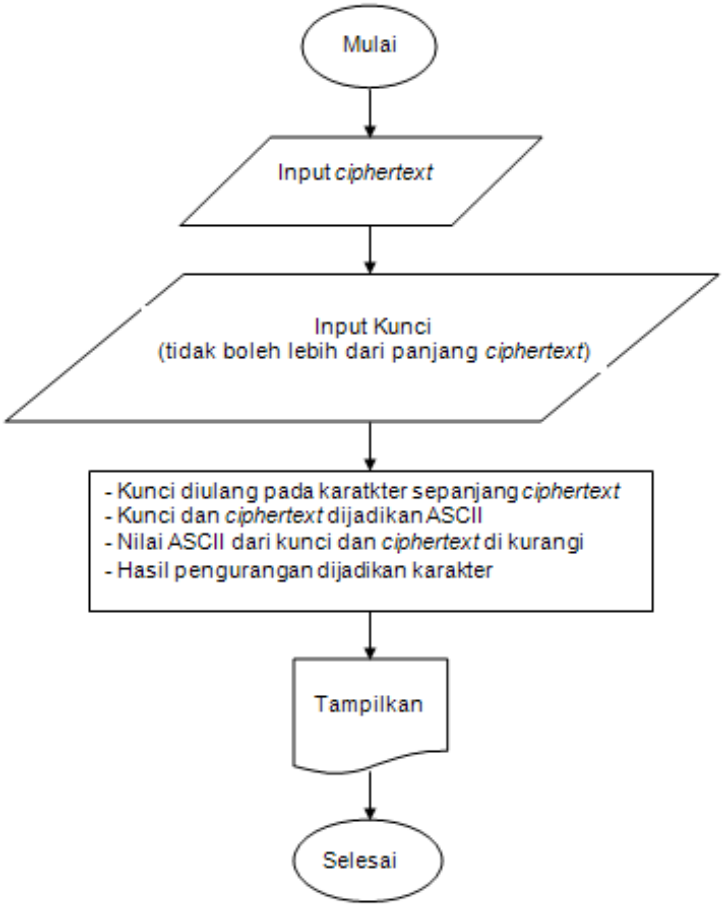


Gambar 2.3 *Flowchart* Enkripsi *Vegenere Cipher*

Berikut adalah algoritma dekripsi dari metode *vigenere cipher* :

1. Masukkan pesan teks.
2. Masukkan kunci (tidak boleh lebih dari panjang pesan).
3. kunci diulang pada karakter sepanjang pesan.
4. Kunci dan pesan dijadikan desimal.
5. Nilai desimal pesan dikurangi nilai desimal kunci.
6. Hasil pengurangan dijadikan ASCII.
7. Data yang dijadikan ASCII ditampilakn.
8. Selesai.

Berikut adalah algoritma enkripsi dari *vegenere cipher* dalam bentuk *flowchart*.



Gambar 2.4 *Flowchart* Dekripsi *Vegenere Cipher*

2.3 Kode ASCII

Kode Standar Amerika untuk Pertukaran Informasi atau *ASCII* (*American Standard Code for Information Interchange*) merupakan suatu standar internasional dalam kode huruf dan simbol seperti *Hex* dan *Unicode* tetapi *ASCII* lebih bersifat *universal*, contohnya 124 adalah untuk karakter "|". Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode *ASCII* sebenarnya memiliki komposisi bilangan biner sebanyak 8 bit. Dimulai dari 0000

0000 hingga 1111 1111. Total kombinasi yang dihasilkan sebanyak 256, dimulai dari kode 0 hingga 255 dalam sistem bilangan Desimal. Tabel kode *ASCII* terlampir.

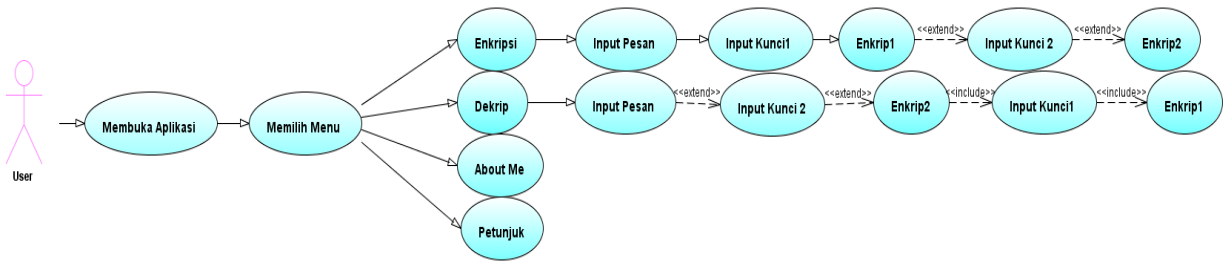
2.4 Perancangan Sistem

2.4.1. Pemodelan

Dalam pemodelan sistem, pada pembuatan aplikasi ini menggunakan metode object oriented dengan UML (*Unified Modelling Language*). Pengembangan system ini menggunakan diagram-diagram yang bisa mewakili dari beberapa yang ada pada UML, diantaranya adalah :

Use Case Diagram

Menggambarkan secara grafis prilaku software aplikasi :



Gambar 2.5 Use Case Diagram

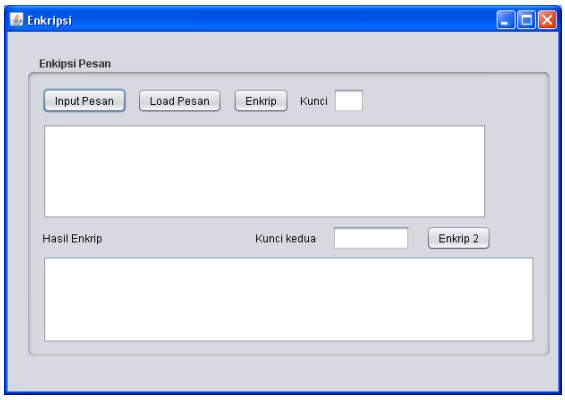
Class Diagram

Menggambarkan hubungan antar kelas dan penjelasan.

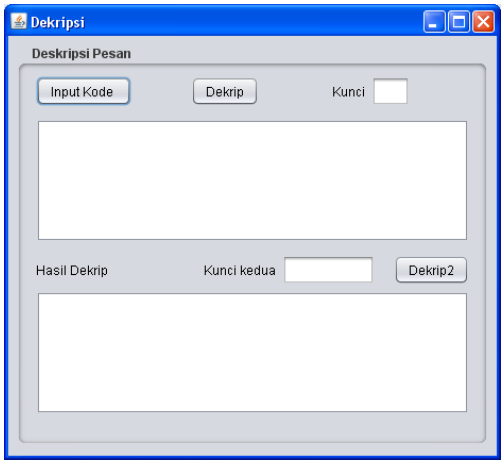
Gambar 2.6 Class Diagram

Gambaran Program

Gambar 2.7 Tampilan Pertama



Gambar 2.8 Tampilan Enkripsi



Gambar 2.9 Tampilan Dekripsi

Gambar 2.10 Tampilan About Me