



**PEMBUATAN APLIKASI KRIPTOGRAFI *FILE*
MENGUNAKAN ALGORITMA
*VIGENERE 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 : YULIARISTIA Br. PURBA
NPM : 1514370165
PROGRAM STUDI : SISTEM KOMPUTER**

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019**

ABSTRAK

YULI ARISTIA BR. PURBA

**Pembuatan Aplikasi Kriptografi *File* Menggunakan Algoritma *Vigenere Cipher*
2019**

Perkembangan teknologi informasi memiliki pengaruh yang sangat signifikan bagi aspek kehidupan. Tidak terkecuali pada aspek komunikasi dan pengiriman pesan. Masalah keamanan dan kerahasiaan isi *file* (teks) merupakan hal yang penting yang bersifat rahasia dan perlu dibuatkan sistem penyimpanan dan pengirimannya agar tidak terbaca atau diubah oleh orang-orang yang tidak bertanggung jawab baik saat informasi teks dikirim melalui e-mail. Adapun tujuan yang ingin dicapai yaitu, untuk mengamankan isi *file* (teks) yang akan dikirim ke e-mail, dan membuat sistem pengamanan informasi teks dengan proses enkripsi dan dekripsi. Metode yang digunakan dalam pembuatan aplikasi kriptografi ini yaitu metode *Vigenere Cipher* yang bersifat sistem substitusi multi-alphabet dan memiliki kunci simetris dimana kunci enkripsi sama dengan kunci dekripsi, disini kunci yang dipakai untuk enkripsi dan dekripsi adalah angka. Hasil penelitian menunjukkan bahwa isi *file* (teks) dapat terenkripsi menggunakan algoritma *Vigenere Cipher*, lalu dapat didekripsikan sesuai dengan kunci yang ditetapkan saat proses enkripsi. Dengan adanya pembuatan aplikasi ini dapat mengamankan kerahasiaan informasi teks yang akan dikirim melalui e-mail agar tidak dapat diubah oleh pihak ketiga atau orang-orang yang tidak bertanggung jawab.

Kata Kunci : Enkripsi, Dekripsi, Kriptografi, *Vigenere Cipher*

DAFTAR ISI

	Halaman
KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iv
DAFTAR TABEL	v
DAFTAR LAMPIRAN	vi
DAFTAR ISTILAH	vii
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB II LANDASAN TEORI	
2.1 Pengertian Aplikasi	5
2.2 Pengertian Kriptografi	6
2.3 Pengertian File	8
2.4 Pengertian Teks	9
2.5 Kode ASCII.....	9
2.6 Algoritma Kriptografi.....	12
2.6.1 Algoritma Simetris	13
2.6.2 Algoritma Asimetris	14
2.7 Algoritma <i>Vigenere Cipher</i>	15
2.8 Bahasa Pemrograman	17
2.9 <i>Visual Basic.Net</i>	19
2.10 <i>Flowchart</i>	21
BAB III METODE PENELITIAN	
3.1 Tahapan Penelitian	24
3.2 Metode Pengumpulan Data.....	25
3.3 Analisis Sistem yang Sedang Berjalan.....	26
3.3.1 Analisa Kelemahan yang Berjalan	26
3.3.2 Solusi Pemecahan Masalah.....	27
3.4 Rancangan Penelitian	28
3.4.1 <i>Flowchart</i> Sistem.....	31
3.4.2 Perancangan Antarmuka	33
BAB IV HASIL DAN PEMBAHASAN	
4.1 Implementasi Sistem	38
4.4.1 Spesifikasi Sistem.....	38
4.2 Pengujian Sistem dan Pembahasan	39
4.2.1 Validasi sistem	48

BAB V PENUTUP

5.1 Simpulan54

5.2 Saran54

DAFTAR PUSTAKA

BIOGRAFI PENULIS

LAMPIRAN-LAMPIRAN

KATA PENGANTAR

Puji Tuhan yang Maha Esa karena dengan berkat dan kasih anugerah-Nya penulis masih diberikan kesehatan sehingga akhirnya Skripsi ini dapat diselesaikan oleh penulis dengan baik, lancar, dan tepat waktu dalam menyelesaikannya.

Skripsi ini dilakukan guna memenuhi salah satu syarat kurikulum dalam menyelesaikan pendidikan pada Program Studi S1 Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi dengan judul Skripsi : **“Pembuatan Aplikasi Kriptografi File Menggunakan Algoritma Vigenere Cipher”**

Dalam kesempatan ini, penulis mengucapkan terima kasih yang sebesar-besarnya kepada banyak pihak yang telah membantu dalam penyelesaian penyusunan Skripsi ini.

Penulis ingin mengucapkan terima kasih kepada :

1. Untuk orang tua saya yang tercinta dan yang saya sayangi ibunda, biuda, Lola Monica serta keluarga yang selalu memberikan semangat, motivasi, do'a yang tiada henti-hentinya dan membantu dalam segi moril maupun materil sehingga penulis dapat menyelesaikan penyusunan Skripsi ini.
2. Rektor Universitas Pembangunan Panca Budi, Bapak Dr. H. Muhammad Isa Indrawan, S.E., M.M.
3. Rektor I, Bapak Ir. Bhakti Alamsyah, M.T., Ph.D.
4. Dekan Fakultas Sains dan Teknologi, Ibu Sri Shindi Indira, ST., M.Sc.
5. Ketua Program Studi Sistem Komputer, Bapak Dr. Muhammad Iqbal, S.Kom., M.Kom
6. Dosen Pembimbing I, Bapak Andysah Putera Utama Siahaan, S.Kom., M.Kom., Ph.D.
7. Dosen Pembimbing II, Bapak Radian Rahim, S.Kom., M.Kom.
8. Untuk sahabat penulis Billy Hardiyanti Ciu dan teman-teman seperjuangan Skripsi lainnya yang telah memberikan masukan, do'a, dukungan serta motivasi sehingga penulis dapat menyelesaikan Skripsi ini.

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

Medan, Agustus 2019
Penulis

(Yuli Aristia Br. Purba)
1514370165

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Matematika merupakan ilmu pengetahuan dasar yang sangat berpengaruh dalam perkembangan ilmu pengetahuan lainnya, termasuk perkembangan teknologi informasi. Perkembangan teknologi informasi memiliki pengaruh yang sangat signifikan bagi aspek kehidupan. Tidak terkecuali pada aspek komunikasi dan pengiriman pesan maupun pengiriman isi *file*. Masalah keamanan dan kerahasiaan isi *file* merupakan hal yang penting. Isi *file* yang bersifat rahasia tersebut perlu dibuatkan sistem penyimpanan dan pengirimannya agar tidak terbaca atau diubah oleh orang-orang yang tidak bertanggung jawab, baik itu saat isi *file* tersebut tersimpan didalam komputer maupun saat isi *file* dikirim melalui *e-mail*.

Untuk menyimpan isi *file* agar benar-benar aman, tentunya dilakukan sistem pengamanan yang baik, dan bebas dari jangkauan orang-orang yang tidak berhak, baik bebas dari jangkauan secara fisik maupun secara sistem. Apalagi jika isi *file* berada dalam suatu jaringan komputer yang terhubung dengan jaringan internet, tentu saja isi *file* tersebut tidak boleh diketahui dan diubah oleh orang yang tidak berhak.

Menurut *Request for Comments* (RFC), kriptografi merupakan ilmu matematika yang berhubungan dengan transformasi data untuk membuat artinya tidak dapat dipahami (untuk menyembunyikan maknanya),

mencegahnya dari perubahan tanpa izin, atau mencegahnya dari penggunaan yang tidak sah. Jika transformasinya dapat dikembalikan, kriptografi juga bisa diartikan sebagai proses mengubah kembali data yang terenkripsi menjadi bentuk yang dapat dipahami. Artinya, kriptografi dapat diartikan sebagai proses untuk melindungi data dalam arti yang luas (Oppliger, 2005). Dalam kriptografi terdapat dua konsep utama yaitu enkripsi dan dekripsi. Enkripsi merupakan proses penyandian pesan asli atau *plainteks* menjadi *cipherteks* (teks tersandi). Sedangkan dekripsi adalah proses penyandian kembali *cipherteks* menjadi *plainteks*.

Pada penelitian yang dilakukan oleh penulis yang berjudul “Pembuatan Aplikasi Kriptografi *File* Menggunakan Algoritma *Vigenere Cipher*” dijelaskan hasil penelitian menunjukkan bahwa isi *file* dapat terenkripsi menggunakan algoritma *Vigenere Cipher* yang bersifat sistem substitusi multi-alphabet, yaitu dengan sistem sandi *Ceasar* tetapi dengan pergeseran alphabet yang berlainan disesuaikan dengan kata kunci. Yang dimaksud sistem sandi substitusi merupakan penyandian dengan cara mengantikan huruf-huruf/teks aslinya dengan huruf-huruf sandi. Cara kerjanya yaitu isi *file* terlebih dahulu akan dienkripsi kemudian isi *file* akan dikirimkan melalui *e-mail*, ketika penerima sudah menerima isi *file* maka akan dideskripsikan sesuai dengan kunci, maka pihak ketiga atau orang yang tidak berhak tidak dapat melihat isi *file* tersebut.

Berdasarkan informasi yang telah dipaparkan, penulis membuat sebuah penerapan enkripsi dan dekripsi dengan menggunakan metode *Vigenere Cipher* dengan mengenkripsi dan mengdekripsi isi *file*. Cara kerja enkripsi dan dekripsi

ini akan dibuat secara mudah dan efektif. Implementasi ini menerapkan sistem enkripsi dan dekripsi menggunakan metode *Vigenere Cipher* simetris dalam pengamanan isi *file* mahasiswa Jurusan Sistem Komputer yang berjudul “**Pembuatan Aplikasi *File* Menggunakan Algoritma *Vigenere Cipher*”.**

1.2 Perumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan, maka rumusan masalah dalam penelitian ini sebagai berikut :

1. Bagaimana merancang sebuah *software* enkripsi dan dekripsi teks menggunakan algoritma *vigenere* sebagai pengaman informasi teks ?
2. Bagaimana membuat aplikasi enkripsi dan dekripsi berbasis *desktop* ?

1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah dipaparkan, maka batasan masalah dalam penelitian ini sebagai berikut :

1. Aplikasi yang dibangun hanya melakukan enkripsi dan dekripsi informasi *text*.
2. Perancangan aplikasi merupakan simulasi
3. Program yang digunakan dalam perancangan aplikasi ini adalah *visual basic .net 2015* menggunakan algoritma *vigenere cipher* dalam proses enkripsi dan dekripsi.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai penulis dalam perancangan aplikasi penerapan algoritma *vigenere* ini adalah :

- 1 Merancang aplikasi keamanan informasi *text* dengan menggunakan algoritma *vigenere cipher*.
- 2 Merancang sistem pengamanan informasi *text* dengan proses enkripsi dan dekripsi menggunakan metode algoritma *vigenere cipher*.

1.5 Manfaat Penelitian

Perancangan aplikasi penerapan algoritma *vigenere* ini bermanfaat bagi masyarakat luas antara lain :

1. Dengan menggunakan aplikasi ini seseorang dapat mengamankan suatu informasi tanpa takut diketahuin oleh orang lain.
2. Dapat digunakan dalam proses kerahasiaan data.
3. Proses pertukaran data atau informasi menjadi aman.

BAB II

LANDASAN TEORI

2.1 Pengertian Aplikasi

Menurut Widiyanto, Arifin dan Soebijono (2016) Aplikasi yang dijelaskan merupakan suatu perangkat lunak komputer yang memiliki fungsional tertentu sesuai dengan tujuan yang diinginkan oleh *programer*. Aplikasi diciptakan untuk mempermudah manusia dalam mengerjakan suatu tugas didalam sebuah komputer, seperti untuk pengolahan data maupun untuk keperluan editing.

Menurut Limbong dan Taufik (2017) Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemerosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemerosesan data yang diharapkan.

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*.

2.2 Pengertian Kriptografi

Secara umum kriptografi merupakan salah satu teknik pengamanan suatu data atau informasi yang rahasia atau pribadi yang dilakukan dengan cara mengolah *plaintext* dengan suatu kunci tertentu menggunakan suatu proses enkripsi, sehingga menghasilkan *chipertext* yang tidak dapat dipahami maknanya oleh orang lain. *Ciphertext* tersebut dikembalikan menjadi *plaintext* setelah melalui proses deskripsi pada algoritma.

Secara etimologi kriptografi berasal dari negara Yunani yang berarti *kryptos* yang bermakna tersembunyi dan *graphein* yang bermakna tulisan. Menurut Muhammad Khoiruddin Harahap (2016) Kriptografi adalah ilmu menulis pesan rahasia dengan tujuan menyembunyikan makna pesan tersebut.

Dari pernyataan diatas dapat disimpulkan bahwa kriptografi merupakan ilmu dan seni untuk menjaga kerahasiaan data atau informasi dengan cara menyembunyikan pesan asli agar tidak dapat dilihat dan dibaca oleh pihak yang tidak memiliki wewenang dalam informasi data tersebut.

A. Istilah-istilah dalam kriptografi antara lain sebagai berikut :

1. *Plaintext* merupakan pesan asli sebelum diubah menjadi pesan rahasia.
2. *Ciphertext* merupakan pesan sandi atau pesan rahasia yang sulit diterjemahkan.
3. *Key* merupakan kunci rahasia yang digunakan untuk mengubah pesan asli menjadi pesan rahasia.
4. Enkripsi merupakan proses mengubah *plaintext* menjadi *ciphertext*.
5. Dekripsi merupakan proses mengubah *ciphertext* menjadi *plaintext*.

a. Tujuan dari kriptografi dalam aspek keamanan informasi itu sendiri meliputi :

1) Kerahasiaan Data (*Confidentiality*)

Menjaga data agar tetap terahasia dari pihak-pihak yang tidak berwenang yang mungkin mencoba membaca data tersebut.

2) Integritas Data (*Integrity*)

Memastikan data yang dikirim masih tetap sama dengan data yang diterima tanpa ada perubahan atau modifikasi terhadap data tersebut.

3) Autentikasi (*Authentication*)

Memastikan bahwa pengirim dan penerima benar-benar terjamin keasliannya. Dua pihak yang berkomunikasi harus saling mengetahui satu dengan yang lainnya.

4) Non-Repudiasi (*Non-Repudiation*)

Pengirim tidak bisa menyangkal bahwa ia telah mengirim data, karena pengirim akan mendapatkan bukti kalau ia telah mengirim data kepada si penerima.

B. Jenis kriptografi berdasarkan perkembangan

Berdasarkan perkembangan jaman dari tahun ke tahun sejak pertama kali kriptografi ditemukan, ada dua jenis algoritma kriptografi antara lain sebagai berikut :

1. Kriptografi Klasik

Algoritma kriptografi yang termasuk kedalam jenis kriptografi klasik ini digunakan pada masa sebelum berlakunya komputerisasi dengan komputer, algoritma kriptografi ini rata-rata masih menggunakan kunci simetris dan menyandikan pesan dengan teknik substitusi atau transposisi.

2. Kriptografi Modern

Algoritma kriptografi yang termasuk jenis kedalam jenis kriptografi modern ini memiliki tingkat kesulitan yang lebih tinggi dan kompleks serta menggunakan pengetahuan matematika dalam penerapan kuncinya. Pada kriptografi modern ini kunci yang digunakan untuk menyandikan pesan berupa kunci asimetris.

2.3 Pengertian *File*

Menurut Dewi Intan Manullang (2018) *File* merupakan entitas dari sebuah data yang disimpan didalam sistem *file* yang dapat diakses dan diatur oleh pengguna. Sebuah *file* memiliki nama yang unik dalam direktori dimana ia berada. Alamat direktori dimana suatu berkas ditempatkan diistilahkan dengan *path*. Sebuah *file* berisi aliran data (atau data *stream*) yang berisi sekumpulan data yang disebut dengan *properties* yang berisi informasi mengenai *file* yang bersangkutan seperti informasi mengenai kapan sebuah berkas dibuat.

File merupakan kumpulan dari data dan informasi yang saling berhubungan juga tersimpan didalam ruang penyimpanan komputer. *File* adalah kumpulan dokumen yang berisi informasi tertentu dan dapat dibuka dengan menggunakan program komputer.

2.4 Pengertian Teks

Menurut Dewi Intan Manullang (2018) Teks adalah permukaan fenomena karya sastra. Teks adalah kata-kata yang membentuk karya dan yang disusun dengan cara sedemikian rupa untuk membelokkan arti yang tetap dan seunik mungkin. Karena teks merupakan tenunan yang dijalin, teks sebagai sebuah jaringan, yang secara konstitutif berhubungan dengan tulisan, maka teks mempunyai fungsi menjaga tetapnya dan permanennya inkripsi yang ditulis agar ingatan terbantu.

2.5 Kode ASCII

Menurut Kharisma, dan Rachman (2017) ASCII (*American Standard Code for Information Interchange*) atau kode Standar Amerika untuk Pertukaran Informasi. Merupakan suatu standar internasional dalam kode huruf dan lumeri seperti Hex dan Unicode tetapi ASCII lebih bersifat universal. Dalam kriptografi, kode ASCII ini merupakan urutan bit yang akan mewakili teks asli yang kemudian dienkrpsi untuk mendapatkan teks kode dalam bentuk urutan bit.

Menurut Imam Marzuki (2018) Kode ASCII sebenarnya memiliki komposisi bilangan biner sebesar 7 bit, namun ASCII disimpan sebagai 8 bit dengan menambahkan nilai 0 sebagai nilai signifikan paling tinggi. Encoding pada ASCII menggunakan 3 tipe bilangan bulat yaitu decimal (2^2), hexadecimal (2^{16}), dan oktadecimal (2^8).

Jumlah kode ASCII adalah 255 kode, kode ASCII 0..127 merupakan kode ASCII untuk manipulasi teks, sedangkan kode ASCII 128..255 merupakan kode ASCII untuk manipulasi grafik.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

Gambar 2.1 Tabel Kode ASCII

Sumber : <http://jti.respati.ac.id/index.php/jurnalijti/articel/view/176>

DOS	WIN	Dec	Hex	DOS	WIN	Dec	Hex	DOS	WIN	Dec	Hex	DOS	WIN	Dec	Hex
Ç	€	128	80	á		160	A0	Ł	À	192	C0	α	à	224	E0
ü		129	81	í	ı	161	A1	Ł	Á	193	C1	β	á	225	E1
é	‚	130	82	ó	¢	162	A2	┘	Â	194	C2	Γ	â	226	E2
â	ƒ	131	83	ú	£	163	A3	┘	Ã	195	C3	π	ã	227	E3
ä	„	132	84	ñ	¤	164	A4	—	Ä	196	C4	Σ	ä	228	E4
à	…	133	85	Ñ	¥	165	A5	┘	Å	197	C5	σ	å	229	E5
á	†	134	86	ª	¦	166	A6	┘	Æ	198	C6	μ	æ	230	E6
ç	‡	135	87	º	§	167	A7	┘	Ç	199	C7	τ	ç	231	E7
ê	ˆ	136	88	¿	¨	168	A8	┘	È	200	C8	Φ	è	232	E8
ë	‰	137	89	¬	©	169	A9	┘	É	201	C9	Θ	é	233	E9
è	Š	138	8A	¬	ª	170	AA	┘	Ê	202	CA	Ω	ê	234	EA
ı	‹	139	8B	½	«	171	AB	┘	Ë	203	CB	δ	ë	235	EB
î	Œ	140	8C	¼	¬	172	AC	┘	Ì	204	CC	∞	ì	236	EC
ï		141	8D	ı	-	173	AD	=	Í	205	CD	ø	í	237	ED
Ä	Ž	142	8E	«	®	174	AE	┘	Î	206	CE	ε	î	238	EE
Á		143	8F	»	—	175	AF	┘	Ï	207	CF	∩	ï	239	EF
É		144	90	◊	°	176	B0	┘	Ð	208	D0	∩	ð	240	F0
æ	‚	145	91	◊	±	177	B1	┘	Ñ	209	D1	±	ñ	241	F1
Æ	‚	146	92	◊	²	178	B2	┘	Ò	210	D2	≥	ò	242	F2
ô	“	147	93	◊	³	179	B3	┘	Ó	211	D3	≤	ó	243	F3
ö	”	148	94	┘	´	180	B4	┘	Ô	212	D4		ô	244	F4
ò	•	149	95	┘	μ	181	B5	┘	Õ	213	D5		õ	245	F5
û	-	150	96	┘	¶	182	B6	┘	Ö	214	D6	+	ö	246	F6
ù	—	151	97	┘	·	183	B7	┘	×	215	D7	≈	÷	247	F7
y	~	152	98	┘	,	184	B8	┘	Ø	216	D8	°	ø	248	F8
Ö	™	153	99	┘	ı	185	B9	┘	Ù	217	D9	•	ù	249	F9
Û	š	154	9A	┘	°	186	BA	┘	Ú	218	DA	·	ú	250	FA
¢	›	155	9B	┘	»	187	BB	■	Û	219	DB	√	û	251	FB
£	œ	156	9C	┘	¼	188	BC	■	Ü	220	DC	“	ü	252	FC
¥		157	9D	┘	½	189	BD	■	Ý	221	DD	²	ý	253	FD
Ps	ž	158	9E	┘	¾	190	BE	■	Þ	222	DE	■	þ	254	FE
f	ÿ	159	9F	┘	¿	191	BF	■	ß	223	DF	■	ÿ	255	FF

Gambar 2.2 Tabel Kode ASCII

Sumber : <http://jti.respati.ac.id/index.php/jurnalijti/articel/view/176>

2.6 Algoritma Kriptografi

Algoritma kriptografi merupakan langkah-langkah logis bagaimana caranya menyembunyikan pesan dari orang yang tidak memiliki wewenang atas pesan tersebut dengan melakukan pembangkitan kunci, enkripsi dan dekripsi.

Algoritma kriptografi terdiri dari 3 fungsi dasar meliputi :

- a. Kunci yang dimaksud disini adalah kunci yang dipakai untuk enkripsi dan dekripsi, kunci terbagi dua bagian yaitu kunci publik (*public key*) dan kunci privat (*private key*).
- b. Enkripsi merupakan hal yang paling 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 tidak mengerti akan sebuah kata, maka akan melihatnya didalam kamus atau daftar istilah-istilah. Beda halnya dengan enkripsi, untuk mengubah *plaintext* ke bentuk *ciphertext* dapat menggunakan algoritma yang mengkodekan data yang diinginkan.
- c. Dekripsi merupakan kebalikan dari enkripsi, pesan yang telah dienkripsi dibalikan ke bentuk asalnya *plaintext* disebut dengan dekripsi pesan.

Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara kedua himpunan tersebut. Misalkan P menyatakan *Plaintext* dan C menyatakan *Ciphertext*, maka fungsi Enkripsi E menyatakan P ke C :

$$E(P) = C \dots\dots\dots(1)$$

Dan fungsi dekripsi D menyatakan C ke P :

$$D(C) = P \dots\dots\dots(2)$$

Karena proses enkripsi kemudian dekripsi mengembalikan pesan sandi ke pesan asal, maka persamaan berikut harus benar,

$$D(E(P)) = P \dots\dots\dots(3)$$

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.

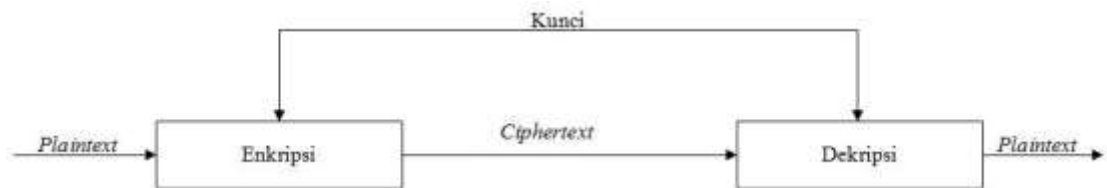
2.6.1 Algoritma Simetris

Menurut Muhammad Dedi Irawan (2017) Algoritma simetris atau sering disebut algoritma kriptografi konvensional adalah algoritma yang menggunakan kunci yang sama untuk proses enkripsi dan proses dekripsi.

Mengasumsikan pengirim dan penerima pesan sudah berbagi kunci yang sama sebelum bertukar pesan. Keamanan sistem kriptografi simetris terletak pada kerahasiaan kuncinya. Kriptografi simetris merupakan satu-satunya jenis kriptografi yang dikenal dalam catatan sejarah hingga tahun 1976. Semua algoritma kriptografi klasik termasuk kedalam sistem kriptografi simetris.

Kelebihan algoritma simetris ini adalah proses enkripsi dan dekripsi yang jauh lebih cepat dibandingkan dengan algoritma asimetris. Sedangkan kelemahan algoritma simetris ini adalah permasalahan distribusi kunci (*distribution key*). Seperti yang telah dibahas, proses enkripsi dan dekripsi menggunakan kunci yang

sama. Sehingga muncul persoalan menjaga kerahasiaan kunci, yaitu pada saat pengiriman kunci pada media yang tidak aman seperti internet. Tentunya jika kunci ini sampai hilang atau sudah dapat oleh orang lain. Berikut merupakan gambar enkripsi dan dekripsi algoritma simetris.

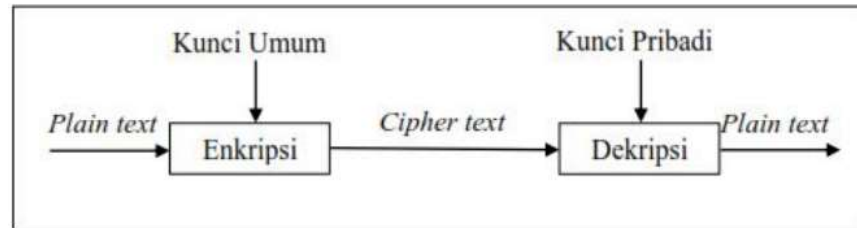


Gambar 2.3 Kriptografi Simetris

2.6.2 Algoritma Asimetris

Menurut Muhammad Dedi Irawan (2017) Algoritma asimetris adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsi. Dimana kunci enkripsi dapat disebarluaskan kepada umum dan dinamakan sebagai kunci publik (*public key*), sedangkan kunci dekripsi disimpan untuk digunakan sendiri dan dinamakan sebagai kunci private (*private key*). Oleh karena itu, kriptografi ini dikenal pula dengan nama kriptografi kunci publik (*public key cryptography*). Adapun pada kriptografi asimetris, dimana setiap pelaku sistem informasi akan memiliki sepasang kunci, yaitu kunci publik dan kunci pribadi, dimana kunci publik di distribusikan kepada umum sedangkan kunci pribadi disimpan untuk diri sendiri. Artinya bila A mengirimkan sebuah pesan kepada B, A dapat menyandikan pesannya dengan menggunakan kunci publik B, dan B ingin membaca pesan tersebut, ia perlu mendekripsikan surat itu dengan kunci

privatnya. Dengan demikian kedua belah pihak dapat menjamin asal surat serta keaslian surat tersebut.



Gambar 2.4 Kriptografi Asimetris

2.7 Algoritma *Vigenere Cipher*

Menurut Anas, Nanda dan Hidayat (2018) Algoritma *Vigenere cipher* dipublikasikan oleh diplomat Prancis, *blaine de vigenere* pada abad 16 tahun 1586. *Vigenere cipher* sangat dikenal karena mudah dipahami dan diimplementasikan. *Cipher* menggunakan bujursangkar *vigenere* untuk melakukan enkripsi. Bujursangkar *vigenere* digunakan untuk memperoleh *ciphertext* dengan menggunakan kunci yang sudah ditentukan, jika panjang kunci lebih pendek daripada panjang *plaintext* maka kunci diulang penggunaannya.

Vigenere Cipher merupakan jenis *cipher* abjad majemuk yang paling sederhana. *Vigenere Cipher* menerapkan metode substitusi poli alfabetik dan termasuk kedalam kategori kunci simetris dimana kunci yang digunakan untuk proses enkripsi adalah sama dengan kunci yang digunakan untuk proses dekripsi. Tujuan utama dari *vigenere cipher* ini adalah menyembunyikan keterhubungan antara *plaintext* dan *ciphertext* dengan menggunakan kata kunci sebagai penentu pergeseran karakternya.

Kekuatan algoritma *vigenere cipher* ini dapat mencegah frekuensi karakter-karakter di dalam *ciphertext* yang memiliki pola tertentu yang sama, seperti yang terjadi pada *cipher* abjad tunggal. Karakter yang paling sering muncul pada *cipher* abjad tunggal dalam *ciphertext* merupakan substitusi dari karakter yang paling sering muncul di *plaintext*. Akibatnya, kriptanalis bisa dengan mudah menebak karakter tersebut dengan teknik analisis. Namun, pada *vigenere cipher* hal tersebut tidak bisa dilakukan karena satu macam karakter pada *plaintext* mungkin dienkripsi menjadi beberapa bagian karakter pada *ciphertext*

Meskipun dapat dikatakan bahwa algoritma *Vigenere cipher* lebih kuat dibanding algoritma *Caesar cipher*, algoritma ini tetap memiliki kelemahan sehingga *ciphertext* hasil dari algoritma *vigenere cipher* ini dapat dibuka secara paksa oleh kriptanalis. Kelemahan ini muncul jika panjang kunci lebih pendek dari panjang *plaintext*-nya sehingga terdapat perulangan kunci yang digunakan untuk mengenkripsi *plaintext* tersebut.

Berikut ini rumus enkripsi dan dekripsi *Vigenere Cipher* :

$$\text{Enkripsi : } C_i = P_i + k_i \text{ mod } 255 \dots\dots\dots(4)$$

$$\text{Dekripsi : } P_i = C_i - k_i \text{ mod } 255 \dots\dots\dots(5)$$

C_i : *Ciphertext*

P_i : *Plaintext*

k_i : *Key* atau kunci

2.8 Bahasa Pemrograman

Bahasa pemrograman adalah perintah-perintah atau intruksi yang dimengerti oleh komputer untuk melakukan tugas tertentu bahasa pemrograman merupakan sebuah intruksi untuk memerintah komputer agar bisa menjalankan fungsi tertentu namun hanya intruksi standar saja. Bahasa pemrograman juga memiliki perhimpunan dari aturan sintaks dan semantik yang tugasnya untuk mendefinisikan program komputer. Bahasa pemrograman yang kita kenal antara lain adalah *Java*, *Visual Basic*, *C ++*, *PHP*, dan bahasa pemrograman lainnya. Namun tentu saja kebutuhan bahasa pemrograman ini harus disesuaikan dengan fungsi dan perangkat yang menggunakannya.

Menurut generasi bahasa pemrograman digolongkan menjadi 4 generasi, meliputi :

- a. Generasi ke-1: *machine language*
- b. Generasi ke-2: *assembly language: Assembler*
- c. Generasi ke-3: *high level programming language*, contoh *c* dan *Pascal*
- d. Generasi ke-4: *4 GL (fourth-generation language)*, contoh *SQL*
- e. Generasi ke-5: *Programming Language Based Object Oriented & Web Development*

Secara umum bahasa pemrograman terdiri dari 4 kelompok meliputi :

- a. *Object Oriented Language* : Seperti bahasa *Visual C*, *Delphi*, *Visual dBase*, *Visual FoxPro*.
- b. *Low Level Language* : Bahasa *Assambly*.
- c. *Middle Level Language* : Bahasa *C*.

d. *High Level Language* : Bahasa *Basic*, dan *Pascal*.

Menurut tingkat kedekatannya dengan mesin komputer, bahasa pemrograman terdiri dari :

- a. Bahasa Mesin, yaitu memberikan perintah kepada komputer dengan memakai kode bahasa biner, contohnya 01100101100110.
- b. Bahasa Tingkat Rendah, atau dikenal dengan istilah bahasa rakitan (bahasa Inggris *Assembly*), yaitu memberikan perintah kepada komputer dengan memakai kode-kode singkat (kode *mnemonic*), contohnya *MOV*, *SUB*, *CMP*, *JMP*, *JGE*, *JL*, *LOOP*, dan lain sebagainya.
- c. Bahasa Tingkat Menengah, yaitu bahasa komputer yang memakai campuran intruksi dalam kata-kata bahasa manusia (lihat Bahasa Tingkat Tinggi dibawah) dan intruksi yang bersifat simbolik, contohnya {, }, ?, <<, >>, &&.
- d. Bahasa Tingkat Tinggi, yaitu bahasa komputer yang memakai intruksi berasal dari unsur kata-kata bahasa manusia, contohnya *begin*, *end*, *if*, *for*, *while*, *and*, *or*, dan lain sebagainya. Komputer dapat mengerti bahasa manusia itu diperlukan dalam program *compiler* atau *interpreter*.

Fungsi dari bahasa pemrograman adalah untuk memerintahkan sebuah komputer agar dapat mengolah data yang sesuai dengan yang diinginkan. *Output* dari bahasa pemrograman ini adalah berupa aplikasi ataupun program khusus. Contohnya lampu lalu lintas di jalan raya.

2.9 *Visual Basic.Net*

Menurut Hadi dan Samad (2019) *Visual basic* merupakan sebuah bahasa pemrograman yang menawarkan *Intergrated Development Environment* (IDE) visual untuk membuat program perangkat lunak berbasis operasi *Microsoft Windows* menggunakan model pemrograman (COM).

Menurut Fernando, Siswanto, dan Suryana (2014) *Platform Microsoft.Net* merupakan model untuk *development* dimana *platform* dan aplikasi bisa dibuat dan dijalankan tanpa bergantung pada alat (*device*) yang dipakai. Teknologi ini memungkinkan beberapa aplikasi bekerja sama. *Visual Basic.Net* merupakan *core* dari pembuatan aplikasi berbasis *.Net* yang merupakan lingkungan pemrograman yang mempermudah tahapan desain, *development*, *debugging*, dan *deployment* dari aplikasi berbasis *.Net* dan *XML web service*, serta meningkatkan efisiensi *developer* dengan menyediakan lingkungan pemrograman yang sudah biasa digunakan.

Bahasa *Basic* pada dasarnya merupakan bahasa yang mudah untuk dimengerti sehingga pemrograman menggunakan bahasa *Basic* dapat dilakukan dengan mudah sekalipun yang melakukannya adalah seorang pemula. Dalam *Microsoft Visual Basic.Net* terdapat dua komponen utama yaitu :

A. *Net Framework Class Library*

Komponen ini digunakan untuk menjalankan sebuah aplikasi melalui objek yang telah didefinisikan antara lain : *label*, *form*, *textbox*, *button*, *listbox*, *datetimepicker* dan lain-lain.

B. *Common Language Runtime (CLR)*

Komponen ini digunakan untuk mengeksekusi program yang ditulis dalam bahasa pemrograman yang ada dalam lingkungan *Microsoft Visual Studio.Net*, seperti : *C.Net*, *C++*, *.Net*, dan juga *Visual Basic.Net*.

Adapun kelebihan dan kekurangan dari *Visual Basic.Net* yaitu :

1. Kelebihan *Visual Basic*

- a. *VB.Net* mempunyai fasilitas *Real Time Background Compiler* yaitu sebagai penanganan dalam *error* atau *bug*.
- b. Lebih cepat dalam pembuatan aplikasi berbasis desktop
- c. Menyediakan untuk *developer* pemrograman data akses *ActiveX Data Object (ADO)*.

2. Kekurangan *Visual basic*

- a. Untuk versi *VB.Net* 2010 dan seterusnya tidak mempunyai komponen *Crystal Report* karena sudah terpisah.
- b. Harus ada *Net framework* agar aplikasi bisa berjalan.
- c. Tidak mempunyai database sendiri.
- d. Memerlukan kapasitas yang besar untuk instalasi *VB.Net*.

2.10 FlowChart


Flowchart atau bagan alir, awal mulanya memang berkembang dari komputer yaitu untuk menggambarkan urutan proses penyelesaian masalah. *Flowchart* atau bagan alir merupakan bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.



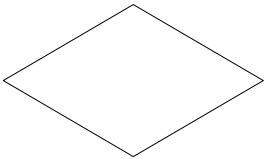
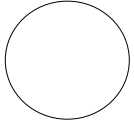
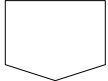
Menurut Hadi dan Samad (2019) *Flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma.





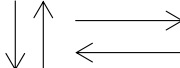
Flowchart sangat berguna khususnya untuk menjelaskan urutan proses yang pelaksanaannya memiliki banyak option pilihan atau percabangan. Menggambar flowchart, memerlukan simbol-simbol yang berbentuk seperti persegi, dan belah ketupat maupun dengan bentuk lain yang kemudian dihubungkan dengan garis-garis yang berarah (garis yang dengan menggunakan simbol anak panah).

Simbol yang digunakan dalam menggambarkan algoritma dalam bentuk diagram alir dan kegunaan dari simbol tersebut adalah sebagai berikut.

Tabel 2.1 Simbol *Flowchart*

No	Simbol	Nama	Fungsi
1		Terminal	Menyatakan permulaan atau akhir suatu program.

2		<i>Input / Output</i>	Menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya.
3		<i>Process</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
4		<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya/tidak.
5		<i>Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
6		<i>Offline Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.

7		<i>Predefined Process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
8		<i>Punched Card</i>	Menyatakan input berasal dari kartu atau <i>output</i> ditulis ke kartu.
9		<i>Punch Tape</i>	
10		<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui printer).
11		<i>Flow</i>	Menyatakan jalannya arus suatu proses.

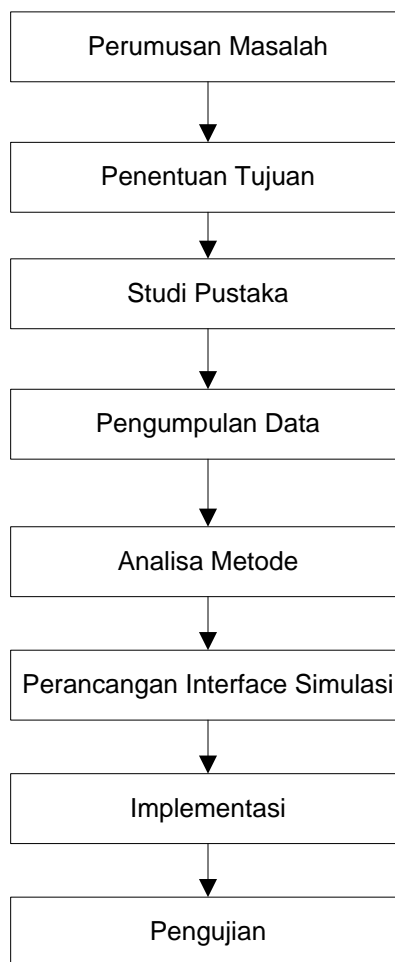
Sumber : <http://j-ilkominfo.org/index.php/ejournalaikom/article/view/15>

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Adapun tahapan penelitian yang dilakukan oleh penulis ini dengan judul Pembuatan Aplikasi *File* Menggunakan Algoritma *Vigenere Cipher* adalah sebagai berikut:



Gambar 3.1 Tahapan Penelitian

3.2 Metode Pengumpulan Data

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

1. Pengamatan (*Observation*)

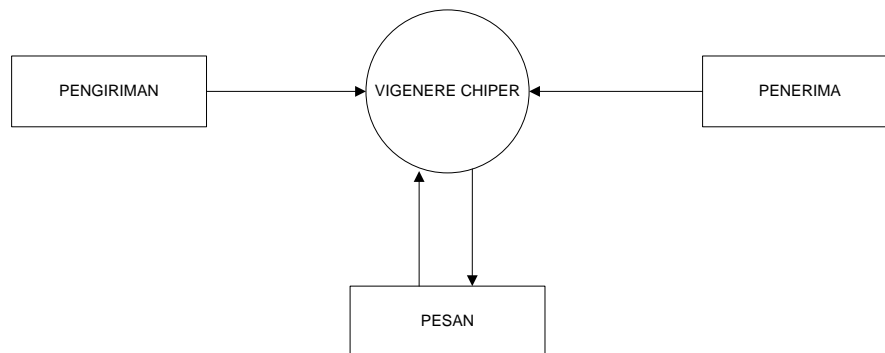
Penulis melakukan pengamatan langsung pada setiap jenis-jenis pengamanan isi *file* untuk menentukan keamanan data yang sesuai untuk dapat mengamankan isi *file*.

2. Penelitian Kepustakaan (*Library Research*)

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

3.3 Analisis Sistem Yang Sedang Berjalan

Pertukaran data dalam hal ini pesan rahasia berbentuk teks dengan menggunakan metode tradisional yaitu dengan cara bertukar kata kunci tunggal. Diagram dibawah adalah penggambaran bagaimana pertukaran pesan rahasia menggunakan kunci tunggal terjadi.



Gambar 3.2 Skema Pengiriman Pesan

Pemberitahuan kata kunci dari pengirim ke penerima menggunakan media yang umum digunakan oleh banyak orang.

3.3.1 Analisa Kelemahan yang Berjalan

1. Penggunaan kata kunci tunggal berpotensi terjadinya salah pemahaman. Dalam hal ini kemungkinan penerima salah mengartikan kunci yang diberikan oleh pengirim adalah hal yang dapat terjadi.
2. Pemberitahuan atau pertukaran kata kunci yang dikirimkan oleh pengirim ke penerima memiliki potensi dapat diketahui oleh orang lain sehingga pesan rahasia dapat terbongkar.

3.3.2 Solusi Pemecahan Masalah

Pemecahan masalah yang penulis lakukan adalah dengan melakukan penerapan metode ini yang didalamnya terdapat Algoritma *Vigenere Cipher*. Penggunaan metode ini dapat digunakan sebagai solusi agar pengirim dan penerima menetapkan kunci untuk membuka pesan yang dikirim oleh pengirim.

Tabel 3.1 Perencanaan Rancangan

No	Sistem yang Berjalan	Sistem yang Diusulkan	Hasil yang Ingin Dicapai
1.	Penggunaan kunci tunggal yang harus diketahui oleh pengirim dan penerima untuk membuka pesan.	Pengirim dan penerima menetapkan kunci untuk membuka pesan.	Tidak ada lagi kesalahan pemahaman atau salah tafsir kunci tunggal karena pengirim dan penerima memiliki kunci yang telah ditetapkan.
2.	Pertukaran kunci tunggal menggunakan media komunikasi yang rentan untuk dapat diketahui orang lain.	Pengirim dan penerima menentukan kunci yang ingin digunakan untuk membuka pesan.	Kemungkinan bocornya kunci saat proses pertukaran informasi kunci tunggal dapat dihindari.

3.4 Rancangan Penelitian

Visual basic 2010 akan menjadi sarana untuk menciptakan perangkat lunak ini. Pada analisa proses ini penggunaan digunakan sebagai metode yang didalamnya terdapat kombinasi dari algoritma *Vigenere Cipher*. Algoritma *Vigenere Cipher* digunakan oleh pengirim untuk mengenkripsi pesan yang akan dikirimkan..

Perhitungan secara matematis dilakukan sebagai penggambaran proses yang akan terjadi pada metode ini yang didalamnya terdapat algoritma *Vigenere Cipher*. Berikut tahapannya:

1. Proses Enkripsi Pesan Asli oleh Pengirim

Tahap ini dilakukan dengan menggunakan Algoritma *Vigenere Cipher* yang akan digunakan untuk meng-enkripsi pesan asli (*plaintext*) pengirim.

Diketahui *Plaintext* “SELAMAT DATANG” dengan kunci “KAMPUS”. Maka untuk mendapatkan *ciphertextnya* harus menggunakan penghitungan seperti di bawah ini:

Langkah Pertama membuat tabel konversi ASCII.

Plaintext : SELAMAT DATANG

Kunci : KAMPUS

Penerima memilih kata KAMPUS sebagai kunci yang akan ia gunakan untuk melakukan proses enkripsi menggunakan Algoritma *Vigenere Cipher*, sehingga pada prosesnya kata KAMPUS akan mengikuti banyak karakter *ciphertext 1* yang didapat.

Ciphertext : SELAMAT DATANG

Kunci : KAMPUS

Selanjutnya akan di enkripsi dengan formula Algoritma *Vigenere Cipher* yaitu:

$$C = P + K \text{ mod } 255 - 1$$

Dalam hal ini *plaintext* adalah *ciphertext* 1 yang didapat.

$$\begin{aligned} C1 &= S + K \text{ mod } 255 \\ &= 83 + 75 \text{ mod } 255 \\ &= 158 = \checkmark \end{aligned}$$

$$\begin{aligned} C2 &= E + A \text{ mod } 255 \\ &= 69 + 65 \text{ mod } 255 \\ &= 134 = \dagger \end{aligned}$$

$$\begin{aligned} C3 &= L + M \text{ mod } 255 \\ &= 76 + 77 \text{ mod } 255 \\ &= 153 = \text{TM} \end{aligned}$$

$$\begin{aligned} C4 &= A + P \text{ mod } 255 \\ &= 65 + 80 \text{ mod } 255 \\ &= 145 = \text{'} \end{aligned}$$

$$\begin{aligned} C5 &= M + U \text{ mod } 255 \\ &= 77 + 85 \text{ mod } 255 \\ &= 162 = \text{¢} \end{aligned}$$

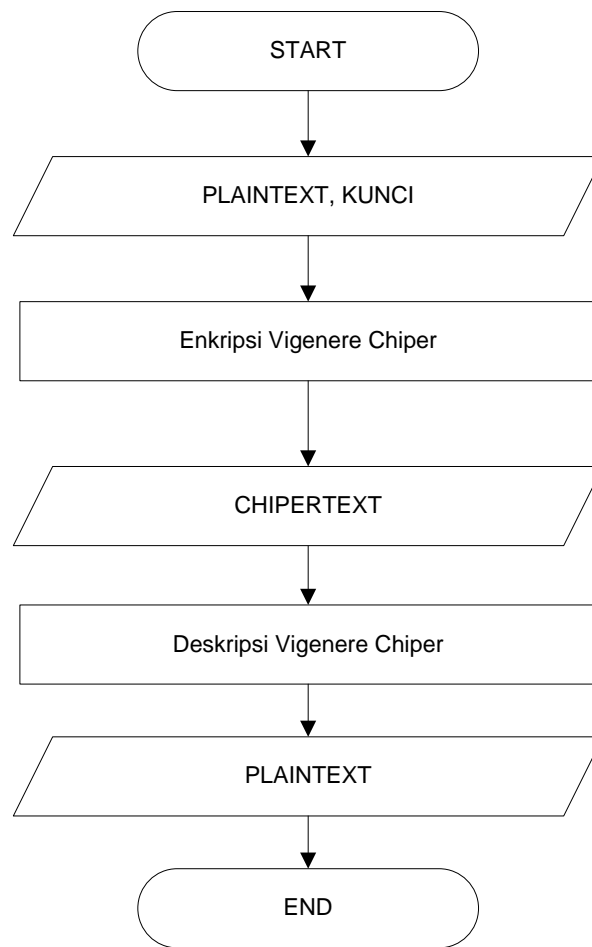
$$\begin{aligned} C6 &= A + S \text{ mod } 255 \\ &= 65 + 83 \text{ mod } 255 \\ &= 148 = \text{'"} \end{aligned}$$

3.4.1 *Flowchart* Sistem

Flowchart merupakan langkah awal pembuatan program. Dengan adanya *flowchart* urutan proses kegiatan menjadi lebih jelas. Bila terdapat penambahan proses maka dapat dilakukan lebih mudah. Setelah *flowchart* selesai disusun, selanjutnya pemrogram (*programmer*) menerjemahkannya ke bentuk program dengan bahasa pemrograman.

Flowchart merupakan urutan-urutan langkah kerja suatu proses yang digambarkan dengan menggunakan simbol-simbol yang disusun secara sistematis.

Flowchart Vigenere Cipher yang digunakan oleh pengirim untuk mengenkripsi dan mendeskripsi *plaintext* hingga mendapatkan *ciphertext* digambarkan sebagai berikut:

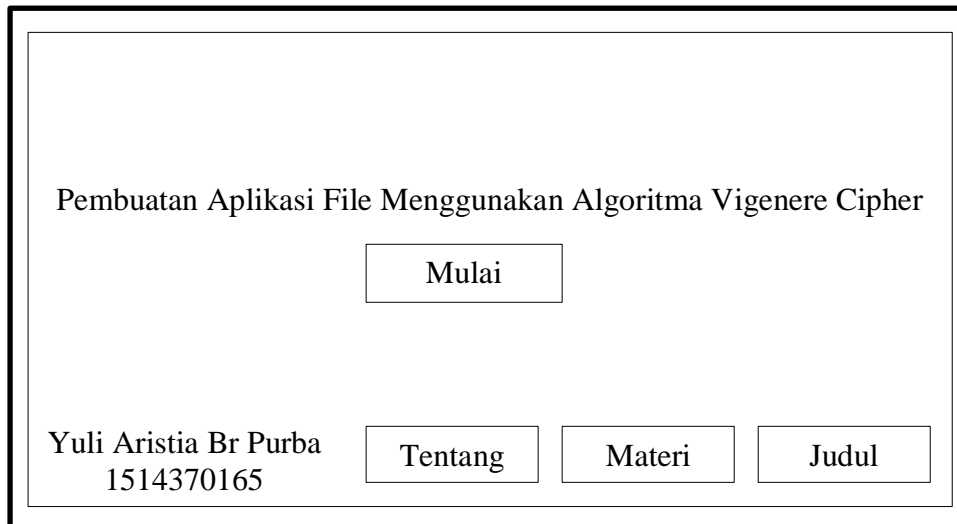


Gambar 3.3 *Flowchart Vigenere Cipher*

3.4.2 Perancangan Antarmuka

1. Rancangan Tampilan Awal/*Home*

Form ini berisi tombol-tombol seperti menu Mulai, Tentang, Judul.



Gambar 3.4 Rancangan Tampilan Awal/*Home*

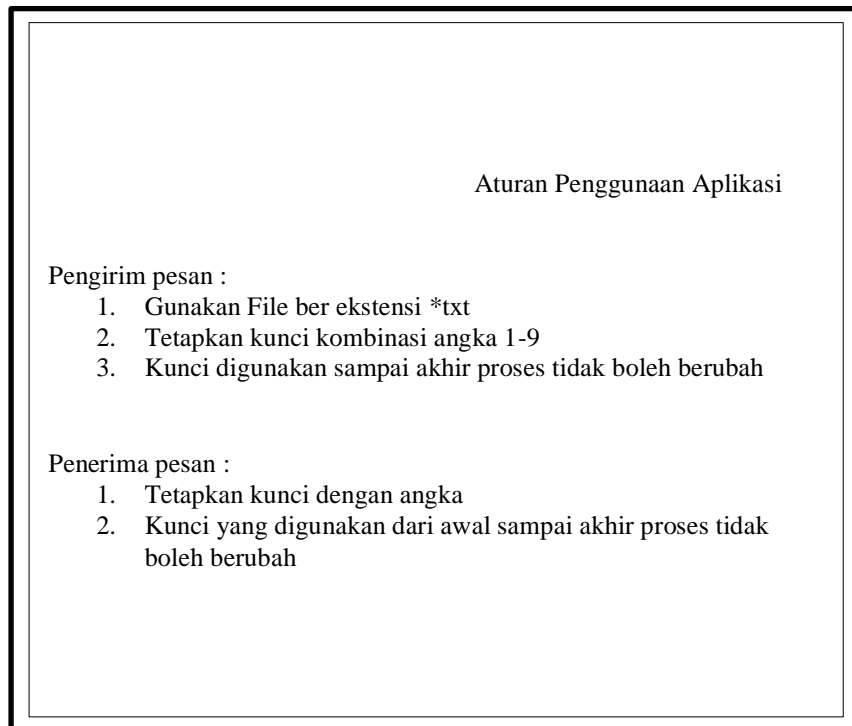
Pada tampilan di atas terdapat 4 tombol yaitu Mulai, Tentang, Materi, Judul.

- Tombol Mulai berfungsi untuk menghubungkan pengguna ke form mulai.
- Tombol Tentang berfungsi untuk menghubungkan pengguna ke form tentang.
- Tombol Materi berfungsi untuk menghubungkan pengguna ke form materi.
- Tombol Judul berfungsi untuk menghubungkan pengguna ke form judul.

2. Rancangan Tampilan Halaman Tentang

Form ini berisikan tentang tata cara penggunaan aplikasi yang dijalankan.

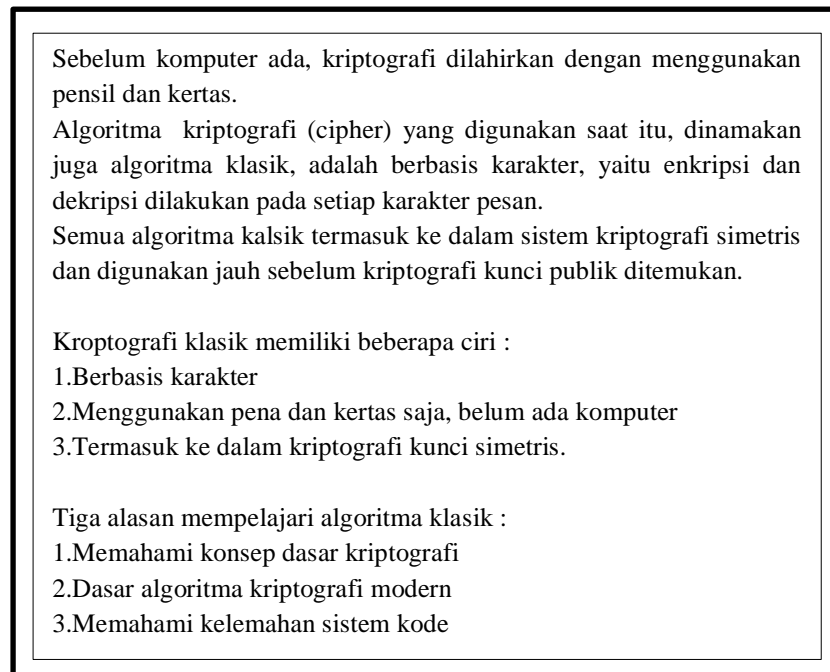
Pada halaman tersebut dijelaskan apa-apa saja yang menjadi kewajiban bagi pengirim dan penerima saat menjalankan simulasi algoritma *vigenere*.



Gambar 3.5 Rancangan Tampilan Halaman Tentang

3. Rancangan Tampilan Halaman Materi

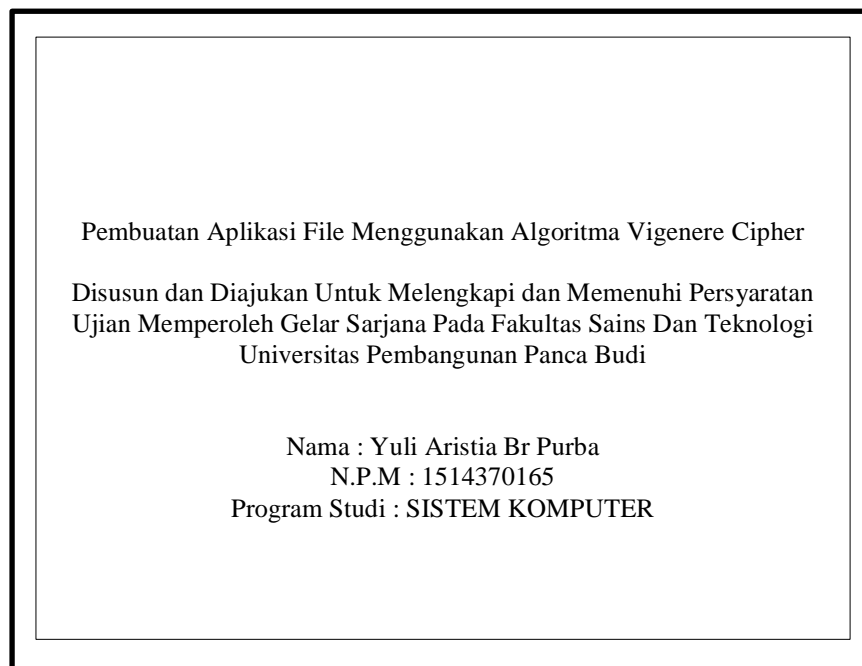
Form ini digunakan untuk menjelaskan apa-apa saja yang menjadi kewajiban bagi pengirim dan penerima saat menjalankan simulasi algoritma *vigenere cipher*.



Gambar 3.6 Rancangan Tampilan Halaman Materi

4. Rancangan Tampilan Halaman Judul

Berisi penjelasan mengenai biodata penulis. Isi dari form judul ini adalah berisikan data dari penulis yang ada mengangkat judul ini.



Pembuatan Aplikasi File Menggunakan Algoritma Vigenere Cipher

Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Persyaratan
Ujian Memperoleh Gelar Sarjana Pada Fakultas Sains Dan Teknologi
Universitas Pembangunan Panca Budi

Nama : Yuli Aristia Br Purba
N.P.M : 1514370165
Program Studi : SISTEM KOMPUTER

Gambar 3.7 Rancangan Tampilan Judul

5. Rancangan Tampilan Halaman Utama Algoritma *Vigenere*

Form ini menampilkan tampilan halaman utama pada aplikasi algoritma *vigenere cipher*.

The wireframe shows a main window with a title bar. At the top, there is a search bar with a 'Cari' button, and three buttons: 'Peraturan', 'About', and 'Tutup'. Below this is a 'Baca File' button. A large empty text area is followed by a 'Kerjakan' button. The interface is divided into two main sections: 'Pengirim' (Sender) and 'Penerima' (Receiver). The 'Pengirim' section is titled 'Enkripsi Vigenere Cipher' and contains a 'Plaintext' input field, a 'Kunci' (Key) input field, an 'Enkripsi' button, and a 'Ciphertext' output field. The 'Penerima' section is titled 'Dekripsi Vigenere Cipher' and contains a 'Ciphertext' input field, a 'Kunci' (Key) input field, a 'Dekripsi' button, and a 'Plaintext' output field. At the bottom, there are two email input fields: 'Email Pengirim' and 'Email Penerima', followed by a 'Kirim Email' button. A 'Clear All' button is located to the right of the 'Email Penerima' field.

Gambar 3.8 Rancangan Tampilan Halaman Utama Algoritma *Vigenere*

BAB IV

HASIL DAN PEMBAHASAN

4.1 Kebutuhan Spesifikasi Minimum *Hardware* dan *Software*

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

4.4.1 Spesifikasi Sistem

Analisis kebutuhan sistem merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen – komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

A. Analisis Perangkat Keras (*Hardware*)

Perangkat keras minimum yang digunakan untuk membangun Sistem Informasi Penjualan ini adalah :

1. *Processor* Berkecepatan 3.0 Ghz
2. RAM 2 Gb
3. *Hardisk* minimal 10 Gb untuk menyimpan data
4. LAN *Card*
5. *Keyboard* dan *Mouse*
6. Monitor 14 *inch*.

B. Analisis Perangkat Lunak (*Software*)

Untuk mendukung dalam penyimpanan informasi, dibutuhkan suatu fasilitas yang memadai. Yaitu berupa perangkat lunak (*software*) yang dirancang untuk memudahkan dalam pembangunan dan menjalankan sisten nantinya. Adapun perangkat lunak yang digunakan adalah sebagai berikut :

1. *Microsoft Windows 10* , *Windows 10* sebagai sistem operasi
2. *Visual Studio 2010*, Sebagai Perancangan Program Aplikasi.

4.2 Pengujian Aplikasi dan Pembahasan

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

Pengujian dilakukan dengan memasukkan karakter atau huruf dari *file* berformat *.txt* selanjutnya diproses oleh aplikasi apakah aplikasi tersebut dapat memberikan hasil yang sesuai. Proses yang akan dilakukan pengujian dalam aplikasi

ini adalah simulasi pengiriman pesan dengan menggunakan metode algoritma *vigenere* antara pengirim kepada penerima dengan kunci yang dimiliki masing-masing pihak tanpa perlu bertukar kunci tunggal hingga pada akhirnya pesan asli yang dikirimkan oleh pengirim dapat dibaca oleh penerima .

A. Tampilan Awal/ *Home*

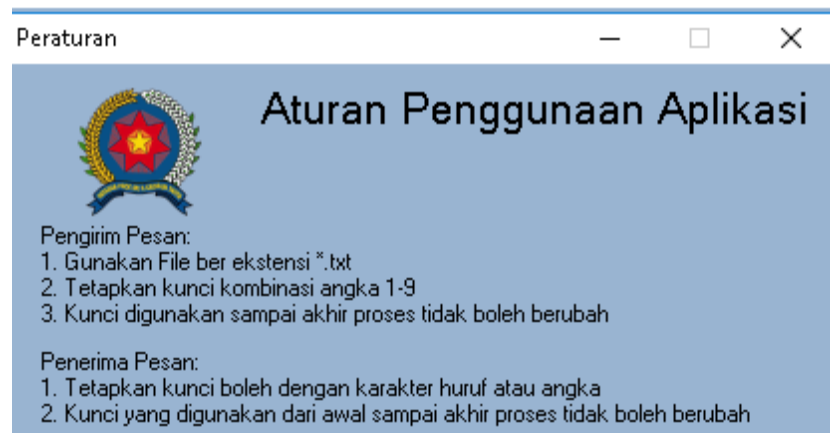
Tampilan pada gambar dibawah merupakan tampilan awal ketika aplikasi dijalankan. Pada form ini pengguna dapat memilih untuk membuka beberapa form lainnya seperti tombol tentang yang akan mengarahkan pengguna menuju form yang menjelaskan profil aplikasi ini, tombol materi dan tombol pengaturan yang akan mengarahkan pengguna ke form yang menjelaskan tata cara penggunaan dari aplikasi ini.



Gambar 4.1 Tampilan Awal/ *Home*

B. Tampilan Halaman Tentang

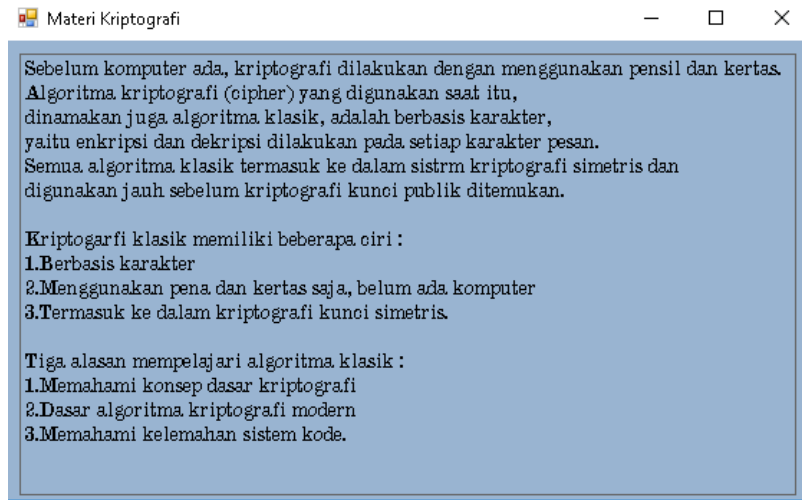
Tampilan halaman tentang merupakan tampilan halaman atau form yang berisi tentang tata cara penggunaan aplikasi yang dijalankan. Pada halaman tersebut dijelaskan apa-apa saja yang menjadi kewajiban bagi pengirim dan penerima saat menjalankan simulasi algoritma *vigenere cipher*.



Gambar 4.2. Tampilan Halaman Tentang

C. Tampilan Halaman Materi

Tampilan halaman materi merupakan tampilan halaman atau form yang berisi tentang materi yang dijalankan. Pada halaman tersebut dijelaskan apa-apa saja yang menjadi kewajiban bagi pengirim dan penerima saat menjalankan simulasi algoritma *vigenere cipher*.



Gambar 4.3 Tampilan Halaman Materi

D. Tampilan Halaman Judul

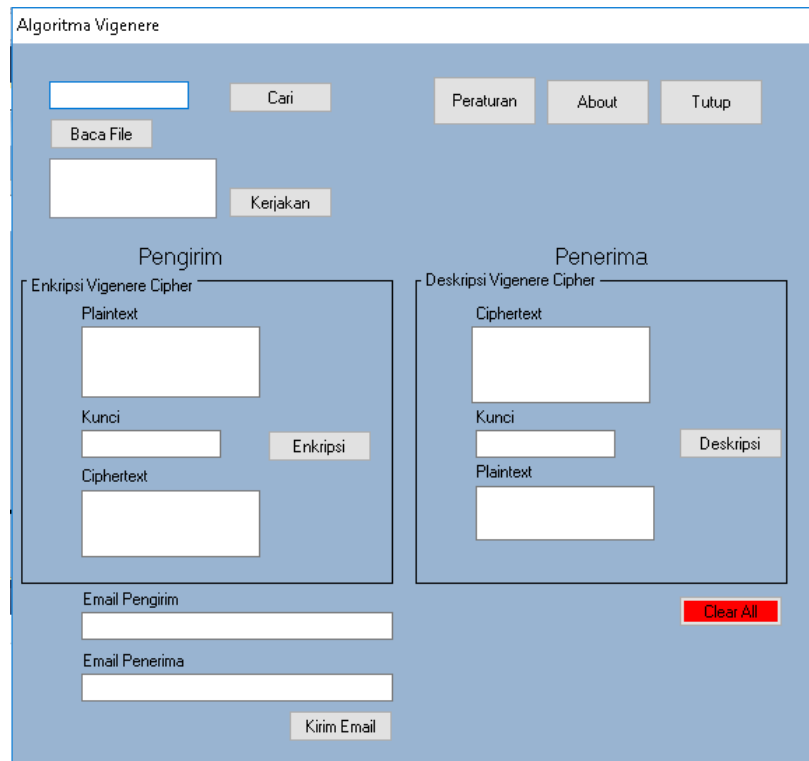
Tampilan berikut ini menampilkan halaman atau form yang berisi tentang profil dari aplikasi ini. Di dalamnya terdapat judul dari aplikasi beserta maksud dari pembuatannya beserta nama dan nomor pondok mahasiswi penulis.



Gambar 4.4 Tampilan Halaman Judul

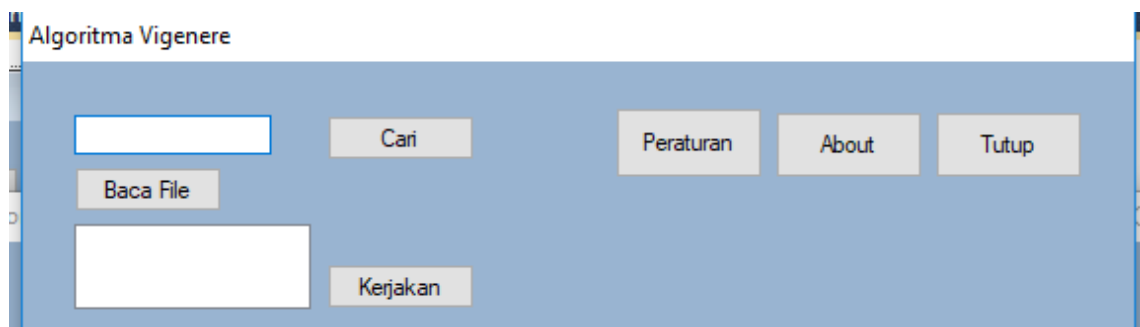
E. Tampilan Halaman Utama Algoritma *Vigenere*

Tampilan berikut merupakan tampilan utama pada aplikasi ini. Algoritma *vigenere* merupakan protokol yang menjamin tidak adanya pertukaran kunci antara pihak-pihak yang melakukan enkripsi dan dekripsi. Kedua belah pihak menggunakan kunci mereka masing-masing untuk mengenkripsi pesan dan kemudian untuk mendekripsi pesan tanpa perlu mengetahui kunci yang lainnya



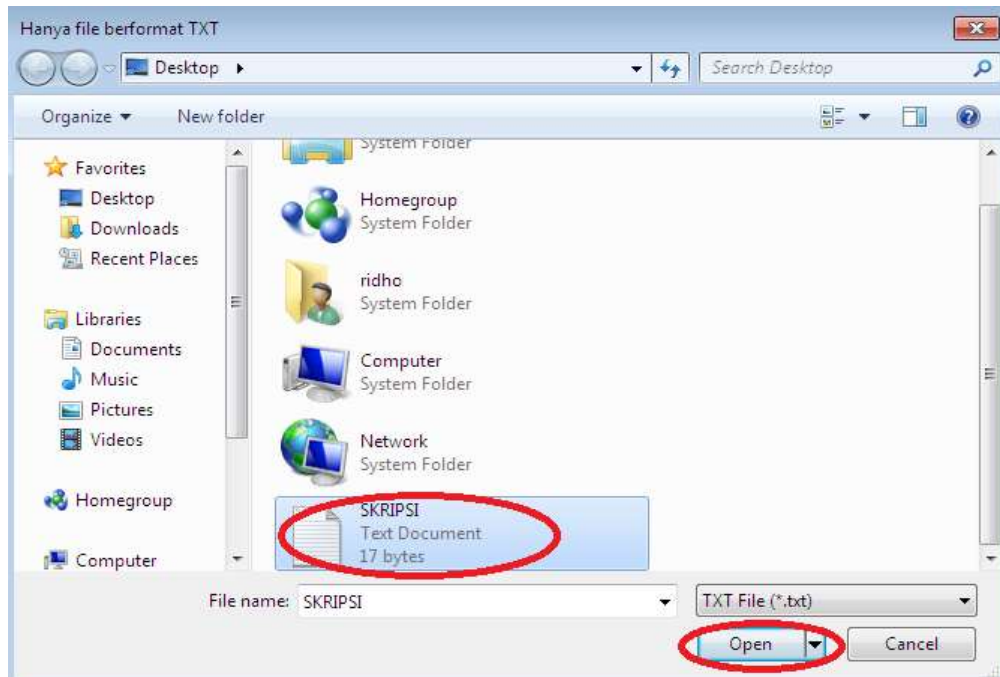
Gambar 4.5 Tampilan Halaman Utama Algoritma Vigenere

Uji coba pada *system* aplikasi ini dilakukan dengan memasukkan *input* teks yang bersumber dari file berekstensi **txt* ,dengan menggunakan tombol pencarian yang berada disisi kanan atas.



Gambar 4.6 Tombol Pencarian Data

Pengguna kemudian akan diarahkan menuju direktori file berekstensi **txt* tersebut berada.



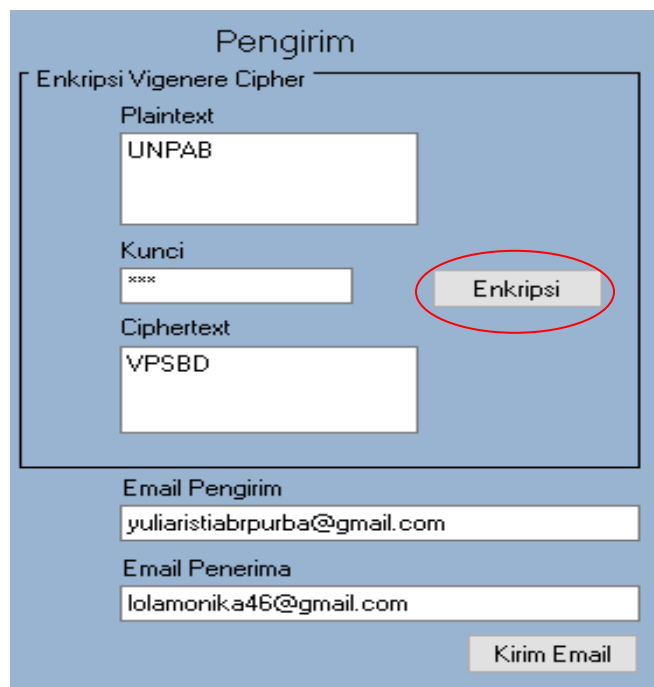
Gambar 4.7 Tampilan Memilih File

Setelah mendapatkan file berekstensi **txt* yang diinginkan pengguna akan diarahkan kembali ke halaman algoritma *vigenere*. Kemudian pengguna dapat menekan tombol buka *file* untuk menampilkan isi dari *file *txt* tersebut kedalam *listbox* yang tersedia.



Gambar 4.8 Tampilan Tombol Baca File

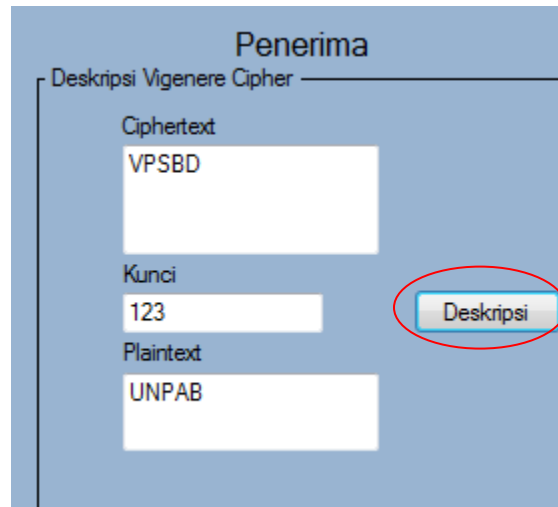
Otomatis rangkaian karakter tersebut akan berpindah ke *textbox* yang berada dibawahnya. Pada tahap awal rangkaian karakter akan berada di sisi bagian pengirim yang akan mengeksekusi rangkaian karakter tersebut untuk diubah menjadi *ciphertext* menggunakan Algoritma *Vigenere Cipher*. Untuk dapat mengeksekusi dibutuhkan kunci yang hanya dapat diisi karakter angka dari 0 sampai 9.



The image shows a web application interface for Vigenere cipher encryption. The main heading is "Pengirim". Below it, there is a section titled "Enkripsi Vigenere Cipher" which contains three text input fields: "Plaintext" with the value "UNPAB", "Kunci" with the value "****", and "Ciphertext" with the value "VPSBD". To the right of the "Kunci" field is a button labeled "Enkripsi" which is circled in red. Below this section are two more text input fields: "Email Pengirim" with the value "yuliaristiabrpurba@gmail.com" and "Email Penerima" with the value "lolamonika46@gmail.com". At the bottom right of the interface is a button labeled "Kirim Email".

Gambar 4.9 Tampilan Enkripsi dengan Algoritma *Vigenere*

Tombol enkripsi yang ditekan setelah memasukkan kunci berupa karakter angka selanjutnya akan mengeksekusi rangkaian karakter pesan asli yang selanjutnya akan dipanggil plaintext. Hasil enkripsi didapatkan pada *textbox* dibawahnya. Tombol kirim yang ditekan oleh penerima berfungsi untuk meneruskan pesan kembali pada pengirim. Selanjutnya *ciphertext* yang merupakan enkripsi dari *ciphertext* yang diterima dari pengirim akan diteruskan ke pengirim.



Gambar 4.10 Tampilan Deskripsi Vigenere Cipher

Aplikasi ditutup dengan menekan tombol tutup yang terdapat disisi kanan atas. Tombol tutup tersebut akan mengarahkan pengguna untuk kembali pada form awal.



Gambar 4.11 Tampilan Tombol Tutup

Selanjutnya setelah pengguna kembali ke halaman awal dari aplikasi ini. Pengguna dapat mengakhiri aplikasi dengan menekan tombol keluar yang terdapat pada pojok kanan atas.

4.2.1 Validasi Sistem

- a. Hasil Perhitungan Manual Proses Enkripsi.

Tabel 4.1. Konversi Huruf Ke 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

Pada tabel diatas berfungsi untuk memindahkan huruf dalam bentuk angka.

Langkah kedua membuat sebuah tabel yang bertujuan memindahkan huruf ke dalam bentuk angka.

Tabel 4.2 *Plaintext*

<i>Plaintext</i>	U	N	P	A	B
	85	78	80	65	66

Tabel 4.3 *Plaintext* dan *key*

Langkah selanjutnya, masukan kunci "1 2 3"

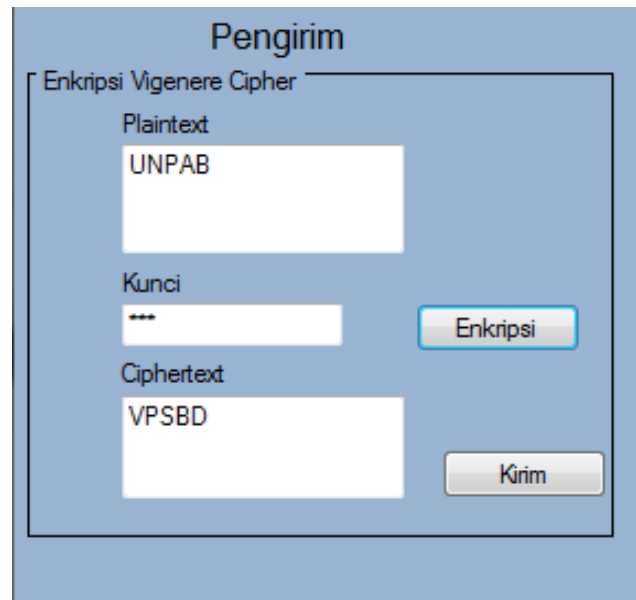
<i>Plaintext</i>	U	N	P	A	B
	85	78	80	65	66
<i>Key</i>	1	2	3	1	2

Pada baris tabel yang ketiga, kunci dimasukkan berulang sampai *cell* pada tabel terpenuhi. Pada langkah selanjutnya dilakukan penjumlahan antara baris kedua dan ketiga.

Tabel 4.4 Penjumlahan Manual Enkripsi

<i>Plaintext</i>	U	N	P	A	B
	85	78	80	65	66
<i>Key</i>	1	2	3	1	2
Kode CT	86	80	83	66	68

Setelah dilakukan perjumlahan maka langkah terakhir adalah mengembalikan hasil nilai angka ke dalam bentuk huruf.



Gambar 4.12 Perhitungan Aplikasi Enkripsi

Tabel 4.5 Perhitungan Manual Hasil Enkripsi

	<i>Plaintext</i>	U	N	P	A	B
		85	78	80	65	66
ENKRIPSI	<i>Key</i>	1	2	3	1	2
	Kode CT	86	80	83	66	68
	<i>Chipertext</i>	V	P	S	B	D

Maka diketahui ciphertext dari plaintext "UNPAB" dengan kunci "123" adalah VPSBD.

Kesimpulan : Berdasarkan proses enkripsi menggunakan aplikasi dan proses perhitungan manual, hasil yang didapat yaitu: proses yang diaplikasi sama dengan hasil yang ada pada perhitungan manual.

b. Hasil perhitungan manual proses deskripsi.

Setelah dienkrpsi, maka *plaintext* "UNPAB" akan berubah menjadi "VPSBD" berdasarkan kunci yang telah ditetapkan.

Tabel 4.6 *Ciphertext*

<i>Chipertext</i>	V	P	S	B	D
	86	80	83	66	68

Tabel 4.7 *Ciphertext* dan *key*

Kunci yang diinputkan adalah sebagai berikut.

<i>Chipertext</i>	V	P	S	B	D
	86	80	83	66	68
<i>Key</i>	1	2	3	1	2

Tabel 4.8 Penjumlahan Manual Dekripsi

Berdasarkan langkah diatas maka diperoleh hasil sebagai berikut.

<i>Chipertext</i>	V	P	S	B	D
	86	80	83	66	68
<i>Key</i>	1	2	3	1	2
Kode PT	85	78	80	65	66

Setelah dilakukan perjumlahan dari enkripsi ke dekripsi maka hasil akhirnya adalah sebagai berikut.

The image shows a software interface for manual decryption of a Vigenere cipher. The window is titled 'Penerima' and contains a section labeled 'Deskripsi Vigenere Cipher'. There are three input fields: 'Ciphertext' containing 'VPSBD', 'Kunci' containing '123', and 'Plaintext' containing 'UNPAB'. A 'Deskripsi' button is positioned to the right of the 'Kunci' field.

Gambar 4.13 Perhitungan Aplikasi Dekripsi

Tabel 4.9 Perhitungan Manual Hasil Dekripsi

DEKRIPSI	<i>Chipertext</i>	V	P	S	B	D
		86	80	83	66	68
	<i>Key</i>	1	2	3	1	2
	Kode PT	85	78	80	65	66
	<i>Plaintext</i>	U	N	P	A	B

Kesimpulan:

Berdasarkan proses deskripsi menggunakan aplikasi dan proses perhitungan manual, hasil yang didapat yaitu: proses yang diaplikasi sama dengan hasil yang ada pada perhitungan manual.

BAB V

PENUTUP

5.1 Simpulan

Berdasarkan pembahasan dalam perancangan Pembuatan Aplikasi *File* Menggunakan Algoritma *Vigenere Cipher*, maka dapat diambil kesimpulan sebagai berikut :

1. Perangkat lunak ini dirancang untuk menampilkan simulasi pengiriman pesan berekstensi yang diinputkan kedalam textbox antara pengirim dan penerima.
2. Pengirim mengirimkan pesan menggunakan dua kunci yang ditentukan sendiri oleh pengirim.
3. Penerima pesan menggunakan kunci yang diberikan oleh pengirim pesan, agar bisa membuka pesan asli yang dikirimkan oleh pengirim.

5.2 Saran

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

1. Diharapkan adanya kombinasi algoritma keamanan data lainnya.
2. Proses pengamanan data yang dilakukan oleh penulis masih menggunakan visual studio, diharapkan ada yang menggunakan diandroid agar bisa digunakan pada mobile.

DAFTAR PUSTAKA

- Anas, Irfan., Nanda, Putra, Arya., Hidayat Abdul. (2018). Implementasi Algoritma *Vigenere Cipher* dan *Gost* dalam Keamanan Data. *Jurnal Penelitian Teknik Informatika*. 2. (2). 19-20. Diakses dari <http://www.polgan.ac.id/jurnal/index.php/sinkron/article/view/146>
- Barus, S., Sitorus, V. M., Napitupulu, D., Mesran, M., & Supiyandi, S. (2018). Sistem Pendukung Keputusan Pengangkatan Guru Tetap Menerapkan Metode Weight Aggregated Sum Product Assesment (WASPAS). *Jurnal Media Informatika Budidarma*, 2(2).
- Fernando, Frenky., Siswanto., Suryana, Eko. (2014). Aplikasi Kriptografi Untuk Mengamankan File Audio Video Menggunakan Visual Basic.Net. *Jurnal Media Informasi*. 10. (1). 28-29. Diakses dari <https://jurnal.unived.ac.id/index.php/jmi/article/view/229>
- Hadi, Muhammad, Sartika., Samad, Abjan. (2019). Sistem Informasi Pengolahan Data Bantuan Beasiswa Siswa Miskin (BSM) Pada Kantor Wilayah Kementerian Agama Provinsi Maluku Utara. *Jurnal Ilmiah ILKOMINFO-Jurnal Ilmu Komputer dan Informatika*.2. (1) 4-5 Diakses dari <https://j-ilkominfo.org/index.php/ejournalaikom/article/view/15>
- Harahap, Muhammad Khoiruddin. (2016). Analisis Perbandingan Algoritma Kriptografi Klasik *Vigenere Cipher* dan *One Time Pad*. *Jurnal Nasional Informatika dan Teknologi Jaringan*. 1. (1). 61-62. Diakses dari <https://jurnal.uisu.ac.id/index.php/infotekjar/article/view/43>
- Hartanto, S. (2017). Implementasi fuzzy rule based system untuk klasifikasi buah mangga. *TECHSI-Jurnal Teknik Informatika*, 9(2), 103-122.
- Herdianto, H. (2018). Perancangan Smart Home dengan Konsep Internet of Things (IoT) Berbasis Smartphone. *Jurnal Ilmiah Core IT: Community Research Information*
- Harumy, T. H. F., & Sulistianingsih, I. (2016). Sistem penunjang keputusan penentuan jabatan manager menggunakan metode mfp pada cv. Sapo durin. In *Seminar Nasional Teknologi Informasi dan Multimedia* (pp. 6-7).
- Irwan, Muhammad, Dedi. (2017). Implementasi Kriptografi *Vigenere Cipher* dengan Php. *Jurnal Teknologi Informasi*. 1. (1). 13-14. Diakses dari <http://jurnal.una.ac.id/index.php/jurti/article/view/21>

- Khairul, K., Haryati, S., & Yusman, Y. (2018). Aplikasi Kamus Bahasa Jawa Indonesia dengan Algoritma Raita Berbasis Android. *Jurnal Teknologi Informasi dan Pendidikan*, 11(1), 1-6.
- Kharisma, Rizqi, Sukma., Rachman, Muhammad, Aziz, Fatchu. (2017). Pembuatan Aplikasi Notes Menggunakan Algoritma Kriptografi Polyalphabetic Substitution Cipher Kombinasi Kode ASCII dan Operasi XOR Berbasis Android. *Jurnal Teknologi Informasi*. 12. (35). 2-3. Diakses dari <http://jti.respati.ac.id/index.php/jurnalijti/article/view/176>
- Limbong, Tonni., Taufik, Insan. (2017). Aplikasi Pengacak Soal Ujian Untuk Type Soal Berbasis Microsoft Word Menggunakan Metode Linear Congruent Method. *Jurnal Manajemen dan Informatika komputer Pelita Nusantara*. 21. (1). 81-82. Diakses dari http://www.ejournal.ust.ac.id/index.php/Jurnal_Means/article/view/25
- Manullang, Dewi, Intan. (2018). Perancangan Aplikasi Penyandian File Teks dengan Algoritma BIFID Cipher. *Jurnal Pelita Informatika*. 17. (1). 65-70. Diakses dari <http://ejournal.stmik-budidarma.ac.id/index.php/pelita/article/view/620>
- Marzuki, Imam. (2018) Perancangan Dan Pembuatan Aplikasi Kriptografi Berbasis Web Menggunakan Algoritma RSA. *Jurnal Seminar Nasional Humaniora & Aplikasi Teknologi Informasi (SEHATI)*. 47-48
- Putri, R. E., & Siahaan, A. (2017). Examination of document similarity using Rabin-Karp algorithm. *International Journal of Recent Trends in Engineering & Research*, 3(8), 196-201.
- Rahim, R., Supiyandi, S., Siahaan, A. P. U., Listyorini, T., Utomo, A. P., Triyanto, W. A., & Khairunnisa, K. (2018, June). TOPSIS Method Application for Decision Support System in Internal Control for Selecting Best Employees. In *Journal of Physics: Conference Series* (Vol. 1028, No. 1, p. 012052). IOP Publishing.
- Sari, R. D., Supiyandi, A. P. U., Siahaan, M. M., & Ginting, R. B. (2017). A Review of IP and MAC Address Filtering in Wireless Network Security. *Int. J. Sci. Res. Sci. Technol*, 3(6), 470-473.
- Siahaan, A. P. U., Aryza, S., Nasution, M. D. T. P., Napitupulu, D., Wijaya, R. F., & Arisandi, D. (2018). Effect of matrix size in affecting noise reduction level of filtering.
- Sidik, A. P. (2018). Algoritma RSA dan Elgamal sebagai Algoritma Tambahan untuk Mengatasi Kelemahan Algoritma One Time Pad pada Skema Three Pass Protocol.

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

Widiyanto, Edy., Arifin, Mochammad., Soebijono, Tony. (2016). Rancang Bangun Aplikasi Simpan Pinjam Pada Koperasi Pegawai Negeri Republik Indonesia Hidup Tulungagung. *Jurnal JSIKA*. 6. (3). 2-3. Diakses dari <http://jurnal.stikom.edu/index.php/jsika/view/1682>

