

APLIKASI ENKRIPSI TEKS DENGAN MENGGUNAKAN METODE VIGENERE UNTUK KEAMANAN DATA *PASSWORD* DAN *EMAIL*

Rudiansyah

Teknik Informatika, STMIK Widya Cipta Dharma
Jl. M. Yamin No.25, Samarinda, 75123
E-mail : Rudi_rdy@.com

ABSTRAK

Penerapan Metode *Vigenere* pada Enkripsi Teks dengan Menggunakan Metode *Vigenere*, merupakan bentuk penelitian untuk membuktikan bahwa algoritma *Vigenere* dapat digunakan untuk pencarian solusi, khususnya pada permasalahan kerahasiaan Teks.

Tujuan dari penelitian ini adalah merancang dan membangun sebuah aplikasi yang dapat menyelesaikan masalah enkripsi teks untuk merahasiakan sebuah teks dengan mengacak nilai dari teks yang terdapat pada kode ASCII dengan menggunakan bahasa pemrograman *Microsoft Visual Studio 2010*. Dalam penelitian ini, teknik pengumpulan data yang digunakan adalah studi pustaka. Metode pengujian yang digunakan adalah dengan pengujian *Waterfall* dan Pengujian *Whitebox*.

Hasil dari penelitian ini adalah dibuatnya aplikasi enkripsi teks yang dapat mengacak teks yang terdapat pada sebuah pesan teks. Pengguna dapat menentukan kunci yang digunakan dalam melakukan proses enkripsi dan kemudian proses enkripsi menggunakan metode *Vigenere*.

Kata Kunci: Aplikasi, Enkripsi, Teks, *Vigenere*

1. PENDAHULUAN

Teknologi komunikasi dan informasi berkembang dengan pesat dan memberikan pengaruh besar bagi kehidupan manusia. Seiring dengan perkembangan teknologi sekarang ini yang semakin pesat maka proses pengiriman data dapat dilakukan dengan mudah dan melalui berbagai macam media yang telah ada antara lain, melalui media internet dengan menggunakan fasilitas email, melalui *transfer* data antara perangkat *mobile* maupun dengan teknologi radio *frequency* hingga dengan menggunakan jaringan komputer.

Perkembangan yang pesat dalam pengiriman data membawa dampak yang besar, yaitu masalah keamanan data yang di kirim. Untuk itu, tidak mungkin mengirim data melalui media-media tersebut secara polos (*plain*). Melainkan harus dilakukan proses pengamanan dengan cara melakukan enkripsi pada sebuah *file*.

Kriptografi dapat menjadi jawaban dari masalah tersebut. sebagai ilmu yang telah diaplikasikan untuk mengamankan data, kriptografi dapat digunakan untuk mengamankan data-data penting pada sebuah *file* data seperti pembuatan atau pengamanan kerahasiaan sebuah *password* dan kerahasiaan isi dari pesan sebuah *email* yang tidak boleh diketahui pada orang yang bukan seharusnya yang terkandung dalam sebuah *file* yang

disandikan atau dienkripsikan untuk diubah menjadi simbol tertentu sehingga hanya orang tertentu saja yang dapat mengetahui isi dari data tersebut.

Dalam perkembangan ilmu kriptografi masa sekarang ini, telah banyak tercipta algoritma-algoritma yang dapat digunakan untuk mengubah data asli (*plaintext*) menjadi simbol tertentu (*chipertext*) salah satu contohnya adalah algoritma *vigenere* yang merupakan algoritma klasik dengan teknik substitusi. Dimana metode ini menggunakan suatu kunci yang memiliki panjang tertentu. Panjang kunci tersebut bisa lebih pendek ataupun sama dengan panjang *plaintext*, maka kunci tersebut akan diulang secara periodik hingga panjang kunci tersebut sama dengan panjang *plaintextnya*.

2. RUANG LINGKUP PENELITIAN

Permasalahan difokuskan pada:

1. Membangun Aplikasi enkripsi menggunakan metode *vigenere* dengan menggunakan *Visual Basic.NET*.
2. Terdapat dua jenis Proses didalam aplikasi yaitu enkripsi dan dekripsi.
3. Menghasilkan Aplikasi enkripsi teks dengan metode *vigenere* dengan menggunakan *Visual Basic.NET*.

3. BAHAN DAN METODE

Adapun bahan dan metode yang digunakan dalam membangun aplikasi ini yaitu:

3.1 Aplikasi

Menurut Shelly (2009), Aplikasi adalah seperangkat instruksi khusus dalam komputer yang dirancang agar kita menyelesaikan tugas-tugas tertentu sebagai contoh, Aplikasi *enkripsi* teks adalah sebuah aplikasi yang diperuntukkan untuk keamanan data. aplikasi *enkripsi* teks adalah aplikasi yang diperuntukkan untuk keamanan data *file* seperti *password* dan *e-mail*

3.2 Kriptografi

Menurut Ariyus (2008), Kriptografi (*Cryptography*) berasal dari bahasa Yunani yaitu dari kata *Crypto* dan *Graphia* yang berarti penulisan rahasia. Kriptografi adalah suatu ilmu yang mempelajari penulisan secara rahasia. Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut *Cryptology*. Kriptografi bertujuan menjaga kerahasiaan informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak sah.

Dalam menjaga kerahasiaan data, kriptografi mentransformasikan data jelas (*plaintext*) ke dalam bentuk data sandi (*ciphertext*) yang tidak dapat dikenali. *Ciphertext* inilah yang kemudian dikirimkan oleh pengirim (*sender*) kepada penerima (*receiver*). Setelah sampai di penerima, *ciphertext* tersebut ditransformasikan kembali ke dalam bentuk *plaintext* agar dapat dikenali.

Proses transformasi dari *plaintext* menjadi *ciphertext* disebut proses *Encipherment* atau *enkripsi* (*encryption*), sedangkan proses mentransformasikan kembali *ciphertext* menjadi *plaintext* disebut proses *dekripsi* (*decryption*).

Untuk mengenkripsi dan mendekripsi data. Kriptografi menggunakan suatu algoritma (*cipher*) dan kunci (*key*). *Cipher* adalah fungsi matematika yang digunakan untuk mengenkripsi dan mendekripsi data. Sedangkan kunci merupakan sederetan bit yang diperlukan untuk mengenkripsi dan mendekripsi data.

Algoritma kriptografi modern tidak lagi mengandalkan keamanannya pada kerahasiaan algoritma tetapi kerahasiaan kunci. *Plaintext* yang sama bila disandikan dengan kunci yang berbeda akan menghasilkan *ciphertext* yang berbeda pula. Dengan demikian algoritma kriptografi dapat bersifat umum dan boleh diketahui oleh siapa saja, akan tetapi tanpa pengetahuan tentang kunci, data tersandi tetap saja tidak dapat terpecahkan.

3.3 Enkripsi

Menurut Ariyus (2008), *Enkripsi* adalah proses mengubah suatu pesan asli yang disebut *Plaintext* menjadi sebuah sandi atau kode yang tidak terbaca yang disebut *Chipertext* dan tidak dapat dimengerti, untuk mengembalikan pesan ke bentuk asli seperti semula diperlukan deskripsi. *Enkripsi* dimaksudkan untuk melindungi dan menyamarkan informasi agar tidak terlihat oleh pihak atau orang yang bukan seharusnya.

3.4 Dekripsi

Ariyus (2008), Deskripsi merupakan proses kebalikan dari *enkripsi* dimana pesan yang telah disandikan menjadi sebuah kode atau *Chipertext* di proses kembali agar dapat terbaca isi dari pesan tersebut. Deskripsi dimaksudkan untuk menampilkan pesan asli atau *Plaintext* yang telah di *Enkripsikan*.

Maka dapat disimpulkan pengertian dari *enkripsi* dan *dekripsi* adalah melindungi pesan yang dirahasiakan atau menjadikan sebuah pesan menjadi sandi-sandi atau kode kemudian dapat di jadikan sebuah pesan yang dapat di baca oleh semua orang.

3.5 Vigenere

Menurut Budi Satrio (2006), *Vigenere* merupakan salah satu algoritma klasik dengan teknik substitusi. Nama *vigenere* diambil dari seorang yang bernama Blaise de *Vigenere*. *Vigenere* menggunakan suatu kunci yang memiliki panjang tertentu. Panjang kunci tersebut bisa lebih pendek ataupun sama dengan panjang *plainteks*. Jika panjang kunci kurang dari panjang *plainteks*, maka kunci yang tersebut akan diulang secara periodik hingga panjang kunci tersebut sama dengan panjang *plainteks*nya.

Algoritma *enkripsi vigenere* :

$$C_i = (P_i + K_i) \bmod 26$$

Algoritma deskripsi *vigenere* :

$$P_i = (C_i - K_i) \bmod 26$$

Dimana :

C_i = nilai desimal karakter *ciphertext* ke- i

P_i = nilai desimal karakter *plaintext* ke- i

K_i = nilai desimal karakter kunci ke- i

26 adalah jumlah huruf dari abjad ($a - z$)

Sebagai contoh, jika *plaintext* adalah THEBEAUTYANDTHEBEAST dan kunci adalah ABC maka proses *enkripsi* yang terjadi adalah sebagai berikut :
Plaintext :

THEBEAUTYANDTHEBEAST

Kunci : ABCABCABCABCABC

Chipertext : UJHCGDVVBPGUJHCGDTV

Pada contoh di atas kata kunci ABC diulang sedemikian rupa hingga panjang kunci sama dengan panjang *plainteks*nya. Kemudian setelah panjang kunci sama dengan panjang *plainteks*, proses *enkripsi* dilakukan dengan melakukan menggeser setiap huruf pada *plainteks* sesuai dengan huruf kunci yang bersesuaian dengan huruf *plainteks* tersebut. Pada contoh *plainteks* di atas huruf pertama adalah T akan dilakukan pergeseran huruf dengan kunci $K_i=0$ (kunci huruf pertama adalah A yang memiliki $K_i=0$) menjadi T. huruf kedua pada *plainteks* adalah H akan dilakukan pergeseran huruf dengan kunci $K_i=1$ (kunci huruf ke dua adalah B yang memiliki $K_i=1$) menjadi I. Begitu seterusnya dilakukan pergeseran sesuai dengan kunci pada tiap huruf hingga semua *plainteks* telah *terenkripsi* menjadi *chiphertext*.

3.6 ASCII

Indah Novita Sari (2013), 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 7 bit. Namun, *ASCII* disimpan sebagai sandi 8 bit dengan menambahkan satu angka 0 sebagai bit significant paling tinggi. Bit tambahan ini sering digunakan untuk uji prioritas. Karakter kontrol pada *ASCII* dibedakan menjadi 5 kelompok sesuai dengan penggunaan yaitu berturut-turut meliputi *logical communication*, *Device control*, *Information separator*, *Code extention*, dan *physical communication*. Kode *ASCII* ini banyak dijumpai pada papan ketik (*keyboard*) komputer atau instrument-instrument digital.

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. Kode *ASCII* sendiri dapat dikelompokkan lagi kedalam beberapa bagian:

Kode yang tidak terlihat simbolnya seperti Kode 10(*Line Feed*), 13(*Carriage Return*), 8(*Tab*), 32(*Space*) khusus (~!@#%&^*()_+?:.”{ }) Kode yang tidak ada di keyboard namun dapat ditampilkan. Kode ini umumnya untuk kode-kode grafik. Dalam pengkodean kode *ASCII* memanfaatkan 8 bit. Pada saat ini kode *ASCII* telah tergantikan oleh kode *UNICODE* (*Universal Code*). *UNICODE* dalam pengkodeannya memanfaatkan 16 bit sehingga memungkinkan untuk menyimpan kode-kode lainnya seperti kode bahasa Jepang, Cina, Thailand dan sebagainya.

Pada papan *keyboard*, aktifkan numlock (tidak terdapat pada laptop), tekan tombol ALT secara bersamaan dengan kode karakter maka akan dihasilkan karakter tertentu. Misalnya: ALT + 44 maka akan muncul karakter koma (,). Mengetahui kode-kode *ASCII* sangat bermanfaat misalnya untuk membuat karakter-karakter tertentu yang tidak ada di *keyboard*. Berikut adalah tabel dari kode *ASCII* :

Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	↵	100 0000	100	64	40	@	110 0000	140	96	60	`
010 0001	041	33	21	!	100 0001	101	65	41	A	110 0001	141	97	61	a
010 0010	042	34	22	"	100 0010	102	66	42	B	110 0010	142	98	62	b
010 0011	043	35	23	#	100 0011	103	67	43	C	110 0011	143	99	63	c
010 0100	044	36	24	\$	100 0100	104	68	44	D	110 0100	144	100	64	d
010 0101	045	37	25	%	100 0101	105	69	45	E	110 0101	145	101	65	e
010 0110	046	38	26	&	100 0110	106	70	46	F	110 0110	146	102	66	f
010 0111	047	39	27	'	100 0111	107	71	47	G	110 0111	147	103	67	g
010 1000	050	40	28	(100 1000	110	72	48	H	110 1000	150	104	68	h
010 1001	051	41	29)	100 1001	111	73	49	I	110 1001	151	105	69	i
010 1010	052	42	2A	*	100 1010	112	74	4A	J	110 1010	152	106	6A	j
010 1011	053	43	2B	+	100 1011	113	75	4B	K	110 1011	153	107	6B	k
010 1100	054	44	2C	,	100 1100	114	76	4C	L	110 1100	154	108	6C	l
010 1101	055	45	2D	-	100 1101	115	77	4D	M	110 1101	155	109	6D	m
010 1110	056	46	2E	.	100 1110	116	78	4E	N	110 1110	156	110	6E	n
010 1111	057	47	2F	/	100 1111	117	79	4F	O	110 1111	157	111	6F	o
011 0000	060	48	30	0	101 0000	120	80	50	P	111 0000	160	112	70	p
011 0001	061	49	31	1	101 0001	121	81	51	Q	111 0001	161	113	71	q
011 0010	062	50	32	2	101 0010	122	82	52	R	111 0010	162	114	72	r
011 0011	063	51	33	3	101 0011	123	83	53	S	111 0011	163	115	73	s
011 0100	064	52	34	4	101 0100	124	84	54	T	111 0100	164	116	74	t
011 0101	065	53	35	5	101 0101	125	85	55	U	111 0101	165	117	75	u
011 0110	066	54	36	6	101 0110	126	86	56	V	111 0110	166	118	76	v
011 0111	067	55	37	7	101 0111	127	87	57	W	111 0111	167	119	77	w
011 1000	070	56	38	8	101 1000	130	88	58	X	111 1000	170	120	78	x
011 1001	071	57	39	9	101 1001	131	89	59	Y	111 1001	171	121	79	y
011 1010	072	58	3A	:	101 1010	132	90	5A	Z	111 1010	172	122	7A	z
011 1011	073	59	3B	;	101 1011	133	91	5B	[111 1011	173	123	7B	{
011 1100	074	60	3C	<	101 1100	134	92	5C	\	111 1100	174	124	7C	
011 1101	075	61	3D	=	101 1101	135	93	5D]	111 1101	175	125	7D	}
011 1110	076	62	3E	>	101 1110	136	94	5E	^	111 1110	176	126	7E	~
011 1111	077	63	3F	?	101 1111	137	95	5F	_					

Gambar 1. Kode ASCII

3.7 Password

Tom Wayteg (2013), *Password* adalah suatu bentuk dari data otentikasi rahasia yang digunakan untuk mengontrol akses ke dalam suatu sumber informasi. *Password* akan dirahasiakan dari mereka yang tidak diijinkan untuk mengakses, dan mereka yang ingin mengetahui akses tersebut akan diuji apakah layak atau tidak untuk memperolehnya.

Walaupun demikian, *password* bukan berarti kata-kata.tentu saja *password* yang bukan suatu kata yang mempunyai arti akan lebih sulit untuk ditebak. sebagai tambahan, *password* sering digunakan untuk menggambarkan sesuatu.

3.8 Email

Woole (2008), *E-mail* adalah surat dalam bentuk elektronik. *E-mail* merupakan salah satu fasilitas atau aplikasi internet yang paling banyak digunakan dalam hal surat menyurat.Hal ini dikarenakan *e-mail* merupakan alat komunikasi yang murah, cepat, dan efisien. Menggunakan *e-mail* memungkinkan kita untuk mengirimkan pesan dalam bentuk surat ke seluruh dunia dalam waktu yang sangat cepat dan biaya yang dikeluarkan pun hanyalah biaya untuk mengakses internet pada saat kita mengirim *e-mail* tersebut.adapun manfaat dan kegunaan dari *e-mail* adalah sebagai media komunikasi, media pengiriman, media promosi, media informasi,yang efektif, efisien,dan murah.

3.9 Microsoft Visual Basic.NET 2010

Edy Winarto (2010), *Microsoft Visual Basic 2010* salah satu bagian dari produk pemrograman yang di keluarkan oleh *Microsoft*, yaitu *Microsoft Visual Studio 2010*. Sebagai produk pengembangan atau *Integrated Development Environment* (IDE) yang di keluarkan oleh *Microsoft*, *Microsoft Visual Studio 2010* berisi beberapa IDE seperti *Visual Basic*, *Visual C++*, *Visual Web Developer*, *Visual C#* dan *Visual F#*. Semua IDE tersebut sudah mendukung penuh implementasi *.Net Framework* terbaru, yaitu *.Net Framework 4.0* yang merupakan pengembangan dari *.Net Framework 3.5*. Adapun *database* standar yang di sertakan adalah *Microsoft SQL Server 2008 Express*.

Beberapa kemampuan dari *Visual Basic* antara lain sebagai berikut :

1. Membuat program aplikasi berbasis *windows*.
2. Membuat objek-objek pembantu program, misalnya *File*, *Help*, *kontrol ActiveX* dan sebagainya.
3. Menguji program dan menghasilkan program akhir berekstensi *EXE* yang langsung dapat dijalankan.

3.10 Algoritma Pseudocode

Cybernur (2010), *Pseudocode* adalah kode atau tanda yang menyerupai atau merupakan penjelasan cara menyelesaikan suatu masalah.*pseudocode* sering digunakan oleh seseorang untuk menuliskan algoritma dari suatu permasalahan. *pseudocode* berisikan langkah-langkah untuk menyelesaikan suatu permasalahan, hampir sama dengan algoritma hanya saja bentuknya

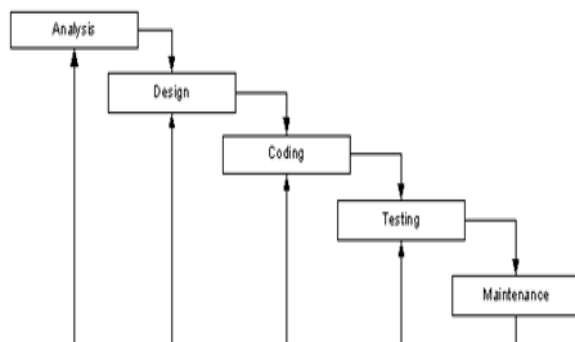
sedikit berbeda dari algoritma. *pseudocode* menggunakan bahasa yang hampir menyerupai bahasa pemrograman, selain itu biasanya *pseudocode* menggunakan bahasa yang mudah dipahami secara *universal* dan juga lebih ringkas dari algoritma.

<p><u>Deklarasi</u> Key newKey, ciphertext : string; Karakter: Array[1...karakter.length] of string; I, j : integer</p>	<p><u>Deklarasi</u> Key newKey, ciphertext : string; Karakter: Array[1...karakter.length] of string; I, j : integer</p>
<p><u>Enkripsi :</u> Input (key); Input (plaintext); For (int i ← 0; i < plaintext.length) do j ← i mod key.length(); newKey ← key(j); end for; Write(newKey); Karakter k ← plaintext; key ← get(key.plaintext); for (int i ← 0; i < karakter.length) do ciphertext ← (char) (((95 + ((karakter[i] + key(i) mod 95 + ''))) mod 95 + '')); end for Write (ciphertext)</p>	<p><u>Dekripsi :</u> Input (key); Input (Chipertext); For (int i ← 0; i < Chipertext.length) do j ← i mod key.length(); newKey ← key(i); end for; Write(newKey); Karakter k ← ciphertext; key ← get(key.ciphertext); for (int i ← 0; i < karakter.length) do plaintext ← (char) (((95 + ((karakter[i] + key(i) mod 95 + ''))) mod 95 + '')); end for Write (plaintext)</p>

Gambar 2. Gambaran Algoritma Pseudocode

3.11 Metode Waterfall

Menurut Pressman (2010), *waterfall* sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahapan pendukung (*support*) pemeliharaan (*maintenance*).



Gambar 3. Metode Waterfall
 Sumber : Pressman (2010).

Berikut adalah penjelasan dari tahapan Metode *waterfall* meliputi?

1. Analisis

Analisis adalah tahapan yang bertujuan untuk menjabarkan segala sesuatu yang nantinya akan di tangani oleh perangkat lunak, atau tujuan dan fungsi yang akan dilakukan aplikasi ini.

2. Design

Design Proses perancangan desain pemilihan elemen yang tepat akan dalam perancangan desain dapat mengoptimalkan hasil yang akan sesuai dengan aplikasi enkripsi teks metode *vigenere* tersebut.

3. Coding

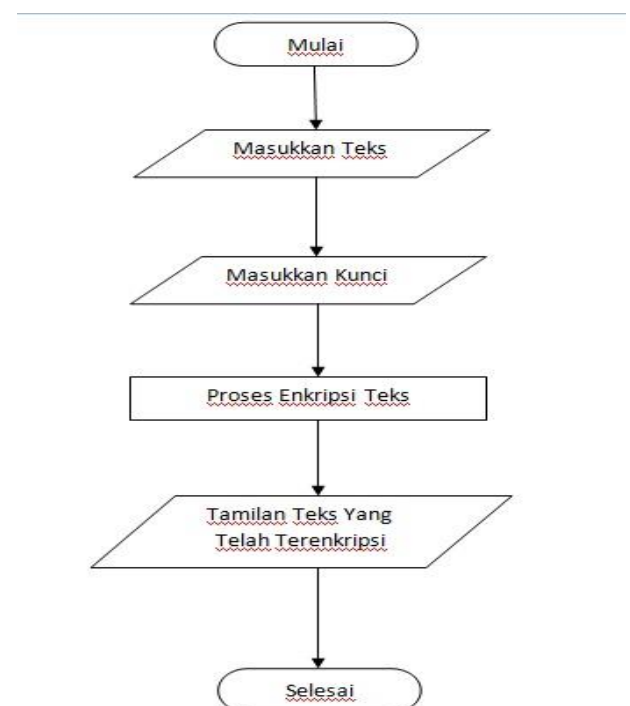
Mengkonversi apa yang telah dirancang sebelumnya ke dalam sebuah bahasa yang dimengerti komputer atau mengimplementasikan desain menjadi program melalui bahasa pemrograman.

4. Testing

Dilakukan pengujian sistem sekaligus pemeriksaan apakah program yang di jalankan sudah sesuai dengan yang di harapkan. Dimana pada penelitian ini menggunakan metode pengujian *whitebox*.

1. RANCANGAN SISTEM/APLIKASI

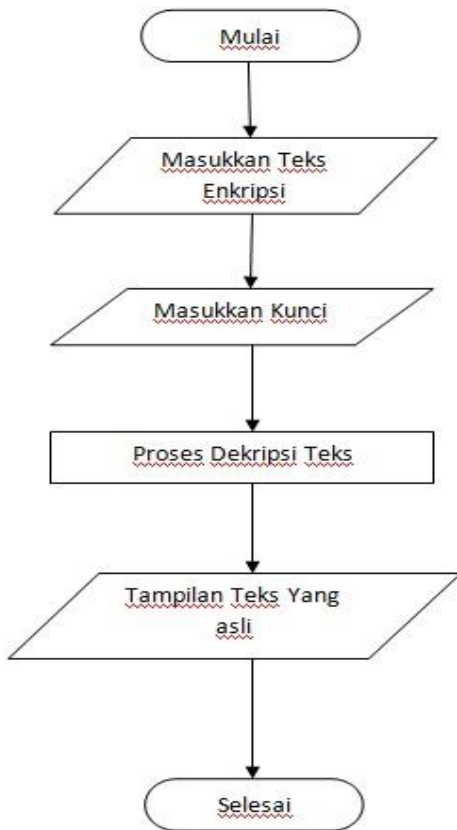
Berikut ini merupakan *flowchart* Proses enkripsi yang akan dibangun pada Aplikasi enkripsi teks menggunakan metode *vigenere* dengan Menggunakan *Microsoft Visual Basic*:



Gambar 4. Flowchart Proses Enkripsi

Pada *flowchart* enkripsi terdapat alur proses untuk mengenkripsi teks sehingga mendapat kan hasil teks yang telah terdekripsi.

Berikut ini adalah *flowchart* alur dekripsi teks :

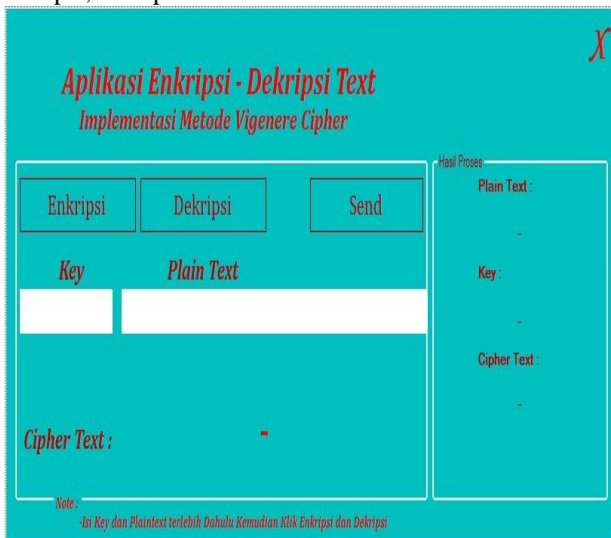


Gambar 5. Flowchart Proses Dekripsi

Pada *flowchart* Dekripsi terdapat alur proses untuk mendekripsikan teks yang telah di enkripsikan menjadi teks asli.

2. IMPLEMENTASI

Berikut merupakan tampilan awal aplikasi enkripsi pada saat awal dijalankan yang mana terdapat tombol enkripsi, dekripsi serta tombol *send*.



Gambar 6. Tampilan awal

Berikut merupakan tampilan pada saat telah melakukan proses enkripsi sehingga mendapatkan hasil teks yang telah terdekripsi.



Gambar 7. Tampilan hasil enkripsi

Berikut merupakan tampilan pada saat proses dekripsi sehingga mendapatkan hasil dekripsi atau mendapatkan teks asli setelah di enkripsi.



Gambar 8. Tampilan hasil dekripsi

3. KESIMPULAN

Dengan adanya hasil penelitian yang dilakukan dan berdasarkan uraian-uraian yang dibahas pada bab-bab sebelumnya, maka dapat diambil kesimpulan :

1. Pembuatan aplikasi enkripsi dan dekripsi dengan menggunakan metode *vigenere* melalui proses demi proses, desain dan pembuatan menggunakan program *Microsoft Visual Basic 2010*, setelah itu menjadi sebuah *file* berformat *.EXE* dapat di jalankan di PC.
2. Metode *Vigenere* yang diaplikasikan pada aplikasi ini dapat berfungsi dengan baik serta menggunakan kunci yang sama untuk enkripsi dan dekripsi.
3. Waktu yang digunakan untuk melakukan proses enkripsi dan dekripsi tergolong cepat.
4. Kunci yang digunakan pada aplikasi kriptografi ini hanya berupa alphabet atau huruf saja.

4. SARAN

Berdasarkan hasil dari penelitian ini akan beberapa saran, yaitu sebagai berikut :

1. Dalam Pemrosesan Enkripsi dan Dekripsi kunci yang digunakan masih tergolong biasa maka lebih baik lagi untuk kunci di enkripsikan menggunakan metode yang berbeda.
Aplikasi Diakses pada tanggal 23 Oktober 2014.
2. Ali, Muhammad Daud. 2008. Pendidikan Agama dapat di kembangkan ke operasi sistem *handphone* seperti android dan *windows phone*.

5. DAFTAR PUSTAKA

- Ariyus Dony. 2008. Pengantar Ilmu Kriptografi. Penerbit : ANDI. Yogyakarta.
- Tri Haryono. 2009. Enkripsi Citra Digital RGB dengan menggunakan metode *Vigenere*. STMIK Widya Cipta Dharma. Samarinda
- Budi Satrio. 2006. *Vigenere Cipher*. Program Studi Teknik Informatika. Bandung.
- Cybernur. 2010. Pseudocode Enkripsi dan Dekripsi *Vigenere Cipher*, (Online) <https://cybernur.wordpress.com/tag/pseudocode-enkripsi-dan-dekripsi-vigenere-cipher/> (Di akses : 01 Februari 2015).
- Daryanto. 2004, Keterampilan Dasar Pengoperasian Komputer. Bandung : Yrama Widya. Kadir, Abdul. 2009. *Dasar Perancangan dan Implementasi Database Relasional*. Yogyakarta : Andi Offset.

Elsa Suryana. 2013. Pengertian tentang keamanan informasi. (Online) <http://elsa-suryana-fisip-12.web.unair.ac.id> (Di akses : 01 Februari 2015).

Liikanen Erkki. 2009. Pengertian Keamanan data. ISSE. Berlin

Muhammad Farid Fachrurozi. 2010. Enkripsi pesan rahasia menggunakan algoritma AES. Universitas Islam. Jakarta.

Indah Novita Sari . 2013. Pengertian kode ASCII, (Online) <http://indahnyopie.wordpress.com/2008/09/20/pengertian-ascii/> (Di akses : 01 Februari 2015).

Kevin Chandra Irwanto, 2010.(Aplikasi bilangan dalam sandi *vigenere* dan *Cesar*).

Pressman, Roger. S, 2010. *Rekayasa Perangkat Lunak : Pendekatan Praktisi*. Yogyakarta : Andi

Shelly. 2009. Pengertian Tentang Aplikasi (Online) <http://juraganandroid.wordpress.com/2014/04/08/pengertian-aplikasi/> (Di akses : 01 Februari 2015).

Tri Haryono. 2009. Enkripsi gambar dengan menggunakan algoritma *secure image protection*. Politeknik Elektronika Negeri. Surabaya.

Wayteg Tom. 2013. Kamus internet, cara cepat masuk dunia *cyber*. Penertbit Liris Cetakan I. Surabaya.

Webster. 2013. Pengertian tentang data, New World Dictionary.

Woole David. 2008. Pengertian dan manfaat atau kegunaan *Email*, (Online) Yana185.pun.bz/pengertian-email-dan-klasifikasi-email.xhtml

Yatini, Indra. 2010, *Flowchart, Algoritma, dan Pemrograman Menggunakan Bahasa C++ Builder*. Yogyakarta: Graha Ilmu. Tjiharjadi Semuil. 2009, Pengaman data menggunakan metode enkripsi Einstein. Sistem Komputer, Fakultas Teknik, Universitas Kristen Maranatha. Bandung.