Studi Informatika: Jurnal Sistem Informasi , 7(2), 2014, 1-11

PENERAPAN KOMBINASI SANDI CAESAR DAN VIGENERE UNTUK PENGAMANAN DATA PESAN PADA SURAT ELEKTRONIK

Faisal Zuli [1] , Ari Irawan [2]

[1]Prodi Teknik Informatika Fakultas Teknik
Universitas Satya Negara Indonesia Jakarta
e-mail : faizal.zuli@yahoo.com

[2]Prodi Sistem Informasi Fakultas Sains dan Teknologi
Universitas Islam Negeri Syarif Hidayatullah Jakarta
e-mail :ari.irawan@uinjkt.ac.id

ABSTRACT

Security of the content of the message is especially important if the message sent by internet or email access. The security techniques may use a combination of algorithms Vigenere cipher and password. Both are part of the science of cryptography. It would be easier if the algorithm is implemented into an application that can later be used to secure the contents of the email message so that the message is protected from acts of interception.

Keywords: *cryptography, caesar cipher, vigenere password, encryption, decryption, ciphertext, plaintext*

ABSTRAK

Keamanan isi pesan sangat penting jika pesan yang dikirim oleh internet atau akses email . Teknik-teknik keamanan mungkin menggunakan kombinasi dari algoritma Vigenere cipher dan password . Keduanya merupakan bagian dari ilmu kriptografi . Akan lebih mudah jika algoritma tersebut diimplementasikan ke dalam sebuah aplikasi yang nantinya dapat digunakan untuk mengamankan isi pesan email sehingga pesan dilindungi dari tindakan intersepsi .

Kata kunci *: kriptografi , cipher caesar , password vigenere , enkripsi , dekripsi , ciphertext , plaintext*

1. Pendahuluan

Maraknya aksi penyadapan saat ini perlu kita sikapi dengan serius karena aksi penyadapan tersebut telah melanggar hak asasi manusia dalam berkomunikasi dengan aman dalam hal ini adalah komunikasi melalui surat elektronik dengan akses internet atau dikenal dengan email. Data atau isi pesan yang ada pada email tersebut harus dijaga kerahasiaannya yaitu dengan salah satu cara menggunakan ilmu kriptografi.

Ilmu kriptografi adalah suatu teknik untuk mengamanankan data atau pesan[1]. Pengamanan data atau pesan dapat dilakukan dengan menggunakan berbagai algoritma, salah satunya dapat menggunakan sandi Caesar dan Vigenere.

Sandi Caesar dan Vigenere merupakan bagian dari awal perkembangan ilmu kriptografi, atau bagian dari kriptografi klasik. Algoritma ini memanfaatkan pergeseran huruf yang ada pada data atau pesan yang akan diamankan dengan menggunakan sebuah kunci berupa jumlah pergeseran dan kata atau susunan kata untuk proses pengacakan data atau pesan. Proses yang dilakukan yaitu adalah enkripsi dan dekripsi[2].

Karya ilmiah ini membahas mengenai pemanfaatan kombinasi sandi Caesar dan Vigenere untuk mengamankan data teks pada pesan email, agar isi pesan email tersebut hanya bisa dibaca oleh pihak penerima pesan email.

2. Landasan Teori

A. Kriptografi

kriptografi adalah suatu ilmu yang mempelajari teknik - teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data[1]. Beberapa definisi mengenai kriptografi :

a. Kriptografi adalah cabang matematika yang menyediakan teknik untuk memungkinkan informasi rahasia yang akan dikirim melalui jaringan publik[3].
b. Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan[.
c. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas, data, serta otentikasi[2].

Dalam ilmu Kriptografi terbagi menjadi 2 aliran, yaitu : kriptografi klasik dan kriptografi modern. Pada Kriptografi klasik terdapat beberapa teknik Enkripsi yaitu : Subtitusi, Transposisi. algoritma Caesar dan Vigenere cipher termasuk dari teknik Enkripsi subtitusi[2].

B. Keamanan Sistem Kriptografi

Salah satu upaya pengamanan sistem informasi yang dapat dilakukan adalah kriptografi. Kriptografi sesungguhnya merupakan studi terhadap teknik matematis yang terkait dengan aspek keamanan suatu sistem informasi, antara lain seperti kerahasiaan, integritas data, otentikasi, dan ketiadaan penyangkalan. Keempat aspek tersebut merupakan tujuan fundamental dari suatu sistem kriptografi[2].

1. Kerahasiaan (*confidentiality*)
   Kerahasiaan adalah layanan yang digunakan untuk menjaga informasi dari setiap pihak yang tidak berwenang untuk mengaksesnya. Dengan demikian informasi hanya akan dapat diakses oleh pihak-pihak yang berhak saja.
2. Integritas data (*data integrity)*
   Integritas data merupakan layanan yang bertujuan untuk mencegah terjadinya pengubahan informasi oleh pihak-pihak yang tidak berwenang. Untuk meyakinkan integritas data ini harus dipastikan agar sistem informasi mampu mendeteksi terjadinya manipulasi data. Manipulasi data yang dimaksud di sini meliputi penyisipan, penghapusan, maupun penggantian data.
3. Otentikasi (*authentication*)
   Otentikasi merupakan layanan yang terkait dengan identifikasi terhadap pihak-pihak yang ingin mengakses sistem informasi (*entity authentication*) maupun keaslian data dari sistem informasi itu sendiri (*data origin authentication*).
4. Ketiadaan penyangkalan (*non-repudiation*)
   Ketiadaan penyangkalan adalah layanan yang berfungsi untuk mencegah terjadinya penyangkalan terhadap suatu aksi yang dilakukan oleh pelaku sistem informasi.

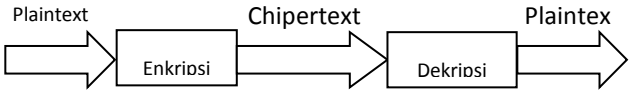
C. Mekanisme Kriptografi

Suatu sistem kriptografi (kriptosistem) bekerja dengan cara menyandikan suatu pesan menjadi suatu kode rahasia yang dimengerti oleh pelaku sistem informasi saja. Pada dasarnya mekanisme kerja semacam ini telah dikenal sejak jaman dahulu. Bangsa Mesir kuno sekitar 4000 tahun yang lalu bahkan telah mempraktekkannya dengan cara yang sangat primitif.

Dalam era teknologi informasi sekarang ini, mekanisme yang sama masih digunakan tetapi tentunya implementasi sistemnya berbeda. Sebelum membahas lebih jauh mekanisme kriptografi modern,berikut ini diberikan beberapa istilah yang umum digunakan dalam pembahasan kriptografi[2].

1. *Plaintext*
   Plaintext (*message*) merupakan pesan asli yang ingin dikirimkan dan dijaga keamanannya. Pesan ini tidak lain dari informasi tersebut.
2. *Chipertext*
   Chipertext merupakan pesan yang telah dikodekan (disandikan) sehingga siap untuk dikirimkan.
3. *Cipher*
   Cipher merupakan algoritma matematis yang digunakan untuk proses Enkripsi *plaintext* menjadi ciphertext.
4. Enkripsi
   Enkripsi (*encryption*) merupakan proses yang dilakukan untuk menyandikan *plaintext* sehingga menjadi *chipertext*.
5. Dekripsi
   Dekripsi (*decryption*) merupakan proses yang dilakukan untuk memperoleh kembali *plaintext* dari *chipertext.*
6. Kriptosistem
   Kriptosistem merupakan sistem yang dirancang untuk mengamankan suatu sistem informasi dengan memanfaatkan kriptografi.

Urutan-urutan proses kriptografi dapat digambarkan sebagai berikut.



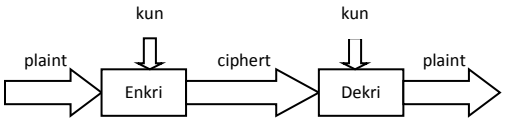
Gambar 2.1. Mekanisme kriptografi

Prosesnya pada dasarnya sangat sederhana. Sebuah plaintext (m) akan dilewatkan pada proses enkripsi (E) sehingga menghasilkan suatu ciphertext (c). Kemudian untuk memperoleh kembali plaintext, maka ciphertext (c) melalui proses dekripsi (D) yang akan menghasilkan kembali plaintext (m). Secara matematis proses ini dapat dinyatakan sebagai,

$E(m) = c$
$D(c) = m$
$D(E(m)) = m$

Kriptografi sederhana seperti ini menggunakan algoritma Enkripsi yang disebut *cipher*. Keamanannya bergantung pada kerahasiaan algoritma Enkripsi tersebut, karena itu algoritmanya harus dirahasiakan. Pada kelompok dengan jumlah besar dan anggota yang senantiasa berubah, penggunaannya akan menimbulkan masalah. Setiap ada anggota yang meninggalkan kelompok, algoritma harus diganti karena anggota ini dapat saja membocorkan algoritma.

Kriptografi modern selain memanfaatkan algoritma juga menggunakan kunci (*key*) untuk memecahkan masalah tersebut. Proses enkripsi dan dekripsi dilakukan dengan menggunakan kunci ini. Setiap anggota memiliki kuncinya masing-masing yang digunakan untuk proses enkripsi dan dekripsi yang akan dilakukannya. Dengan demikian ada sedikit perubahan yang harus dilakukan pada mekanisme yang digambarkan pada gambar 2.1 menjadi seperti gambar 2.2 berikut ini. :



Gambar 2.2 Kriptografi berbasis kunci

D. Sandi Caesar

Sandi Caesar diambil dari nama kaisar romawi Julius Caesar, dalam mengirimkan pesan Julius Caesar mengamankannya dengan cara isi pesan yang ada disandikan dengan mengganti posisi setiap huruf yang ada pada pesan dengan huruf lain yang memiliki posisi selisih huruf yang lain dari urutan alfabet[4]. Adapun langkah – langkah yang dilakukan adalah sebagai berikut :

a. Menentukan besarnya jumlah pergeseran huruf yang akan diganti
b. Mengganti setiap huruf yang ada pada pesan sesuai dengan jumlah pergeseran huruf yang ditentukan.
c. Merangkai kembali jumlah huruf sesuai dengan susunan pesan awal

Tabel 2.1. Susunan Abjad

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

Untuk menyandikan suatu pesan cukup mengganti huruf yang ada pada pesan dengan huruf sandi sesuai dengan jumlah pergeseran huruf yang diinginkan.

Contoh Enkripsi Caesar :

Teks Awal : PESAN INI SANGAT RAHASIA
Jumlah geser (Key) : 12
Teks Sandi : BQEMZ UZU EMZSMF DMTMEUM

E. Sandi Vigenere

Sandi Vigenere adalah suatu algoritma yang digunakan untuk Enkripsi data atau pesan dengan cara data atau pesan akan disandikan dengan menggunakan sebuah kata kunci (*Key*) yang berupa kata atau paduan kata[4]. Setiap huruf yang ada pada data atau pesan dipasangkan tepat dengan huruf yang terdapat pada kata kunci yang ditentukan, lalu kemudian dilakukan proses Enkripsi yaitu enkripsi.

Contoh penggunaan sandi Vigenere :
Teks Awal : PESAN INI SANGAT RAHASIA
Kata Kunci : ARMADA
Teks Sandi : PVEAQ INZ EAQGAK DAKASZM

F. Sandi Playfair

Sandi Playfair adalah salah satu teknik kriptografi. Dalam teknik ini pesan dienkripsi berdasarkan pasangan huruf, bukan huruf tunggal seperti sandi klasik lainnya[7].

Sandi Playfair ditemukan oleh ahli Fisika berkebangsaan Inggris bernama Sir Charles Wheatstone (1802 - 1875) namun dipromosikan oleh Baron Lyon Playfair (1819 - 1898) pada tahun 1854[7].

Dibandingkan dengan sandi-sandi lainnya, sandi *Playfair* dapat meningkatan keamanan dalam pengiriman sebuah pesan rahasia sehingga dapat memberikan jaminan integritasdata serta menjaga kerahasiaan. Sandi *Playfair* pertama kali digunakan untuk tujuan-tujuan taktis oleh pasukan Inggris dalam Perang Boer II dan Perang Dunia I. Australia dan Jerman juga menggunakan sandi ini untuk tujuan yang sama dalam Perang Dunia II.

Sandi *Playfair* paling sering digunakan karena penggunaannya yang sangat sederhana dan tidak memerlukan peralatan khusus untuk membaca atau menerjemahkan suatu sandi yang bersifat rahasia. Pada perkembangan selanjutnya, sandi ini tidak lagi digunakan oleh pasukan militer karena telah muncul berbagai perangkat enkripsi digital untuk menerjemahkannya. Sandi Playfair dianggap tidak aman lagi untuk menjaga suatu kerahasiaan pesan karena komputer dengan piranti lunak tertentu dapat memecahkan suatu sandi dalam hitungan detik.
Sandi *Playfair* menggunakan 25 huruf sebagai kunci yang disusun dalam bujur sangkar dengan menghilangkan huruf J dari abjad. Susunan kunci di dalam bujur sangkar tersebut diperluas dengan menambahkan kolom keenam dan baris keenam[7].
Keunggulan :

1. Proses enkripsi dan dekripsi data menggunakan kombinasi dua huruf sehingga kriptanalis yang menggunakan teknik analisis frekuensi sangat sulit untuk memecahkan sandi playfair.
2. Tabel kunci hanya digunakan sekali karena terdapat kemungkinan tabel kunci tersebut telah dipecahkan oleh pihak yang tidak berkepentingan.

Kelemahan :

1. Sandi Playfair dengan mudah dapat dipecahkan dengan menggunakan teknik frekuensi ditribusi ganda, yaitu dengan menghitung frekuensi kemunculan pasangan dua huruf sandi yang kemudian dibandingkan dengan frekuensi pasangan dua huruf pada suatu bahasa.
2. SandiPlayfair tidak menggunakan huruf J dalam tabel kunci sehingga bisa menimbulkan makna atau arti ganda pada saat memecahkan atau menerjemahkan suatu sandi.
3. SandiPlayfair tidak cocok digunakan untuk menyampaikan pesan rahasia yang cukup panjang.

G. Sandi Blowfish

Blowfish merupakan algoritma kunci simetrik cipher blok yang dirancang pada tahun 1993 oleh Bruce Schneier untuk menggantikan DES. Pada saat itu banyak sekali rancangan algoritma yang ditawarkan, namun hampir semua terhalang oleh paten atau kerahasiaan pemerintah Amerika. Schneier menyatakan bahwa blowfish bebas paten dan akan berada pada domain publik. Dengan pernyataan Schneier tersebut blowfish telah mendapatkan tempat di dunia kriptografi, khususnya bagi masyarakat yang membutuhkan algoritma kriptografi yang cepat, kuat, dan tidak terhalang oleh lisensi[8].

Keberhasilan blowfish dalam menembus pasar telah terbukti dengan diadopsinya blowfish sebagai Open Cryptography Interface (OCI) pada kernel linux versi

2.5 keatas. Dengan diadopsinya blowfish, maka telah menyatakan bahwa dunia open source menganggap blowfish adalah salah satu algoritma yang terbaik. Kesuksesan blowfish mulai memudar setelah kehadiran algoritma-algoritma dengan ukuran blok yang lebih besar, seperti AES. AES sendiri memang dirancang untuk menggantikan DES. Sehingga secara keseluruhan AES lebih unggul dari DES dan juga blowfish[8].

Blowfish adalah algoritma kriptografi kunci simetrik cipher blok dengan panjang blok tetap sepanjang 64 bit[8]. Algortima tersebut juga menerapkan teknik kunci yang berukuran sembarang. Ukuran kunci yang dapat diterima oleh blowfish adalah antara 32 hingga 448 bit, dengan ukuran standar sebesar 128 bit. Blowfish memanfaatkan teknik pemanipulasian bit dan teknik pemutaran ulang dan pergiliran kunci yang dilakukan sebanyak 16 kali. Algoritma utama terbagi menjadi dua sub-algoritma utama, yaitu bagian ekspansi kunci dan bagian enkripsi-dekripsi data.

Pengekspansian kunci dilakukan pada saat awal dengan masukan sebuah kunci dengan panjang 32 hingga 448 bit, dan keluaran adalah sebuah larik sub-kunci dengan total 4168 bita. Bagian enkripsi-dekripsi data terjadi dengan memanfaatkan perulangan 16 kali terhadap jaringan feistel. Setiap perulangan terdiri dari permutasi dengan masukan adalah kunci, dan substitusi data. Semua operasi dilakukan dengan memanfaatkan operasi xor dan penambahan. Operasi penambahan dilakukan terhadap empat larik lookup yang dilakukan setiap putarannya.

H. Sandi RSA

Algortima RSA dijabarkan pada tahun 1977 oleh tiga orang : Ron Rivest, Adi Shamir dan Len Adleman dari Massachusetts Institute of Technology. Huruf RSA itu sendiri berasal dari inisial nama mereka (Rivest—Shamir—Adleman)[9].

Clifford Cocks, seorang matematikawan Inggris yang bekerja untuk GCHQ, menjabarkan tentang sistem equivalen pada dokumen internal pada tahun 1973. Penemuan Clifford Cocks tidak terungkap hingga tahun 1997 karena alasan top-secret classification[9].

Algoritma tersebut dipatenkan oleh Massachusetts Institute of Technology pada tahun 1983 di Amerika Serikat sebagai U.S. Patent 4.405.829. Paten tersebut berlaku hingga 21 September 2000. Semenjak Algoritma RSA dipublikasikan sebagai aplikasi paten, regulasi di sebagian besar negara-negara lain tidak memungkinkan penggunaan paten. Hal ini menyebabkan hasil temuan Clifford Cocks di kenal secara umum, paten di Amerika Serikat tidak dapat mematenkannya[9].

I. Perangkat Lunak Penunjang

1. JAVA NETBEAN
   Netbeans adalah salah satu aplikasi IDE yang digunakan programmer untuk menulis, mengompile, mencari kesalahan, dan menyebarkan program.netbeans ditulis dalam bahasa java namun dapat juga mendukung bahasa pemrogramman lain. program ini bebas digunakan.[5]
2. PHP (Personal Home Page)
   PHP adalah merupakan script untuk pemograman script web server-side, script yang membuat dokumen HTML secara on the fly, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML. [6]

3. Metodologi Penelitian

Metodologi yang digunakan dalam pembuatan tulisan ini adalah sebagai berikut :

Studi Literatur

Adalah usaha pencarian referensi teori yang relevan dengan topik permasalahan yang dibahas. Referensi tersebut berisikan tentang :

1. Pengertian Kriptografi
2. Metode Caesar *Cipher*
3. Metode Vigenere *Cipher*

Referensi ini dapat dicari dari hasil penelitian, situs *website*, dan lain – lain

Studi Pustaka

Studi pustaka adalah kegiatan membaca dan memahami tutorial, panduan – panduan serta keterangan yang bersumber dari buku – buku yang memaparkan secara terperinci mengenai teori – teori yang dapat digunakan untuk penelitian dan penulisan karya ilmiah.

4. Pembahasan

A. Sandi Caesar

Terdapat suatu data atau pesan yang akan disandikan dengan menggunakan algoritma Caesar. Teks data atau pesan awal yang akan di sandikan yaitu PESAN INI SANGAT RAHASIA. Berikut adalah proses Enkripsi dengan menggunakan algoritma Caesar.

Teks awal : PESAN INI SANGAT RAHASIA

Key : 12
Proses enkripsi :
Rumus E(P)=C, C=P+K Mod 26
Keterangan :
E(P) : Enkripsi
P : *Plaintext* (Teks Awal)
K : *Key* (Jumlah Pergeseran)

Teks / pesan sandi (*Ciphertext*) : BQEMZ UZU EMZSMF DMTMEUM

Proses dekripsi :
Rumus : D(C)=P, P=C-K mod 26
D(C) : Dekripsi
C : *Ciphertext* (Teks akhir)
K : *Key* (Jumlah Pergeseran)

Teks / pesan sandi (*Ciphertext*) : BQEMZ UZU EMZSMF DMTMEUM
Teks / pesan awal (*Plaintext*) : PESAN INI SANGAT RAHASIA

B. Sandi Vigenere

Terdapat suatu data atau pesan yang akan disandikan dengan menggunakan algoritma Vigenere. Teks data atau pesan awal yang akan di sandikan yaitu PESAN INI SANGAT RAHASIA dengan kunci ARMADA. berikut adalah proses Enkripsinya dengan algoritma vigenere.
Teks awal (Hasil dari Enkripsi Caesar)= PESAN INI SANGAT RAHASIA
Kunci (*Key*) = ARMADA

Proses Enkripsi :

Tabel 4.1 : Enkripsi Vigenere Cipher

Plaintext	P	E	S	A	N	I	N	I
Posisi abjad	15	4	18	0	13	8	13	8
Kunci (*Key*)	A	R	M	A	D	A	A	R
Posisi abjad	0	17	12	0	3	0	0	17
	15	21	30	0	16	8	13	25
Ciphertext	P	V	E	A	Q	I	N	Z

Tabel 4.2 : Enkripsi Vigenere Cipher

Plaintext	S	A	N	G	A	T	R	A
Posisi abjad	18	0	13	6	0	19	17	0
Kunci (*Key*)	M	A	D	A	A	R	M	A
Posisi abjad	12	0	3	0	0	17	12	0
	30	0	16	6	0	36	29	0
Ciphertext	E	A	Q	G	A	K	D	A

Tabel 4.3 : Enkripsi Vigenere Cipher

Plaintext	H	A	S	I	A
Posisi abjad	7	0	18	8	0
Kunci (*Key*)	D	A	A	R	M
Posisi abjad	3	0	0	17	12
	10	0	18	25	12
Ciphertext	K	A	S	Z	M

Hasil Enkripsi :
PVEAQ INZ EAQGAK DAKASZM

Proses dekripsi :
Rumus : D(C)=P, P=C-K mod 26
D(C) : Dekripsi
C : *Ciphertext* (Teks akhir)
K : *Key* (Kata / Kalimat)

Tabel 4.4 : Dekripsi Vigenere Cipher PESAN INI

Ciphertext	P	V	E	A	Q	I	N	Z
Posisi abjad	15	21	4	0	16	8	13	25
Kunci (*Key*)	A	R	M	A	D	A	A	R
Posisi abjad	0	17	12	0	3	0	0	17
	15	4	-8	0	13	8	13	8
Plaintext	P	E	S	A	N	I	N	I

Tabel 4.5 : Enkripsi Vigenere Cipher SANGAT

Plaintext	E	A	Q	G	A	K
Posisi abjad	4	0	16	6	0	10
Kunci (*Key*)	M	A	D	A	A	R
Posisi abjad	12	0	3	0	0	17
	-8	0	13	6	0	-7
Ciphertext	S	A	N	G	A	T

Tabel 4.6 : Enkripsi Vigenere Cipher RAHASIA

Plaintext	D	A	K	A	S	Z	M
Posisi abjad	3	0	10	0	18	25	12
Kunci (*Key*)	M	A	D	A	A	R	M
Posisi abjad	12	0	3	0	0	17	12
	-9	0	7	0	18	8	0
Ciphertext	R	A	H	A	S	I	A

Hasil Dekripsi : PESAN INI SANGAT RAHASIA

a. Kombinasi Sandi Caesar Dan Vigenere

Jika algoritma Caesar dan Vigenere kita kombinasikan, maka akan menghasilkan kekuatan enkripsi yang cukup kuat karena apabila terjadi penyadapan pesan hasil penyadapan tersebut masih dalam keadaan terenkripsi. Berikut contoh kombinasi dari sandi Caesar dan sandi Vigenere. :

1. Contoh ke-1
   Pesan akan disandikan :
   INDONESIA HARUS BANGKIT

a. Enkripsi dengan sandi Caesar
   Teks Awal : INDONESIA HARUS BANGKIT
   Kunci (*Key*) : 12
   Cipher Text : UZPAZQEUM TMDGE NMZSWUF

   Proses Enkripsi sandi Caesar :
   Rumus enkripsi : E(P)=C, C=P+K Mod 26

Tabel 4.7 Enkripsi *Plaintext* INDONESIA

Plaintext	I	N	D	O	N	E	S	I	A
Posisi abjad	8	13	3	14	13	4	18	8	0
Kunci (*Key*)	12	12	12	12	12	12	12	12	12
	20	25	15	26	25	16	30	20	12
Ciphertext	U	Z	P	A	Z	Q	E	U	M

Tabel 4.8 Enkripsi *Plaintext* HARUS

Plaintext	H	A	R	U	S
Posisi abjad	7	0	17	20	18
Kunci (*Key*)	12	12	12	12	12
	19	12	29	32	30
Ciphertext	T	M	D	G	E

Tabel 4.9 Enkripsi Plaintext BANGKIT

Plaintext	B	A	N	G	K	I	T
Posisi abjad	1	0	13	6	10	8	19
Kunci (*Key*)	12	12	12	12	12	12	12
	13	12	25	18	22	20	31
Ciphertext	N	M	Z	S	W	U	F

Hasil Enkripsi dengan sandi Caesar kemudian menjadi pesan awal yang akan disandikan kembali dengan sandi Vigenere.

b. Enkripsi dengan algoritma Vigenere
   Teks Awal : UZPAZQEUM TMDGE NMZSWUF
   Key : PERMATA
   Cipher Text : JDGMZJEJQ KYDZE CQQEWNF

   Proses Enkripsi sandi Vigenere :
   Rumus enkripsi : E(P)=C, C=P+K Mod 26

Tabel 4.10 *Plaintext* Hasil Caesar INDONESIA

Plaintext	U	Z	P	A	Z	Q	E	U	M
Posisi abjad	17	25	15	0	25	16	4	20	12
Kunci (*Key*)	P	E	R	M	A	T	A	P	E
Posisi abjad	15	4	17	12	0	19	0	15	4
	32	29	32	12	25	35	4	35	16
Ciphertext	J	D	G	M	Z	J	E	J	Q

Tabel 4.11 *Plaintext* Hasil Caesar HARUS

Plaintext	T	M	D	G	E
Posisi abjad	19	12	3	6	4
Kunci (*Key*)	R	M	A	T	A
Posisi abjad	17	12	0	19	0
	36	24	3	25	4
Ciphertext	K	Y	D	Z	E

Tabel 4.12 *Plaintext* Hasil Caesar BANGKIT

Plaintext	N	M	Z	S	W	U	F
Posisi abjad	13	12	25	18	22	20	5
Kunci	P	E	R	M	A	T	A

(*Key*)							
Posisi abjad	15	4	17	12	0	19	0
	28	16	32	30	22	39	5
Ciphertext	C	Q	Q	E	W	N	F

Hasil akhir yang diperoleh adalah berupa pesan yang telah disandikan dengan sandi Vigenere.

2. Contoh ke-2
   Pesan akan disandikan :
   BERSAMA KITA BISA

a. Enkripsi dengan sandi Caesar
   Teks Awal : BERSAMA KITA BISA
   Kunci (Key) : 8
   Cipher Text : JMZAIUI SQBI JQAI

   Proses Enkripsi sandi Caesar :
   Rumus enkripsi : E(P)=C, C=P+K Mod 26

Tabel 4.13 Enkripsi *Plaintext* BERSAMA

Plaintext	B	E	R	S	A	M	A
Posisi abjad	1	4	17	18	0	12	0
Kunci (*Key*)	8	8	8	8	8	8	8
	9	12	25	26	8	20	8
Ciphertext	J	M	Z	A	I	U	I

Tabel 4.14 Enkripsi *Plaintext* KITA

Plaintext	K	I	T	A
Posisi abjad	10	8	19	0
Kunci (*Key*)	8	8	8	8
	18	16	27	8
Ciphertext	S	Q	B	I

Tabel 4.15 Enkripsi *Plaintext* BISA

Plaintext	B	I	S	A
Posisi abjad	1	8	18	0
Kunci (*Key*)	8	8	8	8
	9	16	26	8
Ciphertext	J	Q	A	I

Kemudian hasil sandi Caesar disandikan lagi dengan sandi Vigenere.

b. Enkripsi dengan algoritma Vigenere
   Teks Awal : JMZAIUI SQBI JQAI
   Key : KEKUATAN
   Cipher Text : TQJUINI FAFS DQTI

Tabel 4.16 *Plaintext* Hasil Caesar BERSAMA

Plaintext	J	M	Z	A	I	U	I
Posisi abjad	9	12	25	0	8	20	8
Kunci (*Key*)	K	E	K	U	A	T	A
Posisi abjad	10	4	10	20	0	19	0
	19	16	35	20	8	39	8
Ciphertext	T	Q	J	U	I	N	I

Tabel 4.17 *Plaintext* Hasil Caesar KITA

Plaintext	S	Q	B	I
Posisi abjad	18	16	1	8
Kunci (*Key*)	N	K	E	K
Posisi abjad	13	10	4	10
	31	26	5	18
Ciphertext	F	A	F	S

Tabel 4.18 *Plaintext* Hasil Caesar Bisa

Plaintext	J	Q	A	I
Posisi abjad	9	16	0	8
Kunci (*Key*)	U	A	T	A
Posisi abjad	20	0	19	0
	29	16	19	8
Ciphertext	D	Q	T	I

Kemudian hasil proses dari kombinasi dengan sandi Caesar dan Vigenere pesan sudah siap dikirimkan dengan memanfaatkan layanan surat elektronik.

Untuk penerima pesan yang telah disandikan tersebut, harus melakukan proses dekripsi atau mengembalikan pesan yang tersandikan menjadi pesan awal sebelum kombinasi Enkripsi pesan dilakukan sehingga isi pesan tersebut dapat dibaca dan dipahami.

Berikut adalah proses dekripsinya.

1. *Ciphertext* diambil dari contoh ke-1 hasil dari sandi Vigenere.

   Rumus dekripsi : D(C)=P, P=C-K mod 26

a. Dekripsi dengan sandi Vigenere
Cipher Text : JDGMZJEJQ KYDZE CQQEWNF
Key : PERMATA
Plaintext : UZPAZQEUM TMDGE NMZSWUF

Tabel 4.19 Hasil Dekripsi *Ciphertext* Vigenere kata INDONESIA

Ciphertext	J	D	G	M	Z	J	E	J	Q
Posisi abjad	9	3	6	12	25	9	4	9	16
Kunci (*Key*)	P	E	R	M	A	T	A	P	E
Posisi abjad	15	4	17	12	0	19	0	15	4
	-6	-1	-11	0	25	-10	4	-6	12
Plaintext	U	Z	P	A	Z	Q	E	U	M

Tabel 4.20 Dekripsi *Ciphertext* Vigenere kata HARUS

Ciphertext	K	Y	D	Z	E
Posisi abjad	10	24	3	25	4
Kunci (*Key*)	R	M	A	T	A
Posisi abjad	17	12	0	19	0
	-17	12	3	6	4
Plaintext	T	M	D	G	E

Tabel 4.21 Dekripsi *Ciphertext* Vigenere kata BANGKIT

Ciphertext	C	Q	Q	E	W	N	F
Posisi abjad	2	16	16	4	22	13	5
Kunci (*Key*)	P	E	R	M	A	T	A
Posisi abjad	15	4	17	12	0	19	0
	-13	12	-1	-8	22	-6	5
Plaintext	N	M	Z	S	W	U	F

b. Dekripsi dengan sandi Caesar
Proses Enkripsi sandi Caesar :
Rumus enkripsi : D(C)=P, P=C-K Mod 26
Teks Awal : UZPAZQEUM TMDGE NMZSWUF
Kunci (*Key*) : 12
Plaintext : INDONESIA HARUS BANGKIT

Tabel 4.22 Dekripsi *Ciphertext* Caesar kata INDONESIA

Ciphertext	U	Z	P	A	Z	Q	E	U	M
Posisi abjad	20	25	15	0	25	16	4	20	12
Kunci (*Key*)	12	12	12	12	12	12	12	12	12
	8	13	3	-12	13	4	-8	12	0
Plaintext	I	N	D	O	N	E	S	I	A

Tabel 4.23 Dekripsi *Ciphertext* Caesar kata HARUS

Ciphertext	T	M	D	G	E
Posisi abjad	19	12	3	6	4
Kunci (*Key*)	12	12	12	12	12
	7	0	-9	-6	-8
Plaintext	H	A	R	U	S

Tabel 4.24 Dekripsi *Ciphertext* Caesar kata BANGKIT

Ciphertext	N	M	Z	S	W	U	F
Posisi abjad	13	12	25	18	22	20	5
Kunci (*Key*)	12	12	12	12	12	12	12
	1	0	13	6	10	8	-7
Plaintext	B	A	N	G	K	I	T

Berikut adalah proses dekripsi dari comtoh yang ke-2

2. Ciphertext diambil dari contoh ke-2 hasil sandi Vigenere.

Rumus dekripsi : D(C)=P, P=C-K mod 26

a. Dekripsi dengan sandi Vigenere
Ciphertext : TQJUINI FAFS DQTI
Key : KEKUATAN
Plaintext : JMZAIUI SQBI JQAI

Tabel 4.25 *Plaintext* Hasil Vigenere BERSAMA

Ciphertext	T	Q	J	U	I	N	I
Posisi abjad	19	16	9	20	8	13	8
Kunci	K	E	K	U	A	T	A

(*Key*)							
Posisi abjad	10	4	10	20	0	19	0
	9	12	-1	0	8	-6	8
Plaintext	J	M	Z	A	I	U	I

Tabel 4.26 *Plaintext* Hasil Caesar KITA

Ciphertext	F	A	F	S
Posisi abjad	5	0	5	18
Kunci (*Key*)	N	K	E	K
Posisi abjad	13	10	4	10
	-8	-10	1	8
Plaintext	S	Q	B	I

Tabel 4.27 *Plaintext* Hasil Caesar BISA

Ciphertext	D	Q	T	I
Posisi abjad	3	16	19	8
Kunci (*Key*)	U	A	T	A
Posisi abjad	20	0	19	0
	-17	16	0	8
Plaintext	J	Q	A	I

b. Dekripsi dengan sandi Caesar
Proses Enkripsi sandi Caesar :
Rumus enkripsi : D(C)=P, P=C-K Mod 26

Teks Awal : JMZAIUI SQBI JQAI
Kunci (*Key*) : 8
Plaintext : BERSAMA KITA BISA

Tabel 4.28 Dekripsi *Ciphertext* Caesar kata BERSAMA

Ciphertext	J	M	Z	A	I	U	I
Posisi abjad	9	12	25	0	8	20	8
Kunci (*Key*)	8	8	8	8	8	8	8
	1	4	17	-8	0	12	0
Plaintext	B	E	R	S	A	M	A

Tabel 4.29 Dekripsi *Ciphertext* Caesar kata KITA

Ciphertext	S	Q	B	I
Posisi abjad	18	16	1	8
Kunci (*Key*)	8	8	8	8
	10	8	-7	0
Plaintext	K	I	T	A

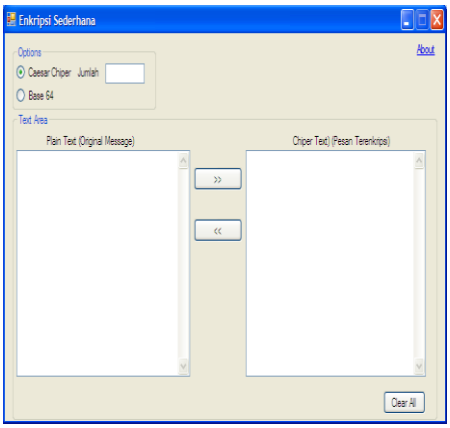
Tabel 4.30 Dekripsi *Ciphertext* Caesar kata BISA

Ciphertext	J	Q	A	I
Posisi abjad	9	16	0	8
Kunci (*Key*)	8	8	8	8
	1	8	-8	0
Plaintext	B	I	S	A

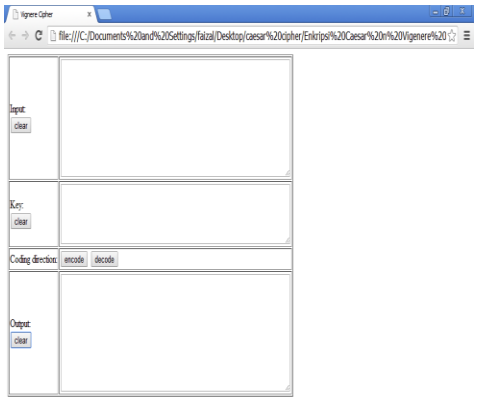
Pesan yang telah di dekripsi dari kombinasi sandi vigenere dan Caesar sudah dapat di baca dan di pahami oleh penerima pesan. Kerahasiaan dan keaslian pesan terjaga sampai kepada penerima.

b. Tampilan Aplikasi

Setelah diketahui bagaimana proses Enkripsi dengan mengkombinasikan kedua algoritma yaitu algoritma Caesar dan Algoritma Vigenere secara tertulis, maka perlu ditransformasikan kedalam bahasa pemprograman agar pemanfaatannya menjadi lebih mudah. Berikut adalah Tampilan aplikasi dari algoritma Caesar dan Algoritma Vigenere. Masing – masing aplikasi dibuat dengan bahasa pemprograman yang berbeda.



Gambar 4.1. Aplikasi Sandi Caesar



Gambar 4.2. Aplikasi Sandi Vigenere

5. Kesimpulan

Dalam membuat pesan surat elektronik melalui akses internet ada baiknya isi dari pesan tersebut dijaga kerahasiaannya agar hanya pengirim dan penerima saja yang dapat membaca isi pesan surat elektronik tersebut.

Menjaga kerahasiaan isi pesan tersebut dapat menggunakan algoritma sandi Caesar dan dikombinasikan dengan algoritma sandi Vigenere agar keamanan isi pesan lebih kuat sehingga seandainya pesan yang dikirimkan dibajak atau disadap oleh penggangu maka isi pesan tersebut tetap terlindungi dan menyulitkan si pembajak untuk mengetahuinya.

Untuk bisa mengamankan pesan surat elektronik tersebut maka algoritma sandi Caesar dan Vigenere ini di kombinasikan dan diterapkan menjadi suatu aplikasi sehingga dapat mudah digunakan.

Daftar Pustaka

[1] LSN. *Jelajah Kriptologi.* Jakarta : LSN
[2] Munir, Rinaldi. 2007. *Kriptografi.* Bandung : Informatika
[3] http://codeindesign.com/dasar-kriptografi-enkripsi-dan-dekripsi/
[4] Bishop, David. Introduction to Cryptography with Java Applets. Grinnell College. 2003.
[5] http://www. biebah-site34.blogspot.com/2013/05/ tentang-netbeans.html
[6] Sidik, Ir, Betha. 2005. MySQL untuk Pengguna, Administrator, dan Pengembang Aplikasi Web. Bandung: Informatika
[7]http://www .id.wikipedia.org/wiki/Sandi_Playfair
[8] http://id.wikipedia.org/wiki/Blowfish_(cipher)
[9] http://id.wikipedia.org/wiki/RSA