Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)
Volume (3) Juli 2018, pp 73-86
ISSN: 2527-5771/EISSN: 2549-7839
http://tunasbangsa.ac.id/ejurnal/index.php/jurasik

IMPLEMENTASI METODE ALGORITMA KVC UNTUK PENGAMANAN PESAN

Yunita Achsanti[1], Heru Abrianto[2], Ninuk Wiliani[3]
Program Studi Teknik Elektro/Telekomunikasi, Fakultas Teknologi Industri
Program Studi Teknik Informatika, Fakultas Sains dan Teknologi Industri
Institut Sains dan Teknologi Nasional
Jl. M.Kahfi II Bhumi Srengseng Indah, Jagakarsa, Jakarta Selatan, telp 021-7270090
yunita.achsanty@gmail.com, heru@istn.ac.id, ninukwiliani15@gmail.com

Abstract

Communication through message media or sms, Sort Message Service is not a point to point message but that communication by sent through sms network and stored in database operator. Message security on the network is threatened be read by people who are not responsible or called intercept. Therefore it will be develop some application on the mobile phone to modify the message be come a Ciphertext and the information content of the messageis not known by others. In this application, the system encrypts the message into Ciphertext using the key entered by the sender and then it will be sends to the destination number. For accepting the message, system will decrypt chipertext become plaintext using key by sender, then the message can be read by receiver. This application can be used by some one who wants to sending the secret message and very important without fear if the information can read by others. This application uses Vigenere Cipher method. The main parameter is the key and the message itself.

Keywords*: Sort Message Service (SMS), Encryption, Decryption, Cryptography, Vigenere Cipher****.***

Abstrak

Komunikasi melalui media pesan atau SMS, Sort Message Service bukan merupakan pesan point-to-poin melainkan komunikasi yang dikirimkan melalui jaringan SMS dan tersimpan pada database operator yang bersangkutan. Pada jaringan tersebut, keamanan pesan sangat terancam untuk dibaca oleh orang yang tidak bertanggung jawab atau yang populer dengan istilah penyadapan. Oleh karena itu akan dikembangkan sebuah aplikasi pada telepon selular untuk memodifikasi pesan menjadi Ciphertext agar isi informasi dari pesan tersebut tidak diketahui oleh orang lain. Dalam aplikasi ini, sistem mengenkripsi pesan menjadi Ciphertext menggunakan kunci yang diinputkan oleh pengirim kemudian mengirimkan ke nomor tujuan. Untuk penerimaan pesan, sistem mendekripsi Ciphertext menjadi plaintext menggunakan key yang diinputkan oleh penerima kemudian menampilkan pesan asli kepada penerima. Aplikasi ini dapat dimanfaatkan oleh seseorang yang ingin mengirimkan suatu informasi rahasia kepada orang lain melalui SMS tanpa takut informasi dari pesan tersebut akan diketahui oleh orang lain. Metode yang digunakan aplikasi dalam mengenkripsi dan mendekripsi pesan adalah metode algoritma kriptografi Vigenere Cipher (KVC) dengan parameter utamanya adalah kunci, dan pesan itu sendiri.

Kata Kunci: *Sort message service (SMS), enkripsi, dekripsi, kriptografi, Vigenere Cipher.*

1. PENDAHULUAN

Perkembangan teknologi di bidang telekomunikasi khususnya dalam perkembangan teknologi seluler terlihat sangat pesat. Kini, perangkat seluler ada yang telah memiliki spesifikasi setara dengan notebook dan netbook. Sehingga sebagian besar kegiatan yang dilakukan di perangkat notebook dan netbook, kini bisa dilakukan di perangkat seluler. Perangkat ini biasa disebut Smartphone. Oleh karenanya, masyarakat lebih memilih menggunakan smartphone untuk mendukung aktifitas hariannya karena lebih simpel, efektif dan efisien. Sejalan

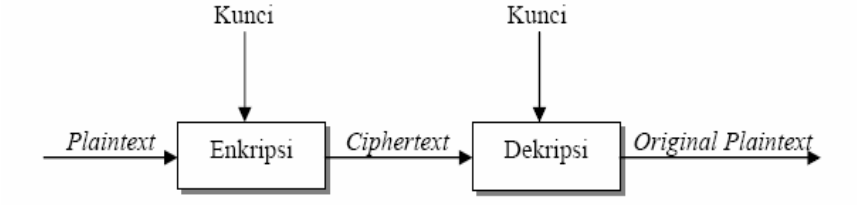
dengan itu semua, kejahatan di dunia cyber makin merebak. Salah satu sasarannya yaitu media perpesanan atau SMS. Komunikasi melalui media pesan atau SMS ini bukanlah komunikasi *point-to-point*. Pesan yang dikirimkan melalui media pesan tidak langsung sampai pada tujuan, melainkan melalui jaringan SMS dan tersimpan pada database operator yang bersangkutan. Pada jaringan SMS tersebut, keamanan pesan sangatlah terancam untuk dibaca oleh orang yang tidak bertanggung jawab atau yang populer dengan istilah penyadapan[7]. Untuk membantu pengguna dalam mengamankan pesannya, agar tidak ada pihak yang tidak dikehendaki yang dapat membacanya, akan dikembangkan sebuah aplikasi dengan menggunakan metode Algoritma *Vigenere* yang bertujuan untuk mengamankan sebuah pesan terkirim.

Short Message Service (SMS) merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel, memungkinkan dilakukannya pengiriman pesan dalam bentuk *alphanumeric* antara terminal pelanggan atau antara terminal pelanggan dengan sistem eksternal seperti *email, paging, voice mail*, dan lain - lain. Disebut pesan teks pendek karena pesan yang dikirimkan hanya berupa karakter[4]. Teknologi yang mendukung SMS antara lain adalah *Global System for Mobile* (GSM), *Time Division Multiple Access* (TDMA) dan *Code Division Multiple Access* (CDMA). Dengan didukung oleh ketiga teknologi ini, SMS telah menjadi layanan data bergerak yang bersifat *universal*.

2. METODOLOGI PENELITIAN

2.1. Kriptografi

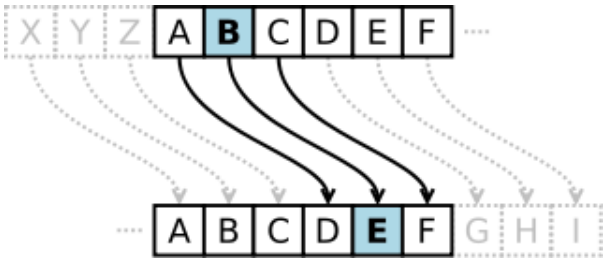
Kriptografi adalah sebuah ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalamigangguan dari pihak ketiga. Menurut Bruce Scheiner dalam bukunya "*AppliedCryptography*", kriptografi adalah ilmu pengetahuan dan seni menjaga pesan (informasi)agar tetap aman (*secure*)[2]. Pada bidang kriptografi terdapat istilah-istilah yang digunakan, seperti *plaintext, chipertext, enkripsi*, dekripsi, dan kunci. Dimana, Plaintext (M) adalah pesan yang hendak dikirimkan (berisi data asli), Chipertext (C) adalah pesan ter-enkrip (tersandi) yang merupakan hasil *enkripsi*, *Enkripsi* adalah proses pengubahan *plaintext* menjadi *Ciphertext*, Dekripsi adalah kebalikan dari *enkripsi* yakni mengubah *Ciphertext* menjadi *plaintext*, sehingga berupa data awal/asli, dan Kunci adalah suatu bilangan yang dirahasiakan yang digunakan dalam proses *enkripsi* dan dekripsi. Hal-hal tersebut merupakan dasar dari struktur *enkripsi* dekripsi seperti pada gambar 1[5].



Gambar 1. Proses *Kriptografi*

2.2. *Algoritma Kriptografi Caesar Cipher*

Nama Sandi Caesar diambil dari Julius Caesar, yang menurut buku Suetonius *Kehidupan Dua belas Caesar*, menggunakan sandi ini dengan geseran tiga, untuk mengirim pesan yang mengandung rahasia atau taktik militer.
Caesar *Cipher* sering juga disebut dengan shift *Cipher*, karena dasar dari algoritma ini adalah dengan menggeser beberapa karakter dari plaintext.



Gambar 2. *Metode Kriptografi Caesar Cipher*

Pada gambar 2 merupakan contoh dari metode kriptografi caesar *Cipher* dengan menggeser 3 karakter. Misalnya, plaintext pada huruf A jika digeser 3 karakter maka menghasilkan *Cipher*text huruf D, plaintext pada huruf B jika digeser 3 karakter maka menghasilkan *Cipher*text huruf E, plaintext pada huruf C jika digeser 3 karakter maka menghasilkan *Cipher*text huruf F, dan begitu seterusnya.

2.3. Enkripsi Kriptografi Caesar Cipher

Untuk melakukan *enkripsi* dari caesar *Cipher*, dapat diambil rumus untuk *enkripsi*nya adalah sebagai berikut:
E(x) = (x+key) mod 25
Nilai mod 25, bukan mod 26 karena index karakter memulai dari 0. (A = 0, B = 1, C = 2, … , Z = 25)

2.4. Deskripsi Kriptografi Caesar Cipher

Untuk melakukan dekripsi dari caesar *Cipher*, harus melakukan pembalikan rumus *enkripsi*nya[3]:
Dari **E(x) = (x+key) mod 25**
Menjadi **D(x) = (x-key) mod 25**

2.5. Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi. Awalnya, Google Inc. Mengakuisisi Android Inc. Yang merupakan pendatang baru sebagai pembuat piranti lunak ponsel atau *smartphone*. Saat ini Android diklaim sebagai *platform mobile* pertama yang lengkap. Terbuka, dan bebas[6].

Untuk mengimplementasikan algoritma pengacak pesan, maka dibangun aplikasi *Secure SMS*. Aplikasi tersebut menggunakan metode algoritma KVC yang berperan untuk meng*enkripsi* dan mendeskripsi pesan.
Sebelum membangun aplikasi *Secure SMS* ini, terlebih dahulu harus mengetahui tentang apa saja yang diperlukan untuk membuat aplikasi tersebut. Software dapat di download sendiri tanpa harus membelinya. Berikut peralatan atau software yang digunakan untuk membangun aplikasi tersebut.

Software pendukung yang digunakan untuk membuat aplikasi adalah eclipse juno versi 3.8 windows 32bit, android sdk versi 24.1.1, perangkat PC dengan minimal RAM 2GB & minimal free HDD 50GB, dan java jdk minimal versi 7 untuk bisa menjalankan eclipse juno seperti dijelaskan pada tabel 1.

Tabel 1. Software Pendukung

Equipment	Version	Size	Keterangan
Eclipse Juno	3.8 windows 32bit	179 MB	Release 2012
Android SDK	24.1.1	149 MB	149 MB – 50 GB
Perangkat PC	Windows 7 32 bit Intel core 2 duo	Min RAM 2 GB Min free HDD 50GB	Free HDD mengacu pada besaran size android SDK
Java JDK	Java versi 7, x86	127 MB	Di atas versi 7

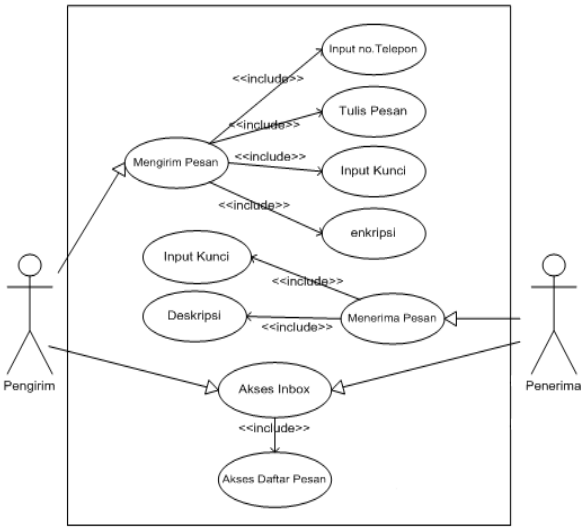
Penelitian ini menggunakan dua buah perangkat untuk pengujian. Satu perangkat sebagai pengirim dan perangkat lainnya sebagai penerima. Perangkat yang digunakan sebagai pengirim adalah xiomi redmi 4x dan perangkat yang digunakan sebagai penerima adalah LG G3 dengan spesifikasi minimum seperti yang dijelaskan pada tabel 2.

Tabel 2. Spesifikasi perangkat Android yang digunakan

Spesifikasi	Xiaomi Redmi 4x (Pengirim)	LG G3 (Penerima)	Spesifikasi Minimum
Layar	IPS-LCD 5.0 inches, 720x 1280 pixels	IPS-LCD 5,5 inches, 1440x2560 pixels	3,0 inches 240x320 pixels
CPU	Qualcomm MSM8940 Snapdragon 435 Octa-core 1,4 GHz Cortex-A53	Qualcomm MSM8974 AC Snapdragon 801 Quad-core 2.5 GHz Krait 400	512 MHz ARMv6
Memori	32 GB, 3 GB RAM	16 GB, 2GB RAM	1 GB, 512 MB RAM
OS	AndroidOS v.6.0 (Marsmallow)	AndroidOS v 5.0 (Lolipop)	AndroidOS v.2.3.3 (Gingerbread)

Hal–hal yang dapat dilakukan pengguna terhadap sistem atau aplikasi *Secure SMS* dijelaskan dalam gambar 3. Hal yang dapat dilakukan pengirim adalah mengirimkan pesan yang telah di *enkripsi* ke nomor tujuan *include* tulis pesan, input kontak, input kunci, dan *enkripsi* pesan serta melihat dan membaca pesan di dalam kotak masuk. Hal yang dapat dilakukan penerima adalah mendekripsikan

terlebih dahulu pesan yang di terima dalam bentuk ter*enkripsi include* input kunci dan dekripsi pesan serta melihat dan membaca pesan di dalam kotak masuk



Gambar 3. Interaksi Diagram aplikasi

2.6. Metode Algoritma Kriptografi Vigenere Cipher

Vigenere Cipher merupakan bentuk pengembangan dari Caesar *Cipher*. Kelebihan sandi ini dibanding Caesar *Cipher* adalah *Cipher* ini tidak begitu rentan terhadap metode pemecahan *Cipher*. *Vigènere Cipher* menggunakan Bujursangkar *Vigènere* untuk melakukan *enkripsi* dan dekripsi. Jika pada Caesar *Cipher* setiap huruf digeser dengan besar geseran yang sama, maka pada *Vigènere Cipher* setiap huruf digeser dengan besar yang berbeda sesuai dengan kuncinya.

Algoritma *enkripsi* jenis ini sangat dikenal karena mudah dipahami dan diimplementasikan. Teknik untuk menghasilkan *Cipher*text bisa dilakukan menggunakan substitusi angka maupun bujursangkar *Vigenere*. Teknik susbtitusi *Vigenere* dengan menggunakan angka dilakukan dengan menukarkan huruf dengan angka secara urut, di mulai dari huruf A-Z dan angka 0-25 seperti pada tabel 3.3.

Tabel 3. Substitusi Algoritma Kriptografi *Vigenere Cipher*

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

Sedangkan teknik lain untuk melakukan proses *enkripsi* dengan metode *Vigenere Cipher* yaitu menggunakan tabula recta (disebut juga bujursangkar *Vigenere*) seperti pada gambar 4. Kolom paling kiri dari bujursangkar menyatakan huruf-huruf kunci, sedangkan baris paling atas menyatakan huruf-huruf plaintext. Setiap baris di dalam bujursangkar menyatakan huruf-huruf *Cipher*text yang diperoleh

dengan Caesar *Cipher*, yang mana jumlah pergeseran huruf plaintext ditentukan nilai numerik huruf kunci tersebut (yaitu, a=0, b=1, c=2, …, z=25).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar 4. Tabula Recta Algoritma Kriptografi *Vigenere Cipher*

2.7. Perhitungan Manual Algoritma KVC

Pada perhitungan manual ini diberikan contoh kunci dan *plaintext* sebesar 19 karakter. Dengan menggunakan rumus sebagai berikut :

$$Ci = (Pt + K) \bmod 26$$
$$Pt = (Ci - K) \bmod 26$$

Dimana:
Ci = nilai desimal karakter *Cipher* text ke-n
Pt = nilai desimal karakter plaintext ke-n
K = nilai desimal karakter kunci ke-n
Mod 26 = akan dilakukan perulangan jika melebihi 26

Plaintext **: SAYA YUNITA ACHSANTY MAHASISWI ISTN CIKINI YANG INGIN LULUS CUMLAUDE**
Key **: AMIN**

Proses *Enkripsi* Algoritma Kriptografi *Vigenere Cipher*

1. Menyamakan panjang *key* dengan panjang *plaintext*.

Plaintext	S	A	Y	A	Y	U	N	I	T	A	A	C	H	S	A	N	T	Y
Key	A	M	I	N	A	M	I	N	A	M	I	N	A	M	I	N	A	M

2. Mengubah setiap huruf menjadi bilangan angka dan memasukannya ke dalam rumus.

Plaintext	S	A	Y	A	
Desimal Plaintext	18	0	24	0	
Key	A	M	I	N	
Desimal Key	0	12	8	13	

a. Ci(S) = (Pt + K) mod 26
   Ci(S) = (18 + 0) mod 26
   Ci(S)= (18) = **S**
b. Ci(A) = (Pt + K) mod 26
   Ci(A) = (0 + 12) mod 26
   Ci(A) = (12) = **M**
c. Ci(Y) = (Pt + K) mod 26
   Ci (Y) = (24 + 8) mod 26
   Ci (Y) = (32) mod 26
   Ci (Y) = (32 – 26)
   Ci (Y) = (6) = **G**
d. Ci(A) = (Pt + K) mod 26
   Ci(A) = (0 + 13) mod 26
   Ci(A) = (12) = **N**

3. Maka didapatkan hasil *Ciphertext* adalah **SMGNYGVVTMIPHEIATK UNHMAVSIQVSFVPIWQAIKIAGUVTIZTHLGAPUYTNUPM**.

Proses Deskripsi Algoritma Kriptografi *Vigenere Cipher*

1) Menyamakan panjang *key* dengan panjang *Ciphertext*.

Plaintext	S	M	G	N	Y	G	V	V	T	M	I	P	H	E	I	A	T	K
Key	A	M	I	N	A	M	I	N	A	M	I	N	A	M	I	N	A	M

2) Mengubah setiap huruf menjadi bilangan angka dan memasukannya ke dalam rumus.

Plaintext	S	M	G	N	Y	G	V	V	T	M	I	P	H	E	I	A	T	K	U	N	H	M	A	V	S	I	Q
Desimal Plaintext	18	12	6	13	24	6	21	21	19	12	8	15	7	4	8	0	19	10	20	13	7	12	0	21	18	8	16
Key	A	M	I	N	A	M	I	N	A	M	I	N	A	M	I	N	A	M	I	N	A	M	I	N	A	M	I
Desimal Key	0	12	8	13	0	12	8	13	0	12	8	13	0	12	8	13	0	12	8	13	0	12	8	13	0	12	8

a. Pt (S) = (Ci - K) mod 26
   Pt (S) = (18 – 0) mod 26
   Pt (S) = (18) = **S**
b. Pt (M) = (Ci - K) mod 26
   Pt (M) = (12 – 12) mod 26
   Pt (M) = (0) = **A**
c. Pt (G) = (Ci - K) mod 26
   Pt (G) = (6 – 8) mod 26
   Pt (G) = (-2) mod 26
   Pt (G) = (-2 + 26)
   Pt (G) = (24) = **Y**
d. Pt (N) = (Ci - K) mod 26
   Pt (N) = (13 – 13) mod 26
   Pt (N) = (0) = **A**

3) Maka didapatkan hasil *plaintext* adalah **SAYA YUNITA ACHSANTY MAHASISWI ISTN CIKINI YANG INGIN LULUS CUMLAUDE.**

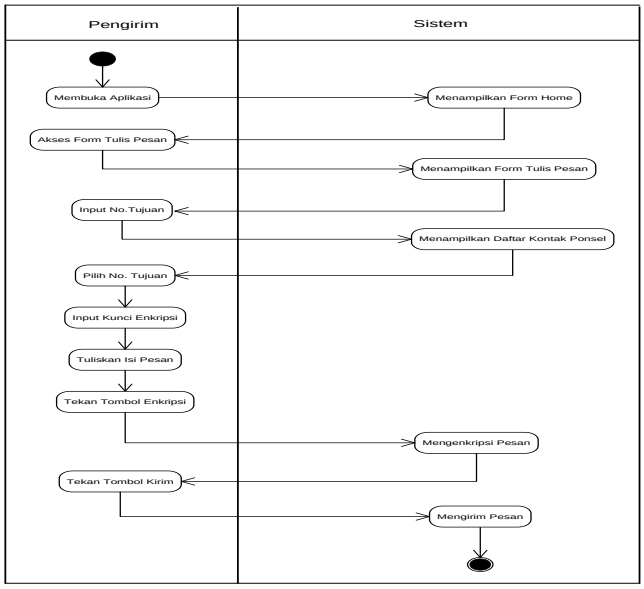
3. HASIL DAN PEMBAHASAN

3.1. Desain

Pada setiap perancangan aplikasi, pasti akan menyinggung perihal desain. Desain disini lebih kepada pembahasan tentang bagaimana interaksi yang akan dibangun antara user dengan aplikasi, atau yang biasa disebut dengan *user interface.*

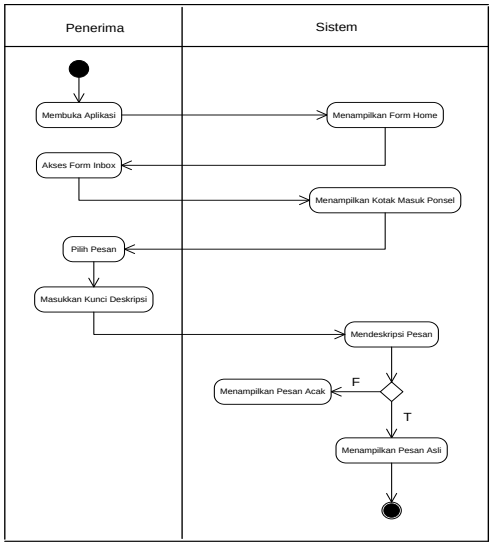
3.2. Rancangan Proses

Pada *Activity* Diagram 5 diperlihatkan aktifitas yang dilakukan oleh pengirim terhadap sistem pada aplikasi *Secure SMS.* Mulai dari pengirim membuka aplikasi, mengakses form tulis pesan, input nomor tujuan, *input* kunci *enkripsi*, menuliskan pesan, meng*enkripsi* dan mengirim pesan atau SMS, sampai dengan pesan itu diterima oleh penerima.



Gambar 5. *Activity* Diagram Pengirim

Pada *Activity* Diagram 6. diperlihatkan aktifitas yang dilakukan oleh penerima terhadap sistem pada aplikasi *Secure SMS.* Mulai dari penerima membuka aplikasi, mengakses form *inbox*, memilih pesan yang akan dideskripsikan, *input* kunci deskripsi, sampai dengan pesan itu menampilkan pesan aslinya.



Gambar 6. *Activity* DiagramPenerima

3.3. Pengujian Proses

Untuk lebih memastikan apakah algoritma yang dihitung manual berjalan sama persis pada aplikasi, maka dilakukan pembuktian pada aplikasinya langsung.

3.3.1. Pengujian *Enkripsi* Pesan

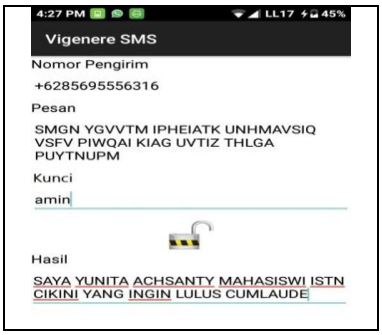
Pengujian dilakukan dengan meng-input nomor penerimanya, memasukan kunci dan isi pesan. Untuk **kunci** itu sendiri **ditentukan atas dasar kesepakatan pihak pengirim dan penerima**. Karena bila kunci berbeda maka kelak penerima tidak akan bisa mendeskripsikan pesan tersebut. Tombol *enkripsi* (*icon* gembok) akan bekerja bila pesan dan kunci sudah terisi. Begitu *icon* gembok diklik, maka hasil *enkripsi* akan tertampil dibawahnya. Lalu bisa mengirimkan pesan dengan menyentuh icon Kirim Pesan.

Gambar 7. Tampilan *enkripsi* pesan

Gambar 7 merupakan hasil dari perhitungan manual yang didapatkan hasil *enkripsi* dari pesan "Saya yunita achsanty mahasiswi istn cikini yang ingin lulus cumlaude" dengan kunci "Amin" adalah SMGN YGVVTM IPHEIATK UNHMAVSIQ VSFV PIWQAI KIAG UVTIZ THLGA PUYTNUPM.

3.3.2. Pengujian Deskripsi Pesan

Pesan yang telah terkirim akan sampai ke penerima. Penerima harus memiliki aplikasi yang sama dan mengetahui kunci yang sama untuk melakukan proses deskripsi. Untuk mengakses baca pesan dapat dilakukan dengan sentuh icon inbox pada halaman utama*.* Kemudian memilih pesan yang dimaksud, maka akan tampil seperti dibawah ini.



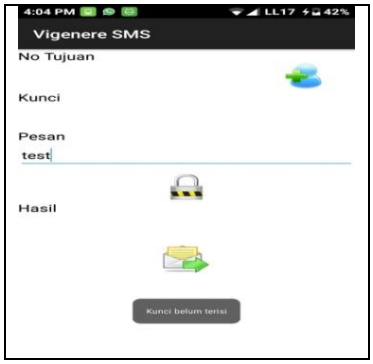
Gambar 8. Tampilan deskripsi pesan

Secara *default*, user hanya harus menginputkan kuncinya. Dan jika tombol deskripsi (gembok terbuka) disentuh, maka hasil deskripsi akan tampil dibawahnya. Gambar 8 merupakan bukti dari perhitungan manual didapatkan hasil deskripsi dari pesan "Smgn ygvvtm ipheiatk unhmavsiq vsfv piwqai kiag uvtiz thlga puytnupm" dengan kunci "Amin" adalah SAYA YUNITA ACHSANTY MAHASISWI ISTN CIKINI YANG INGIN LULUS CUMLAUDE.

3.4. Pengujian Kasus

Proses pengujian dilakukan untuk mengetahui apakah data yang dimasukkan (*input*) sudah sesuai dengan yang diharapkan (*output*).

3.4.1. Pengujian Ketika Kunci Belum Terisi



Gambar 9. Pesan peringatan kunci pesan belum terisi

Ketika kunci belum terisi namun isi pesan sudah terisi, maka saat *klik* tombol *enkripsi* (gembok), akan didapatkan pesan peringatan seperti pada gambar 9 *"Kunci belum terisi"*.

3.4.2. Pengujian Ketika Isi Pesan Belum Terisi

Gambar 10. Pesan peringatan isi pesan belum terisi

Ketika isi pesan belum terisi namun kunci sudah terisi, maka saat *klik* tombol *enkripsi* (gembok), akan didapatkan pesan peringatan seperti pada gambar 10

3.4.3. Pengujian Ketika Mengirim Pesan Tanpa Di *Enkripsi*

Gambar 11. Pesan peringatan mengirim pesan tanpa di *enkripsi*

Ketika isi pesan dan kunci sudah terisi namun belum di *enkripsi*, makasaat *klik* tombol *send*, akan didapatkan pesan peringatan seperti pada gambar 11.

3.4.4. Pengujian Ketika Salah Memasukan Kunci

Gambar 12. Salah memasukkan kunci

Ketika pesan hendak didekripsi, saat input kunci namun kunci yang dimasukan salah, maka akan didapatkan pesan yang sama seperti pada gambar 12 Pesan tidak berubah / masih ter*enkripsi*.

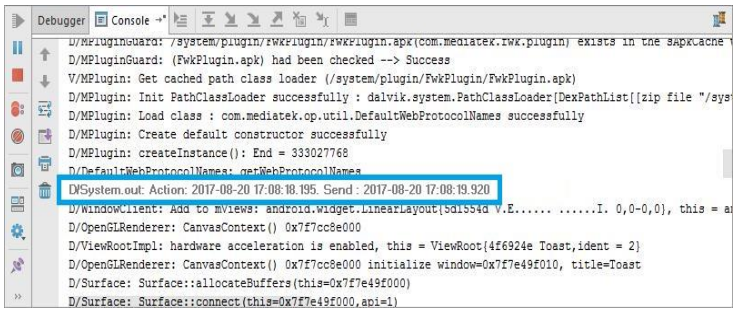
3.5. Pengujian Kapasitas SMS & Delay

Proses pengiriman SMS pasti besinggungan dengan berapa besar data yang dapat dikirim dalam suatu proses dan berapa lama waktu yang dibutuhkan untuk proses tersebut.

3.5.1. Waktu Pengiriman SMS

Dalam pengaplikasiannya, proses pengiriman SMS sudah pasti memiliki jeda waktu bergantung pada panjang karakter pada SMS tersebut dan juga *provider* selular yang digunakan. Satu pengiriman SMS hanya dapat memuat 160 karakter. Sehingga bila lebih dari itu akan terhitung lebih dari 1 SMS dalam sekali pengiriman, dengan kata lain *cost* yang dikeluarkan juga akan lebih besar. Untuk mengetahui berapa waktu yang dibutuhkan untuk pengiriman pesan, dilakukan

pengujian terhadap waktu kirim dan waktu terima pesan. Pengujian ini dilakukan pada dua pesan dengan panjang teks yang berbeda.



Gambar 13. Log Waktu Pengiriman SMS

Gambar 13 adalah tampilan log pada saat proses *debug* pada aplikasi *console* Eclipse yang menunjukkan lama waktu proses pengiriman SMS. Log yang ditampilkan adalah waktu aplikasi mulai menjalankan perintah kirim dan ketika status pesan terkirim. Contoh *logdebug* diatas adalah menggunakan 450 karakter (2 SMS dalam sekali kirim). Jadi dapat disimpulkan perhitungan waktu pengiriman SMS adalah waktu pesan terkirim dikurang waktu pesan di kirim. Pengujian dilakukan 5 kali untuk pengiriman jumlah SMS yang sama, namun jumlah karakter berbeda. Hasil pengukuran waktu SMS ditunjukkan pada tabel 4.

Tabel 4. Pengujian Waktu Pengiriman SMS

Panjang Pesan	Panjang *Chipertext*	Jumlah SMS	Waktu Pengiriman
1 Karakter	1 Karakter	1	528ms
15 Karakter	15 Karakter	1	399ms
30 Karakter	30 Karakter	1	433ms
64 Karakter	64 Karakter	1	442ms
80 Karakter	80 Karakter	1	502ms
161 Karakter	161 Karakter	2	865ms
200 Karakter	200 Karakter	2	829ms
240 Karakter	240 Karakter	2	717ms
280 Karakter	280 Karakter	2	811ms
320 Karakter	320 Karakter	2	816ms
350 Karakter	350 Karakter	3	1.093s
390 Karakter	390 Karakter	3	1.316s
420 Karakter	420 Karakter	3	1.554s
450 Karakter	450 Karakter	3	1.725s
480 Karakter	480 Karakter	3	1.639s

3.5.2. Perhitungan *Throughput*

Dapat dilihat pada tabel 6 bahwa panjang karakter pada plaintext (teks asli) dan karakter setelah proses *enkripsi* adalah sama. Hal ini menjadi salah satu keunggulan dari kesederhanaan *enkripsi Vigenere* dibanding algoritma *enkripsi* lain yang lebih rumit. Jumlah karakter menyebabkan pertambahan waktu lamanya

pengiriman SMS, karena pertambahan karakter juga menambah jumlah SMS yang dikirim dalam 1 waktu.

Tetapi itu tidaklah mutlak, karena disi lain juga bergantung pada *provider* selular yang digunakan. Jadi bisa saja jumlah karakter SMS yang lebih sedikit memerlukan waktu kirim yang lebih lama dibanding jumlah karakter yang lebih banyak. Berikut dilakukan perhitungan untuk mengetahui rata-rata waktu pengiriman sesuai dengan jumlah SMS yang dikirimkan:

Rata-rata waktu pengiriman 1 SMS:

$$\frac{528\text{ms} + 399\text{ms} + 433\text{ms} + 442\text{ms} + 502\text{ms}}{5} = 460{,}8ms$$

Rata-rata waktupengiriman 2 SMS:

$$\frac{865\text{ms} + 829\text{ms} + 717\text{ms} + 811\text{ms} + 816\text{ms}}{5} = 807{,}6ms$$

Rata-rata waktupengiriman 3 SMS:

$$\frac{1.093\text{ms} + 1.316\text{ms} + 1.554\text{ms} + 1.725\text{ms} + 1.639\text{ms}}{5} = 1.465ms$$

Dari perhitungan di atas dapat diketahui rata-rata waktu untuk pengiriman satu SMS adalah 460,8ms, waktu untuk pengiriman dua SMS adalah 807,6ms dan waktu untuk pengiriman tiga SMS adalah 1.465ms. Berdasarkan peningkatan jumlah SMS yang di kirimkan,waktu pengiriman juga mengalami peningkatan.

Dari perhitungan rata-rata tersebut diatas, dapat pula diketahui peningkatan waktu pengiriman 2 SMS ke 3 SMS dalam sekali kirim hampir satu kali lipat dibanding waktupengiriman satu SMS (1 SMS ke 2 SMS=346,8ms. 2 SMS ke 3 SMS=657,4ms). Jadi, dapat dihitung rata-rata waktu pengiriman dari ketiga SMS adalah ±502ms. Setelah diketahui waktu pengiriman, maka dapat di hitung *throughput* sebagai berikut:

$$Throughput = \frac{JumlahDatayangDikirim}{WaktuPengiriman}$$

$$\frac{140}{502ms} = 0.2788Bytes/ms$$

Jumlah data pada satuan SMS maksimum adalah 140Bytes, atau 160 karakter 7-bit. Berdasarkan perhitungan di atas, dapat diketahui bahwa *throughput* data pengiriman SMS adalah 0.2788 Bytes/s.

4. SIMPULAN

Berdasarkan penelitian yang sudah dilakukan, akhirnya dapat diambil kesimpulan:

1. Penerapan algoritma kunci privat untuk *enkripsi* SMS pada perangkat android dapat meningkatkan keamanan. Pesan yang ter*enkripsi* tidak akan dapat dibaca jika tidak didekripsi dengan menggunakan kunci yang benar.
2. Dengan algoritma KVC, maka hasil *enkripsi*nya sama dengan jumlah karakter plaintextnya.
3. Waktu yang diperlukan untuk proses pengiriman SMS dengan mengunakan aplikasi SMS biasa dan dengan menggunakan aplikasi *Vigenere* SMS tidak berbeda. Hal ini dikarenakan pada dasarnya aplikasi *Vigenere* SMS ini me-link langsung ke inbox bawaan perangkat selular itu sendiri. Jadi pada aplikasi *Vigenere* SMS, user hanya "menumpang " untuk meng*enkripsi* pesan saja.

4. Waktu pengiriman SMS bergantung pada jaringan sinyal pengirim provider selular yang digunakan.

DAFTAR PUSTAKA

[1] Andi. Java for Mobile Programming, Wahana Komputer, Semarang 2012
[2] Ariyus, Dony, Pengantar Ilmu Kriptografi: Teori, Analisis, dan Implementasi. Penerbit Andi, Yogyakarta, 2008.
[3] Hendrayanto, Rudi dan Ramadona Nilawati. PROGRAM APLIKASI *ENKRIPSI* DAN DEKRIPSI SMSPADA PONSEL BERBASIS ANDROID DENGAN ALGORITMA DES. ISSN:2302-3740. Depok: Prosiding Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2012)
[4] Kajian pustaka.. "Teori SMS (Short MessageService)" 2012 (http://www.kajianpustaka.com/2012/12/teori-sms-short-message-service.html). Diakses tanggal 18 Agustus 2017.
[5] Munir, Rinaldi.. Pengantar Kriptografi. Angkasa Bandung:. 2006
[6] Safaat, Nazruddin. Aplikasi Berbasis Android.: Informatika Bandung. 2013.
[7] Setiawan, Deris..*Sistem Keamanan Komputer.* : Elex Media Komputindo Jakarta 2005
[8] Sun-zine, Berkenalan dengan Algoritma Kriptografi Klasik: *Vigènere Cipher*" 2012(http://sun-coolin.blogspot.co.id/2012/07/algoritma-kriptografi-klasik-*Vigenere-Cipher*.html) Diakses tanggal 5 Agustus 2017.
[9] Telekomunikasi. 2010. "SMS" (http://serenitysparkling.blogspot.co.id/2010/10/sms.html). Diakses tanggal 18 Agustus 2017