PENGAMANAN PESAN YAHOO MESSENGER DENGAN HYBRID CRYPTOSYSTEM KOMBINASI RSA DAN VIGENERE DOUBLE COLUMNAR TRANSPOSITION BERBASIS ANDROID

Ahmad Pudoli

Program Studi Magister Ilmu Komputer, Program Pascasarjana,Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Uttara, Kebayoran Lama, Jakarta Selatan 12260
Telp. (021) 5853753, Fax. (021) 5869225
ahmad.pudoli@explorindo.com

ABSTRAK

Teknologi Informasi membuat seseorang dengan mudah berkomunikasi kapanpun dan dimanapun. Yahoo Messenger merupakan layanan yang dapat digunakan melalui jaringan internet. Namun, tidak ada pengamanan pada pesan yang dikirim oleh layanan tersebut. Jika ada seseorang melakukan serangan contohnya sniffing, maka dapat dengan mudah pesan dapat dibaca. Hal ini dapat merugikan pengguna, terlebih pesan tersebut bersifat rahasia, maka perlu dibuat sistem pengamanan pada pesan tersebut. Salah satu cara untuk melakukan pengamanan tersebut adalah dengan enkripsi. Banyak teknik kriptografi yang dapat digunakan. Pada penelitian ini pengamanan dilakukan dengan menggunakan Hybrid Cryptosystem. Untuk menjaga keabsahan dari pesan yang dikirim perlu ditambahkan teknik error detection yang dalam hal ini menggunakan hash function. Pada penelitian ini menghasilkan aplikasi yang dapat meningkatkan keamanan pesan Yahoo Messenger, dimana pesan tersebut dienkripsi dengan metode Hybrid Cryptosystem kombinasi RSA dan Vigenere Double Columnar Transposition.

Kata kunci : Kriptografi, *Hybrid Cryptosystem, RSA, Vigenere, Double Columnar Transposition*

I. PENDAHULUAN

Informasi berkembang dengan sangat cepat, hal ini secara tidak disadari telah membuat bentuk komunikasi dalam berinteraksi manusia menjadi beragam. Dengan perkembangan Teknologi Informasi juga membuat seseorang dengan mudah mendapatkan informasi, dan komunikasi juga dapat dilakukan dimanapun dan kapanpun.

Dalam perkembangannya untuk melakukan komunikasi bukan hanya dengan telpon atau SMS, tetapi juga layanan *chatting*. Layanan *chating* ini ada yang dapat digunakan secara gratis maupun berbayar. Pada umumnya layanan *chatting* memerlukan jaringan internet untuk dapat saling terhubung. Banyak layanan *chatting* yang dapat digunakan antara lain *Yahoo Messenger, Facebook Chat, WeChat, Line, Whats App* dan lain-lain. Penelitian ini fokus pada *Yahoo Messenger* karena pesan yang dikirim pada layanan tersebut tidak dilakukan pengamanan, sehingga jika dilakukan *snifing* oleh seseorang maka pesan dapat dibaca. Hal ini sangat merugikan bagi pemilik pesan, terlebih pesan tersebut bersifat rahasia.

Pada penelitian ini menghasilkan aplikasi untuk pengamanan pesan dengan menggunakan metode *Hybrid Cryptosystem*. Alasan penggunaan teknik *Hybrid Cryptosystem* karena keamanan dan efisiensi[1]. *Hybrid Cryptosystem* dapat dibangun dengan menggunakan dua kriptografi atau lebih secara terpisah. Pada penelitian ini melakukan pengamanan pesan pada *Yahoo Messenger* dengan metode *Hybrid Cryptosystem.* Dimana dengan menggunakan kombinasi kriptografi RSA dan *Vigenere Cipher* dengan kombinasi *Double Columnar Transposition*.

II. STUDI LITERATUR

Penelitian ini mengacu pada beberapa penulisan terkait penelitian yang telah dilakukan sebelumnya, yaitu sebagai berikut:

1) Jigar Chauhan, Neekhil Dedhia dan Bhagyashri Kulkarni melakukan penelitian tentang *Hybrid Cryptograph* dengan menggunakan AES-DES. Pada penelitian ini mengusulkan metode pengamanan data dengan merancang konsep gabungan AES dan DES untuk mendapatkan model *hybrid* agar dapat digunakan untuk semua jenis data. Dengan menggunakan model *hybrid* AES-DES didapatkan difusi yang lebih baik. Oleh karena itu dengan menggunakan model ini serangan dapat diminimalisir. Model *hybrid* AES-DES membutuhkan proses yang lebih dibandingkan AES atau DES saja, dengan demikian waktu yang digunakan *hybrid* AES-DES untuk melakukan enkripsi dan dekripsi jauh lebih besar [2].
2) Penelitian tentang pengulangan enkripsi pada RSA yang dilakukan oleh Anjana S. Chandran mengatakan RSA masih menjadi algoritma yang kuat dari *public key cryptosystem*. semakin panjang kunci maka semakin kuat algoritma yang digunakan. Namun terjadi sebuah kasus pada saat melakukan enkripsi *plaintext* dengan *key* yang merupakan bilangan kecil secara berulang akan mendapatkan kembali *plaintext*-nya. Oleh karena itu untuk menghindari kasus seperti itu *key* yang digunakan pada RSA harus bilangan bulat yang besar, sehingga sulit

mendapatkan *plaintext* kembali jika *ciphertext* di enkripsi secara berulang[3].

3) Anjali Patil dan Rajeshwari Goudar melakukan studi literatur perbandingan antara kriptografi yang berbeda untuk perangkat *wireless.* Algoritma kriptografi mengkonsumsi sejumlah besar *resource* seperti *memory*, daya baterai dan waktu CPU dalam melakukan enkripsi dan dekripsi. Pilihan algoritma yang lebih baik tergantung pada kelebihan dan kekurangan masing-masing algoritma. Pada penelitian ini memberikan rincian tentang algoritma simetris dan algoritma asimetris. Hasil evaluasi yang didapat bahwa *memory* yang digunakan oleh algoritma simetris lebih sedikit dibandingkan dengan asimetris. Dan algoritma simetris berjalan lebih cepat daripada algoritma asimetris[4].
4) Kemanan informasi merupakan masalah penting pada setiap domain. Konsep utama dalam keamanan data adalah kerahasiaan, integritas, ketersediaan dan otentikasi. Konsep-konsep tersebut harus dicapai oleh setiap sistem keamanan. Untuk mencapai tujuan tersebut dapat dilakukan dengan menggunakan kriptografi. Pada penelitian yang dilakukan oleh Georgiana Mateescu dan Marius Vladescu tentang pendekatan *hybrid cryptosystem* dengan menggunakan kombinasi AES, RSA dan MD5. Mendapat kesimpulan bahwa sebuah *cryptosystem* yang kuat dapat memastikan semua tujuan keamanan berhasil dicapai. Kombinasi algoritma kriptografi yang berbeda memberikan efisiensi maksimal, memperbaiki atau mengatasi kelemahan masing-masing[5].
5) Nishith Sinha dan Kishore Bhamidipati melakukan penelitian untuk meningkatkan sandi *Vigenere Cipher* dengan menggunakan kombinasi *Double Columnar Transposition* untuk menangani kelemahan terhadap serangan Kasiski. Ada dua cara yang dilakukan dalam enkripsi yaitu transposisi dan subtitusi. Hasil dari penelitian ini bahwa kombinasi *Vigenere Cipher* dengan *Double Columnar Transposition* dapat membuat *ciphertext* sulit untuk dikriptanalisis. Dan kompleksitas komputasi yang dilakukan jauh lebih rendah dibandingkan dengan *cipher* modern. Hal ini menjadikan metode yang diusulkan sangat cocok untuk aplikasi yang ringan dan sumber daya yang terbatas. Waktu yang dibutuhkan untuk enkripsi dan dekripsi secara signifikan lebih rendah daripada kebanyakan *cipher* modern [6].
6) Fairouz Mushtaq Sher Ali dan Falah Hassan Sarhan melakukan penelitian untuk meningkatkan keamanan pada *Vigenere Cipher* dengan menggunakan kombinasi *Stream Cipher. Vigenere* merupakan *cipher* subtitusi yang rentan terhadap serangan. Sedangkan pada *cipher* modern seperti *cipher stream*, sandi yang dihasilkan lebih sulit untuk dipecahkan. *Stream Cipher* menggunakan bit, dimana *plaintext*, *ciphertext*, dan *key* dengan menggunakan bit. Pada kriptografi klasik *ciphertext* mudah dipecahkan, hal ini disebabkan karena *cipher* subtitusi hanya menggunakan huruf tidak seperti pada *stream cipher.* Sedangkan pada metode yang diusulkan menggabungkan *Vigenere Cipher* dengan *Stream Cipher,* sehingga menghasilkan *cipher* yang lebih sulit dipecahkan dan dapat meningkatkan keamanan data. Akan tetapi masih membutuhkan waktu dan *resource* yang lebih dibandingkan dengan menggunakan *Vigenere Cipher* saja[7].
7) Penelitian dengan melakukan implementasi *Vigenere Cipher* untuk pengamanan data yang dilakukan oleh Putu H. Arjana, Tri Puji Rahayu, Yakub dan Hariyanto. Pada penelitian ini menyajikan tentang penerapan metode *Vigenere Cipher* untuk mengamankan data pelanggan pada perusahaan. Hasil evaluasi yang didapat bahwa algoritma *Vigenere cipher* dapat meninggkatkan keamanan data pelanggan dan penjualan[8].
8) Hendra dan Sukiman melakukan penelitian dengan membuat aplikasi untuk mengamankan *Short Message Messaging* (SMS) dengan menggunakan Algoritma RSA. Teknologi SMS sendirinya memiliki kelemahan yaitu yaitu enkripsi hanya dilakukan antara *Mobile Station* (MS) dan *Base Transceiver Station* (BTS) sedangkan pada bagian lain terbuka sama sekali, sehingga memungkinkan serangan berupa penyadapan maupun modifikasi. Tujuan pada penelitian ini membangun aplikasi SMS dengan menggunakan kriptografi RSA [9].

III. STUDI PUSTAKA

A. *Kriptografi*

Kriptografi merupakan sebuah seni perlindungan keamanan pesan rahasia dengan mengacaukan dan menyandikan pesan rahasia menjadi kode-kode rahasia atau *ciphertext*[10]. Dengan menggunakan kriptografi orang lain dapat menyadari keberadaan pesan rahasia tersebut, tetapi hanya orang yang memiliki kunci yang dapat membacanya. Kriptografi digunakan untuk menjaga kerahasiaan pesan yang dikirim dengan menggunakan media tertentu sehingga pesan tidak dapat dibaca oleh orang yang tidak berhak.

Tujuan utama penggunaan teknik kriptografi dalam pengiriman pesan rahasia terbagi menjadi beberapa poin-poin penting, yaitu *Confidentialy* (Kerahasiaan), *Authenctication* (Keaslian), *Data Integrity* (Integritas Data), *Non- Repudiation* (Anti Penyangkalan) dan *Access Control* (Kendali Akses) [Abutaha 2011].

Berdasarkan kunci yang digunakan algoritma kriptografi terbagi menjadi dua golongan yaitu simetris dan asimetris. Pada Algoritma Simetris untuk melakukan enkripsi dan dekripsi menggunakan kunci yang sama. Kunci-kunci ini harus dirahasiakan, oleh karena itu kunci ini disebut *secret key chipersystem*[11]. Sebelum data dienkripsi dan dikirim ke penerima lebih dahulu melakukan kesepakatan kunci yang akan digunakan. Oleh karena itu kerahasian kunci menjadi faktor penting untuk tingkat keamanan pada kriptografi dengan algoritma ini. Pada Algoritma Asimetris untuk melakukan enkripsi dan dekripsi menggunakan kunci yang berbeda, yaitu *public key* dan *private key*. Metode kunci asimetris yang dikenal juga kunci publik. Metoda kunci publik ini pertama kali ditemukan oleh Whitfield Diffie dan Martin Hellman, serta Ralph Merkle secara independen[12].

B. *Rhivest – Shamir - Adleman (RSA)*

RSA merupakan salah satu kriptografi asimetris, yaitu kriptografi yang memiliki kunci enkripsi dan dekripsi yang berbeda. Algoritma ini ditemukan oleh tiga orang dari Massachusetts Institute Of Technology pada tahun 1977. Algoritma ini dinamakan berdasarkan dari ketiga nama penemu tersebut, yaitu Ron Rivest, Adi Shamir dan Len Adlement.

Selain digunakan untuk enkripsi, RSA juga dapat digunakan untuk pertukaran kunci dan tanda tangan digital. RSA menggunakan variabel ukuran enkripsi blok dan ukuran kunci variabel. *Keypair* ini berasal dari jumlah yang sangat besar, n yang merupakan produk dari dua bilangan prima. RSA telah banyak digunakan untuk membangun komunikasi data pada jaringan dengan aman dan untuk otentikasi identitas penyedia layanan. Dalam penggunaannya untuk otentikasi, server mengimplementasikan *public key* dengan *client* dengan memberikan *signature* pada pesan dengan menggunakan *private key*. Kemudian signature tersebut dikembalikan ke *client* selanjutnya diverifikasi dengan menggunakan *public key* yang diketahui *server*[13].

Algoritma RSA menggunakan dua kunci yang berbeda, yaitu kunci publik *(public key)* dan kunci rahasia *(private key)*. *Public key* tidak bersifat rahasia dan boleh diberikan kepada orang lain. *Public key* digunakan untuk mengenkripsi pesan atau *plaintext*. Kemudian *private key* sesuai namanya *key* ini bersifat rahasia. Tidak boleh seorang pun yang mengetahui *key* ini, karena *key* ini digunakan untuk mendekripsi *ciphertext*. RSA mengkonsumsi waktu yang lebih lama dan penggunaan *memory* yang lebih tinggi dibandingkan dengan AES dan DES[13].

Sebelum melakukan enkripsi dan dekripsi dengan menggunakan RSA, terlebih dahulu membuat *public key* dan *private key*. Berikut algoritma untuk melakukan enkripsi dan dekripsi pada RSA[14] :

1) Memilih dua buah bilangan prima yang diberi p dan q (disarankan untuk memilih bilangan yang besar)
2) Menghitung nilai n = p . q
3) Kemudian menghitung nilai □(n) = (p–1) x (q–1), dimana □(n) adalah *Euler totient* dari n yaitu bilangan positif kurang dari n dan relatif prima dengan n (gcd □(n) dan n sama dengan 1).
4) Cari bilangan e, yang relatif prima terhadap □(n) dan harus lebih kecil dari □(n).
5) Hitung d dimana d = e^{-1} mod □(n) atau e*d mod □(n) = 1.
6) Untuk melakukan enkripsi yaitu n dan e dimana C = P^e mod n.
7) Untuk melakukan dekripsi yaitu n dan d dimana P = C^d mod n.

C. *Vigenere Cipher*

Vigenere Cipher merupakan perkembangan dari *Caesar Cipher*. Pada *Caesar Cipher* pergeseran setiap karakter dilakukan dengan menggunakan kunci rotasi yang sama, sedangkan pada *Vigenere* setiap karakter pada *plaintext* mengalami pergeseran dengan jumlah yang berbeda sesuai dengan kunci yang digunakan. Dalam mengenkripsi pesan kita membutuhkan sebuah tabel yang berisi alfabet dan terdiri dari baris dan kolom. Baris dinotasikan sebagai bagian dari *key* sedangkan kolom sebagai *plaintext*. Masing-masing baris digeser satu urutan ke kiri dari baris sebelumnya. Setiap huruf disandikan dengan menggunakan baris yang berbeda-beda, sesuai kata kunci yang diulang. Gambar 13 menggambarkan tabel *Vigenere Cipher*.

	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 13 : Tabel Vigenere Cipher

Vigenere ditemukan oleh seorang kriptanalis dan juga merupakan seorang diplomat Perancis yang bernama Blaise de Vigenere pada abad 16 (tahun 1586). *Vigenere* melakukan enkripsi dengan menggunakan sejumlah sandi *Caesar* yang berbeda huruf dari sebuah kunci. Sandi ini merupakan kunci sederhana dari subtitusi polialfabetik[8].

Algoritma enkripsi *vigenere cipher* :

$$Ci = (Pi + Ki) \bmod 26$$

Algoritma dekripsi *vigenere cipher*:

$$Pi = (Ci + 26 – Ki) \bmod 26$$

Dimana :

Ci = nilai desimal karakter *ciphertext* ke-i
Pi = nilai desimal karakter *plaintext* ke-i
Ki = nilai desimal karakter kunci ke-i.

Misalnya, *plaintext* yang akan dienkripsi adalah "TESISKU". Sedangkan kata kunci yang digunakan adalah "DEFAN", karena ada 7 huruf maka sandi akan diulang, sehingga menjadi “DEFANDE”. Sesuaikan kunci dengan jumlah huruf pda *plaintext*. Pada tabel tersebut, T bertemu dengan D berada di huruf 'W', lalu huruf O bertemu dengan E di huruf 'S', dan seterusnya.

Teks Asli : T E S I S K U
Kata kunci : D E F A N D E

Hasil Vinegere : W I X I F N Y
Jadi, hasil enskripsi *Vinegere* dari kata "TESISKU" dengan menggunakan kata kunci "DEFAN" adalah "WIXIFNY".

D. *Vigenere Cipher Kombinasi Double Columnar Transposition*

Pada dasarnya enkripsi dicapai dengan menggunakan dua metode yaitu Subtitusi dan Transposisi. Kriptografi *Vigenere* merupakan subtitusi huruf, yaitu dengan menggantikan setiap karakter *plaintext* dengan karakter lainnya. Sedangkan pada *transposition* mengacu pada perubahan urutan karakter. Penggunaan keduanya secara bersamaan dapat meningkatkan keamanan jika dibandingkan dengan salah satu penggunaannya secara terpisah. Inilah yang diterapkan pada modifikasi *Vigenere Cipher* dengan menggunakan kombinasi *Double Columnar Transposition*[6].

Pada awalnya algoritma *vigenere* dirasa aman, namun pada tahun 1917 dapat dipecahkan oleh Kasiski dan Friedman. *Vigenere* merupakan algoritma yang cepat dan sedikit menggunakan *resource*. Hal ini menjadi alasan untuk pengembangan aplikasi yang membutuhkan *resource* yang terbatas. Oleh karena itu diperlukan peningkatan ketahanan sandi terhadap serangan Kasiski tersebut.

Pada *Vigenere* yang telah dimodifikasi ini menggunakan 2 kunci, yaitu K1 dan K2. Dimana K1 merupakan kunci yang digunakan untuk melakukan subtitusi, kunci ini terdiri dari huruf. Sedangkan K2 merupakan kunci yang digunakan untuk melakukan transposisi, kunci ini terdiri dari kumpulan angka yang unik.

Berikut ini algoritma enkripsi yang digunakan dalam peningkatan *Vigenere Chiper* dengan menggunakan kombinasi *Double Columnar Transposition:*

1) Mengenkripsi *plaintext* dengan menggunakan metode *Vigenere* (dengan kunci K1). Lalu akan menghasilkan C1 (*ciphertext* yang pertama).
2) Setelah itu, melakukan transposisi hingga membentuk sebuah matrik dengan menggunakan kunci transposisi (K2).
3) Susun hasil matrik berdasarkan urutan kolom yang paling kecil. Disini kita dapatkan C2 (*cipher* yang kedua).
4) Ulangi Tahap b dan c dengan menggunakan C2. Disinilah kita mendapatkan *cipertext* yang merupakan hasil enkripsi *Vigenere* dengan menggunakan *Double Columnar Transposition.*

Berikut ini algoritma dekripsi yang digunakan dalam peningkatan *Vigenere Chiper* dengan menggunakan kombinasi *Double Columnar Transposition:*

1) Menghitung jumlah maksimum anggota yang dapat dimiliki pada tiap kolom, dinotasikan dengan m. Nilai m didapat dari total huruf pada pesan dan dimodulasikan dengan jumlah kolom, kemudian hasilnya dilakukan pembulatan keatas. Berikut ini rumus yang digunakan:

$$m = \text{RoundUp}(P / K)$$

Ket:
M = Jumlah maksimum anggota yang dapat dimiliki pada tiap kolom
P = Total huruf pada pesan
K = Jumlah kolom yang didapat berdasarkan kunci transposisi (K2)

2) Menghitung jumlah kolom yang memiliki kekurangan anggota sebanyak 1 anggota (m-1), dinotasikan dengan n. Berikut rumus untuk mendapatkan nilai n:

$$n = (m \times K) - P$$

Ket:
n = Jumlah kolom yang memiliki anggota m-1 (dimulai kolom yang berada disisi kanan)

3) Setelah mendapatkan nilai m dan n, selanjutnya melakukan proses dekripsi transposisi dengan menggunakan kunci transposisi. Susun pesan hingga membentuk matrik.
4) Susun hasil matrik dimulai dari baris pertama. Kita akan mendapatkan *plaintext* yang merupakan *chipertext* kedua dari hasil enkripsi (C2).
5) Ulangi langkah c dan d dengan menggunakan C2. Kita akan mendapatkan *plaintext* yang merupakan *ciphertext* pertama dari hasil enkripsi (C1).
6) Tahapan selanjutnya, setelah mendapatkan C1 maka di dekripsi dengan *Vigenere* dengan menggunakan kunci *Vigenere* (K1). Pada proses ini kita akan menghasilkan *plaintext* yang merupakan pesan asli.

E. *Hybrid Cryptosystem*

Hybrid Cryptograph merupakan teknik kriptografi dengan menggunakan dua atau lebih cipher yang berbeda dalam waktu bersamaan. Sedangkan Hybrid Cryptosystem dibangun dengan menggunakan dua atau lebih kriptografi yang terpisah. Pada Hybrid Cryptosystem untuk melakukan enkripsi atau dekripsi pesan yang panjang akan efisien dengan menggunakan symetryc-key. Sedangkan kunci publik hanya digunakan untuk mengenkripsi/mendekripsi kunci simetris yang pendek.

F. *Checksum*

Checksum merupakan salah satu skema dari *Redudancy Check* (RC). RC adalah proses pendeteksian dan pengkoreksian *error* dari sebuah data, ini merupakan solusi untuk melindungi integritas/keaslian dari sebuah data. Biasanya *Checksum* disimpan dibagian *header* dari data.

Berikut ini salah satu contoh pencarian dalam mencari nilai *Checksum* pada sebuah data:

1) Jumlahkan semua byte pada data.
2) Hilangkan *carry* (Sisa hasil penjumlahan pada bilangan hexa) bila ada.
3) Cari *two's* complement hasil nomor 2, maka didapatkanlah nilai *checksum.*

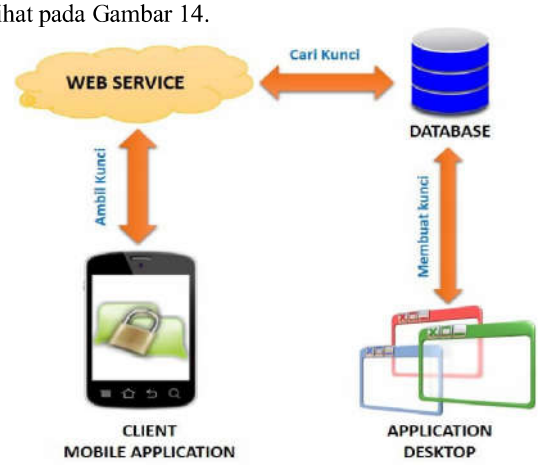
Contoh:

1) Diberikan 4 byte: 0x15, 0x7F, 0x86, 0x5C
2) 0x15 + 0x7F + 0x86 + 0x5C = 0x176
3) 0x176 → 0x76
4) *Two's complement*(0x76) = 0x8A, jadi nilai *Checksum* = 0X8A

Salah satu algoritma *checksum* yaitu *Message Digest* 5 (MD5). MD5 diciptakan oleh Ron Rivest merupakan fungsi *hash* satu arah, dimana dengan mudah melakukan enkripsi untuk mendapatkan *cipher*-nya tetapi sangat sulit untuk mendapatkan *plaintext*-nya. MD5 adalah salah satu algoritma yang digunakan untuk mengetahui bahwa pesan yang dikirim tidak ada perubahan sewaktu berada di jaringan[15].

IV. RANCANGAN APLIKASI

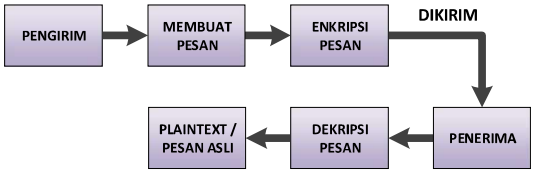
Dalam pembuatan sistem pengamanan pesan *Yahoo Messenger* yang dikembangkan dalam penelitian ini ada beberapa aplikasi yang dihasilkan, yaitu aplikasi *desktop, web service* dan *mobile application*. Pengembangan aplikasi pengamanan pesan dibangun dengan arsitektur sistem yang terlihat pada Gambar 14.



Gambar 14 : Arsitektur Sistem Yang Dibangun

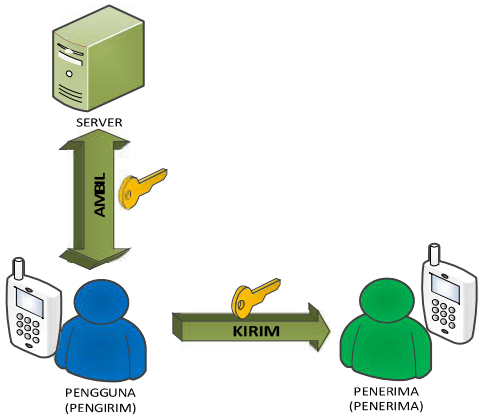
Aplikasi *desktop* digunakan untuk membuat atau *generate* kunci publik. Nantinya kunci-kunci tersebut akan digunakan oleh pengguna untuk pengamanan pesan. Dengan menggunakan aplikasi *mobile,* kunci tersebut diambil oleh pengguna melalui *web service* dan selanjutnya dikirim ke pengguna lainnya. Aplikasi *mobile* merupakan aplikasi yang akan digunakan oleh pengguna terakhir (*end user*). Pada aplikasi inilah pengguna dapat melakukan *chatting* dengan pengamanan pesan.

Alur dari proses pengiriman atau penerimaan pesan secara umum dapat dilihat pada Gambar 15. Proses dimulai dari pembuatan pesan yang dilakukan oleh pengirim. Oleh karena itu, penerima membutuhkan proses dekripsi pesan untuk mendapatkan pesan asli yang dikirim oleh pengirim. Proses enkripsi pesan dilakukan untuk pengamanan pesan agar hanya penerima yang dapat membaca pesan yang asli. Oleh karena itu, penerima membutuhkan proses dekripsi pesan untuk mendapatkan pesan asli yang dikirim oleh pengirim.



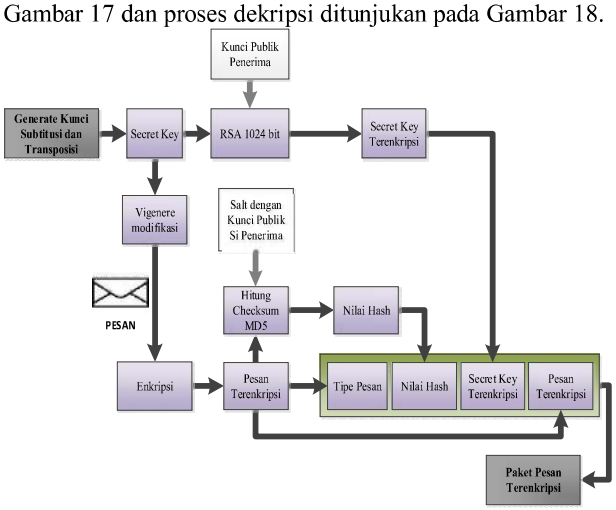
Gambar 15 : Gambaran Umum Sistem yang Dibangun

Selanjutnya, agar proses komunikasi berjalan dengan pengamanan pesan maka pengirim dan penerima harus memiliki kunci publik. Pengirim memiliki kunci publik penerima dan sebaliknya. Pada sistem yang dibangun memiliki fitur untuk mengambil kunci dari *server* dan mengirim kunci ke teman. Proses pengambilan kunci dari *server* dan pengiriman kunci ke teman ditunjukan pada Gambar 16:



Gambar 16 : Proses Mengambil Kunci dari Server dan Mengirim Kunci Ke Teman

Pada Gambar 15 terdapat proses enkripsi dan dekripsi pesan. Untuk menjelaskan proses enkripsi ditunjukan pada Gambar 17 dan proses dekripsi ditunjukan pada Gambar 18.

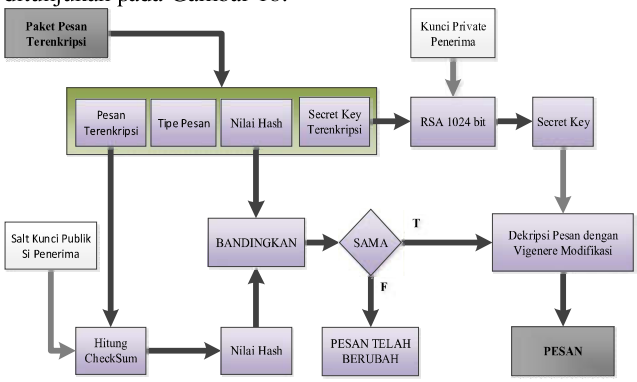


Gambar 17 : Desain Enkripsi dengan Hybrid Cryptosystem Usulan

Pada Gambar 17 merupakan blok diagram proses enkripsi yang dibangun. Dimulai dari *Generate Key* sampai mendapatkan paket pesan yang terenkripsi. Berikut ini algoritma yang menjelaskan alur proses tersebut:

1) Melakukan *Generate Key* untuk mendapatkan kunci subtitusi dan transposisi. Dua kunci tersebut selanjutkan menjadi *Secret Key.*
2) Pesan dienkripsi dengan menggunakan *Vigenere Chiper* yang dimodifikasi dengan kombinasi *Double Columnar Transposition.* Pada proses enkripsi ini menggunakan dua kunci, yaitu kunci subtitusi dan transposisi.
3) Setelah pesan dienkripsi maka proses selanjutnya menghitung nilai *checksum*-nya dengan menggunakan *Message Digest 5 (MD5)*. Pada proses ini ditambahkan *salt* dengan menggunakan kunci publik penerima.
4) Enkripsi *Secret Key* (kunci subtitusi dan transposisi) dengan Algoritma RSA, dimana dengan menggunakan kunci publik penerima pesan.
5) Membuat paket pesan terenkripsi dengan menggabungkan tipe pesan, *hash,* kunci terenkripsi dan pesan yang terenkripsi.

Berikut ini akan dijelaskan proses dekripsi pesan yang ditunjukan pada Gambar 18:

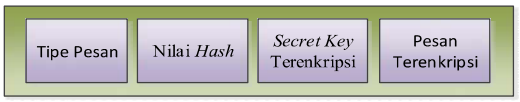


Gambar 18 : Desain Dekripsi dengan Hybrid Cryptosystem Usulan

Pada Gambar 18 merupakan blok diagram proses dekripsi yang dibangun. Dimulai dari memisahkan pesan yang terenkripsi sampai mendapatkan pesan asli. Berikut ini algoritma yang menjelaskan alur proses yang terdapat pada Gambar 18:

1) Memisahkan paket pesan terenkripsi sehingga masing-masing menjadi tipe pesan, *hash* dan kunci terenkripsi dan pesan terenkripsi.
2) Hitung nilai *hash* pesan terenkripsi menggunakan *Checksum* MD5. Pada proses ini ditambahkan *salt* dengan menggunakan kunci publik penerima, kemudian bandingkan dengan nilai *hash* yang disimpan.
3) Jika nilai *hash* sama, maka *secret key* yang dienkripsi dengan RSA didekripsi dengan menggunakan *private key* si penerima. Nanti akan mendapatkan *Secret Key* (kunci subtitusi dan transposisi) untuk melakukan dekripsi *Vigenere.*
4) Setelah mendapatkan kunci subtitusi dan transposisi, maka dekripsi pesan dengan menggunakan *Vigenere Cipher* yang dikombinasikan dengan *Double Columnar Transposition.*

Protokol atau aturan standar yang digunakan pada pesan untuk melakukan komunikasi pada sistem. Protokol pesan yang terdapat pada sistem yang akan dibangun adalah Tipe Pessan, *Hash, Secret Key* Terenkripsi dan Pesan Terenkripsi. Paket pesan yang dikirim akan disimpan dalam bentuk data *gson.* Gambar 19 menunjukan protokol pesan yang digunakan pada sistem.



Gambar 19 : Protokol Pesan

Ada tiga tipe pesan yang terdapat pada sistem. Tipe-tipe pesan ini digunakan berdasarkan kebutuhan sistem. Berikut Tabel 3 yang menjelaskan tipe-tipe pesan tersebut:

Tabel 3 : Tipe Pesan yang Digunakan Pada Sistem

No.	Kode	Nama	Deskripsi
1.	01	Pesan Biasa	Tipe pesan ini merupakan pesan biasa yang diterima dan dapat didekripsi oleh pengguna atau penerima.
2.	02	Kiriman *Public Key*	Pesan ini merupakan pesan yang berisi *Public Key* pengirim dan akan disimpan oleh penerima

Pada pesan yang dikirim terdapat nilai *hash* yang berfungsi untuk validasi pesan. Proses validasi bertujuan memastikan pesan belum berubah dan untuk penerima yang dimaksud. Oleh karena itu digunakan algoritma *Message Digest 5 (MD5)* ditambahkan dengan sebuah *salt. Salt* tersebut merupakan *public key* dari penerima. Jika isi pesan sama tetapi tujuan penerima berbeda maka nilai *hash* pun akan berbeda, sehingga dapat mengidentifikasi pesan sudah berubah dan ditujukan kepada siapa. Dengan menggunakan proses validasi ini keamanan pesan semakin meningkat.

V. HASIL PENELITIAN

A. *Tampilan Aplikasi*

Tampilan aplikasi dibuat menarik dan mudah untuk digunakan. Penjelasan tampilan akan dijabarkan pada tiap-tiap tampilan form pada aplikasi.

1. Tampilan *Form Login*

*Form L*ogin merupakan tampilan awal pada saat pertama kali aplikasi dijalankan. *Form Login* digunakan untuk masuk dan menggunakan aplikasi. Gambar 20 merupakan tampilan *Form Login.*

Gambar 20 : Tampilan Form Login

2. Tampilan *Form* Daftar Teman

Setelah pengguna berhasil melakukan login, maka aplikasi akan menampilkan *Form* Daftar Teman. Pada *form* ini pengguna dapat memilih teman untuk *chatting,* pencarian teman, mengambil kunci, mengubah status, melihat percakapan dan *sign out* atau keluar dari aplikasi. Gambar 21 merupakan tampilan dari *Form* Daftar Teman.

Gambar 21 : Tampilan Form Daftar Teman

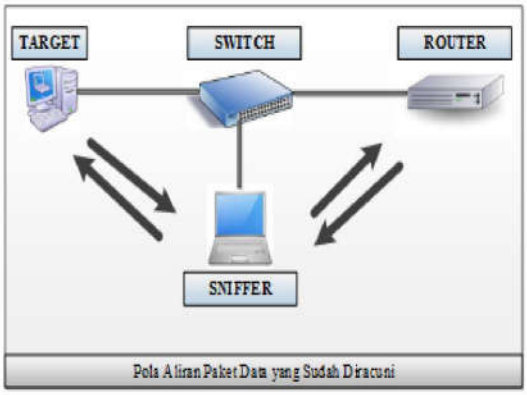
3. Tampilan *Form Chat*

Pada *Form Chat* pengguna dapat melakukan *chatting* dengan teman yang dipilih. Ada beberapa indikasi pada setiap *chat* yang dikirim atau diterima ditandai dengan warna latar pada pesan. Chat dengan warna latar hijau menandakan bahwa pesan tersebut sudah aman atau dienkripsi. Warna latar biru toska adalah informasi yang diberikan untuk pengguna, missal seperti *"BUZZ",* mengirimkan kunci, pesan tidak valid dan lain-lain. Warna latar merah muda menandakan pesan tidak aman atau tidak dienkripsi. Selain *chat* pada *form* ini pengguna juga dapat membersihkan pesan dan mengirimkan kunci ke teman. Untuk lebih jelasnya dapat dilihat pada Gambar 22.

Gambar 22 : Tampilan Form Chat

B. *Pengujian Sniffing Chat dengan Cain And Abel dan Wireshark*

Pengujian dilakukan dengan melakukan penyerangan terhadap aplikasi *Yahoo Messenger* dan juga aplikasi yang dibangun. Penyerangan dilakukan dengan metode *Man In The Middle,* dengan cara teknik *sniffing.* Penyadapan paket data yang mengalir pada jaringan dari *host* yang diserang. Tools yang digunakan adalah *Cain And Abel* dan *Wireshark.*



Gambar 23 : Skema Penyerangan dengan Sniffing

Tujuan dari *Sniffing Chat* ini adalah untuk memeriksa pesan yang dikirim dari aplikasi sudah terenkripsi. Pada pengujian ini akan membandingkan pesan yang dikirim dari aplikasi yang dibangun dengan pesan yang dikirim dari aplikasi *Yahoo Messenger.* Dengan menggunakan teknik *sniffing* kita akan mendapatkan paket data yang mengalir pada jaringan dan memilih pesan yang dikirim dari aplikasi. Berikut langkah-langkah pengujian yang dilakukan:

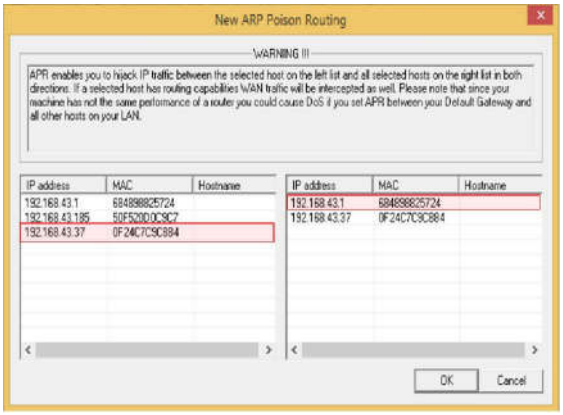
1. Menentukan target dan pesan yang dikirim pada pengujian *Sniffing Chat*

Ada dua target dalam pengujian ini, yaitu *mobile device* dengan IP **192.168.43.185** dan personal komputer dengan IP **192.168.43.37**. Pada *mobile device* terpasang atau di-*install* aplikasi yang dibangun, sedangkan pada komputer personal terpasang aplikasi *Yahoo Messenger.* Kedua aplikasi digunakan

untuk mengirim pesan **"*testing sniffing chat dengan cain and abel dan wireshark*"** dengan pengguna ahmad.pudoli4tesis2, yang nantinya pesan akan dicuri dengan menggunakan serangan *Man In The Middle Attact.*

2. Meracuni ARP dengan aplikasi *Cain And Abel*

Tahap ini melakukan penyerangan dengan meracuni ARP. Proses meracuni ARP target dilakukan dengan menggunakan *Tool Cain And Abel.* Setelah ARP target diracuni, langkah selanjutnya adalah mengamati aliran paket data yang terjadi antara target dan *Gateway* yang sudah diketahui oleh *sniffer.* Gambar 24 menunjukan proses meracuni ARP pada personal komputer yang menjadi target. Hal yang sama dilakukan untuk *mobile device.*

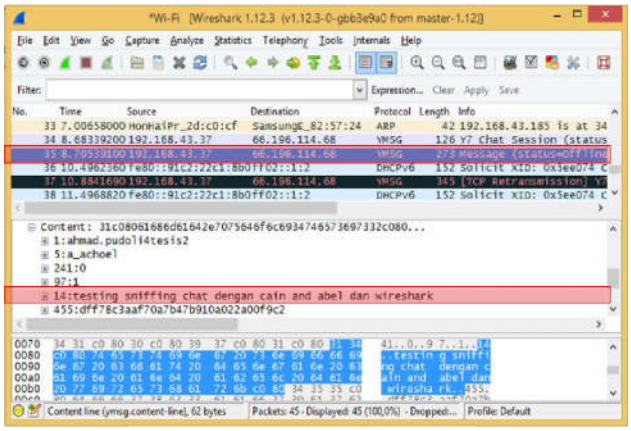


Gambar 24 : Meracuni ARP Target Pengujian

3. Mencari pesan yang dikirim dengan aplikasi *Wireshark*

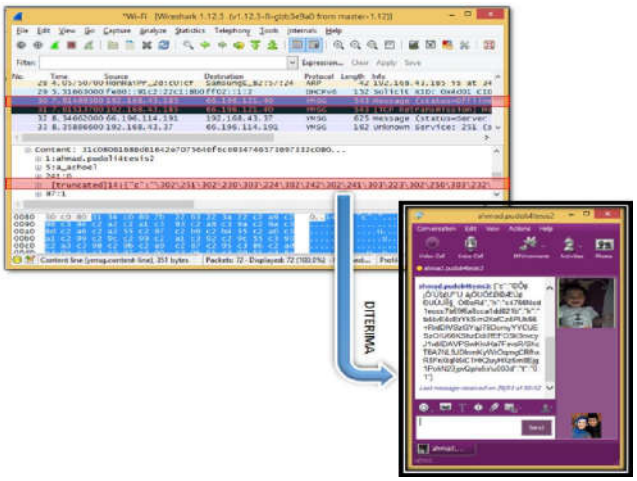
Tahap selanjutnya adalah adalah mengamati paket yang berasal dari IP target dengan menggunakan *Tool Wireshark.* Peneliti mengamati komunikasi antara target dengan *Gateway* dan mencari paket data dari *Yahoo Messenger.* Pada tabel aliran paket data yang terdapat pada *Wireshark* cari yang menggunakan protokol YMSG dengan Info *Message.* Pada paket data tersebut terdapat pesan yang dikirim oleh target. Pilih paket data tersebut, kemudian pada detail paket data *expand content*. Pada *content* di tag 14 merupakan pesan yang dikirim pada *Yahoo Messenger.*

Pada Gambar 25 menunjukan aliran paket data pada jaringan yang dapat ditangkap oleh aplikasi *Wireshark.* Didapatkan paket pesan yang dikirim melalui aplikasi *Yahoo Messenger!.* Terlihat pesan tidak dienkripsi dan tidak dapat dibaca.



Gambar 25 : Pesan Dikirim Dari Aplikasi Yahoo Messenger

Pada Gambar 26 menunjukan paket pesan yang dikirim dari aplikasi yang dibangun. Terlihat pada aplikasi *Wireshark,* pesan berhasil didapatkan dan tidak dapat dibaca karena pesan tersebut telah dilakukan pengamanan dengan enkripsi.



Gambar 26 : Pesan yang Dikirim Melalui Aplikasi Yang Dibangun

Terlihat jelas perbedaan pesan yang dikirim melalui aplikasi *Yahoo Messenger* dengan pesan yang dikirim oleh aplikasi yang dibangun. Pesan yang dikirim pada Aplikasi *Yahoo Messenger* tidak dilakukan pengamanan. Jika pesan dicuri oleh orang lain, maka dengan mudah dapat dibaca. Pada aplikasi yang dibangun, pesan dienkripsi terlebih dahulu sebelum dikirim ke tujuan. Sehingga dapat memberikan pengamanan pada pesan.

C. *Pengujian Error Detection*

Error Detection digunakan untuk mengetahui pesan yang dikirim dari aplikasi sudah mengalami perubahan atau tidak. Dengan mengunakan fungsi *hash,* yaitu *Message Digest Algorithm 5* (MD5). Setelah pesan asli dienkripsi dengan menggunakan *Vigenere* kombinasi *Double Columnar Transposition,* maka hasil enkripsi dihitung nilai *hash*-nya.

Nilai *hash* ini yang akan disimpan dalam paket pesan. Pada saat pesan diterima, aplikasi akan menghitung nilai *hash* pesan terenkripsi dan membandingkan dengan nilai *hash* yang tersimpan pada paket pesan. Bila nilai *hash* berbeda, maka dapat dipastikan bahwa pesan telah berubah.

Dalam pengujian ini peneliti akan mengirim pesan dari aplikasi yang dibuat dan selanjutnya akan diterima melalui aplikasi *Yahoo Messenger*. Setelah pesan diterima dan diubah, selanjutnya pesan akan dikirim kembali melalui aplikasi *Yahoo Messenger* dan akan diterima oleh aplikasi yang dibuat. Aplikasi yang dibuat akan memberikan informasi bahwa pesan tidak valid ketika pesan diterima. Berikut ini langkah-langkah pengujian yang dilakukan:

1. Menentukan pengguna dan pesan yang dikirim

   Tahap atau langkah pertama yang dilakukan pada pengujian ini adalah menentukan pengguna dan pesan yang akan dikirim. Pada pengujian ini membutuhkan dua pengguna yaitu sebagai pengirim dan penerima pesan. Pengguna tersebut adalah **ahmad.pudoli4tesis2** dan **a_achoel**. Pengguna ahmad.pudoli4tesis2 sebagai pengirim dan a_achoel sebagai penerima. Pesan yang akan dikirim adalah **"testing error detection tesisku".**

2. Pengguna mengirimkan pesan dari aplikasi yang dibangun ke aplikasi *Yahoo Messenger*

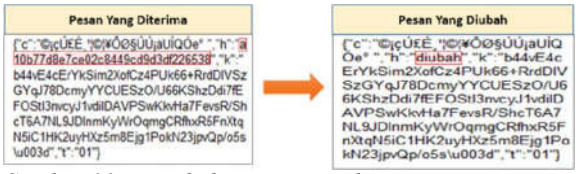
   Setelah menentukan pengguna dan pesan yang akan dikirim. Tahapan selanjutnya adalah mengirimkan pesan "testing error detection tesisku". Proses pengiriman dilakukan oleh pengguna ahmad.pudoli4tesis2 dengan menggunakan aplikasi yang dibangun dan pengguna a_achoel akan menerima pesan dengan aplikasi *Yahoo Messenger*. Pada

   Gambar 27 terlihat bahwa pesan yang dikirim oleh ahmad.pudoli4tesis2 dapat diterima oleh a_achoel akan tetapi tidak dapat dibaca karena sudah dilakukan enkripsi atau pengamanan.

Gambar 27 : Proses Pengiriman Pesan Pada Pengujian Error Detection

3. Mengubah pesan yang diterima pada aplikasi *Yahoo Messenger*

   Pesan yang diterima pada aplikasi *Yahoo Messenger,* selanjutnya akan diubah. Pada pengujian ini ada bagian dari pesan yang diubah menjadi **"diubah"**. Sehingga menjadi sebuah pesan baru yang pada tahap selanjutnya akan dikirim kembali. Gambar 28 menunjukan perubahan pesan pada proses pengujian *error detection.*



Gambar 28 : Perubahan Pesan Pada Pengujian Error Detection

4. Pengguna mengirimkan pesan yang sudah diubah dari aplikasi *Yahoo Messenger* ke aplikasi yang Dibangun

   Tahapan terakhir pada proses pengujian ini adalah mengirimkan kembali pesan yang sudah diubah. Perubahan pesan telah dilakukan pada tahap sebelumnya. Pesan yang sudah diubah akan dikirim oleh pengguna ahmad.pudoli4tesis2 melalui aplikasi *Yahoo Messenger* dan akan diterima oleh pengguna a_achoel pada aplikasi yang dibangun. Ketika a_achoel dengan aplikasi yang dibangun menerima pesan tersebut maka akan menampilkan pesan informasi "Pesan tidak valid" dengan warna latar biru toska. Gambar 29 menunjukan hasil akhir dari proses dari pengujian *Error Detection.*

Gambar 29 : Proses Pengiriman Pesan yang Sudah Diubah Pada Pengujian Error Detection

Berdasarkan hasil dari langkah-langkah pengujian *error detection* yang telah dilakukan menunjukan bahwa metode *Message-Digest Algorithm 5 (MD5)* yang digunakan untuk mendeteksi kesalahan sudah cukup baik. Gambar 29 menunjukan aplikasi telah dapat mendeteksi pesan yang sudah diubah dengan memberikan informasi "Pesan tidak valid !".

VI. KESIMPULAN

A. Kesimpulan

Berdasarkan analisis dan hasil pengujian yang telah dilakukan pada pengembangan model pengamanan pesan *Yahoo Messenger* dengan *Hybrid Cryptosystem* kombinasi RSA dan *Vigenere Double Columnar Transposition* berbasis *Android,* maka dapat ditarik kesimpulan sebagai berikut:

1) Penggunaan metode *Hybrid Cryptosystem* yang dibangun dapat memberikan dalam pengiriman pesan dan efisiensi dalam memberikan kunci rahasia (*secret key).*
2) aplikasi dapat berjalan dengan baik meskipun dalam lingkungan dengan sumberdaya yang rendah.
3) Dengan penambahan *error detection*, maka dapat diketahui jika pesan sudah berubah. Penggunaan teknik *error detection* menggunakan MD5, mampu memberikan keamanan bahwa pesan yang dikirim tidak berubah dengan memeriksa nilai *hash* yang dimiliki.
4) Berdasarkan hasil pengujian *sniffing chat* terlihat jelas perbedaan antara pesan yang dikirim melalui aplikasi *Yahoo Messenger* dengan aplikasi yang dibangun. Pesan dari aplikasi *Yahoo Messenger* jika paket pesan dicuri oleh orang lain maka dengan mudah dapat dibaca, sedangkan pesan dari aplikasi yang dibangun tidak dapat dibaca. Karena pesan yang dikirim dari aplikasi yang dibangun sudah dienkripsi.

B. Saran

Beberapa saran untuk penelitian lebih lanjut dan penyempurnaan penelitian tentang penelitian ini adalah sebagai berikut:

1) Pesan yang dapat dilakukan pengamanan atau enkripsi bukan hanya teks tetapi juga berupa gambar atau file. Untuk meningkatkan ruang lingkup aplikasi dan melakukan pengamanan pada seluruh jenis pesan yang ada.
2) Perlunya mencoba metode kriptografi lainnya yang lebih baik dari metode *Vigenere* dengan kombinasi *Double Columnar Transposition.* Tidak hanya fokus meningkatkan keamanan, tetapi juga memperhatikan aspek efisiensi dalam penggunaan sumber daya dan waktu.
3) Perlunya sebuah metode yang lebih efisien dalam melakukan pendistribusian kunci. Sehingga dapat mempermudah pengguna dalam memperoleh kunci dari daftar teman yang dimiliki.

DAFTAR PUSTAKA

[1] Gupta, Ravindra Kumar and Parvinder Singh. "A New Way to Design and Implementation of Hybrid Cryptosystem for Security of The Information in Public Network". *International Journal of Emerging Technology and Advanced Engineering,* Vol. 3, Agustus 2013.

[2] Chauhan, Jigar, et.al. "Enhancing Data Security by using Hybrid Cryptography Algorithm". *International Journal of Engineering Science and Innovative Technology (IJESIT),* Vol. 2, No. 3, Mei 2013.

[3] Chandran, Anjana S. "Repeated Encryption on RSA", *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE),* Vol. 4, No. 4, April 2014.

[4] Patil, Anjali and Rajeshwari Goudaar. "A Comparative Survey Of Symetric Encryption Techniques For Wireless Devices". *International Journal Of Scientific and Technology Research (IJSTR),* Vol. 2, No. 8, Agustus 2013.

[5] Mateescu, Georgiana and Marius Vladescu. "A Hybrid Approach of System Security for Small and Medium Enterprises: combining different Cryptography techniques". *Federated Conference on Computer Science and Information System,* (2013):659-662.

[6] Sinha, Nishith and Kishore Bhamidipati. "Improving Security of Vigenere Cipher by Double Columnar Transposition". *International Journal of Computer Application,* Vol. 100, No. 14, Agustus 2014.

[7] Ali, Fairous Mushtaq Sher and Falah Hassan Sarhan. "Enhancing Security of Vigenere Cipher by Stream Cipher", *International Journal of Computer Application,* Vol. 100, No. 1, Agustus 2014.

[8] Arjana, Putu H., et.al. "Implementasi Enkripsi Data dengan Algoritma Vigenere Cipher". *Seminar Nasional Teknologi Informasi dan Komunikasi (SENTIKA),* Maret 2012.

[9] Hendra and Sukiman. "Aplikasi Pengaman Pertukaran SMS pada Perangkat Android dengan Metode RSA". *Seminar Nasional Teknologi Informasi dan Komunikasi (SNASTIKOM),* 2012.

[10] Gutub, Adnan Abdul-Aziz. "Pixel Indicator Technique for RGB Image Steganography". *Journal of Emerging Technologies in Web Intelligence,* Vol.2, No.1, Februari 2010.

[11] Santi, Rina Chandra Noer. "Implementasi Algoritma Enkripsi Playpair pada File Teks". *Jurnal Teknologi Informasi DINAMIK*, vol. XV, (Januari, 2010): 27-33.

[12] Tjiharjadi, Semuil and Marvin Chandra Wijaya. "Pengamanan Data Menggunakan Metoda Enkripsi Simetri dengan Algoritma FEAL". *Seminar Nasional Aplikasi Teknologi Informasi (SNATI),* Juni 2009.

[13] Seth, Shashi Mehrotra and Rajan Mishra. "Comparative Analysis Encryption Algorithms For Data Communication". *International Journal of Computer Science and Technology (IJCST),* Vol. 2, No. 2, Juni 2011.

[14] Lukas, Samuel and Ni Putu Sri Artati. "Analisis Waktu Enkripsi-Dekripsi File Text Menggunakan Metoda One-Time Pada (OTP) dan Rivest, Shamir, Adleman (RSA)". *Seminar Nasional Sistem dan Informatika,* November 2007.

[15] Aghus Sofwan, Agung Budi P., Toni Susanto, Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro, "Aplikasi Kriptografi Dengan Algoritma Message Digest 5 (Md5)". *Transmisi*, vol. 11, (Juni, 2006): 22-27