APLIKASI MENGAMANKAN PESAN *CHATTING* DENGAN ALGORITMA KNAPSACK BERBASIS *WEBSITE*

Widana Puguh Saksono[1], Siswanto[2]
Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
Telp. (021) 5853753, Fax. (021) 5866369
E-mail : widanapuguh@gmail.com[1], siswanto@budiluhur.ac.id[2]

ABSTRAK

Aplikasi ini dibuat untuk mengirimkan suatu pesan instan yang mempu mempermudah sebuah komunikasi dengan kemananan pesan teks. Keamanan suatu data teks merupakan hal yang perlu diperhatikan dalam menjaga kerahasiaan data itu sendiri, terutama bila data teks tersebut hanya boleh diketahui oleh pihak tertentu saja. Melindungi akses data dari pihak yang tidak berkepentingan, maka sangat diperlukan kemanan seperti proses enkripsi dan dekripsi dengan algoritma tertentu. Algoritma yang digunakan disini adalah Algoritma Merkle Hellman Knapsack. Algoritma ini menciptakan sebuah subset masalah yang dapat diselesaikan dengan mudah dan kemudian untuk menyembunyikan sifat super meningkat oleh perkalian modular dan permutasi. Pendekatan yang dilakukan untuk mewujudkan kerahasiaan data tersebut diubah kedalam bentuk algoritma berbasis matematika yang membuat data menjadi tidak terbaca. Aplikasi ini dibangun dengan Bahasa pemrograman php untuk proses chatting dan MySQL sebagai database. Aplikasi ini berbasis website.

Kata Kunci : Aplikasi *Chatting*, Keamanan data teks, Knapsack, Enkripsi, Dekripsi

1. PENDAHULUAN

Karena sebagian orang terkadang lupa mengenai informasi yang telah disampaikan dan dipilihlah *chatting* sebagai tempat penyimpanannya.PT. Yudhakarsa Indonesia merupakan perusahaan yang bergerak di bidang jasa. PT Yudhakarsa Indonesia berperan dalam pengadaan barang, alat-alat peraga, serta buku pelajaran untuk sekolah di Indonesia.Dalam Proses Bisnisnya, komunikasi yang terjalin di PT. Yudhakarsa Indonesia untuk melakukan pemesanan masih kurang efektif dan praktis karena masih melalui *email*tanpa bantuan dari aplikasi lainnya atau dengan datang langsung ke kantor. Keamanan informasi sangat dibutuhkan dalam dunia bisnis, terlebih jika pesan yang dikirim bersifat penting dan rahasia. PT. Yudhakarsa Indonesia menggunakan *email* dalam kegiatan bisnisnya, baik untuk pertukaran informasi, pesan finansial ataupun untuk melakukan pemesanan buku maupun alat peraga.Dengan adanya aplikasi *chatting* berbasis website dengan algoritma kriptografi*knapsack* ini diharapkan proses komunikasi antara seller dan buyer menjadi lebih mudah dan efektif serta memberikan rasa aman bagi setiap individu yang menggunakan aplikasi ini.

Di dunia internet, *Chat* (mengobrol) merupakan cara untuk berkomunikasi langsung sesama pengguna internet yang sedang sama-sama menggunakan internet. *Chatting* adalah suatu pesan instan di sebuah teknologi jaringan komputer yang mengijinkan pemakainya untuk mengirimkan pesan ke pengguna lain yang tersambung dalam sebuah jaringan komputer ataupun internet. Komunikasi ini dapat berupa teks ataupun suara [7].

Algoritma Knapsack adalah algoritma kriptogradi kunci publik yangkeamanan algoritma ini terletak pada sulitnya memecahkan persoalan Knapsack (Knapsack *Problem*). Knapsack juga dapat diartikan sebagai sebuah karung yang dapatdigunakan untuk memasukan sejumlah barang. Karung tersebut memiliki kapasitas yang terbatas, sehingga tidak semua barang dapat masuk ke dalam karung. Untuk mengoptimalkan penggunaan kapasitas karung terbatas, maka perlu pemilihan yang tepat terhadap jenis barang yang dimasukan.

Masalah knapsack adalah masalah lengkap NP di optimasi kombinatorial. Yang ditunjuk oleh masalah ransel item yang paling berguna dari sejumlah item mengingat bahwa yang ransel atau ransel memiliki kapasitas tertentu . Masalah ransel secara luas digunakan untuk memodelkan solusi masalah industri seperti kriptografi kunci publik.

Masalah 0-1 ransel menyatakan bahwa jika ada ransel dengan kapasitas tertentu dan sejumlah item yang perlu dimasukkan ke dalam ransel . Setiap item memiliki nilai dan berat yang terkait dengannya . Yang ditunjuk oleh masalah ransel

item yang dapat dimasukkan ke dalam ransel sehingga nilai semua item dimaksimalkan dan berat tidak meningkatkan total kapasitas ransel . Hal ini dapat dinyatakan dengan rumus:

Maximize = Maximize

$$\sum_{i=1}^{n} Pi\ Xi$$

$V = (v1, v2, v3, \ldots vn)$.....[1]

dimana,*p* adalah nilai yang terkait dengan setiap item *I w* adalah berat badan yang berhubungan dengan masing-masing item *I W* adalah kapasitas maksimum *knapsack*, *n* adalah jumlah item Masalah bagian sum adalah kasus khusus dari *Knapsack* Masalah ini menemukan sekelompok bilangan bulat dari daftar vektor *V* ,di mana subset elemen dalam vektor *V* yang memiliki jumlah yang diberikan *S*. hal ini juga menentukan apakah vektor $X = (x1, x2, x3 \ldots xn)$ (2)
ada di mana *xi* unsur { 0,1} sehingga $V * X = S$.Ralph Merkle dan Martin Hellman menggunakan subset masalah untuk membuat sistem kriptografi untuk mengenkripsi data. *A* superincreasing vektor ransel *s* dibuat dan Properti *superincreasing* disembunyikan dengan membuat kedua vektor *M* oleh perkalian *modular* dan permutasi . itu vektor *M* adalah kunci publik *cryptosystem* dan *s* adalah digunakan untuk mendekripsi pesan.

Langkah-langkah penjelasan Matematika sebagai berikut:

1) memilih urutan *superincreasing* dari jumlah bilangan bulat positif. Urutan superincreasing adalah salah satu di mana setiap nomor lebih besar dari jumlah semua sebelumnya angka .s = (s1 , s2 , s3 , ... sn).
2) untuk mengkonversi semua karakter dari pesan ke biner . Urutan biner diwakili oleh variabel *b* .
3) untuk memilih dua nomor - integer (*a*) yang lebih besar dari pada jumlah semua nomor di urutan dan *co -prime* (*r*).
4) Urutan dan nomor dan r membentuk kolektif kunci pribadi *cryptosystem* tersebut.
5) Semua elemen - s1 , s2 , s3 , ... sn, dari urutan *s* adalah dikalikan dengan jumlah *r* dan *modulus* dari beberapa diambil dengan membagi dengan angka.
6) Oleh karena itu , $pi = r * si \bmod a$
7) Semua elemen p1 , p2 , p3 , ... *pn* urutan *p* adalah dikalikan dengan dengan unsur-unsur yang sesuai dari biner urutan *b*.
8) Angka-angka tersebut kemudian ditambahkan untuk membuat pesan terenkripsi *Mi*.

Superincreasing knapsack adalah persoalan *knapsack* yang dapat dipecahkan dalam orde *O*(*n*) (jadi, polinomial). Ini adalah persoalan *knapsack* yang mudah sehingga tidak disukai untuk dijadikan sebagai algoritma kriptografi yang kuat.Jika senarai bobot disebut barisan *superincreasing*, maka kita dapat membentuk *superincreasing knapsack*. Barisan *superincreasing* adalah suatu barisan di mana setiap nilai di dalam barisan lebih besar dari pada jumlah semua nilai sebelumnya. Misalnya {1, 3, 6, 13, 27, 52} adalah barisan *superincreasing* tetapi {1, 3, 4, 9, 15, 25} bukan.

Dalam penerapan algoritma *knapsack* ini akan digunakan sebuah contoh kasus untuk melihat langkah demi langkah proses keamanan data teks dengan menggunakan algoritma *Knapsack*, baik pada proses *enkripsi* maupun *dekripsi* sebagai berikut:

1) Enkripsi
   a) *Plaintext* :"**Widan**".
   b) dikonversi ke bilangan biner.
   W= 01001101 **i**= 01110101 **d** = 01110010 **a** = 01100001 **n** = 01101110.
   c) menentukan *superincreasing*. w = { 62, 93, 81, 88, 102, 37 }
   d) nilai w , pdan r : q = 105 r = 61
   e) Selanjutnya, dilakukan pembagian 6*plaintext* ke dalam blok-blok berdasarkan jumlah elemen *p*sebagai berikut :
      Blok 1 = 101011
      Blok 2 = 111010
      Blok 3 = 011100
      Blok 4 = 100110
      Blok 5 = 000111
      Blok 6 = 011100
   f) Selanjutnya, setiap blok akan dikalikan dengan setiap elemen *p*, sehingga diperoleh *ciphertext* sebagai berikut :
      Cipher I = (1 * 62) + (0 * 93) + (1 * 81) + (0 * 88) + (1 * 102) + (1 * 37) = 62 + 81 + 102 + 37 = 282
      Cipher II = (1 * 62) + (1 * 93) + (1 * 81) + (0 * 88) + (1 * 102) + (0 * 37) = 62 + 93 + 81 + 102 = 338
      Cipher III = (0 * 62) + (1 * 93) + (1 * 81) + (1 * 88) + (0 * 102) + (0 * 37) = 93 + 81 + 88 = 262
      Cipher IV = (1 * 62) + (0 * 93) + (0 * 81) + (1 * 88) + (1 * 102) + (0 * 37) = 62 + 88 + 102 = 252
      Cipher V = (0 * 62) + (0 * 93) + (0 * 81) + (1 * 88) + (1 * 102) + (1 * 37) = 88 + 102 + 37 = 227
      Cipher VI = (0 * 62) + (1 * 93) + (1 * 81) + (1 * 88) + (0 * 102) + (0 * 37) = 93 + 81 + 83 = 257

g) Memperoleh *ciphertext* hasil *enkripsi* sebagai berikut : {282,338,262,252,227,257}

2) Dekripsi
   a) Ciphertext
      Untuk proses Dekripsi *Algoritma merkle-hellmanm knapsack* ini, digunakan *private key*{2,3,6,13,27,52}
   b) Modular *invers*
      nilai *modulo invers* dari (*r-1*) sebesar 61, dengan menggunakan nilai *r-1* ini, akan dilakukan perkalian seluruh cipherteks dengan nilai *r-1* mod *q*, sehingga diperoleh nilai-nilai sebagai berikut :
      P1 = 282 * 61 mod 105 = 87
      P2 = 338 * 61 mod 105 = 38
      P3 = 262 * 61 mod 105 = 22
      P4 = 252 * 61 mod 105 = 42
      P5 = 227 * 61 mod 105 = 92
      P6 = 257 * 61 mod 105 = 22
   c) nilai P1 sampai P6 akan didekomposisi menggunakan setiap nilai pada *w*. Dekomposisi ini dilakukan dengan cara melakukan pengurangan terhadap nilai terbesar hingga terkecil.
      P1 = 87 – 52 – 27 – 6 – 2 = 0
      diperoleh : 101011
      P2 = 38 – 27 – 6 – 3 – 2 = 0
      diperoleh : 111010
      P3 = 22 – 13 – 6 – 3 = 0
      diperoleh : 011100
      P4 = 42 – 27 – 13 – 2 = 0
      diperoleh : 100110
      P5 = 92 –52 – 27 – 13 = 0
      diperoleh : 000111
      P6 = 22 – 13 – 6 – 3 = 0
      diperoleh : 011100 }
   d) mengubah nilai binernya menjadi nilai desimal, sehingga diperoleh hasil sebagai berikut : { 87, 105, 100, 97, 110 }
   e) melakukan konversi nilai desimal *plaintext* menjadi karakter, sehingga diperoleh pesan awal sebagai berikut : "**Widan**".

2. METODE PENELITIAN

2.1. Metode Literatur (Studi Pustaka)

Merupakan metode pengumpulan data dengan melakukan studi kepustakaan melalui buku-buku referensi untuk mendapatkan data yang berhubungan dengan topic penilitian.

2.2. Metode Pengamatan

Metode ini dilakukan dengan cara mengamati hal-hal yang berkaitan dengan aplikasi kriptografi menggunakan algoritma *Knapsack* yang nantinya digunakan dalam penelitian ini.

2.3. Metode Pengujian sistem

Metode pengujian digunakan untuk menguji aplikasi tanpa memperhatikan proses penyajian keluaran dari fungsi pada sistem yang dibuat.

2.4. Metode Pengembangan Sistem

Merupakan suatu proses pengumpulan kebutuhan *software* untuk mengerti sifat-sifat program yang dibentuk *software engineering*, atau analis harus mengerti fungsi software yang diinginkan, *performance* dan *interface*terhadap elemen lainnya.

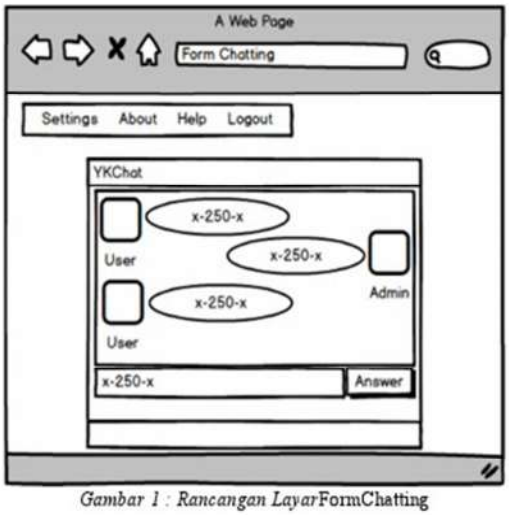
3. HASIL DAN PEMBAHASAN

3.1. Rancangan Layar

Rancangan layar sangat penting dalam membuat suatu program. Oleh karena itu rancangan layar harus bersifat *user friendly* atau mudah dipahami oleh *user* sehingga *user* tidak akan kebingungan dan kesulitan dalam menggunakan aplikasi. Berikut ini adalah rancangan layar untuk aplikasi enkripsi dan dekripsi *chatting* PT Yudhakarsa Indonesia.

3.1.1. *FormChatting User*

Menu ini menampilkan halaman utama aplikasi ini, yang terdiri dari menu *button settings, button about*, dimana user bisa memilih menu mana yang diingin kan oleh *user* tersebut. Menu help adalah menu bantuan yang di sediakan untuk memberi petunjuk atau arahan tahapan-tahapan aplikasi ini. Menu *about* adalah menu yang disediakan untuk memberi tahu kepada *user* biodata pembuat aplikasi ini. Berikut adalah rancangan layar menu utama :

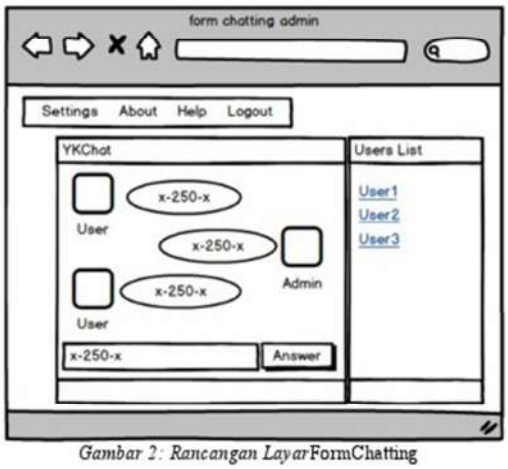


*Gambar 1 : Rancangan Layar*FormChatting

3.1.2. *FormChatting Admin*

Menu ini menampilkan halaman utama aplikasi ini, yang terdiri dari menu *button settings, button*

help, button about, dimana admin bisa memilih menu mana yang diingin kan oleh *admin* tersebut. Admin harus terlebih dahulu memilih *user* pada menu *userslist*yang terletak di sisi kanan untuk menentukan kepada siapa pesan akan dikirim. Menu *about* adalah menu yang disediakan untuk memberi tahu kepada *user* biodata pembuat aplikasi ini. Berikut adalah rancangan layar *form chatting admin* :

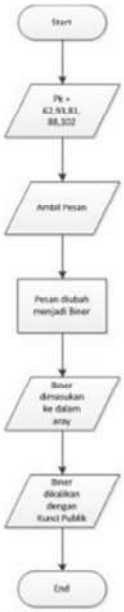


*Gambar 2: Rancangan Layar*FormChatting

3.2. *Flowchart*

3.2.1. *Flowchart* Proses Enkripsi Pesan

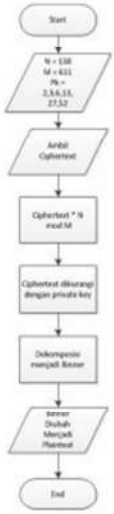
Pada*flowchart* Enkripsi pesan ini menjelaskan bagaimana cara mengubah *plaintext* menjadi *ciphertext*menggunakan algoritma *knapsack*. Berikut adalah gambar *flowchart* enkripsi pesan menggunakan algoritma *knapsack*:



Gambar 3: Flowchart Enkripsi Knapsack

3.2.2. *Flowchart* Proses Dekripsi pesan

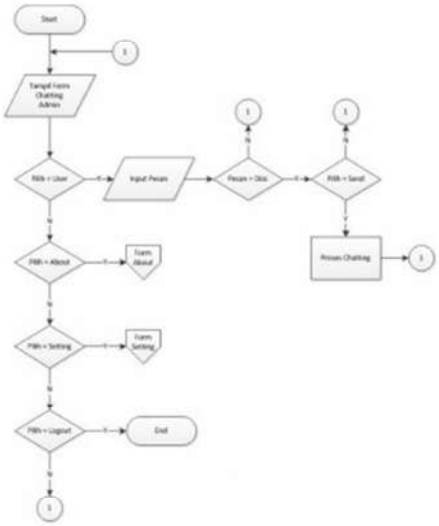
Pada *flowchart* dekripsi pesan ini menjelaskan bagaimana cara mengubah *ciphertext* menjadi *plaintext* menggunakan algoritma *knapsack*. Berikut adalah gambar *flowchart* dekripsi pesan menggunakan algoritma *knapsack*:



Gambar 4 :Flowchart *Dekripsi* Knapsack

3.2.3. *Flowchart Form Chatting Admin*

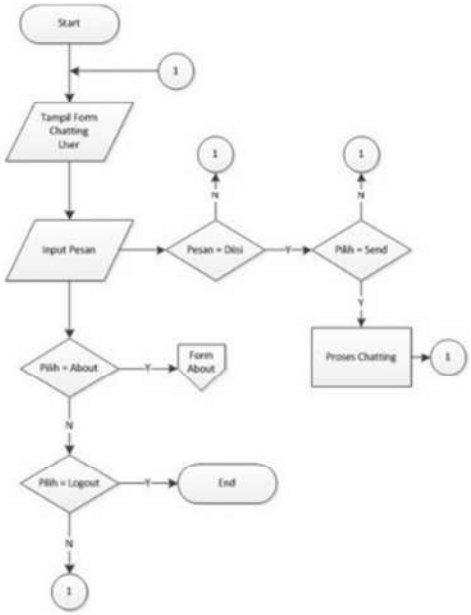
Pada *flowchartform chatting admin* ini menjelaskan bagaimana proses yang terjadi setelah admin melakukan proses login. Admin harus memilih user terlebih dahulu sebelum mengirimkan pesan. Berikut adalah gambar *flowchartform chatting admin*:



Gambar 5 : Flowchart Form Chatting Admin

3.2.4. *Flowchart Form Chatting User*

Pada *flowchartform chatting user* ini menjelaskan bagaimana proses yang terjadi setelah *user* melakukan proses login. Berbeda dengan *admin*, *user* tidak perlu lagi memilih kepada siapa *user* mengirim pesan karena hanya mampu mengirimkan pesan kepada admin saja. Berikut adalah gambar *flowchartform chatting user*:



Gambar 6 : Flowchart Form Chatting user

3.3. Implementasi Program

Agar aplikasi dapat berjalan secara optimal maka dibutuhkan perangkat yang mempuni dalam implementasi program. Implementasi aplikasi terdiri dari dua bagian, yaitu kebutuhan perangkat keras dan kebutuhan perangkat.

3.4. Kebutuhan Perangkat Keras

Berikut ini adalah spesifikasi minimum dan spesifikasi saat implementasi aplikasi keamanan *Chatting* pada PT. Yudhakarsa Indonesia.

Tabel 1 : Spesifikasi Minimum Perangkat Keras

No.	Perangkat	Kebutuhan
1	Processor	Intel Pentium IV @ 2.6 GHZ
2	Harddisk	320.00 GB
3	RAM	2.00 GB

Tabel 2 : Spesifikasi Implementasi Perangkat Keras

No.	Perangkat	Kebutuhan
1	Processor	Intel Core i3-2330 2.2 GHz
2	Harddisk	500 GB
3	RAM	2.00 GB

3.5. Kebutuhan Perangkat Lunak

Dalam pembuatan aplikasi ini, perangkat lunak yang digunakan untuk implementasi aplikasi ini sebagai berikut :

Tabel 3 : Spesifikasi Minimum Perangkat Lunak

No.	Kebutuhan
1	Microsoft Windows 7
2	Xampp 7.1.12

Tabel 4 : Spesifikasi Implementasi Perangkat Lunak

No.	Kebutuhan
1	Microsoft Windows 7
2	Xampp 7.1.12

3.6. Tampilan Layar

3.6.1. Tampilan Layar *Form Login User*

Tampilan layar halaman *login* merupakan layar yang akan tampil pertama kali ketika aplikasi dijalankan yang menjadi penghubung ke halaman utama. Pengguna yang lupa dengan password tidak perlu khawatir, karena terdapat tombol *forgotpassword*. Bagi pengguna yg belum memiliki akun juga dapat melakukan registrasi dengan mengklik tombol *register*. Pada bagian ini pengguna harus memasukan *username* dan *password*. Berikut adalah gambar tampilan layar halaman *login*.

Gambar 7: Tampilan Layar FormLoginUser

3.6.2. Tampilan Layar Validasi Form Login

Jika *user* tidak memasukan *username* dan *password* maka akan muncul pesan dialog seperti pada gambar 8 berikut:

Gambar 8 : Pesan Informasi username *dan* Password *Kosong*

Jika *user* salah memasukan *username* dan *password* maka akan muncul pesan dialog seperti pada gambar 9 berikut:

Gambar 9 : Pesan Informasi username *dan* Password *Salah*

3.6.3. Tampilan layar *Chatting Admin*

Halaman *Chatting Admin*ini akan tampil ketika *admin* sudah berhasil melakukan *login*. Pada halaman ini *admin* harus memiilih *user* pada bagian *friendonline* agar dapat membaca pesan masuk dari *user*. Untuk lebih jelasnya berikut adalah gambar tampilan layar *chatting admin*:

Gambar 10: Tampilan Layar Chatting Admin

3.6.4. Tampilan layar *Chatting User*

Halaman *Chatting User*ini akan tampil ketika *user* sudah berhasil melakukan *login*. Pada halaman ini *user* tidak perlu memiilih penerimakarena hanya dapat mengirimkan pesan kepada *admin* saja. Untuk lebih jelasnya berikut adalah gambar tampilan layar *chatting user*:

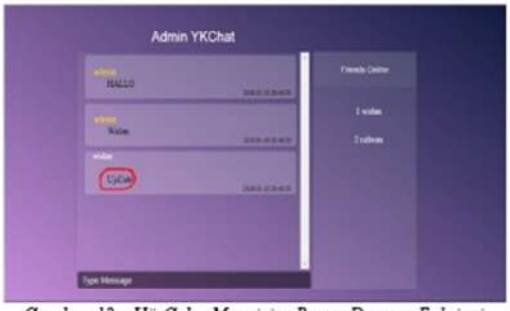
Gambar 11: Tampilan Layar Chatting User

3.7. Pengujian Aplikasi

Setelah spesifikasi perangkat keras dan perangkat lunak terpenuhi, maka aplikasi dapat dilakukan uji coba.

3.7.1. Pengujian Enkripsi

Dalam pengujian ini dilakukan dengan cara mengirim pesanmenggunakan aplikasi. Pesan akan di enkripsi menggunakan algoritma knapsack. Gambar berikut adalah contoh pesan yang akan dikirimkan:



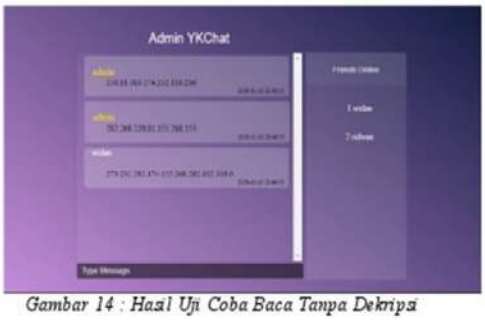
Gambar 12 : Uji Coba Mengirim Pesan Dengan Enkripsi

Apabila pesan yang sudah terkirim dan berhasil masukpada database maka pesan tidak akan dapat dibaca, seperti gambar di bawah:

Gambar 13 : Uji CobaBaca Pesan Masuk Database

3.7.2. Pengujian Dekripsi

Pengujian proses dekripsi merupakan pengujian untuk merubah *ciphertext* menjadi *plaintext* dan juga dapat mengetahui hasil dari proses sebelum dan sesudah dekripsi pada aplikasi ini.Pesan yang dibaca tanpa melalui proses dekripsi seperti gambar dibawah:



Gambar 14 : Hasil Uji Coba Baca Tanpa Dekripsi

3.7.3. Pengujian Lama Waktu Enkripsi dan Dekripsi

Pengujian selanjutnya dilakukan untuk mengetahui pengaruh panjang pesan terhadap lamanya proses enkripsi dan dekripsi. Lamanya proses enkripsi dan dekripsi berbanding lurus dengan panjang pesan yang dikirimkan. Semakin panjang pesan yang akan dikirimkan maka semakin lama pula waktu yang dibutuhkan untuk proses enkripsi dan dekripsi. Table berikut adalah hasil pengujian lamanya proses enkripsi dan dekripsi:

Tabel 5 : Hasil Pengujian Lamanya Proses Enkripsi

No.	Plaintext	Ukuran Plaintext (Byte)	Ukuran Ciphertext (Byte)	Waktu (Milli Detik)
1.	' '	1	4	0.00057
2.	'du'	3	9	0.000635
3.	'dunia'	6	23	0.000779
4.	'duniailkom'	11	48	0.002081
Rata-rata		5.25	21	0.0010162 5

Tabel 6 : Hasil Pengujian Lamanya Proses Dekripsi

No.	Ciphertext
1.	279
2.	926.1960
3.	926.1960.1897.1415.848
4.	926.1960.1897.1415.848.1415.1493.1819.2342.1938
Rata-rata	

Ukuran Plain text (Byte)	Ukuran Cipher text (Byte)	Waktu (Milli Detik)
1	4	0.000002
3	9	0.000024
6	23	0.000169
11	48	0.001256
5.25	21	0.00036275

Tabel 7 : Hasil Perbandingan Pengujian Lamanya Proses Antara Enkripsi dengan Dekripsi

No.	Ukuran Cipher text (Byte)	Ukuran Plain text (Byte)	Waktu Enkripsi (Milli Detik)	Waktu Dekripsi (Milli Detik)
1.	4	1	0.00057	0.000002
2.	9	3	0.000635	0.000024
3.	23	6	0.000779	0.000169
4.	48	11	0.002081	0.001256
Rata-rata	84	5.25	0.0010162 5	0.0003627 5

3.8. Evaluasi Program

Evaluasi program adalah tahap terakhir yang perlu dilakukan dalam pengembangan suatu aplikasi perangkat lunak. Evaluasi program bertujuan untuk mengetahui hasil yang telah di capai oleh aplikasi yang dibuat dan menentukan kekurangan dan kelebihan aplikasi yang dibuat. Berdasarkan hasil uji coba program dan eksekusi aplikasi yang dilakukan, maka didapati beberapa kelebihan dan kekurangan pada aplikasi yang yang telah dibuat dalam penelitian ini.

3.8.1. Kelebihan

1) Pesan teksyang telah dienkripsi bukan pesan teks duplikat sehingga menghindari kemungkinan pesan teks asli dicuri.
2) Pesan teks yang dimasukan ke dalam sebuah database merupakan hasil dari pesan teks yang telah dienkripsi sehingga jika ada kemungkinan database hilang, pencuri tidak dapat membaca pesan teks yang asli.
3) Proses dekripsi pesan teks cepat bahkan tidak terlihat seperti adanya proses dekripsi.

3.8.2. Kekurangan

1) Aplikasi ini hanya mampu mengirim pesan teks.
2) Aplikasi belum memiliki fiturnotifikasi yang berperansebagai penanda adanya pesan masuk.

4. KESIMPULAN

Melalui proses pengerjaan dan pengujian dalam penelitian ini, maka dapat disimpulkan beberapa hal, yaitu :

a) Meminimalisir kemungkinan kebocoran pesan yang terdapat di aplikasi apabila menjadi korban *hacker*, karena pesan teks tersebut sudah terenkripsi.
b) Kecepatan aplikasi ini sangat tergantung pada koneksi internet yang tersambung.

Untuk pengembangan lebih lanjut mengenai aplikasi ini menjadi lebih baik lagi adapun saran yang bisa diberikan antara lain:
Aplikasi ini hanya mampu mengirim pesan teks, untuk itu kedepannya perlu dikembangkan untuk dapat menyisipkan file *.txt, *.doc, .docx, *.pdf,

*.ppt, *.pptx, *.xls, *.xlsx dan menambahkan *fileextension* lainnya. Seperti gambar *.jpg, *.jpeg, *png, dll.*Interface* yang masih sangat sederhana sehingga diharapkan dapat ditambahkan beberapa fitur seperti notifikasi, dan lain sebagainya.

5. DAFTAR PUSTAKA

[1] Abdala, P., Budiman, M. A., & Herriyance, H. (2017). Implementasi Algoritma Kriptografi Vernam Cipher dan DES (Data Encryption Standard) pada Aplikasi Chatting berbasis Android. *Jurnal Ilmiah CORE IT*, *5*(1), 1–19.

[2] Apriani, F. (2016). Aplikasi chatting dengan Sistem enkripsi Menggunakan Algoritma Blowfish Berbasis Android.*Jurnal Informatika Politeknik Senggarang*. *1*(1), 1-10.

[3] Dias, M., Suhery, C., Rismawan, T., & Komputer, J. S. (2016). Penerapan Kriptografi Menggunakan Algoritma Knapsack, Algoritma Genetika, dan Algoritma Arnold's Catmap Pada Citra. *Jurnal Coding, Sistem Komputer Untan*, *4*(2), 119–129.

[4] Munawar. (2012). PERANCANGAN ALGORITMA SISTEM KEAMANAN DATA Munawar Jurnal Komputer dan Informatika (KOMPUTA). *Jurnal Komputer Dan Informatika*, *1*(1), 11–16.

[5] Murdani, M. (2017). Perancangan Aplikasi Keamanan Data Teks Menggunakan Algoritma Merkle Hellman Knapsack. *Pelita Informatika: Informasi Dan Informatika*, *16*(3), 302–305.

[6] Widarma, A. (2016). Dalam Skema Hybrid Untuk Keamanan Data, *Jurnal Coding, Sistem Komputer Untan*, (1), 1–8.

[7] Yulianingsih, P., & Maharani, S. (2014). Aplikasi Chatting Rahasia Menggunakan Algoritma Vigenere Cipher, *Jurnal INFORMARIKA Mulawarman*, *9*(1), 1–4.