ENKRIPSI PESAN TEKS PADA EMAIL MENGGUNAKAN ALGORITMA HILL CIPHER

Awang Harsa Kridalaksana [1)], Reza Andrea [2)], Miftahul Huda[3)]

[1]Teknik Informatika, STMIK Widya Cipta Dharma
[2]Teknik Informatika, STMIK Widya Cipta Dharma
[3]Teknik Informatika, STMIK Widya Cipta Dharma
[1,2,3]Jl. M. Yamin No.25, Samarinda, 75123
E-mail : qonita23@yahoo.com [1)], reza@bibirdesign.com[2)], miftahulh2@gmail.com[3)]

ABSTRAK

Penelitian ini bertujuan membuat program enkripsi pesan *text* dalam surat elektronik (*email*) dengan algoritma *hill cipher* untuk menjaga kerahasiaan pesan *text* dalam surat elektronik (*email*) agar tidak dapat diketahui oleh pihak yang tidak berhak (*anauthorized persons*), serta memudahkan pengguna untuk mengirimkan surat elektronik (*email*) secara rahasia dan pribadi.

Dalam penelitian ini algoritma yang digunakan adalah *hill cipher*, *software* yang digunakan adalah *visual basic.net* untuk bahasa pemprogamannya dan *microsoft visio* 2003 untuk membuat *flowchart*nya. Metode pengembangan sistem yang digunakan yaitu *Waterfall.*

Adapun hasil akhir dari penelitian ini yaitu diharapkan dapat membentuk sebuah aplikasi pengamanan *email* yang berupa data teks, sehingga bisa lebih mempersulit aktivitas pelaku *cyber crime*.

Kata Kunci: *Hill Cipher, Email*

1. PENDAHULUAN

Kriptografi adalah suatu ilmu yang mempelajari penulisan secara rahasia. Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut *Cryptology*. Kriptografi bertujuan menjaga kerahasiaan informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak sah. Dalam kriptografi, terdapat dua konsep utama, yaitu enkripsi dan dekripsi. Enkripsi adalah suatu proses, di mana informasi atau data yang akan dikirim diubah menjadi bentuk yang hampir tidak dikenali sebagaimana informasi awalnya dengan menggunakan suatu algoritma tertentu. Hasil enkripsi disebut *ciphertext*. Dekripsi adalah suatu proses mengubah kembali *ciphertext* menjadi bentuk (informasi atau data) awalnya, yang disebut *plaintext*.

2. RUANG LINGKUP PENELITIAN

1. Cakupan Permasalahan

Berdasarkan uraian latar belakang masalah diatas, maka dapat diambil rumusan masalah dari penelitian ini adalah, " Bagaimana membuat aplikasi enkripsi pesan teks pada *email* dengan metode kriptografi *hill cipher* ?

2. Batasan masalah pada penelitian ini adalah
   1). Aplikasi ini hanya berfungsi sebagai pengirim *email* dan untuk mendekripsi pesan *text* dalam email yang terenkripsi.
   2). Pengguna aplikasi ini diharuskan memakai akun *email gmail*, *yahoo* dan outlook.
   3). Yang dapat di enkripsi dan dekripsi berupa string atau hanya pesan *text* dalam *email* bukan *attachment file*.
   4). Diperlukan koneksi *internet* agar pesan dapat dikirimkan ke tujuan.
   5). Pengirim dan penerima harus sama-sama menggunakan aplikasi ini.
   6). Aplikasi pengamanan pesan *text* dalam *email* ini menggunakan Kriptografi *hill cipher* untuk proses enkripsi dan dekripsinya
   7). Data yang diproses berupa karakter (*string*)
   8). Batas matriks kunci yang digunakan adalah perkalian 3 x 3 dan berupa bilangan bulat

3. Tujuan penyusunan penelitian ini adalah untuk Tujuan dari penulisan tugas akhir ini adalah merancang dan mengimplementasikan metode kriptografi *hill cipher* guna mengamankan data dan informasi isi pesan *email*.

3. BAHAN DAN METODE

3.1 Algoritma

Istilah algoritma berasal dari nama seorang pengarang berkebangsaan Arab bernama Ja'fat Mohammed bin Musa al Khowarizmi tahun (790 – 840), yang sangat terkenal dengan sebutan bapak Aljabar. Secara defenisi algoritma adalah alur pemikiran yang logis yang dapat dituangkan ke dalam bentuk tulisan. Sebuah algoritma dikatakan benar (*correct*) jika algoritma tersebut berhasil mengeluarkan *output* yang benar untuk semua kemungkinan *input*. (Rachmat, 2010).

3.2 Algoritma *Hill Cipher*

Menurut Howard Anton dan Chris Rorres *Hill Cipher* termasuk kepada algoritma kriptografi klasik yang sangat sulit dipecahkan oleh kriptanalis apabila dilakukan hanya dengan mengetahui berkas *ciphertext* saja. Karena *Hill Cipher* tidak mengganti setiap abjad yang sama pada *plaintext* dengan abjad lainnya yang sama pada *ciphertext* karena menggunakan perkalian matriks pada dasar enkripsi dan dekripsinya.

Hill Cipher diciptakan oleh Lester S. Hill pada tahun 1929. *Hill cipher* yang merupakan *polyalphabetic cipher* dapat dikategorikan sebagai *block cipher*, karena teks yang akan diproses akan dibagi menjadi blok-blok dengan ukuran tertentu. Setiap karakter dalam satu blok akan saling mempengaruhi karakter lainnya dalam proses enkripsi dan dekripsinya,

Contoh kasus:

Misalkan kata yang dienkripsi adalah : "abc" dan kunci yang digunakan adalah matriks 3x3 sebagai berikut: $\begin{bmatrix}6 & 5 & 5\\1 & 1 & 1\\1 & 2 & 3\end{bmatrix}$

Penyelesaian:

1. Tahapan enkripsi

Langkah 1:

Transformasikan tiap huruf ke dalam kode ASCII, sehingga dari kata "abc" menjadi "97 98 99"

Langkah 2:

Hitung C = K x P (*mod* 127 untuk tiap vektor P), dimana C=*ciphertext*, K=*key*(matriks), dan P=*Plaintext*.

Langkah 2:

Hitung C = K x P (*mod* 127 untuk tiap vektor P), dimana C=*ciphertext*, K=*key*(matriks), dan P=*Plaintext*.

Kunci 3 x 3			abjad	nilai	Hasil Perkalian	Mod 127	ciphertext
6	5	5	a	97	1963	58	:
1	1	1	b	98	294	40	(
1	2	3	c	99	590	82	**R**

Sehingga nilai vektor kolom "abc" {97, 98, 99 } maka setelah di *Encipher* menghasilkan *ciphertext* "**:(R**" {58, 40, 82}

2. Tahapan deskripsi

$P = K^{-1}.C$

Keterangan :

P = *Plaintext*

K^{-1} = *Key invers*

C = *Ciphertext*

Tahap 1:

Transformasikan tiap huruf ke dalam kode ASCII, sehingga dari kata "**:(R**" menjadi "58 40 82"

Tahap 2:

Cari nilai *invers* dari

Maka tahapan mencari *invers* sebagai berikut:

Pertama, harus mencari kofaktornya terlebih dahulu

$$\text{Kofaktor} = \begin{bmatrix}A_{11} & A_{12} & A_{21}\\A_{21} & A_{22} & A_{23}\\A_{31} & A_{32} & A_{33}\end{bmatrix}$$

$$\text{Kofaktor} = \begin{bmatrix}+\begin{vmatrix}1 & 1\\2 & 3\end{vmatrix} & -\begin{vmatrix}1 & 1\\1 & 3\end{vmatrix} & +\begin{vmatrix}1 & 1\\1 & 2\end{vmatrix}\\-\begin{vmatrix}5 & 5\\2 & 3\end{vmatrix} & +\begin{vmatrix}6 & 5\\1 & 3\end{vmatrix} & -\begin{vmatrix}6 & 5\\1 & 2\end{vmatrix}\\+\begin{vmatrix}5 & 5\\1 & 1\end{vmatrix} & -\begin{vmatrix}6 & 5\\1 & 1\end{vmatrix} & +\begin{vmatrix}6 & 5\\1 & 1\end{vmatrix}\end{bmatrix}$$

Kedua, setelah kofaktor berhasil ditemukan, maka cari *adjoin* nya

$$\text{Matriks Kofaktor} = \begin{bmatrix}1 & -2 & 1\\-5 & 13 & -7\\0 & -1 & 1\end{bmatrix}$$

$$\text{Menjadi } adjoin = \begin{bmatrix}1 & -5 & 0\\-2 & 13 & -1\\1 & -7 & 1\end{bmatrix}$$

Ketiga, mencari nilai determinan

Det (a) = $A_{11}A_{22}\,A_{33} + A_{12}A_{23}A_{31} + A_{13}A_{21}A_{32} - A_{31}A_{22}\,A_{13} - A_{32}A_{23}A_{11} - A_{33}A_{21}A_{12}$

Det (a) = [(6)(1)(3) + (5)(1)(1) + (5)(1)(2) – (1)(1)(5) – (2)(1)(6) - (3)(1)(5)]

Det (a) = [18 + 5 + 10 – 5 – 12 – 15]

Det (a) = 1

Ketiga, mencari nilai *invers*

$$A^{-1} = \frac{1}{1}\begin{bmatrix}1 & -5 & 0\\-2 & 13 & -1\\1 & -7 & 1\end{bmatrix}$$

$$A^{-1} = \begin{bmatrix}1 & -5 & 0\\-2 & 13 & -1\\1 & -7 & 1\end{bmatrix} = \begin{vmatrix}1 & 5 & 0\\2 & 13 & 1\\1 & 7 & 1\end{vmatrix}$$

Tahap 3:

Setelah dapat matrik *invers*nya, kemudian dikalikan nilai ASCII *ciphertext*

Matriks 3 x 3			abjad	nilai	Hasil perkalian	Mod 127	plaintext
1	5	0	:	58	341	87	a
2	13	1	(	40	813	88	b
1	7	1	**R**	82	470	89	c

Sehingga nilai vektor kolom "**:(R**" {58, 40, 82 } maka setelah di *Encipher* menghasilkan *plaintext* "abc" {87, 88, 89}

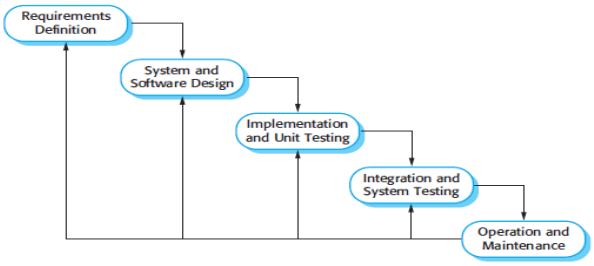
3.3 Aplikasi

Perangkat lunak aplikasi adalah program yang ditulis dan diterjemahkan oleh *language software* untuk menyelesaikan suatu aplikasi tertentu. Aplikasi juga

merupakan program siap pakai yang digunakan manusia dalam melakukan pekerjaan dengan menggunakan komputer. (Jogiyanto,2005).

4. Metode Pengembangan Sistem

Dalam mengembangkan aplikasi ini, metode yang digunakan adalah model prototipe *(prototype).* Menggunakan model ini karena aplikasi yang dibuat yaitu aplikasi kamus istilah kesehatan dalam perancangannya antara *developer* dan *user* harus saling berkaitan, terlebih di dalam proses *input* istilah kosa kata dan definisinya yang dimana apabila terdapat kekurangan ataupun penambahan istilah kosa kata baru yang ingin ditambahkan, *user* dapat memberi masukkan kepada *developer* dan *developer* pun harus mendengarkan, memperbaiki dan kemudian dipresentasikan kepada pelanggan sehingga perangkat lunak yang dihasilkan nantinya sesuai dengan kebutuhan pelanggan.



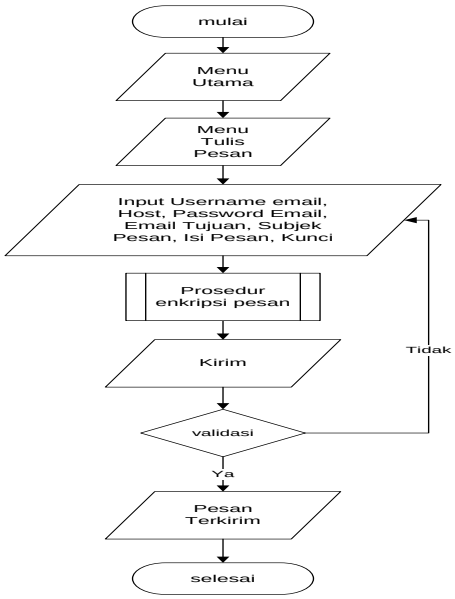
Gambar 1. *Waterfall*

5. IMPLEMENTASI

5.1 Implementasi Rancangan Sistem

Di dalam merancang aplikasi kamus istiah kesehatan ini, digunakan alat bantu untuk mempermudah proses perancangan sistem. Perancangan sistem ini menggunakan diagram alir *(flowchart).* Adapun *flowchat* pada aplikasi kamus istilah kesehatan digambarkan seperti berikut.

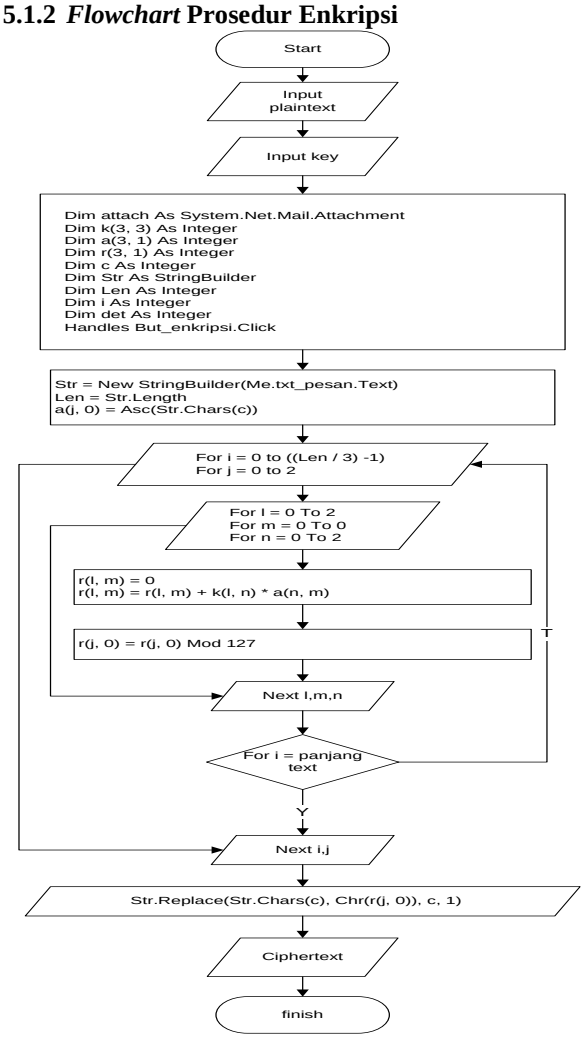
5.1.1 *Flowchart* Enkripsi Pesan



Gambar 2. *Flowchart* Enkripsi Pesan

Pada gambar 2. terlihat bagaimana diagram alir bagaimana *user* dapat mengenkripsi pesan. Di mulai dengan *user* masuk ke menu utama. Di menu utama *user* memilih menu Tulis Pesan, di menu Tulis Pesan *user* mengisi *input*an berupa *username email*, *host server*, *password*, *email* tujuan, subjek pesan, isi pesan dan kunci enkripsi untuk mengenkripsi pesan *text* yang akan dikirim.

5.1.2 *Flowchart* Prosedur Enkripsi



Gambar 3. *Flowchart* Prosedur Enkripsi

Pada gambar 3 terlihat bagaimana diagram enkripsi alir algoritma *hill cipher*. Di mulai dengan meng*input*kan *plaintext*, dan menginputkan *key*.

```
Dim attach As System.Net.Mail.Attachment,
Dim k(3, 3) As Integer,
Dim a(3, 1) As Integer,
Dim r(3, 1) As Integer,
Dim c As Integer,
Dim Str As StringBuilder,
Dim Len As Integer,
Dim i As Integer,
Dim det As Integer,
Handles But_enkripsi.Click,
```

Merupakan variabel dari *hill cipher* yang digunakan pada program

```
Str = New StringBuilder(Me.txt_pesan.Text),
Len = Str.Length,
a(j, 0) = Asc(Str.Chars(c)),
```

Proses pengambilan *string* pada *txt_pesan* untuk dirubah ke karakter ASCII

```
For i = 0 to ((Len / 3) -1),
For j = 0 to 2,
```

Merupakan perulangan pada *txt_pesan* untuk setiap pengambilan 3 karakter

```
For l = 0 To 2,
For m = 0 To 0,
For n = 0 To 2,
```

Merupakan perulangan *array* untuk pengisian bilangan pada kolom matriks

```
r(l, m) = 0,
r(l, m) = r(l, m) + k(l, n) * a(n, m),
```

Proses perkalian matriks *plaintext* dengan matriks *key*

```
r(j, 0) = r(j, 0) Mod 127,
```

proses dari hasil perkalian matriks dikalikan dengan *mod* 127,

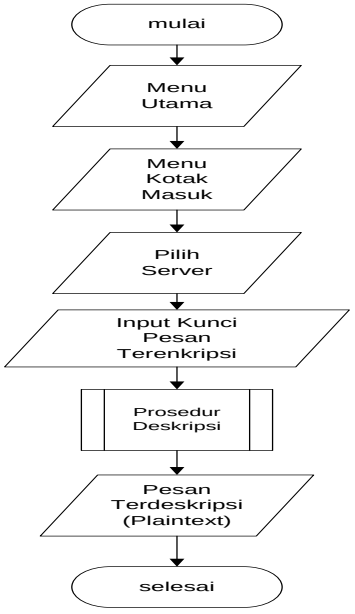
```
Str.Replace(Str.Chars(c), Chr(r(j, 0)), c, 1),
```

Merupakan proses konversi dari hasil *mod* 127 ke karakter ASCII,

Setelah itu, pesan hasil enkripsi ditampilkan ke *ciphertext*, dan selesai

5.1.3 *Flowchart* Deskripsi Pesan

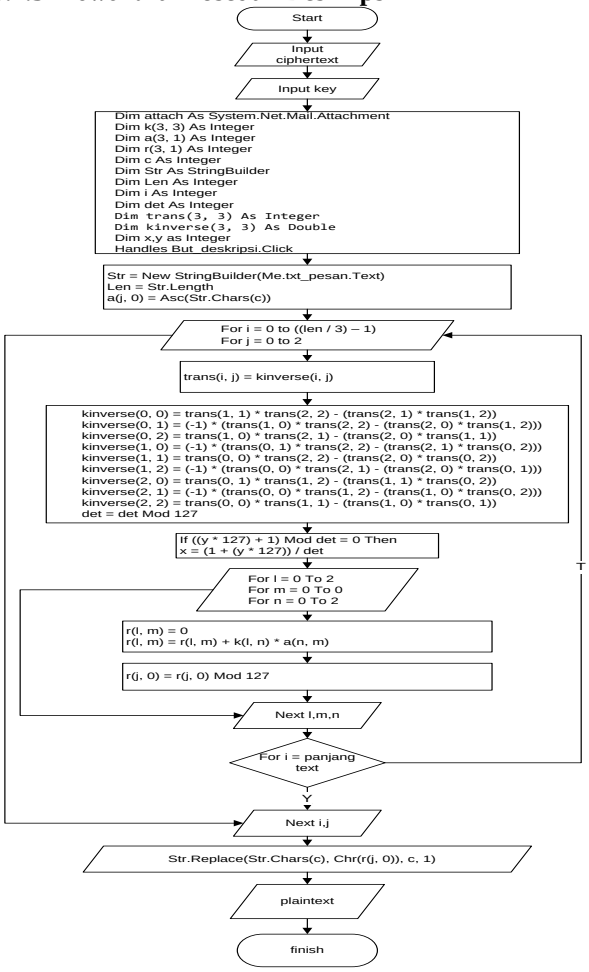
Pada gambar 4 terlihat bagaimana diagram alir bagaimana *user* dapat mendekripsi pesan. Di mulai dengan *user* masuk ke menu utama. Di menu utama *user* memilih menu kotak masuk, kemudian di dalam kotak masuk user memilih server email yang akan digunakan. Setelah itu, *user* mengisi inputan berupa pesan yang terenkripsi (*ciphertext*) dan kunci yang digunakan untuk mendekripsi pesan



Gambar 4. *Flowchart* Deskripsi Pesan

.

5.1.3 *Flowchart* Prosedur Deskripsi



Gambar 5. *Flowchart* Prosedur Deskripsi

Pada gambar 5 terlihat bagaimana diagram enkripsi alir algoritma *hill cipher*. Di mulai dengan meng*input*kan *plaintext*, dan menginputkan *key*.

```
Dim attach As System.Net.Mail.Attachment,
Dim k(3, 3) As Integer,
Dim a(3, 1) As Integer,
Dim r(3, 1) As Integer,
Dim c As Integer,
Dim Str As StringBuilder,
Dim Len As Integer,
Dim i As Integer,
Dim det As Integer,
Dim trans(3, 3) As Integer
Dim kinverse(3, 3) As Double
Dim x,y as Integer
Handles But_enkripsi.Click,5
```

Merupakan variabel dari *hill cipher* yang digunakan pada program

```
Str = New StringBuilder(Me.txt_pesan.Text),
Len = Str.Length,
a(j, 0) = Asc(Str.Chars(c)),
```

Proses pengambilan *string* pada *txt_pesan* untuk dirubah ke karakter ASCII

```
For i = 0 to ((Len / 3) -1),
For j = 0 to 2,
```

Merupakan perulangan pada *txt_pesan* untuk setiap pengambilan 3 karakter

```
trans(i, j) = kinverse(i, j),
```

Proses merubah matriks *key* ke bentuk *transpose*

```
  kinverse(0, 0) = trans(1, 1) * trans(2, 2) -
(trans(2, 1) * trans   (1, 2)),
  kinverse(0,  1)  =  (-1)  *  (trans(1,  0)  *
trans(2,      2)      -      (trans(2,      0)
*   trans(1, 2))),
 kinverse(0, 2) = trans(1, 0) * trans(2, 1) -
(trans(2, 0) * trans   (1, 1)),
  kinverse(1,  0)  =  (-1)  *  (trans(0,  1)  *
trans(2,  2)  -  (trans(2,  1)       *  trans(0,
2))),
   kinverse(1, 1) = trans(0, 0) * trans(2, 2)
- (trans(2, 0) * trans (0, 2)),
   kinverse(1,  2)  =  (-1)  *  (trans(0,  0)  *
trans(2, 1) - (trans(2, 0)   * trans(0, 1))),
    kinverse(2, 0) = trans(0, 1) * trans(1, 2)
- (trans(1, 1) * trans (0, 2)),
    kinverse(2,  1)  =  (-1)  *  (trans(0,  0)  *
trans(1, 2) - (trans(1, 0)   * trans(0, 2))),
    kinverse(2, 2) = trans(0, 0) * trans(1, 1)
- (trans(1, 0) * trans (0, 1)),
 det = det Mod 127,
```

Proses mencari nilai determinan,

```
If ((y * 127) + 1) Mod det = 0 Then
x = (1 + (y * 127)) / det
```

Proses mencari nilai *invers*,

```
For l = 0 To 2,
```

```
For m = 0 To 0,
For n = 0 To 2,
```

Merupakan perulangan *array* untuk pengisian bilangan pada kolom matriks,

```
r(l, m) = 0,
r(l, m) = r(l, m) + k(l, n) * a(n, m),
```

Proses perkalian matriks *plaintext* dengan matriks *key*

```
r(j, 0) = r(j, 0) Mod 127,
```

proses dari hasil perkalian matriks dikalikan dengan *mod* 127,

```
   Str.Replace(Str.Chars(c), Chr(r(j, 0)), c,
1),
```

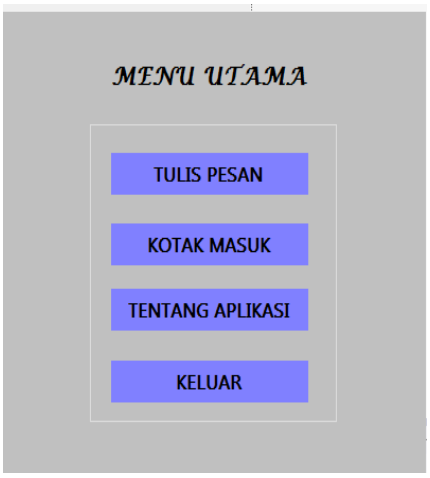
Merupakan proses konversi dari hasil *mod* 127 ke karakter ASCII,

Setelah itu, pesan hasil deskripsi ditampilkan ke *plaintext*, dan selesai

5.2 Implementasi Program

Implementasi merupakan tahapan yang bertujuan mengubah hasil dari rancangan sistem menjadi bentuk nyata. Pada saat pertama kali aplikasi dijalankan maka akan muncul sebuah tampilan.

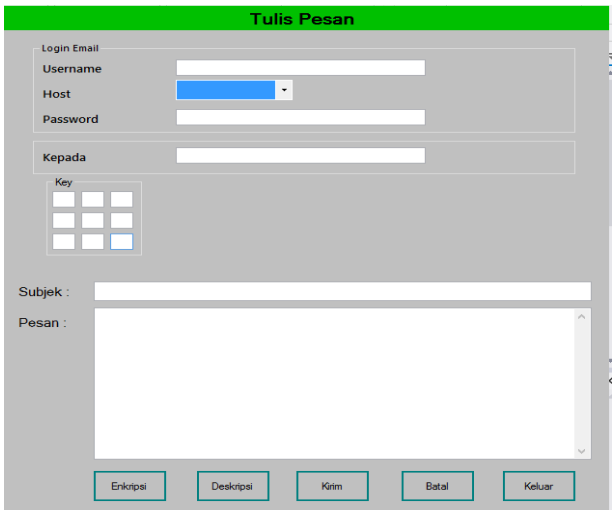
5.2.1 *Form Menu* Utama



Gambar 6. Tampilan *Menu* Utama

Gambar 6 merupakan tampilan menu utama. Di dalam *form* menu utama terdapat beberapa menu, yaitu tulis pesan, kotak masuk, tentang aplikasi dan menu keluar.

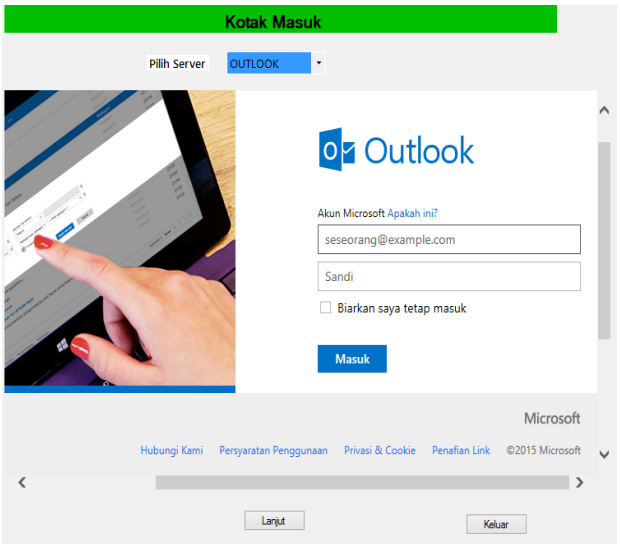
5.2.2 *Form* Tulis Pesan



Gambar 7. Tampilan Tulis Pesan

Gambar 7 merupakan tampilan menu tulis pesan. *Form* ini digunakan pengguna untuk mengirim pesan *text email* yang dapat dienkripsi. Di *form* ini terdapat beberapa inputan yaitu, *username email*, *host server email*, *password email*, *email* tujuan, kunci untuk mengenkripsi, subjek pesan, dan isi pesan.

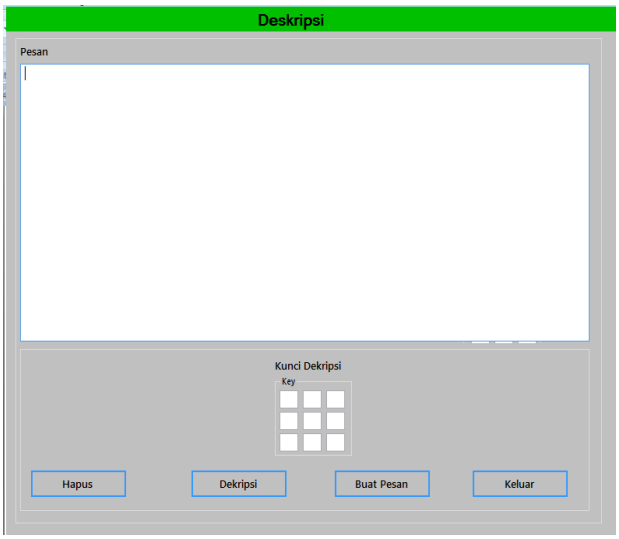
5.2.3 *Form* Buka Email



Gambar 8. Tampilan *Form* Admin

Gambar 8 merupakan tampilan menu kotak masuk. *Form* ini digunakan untuk mengakses akun *email*.

5.2.4 *Form* Deskripsi Pesan



Gambar 9. Tampilan Deskripsi Pesan

Gambar 9 merupakan tampilan menu deskripsi pesan. *Form* ini merupakan tampilan menu untuk mendekripsi pesan *text* yang yang telah di salin (*copy*) dari *form* buka *email*. Pesan yang telah di salin (*copy*) lalu di tempel (*paste*) di *textbox* pesan enkripsi. *Textbox* kunci merupakan kunci yang disepakati saat mengenkripsi pesan.

5.2.5 *Form* Tentang Aplikasi

Gambar 10. Tampilan Tentang Aplikasi

Adapun penjelasan sebagai berikut:

No	Keterangan
1	isi kolom *username* dengan *username* email tanpa akhiran "@...*com*", lalu *host* email, serta masukkan *password* email.
2	Masukkan email tujuan
3	Masukkan kunci enkripsi
4	Masukkan subjek pesan
5	Masukkan isi pesan
6	Tekan tombol enkripsi untuk mengenkripsi pesan
7	Tekan tombol kirim untuk mengirim pesan
8	Tekan tombol deskripsi untuk mendeskripsi pesan
9	Tekan tombol batal untuk membatalkan dan membersihkan kolom
10	Tekan tombol keluar bila ingin keluar

6. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka diambil beberapa kesimpulan, diantaranya adalah sebagai berikut:

1. Aplikasi ini dibangun menggunakan bahasa pemprogaman *visual basic.net* dengan metode *hill cipher*, pengujiannya menggunakan *beta testing* dan *white-box testing* dan hasilnya berjalan dengan baik.
2. Dengan adanya aplikasi enkripsi teks pada email, diharapkan mampu melindungi pesan agar tidak bisa dibaca oleh orang yang tidak bertanggung jawab.
3. Pesan yang dikirim, hanya bisa mengirim ke satu email tujuan. Sehingga untuk mengirim lebih dari email tujuan harus mengulang mengirim pesan.

7. SARAN

Aplikasi yang dibuat ini masih terdapat beberapa kekurangan dan masih perlu penyempurnaan, berikut saran-saran yang dapat penulis sampaikan:

1. Aplikasi yang dibuat masih sederhana. Diharapkan selanjutnya bisa menjadi lebih baik lagi.
2. Aplikasi dapat dikembangkan di *handphone* seperti *android* dan *windows phone*
3. Perangkat lunak ini dapat dikembangkan dengan algoritma metode lain. Untuk lebih meningkatkan keamanan pesan yang dikirim.

8. DAFTAR PUSTAKA

Anton, Howard, Chris Rorres. 2005. *Aljabar Linier Elementer Versi Aplikasi*. Jakarta: Erlangga

Ariyus Dony. 2008, *Pengantar Ilmu Kriptografi*. Yoyakarta: CV ANDI OFFSET.

Bin Ladjamudin, Al-Bahra. 2005, *Analisi dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu.

Dhanta, Risky. 2009, *Kamus Istilah Komputer Grafis & Internet*. Surabaya : Indah.

Jogiyanto. 2005, *Analisis dan Desain Sistem Informasi*. Yogyakarta : CV ANDI OFFSET.

Kurniawan, Agus. 2008, *Konsep dan Implementasi Cryptography Dengan .NET*. Jakarta: PC Media.

Imrona, Mahud, 2013, *Aljabar Linier Dasar Edisi Kedua*. Jakarta: Erlangga.

Munir, Rinaldi, 2006, *Pengantar Kriptografi*, Bandung: Informatika.

Prasetyo, Didik Dwi, 2005, *Buku Pintar Internet*, Jakarta : Elex Media Komputindo.

Pressman, Roger.S, 2005, *Rekayasa Perangkat Lunak*, Yogyakarta : CV ANDI OFFSET.

Sommerville, Ian. 2011. *Software Engineering (Rekayasa Perangkat Lunak)*, Jakarta: Erlangga

Supriyanto, Wahyu. 2008, *Teknologi Informasi Perpustakaan : Strategi Perancangan Perpustakaan Perpustakaan Digital*. Yogyakarta : Kanisius.

T.Sutojo, 2010, *Teori dan Aplikasi Aljabar Linier dan Matriks*, Semarang : CV ANDI OFFSET.