APLIKASI KRIPTOGRAFI DATABASE MENGGUNAKAN ALGORITMA VIGENERE CIPHER DAN RIVEST SHAMIR ADLEMAN BERBASIS DESKTOP

Farhan Abdul Majid[1)], **Subandi, M.Kom**[)]
[1]Program studi, Fakultas Teknologi Informasi, Universitas Budi Luhur
[1,2]Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail : farhan.officials@gmail.com[1)], subandionline@gmail.com[2)]

Abstrak

Database adalah kumpulan informasi yang disusun sedemikian rupa dengan ketentuan dan aturan tertentu sehingga dapat dikontrol dengan program komputer untuk memperoleh informasi dari *database* tersebut. Terhubungnya *Local Area Network* atau Komputer ke Internet membuka celah keamanan bagi database tersebut. Salah satu cara untuk mengamankan data pada *database* adalah dengan menggunakan kriptografi. Kriptografi merupakan suatu ilmu yang mempelajari cara menyandikan data atau pesan yang akan dikirim ke penerima sehingga data atau pesan menjadi aman dan tidak diketahui oleh pihak ketiga. Algoritma kriptografi yang digunakan dalam Jurnal ini adalah algoritma Vigenere Cipher dan Rivest Shamir Adleman (RSA). Dalam pembangunan aplikasi ini, penulis menggunakan bahasa pemrograman *Java* dan berbasis *desktop*. Uji coba yang dilakukan dengan membandingkan *table database* yang telah dienkrip hasilnya *table database* yang telah dienkrip tersandikan. Kemudian uji coba kedua mengembalikan isi *table database* ke bentuk semula. Waktu yang diperlukan untuk enkripsi dan dekripsi berbading lurus dengan banyaknya jumlah kolom dan baris pada *table database* yang diproses, Semakin banyak jumlah kolom dan baris maka akan semakin lama proses enkripsi dan dekripsi. Sebaliknya, semakin sedikit jumlah kolom dan baris maka proses enkripsi dan dekripsi semakin cepat. Hasil dari Jurnal ini adalah Aplikasi Kriptografi *Database* Menggunakan Algoritma Vigenere Cipher dan Rivest Shamir Adleman.

Kata kunci: Kriptografi, Vigenere, RSA, Enkripsi, Dekripsi

1 PENDAHULUAN

1.1 Latar Belakang Masalah

Database adalah kumpulan informasi yang disusun sedemikian rupa dengan ketentuan dan aturan tertentu sehingga dapat dikontrol dengan suatu program untuk memperoleh informasi dari *database* tersebut. *Database* merupakan aspek yang sangat penting dalam sistem informasi karena *database* merupakan tempat penyimpanan data yang akan diolah lebih lanjut menjadi informasi yang akurat. *Database* menjadi penting karena dapat mengurangi duplikasi data, hubungan antar data yang tidak jelas, organisasi data, dan juga update yang rumit.

Untuk dapat mengakses kedalam database pengguna harus terkoneksi dengan jaringan komputer. Terkoneksinya jaringan komputer dapat menambah celah keamanan bagi *database* tersebut, seperti ancaman *hacking* atau *cracking* dan sebagainya. Banyak *database* yang bersifat *private* dan tidak bisa dirubah oleh pihak yang tidak berwenang untuk merubahnya. Keamanan *database* adalah suatu cara untuk melindungi *database* dari ancaman. Ancaman adalah segala situasi atau kejadian baik secara sengaja maupun tidak yang bersifat merugikan dan mempengaruhi sistem serta secara konsekuensi terhadap perusahaan yang memiliki sistem *database*.

Salah satu cara untuk mengamankan data pada *database* dengan menggunakan kriptografi. Algoritma kriptografi yang digunakan dalam jurnal ini adalah algoritma Vigenere Cipher dan Rivest Shamir Adleman (RSA). Algoritma Vigenere Cipher merupakan salah satu algoritma kriptografi *classic* untuk menyandikan suatu *plaintext* dengan menggunakan teknik substitusi. Sedangkan Algoritma RSA termasuk dalam algoritma kriptografi asimetris yang memiliki dua kunci, yaitu kunci publik (*public key*) dan kunci pribadi (*private key*).

1.2 Perumusan Masalah

Berdasarkan latar belakang masalah diatas, maka dihasilkan rumusan masalah sebagai berikut :

1) Melakukan proses enkripsi dan dekripsi terhadap data pada table *database*
2) Menerapkan Algoritma Vigenere Cipher dan Rivest Shamir Adleman pada aplikasi
3) Merancang sebuah aplikasi kriptografi *database* menggunakan Algoritma Vigenere Cipher dan Rivest Shamir Adleman

1.3 Batasan Masalah

Pada Proses perancangan aplikasi ini, penulis membatasi permasalahan yang akan dibahas, diantaranya adalah :

1) Aplikasi yang dibangun hanya dapat melakukan proses enkripsi dan dekripsi pada table database.
2) Hanya menggunakan Algoritma Vigenere Cipher dan Rivest Shamir Adleman.
3) Menggunakan bahasa pemrograman Java.

1.4 Tujuan Penelitian

Adapun tujuan yang diperoleh dari hasil penelitian yang dilakukan diantaranya adalah :

1) Menerapkan Algoritma Vigenere Cipher dan Rivest Shamir Adleman dalam penyandian data.
2) Merancang aplikasi kriptografi yang dapat melakukan pengamanan data *database*.

2 LANDASAN TEORI

2.1 Kriptografi

Kriptografi merupakan suatu ilmu yang mempelajari cara mengamankan data atau pesan yang akan dikirim ke penerima sehingga data atau pesan menjadi aman dan tidak diketahui oleh pihak ketiga. Data atau pesan yang akan dikirim diubah menjadi kode-kode yang tidak dipahami oleh pihak ketiga.

Kriptografi membuat data atau pesan menjadi kode-kode terlebih dahulu oleh pengirim. Proses ini dikenal dengan enkripsi. Enkripsi diartikan sebagai proses diubahnya data atau pesan yang hendak dikirim menjadi bentuk yang hampir tidak dikenali oleh pihak ketiga. Setelah data atau pesan itu sampai kepada penerima, maka penerima melakukan dekripsi yang merupakan kebalikan dari enkripsi. Dekripsi diartikan sebagai proses mengubah data atau pesan kembali kebentuk semula sehinggan data atau pesan dapat tersampaikan dan dimengerti oleh penerima. Proses enkripsi dan dekripsi memerlukan kunci dalam mekanismenya. [3]

Ada empat tujuan mendasar dari kriptografi yang juga merupakan aspek keamanan informasi menurut Rifki Sadikin [4] yaitu:

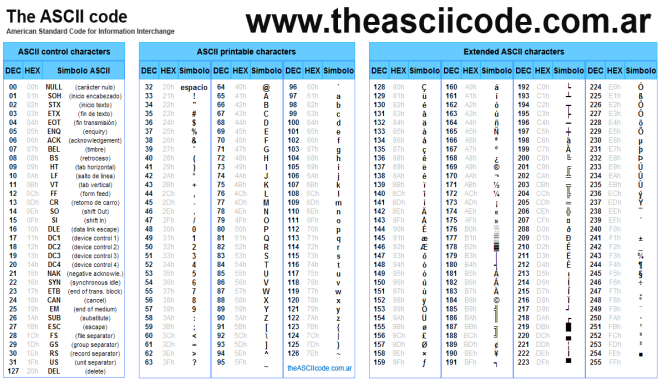
1) Kerahasiaan.
2) Integritas.
3) Autentifikasi.
4) *Non-repudiation*.

2.2 Algoritma Vigenere Cipher

Vigenere cipher menggunakan bujursangkar vigenere untuk melakukan enkripsi. Tiap baris-baris di dalam bujursangkar menyatakan kumpulan huruf Ciphertext yang diperoleh dengan Caesar cipher. Namun pada jurnal ini penulis menggunakan tabel ACII, dimana kunci-nya berjumlah 256 karakter. Sehingga relatif lebih aman dibanding dengan vigenere alfabet biasa.

Berikut rumus enkripsi dan dekripsi vigenere cipher:

a. Enkripsi
   $C_i = (P_i + K) \bmod 256$
b. Dekripsi
   $P_i = (C_i - K) \bmod 256$



Gambar 1 ASCII *CODE*

Contoh :

Diketahui plaintext “FARHAN”, jika menggunakan nilai Z = 97 (berdasrkan *source code*), dalam tabel ASCII yaitu huruf “a”. dan kuncinya “kunci”. Maka proses enkripsinya sebagai berikut :

a. Enkripsi
   Rumus $C_i = (P_i + K) \bmod 256$

- F = 70
  Shift = nilai desimal kunci (k) – 97
  = 98 – 97 = 1
  $C_i = (70 + Shift) \bmod 256$
  $= (70+1) \bmod 256$
  $= 71 \bmod 256 = 71$

Nilai desimal 71 pada table ASCII mempunyai karakter “G”

- A = 65
  Shift = nilai desimal kunci (u) – 97
  = 117 – 97 = 20
  $C_i = (65 + Shift) \bmod 256$
  $= (65 + 20) \bmod 256$
  $= 85 \bmod 256 = 85$

Nilai desimal 85 pada table ASCII mempunyai karakter “U”

- R = 82
  Shift = nilai desimal kunci (n) – 97
  = 110 – 97 = 13
  $C_i = (82 + Shift) \bmod 256$
  $= (82 + 13) \bmod 256$
  $= 95 \bmod 256 = 95$

Nilai desimal 95 pada table ASCII mempunyai karakter “_”

- H = 72
  Shift = nilai desimal kunci (c) – 97
  = 99 – 97 = 2
  $C_i = (72 + Shift) \bmod 256$
  $= (72 + 2) \bmod 256$
  $= 74 \bmod 256 = 74$

Nilai desimal 74 pada table ASCII mempunyai karakter “J”

- A = 65

  Shift = nilai desimal kunci (i) – 97
  = 105 – 97 = 8
  C_i = (65 + Shift) mod 256
  = (65 + 8) mod 256
  = 73 mod 256 = 73

Nilai desimal 73 pada table ASCII mempunyai karakter "I"

- N = 78

  Shift = nilai desimal kunci (k) – 97
  = 98 – 97 = 1
  C_i = (78 + Shift) mod 256
  = (78 + 1) mod 256
  = 79 mod 256

Nilai desimal 79 pada table ASCII mempunyai karakter "O"

Jadi, Plaintext "FARHAN" dengan kunci "kunci" mempunyai ciphertext GU_JIO. Untuk melakukan proses dekripsi, bisa menggunkan rumus dekripsi vigenere cipher.

b. Dekripsi

Rumus $P_i = (C_i + K) \bmod 256$

- G = 71

  Shift = nilai desimal kunci (k) – 97
  = 98 – 97 = 1
  C_i = (70 - Shift) mod 256
  = (70 - 1) mod 256
  = 70 mod 256 = 70

Nilai desimal 70 pada table ASCII mempunyai karakter "F"

- U = 85

  Shift = nilai desimal kunci (u) – 97
  = 117 – 97 = 20
  C_i = (65 - Shift) mod 256
  = (65 - 20) mod 256
  = 65 mod 256 = 65

Nilai desimal 65 pada table ASCII mempunyai karakter "A"

- _ = 95

  Shift = nilai desimal kunci (n) – 97
  = 110 – 97 = 13
  C_i = (95 - Shift) mod 256
  = (95 - 13) mod 256
  = 82 mod 256 = 82

Nilai desimal 82 pada table ASCII mempunyai karakter "R"

- J = 74

  Shift = nilai desimal kunci (c) – 97
  = 99 – 97 = 2
  C_i = (74 + Shift) mod 256
  = (74 + 2) mod 256
  = 72 mod 256 = 72

Nilai desimal 72 pada table ASCII mempunyai karakter "H"

- I = 73

  Shift = nilai desimal kunci (i) – 97
  = 105 – 97 = 8
  C_i = (73 - Shift) mod 256
  = (73 - 8) mod 256
  = 65 mod 256 = 65

Nilai desimal 65 pada table ASCII mempunyai karakter "A"

- O= 79

  Shift = nilai desimal kunci (k) – 97
  = 98 – 97 = 1
  C_i = (79 - Shift) mod 256
  = (79 - 1) mod 256
  = 78 mod 256

Nilai desimal 79 pada table ASCII mempunyai karakter "N"

Jadi, Ciphertext "GU_JIO" dengan kunci "kunci" mempunyai plaintext FARHAN, artinya proses dekripsi berhasil.

2.3 Algoritma Rivest Shamir Adleman (RSA)

Algoritma RSA ditemukan oleh 3 orang peneliti dari MIT (*Masschussets Institute of Technology*), yaitu Ron Rivest, Adi Shamir, dan Len Adleman, pada tahun 1976.

2.3.1 Algoritma Pembangkitan Pasangan Kunci

Digunakan algoritma sebagai berikut:

1) Pilih 2 (dua) bilangan prima secara acak yang diberi nama p dan q. Nilai p dan q harus dirahasiakan.
2) Hitung nilai n = p x q besaran n tidak harus dirahasiakan.
3) Hitung m = (p-1).(q-1)
4) Pilih nilai e dengan syarat e > 1, dan GCD(e,m) = 1
5) Pilih nilai d, dengan ketentuan (d.e) mod m = 1

Maka hasil dari algoritma tersebut diperoleh:

1) *Public Key* adalah pasangan (e,n).
2) *Private Key* adalah pasangan (d,n).

2.3.2 Enkripsi Pesan

1) Menggunakan *public key* (e,n).
2) Rumus $c = M^e \bmod n$.

2.3.3 Dekripsi Pesan

1) Menggunakan kunci pribadi (d,n).
2) Pilih *ciphertext* C.
3) Rumus $M = c^d \bmod n$.

2.3.4 Penggunaan Algoritma RSA

Berikut ini adalah contoh penggunaan algoritma RSA

1) p = 17 dan q = 11
2) n = p x q
   n = 17 x 11 = 187
3) m = (p – 1)(q – 1)
   m = (17-1)(11-1) = 160
4) Pilih nilai e dengan syarat e > 1, dan GCD(e,m) = 1
   e = 7

Sebelumnya, sesuai persyaratan, apakah GCD (7,160) = 1 ?

160 mod 7 = 6
7 mod 6 = 1
6 mod 1 = 0

Ternyata benar GCD(7,160) = 1, berarti angka 7 dapat digunakan sebagai nilai e.

5) Pilih nilai d, dengan syarat (d.e) mod m = 1
   d = 23

Sebelumnya sesuai persyaratan, apakah (23 x 7) mod 160 = 1 ?

(23 x 7) mod 160 = 1
= 161 mod 160
= 1

Ternyata benar (23 x 7) mod 160 = 1. Berarti persyaratan terpenuhi & 23 sudah bisa dipastikan dapat mengisi nilai d.

Sehingga didapatkan :

1) Public key = (e, n) = (7, 187)
2) Private key = (d, n) = (23, 187)

2.3.5 Proses Enkripsi

Setelah didapat perhitungan di atas, maka akan dilakukan enkripsi plaintext M = FARHAN. Pertama-tama *plaintext* tersebut diubah menjadi format ASCII (lihat gambar 2.5) sebagai berikut:

Plaintext	F	A	R	H	A	N
ASCII	70	65	82	72	65	78

Rumus $c = M^e \bmod n$, Maka :
M = F(70), $c = 70^7 \bmod 187 = 60$
M = A(65), $c = 65^7 \bmod 187 = 142$
M = R(82), $c = 82^7 \bmod 187 = 91$
M = H(72), $c = 72^7 \bmod 187 = 30$
M = A(65), $c = 65^7 \bmod 187 = 142$
M = N(78), $c = 78^7 \bmod 187 = 56$

Ciphertext = 60 142 91 30 142 56

2.3.6 Proses Dekripsi

Setelah *chipertext* didapat yaitu 60 142 91 30 142 56 , untuk mengubahnya kembali jadi *plaintext* menggunakan dekripsi dengan rumus $M = c^d \bmod n$.

c = 60, $M = 60^{23} \bmod 187 = 70 = F$
c = 142, $M = 142^{23} \bmod 187 = 65 = A$
c = 91, $M = 91^{23} \bmod 187 = 82 = R$
c = 30, $M = 30^{23} \bmod 187 = 72 = H$
c = 142, $M = 142^{23} \bmod 187 = 65 = A$
c = 56, $M = 56^{23} \bmod 187 = 78 = N$

Setelah nilai *plaintext* didapatkan, maka selanjutnya merubahnya menjadi karakter ASCII yaitu FARHAN.

3 METODOLOGI PENELITIAN

3.1 Mengumpulkan Data

Tahapan ini dilakukan untuk mengumpulkan data dengan mencari informasi pembahasan yang sudah pernah dibahas di internet, wawancara narasumber, meminta data yang akan dibutuhkan dalam pembangunan aplikasi ke perusahaan dan membaca buku-buku referensi.

3.2 Analisis Data

Menganalisa algoritma kriptografi Vigenere Cipher dan Rivest Shamir Adleman (RSA) dan teknik yang dipakai.

3.3 Perancangan Sistem

Merancang sistem untuk menentukan spesifikasi dari sistem sesuai hasil analisa yang dilakukan.

3.4 Implementasi

Melakukan tahapan untuk mengembangkan perangkat lunak dengan pengkodean program, pengujian, dan menerapkan berdasarkan hasil analisa kedalam bentuk program dengan bahasa pemograman Java.

3.5 Pengujian Sistem

Melakukan pengujian terhadap program yang telah dirancang serta menyimpulkan hasil pengujian.

4 ANALISA DAN PERANCANGAN SISTEM

4.1 Analisa Masalah dan Penyelesaiannya

Pada saat ini sistem keamanan database umumnya masih sangat kurang. Misalnya pengguna ingin menyimpan data penting ke dalam *database* yang hanya boleh diketahui oleh pihak tertentu. Seandainya data tersebut jatuh kepada pihak yang tidak berhak dan tidak bertanggung jawab, maka pemilik *database* akan mengalami kerugian. Untuk mengantisipasi hal yang tidak diinginkan seperti pencurian data pada *database*, maka dibutuhkan suatu

aplikasi untuk mengamankan *database*. Keamanan *database* merupakan suatu masalah yang amat penting bagi sebuah perusahaan.

Itu sebabnya dibuatlah aplikasi kriptografi yang mampu mengamankan *database* dengan metode enkripsi. Metode enkripsi yang digunakan aplikasi tersebut harus memiliki tingkat keamanan yang tinggi. Aplikasi tersebut nantinya dapat merubah sebuah *database* menjadi *database* yang isinya tidak bisa di baca dan *database* tersebut dapat terjaga kerahasiaannya. Kemudian mengembalikan *database* tersebut menjadi seperti semula tanpa mengalami perubahan sedikitpun.

4.2 Gambaran Umum Aplikasi

Kebutuhan sistem yang akan di bangun pada aplikasi ini adalah sebagai berikut:

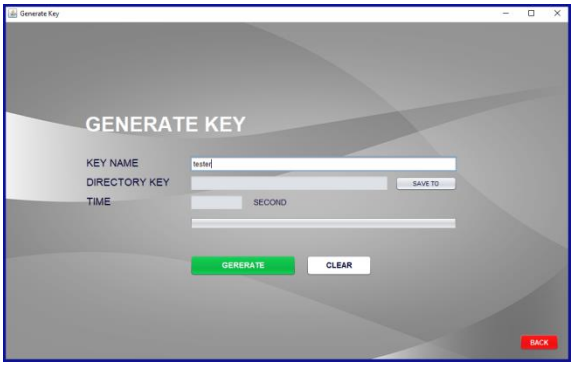
1) Aplikasi mampu mengubah isi data asli dalam sebuah *database* menjadi data acak yang isinya tidak dapat di baca.
2) Aplikasi juga mampu mengembalikan isi *database* asli yang sudah diubah ke data acak kembali seperti semula, tanpa adanya perubahan pada isi *database* tersebut.
3) Aplikasi yang dibuat berbasis *Java*.

5 HASIL DAN PEMBAHASAN

Aplikasi Pengamanan *Database* ini harus diuji untuk dapat mengetahui hasil yang diperoleh setelah sistem dijalankan, fungsi utamadari aplikasi ini adalah melakukan enkripsi dan dekripsi *table database*.

5.1 Tampilan Layar Menu Generate Key

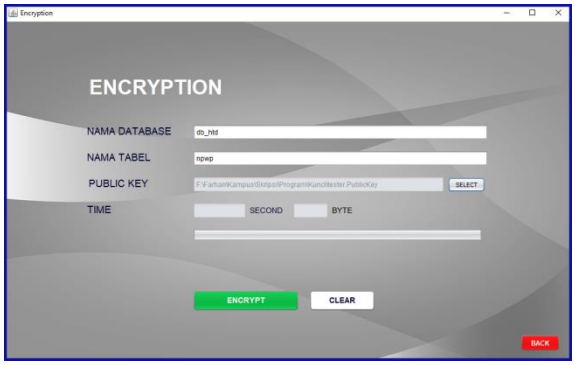
Pada gambar dibawah ini ada tampilan dari layar menu generate key, berfungsi untuk melakukan pembuatan kunci.



Gambar 2 Menu *Generate Key*

5.2 Tampilan Layar Menu Encryption

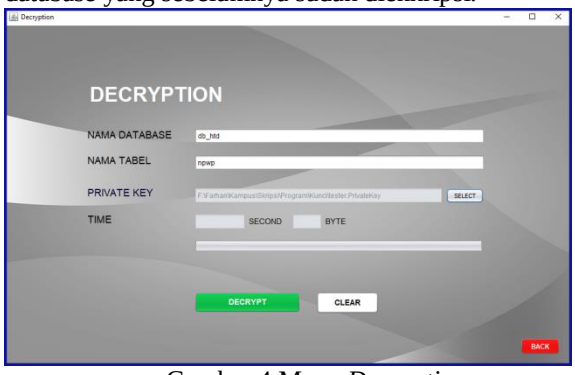
dibawah ini adalah tampilan layar dari form encryption, fungsinya adalah melakukan enkripsi pada database user.



Gambar 3 Menu *Encryption*

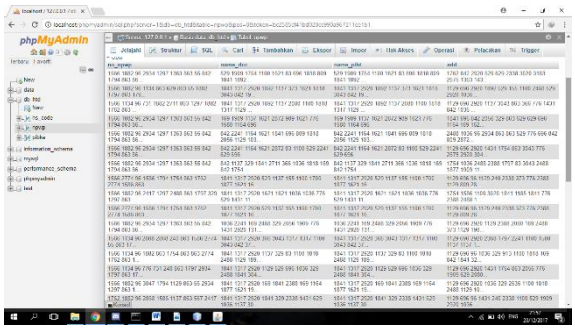
5.3 Tampilan Layar Menu Decryption

Dibawah ini adadlaah tampilan layar form decryption, fungsinya untuk melakukan dekripsi pada database yang sebelumnya sudah dienkripsi.



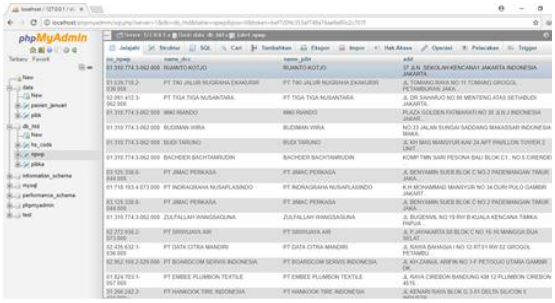
Gambar 4 Menu *Decryption*

5.4 Hasil Enkripsi



Gambar 5 Hasil Enkripsi

5.5 Hasil Dekripsi



Gambar 6 Hasil Dekripsi

6 KESIMPULAN

Dibawah ini adalah beberapa kesimpula yang bisa didapat. Diantaranya adalah :

1) Proses enkripsi dan dekripsi berhasil dilakukan.
2) Keamanan database menjadi lebih kuat karena aplikasi ini menggunakan 2 metode kriptografi.
3) Waktu yang dibutuhkan untuk proses enkripsi dan dekripsi terbilang cukup cepat dan tidak merubah ukuran dari *table database* tersebut.
4) Waktu yang diperlukan untuk proses enkripsi dan dekripsi berbading lurus dengan banyaknya jumlah kolom dan baris pada *table database* yang diproses, Semakin banyak jumlah kolom dan baris maka akan semakin lama proses enkripsi dan dekripsi. Sebaliknya, semakin sedikit jumlah kolom dan baris maka proses enkripsi dan dekripsi akan semakin cepat.

7 DAFTAR PUSTAKA

[1]Ariyus, Doni. 2008. Pengantar Ilmu Kriptografi (Teori, Analisis, dan Implementasi). Yogyakarta : Andi.

[2]Chandra. Wico.2010, *Kriptografi Dan Algoritma RSA*, Bandung : Institut Teknologi Bandung. Makalah II2092 Probabilitas dan Statistik.

[3]I. Fitriasih,T. B . Prayitno,& S.Sidopekso. (2012). *Studi Model Kriptografi* . Jurnal Fisika dan Aplikasi, Vol 13, No 1

[4]Sadikin, Rifki., 2012. *Kriptografi Untuk Keamanan Jaringan*. Yogyakarta : Andi.