# PENERAPAN ALGORITMA KRIPTOGRAFI VIGENERE CIPHER DAN RC4 (RIVEST CODE 4) PADA DATABASE BERBASIS JAVA

**Lutfi Risnanda[1)] , Noni Juliasari[2)]**
Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail : lutfirisnanda2@gmail.com[1)] , noni.juliasari@budiluhur.ac.id[2)]

***Abstrak***

*Seiring dengan perkembangannya teknologi dan telekomunikasi, Salah satu dampak perkembangan negatif dalam teknologi dan telekomunikasi adalah adanya pencurian data, yang merupakan salah satu masalah yang paling ditakuti. Dimana data tersebut sangat rentan jika ada seseorang yang tidak bertanggung jawab merubah atau memanipulasi database tersebut. Dengan demikian maka dibutuhkan suatu sistem pengamanan yang memiliki tingkat kemanan dan kerahasiaan yang cukup tinggi untuk menjamin keamanan dan kerahasian data atau informasi penting tersebut. Pada penelitian ini algoritma yang dipakai yaitu RC4 (Rivest Code 4) dan algoritma Vigenere Cipher. Aplikasi ini hanya dapat mengenkripsi dan dekripsi satu table database saja, selain itu aplikasi ini juga dapat menginput data siswa, data guru, dan data nilai untuk di masukan ke dalam database. Waktu yang digunakan untuk melakukan proses enkripsi dan dekripsi berbanding lurus dengan banyaknya isi database yang diproses (semakin sedikit isi database yang diproses, semakin cepat proses enkripsi dan dekripsi yang dilakukan, semakin banyak isi database yang diproses, semakin lama pula proses enkripsi dan dekripsi yang dilakukan). Dengan menggunakan kedua metode tersebut hasil yang diharapkan memiliki tingkat keamanan yang cukup tinggi sehingga dapat menjaga keamanan dan kerahasiaan data pada database agar tidak mudah dicuri atau di akses oleh pihak yang tidak bertanggung jawab.*

**Kata Kunci** : Kriptografi, RC4, Rivest Code 4, Vigenere Cipher, Database

## 1. PENDAHULUAN

Seiring dengan perkembangan teknologi dan telekomunikasi, Salah satu dampak perkembangan negatif dalam teknologi dan telekomunikasi adalah adanya pencurian data, yang merupakan salah satu masalah yang paling ditakuti. Dengan adanya masalah tersebut maka aspek keamanan dalam penyimpanan data di anggap penting. Untuk itu keamanan penyimpanan data yang digunakan haruslah terjamin keamanannya. Salah satu data-data yang sangat penting yang disimpan adalah *database.* Dimana *database* tersebut sangat rentan jika ada seseorang yang tidak bertanggung jawab merubah atau memanipulasi *database* tersebut.

Dari latar belakang diatas, terdapat beberapa masalah yang dihadapi, Bagaimana cara mengamankan informasi yang disimpan didalam *database* sehingga informasi tersebut terjaga keamanannya? Bagaimana mengimplementasikan Algoritma Kriptografi dalam *database* ?, Pertanyaan- pertanyaan tersebut sering sekali dipertanyakan.

Maka dari itu penulis bertujuan untuk membuat suatu program pengamanan Dengan mengubah informasi yang ada di *database* menjadi kode-kode yang sulit dimengerti agar keamanan data dan kerahasiaan informasi dapat terlindungi dengan baik dan Dengan menerapkan kombinasi metode algoritma RC4 (*Rivest Code* 4) dan algoritma *Vigenere Cipher.* Dengan menggunakan kedua metode tersebut diharapkan memiliki tingkat keamanan yang cukup tinggi sehingga dapat menjaga keamanan dan kerahasiaan data pada *database* agar tidak mudah dicuri atau di akses oleh pihak yang tidak bertanggung jawab.

## 2. METODE PENELITIAN

### 2.1 Alur Penelitian

Adapun beberapa metode yang digunakan penulis untuk menyelesaikan masalah yang ditemui. metode-metode itu adalah sebagai berikut

a. Studi Literatur
   Melalui penelitian kali ini penulis memperoleh data dengan mengumpulkan, mempelajari dan membaca berbagai referensi baik dari buku, jurnal, makalah, dan internet mengenai *database*, algoritma *Rivest Code 4* dan *Vigenere Cipher*, konsep perhitungan yang mendasar, dan beberapa referensi lainnya.
b. Analisa Data
   Metode ini digunakan untuk menganalisa algoritma *Rivest Code 4* dan *Vigenere Cipher.*
c. Perancangan Sistem
   Metode ini digunakan dengan cara merancang system aplikasi yang akan dibangun dan mengimplementasikan langsung algoritma *Rivest Code 4* dan *Vigenere Cipher* ke dalam sebuah aplikasi berbasis *Java Desktop*.
d. Pengujian Sistem
   Metode ini dilakukan dengan menguji dan megecek alur program yang sedang atau telah

dirancang lalu disimpulkan hasil dari pengujian tersebut.

**2.2 Algoritma *Vigenere Cipher***

Kode vigenere ini adalah kode abjad-majemuk yang termasuk satu pemicu perang sipil di Amerika. Para Tentara Konfederasi pada perang sipil Amerika menggunakan kode vigenere untuk mengirim pesan/ perintah untuk para tentaranya. Pada pertengahan abad ke-19 Babbage dan Kasiski berhasil memecahkan kode vigenere.[3] Pada jurnal ini penulis menggunakan tabel ASCII, dimana *key*-nya sebanyak 256 karakter. Sehingga tingkat kerahasiaannya relatif lebih aman dibandingkan dengan karakter vigenere alfabet (26 karakter).
Dibawah ini adalah rumus vigenere cipher:

a. Enkripsi
$$\mathbf{C_i = (P_i + K) \bmod 256}$$
b. Dekripsi
$$\mathbf{P_i = (C_i - K) \bmod 256}$$
Keterangan:
$C_i$ = nilai desimal karakter ciphertext ke-i
$P_i$ = nilai desimal karakter plaintext ke-i
K = nilai desimal karakter kunci ke-I
mod 256 = berdasarkan ASCII

Contoh :

Diketahui plaintext "LU", jika menggunakan nilai Z = 97, kalau di tabel ASCII yaitu huruf "a". dan kuncinya "pu". Berikut ini adalah proses enkripsinya:

a. Enkripsi
Rumus $C_i = (P_i + K) \bmod 256$
- L = 76
  Shift = nilai desimal kunci (p) – 97
  = 112 – 97 = 15
  $C_i$ = (76 + Shift) mod 256
  = (76+15) mod 256
  = 91 mod 256 = 91

  Nilai desimal 91 pada table ASCII mempunyai karakter "["
- U = 86
  Shift = nilai desimal kunci (u) – 97
  = 117 – 97 = 20
  $C_i$ = (86 + Shift) mod 256
  = (86 + 20) mod 256
  = 106 mod 256 = 106

  Nilai desimal 106 pada table ASCII mempunyai karakter "j"

Jadi, Plaintext "LU" dengan kunci "pu" mempunyai ciphertext "[j". Untuk melakukan proses dekripsi, Berikut ini adalah proses dekripsi vigenere cipher.

b. Dekripsi
Rumus $P_i = (C_i - K) \bmod 256$
- [ = 91
  Shift = nilai desimal kunci (p) – 97
  = 112 – 97 = 15
  $C_i$ = (91 - Shift) mod 256
  = (91 - 15) mod 256
  = 76 mod 256 = 76

  Nilai desimal 76 pada table ASCII mempunyai karakter "L"
- j = 106
  Shift = nilai desimal kunci (u) – 97
  = 117 – 97 =20
  $C_i$ = (106 - Shift) mod 256
  = (106 - 20) mod 256
  = 86 mod 256 = 86

  Nilai desimal 86 pada table ASCII mempunyai karakter "U"

Jadi, Ciphertext "[j" dengan kunci "pu" mempunyai plaintext LU, artinya proses dekripsi berhasil.

**2.3 Algoritma *Rivest Code 4*(RC4)**

Kriptografi RC4 adalah enkripsi stream simetrik yang dibuat oleh *RSA Data Security, Inc* atau yang disingkat dengan *RSADSI* [4]. Proses enkrip dan deskrip mempunyai proses yang sama sehingga hanya ada satu kunci untuk menjalankan kedua proses tersebut. RC4 mempunyai sebuah S - Box dan key dalam bentuk arrray 256 byte yaitu : dari S0, sampai S255 yang berisi bilangan 0 - 255, dari K0, sampai K255.

Secara garis besar algoritma RC4 Stream Chiper ini terbagi dua bagian, yaitu : key setup dan *Key Scheduling Algorithm* (KSA) dan *stream generation* atau *Pseudo Randon Generation Algorithm* ( PRGA ) dan proses XOR dengan *steam data* [5].

1. Key Setup / Key Scheduling Algorithm (KSA) Pada bagian ini, terdapat tiga tahapan proses didalam nya yaitu:

   a) Inisialisasi S-box Pada tahapan ini, S-Box akan diisi dengan nilai sesuai indeksnya untuk mendapatkan S-Box awal.
      1) Untuk i=0 hingga i=255 lakukan
      2) Isikan S dengan nilai i
      3) Tambahkan i dengan 1, kembali ke 2

   b) Menyimpan kunci dalam Key Byte Array Pada tahapan ini, kunci (key) yang akan digunakan untuk mengenkripsi atau dekripsi akan dimasukan ke dalam array berukuran 256 secara berulang sampai seluruh array terisi.
      1) Isi j dengan 1
      2) Untuk i=0 hingga i-255 lakukan
      3) Jika j > panjang kunci maka
      4) j diisi dengan nilai 1
      5) akhir jika
      6) isi k ke i dengan nilai ASCII karakter kunci ke j
      7) nilai j dinaikan 1
      8) tambahkan i dengan 1, kembali ke 2.

Maka akan di dapat urutan array key sebagai berikut untuk kunci dengan panjang 8 karakter dengan urutan karakter dalam ASCII " 109 97 98 98 97 104"

c) Permutasi pada S-Box
Sedangkan untuk inisialisasi S - Box yaitu dengan mengisikan nilai 1 sampai dengan 255 mulai S0 sampai S255, isi S-Box secara berurutan, yaitu S0=0, S1=1, sampai dengan S255=255. Cara inisalisasi key yaitu dengan mengisikan array K255 byte dengan kunci yang diulangi sampai seluruh array terisi sepenuhnya mulai K0,K1sampai K255.
1) Isi nilai j dengan 0
2) Untuk i=0 hingga i=255 lakukan
3) Isi nilai j dengan nilai operasi (j+S(i)+K(i)) mod 256
4) Tukar nilai S(i) dengan S(j)
5) Tambahkan i dengan 1, kembali ke 2

Dari algoritma tersebut akan diperoleh nilai SBox yeng telah mengalami proses transposisi sehingga urutannya diacak untuk kunci, contohnya sebagai berikut

2. Stream Generation

Pada tahapan ini akan dihasilkan pseudo random yang menggunakan operasi XOR untuk menghasilkan ciphertext menjadi plaintext ataupun sebaliknya. Berikut adalah algoritmanya:

a. Isi indeks i dan j dengan nilai 0
b. Untuk i=0 hingga I = panjang plaintext
c. Isikan i dengan hasil (i+1) mod 256
d. Isikan j dengan hasil (j+S(i)) mod 256
e. Tukar nilai S(i) dan S(j)
f. Isikan t dengan hasil (S(i)+(S(j) mod 256)) mod 256
g. Isi nilai y dengan nilai S(t)
h. Nilai y dikenakan operasi XOR terhadap plaintext
i. Tambahkan i dengan 1, kembali ke 2

Dengan demikian akan dihasilkan misalkan ciphertext dengan hasil XOR antar stream key dari S-Box dan plaintext secara berurutan.

**2.4 Rancangan Basis Data**

**1) Tabel Data Siswa**

Nama Tabel : datasiswa
Isi : informasi siswa
*Primary Key* : nis

*Tabel 3.1 : Spesifikasi Tabel datasiswa*

| Field | Type | Length | Keterangan |
|---|---|---|---|
| Nis | *Varchar* | 255 | Nomor induk siswa |
| Nama | *Varchar* | 255 | Nama siswa/i |
| Kelas | *Varchar* | 255 | Kelas siswa/i |
| Jeniskelamin | *Varchar* | 255 | Jenis kelamin siswa/i |
| Tempatlahir | *Varchar* | 255 | Tempat lahir siswa/i |
| Tanggallahir | *Varchar* | 255 | Tanggal lahir siswa/i |
| Alamat | *Varchar* | 255 | Alamat siswa/i |
| Agama | *Varchar* | 255 | Agama siswa/i |
| Jurusan | *Varchar* | 255 | Jurusan siswa/i |

**2.5 Rancangan Layar**

1) Rancangan Layar *Form Login*

Pada rancangan menu *form login* merupakan halaman awal ketika pengguna membuka aplikasi ini, sebelum pengguna mengakses ke halaman utama, pengguna diharuskan memasukan *username dan password* terlebih dahulu. Berikut adalah rancangan menu *form login* :

*Gambar 1: Rancangan Layar Form Login*

2) Rancangan Layar *Form* Data Siswa

Pada rancangan menu *Form* Data Siswa, berfungsi untuk melakukan *input data,* simpan, *edit,* dan hapus data siswa.

*Gambar 2: Rancangan Layar Form Data Siswa*

3) Rancangan Layar *Form* Enkripsi

Pada rancangan menu *Form* Enkripsi *Database,* berfungsi untuk melakukan enkripsi *database, per-table database*. Dengan cara pilih *table database* yang ingin di enkripsi, lalu ketikan kunci

enkripsinya,dan klik *button* enkrip Berikut adalah rancangan layar *Form* Enkripsi:

*Gambar 3: Rancangan Layar Form Enkripsi*

4) Rancangan Layar *Form* Dekripsi

Pada rancangan menu *Form* Dekripsi *Database,* berfungsi untuk melakukan dekripsi *database, per-table database*. Dengan cara pilih *table database* yang ingin di dekripsi, lalu ketikan kunci yang sama saat *user* enkripsi sebelumya,dan klik *button* enkrip Berikut adalah rancangan layar *Form* Dekripsi.

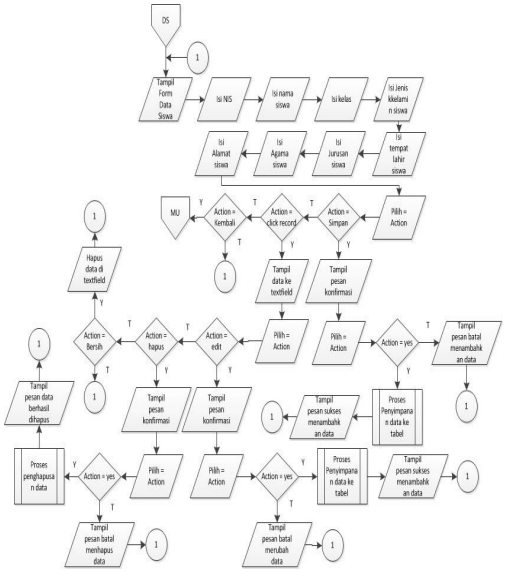
*Gambar 4: Rancangan Layar Form Dekripsi*

**2.6 *Flowchart***

Flowchart adalah sebuah jenis diagram yang menampilkan langkah-langkah atau alur dari proses aplikasi yang digunakan atau memperjelas alur proses aplikasi. Dibawah ini akan digambarkan beberapa flowchart untuk masing-masing proses.

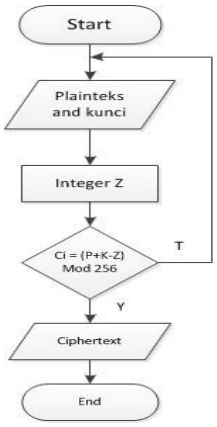
1) *Rancangan Layar Form Data Siswa*

Flowchart form Data Siswa merupakan gambaran alur proses dari form Data siswa. Pada proses ini user dapat meng*input* data siswa. Di form data siswa tersedia beberapa tombol antara lain tombol untuk simpan data siswa, tombol untuk hapus data siswa, tombol untuk edit data siswa, tombol bersih untuk bersihkan *textfield* data siswa, dan tombol back untuk kembali ke menu utama



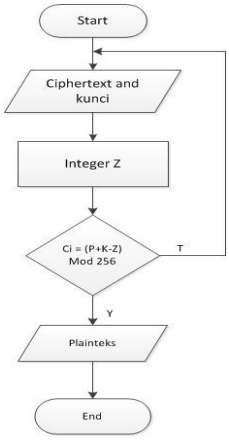
*Gambar 5: Flowchart Proses Enkripsi Vigenere Cipher*

2) *Flowchart Proses Enkripsi Vigenere Cipher*



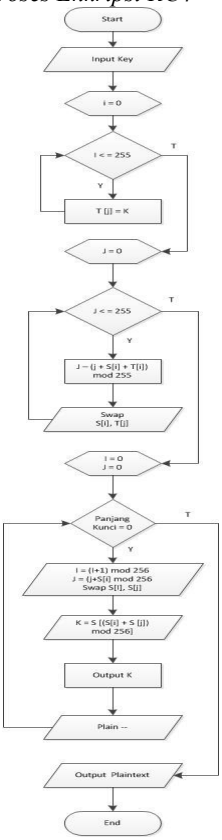

*Gambar 6: Flowchart Proses Enkripsi Vigenere Cipher*

3) *Flowchart Proses Dekripsi Vigenere Cipher*



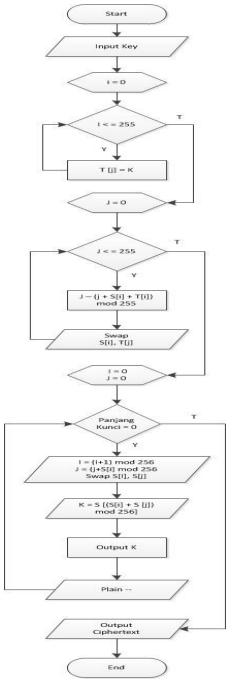
*Gambar 7: Flowchart Proses Dekripsi Vigenere Cipher*

4) *Flowchart Proses Enkripsi RC4*




*Gambar 8: Flowchart Proses Enkripsi RC*

5) *Flowchart Proses Dekripsi RC4*

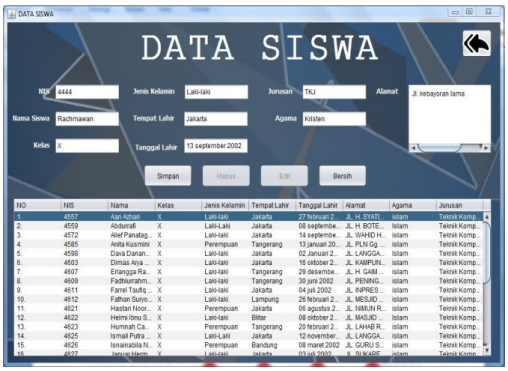



*Gambar 9: Flowchart Proses Dekripsi RC4*

## 3. HASIL DAN PEMBAHASAN

### 3.1 Form Data Siswa

Tampilan *form* data siswa ini muncul pada saat user memilih tombol Data Siswa pada form menu utama.



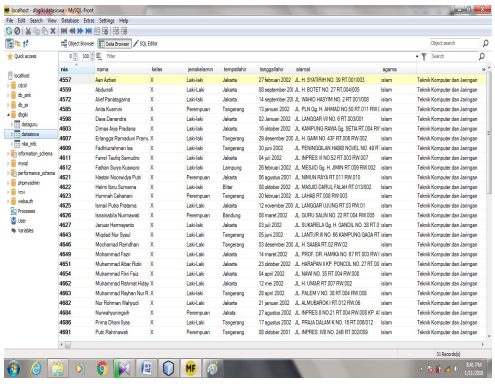

*Gambar 10: Tampilan Form Data Siswa*

### 3.2 Proses Enkripsi dan Dekripsi

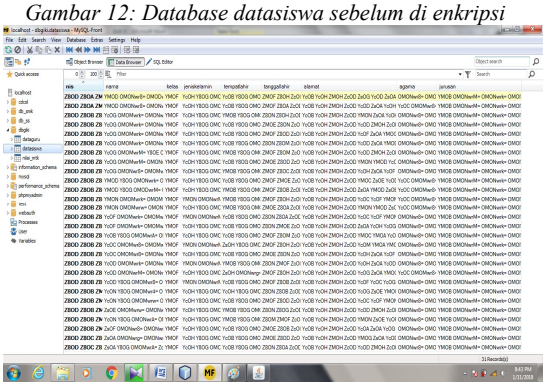
Pengujian enkripsi ini dimulai dari user memilih nama *table database* yang ingin di enkripsi kan, lalu ketikan kunci enkripsi, lalu tekan klik *button* enkrip. Berikut akan diperlihatkan pada gambar di bawah ini:


*Gambar 11: Uji Coba Enkripsi Databse*

Dalam pengujian ini *user* ingin mengenkripsi table "datasiswa" dan key/kuncinya "Tugas", untuk tampilan *database* sebelum dan dan sesudah dienkripsi adalah sebagai berikut:



*Gambar 12: Database datasiswa sebelum di enkripsi*

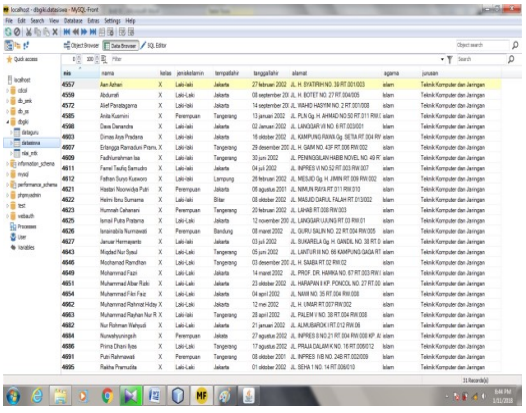


*Gambar 13: Database datasiswa setelah di enkripsi*

Lalu dalam pengujian dekripsi ini penggunaan *form* dekrip ini sama seperti *form* enkripsi, pada pengujian ini *user* ingin mendekripsi table "datasiswa" yang sebelumnya telah terenkripsi dan memasukan key/kuncinya yang sama saat user mengenkripsi sebelumnya yaitu "Tugas", untuk lebih jelasnya akan diperlihatkan pada gambar di bawah ini.


*Gambar 14: Uji Coba Dekripsi Database*

Untuk tampilan *database* dan hasil setelah di dekripsi adalah sebagai berikut:



*Gambar 15: Database datasiswa setelah di dekripsi*

### 3.3 Tabel Pengujian

Dalam pengujian ini penulis akan membahas perbandingan antara proses enkripsi dan dekripsi.

*Tabel 4.1: Tabel hasil Pengujian*

| Input Tabel | Waktu | | Ukuran Tabel Database | | |
|---|---|---|---|---|---|
| | Enkripsi | Dekripsi | Asli | Enkripsi | Dekripsi |
| **datasiswa** | 2.995 Detik | 5.156 Detik | 16Kb | 16Kb | 16Kb |
| **dataguru** | 2.283 Detik | 5.274 Detik | 16Kb | 16Kb | 16Kb |

## 4. KESIMPULAN

### 4.1 Kesimpulan

Berdasarkan penelitian diatas, maka dapat diambil suatu kesimpulan antara lain:

a. Dengan adanya aplikasi kriptografi ini, proses penyimpanan menjadi lebih aman.
b. Satu kunci bisa digunakan berkali-kali dengan jenis ataupun *database* yang berbeda.
c. Proses dekripsi dengan kunci yang sesuaiakan mengembalikan *database* menjadi *database* semula tanpa mengalami perubahan sedikit pun.
d. Waktu yang digunakan untuk melakukan proses enkripsi dan dekripsi berbanding lurus dengan banyaknya isi *database* yang diproses (semakin sedikit isi *database* yang diproses, semakin cepat proses enkripsi dan dekripsi dilakukan, semakin banyak isi database yang diproses, semakin lama proses pula enkripsi dan dekripsi yang dilakukan).
e. Aplikasi ini tidak dapat mengenkripsi dua atau lebih dari satu table *database*.
f. Pengamanan data dilakukan hanya pada *database* yang telah ditentukan.

### 4.2 Saran

Berikut saran yang mungkin diperlukan untuk mengembangkan aplikasi ini agar dapat berjalan lebih baik lagi kedepannyaantara lain:

a. Aplikasi ini diharapkan dapat ditingkatkan kinerjanya sehingga tidak hanya dapat mengenkripsi satu *table database* saja, tetapi bisa mengenkripsi seluruh *table* didalam *database*.
b. Waktu proses enkripsi dan dekripsi *database* yang rata-rata berisi banyak data diharapkan dapat berjalan lebih cepat pada *hardware* yang lebih baik.
c. Aplikasi ini diharapkan dapat meyimpan kunci enkripsi, atau tidak perlu menginput kunci lagi ketika ingin mendekripsi *database*.

## 5. DAFTAR PUSTAKA

[1] B. Candra, J. Wahyudi, and Hermawansyah, "PENGEMBANGAN SISTEM KEAMANAN UNTUK TOKO ONLINE BERBASIS KRIPTOGRAFI AES MENGGUNAKAN

BAHASA PEMROGRAMAN PHP DAN MYSQL," *J. Media Infotama*, vol. 11, no. 1, pp. 31–39, 2014.

[2] H. Hasrul and L. H. Siregar, "PENERAPAN

TEKNIK KRIPTOGRAFI PADA DATABASE MENGGUNAKAN," vol. 2, no. 2, pp. 41–52, 2016.

[3] P. Yulianingsih, Hamdani, and S. Maharani, "Aplikasi Chatting Rahasia Menggunakan Algoritma Vigenere Cipher," *Inform. Mulawarman*, vol. 9, no. 1, pp. 1–4, 2014.

[4] N. B. Nugroho, Z. Azmi, and S. N. Arif, "Aplikasi Keamanan Email Menggunakan Algoritma Rc4," *J. SAINTIKOM*, vol. 15, no. 3, pp. 81–88, 2016.

[5] Febrian Wahyu. C. et al., "PENERAPAN ALGORITMA GABUNGAN RC4 DAN BASE64 PADA SISTEM KEAMANAN E-COMMERCE," vol. 2012, no. Snati, pp. 15–16, 2012.