Perangkat Lunak Pembelajaran Metode Kriptografi *Word Auto Key Encryption*

Agustian Noor

Jurusan Teknik Informatika, Politeknik Negeri Tanah Laut
Jl. A Yani Km 6 Pelaihari Tanah Laut Kalimantan Selatan
Telepon / Fax (0512) 21537
E-mail: agustiannoor@ymail.com.com

Abstrak *– Metode kriptografi dapat digunakan untuk mengamankan data yang bersifat rahasia agar data tersebut tidak diketahui oleh orang lain yang tidak berkepentingan. Metode WAKE (word auto key encryption) merupakan salah satu metode yang telah digunakan secara komersial. Metode ini menggunakan kunci 128 bit, plaintext 32 bit dan sebuah tabel 256 x 32 bit. Dalam algoritmanya, metoda ini menggunakan operasi XOR, AND, OR dan Shift Right. Inti dari metode WAKE terletak pada proses pembentukan tabel S-Box dan proses pembentukan kunci. Tabel S-Box dari metode WAKE bersifat fleksibel dan berbeda-beda untuk setiap putaran. Perancangan perangkat lunak menggunakan metode perancangan RAD (Rapid Application Development), adapun langkah-langkah yang dilakukan yaitu: Bussiness Modeling, Data modelling, Process Modeling, Generation Application. Metode WAKE dapat dibagi menjadi beberapa proses yaitu proses pembentukan tabel dan kunci, enkripsi dan dekripsi. Proses penyelesaian metode ini cukup rumit dan sulit untuk dikerjakan secara manual berhubung karena algoritmanya yang cukup panjang dan kompleks*.

Kata Kunci: Data, Metode Kriptografi, Wake.

1. PENDAHULUAN

1.1 Latar Belakang

Dalam ilmu kriptografi, selain metode WAKE, masih banyak metode yang dapat digunakan untuk mengamankan data.Setiap metode memiliki kelebihan dan kekurangannya masing-masing.Namun, yang menjadi permasalahan dalam memilih metode kriptografi yang cocok adalah bagaimana mengetahui dan memahami cara kerja dari metode kriptografi tersebut. Oleh karena itu, diperlukan suatu perangkat lunak untuk mempelajari metode kriptografi tersebut.Penulis memilih metode WAKE karena metode ini cukup cepat dalam implementasinya pada perangkat lunak.

Berdasarkan uraian di atas maka penulis ingin membuat jurnal dengan judul "Perancangan Perangkat Lunak Pembelajaran Metode Kriptografi WAKE (*Word Auto Key Encryption*)".

1.2 Permasalahan Penelitian

Yang menjadi permasalahan dalam menyusun jurnal ini adalah bagaimana menampilkan langkah-langkah proses pembentukan tabel, kunci, enkripsi dan dekripsi dari metode kriptografi WAKE.

2. TINJAUAN PUSTAKA

2.1 Penelitian terkait

2.1.1 Implementasi Kriptografi Idea pada priority Dealer untuk Layanan dan Pemesanan Penjualan Handphone Berbasis WEB

Rancangan Aplikasi oleh Kholidya Yuli Wardani, M.Zen Syaiful Hadi ini hanya berpatokan pada keamanan pemesanan dan akun pengguna layanan web saja tidak menjelaskan struktur perhitungan enkripsi data.

2.1.2 Pembelajaran Enkripsi Metode Word Auto Key Encryption

Aplikasi rancangan Eddy dan Muhammad Reza Pahlevi ini hanya terbatas pada pembelajaran saja. Perangkat lunak ini menunjukkan setiap langkah dan tahapan proses proses (proses pembentukan table S-Box, proses pembentukan kunci, proses enkripsi dan proses dekripsi) yang terdapat di dalam kriptografi metode WAKE, sehingga dapat membantu pemahaman atau pembelajaran prosedur kerja atau algoritma dari metode kriptografi tersebut.

2.2 Kriptografi

Menurut Stalling, ada beberapa tuntutan yang terkait dengan isu keamanan data yaitu :

1. *Confidentiality*
   Menjamin bahwa data-data tersebut hanya bisa diakses oleh pihak-pihak tertentu saja.
2. *Authentication*
   Baik pada saat mengirim atau menerima informasi, kedua belah pihak perlu mengetahui bahwa pengirim dari pesan tersebut adalah orang yang sebenarnya seperti yang diklaim.
3. *Integrity*
   Tuntutan ini berhubungan dengan jaminan setiap pesan yang dikirim pasti sampai pada penerimanya tanpa ada bagian dari pesan tersebut yang diganti, diduplikasi, dirusak, diubah urutannya, dan ditambahkan.
4. *Nonrepudiation*
   Nonrepudiation mencegah pengirim maupun penerima mengingkari bahwa mereka telah mengirimkan atau menerima suatu pesan/informasi.Jika sebuah pesan dikirim, penerima dapat membuktikan bahwa pesan

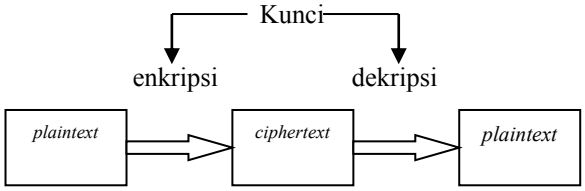
tersebut memang dikirim oleh pengirim yang tertera.Sebaliknya, jika sebuah pesan diterima, pengirim dapat membuktikan bahwa pesannya telah diterima oleh pihak yang ditujunya.

5. *Access Control*
   Membatasi sumber-sumber data hanya kepada orang-orang tertentu.
6. *Availability*
   Jika diperlukan setiap saat semua informasi pada sistem komputer harus tersedia bagi semua pihak yang berhak atas informasi tersebut.

Dari keenam aspek keamanan data tersebut, empat diantaranya dapat diatasi dengan menggunakan kriptografi yaitu *confidentiality, integrity, authentication,* dan *nonrepudiation.* Kriptografi dapat didefinisikan sebagai ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek-aspek pada keamanan informasi misalnya kerahasiaan, integritas data, otentikasi pengirim / penerima data, dan otentikasi data.*Cryptanalysis* adalah studi tentang bagaimana mengalahkan (memecahkan) mekanisme kriptografi, dan *cryptology* yang berasal dari kata kryptos dan logos (bahasa Yunani) yang artinya kata tersembunyi, adalah penggabungan disiplin *cryptography* dan *cryptanalysis*. (Jusuf, 2002)

2.3 Sistem Kriptografi Simetris

Enkripsi simetris sering juga disebut sebagai enkripsi konvensional atau enkripsi kunci-tunggal (*single key*). Pada model enkripsi simetris ini digunakan algoritma yang sama untuk proses enkripsi/dekripsi dengan memakai satu kunci yang sama.



Gambar 1. Model sederhana Sistem Kriptografi Simetris

Keamanan dari enkripsi simetris bergantung pada beberapa faktor, yaitu :

1. Algoritma enkripsi harus cukup kuat sehingga tidaklah praktis untuk mendekripsi suatu pesan hanya dengan memiliki *cyphertext* saja.
2. Keamanan dari enkripsi simetris adalah bergantung pada kerahasiaan kunci, bukan kerahasiaan dari algoritma enkripsi itu sendiri. Semakin panjang kunci yang dipakai maka semakin sulit untuk menebak kunci dengan menggunakan metode *brute force attacks* (mencoba semua kemungkinan kunci).

Algoritma enkripsi simetris yang popular dewasa ini adalah DES (*Data Encryption Standard*) dengan panjang kunci 56-bit, IDEA (128-bit), Twofish (sampai dengan 256-bit), Rijndael (sampai dengan 256-bit) dan lain-lain (Wardani, 2012).

2.4 Sistem Kriptografi Asimetris

Sistem kriptografi asimetris biasanya lebih dikenal dengan kriptografi kunci-publik (*public - keycryptography)*. Ide kriptografi asimetris ini pertama kali dimunculkan oleh Whitfield Diffie dan Martin Hellman pada tahun 1976.Diffie dan Hellman mempostulatkan sistem ini tanpa menunjukkan algoritmanya. Walaupun demikian mereka menjabarkan syarat-syarat yang harus dipenuhi oleh suatu algoritma *public-key* yaitu :

1. Mudah secara komputasi bagi suatu pihak B untuk mengkonstruksi sepasang kunci asimetris (kunci public KU, kunci pribadi KR).
2. Mudah secara komputasi bagi pengirim A, dengan memiliki kunci public B dan pesan yang ingin dienkripsi, M, untuk menghasilkan *ciphertext* (C) :

$$C = E_{KUb}(M) \quad (1)$$

3. Mudah secara komputasi bagi penerima B untuk mendekripsi *ciphertext* yang dihasilkan dengan menggunakan kunci pribadinya untuk mengembalikan pesan aslinya.

$$M = D_{KRb}(C) = D_{KRb}[E_{KUb}(M)] \quad (2)$$

4. Tidak bisa secara komputasi bagi pihak ketiga untuk memperoleh kunci pribadi KRb hanya dengan mengetahui kunci public KUb.
5. Tidak bisa secara komputasi bagi pihak ketiga untuk mengembalikan data asli M hanya dengan mengetahui kunci public KUb dan *ciphertext* C.Walaupun bukanlah suatu keharusan bagi semua aplikasi *public-key*, namun persyaratan keenam bisa ditambahkan :
6. Fungsi enkripsi dan dekripsi bisa diterapkan dengan urutan yang dibalik :

$$M = E_{KUb}[D_{KRb}(M)] \quad (3)$$

Kegunaan dari persyaratan keenam adalah untuk penerapan tanda tangan digital (*digital signature*) yang digunakan memecahkan isu otentikasi (*authentication*) dalam masalah keamanan data.

Menurut Stalling, proses enkripsi *public-key* sederhana melibatkan empat tahap berikut :

1. Setiap *user* di dalam jaringan membuat sepasang kunci untuk digunakan sebagai kunci enkripsi dan dekripsi dari pesan yang akan diterima.
2. *User* mempublikasikan kunci enkripsinya dengan menempatkan kunci publiknya ke tempat umum. Pasangan kunci yang lain tetap dijaga kerahasiaannya.
3. Jika *user*A ingin mengirimkan sebuah pesan ke *user* B, ia akan mengenkripsi pesan tersebut dengan menggunakan kunci publik *user* B.
4. Pada saat *user* B ingin mengirimkan sebuah pesan ke *user* B, ia akan menggunakan kunci pribadinya sendiri. Tidak ada pihak lain yang bisa mendekripsi pesan itu karena hanya B sendiri yang mengetahui kunci pribadi B.

Gambar 2. Model Sederhana Sistem Kriptografi Asimetris

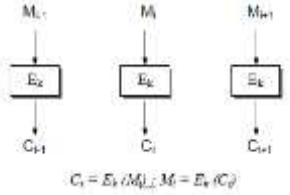
Sampai saat ini hanya ada beberapa sistem *cryptography* asimetris yang dipublikasikan.Yang paling berhasil sejauh ini adalah algoritma RSA yang memenuhi keenam persyaratan *public-key* di atas. Di samping itu, algoritma enkripsi *public-key* yang lain adalah LUC, DSS, Diffie-Hellman dan lain-lain.

Kunci publik adalah kunci yang tidak disembunyikan dan boleh diketahui oleh orang lain. Kunci publik digunakan dalam proses enkripsi.

Kunci *private* adalah kunci rahasia yang tidak boleh diketahui oleh orang lain. Kunci *private* digunakan dalam proses dekripsi. (Wardani, 2012)

2.5 *Block Cipher* dan Mode Operasi

Mode operasi ECB membagi-bagi *plaintext* menjadi blok-blok yang panjangnya n-bit dan masing-masing blok dienkripsi secara terpisah dengan menggunakan kunci yang sama. Dengan demikian, untuk blok n-bit *plaintext* yang sama dalam satu pesan akan menghasilkan n-bit *ciphertext* yang sama pula.



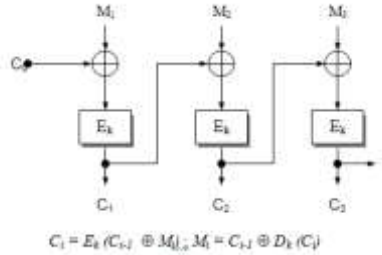
Gambar 3. Mode Electronic Code Book (ECB)

Dari gambar di atas dapat dilihat *plaintext* dipecah menjadi urutan yang terdiri dari blok-blok n-bit, M_{i-1}, M_i, M_{i+1} dengan hasil enkripsinya berupa blok-blok *ciphertext* n-bit, C_{i-1}, C_i, C_{i+1}. (Wardani, 2012)

2.6 Mode Operasi ECB (*Electronic Code Book*)

Salah satu alternatif untuk mencegah munculnya blok-blok *ciphertext* yang sama dari blok *plaintext* yang sama pada satu pesan adalah dengan menggunakan mode CBC. Pada skema ini setiap blok n-bit plaintext di-XOR-kan dengan blok n-bit *ciphertext* sebelumnya. Kecuali blok *plaintext* pertama di-XOR-kan dengan suatu konstanta awal atau initialization vector (IV), sebesar n-bit. Hasil dari proses XOR tersebut yang kemudian dienkripsi.

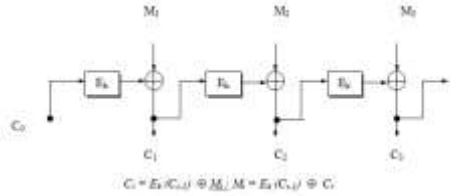
Untuk proses dekripsi, hasil dekripsi blok *ciphertext* di-XOR-kan dengan blok *ciphertext* sebelumnya untuk menghasilkan blok *plaintext*. Untuk blok pertama, hasil dekripsi blok *ciphertext* pertama di-XOR-kan dengan IV untuk menghasilkan blok *plaintext* pertama. Walaupun nilai IV tidak perlu dirahasiakan akan tetapi integritas dari nilai IV harus dilindungi. (Wardani, 2012)



Gambar 4. Mode Cipher Block Chaining (CBC)

2.7 Mode Operasi CFB (Cipher Feedback)

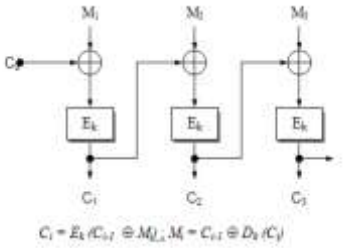
Dalam mode *Cipher Feedback* (CFB) blok*ciphertext* sebelumnya di-enkrip dan outputnya digabungkan dengan blok *plaintext* dengan menggunakan XOR untuk menghasilkan blok *ciphertext* sekarang. Kita dapat mendefinisikan mode CFB sedemikian hingga mode tersebut menggunakan *feedback* yang lebih kecil dari 1 blok penuh data. Sebuah vektor inisialisasi C_0 digunakan sebagai sebuah "seed" untuk prosesnya, seperti terlihat pada Gambar 2.5 di bawah ini. (Wardani, 2012)



Gambar 5 Mode Cipher Feedback

2.8 Mode Output Feedback

Mode *Output Feedback* (OFB mode) mirip dengan mode CFB kecuali bahwa jumlah operasi XOR dengan setiap blok*plaintext* dihasilkan secara independen dari baik *plaintext* maupun *ciphertext*. Sebuah vektor Inisialisasi C_0 digunakan sebagai suatu "seed" untuk sebarisan blok data s_i, dan setiap blok data s_i diperoleh dari proses enkripsi terhadap blok data s_{i-1} sebelumnya. Proses enkripsi blok*plaintext* diperoleh dengan melakukan operasi XOR antara blok *plaintext* dengan blok data yang relevan. [(Wardani, 2012)



Gambar 6 Mode Output Feedback

3. METODE PENELITIAN

3.1 Proses

Proses kriptografi metode WAKE terdiri atas 4 (empat) proses, yaitu :

1. Proses Pembentukan Tabel *S-Box*.
2. Proses Pembentukan Kunci.

3. Proses Enkripsi.
4. Proses Dekripsi.

Inti dari metode WAKE terletak pada proses pembentukan tabel *S-Box* dan proses pembentukan kunci. Proses enkripsi dan dekripsi hanya berupa operasi XOR dari *plaintext* dan kunci untuk menghasilkan *ciphertext* dan operasi XOR dari *ciphertext* dan kunci untuk menghasilkan *plaintext*.

3.2 Identifikasi Masalah

Proses pembentukan tabel *S-Box* terdiri atas 8 (delapan) proses utama. Dalam prosesnya, pembentukan tabel *S-Box* memerlukan *inputkunci* dengan panjang 128 bit biner atau 16 karakter *ascii*. Untuk lebih jelas, proses ini dapat dilihat pada contoh berikut :

Misalkan *inputkey* = 'WAKE, EDDY SALIM', maka proses pembentukan tabel *S-Box* dalam heksadesimal adalah sebagai berikut :

1. Inisialisasi nilai TT[0] ... TT[7].
   TT[0] = 726A8F3B (dalam heksadesimal)
   TT[1] = E69A3B5C
   TT[2] = D3C71FE5
   TT[3] = AB3C73D2
   TT[4] = 4D3A8EB3
   TT[5] = 0396D6E8
   TT[6] = 3D4C2F7A
   TT[7] = 9EE27CF3
2. Pecah kunci menjadi 4 kelompok dan masukkan pada T[0] ... T[3].
   Kunci : 'WAKE, EDDY SALIM'
   Kode ascii dari 'W' = 87 = 57
   Kode ascii dari 'A' = 65 = 41
   Kode ascii dari 'K' = 75 = 4B
   Kode ascii dari 'E' = 69 = 45
   Kode ascii dari ',' = 44 = 2C
   Kode ascii dari ' ' = 32 = 20
   Kode ascii dari 'E' = 69 = 45
   Kode ascii dari 'D' = 68 = 44
   Kode ascii dari 'D' = 68 = 44
   Kode ascii dari 'Y' = 89 = 59
   Kode ascii dari ' ' = 32 = 20
   Kode ascii dari 'S' = 83 = 53
   Kode ascii dari 'A' = 65 = 41
   Kode ascii dari 'L' = 76 = 4C
   Kode ascii dari 'I' = 73 = 49
   Kode ascii dari 'M' = 77 = 4D
   Kunci (dalam heksa) = 57414B452C20454444592053414C494D
   T[0] = K[0] = 57414B45
   T[1] = K[1] = 2C204544
   T[2] = K[2] = 44592053
   T[3] = K[3] = 414C494D
3. Untuk n = 4 sampai 255, lakukan prosedur berikut:

   ```
   X = T[n-4] + T[n-1]

   T[n] = X>>3 XOR TT(X AND 7)
   ```

   -> X = T[0] + T[3] = 57414B45 + 414C494D = 988D9492
   -> X >> 3 (Shift Right 3 bit) = 988D9492 >> 3 = 1311B292
   X AND 7 = 988D9492 AND 7(10) = 2
   T[4] = X >> 3 XOR TT[X AND 7] = 1311B292 XOR TT[2] = C0D6AD77

   n = 5
   -> X = T[1] + T[4] = 2C204544 + C0D6AD77 = ECF6F2BB
   -> X >> 3 (Shift Right 3 bit) = ECF6F2BB >> 3 = 1D9EDE57
   X AND 7 = ECF6F2BB AND 7(10) = 3
   T[5] = X >> 3 XOR TT[X AND 7] = 1D9EDE57 XOR TT[3] = B6A2AD85

   n = 6
   -> X = T[2] + T[5] = 44592053 + B6A2AD85 = FAFBCDD8
   -> X >> 3 (Shift Right 3 bit) = FAFBCDD8 >> 3 = 1F5F79BB
   X AND 7 = FAFBCDD8 AND 7(10) = 0
   T[6] = X >> 3 XOR TT[X AND 7] = 1F5F79BB XOR TT[0] = 6D35F680
   (dan seterusnya hingga n = 255).
4. Untuk n = 0 sampai 22, lakukan prosedur berikut :

   ```
   T[n] = T[n] + T[n + 89]
   ```

   n = 0
   T[0] = T[0] + T[89] = 57414B45 + 15F12D0E = 6D327853
   n = 1
   T[1] = T[1] + T[90] = 2C204544 + 72BFCF87 = 9EE014CB
   n = 2
   T[2] = T[2] + T[91] = 44592053 + E3163C25 = 276F5C78
   n = 3
   T[3] = T[3] + T[92] = 414C494D + A5D89206 = E724DB53
   (dan seterusnya hingga n = 22).
5. Set nilai untuk beberapa variabel di bawah ini.
   X = 8A1B6650
   Z = T[59] OR 01000001 = EC8DC527 OR 01000001 = ED8DC527
   Z = Z AND FF7FFFFF = ED8DC527 AND FF7FFFFF = ED0DC527
   X = X AND FF7FFFFF = 8A1B6650 AND FF7FFFFF = 77292B77
6. Untuk n = 0 sampai 255, lakukan prosedur berikut:

   ```
   X = (X AND FF7FFFFF) + Z

   T[n] = T[n] AND 00FFFFFF XOR X
   ```

   n = 0

X = (77292B77 AND FF7FFFFF) + ED0DC527 = 6436F09E
T[0] = [6D327853] AND 00FFFFFF XOR 6436F09E = 640488CD
n = 1
X = (6436F09E AND FF7FFFFF) + ED0DC527 = 5144B5C5
T[1] = [9EE014CB] AND 00FFFFFF XOR 5144B5C5 = 51A4A10E
n = 2
X = (5144B5C5 AND FF7FFFFF) + ED0DC527 = 3E527AEC
T[2] = [276F5C78] AND 00FFFFFF XOR 3E527AEC = 3E3D2694
(dan seterusnya hingga n = 255).

7. Set nilai untuk beberapa variabel berikut.
   T[256] = T[0] = 640488CD
   X = X AND 255(10) = 776E5277 AND 255(10) = 00000077
8. Untuk n = 0 sampai 255, lakukan prosedur berikut.

```
Temp = (T[n XOR X] XOR X) AND 255

T[n] = T[Temp]

T[X] = T[n+1]
```

n = 0
Temp = T[119] XOR X AND 255 = 8F8F3242 XOR 00000077 AND 255 = 00000035
T[0] = T[53] = 75B47C03
T[119] = T[1] = 51A4A10E
n = 1
Temp = T[118] XOR X AND 255 = A2B3307A XOR 00000077 AND 255 = 0000000D
T[1] = T[13] = 6D7399BB
T[119] = T[2] = 3E3D2694
n = 2
Temp = T[117] XOR X AND 255 = B5758EDE XOR 00000077 AND 255 = 000000A9
T[2] = T[169] = D9F40A36
T[119] = T[3] = 2B449B40
n = 3
Temp = T[116] XOR X AND 255 = C8A36D5D XOR 00000077 AND 255 = 0000002A
T[3] = T[42] = 4640C5A0
T[119] = T[4] = 187463F2
(dan seterusnya hingga n = 255).

3.3 Proses Enskripsi

Proses enkripsi dari metode WAKE untuk menghasilkan *ciphertext* adalah berupa hasil operasi XOR dari *plaintext* dan 32 bit kunci yang dihasilkan dari proses pembentukan kunci.

Ciphertext(C) = *Plaintext*(P) XOR *Key*(K)

3.4 Proses Deskripsi

Proses dekripsi dari metode WAKE untuk menghasilkan *plaintext* adalah berupa hasil operasi XOR dari *ciphertext* dan 32 bit kunci yang dihasilkan dari proses pembentukan kunci.

Plaintext (P) = *Ciphertext*(C) XOR *Key*(K)

3.5 Teknik Perancangan

Perangkat lunak pembelajaran kriptografi metode WAKE dirancang dengan menggunakan bahasa pemrograman *Microsoft Visual Basic 6.0* dengan beberapa komponen standar seperti *Microsoft Flex Grid, Text Box, Picture Box, Label, Shape*, dan sebagainya. Penggunakan aplikasi *Microsoft Visio* untuk menggambarkan diagram proses pembentukan kunci, proses enkripsi dan proses dekripsi. Desain perangkat lunak dirancang dengan menggunakan aplikasi *Adobe Photoshop C.S.*

Proses pembentukan tabel *S-Box* memerlukan *input* kunci 16 karakter *ascii* atau 128 bit biner, sehingga tabel *S-Box* pada metode WAKE adalah berbeda untuk *input* kunci yang berbeda.

Algoritma proses pembentukan tabel *S-Box* adalah sebagai berikut.

```
{1. Inisialisasi nilai TT[0] - T[7]}
    TT(0) = FHexToBiner("726a8f3b")
    TT(1) = FHexToBiner("e69a3b5c")
    TT(2) = FHexToBiner("d3c71fe5")
    TT(3) = FHexToBiner("ab3c73d2")
    TT(4) = FHexToBiner("4d3a8eb3")
    TT(5) = FHexToBiner("0396d6e8")
    TT(6) = FHexToBiner("3d4c2f7a")
    TT(7) = FHexToBiner("9ee27cf3")
{2. Pecah kunci (128 bit) menjadi 4 kelompok dan
masukkan nilainya ke T[0], T[1], T[2], T[3]}
    X = ""
    For N = 1 To Len(pcKunci)
      X = X &
    FormatS(FDecToBiner(Asc(Mid(pcKunci,N,
    1))),"0",8)
    Next N
    T(0) = Mid(X, 1, 32)
    T(1) = Mid(X, 33, 32)
    T(2) = Mid(X, 65, 32)
    T(3) = Mid(X, 97, 32)
{3. Untuk N = 4 sampai 255, lakukan proses berikut}
    For N = 4 To 255
      {X = T[n-4] + T[n-1]}
      X = FAddBiner(T(N - 4), T(N - 1), 32)

      {T[n] = X>>3 XOR TT[X AND 7]}
      T(N) = FOpBiner("XOR", FShiftRight(X, 3), _
      TT(FBinerToDec(FOpBiner("AND", X, "111"))),
      32)
    Next N
{4. Untuk N = 0 sampai 22, lakukan proses berikut}
    For N = 0 To 22
      {T[n] = T[n] + T[n+89]}
      T(N) = FAddBiner(T(N), T(N + 89), 32)
```

Gambar 7. Algoritma proses pembentukan tabel *S-Box*

```
    Next N
{5. Set nilai untuk variabel di bawah ini}
    X = T(33)
    Z = FOpBiner("OR", T(59),
    FHexToBiner("01000001"), 32)
    Z = FOpBiner("AND", Z,
    FHexToBiner("FF7FFFFF"), 32)
    X = FAddBiner(FOpBiner("AND", X,
    FHexToBiner("FF7FFFFF"),32), Z, 32)
{6. Untuk N = 0 sampai 255, lakukan proses berikut}
    For N = 0 To 255
      {X = (X And FF7FFFFF) + Z}
      X = FAddBiner(FOpBiner("AND",X,_
          FHexToBiner("FF7FFFFF"),32),Z,32)
      {T[n] = T[n] AND 00FFFFFF XOR X}
      T(N) = FOpBiner("XOR", FOpBiner("AND",
    T(N), _
     FHexToBiner("00FFFFFF"), 32), X, 32)
    Next N
{7. Inisialisasi nilai untuk beberapa variabel berikut}
    T(256) = T(0)
    X = FOpBiner("AND", X, FDecToBiner(255), 32)
{8. Untuk N = 0 sampai T[255], lakukan proses berikut}
    For N = 0 To 255
      {Temp = (T[n XOR X] XOR X) AND 255}
      Temp = T(FBinerToDec(FOpBiner("XOR",
       FDecToBiner(N), X, 32)))
      Temp = FOpBiner("XOR", Temp, X, 32)
      Temp = FOpBiner("AND", Temp,
       FDecToBiner(255), 32)
      {T[n] = T[Temp]}
      T(N) = T(FBinerToDec(Temp))
        {T[X] = T[n+1]}
      T(FBinerToDec(X)) = T(N + 1)
Next N
```

Gambar 7. Algoritma proses pembentukan tabel *S-Box* (lanjutan)

Perangkat lunak pembelajaran ini memiliki beberapa *form*, seperti *FormMain*.

1. *Form*Teori.
2. *FormInput* Proses *S-Box*.
3. *Form* Proses *S-Box*.
4. *FormInput* Proses Pembentukan Kunci.
5. *Form* Proses Pembentukan Kunci.
6. *FormInput* Proses Enkripsi.
7. *FormInput* Proses Dekripsi.
8. *Form* Proses Enkripsi / Dekripsi.
9. *Form* Tabel *S-Box*.
10. *Form* Hasil Pembentukan Kunci.
11. *Form About*.
12. *Form* Teori Kriptografi.

3.6 Bahasa Pemograman dan Desain

Bahasa pemograman dan desain yang digunakan pada pembuatan aplikasi ini adalah: menggunakan *Visual Basic 6.*

Proses kriptografi metode WAKE terdiri atas 4 (empat) proses, yaitu :

1. Proses Pembentukan Tabel *S-Box*.
2. Proses Pembentukan Kunci.
3. Proses Enkripsi.
4. Proses Dekripsi.

Inti dari metode WAKE terletak pada proses pembentukan tabel *S-Box* dan proses pembentukan kunci. Proses enkripsi dan dekripsi hanya berupa operasi XOR dari *plaintext* dan kunci untuk menghasilkan *ciphertext* dan operasi XOR dari *ciphertext* dan kunci untuk menghasilkan *plaintext*.

4. PEMBAHASAN

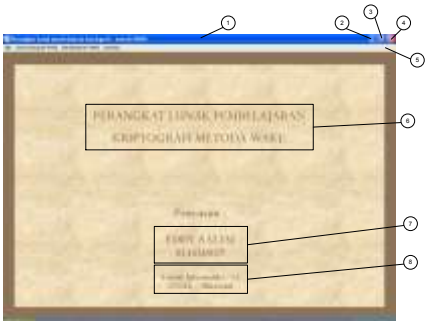
Proses dekripsi pada metode WAKE adalah melakukan operasi XOR dari *ciphertext* (32 bit) dan kunci (32 bit) untuk menghasilkan *plaintext* (32 bit).

Algoritma proses dekripsi adalah sebagai berikut.

```
    {Text untuk di-dekripsi (ciphertext)}
    strCipher = pcDekripsi

    {Ubah ciphertext dari karakter ascii ke bentuk biner}
    X = ""
    For N = 1 To Len(strCipher)
      X = X &
FormatS(FDecToBiner(Asc(Mid(strCipher,N,1))),"0",8)
    Next N

    {XOR-kan ciphertext dengan kunci}
    X = FOpBiner("XOR", X, strKunciBiner, 32)

    {Hasil XOR - ubah menjadi karakter ascii}
    cHasil = ""
    For N = 8 To 32 Step 8
      cHasil = cHasil & Chr(FBinerToDec(Mid(X, N -
7, 8)))
    Next N

    {Plaintext – hasil dekripsi}
strPlain = cHasil
```

Gambar 8. Algoritma Proses Deskripsi

4.1 Form *Menu*



Gambar 9 *User* Bagian *Index*

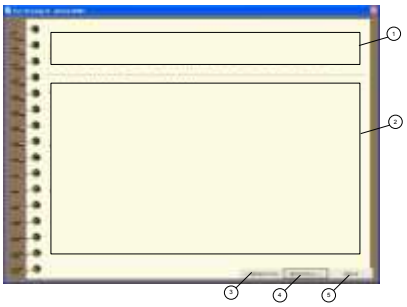
Keterangan :

1 : *title bar*, berisikan tulisan ‘Perangkat lunak pembelajaran kriptografi – metode WAKE’.
2 : tombol ‘Minimize’, berfungsi untuk mengecilkan tampilan *form*.
3 : tombol ‘Maximize’, berfungsi untuk memperbesar tampilan *form*.
4 : tombol ‘*Close*’, berfungsi untuk menutup *form* dan keluar dari *program*.
5 : *menu bar* yang berisikan menu ‘File’, ‘Teori Kriptografi WAKE’, ‘Pembelajaran WAKE’, dan ‘Tentang’.
6 : nama perangkat lunak.

ISSN: 2460-173X

7 : data penyusun skripsi atau pembuat perangkat lunak.

4.2 Form Teori



Gambar 10. *Form Teori*

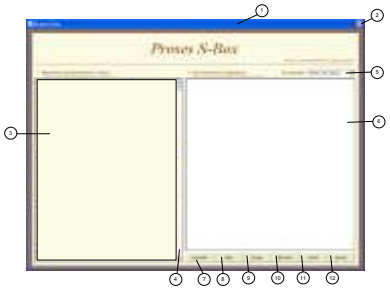
Keterangan:
1: judul teori.
2: daerah tampilan teori.
3: tombol '*Sebelumnya*', berfungsi melihat halaman teori sebelumnya.
4: tombol '*Berikutnya*', berfungsi melihat ke halaman teori selanjutnya.
5: tombol '*Keluar*', berfungsi untuk keluar dari *form* 'Teori' dan kembali ke *form* '*Main*'.

4.3 Form Input Proses S-Box

Gambar 11. Form Input Proses S-Box

Keterangan:
1 : *title bar*, berisikan tulisan 'Proses S-Box'.
2 : tombol '*Close*', berfungsi untuk menutup *form* 'Proses S-Box'.
3 : *textbox* sebagai tempat pengisian *input* kunci.
4 : *option button* 'Biner', berfungsi untuk memilih proses ditampilkan dalam biner.
5 : *option button* 'Heksa Desimal', berfungsi untuk memilih proses ditampilkan dalam bentuk heksa desimal.
6 : *label* 'Lihat Hasil / Tabel S-Box', berfungsi untuk menampilkan *form* 'Tabel S-Box'.
7 : tombol 'Proses kerja S-Box', berfungsi untuk menampilkan *form* 'Proses S-Box'.
8 : tombol 'Keluar', berfungsi untuk keluar dari *form* 'Input Proses S-Box' dan kembali ke *form* 'Main'.

4.4 Form Proses S-Box



Gambar 12. Form Proses *S-Box*

Keterangan:
1 : *title bar*, berisikan tulisan 'Proses S-Box'.
2 : tombol '*Close*', berfungsi untuk menutup *form* 'Proses S-Box'.
3 : daerah tampilanalgoritma untuk pembentukan tabel S-Box.
4 : *vertical scroll bar*, berfungsi untuk menggulung (*scroll*) tampilan algoritma secara vertikal.
5: *combo box* 'Kecepatan', berfungsi untuk memilih kecepatan proses.
6 : daerah tampilan hasil eksekusi proses.
7 : tombol 'Jalankan', berfungsi untuk memulai proses pembentukan tabel *S-Box*.
8 : tombol 'Skip', berfungsi untuk menghasilkan tabel *S-Box* tanpa melalui animasi tahapan–tahapan proses yang ada.
9 : tombol 'Ulang', berfungsi untuk mengulangi proses pembentukan tabel *S-Box*.
10: tombol 'Simpan', berfungsi untuk menyimpan hasil eksekusi.
11: tombol 'Cetak', berfungsi untuk mencetak hasil eksekusi.
12: tombol 'Keluar', berfungsi untuk keluar dari *form* 'Proses S-Box'.

4.2 Form Input Proses Pembentukan Kunci

Gambar 13. Form Input Proses Pembentukan Kunci

Keterangan:
1 : *title bar*, berisikan tulisan 'Proses Pembentukan Kunci'.
2 : tombol '*Close*', berfungsi untuk menutup *form* 'Input Proses Pembentukan Kunci'.

 ISSN: 2460-173X

3 : *option button* 'Biner', berfungsi untuk memilih proses ditampilkan dalam biner.
4 : *option button* 'Heksa Desimal', berfungsi untuk memilih proses ditampilkan dalam bentuk heksa desimal.
5 : *textbox* sebagai tempat pengisian banyaknya putaran untuk menghasilkan kunci.
6 : *updown* untuk mengatur isi dari *textbox* (5).
7 : *label* 'Lihat Tabel S-Box', berfungsi untuk menampilkan *form* 'Tabel S-Box'.
8 : *label* 'Kunci yang terbentuk', berfungsi untuk menampilkan *form* 'Hasil Pembentukan Kunci'.
9 : tombol 'Proses', berfungsi untuk menampilkan *form* 'Proses Pembentukan Kunci'.
10: tombol 'Keluar', berfungsi untuk keluar dari *form* 'Proses Pembentukan Kunci' dan kembali ke *form* '*Main*'.

4.3 Form Proses Pembentukan Kunci

Gambar 14. Form Proses Pembentukan Kunci

Keterangan:
1 : *title bar*, berisikan tulisan 'Proses Pembentukan Kunci'.
2 : tombol '*Close*', berfungsi untuk menutup *form* 'Proses Pembentukan Kunci'.
3 : daerah tampilandiagram pembentukan kunci.
4 : *combo box* 'Kecepatan', berfungsi untuk memilih kecepatan proses.
5 : daerah tampilan hasil eksekusi proses.
6 : tombol 'Jalankan', berfungsi untuk memulai proses pembentukan kunci.
7 : tombol 'Skip', berfungsi untuk menghasilkan kuncitanpa melalui animasi tahapan– tahapan proses yang ada.
8 : tombol 'Ulang', berfungsi untuk mengulangi proses pembentukan kunci.
9 : tombol 'Simpan', berfungsi untuk menyimpan hasil eksekusi.
10 : tombol 'Cetak', berfungsi untuk mencetak hasil eksekusi.
11 : tombol 'Keluar', berfungsi untuk keluar dari *form* 'Proses Pembentukan Kunci'.

4.4 Form Input Proses Enkripsi

Gambar 15. Rancangan Form Input Proses Enkripsi

Keterangan :
1 : *title bar*, berisikan tulisan 'Proses Enkripsi'.
2 : tombol '*Close*', berfungsi untuk menutup *form* 'Input Proses Enkripsi'.
3 : *textbox* sebagai tempat pengisian *inputplaintext*.
4 : *option button* 'Biner', berfungsi untuk memilih proses ditampilkan dalam biner.
5 : *option button* 'Heksa Desimal', berfungsi untuk memilih proses ditampilkan dalam bentuk heksa desimal.
6 : *label* 'Lihat Tabel S-Box', berfungsi untuk menampilkan *form* 'Tabel S-Box'.
7 : *label* 'Kunci yang terbentuk', berfungsi untuk menampilkan *form* 'Hasil Pembentukan Kunci'.
8 : tombol 'Proses', berfungsi untuk menampilkan *form* 'Proses Enkripsi / Dekripsi'.
9 : tombol 'Keluar', berfungsi untuk keluar dari *form* 'Input Proses Enkripsi' dan kembali ke *form* '*Main*'.

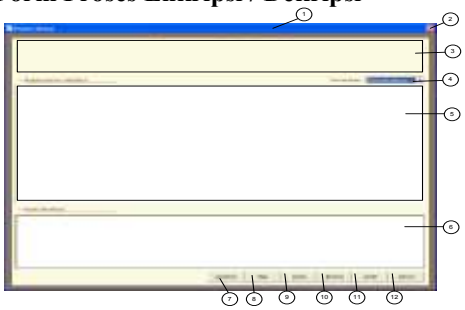
4.5 Form Input Proses Dekripsi

Gambar 16. Rancangan Form Input Proses Dekripsi

Keterangan:
1 : *title bar*, berisikan tulisan 'Proses Dekripsi'.
2 : tombol '*Close*', berfungsi untuk menutup *form* 'Input Proses Dekripsi'.
3 : *textbox* sebagai tempat pengisian *inputciphertext*.
4 : *option button* 'Biner', berfungsi untuk memilih proses ditampilkan dalam biner.
5 : *option button* 'Heksa Desimal', berfungsi untuk memilih proses ditampilkan dalam bentuk heksa desimal.

6 : *label* 'Lihat Tabel S-Box', berfungsi untuk menampilkan *form* 'Tabel S-Box'.
7 : *label* 'Kunci yang terbentuk', berfungsi untuk menampilkan *form* 'Hasil Pembentukan Kunci'.
8 : tombol 'Proses', berfungsi untuk menampilkan *form* 'Proses Enkripsi / Dekripsi'.
9 : tombol 'Keluar', berfungsi untuk keluar dari *form* 'Input Proses Dekripsi' dan kembali ke *form* '*Main*'.

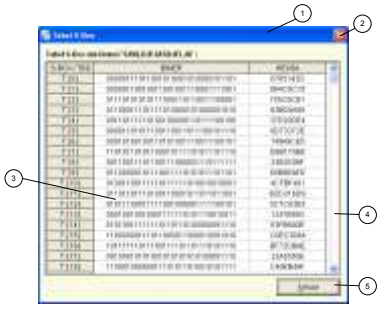
4.6 Form Proses Enkripsi / Dekripsi



Gambar 17. Rancangan Form Proses Enkripsi / Dekripsi

Keterangan:
1 : *title bar*, berisikan tulisan 'Proses Enkripsi' atau 'Proses Dekripsi'.
2 : tombol '*Close*', berfungsi untuk menutup *form*.
3 : *label*, untuk menampilkan nama proses (proses enkripsi atau dekripsi).
4 : *combo box* 'Kecepatan', berfungsi untuk memilih kecepatan proses.
5 : daerah tampilandiagram enkripsi atau dekripsi.
6 : daerah tampilan hasil eksekusi proses.
7 : tombol 'Jalankan', berfungsi untuk memulai proses enkripsi atau dekripsi.
8 : tombol 'Skip', berfungsi untuk menghasilkan *plaintext* atau *ciphertext*tanpa melalui animasi tahapan– tahapan proses yang ada.
9 : tombol 'Ulang', berfungsi untuk mengulangi proses enkripsi atau dekripsi.
10 : tombol 'Simpan', berfungsi untuk menyimpan hasil eksekusi.
11 : tombol 'Cetak', berfungsi untuk mencetak hasil eksekusi.
12 : tombol 'Keluar', berfungsi untuk keluar dari *form*.

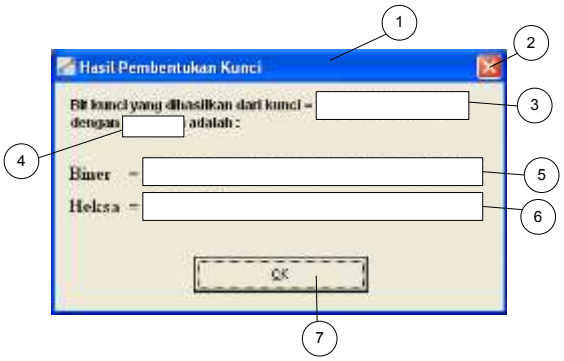
4.7 Form Tabel S-Box



Gambar 18 *Rancangan Form Tabel S-Box*

Keterangan:
1 : *title bar*, berisikan tulisan 'Tabel S-Box'.
2 : tombol '*Close*', berfungsi untuk menutup *form* 'Tabel S-Box'.
3 : tabel S-Box.
4 : *vertical scroll bar,* untukmenggulung (*scroll*) tabel S-Box secara horizontal.
5 : tombol 'Keluar', berfungsi untuk keluar dari form 'Tabel S-Box'.

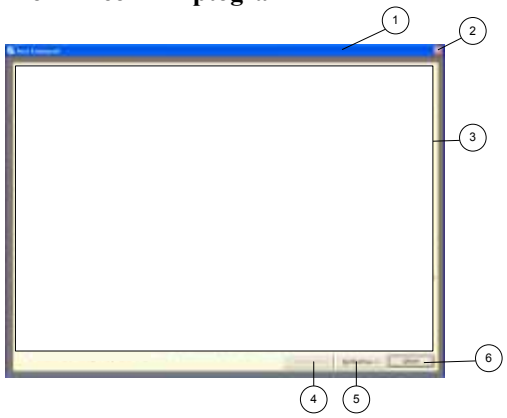
4.8 Form Hasil Pembentukan Kunci



Gambar 19. Rancangan Form Hasil Pembentukan Kunci

Keterangan:
1 : *title bar*, berisikan tulisan 'Hasil Pembentukan Kunci'.
2 : tombol '*Close*', berfungsi untuk menutup *form*.
3 : *label*, untuk menampilkan *input key*.
4 : *label*, untuk menampilkan *input* putaran kunci.
5 : *label,* untukmenampilkan hasil pembentukan kunci dalam bentuk biner.
6 : *label,* untukmenampilkan hasil pembentukan kunci dalam bentuk heksa.
7 : tombol 'OK', berfungsi untuk keluar dari *form*.

4.9 Form Teori Kriptografi



Gambar 20. Rancangan Form Teori Kriptografi

Keterangan:
1 : *title bar*, berisikan tulisan 'Teori Kriptografi'.
2 : tombol '*Close*', berfungsi untuk menutup *form*.
3 : daerah tampilan teori kriptografi.
4 : tombol '< Sebelumnya' untuk menampilkan halaman teori sebelumnya.

5 : tombol 'Berikutnya >' untuk menampilkan halaman teori selanjutnya.
6 : tombol 'Keluar' untuk keluar dari *form* dan kembali ke *form* '*About*'.

5. KESIMPULAN

Setelah selesai menyelesaikan perancangan perangkat lunak pembelajaran kriptografi metode WAKE ini, dapat ditarik kesimpulan sebagai berikut :

1. Perangkat lunak ini menunjukkan setiap langkah dan tahapan proses – proses (proses pembentukan tabel *S-Box*, proses pembentukan kunci, proses enkripsi dan proses dekripsi) yang terdapat di dalam kriptografi metode WAKE, sehingga dapat membantu pemahaman atau pembelajaran prosedur kerja atau algoritma dari metode kriptografi tersebut.
2. Aplikasi integrasi *client server* tidak dapat mensupport karakter huruf kapital dan *form choicegroup*. Hal ini disebabkan karena dalam pemrograman IDEA perlu dilakukan konvert data beberapa kali untuk memasukan data inputan ke dalam fungsi enkrip dan dekrip, sedangkan pada PHP dapat langsung mengenali berbagai tipe data tanpa mengkonvert.

DAFTAR PUSTAKA

Ario, S., 2001, ***Microsoft Visual Basic 6.0***, PT. Elex Media Komputindo.

Bruce, S., 1996, *Applied Crytography*, Second Edition, John Wiley & Sons, Inc.

K. Jusuf Ir, M.T., 2002,Kriptografi, *Keamanan Internet dan Jaringan Komunikasi*, Penerbit Informatika Bandung.

Reza , P., *Kriptografi, Jurnal Saint, Vol 4, Sekolah Tinggi Manajemen Informatika dan Komputer Pontianak, 2013.*

Sodhi, Jag, , 1991, *Software Engineering Methods, Management, and CASE Tools, TAB Professional dan Reference Books*, Amerika.

Wardani , Kholidya Yuli, M.Zen S.Hadi, ST. MSc, Mike Yuliana, ST.MT, 2012, *Implementasi Metode Kriptografi Idea pada Priority Dealer Untuk Layanan Pemesanan dan Laporan Penjualan* Politeknik Elektronika Negeri Surabaya.

http://www.cix.co.uk/~klockstone/wake.htm, tanggal 11 Juli 2005.

http://www.cix.co.uk/~klockstone/hereward.htm, tanggal 11 Juli 2005.

http://eprint.iacr.org/2001/065.pdf, tanggal 11 Juli 2005.

Biodata Penulis

Agustian Noor**,** lahir di Banjarmasin pada tanggal 2 Agustus 1984. Penulis mendapatkan gelar M.Kom dari Teknik Informatika Universitas Dian Nusawantoro Semarang Indonesia pada tahun 2015. Sejak tahun 2016, penulis bekerja sebagai Dosen Teknik Informatika Politeknik Negeri Tanah Laut.