e-ISSN : 2722-4368

Aplikasi Penyandian Record Menggunakan Metode Twofish

Asmawi
STMIK Budi Darma Medan, Jl.Sisingamangaraja No.338 Simpang Limun Medan, Indonesia
asmawi.asma@yahoo.com

Abstrak. *Perkembangan teknologi semakin mengalami kemajuan, misalnya untuk keamanan data. Keamanan data merupakan hal yang sangat penting dalam menjaga kerahasiaan informasi terutama informasi yang sensitive yang hanya boleh diketahui isinya hanya pihak yang berhak saja, apalagi penyimpanan data dilakukan pada komputer, apabila data tersebut tidak diamankan terlebih dahulu maka data tersebut akan diketahui oleh pihak-pihak yang tidak berhak dengan adanya pengamanan data pada record maka data yang ada pada komputer tidak mudah dilihat dan tidak bisa dirubah oleh sembarangan pihak yang tidak bertanggung jawab sehingga data akan aman dan tidak bisa diganggu gugat oleh pihak manapun. Salah satu pengamanan data adalah dengan menggunakan sistem kriptografi yaitu penyandian isi informasi (plaintext) tersebut menjadi isi yang tidak bisa dipahami melalui proses enkripsi (enchipher) dan untuk memperoleh kembali informasi yang asli, dilakukan proses deskripsi (deschiper) disertai dengan menggunakan kunci yang benar. Penyandian data atau kriptografi adalah ilmu yang mempelajari tentang penyembunyian huruf atau tulisan sehingga membuat tulisan tersebut tidak dapat dibaca oleh orang yang tidak berkepentingan. Kriptografi sudah dipakai sejak jaman Julius Caesar dimana akan mengirimkan pesan kepada panglimanaya tetapi tidak mempunyai kurir pembawa pesan tersebut. Kriptografi mempunyai 2 (dua) bagian yang penting yaitu enkripsi dan deskripsi. Enkripsi adalah adalah proses dari penyandian pesan asli menjadi pesan yang tidak dapat diartikan seperti aslinya. Deskripsi adalah merubah pesan yang sudah diartikan menjadi aslinya pesan asli biasanya disebut plaintext, sedangkan pesan yang sudah diartikan disebut chipertext.*

Kata Kunci : Kriptografi, Pesan Rahasia, Enkripsi, Dekripsi pada penyandian file record.

Abstract. *Technological developments are increasingly progressing, for example for data security. Data security is very important in maintaining the confidentiality of information, especially sensitive information, which only authorized parties can know, moreover data storage is done on a computer, if the data is not secured in advance then the data will be known by the parties who are entitled to it. is not entitled to the security of data on the record, the data on the computer is not easily seen and cannot be changed by irresponsible parties so that the data will be safe and cannot be contested by any party One of the data safeguards is by using a cryptographic system, which is the encryption of the contents of the information (plaintext) into incomprehensible content through an encryption process (encoding) and to retrieve the original information, a description process (deschiper) is carried out accompanied by using the correct key. Data encryption or cryptography is the study of hiding letters or writings so that they cannot be read by unauthorized people. Cryptography has been used since the time of Julius Caesar, who sent messages to the commander but did not have a courier to carry the message. Cryptography has 2 (two) important parts, namely encryption and description. Encryption is the process of encrypting the original message into a message that cannot be interpreted as the original. Description is to change a message that has been interpreted to be the original message, usually called plaintext, while the message that has been interpreted is called ciphertext.*

Keyword : Cryptography, Secret Message, Encryption, Decryption on encoding file records.

PENDAHULUAN

Keamanan data merupakan hal yang sangat penting dalam menjaga kerahasiaan informasi terutama informasi yang sensitif yang hanya boleh diketahui isinya hanya oleh pihak yang berhak saja, apalagi penyimpanan data dilakukan pada komputer, apabila data tersebut tidak diamankan terlebih dahulu maka data tersebut akan diketahui oleh pihak-pihak yang tidak berhak dengan adanya pengamanan data pada record maka data yang ada pada komputer tidak mudah dilihat dan

tidak bisa dirubah oleh sembarangan pihak yang tidak bertanggung jawab khususnya penyandian pada database pada laptop anda [1], [2]. Aplikasi penyandian record memberikan manfaat yang sangat besar dalam industry pemonggraman saat ini. Salah satunya adalah dalam penyimpanan data dan kerahasiaan data tersebut tidak bisa diganggu gugat oleh pihak yang tidak berkepentingan, salah satunya adalah dalam penyimpanan data atau informasi yang sangat rahasia didalamnya. Data record yang telah disandiakan akan lebih sulit untuk membukanya oleh orang-orang lain yang tidak bertanggung jawab. Selain pada proses penyandian data record juga bisa digunakan dalam proses penyandian field dan yang lain-lain[3], [4].

Tujuan penelitian adalah untuk melakukan proses penyandian data dan mengenkripsi dan mendeskripsi menggunakan metode twofish, Menerapakan metode twofish dan penyandian record [5], [6] menggunakan metode tersebut dan menerapkan metode twofish dan memprosesnya menggunakan Microsoft visual basic net 2008.

METODOLOGI PENELITIAN

2.1 Kriptografi

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Crypto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Jadi kriptografi berarti *secret writing* (tulisan rahasia). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ketempat lain [4], [7]. Kriptografi pada dasarnya sudah dikenal sejak lama. Menurut catatan sejarah, kriptografi sudah digunakan oleh bangsa Mesir sejak 4000 tahun yang lalu oleh raja - raja Mesir pada saat perang untuk mengirimkan pesan rahasia kepada panglima perangnya melalui kurir - kurinya. Orang yang melakukan penyandian ini disebut kriptografer, sedangkan orang yang mendalami ilmu dan seni dalam membuka atau memecahkan suatu algoritma kriptografi tanpa harus mengetahui kuncinya disebut kriptanalis. 4 tujuan mendasar:

1. Kerahasiaan (*Confidentiality*)
   Yaitu layanan agar isi pesan yang dikirimkan tetap rahasia dan tidak diketahui oleh pihak lain (kecuali pihak pengirim, pihak penerima / pihak - pihak memiliki ijin). Layanan ini direalisasikan dengan menyandikan pesan menjadi cipherteks.
2. Integritas data (data *integrity*)
   Adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain menyangkut penyisipan, penghapusan, dan pensubtitusian data lain ke dalam data yang sebenarnya.
3. Otentikasi (*authentication*)
   Yaitu layanan yang berhubungan dengan identifikasi. Baik otentikasi pihak-pihak yang terlibat dalam pengiriman data maupun otentikasi keaslian data/informasi.
4. Nirpenyangkalan (*non-repudiation*)
   Yaitu layanan yang dapat mencegah suatu pihak untuk menyangkal aksi yang dilakukan sebelumnya (menyangkal bahwa pesan tersebut berasal dirinya).

Dalam penelitian ini menggunakan pengkodean ASCII, karena kode ASCII, karena kode ASCII merupakan kode standar yang digunakan dalam komputer mikro atau Personal Computer (PC). Pengujian dan aplikasi penelitian ini nantinya akan menggunakan personal komputer.

2.2. Twofish

Pada tahun 1972 dan 1974, US the National Bureau of Standards (sekarang bernama the National Institute of Standards and Technology, atau NIST) mengeluarkan publikasi pertama untuk sebuah standar enkripsi, yang menghasilkan algoritma data Encryption Standard (DES), yang tidak dapat disangkal sebagai algoritma kriptografi yang sangat terkenal dan sangat berhasil [8], [9]. Algoritma Twofish merupakan algoritma kuat yang sampai saat ini dinyatakan aman karena masih

belum ada serangan kriptanalisis yang benar – benar dapat mematahkan algoritma ini. Algoritma ini juga tidak dipatenkan sehingga penggunaannya pada alat enkripsi tidak perlu mengeluarkan biaya.

2.3. Tujuan Desain *TwoFish*

Algoritma *Twofish* didesain untuk memenuhi kriteria yang ditetapkan oleh NIST untuk sayembara penentuan standar algoritma[10] . Kriteria tersebut diantaranya adalah :

1. Menggunakan 128-bit enkripsi dengan metode blok *cipher*.
2. Panjang kunci 128 bit, 192 bit, dan 256 bit.
3. Tidak memiliki kunci lemah
4. Efisien baik jika digunakan di Intel Pentium Pro maupun perangkat lunak ataupun keras lainnya
5. Memiliki desain yang fleksibel sehingga dapat digunakan untuk stream chiper, hash function, dan MAC.
6. *Design* yang sederhana.

Kriteria tambahan yang dimiliki oleh algoritma *Twofish* adalah :

1. Dapat menerima kunci lebih dari 256 bit
2. Untuk versi dengan optimasi penuh proses enkripsi data dapat dilakukan kurang dari 500 *clock cycle* per blok pada Pentium, Pentium Pro.
3. Untuk pemrosesan 32 blok dengan 128 bit kunci dapat memakan waktu yang lebih sedikit.
4. Tidak memiliki operasi yang dapat mengurangi efisiensi jika digunakan pada mikroprosesor 8-bit, 16-bit, 32-bit maupun 64 bit.
5. Memiliki berbagai variasi performansi dari *key schedule*.

2.4. Blok Pembangun TwoFish

Secara garis besar algoritma twofish dibangun dari beberapa algoritma utama, algoritma - algoritma tersebut diambil dari prinsip pembangunan algoritma cipher blok. Ada 6 prinsip yang digunakan yaitu :

1. Jaringan Feistel
   Hampir semua algoritma *cipher* blok bekerja dalam model jaringan Feistel. Jaringan Feistel ditemukan oleh Horst Feistel dalam desainnya tentang Lucifer, dan dipopulerkan oleh DES. Jaringan Feistel adalah metode umum untuk mentransformasi fungsi apapun (biasa disebut fungsi F) ke dalam permutasi. Beberapa algoritma kriptografi lain yang menggunakan jaringan Feistel misalnya LOKI, GOST, FEAL, Blowfish, Khufu Khafre, dan RC-5. Model jaringan Feistel bersifat reversible, untuk proses enkripsi dan dekripsi, sehingga kita tidak perlu membuat algoritma baru untuk mendekripsi cipherteks menjadi plainteks.
2. Kotak S (S-Boxes)
   Kotak-S adalah matriks yang berisi subst itusi non-linear yang memetakan satu atau lebih bit dengan satu atau lebih bit yang lain dan digunakan di banyak *cipher* blok. Kotak-S memiliki ukuran input dan ukuran output yang bervariasi. Ada empat pendekatan yang digunakan dalam mengisi Kotak-S : dipilih secara acak, dipilih secara acak lalu diuji, dibut oleh orang, dihitung secara matematis. Kotak-S pertama digunakan di Lucifer, lalu DES dan diikuti banyak algoritma enkripsi yang lain. *Twofish* menggunakan empat buah 8x8 bit Kotak-S yang berbeda, bijektif, dan bergantung pada kunci. Kotak-S ini dibuat menggunakan 8x8 bit permutasi dan material kunci.
3. MDS Matrices
   Kode MDS (Maximum Distance Separable) pada sebuah *field* adalah pemetaan liner dari x elemen *field* ke y elemen *field*, dan menghasilkan vektor komposit x + y elemen, dengan ketentuan bahwa jumlah minimum dari elemen bukan nol pada setiap vektor bukan nol paling sedikit y + 1. Dengan kata lain, jumlah elemen yang berbeda diantara dua vektor berbeda yang dihasilkan oleh pemetaan MDS paling sedikit y + 1. Dapat dibuktikan dengan mudah bahwa tidak ada pemetaan yang dapat memiliki jarak pisah yang lebih besar diantara dua vektor yang

berbeda, maka disebut jarak pisah maksimum (maximum distance separable). Pemetaan MDS dapat direpresentasikan dengan sebuah MDS matriks yang teridiri dari x × y elemen.
Kode perbaikan-kesalahan Reed-Solomon (RS) adalah MDS. Kondisi yang diperlukan untuk sebuah x × y matriks untuk menjadi MDS adalah semua kemungkinan submatriks kotak, yang diperoleh dengan membuang kolom atau baris, adalah tidak singular. Serge Vaudenay pertama kali mengajukan matriks MDS sebagai elemen desain kode. Shark dan Square menggunakan matrika MDS, meskipun konstruksinya pertama kali ditemukan di kode Manta yang tidak dipublikasikan. Twofish menggunakan matriks MDS tunggal 4x4.

4. Transformasi Pseudo Hadamard
   Transformasi Pseudo-Hadamard (PHT) adalah sebuah operasi pencampuran sederhana yang berjalan secara cepat dalam perangkat lunak. 32-bit PHT dengan dua masukkan didefinisikan sebagai :

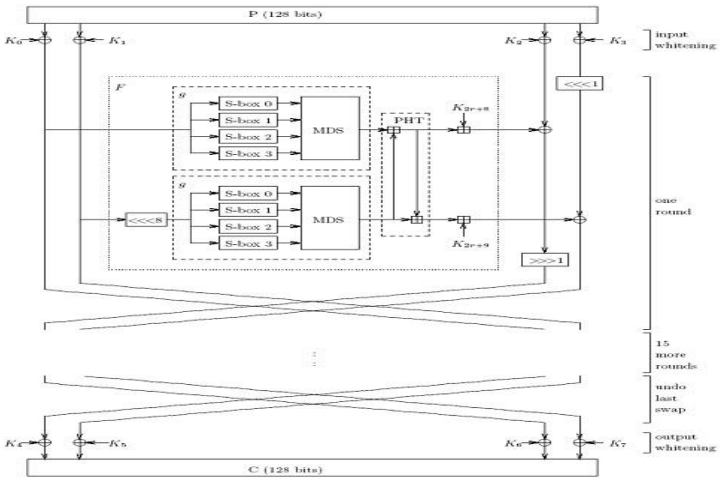
$$a' = a + b \bmod 2^{32}$$
$$b' = a + 2b \bmod 2^{32}$$

   SAFER menggunakan 8-bit PHT untuk difusinya. Twofish menggunakan 32-bit PHT untuk mengubah keluaran dari fungsi g- nya. PHT ini dapat dieksekusi dalam dua opcodes di mikroprosesor modern seperti keluarga Pentium.
5. *Whitening*
   Whitening, sebuah teknik meng-XOR-kan material kunci sebelum putaran pertama dan setelah putaran terakhir, digunakan oleh Merkle dalam Khufu/Khafre, dan ditemukan oleh Rivest untuk DES-X. *Whitening* menambah tingkat kesulitan serangan pencarian kunci terhadap ciphertext, dengan menyembunyikan masukkan spesifik terhadap putaran pertama dan putaran terakhir dari fungsi F.
   Twofish meng-XOR-kan 128-bit sub-kunci sebelum putaran Feistel yang pertama, dan 128-bit lagi setelah putaran Feistel terakhir. Sub-kunci ini diperhitungkan dengan cara yang sama seperti sub-kunci putaran, tetapi tidak digunakan di tempat lain dalam cipher.
6. Penjadwalan Kunci
   Penjadwalan kunci adalah proses pengubahan bit-bit kunci menjadi sub-kunci tiap putaran yang dapat digunakan oleh cipher. Twofish memerlukan banyak material kunci dan memiliki penjadwalan kunci yang rumit. Untuk memfasilitasi analisis, penjadwalan kunci menggunakan primitif yang sama seperti fungsi putaran.

2.5. Algoritma *Twofish*

Twofish merupakan algoritma kriptografi kunci simetrik *cipher* blok dengan panjang setiap blok adalah tetap 128 bit. Sedangkan kunci yang dapat diterima adalah: 128, 192, atau 256 bit. Twofish memanfaatkan teknik pemanipulasian bit, kotak permutasi/pemutihan, jaringan feistel, pemutaran ulang dengan pergiliran kunci dengan jumlah perputaran dan pergiliran kunci sebanyak 16 kali, tranformasi pseudo-Hadamard, ekspansi dan filter, dan kotak MDS (Most Distance Separable).

e-ISSN : 2722-4368



Gambar 1. Struktur Algoritma TwoFish

HASIL DAN PEMBAHASAN

3.1 Proses Pembentukan S-Box

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* = ‘STMIK BUDIDARMA ‘, maka proses pembentukan tabel *S-Box* dalam heksadesimal adalah sebagai berikut :
LANGKAH 1 - Inisialisasi nilai TT[0] ... TT[7]

TT[0] = 726A8F3B
TT[1] = E69A3B5C
TT[2] = D3C71FE5
TT[3] = AB3C73D2
TT[4] = 4D3A8EB3
TT[5] = 0396D6E8
TT[6] = 3D4C2F7A
TT[7] = 9EE27CF3

LANGKAH 2 - Pecah kunci menjadi 4 kelompok besar dan masukkan pada T[0] ... T[3]
Kunci : 'STMIK BUDIDARMA '
Hasil S:DEC=HEX
Kode ascii dari 'S' = 83 = 53
Kode ascii dari 'T' = 84 = 54
Kode ascii dari 'M' = 77 = 4D
Kode ascii dari 'I' = 73 = 49
Kode ascii dari 'K' = 75 = 4B
Kode ascii dari ' ' = 32 = 20
Kode ascii dari 'B' = 66 = 42
Kode ascii dari 'U' = 85 = 55
Kode ascii dari 'D' = 68 = 44
Kode ascii dari 'I' = 73 = 49
Kode ascii dari 'D' = 68 = 44

e-ISSN : 2722-4368

Kode ascii dari 'A' = 65 = 41
Kode ascii dari 'R' = 82 = 52
Kode ascii dari 'M' = 77 = 4D
Kode ascii dari 'A' = 65 = 41
Kode ascii dari ' ' = 32 = 20
Kunci (dalam heksa) = 53544D494B20425544494441524D4120
T[0] = K[0] = 53544D49
T[1] = K[1] = 4B204255
T[2] = K[2] = 44494441
T[3] = K[3] = 524D412

LANGKAH 3 - Untuk n = 4 sampai 255, lakukan prosedur berikut.
n = 4
-> X = T[0] + T[3] = 53544D49 + 524D4120 = A5A18E69
-> X >> 3 (Shift Right 3 bit) = A5A18E69 >> 3 = 14B431CD
X AND 7 = A5A18E69 AND 7(10) = 1
T[4] = X >> 3 XOR TT[X AND 7] = 14B431CD XOR TT[1] = F22E0A91
n = 5
n = 11
-> X = T[7] + T[10] = 8288E7A5 + 7F8E5CF4 = 02174499
-> X >> 3 (Shift Right 3 bit) = 02174499 >> 3 = 0042E893
X AND 7 = 02174499 AND 7(10) = 1
T[11] = X >> 3 XOR TT[X AND 7] = 0042E893 XOR TT[1] = E6D8D3CF

n = 12
dan seterus nya……

LANGKAH 4 - Untuk n = 0 sampai 22, lakukan prosedur berikut.
n = 0
T[1] = T[1] + T[90] = 4B204255 + 773D8DCD = C25DD022

T[0] = T[0] + T[89] = 53544D49 + 1CE3B995 = 703806DE
n = 1
n = 2
T[2] = T[2] + T[91] = 44494441 + 72A1384A = B6EA7C8B
n = 3
T[3] = T[3] + T[92] = 524D4120 + 32715E5D = 84BE9F7D
n = 4
T[4] = T[4] + T[93] = F22E0A91 + DA2DBD1B = CC5BC7AC
n = 5
T[5] = T[5] + T[94] = 3AE5E6E6 + 7847E666 = B32DCD4C
n = 6
T[6] = T[6] + T[95] = 91079997 + 6F37ACED = 003F4684
n = 7
T[7] = T[7] + T[96] = 8288E7A5 + C7F23E8C = 4A7B2631
n = 8
T[8] = T[8] + T[97] = 33DAF13C + 8AA18387 = BE7C74C3
n = 9
T[9] = T[9] + T[98] = DE1F04E1 + 03CBFBD5 = E1EB00B6
n = 10

e-ISSN : 2722-4368

T[10] = T[10] + T[99] = 7F8E5CF4 + DDA76AFD = 5D35C7F1
n = 11
T[11] = T[11] + T[100] = E6D8D3CF + F2290E6D = D901E23C
n = 12
dan seterusnya…

LANGKAH 5 - Set nilai untuk variabel di bawah ini.
X = 32FFA237
Z = T[59] OR 01000001 = FFF256A8 OR 01000001 = FFF256A9
Z = Z AND FF7FFFFF = FFF256A9 AND FF7FFFFF = FF7256A9
X = X AND FF7FFFFF = 32FFA237 AND FF7FFFFF = 31F1F8E0

LANGKAH 6 - Untuk n = 0 sampai 255, lakukan prosedur berikut.
n = 0
X = (31F1F8E0 AND FF7FFFFF) + FF7256A9 = 30E44F89
T[0] = 703806DE] AND 00FFFFFF XOR 30E44F89 = 30DC4957
n = 1
X = (30E44F89 AND FF7FFFFF) + FF7256A9 = 2FD6A632
T[1] = C25DD022] AND 00FFFFFF XOR 2FD6A632 = 2F8B7610

n = 2
X = (2FD6A632 AND FF7FFFFF) + FF7256A9 = 2EC8FCDB
T[2] = B6EA7C8B] AND 00FFFFFF XOR 2EC8FCDB = 2E228050

n = 3
X = (2EC8FCDB AND FF7FFFFF) + FF7256A9 = 2DBB5384
T[3] = 84BE9F7D] AND 00FFFFFF XOR 2DBB5384 = 2D05CCF9

n = 4
X = (2DBB5384 AND FF7FFFFF) + FF7256A9 = 2CADAA2D
T[4] = CC5BC7AC] AND 00FFFFFF XOR 2CADAA2D = 2CF66D81

n = 5
X = (2CADAA2D AND FF7FFFFF) + FF7256A9 = 2BA000D6
T[5] = B32DCD4C] AND 00FFFFFF XOR 2BA000D6 = 2B8DCD9A

n = 6
X = (2BA000D6 AND FF7FFFFF) + FF7256A9 = 2A92577F
T[6] = 003F4684] AND 00FFFFFF XOR 2A92577F = 2AAD11FB

n = 7
X = (2A92577F AND FF7FFFFF) + FF7256A9 = 2984AE28
T[7] = 4A7B2631] AND 00FFFFFF XOR 2984AE28 = 29FF8819

n = 8
X = (2984AE28 AND FF7FFFFF) + FF7256A9 = 287704D1
T[8] = BE7C74C3] AND 00FFFFFF XOR 287704D1 = 280B7012

n = 9
X = (287704D1 AND FF7FFFFF) + FF7256A9 = 27E95B7A

e-ISSN : 2722-4368

T[9] = E1EB00B6] AND 00FFFFFF XOR 27E95B7A = 27025BCC

n = 10
X = (27E95B7A AND FF7FFFFF) + FF7256A9 = 26DBB223
T[10] = 5D35C7F1] AND 00FFFFFF XOR 26DBB223 = 26EE75D2

n = 11
X = (26DBB223 AND FF7FFFFF) + FF7256A9 = 25CE08CC
T[11] = D901E23C] AND 00FFFFFF XOR 25CE08CC = 25CFEAF0

n = 12
dan seterusnya….

LANGKAH 8 - Untuk n = 0 sampai 255, lakukan prosedur berikut.
n = 0
Temp = T[224] XOR X AND 255 = 50CCC0AD XOR 000000E0 AND 255 = 0000004D
T[0] = T[77] = E3D54644
T[224] = T[1] = 2F8B7610

n = 1
Temp = T[225] XOR X AND 255 = 4FBF34E3 XOR 000000E0 AND 255 = 00000003
T[1] = T[3] = 2D05CCF9
T[224] = T[2] = 2E228050
Sampai 10….

n = 10
Temp = T[234] XOR X AND 255 = 46BADA9F XOR 000000E0 AND 255 = 0000007F
T[10] = T[127] = B189B3FD
T[224] = T[11] = 25CFEAF0
Dan seterusnya,…..

Nilais-bokuntukkunci‘STMIK BUDIDARMA’adalah :

T[0] = E3D54644
T[1] = 2D05CCF9
T[2] = 27025BCC
T[3] = 68EEF7AE
T[4] = BC2992DF
T[5] = EE50CD18
T[6] = 673B98DF
T[7] = 3BABA719
T[8] = 531835E2
T[9] = 9B99C275
T[10] = B189B3FD
T[11] = C9BF4B0E
T[12] = 609FF4BF
T[13] = F186130C
T[14] = 1830EA55
T[15] = 8C21AF54
T[16] = 46BADA9F

T[17] = DD484478
T[18] = 3229626C
T[19] = 4B96EAA2
T[20] = BE6A767C
T[21] = 37C1D252
T[22] = E6869B95
T[23] = CCA727E0
T[24] = E6869B95
T[25] = 7EB0F5DE
T[26] = BC2992DF
T[27] = 34D0A54C
T[28] = 84F4EAE8
T[29] = 72FD1692
T[30] = A4B5A8F8
Sampai menuju ;
T[256] = 30DC4957

3.2 Proses Pembentukan Kunci

Proses pembentukan kunci memerlukan *inputkunci* dengan panjang 128 bit biner atau 16 karakter *ascii*. Pertama-tama, *input* kunci dipecah menjadi 4 kelompok dan di-*set* sebagai nilai awal dari variabel A_0, B_0, C_0, D_0. Kemudian isi variabel A, B, C dan D dan ulangi sebanyak n-putaran yang di-*input*.

$A_{i+1} = M(A_i, D_i)$
$B_{i+1} = M(B_i, A_{i+1})$
$C_{i+1} = M(C_i, B_{i+1})$
$D_{i+1} = M(D_i, C_{i+1})$

Fungsi M(X, Y) = (X + Y)>>8 XOR T[(X + Y) AND 255]. Nilai dari D_i merupakan nilai dari kunci K_i. Proses ini dapat dilihat pada contoh berikut :
Misalkan *inputkey* : 'STMIK BUDIDARMA ' dan putaran kunci sebanyak 4 kali, maka proses pembentukan kunci dalam heksadesimal adalah sebagai berikut :
Kunci 'STMIK BUDIDARMA' diubah dalam bentuk BINER= 10011010 10101101 00000110 00110111

Kunci 'STMIK BUDIDARMA' diubah dalam bentuk biner = 01010011 01010100 01001101 01001001 01001011 00100000 01000010 01010101 01000100 01001001 01000100 01000001 01010010 01001101 01000001 00100000

Pecah kunci menjadi 4 kelompok dan masukkan ke A(0), B(0), C(0) dan D(0).
A(0) = 01010011010101000100110101001001
B(0) = 01001011001000000100001001010101
C(0) = 01000100010010010100010001000001
D(0) = 01010010010011010100000100100000

KUNCI PUTARAN 1

FungsiM(A[0],D[0]) =
FungsiM(01010011010101000100110101001001,01010010010011010100000100100000) = (01010011010101000100110101001001 + 01010010010011010100000100100000)>>8 XOR T[(01010011010101000100110101001001 + 01010010010011010100000100100000) AND

e-ISSN : 2722-4368

255(10)] = 10100101101000011000111001101001>>8 XOR T[105] = 00000000101001011010000110001110 XOR 10001010011111110111011011111110 = 10001010110110101101101011110000
A[1] = 10001010110110101101101011110000

FungsiM(B[0],A[1]) =
FungsiM(01001011001000000100001001010101,10001010110110101101101011110000) =
(01001011001000000100001001010101 + 10001010110110101101101011110000)>>8 XOR
T[(01001011001000000100001001010101 + 10001010110110101101101011110000) AND
255(10)] = 11010101111110110001110101000101>>8 XOR T[69] =
00000000110101011111101100011101 XOR 01101110001100010011011000011100 =
01101110111001001100110100000001
B[1] = 01101110111001001100110100000001

FungsiM(C[0],B[1]) =
FungsiM(01000100010010010100010001000001,01101110111001001100110100000001) =
(01000100010010010100010001000001 + 01101110111001001100110100000001)>>8 XOR
T[(01000100010010010100010001000001 + 01101110111001001100110100000001) AND
255(10)] = 10110011001011100001000101000010>>8 XOR T[66] =
00000000101100110010111000010001 XOR 01010101111110100111100000101110 =
01010101010010010101011000111111
C[1] = 01010101010010010101011000111111

FungsiM(D[0],C[1]) =
FungsiM(01010010010011010100000100100000,01010101010010010101011000111111) =
(01010010010011010100000100100000 + 01010101010010010101011000111111)>>8 XOR
T[(01010010010011010100000100100000 + 01010101010010010101011000111111) AND
255(10)] = 10100111100101101001011101011111>>8 XOR T[95] =
00000000101001111001011010010111 XOR 01100111001110111001100011011111 =
01100111100111000000111001001000
D[1] = 01100111100111000000111001001000

KUNCI PUTARAN 2
Sampai:

FungsiM(D[3],C[4]) =
FungsiM(01101101001100001110001101111110,10011011100111011001011010111011) =
(01101101001100001110001101111110 + 10011011100111011001011010111011)>>8 XOR
T[(01101101001100001110001101111110 + 10011011100111011001011010111011) AND
255(10)] = 00001000110011100111101000111001>>8 XOR T[57] =
00000000000010001100111001111010 XOR 10011010101001011100100001001101 =
10011010101011010000011000110111
D[4] = 10011010101011010000011000110111
KUNCI = D[4] = 10011010101011010000011000110111

3.3 Proses Enkripsi

Proses enkripsi dari metode *twofish* untuk menghasilkan ciphertext adalah berupa hasil operasi XOR dari plaintext dan 128 bit kunci yang dihasilkan dari proses pembentukan kunci.

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

e-ISSN : 2722-4368

Contoh :
PlainTeks = SAYA SEDANG MAKAN NASI
Key = STMIK BUDIDARMA
Maka adapun hasil proses enkripsinya adalah sebagai berikut:
Plain Text : 'SAYA SEDANG MAKAN NASI
Kode ascii dari 'S' = 01010011
Kode ascii dari 'A' = 01000001
Kode ascii dari 'Y' = 01011001
Kode ascii dari 'A' = 01000001
Kode ascii dari ' ' = 00100000

Kode ascii dari ' ' = 00100000
Kode ascii dari 'N' = 01001110
Kode ascii dari 'A' = 01000001
Kode ascii dari 'S' = 01010011
Kode ascii dari 'I' = 00100001

Plain Text (dalam biner) = 01000001 01011001 01010101 00100000 01010011 01000101 01000100 01000001 01001110 01000111 00100000 01001101 01000001 01001011 01000001 01001110 00100000 01000001 01011001 01000001 010011010 01000000

Kunci dari proses pembentukan kunci = 10011010101011010000011000110111
Cipher Text = Plain Text XOR Key
01010011 XOR 10011010 = 11001001 = 'É'
01000001 XOR 00110111 = 01110110 = 'v'
01011001 XOR 00000110 = 01011111 =’_’
01000001 XOR 00110111 = 01110110 = 'v'
00100000 XOR 00110111 = 00010111 = '_'
01010011 XOR 10011010 = 11001001 = 'É'
01000101 XOR 10101101 = 11101000 = 'è'
01000100 XOR 00000110 = 01000010 = 'B'
01000001 XOR 00110111 = 01110110 = 'v'
01001110 XOR 10011010 = 11010100 = 'Ô'
01000111 XOR 10101101 = 11101010 = 'ê'
00100000 XOR 00000110 = 00100110 = '&'
01001101 XOR 00110111 = 01111010 = 'z'
01000001 XOR 10011010 = 11011011 = 'Û'
01001011 XOR 10101101 = 11100110 = 'æ'
01000001 XOR 00000110 = 01000111 = 'G'
01001110 XOR 00110111 = 01111001 = 'y'
00100000 XOR 10011010 = 10111010 = 'º'
01001110 XOR 10011010 = 11010100 = 'Ô'
01000001 XOR 00110111 = 01110110 = 'v'
01011001 XOR 00000110 = 01011111 =’_’
01001011 XOR 10101101 = 11100110 = 'æ'
Hasil proses enkripsi = Évìv_ÉèBvÔê&zÛæGyº_ÔvÉz

3.4 Proses Dekripsi

e-ISSN : 2722-4368

Proses dekripsi dari metode *twofish* untuk menghasilkan *plaintext* adalah berupa hasil operasi XOR dari *ciphertext* dan 128 bit kunci yang dihasilkan dari proses pembentukan kunci.

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

Cipher Text : 'Évìv_ÉèBvÔê&zÛæGyº_ÔvÉz
Kode ascii dari 'É' = 11001001
Kode ascii dari 'v' = 01110110
Kode ascii dari 'ì' = 11101100
Kode ascii dari 'v' = 01110110
Kode ascii dari '_' = 00010111
Kode ascii dari 'É' = 11001001
Kode ascii dari 'è' = 11101000
Kode ascii dari 'B' = 01000010
Kode ascii dari 'v' = 01110110
Kode ascii dari 'Ô' = 11010100
Kode ascii dari 'ê' = 11101010
Kode ascii dari '&' = 00100110
Kode ascii dari 'z' = 01111010
Kode ascii dari 'Û' = 11011011
Kode ascii dari 'æ' = 11100110
Kode ascii dari 'G' = 01000111
Kode ascii dari 'y' = 01111001
Kode ascii dari 'º' = 10111010
Kode ascii dari 'Ô' = 11010100
Kode ascii dari 'y ' = 11001001
Kode ascii dari 'É' = 11001011
Kode ascii dari 'v' = 11001010
Cipher Text (dalam biner) =
11011011111101000101001100010111110010011110100001000010011101101101010011101010
00100110011110101101101111100110010001110111100110111010111011000101111101110110
110101111000110101

Kunci dari proses pembentukan kunci = 10011010101011010000011000110111

Plain Text = Cipher Text XOR Key
11001001 XOR 10011010 = 01010011 = 'S'
11011011 XOR 10011010 = 01000001 = 'A'
11110100 XOR 10101101 = 01011001 = 'Y'
01110110 XOR 00110111 = 01000001 = 'A'
00010111 XOR 00110111 = 00100000 = ' '
11001001 XOR 10011010 = 01010011 = 'S'
11101000 XOR 10101101 = 01000101 = 'E'
01000010 XOR 00000110 = 01000100 = 'D'
01110110 XOR 00110111 = 01000001 = 'A'
11010100 XOR 10011010 = 01001110 = 'N'
11101010 XOR 10101101 = 01000111 = 'G'
00100110 XOR 00000110 = 00100000 = ' '
01111010 XOR 00110111 = 01001101 = 'M'
11011011 XOR 10011010 = 01000001 = 'A'

e-ISSN : 2722-4368

11100110 XOR 10101101 = 01001011 = 'K'
01000111 XOR 00000110 = 01000001 = 'A'
01111001 XOR 00110111 = 01001110 = 'N'
10111010 XOR 10011010 = 00100000 = ' '
01111001 XOR 00110111 = 01001110 = 'N'
01000111 XOR 00000110 = 01000001 = 'A'
11001001 XOR 10011010 = 01010011 = 'S'
11001001 XOR 10011011 = 01010011 = 'I'
Hasil proses dekripsi = SAYA SEDANG MAKAN NASI

KESIMPULAN

Setelah melakukan perancangan dan implementasi maka penulis dapat memberikan beberapa kesimpulan sebagai berikut :

1. Proses penyandian data pada record menggunakan metode twofish agar data tersebut tidak diketahui oleh pihak yang tidak bertanggung jawab.
2. Proses penyandian record dengan menggunakan metode twofish dilakukan dengan cara pembentukan S-Box proses pembentukan kunci proses enkripsi dan proses deskripsi sehinga data atau recordakan tersandi.
3. Dalam penerapan aplikasi penyandian record menggunakan sofware Microsoft visual basic net 2008.

DAFTAR PUSTAKA

[1] R. Sadikin, *Kriptografi untuk keamanan jaringan*. Penerbit ANDI, 2012.
[2] D. Ariyus, "Kriptografi keamanan data dan komunikasi," *Yogyakarta Graha Ilmu*, 2006.
[3] R. Santi, "Implementasi Algoritma Enkripsi Playfair pada File Teks," *None*, vol. 15, no. 1, p. 246766, 2010.
[4] R. Munir, "Kriptografi," *Inform. Bandung*, 2006.
[5] A. Farisi, "Analisis Kinerja Algoritma Kriptografi Kandidat Advanced Encryption Standard (AES) pada Smartphone," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 4, no. 2, pp. 199–208, 2018, doi: 10.35957/jatisi.v4i2.103.
[6] D. R. Yunia, Adiwijawa, and Setyorini, "Analisis dan implementasi algoritma twofish pada penyandian citra digital," pp. 0–5, 2012.
[7] M. Qamal, "Kriptografi File Citra Menggunakan Algoritma Tea (Tiny Encryption Algorithm)," *TECHSI - J. Penelit. Tek. Inform.*, 2014.
[8] Z. Hercigonja, "Comparative Analysis of Cryptographic Algorithms," *Int. J. Digit. Technol. Econ.*, vol. 1, no. 2, pp. 127–134, 2016.
[9] A. M. Hasibuan, "Rancang Bangun Aplikasi Keamanan Data Menggunakan Metode AES Pada Smartphone," *MEANS (Media Inf. Anal. dan Sist.*, vol. 2, no. 1, pp. 29–35, Jun. 2017, doi: 10.17605/JMEANS.V2I1.20.
[10] D. A. Trianggana and H. Latipa Sari, "Analisis Perbandingan Kinerja Algoritma Blowfish Dan Algoritma Twofish Pada Proses Enkripsi Dan Dekripsi," *Pseudocode*, vol. 2, no. 1, pp. 37–44, 2015, doi: 10.33369/pseudocode.2.1.37-44.