# Studi Perbandingan Cipher Block Algoritma Blowfish dan Algoritma Twofish

Candra Alim Sutanto
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
*if18069@students.if.itb.ac.id*

***Abstract*—Dalam kriptografi, terdapat banyak algoritma yang bisa digunakan untuk mengenkripsi data. Algoritma kriptografi terbagi dalam algoritma kriptografi klasik dan modern. Beberapa contoh dari algoritma klasik adalah *caesar chiper,* sedangakan contoh dari algoritm modern adalah Algoritma Blowfish dan Twofish.**

**Makalah akan membahas perbandingan mengenai dua buah algortima kriptografi Blowfish dan Twofish. Kedua algoritma tersebut kedalam golongan algoritma kunci simetrik cipher blok. Cipher blok berarti data diolah dalam satuan blok, dan algoritma kunci simetrik berarti kunci yang dipakai untuk mengenkripsi data sama dengan kunci yang dipakai untuk mendekripsi.**

**Blowfish dan twofish keduanya dirancang oleh orang yang sama, yaitu Bruce Schneier. Algoritma Blowfish dan Twofish banyak digunakan dalam dunia kriptografi karena tidak dipatenkan dan memiliki lisensi yang bebas dan gratis.**

**Twofish dan blowfish keduanya memiliki keunggulan tersendiri dibandingkan rivalnya (DES dan AES). Dan diantara keduanya pun masing-masing memiliki keunggulan dan kelemahan masing-masing.**

**Perlu studi pembandingan kedua algoritma tersebut secara mendalam. Mulai dari teknik-teknik yang dipakai oleh keduanya, hingga performansi keduanya. Sehingga bisa diketahui mana algoritma memiliki performansi lebih baik dalam mengenkripsi data.**

***Kata Kunci*—Blowfish, Kriptografi, Perbandingan, Twofish.**

## I. PENDAHULUAN

Kriptografi, secara umum adalah ilmu dan seni untuk menjaga kerahasiaan berita [bruce Schneier - Applied Cryptography]. Selain pengertian tersebut terdapat pula pengertian ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integrasi data, serta autentifikasi [A. Menezes, P. van Oorschot and S. Vanstone - Handbook of Applied Cryptography]. Tidak semua aspek keamanan informasi ditangani oleh kriptografi.

Penggunaan komputer dijital mendorong berkembangnya kriptografi modern yang beroperasi dalam mode bit (satuan terkecil dalam dunia dijital) daripada kiptografi dalam mode karakter yang biasa digunakan dalam kriptografi klasik. Tetapi, kriptografi modern tetap saja menggunakan prinsip substitusi dan transposisi yang sudah digunakan sejak kriptografi klasik berkemban. Algoritma kriptografi modern sendiri dapat dibagi ke dalam kelompok algoritma simetri dan kunci publik. Algoritma simetri merupakan algoritma kriptografi yang beroperasi dengan kunci enkripsi dan dekripsi yang sama. Sedangkan pada algoritma kunci publik, kunci yang digunakan untuk proses enkripsi dan dekripsi berbeda. Algoritma simetri yang beroperasi dalam mode bit dapat dikelompokkan menjadi dua kategori, yaitu *cipher* aliran dan *cipher* blok. *Cipher* aliran merupakan algoritma kriptografi yang beroperasi dalam bentuk bit tunggal. Sedangkan algoritma kriptografi kategori *cipher* blok beroperasi dalam bentuk blok bit.

## II. CHIPER BLOK

Dalam proses enkripsi dan dekripsi algoritma kriptografi simetrik terdapat banyak modus untuk membantu proses tersebut secara keseluruhan dalam hal penyembunyian data/informasi. Hampir semua modus kecuali ECB juga berfungsi untuk mengacak blok masukan sebelum memasuki proses enkripsi dan dekripsi. Proses tersebut ikut membantu menyamarkan informasi yang ada sebagai bagian proses enkripsi dan dekripsi secara keseluruhan.

Proses enkripsi dengan kunci $K$ dinyatakan secara formal dengan persamaan

$$E_K(P) = C$$

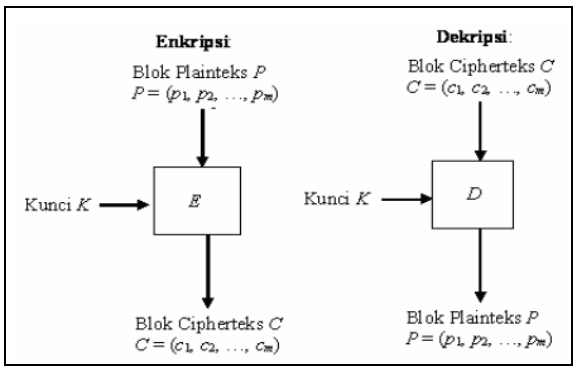
Sedangkan persamaan yang menyatakan proses dekripsi dengan kunci $K$ adalah

$$D_K(C) = P$$

Fungsi $E$ yang digunakan dalam proses enkripsi harus merupakan fungsi yang berkoresponden satu-ke-satu, sehingga

$$E^{-1} = D$$

Skema enkripsi dan dekripsi cipher blok dapat dilihat pada Gambar 1.



Gambar 1. Skema Block Chiper

Terdapat empat mode dalam kriptografi chiper blok

1. Electronic Code Book (ECB)

   Pada mode ECB, setiap blok plainteks $P_i$ dienkripsi secara individual dan independen menjadi blok cipherteks $C_i$. Secara matematis proses enkripsi dengan mode ECB dapat dinyatakan sebagai berikut:

   $$C_i = E_K(P_i)$$

   dan proses dekripsi sebagai berikut:

   $$P_i = D_K(C_i)$$

   Dalam hal ini, $P_i$ dan $C_i$ merupakan blok plainteks dan cipherteks ke-i.

2. Cipher Block Chaining (CBC)

   Pada mode CBC, proses enkripsi dibuat agar terjadi ketergantungan antar blok. Proses enkripsi setiap blok $P_i$ seperti halnya dalam mode ECB namun sebelumnya sudah dilakukan proses xor dengan blok chiper sebelumnya ($C_{i-1}$). Sedangkan blok pertama dilakukan proses xor dengan blok semu ($C_0$) yang disebut *IV* (*initialization vector*).

   Secara matematis proses enkripsi dapat dinyatakan sebagai berikut:

   $$C_i = E_K(P_i \oplus C_{i-1})$$

   sedangkan proses dekripsi dapat dinyatakan sebagai berikut:

   $$P_i = D_K(C_i) \oplus C_{i-1}$$

   Dalam hal ini $C_0$ merupakan IV (Initialization Vactor).

3. Cipher Feedback (CFB)

   Mode CFB mengenkripsikan data dalam unit yang lebih kecil daripada ukuran blok. Proses enkripsi pada unit yang lebih kecil daripada ukuran blok ini membuat mode CFB berlaku seperti *cipher* aliran. Unit yang dienkripsi dapat berupa bit per bit. Bila unit yang dienkripsi berupa satu karakter setiap kalinya, maka mode CFB ini disebut CFB 8-bit. Mode ini membutuhkan sebuah antrian yang berukuran sama dengan ukuran blok 3 masukan.

   Secara formal, proses enkripsi mode CFB n-bit dapat dinyatakan sebagai berikut:

   $$C_i = P_i \oplus MSB_m(E_K(X_i))$$
   $$X_{i+1} = LSB_{m-n}(X_i) \parallel C_i$$

   sedangkan proses dekripsi dapat dinyatakan sebagai berikut:

   $$P_i = C_i \oplus MSB_m(D_K(X_i))$$
   $$X_{i+1} = LSB_{m-n}(X_i) \parallel C_i$$

   Keterangan:
   $X_i$ = isi antrian dengan $X_1$ adalah IV
   E = fungsi enkripsi
   K = kunci
   m = panjang blok enkripsi
   n = panjang unit enkripsi
   $\parallel$ = operator penyambungan (concatenation)
   MSB = Most Significant Byte
   LSB = Least Significant Byte

4. Output Feedback (OFB)

   Mode OFB berkerja dengan cara yang mirip dengan mode CFB, kecuali n-bit dari hasil fungsi enkripsi terhadap antrian disalin menjadi elemen paling kanan antrian. Pada mode OFB tidak terdapat perambatan kesalahan. Kesalahan satu bit pada plainteks hanya mengakibatkan kesalahan satu bit yang berkoresponden pada cipherteks. Sebaliknya kesalahan satu bit pada cipherteks hanya mengakibatkan kesalahan satu bit yang berkoresponden pada plainteks.

Penggunaan mode-mode operasi tersebut tidak merubah fungsi enkripsi dan dekripsi yang telah didefinisikan.

## III. ALGORITMA BLOWFISH

Blowfish atau yang disebut juga "OpenPGP.Cipher.4″ adalah algoritma kunci simetrik cipher blok yang dirancang pada tahun 1993 oleh Bruce Schneider untuk menggantikan DES (Data Encryption Standard). Algoritma Blowfish dibuat untuk digunakan pada komputer yang mempunyai microposesor besar (32-bit keatas dengan cache data yang besar).

Pada saat itu banyak sekali rancangan algoritma yang ditawarkan, namun hampir semua terhalang oleh paten atau kerahasiaan pemerintah Amerika. Schneier menyatakan bahwa blowfish bebas paten dan akan berada pada domain publik. Dengan pernyataan Schneier tersebut blowfish telah mendapatkan tempat di dunia kriptografi, khususnya bagi masyarakat yang membutuhkan algoritma kriptografi yang cepat, kuat, dan tidak terhalang oleh lisensi.

Blowfish dirancang dan diharapkan mempunyai kriteria perancangan yang diiginkan sebagai berikut :

1. Cepat, Blowfish melakukan enkripsi data pada microprocessor 32-bit dengan rate 26 clock cycles per byte.
2. Compact, Blowfish dapat dijalankan pada memory kurang dari 5K.
3. Sederhana, Blowfish hanya menggunakan operasi – operasi sederhana, Blowfish hanya menggunakan

operasi – operasi sederhana, seperti : penambahan, XOR, dan lookup tabel pada operan32-bit.

4. Memiliki tingkat keamanan yang bervariasi, panjang kunci yang digunakan oleh Blowfish dapat bervariasi dan bisa sampai sepanjang minimal 32-bit, maksimal 448 -bit, Multiple 8 bit, default 128 bit

Namun, dalam penerapannya sering kali algortima ini menjadi tidak optimal. Karena strategi implementasi yang tidak tepat. Algoritma Blowfish akan lebih optimal jika digunakan untuk aplikasi yang tidak sering berganti kunci, seperti jaringan komunikasi atau enkripsi file otomatis.

## A. STRUKTUR ALGORITMA BLOWFISH

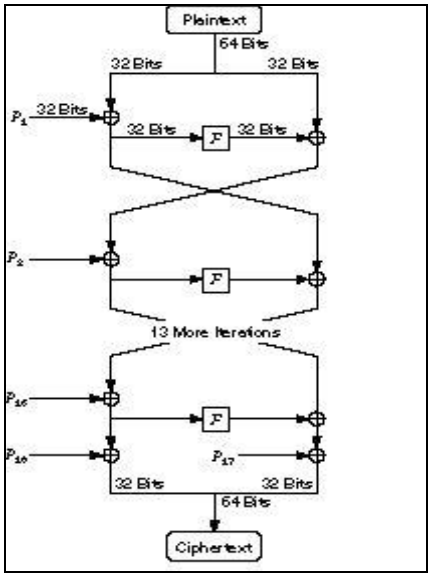
Blowfish merupakan blok cipher 64-bit dengan panjang kunci variabel. Algoritma ini terdiri dari dua bagian: key expansion atau perluasan kunci dan enkripsi data.

1. Key-Expansion
   Berfungsi merubah kunci (Minimum 32-bit, Maksimum 448-bit) menjadi beberapa array subkunci (subkey) dengan total 4168 byte.
2. Enkripsi Data
   Terdiri dari iterasi fungsi sederhana (Feistel Network) sebanyak 16 kali putaran. Setiap putaran terdiri dari permutasi kunci-dependent dan substitusi kunci- dan data-dependent. Semua operasi adalah penambahan (addition) dan XOR pada variabel 32-bit. Operasi tambahan lainnya hanyalah empat penelusuran tabel (table lookup) array berindeks untuk setiap putaran.

Untuk alur algoritma enkripsi dengan metoda Blowfish dijelaskan sebagai berikut :

1. Bentuk inisial array P sebanyak 18 buah (P1,P2,……..P1 masing-msing bernilai 32-bit. Array P terdiri dari delapan belas kunci 32-bit subkunci :
   P1,P2,…….,P18
2. Bentuk S-box sebanyak 4 buah masing-masing bernilai 32-bit yang memiliki masukan 256.
   Empat 32-bit S-box masing-masing mempunyai 256 entri :
   S1,0,S1,1,……..,S1,255
   S2,0,S2,1,……..,S2,255
   S3,0,S3,1,……..,S3,255
   S4,0,S4,1,……..,S4,255
3. Plainteks yang akan dienkripsi diasumsikan sebagai masukan, Plainteks tersebut diambil sebanyak 64-bit, dan apabila kurang dari 64-bit maka kita tambahkan bitnya, supaya dalam operasi nanti sesuai dengan datanya.
4. Hasil pengambilan tadi dibagi 2, 32-bit pertama disebut XL, 32-bit yang kedua disebut XR.
5. Selanjutnya lakukan operasi XL = XL xor Pi dan XR = F(XL) xor XR
6. Hasil dari operrasi diatas ditukar XL menjadi XR dan XR menjadi XL.
7. Lakukan sebanyak 16 kali, perulangan yang ke-16 lakukan lagi proses penukaran XL dan XR.
8. Pada proses ke-17 lakukan operasi untuk XR = XR xor P17 dan XL = XL xor P18.
9. Proses terakhir satukan kembali XL dan XR sehingga menjadi 64-bit kembali.

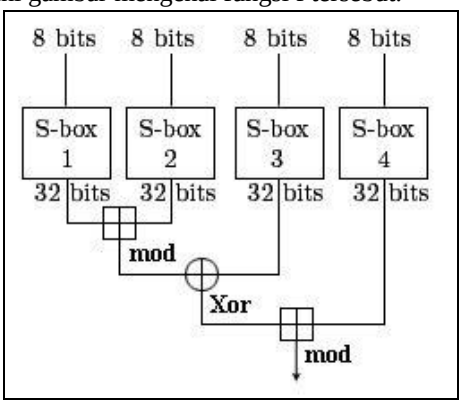
Blowfish menggunakan jaringan Feistel yang terdiri dari 16 buah putaran. Skema jaringan Feistel dapat dilihat di gambar .



Gambar 2. Jaringan Feistel untuk Algoritma Blowfish

Algoritma Blowfish memiliki keunikan dalam hal proses dekripsi, yaitu proses dekripsi dilakukan dengan urutan yang sama persis dengan proses enkripsi, hanya saja pada proses dekripsi P1, P2, …, P18 digunakan dalam urutan yang terbalik.

Dalam algoritma Blowfish juga terdapat fungsi f. Berikut ini gambar mengenai fungsi f tersebut.



Gambar 3. Fungsi f dalam algoritma Blowfish

Sebelumnya dijelaskan bahwa Array P terdiri dari delapan belas subkunci. Subkunci dihitung menggunakan algoritma Blowfish, metodenya adalah sebagai berikut :

1. Pertama-tama inilialisasi P-array dan kemudian empat S-box secara berurutan dengan string yang tetap. String ini terdiri atas digit hexadesimal dari Pi.

2. XOR P1 dengan 32-bit pertama kunci, XOR P2 dengan 32-bit kedua dari kunci dan seterusnya untuk setiap bit dari kunci (sampai P18).Ulangi terhadap bit kunci sampai seluruh P-array di XOR dengan bit kunci.
3. Enkrip semua string nol dengan algoritma Blowfish dengan menggunakan subkunci seperti dijelaskan pada langkah (1) dan (2).
4. Ganti P1 dan P2 dengan keluaran dari langkah (3).
5. Enkrip keluaran dari langkah (3) dengan algoritma Blowfish dengan subkunci yang sudah dimodifikasi.
6. Ganti P3 dan P4 dengan keluaran dari langkah (5).
7. Lanjutkan proses tersebut, ganti seluruh elemen dari P-array, kemudian seluruh keempat S-box berurutan, dengan keluaran yang berubah secara kontiyu dari algoritma Blowfish.

Secara keseluruhan terdapat 521 iterasi atau putaran yang dibutuhkan untuk membangkitkan seluruh upa-kunci yang dibutuhkan. Aplikasi kemudian dapat menyimpan upa-kunci yang telah dihasilkan. Proses pembangkitan kunci ini tidak perlu selalu dilakukan setiap saat.

### B. KEAMANAN ALGORITMA BLOWFISH

Sampai saat ini algoritma Blowfish belum ditemukan kelemahan yang berarti hanya adanya weak key dimana dua entri dari S-box mempunyai nilai yang sama. Belum ada cara untuk mengecek weak key sebelum melakukan key expansion, tetapi hal ini tidak berpengaruh terhadap hasil enkripsi.

Hasil enkripsi dengan algoritma Blowfish sangat tidak mungkin dan tidak praktis untuk di terjemahkan tanpa bantuan kunci. Sampai kini belum ada Cryptoanalyst yang dapat membongkar pesan tanpa kunci yang dienkripsi dengan memakai bantuan algoritma Blowfish. Agar aman dari pembongkaran pesan maka dalam algoritmanya harus menggunakan 16 putaran agar pesan tersebut tidak dapat dibongkar.

Algoritma Blowfish pun dapat digabungkan dengan algoritma-algoritma enkripsi yang lain dalam pengkripsian sebuah pesan untuk lebih menjamin isi dari pesan tersebut. Sehingga algoritma Blowfish cukup aman jika ingin digunakan untuk mengenkripsi data yang ingin di amankan.

### C. PENERAPAN ALGORITMA BLOWFISH

Blowfish adalah salah satu algoritma cipher blok yang tercepat dan digunakan secara luas di dunia, kecuali ketika pergantian kunci. Setiap kunci baru memerlukan pemrosesan awal yang sebanding dengan mengenkripsikan teks dengan ukuran sekitar 4 kilobyte. Pemrosesan awal ini sangat lambat dibandingkan dengan algoritma cipher blok lainnya. Hal ini menyebabkan Blowfish tidak mungkin digunakan dalam beberapa aplikasi, tetapi tidak menimbulkan masalah dalam banyak aplikasi lainnya.

Dalam beberapa implementasi, Blowfish memerlukan memori yang relatif besar, yaitu sekitar 4 kilobyte. Hal ini tidak menjadi masalah bahkan untuk komputer desktop dan laptop yang sudah berumur tua. Tetapi hal ini juga membuat implementasi Blowfish pada embedded system terkecil (seperti pada smartcard pada awal kemunculannya) tidak mungkin untuk dilakukan.

Hal-hal yang harus dipertimbangkan dalam perancangan perangkat lunak yang menerapkan algoritma Blowfish antara lain adalah prinsip perancangan algoritma Blowfish (sifat dasar algoritma Blowfish), yaitu:

1. Lebih optimal jika digunakan untuk aplikasi yang tidak sering berganti kunci.
2. Lebih cepat jika diterapkan pada 32-bit mikroprosessor dengan data cache yang besar. Namun, karena hal ini merupakan batasan teknis sehingga tidak akan dipakai dalam strategi perancangan.
3. Lebih aman jika diterapkan tanpa adanya pengurangan jumlah iterasi (dengan asumsi user tidak menggunakan weak key).

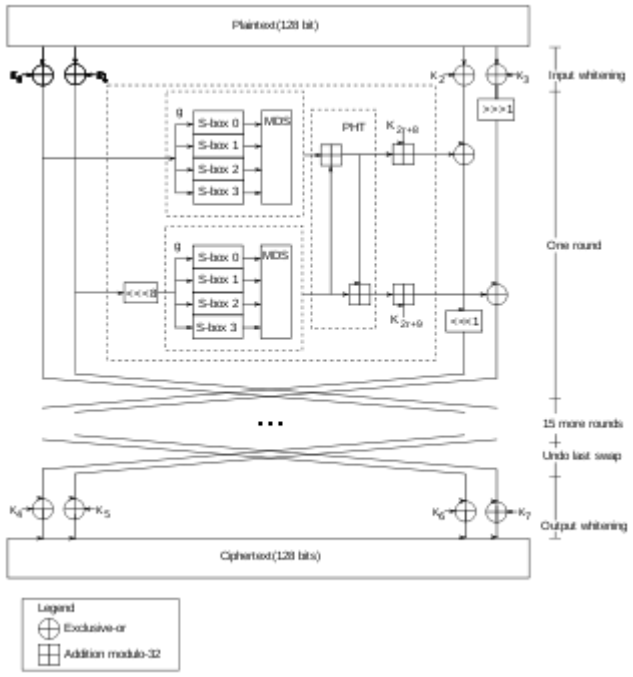
## IV. ALGORITMA TWOFISH

Twofish merupakan algoritma kriptografi yang beroperasi dalam mode blok cipher berukuran 128 bit dengan ukuran kunci sebesar 256 bit, ukuran kunci yang besar ditujukan untuk meniadakan kemungkinan kunci lemah (weak-key). Algoritma Twofish sendiri merupakan pengembangan dari algoritma Blowfish. Perancangan Twofish dilakukan dengan memperhatikan kriteria-kriteria yang diajukan National Institute of Standards and Technology (NIST) untuk kompetisi Advanced Encryption Standard (AES), namun algoritma ini tidak terpilih sebagai basis standarisasi.

### A. STRUKTUR ALGORITMA BLOWFISH

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

1. Feistel Network
2. S-boxes
3. MDS Matrices
4. Pseudo-Hadamard Transforms
5. Whitening
6. Key Schedule




Gambar 4. Struktur algoritma Twofish

Berikut ini penjelasan masing-masing prinsip yang terdapat dalam algoritma twofish

1. Fiestal Network

   Jaringan feistel asli menerapkan beberapa langkah yang sudah terstandardisasi. Langkah-langkah tersebut dapat berupa metoda-metoda lain, seperti pergiliran kunci, ekspansi dan filter, dll. Inti dari jaringan feistel adalah sebuah blok dibagi menjadi dua buah blok sama besar. Setengah blok kanan dikopikan ke setengah blok kiri blok hasil dan juga dimasukkan ke metoda ekspansi dan filter bersamaan dengan setengah blok kiri. Hasil dari fungsi ekspansi dan filter tersebut dikopikan ke setengah blok kanan blok hasil.

   Untuk mendekripsikan metoda ini cukup mudah, yaitu dengan cara mengkopikan setengah blok kiri result cipher ke setengah blok kanan blok hasil sambil dimasukkan ke fungsi ekspansi dan filter.

2. S-boxes

   S-Box adalah tabel substitusi untuk operasi algoritma yang yang tidak linear yang digunakan pada kebanyakan block cipher. S-Box memiliki berbagai variasi dalam ukuran input maupun ukuran output dan dapat digunakan baik secara acak maupun secara algoritmik.

   S-box dirancang agar tahan terhadap dua serangan besar awal 1990-an diferensial pembacaan sandi & pembacaan sandi linear dan tahan terhadap serangan apa pun yang tidak diketahui datang berikutnya.

   Dalam algoritma *Twofish s-box* yang digunakan adalah 8 x 8 bit *s-box*. *Twofish* memiliki 4 kunci yang berbeda. *S*-box yang digunakan pada algoritma *twofish* dibangun dengan melakukan 8x8 bit permutasi dengan kunci tertentu

3. MDS Matrices

   Kotak Most Distance Separable (MDS) adalah sebuah matriks yang dapat dioperasikan secara linear kepada sebuah vektor menjadi sebuah vektor hasil. Vektor hasil tersebut merupakan vektor yang terbesar yang dapat dihasilkan namun dengan jarak minimum dari vektor awal.

   Jika ada vektor dengan “a” elemen, dan ada vektor lain dengan “b” elemen, akan menghasilkan vektor baru dengan “a+b” elemen. Dengan mengandung jumlah minimum elemen bukan nol pada setiap bukan nol vektor adalah “b+1”. Sehingga “jarak” antara dua vektor yang berbeda yang dihasilkan dengan kotak MDS adalah sekurang-kurangnya “b+1”.

4. Pseudo-Hadamard Transforms

   Metoda asli pseudo-Hadamard yaitu dengan membagi blok menjadi dua buah subblok sama besar dan menerapkan operasi aritmatika sederhana.

   Enkripsi pseudo-Hadamard:

   a’ = a + b (mod 2n)

   b’ = a + 2b (mod 2n)

   Dekripsi pseudo-Hadamard:

   b = b’ - a’ (mod 2n)

   a = 2a’ - b’ (mod 2n)

   Persamaan diatas dapat juga dituliskan sebagai matriks aljabar, dengan memandang a dan b sebagai dua elemen sebuah vektor, dan transformasinya sendiri dipandang sebagai perkalian matriks dengan bentuk:

$$H_1 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

   Untuk proses dekripsi dapat dengan cara menginvers matriks tersebut diatas. Kelebihan lain adalah matriks tersebut dapat digeneralkan menjadi matriks dengan dimensi yang lebih tinggi untuk mentransformasikan vektor yang lebih dari dua variabel, dengan fungsi rekursif:

$$H_n = \begin{bmatrix} 2 \times H_{n-1} & H_{n-1} \\ H_{n-1} & H_{n-1} \end{bmatrix}$$

5. Whitening

   Tujuan dari metode ini adalah untuk mengacak urutan bit-bit pada sebuah blok. Metode ini berbeda dengan metode substitusi pada pemanipulasian bit.

   Perbedaannya adalah pada metode ini digunakan acuan yang telah pasti dalam pensubstitusian bit.

   Acuan tersebut tidak memiliki pola khusus, dan pada kebanyakan algoritma kriptografi, acuan tersebut telah ditetapkan oleh si perancang algoritma.

6. Key Schedule

   Key schedule adalah schedule yang mengatur

perubahan key bits menjadi round key yang dapat digunakan oleh cipher. Twofish membutuhkan banyak material key dan memiliki key schedule yang lengkap. Untuk mmfasilitasi analisis, key schedule menggunakan primitif yang sama seperti pada fungsi yang mengalami pengulangan.

## V. PERBANDINGAN BLOWFISH DAN TWOFISH

Terdapat beberapa perbedaan antara algoritma Blowfish dan algoritma twofish.

### A. PERBEDAAN SECARA UMUM

| Komponen | Blowfish | Twofish |
|---|---|---|
| Tahun Perancangan | 1993 | 1997 |
| Perancang | Bruce Scheneier | |
| Algoritma Pembanding | DES | AES |
| Panjang Blok | 64 bit | 128 bit |
| Panjang Kunci | 32 – 448 bit, kelipatan 8 bit | 128, 192, 256 bit |
| Transformasi Pseudo-Hadamard | Tidak | Ya |
| Ekspansi dan Filter | Tidak | Ya |
| Kotak MDS | Tidak | Ya |
| Jumlah putaran | 16 Putaran | |
| Lisensi | Tidak Berlisensi | |
| Biaya | Gratis | |

### B. KEAMANAN

Tingkat keamanan suatu algoritma kunci simetri tipe *cipher* blok dapat diukur dari tingkat kerumitan algoritma, panjang blok yang digunakan, panjang kunci yang digunakan, dan tingkat pengacakan plainteks terhadap cipherteks. Algoritma Blowfish dan Twofish menggunakan jaringan Feistel dan kotak-S dalam implementasinya. Karena itu tingkat keamanan algoritma ini juga dipengaruhi oleh cara penjadwalan kunci internal dan cara pembangkitan kotak-S.

| Komponen | Blowfish | Twofish |
|---|---|---|
| Algoritma | 16 putaran; menggunakan jaringan Feistel dan kotak-S; menggunakan operasi penambahan, XOR, dan *lookup table* | 16 putaran; menggunakan jaringan Feistel, kotak-S, matriks MDS, teknik Pseudo-Hadamard Transform (PHT), dan teknik *whitening*; menggunakan operasi penambahan, XOR, *lookup table*, aljabar vektor, aljabar polinom dan aljabar medan ;menggunakan fungsi dengan derajat aljabar yang tinggi (merupakan gabungan dari banyak kelompok aljabar) |
| Tingkat pengacakan | Rumit, karena menggunakan jaringan Feistel 16 putaran dan 4 kotak-S yang bergantung pada kunci. | Sangat rumit, karena menggunakan jaringan Feistel 16 putaran, 4 kotak-S yang bergantung pada kunci, matriks MDS, teknik PHT, teknik *whitening*. |
| Penjadwalan kunci internal | Penjadwalan kunci internal bergantung pada kunci eksternal; cara pembangkitan dengan menggunakan operasi penambahan dan XOR | Penjadwalan kunci internal bergantung pada kunci eksternal; cara pembangkitan kunci dengan menggunakan operasi penambahan, XOR, aljabar vektor, aljabar polinom, dan matriks MDS |
| Tingkat pengacakan | Rumit, karena menggunakan jaringan Feistel 16 putaran dan 4 kotak-S yang bergantung pada kunci. | Sangat rumit, karena menggunakan jaringan Feistel 16 putaran, 4 kotak-S yang bergantung pada kunci, matriks MDS, teknik PHT, teknik *whitening*. |

Berdasarkan tabel di atas, algoritma Twofish dirancang jauh lebih rumit dibandingkan dengan algoritma Blowfish. Algoritma Blowfish hanya lebih unggul dalam hal panjang kunci dibandingkan dengan algoritma Twofish. Dari perbandingan pada tabel di atas dapat ditarik kesimpulan bahwa rancangan Twofish jauh lebih aman dibandingkan dengan Blowfish.

### C. PENGGUNAAN

Blowfish dan twofish keduanya sangat terkenal di dunia kriptografi, hal ini ditandai dengan banyaknya aplikasi yang mengimplementasikan blowfish, twofish, maupun keduanya. Aplikasi yang paling menonjol adalah kernel Linux versi 2.5 keatas, dimana blowfish menjadi Open Cryptography Interface. Kelebihan lain yang ditawarkan oleh keduanya adalah bebas lisensi dan paten, sehingga semua aplikasi dapat memanfaatkan algoritma blowfish yang hebat tanpa perlu membeli paten apapun.

Dari segi kesulitan pengkodean, tentu twofish lebih sulit untuk dikode daripada blowfish, mengingat langkah-langkah twofish lebih banyak daripada blowfish. Namun untuk bahasa-bahasa standar, seperti C, C++, Java, dll. Telah ada beberapa versi yang tersedia secara open source maupun gratis yang dapat diterapkan pada program yang membutuhkan.

## VI. KESIMPULAN

Berdasarkan hasil studi dapat disimpulkan bahwa:

1. Algoritma Twofish jauh lebih kuat dibandingkan dengan Blowfish.
2. Sampai saat ini belum ditemukan metode kriptanalisis yang efisien untuk algoritma Twofish dan Blowfish yang tidak diperlemah.
3. Algoritma Blowfish berjalan lebih cepat dibandingkan dengan Twofish.
4. Algoritma Blowfish dapat digunakan untuk aplikasi yang membutuhkan waktu enkripsi dan dekripsi yang cepat serta tingkat keamanan data tidak terlalu penting.
5. Algoritma Twofish dapat digunakan untuk aplikasi yang membutuhkan tingkat keamanan data yang sangat penting serta waktu enkripsi dan dekripsi tidak terlalu dipersoalkan.

## REFERENCES


[1] Munir, Rinaldi. Bahan Kuliah IF5054 Kriptografi. (2004). Departemen Teknik Informatika, Institut Teknologi Bandung.
[2] http://id.wikipedia.org/wiki/blowfish
[3] http://id.wikipedia.org/wiki/kriptografi
[4] http://www.schneier.com/blowfish.html
[5] http://www.schneier.com/paper-blowfish-oneyear.html
[6] http://www.splashdata.com/splashid/blowfish.htm
[7] http://drdobbs.com/architecture-and-design/184410744
[8] http://www.schneier.com/twofish.html
[9] http://www.brighthub.com/computing/smb-security/articles/53270.aspx


## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 22 Maret 2011

Candra Alim Sutanto