ALGORITMA BLOWFISH UNTUK MENINGKATKAN KEAMANAN DATABASE MYSQL

Harlen Gilbert Simanullang[1], Arina Prima Silalahi[2]

[1,2]Fakutas Ilmu Komputer - Universitas Methodist Indonesia

[1]Jl.Hang Tuah No.8, Medan, Sumatera Utara

harlengilbert@gmail.com[1], primaarinasilalahi@gmail.com[2]

Abstract

The limitations of humans to maintain the security and confidentiality of stored data can have a bad impact if it is used by unauthorized parties. One way to secure shipping is to convert data into one that is not understood by encoding and insertion using cryptographic techniques. Cryptography is the study of system security, where the system needs to be secured from interference or threats from irresponsible parties. One of the cryptographic algorithms is the Blowfish Algorithm. The Blowfish algorithm is a modern cryptographic algorithm symmetrical key shaped block cipher. The Blowfish algorithm is quite simple but strong enough because it has a long key space, making it difficult to attack. Encryption is done by using a certain key, so as to produce ciphertext (files that have been encrypted or encoded) that cannot be read or understood. The ciphertext can be restored as if it was decrypted using the same key when encrypting the file. The key length used can affect the security of the algorithm. The decryption flow is almost the same as the Blowfish Algorithm encryption, except in the P-array (P1, P2, …… .., P18) done by invading. Information of a system is certainly stored in a database that is always possible to attack, but using cryptography then the problem can be overcome. By using the Blowfish algorithm, information security in a MySQL database can be improved.

Keywords: Cryptography, Blowfish Algorithm, Block Cipher, Ciphertext, Encryption, Decryption, MySQL

1. PENDAHULUAN

1.1 Latarbelakang

Kemudahan akses informasi membawa pengaruh terhadap keamanan informasi yang menggunakan media penyampaian. Seiring dengan perkembangan teknologi, maka semakin banyak keinginan pihak-pihak tertentu untuk mengambil ataupun merusak data/informasi secara sengaja. Informasi menjadi sangat rentan untuk diketahui, diambil atau bahkan dimanipulasi dan disalahgunakan. Pihak-pihak tersebut dengan tidak bertanggungjawab masuk ke sistem melalui akses tertentu dan berusaha mendapatkan informasi yang diinginkan. Keterbatasan pengguna/petugas sistem untuk terus memantau perkembangan sistem menjadi sebuah celah bagi pihak yang tidak berhak untuk menyalahgunakan informasi.

Selama pengiriman dan ketika sampai di tujuan, informasi tersebut harus tetap rahasia dan terjaga keasliannya atau tidak dimodifikasi. Penerima informasi harus yakin bahwa informasi tersebut memang benar berasal dari pengirim yang tepat, begitu juga sebaliknya, pengirim yakin bahwa penerima informasi adalah orang yang sesungguhnya. Selain itu penerima tidak ingin pengirim membantah pernah mengirim informasi tersebut, dan jika hal tersebut terjadi penerima perlu membuktikan ketidakbenaran penyangkalan tersebut. Untuk permasalahan-permasalahan keamanan tersebut diperlukan suatu metode untuk menjaga keamanan informasi. Salah satu metodenya adalah kriptografi [7].

Kriptografi merahasiakan informasi dengan menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Saat ini banyak bermunculan algoritma kriptografi yang terus dianalisis, dicoba dan disempurnakan untuk mencari algoritma yang dianggap memenuhi standar keamanan. Blowfish merupakan salah satu algoritma yang tidak dipatenkan dan cukup kuat karena memiliki ruang kunci yang besar dan panjangnya bisa beragam, sehingga tidak mudah diserang pada bagian kuncinya. Suatu sistem kriptografi yang baik terletak pada kerahasiaan kunci dan bukan pada kerahasiaan algoritma yang digunakan. Blowfish pada strategi implementasi yang tepat akan lebih optimal, dapat berjalan pada memori kurang dari 5KB dan kesederhanaan pada algoritmanya. Algoritma Blowfish akan lebih optimal jika digunakan untuk aplikasi yang tidak sering berganti kunci, seperti jaringan komunikasi atau enkripsi file otomatis. Selain itu, karena algoritma ini membutuhkan memori yang besar, maka algoritma ini tidak dapat diterapkan untuk aplikasi yang memiliki memori kecil seperti smart card. Panjang kunci yang digunakan, juga mempengaruhi keamanan penerapan algoritma ini [8]. Dengan menggunakan algoritma Blowfish, keamanan informasi dalam Database MySQL dapat ditingkatkan.

1.2 Rumusan Masalah

Sesuai dengan latarbelakang yang telah dijelaskan, maka rumusan masalah dalam penelitian ini adalah :

1. Kriptografi menjadi peran penting dalam membantu keamanan informasi.
2. Penerapan Algoritma Blowfish dalam sebuah sistem dengan database MySQL.
3. Enkripsi dan Dekripsi dengan menggunakan Algoritma Blowfish.

1.3 Batasan Masalah

Untuk mencegah melebarnya masalah yang akan diteliti, maka batasan masalah dalam penelitian ini adalah :

1. Keamanan informasi dilakukan untuk database MySQL, tidak mencakup Oracle, Acces ataupun SQL Server.
2. Kriptografi simetris yang digunakan hanya Algortima Blowfish.

3. Algoritma Blowfish pada penelitian berikut dapat diterapkan dalam sistem apapun yang memiliki banyak informasi penting.

1.4 Tujuan dan Manfaat

1.4.1Tujuan Penelitian

Adapun tujuan penelitian ini adalah menerapkan algoritma Blowfish sebagai metode untuk meningkatkan keamanan data dalam database MySQL.

1.4.2Manfaat Penelitian

Sedangkan manfaat penelitian adalah memberikan privasi pesan yang menjamin kerahasiaan, integritas dan tidak ada pengulangan data dalam bertukar informasi.

2. LANDASAN TEORI

2.1 Kriptografi

Kriptografi berasal dari dua kata bahasa Yunani, *Cryptos* dan *Graphein. Cryptos* 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 ke tempat lain [7].

Kriptografi adalah ilmu dan seni dalam mengamankan pesan dengan cara mengubah pesan menjadi sesuatu yang tidak dapat dimerngerti oleh orang lain dengan teknik-teknik dan metode-metode tertentu. Kriptografi tidak berarti hanya memberikan keamanan informasi saja, namun lebih ke arah teknik-tekniknya.

Algoritma kriptografi yang baik tidak ditentukan oleh kerumitan dalam mengolah data atau pesan yang akan disampaikan. Ada empat syarat yang perlu dipenuhi dalam ilmu kriptografi (sebagai aspek-aspek keamanan), yaitu [4] :

1. Kerahasiaan (*confidentiality*), adalah menyediakan privasi untuk pesan dan menyimpan data dengan menyembunyikannya.
2. Integritas data (*data integrity*), 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, yang tidak dimodifikasi ketika sedang dalam proses transmisi data.
3. Autentikasi (*authentication*), beruhubunagn dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user/entity authentification*)maupun mengidentifikasi kebenaran sumber pesan (*data origin authentification*),agar penerima informasi dapat memastikan keaslian pesan, bahwa pesan itu datang dari orang yang dimintai informasi.
4. Non-repudiasi (*non-repudiation*), yang berarti dapat membuktikan bahwa dokumen memang benar datang dari orang yang dimintai informasi.

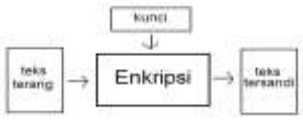
Algoritma kriptografi adalah bagian dari kriptografi yang berisi kumpulan langkah-langkah logis yang digunakan untuk melakukan enkripsi dan dekripsi. Biasanya langkah-langkah ini berupa sekumpulan fungsi matematik. Berdasarkan kuncinya, algoritma kriptografi dibedakan menjadi dua, yaitu algoritma simetris (kunci privat) dan algoritma asimetris (kunci publik). Secara umum kriptografi terdiri dari dua proses yaitu enkripsi dan dekripsi. *Plaintext* adalah pesan yang akan dirahasiakan, dinotasikan dengan m (Message), yang dapat berupa *bit tream, file text, digitized voice stream, digital video image* atau lebih singkatnya m adalah data *binary*.

Enkripsi adalah proses pengaman data atau informasi dengan membuat informasi tersebut seolah tidak bermakna atau tidak dapat dibaca. enkripsi dinotasikan dengan E, berfungsi untuk mengubah m menjadi c, dalam matematika dinotasikan dengan: E(m) = c.

Keterangan rumus:

E = enkripsi
m = plaintext
c = ciphertext

Skema Proses enkripsi dapat dilihat pada Gambar 1 berikut :



Gambar 1 Skema Proses Enkripsi
(Sumber : Munir, 2006)

Ciphertext adalah hasil dari proses enkripsi, dinotasikan dengan c, juga berupa *data binary* yang kadang-kadang mempunyai ukuran yang sama dengan m, lebih kecil dari m atau lebih besar dari m.

Dekripsi adalah kebalikan dari enkripsi yaitu proses mengubah data menjadi bermakna. Fungsi dekripsi D, berfungsi untuk mengubah c menjadi m, dalam matematika dinotasikan dengan: D(c) = m.

Keterangan rumus:

D = dekripsi
c = ciphertext
m = plaintext

Crypanalyst adalah orang mempelajari ilmu dan seni ilmu membongkar ciphertext. Menurut *ISO 7498-2* istilah yang lebih tepat untuk decryption adalah *decipher*.

Cipher adalah kata lain dari algoritma yang digunakan untuk melakukan melakukan proses kriptografi. Cipher juga sering disebut teknik yang digunakan untuk proses enkripsi dan deskripsi. Secara umum berdasarkan kesamaan kuncinya, algoritma kriptografi dibedakan menjadi :

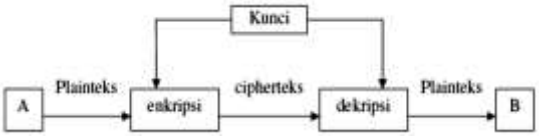
1. Kunci-simetris / *symetric-key*, kunci yang digunakan untuk melakukan enkripsi sama dengan kunci yang digunakan untuk dekripsi
2. Kunci-asimetris / *asymetric-key* , kunci enkripsi dan kunci dekripsi tidak sama.

Berdasarkan kerahasiaan kuncinya, algoritma kriptografi dibedakan menjadi :

1. Algoritma kriptografi kunci rahasia *secret-key*
2. Algoritma kriptografi kunci publik *publik-key*

2.2 Algoritma Simetris

Algoritma simetris adalah algoritma kriptografi yang menggunakan kunci enkripsi yang sama dengan kunci dekripsinya. Algoritma ini mengharuskan pengirim dan penerima menyetujui suatu kunci tertentu sebelum mereka saling berkomunikasi. Keamanan algoritma simetris tergantung pada kunci, membocorkan kunci berarti bahwa orang lain dapat mengenkripsi dan mendekripsi pesan. Agar komunikasi tetap aman, kunci harus tetap dirahasiakan. Algoritma simetris sering juga disebut dengan algoritma kunci rahasia, algoritma kunci tunggal, atau algoritma satu kunci.



Gambar 2. Skema algoritma simetris
(Sumber : Munir, 2006)

2.3 Algoritma Blowfish

Blowfish diciptakan oleh seorang Cryptanalyst bernama Bruce Schneier, Presiden perusahaan Counterpane Internet Security, Inc (Perusahaan konsultan tentang kriptografi dan keamanan komputer) dan dipublikasikan tahun 1994. Dibuat untuk digunakan pada komputer yang mempunyai microposesor besar (32-bit keatas dengan cache data yang besar). Blowfish merupakan algoritma yang tidak dipatenkan dan license free, dan tersedia secara gratis untuk berbagai macam kegunaan.

Pada saat Blowfish dirancang, diharapkan mempunyai kriteria perancangan sebagai berikut:

1. Cepat, Blowfish melakukan enkripsi data pada microprocessors 32-bit dengan rate 26 clock cycles per byte.
2. Compact (ringan), Blowfish dapat dijalankan pada memori kurang dari 5K.
3. Sederhana, Blowfish hanya menggunakan operasi-operasi sederhana: penambahan, XOR, dan lookup tabel pada operan 32-bit.

Memiliki tingkat keamanan yang bervariasi, panjang kunci yang digunakan oleh Blowfish dapat bervariasi dan bisa sampai sepanjang 448 bit.

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. Selain itu, karena algoritma ini membutuhkan memori yang besar, maka algoritma ini tidak dapat diterapkan untuk aplikasi yang memiliki memori kecil seperti smart card. Panjang kunci yang digunakan, juga mempengaruhi keamanan penerapan algoritma ini. Blowfish merupakan blok cipher 64-bit dengan panjang kunci variabel[2].

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.

2.3.1 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 [9].

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.

2.3.2 Penggunaan 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 denga 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.

Pemrosesan awal yang lama pada Blowfish digunakan sebagai ide untuk metode *password-hashing* yang digunakan pada OpenBSD. Metode *password-hashing* ini menggunakan algoritma yang diuturnnkan dari algoritma Blowfish yang menggunakan penjadwalan kunci yang lambat. Algoritma ini digunakan dengan pertimbangan bahwa usaha komputasi ekstra yang harus dilakukan dapat memberikan proteksi lebih terhadap serangan terhadap password berbasiskan kamus (*dictionary attacks*).

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[8].

2.4 MySQL

MySQL adalah sistem manajemen *database* SQL yang bersifat *Open Source* dan paling populer saat ini. Konsekuensi dari *open source*, perangkat lunak ini dapat dipakai oleh siapa saja tanpa membayar dan *source code*-nya bisa diunduh oleh siapa saja. Sistem *database* MySQL mendukung beberapa fitur seperti *multithreaded*, *multi-user*, dan *SQL database* managemen sistem (DBMS). *Database* ini dibuat untuk keperluan sistem *database* yang cepat, handal dan mudah digunakan[5]. Berikut ini beberapa kelebihan MySQL sebagai *database server* antara lain :

1. *Source* MySQL dapat diperoleh dengan mudah dan gratis.Sintaksnya lebih mudah dipahami dan tidak rumit.
2. Pengaksesan *database* dapat dilakukan dengan mudah.
3. MySQL merupakan program yang *multithreaded*, sehingga dapat dipasang pada *server* yang memiliki multi CPU.
4. Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python.
5. Bekerja pada berbagai *platform*. (Tersedia berbagai versi untuk berbagai sistem operasi).
6. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem *database*.
7. Memiliki sistem keamanan yang cukup baik dengan verifikasi *host*.

8. Mendukung ODBC untuk sistem operasi *Windows*.
9. Mendukung *record* yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

MySQL dan PHP merupakan sistem yang saling terintegrasi. Maksudnya adalah pembuatan *database* dengan menggunakan sintak PHP dapat di buat. Sedangkan *input* yang di masukkan melalui aplikasi *web* yang menggunakan *script serverside* seperti PHP dapat langsung dimasukkan ke *database* MySQL yang ada di *server* dan tentunya *web* tersebut berada di sebuah *web server*. MySQL ini bermanfaat untuk mengelola data dengan cara yang sangat fleksibel dan cepat[6].

3. HASIL DAN PEMBAHASAN

3.1 Alur Enkripsi Algoritma Blowfish

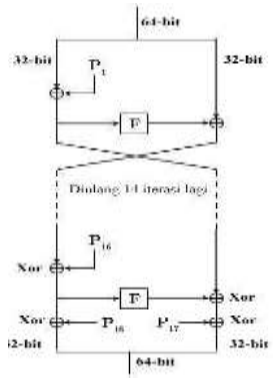
Untuk alur algoritma enkripsi dengan metode Blowfish dijelaskan sebagai berikut [2] :

1. Bentuk inisial array P sebanyak 18 buah (P1,P2,……..P1 masing-masing 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.

Alur *pseudocode* enkripsi dengan Blowfish adalah sebagai berikut [3] :

1. Bagi X menjadi dua 32-bit: XL, XR untuk i = 1 sampai 16
2. XL = XL xor Pi
3. XR = F(XL) xor XR
4. Tukar XL dan XR
5. Tukar XL dan XR (batalkan penukaran terakhir)
6. XR = XR xor P17
7. XL = XL xor P18
8. Kombinasikan kembali XL dan XR
9. Fungsi F adalah sebagai berikut:
10. Bagi XL, menjadi empat bagian 8-bit: a, b, c dan d
11. F(XL) = ((S1,a + S2,b mod 232) xor S3,c) + S4,c mod 232

Berikut adalah enkripsi algoritma Blowfish dapat dilihat pada Gambar 3.



Gambar 3. Enkripsi Algoritma Blowfish

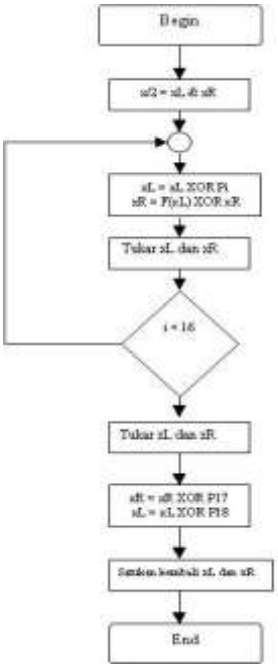
3.2 Alur Dekripsi Algoritma Blowfish

Untuk deskripsi sama persis dengan enkripsi, kecuali pada P-array (P1,P2,……..,P18) digunakan dengan urutan terbalik atau di inverskan [10].

Dekripsi sama persis dengan enkripsi, kecuali bahwa P1, P2,…P18 digunakan pada urutan yang berbalik (reverse). Algoritmanya dapat dinyatakan sebagai berikut:

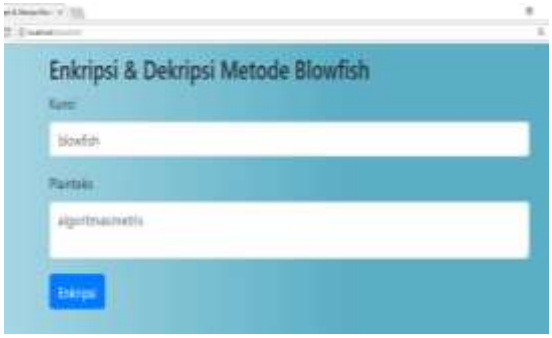
for i = 1 to 16 do
XRi = XLi-1 ⊕ P19-i
XLi = F[XRi] ⊕ XRi-1;
XL17 = XR16 ⊕ P1;
XR17 = XL16 ⊕ P2;

Flowchat pada Algoritma Blowfish dapat dilihat pada Gambar 4 berikut.



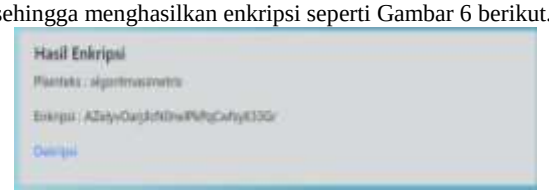
Gambar 4. Flowchart pada Algoritma Blowfish

Penerapan alur enkripsi Algoritma Blowfish dapat diterapkan dengan cara berikut: kunci yang diberikan adalah “blowfish” dan Plainteks adalah “algoritmasimetris” dapat dilihat pada Gambar 5.



Gambar 5. Kunci dan Plainteks Algoritma Blowfish

Kunci dan plainteks yang diberikan akan diproses untuk dienkripsi menggunakan Algoritma Blowfish sehingga menghasilkan enkripsi seperti Gambar 6 berikut.



Gambar 6. Hasil Enkripsi

Alur untuk dekripsi Algoritma Blowfish menerapkan urutan terbalik atau invers pada array P sebanyak 18 buah yang masing-masing bernilai 32-bit. Hasil Dekripsi Algoritma Blowfish dapat dilihat pada Gambar 7.

Gambar 7. Hasil Dekripsi

4. KESIMPULAN

Dari hasil uraian diatas, maka kesimpulan yang diperoleh adalah sebagai berikut :

1. Penerapan algoritma blowfish dapat dilakukan dengan membuat tampilan aplikasi untuk memperlihatkan hasil dari enkripsi dan dekripsi.
2. Dengan menginputkan kunci dan plainteks, maka akan diproses sehingga menghasilkan enksipsi.
3. Proses Enkripsi dari Algoritma Blowfish memperlihatkan Ciphertext yang kemudian digunakan untuk menghasilkan Dekripsi Algoritma Blowfish.
4. Algoritma blowfish merupakan algoritma simetris, yaitu menggunakan kunci enkripsi yang sama dengan kunci dekripsinya.
5. Algoritma Blowfish pun dapat digabungkan dengan algoritma-algoritma enkripsi yang lain dalam pengkripsian sebuah pesan untuk lebih menjamin isi dari pesan, namun dalam hal ini hanya menggunakan algoritma tersebut untuk enkripsi dan dekripsi.

5. SARAN

Berdasarkan penelitian yang telah dilakukan, saran yang diberikan adalah :

1. Dengan mengkombinasikan enkripsi Algoritma Blowfish dengan dekripsi dengan Algoritma simetris yang lain.
2. Memberikan keamanan pada Database lain seperti SQL Server ataupun Oracle.

DAFTAR PUSTAKA

[1] Ariyus, Dony. 2007. *Intrusion Detection System.* Yogyakarta : Penerbit Andi

[2] Christina L, Joe Irudayaraj V S,Optimized Blowfish Encryption Technique, International Journal of Innovative Research in Computerand Communication Engineering

[3] E.A.Shanty, *Implementasi Algoritma Kriptografi Blowfish Untuk Keamanan Dokumen Pada Microsoft Office*, J. Chem. Inf. Model., vol. 53, no. 9, pp. 1689–1699, 2013.

[4] Kadir,Abdul. 2009. *Mudah mempelajari database MySQL*. Andi, Yogyakarta

[5] Munir, R., 2006. *Kriptografi*, Bandung: Informatika

[6] Pratama, I Putu. 2014. Sistem Informasi dan Implementasinya. Bandung: Informatika

[7] Sadikin, Rifki. 2012. Kriptografi untuk Jaringan. Yogyakarta: Andi Publisher

[8] Sitinjak, Suriski. Fauziah, Yuli. Juwairiah. 2010. Aplikasi Kriptografi File Menggunakan Algoritma Blowfish. Jurnal SemnasIf:78-86.

[9] Sutanto, Candra. 2009. Penggunaan Algoritma Blowfish dalam Kriptografi. Institut Teknologi Bandung, ISSN: 2320-9801 , Vol. 2, Issue 7, July 2014.

[10] Tanjyot Aurora, Parul Arora, "Blowfish Algorithm", IJCSCE, ISSN 2319-7080,NCRAET , 2013.