OPTIMASI METODE BLOWFISH UNTUK MENGAMANKAN PASSWORD PADA KRIPTOGRAFI

Yani Parti Astuti[1], Christy Atika Sari[2], Eko Hari Rachmawanto[3]
Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Dian Nuswantoro Semarang
Jl Imam Bonjol No 207 Semarang
Telp. (024) 3560567
E-mail:sekretariat@dinus.ac.id

ABSTRAK

Berkembangnya ilmu komputerisasi, isu mengenai keamanan data baik dalam bentuk file maupun password menjadi sangat penting. Untuk menjaga keamanan pada data tersebut, maka dikenal salah satu teknik yaitu kriptografi. Adapun algoritma yang digunakan adalah Algoritma *Blowfish*. *Blowfish* merupakan algoritma yang aman, sederhana, kuat dan cepat dengan menggunakan 16 putaran, dengan blok cipher 64-bit yang merupakan panjang dari kunci variabel. Keunggulan blowfish yaitu mempunyai dua bagian yaitu perluasan kunci atau *key expansion* dan enkripsi data. *Blowfish* disebut juga dengan algoritma kunci simetrik *cipher* blok yang dikenal dengan istilah "OpenPGP.Cipher.4". Pesan tanpa kunci yang dienkripsi oleh *blowfish* belum bisa dibongkar oleh kriptanalis sampai sekarang. Hasil dari implementasi b*lowfish* ini menunjukkan bahwa aplikasi dapat menjalankan fungsi pada proses enkripsi dan dekripsi data dengan baik dalam mengamankan password dalam ranah kunci heksa string dan alfabet.

Kata kunci: kriptografi, *blowfis*, *key expansion.*

1. PENDAHULUAN

Data yang disimpan secara digital menggunakan media internet memiliki berbagai macam resiko, yang mana pihak lain yang tidak berkepentingan bisa mengambil data dan menggunakannya untuk kepentingan sendiri. Sebagai contoh apabila seseorang mengetahui password orang lain maka mereka dapat memanipulasi informasi yang diketahuinya dan bias merugikan pemilik password tersebut. Berdasarkan contoh kasus tersebut maka diperlukan teknik pengamanan data yang bisa digunakan oleh pemilik password agar tidak diketahui oleh orang lain, sehingga password yang dimiliki aman.

Dalam pengamanan data dikenal beberapa teknik, salah satunya adalah kriptografi. Kriptografi berfungsi mengamankan data dengan teknik menyisipkan data melalui suatu algoritma tertentu agar tidak diketahui orang lain(Munir et al. n.d.). teknik kriptografi tidak hanya digunakan untuk menyandikan data, namun bisa juga digunakan untuk menjaga integritas, autentikasi serta non repudiasi(Abd Elminaam et al. 2010). Hasil dari kriptografi hanya bisa diketahui oleh pemilik dan orang yang berkepentingan .

Dalam teknik kriptografi terdapat bebrapa algoritma yang digunakan, antara lain Blowfish, *Most Significant Bit* (MSB), *End Of File* (EOF), *Advanced Encryption Standard* (AES) dan *Data Encryption Standard* (DES). Blowfish adalah salah satu algoritma kriptografi yang terkenal, cepat, sederhana serta tahan terhadap serangan dan manipulasi data (Mandal 2012a). *Blowfish* melakukan teknik pemutaran ulang dan juga pergantian kunci sebanyak 16 kali putaran yang disebut dengan teknik manipulasi bit. Algoritma utama pada *blowfish* dibagi

menjadi dua sub algoritma utama, yaitu ekspansi kunci dan bagian enkripsi-dekripsi data, sehingga *blowfish* lebih aman.

Penelitian yang sudah dilakukan dalam menganalisa kinerja algoritma blowfish adalah Ema Utami yang penelitiannya adalah menganalisa performa *blowfish* menggunakan simulasi data yang terbatas(Utami & Tambunan 2010). Penelitian berikutnya dilakukan oleh Mandal yang membandingkan bahwa algoritma blowfish lebih handal dibanding dengan algoritma DES, AES dan 3DES. Kehandalannya dibuktikan dengan proses enkripsi maupun dekripsi dari algoritma *blowfish* lebih cepat(Mandal 2012a). Sitinjak dan Fauziah juga melakukan penelitian menggunakan algoritma *blowfish* untuk mengamankan file txt, pdf dan juga rtf. Sehingga blowfish juga mampu dalam pengamanan file(Sitinjak & Fauziah 2010).

Melalui penelitian – penelitian di atas, maka pengamanan data menggunakan *blowfish* sangat penting dan sudah terbukti unggul dalam teknik kriptografi. Sehingga pada penelitian ini akan dibuat sebuah pengembangan aplikasi pengamanan password dengan teknik kriptografi berupa algoritma *blowfish* menggunakan bahasa pemrograman *Visual Basic* versi 6.0.

Pada penelitian ini diberikan aplikasi *blowfish* yang semakin luas yaitu bisa memasukkan kunci dengan heksadesimal maupun alphabet dengan panjang karakter tak terbatas. Sehingga akan memudahkan bagi pengguna aplikasi *blowfish* dalam pengamanan password yang diinginkan.

1.1 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah bagaimana membuat aplikasi untuk mengamankan password menggunakan algoritma blowfish melalui Bahasa Pemrograman *Visual Basic* 6.0.

1.2 Tujuan Penelitian

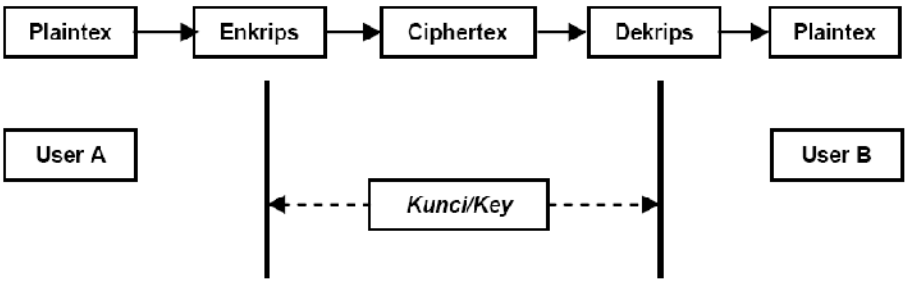
Tujuan dilakukannya penelitian yaitu mengevaluasi keberhasilan algoritma blowfish dalam mengamankan data dalam aplikasi pengamanan password menggunakan bahasa pemrograman *Visual Basic* 6.0.

2. TINJAUAN PUSTAKA

2.1 Kriptografi

Kriptografi adalah ilmu yang mempelajari tentang penyembunyian data ke dalam suatu bentuk tertentu agar tidak diketahui oleh orang lain(Munir 2012). Sedang algoritma kriptografi merupakan algoritma yang digunakan untuk menyelesaikan tujuan dari ilmu kriptografi (Munir 2011; Munir 2012). Dalam teknik kriptografi dikenal adanya proses enkripsi dan dekripsi. Enkripsi merupakan proses penyembunyian data yang dimengerti oleh orang lain, sedang dekripsi adalah kebalikan dari proses enkripsi yang hasilnya merupakan data utama dan pesan yang tersembunyi. Proses dekripsi dikatakan berhasil apabila data utama dan pesan yang disembunyikan hasilnya sama dengan kondisi awal(Munir et al. n.d.). Proses enkripsi dan dekripsi tersebut bisa dilihat sebagai berikut.

ISBN: 978-979-3649-96-2 Unisbank Semarang, 28 Juli 2016



Gambar 1 : Proses Enkripsi dan Dekripsi dalam Kriptografi(Utami & Tambunan 2010)

Kriptografi pada era digitalisasi seperti saat ini digunakan dalam berbagai bidang misalnya militer, kesehatan dan bisnis(Munir 2005). Kriptografi digunakan pula untuk mengamankan transaksi ATM, transaksi dengan kartu kredi, percakapan telpon maupun pengaktifan peluru kendali. Teknik ini dimaksudkan untuk menghindari penyadapan dan modifikasi data oleh pihak yang tidak berkepentingan.

2.2 Algoritma Blowfish

Pada tahun 1993, Bruce Schneier menyatakan bahwa algoritma *blowfish* adalah algoritma kunci simetrik cipher blok yang sudah dirancang sebagai pengganti DES. Waktu itu banyak sekali tawaran tentang rancangan algoritma, tetapi hak paten atau kerahasiaan pemerintah Amerika telah menghalangi. Untuk itu Schneier membuat pernyataan bahwa algoritma *blowfish* bebas hak paten dan juga berada dalam domain publik. Dengan pernyataan Schneier tersebut membuat *blowfish* mendapatk tempat baik dalam dunia kriptografi, khususnya bagi masyarakat yang memerlukan algoritma kriptografi yang kuat, sederhana, cepat, dan tidak terhalang pada lisensi (Andriyanto & C. Pardede n.d.).

Algoritma blowfish memiliki kriteria perancangan sebagai berikut(Utami & Tambunan 2010; Mandal 2012a; Andriyanto & C. Pardede n.d.):

a. *Blowfish* melakukan enkripsi data dengan cepat melalui microprocessor sebasar 32-bit dengan rate 26 *clock cycles* per byte.
b. *Blowfish* dapat dijalankan menggunakan memory kurang dari 5K. Sehingga dikatakan bahwa algoritma *blowfish* compact
c. *Blowfish* dikatakan juga sederhana karena hanya memerlukan operasi – operasi yang sederhana, yaitu : penambahan, XOR, dan juga lookup tabel menggunakan operan 32-bit.
d. *Blowfish* memiliki tingkat keamanan dan menggunakan panjang kunci yang variasi dari minimal 32 bit sampai batas maksimal 448 bit dengan multiple 8 bit serta default 128 bit.

Blowfish juga merupakan blok cipher 64-bit dengan panjang kunci variabel. Algoritma ini terdiri atas dua bagian yaitu key expansion atau disebut perluasan kunci dan suatu enkripsi data (Thakur & Kumar 2011; Octamanullah 1997; Sitinjak & Fauziah 2010).

ISBN: 978-979-3649-96-2 Unisbank Semarang, 28 Juli 2016

2.3 Enkripsi Data dan dekripsi

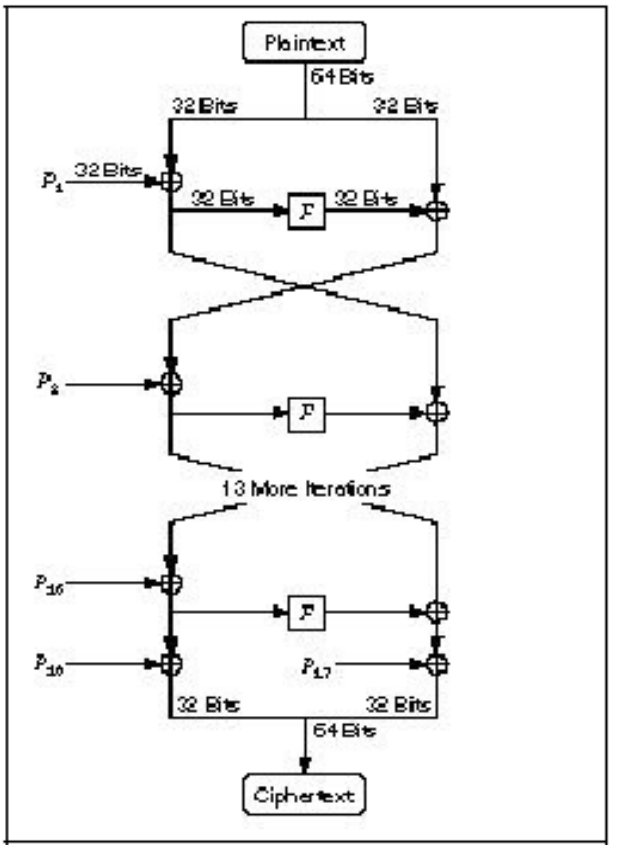
Pada algoritma *blowfish* dikenal adanya proses enkripsi yang terdiri dari iterasi fungsi yang sederhana yang disebut dengan *Feistel Network* sebanyak 16 kali putaran. Setiap perputaran yang dilakukan terdapat permutasi kunci dependen dan substitusi kunci serta data dependen. Operasi yang digunakan sangat sederhana yaitu penambahan (*addition*) dan juga XOR dengan menggunakan variable 32 bit. Selain dua operasi tersebut, digunakan juga empat penelusuran table (table lookup) array indeks yang dipakai setiap kali putaran. Tahapan enkripsi pada algoritma blowfish adalah sebagai berikut (Ratih n.d.):

1. Inisial array P dibentuk sebanyak 18 buah yaitu (P1,P2,……..P1) dengan masing - masing mempunyai nilai 32 bit. Pada array P terdiri atas 18 kunci 32 bit menggunakan subkunci : P1,P2,…….,P18
2. S-box dibentuk sebanyak 4 buah dengan masing - masing bernilai 32 bit dan memiliki masukan 256 .

   Empat 32-bit S-box yang masing-masing mempunyai 256 entri adalah sebagai berikut:
   S1,0,S1,1,……..,S1,255
   S2,0,S2,1,……..,S2,255
   S3,0,S3,1,……..,S3,255
   S4,0,S4,1,……..,S4,255
2. Plainteks yang sudah dienkripsi akan diasumsikan menjadi masukan. Plainteks akan diambil 64 bit. Jika terdapat kurang dari 64 bit, maka harus ditambahkan bitnya agar dapat dioperasikan sesuai dengan datanya.
3. Hasil pengambilan yang sudah dilakukan tersebut akan dibagi menjadi dua yaitu 32 bit pertama sebagai XL dan 32 bit kedua sebagai XR
4. Langkah selanjutnya dilakukan operasi XL = XL XOR Pi dan XR = F(XL) XOR XR
5. Setelah operasi tersebut, maka dilakukan penukaran XL menjadi XR, sedang XR menjadi XL.
6. Proses tersebut dilakukan sebanyak 16 kali putaran, dan perulangan yang ke-16 dilakukan lagi proses penukaran antara XL dan XR.
7. Pada proses yang ke-17 dilakukan operasi untuk XR = XR XOR P17 dan XL = XL XOR P18.
8. Proses terakhir disatukan kembali antara XL dan XR sehingga menjadi 64-bit kembali.

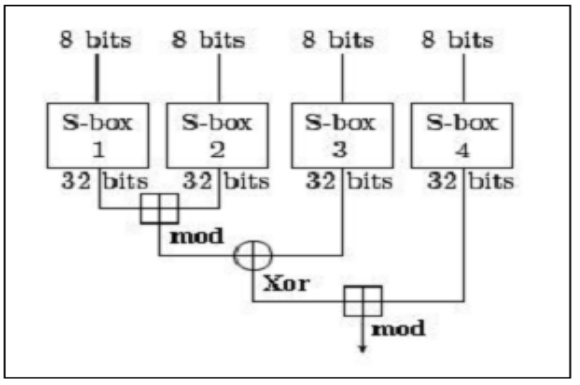
Blowfish menggunakan jaringan *Feistel* yang terdiri dari 16 kali putaran. Skema jaringan *Feistel* dapat dilihat di gambar berikut.

ISBN: 978-979-3649-96-2 Unisbank Semarang, 28 Juli 2016



Gambar 2 : Jaringan *Feistel* untuk Algoritma *Blowfish*(Mandal 2012a)

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.

ISBN: 978-979-3649-96-2 Unisbank Semarang, 28 Juli 2016

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*.

Blowfish memerlukan 64 bit blok-blok plaintext yang digunakan sebagai masukannya dan menghasilkan 64 bit chipertext. Blowfish mempunyai ukuran kunci yang bisa dipilih dari range 32 bit sampai 448 bit dimana semakin besar ukurannya maka semakin kuat juga keamanannya. Blok masukan dipecah dalam paro L0 dan R0 dimana tiap-tiap paro tersebut mengandung 32 bit. Secara sederhana algoritmanya bisa digambarkan sebagai berikut :

```
j = 1
loop from j to 16
Rj = Lj-1 XoR Pj
Lj = F (Rj) XoR Rj-1
end loop
L17 = R16 XoR P18
R17 = L16 XoR P17
```

Gambar 4:Algoritma enkripsi *blowfish*

Proses dekripsi pada *blowfish* mempunyai sifat maju. Dekripsi akan bekerja sebagaimana yang dilakukan enkripsi. Akan tetapi masukannya dimulai dengan chipertext. Sehingga sub kunci yang digunakan dalam posisi urutan terbalik. Algoritma dekripsi bisa digambarkan sebagai berikut :

```
j = 1
loop from j to 16
Rj = Lj-1 XoR P19-j
Lj = F (Rj) XoR Rj-1
end loop
L17 = R16 XoR P1
R17 = L16 XoR P2
```

Gambar 5 : Algoritma dekripsi *blowfish*

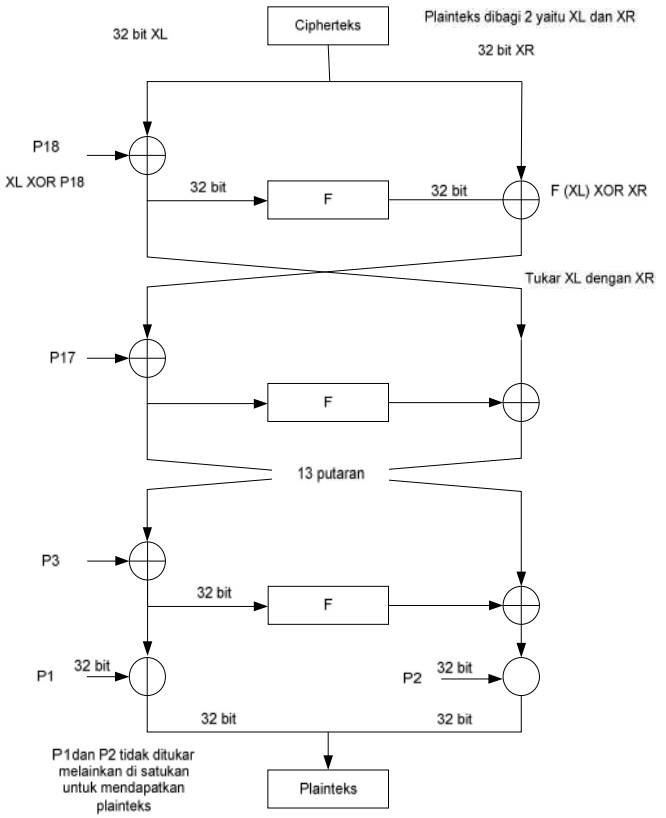
Pada algoritma *blowfish* belum ditemukan kelemahannya sampai periode ini. Hal itu karena adanya *weak key* yang terdiri dari dua entri S-box yang mempunyai nilai sama. Sampai saat ini juga belum ada cara untuk mengecek *weak key* sebelum melakukan *key expansion*. Namun hal tersebut tidak mempengaruhi dari hasil enkripsi. Hasil enkripsi dengan algoritma blowfish sangat tidak mungkin dan tidak praktis untuk diterjemahkan tanpa bantuan kunci.

ISBN: 978-979-3649-96-2 Unisbank Semarang, 28 Juli 2016

Sampai sekarang pun belum ada kripnalis yang bisa membongkar pesan tanpa kunci yang telah dienkripsi menggunakan algoritma *blowfish*. (Munir 2005; Thakur & Kumar 2011). Untuk menjaga keamanan dari pembongkaran pesan, maka dalam algoritmanya melakukan 16 kali putaran agar pesan yang disimpan tidak bisa dibongkar dan aman. Dengan adanya proses enkripsi, maka algoritma *blowfish* selalu aaman dalam hal pengenkripsian sebuah pesan.

2.4 Hasil dan Pembahasan

Dalam penelitian ini diberikan flowchat proses ekstraksi password dengan menggunakan algoritma *blowfish* sebagai berikut :



Gambar 6 : Flowchart proses ekstraksi password dengan Algoritma *Blowfish*

Aplikasi keamanan password dalam penelitian ini menggunakan bahasa pemrograman *Visual Basic* versi 6.0. Aplikasi ini untuk memanipulasi password sebelum digunakan agar tidak diketahui orang lain. Di sisi lain, password yang digunakan memiliki dua kali pertahanan. Berikut hasil dari implementasi aplikasi bahasa pemrograman visual basic 6.0 dalam pengamanan password dengan algoritma *blowfish.*

ISBN: 978-979-3649-96-2

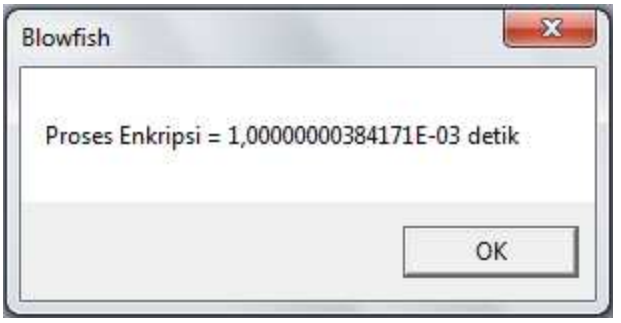
Gambar 7 : Tampilan Awal Aplikasi Enkripsi Password

Pada gambar 7 merupakan tampilan awal dari aplikasi enkripsi password menggunakan algoritma *blowfish*. Pada tampilan tersebut terdapat pilihan opsional yang digunakan untuk mengaktifkan kunci yang akan digunakan, yaitu hexastring dan alphabet. Setelah kunci tersebut ditetapkan menjadi kunci untuk menggenarate berdasarkan opsi yang telah dipilih oleh pengguna. Kemudian pengguna harus menuliskan plainteks yang akan digunakan.

Gambar 8 : Proses Enkripsi Password

Pada gambar 8 merupakan proses enkripsi menggunakan algoritma *blowfish*. Pada percobaan yang dilakukan, opsi dari model kunci yang digunakan adalah mode alfabet. Sedang waktu yang digunakan untuk enkripsi adalah sebagai berikut :

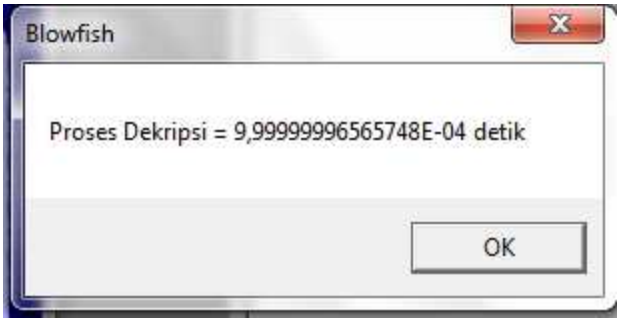
ISBN: 978-979-3649-96-2



Gambar 9 : Waktu proses enkripsi

Gambar 10 : Proses Dekripsi Password

Pada gambar 10 merupakan proses dekripsi dilakukan untuk mengevaluasi apakah proses enkripsi sudah dilakukan dengan benar. Dalam percobaan yang telah dilakukan, eksraksi berjalan dengan baik dan menghasilkan deciphered yang sama dengan plainteks (file induk asli). Sedang waktu yang digunakan untuk proses dekripsi adalah sebagai berikut:



Gambar 11 : Waktu Proses dekripsi

ISBN: 978-979-3649-96-2

2.5 Kesimpulan

Algoritma *blowfish* merupakan algoritma yang mudah dan cepat serta tersusun rapi. Algoritma ini juga mudah dijalankan, sederhana dan terjamin keamanannya. Sampai saat ini belum ada kripnalis yang bisa menembus keamanan yang dibuat oleh algoritma *blowfish* dengan menggunakan 16 kali putaran. Dari hasil penelitian yang sudah dilakukan dapat disimpulkan bahwa algoritma *blowfish* terbukti handal dalam pengamanan password.

UCAPAN TERIMA KASIH

Kami mengucapkan terima kasih kepada Universitas Dian Nuswantoro atas dana yang diberikan pada kami, sehingga penelitian ini bisa terlaksana dengan baik.

DAFTAR PUSTAKA

Abd Elminaam, D.S., Kader, H.M.A. & Hadhoud, M.M., 2010. Evaluating the performance of symmetric encryption algorithms. *International Journal of Network Security*, 10(3), pp.213–219.

Andriyanto, T. & Pardede, C., Studi Dan Perbandingan Algoritma Idea Dan Algoritma Blowfish. In Universitas Gunadarma.

Andriyanto, T. & Pardede, D.L.C., STUDI DAN PERBANDINGAN ALGORITMA IDEA DAN ALGORITMA BLOWFISH.

Kromodimoeljo, S., 2009. *Teori dan Aplikasi Kriptografi* S. Kromodimoeljo, ed., SPK IT Consulting.

Mandal, P.C., 2012a. Superiority of Blowfish Algorithm. *International Journal of Advance Research in Computer Science and Software Engineering*, 2(9), pp.196–201.

Mandal, P.C., 2012b. Superiority of Blowfish Algorithm. *International Journal*, 2(9), pp.196–201.

.