Jurnal Elektronik Ilmu Komputer Udayana
Volume 8, No 3. February 2020

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Rancang Bangun Aplikasi Enkripsi dan Dekripsi Objek 3 Dimensi menggunakan Algoritma Blowfish

Barneci Henderika Nuboba [a1], I Gusti Ngurah Anom Cahyadi Putra[a2], I Ketut Gede Suhartana[b3]

Program Studi Teknik Informatika,
Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Jalan Kampus Bukit Jimbaran, Badung, Bali. Kode Pos: 80361. Indonesia
[1]nbarneci@yahoo.com, [2]anom.cp@unud.ac.id, [3]ikg.suhartana@unud.ac.id

Abstrak

Kemajuan Teknologi Digital telah mempengaruhi cara berpikir dalam desain pada Arsitektur. Sekarang banyak Arsitek yang menggunakan *software* tidak hanya sebagai alat menggambar objek 2 dimensi tetapi telah berkembang menjadi alat untuk menggambar objek 3 dimensi (*modeling*) dan presentasi multimedia. 3 dimensi *modeling* telah menjadi bagian yang tak terpisahkan dalam proses desain arsitektur. 3 dimensi semakin sering digunakan karena didapati informasi secara utuh dan akurat sesuai keadaan nyata. Dengan begitu banyaknya perkembangan yang terjadi di dunia arsitektur, keamanan pada objek 3 dimensi sangat dibutuhkan mengingat telah terjadi banyak kasus plagiarisme desain arsitektur.

Berdasarkan hasil survey yang dilakukan di 72 Perguruan Tinggi di Indonesia oleh Tim Alumni Reference Group (ARG), disebutkan bahwa masih banyak yang belum mengetahui dan menyadari tentang plagiarisme sehingga mereka sering tidak menyadari jika telah melakukan tindakan plagiarisme, selain itu juga masih adanya mindset atau persepsi yang keliru dan menganggap bahwa mengambil atau mencuri ide, hak cipta dan hak intelektual seseorang bukan merupakan masalah serius atau mungkin sudah lazim (Mochtar, 2014).

Dengan latar belakang di atas, penulis memilih judul "Keamanan Objek 3 Dimensi menggunakan Algoritma *Blowfish*" dimana penulis mengimplementasikan ilmu kriptografi untuk mengamankan data berupa desain arsitektur objek 3 dimensi agar terhindar dari praktek plagiarism menggunakan algoritma *blowfish.*

Kata Kunci: *kriptografi, enkripsi, dekripsi, objek 3 dimensi, algoritma blowfish*

1. Pendahuluan

Kriptografi merupakan salah satu metode pengamanan data yang dapat digunakan untuk menjaga kerahasiaan data, keaslian data serta keaslian pengirim. Metode ini bertujuan agar informasi yang bersifat rahasia yang dikirim melalui telekomunikasi umum seperti LAN atau internet, tidak dapat diketahui atau dimanfaatkan oleh orang yang tidak berkepentingan atau yang tidak berhak menerimanya (Ariyus, 2008).

Kemajuan Teknologi telah mempengaruhi cara berpikir dalam dunia gambar maupun desain grafis. Sekarang banyak pegiat desain grafis yang menggunakan *software* tidak hanya sebagai alat menggambar objek 2 dimensi tetapi telah berkembang menjadi alat untuk menggambar objek 3 dimensi (*modeling*) dan presentasi multimedia. Selain memiliki nilai seni yg sangat tinggi, desain 3 dimensi ini juga penting bagi kehidupan sehari-hari untuk mengetahui perkembangan dunia digital dan juga dunia gambar. Desain 3 dimensi juga dimanfaatkan untuk pembuatan sebuah sketsa

bangunan di bidang arsitektur, otomotif dan juga denah geografis. Dengan begitu banyaknya perkembangan teknologi yang terjadi, keamanan pada desain objek 3 dimensi sangat dibutuhkan untuk menghindari praktek pencurian ide dalam berbagai bidang yang menggunakan objek 3 dimensi sebagai sarana informasi maupun seni. Kriptografi adalah metode yang tepat untuk mengatasi masalah diatas.

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 5 KB dan kesederhanaan pada algoritmanya. (Sitinjak, 2010).

Dengan latar belakang di atas, penulis memilih judul "Implementasi Aplikasi Enkripsi dan Dekripsi Objek 3 Dimensi menggunakan Algoritma *Blowfish*" dimana penulis mengimplementasikan ilmu kriptografi untuk mengamankan data berupa desain arsitektur objek 3 dimensi agar terhindar dari praktek pencurian ide menggunakan algoritma *blowfish.*

2. Metode Penelitian

Penelitian ini menggunakan model SDLC *(Software Development Life Cycle). System Development Life Cycle* (SDLC) terdiri dari tahap perencanaan sistem, analisis sistem, perancangan sistem, implementasi sistem dan pengujian sistem.

2.1 Perencanaan Sistem

Kegiatan yang dilakukan pada tahap ini adalah sebagai berikut:

1. Mendefinisikan Masalah

Definisi permasalahan meliputi rumusan masalah serta batasan-batasan masalah yang ada dalam penelitian ini yaitu enkripsi dan dekripsi objek 3 dimensi dengan menggunakan algoritma *blowfish.*

2. Menentukan Tujuan Sistem

Setelah mendefinisikan masalah, selanjutnya dilakukan penentuan tujuan dari penelitian ini yaitu menghasilkan sistem yang dapat melakukan proses enkripsi dan dekripsi objek 3 dimensi menggunakan algoritma *blowfish.*

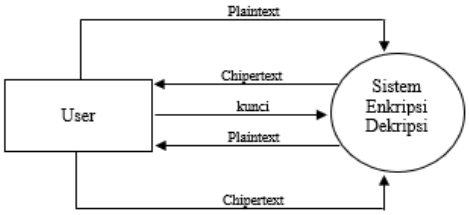
2.2 Analisis Sistem

Dilakukan proses pencarian informasi dan data yang dibutuhkan oleh sistem yang akan dirancang dalam tahap analisis sistem ini. Teknik pengumpulan data yang digunakan adalah studi literature yang berguna untuk mendapatkan dasar teori sebagai pedoman yang sangat dibutuhkan dan berhubungan dengan algoritma *blowfish.*

2.3 Perancangan Sistem

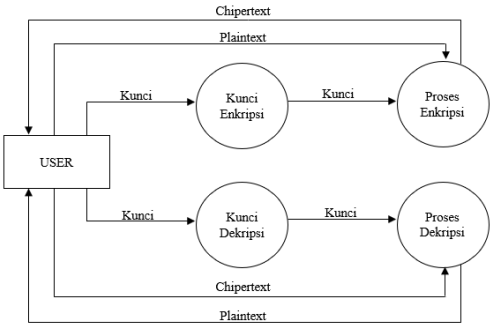
Pada tahap perancangan sistem, dirumuskan tahapan yang akan digunakan dalam proses enkripsi dan dekripsi objek 3 dimensi menggunakan algoritma *blowfish.* Berikut DFD dari sistem yang akan dibangun:

p-ISSN: 2301-5373
e-ISSN: 2654-5101



Gambar 1. Context Diagram

Alir data yang terjadi pada sistem dapat dijelaskan pada gambar 1 dimana pengguna memberikan *input* berupa *plaintext,* kunci dan *chipertext.* Data yang di *input* tersebut kemudian akan diproses sesuai dengan menu yang dipilih. Data masukan yang berupa *plaintext* akan diproses pada menu enkripsi dan menghasilkan *chipertext,* sedangkan data yang berupa *chipertext* akan diproses

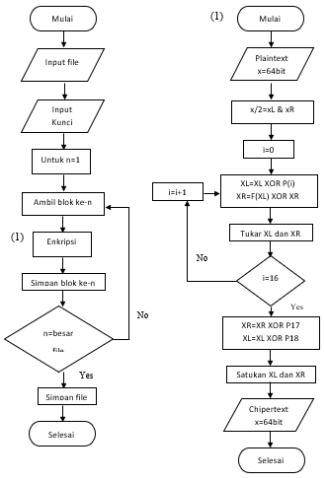


pada menu dekripsi sehingga menghasilkan *plaintext.*

Gambar 2. DFD Level 0

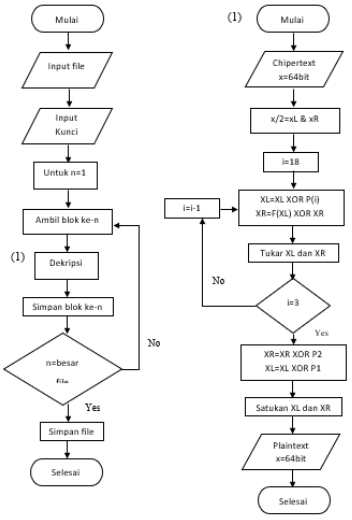
Pada Gambar 2 dijelaskan bahwa *user* memilih *plaintext* yang berupa objek 3 dimensi dengan format (.obj) kemudian memasukan kunci dan melakukan proses enkripsi sehingga mengembalikan *chipertext* kepada *user.* Kemudian untuk proses dekripsi, *user* memilih *chipertext* yaitu objek 3 dimensi dengan format (.obj) berupa hasil enkripsi dan memasukan kunci. Pada kunci yang dimasukan akan diperiksa, apa bila benar maka *output* yang dihasilkan berupa *plaintext* yaitu objek 3 dimensi.

Berikut *flowchart* perancangan sistem yang akan digunakan dalam enkripsi objek 3 dimensi pada aplikasi algoritma *Blowfish:*



Gambar 3. Flowchart Enkripsi Sistem

Berikut *flowchart* perancangan sistem yang akan digunakan dalam dekripsi objek 3 dimensi pada aplikasi algoritma *Blowfish:*



Gambar 4. Flowchart Dekripsi Sistem

1. Flowchart Enkripsi Sistem

   Pada gambar 3 dijelaskan alur enkripsi sistem menggunakan algoritma *blowfish*. Pertama, user menginput masukan berupa objek 3 dimensi, kemudian menginput kunci untuk mengenkripsi file objek 3 dimensi tersebut. Pada proses enkripsi, dijalankan sesuai algoritma *blowfish* yaitu:

   a. Inisialisasi blok pertama
   b. Kemudian ambil blok ke-n dari jumlah file yang di inputkan.
   c. Proses enkripsi *plaintext* dimulai dengan x = 64 bit. Kemudian x dibagi menjadi 2 bagian yaitu XL (X *Left* = 32 bit) dan XR (X *Right* = 32 bit) dengan putaran yang dimulai dari i=0
   d. Proses selanjutnya yaitu operasi XL = XL XOR Pi dan XR=F(XL) XOR XR. Dengan menukar hasil dari XL dan XR (XL=XR dan XR=XL) yang dilakukan sebanyak 16 kali.
   e. Pada proses ke-17, lakukan operasi untuk XR=XR XOR P17 dan XL=XL XOR P18.
   f. Kemudian gabungkan kembali XL dan XR (sehingga menjadi x = 64bit)
   g. Hasilnya adalah *chipertext* dengan x = 64 bit
   h. Simpan blok ke-n tersebut
   i. Lakukan sampai semua blok dari data terenkripsi
   j. Terakhir, simpan hasil enkripsi

2. Flowchart Dekripsi Sistem

   Pada gambar 4 dijelaskan alur dekripsi sistem menggunakan algoritma *blowfish*. Pertama, user menginput masukan berupa chipertext yang ingin didekripsi, kemudian menginput kunci untuk mendekripsi chipertext tersebut. Pada proses enkripsi, dijalankan sesuai algoritma *blowfish* yaitu:

   a. Inisialisasi blok pertama
   b. Kemudian ambil blok ke-n dari jumlah file yang di inputkan.
   c. Proses enkripsi *chipertext* dimulai dengan x = 64 bit. Kemudian x dibagi menjadi 2 bagian yaitu XL (X *Left* = 32 bit) dan XR (X *Right* = 32 bit) dengan putaran yang dimulai dari i=18
   d. Proses selanjutnya yaitu operasi XL = XL XOR Pi dan XR=F(XL) XOR XR. Dengan menukar hasil dari XL dan XR (XL=XR dan XR=XL)
   e. Pengulangan dilakukan sebanyak (i = i - 1) sampai jumlah perulangna i=3.

f. Pada proses selanjutnya lakukan operasi untuk XR = XR XOR P2 dan XL = XL XOR P1.
g. Gabungkan kembali XL dan XR (sehingga menjadi x = 64 bit) agar *plaintext* kembali menjadi x = 64 bit
h. Simpan blok data
i. Lakukan sampai seluruh data terdekripsi
j. Terakhir, simpan hasil dekripsi

2.4 Implementasi Sistem

Pada tahap ini dilakukan penerapan hasil dari perancangan sistem ke dalam baris-baris kode program sehingga dapat dimengerti oleh mesin. Komponen pendukung yang akan diimplementasikan adalah algoritma *blowfish* untuk mengenkripsi dan mengdekripsi objek 3 dimensi.

2.5 Pengujian Sistem

Tahap pengujian merupakan tahap untuk memastikan apakah sistem yang dibuat telah sesuai dengan tujuan yang ingin dicapai. Pada pengujian ini dilakukan dengan menggunakan metode *black-box* dan RMS (*Root Mean Square*)

1. *Black-box testing*

Pengujian black-box borfokus pada persyaratan fungsional dari perangkat lunak. Pada pengujian ini memungkinkan analisis sistem memperoleh kumpulan kondisi input yang akan mengerjakan seluruh keperluan fungsional program. Dengan melakukan berbagai macam skenario terdapat sistem dilihat hasilnya dan disimpulkan validasinya.

2. RMS (*Root Mean Square*)

Pada tahap ini, dilakukan pengujian dengan menggunakan Metode RMS untuk mengetahui tingkat keamnan dan kualitas hasil pengujian dari sistem yang dibuat.

a. Menguji file asli dengan file yang telah dienkripsi. Hasil perbandingan antara file asli dan file enkripsi tidak boleh sama atau hasil pengujian RMS tidak boleh bernilai 0. Apabila bernilai 0, maka file enkripsi tersebut memiliki keamanan yang buruk
b. Menguji file enkripsi dengan file dekripsi. Hasil perbandingan antara file enkripsi dan file dekripsi tidak boleh sama atau hasil pengujian RMS tidak boleh bernilai 0. Apabila bernilai 0, maka file enkripsi tersebut memiliki keamanan yang buruk.
c. Menguji file asli dengan file yang telah dienkripsi. Hasil perbandingan antara file asli dan file dekripsi harus sama atau hasil pengujian RMS harus bernilai 0. Apabila bernilai 0, maka proses dekripsi berhasil.

3. Implementasi Sistem

Implementasi aplikasi enkripsi dan dekripsi objek 3 dimensi menggunakan algoritma *Blowfish* dibagi menjadi dua proses yaitu proses enkripsi dan dekripsi. Pada proses enkripsi, objek 3 dimensi dengan format (.obj) di*input* terlebih dahulu, kemudian dilanjutkan dengan kunci. Setelah kunci telah di *setting,* kemudian pilih proses enkripsi lalu lokasi penyimpanan file yang telah dienkripsi, aplikasi kemudian menghasilkan output berupa *chipertext* dengan format (.blf), sehingga *file* tidak dapat diakses. Pada proses dekripsi, merupakan kebalikan dari proses enkripsi, pertama-tama pilih proses dekripsi, kemudian pilih *chipertext* dengan format (.blf), lalu masukan kunci yang harus sama dengan proses enkripsi sebelumnya, pilih lokasi penyimpanan proses, lakukan proses dekripsi sehingga output yang didapatkan merupakan *file* dengan format (.obj).

3.1 Proses Ekspansi Algoritma *Blowfish*

Pada bagian ekspansi kunci ini, user memasukan sebuah kunci berukuran 32bit sampai 448bit yang akan memberi output berupa array subkunci dengan total 4168bit. Terjadi inisialisasi P-array, kemudian menentukan S-block secara berurutan dengan string yang tetap, string ini adalah digitan hexadecimal dari Pi.

```
#region Cryptography

        /// <summary>
        /// Sets up the S-blocks and the key
        /// </summary>
        /// <param name="cipherKey">Block cipher key (1-448 bits)</param>
        private void SetupKey(byte[] cipherKey)
        {
            bf_P = SetupP();
            //set up the S blocks
            bf_s0 = SetupS0();
            bf_s1 = SetupS1();
            bf_s2 = SetupS2();
            bf_s3 = SetupS3();

            key = new byte[cipherKey.Length]; // 448 bits
            if (cipherKey.Length > 56)
            {
                throw new Exception("Key too long. 56 bytes required.");
            }
```

Kemudian terjadi tahap pembangkitan Subkunci dimana proses xor P1 dengan 32bit pertama kunci, xor P2 dengan 32bit kedua kunci, dan seterusnya hingga P18.

```
Buffer.BlockCopy(cipherKey, 0, key, 0, cipherKey.Length);
            int j = 0;
            for (int i = 0; i < 18; i++)
            {
                uint d = (uint)(((key[j % cipherKey.Length] * 256 + key[(j + 1) %
cipherKey.Length]) * 256 + key[(j + 2) % cipherKey.Length]) * 256 + key[(j + 3) %
cipherKey.Length]);
                bf_P[i] ^= d;
                j = (j + 4) % cipherKey.Length;
            }

            xl_par = 0;
            xr_par = 0;
            for (int i = 0; i < 18; i += 2)
            {
                encipher();
                bf_P[i] = xl_par;
                bf_P[i + 1] = xr_par;
            }

            for (int i = 0; i < 256; i += 2)
            {
                encipher();
                bf_s0[i] = xl_par;
                bf_s0[i + 1] = xr_par;
            }
            for (int i = 0; i < 256; i += 2)
```

p-ISSN: 2301-5373
e-ISSN: 2654-5101

```
            {
                encipher();
                bf_s1[i] = xl_par;
                bf_s1[i + 1] = xr_par;
            }
            for (int i = 0; i < 256; i += 2)
            {
                encipher();
                bf_s2[i] = xl_par;
                bf_s2[i + 1] = xr_par;
            }
            for (int i = 0; i < 256; i += 2)
            {
                encipher();
                bf_s3[i] = xl_par;
                bf_s3[i + 1] = xr_par;
            }
        }
```

3.2 Penjelasan Proses Enkripsi Algoritma *Blowfish*

Plainteks yang diambil sebesar 64bit kemudian dibagi menjadi dua bagian, 32bit pertama disebut XL, 32bit kedua disebut XR. Selanjutnya lakukan operasi XL = XL xor P1 dan XR = XR xor F(XL). Lakukan perulangan sebanyak 16 kali, perulangan ke 16 lakukan lagi proses penukaran XL dan XR. Dan yang terakhir pada P17 dan P18 tidak dilakukan penukaran, tetapi penyatuan antara XL dan XR untuk mendapatkan chiperteks. Berikut adalah *pseudocode* enkripsi dari algoritma Blowfish:

```
/// <summary>
        /// Runs the blowfish algorithm (standard 16 rounds)
        /// </summary>
        private void encipher()
        {
            xl_par ^= bf_P[0];
            for (uint i = 0; i < ROUNDS; i += 2)
            {
                xr_par = round(xr_par, xl_par, i + 1);
                xl_par = round(xl_par, xr_par, i + 2);
            }
            xr_par = xr_par ^ bf_P[17];

            //swap the blocks
            uint swap = xl_par;
            xl_par = xr_par;
            xr_par = swap;
        }
```

3.3 Penjelasan Proses Dekripsi Algoritma *Blowfish*

Proses Dekripsi pada algoritma *Blowfish* sama dengan proses enkripsinya. Hanya subkey saja yang dibalik. Chiperteks yang diambil sebesar 64bit. Dengan membalikkan 18 subkey untuk mendekripsi. Proses dekripsi sama dengan proses enkripsi, hanya masukan awal saja yang berbeda.

Pada proses dekripsi P18 diproses terlebih dahlulu. P18 xor XL, kemdudian F(XL) xor XR lalu ditukar tempat, begitu seterusnya. Pada bagian akhir P1 dan P2 tidak melakukan proses penukaran, namun melakukan proses penyatuan untuk mendapatkan plainteks. Berikut adalah pseudocode dekripsi dari algoritma *Blowfish:*

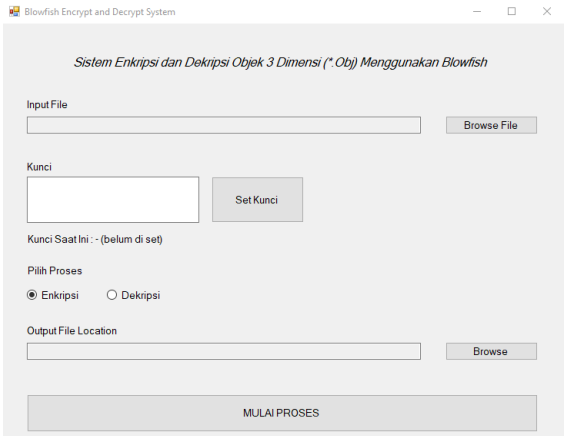
```
/// <summary>
 /// Runs the blowfish algorithm in reverse (standard 16 rounds)
 /// </summary>
 private void decipher()
 {
     xl_par ^= bf_P[17];
     for (uint i = 16; i > 0; i -= 2)
     {
         xr_par = round(xr_par, xl_par, i);
         xl_par = round(xl_par, xr_par, i - 1);
     }
     xr_par = xr_par ^ bf_P[0];

     //swap the blocks
     uint swap = xl_par;
     xl_par = xr_par;
     xr_par = swap;
 }
```

3.4 Tampilan Antarmuka Aplikasi

Form utama terdiri dari beberapa menu yang bisa diakses oleh user sesuai dengan fungsinya masing-masing. Berikut keterangan dari menu tersebut:

1. Menu Brows File
   Menu ini berfungsi untuk membuka dan memilih objek 3 dimensi yang akan diproses.
2. Menu Set Kunci
   Menu ini berfungsi untuk menginput kunci yang diinginkan oleh user. Dimana dalam kunci yang di input harus bersifat rahasia
3. Menu Enkripsi / Dekripsi
   Menu ini berfungsi untuk memilih proses apa yang ingin dilakukan oleh user.
4. Menu Browse
   Menu ini berfungsi untuk menentukan lokasi penyimpanan file yang akan diproses.
5. Menu Mulai Proses
   Menu ini berfungsi untuk menjalankan proses enkripsi atau dekripsi sesuai dengan pilihan user.



p-ISSN: 2301-5373
e-ISSN: 2654-5101

Gambar 5. Tampilan Antarmuka Sistem

3.5 Pengujian Sistem

Pengujian sistem dimaksudkan untuk mengetahui sejauh mana tingkat keberhasilan dalam mengamankan file yang berupa objek 3 dimensi, sehingga dapat berfungsi sesuai dengan tujuan awal pembuatan sistem ini yaitu untuk keamanan objek 3 dimensi.

3. 6 Black Box Testing

Pengujian *Black Box* dilakukan untuk mengetahui fungsi spesifik dai aplikasi. Pengujian ini mendemonstrasikan setiap fungsi dari aplikasi dan mengetahui apakah terjadi error atau tidak. Pengujian black box digunakan untuk mengetahui apakah input atau output yang dihasilkan aplikasi sudah sesuai dengan yang diinginkan.

No.	Uji Coba	Skenario Sistem	Keterangan
1	Browse File	Aplikasi menampilkan pilihan lokasi objek 3 dimensi yang akan diproses	Berhasil
2	Set Kunci	Aplikasi menampilkan kunci yang akan diinput	Berhasil
3	Pilih Proses	Aplikasi dapat memilih proses enkripsi atau dekripsi	Berhasil
4	Browse	Aplikasi menampilkan lokasi penyimpanan file setelah diproses	Berhasil
5	Mulai Proses	Aplikasi dapat melakukan proses enkripsi atau dekripsi objek 3 dimensi	Berhasil

Tabel 2. Black Box Testing

3.8 RMS (Root Mean Square)

Pada tahap kedua dilakukan pengujian dengan menggunkan metode RMS (Root Mean Square) untuk mengetaui perbedaan dan kualitas hasil pengujian dari sistem yang dibuat. Berikut tabel perbandingan nilai RMS dari objek 3 yang melewati 3 tahap perbandingan:

1. Perbandingan Nilai RMS file asli dengan file enkripsi

No.	Nama File	Nilai RMS
1.	aa.obj	3600.67
2.	aa.blf	
3.	ee.obj	24488.67
4.	ee.blf	

Tabel 3. Perbandingan Nilai RMS file asli dengan file enkripsi

Dari Tabel pengujian 3 diperoleh bahwa hasil pengujian RMS antara file asli dan file enkripsi tidak menghasilkan 0 ataupun sama, sehingga keamanan dari file asli tetap terjaga. Ini membuktikan bahwa file asli telah berhasil disamarkan dengan baik oleh proses enkripsi *B*lowfish.

2. Perbandingan Nilai RMS file enkripsi dan dekripsi

No.	Nama File	Nilai RMS
1.	aa.obj	2547.355
2.	aa.blf	
3.	ee.obj	24490.5
4.	ee.blf	

Tabel 4. Perbandingan Nilai RMS file asli dengan file enkripsi

Dari Tabel pengujian 4 diperoleh bahwa hasil pengujian RMS antara file enkripsi dan dekripsi tidak sama dan juga tidak berniai 0 sehingga bias dikatakan keamanan dari file asli tetap terjaga.

3. Perbandingan Nilai RMS file asli dengan file dekripsi

No.	Nama File	Nilai RMS
1.	aa.obj	3597
2.	aa.blf	
3.	ee.obj	24485
4.	ee.blf	

Tabel 5. Perbandingan Nilai RMS file asli dengan file dekripsi

Dari Tabel pengujian 5 diperoleh bahwa pengujian RMS file asli dan file dekripsi menghasilkan nilai yang sama. Yang membuktikan file asli dan file dekripsi adalah sama.

4. Kesimpulan

Adapun kesimpulan yang dapat diambil dari penelitian ini adalah:

1. Algoritma dapat diimplementasikan untuk mengenkripsi dan dekripsi file berupa objek 3 dimensi
2. Melalui pengujian dengan metode RMS didapatkan bahwa keamanan dari Algoritma Blowfish dapat terjaga dengan baik, hal ini terlihat dari perbandingan kemiripan antara file asli dengan file enkripsi yang menghasilkan nilai rata-rata 3600.67.

DAFTAR PUSTAKA

[1] Ariyus, Dony. (2006). *Kriptografi Keamanan Data dan Komunikasi.* Yogyakarta: Graha Ilmu.
[2] Ariyus, Dony. (2008). *Pengantar Ilmu Kriptografi: Teori Analisis & Implementasi.* Yogyakarta: Andi Offset.
[3] *Budiyono*, Avon. (*2004*). *Enkripsi Data Kunci Simetris dengan Algoritma Kriptografi LOKI97*. Bandung: Institut Teknologi Bandung.
[4] E Pratiwi, Apriyanti. (2011). *Implementasi Enkripsi Data Dengan Algoritma Blowfish Menggunakan Java Pada Aplikasi Email.* Bandung: Jurnal Jurusan Teknik Komputer Politeknik Telkom Bandung
[5] Guritman, Sugi., Rachmaniah, Meuthia., Mardiana, Dian. (2003). *Algoritma Blowfish untuk Penyandian Pesan.* Bogor: Jurusan Ilmu Komputer, IPB.
[6] Kristanto, Andri. (2003). *Keamanan Data pada Jaringan Komputer.* Yogyakarta: Gava Media.
[7] Kristiawan, Natanael. (2014). *Implementasi Kriptografi dan Steganografi pada File Audio Menggunakan Metode Blowfish dan Parity Coding. Bali: Jurusan Ilmu Komputer, Universitas Udayana.*
[8] Munir, Rinaldi. (2003). *Kriptogafi.* Bandung: Informatika.
[9] Zuli, Faizal., Irawan, Ari. (2006). *Implementasi Kriptografi Dengan Algoritma Blowfish dan Riverst Shamir Adleman (RSA) Untuk Proteksi File.* Jakarta: Jurnal Universitas Satya Negara.