TUGAS AKHIR

PORTAL ONLINE PENYEDIA FASILITAS ENKRIPSI DAN DEKRIPSI MENGGUNAKAN METODE *CIPHER BLOCK CHAINING* (CBC)

Disusun oleh :

HARI SETIADY WIBOWO

121 065 1167

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH JEMBER

2017

TUGAS AKHIR

PORTAL ONLINE PENYEDIA FASILITAS ENKRIPSI DAN DEKRIPSI MENGGUNAKAN METODE *CIPHER BLOCK CHAINING* (CBC)

Disusun Untuk Melengkapi dan Memenuhi Syarat Kelulusan
Guna Meraih Gelar Sarjana Komputer
Program Studi Teknik Informatika Universitas Muhammadiyah Jember

Disusun oleh :

HARI SETIADY WIBOWO

121 065 1167

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH JEMBER

2017

HALAMAN PENGESAHAN

PORTAL ONLINE PENYEDIA FASILITAS ENKRIPSI DAN DEKRIPSI MENGGUNAKAN METODE *CIPHER BLOCK CHAINING* (CBC)

Hari Setiady Wibowo
121 065 1167

Telah mempertanggung jawabkan Laporan Tugas Akhir pada sidang Tugas Akhir tanggal 10 April 2017 sebagai salah satu syarat kelulusan dan mendapat gelar Sarjana Komputer (S.Kom) di Universitas Muhammadiyah Jember

Disetujui oleh,

Dosen Penguji :

Penguji I

Agung Nilogiri, S.T, M.Kom
NIP. 197703302005011002

Penguji II

Lutfi Ali Muharrom, S.Si, M.Si
NPK. 09 03 521

Dosen Pembimbing :

Taufik Timur W, S.Kom, M.Kom
NPK. 08 04 486

Jember, April 2017

Mengesahkan,
Dekan Fakultas Teknik

Ir. Suhartina, MT
NPK. 95 05 246

Mengetahui,
Ketua Program Studi Teknik Informatika

Yeni Dwi Rahayu, S.St.,M.Kom
NPK. 11 03 590

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini:

NAMA : HARI SETIADY WIBOWO
NIM : 121 065 1167

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul **"PORTAL *ONLINE* PENYEDIA FASILITAS ENKRIPSI DAN DEKRIPSI MENGGUNAKAN ALGORITMA *CIPHER BLOCK CHAINING* (CBC)"** adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, April 2017

HARI SETIADY WIBOWO
NIM. 12 1065 1167

PERSEMBAHAN

Tugas akhir ini dengan hormat penulis persembahkan kepada:

1. Allah SWT yang Maha Pengasih lagi Maha Penyayang, begitu besar Rahmat dan KaruniaMu sehingga saya dapat menyelesaikan Tugas Akhir ini;
2. Ibunda Maryam Vergalia dan Ayahanda Abdul Latief yang selalu memberikan dukungan lahir dan batin;
3. Adik Indra Koerniawan yang selalu memberi dukungan tanpa henti;
4. Dosen-dosen Universitas Muhammadiyah Jember yang tiada letihnya memberikan ilmunya untuk saya;
5. Teman-teman seperjuangaan untuk menggapai mimpi dan cita-cita bersama, Diyah Apriliana Puspita Dewi, Deny, Hendri, Darmawan, Rofik, Solehan, Nanang, Bayu, Kekel, Afan, Wafi, Rio dan juga sahabatku semuanya yang tidak bisa saya sebutkan satu persatu yang telah memberikan semangat dalam perjalanan hidup saya;
6. Sahabat sekaligus rekan kerja *Support System* Pusat Data dan Informasi yang selalu memberi motivasi tiada henti;
7. Almamaterku tercinta, Universitas Muhammadiyah Jember.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur kita panjatkan kehadirat Tuhan Yang Maha Esa atas rahmat dan karuniaNya yang telah dilimpahkan sehingga kami bisa menyelesaikan Laporan Tugas Akhir. Penyusunan Laporan Tugas Akhir disusun untuk melengkapi dan memenuhi syarat kelulusan program Strata Satu (S1) Jurusan Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Jember dan juga sebagai syarat untuk memperoleh Gelar Sarjana Komputer (S.Kom).

Ucapan terima kasih penulis sampaikan kepada pihak-pihak yang telah membantu penulis, baik selama pembuatan aplikasi maupun selama penyusunan Laporan Tugas Akhir, di antaranya :

1. Ibu Ir. Suhartina, MT. selaku dekan Fakultas Teknik Informatika;
2. Ibu Yeni Dwi Rahayu, M.Kom. selaku Ketua Program Studi Teknik Informatika Universitas Muhammadiyah Jember;
3. Bapak Taufiq Timur W, S.Kom. M.Kom selaku dosen pembimbing yang telah memberikan bimbingan dan pengarahan kepada penulis sehingga tugas akhir ini dapat penulis selesaikan;
4. Bapak Agung Nilogiri, S.T. M.Kom. selaku dosen penguji 1 yang memberikan saran dan kritik yang membangun dalam penelitian ini;
5. Bapak Lutfi Ali Muharrom, S.Si. M.Si. selaku dosen penguji 2 yang memberikan saran dan kritik yang membangun dalam penelitian ini;
6. Dosen Fakultas Teknik Informatika, terima kasih atas semua ilmu yang telah diberikan;
7. Teman-teman yang telah mendukung dan memberi semangat kepada penulis;
8. Keluarga penulis yang telah memberikan do'a dan juga bantuan secara moril dan materil;
9. Serta pihak-pihak yang telah membantu dan tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa masih banyak kekurangan dan kelemahan dalam penyusunan Laporan Skripsi ini. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dan menambah wawasan dan wacana ilmu penulis.

Besar harapan penulis semoga skripsi ini dapat bermanfaat bagi semua pihak dan dapat dimanfaatkan sebaik-baiknya.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Jember, April 2017

Penulis,

ABSTRAK

Kerahasiaan dan keamanan data adalah hal yang sangat penting dalam komunikasi data. Keamanan merupakan bentuk tindakan untuk mempertahankan suatu hal dari berbagai macam gangguan dan ancaman. Terdapat banyak faktor yang mengancam keamanan komunikasi data. Ancaman-ancaman tersebut menjadi masalah terutama dengan semakin meningkatnya komunikasi data yang bersifat rahasia.

Algoritma *Cipher Block Chaining* (CBC) merupakan penerapan mekanisme umpan balik pada sebuah blok *bit* dimana hasil enkripsi blok sebelumnya diumpan balikkan ke dalam proses enkripsi blok *current*. Caranya, blok *plaintext* yang *current* di-XOR-kan terlebih dahulu dengan blok *ciphertext* hasil enkripsi sebelumnya, selanjutnya hasil peng-XOR-an ini masuk ke dalam fungsi enkripsi. Dengan algoritma CBC, setiap blok *ciphertext* tidak hanya bergantung pada blok *plaintext* tetapi juga pada seluruh blok *plaintext* sebelumnya.

Web portal yang terbentuk telah dapat menyediakan fasilitas enkripsi dan dekripsi pada *file* *.docx, *.txt, *.jpg dan *.png menggunakan algoritma *Cipher Block Chaining* (CBC) dengan tingkat akurasi pengembalian kalimat (*.docx dan *.txt) pada 3 skenario (10, 50 dan 150 kata) pengujian adalah 100%, serta akurasi pengembalian ukuran *pixels* dan ukuran *file* (*.jpg dan *.png) dari 3 kali pengujian pada 3 *file* dengan ukuran *pixels* dan ukuran *file* yang berbeda adalah 100%. Namun tidak dapat mengembalikan gambar dan tabel yang terdapat pada *file* *.docx.

Kata Kunci : Kerahasiaan dan Keamanan, Pengiriman Data, *Cipher Block Chaining* (CBC)

ABSTRACT

Secrecy and security data is of crucial importance in communication data, security is a form of action to maintain a thing of various disorder and threat.There are many factors that threatens security data communication. Threats become the problem especially with the increase data communication that are secret.

Cipher Block Chaining algorithm is an implementation of a feedback mechanism at a result bit block encryption block previously routed invert into the encryption process blocks current. Here's how the current plaintext block, XOR operation advance with blocks of ciphertext encryption results results of previous, next it was entered in XOR operation to the function of encryption. CBC algorithm, with each block of ciphertext depends not only on the block of plaintext but also on the entire plaintext block before.

*Web portal that is formed has been able to provide encryption and decryption facilities in the * .docx, * .txt, * .jpg, and * .png algorithm using Cipher Block Chaining (CBC) with the accuracy of the return of the sentence (* .docx and * .txt) at 3 scenarios (10, 50 and 150 words) the test was 100%, and accuracy of the return of the size of pixels and the size of the files (* .jpg, and * .png) of 3 times the test on three files with a size of pixels and a different file size is 100 %. Yet it can not restore images and tables contained in the file * .docx.*

Keywords : Confidentiality and security, Shipping Data, Cipher Block Chaining (CBC)

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN ii
HALAMAN PERNYATAAN iii
ABSTRAK iv
ABSTRACT v
HALAMAN PERSEMBAHAN vi
KATA PENGANTAR vii
DAFTAR ISI ix
DAFTAR GAMBAR xi
DAFTAR TABEL xiii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 2
1.2 Rumusan Masalah 2
1.3 Tujuan 2
1.4 Batasan masalah 2
1.5 Manfaat 3
BAB II TINJAUAN PUSTAKA 4
2.1 Dokumen 4
2.1.1 Jenis-jenis dokumen dari segi pemakaiannya 4
2.1.2 Jenis-jenis dokumen dari segi fungsinya 5
2.1.3 Jenis-jenis dokumen dari segi ruang lingkupnya 5
2.2 JPG / JPEG (Joint Photographic Experts Assemble) 5
2.3 PNG (Portable Network Graphics) 6
2.4 Kriptografi 7
2.5 Tujuan Kriptografi 9
2.5.1 Algoritma Kriptografi 10
2.5.2 Macam-Macam Algoritma Kriptografi 11
2.5.3 Kriptografi Kunci Simetris 12
2.5.4 *Stream Cipher* 12

2.5.5 *Block Cipher* 12
2.5.6 Algoritma *Cipher Block Chaining*(CBC) 12
2.6 Operasi XOR (Munir, 2004) 14
2.7 *Personal Home Page* (PHP) 16
2.8 *MySQL* 16
2.9 ASCII 17
BAB III METODE PENELITIAN 18
3.1 Metode Penelitian 18
3.2 Perancangan Sistem 19
3.2.1 Diagram Blok 19
3.2.2 *Flowchart* 20
3.2.3 Proses Enkripsi Karakter 22
3.2.4 Proses Dekripsi Karakter 25
BAB IV IMPLEMENTASI DAN PENGUJIAN 29
4.1 Metode Pengujian 29
4.1.1 Pengujian Antarmuka 29
4.1.2 Pengujian Enkripsi 33
4.1.3 Pengujian Dekripsi 37
4.1.4 Pengujian *File* *.docx Berisi Gambar 41
4.1.5 Pengujian *File* *.docx Berisi Tabel 42
4.1.6 Pengujian Akurasi Dekripsi Kalimat 44
4.1.7 Pengujian Akurasi Gambar 71
BAB IV KESIMPULAN DAN SARAN 74
5.1 Kesimpulan 74
5.2 Saran 74
DAFTAR PUSTAKA 75
LAMPIRAN 76
IDENTITAS PENULIS 89

DAFTAR GAMBAR

Gambar 2.1 Proses Enkripsi dan Dekripsi 11
Gambar 2.2 Skema Enkripsi dengan Algoritma CBC 14
Gambar 2.3 Skema Dekripsi dengan Algoritma CBC 15
Gambar 3.1 Blok Diagram 19
Gambar 3.2 *Flowchart* Enkripsi 20
Gambar 3.3 *Flowchart* Dekripsi 21
Gambar 4.1 Halaman Beranda 29
Gambar 4.2 Halaman Registrasi 30
Gambar 4.3 Halaman *Login* 30
Gambar 4.4 Halaman *User* 31
Gambar 4.5 Halaman Enkripsi 31
Gambar 4.6 Halaman Dekripsi 32
Gambar 4.7 Notifikasi *User* 32
Gambar 4.8 Detail *File* 33
Gambar 4.9 *Plaintext* pengujian enkripsi *.docx 33
Gambar 4.10 *Ciphertext* pengujian enkripsi *.docx 34
Gambar 4.11 *Plaintext* pengujian enkripsi *.txt 34
Gambar 4.12 *Ciphertext* pengujian enkripsi *.txt 35
Gambar 4.13 *Plaintext* pengujian enkripsi *.jpg 35
Gambar 4.14 *Ciphertext* pengujian enkripsi *.jpg 36
Gambar 4.15 *Plaintext* pengujian enkripsi *.png 36
Gambar 4.16 *Ciphertext* pengujian enkripsi *.png 37
Gambar 4.17 *Ciphertext* pengujian dekripsi *.docx 37
Gambar 4.18 *Plaintext* pengujian dekripsi *.docx 38
Gambar 4.19 *Ciphertext* pengujian dekripsi *.txt 38
Gambar 4.20 *Plaintext* pengujian dekripsi *.txt 39
Gambar 4.21 *Ciphertext* pengujian dekripsi *.jpg 39
Gambar 4.22 *Plaintext* pengujian dekripsi *.jpg 40
Gambar 4.23 *Ciphertext* pengujian dekripsi *.png 40

Gambar 4.24 *Plaintext* pengujian dekripsi *.png 41
Gambar 4.25 *Plaintext* berisi gambar 41
Gambar 4.26 *Ciphertext* setelah dienkripsi 42
Gambar 4.27 Hasil dekripsi 42
Gambar 4.28 *Plaintext* berisi tabel 43
Gambar 4.29 *Ciphertext* setelah dienkripsi 43
Gambar 4.30 Hasil dekripsi 44

DAFTAR TABEL

Tabel 4.1 Pengujian pada skenario 1 44
Tabel 4.2 Pengujian pada skenario 2 46
Tabel 4.3 Data uji skenario 2 51
Tabel 4.4 Pengujian pada skenario 3 53
Tabel 4.5 Data uji skenario 3 66
Tabel 4.6 Pengujian akurasi dekripsi gambar 1 72
Tabel 4.7 Pengujian akurasi dekripsi gambar 2 72
Tabel 4.8 Pengujian akurasi dekripsi gambar 3 73

BAB I
PENDAHULUAN

1.1 Latar Belakang

Bertukar informasi/data di jaman teknologi yang sudah sangat berkembang pesat seperti saat ini umum dilakukan oleh berbagai pihak, dalam bertukar data pihak pengirim dan penerima terkadang mengenyampingkan hal keamanan dan kerahasiaan. Seiring dengan itu, kebutuhan pada keamanan terhadap kerahasiaan data yang saling dipertukarkan tersebut semakin meningkat. Oleh karena itu dikembangkan cabang ilmu yang mempelajari tentang cara-cara pengamanan data atau dikenal dengan istilah Kriptografi.

Kerahasiaan dan keamanan data adalah hal yang sangat penting dalam komunikasi data, keamanan merupakan bentuk tindakan untuk mempertahankan suatu hal dari berbagai macam gangguan dan ancaman. Terdapat banyak faktor yang mengancam keamanan komunikasi data. Ancaman-ancaman tersebut menjadi masalah terutama dengan semakin meningkatnya komunikasi data yang bersifat rahasia.

Selama ini aplikasi yang dapat melakukan enkripsi *file* yang berbasis web belum tersedia. Hanya tersedia dalam bentuk aplikasi *offline*. Seperti aplikasi enkripsi dan dekripsi dengan menggunakan algoritma MD5 dan aplikasi enkripsi dan dekripsi dengan algoritma RSA. Kedua metode tersebut cukup memiliki tingkat kerahasiaan dan algoritma tersendiri dalam menjadikan sebuah *plaintext* menjadi *ciphertext.*

Algoritma *Cipher Block Chaining* merupakan penerapan mekanisme umpan balik pada sebuah blok *bit* dimana hasil enkripsi blok sebelumnya diumpan balikkan ke dalam proses enkripsi blok *current.* Caranya, blok *plaintext* yang *current* di-XOR-kan terlebih dahulu dengan blok *ciphertext* hasil enkripsi sebelumnya, selanjutnya hasil peng-XOR-an ini masuk ke dalam fungsi enkripsi. Dengan algoritma CBC, setiap blok *ciphertext* tidak hanya bergantung pada blok *plaintext* tetapi juga pada seluruh blok *plaintext* sebelumnya.

Pemberian fasilitas enkripsi dan dekripsi dengan algoritma kriptografi *Cipher Block Chaining* (CBC) ke dalam sebuah web portal ini dilakukan karena metode ini diimplementasikan pada level *binary digit* (*bit*), sehingga pola proses enkripsi tidak dapat terbaca, serta proses enkripsi dan deskripsi memerlukan waktu singkat.

1.2 Rumusan Masalah

Dari uraian yang dikemukakan pada latar belakang, dapat dirumuskan masalah – masalah sebagai berikut :

1. Bagaimana membuat sebuah web yang menyediakan fasilitas enkripsi dan deskripsi dengan algoritma *Cipher Block Chaining* (CBC)
2. Berapa tingkat akurasi pengembalian atau deskripsi kalimat yang telah dienkripsi.

1.3 Tujuan

Adapun tujuan penulis melakukan penelitian ini adalah :

1. Membuat portal *online* penyedia fasilitas enkripsi dan dekripsi dengan metode *Cipher Block Chaining* (CBC) dengan menggunakan Bahasa pemrograman PHP dan MySQL
2. Mengukur keakurasian pengembalian atau deskripsi dari kalimat yang telah terenkripsi

1.4 Batasan Masalah

Agar pembahasan tidak menyimpang dari topik permasalahan yang ada, maka penulis membuat batasan masalah dalam penelitian ini, antara lain :

1. Data/*file* yang akan dienkripsi terdiri dari *file* teks (*.txt), file dokumen (*.docx), dan *file* gambar (*.jpg) dan (*.png).
2. Ukuran *file* gambar yang akan diproses tidak lebih dari 600 x 600 *pixels*.
3. Isi *file* *.docx yang dienkripsi tidak mengandung unsur tabel dan gambar.
4. *File* gambar (*.jpg dan *.png) yang dienkripsi adalah data *bit*-nya, bukan warna.

1.5 Manfaat

Manfaat dari pembangunan portal online ini dapat digunakan untuk menjaga kerahasiaan *file*/data dengan proses enkripsi dan dekripsi menggunakan algoritma *Cipher Block Chaining* (CBC) sebelum melakukan pengiriman atau berbagi *file*/data melalui media pengiriman data.

BAB II
TINJUAN PUSTAKA

2.1 DOKUMEN

Istilah Dokumentasi dari kata *document* (Belanda), *document* (Inggris), *documentum* (Latin). Sebagai kata kerja *document* berarti menyediakan dokumen, membuktikan dengan menunjukkan adanya dokumen sebagai kata benda berarti wahana (wahana = kebenaran, alat pengangkut, angkutan, alat untuk mencapai tujuan) informasi, data yang terekam atau dimuat dalam wahana tersebut beserta maknanya yang digunakan untuk belajar, kesaksian, penelitian, rekreasi, dan sebagainya. (Purwono, 2009)

2.1.1 Jenis dokumen dari segi pemakaiannya

1. Dokumen pribadi

   Dokumen pribadi yaitu surat keterangan penting yang kegunaannya untuk kepentingan pribadi contohnya adalah KTP, Ijazah, Akte Kelahiran, Surat Nikah dll.

2. Dokumen Niaga

   Dokumen niaga yaitu surat berharga yang kegunaannya adalah untuk bukti dalam melakukan transaksi contohnya adalah Surat Pengantar, Faktur dll, Ressi pengiriman barang dll.

3. Dokumen Pemerintah

   Dokumen pemerintah yaitu surar-surat penting yang di gunakan dalam instansi pemerintahan contohnya adalah Undang-undang, RAPBN dll.

4. Dokumen sejarah

   Dokumen sejarah yaitu surat-surat penting yang digunakan sebagai bukti peristiwa di masa lampau contohnya adalah Pancasila, teks proklamasi dll.

2.1.2 Jenis jenis dokumen dari segi fungsinya

1. Dokumen dinamis adalah dokumen yang digunakan secara langsung di dalam proses kerja.
2. Dokumen statis adalah kebalikan dari dokumen dinamis yaitu dokumen yang tidak digunakan secara langsung dalam proses pekerjaan.

2.1.3 Jenis jenis dokumen dari segi ruang lingkupnya

1. Dokumen korporal adalah dokumen yang di pakai secara terus-menerus dalam proses penyelenggaraan pekerjaan
2. Dokumen riteral adalah dokumen yang di tulis, di rekan, dicetak dan di gambarkan.
3. Dokumen privat adalah dokumen yang berupa surat ataupun yang bersifat pribadi.

2.2 JPG / JPEG (Joint Photographic Experts Assemble)

JPG adalah jenis data yang dikembangkan oleh *Joint Photographic Experts Assemble* (JPEG) yang dijadikan standar untuk para fotografer profesional. Seperti metode yang digunakan oleh format ZIP yang digunakan untuk menemukan pengulangan (*redundancy*) dalam data untuk kemudian dikompresi, JPG mengompresi data gambar dengan cara mengurangi bagian-bagian dari gambar untuk memblok *pixel* dalam gambar tersebut. Kompresi JPG mempunyai kekurangan yang bersifat permanen, namun teknologi ini hanya digunakan untuk menyimpan data yang besar di media penyimpanan yang terbatas, bukan untuk manipulasi foto.

JPG sudah digunakan dan menjadi standar gambar di internet karena ia bisa dikompresi hingga ukuran kecil. Data JPG tertentu bisa dikompres dengan rasio perbandingan 2:1 sampai paling tinggi 100:1, tergantung pengaturan yang anda berikan. Sewaktu koneksi internet yang tersedia di bumi ini masih berupa koneksi *dial-up,* JPG adalah satu-satunya jenis data yang bisa dikirimkan dan dilihat secara bebas.

File JPG menggunakan teknik kompresi yang menyebabkan kualitas gambar turun (*lossy compression*), maka format gambar ini tidak terlalu baik untuk digunakan menyimpan gambar pajangan atau artistik. Setiap kali menyimpan ke tipe JPG dari tipe lain, ukuran gambar biasanya mengecil, dan kualitasnya turun dan tidak dapat dikembalikan lagi. Ukuran file BMP dapat turun menjadi seper sepuluh setelah dikonversi menjadi JPG. Meskipun dengan penurunan kualitas gambar, pada gambar-gambar tertentu (misalnya pemandangan), penurunan kualitas gambar hampir tidak terlihat mata.

File JPG cocok digunakan untuk gambar yang memiliki banyak warna, misalnya foto wajah dan pemandangan dan tidak cocok digunakan untuk gambar yang hanya memiliki sedikit warna seperti kartun atau komik.

JPG juga bukan media ideal untuk penggunaan *typography*, *crisp line*, atau bahkan hasil fotografi dengan sudut yang tajam, karena obyek itu kadang menjadi samar/blur. Memang lebih enak karena file ini sangat umum dan sudah sangat memasyarakat.

JPG mendukung 24-*bit* RGB dan CMYK, dan 8-*bit Grayscale*. Tidak disarankan untuk Anda menggunakan palet CMYK dalam format JPG. Perlu dicatat juga bahwa Grayscale tidak banyak dikompres jika dibandingkan dengan versi berwarnanya. (Mulyanta, 2006)

2.3 PNG (Portable Network Graphics)

PNG adalah kepanjangan dari *Portable Network Graphics* atau bisa diplesetkan menjadi “PNG-*Not*-GIF“. Dikembangkan sebagai alternatif lain untuk GIF, yang menggunakan paten dari LZW–algoritma kompresi. PNG adalah format gambar yang sangat baik untuk grafis internet, karena mendukung transparansi didalam perambah (*browser*) dan memiliki keindahan tersendiri yang tidak bisa diberikan GIF atau bahkan JPG. Bisa disebut sebagai salah satu format yang merupakan gabungan dari format JPG dan GIF. Untuk tipe ini mampu untuk gradiasi warna.

Karena sifat transparannya yang tidak pecah-pecah, PNG yang masuk kelas 24-*bit* ini cocok untuk membuat *screenshoot*. Ia bisa mereproduksi gambar desktop

dari tiap piksel ke piksel secara detil. PNG juga mendukung kelas 8-*bit* seperti GIF, sekaligus 24-*bit* seperti JPG. Ia juga tidak pecah-pecah, bisa mengkompresi gambar dari proses fotografi tanpa banyak mengurangi kualitas gambarnya. Namun PNG cukup besar ukurannya diantara JPG dan GIF, serta tidak didukung oleh perambah / *browser* yang lama.

Tipe file PNG merupakan solusi kompresi yang *powerful* dengan warna yang lebih banyak (24 *bit* RGB + *alpha*). Berbeda dengan JPG yang menggunakan teknik kompresi yang menghilangkan data, file PNG menggunakan kompresi yang tidak menghilangkan data (*lossles compression*). Kelebihan file PNG adalah adanya warna transparan dan alpha. Warna alpha memungkinkan sebuah gambar transparan, tetapi gambar tersebut masih dapat dilihat mata seperti samar-samar atau bening. File PNG dapat diatur jumlah warnanya hingga 64 *bit* (*true color* + *alpha*) sampai *indexed* color 1 *bit*. Dengan jumlah warna yang sama, kompresi file PNG lebih baik dari pada GIF, tetapi memiliki ukuran file yang lebih besar daripada JPG. Kekurangan tipe PNG adalah belum populer sehingga sebagian *browser* tidak mendukungnya.

Format PNG ini diperkenalkan untuk menggantikan format GIF. PNG mempunyai faktor kompresi yang lebih baik dibandingkan dengan GIF (kurang lebih 5%-25% lebih baik dibanding format GIF). Tetapi ada satu fasilitas dari GIF yang tidak terdapat pada PNG format yaitu dukungan terhadap penyimpanan multi format untuk keperluan animasi. Untuk keperluan pengolahan gambar, meskipun format PNG bisa dijadikan alternatif selama proses pengolahan grafis namun format JPEG masih menjadi pilihan yang lebih baik. (Mulyanta, 2006)

2.4 Kriptografi

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Crypto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain. Dalam perkembangannya, kriptografi juga digunakan untuk mengidentifikasi pengiriman pesan dan tanda tangan digital dan kieaslian pesan dengan sidik jari digital dan keaslian pesan

dengan sidik jari digital (Ariyus, 2005). Berikut berbagai istilah atau terminology pada kriptografi (Munir, 2006) yang harus diketahui yaitu:

1. Pesan, *plaintext* dan *ciphertext*

   Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah (*plaintext*) atau teks jelas (*cleartext*). Agar pesan tidak dapat dimengerti maknanya oleh pihak lain yang tidak berkepentingan, maka pesan perlu disandikan kebentuk lain yang tidak dapat dipahami. Bentuk pesan yang tersandi disebut *chipertext* atau kriptogram (*cryptogram*). *Chipertext* harus dapat ditrnasformasikan kembali menjadi *plaintext* semula agar dapat diterima dan bisa dibaca.

2. Pengirim dan penerima

   Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan. Pengirim tentu menginginkan pesan dapat dikirim secara aman, yaitu pengirim yakin bahwa pihak lain tidak dapat membaca isi pesan yang dikirim. Solusinya adalah dengan cara menyandikan pesan menjadi *ciphertext.*

3. Enkripsi dan Deskripsi

   Proses menyandikan *plaintext* menjadi *ciphertext* disebut enkripsi (*encryption*) atau *enciphering.* Sedangkan proses mengembalikan *ciphertext* menjadi *plaintext* disebut deskripsi (*decryption*) atau *deciphering.*

4. *Cipher* dan Kunci

   Algoritma kriptografi disebut juga *cipher*, yaitu aturan untuk enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk *enciphering* dan *deciphering*.

   Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yang berisi elemen-elemen *plaintext* dan himpunan yang berisi *ciphertext*. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara dua himpunan tersebut. Misalkan *P* menyatakan *plaintext* dan *C* menyatakan *ciphertext*, maka fungsi enkripsi *E* memetakan *P*

ke *C*. Karena proses enkripsi kemudia n dekripsi mengembalikan pesan ke pesan semula, maka kesamaan berikut harus benar.

Kriptografi mengatasi masalah keamanan data dengan menggunakan kunci, yang dalam hal ini algoritma tidak dirahasiakan lagi, tetapi kunci harus tetap dijaga kerahasiaannya. Kunci (*key*) adalah parameteryang digunakan untuk transformasi *enciphering* dan *deciphering.* Kunci biasanya berupa *string* atau deretan bilangan.

5. Sistem kriptografi

   Kriptografi membentuk sebuah sistem yang dinamakan sistem kriptografi. Sistem kriptografi (*criptosystem*) adalah kumpulan yang terdiri dari algoritma kriptografi, semua *plaintext* dan *ciphertext* yang mungkin, dan kunci. Di dalam kriptografi *cipher* hanyalah salah satu komponen saja.

6. Penyadap

   Penyadap (*eavesdropper*) adalah orang yang mencoba menangkap pesan selama ditransmisikan. Tujuan penyadap adalah untuk menapatkan informasi sebanyak-banyaknya mengenai sistem kriptografi yang digunakan untuk berkomunikasi dengan maksud untuk memecahkan *ciphertext*. Nama lain penyadap : *enemy, adversary, intruder, interceptor, bad guy.*

7. Kriptanalisis dan kriptologi

   Kroptografi berkembang sedemikian rupa sehingga melahirkan bidang yang berlawanan yaitu kriptanalisis. Kriptanalisis (*cryptanalysis*) adalah ilmu dan seni untuk memecahkan *ciphertext* menjadi *plaintext* tanpa mengetahui kunci yang digunakan. Pelakunya disebut kriptanalis. Jika seorang kriptografer (*cryptographer*) mentransformasikan *plaintext* menjadi *ciphertext* dengan suatu algoritma dan kunci, maka sebaliknya seorang kriptanalis berusaha untuk memecahkan *ciphertext* tersebut untuk menemukan *plaintext* atau kunci. Kriptologi (*cryptology*) adalah studi mengenai kriptografi dan kriptanalisis.

2.5 Tujuan Kriptografi

Menurut (Munir, 2006) kriptografi bertujuan untuk memberi layanan keamanan. Yang dinamakan aspek-aspek keamanan sebagai berikut:

1. Kerahasiaan (*confidentiality*)

   Adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak. Di dalam kriptografi layanan ini direalisasikan dengan menyandikan *plaintext* menjadi *ciphertext*. Istilah lain yang senada dengan *confidentiality* adalah *secrecy* dan *privacy.*

2. Integritas data (*data integrity*)

   Adalah layanan yang menjamin bahwa pesan masih asli/utuh atau belum pernah dimanipulasi selama pengiriman. Dengan kata lain, aspek keamanan ini dapat diungkapkan sebagai pertanyaan: " apakah pesan yang diterima masih asli atau tidak mengalami perubahan (modifikasi)?".

3. Otentikasi (*authentication*)

   Adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user authentication*). Dua pihak yang saling berkomunikasi harus dapat mengontetifikasi satu sama lain agar dapat memastikan sumber pesan.

4. Nirpenyangkalan (*non-repudiation*)

   Adalah layanan untuk menjaga entitas yang berkomunikasi melakukan penyangkalan, yaitu pengirim pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menrima pesan.

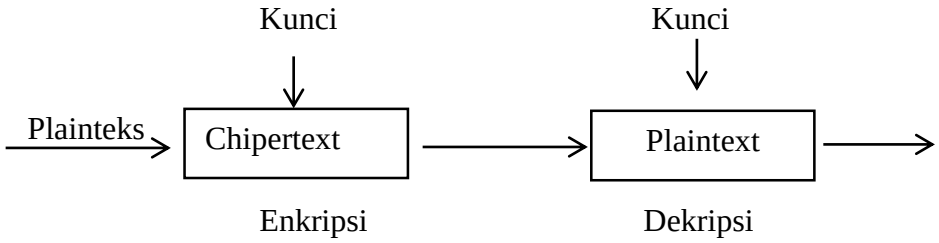
2.5.1 Algoritma Kriptografi

Algoritma kriptografi disebut juga *cipher*, yaitu aturan untuk *enciphering* dan *deciphering*, atau fungsi matematika yang digunakan untuk enkripsi dan deskripsi. Keamanan algoritma kriptografisering diukur dari banyaknya kerja yang dibutuhkan untuk memecahkan *ciphertext* menjadi *plaintext* tanpa mengetahui kunci yang digunakan. Apabila semakin banyak proses yang diperlukan berarti juga semakin lama waktu yang dibutuhkan, maka semakin kuat algoritma yang tersebut dan semakin aman digunakan untuk menyandikan pesan.

Algoritma kriptografi terdiri dari fungsi dasar, yaitu :

1. Enkripsi, merupakan hal yang sangat penting dalam kriptografi yang merupakan pengamanan data yang dikirimkan terjaga rahasianya, pesan asli disebut plainteks yang dirubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan dengan *cipher* atau kode.
2. Deskripsi, merupakan kebalikan dari enkripsi, pesan yang telah dienkripsi dikembalikan kebentuk asalnya (plainteks) disebut dengan dekripsi pesan. Algoritma yang digunakan untuk dekripsi berbeda dengan yang digunakan untuk enkripsi.
3. Kunci, yang dimaksud di sini adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi, kunci terbagi menjadi dua bagian yaitu kunci pribadi (*private key*) dan kunci umum (*public key*).

Gambar dibawah memperlihatkan skema enkripsi dan dekripsi dengan menggunakan kunci.



Gambar 2.1 Proses Enkripsi dan Dekripsi

2.5.2 Macam-Macam Algoritma Kriptografi

Kriptografi modern merupakan suatu perbaikan yang mengacu pada kriptografi klasik. Pada kriptografi modern terdapat berbagai macam algoritma yang dimaksudkan untuk mengamankan informasi yang dikirim melalui jaringan computer. Algoritma kriptografi modern terdiri dari dua bagian :

a. Kriptografi simetris
b. Kriptografi Asimetris

2.5.3 Kriptografi Kunci Simetris

Algoritma kunci simetris mengacun pada metode enkripsi dan dekripsi yang dalam hal ini memiliki kunci yang sama. Algoritma kunci simetri modern beroperasi dalam mode *bit* dan dapat dikelompokkan menjadi dua kategori:

a. *Cipher* aliran (*streamcipher*)
b. *Cipher* blok (*blok cipher*)

2.5.4 *Stream Cipher*

Algoritma kriptografi beroperasi pada teks asli (*plaintext*) atau teks acak (*ciphertext*) dalam bentuk bit tunggal, yang dalam hal ini rangkaian bit dienkripsikan atau didekripsikan bit per bit. Acak (*cipher*) aliran mengenkripsi satu bit setiap kali transformasi atau *byte* per *byte* (1 karakter = 1 *byte*). Nama lain *streamchiper* adalah *cipher* status sebab enkripsi tiap bit bergantung pada status.

2.5.5 Block *Cipher*

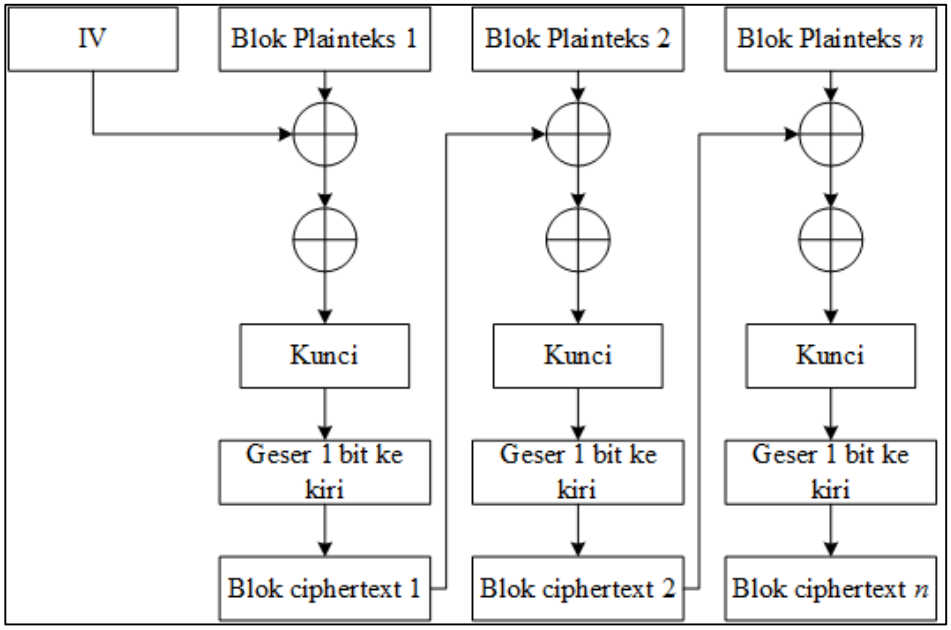
Algoritma kriptografi beroperasi pada teks asli (*plaintext*) atau teks acak (*ciphertext*) dalam bentuk blok bit, yang dalam hal ini rangkaian nit dibagi menjadi blok-blok bit yang panjangnya sudah ditentukan sebelumnya. Misalnya panjang blok adalah 64 bit, maka itu berarti algoritma enkripsi memerlukan 8 karakter setiap kali enkripsi (1 karakter = 8 bit dalam pengkodean ASCII). Acak (*cipher*) blok mengenkripsi satu blok bit setiap kali.

2.5.6 Algoritma *Cipher Block Chaining* (CBC)

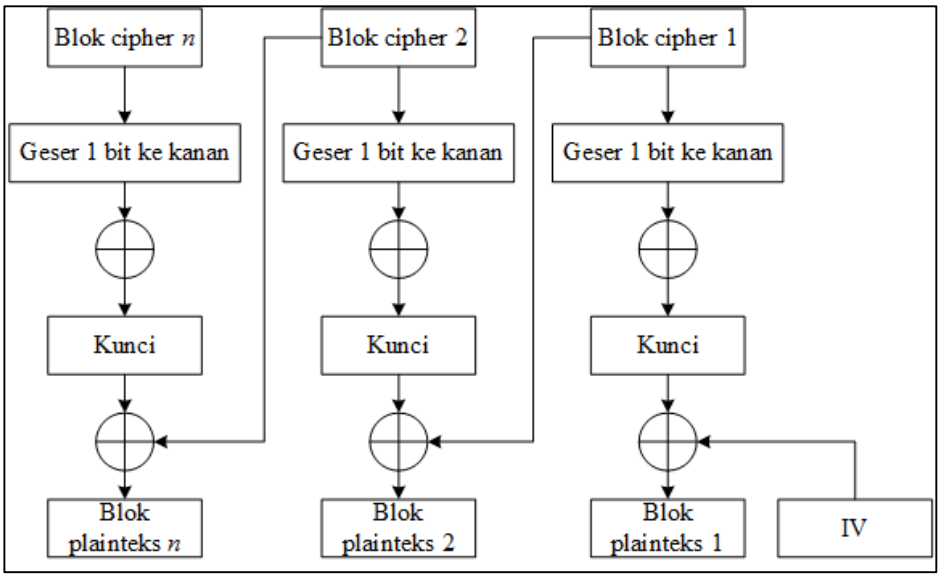
Algoritma *Cipher Block Chaining* merupakan penerapan mekanisme umpan balik pada sebuah blok *bit* dimana hasil enkripsi blok sebelumnya diumpan balikkan ke dalam proses enkripsi blok *current.* Caranya, blok *plaintext* yang *current* di-XOR-kan terlebih dahulu dengan blok *ciphertext* hasil enkripsi sebelumnya, selanjutnya hasil peng-XOR-an ini masuk ke dalam fungsi enkripsi. Dengan algoritma CBC, setiap blok

ciphertext tidak hanya bergantung pada blok *plaintext* tetapi juga pada seluruh blok *plaintext* sebelumnya.

Dekripsi dilakukan dengan memasukkan blok *ciphertext* yang *current* ke fungsi dekripsi, kemudian meng-XOR-kan hasilnya dengan blok *ciphertext* sebelumnya. Dalam hal ini, block *ciphertext* sebelumnya berfungsi sebagai umpan maju (*feedforward*) pada akhir proses deskripsi. (Rosmala, 2012).



Gambar 2.2 Skema Enkripsi dengan Algoritma CBC



Gambar 2.3 Skema Dekripsi dengan Algoritma CBC

2.6 Operasi XOR

Menurut (Munir, 2006) operasi XOR dapat digambarkan sebagai berikut :

a. Operator biner yang sering digunakan dalam *cipher* yang yang beroperasi dalam mode bit adalah *XOR* atau *exclusive-or*.
b. Notasi matematis untuk operator *XOR* adalah $\oplus$ (dalam Bahasa *C*, operator *XOR* dilambangkan dengan ^).
c. Operator *XOR* diperasikan pada dua bit dengan aturan sebagai berikut:

$$0 \oplus 0 = 0$$
$$0 \oplus 1 = 1$$
$$1 \oplus 0 = 1$$
$$1 \oplus 1 = 0$$

Perhatikan bahwa operator *XOR* identik dengan penjumlahan modulo 2:

$$0 + 0 \pmod 2 = 0$$
$$0 + 1 \pmod 2 = 1$$
$$1 + 0 \pmod 2 = 1$$
$$1 + 1 \pmod 2 = 0$$

d. Misalkan *a*, *b*, dan *c* adalah peubah Boolean. Hukum-hukum yang terkait dengan operator XOR:

(i) $a \oplus a = 0$

(ii) $a \oplus b = b \oplus a$ (Hukum komutatif)

(iii) $a \oplus (b \oplus c) = (a \oplus b) \oplus c$ (Hukum asosiatif)

e. Jika dua rangkaian dioperasikan dengan *XOR*, maka operasinya dilakukan dengan meng-*XOR*-kan setiap bit yang berkoresponden dari kedua rangkaian bit tersebut.

Contoh: 10011 ⊕ 11001 = 01010

yang dalam hal ini, hasilnya diperoleh sebagai berikut:

1	0	0	1	1	
1	1	0	0	1	⊕
1 ⊕ 1	0 ⊕ 1	0 ⊕ 0	1⊕ 0	1 ⊕ 1	
0	1	0	1	0	

f. Algoritma enkripsi sederhana yang menggunakan *XOR* adalah dengan meng-*XOR*-kan plainteks (*P*) dengan kunci (*K*) menghasilkan cipherteks:

$$C = P \oplus K$$

Karena meng-*XOR*-kan nilai yang sama dua kali berturut-turut menghasilkan nilai semula, maka dekripsi menggunakan persamaan:

$$P = C \oplus K$$

Contoh:

plainteks	01100101		(karakter ‘e’)
kunci	00110101	⊕	(karakter ‘5’)
cipherteks	01010000		(karakter ‘P’)
kunci	00110101	⊕	(karakter ‘5’)
plainteks	01100101		(karakter ‘e’)

2.7 *Personal Home Page* (PHP)

PHP adalah *script* untuk pemrograman *script web server-side*, *script* yang membuat dokumen HTML secara *on the fly*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML. Dengan menggunakan PHP maka *maintenance* suatu situs web menjadi lebih mudah. Proses *update* data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan *script* PHP.

PHP/FI merupakan nama awal dari PHP. PHP merupakan singkatan dari *Personal Home Page* dan FI adalah singkatan dari *Form Interface*. Dibuat pertama kali oleh Rasmus Lerdoff. PHP, awalnya merupakan program CGI yang dikhususkan untuk menerima input melalui *form* yang ditampilkan dalam *browser web*. Software ini disebarkan dan dilisensikan sebagai perangkat lunak *Open Source*. PHP secara resmi merupakan kependekan dari PHP: *HyperText Preprocessor*, merupakan bahasa *script server-side* yang disisipkan pada HTML (Rudianto, 2011).

2.8 *MySQL*

MySQL dikembangkan oleh perusahaan swedia bernama MySQL AB yang pada saat ini bernama Tcx DataKonsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak tahun 1979. Awalnya Tcx merupakan perusahaan pengembang software dan konsultan database, dan saat ini MySQL sudah diambil alih oleh Oracle Corp.

Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database*-nya sehingga mudah untuk digunakan, kinerja *query* cepat, dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan yang berskala kecil sampai menengah, MySQL juga bersifat *open source* (tidak berbayar) .

MySQL merupakan *database* yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan *software* pembangun aplikasi web yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis web, umumnya

pengembangan aplikasinya menggunakan bahasa pemrograman *script* PHP (Rudiyanto, 2011).

2.9 *ASCII*

Kode Standar Amerika untuk Pertukaran Informasi atau ASCII (*American Standard Code for Information Interchange*) merupakan suatu standar internasional dalam kode huruf dan simbol seperti *Hex* dan *Unicode* tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 7 *bit*. Namun, ASCII disimpan sebagai sandi 8 *bit* dengan menambakan satu angka 0 sebagai *bit significant* paling tinggi. *Bit* tambahan ini sering digunakan untuk uji prioritas. Karakter control pada ASCII dibedakan menjadi 5 kelompok sesuai dengan penggunaan yaitu berturut-turut meliputi *logical communication*, *Device control*, *Information separator*, *Code extention*, dan *physical communication*. Code ASCII ini banyak dijumpai pada papan ketik (*keyboard*) computer atau instrument-instrument digital.

Jumlah kode ASCII adalah 255 kode. Kode ASCII 0..127 merupakan kode ASCII untuk manipulasi teks; sedangkan kode ASCII 128..255 merupakan kode ASCII untuk manipulasi grafik. Kode ASCII sendiri dapat dikelompokkan lagi kedalam beberapa bagian:

- Kode yang tidak terlihat simbolnya seperti Kode 10(*Line Feed*), 13(*Carriage Return*), 8(Tab), 32(*Space*)
- Kode yang terlihat simbolnya seperti abjad (A..Z), numerik (0..9), karakter khusus (~!@#$%^&*()_+?:”{})
- Kode yang tidak ada di keyboard namun dapat ditampilkan. Kode ini umumnya untuk kode-kode grafik.

Dalam pengkodean kode ASCII memanfaatkan 8 *bit*. Pada saat ini kode ASCII telah tergantikan oleh kode UNICODE (*Universal Code*). UNICODE dalam pengkodeannya memanfaatkan 16 *bit* sehingga memungkinkan untuk menyimpan kode-kode lainnya seperti kode bahasa Jepang, Cina, Thailand dan sebagainya.

BAB III
METODE PENELITIAN

3.1 Metode Penelitian

1. *Study Literature*

   Study Literature dilaksanakan dengan cara mengumpulkan dan mempelajari segala macam informasi yang berhubungan dengan kriptografi dan segala yang berhubungan dengan metode *Cipher Block Chaining* (CBC).

2. Desain Sistem

   Pada tahap dilaksanakan perancangan sistem aplikasi yang akan dibuat berdasarkan hasil *study literature* yang ada. Perancangan sistem aplikasi ini meliputi desain antarmuka, desain akses user, dan pemograman. Perencanaan menggunakan bahasa pemrograman PHP.

3. Analisis Metode

   Analisis metode yaitu analisa yang dilakukan oleh penulis tentang metode yang dipakai untuk pengamanan data. Analisis metode yang dilakukan seperti alur perancangan sistem dan uji perhitungan manual metode.

   Pada proses enkripsi data, akan digunakan algoritma *Cipher Block Chaining* (CBC). Pada enkripsi *Cipher Block Chaining* (CBC) dimulai dengan proses membagi *plaintext* menjadi blok yang telah *ditentukan* ukurannya, pada sistem aplikasi ini tiap blok berukuran 8 bit. Satu blok *plaintext* yang telah dibagi di-XOR-kan dengan IV (*Initialization Vector*) yang telah ditentukan, kemudian hasil tersebut di-XOR-kan lagi dengan kunci. Hasil XOR tersebut digeser 1 *bit* ke kiri, hasil tersebut menjadi IV untuk blok berikutnya. Proses diulang sampai blok berakhir.

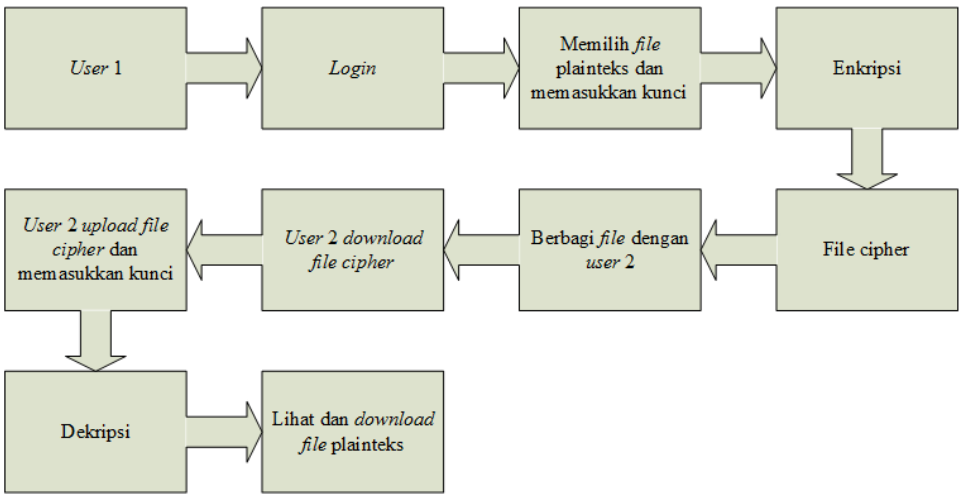
Untuk proses dekripsi, hal yang dilakukan adalah kebalikan dari proses enkripsi dengan algoritma *Cipher Block Chaining* (CBC).

3.2 Perancangan Sistem

Pada tahap ini akan dijelaskan bagaimana kerja aplikasi yang akan dibangun, sehingga sebelum dibangun, peneliti sudah bisa mendapatkan dugaan hasil yang akan diperoleh setelah terbangunnya aplikasi ini.

Tujuan dari perancangan sistem adalah untuk memenuhi kebutuhan pengguna mengenai gambaran yang jelas tentang perancangan sistem yang akan dibuat serta diimplementasikan. Untuk membangun suatu sistem aplikasi kriptografi, maka penulis terlebih dahulu merencanakan alur kerja yang akan dipergunakan oleh pengguna.

3.2.1 Diagram Blok

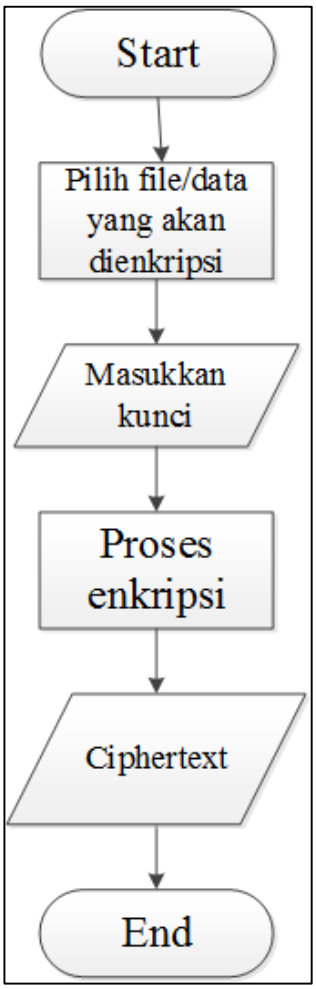


Gambar 3.1 Diagram blok aplikasi

Dari diagram diatas dapat dijelaskan, *User1* akan melakukan *login* untuk melakukan *upload file* dan memasukkan kunci untuk men-enkripsi *file*. Setelah didapatkan *file cipher*, *user 1* akan berbagi *file* dengan *user 2*. *User 2* menerima *file cipher* selanjutnya di-*download*. Kemudian *user 2* mengupload *file cipher* dan memasukkan kunci untuk melakukan proses dekripsi. Sehingga didapatkan *file plainteksnya*.

3.2.2 *Flowchart*

Enkripsi



Gambar 3.2 *Flowchart* Enkripsi

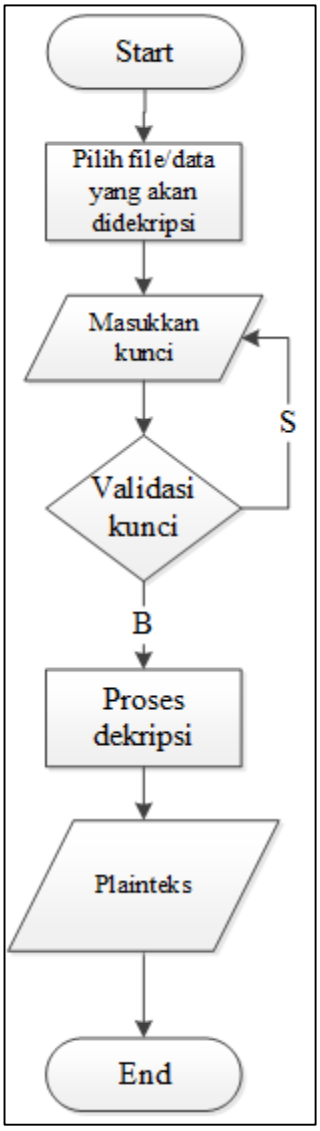
Pada *flowchart* enkripsi diatas dapat diamati :

1. Pertama data yang dipilih adalah data yang akan di enkripsi
2. Kedua memasukkan kunci yang akan digunakan untuk proses deskripsi
3. Ketiga dalah proses enkripsi, dimana:
   a. *Plaintext* dibagi dengan tiap blok berukuran 8 bit
   b. *Plaintext* kemudian di-XOR-kan dengan IV (*Initialization Vector*) yang ditentukan
   c. Hasil dari XOR kemudian di-XOR-kan lagi dengan kunci

d. Hasil peng-XOR-an yang terakhir kemudian digeser 1 bit ke kiri
e. Hasil tersebut menjadi IV untuk blok berikutnya
f. Proses diulang sampai blok berakhir

4. Keempat didapatkan *chipertext*.

Dekripsi



Gambar 3.3 *Flowchart* Deskripsi

Pada *flowchart* deskripsi, dapat diamati:

1. Pertama, *user* akan memilih data untuk dekripsi

2. Kedua, *user* memasukkan kunci yang akan digunakan untuk proses deskripsi
3. Ketiga, jika kunci salah maka akan diminta untuk memasukkan kunci lagi
4. Keempat, adalah proses deskripsi, dimana:
   a. Proses dimulai dari blok paling akhir
   b. Hasil pembagian blok digeser 1 bit ke kanan
   c. Hasil penggeseran di-XOR-kan dengan kunci
   d. Hasil peng-XOR-an dengan kunci kemudian di-XOR-kan lagi dengan blok *chipertext* sebelumnya
   e. Proses diulang sampai blok paling awal, blok paling awal di-XOR-kan dengan IV.
5. Kelima, didapatkan *plaintext.*

3.2.3 Proses Enkripsi Karakter

Proses enkripsi dimulai dari menkorversikan karakter dalam plainteks kedalam bentuk *binary* agar dapat dengan mudah melakukan operasi XOR. Contoh enkripsi dengan algoritma *Cipher Block Chaining* :

Plainteks (P) : HARISETIADY
IV : K (01001011)
Kunci (K) : V (01010110)

Di bawah ini adalah hasil konversi dari plainteks “HARISETIADY” kedalam bentuk *binary*.

P1 P2 P3 P4 P5
P = 01001000,01000001,01010010,01001001,01010011,
P6 P7 P8 P9 P10
01000101,01010100,01001001,01000001,01000100,
P11
01011001

Di bawah ini adalah hasil operasi XOR dari *plaintext* dengan IV yang kemudian hasil dari blok sebelumnya menjadi IV untuk *block* berikutnya sehingga menghasilkan *ciphertext*.

C_1 = $P_1 \oplus IV$

= 01001000 $\oplus$ 01001011

= 00000011

$C_1 \oplus$ Kunci

= 00000011 $\oplus$ 01010110

= 01010101

C_1 digeser 1 *bit* ke kiri = **10101010**

C_2 = $P_2 \oplus C_1$

= 01000001 $\oplus$ 10101010

= 11101011

$C_2 \oplus$ Kunci

= 11101011 $\oplus$ 01010110

= 10111101

C_2 digeser 1 *bit* ke kiri = **01111011**

C_3 = $P_3 \oplus C_2$

= 01010010 $\oplus$ 01111011

= 00101001

$C_3 \oplus$ Kunci

= 00101001 $\oplus$ 01010110

= 01111111

C_3 digeser 1 *bit* ke kiri = **11111110**

C_4 = $P_4 \oplus C_3$

= 01001001 $\oplus$ 11111110

= 10110111

$C_4 \oplus$ Kunci

= 10110111 $\oplus$ 01010110

= 11100001

C_4 digeser 1 *bit* ke kiri = **11000011**

C_5 = $P_5 \oplus C_4$

= 01010011 $\oplus$ 11000011

= 10110010

$C_4 \oplus$ Kunci

= 10110010 $\oplus$ 01010110

= 11100100

C_4 digeser 1 *bit* ke kiri = **11001001**

C_6 = $P_6 \oplus C_5$

= 01000101 $\oplus$ 11001001

= 10001100

$C_6 \oplus$ Kunci

= 10001100 $\oplus$ 01010110

= 11011010

C_6 digeser 1 *bit* ke kiri = **10110101**

C_7 = $P_7 \oplus C_6$

= 01010100 $\oplus$ 10110101

= 11100001

$C_7 \oplus$ Kunci

= 11100001 $\oplus$ 01010110

= 10110111

C_7 digeser 1 *bit* ke kiri = **01101111**

C_8 = $P_8 \oplus C_7$

= 01001001 $\oplus$ 01101111

= 00100110

$C_8 \oplus$ Kunci

= 00100110 $\oplus$ 01010110

= 01110000

C_8 digeser 1 *bit* ke kiri = **11100000**

C_9 = $P_9 \oplus C_9$

= 01000001 $\oplus$ 11100000

= 10100001

C9 ⊕ Kunci

= 10100001 ⊕ 01010110

= 11110111

C9 digeser 1 *bit* ke kiri = **11101111**

C10 = C10 ⊕ C9

= 01000100 ⊕ 11101111

= 10101011

C10 ⊕ Kunci

= 10101011 ⊕ 01010110

= 11111101

C10 digeser 1 *bit* ke kiri = **11111011**

C11 = P11 ⊕ C9

= 01011001 ⊕ 11111011

= 10100010

C11 ⊕ Kunci

= 10100010 ⊕ 01010110

= 11110100

C11 digeser 1 *bit* kekiri = **11101001**

Dari perhitungan proses enkripsi di atas menghasilkan *chiphertext* "**ª{þ ÃÉµoàïûé**"

3.2.4 Proses Dekripsi Karakter

Proses dekripsi dimulai dari menkorversi karakter *ciphertext* ke dalam bentuk *binary* agar mudah untuk melakukan operasi XOR. Berikut proses dekripsi dari *ciphertext* di atas :

Cipherteks (C): ª{þ ÃÉµoàïûé

Kunci (K) : V (01010110)

IV : K (01001011)

Pertama *ciphertext* akan dikonversi ke dalam bentuk *binary* kemudian dibagi menjadi beberapa blok dan digeser 1 *bit* ke kanan.

C_1 C_2 C_3 C_4 C_5 C_6

C = 01010101,10111101,01111111,11100001,11100100,11011010

C_7 C_8 C_9 C_{10} C_{11}

10110111,01110000,11110111,11111101,11110100

Setelah *ciphertext* dibagi menjadi beberapa blok, proses dimulai dari blok paling belakang. Blok paling akhir di XOR kan dengan kunci hasilnya di XOR kan dengan blok *cipher* sebelumnya hingga blok paling depan di XOR kan dengan IV.

P_{11} = $C_{11} \oplus$ Kunci

= 11110100 $\oplus$ 01010110

= 10100010

$P_{11} \oplus$ Blok *cipher* sebelumnya

= 10100010 $\oplus$ 11111011

= **01011001**

P_{10} = $C_{10} \oplus$ Kunci

= 11111101 $\oplus$ 01010110

= 10101011

$P_{10} \oplus$ Blok *cipher* sebelumnya

= 10101011 $\oplus$ 11101111

= **01000100**

P_9 = $C_9 \oplus$ Kunci

= 11110111 $\oplus$ 01010110

= 10100001

$P_9 \oplus$ Blok *cipher* sebelumnya

= 10100001 $\oplus$ 11100000

= **01000001**

P_8 = $C_8 \oplus$ Kunci

= 01110000 $\oplus$ 01010110

= 00100110

$P_8 \oplus$ Blok *cipher* sebelumnya

= 00100110 $\oplus$ 01101111

= **01001001**

P_7 = $C_7 \oplus$ Kunci

= 10110111 $\oplus$ 01010110

= 11100001

$P_7 \oplus$ Blok *cipher* sebelumnya

= 11100001 $\oplus$ 10110101

= **01010100**

P_6 = $C_6 \oplus$ Kunci

= 11011010 $\oplus$ 01010110

= 10001100

$P_6 \oplus$ Blok *cipher* sebelumnya

= 10001100 $\oplus$ 11001001

= **01000101**

P_5 = $C_5 \oplus$ Kunci

= 11100100 $\oplus$ 01010110

= 10110010

$P_5 \oplus$ Blok *cipher* sebelumnya

= 10110010 $\oplus$ 11000011

= **01110001**

P_4 = $C_4 \oplus$ Kunci

= 11100001 $\oplus$ 01010110

= 10110111

$P_4 \oplus$ Blok *cipher* sebelumnya

= 10110111 $\oplus$ 11111110

= **01001001**

P_3 = $C_3 \oplus$ Kunci

= 01111111 $\oplus$ 01010110

= 00101001

$P_3 \oplus$ Blok *cipher* sebelumnya

= 00101001 $\oplus$ 01111011

= **01010010**

P_2 = $C_2 \oplus$ Kunci

= 10111101 $\oplus$ 01010110

= 11101011

$P_2 \oplus$ Blok *cipher* sebelumnya

= 11101011 $\oplus$ 10101010

= **01000001**

P_1 = $C_1 \oplus$ Kunci

= 01010101 $\oplus$ 01010110

= 00000011

$P_1 \oplus$ IV

= 00000011 $\oplus$ 01001011

= **01001000**

Setelah dilakukan proses dekripsi diatas maka didapatkan *plaintext* nya adalah “HARISETIADY”.

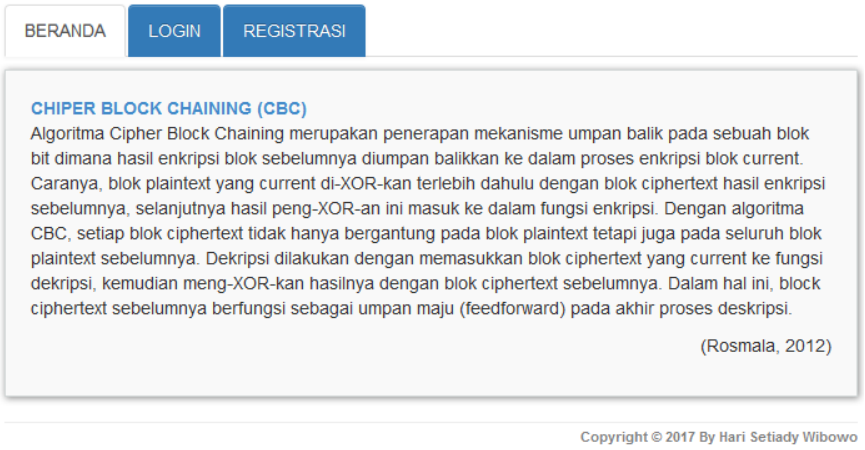
BAB IV
IMPLEMENTASI DAN PENGUJIAN

Dalam pembuatan suatu sistem, implementasi dan pengujian sangat diperlukan untuk mengetahui seberapa baik sistem yang telah dibuat.

4.1 Metode Pengujian

4.1.1 Pengujian Antarmuka

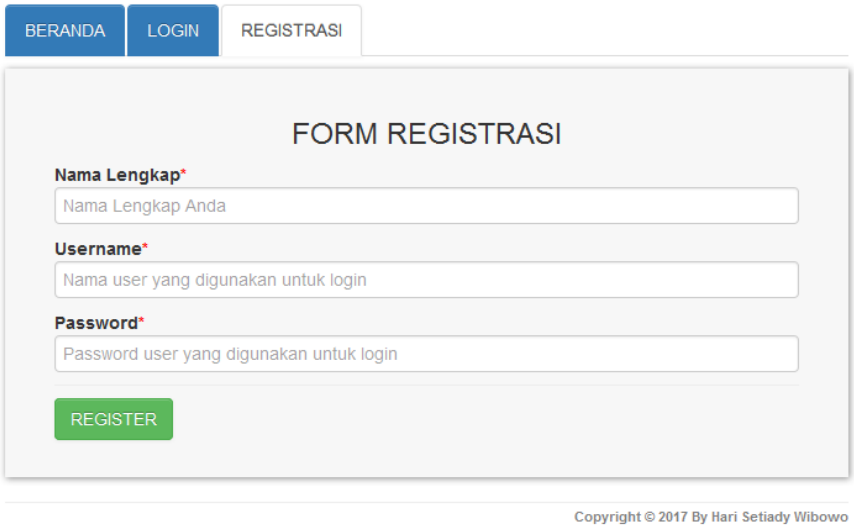
Halaman Beranda



`Gambar 4.1 Halaman Beranda

Pada halaman beranda ini terdapat penjelasan mengenai metode *cipher block chaining* yang dikutip dari Dewi Rosmala pada tahun 2012.

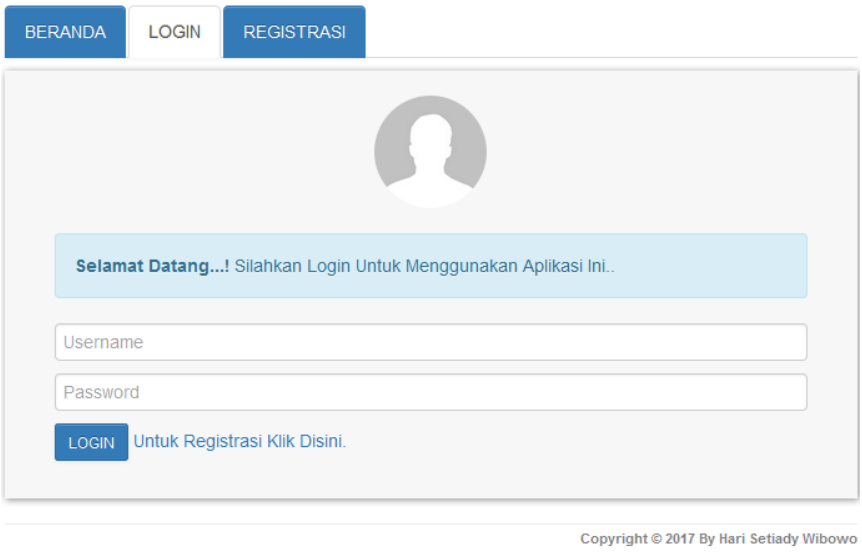
Halaman Registrasi



Gambar 4.2 Halaman Registrasi

Pada halaman ini *user* mengisi *form* registrasi Nama Lengkap, *Username* dan *Password* untuk mendapatkan akun *login.*

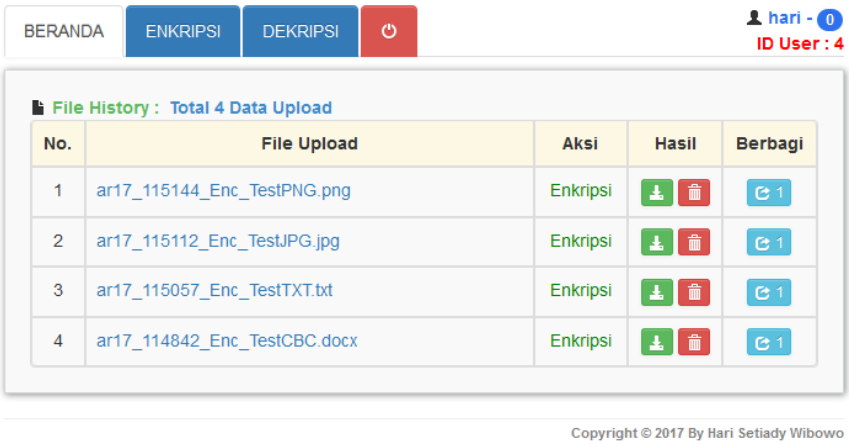
Halaman *Login*



Gambar 4.3 Halaman *Login*

Pada halaman ini, *user* atau pengguna harus melakukan login dengan menggunakan *username* dan *password* yang terdaftar. Bila *user* atau pengguna belum terdaftar, silahkan mendaftar.

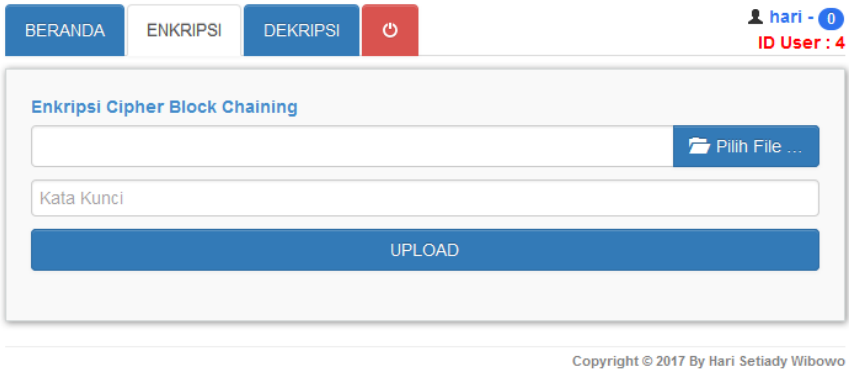
Halaman *User*



Gambar 4.4 Halaman *User*

Setelah berhasil melakukan *login*, *user* akan masuk ke halaman beranda *user*. Pada menu beranda ini terdapat *username*, *ID User* dan tabel *file history*. *ID User* digunakan untuk berbagi file. *File history* disini menampilka *history file* apa saja yang sudah di *upload* untuk di enkripsi atau dekripsi *user* tersebut. Dalam tabel ini *user* juga bisa melihat *file* yang telah terenkripsi, men-*download file*, menghapus *file* dan berbagi *file* ke *user* lain.

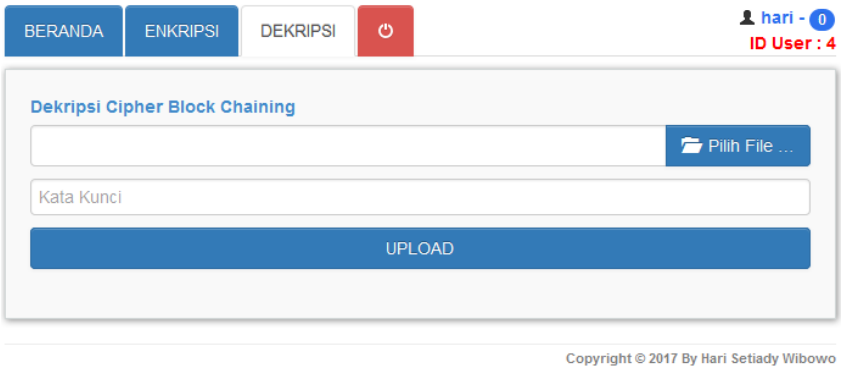
Halaman Enkripsi



Gambar 4.5 Halaman Enkripsi

Pada menu Enkripsi, *user* mengupload *file* yang akan di enkripsi. Kemudian mengisi kunci untuk melakukan proses *upload* dan enkripsi.

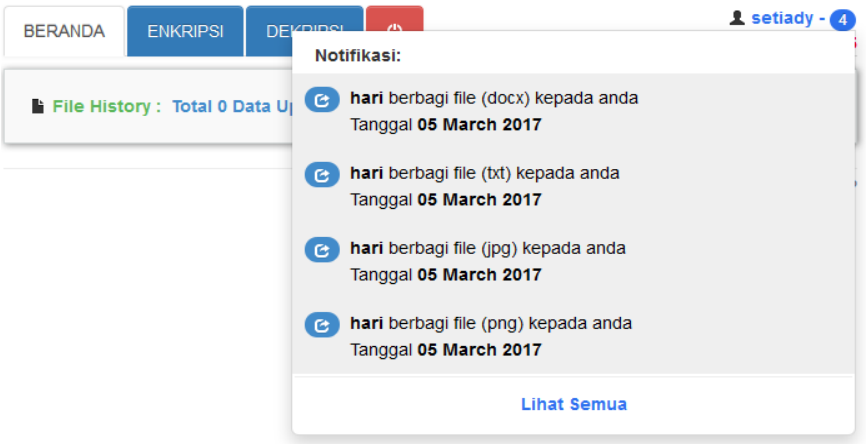
Halaman Dekripsi



Gambar 4.6 Halaman Dekripsi

Tampilan pada halaman dekripsi ini hampir sama dengan halaman enkripsi, dimana *user* memilih *file* yang akan didekripsi dan memasukkan kunci *file* untuk di dekripsi.

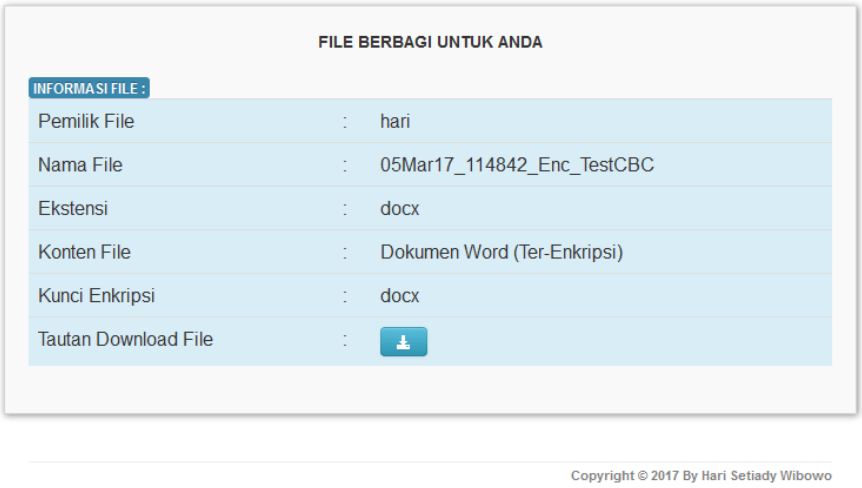
Notifikasi *User*



Gambar 4.7 Notifikasi *User*

Pada notifikasi ini user dapat melihat *file* apa saja yang telah dibagikan.

Detail *File*

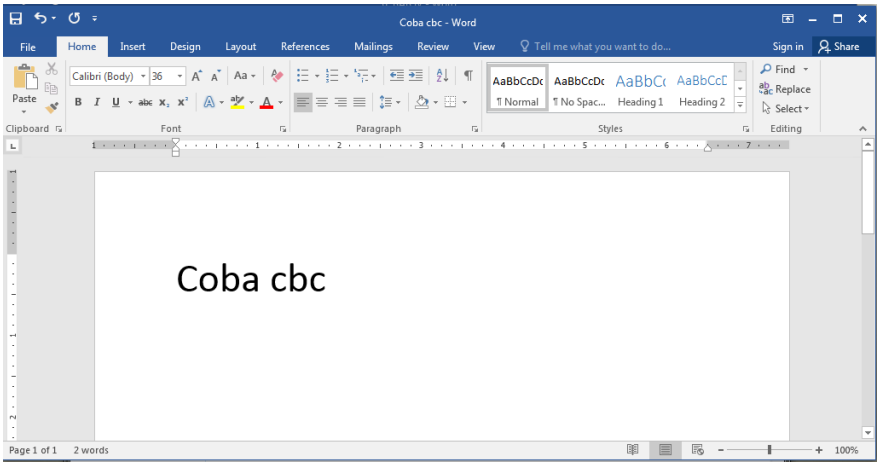


Gambar 4.8 Detail *File*

Setelah *user* mengklik notifikasi *file* yang dipilih, akan muncul *detail* pemilik *file*(yang membagikan *file*), Nama *file*, Ekstensi, Konten *file,* Kunci enkripsi, dan Tautan *download file*(terdapat tombol untu men-*download file*).

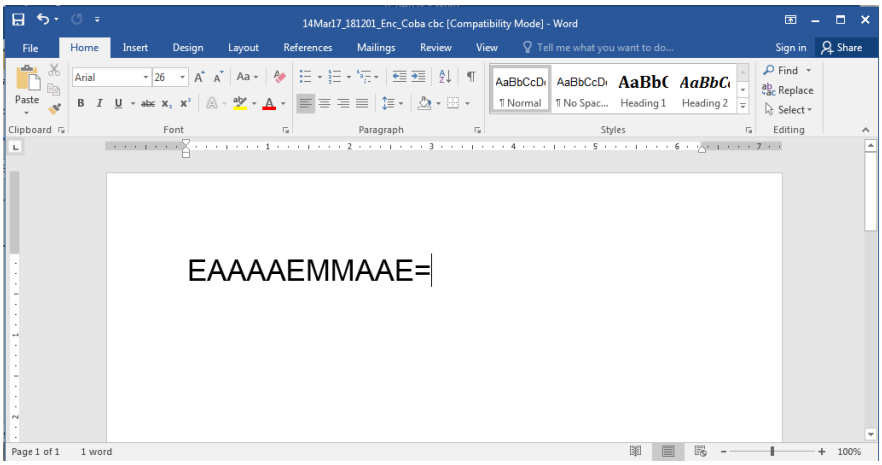
4.1.2 Pengujian Enkripsi

***File* ekstensi *.docx**



Gambar 4.9 *Plaintext* pengujian enkripsi *.docx

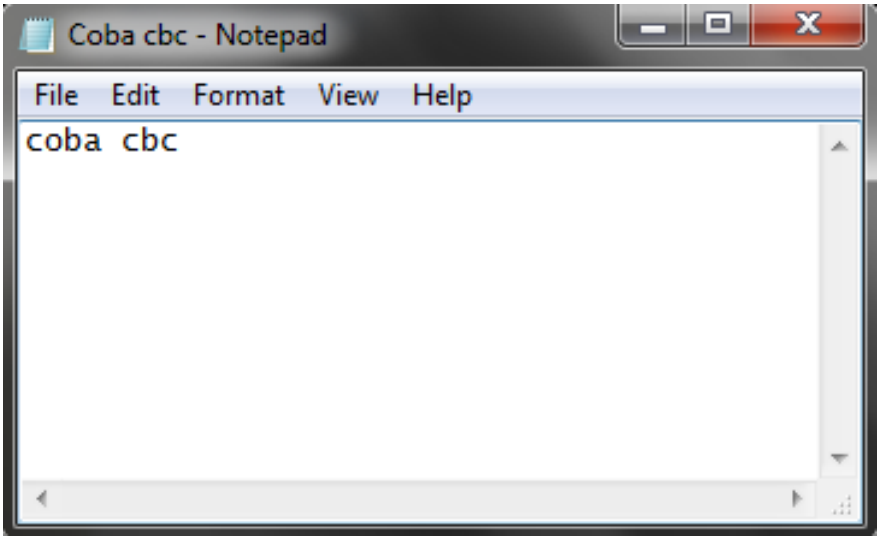
File plainteks Coba cbc.docx sebelum dienkripsi memiliki isi "**Coba cbc**"



Gambar 4.10 *Ciphertext* pengujian enkripsi *.docx

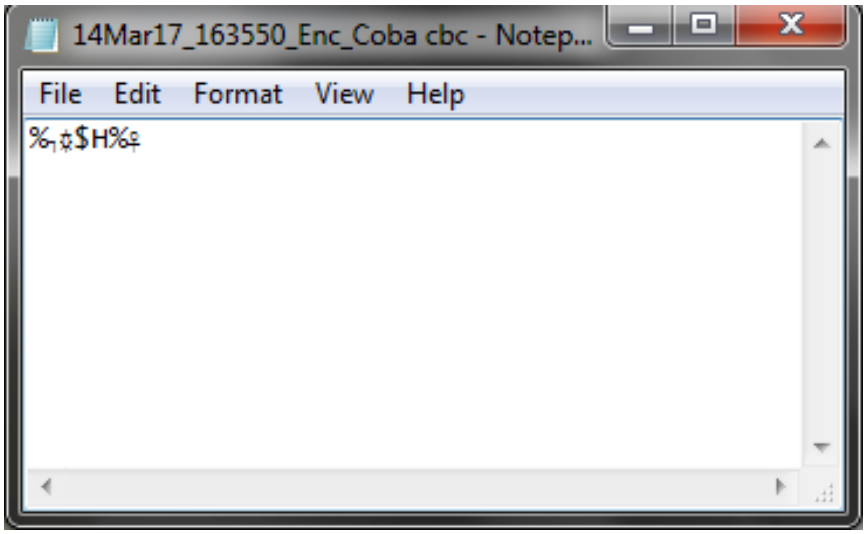
Setelah di enkripsi akan menghasilkan *file cipher* 14Mar17_181201_Enc_Coba cbc.docx yang memiliki isi "**EAAAAEMMAAE=**".

***File* ekstensi *.txt**



Gambar 4.11 *Plaintext* pengujian enkripsi *.txt

Sebelum dienkripsi *Plaintext* Coba cbc.txt ini berisi "**coba cbc**"



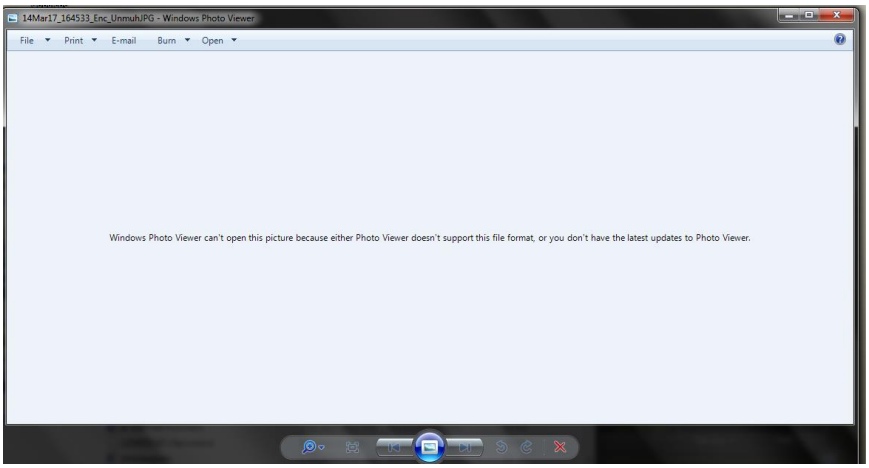
Gambar 4.12 *Ciphertext* pengujian enkripsi *.txt

Setelah dienkripsi akan menghasilkan *file cipher* 14Mar17_163550_Enc_Coba cbc.txt yang berisi "**%$H%**".

***File* *.jpg**

Gambar 4.13 *Plaintext* pengujian enkripsi *.jpg

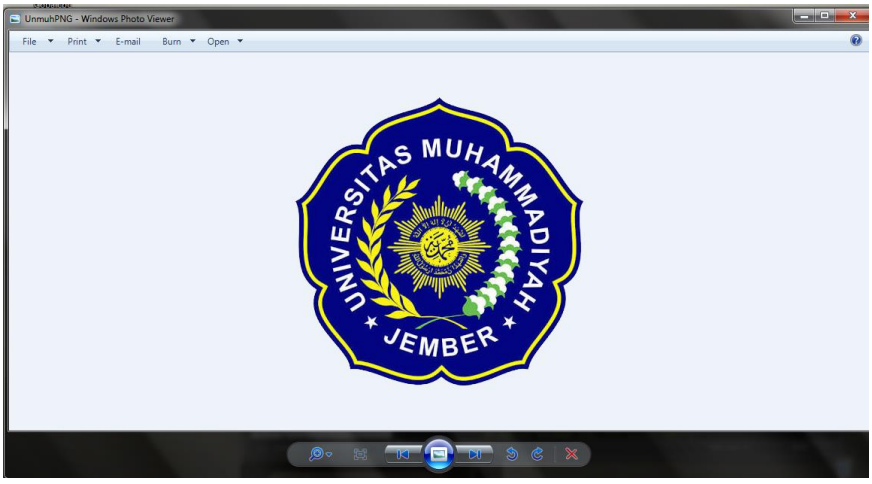
Plaintext UnmuhJPG.jpg sebelum dienkripsi memiliki isi gambar logo Universitas Muhammadiyah Jember.



Gambar 4.14 *Ciphertext* pengujian enkripsi *.jpg

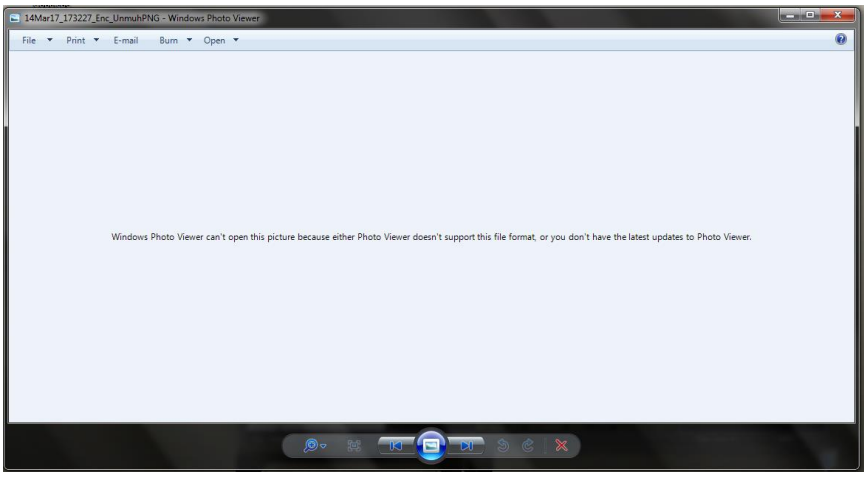
Setelah dienkripsi menghasilkan *file cipher* dengan nama 14Mar17_164533_Enc_UnmuhJPG.jpg yang tidak bisa dilihat gambarnya.

***File* *.PNG**



Gambar 4.15 *Plaintext* pengujian enkripsi *.png

Plaintext sebelum dienkripsi adalah UnmuhPNG.png yang memiliki isi berupa gambar logo Universitas Muhammadiyah Jember.

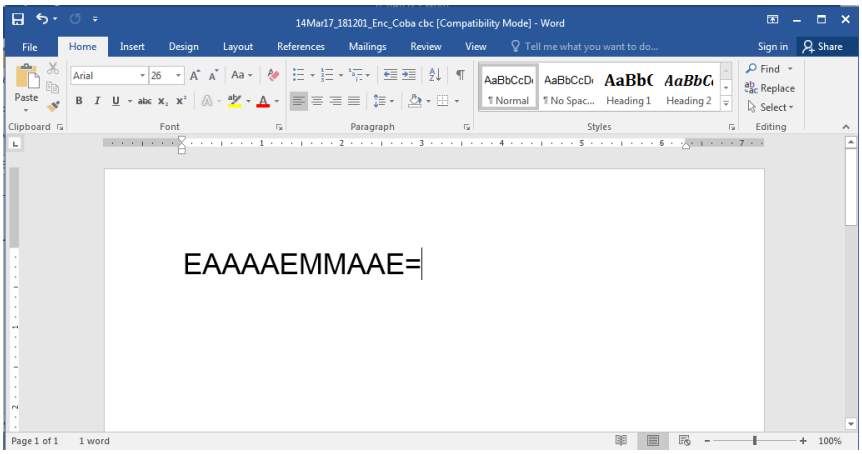


Gambar 4.16 *ciphertext* pengujian enkripsi *.png

Setelah dienkripsi, *file* UnmuhPNG.png akan menjadi *file cipher* 14Mar17_173227_Enc_UnmuhPNG.png yang tidak dapat dilihat isi gambarnya.

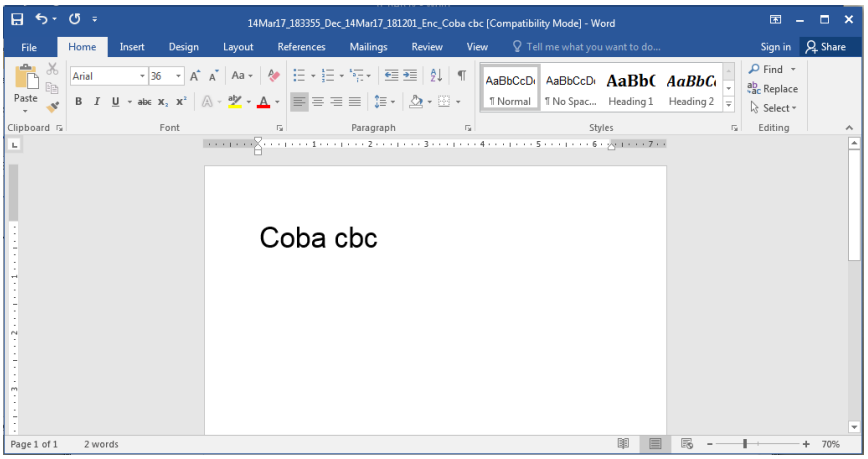
4.1.3 Pengujian Dekripsi

***File* ekstensi *.docx**



Gambar 4.17 *Ciphertext* pengujian dekripsi *.docx

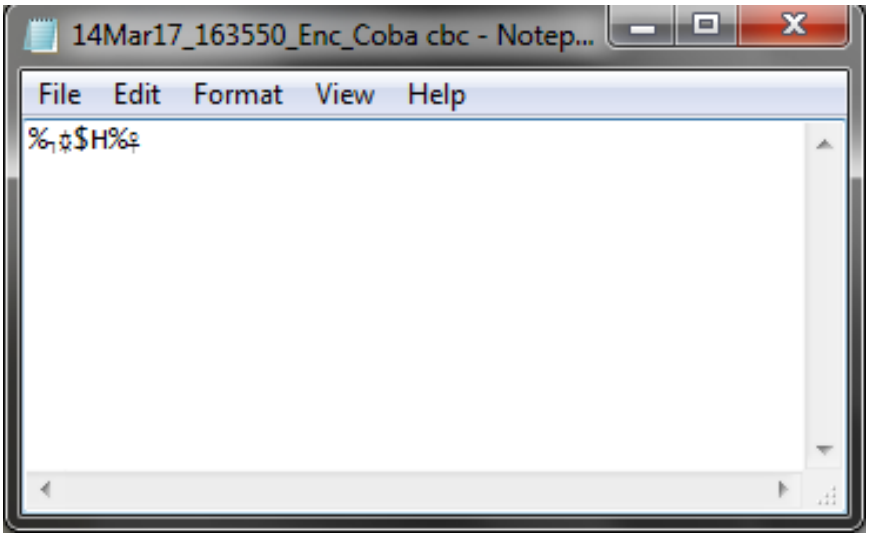
Sebelum didekripsi *file cipher* berisi "**EAAAAEMMAAE=**". Dan memiliki nama *file* 14Mar17_181201_Enc_Coba cbc.docx.



Gambar 4.18 *Plaintext* pengujian dekripsi *.docx

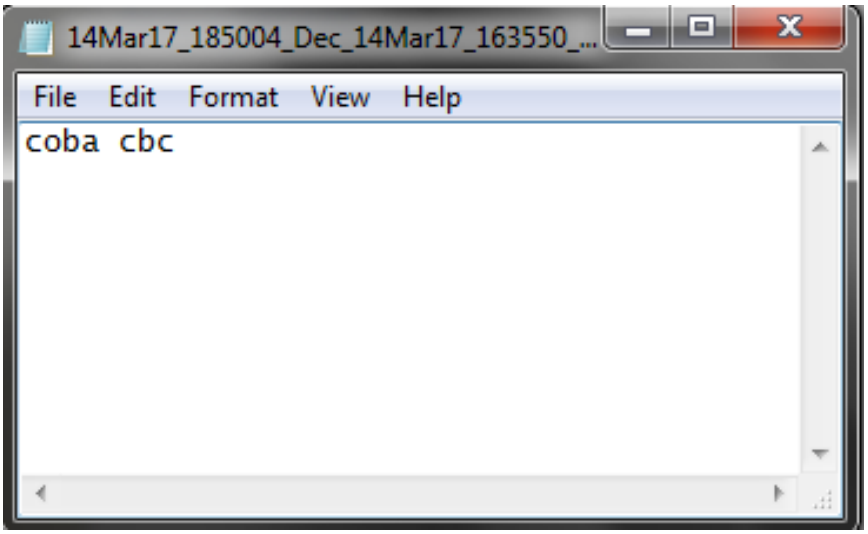
Setelah didekripsi *file* 14Mar17_181201_Enc_Coba cbc.docx ini menghasilkan *file* bernama 14Mar17_183355_ Dec_14Mar17_181201 _Enc_Coba cbc.docx yang memiliki isi "**Coba cbc**".

***File* ekstensi *.txt**



Gambar 4.19 *Ciphertext* pengujian dekripsi *.txt

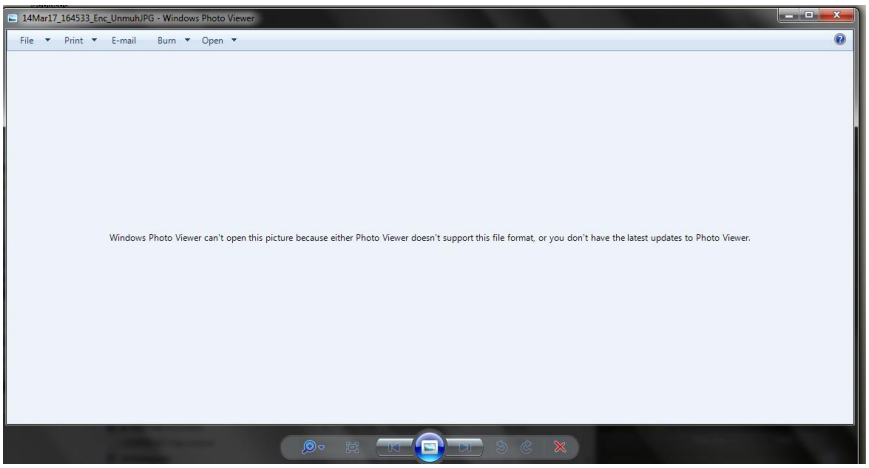
Sebelum didekripsi, *file cipher* 14Mar17_163550_Enc_Coba cbc.txt ini berisi "**%$H%**".



Gambar 4.20 *Plaintext* pengujian dekripsi *.txt

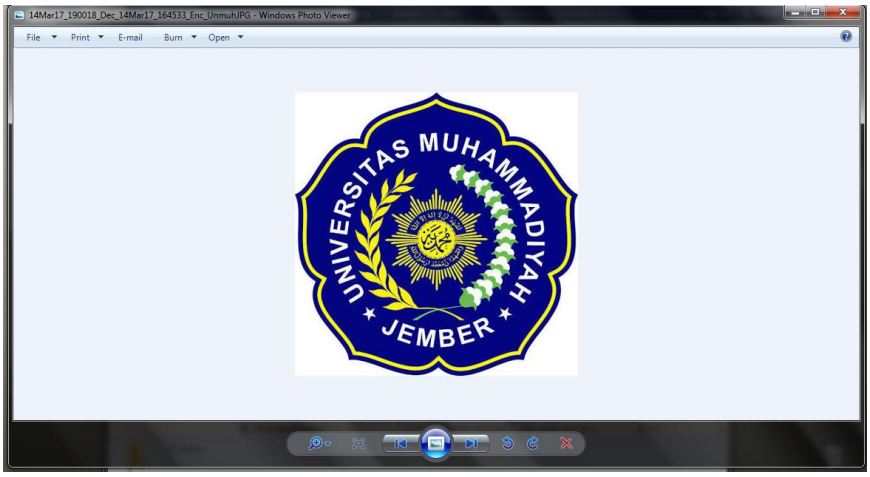
Setelah didekripsi *file* 14Mar17_163550_Enc_Coba cbc.txt berganti nama menjadi 14Mar17_185004_Dec_14Mar17_ 163550_Enc_Coba cbc.txt yang berisi "**coba cbc**".

***File* ekstensi *.jpg**



Gambar 4.21 *Ciphertext* pengujian dekripsi *.jpg

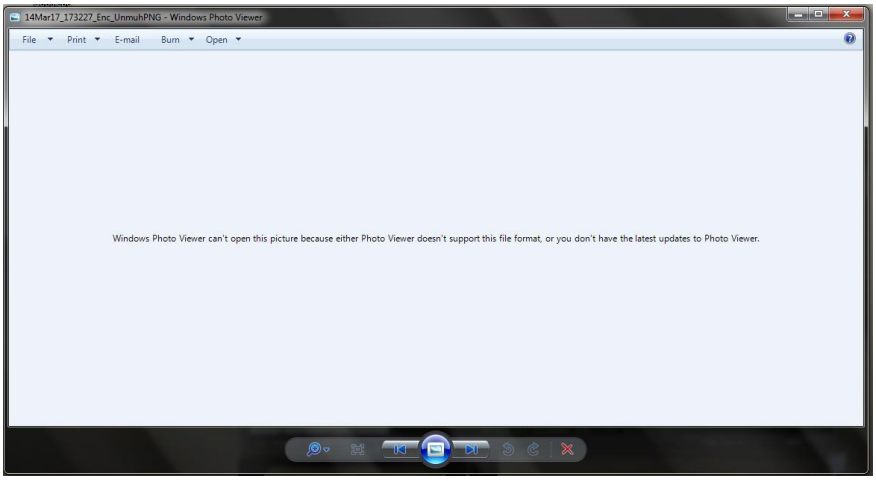
Sebelum didekripsi isi dari *file* 14Mar17_164533_Enc_ UnmuhJPG.jpg tidak dapat terlihat.



Gambar 4.22 *Plaintext* pengujian dekripsi *.jpg

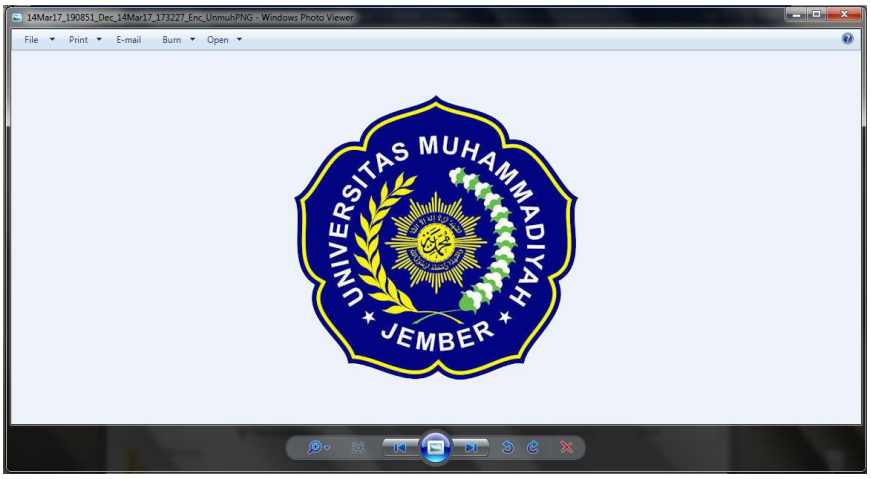
Setelah didekripsi, *file* yang tadinya memiliki nama 14Mar17_ 164533_Enc_UnmuhJPG.jpg berubah menjadi 14Mar17_190018_Dec_ 14Mar17_ 164533_Enc_UnmuhJPG.jpg dan gambar hasil dekripsi dapat terlihat.

***File* ekstensi *.png**



Gambar 4.23 *Ciphertext* pengujian dekripsi *.png

Sebelum melalui proses dekripsi, *file* 14Mar17_173227_ Enc_UnmuhPNG.png isi gambar tidak dapat dilihat.



Gambar 4.24 *Plaintext* pengujian dekripsi *.png

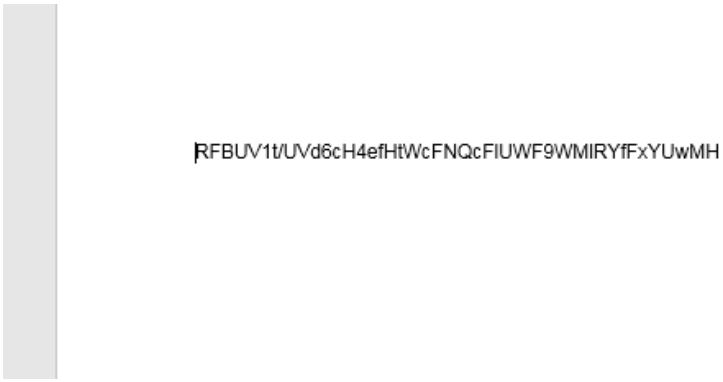
Setelah didekripsi, *file* yang tadinya meiliki nama 14Mar17_ 173227_ Enc_UnmuhPNG.png, berganti nama menjadi 14Mar17_190851_Dec_14Mar17_ 173227_Enc_UnmuhPNG.png dan gambar dari *file* dapat terlihat.

4.1.4 Pengujian *file* *.docx berisi gambar

Pada tahap pengujian ini akan dilakukan proses enkripsi dan dekrispsi pada *file* ber-ekstensi *.docx yang mengandung unsur gambar sehingga didapatkan hasil pengujian sebagai berikut :

Gambar 4.25 *Plaintext* berisi gambar

Pada *plaintext* pengujian enkripsi di atas berisi gambar logo Universitas Muhammadiyah Jember dan kalimat "Universitas Muhammadiyah Jember".



Gambar 4.26 *Ciphertext* setelah dienkripsi

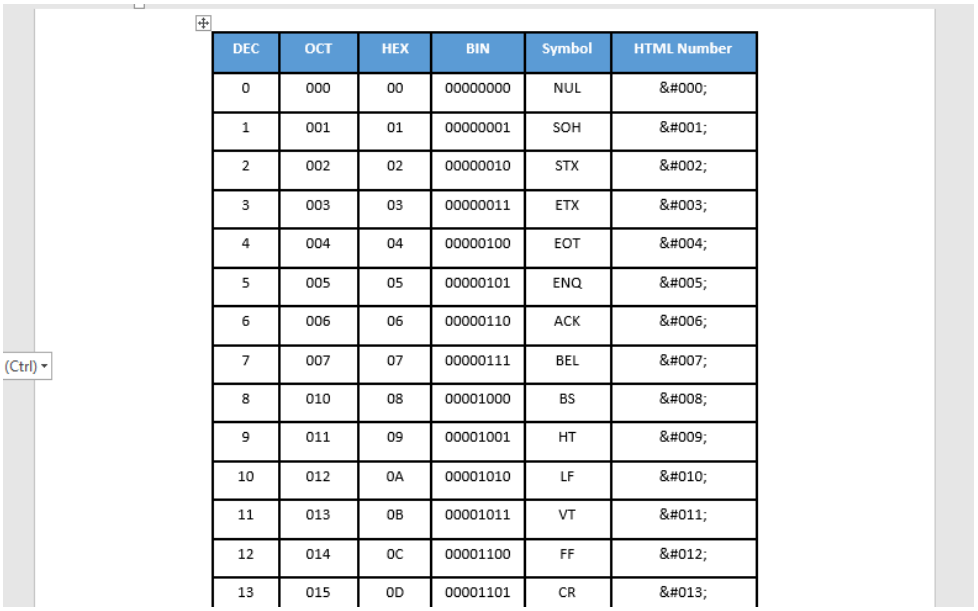
Hasil proses enkripsi dari *file* *.docx yang mengandung unsur gambar adalah *file cipher* yang berisi kalimat "RFBUV1t/UVd6cH4efHtWcFNQcFlUWF9WMlRYfFxYUwMH"

Gambar 4.27 Hasil dekripsi

Setelah dilakukan proses dekripsi, *plaintext* yang semula terdapat logo Universitas Muhammadiyah Jember dan kalimat "Universitas Muhammadiyah Jember" hanya akan mengembalikan kalimatnya saja.

4.1.5 Pengujian *file* *.docx berisi tabel

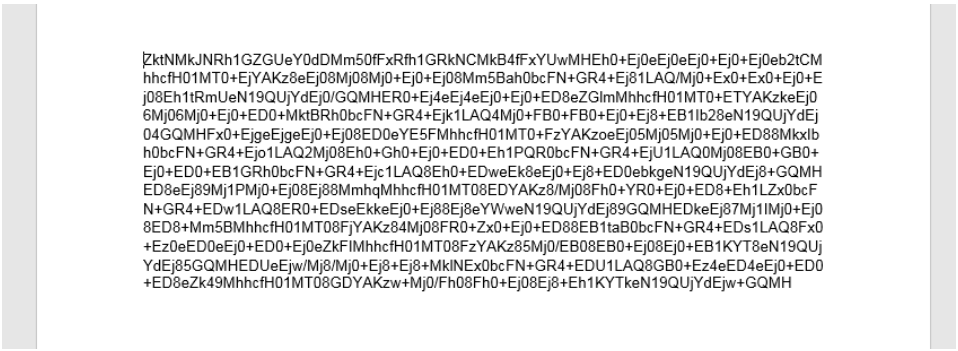
Pada tahap pengujian ini, akan dilakukan proses enkripsi dan dekripsi pada *file* ber-ekstensi *.docx yang mengandung unsur tabel sehingga didapatkan hasil sebagai berikut :



DEC	OCT	HEX	BIN	Symbol	HTML Number
0	000	00	00000000	NUL	&#000;
1	001	01	00000001	SOH	&#001;
2	002	02	00000010	STX	&#002;
3	003	03	00000011	ETX	&#003;
4	004	04	00000100	EOT	&#004;
5	005	05	00000101	ENQ	&#005;
6	006	06	00000110	ACK	&#006;
7	007	07	00000111	BEL	&#007;
8	010	08	00001000	BS	&#008;
9	011	09	00001001	HT	
10	012	0A	00001010	LF
11	013	0B	00001011	VT	&#011;
12	014	0C	00001100	FF	
13	015	0D	00001101	CR

Gambar 4.28 *Plaintext* berisi tabel

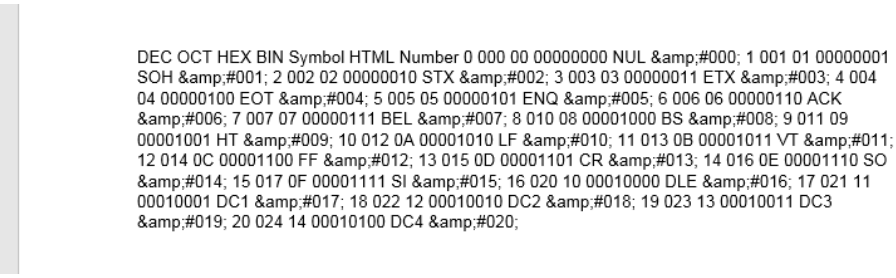
Pada *plaintext* pengujian *file* *.docx yang mengandung unsur tabel diatas, memiliki isi *file* sebagian tabel ASCII.



ZktNMkJNRh1GZGUeY0dDMm50fFxRfh1GRkNCMkB4fFxYUwMHEh0+Ej0eEj0eEj0+Ej0+Ej0eb2tCM
hhcfH01MT0+EjYAKz8eEj08Mj08Mj0+Ej0+Ej08Mm5Bah0bcFN+GR4+Ej81LAQ/Mj0+Ex0+Ex0+Ej0+E
j08Eh1tRmUeN19QUjYdEj0/GQMHER0+Ej4eEj4eEj0+Ej0+ED8eZGImMhhcfH01MT0+ETYAKzkeEj0
6Mj06Mj0+Ej0+ED0+MktBRh0bcFN+GR4+Ejk1LAQ4Mj0+FB0+FB0+Ej0+Ej8+EB1Ib28eN19QUjYdEj
04GQMHFx0+EjgeEjgeEj0+Ej08ED0eYE5FMhhcfH01MT0+FzYAKzoeEj05Mj05Mj0+Ej0+ED88Mkxlb
h0bcFN+GR4+Ejo1LAQ2Mj08Eh0+Gh0+Ej0+ED0+Eh1PQR0bcFN+GR4+EjU1LAQ0Mj08EB0+GB0+
Ej0+ED0+EB1GRh0bcFN+GR4+Ejc1LAQ8Eh0+EDweEk8eEj0+Ej8+ED0ebkgeN19QUjYdEj8+GQMH
ED8eEj89Mj1PMj0+Ej08Ej88MmhqMhhcfH01MT08EDYAKz8/Mj08Fh0+YR0+Ej0+ED8+Eh1LZx0bcF
N+GR4+EDw1LAQ8ER0+EDseEkkeEj0+Ej88Ej8eYWweN19QUjYdEj89GQMHEDkeEj87Mj1IMj0+Ej0
8ED8+Mm5BMhhcfH01MT08FjYAKz84Mj08FR0+Zx0+Ej0+ED88EB1taB0bcFN+GR4+EDs1LAQ8Fx0
+Ez0eED0eEj0+ED0+Ej0eZkFIMhhcfH01MT08FzYAKz85Mj0/EB08EB0+Ej08Ej0+EB1KYT8eN19QUj
YdEj85GQMHEDUeEjw/Mj8/Mj0+Ej8+Ej8+MklNEx0bcFN+GR4+EDU1LAQ8GB0+Ez4eED4eEj0+ED0
+ED8eZk49MhhcfH01MT08GDYAKzw+Mj0/Fh08Fh0+Ej08Ej8+Eh1KYTkeN19QUjYdEjw+GQMH

Gambar 4.29 *Ciphertext* setelah dienkripsi

Setelah dilakukan proses enkripsi pada *plaintext* yang mengandung unsur tabel, menghasilkan *file cipher* seperti pada gambar 4.29.



Gambar 4.30 Hasil dekripsi

Setelah dilakukan proses dekripsi pada *file cipher* (gambar 4.29) dihasilkan *file* *.docx seperti pada gambar 4.30 di atas.

4.1.6 Pengujian Akurasi Dekripsi Kalimat

Pada tahap ini akan dilakukan pengujian terhadap 3 skenario di mana skenario pertama terdiri dari 10 data yang diuji, setiap data terdiri dari 10 kata. Skenario 2 terdiri dari 10 data uji, setiap data terdiri dari 50 kata dan skenario 3 terdiri dari 10 data uji, setiap data terdiri dari 150 kata. Dengan pengujian melakukan proses enkripsi pada data yang diuji, kemudian mendekripsinya kembali ke dalam bentuk *plaintext.* Hasil dekripsi kemudian membandingkan karakter, tanda baca dan spasi dengan *plaintext* awal sebelum dilakukan proses enkripsi.

Pengujian pada skenario 1

Tabel 4.1 Pengujian pada skenario 1

No	Data uji	Hasil enkripsi	Hasil dekripsi	Ket
1	Kerahasiaan dan keamanan data adalah hal yang sangat penting dalam	i[• pU\QW\pP-v _S2VXpS\ • _S2 Y\V_-pY\~_V2U \~tpPY2~\ • Z\V~ tPzxPY2Y\~_P	Kerahasiaan dan keamanan data adalah hal yang sangat penting dalam	Berhasil
2	komunikasi data, baik dengan tujuan keamanan bersama, maupun untuk privasi	yRPTPTy_}xZpy \>_pWU2YX • Z\ • zTTxpP-y[|_Sp P-s[• Q_Pp-|_xR {S2{SV{U2} • x x\QWZpQ|	komunikasi data, baik dengan tujuan keamanan bersama, maupun untuk privasi	Berhasil

3	individu. Para pengguna komputer yang menginginkan agar datanya tidak diketahui	xPZxxTv{2m\S_-R[SuZx • _-yRPR{zt|-X_SuPtPYxPYxPUpP-pZ\SZpy\ • w\2yTv_U2YTy[zpUxx	individu. Para pengguna komputer yang menginginkan agar datanya tidak diketahui	Berhasil
4	oleh pihak-pihak yang tidak berkepentingan selalu berusaha menyiasati cara mengamankan	}QXz~xU\y~xU\ytpPY2yTv_U2\XSVXR[SVWSu_S2~X~_RT_t|xQ_VpPtPtx_}pyT2^\S_-|[Su_PpPUpP	oleh pihak-pihak yang tidak berkepentingan selalu berusaha menyiasati cara mengamankan	Berhasil
5	informasi yang akan dikomunikasikan atau yang akan disimpan. Perlindungan terhadap	xP[}|Pp~T2w\ • Z-pV\ • ZxVQ|{SxV\QWUpP-py\TtpPY2_UpP-vW}xS~pP2mXSQT • Yx • Z\ • zt|VpY\R	informasi yang akan dikomunikasikan atau yang akan disimpan. Perlindungan terhadap	Berhasil
6	kerahasiaan data meningkat, salah satu caranya dengan menerapkan ilmu kriptografi.	y[• pU\QW\pP-v_zpPtPT • ZUpy2~\~_V2~\V{-q_ • pPtpZtPYpP-|[St|\RV\ • T~Sx2V • x}z}Z • pXT?	kerahasiaan data meningkat, salah satu caranya dengan menerapkan ilmu kriptografi.	Berhasil
7	Kriptografi adalah salah satu ilmu yang digunakan untuk menjaga kerahasiaan	i|TRyQu|\wW-pY\~_V2~\~_V2~\V{-xQPTtpPY2YTu{SpV\ • x • yxyPtPWpZ\2VXS_Vp~Tp_S2	Kriptografi adalah salah satu ilmu yang digunakan untuk menjaga kerahasiaan	Berhasil
8	dan keamanan data sudah berkembang sejak jaman Yunani kuno. Kriptografi	v_S2VXpS\ • _S2Y\V_-Q{ZpU-s[• y[Ps_Su}tT\yWpS\ • dTP\ • W-y{S}-i|TRyQu|\wW	dan keamanan data sudah berkembang sejak jaman Yunani kuno. Kriptografi	Berhasil
9	semakin berkembang dari jaman ke jaman sampai saat ini. Salah	Q[PpVT • _t|UtS_pPY2Y\SW-{_PpP-y[-{_PpP-Q_PR_T2~\py-xPT?mpQ\z	semakin berkembang dari jaman ke jaman sampai saat ini. Salah	Berhasil

10	cukup handal, stabil dan menjadi induk dari algoritma – algoritma kriptografi	q{UT}-z_Sv_R> }V__xQ-v_S2SX • T\vW-xPZTV-v_ • x\~ZQSWz|_ -‡\~ZQSWz|_-y| TRyQu|\wW	cukup handal, stabil dan menjadi induk dari algoritma – algoritma kriptografi	Berhasil

Dari tabel pengujian 4.1 di atas, dapat di hasilkan nilai akurasi sebagai berikut :

$$\text{Akurasi} = \frac{\Sigma\, berhasil - \Sigma gagal}{\Sigma data} \times 100\%$$

$$\text{Akurasi} = \frac{10-0}{10} \times 100\% = 100\%$$

Pengujian pada skenario 2

Tabel 4.2 Pengujian pada skenario 2

No	Data uji	Hasil enkripsi	Hasil dekripsi	Ket
1	A	i[• pU\QW\pP-v_S2V XpS\ • _S2Y\V_-pY\~_ V2U\~tpPY2~\ • Z\V~t PzxPY2Y\~_P2VQ|{Sx V\QW-v_zp- s_TyZtP YpP-V{WT_S2VXpS\ • _S2\XS~|_2S\T}x • x • yxy~SW{p~T2WSv W{xYx?np|\2}X • ZYT P\2VQ|}xV[• 2w\ • Z-|[SuWSuWSy_S2_Yp|-v _zpPtpzxY\yZxVXV_ VTW-}QXz~xU\y~xU\ ytpPY2yTv_U2\XSVX R[SVWSu_S2~X~_RT _t|xQ_VpPtPtx_}pyT2 ^\S_-|[Su_PpPUpP-xP[}|Pp~T2w\ • Z-pV\ • Zx VQ|{SxV\QWUpP-py\ TtpPY2_UpP-vW}xS~ pPA[VxPYu_-R[• ~W Sv{Su_S	Kerahasiaan dan keamanan data adalah hal yang sangat penting dalam komunikasi data, baik dengan tujuan keamanan bersama, maupun untuk privasi individu. Para pengguna komputer yang menginginkan agar datanya tidak diketahui oleh pihak-pihak yang tidak berkepentingan selalu berusaha menyiasati cara mengamankan informasi yang akan dikomunikasikan atau yang akan disimpan.Sehingga perlindungan	Berhasil
2	B	V[• z_Zp}-2VXS_Vp~ Tp_S2Y\V_-|[SxPYy_z >}pQ\z}pyx2^\S_-pY\ ~_V2}X • w\ • YTpP-V [UQZpQ|X • V • x}}x-	terhadap kerahasiaan data meningkat, salah satu cara adalah penyandian teks dalam enkripsi.	Berhasil

		dPUSW~QW-|[• T}\y_ S2~xpyx2} • }~XQ~tP YT\\z_S2}XQ_S2_}~ W-|[S{_ZxUp|\yyXStp PY2yTv_U2Y\R_z2YT s_]p-`Y\2\Xs[ • p}\2_R uR • xyPpX • V • x}}xt pPY2\Tp~\2ZxZx • _U pP-Q[~t|zxO~R]yMx} Vt|2nzS[|Mx}Vt|2I\V_ -dP]Sw~VWQ • mV_S v_ • vfKm8-F|TRQX2I HA-`Y{pP]tY-dP]Sw~ VWQ • mV_Sv_ • v`K m8-v_S2~Xs_YpWSX _2IT|_Sp	Enkripsi merupakan suatu proses pengubahan pesan asli menjadi karakter yang tidak dapat dibaca. Ada beberapa algoritma enkripsi yang biasa digunakan seperti Block Cipher, Stream Cipher, Data Encryption Standard (DES), Triple DES, Advanced Encryption Standard (AES), dan sebagainya. Dimana	
3	C	Q[zx_~2_RuR • xyPpPtS T~WUxUp|\yyXSW}VW U2yXS~X • YTSW2nXv _SuV\ • ~SR}t~-R[Su{_p U\ • UtS_pQT2U\QWR2[Sy|TR~T2SX • T\vW-R[} pP-p~RxZxP|_UpP-v[US W~QW2kSV{U2SXS_V p~TpV\ • Zpy\2w\ • Z-Q_ Su_z2}X • yT • Z-|_UpZx Zx • _UpPRpU-|[z}YX2 V • x}z}Z • pXT2w\ • Z-p V\ • PtPYtPUSW~QW-v _S2YXQV • x}}xV\ • Zp y\? --- A_RpU-Q_zTPtyQv[-X_ Su\y_S2YTu{SpV\ • ZpQ |~tS_T_zpP-R[SX_SvW\ • ztV}2WSx\v_RpU-|[z} YX2NTRUXSO~R]yMz _T • WSuaLM8-y_ • tP\,	setiap algoritma memiliki karakteristik tersendiri. Sedangkan proses pengubahan kembali hasil enkripsi menjadi pesan asli dinamakan dekripsi. Untuk merahasiakan data yang sangat penting maka digunakanlah metode kriptografi yang akan mengenkripsi dan deskripsikan data. Salah satu metode yang akan digunakan dalam pembuatan penyandian teks ini adalah metode Cipher Block Chaining (CBC), karena	Berhasil
4	D	i|TRyQu|\wW-:^ • X}z }Z • p}VX-s[• p~\~Zp| T2\\z_}pdTP\ • W-†^ • X}z}~ª2_ • VWSX_-†~ Xq|XV©-:|\z_}x_>}tY\ • ZUpP-†Z□p}VtWS† \SyT • w\2©ySWzxPY †V{Rx~\ • 2D\vW-y|T RyQu|\wW-s[• p|zxªQ[	Kriptografi (cryptography) berasal dari bahasa Yunani ”cryptos” artinya ”secret” (rahasia), sedangkan ”graphein” artinya ”writing” (tulisan). Jadi kriptografi	Berhasil

		]S[z2z • xyT • Zª2zTQTQ_S2|\z_}x_?ESW~VRYS_[x\v_RpU-xQPTZpP-Q[Sxx • yxyPtPWpZ\2VXpS\ • _S2}XQ_S2OS{]tmqUStWXS-742I\~_P2V • x}z}Z • pXT2~XSWSuZxyX|{UpP-x~zxQ\z\V_x2yXSST • RR}ZT>}t}XSyT2}XQ_S2Pt~}pZX8\v_RpU-v_zp\V_x2WSwR • |_}xtpPY2Y\R_z2	berarti ”secret writing” (tulisan rahasia). Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan (Bruce Schneier, 1996). Dalam kriptografi sering ditemukan istilah atau terminologi, seperti pesan (message) adalah data atau informasi yang dapat	
5	E	vW_p^\\2Y\ • ZxSX • ZXSyT2S\yP\ • w\?CpS\2Q\xP-TPzTV-R[}pP-pY\~_V2}RpWSV[UQRQ\xPztuz8\V_x2yXy~+TX~_}2]~[\SyXZy?nt~\ • Zp}\V_t|xR_-v_zp\V_x2WSwR • |_}xtpPY2YTyW • xS-:SX~_RTW-y{ • x|2~\~{ • pP-V[RtVQ|{SxV\QW2Y}s-py\TtpPY2YTQWPR_S2YT2Y\~_P2SXvW\2}XS[UpS\ • y[• V_}>}VR • pZX>ZQ\?nt~\ • tpPY2yXS~T|}\ • zxY\yVpPtp_t|xR_-V[UQ-V[zp}T2Y\R_z2\XS\X • yxy]xy • pxS\u[>	dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah plainteks (plaintext) atau teks jelas (cleartext). Pesan dapat berupa data atau informasi yang dikirim (melalui kurir, saluran telekomunikasi, dsb) atau yang disimpan di dalam media perekaman (kertas, storage, dsb). Pesan yang tersimpan tidak hanya berupa teks, tetapi dapat berbentuk citra (image),	Berhasil
6	F	f_RpS-R|QQ[}2[Sy|TR~T • w\>_~R]y]x}Vt|-|[SuZx • _UpP-s[_t|\R_-w{Su~T2S\V[PpyTy_2YTpPzp|\ • w\2Xx • Z}x~t|PTy\QW-v_S2Xx • Z}x}T\}VWzT~T>}tUT • ZYpU}P[T~T2]}P[T~}xRS8ZpP-vW[T~T2ZxX[T~T}P2}\v_-sQQqV-qW~z[• 2yXS}X • {Vx-h~zxQ\zU}P[T~T2Y\ • ZxXxQW-vW~t	Dalam proses enkripsinya, block cipher menggunakan beberapa fungsi matematika, diantaranya fungsi permutasi dan fungsi substitusi, sehingga konfusi (confussion) dan difusi (diffusion) pada block cipher terpenuhi. Istilah konfusi dan difusi	Berhasil

		|UtP\~V\ • Q~[V2NRp{ Ztmz_S • RS2}\v_-V_ VTP-7:-l[ST|xVPtp-yR Sw{}xZpP-vW[T~T2S XS{~pV\ • VpQ-X_Su Vp|xQZx}XSU\VWUp P-v_RpS-QW}V[P2V • x}z}Z • pXT>Up|X • _-f_~py	diperkenalkan oleh Claude Shannon pada tahun 1949. Menurutnya, konfusi dan difusi merupakan hal yang harus diperhatikan dalam sistem kriptografi, karena Dapat	
7	G	c[• v_}p|UpP-z_}xQ-tV} V|\y~T2\TV_xy-V[• Q[_ Ty-v_~py +YTQWPR{Ry_S2\\zz\2 }X • Zxs_VpP-sWz<\TV @AL-v_~py +SX|\xpy-qWzS_-|[S{_Z x-S{}pV^{a}2_zp{-VWZpV- v_~py+YTy[SpQT2Q\u W2~Xv_SuV\ • ~tPYT\\z _S2\TV_xy-nnO2yTv_U, --- PtS~tPYp|xzW-qWzS_-Q []p|\2VXQ[RT|xz_S?A~[V2V\S[Sp --- xyx2U\ • w\2\TV_xy-lnO 2~\{_-X_SuZx}T~WV2{ SV{U, --- Zx[Sy|TR~T2~Xs__2YX • Z\ • VpPtpPtPYtPUSW ~QW-sWz<\TVzt|}t\xVP pV\2VXQ[RT|xz_S2^TV |\2	Berdasarkan hasil ekstraksi bit-bit tersebut dapat disimpulkan bahwa pengubahan bit-bit MSB dapat membuat citra menjadi "rusak" atau tidak dapat dikenali lagi, sedangkan pengubahan bit-bit LSB tidak mempengaruhi citra secara keseluruhan. Oleh karena itu hanya bit-bit MSB saja yang dipilih untuk dienkripsi sebab dengan hanya mengenkripsi bit-bit tersebut maka keseluruhan citra	Berhasil
8	H	|[S{_ZxzxY\yZp}\VZxV X • _RxRpZT? --- k{P~_V2\TV@AL-X_Su Zx[Sy|TR~T2SX|}X • Z\ S{Vx --- VWSuV\VUt_PpP\ • -kW UpVpPtp}pyx2\TVtpPY2 YTtPUSW~QW, --- PpV\2yx{{V2\TV}x~\ • w\2tpPY2yTv_U2WUTy -vWX • V • x}}x+S\QW	menjadi tidak dapat dikenali lagi. Jumlah bit MSB yang dienkripsi mempengaruhi tingkat keamanan. Jika hanya satu bit yang dienkripsi, maka tujuh bit sisanya (yang tidak ikut dienkripsi) masih dapat memperlihatkan	Berhasil

		V2Y\R_z2SX|}XSQTz_zy_S2zx{{Z2R_{[U2YT2Y\~_P2^TV|\>}tUT • ZYpzxPYy_z2VXpS\ • _S • w\2|X • Y\z-mQXz --- y_ • tP\2WzT}tyX~_V2[Sy|TR~T2?sWz2CmcPp~TzZx}XSQxy_S2} • }~Xv{ • 2}XSSxV_}x ---	wujud objek di dalam citra, sehingga tingkat keamanannya rendah. Oleh karena itu setelah enkripsi 1-bit MSB masih diperlukan prosedur permutasi	
9	I	V_Ps_VpP-TPzTV-|[Su_]pV-RWvtQ-pZ\SZx}XSRRtU-tXXy]}P[T~T}P-I:`?jpR-JW\ • Z-vW-v_RpS-|_UpQ\zPtpPtPtt\xVV\ • _pUUpP-tPUSW~QW-~[_xU-v_ • xZT_-sWz2CmcV_SR_-R|QQ[ZT|-R[Su_]pV\ • }t~xv_V • w\8Pp~Tzzty\R_tQx|L~ZQSWz|_-tPUSW~QW-Q[RtVzxX-X_SuZx{}TQUpP-vW-v_RpS-|_UpQ\zT • W-|[Su[Sy|TR~T2^TV|\2Y\~_P2|\ • _V2~~p~TpQ2Dx|Q\z_xy-lnO2w\ • Z-vWX • V • x}}x\v_RpU-tS~py	tambahan untuk mengacak pixel agar diperoleh efek confusion [7]. Tao Xiang di dalam makalahnya menyebutkan bahkan enkripsi lebih dari dua bit MSB (tanpa prosedur pengacakan sesudahnya) masih tetap belum Algoritma enkripsi selektif yang diusulkan di dalam makalah ini mengenkripsi citra dalam ranah spasial. Jumlah bit MSB yang dienkripsi adalah empat	Berhasil
10	J	sWz2~XQ{\xVp~T~~tPX~Wzx_S2YT2Y\~_P2f<L-DPzTV-|[PR[• }QXz]x}Vt|xS\u[-X_SuzxY\yPtS~TPtpW-z{_TPYpP-Qy\VW}VWU2YX • Z\ • ~~_T • T|_Yt-|_UpZxZx • _UpP-|RZtMcN~WUt-A[zx_~2\R}V-v_zp_t|xy{ • pP-t	bit sesuai hasil penelitian di dalam [1]. Untuk memperoleh cipher-image yang tidak mempunyai hubungan statistik dengan plain-image, maka digunakan mode CBC-like.	Berhasil

		S~py-sWz2w\xyx29sWz2CmcZp|T2~XVW\R ~xuX~-e_Ps_ • 2<p-|[P R[• ~WVpyUpP-QVX| _-pQY}|TVS\2[Sy|TR~ T2YX • Z\ • P}YX2NO aRxVX?jxPWp{-V[• ~ [_xU-v_VTQx2[Sy|TR ~T<YXy|TR~T2}\v_-q WzS_-u|\X~]pQX?@x ~\~V\ •	Setiap blok data berukuran empat bit yaitu 4-bit MSB dari setiap pixel. Gambar 2(a) memperlihatkan skema algoritma enkripsi dengan mode CBC-like. Tinjau terlebih dahulu enkripsi-dekripsi pada citra grayscale. Misalkan	

Tabel 4.3 Data uji skenario 2

No	Data	Isi
1	A	Kerahasiaan dan keamanan data adalah hal yang sangat penting dalam komunikasi data, baik dengan tujuan keamanan bersama, maupun untuk privasi individu. Para pengguna komputer yang menginginkan agar datanya tidak diketahui oleh pihak-pihak yang tidak berkepentingan selalu berusaha menyiasati cara mengamankan informasi yang akan dikomunikasikan atau yang akan disimpan.Sehingga perlindungan
2	B	terhadap kerahasiaan data meningkat, salah satu cara adalah penyandian teks dalam enkripsi. Enkripsi merupakan suatu proses pengubahan pesan asli menjadi karakter yang tidak dapat dibaca. Ada beberapa algoritma enkripsi yang biasa digunakan seperti Block Cipher, Stream Cipher, Data Encryption Standard (DES), Triple DES, Advanced Encryption Standard (AES), dan sebagainya. Dimana
3	C	setiap algoritma memiliki karakteristik tersendiri. Sedangkan proses pengubahan kembali hasil enkripsi menjadi pesan asli dinamakan dekripsi. Untuk merahasiakan data yang sangat penting maka digunakanlah metode kriptografi yang akan mengenkripsi dan deskripsikan data. Salah satu metode yang akan digunakan dalam pembuatan penyandian teks ini adalah metode Cipher Block Chaining (CBC), karena
4	D	Kriptografi (cryptography) berasal dari bahasa Yunani ”cryptos” artinya ”secret” (rahasia), sedangkan ”graphein” artinya ”writing” (tulisan). Jadi kriptografi berarti ”secret writing” (tulisan rahasia). Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan (Bruce Schneier, 1996). Dalam kriptografi sering ditemukan istilah atau terminologi, seperti pesan (message) adalah data atau informasi yang dapat

5	E	dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah plainteks (plaintext) atau teks jelas (cleartext). Pesan dapat berupa data atau informasi yang dikirim (melalui kurir, saluran telekomunikasi, dsb) atau yang disimpan di dalam media perekaman (kertas, storage, dsb). Pesan yang tersimpan tidak hanya berupa teks, tetapi dapat berbentuk citra (image),
6	F	Dalam proses enkripsinya, block cipher menggunakan beberapa fungsi matematika, diantaranya fungsi permutasi dan fungsi substitusi, sehingga konfusi (confussion) dan difusi (diffusion) pada block cipher terpenuhi. Istilah konfusi dan difusi diperkenalkan oleh Claude Shannon pada tahun 1949. Menurutnya, konfusi dan difusi merupakan hal yang harus diperhatikan dalam sistem kriptografi, karena Dapat
7	G	Berdasarkan hasil ekstraksi bit-bit tersebut dapat disimpulkan bahwa pengubahan bit-bit MSB dapat membuat citra menjadi "rusak" atau tidak dapat dikenali lagi, sedangkan pengubahan bit-bit LSB tidak mempengaruhi citra secara keseluruhan. Oleh karena itu hanya bit-bit MSB saja yang dipilih untuk dienkripsi sebab dengan hanya mengenkripsi bit-bit tersebut maka keseluruhan citra
8	H	menjadi tidak dapat dikenali lagi. Jumlah bit MSB yang dienkripsi mempengaruhi tingkat keamanan. Jika hanya satu bit yang dienkripsi, maka tujuh bit sisanya (yang tidak ikut dienkripsi) masih dapat memperlihatkan wujud objek di dalam citra, sehingga tingkat keamanannya rendah. Oleh karena itu setelah enkripsi 1-bit MSB masih diperlukan prosedur permutasi
9	I	tambahan untuk mengacak pixel agar diperoleh efek confusion [7]. Tao Xiang di dalam makalahnya menyebutkan bahkan enkripsi lebih dari dua bit MSB (tanpa prosedur pengacakan sesudahnya) masih tetap belum Algoritma enkripsi selektif yang diusulkan di dalam makalah ini mengenkripsi citra dalam ranah spasial. Jumlah bit MSB yang dienkripsi adalah empat
10	J	bit sesuai hasil penelitian di dalam [1]. Untuk memperoleh cipher-image yang tidak mempunyai hubungan statistik dengan plain-image, maka digunakan mode CBC-like. Setiap blok data berukuran empat bit yaitu 4-bit MSB dari setiap pixel. Gambar 2(a) memperlihatkan skema algoritma enkripsi dengan mode CBC-like. Tinjau terlebih dahulu enkripsi-dekripsi pada citra grayscale. Misalkan

Dari tabel pengujian 4.2 di atas, dapat di hasilkan nilai akurasi sebagai berikut :

$$\text{Akurasi} = \frac{\Sigma\, berhasil - \Sigma gagal}{\Sigma data} \times 100\%$$

$$\text{Akurasi} = \frac{10-0}{10} \times 100\% = 100\%$$

Pengujian skenario 3

Tabel 4.4 Pengujian skenario 3

No	Data uji	Hasil enkripsi	Hasil dekripsi	Ket
1	A	A[]p|\2S\V[PpyTQ-t PUSW~QW-v[Su_S2 SQv[-aLM2Y\R_z2YT • w\ V_UpP-Q[_pZ\x :2<23-ä xE2W-xM2K-BM, T, +Y\ • ZtV • x}}x}t\\u _T, <2-823-ä xE2W-xn2K- aM, T, + kSV{U2SX~_UTV\ • X • V • x}}xZtV • x}}x~pY\2\R} V-R[• V_Pp vW~t|RTV\ • MtpPY2Y\ ~_P2U\~T • W-a=-pY\~_ V, T • Wzx_Rxt\VWQ • {t^z}|- py\TD GR_Zp\~ZQSWz|_-xPT2Gk 23+¯>=>f2Gk2yTv_U2}X SQx2|\z_}x_-V[zp}T2U\S{ }2~|_+PT~_T • w\2}\v_-R| QQ[}2YXy|TR~T? g{Su~T2K-X_SuZxZx • _U pP-v W-v_RpS-e_Ps_ • 2<+_ZpQ \z[TP YQW-Q[Zt|VpP\2w\xyx2R ~t|\QW-JBo2_SV_ • p_xy_x	Secara matematis, enkripsi dengan mode CBC dapat dinyatakan sebagai ( ) −1 = Å i K i i C E P C i (1) dan dekripsi sebagai 1 ( ) − = Å i K i i P E C C i (2) Untuk melakukan enkripsi/dekripsi pada blok pertama diperlukan C0 yang dalam hal ini C0 adalah initialization vector atau IV (pada algoritma ini IV = ‘0000’). IV tidak perlu rahasia tetapi harus sama nilainya pada proses dekripsi. Fungsi E yang digunakan di dalam Gambar 2 adalah fungsi sederhana yaitu operasi XOR antara bitbit	Berhasil

		y+Vx • ^T2FT 2YX • Z\ • Vp~T~~ tPY<eAC\ • }t\X~{P • w\ iT2W-xH2e-JE, T, 2-È 2=8 X_SuZpQ| VpQ-xPT2eT23-BW-äMx« 2?- R_Zp}y[PpX • V • x}}x v_S2eT23-aW-R_Zp }y[PpZtV • x}}x+CQ v[-X_SuZxZx • _U p P-vW-v_RpS-pQY}|TVS\ 2WSx vW}t\xV McN~WUtUp|X • _-TPzTV -Q[zx_~2\R}V +SX • ZYTP\y_S2Vx • ^T2 w\ • Z-,	kunci Ki dengan hasil peng-XOR-an sebelumnya: K i i i E X X K i ( ) = Å (3) yang dalam hal ini Xi = Pi Å Ci – 1 pada skema enkripsi dan Xi = Ci pada skema dekripsi. Mode yang digunakan di dalam algoritma ini disebut CBC-like karena untuk setiap blok menggunakan kunci yang	
2	B	s[• s[Zp_tY\2~pY\2SQ v[-aLM2w\ • Z-p~RxU TP]x~pY\2~XVW\R_~ RU2_ZpQ\z}pS\2w\xy x2Vx • ^T2[UQ7:43 PM 4/2/2017yXSP\~E8-i{S qW-iW-R_Zp}tyTp}-s QQyZpy\2YTQ[_Ty-x Pzt|SpQ-y[t2w\ • Z-R_ S{_SuPtpWTZ\29sWz? ETP]xT • yXSP\~T • W -vW_pPYyWzy_S2Y\S W-w{Su~T2^VpR}2Q QuW}VW]2S\R-ZW-9 <23-Svx«2uT8-v[Su_S 2=-±vx • 2?2W->><> ?>?2Y\□>2ž-S • 292@T ~_T2_ypQ-:~XtY2Wzt |\QW-pY\~_V2u>2w\ • Z-s[• R[• pP-Q[_pZ\x UTP]x • pU\QW\?H|}\ V_xy-y{SqW-xPzt|Sp	berbeda-beda (pada mode CBC yang asli kunci pada setiap blok adalah sama yaitu kunci eksternal K). Kunci Ki pada setiap blok data disebut internal key yang panjangnya juga 4-bit. Kunci internal ini dibangkitkan dari fungsi chaos logistic map, xi + 1 = r xi (1 – xi) (4) dengan 0 £ xi £ 1, i = 0, 1, 2, ... dan 0 £ r £ 4. Nilai awal (seed) iterasi adalah x0 yang berperan sebagai kunci rahasia. Empat bit kunci internal diperoleh	Berhasil

		Q-vW~t|Q~[V2~Xs_Y pW-s[• xVxVe=`SxQ\ x]z_QQvxZxV\~WUpP -v[Su_S2?>2\XS{RpP Yy_Rx}pS~pW-x_-|[S q_~pW-R_S{_Su\ • ZU pQWwt-X_SuZxWSu WSy_S>}tQ\ • TxVPtp ~}yQ • Z-z_}xQ-R[• y _Rx_S2yXS~Xs{z2{S V{U2SX • Z|\T~_pZT pP-xPztZXSPtp}pT\?m t^\S_-|_ztS\VW}>SxQ\ x]z_QQv2YTyRSW[• QW-y[-xPztZXSZtPYp P-|[SuZx • _UpP-R[• Q _Pp_S2\XSWUTy72iZ -QWwt-v2<2u-˜>2^QT Pz288tpPY2Y\~_P2U\ ~T • W-qRx • y-vWPT Q\xZp|T2?-v_S2\XSy|\ \z?-zWSuZ\2u-;<^QTP z20-=}xtX2¨-	sebagai berikut [10]: nilai chaos xi dikalikan dengan 10 berulangkali sampai ia mencapai panjang angka (size) yang diinginkan, selanjutnya potong hasil perkalian tersebut untuk mengambil bagian integernya saja. Secara matematis, nilai chaos x dikonversi ke integer dengan menggunakan persamaan berikut: T(x, size) = x *10 , x ¹ 0 count (5) yang dalam hal ini count dimulai dari 1 dan bertambah 1 hingga x * 10count > 10size – 1.	
3	C	j_}xQSX_-y[PTYTpP +YTpS_xQ-s_Yx_S2WS V[Yt|-Q_WpvWRpS_pP Yy_S2YX • Z\ • --- R_}pPYpP-u_ • x~-u_Sv _-R_Zp~t|}pS\pP-2nXs_ YpW +^Q • yQz-|W}pQUpP-Z W->?=>?9;>ZpP-QWwt0 292S\y_ +YT|{RpW-v_ • x]}{SV0 2?-Q_PR_T2^QTPz23-Z x}XSRRtU +===:;=<2-=;23-?98?-<> +VX|{Zx_S2_PsWR2\\u W\ • T • yXu[• <PtpZtPY pP +><8=<23-?9 --- dS~py-sWz2yXS_UzW • 2Y\SW-S[~S[}tPzp~T2\T	Hasilnya kemudian diambil bagian integer saja (dilambangkan dengan pasangan garis ganda pada persamaan 5). Sebagai contoh, misalkan xi = 0.003176501 dan size = 4, maka dimulai dari count = 1 sampai count = 6 diperoleh 0.003176501 * 106 = 3176.501 > 103 kemudian ambil bagian integer-nya dengan 3176.501 = 3176	Berhasil

		• [• 2><8 +YT{_ZxV\ • }t\\u_T2FT 2w\xyx2¯<=>ƒ+N<>M-¤ -aP-v_ • xVp~T~X • V • x }}x}tQ\ • TxVPtp |[SuZ\ • yTy_S29sWz2C mcZp|T2~XVW\R~xuX~ tpPY, Zx} • }~XQ-j_}xQ-tPUS W~QW-V[• {_Zp}-Q[R T|xz~xuX~\v_RpU +^TV|\2yXS[Sy|TR~T2] x}Vt|xS\u[?h • yxy~SR}t ~-v[USW~QW +YT~_UTV\ • ~SR}t~-s[• y[_pQTy_S2~XR[• V W-X_Su vWzTPWTVUpP-R_ZpI pS_p|-_8+ORuR • xyPpZ x\V_}2Y\R_z2YTS_PR_ zy_S2{SV{U2^TV|\, _t|yp|Sp-B|QQ[} • w\2YT ~_UTV\ • zxZ\2V\~W2S\ QWSuS\QWSu TPzTV-y_SpQ-S[Z2o8-u| XtP-:J>ZpP-sQxtc, GpYT>~pY\2~XVW\R_ XyX2V\ • _R2z\SP\2YTp S_xQ-_xy-lnO, UtSxvW\ • ZxR~t|\QWU pP-v[Su_S2SQv[-aLM2~ Xq_ • p V[• RW}pU,	Empat bit terakhir dari representasi biner 3176 dijadikan sebagai Ki yaitu '1000'. C1, C2, …, Cn dari hasil enkripsi selanjutnya menggantikan 4-bit MSB dari setiap pixel yang diproses. Hasil enkripsi terjadap seluruh pixel adalah citra terenkripsi (cipher-image). Untuk proses dekripsi dilakukan proses berkebalikan seperti yang ditunjukkan pada Gambar 2(b). Algoritma di atas dapat dirampatkan untuk citra berwarna. Prosesnya dilakukan tiga kali, masingmasing untuk kanal red (R), green (G), dan blue (B). Jadi, pada setiap byte kanal warna diambil 4-bit MSB kemudian dioperasikan dengan mode CBC secara terpisah.	
4	D	jW}VRYS_P2SXS{~pV\ • ~SR~t|zx]xy • ptpPY2}X • y T • Z +~Xs__2~Xs{\zVx~z}Z • p S-|[PR[• ~WVpyUpP-vW} V|Ts{}x xPztP}xy\Q~xuX~ZxZpQ\\|] xy • pzt|}t\xV-DPzTV-qWz	Histogram merupakan properti citra yang penting sebab sebuah histogram memperlihatkan distribusi	Berhasil

		S_ +}RpWS<WPpZX2UTQyQu||PtpPtS_tPzTV-Q{\V{-R RRp X_SuUz_}>tpWzT\v_-R{Sq_U<}x • ^\yZpP-~[Ps_V~[Ps_V? DPzTV-|[Sq[YpU-R[SX[• pPY2SX • ZYTP\y_S, Vx~z}Z • pS-TPzTV-|[RpVxy_S2_SpQTQW}2X • tQxtP}x-|_Up zW}VRYS_P2}RpWS<WPpZX2Y\ • Vx~z}Z • pS-qW~z[• <WPpZX, }tU\S{} • w\2yTv_U2SX|WRxVT2VX|W • x}\ • }t^\S_-Qy\VW}VWU? mQXzUp|X • _-xyx>Vx~z}Z • pS-qW~z[• <WPpZX2~Xz_ • T~SX_ +|X~_zxX-v_zp|-:XRpy2~XzWSuZ\2y\z_S2yXSU\v_~2~XS_Su_S, }V_zx~zxV2ITQy • x\xQW-X_Su • tQ\VW[2{SxXQSS-R_Zp]x}Vt|T|_Yt pY\~_V2~Xs{\zT • YTy_}x_pUyp\~ZQSWz|_ +[Sy|TR~T2^TV|\2SX|WRxVT2VxpQTV_}2w\ • Z-s_YT~-I7`? e_Ps_ • 28p-|[PR[• ~WVpyUpP-zW}VRYS_P2^TV|\, ¬c_ • s_ • p¬-Q[_tQx|Zx[Sy|TR~T>ZpP-e_Ps_ • 28s-pY\~_V, Vx~z}Z • pS-qW~z[• <WPpZX<Ptp-f_~py-vWRxU\V_pUyp zW}VRYS_P2^TRUXST|_YtPtST~WUxZx~zSW_T~T2{SxXQSS	intensitas pixel di dalam citra tersebut. Untuk citra plain-image histogramnya membentuk suatu pola yang khas, yaitu ada puncak-puncak dan lembahlembah. Untuk mencegah penyerang menggunakan histogram untuk melakukan analisis freluensi, maka histogram plain-image dan histogram cipher-image seharusnya tidak memiliki kemiripan secara statistik. Oleh karena itu, histogram cipher-image seharusnya relatif datar (flat) sehingga tahan terhadap serangan statistik. Distribusi yang relatif uniform pada cipherimage adalah sebuah indikasi bahwa algoritma enkripsi citra memiliki kualitas yang bagus [9]. Gambar 6(a) memperlihatkan histogram citra 'Barbara' sebelum dienkripsi, dan Gambar 6(b) adalah histogram cipher-image-nya. Dapat dilihat bahwa	

		+w\ • Z-|_Sp_t|_tY\2YX • Z\ • Vx~z}Z • pS-RQ\xPxS\u[, IpS_p|-\8}pS~pW-]8PtS~t|RxU\VV\ • zW}VRYS_P2^TV|\2¯BtP\ƒRQ\xPxS\u[2{SV{U2~XVW\R y_SpQ-U_ • • _-CJO2Y\ • IpS_p|-Z8}pS~pW-[8\v_RpU +UTQyQu|\Pp~T • Z|_}pPY2V\ • _R2z\SP\2}\v_-qW~z[• xS\u[, mpS\2~XR[• VW-qWzS_-€L\S\\S_¯>Vx~z}Z • pS-qW~z[• xS\u[+}\v_-Q[zx_~2V\ • _R2lIcWTZ\2yXSQTz_z2	histogram cipher-image memiliki distribusi uniform yang mana berbeda dengan histogram plain-image. Gambar 7(a) sampai 7(c) memperlihatkan histogram citra ‘Lena’ (plain-image) untuk setiap kanal warna RGB dan Gambar 7(d) sampai 7(f) adalah histogram masing-masang kanal warna pada cipherimage. Sama seperti citra ‘Barbara’, histogram cipherimage pada setiap kanal RGB juga terlihat	
5	E	wQ\V\V_x, zt|Zx~zSW_T~T2{SxXQSS, IpS_p|-\8PtS~t|RxU\VV\ • Vx~z}Z • pS-qWzS_-€i\{ l_VpQ¯2~~_T • T|_Yt-v_S2J|\\S6:\2_ZpQ\zVx~z}Z • pS-qW~z[• <WPpZX<Ptp-A[ZxVTV_t|_tY\2YX • Z\ • Vx~z}Z • pS-qW~z[• <WPpZX2Y\SW-v{\2^TV|\2~Xs[RTSSX_2UTQyQu|\]x}Vt|xS\u[-v_ • x¬F_W2C\z_Rƒ PtST~WUxZx~zSW_T~T2w\ • Z-S[RpyTwx • W[}|P?Ot|Zp~\SV\ • Vp~T~Vp~T~\ • _Rx~TQVx~z}Z • pS-vW-py\QZp}\V	flat atau terdistribusi uniform. Gambar 8(a) memperlihatkan histogram citra ‘Taj Mahal’ (plain-image) dan Gambar 8(b) adalah histogram cipher-image-nya. Sedikit berbeda dengan histogram cipher-image dari dua citra sebelumnya, histogram cipher-image dari ‘Taj Mahal’ memiliki distribusi yang relatif uniform. Berdasarkan hasil-hasil analisis histogram di atas dapat disimpulkan bahwa cipher-image memiliki histogram yang (relatif) flat	Berhasil

		Zx~T|}x~V\ • _pUyp]x }Vt|xS\u[-|[PxQTyW-z W}VRYS_P2w\ • Z-:|X ~_zxX2XRpy-Q[VxPY u_-|[SX{RxyUpP-R[S X[• pPY2SX~_UTV\ • \ • _Rx~TQ}V_zx~zxV -TPzTV-|[Sv[ZTV}x~x uX~\V_x2Vx • ^T?Fp~ T~T • W-|[STPWTVUp P-s_VU_-pQY}|TVS\2 [Sy|TR~T2^TV|\2w\ • Z-vWxQ{Ry_S2WSxP tST~WUxUt_PpP\ • tp PY2\\u{}?L~ZQSWz|_ -tPUSW~QW-qWzS_- Q[Vp|xQPtp}tP}xyTw zt|VpY\RUTP]x-A[SQ WzxX-p|zxPtpWxV\2V x • ^T2YTT\\z}tYTyW z2~\{_-|_UpVp~T~ZtV • x}}xzt|VpY\R]x}Vt|x S\u[-|[SuU\QWRy_S2^ TRUXST|_YtRpWS2w \ • Z-s[• s[Zp}xZSxXT y_S?Ep|X • _-pQY}|T VS\2w\ • Z-vWxQ{Ry_ S2WSxPtPYu{SpV\ • } x~ztS-qU\}~2S\y_-QW [py-qU\}~-X_Su}tP}xy Twzt|VpY\R~t|xs_VpP -y[]xQ- • WRpW-pz\~Z =2SXS{~pV\ • ~SR~t|z xUt_PpP\ • tpPY2}X • yT • Z2@T~_T2u>2\X S}XS_S2~Xs_YpW-y{ SqW-X_SuZx\XSWUp P-}QXz~tPYu{Sp-B_Z pXy~~t|T|[S2WSxSxQ\ x\U_R2QQuW}VW]2S \RZx{_pU-Q[_t~\SJ2~ XzWSuZ\2SX • T\vW- Z=-9J>+	sehingga menyulitkan penyerang melakukan analisis statistik untuk mendeduksi pixel atau kunci. Hasil ini menunjukkan bahwa algoritma enkripsi citra yang diusulkan ini memiliki keamanan yang bagus. Algoritma enkripsi citra seharusnya sensitif terhadap kunci. Sensitif artinya jika kunci diubah sedikit saja maka hasil dekripsi terhadap cipher-image menghasilkan cipher-image lain yang berbeda signifikan. Karena algoritma yang diusulkan ini menggunakan sistem chaos, maka sifat chaos yang sensitif terhadap perubahan kecil nilai awal (x0) merupakan properti keamanan yang penting. Nilai x0 berperan sebagai kunci yang diberikan oleh pengguna. Pada eksperimen ini nilai awal logistic map diubah sebesar D sehingga menjadi x0 + D,	
6	F	B|QQ[}2WSxVpPtpZp }\VZxQ\y{UpP-}QXz QS_SutpPY2SX • ZYT	Proses ini hanya dapat dilakukan oleh orang yang menggunakan	Berhasil

		P\y_S2SXVRZttpPY2~XQ{\x-B|QQ[}2[Sy|TR~T=YXy|TR~T2yXSZ\ • yx • Z-R_ZpUTP]xy[t8 • pU\QW\2w\ • Z-z_SX_-vWUty\z{T2RRtU-R[SuW • xS-v_S2}X • [• xS\?mtS\yWS2|x|Wz2_RuR • xyPpUSW~VRYS_[xtpPY2YTu{SpV\ • -Q[PpVT • }TQTV~TQ\2{SV{U2}\S_-z_]y[• >]S_]y[• 2w\ • Z-xPYxP-|[PsR_}Q-Q[]T|TVw-Qw}V[P2w\ • Z-vWYTP\y_S>}tUT • ZYpU}Sx • WUp~T2Y\V_-pPzp|\2}X • ZTSWP2Y\ • ~tPXSWPp|_S?ESW~VRYS_[x~pY\2Y}p|SX_-vWYTP\y_S2{SV{U2SX • T|WS2} • xx\QW72SX • ^Xu_V2WSwR • |_}xPtPtt\\SUt}\v_-RWVpV-X_SuzxY\yzt|\TyX • yTwWUp~T2~Xq_ • pRtZ\~x • yxyPtPYpV}t~-wWRttpPY2_Zp~pY\2T\SWSu_S?A~[V2V\S[SpTV{-y|TRyQu|\wW-|[PxQTyW-z_R<U\~tpPY2~\ • _YV_t|UpWzpP-t|\V~}VQytpWzT~SW{p~T>\TyVtPzxXTq_zxRS2PtS{t|TwWUp~T2WZtPzxy\Q~tPYu{Sp-v_S2WSV[YSWzp~-:SX|_}VWUpP-s_VU_-R[}pP-s[RTS-vWxs_V8-i|TRyQu|\wW-R_Zp[x|XU_R~ZxZx • _UpP-TPzTV-|[Sq[YpU-pP]pS\ • Zp|T2{}t|-X_SuzxY\y_t|VpV-TPzTV-|[Pp~xyW-yRPTPTy_}x\V_	metode yang sesuai. Proses enkripsi/dekripsi tergantung pada kunci (key) rahasia yang hanya diketahui oleh pengirim dan penerima. Semakin rumit algoritma kriptografi yang digunakan, semakin sulit pula untuk para hacker, cracker yang ingin membobol security system yang digunakan, sehingga komunikasi data antara pengirim dan penerima aman. Kriptografi pada dsarnya digunakan untuk menjamin privasi: mencegah informasi menyebar kepada pihak yang tidak terautentifikasi secara legal untuk mengakses file yang ada pada jaringan. Oleh karena itu kriptografi memiliki hal-hal yang sanagt berkaitan erat pokok yaitu privasi, authentification (memverifikasi identitas pengguna) dan integritas (memastikan bahwa pesan belum diubah). Kriptografi pada firewall digunakan untuk mencegah ancaman dari user yang tidak berhak untuk memasuki	

		x2T\SWSu_S>}tUT • ZYpUt|\z_}x_\ • Zpy\2Y\R_z2YT~WSv{SuW2nXq_ • pYp|TQ_t~\S-y|TRyQu|\wW-vWYTP\y_S2{SV{U2SX • ZTSWP2Y\ • PtPXSWPp~t~\ • -i|TRyQu|\wW-R_ZpZp~\SPtp_t|~pyQy_S2}\v_-y{SqW-X_Su}t^\\S_-Q[RtVzxX-V[RpU-vW}t\\S~pY\2VQ|}xV[• <VQ|}xV[• 2w\ • Z-s[• pY\2Y\~_P2	komunikasi atau jaringan, sehingga kerahasiaan data dapat dilindungi. Secara garis besar, kriptografi digunakan untuk mengirim dan menerima pesan. Kriptografi pada dasarnya berpatokan pada kunci yang secara selektif telah disebar pada komputer-komputer yang berada dalam	
7	G	gW • tz\~Q-{[Sx~-~_T • tpPY2\Xy[• {_-vW-xPzt|Sty-pY\~_V2[Sy|TR~T2XTS[ypQR<yQ<XTS[ypQR?mx~ztS-xPT2}XSy|_-y_RxZx_~~WUp~Ty_S2}\v_-wW • tz\~Q-`@m2GSV[• nR]y-A_\VT • W2VQ • [UQW-Q[Pp^|T • W-vW}t\xV}t\\u_T2hTSyxpQ-B|TW_ztCtyy}|U2kB@?mt\X • _ • • w\2VQ • ~XRkB@-Q[SvW • xZp}\VZxyXS_~y_S2YX • Z\ • _t\XS_~p]p|\>SpSx • ~pY\2~xs_p\-xPT2U\ • w\2_UpP-vW_pU\QPtPYtP\x~tPYu{Sp_S2YX • Z\ • PtyQv[-tPUSW~QW-v[USW~QW-sQQUXTQU, --- O~RywW}zPt|xR_UpP-pQY}|TVS\2Vx • ^T2~T|[zSWU2^TRUXS_~RU2w\ • Z-vW • pP]pPY2}\v_-V_VTP-74Q~[V2L • T^X2n]zPXx[• 2{SV{U2SX • ZYpPzxV\ • Jdn2m\v_-Q_\VT	Firewall jenis lain yang bekerja di internet adalah enkripsi firewall-to-firewall. Sistem ini pertama kali diaplikasikan pada firewall ANS InterLock. Saat ini, koneksi semacam ini disebut sebagai Virtual Private Network (VPN). Sebenarnya konsep VPN sendiri dapat diterapkan dengan beberapa cara, namun pada sub bab ini hanya akan dibahas mengenai penggunaan dengan metode enkripsi dekripsi blowfish. Blowfish merupakan algoritma kunci simetrik cipher blok yang dirancang pada tahun 1993 oleh Bruce Schneier untuk menggantikan DES. Pada saat itu	Berhasil

		V{-pQY}|TVS\2{SV{ U2SX • X • V • x}}x}T _zT~t~\ • }pPYpy-s_S X_U>\y_S2yXV_~x~t PYu{Sp_S • w\2~\ • Z\ Vzt|_py\Qx • yxy~T\Rx V2U\~T • W-vWUp|X • _UpP-pYSX_-z_U2} \V[S2Y\ • Rx~X • ~T2{ SV{U2~XVW\R\~ZQS Wz|_-X_SuZxZx • _Up P2n]zPXx[• 2SX • w\V _UpP-s_VU_-sQQUX TQU-s[_p~-R_ztP-v_S 2_UpP-s[• pY\2}\v_-v RPpWS2}xsQTy-B[• • w\V_\ • mqUStWXS} t\\u_T2}X • [PT\~ZQS Wz|_-sQQUXTQU-V[• Q[_Ty-|[Ps{UpYt|_p PY2\\S{-v{Sx_-y|TRy Qu|\wW-TPzTV-v_~py -V[• {{S2Y\ • _t|UtS_p PY2YT2}xsQTy-yUxQ {} • w\2\\uW-|_}X_ • p V\VtpPY2SX|\xV{Vy_ S2_RuR • xyPpUSW~ VRYS_[xtpPY2^XR_z >UT_z>ZpP-VWZpV- V[• z_RpPY2RRtU-~ W}tP}x-B|QQ[}2Y\S W +	algoritma untuk men-enkripsi suatu pesan sangat banyak, akan tetapi penggunaannya sangat terbatas untuk publik, hal ini dikarenakan adnya hak paten dan lisensi untuk setiap algoritma yang digunakan. Schneier menyatakan bahwa blowfish bebas paten dan akan berada pada domain publik. Pernyataan Schneier sebagai penemu algoritma blowfish tersebut membuka gerbang baru dunia kriptografi untuk dapat terjun dan berkembang di publik, khususnya bagi masyarakat yang membutuhkan algoritma kriptografi yang cepat, kuat, dan tidak terhalang oleh lisensi. Proses dari	
8	H	R[SuZx • _\ • PtyQv[-s QQUXTQU-xyx2~X • YTSW-x_RpU-v[Su_S 2SX|}XSU\VWUpP-{[Sx~{[Sx~-Q[• pPYpP- X_SuZp}\VPtS_pU\X_ UpP-v_zp\V_x2T\SWS u_S>RpQx2SX • Z\ • _ Rx~TQPtyQv[-R[Su_P pP\ • -X_SuUtSxvW\ • Zp|T2_SpQTQW}2yX S~Xs{z2SXVRZt\V_x 2}X~_tpP\ • }X~ztS-p} \2w\ • Z-vWzt|\RV\ • Zt	penggunaan metode blowfish itu sendiri ialah dengan memperhatikan jenis-jenis serangan yang dapat membahayakan data atau jaringan, lalu menganalisis metode pengamanan, yang kemudian dari analisis tersebut metode atau pelayanan system apa yang diterapkan	Berhasil

		PYpP-|[PR[• z_zxV\ • Xw[UVW[xy\Q~tPYu{ Sp_S2}X~_tpP\ • tpPY 2YTV[• p}UpP2CXV RZtzt|}t\xVPtPXS_~y_ S2SXy_Sx~PtUSW~V RYS_[x~pY\2SQv[R2 Bmh-c[• v_}p|UpP-Q{ Ps[• >\~ZQSWz|_-sQQ UXTQU-q{UT}-tXXy yTwx • yxyZxZx • _Up P-v_RpS-Qw}V[P2VX pS\ • _S2Y\~_P2VQ • ~ XR[x|XU_R~z}[x|XU_ R~\V_x2hno-cQQUXT QU-pY\~_V2_RuR • x yPpUSW~VRYS_[xU TP]x}xSXV|Ty]x}Vt|-s QQyZtPYpP-R_S{_Su _~RU2yXV_~2~XR_S {_Su;_xy2ORuR • VW Ppzt|}t\xVWTZ\2SX • [• p}UpP-V[U • WU2V x • ^T2w\ • Z-s[• TVxS _S2~X|\\S_Su-DVxS_S 2Vx • ^T2w\ • Z-v_~py -vWzt|T|_-}QXz_~Ry wW}z\v_RpU-pPzp|\2 >?2UT • ZYp:5-sWz>Z tPYpP-TVxS_S2~zpPZ p|-Q[_t~\S<5-sWz?O~ RywW}zPtS\ • X\pyUp P-V[U • WU2}X|_Sx}x ~_}x_S2\TVZpP-V[U • WU2}X|{zp|\ • x~_S uZpP-R[• uWRx|\ • UT P]xtpPY2YT~_UTV\ • }t\\ • w\y<UpQT?Et[[t VzxX\ • _~RywW}zZx QTz_z2YX • Z\ • ~t|Vx yx • Z\ • ~pPWpPY2Vx • ^T=VXXX • ^ • X}z2 SXQ~\u[2ORuR • xyPp xV_Ppzt|_pZT2SX • T\ vW-v{\2~xs\~ZQSWz| _-Ty|_2w\xyx2\\uW\ •	dengan memperhatikan efektifitas penggunaan pelayanan yang diterapkan. Metode tersebut menerapkan mekanisme kriptografi pada model OSI. Berdasarkan sumber, algoritma blowfish cukup efektif untuk digunakan dalam system keamanan dalam konsep firewall to firewall atau VPN. Blowfish adalah algoritma kriptografi kunci simetrik cipher blok dengan panjang blok tetap sepanjang 64 bit. Algortima tersebut juga menerapkan teknik kunci yang berukuran sembarang. Ukuran kunci yang dapat diterima oleh blowfish adalah antara 32 hingga 448 bit, dengan ukuran standar sebesar 128 bit. Blowfish memanfaatkan teknik pemanipulasian bit dan teknik pemutaran ulang dan pergiliran kunci yang dilakukan sebanyak 16 kali. Keefektifan blowfish dilihat dengan perhitungan panjang kunci/key encrypt message. Algoritma utama terbagi menjadi	

		Xy~~pP}xUTP]xZpP-s_Yx_S2[Sy|TR~T<YXy|TR~T2Y\V_2LXSWUTy	dua sub-algoritma utama, yaitu bagian ekspansi kunci dan bagian enkripsi-dekripsi data. Berikut	
9	I	B_Zp_pZTpP-Q{_2\\sT • W-pV\ • ZxTX~_}y_S2VQ • ~XR}t^xSWzXtpPY2Txu_-vWzt|\RV\ • ~pY\2XTS[ypQR2Y\~_P2Uxs{Su_S • w\2YX • Z\ • DB\vY • t~}2w\ • Z-vWYTP\y_S2RRtU-T~XS-hm}t^-xPT2YZp}\VZxZx • _UpP-{{Yp}t\\u_T2_~~WUp~T2Y\SW-wW • tz\~Q-VR-wW • tz\~Q-Qw}V[P2_zp{-GmC2~X~_T • ~tPYu{Sp_S2VQ • ~XR\~ZQSWz|_-sQQUXTQU-y[|_SpP-hm-xPT2YTy[SpQ-vWUtP\~ZtPYpP-hm-A[]T|TVw-:GnQ[]8ZpP-V[RpU-|[S{_Zx}V_Sv_ • x~\QW-y[|_SpP-xPzt|Sty2IT2~T • W-VWZpV-s[Yxyx2\\ • w\yZxTX~_}y_S2VQ • ~XRDB~XqUp|X • _-R[SuZx • _\ • \V_x2_~~WUp~T2w\ • Z-vWzt|\RV\ • x • yxy[x|XU_R~z}[x|XU_R~zxY\yQ\z_pPtpV2GnQ[]2YTv[}pWS2{SV{U2SX~WSv{SuW-yRPTPTy_}xZtPYpP-q_ • pPtPYu{SpV\ • jamhm2m • }yQqRR2GnQ[]2YTy[Ps_SuV\ • Q~[V2GHFH2Y\ • DB~XqPtS_t|Ty_S2Q\X_SpP-V[• z_Zp}-R|TW_]XZpP-p{ztPzxXTy_}xPtPYu{SpV\ • \~ZQSWz|_-y|TR	Pada bagian sub bab ini akan dijelaskan konsep security yang juga diterapkan pada firewall dalam hubungannya dengan IP address yang digunakan oleh user. IPsec ini ddapat digunakan juga sebagai aplikasi dari firewall to firewall system atau VPN selain penggunaan konsep algoritma blowfish keamanan IP ini dikenal dikenal dengan IP Security (IPsec) dan telah menjadi standarisasi keamanan internet. Di sini tidak begitu banyak dijelaskan konsep IPsec karena penggunaan atau aplikasi yang diterapkan untuk firewall to firewall tidaklah banyak. IPsec didesain untuk melindungi komunikasi dengan cara menggunakan TCP/IP. Protocol IPsec dikembangkan oleh IETF, dan IPsec memberikan layanan terhadap privacy dan autentifikasi menggunakan algoritma kriptografi	Berhasil

		yQu|\wW-|RZt|S?h • yx yPtQT • Yx • ZT2Gn2 Y\V_YS_P>Zpy\2YTV | \ • ~[}|Pp~Ty_S2SX • ZYTP\y_S2_RuR • xyP pX • V • x}}x-`Y\2Yxp zx}X2{PTS-v_ • xZp~\ SDB~Xq4-2OxVUX • yTq_zxRS2EXpYXS`E 2OZpQ\z~SRz}^Q~Z} Vx|[S2w\ • Z-|[Rx}xV W-wR • |_z2}\y[z2Y\ • TQ{-TSx|tpPY2\XSUx s{Su_S2YX • Z\ • ~tPY u{Sp_S2OF2{SV{U2} X • ZXQ_VpP-R_Uty2 <?H • ^\R~x~_zxPY2n Xq{ • xyt2m\XQQpY-: KmB-`Y\~_V2YQy{Pt P-X_SuPtQTR{zx	modern. Untuk melindungi IP datagram, data ditransformasikan menggunakan algoritma enkripsi. Ada dua tipe umum dari dasar IPsec: 1). Authentication Header (AH) Adalah protocol dokumen yang meliputi format paket dan isu umum yang berhubungan dengan penggunaan AH untuk pengesahan paket. 2). Encapsulating Security Payload (ESP) Adalah dokumen yang meliputi	
10	J	F_Sv_-V_Su_S2YTuW zpQ-v_~py-vWYTP\y_ S2{SV{U2SX~_UTV\ • ~tS_TVzx_S2~Xq_ • pPpyX|_zx~-s_VU_-v_ zpzxY\yPtPYpQ|W-|R ZxXTy_}x}t^\S_-xQX u_R>}tUT • ZYp_x~\2 YTu{SpV\ • }t\\u_T2~\ ~_V2~\V{-QRRT~T2{ SV{U2SX~_UTV\ • {t| TwWUp~T2Y\V_2ix{ {\ • Zp|T2}X • {Rx~\ • }y|TR~T2WSx\v_RpU -TPzTV-|[Su[zpUxx~S R}t~-R[Ps{\V_S2y\ • Y \2y\ • Z\ • ZxZTV_R2S X • ZYTP\y_S2~TQyX |USW~VRYS_[xH~J|_ R2_zp~-B|QQ[}2}X|\x py\ • zpPZpzpPYpP-v WYxy\~Zx_ypQT2YX • Z\ • ~tS_T_zpP-y{Sq W-R{_~WU2Y\ • UTP	Tanda tangan digital dapat digunakan untuk melakukan pembuktian secara matematis bahwa data tidak mengalami modifikasi secara ilegal, sehingga bisa digunakan sebagai salah satu solusi untuk melakukan verifikasi data. Tujuan dari penulisan skripsi ini adalah untuk mengetahui proses pembuatan tanda tangan digital menggunakan sistem kriptografi ElGamal atas Proses pembuatan tanda tangan digital diawali dengan pembuatan kunci publik dan	Berhasil

		]x~SW{py2m\v_-R|QQ[}2}X|\X • yxy_S2Vx • ^T2YTRWRxU->Y\ • ~>ZpP-vW~xQTz_xQ\ • Z\ • • pU\QW\2-i[PTYTpP-vW]p|T2-i{SqW-R{_~WU2YTyW • xS-y[~pY\2}X • ZTSWP2}XQ_S?nSR}t~-Q[RpPWTySX_-pY\~_V2}XSUTV{Su_S2PT~_T2U\QU-v_ • x}T_zT~t~\ • -n_RTPtST~WV2\T~_Su_S2[->YX • Z\ • YqYt-R<83<2Y\ • PtPYzWzTPY2~XSy\2-fW~t|Q~[V2y\ • Y\2y\ • Z\ • Cj8tpPY2VX|{Zx_S2YTs{_TUUpP-R_ZpZ}Vx|[S2Y\ • ZxVTSWPy_S?mtyX~_V2SX • [• xS\2YQy{PtP2S\y_-R[St|T|_-pV\ • PtS{t|TwWUp~T2y\ • Y\2y\ • Z\ • -v[Su_S2SX • ^\SW- • WRpW-lI-v_VTQx>RpQx2SX • ZXq[U2\\zz\2?02l-~<?2TTy_-V[• R[STUT2Q\ • TxVPtPYzWzTPY2SQv~>}tQ\ • TxVPtpZx}XSWUQ_-s_VU_-|RZ2}2m • }~XQ~tS_T_zpP-y{SqW-|[SuU\QWRy_S2Vx • ^T2}xsQTy	kunci privat. Pada proses pembentukan kunci dipilih ,dan p, dan dipilih bilangan rahasia . Kemudian dicari . Kunci publik dikirim kepada pengirim pesan. Proses selanjutnya adalah perhitungan nilai hash dari suatu pesan. Lalu memilih bilangan e ,dengan gcd(e, p-1)=1 dan menghitung serta . Diperoleh tanda tangan (R,T) yang kemudian dibubuhkan pada dokumen dan dikirimkan. Setelah menerima dokumen, maka penerima akan memverifikasi tanda tangan, dengan mencari nilai MD dahulu, lalu mengecek bahwa 1≤ R ≤ p-1, jika terpenuhi lanjut menghitung mod p, selanjutnya diperiksa bahwa mod p. Proses pembuatan kunci menghasilkan kunci publik	

Tabel 4.5 Data uji skenario 3

No	Data	Isi
1	A	Secara matematis, enkripsi dengan mode CBC dapat dinyatakan sebagai ( ) −1 = Å i K i i C E P C i (1)

		dan dekripsi sebagai 1 ( ) − = Å i K i i P E C C i (2) Untuk melakukan enkripsi/dekripsi pada blok pertama diperlukan C0 yang dalam hal ini C0 adalah initialization vector atau IV (pada algoritma ini IV = '0000'). IV tidak perlu rahasia tetapi harus sama nilainya pada proses dekripsi. Fungsi E yang digunakan di dalam Gambar 2 adalah fungsi sederhana yaitu operasi XOR antara bitbit kunci Ki dengan hasil peng-XOR-an sebelumnya: K i i i E X X K i ( ) = Å (3) yang dalam hal ini Xi = Pi Å Ci – 1 pada skema enkripsi dan Xi = Ci pada skema dekripsi. Mode yang digunakan di dalam algoritma ini disebut CBC-like karena untuk setiap blok menggunakan kunci yang
2	B	berbeda-beda (pada mode CBC yang asli kunci pada setiap blok adalah sama yaitu kunci eksternal K). Kunci Ki pada setiap blok data disebut internal key yang panjangnya juga 4-bit. Kunci internal ini dibangkitkan dari fungsi chaos logistic map, xi + 1 = r xi (1 – xi) (4) dengan 0 £ xi £ 1, i = 0, 1, 2, ... dan 0 £ r £ 4. Nilai awal (seed) iterasi adalah x0 yang berperan sebagai kunci rahasia. Empat bit kunci internal diperoleh sebagai berikut [10]: nilai chaos xi dikalikan dengan 10 berulangkali sampai ia mencapai panjang angka (size) yang diinginkan, selanjutnya potong hasil perkalian tersebut untuk mengambil bagian integernya saja. Secara matematis, nilai chaos x dikonversi ke integer dengan menggunakan persamaan berikut: T(x, size) = x *10 , x ¹ 0 count (5) yang dalam hal ini count dimulai dari 1 dan bertambah 1 hingga x * 10count > 10size – 1.
3	C	Hasilnya kemudian diambil bagian integer saja (dilambangkan dengan pasangan garis ganda pada persamaan 5). Sebagai contoh, misalkan xi = 0.003176501 dan size = 4, maka dimulai dari count = 1 sampai count = 6 diperoleh 0.003176501 * 106 = 3176.501 > 103 kemudian ambil bagian integer-nya dengan 3176.501 = 3176 Empat bit terakhir dari representasi biner 3176 dijadikan sebagai Ki yaitu '1000'.

		C1, C2, …, Cn dari hasil enkripsi selanjutnya menggantikan 4-bit MSB dari setiap pixel yang diproses. Hasil enkripsi terjadap seluruh pixel adalah citra terenkripsi (cipher-image). Untuk proses dekripsi dilakukan proses berkebalikan seperti yang ditunjukkan pada Gambar 2(b). Algoritma di atas dapat dirampatkan untuk citra berwarna. Prosesnya dilakukan tiga kali, masingmasing untuk kanal red (R), green (G), dan blue (B). Jadi, pada setiap byte kanal warna diambil 4-bit MSB kemudian dioperasikan dengan mode CBC secara terpisah.
4	D	Histogram merupakan properti citra yang penting sebab sebuah histogram memperlihatkan distribusi intensitas pixel di dalam citra tersebut. Untuk citra plain-image histogramnya membentuk suatu pola yang khas, yaitu ada puncak-puncak dan lembahlembah. Untuk mencegah penyerang menggunakan histogram untuk melakukan analisis freluensi, maka histogram plain-image dan histogram cipher-image seharusnya tidak memiliki kemiripan secara statistik. Oleh karena itu, histogram cipher-image seharusnya relatif datar (flat) sehingga tahan terhadap serangan statistik. Distribusi yang relatif uniform pada cipherimage adalah sebuah indikasi bahwa algoritma enkripsi citra memiliki kualitas yang bagus [9]. Gambar 6(a) memperlihatkan histogram citra 'Barbara' sebelum dienkripsi, dan Gambar 6(b) adalah histogram cipher-image-nya. Dapat dilihat bahwa histogram cipher-image memiliki distribusi uniform yang mana berbeda dengan histogram plain-image. Gambar 7(a) sampai 7(c) memperlihatkan histogram citra 'Lena' (plain-image) untuk setiap kanal warna RGB dan Gambar 7(d) sampai 7(f) adalah histogram masing-masang kanal warna pada cipherimage. Sama seperti citra 'Barbara', histogram cipherimage pada setiap kanal RGB juga terlihat
5	E	flat atau terdistribusi uniform. Gambar 8(a) memperlihatkan histogram citra 'Taj Mahal' (plain-image) dan Gambar 8(b) adalah histogram cipher-image-nya. Sedikit berbeda dengan histogram cipher-image dari dua citra sebelumnya, histogram cipher-image dari 'Taj Mahal' memiliki distribusi yang relatif uniform. Berdasarkan hasil-hasil analisis histogram di atas dapat disimpulkan bahwa cipher-image memiliki histogram yang (relatif) flat sehingga menyulitkan penyerang melakukan analisis statistik untuk mendeduksi pixel atau kunci. Hasil ini menunjukkan bahwa algoritma enkripsi citra yang diusulkan ini memiliki keamanan yang bagus.

		Algoritma enkripsi citra seharusnya sensitif terhadap kunci. Sensitif artinya jika kunci diubah sedikit saja maka hasil dekripsi terhadap cipher-image menghasilkan cipher-image lain yang berbeda signifikan. Karena algoritma yang diusulkan ini menggunakan sistem chaos, maka sifat chaos yang sensitif terhadap perubahan kecil nilai awal (x0) merupakan properti keamanan yang penting. Nilai x0 berperan sebagai kunci yang diberikan oleh pengguna. Pada eksperimen ini nilai awal logistic map diubah sebesar D sehingga menjadi x0 + D,
6	F	Proses ini hanya dapat dilakukan oleh orang yang menggunakan metode yang sesuai. Proses enkripsi/dekripsi tergantung pada kunci (key) rahasia yang hanya diketahui oleh pengirim dan penerima. Semakin rumit algoritma kriptografi yang digunakan, semakin sulit pula untuk para hacker, cracker yang ingin membobol security system yang digunakan, sehingga komunikasi data antara pengirim dan penerima aman. Kriptografi pada dsarnya digunakan untuk menjamin privasi: mencegah informasi menyebar kepada pihak yang tidak terautentifikasi secara legal untuk mengakses file yang ada pada jaringan. Oleh karena itu kriptografi memiliki hal-hal yang sanagt berkaitan erat pokok yaitu privasi, authentification (memverifikasi identitas pengguna) dan integritas (memastikan bahwa pesan belum diubah). Kriptografi pada firewall digunakan untuk mencegah ancaman dari user yang tidak berhak untuk memasuki komunikasi atau jaringan, sehingga kerahasiaan data dapat dilindungi. Secara garis besar, kriptografi digunakan untuk mengirim dan menerima pesan. Kriptografi pada dasarnya berpatokan pada kunci yang secara selektif telah disebar pada komputer-komputer yang berada dalam
7	G	Firewall jenis lain yang bekerja di internet adalah enkripsi firewall-to-firewall. Sistem ini pertama kali diaplikasikan pada firewall ANS InterLock. Saat ini, koneksi semacam ini disebut sebagai Virtual Private Network (VPN). Sebenarnya konsep VPN sendiri dapat diterapkan dengan beberapa cara, namun pada sub bab ini hanya akan dibahas mengenai penggunaan dengan metode enkripsi dekripsi blowfish. Blowfish merupakan algoritma kunci simetrik cipher blok yang dirancang pada tahun 1993 oleh Bruce Schneier untuk menggantikan DES. Pada saat itu algoritma untuk men-enkripsi suatu pesan sangat banyak, akan tetapi penggunaannya sangat terbatas untuk publik, hal ini dikarenakan adnya hak

		paten dan lisensi untuk setiap algoritma yang digunakan. Schneier menyatakan bahwa blowfish bebas paten dan akan berada pada domain publik. Pernyataan Schneier sebagai penemu algoritma blowfish tersebut membuka gerbang baru dunia kriptografi untuk dapat terjun dan berkembang di publik, khususnya bagi masyarakat yang membutuhkan algoritma kriptografi yang cepat, kuat, dan tidak terhalang oleh lisensi. Proses dari
8	H	penggunaan metode blowfish itu sendiri ialah dengan memperhatikan jenis-jenis serangan yang dapat membahayakan data atau jaringan, lalu menganalisis metode pengamanan, yang kemudian dari analisis tersebut metode atau pelayanan system apa yang diterapkan dengan memperhatikan efektifitas penggunaan pelayanan yang diterapkan. Metode tersebut menerapkan mekanisme kriptografi pada model OSI. Berdasarkan sumber, algoritma blowfish cukup efektif untuk digunakan dalam system keamanan dalam konsep firewall to firewall atau VPN. Blowfish adalah algoritma kriptografi kunci simetrik cipher blok dengan panjang blok tetap sepanjang 64 bit. Algortima tersebut juga menerapkan teknik kunci yang berukuran sembarang. Ukuran kunci yang dapat diterima oleh blowfish adalah antara 32 hingga 448 bit, dengan ukuran standar sebesar 128 bit. Blowfish memanfaatkan teknik pemanipulasian bit dan teknik pemutaran ulang dan pergiliran kunci yang dilakukan sebanyak 16 kali. Keefektifan blowfish dilihat dengan perhitungan panjang kunci/key encrypt message. Algoritma utama terbagi menjadi dua sub-algoritma utama, yaitu bagian ekspansi kunci dan bagian enkripsi-dekripsi data. Berikut
9	I	Pada bagian sub bab ini akan dijelaskan konsep security yang juga diterapkan pada firewall dalam hubungannya dengan IP address yang digunakan oleh user. IPsec ini ddapat digunakan juga sebagai aplikasi dari firewall to firewall system atau VPN selain penggunaan konsep algoritma blowfish keamanan IP ini dikenal dikenal dengan IP Security (IPsec) dan telah menjadi standarisasi keamanan internet. Di sini tidak begitu banyak dijelaskan konsep IPsec karena penggunaan atau aplikasi yang diterapkan untuk firewall to firewall tidaklah banyak. IPsec didesain untuk melindungi komunikasi dengan cara menggunakan TCP/IP. Protocol IPsec dikembangkan oleh IETF, dan IPsec memberikan layanan terhadap privacy dan autentifikasi menggunakan algoritma kriptografi modern. Untuk melindungi IP datagram, data

		ditransformasikan menggunakan algoritma enkripsi. Ada dua tipe umum dari dasar IPsec: 1). Authentication Header (AH) Adalah protocol dokumen yang meliputi format paket dan isu umum yang berhubungan dengan penggunaan AH untuk pengesahan paket. 2). Encapsulating Security Payload (ESP) Adalah dokumen yang meliputi
10	J	Tanda tangan digital dapat digunakan untuk melakukan pembuktian secara matematis bahwa data tidak mengalami modifikasi secara ilegal, sehingga bisa digunakan sebagai salah satu solusi untuk melakukan verifikasi data. Tujuan dari penulisan skripsi ini adalah untuk mengetahui proses pembuatan tanda tangan digital menggunakan sistem kriptografi ElGamal atas Proses pembuatan tanda tangan digital diawali dengan pembuatan kunci publik dan kunci privat. Pada proses pembentukan kunci dipilih ,dan p, dan dipilih bilangan rahasia . Kemudian dicari . Kunci publik dikirim kepada pengirim pesan. Proses selanjutnya adalah perhitungan nilai hash dari suatu pesan. Lalu memilih bilangan e ,dengan gcd(e, p-1)=1 dan menghitung serta . Diperoleh tanda tangan (R,T) yang kemudian dibubuhkan pada dokumen dan dikirimkan. Setelah menerima dokumen, maka penerima akan memverifikasi tanda tangan, dengan mencari nilai MD dahulu, lalu mengecek bahwa $1 \leqslant R \leqslant$ p-1, jika terpenuhi lanjut menghitung mod p, selanjutnya diperiksa bahwa mod p. Proses pembuatan kunci menghasilkan kunci publik

Dari tabel pengujian 4.4 di atas, dapat di hasilkan nilai akurasi sebagai berikut :

$$\text{Akurasi} = \frac{\Sigma\, berhasil - \Sigma gagal}{\Sigma data} \times 100\%$$

$$\text{Akurasi} = \frac{10-0}{10} \times 100\% = 100\%$$

4.1.7 Pengujian Akurasi Gambar

Pada pengujian gambar ini akan dilakukan 3 kali pengujian proses enkripsi dan dekripsi pada 3 ukuran gambar yang berbeda untuk dapat melihat berapa tingkat akurasi pengembalian pada ukuran *pixels* dan ukuran *file* tersebut. Sehingga didapatkan hasil sebagai berikut :

Pengujian 1

Tabel 4.6 Pengujian akurasi dekripsi gambar 1

No	Data uji		Hasil enkripsi		Hasil dekripsi		Keterangan
	Ukuran *pixels*	**Ukuran *file***	**Ukuran *pixels***	**Ukuran *file***	**Ukuran *pixels***	**Ukuran *file***	
1	100x100	10.1 kb	-	10.1 kb	100x100	10.1 kb	Berhasil
2	300x300	39.3 kb	-	39.3 kb	300x300	39.3 kb	Berhasil
3	600x600	91.6 kb	-	91.6 kb	600x600	91.6 kb	Berhasil

Dari tabel pengujian 4.6 di atas didapatkan tingkat akurasi pengembalian ukuran *pixels* dan ukuran *file* sebagai berikut :

Akurasi = $\frac{\Sigma\, berhasil - \Sigma\, gagal}{3} \times 100\%$

Akurasi = $\frac{3-0}{3} \times 100\% = 100\%$

Penguijan 2

Tabel 4.7 Pengujian akurasi dekripsi gambar 2

No	Data uji		Hasil enkripsi		Hasil dekripsi		Keterangan
	Ukuran *pixels*	**Ukuran *file***	**Ukuran *pixels***	**Ukuran *file***	**Ukuran *pixels***	**Ukuran *file***	
1	100x100	10.1 kb	-	10.1 kb	100x100	10.1 kb	Berhasil
2	300x300	39.3 kb	-	39.3 kb	300x300	39.3 kb	Berhasil
3	600x600	91.6 kb	-	91.6 kb	600x600	91.6 kb	Berhasil

Dari tabel pengujian 4.7 di atas didapatkan tingkat akurasi pengembalian ukuran *pixels* dan ukuran *file* sebagai berikut :

Akurasi = $\frac{\Sigma\, berhasil - \Sigma\, gagal}{3} \times 100\%$

Akurasi = $\frac{3-0}{3} \times 100\% = 100\%$

Pengujian 3

Tabel 4.8 Pengujian akurasi dekripsi gambar 3

No	Data uji		Hasil enkripsi		Hasil dekripsi		Keterangan
	Ukuran *pixels*	**Ukuran *file***	**Ukuran *pixels***	**Ukuran *file***	**Ukuran *pixels***	**Ukuran *file***	
1	100x100	10.1 kb	-	10.1 kb	100x100	10.1 kb	Berhasil
2	300x300	39.3 kb	-	39.3 kb	300x300	39.3 kb	Berhasil
3	600x600	91.6 kb	-	91.6 kb	600x600	91.6 kb	Berhasil

Dari tabel pengujian 4.8 di atas didapatkan tingkat akurasi pengembalian ukuran *pixels* dan ukuran *file* sebagai berikut :

Akurasi = $\frac{\Sigma\, berhasil - \Sigma\, gagal}{3} \times 100\%$

Akurasi = $\frac{3-0}{3} \times 100\% = 100\%$

BAB V
KESIMPILAN DAN SARAN

5.1 Kesimpulan

Berdasarkan implementasi dan pengujian yang telah dilakukan, dapat di simpulkan sebagai berikut :

1. Web portal yang terbentuk telah dapat menyediakan fasilitas enkripsi dan dekripsi pada *file* *.docx, *.txt, *.jpg dan *.png dengan menggunakan metode *Cipher Block Chaining* (CBC)
2. Tingkat akurasi pengembalian kalimat dari 3 skenario pengujian (10 kata, 50 kata dan 150 kata) adalah 100%
3. Tingkat akurasi pengembalian *file* untuk ukuran *pixels* dan ukuran *file* pada gambar dari 3 data yang diuji (*file* resolusi 100 x 100 px dengan ukuran *file* 10.1 kb, *file* resolusi 300 x 300 px dengan ukuran *file* 39.3 kb dan *file* resolusi 600 x 600 px dengan ukuran *file* 91.6 kb) adalah 100%
4. Proses enkripsi dan dekripsi pada *file* *.docx yang mengandung unsur gambar dan tabel tidak dapat mengembalikan gambar dan bentuk tabel seperti pada saat sebelum dienkripsi atau *plaintext* awal.

5.2 Saran

Berdasarkan dari pengalaman dalam proses pembuatan aplikasi ini, terdapat beberapa saran yang diusulkan untuk pengembangan sistem lebih lanjut. Antara lain:

1. Agar dapat dikembangkan dengan menambah ekstensi *file* yang dapat di enkripsi maupun didekripsi.
2. Dapat dikembangkan pada sistem operasi *android* atau *iOS.*

DAFTAR PUSTAKA

Lutfillah, F. (2015). *Implementasi Kriptografi Blowfish Pada Sebuah Informasi Dalam Bentuk QR Code*. Universitas Muhammadiyah Jember, Jember.

Mardianto. (2010). *Enkripsi Curve Cryptography (ECC).* Institut Teknologi Telkom, Bandung.

Mulyanta S. Edi (2006). *Dari Teori Hingga Praktik : Pengolahan Digital Image dengan Photoshop CS2*. ANDI, Yogyakarta.

Munir, R. (2006). *Kriptografi.* Informatika, Bandung.

Purwono. (2009). *Buku Materi Pokok: Dasar-dasar Dokumentasi*. Universitas Terbuka. Modul 1, Jakarta.

Putra, D. (2010). *Pengolahan Citra Digital.* ANDI, Yogyakarta.

Rosmala, D. (2012). *Implementasi Mode Operasi Cipher Block Chaining (CBC) Pada Pengamanan Data.* Institut Teknologi Nasional Bandung, Bandung.

Rudianto,A. M. (2011). *Pemrograman Web Dinamis Menggunakan Php dan Mysql*. ANDI, Yogyakarta.

Wisnu, R. (2008). *Implementasi Algoritma RC6 Untuk Enkripsi SMS pada Telepon Seluler.* Institut Teknologi Bandung, Bandung.

LAMPIRAN

Lampiran I

TABEL ASCII

DEC	OCT	HEX	BIN	Symbol	HTML Number
0	000	00	00000000	NUL	&#000;
1	001	01	00000001	SOH	&#001;
2	002	02	00000010	STX	&#002;
3	003	03	00000011	ETX	&#003;
4	004	04	00000100	EOT	&#004;
5	005	05	00000101	ENQ	&#005;
6	006	06	00000110	ACK	&#006;
7	007	07	00000111	BEL	&#007;
8	010	08	00001000	BS	&#008;
9	011	09	00001001	HT	
10	012	0A	00001010	LF
11	013	0B	00001011	VT	&#011;
12	014	0C	00001100	FF	
13	015	0D	00001101	CR
14	016	0E	00001110	SO	&#014;
15	017	0F	00001111	SI	&#015;
16	020	10	00010000	DLE	&#016;
17	021	11	00010001	DC1	&#017;
18	022	12	00010010	DC2	&#018;
19	023	13	00010011	DC3	&#019;
20	024	14	00010100	DC4	&#020;
21	025	15	00010101	NAK	&#021;

Lampiran I

DEC	OCT	HEX	BIN	Symbol	HTML Number
22	026	16	00010110	SYN	&#022;
23	027	17	00010111	ETB	&#023;
24	030	18	00011000	CAN	&#024;
25	031	19	00011001	EM	&#025;
26	032	1A	00011010	SUB	&#026;
27	033	1B	00011011	ESC	&#027;
28	034	1C	00011100	FS	&#028;
29	035	1D	00011101	GS	&#029;
30	036	1E	00011110	RS	&#030;
31	037	1F	00011111	US	&#031;
32	040	20	00100000		&#32;
33	041	21	00100001	!	&#33;
34	042	22	00100010	"	&#34;
35	043	23	00100011	#	&#35;
36	044	24	00100100	$	&#36;
37	045	25	00100101	%	&#37;
38	046	26	00100110	&	&#38;
39	047	27	00100111	'	&#39;
40	050	28	00101000	(	&#40;
41	051	29	00101001	)	&#41;
42	052	2A	00101010	*	&#42;
43	053	2B	00101011	+	&#43;
44	054	2C	00101100	,	&#44;
45	055	2D	00101101	-	&#45;
46	056	2E	00101110	.	&#46;

Lampiran I

DEC	OCT	HEX	BIN	Symbol	HTML Number
47	057	2F	00101111	/	&#47;
48	060	30	00110000	0	&#48;
49	061	31	00110001	1	&#49;
50	062	32	00110010	2	&#50;
51	063	33	00110011	3	&#51;
52	064	34	00110100	4	&#52;
53	065	35	00110101	5	&#53;
54	066	36	00110110	6	&#54;
55	067	37	00110111	7	&#55;
56	070	38	00111000	8	&#56;
57	071	39	00111001	9	&#57;
58	072	3A	00111010	:	&#58;
59	073	3B	00111011	;	&#59;
60	074	3C	00111100	<	&#60;
61	075	3D	00111101	=	&#61;
62	076	3E	00111110	>	&#62;
63	077	3F	00111111	?	&#63;
64	100	40	01000000	@	
65	101	41	01000001	A	&#65;
66	102	42	01000010	B	&#66;
67	103	43	01000011	C	&#67;
68	104	44	01000100	D	&#68;
69	105	45	01000101	E	&#69;
70	106	46	01000110	F	&#70;
71	107	47	01000111	G	&#71;
72	110	48	01001000	H	&#72;
73	111	49	01001001	I	&#73;

Lampiran I

DEC	OCT	HEX	BIN	Symbol	HTML Number
74	112	4A	01001010	J	&#74;
75	113	4B	01001011	K	&#75;
76	114	4C	01001100	L	&#76;
77	115	4D	01001101	M	&#77;
78	116	4E	01001110	N	&#78;
79	117	4F	01001111	O	&#79;
80	120	50	01010000	P	&#80;
81	121	51	01010001	Q	&#81;
82	122	52	01010010	R	&#82;
83	123	53	01010011	S	&#83;
84	124	54	01010100	T	&#84;
85	125	55	01010101	U	&#85;
86	126	56	01010110	V	&#86;
87	127	57	01010111	W	&#87;
88	130	58	01011000	X	&#88;
89	131	59	01011001	Y	&#89;
90	132	5A	01011010	Z	&#90;
91	133	5B	01011011	[	&#91;
92	134	5C	01011100	\	&#92;
93	135	5D	01011101	]	&#93;
94	136	5E	01011110	^	&#94;
95	137	5F	01011111	_	&#95;
96	140	60	01100000	`	&#96;
97	141	61	01100001	a	&#97;
98	142	62	01100010	b	&#98;
99	143	63	01100011	c	&#99;
100	144	64	01100100	d	&#100;

Lampiran I

DEC	OCT	HEX	BIN	Symbol	HTML Number
101	145	65	01100101	e	&#101;
102	146	66	01100110	f	&#102;
103	147	67	01100111	g	&#103;
104	150	68	01101000	h	&#104;
105	151	69	01101001	i	&#105;
106	152	6A	01101010	j	&#106;
107	153	6B	01101011	k	&#107;
108	154	6C	01101100	l	&#108;
109	155	6D	01101101	m	&#109;
110	156	6E	01101110	n	&#110;
111	157	6F	01101111	o	&#111;
112	160	70	01110000	p	&#112;
113	161	71	01110001	q	&#113;
114	162	72	01110010	r	&#114;
115	163	73	01110011	s	&#115;
116	164	74	01110100	t	&#116;
117	165	75	01110101	u	&#117;
118	166	76	01110110	v	&#118;
119	167	77	01110111	w	&#119;
120	170	78	01111000	x	&#120;
121	171	79	01111001	y	&#121;
122	172	7A	01111010	z	&#122;
123	173	7B	01111011	{	&#123;
124	174	7C	01111100	|	&#124;
125	175	7D	01111101	}	&#125;
126	176	7E	01111110	~	&#126;
127	177	7F	01111111		&#127;

Lampiran I

DEC	OCT	HEX	BIN	Symbol	HTML Number
128	200	80	10000000	€	&#128;
129	201	81	10000001		
130	202	82	10000010	‚	&#130;
131	203	83	10000011	ƒ	&#131;
132	204	84	10000100	„	&#132;
133	205	85	10000101	…	&#133;
134	206	86	10000110	†	&#134;
135	207	87	10000111	‡	&#135;
136	210	88	10001000	ˆ	&#136;
137	211	89	10001001	‰	&#137;
138	212	8A	10001010	Š	&#138;
139	213	8B	10001011	‹	&#139;
140	214	8C	10001100	Œ	&#140;
141	215	8D	10001101		
142	216	8E	10001110	Ž	&#142;
143	217	8F	10001111		
144	220	90	10010000		
145	221	91	10010001	‘	&#145;
146	222	92	10010010	’	&#146;
147	223	93	10010011	“	&#147;
148	224	94	10010100	”	&#148;
149	225	95	10010101	•	&#149;
150	226	96	10010110	–	&#150;
151	227	97	10010111	—	&#151;
152	230	98	10011000	˜	&#152;
153	231	99	10011001	™	&#153;
154	232	9A	10011010	š	&#154;

Lampiran I

DEC	OCT	HEX	BIN	Symbol	HTML Number
155	233	9B	10011011	›	&#155;
156	234	9C	10011100	œ	&#156;
157	235	9D	10011101		
158	236	9E	10011110	ž	&#158;
159	237	9F	10011111	Ÿ	&#159;
160	240	A0	10100000		
161	241	A1	10100001	¡	&#161;
162	242	A2	10100010	¢	&#162;
163	243	A3	10100011	£	&#163;
164	244	A4	10100100	¤	&#164;
165	245	A5	10100101	¥	&#165;
166	246	A6	10100110	¦	&#166;
167	247	A7	10100111	§	&#167;
168	250	A8	10101000	¨	&#168;
169	251	A9	10101001	©	&#169;
170	252	AA	10101010	ª	&#170;
171	253	AB	10101011	«	&#171;
172	254	AC	10101100	¬	&#172;
173	255	AD	10101101		&#173;
174	256	AE	10101110	®	&#174;
175	257	AF	10101111	¯	&#175;
176	260	B0	10110000	°	&#176;
177	261	B1	10110001	±	&#177;
178	262	B2	10110010	²	&#178;
179	263	B3	10110011	³	&#179;
180	264	B4	10110100	´	&#180;
181	265	B5	10110101	µ	&#181;

Lampiran I

DEC	OCT	HEX	BIN	Symbol	HTML Number
182	266	B6	10110110	¶	&#182;
183	267	B7	10110111	·	&#183;
184	270	B8	10111000	¸	&#184;
185	271	B9	10111001	¹	&#185;
186	272	BA	10111010	º	&#186;
187	273	BB	10111011	»	&#187;
188	274	BC	10111100	¼	&#188;
189	275	BD	10111101	½	&#189;
190	276	BE	10111110	¾	&#190;
191	277	BF	10111111	¿	&#191;
192	300	C0	11000000	À	&#192;
193	301	C1	11000001	Á	&#193;
194	302	C2	11000010	Â	&#194;
195	303	C3	11000011	Ã	&#195;
196	304	C4	11000100	Ä	&#196;
197	305	C5	11000101	Å	&#197;
198	306	C6	11000110	Æ	&#198;
199	307	C7	11000111	Ç	&#199;
200	310	C8	11001000	È	&#200;
201	311	C9	11001001	É	&#201;
202	312	CA	11001010	Ê	&#202;
203	313	CB	11001011	Ë	&#203;
204	314	CC	11001100	Ì	&#204;
205	315	CD	11001101	Í	&#205;
206	316	CE	11001110	Î	&#206;
207	317	CF	11001111	Ï	&#207;
208	320	D0	11010000	Ð	&#208;

Lampiran I

DEC	OCT	HEX	BIN	Symbol	HTML Number
209	321	D1	11010001	Ñ	&#209;
210	322	D2	11010010	Ò	&#210;
211	323	D3	11010011	Ó	&#211;
212	324	D4	11010100	Ô	&#212;
213	325	D5	11010101	Õ	&#213;
214	326	D6	11010110	Ö	&#214;
215	327	D7	11010111	×	&#215;
216	330	D8	11011000	Ø	&#216;
217	331	D9	11011001	Ù	&#217;
218	332	DA	11011010	Ú	&#218;
219	333	DB	11011011	Û	&#219;
220	334	DC	11011100	Ü	&#220;
221	335	DD	11011101	Ý	&#221;
222	336	DE	11011110	Þ	&#222;
223	337	DF	11011111	ß	&#223;
224	340	E0	11100000	à	&#224;
225	341	E1	11100001	á	&#225;
226	342	E2	11100010	â	&#226;
227	343	E3	11100011	ã	&#227;
228	344	E4	11100100	ä	&#228;
229	345	E5	11100101	å	&#229;
230	346	E6	11100110	æ	&#230;
231	347	E7	11100111	ç	&#231;
232	350	E8	11101000	è	&#232;
233	351	E9	11101001	é	&#233;
234	352	EA	11101010	ê	&#234;
235	353	EB	11101011	ë	&#235;

Lampiran I

DEC	OCT	HEX	BIN	Symbol	HTML Number
236	354	EC	11101100	ì	&#236;
237	355	ED	11101101	í	&#237;
238	356	EE	11101110	î	&#238;
239	357	EF	11101111	ï	&#239;
240	360	F0	11110000	ð	&#240;
241	361	F1	11110001	ñ	&#241;
242	362	F2	11110010	ò	&#242;
243	363	F3	11110011	ó	&#243;
244	364	F4	11110100	ô	&#244;
245	365	F5	11110101	õ	&#245;
246	366	F6	11110110	ö	&#246;
247	367	F7	11110111	÷	&#247;
248	370	F8	11111000	ø	&#248;
249	371	F9	11111001	ù	&#249;
250	372	FA	11111010	ú	&#250;
251	373	FB	11111011	û	&#251;
252	374	FC	11111100	ü	&#252;
253	375	FD	11111101	ý	&#253;
254	376	FE	11111110	þ	&#254;
255	377	FF	11111111	ÿ	&#255;

Lampiran II

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH JEMBER

Jl. Karimata 49 Telp. (0331) 336728 Fax. (0331) 337957 Kotak Pos 104 Jember 68121

DAFTAR REVISI PENGUJI 1 SIDANG TUGAS AKHIR

Nama Mahasiswa	:**HARI SETIADY WIBOWO**
Nomor Induk Mahasiswa	:**1210651167**
Judul Tugas Akhir	: **PORTAL *ONLINE* PENYEDIA FASILITAS ENKRIPSI DAN DESKRIPSI MENGGUNAKAN METODE *CHIPER BLOCK CHAINING* (CBC)**
Hari / Tanggal	:**Senin / 10 April 2017**
Jam	:**10:00 WIB**
Tempat	: **CC 2.2**

Bab/Halaman	Uraian	Keterangan
	abstrak . diperbaiki	
	tata tulis diperbaiki	
	manfaat dan batasan disesuaikan	
	referensi ditambahkan .	
	tabel ada penjelasan gunanya .	
	flowchart diperbaiki	
	simpulan disesuaikan .	

Dosen Penguji 1

AGUNG NILOGIRI, S.T, M.Kom.

NB : Untuk Mahasiswa

Lampiran II

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH JEMBER

Jl. Karimata 49 Telp. (0331) 336728 Fax. (0331) 337957 Kotak Pos 104 Jember 68121

DAFTAR REVISI PENGUJI 2 SIDANG TUGAS AKHIR

Nama Mahasiswa	:HARI SETIADY WIBOWO
Nomor Induk Mahasiswa	:1210651167
Judul Tugas Akhir	: PORTAL *ONLINE* PENYEDIA FASILITAS ENKRIPSI DAN DESKRIPSI MENGGUNAKAN METODE *CHIPER BLOCK CHAINING* (CBC)
Hari / Tanggal	:Senin/ 10 April 2017
Jam	:10:00 WIB
Tempat	: CC 2.2

Bab/Halaman	Uraian	Keterangan
I → IV.	Batasan Masalah → * dan	Oke.
	tabel & gbr → perlu di ujicoba.	
IV.	Uji keamanan pada hasil enkripsi dgn	Oke.
	menggunakan tool yg berbeda.	
All Bab.	Tata tulis : spasi, typo . dll .	Oke.

10/4/17.

4/4/17. Dosen Penguji 2

LUTFI ALI MUHARROM, S.Si, M.Si

NB : Untuk Mahasiswa

IDENTITAS PENULIS

NIM	: 121 065 1167
Nama	: Hari Setiady Wibowo
TTL	: Makassar, 9 Maret 1993
Jenis Kelamin	: Laki-laki
Alamat	: Jl. Slamet Riyadi Gg. Central Blok C-21, Patrang, Jember
Email	: *harisetiadyw@gmail.com*
Nomor Telepon	: 0811 3588 693

Pendidikan Formal

Jenjang	**Nama Institusi**	**Jurusan**	**Tahun Masuk - Lulus**
TK	TK Kartika V-92, Surabaya		1997 – 1999
SD	04 Negeri Bandulan, Malang		1999 – 2005
SMP	6 Negeri Bondowoso		2005 – 2008
SMA	1 Negeri Bondowoso	IPA	2008 – 2011
PT	Universitas Muhammadiyah Jember	Tek.Informatika	2012 - 2017