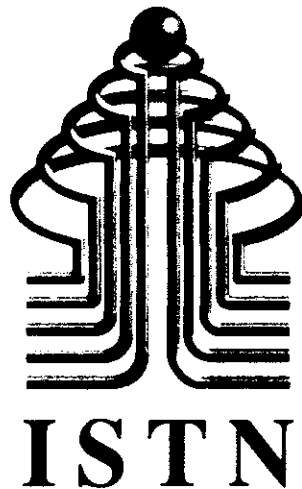


SKRIPSI
PERANCANGAN APLIKASI KRIPTOGRAFI ALGORITMA BASE64
BERBASIS WEB

Disusun oleh :
Himawan Dewabrata
10364016

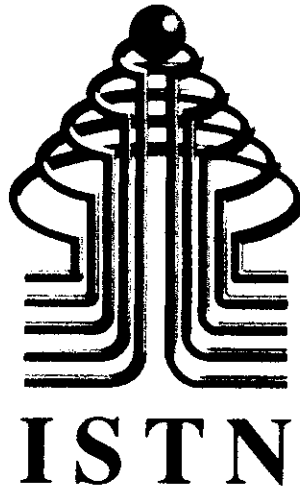


PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT SAINS DAN TEKNOLOGI NASIONAL
JAKARTA SELATAN

2015

SKRIPSI
PERANCANGAN APLIKASI KRIPTOGRAFI ALGORITMA BASE64
BERBASIS WEB

Disusun oleh :
Himawan Dewabrata
10364016



Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Teknik Informatika (S.TI)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT SAINS DAN TEKNOLOGI NASIONAL
JAKARTA SELATAN

2015

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : Himawan Dewabrata
NIM : 10364016
Program Studi : Teknik Informatika
Konsentrasi Peminatan : *Security*

Dengan ini menyatakan bahwa skripsi yang telah saya buat dengan judul : **“Perancangan Aplikasi Kriptografi Algoritma Base64 Berbasis Web”** adalah hasil karya sendiri, dan semua sumber baik yang kutip maupun yang di rujuk telah saya nyatakan dengan benar dan skripsi belum pernah di terbitkan atau di publikasikan di manapun dalam bentuk apapun.

Demikian surat pernyataan ini saya buat dengan sebenar benarnya. Apabila dikemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa skripsi yang telah saya buat adalah hasil karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari Program Studi Teknik Informatika Institut Sains dan Teknologi Nasional di cabut/dibatalkan.

Jakarta, 28 Agustus 2015

Yang menyatakan



Himawan Dewabrata

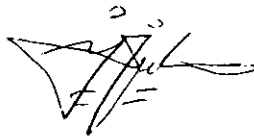
SURAT PERSETUJUAN PEMBIMBING

Yang bertanda tangan dibawah ini :

NIM : 10364016
Nama Mahasiswa : Himawan Dewabrata
Program Studi : Teknik Informatika
Peminatan : Komputer Security
Jenjang pendidikan : Strata satu (S1)
Judul Skripsi : Perancangan Aplikasi Kriptografi
Algoritma Base64 Berbasis Web

Dinyatakan SELESAI / BELUM SELESAI melakukan penulisan tugas akhir, sehingga yang bersangkutan DIIZINKAN / TIDAK DIIZINKAN mengikuti sidang Yodisium atau sidang ujian tugas akhir/Skripsi.

Pembimbing I



Drs. Syarif Hidayatullah, M.Kom
NIDN 03.22026401

Pembimbing II



Siti Madinah L, S.Kom, M.Kom
NIDN 0307107201

Mengetahui

Kepala Program Studi Teknik Informatika FMIPA ISTN



Drs. Syarif Hidayatullah, M.Kom
NIDN 03.22026401

HALAMAN PENGESAHAN

Skripsi ini di ajukan oleh:

Nama Mahasiswa : Himawan Dewabrata
NIM : 10364016
Program Studi : Teknik Informatika
Peminatan : Komputer Scurity
Judul Skripsi : Perancangan Aplikasi Kriptografi
Algoritma Base64 Berbasis Web

Telah di uji dan di nyatakan lulus oleh panitia penguji tanggal: _____

Jakarta, 29 Agustus 2015

Penguji I



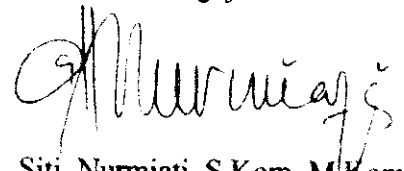
Ir. Andi Suprianto, M.Kom
NIDN 0327025904

Penguji II



Ir. Dadang Rusmana, M.Kom
NIDN 0310125602

Penguji III



Siti Nurmiati, S.Kom, M.Kom
NIDN 0402107703

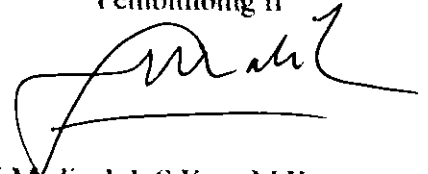
DI SETUJUI OLEH:

Pembimbing I



Drs. Syarif Hidayatullah, M.Kom
NIDN 0322026401

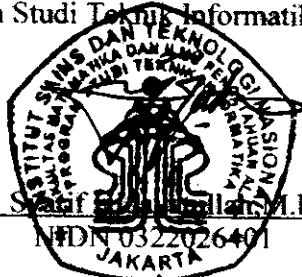
Pembimbing II



Siti Madinah L, S.Kom, M.Kom
NIDN 0307107201

Mengetahui

Kepala Program Studi Teknik Informatika FMIPA ISTN



Drs. Syarif Hidayatullah, M.Kom
NIDN 0322026401

**SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH
UNTUK KEPENTINGAN AKADEMIS**

Yang bertanda tangan dibawah ini, saya :

Nama : Himawan Dewabrata
NIM : 10364016
Program Studi : Teknik Informatika
Peminatan : Security
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, dengan ini menyetujui untuk memberikan ijin kepada pihak Program Studi Teknik Informatika Institut Sains dan Teknologi Nasional **Hak Bebas Royalti Non-Eksklusif (*Non exclusive Royalti Free Right*)** atas karya ilmiah kami yang berjudul : **“Perancangan Aplikasi Website Encode dan Decode File Teks dan Gambar Berbasis Algoritma Base64”**

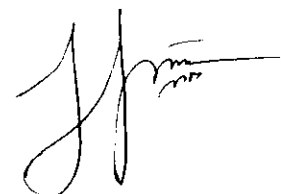
Dengan **Hak Bebas Royalti Non-Eksklusif** ini pihak Institut Sains dan Teknologi Nasional berhak menyimpan, mengalih media atau bentukkan, mengolahnya dalam pangkalan data (*database*), mendistribusikannya dan menampilkan atau mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa perlu minta izin dari kami selama tetap mencantumkan nama kami sebagai penulis/pencipta karya ilmiah tersebut.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak ISTN segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini saya buat dengan sebenar benarnya.

Jakarta, 28 Agustus 2015

Yang menyatakan



Himawan Dewabrata

ABSTRACT

Data security is important to keep its secret. One way to protect a data is with encrypting the data. Encrypting is the transformation process from the original data (plaintext) to encoded form (ciphertext) so it is cannot be recognized as a word or phrase again. While transformation process from ciphertext to plaintext called as decrypting. Input base 64 in form binary data block in various extension such as txt, jpg, png, gif etc, ASCII, and even graphics and sound. administrator has a full responsibility to all of the resource that provide in a network system, including protect the source code which in the web server. The usage of system security application facility in default cannot be guaranteed to save source code they have. This research is focused at a method expansion by designing a function that can be used to encrypted source code in web server based on base64 RFC 2045 and RFC 4648. A function, which has ability to encrypt and decrypt various type of file, such as character, graphics, etc. So that only certain people are capability to process the files Using cryptographic base64 algorithm applications.

Keywords : Encode, Decode, Plaintext, Ciphertext, BASE64 Algorithm.

DAFTAR ISI

Sampul	i
Judul	ii
Lembar Pernyataan Orisinalitas.....	iii
Lembar Persetujuan Pembimbing.....	iv
Lembar Pengesahan	v
Lembar Pernyataan Persetujuan Publikasi Karya Ilmiah	vi
Lembar Persembahan.....	vii
Abstraksi	viii
Abstract	ix
Kata Pengantar	x
Daftar Isi	xi
Daftar Tabel	xiv
Daftar Gambar	xv

BAB I PENDAHULUAN

1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	4

BAB II TINJAUAN PUSTAKA

2.1. Kriptografi	5
2.1.1. Tujuan Kriptografi.....	5
2.2. Algoritma BASE64	6
2.3. Kode ASCII	7
2.4. Bitmap	8
2.5. Request For Comments (RFC).....	9
2.5.1. Sejarah RFC	9
2.6. PHP5.....	10

2.7. HTML5	10
2.8. JavaScript.....	10
2.9. Flowchart	11
2.10. Pingdom.....	12

BAB III METODOLOGI PENELITIAN

3.1. Metodologi Penelitian	13
3.1.1. Objek Penelitian	13
3.1.2. Proses Pengumpulan Data	13
3.1.3. Teknik Pembuatan Aplikasi Website	13
3.2. Perangkat Keras dan Perangkat Lunak yang Digunakan.....	14
3.3. Instalasi dan Integrasi Perangkat Pendukung	14
3.4. Analisa	18
3.4.1 Analisa Algoritma	18
3.5. Metode Perancangan Aplikasi	20
3.5.1. Alur Program Enkripsi Base64 Teks.....	20
3.5.2. Alur Program Dekripsi Base64 Teks.....	21
3.5.3. Alur Program Enkripsi Base64 File Gambar	23
3.5.4. Alur Program Dekripsi Base64 File Gambar.....	25
3.6. Rancangan Aplikasi.....	26

BAB IV HASIL DAN PEMBAHASAN

4.1. Peralatan Pengujian Sistem.....	29
4.2. HASIL	29
4.2.1. Simulasi Enkripsi dan Dekripsi Base64 Pada Teks	29
4.2.2. Simulasi Enkripsi Base64 Pada File Gambar	32
4.3. Pengujian dan Pembahasan.....	34
4.3.1. Uji Coba Enkripsi dan Dekripsi BASE64 Teks	34
4.3.2. Uji Coba Enkripsi BASE64 File Gambar.....	37

KATA PENGANTAR

Alhamdulillah atas segala nikmat iman, Islam, kesempatan, serta kekuatan yang telah diberikan Allah *Subhanahuwata'ala* sehingga Penulis dapat menyelesaikan skripsi ini. *Shalawat* beriring salam untuk tuntunan dan suri tauladan Rasulullah *Shallallahu'alaihiwasallam* beserta keluarga dan sahabat beliau.

Skripsi ini dibuat sebagai salah satu syarat untuk menempuh sidang Sarjana guna mencapai gelar kesarjanaan pada Jurusan Teknik Informatika Di Institut Sains dan Teknologi Nasional. Judul skripsi ini adalah “**Perancangan Aplikasi Kriptografi Algoritma Base64 Berbasis Web**”.

Penulisan Skripsi ini tidak lepas dari dukungan bantuan dan bimbingan dosen pembimbing Drs. Syarif Hidayatullah, M.Kom dan Ibu Siti Madinah L, S.Kom, M.Kom. Dalam kesempatan ini penulis mengucapkan terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Prof. Dr, Amlius Thalib. selaku Dekan FMIPA Institut sains Dan Teknologi Nasional.
2. Bapak Drs. Syarif Hidayatullah, M.Kom. selaku Ketua Program Studi Teknik Informatika Fakultas MIPA Institut Sains dan teknologi Nasional.
3. Orang tua saya tercinta (Prayitno dan Hariyani).
4. Wali Saya Almh Ir. Hj. Bikaningsih yang selalu memberikan dukungan baik secara moril maupun material sehingga dapat melaksanakan dan menyusun Laporan Skripsi.
5. Keluarga besar Ir. Aulia Akbar Harahap, dan juga selaku pimpinan PT. Dyraftama Indotrimitra yang telah mendukung hingga dapat terselesaikannya Laporan Skripsi
6. Teman-teman Teknik Informatika 2010 dan 2011.
7. Semua pihak yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari masih banyak kekurangan , maka mengharapkan kritik dan saran dari pembaca untuk penyempurnaan karya-karya yang akan datang. Harapan dari penulis, semoga Skripsi ini dapat bermanfaat bagi siapa saja yang menggunakannya.

Jakarta , 28 Agustus 2015

Himawan Dewabrata

BAB V PENUTUP

5.1. Simpulan.....	44
5.2. Saran.....	44

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Tabel 2.1. Tabel Karakter Base64	7
Tabel 2.2. Tabel Kode ASCII.....	8
Tabel 2.3. Simbol-simbol dasar flowchart	11
Tabel 4.1. Spesifikasi Leptop	29
Tabel 4.2. Pengujian Black-box.....	36
Tabel 4.3. Perbandingan Kecepatan Loading File.....	43

BAB I

PENDAHULUAN

1.1. Latar Belakang

Seiring perkembangan teknologi dan informasi, manipulasi terhadap gambar, teks, atau berkas-berkas dokumen, sangat mudah dilakukan. Sehingga dapat memberikan celah untuk melaksanakan praktik pemalsuan dokumen. Pemalsuan dokumen umumnya dilakukan dengan cara memanipulasi isi dari dokumen setelah melalui proses *scan* atau membuat dokumen baru dengan desain dan tampilan yang serupa dengan aslinya.

Dalam kehidupan sehari-hari seorang programmer atau penulis *blog*, pasti menginginkan keamanan dan kecepatan dalam menampilkan dan mengelolah halaman *web* atau *blog* yang dimiliki. seperti dalam menampilkan gambar yang tidak tersendat-sendat, halaman *web* atau *blog* yang ringan dll. Hal tersebut dapat dilakukan dengan melakukan kompresi pada file dan data atau menggunakan algoritma yang dapat mengubah kebentuk *string*.

Dari jurnal Annisa Rarasayu, yang berjudul “ Aplikasi Kriptografi Menggunakan Algoritma *Caesar Cipher*” implementasi algoritma klasik seperti *Caesar Cipher* menggunakan metode substitusi transformasi dalam pengenkripsian dan pendekripsian sebuah data yang berupa teks dan angka dengan nilai karakter pada kode ASCII.

Saat ini, banyak algoritma-algoritma kriptografi bermunculan sebagai teknik untuk mengamankan data. Algoritma ini pada dasarnya dibagi menjadi algoritma klasik dan modern. Algoritma klasik beroperasi dalam mode karakter, sedangkan algoritma modern beroperasi dalam mode bit.^[1]

Dan dari jurnal Isnandah Rachman Narima, yang berjudul “Implementasi Metode *Caesar Cipher* Substitusi dan Algoritma Base64 Untuk Penyandian Source Code PHP” algoritma ini cukup mampu untuk mengamankan informasi termasuk *file* format PHP yang didalamnya berisi *Source code*.

Dengan pertimbangan tersebut, penulis ingin menanggapi masalah keamanan data dan sebagai bahan dalam penyusunan skripsi, maka penulis mencoba mengembangkan aplikasi kriptografi. Dengan merancang sebuah aplikasi berbasis *web* yang dapat mengenkrip dan mendekrip *file* teks dan *file* gambar menggunakan algoritma Base64. Serta dapat membantu mempercepat *loading* halaman *web* atau *Blog*. Sebagai salah satu faktor SEO (*Search Engine Optimization*), dengan performa *web* sangat berpengaruh terhadap peringkat hasil pencarian Google. Maka judul yang diambil adalah “Perancangan Aplikasi Kriptografi Algoritma Base64 Berbasis Web”.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah di atas maka dalam penyusunan tugas akhir ini ditemui beberapa permasalahan diantaranya adalah :

1. Bagaimana merancang dan membangun sebuah aplikasi berbasis menggunakan algoritma Base64 ?
2. Bagaimana membuat aplikasi yang dapat mengamankan *file* teks dan *file* gambar?
3. Bagaimana *file* gambar yang sudah dienkripsi kedalam Base64 dapat mempercepat *loading* tampilan sebuah *web* atau *blog*?

1.3. Batasan Masalah

Dalam penyusunan tugas akhir ini dibatasi pada kegiatan dan pembuatan aplikasi. Adapun batasan-batasan masalah yang dibahas adalah :

DAFTAR GAMBAR

Gambar 3.1. Pilihan Bahasa XAMPP	14
Gambar 3.2. Setup Wizard	15
Gambar 3.3. Penyimpanan File Instalasi.....	15
Gambar 3.4. Pilihan Instalasi	16
Gambar 3.5. Proses Instalasi	16
Gambar 3.6. Instalasi dan Integrasi Selesai.....	17
Gambar 3.7. Control Panel	17
Gambar 3.8. Alur Program Enkripsi Base64.....	20
Gambar 3.9. Alur Program Dekripsi Base64	22
Gambar 3.10. Alur Program Enkripsi Base64 File Gambar	24
Gambar 3.11. Alur Program Dekripsi Base64 File Gambar	25
Gambar 3.12. Menu Utama Aplikasi	26
Gambar 3.13. Rancangan Tampilan Enkripsi dan Dekripsi BASE64 Teks.....	27
Gambar 3.14. Rancangan Tampilan Enkripsi File Gambar	27
Gambar 4.1. Halaman Menu Utama Aplikasi	30
Gambar 4.2. Halaman Enkripsi BASE64	31
Gambar 4.3. Halaman Dekripsi BASE64.....	31
Gambar 4.4. Halaman Menu Utama Aplikasi	32
Gambar 4.5. Halaman Enkripsi Base64 File Gambar.....	33
Gambar 4.6. Proses Upload File Gambar.....	33
Gambar 4.7. Hasil Uji Enkripsi Base64	35
Gambar 4.8. Halaman Demo Website	38
Gambar 4.9. Kecepatan Koneksi Internet	38
Gambar 4.10. Memasukkan Skrip Enkripsi Base64 File Gambar.....	39
Gambar 4.11. Tes Kecepatan Pingdom File Gambar (jpg, png, gif)	40
Gambar 4.12 Tes Kecepatan Webpagetest File Gambar (jpg, png, gif)	40
Gambar 4.13. Tes Kecepatan Pingdom Enkripsi Base64	41
Gambar 4.14. Detail Tes Kecepatan Webpagetest	42

1. Merancang dan membangun sebuah aplikasi menggunakan bahasa pemrograman HTML5 dengan algoritma Base64.
2. Aplikasi membahas tentang kriptografi pada teks dan file gambar (jpg, png dan gif) menggunakan algoritma Base64.
3. Aplikasi berjalan pada *localhost*.
4. *Input* yang dimasukan oleh pengguna berupa teks *alphanumeric* kombinasi huruf besar dan huruf kecil maupun angka dan berupa *file* gambar dengan ukuran *file* maksimal sebesar 1.5 MB dengan format gambar (jpg, png dan gif). Tujuan dibatasinya ukuran *file* gambar adalah memperingan kerja *server* pada saat melakukan enkripsi dan dekripsi BASE64.
5. Uji coba dilakukan menggunakan demo *website*. *Pingdom Website Speed Test* dan *Webpagetest* digunakan untuk menganalisa kecepatan *loading* halaman *website*.
6. Bahasan tugas akhir ini hanya mengenai aplikasi kriptografi, sedangkan demo *website*, *Pingdom Website Speed Test* dan *webpagetest* hanya sebagai penunjang tidak dibahas lebih dalam.
7. Batasan fungsi pada aplikasi hanya sebatas menghasilkan teks acak yang tidak dapat dimengerti atau dibaca oleh seseorang.

1.4. Tujuan Penelitian

Tujuan dari pembuatan aplikasi *kriptografi algoritma Base64* ini adalah:

1. Menerapkan algoritma Base64 bukan hanya untuk enkripsi dan dekripsi US-ASCII binar yang digunakan *Multipurpose Internet Mail Extensions* (MIME) sebagai format *email standard internet* tetapi dapat diaplikasikan kedalam bahasa pemrograman *web*.

2. Membuat aplikasi yang dapat digunakan untuk mengamankan *file* teks dan *file* gambar yang akan disimpan kedalam basis data atau dilampirkan pada *script* pemrograman.
3. Membuat aplikasi yang mudah dalam pengoperasian dan penggunaannya.

1.5. Manfaat Penelitian

Manfaat yang didapat diperoleh dari tugas akhir berikut adalah:

1. Menambah pengetahuan tentang algoritma Base64 dalam proses *encode* dan *decode file* teks dan *file* gambar.
2. Membantu pengembang *web* dalam mengamankan *file* yang akan ditampilkan dan disimpan kedalam basis data.
3. Mempermudah bagi pengguna dalam membuat *encode* dan *decode file* teks dan *file* gambar menggunakan algoritma Base64.
4. Dapat dijadikan media pembuat *file* yang rahasia, dan hanya orang yang dikehendaki yang bisa mengaksesnya.

BAB II

TINJAUAN PUSTAKA

2.1. Kriptografi

Kata kriptografi berasal dari bahasa Yunani, “*kryptós*” yang berarti tersembunyi dan “*gráphein*” yang berarti tulisan, kriptografi adalah pelatihan dan mempelajari tulisan rahasia (atau informasi yang tersembunyi).^[2] Kriptografi mengacu hampir secara eksklusif pada enkripsi, yang merupakan proses mengkonversikan informasi biasa menjadi teks yang tak dapat dipahami (disebut *Cipher Text*). Deskripsi merupakan kebalikan, dengan kata lain, memindahkan teks sandi yang tidak dapat dibaca menjadi teks yang dapat dibaca. sandi atau (*Cipher*) merupakan sepasang algoritma yang menciptakan enkripsi dan membalikan dekripsi.^[3] Kriptografi memiliki sejarah panjang dan menarik. *Account non-teknis* yang paling lengkap dari subjek Kahn *The codebreakers*. Buku ini melacak kriptografi dari penggunaan awal dan terbatas oleh bangsa Mesir sekitar 4000 tahun yang lalu, untuk abad kedua puluh di mana ia memainkan peran penting dalam hasil perang dunia kedua. Selesai pada tahun 1963, buku Kahn meliputi aspek-aspek sejarah yang paling signifikan (sampai saat itu) untuk pengembangan subjek. Para praktisi dominan seni itu yang terkait dengan militer, layanan diplomatik dan pemerintah pada umumnya. Kriptografi digunakan sebagai alat untuk melindungi rahasia dan strategi nasional.^[4]

2.1.1. Tujuan Kriptografi

Aspek-aspek keamana didalam kriptografi merupakan aspek – aspek keamanan informasi untuk menjamin sebuah informasi terlindungi dengan aman dan tidak ada tangan yang menyentuh informasi tersebut. Diantaranya adalah:

1. Kerahasiaan (*Confidentiality*)

Seperti hal – hal yang bersifat rahasia pada umumnya, kita harus bisa menjamin sebuah informasi tersebut rahasia, jangan sampai informasi tersebut jatuh kepada pihak – pihak yang tidak berkepentingan. Dalam kriptografi, teknik enkripsi akan membantu memenuhi aspek kriptografis ini.

2. Integritas

Faktor apakah informasi tersebut utuh sesuai dengan yang dikirimkan sebelumnya oleh pengguna kepada pengguna lainnya juga turut memenuhi aspek kriptografis, karena jika terjadi perubahan misalnya ada bagian data yang hilang atau diganti, maka keamanan data ini diragukan karena pihak yang berkepentingan sudah dapat mengakses informasi dan melakukan modifikasi terhadap data tersebut sehingga data yang sampai tidak sesuai dengan data yang sebelumnya dikirimkan.

3. Otentikasi

ini merupakan sebuah teknik untuk mengecek keaslian dari data dan pengguna yang melakukan pengiriman data.

4. Penyangkalan (*non-repudiation*)

adalah layanan yang mencegah sebagian atau salah satu pihak menyangkal komitmen atau tindakan yang dilakukan sebelumnya.

2.2. Algoritma BASE64

BASE64 *encoding* dirancang untuk mewakili urutan acak bilangan oktet dalam bentuk penggunaan huruf kecil dan huruf besar tetapi tidak dapat dibaca oleh manusia.^[5] Ada 64 karakter dari US-ASCII yang digunakan, 6 bit per karakter yang memungkinkan untuk ditampilkan yang dapat dicetak. (tambahan karakter ke 64, “=” digunakan untuk menandakan fungsi pengolahan khusus.).

ditampilkan pada layar monitor, kertas atau media tampilan lainnya. Secara teknis gambar bitmap digambarkan dengan lebar dan tinggi dalam piksel dan dalam angka bit per piksel. Beberapa format gambar bitmap yang sering dijumpai: *GIF*, *JPEG*, *BMP* dan *PNG*.

Keunggulan dari gambar bitmap antara lain:

- Pemakaian *Processor* lebih kecil
- Mampu menangkap nuansa warna dan bentuk yang natural

Sedangkan kelemahannya adalah:

- Ukuran file lebih besar

Ukuran sebenarnya untuk n-bit (2n warna) bitmap dalam *byte* dapat dihitung:

$$\text{Ukuran file bitmap,} \quad 54 + 4 \cdot 2^n + \frac{\text{lebar.tinggi.n}}{8} \quad (2.1)$$

dimana tinggi dan lebar dalam *pixel*.

2.5. Request For Comments (RFC)

RFC adalah publikasi *Internet Engineering Task Force* (IETF) dan pengguna internet, pengembangan teknis dan penetapan standar tubuh utama untuk Internet. RFC ditulis oleh insinyur dan ilmuwan komputer dalam bentuk nota menjelaskan metode, perilaku, penelitian, atau inovasi berlaku untuk kerja internet dan sistem yang tersambung ke internet. Hal ini disampaikan baik untuk penilaian ulasan atau hanya untuk menyampaikan konsep-konsep baru, informasi, atau rekayasa. IETF mengadopsi beberapa proposal diterbitkan sebagai RFC sebagai standar Internet.^[9]

2.5.1 Sejarah RFC

Para penulis dari jenis RFC pertama menulis karya mereka dan beredar salinan tulisan di antara para peneliti ARPA. Berbeda dengan RFC modern, banyak dari RFC awal adalah permintaan yang sebenarnya untuk komentar dan untuk mendorong diskusi. Di mulai dengan RFC pada tahun 1969 pada saat Internet belum ada. Waktu itu namanya ARPANET, RFC 1 berjudul "*Host Software*" ditulis oleh

Steve Crocker dari *University of California, Los Angeles* (UCLA), dan diterbitkan pada tanggal 7 April 1969. Meskipun ditulis oleh Steve Crocker, RFC muncul dari diskusi kelompok kerja awal antara Steve Crocker, Steve Carr dan Jeff Rulifson.

2.6. PHP 5

Perilisan PHP 4.0 pada pertengahan tahun 1999 oleh *Zend*. PHP 4.0 merupakan interpreter PHP terbaru dan merupakan versi PHP yang paling banyak dipakai pada awal abad ke-21 ini. PHP 4.0 banyak dipakai karena memiliki kemampuan membangun aplikasi web kompleks, tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Akhirnya pada juni 2004, *Zend* merilis PHP 5.0 perubahan besar pada inti interpreter PHP. Hal ini karena PHP 5.0 juga memasukan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman kedalam paradigma berorientasi objek.^[10]

2.7. HTML 5

HTML 5 dirilis pada tanggal 28 oktober 2014, HTML 5 memiliki kapabilitas baru dibanding versi sebelumnya HTML 4. Misalnya saja, HTML 5 memiliki tag baru bernama `<audio>` dan `<video>` yang secara khusus berfungsi untuk memasang suara/musik dan video ke dalam website. Ketika menggunakan HTML 4, cara memasang video dan audio kedalam website adalah dengan memanfaatkan tag `<embed>` yang sebenarnya berfungsi sangat luas.^[11]

2.8. JavaScript

JavaScript merupakan bahasa pemrograman berbasis *client*, artinya Javascript merupakan bahasa *scripting* yang digunakan untuk membuat aplikasi *web*, sifatnya *client-side* sehingga dapat diolah langsung di *browser* tanpa harus terhubung ke *server* terlebih dahulu. *JavaScript* sebagai bahasa pemrograman yang berjalan khusus untuk halaman *web* agar halaman *web* menjadi lebih hidup. Walaupun namanya menggunakan kata “*Java*”,

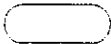
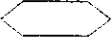
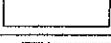

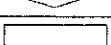
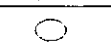


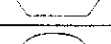
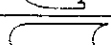
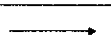

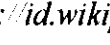
Javascript tidak berhubungan dengan bahasa pemrograman *java*, meskipun keduanya memiliki kemiripan dalam hal *syntax* yang meniru bahasa pemrograman C++ dan C.^[12]

2.9. Flowchart

Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya, gambaran ini dinyatakan dengan simbol, dengan demikian setiap simbol menggambarkan proses tertentu, sedangkan hubungan antar proses digambarkan dengan garis penghubung. *Flowchart* ini merupakan langkah awal pembuatan program. Dengan adanya *flowchart* urutan poses kegiatan menjadi lebih jelas. Setelah *flowchart* selesai disusun, selanjutnya pemrogram (*programmer*) menerjemahkannya ke bentuk program dengan bahasa pemrograman.

Berikut adalah beberapa simbol yang digunakan dalam menggambar suatu flowchart :

Tabel 2.3. Simbol – simbol dasar *flowchart*

Simbol	Nama	Fungsi
	Terminator	Permulaan/akhir program
	Preparation	Persiapan/pemberi harga awal
	Proses	Penugasan/perhitungan/proses
	Input/Output	Proses input/output data
	Decision	Penyeleksian data
	Document	Mencetak hasil
	On page connector	Penghubung dalam satu halaman
	Off page connector	Penghubung beda halaman
	Display	Tampilan (dilayar/monitor)
	Manual Operation	Pekerjaan/operasi secara manual
	Magnetic tape	Input/output berasal dari pita magnetik
	Storage Data	Input/output berasal dari disk
	Garis alir	Menunjukkan arah aliran program

(Sumber: https://id.wikipedia.org/wiki/Diagram_alir, diakses 2015-06-14)

Tabel 2.1. Tabel Karakter Base64

Value	Encoding
0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H
8	I
9	J
10	K
11	L
12	M
13	N
14	O
15	P
16	Q
17	R
18	S
19	T
20	U
21	V
22	W
23	X
24	Y
25	Z
26	a
27	b
28	c
29	d
30	e
31	f
32	g
33	h
34	i
35	j
36	k
37	l
38	m
39	n
40	o
41	p
42	q
43	r
44	s
45	t
46	u
47	v
48	w
49	x
50	y
51	z
52	0
53	1
54	2
55	3
56	4
57	5
58	6
59	7
60	8
61	9
62	+
63	/
64	(pad) =

(Sumber: *The Base 16, Base 32, and Base 64 Data Encodings*. IETF. October 2006. RFC 4648; diakses 2015-06-14)

2.3. Kode ASCII

ASCII (*American Standard Code for Information Interchange*) merupakan suatu standar internasional dalam kode huruf dan simbol seperti Hexadesimal 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 signifikan paling tinggi.^[6] Bit tambahan ini sering digunakan untuk uji prioritas. Karakter

kontrol pada ASCII dibedakan menjadi 5 kelompok sesuai dengan penggunaan yaitu berturut-turut meliputi *logical communication*, *Device control*, *Information separator*, *Code extention*, dan *physical communication*. Jumlah kode ASCII adalah 255 kode. Kode ASCII 0 s/d 127 merupakan kode ASCII untuk manipulasi teks; sedangkan kode ASCII 128 s/d 255 merupakan kode ASCII untuk manipulasi grafik.^[7]

Tabel 2.2. Tabel Kode ASCII

Bit	B7 →				0	0	0	0	1	1	1	1
	B6 →				0	0	1	1	0	0	1	1
	B5 →				0	1	0	1	0	1	0	1
B4	B3	B2	B1	Kolom baris	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(8	H	X	h	x
1	0	0	1	9	HT	EM)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	:	K	[k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	
1	1	0	1	13	CR	GS	-	=	M]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

(Sumber: *ASCII format for network interchange*. IETF. October 16, 1969. RFC 20; Diakses 2015-06-14)

2.4 Bitmap

Bitmap adalah representasi dari citra grafis yang terdiri dari susunan titik yang tersimpan di memori komputer. Dikembangkan oleh *Microsoft* dan nilai setiap titik diawali oleh satu bit data untuk gambar hitam putih, atau lebih bagi gambar berwarna.^[8] Dalam grafik komputer, gambar bitmap adalah sebuah struktur data yang mewakili susunan piksel warna yang

2.10. *Pingdom*

Pingdom adalah solusi pemantauan kinerja global untuk *website* dan aplikasi *web*. Hasil pengujian dan skornya akan bisa langsung lihat. Performa *website* dengan nilai maksimal 100, total permintaan (*request*) ke *server* untuk membuka halaman tersebut, berapa lama waktu yang dibutuhkan untuk memuat *website* secara penuh (*load time*), dan seberapa besar ukuran halaman *website* tersebut (*page size*). Google sekarang juga memperhitungkan kecepatan *loading* sebuah situs sebagai salah satu faktor SEO (*Search Engine Optimization*). Singkatnya performa *website* sangat berpengaruh terhadap peringkat di hasil pencarian Google.

BAB III METODOLOGI PENELITIAN

3.1. Metodologi Penelitian

3.1.1. Objek Penelitian

Objek dari penelitian ini adalah menerapkan algoritma Base64 pada aplikasi kriptografi menggunakan bahasa pemrograman HTML5, adapun lokasi tempat dan waktu penelitian yang telah dilakukan bertempat di PT. Dyraftama Indotrimitra Jl. RSPP No. 15/15C Cilandak Jakarta Selatan, dimulai dari bulan April 2015 sampai dengan Agustus 2015.

3.1.2. Proses Pengumpulan Data

Pada penelitian ini, penulis berusaha mengumpulkan data dan informasi yang akurat untuk menjang proses penelitian. Metode pengumpulan data yang penulis lakukan yaitu:

1. Eksplorasi dan studi literatur

Pengumpulan data dengan cara mengumpulkan literature, jurnal, buku-buku yang berkaitan dengan topik penelitian, dan *browsing internet*.

2. Observasi

Teknik pengumpulan data dengan mengadakan penelitian dan memahami objek permasalahan yang diambil.

3.1.3. Teknik Pembuatan Aplikasi Web

Teknik yang digunakan dalam pembuatan aplikasi kriptografi algoritma Base64 berbasis web ini yaitu *Extreme Programming*. *Extreme Programming* adalah Salah satu metodologi dalam

pengembangan dan rekayasa perangkat lunak. Secara umum *Extreme Programming* (XP) dapat dijabarkan sebagai sebuah pendekatan pengembangan perangkat lunak yang mencoba meningkatkan efisiensi dan fleksibilitas dari sebuah proyek pengembangan perangkat lunak dengan mengkombinasikan berbagai ide simpel atau sederhana tanpa mengurangi kualitas perangkat lunak yang akan dibangun.^[13]

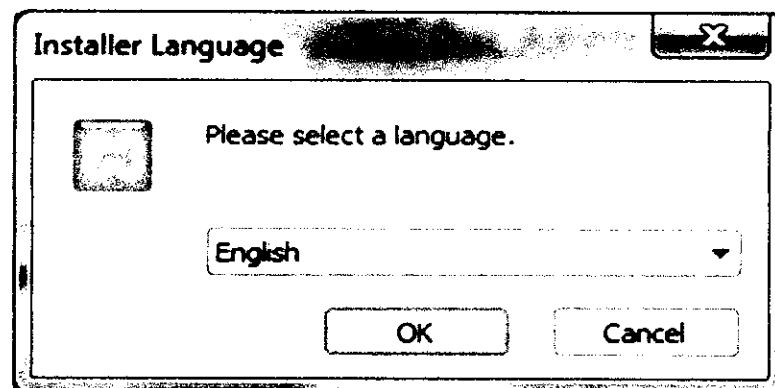
3.2. Perangkat Keras dan Perangkat Lunak yang Digunakan

Alat yang digunakan dalam penelitian ini adalah:

1. Laptop Lenovo G450 dengan spesifikasi:
 - Processor intel dual core 2.10 GHz
 - Ram 2 Gb DDR3
 - Harddisk 150 Gb
2. XAMPP 1.7.7 (yang sudah terinstal PHP 5.3.8 dan *MySQL server*)

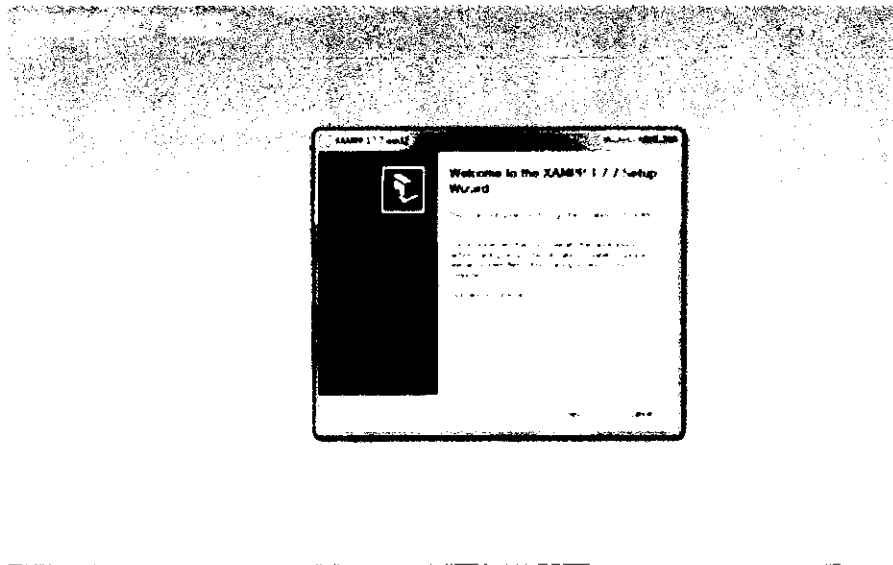
3.3. Instalasi dan Integrasi Perangkat Pendukung

Hal yang pertama kali dilakukan adalah menginstall *local server* XAMPP 1.7.7 win32 <https://www.apachefriends.org/download.html> buka file `xampp-win32-1.7.7-VC9-installer.exe` lalu jalankan. Kemudian akan keluar pilihan bahasa, seperti gambar dibawah ini:



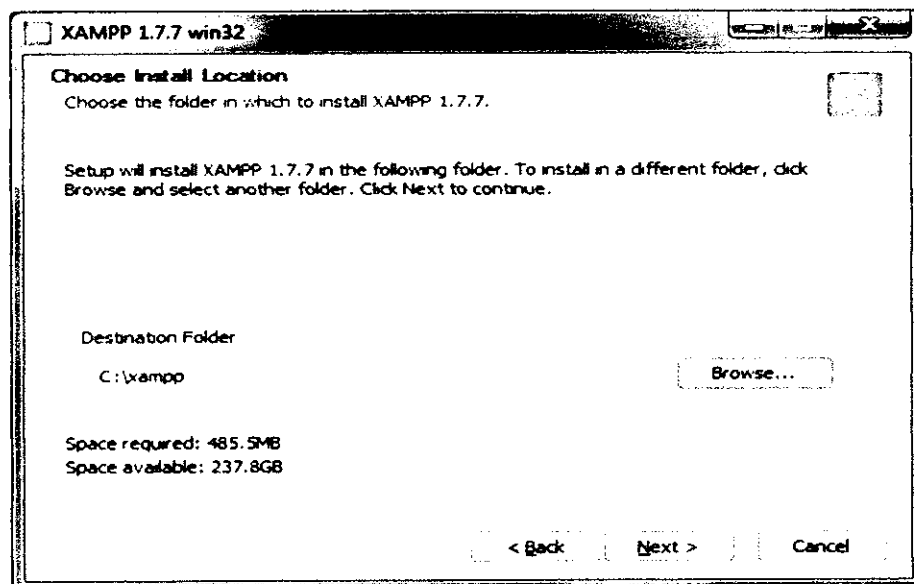
Gambar 3.1. Pilihan Bahasa XAMPP

Selanjutnya akan tampil jendela *setup wizard*, seperti gambar dibawah ini:



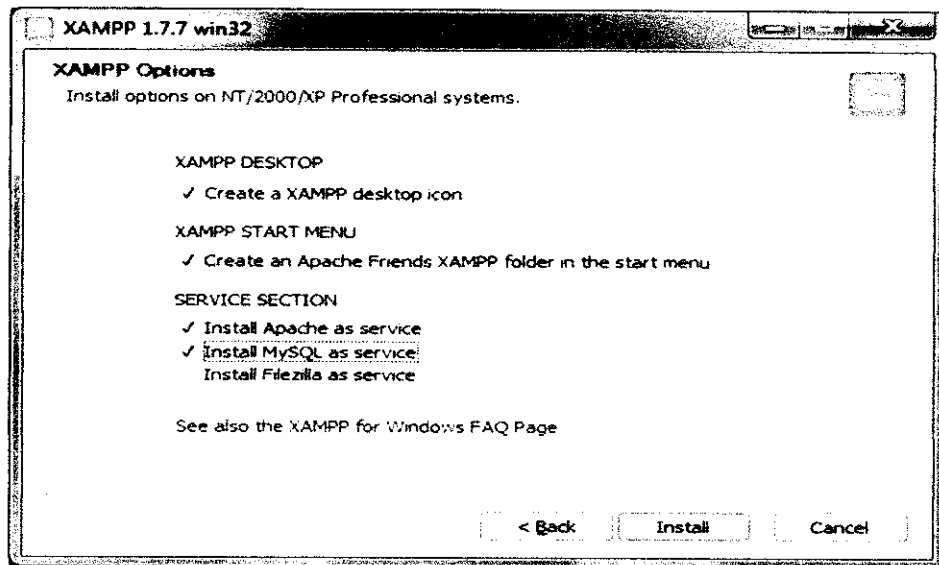
Gambar 3.2. *Setup Wizard*

Pada proses berikutnya akan diminta untuk menentukan dimana *file* instalasi akan di simpan, penulis sarankan penyimpanan *file* instalasi di letakkan diluar dari folder program *file*.



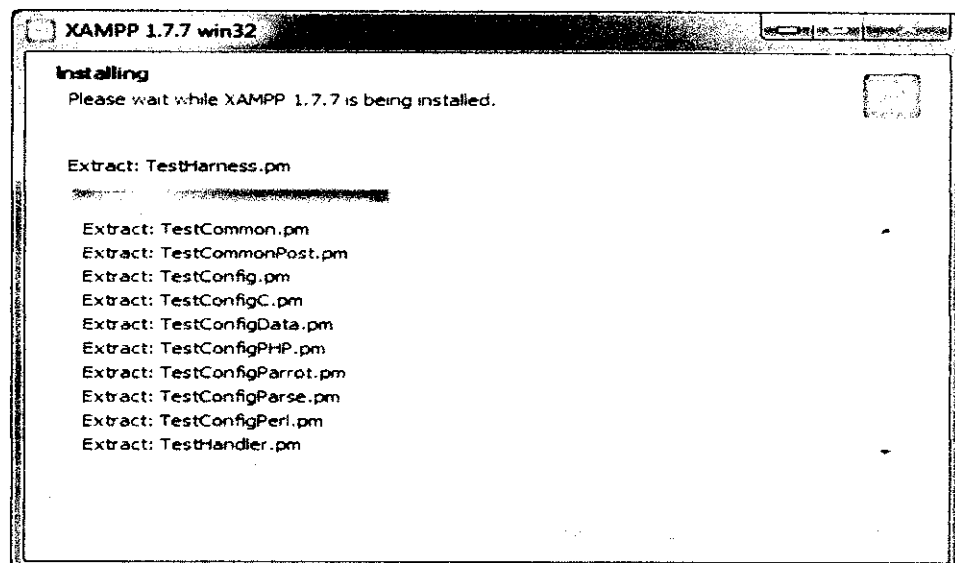
Gambar 3.3. Penyimpanan File Instalasi

Setelah menentukan tempat penyimpanan *file* instalasi, kemudian akan diminta apakah akan dibuatkan *icon* xampp 1.7.7 di dekstop dan start menu, serta pada bagian *Service Section*, beri centang pada Apache dan MySQL. seperti gambar dibawah ini:



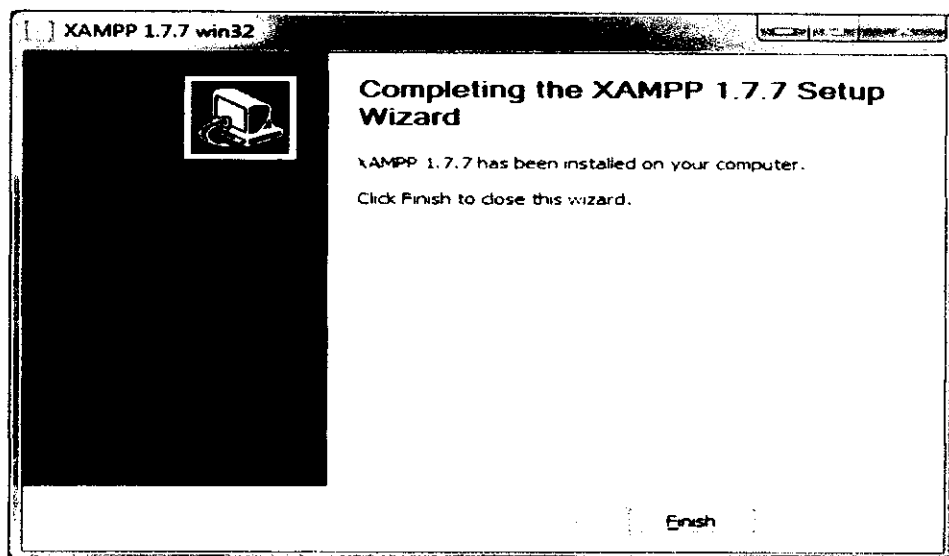
Gambar 3.4. Pilihan Instalasi

Tunggu sampai proses instalasi selesai.



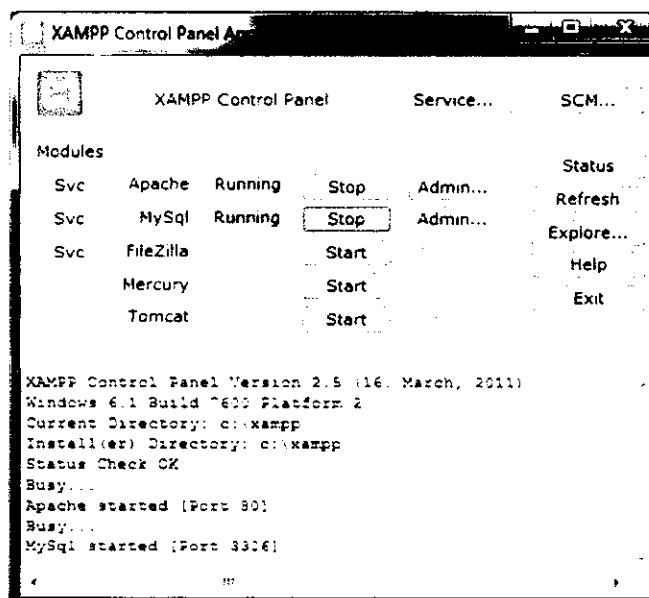
Gambar 3.5. Proses Instalasi

Tampilan instalasi saat proses instalasi selesai klik tombol *finish*.



Gambar 3.6. Instalasi dan Integrasi Selesai

Setelah proses instalasi selesai untuk dapat menjalankan program BASE64 yang penulis kerjakan, *local server Xampp* harus dijalankan terlebih dahulu untuk dapat berintegrasi dengan *browser* yang penulis gunakan. Klik tombol *start* pada pilihan Apache dan MySQL hingga berjalan. Seperti gambar dibawah ini:



Gambar 3.7. Control Panel

3.4. Analisa

3.4.1. Analisa Algoritma

Algoritma yang digunakan dalam perancangan aplikasi ini adalah BASE64 berdasarkan RFC 4648, pada algoritma Base64 yang dijabarkan dalam RFC 4648. Ariyus (2008) menyebutkan jika ada satu (*string*) *bytes* yang akan disandikan ke Base64 maka caranya adalah:

1. Pecah *string byte* menjadi 3 *byte* masing-masingnya
2. Gabungkan 3 *byte* menjadi 24 *bit*. Diketahui bahwa 1 *byte* = 8 *bit*
3. Lalu 24 *bit* yang disimpan dipecah-pecah menjadi 6 *bit*. Maka akan menghasilkan 4 pecahan
4. Masing-masing pecahan diubah kedalam nilai desimal, dimana nilai 6 *bit* adalah 63 karakter Base64
5. Terakhir, jadikan nilai desimal tersebut *index* untuk memilih karakter penyusun dari Base64

Dan seterusnya sampai akhir *string byte* yang akan dienkrripsikan. Jika ternyata dalam proses enkripsi terdapat sisa pembagi, maka tambahkan sebagai penggenap karakter *pad* “=”. Terkadang pada Base64 akan muncul satu atau dua karakter *pad* “==”.

```

+---first octet---+---second octet---+---third octet---+
|7 6 5 4 3 2 1 0|7 6 5 4 3 2 1 0|7 6 5 4 3 2 1 0|
+-----+-----+-----+-----+
|5 4 3 2 1 0|5 4 3 2 1 0|5 4 3 2 1 0|5 4 3 2 1 0|
+----1.index----+----2.index---+----3.index----+----4.index----+

```

Sebagai contoh masukan *string plaintext* karakter ASCII: ISTN dengan hexadesimal 0x4953544E

```
Hex      :49          53          54      | 4E
```

Ubah ke bentuk biner

```
8-bit    :01001001 01010011 01010100    | 01001110
```

Pecah biner ke 4 pecahan 6 bit

6-bit : 010010 010101 001101 010100 | 010011 100000

Dengan pad ==

Desimal : 18 21 13 20 | 19 32

Konversi ke index base64

Base64 : S V N U T g ==

Dapat disimpulkan berdasarkan RFC 2045 setiap 3 karakter ASCII yang akan diencodingkan kedalam Base64 menghasilkan 4 karakter baru artinya 4/3 atau 33% lebih banyak.^[14]

Sedangkan pada proses dekripsi Base64 merupakan kebalikan dari proses enkripsi. Sebagai contoh proses dekripsi pada rangkaian teks: SVNUTg==

Base64 : S V N U T g ==

Desimal : 18 21 13 20 | 19 32 ==

Hilangkan pad “==”

Ubah ke bentuk biner

6-bit : 010010 010101 001101 010100 | 010011 100000

Pecah biner ke 3 pecahan 8 bit

8-bit : 01001001 01010011 01010100 | 01001110

Ubah ke bentuk desimal

Hex : 49 53 54 | 4E

Konversi ke index ASCII

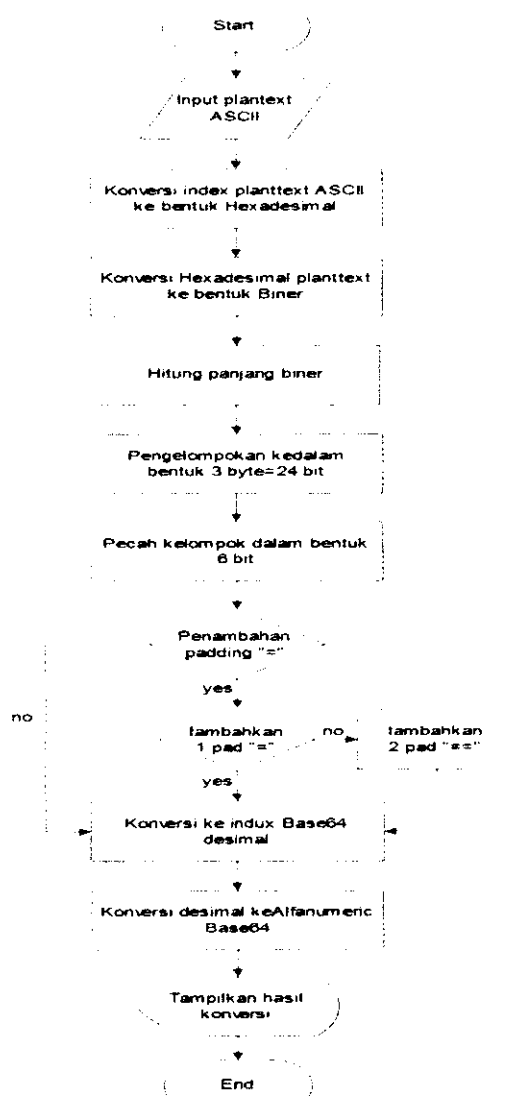
ASCII : I S T | N

3.5. Metode Perancangan Aplikasi

Pada tahap perancangan aplikasi kriptografi dengan menggunakan algoritma Base64, desain perancangan alur program sebagai berikut.

3.5.1. Alur Program Enkripsi Base64 Teks

Dalam pembuatan aplikasi, perancangan pembuatan alur program dimulai dari enkripsi Base64 teks pada gambar dibawah ini:



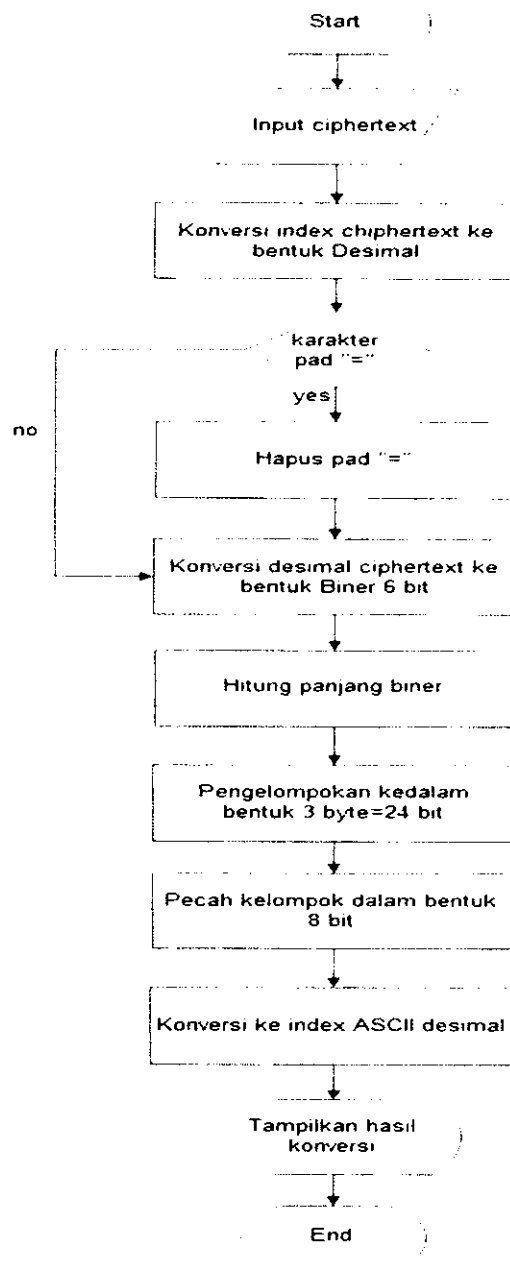
Gambar 3.8. Alur Program Enkripsi Base64

Pada gambar 3.8 alur program enkripsi Base64 teks dapat dijabarkan sebagai berikut:

1. Masukkan *plaintext* dalam bentuk kode ASCII
2. Konversi index *plaintext* ASCII ke bentuk Hexadesimal
3. Konversi hexadesimal ke bentuk biner
4. Hitung panjang biner
5. Kelompokkan ke bentuk 3 byte. 1 byte=8 bit, 3x8=24bit
6. Pecah kelompok ke dalam bentuk 6 bit
7. Penambahan pad “=” untuk *input plaintext* 8 bit dan 16 bit
8. Konversi ke dalam bentuk index BASE64
9. Konversi desimal BASE64 ke bentuk *alfanumeric*
10. *Output* hasil

3.5.2. Alur Program Dekripsi Base64 Teks

Pada proses alur program dekripsi ini merupakan kebalikan dari alur program enkripsi dimana *ciphertext* yang dimiliki oleh pengguna dikembalikan ke bentuk *plaintext*. Maka dapat digambarkan alur program dekripsi Base64 seperti gambar berikut:



Gambar 3.9. Alur Program Dekripsi Base64

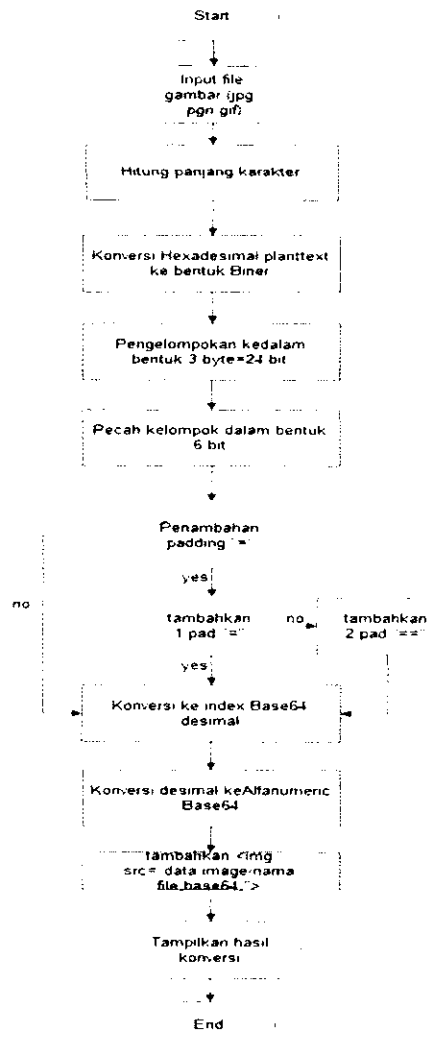
Pada gambar 3.9. alur program dekripsi Base64 teks dapat dijabarkan sebagai berikut:

1. Masukkan ciphertext
2. Konversi index *ciphertext* desimal
3. Hapus penambahan pad "="

4. Konversi desimal ciphertext ke bentuk biner 6 bit
5. Hitung panjang biner
6. Kelompokkan ke bentuk 3 byte. 1 byte=8 bit, 3x8=24bit
7. Pecah kelompok ke dalam bentuk 8 bit
8. Konversi 8 bit biner ke bentuk desimal
9. Konversi ke dalam bentuk index ASCII
10. *Output* hasil

3.5.3. Alur Program Enkripsi Base64 *File* Gambar

Alur program enkripsi Base64 pada file gambar sedikit berbeda dengan alur program pada teks, ini dikarenakan ada penambahan URL pada hasil enkripsi Base64.



Gambar 3.10. Alur Program Enkripsi *file* Gambar

Pada gambar 3.10 alur program enkripsi Base64 *file* gambar dapat dijabarkan sebagai berikut:

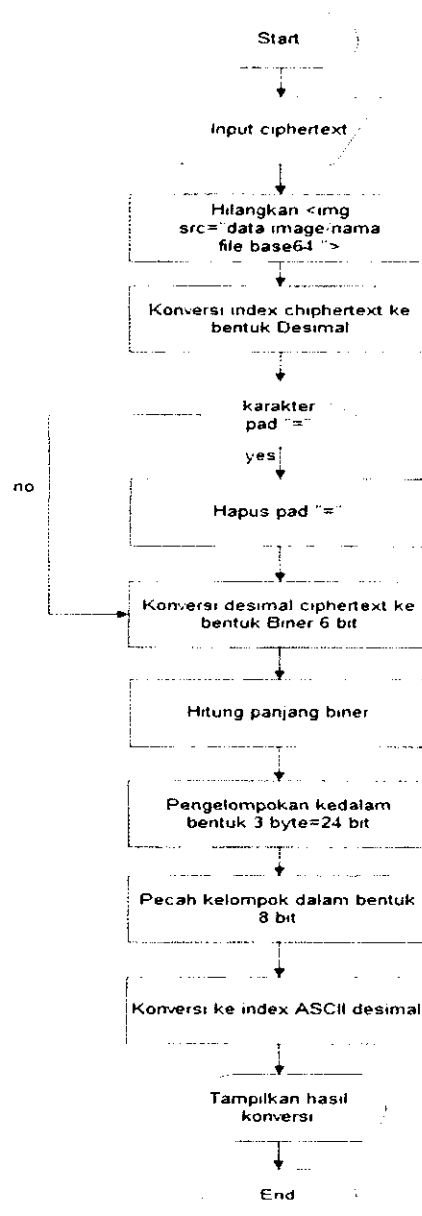
1. Masukan *file gambar* dalam bentuk jpg, png atau gif
2. Hitung panjang karakter
3. Konversi hexadesimal ke bentuk biner
4. Kelompokkan ke bentuk 3 byte. 1 byte=8 bit, 3x8=24bit
5. Pecah kelompok kedalam bentuk 6 bit
6. Penambahan pad “=” untuk *input plaintext* 8 bit dan 16 bit
7. Konversi kedalam bentuk desimal index BASE64
8. Konversi index BASE64 ke bentuk karakter

9. Gabungkan ""

10. Output hasil

3.5.4. Alur Program Dekripsi Base64 File Gambar

Pada proses alur program dekripsi *file* gambar ini merupakan kebalikan dari alur program enkripsi *file* gambar dimana *file* Base64 yang dimiliki oleh pengguna dikembalikan ke bentuk *file* gambar kembali. Maka dapat digambarkan alur program dekripsi Base64 *file* gambar seperti gambar berikut:



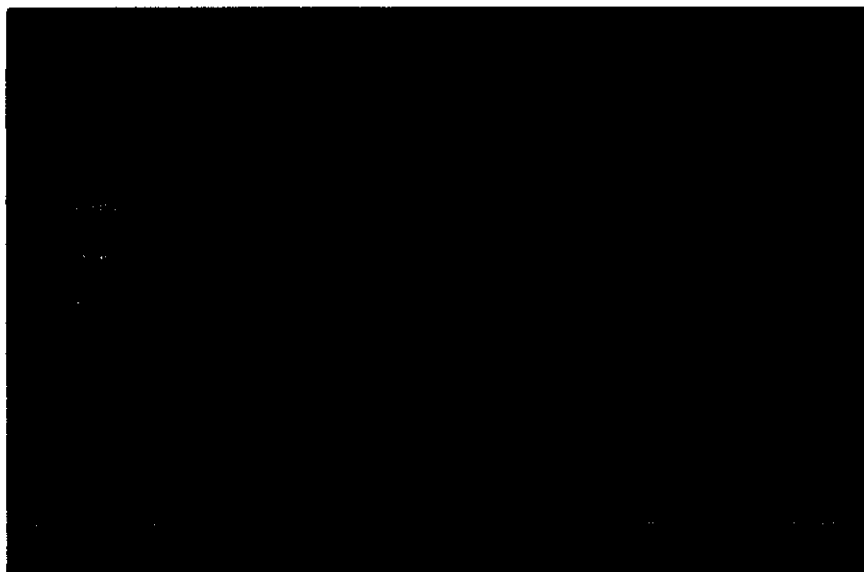
Gambar 3.11. Alur Program Dekripsi *File* Gambar

Proses langkah-langkah pada gambar 3.11. dijelaskan sebagai berikut:

1. Masukkan string Base64 file gambar
2. Hapus ""
3. Konversi index *ciphertext* desimal
4. Hapus penambahan pad "="
5. Konversi desimal ciphertext ke bentuk biner 6 bit
6. Hitung panjang biner
7. Kelompokkan ke bentuk 3 byte. 1 byte=8 bit, 3x8=24bit
8. Pecah kelompok kedalam bentuk 8 bit
9. Konversi kedalam bentuk index ASCII
10. *Output* hasil

3.6. Rancangan Aplikasi

Tujuan dari perancangan antar muka adalah untuk memberikan gambaran aplikasi yang akan dirancang sehingga mempermudah dalam mengimplementasikan ke dalam aplikasi.

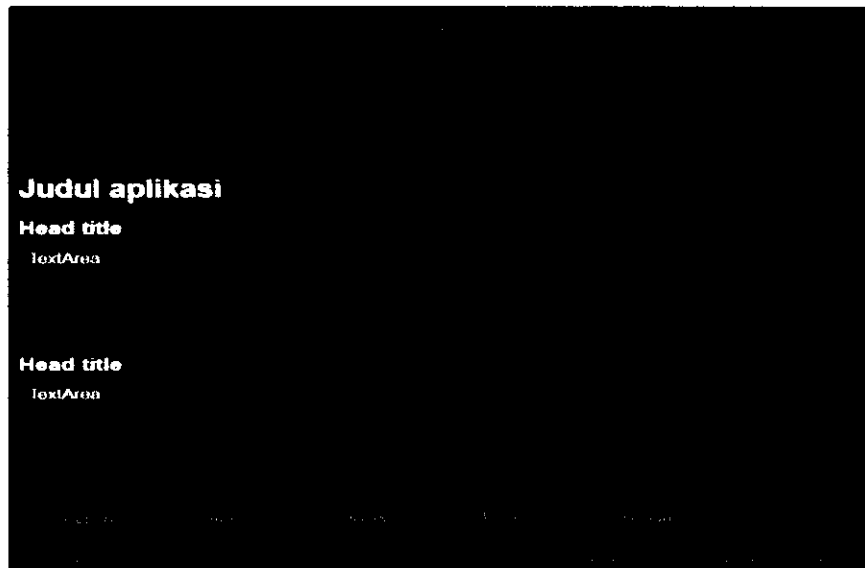


Gambar 3.12. Menu Utama Aplikasi

Gambar 3.12 merupakan rancangan tampilan menu utama dari aplikasi kriptografi algoritma Base64. Dalam form aplikasi terdapat 3 tombol nantinya

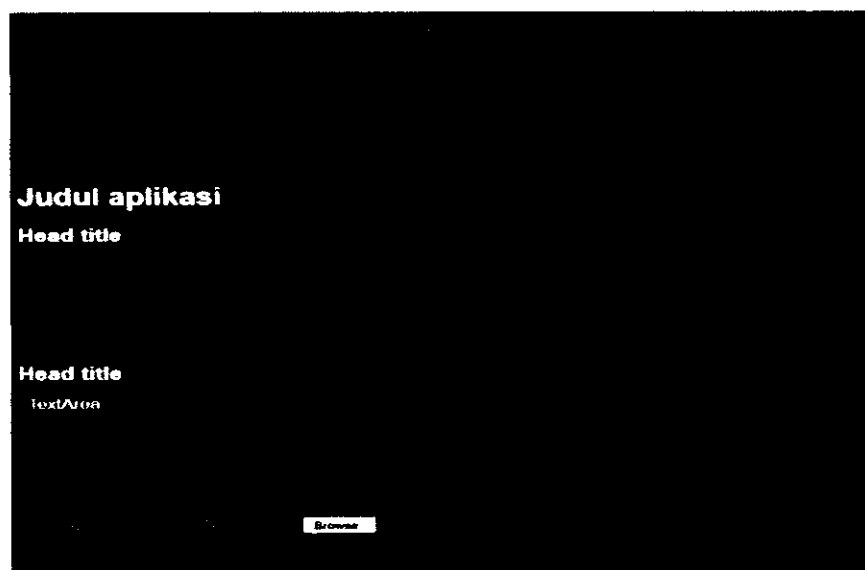
digunakan sebagai tombol bineri, tombol Base64 Gambar dan tombol Base64teks.

Berikutnya rancangan tampilan aplikasi kriptografi algoritma Base64 teks dibawah ini:



Gambar 3.13. Rancangan Tampilan Enkripsi dan Dekripsi BASE64 teks

Tampilan rancangan halaman aplikasi kriptografi algoritma Base64 *file* gambar dibawah ini:



Gambar 3.14. Rancangan Tampilan Enkripsi BASE64 *file* gambar

Pada gambar 3.13 dan 3.14, Setelah menu Base64 di pilih pengguna akan masuk kedalam tampilan aplikasi, pengguna dapat memasukan teks kedalam *textarea* atau memilih *file-file* gambar yang ada di komputer pengguna untuk dilakukanya enkripsi pada *file* gambar tersebut.

BAB IV HASIL DAN PEMBAHASAN

4.1. Peralatan Pengujian Sistem

Pada proses pengujian aplikasi ini membutuhkan alat dan perangkat pendukung agar proses tersebut dapat berjalan sesuai yang diharapkan. Alat dan perangkat pendukung yang digunakan dalam pengujian “Perancangan Aplikasi Kriptografi Algoritma Base64 Berbasis *Web*” ini adalah sebagai berikut:

1. Perangkat Keras (Hardware)

Spesifikasi *Leptop* yang digunakan:

Tabel 4.1. Spesifikasi *Leptop*

Nama	Lenovo
Seri	G450
Processor	Pentium(R) Dual-Core CPU T4300 2.10GHz
Memory	DDR3 2 GB
Tipe Sistem	32 bit
Hard Drive	150 GB Sata

2. Perangkat Lunak (Software)

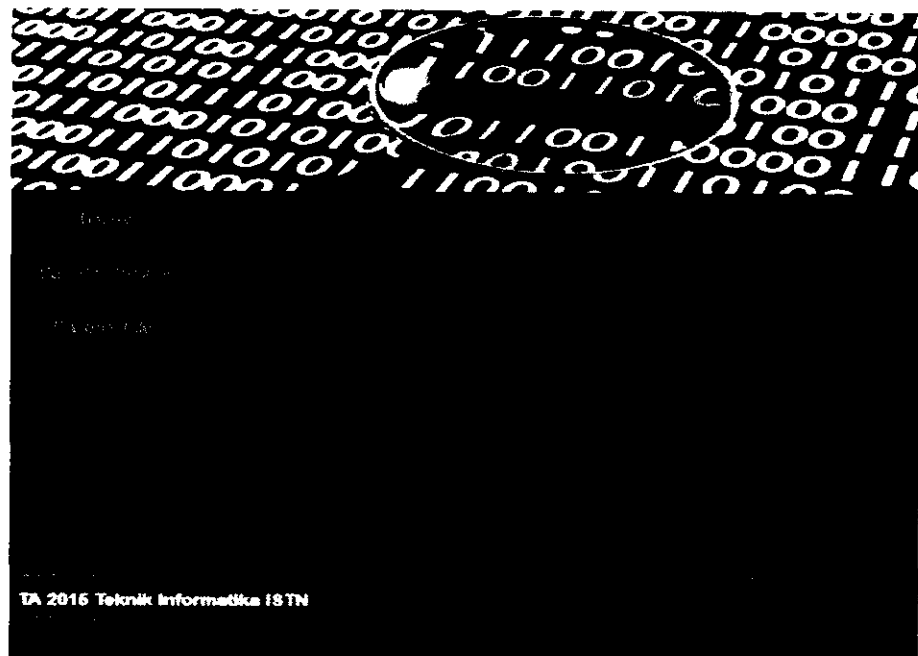
- Sistem Operasi Windows 7 32bit
- *Web Server* XAMPP 1.7.7 yang terdapat didalamnya PHP 5.3.8 dan *Apache* 2.2.21
- *Browser Mozilla Firefox* Versi. 39.0.3

4.2. Hasil

4.2.1. Simulasi Enkripsi dan Dekripsi Base64 Pada Teks

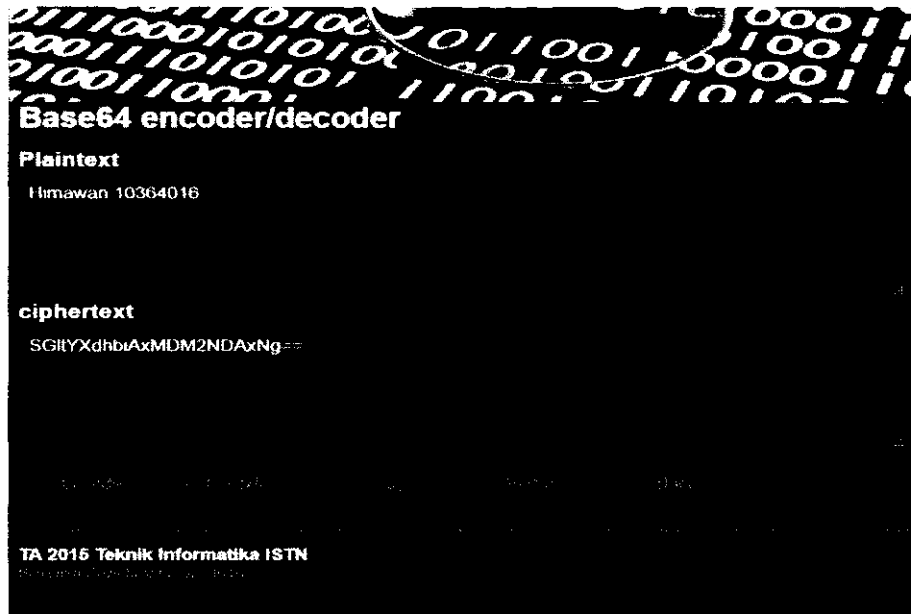
Aplikasi enkripsi dan dekripsi teks dan *file* gambar ini merupakan aplikasi berbasis *web*, dalam penelitian ini penulis menjalankannya melalui lokal *server* menggunakan XAMPP 1.7.7 dikarenakan dapat mempercepat pengujian tanpa harus terhubung ke

internet. Saat pengguna aplikasi enkripsi dan dekripsi *file* teks ini dijalankan pengguna akan langsung dihadapkan pada tampilan menu utama aplikasi enkripsi dan dekripsi BASE64. Berikut tampilan menu utama:



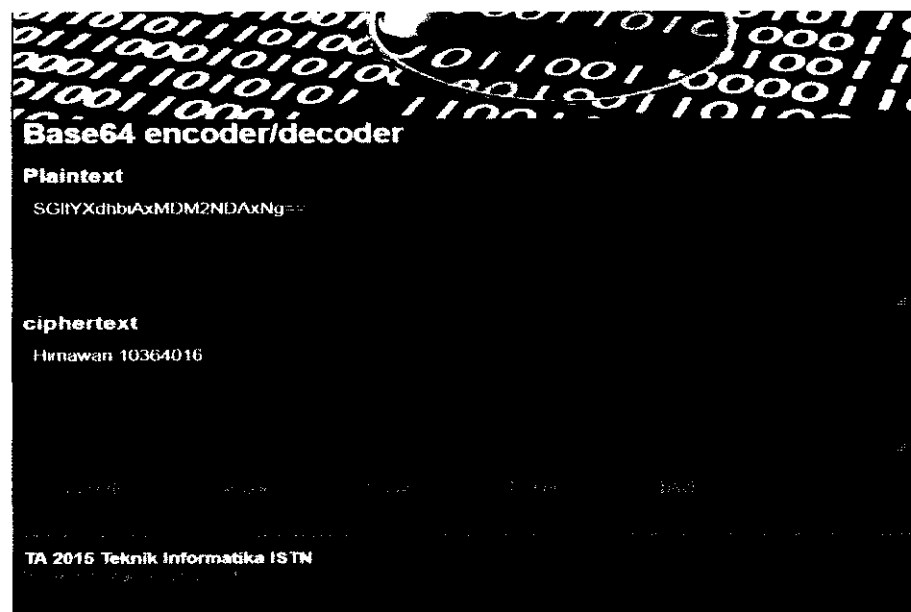
Gambar 4.1. Halaman Menu Utama Aplikasi

Pada Gambar 4.1. pengguna dapat menekan tombol Base64 Teks untuk menggunakan enkripsi atau dekripsi BASE64 pada teks. Cara penggunaannya, *plaintext* yang dimiliki pengguna dapat dimasukkan ke dalam *textarea* untuk dilakukannya proses enkripsi ke bentuk Base 64 dengan menekan tombol *encode* pada aplikasi. Berikut tampilan penggunaan aplikasi enkripsi teks BASE64:



Gambar 4.2. Halaman Enkripsi BASE64

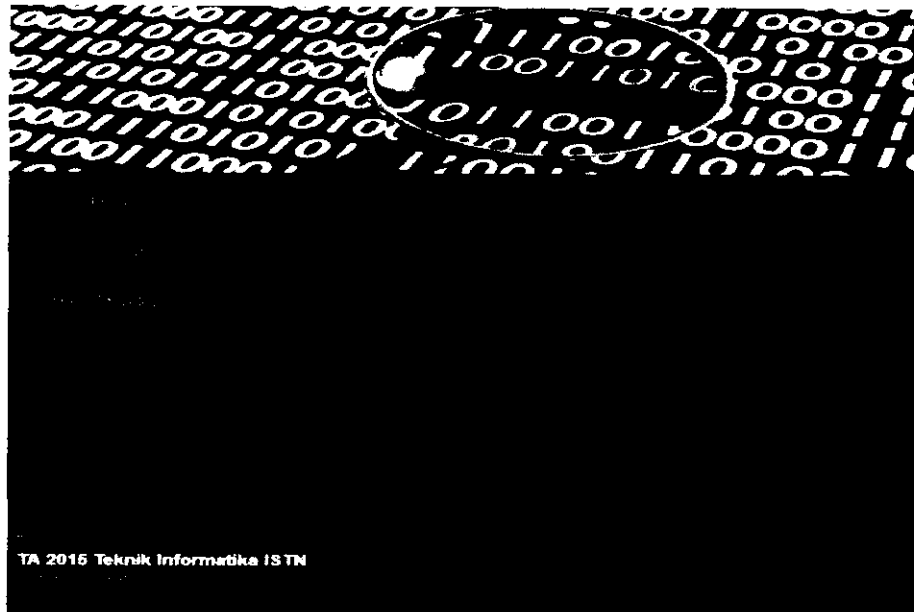
Pengguna aplikasi juga dapat mengembalikan nilai *ciphertext* Base64 menjadi *plaintext* dengan mengisi *textarea plaintext* dengan *ciphertext* Base64 yang dimilikinya dan menekan tombol *Decode*. Berikut tampilan penggunaan aplikasi dekripsi teks BASE64:



Gambar 4.3. Halaman Dekripsi BASE64

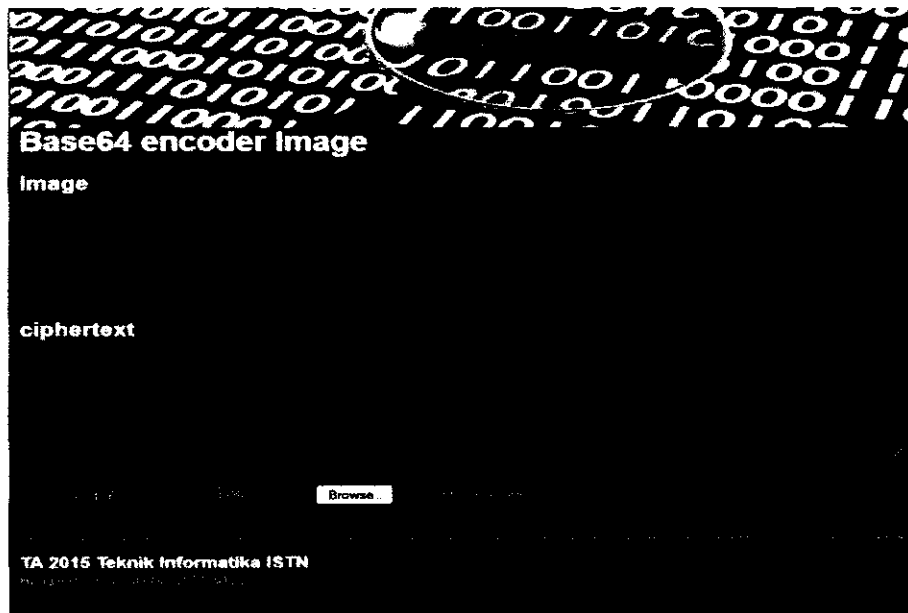
4.2.2. Simulasi Enkripsi Base64 Pada File Gambar

Tidak jauh berbeda cara penggunaan aplikasi enkripsi BASE64 pada file gambar dengan penggunaan pada teks. Pada tampilan halaman menu utama pengguna dapat memilih tombol Base64Gambar. Berikut tampilan halaman menu Base64Gambar:



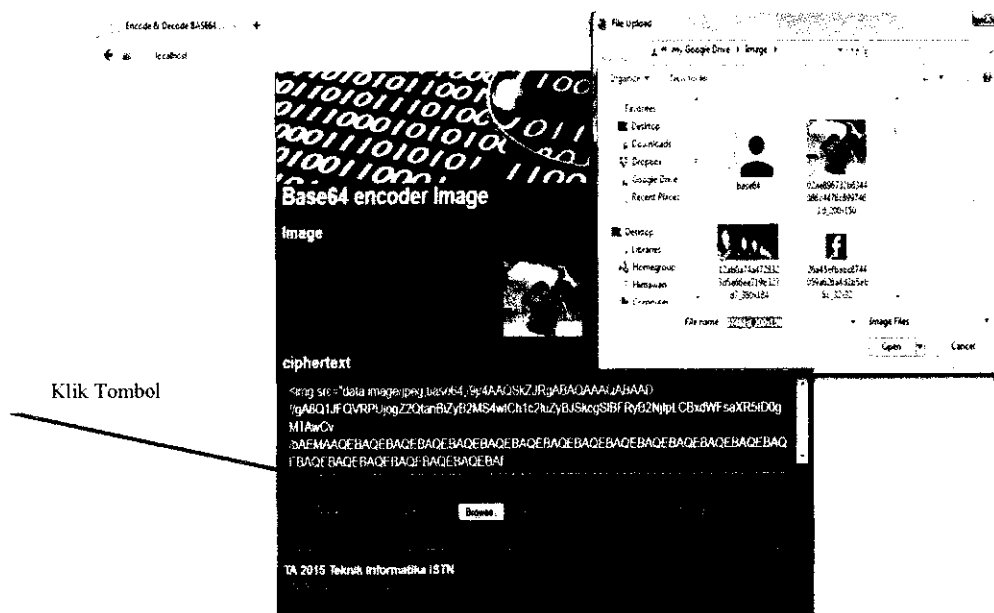
Gambar 4.4. Halaman Menu Utama Aplikasi

Tombol Base64Gambar akan melakukan panggilan satu kali pada file *base64Image.php*. Kemudian pengguna akan masuk ke dalam halaman enkripsi base64 *file* gambar. Berikut tampilan halaman Base64Gambar:



Gambar 4.5. Halaman Enkripsi Base64 *File* Gambar

Setelah pengguna aplikasi masuk kedalam halaman enkripsi base64 *file* gambar, pengguna dapat menekan tombol *Browse* pada aplikasi untuk melakukan *upload file* gambar yang ada pada *folder* komputer pengguna.



Gambar 4.6. Proses *Upload File* Gambar

hasil *file* enkripsi Base64 akan ditampilkan pada kolom *textarea*.

4.3. Pengujian dan Pembahasan

4.3.1. Uji Coba Enkripsi dan Dekripsi BASE64 Teks

Uji yang dilakukan penulis pada enkripsi dan dekripsi Base64 teks dengan menggunakan *plaintext* berupa nama dan nim penulis "Himawan 10364016". Yang kemudian akan dilakukan perhitungan manual, hasilnya akan disamakan dengan aplikasi yang penulis rancang.

Data masukan : Himawan 10364016

Hexadesimal : 0x48696d6177616e203130333634303136

Hex : 48 69 6d | 61 77 61 |

Ubah kebentuk biner

8bit : 01001000 01101001 01101101 | 01100001 01110111 01100001 |

Pecah biner ke 4 pecahan 6 bit

6bit : 010010 000110 100101 101101 | 011000 010111 011101 100001 |

Ubah kebentuk desimal

Desimal : 18 6 37 45 | 24 23 29 33 |

Konversi ke index base64

Keluaran : S G l t | Y X d h |

Lanjutan...

Hex : 6e 20 31 | 30 33 36 |

Ubah kebentuk biner

8bit : 01101110 00100000 00110001 | 00110000 00110011 00110110 |

Pecah biner ke 4 pecahan 6 bit

6bit : 011011 100010 000000 110001 | 001100 000011 001100 110110 |

Ubah kebentuk desimal

Desimal : 27 34 0 49 | 12 3 12 54 |

Konversi ke index base64

Keluaran : b i A x | M D M 2 |

Lanjutan...

Hex : 34 30 31 | 36

Ubah kebentuk biner

8bit : 00110100 00110000 00110001 | 00110110

Pecah biner ke 4 pecahan 6 bit

6bit : 001101 000011 000000 110001 | 001101 100000 000000 000000

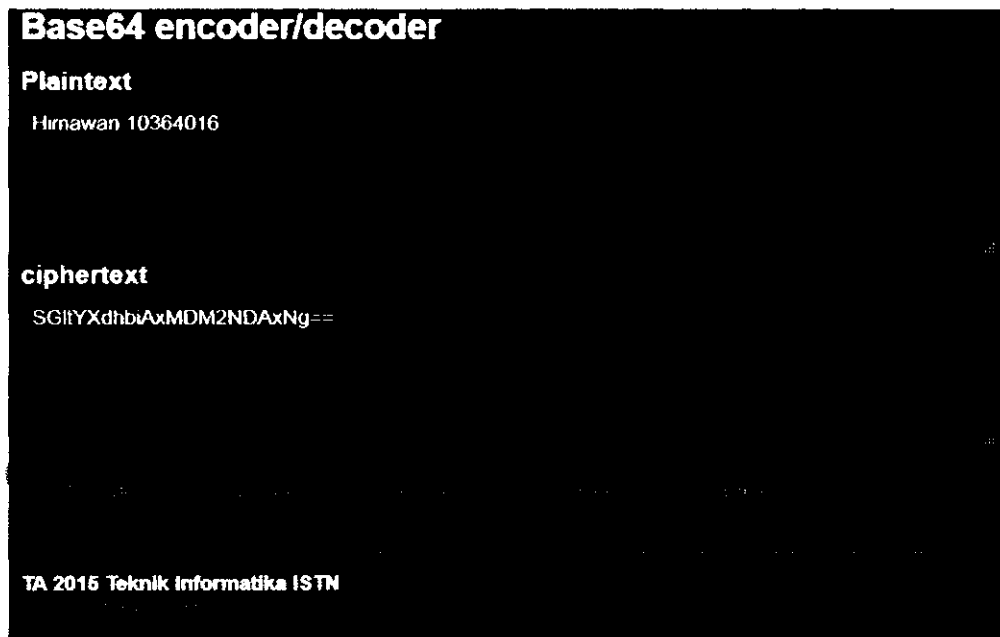
Ubah kebentuk desimal

Desimal : 13 3 0 49 | 13 32 = =

Konversi ke index base64

Keluaran : N D A x | N g = =

Penggabungan hasil keluaran “SGltYXdhbiAxMDM2NDAxNg==” kemudian penulis melakukan *crosscheck* pada aplikasi yang telah penulis rancang, seperti pada gambar berikut:



Gambar 4.7. Hasil Uji Enkripsi Base64

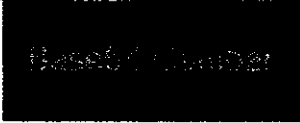
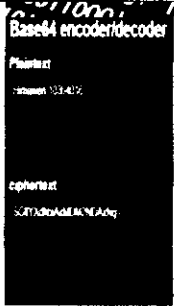

Dari hasil perhitungan secara manual yang telah dilakukan didapatkan hasil yang sama dengan aplikasi kriptografi algoritma Base64. Dari hasil pengujian tersebut penulis dapat menyimpulkan sebagai berikut:





1. Masukan yang bernilai 24 bit akan menghasilkan keluaran 4 karakter tanpa “=” *pad*.

2. Masukan yang bernilai 8 bit akan menghasilkan 2 karakter diikuti 2 “=” *pad*.
3. Masukan yang bernilai 16 bit akan menghasilkan 3 karakter diikuti 1 “=” *pad*.

Kemudian perancangan aplikasi ini, dilakukan dengan pengujian *black-box* dengan melakukan *test-case* yang diisi saat pengujian.

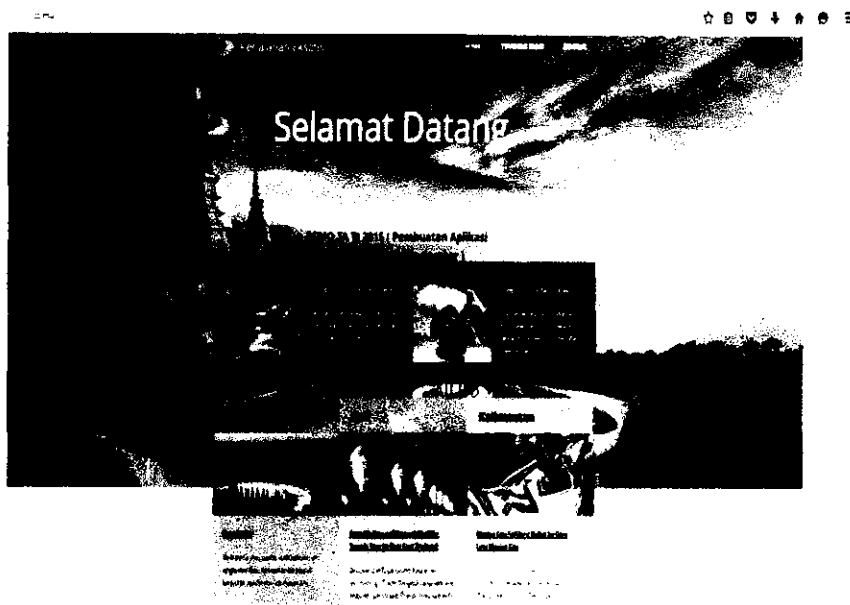
Tabel 4.2. Pengujian *Black-box*

No	Skenario pengujian	Test Case	Hasil Yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Membuka aplikasi <i>web</i>	http://localhost/Algoritma BASE64/	Tampilan menu utama	Gambar 4.1.	Valid
2.	Membuka halaman Enkripsi dan Dekripsi <i>File Teks</i>		Tampilan halaman Enkripsi dan Dekripsi <i>File Teks</i>	Gambar 4.2	Valid
3.	Memasukkan <i>plaintext</i> dan mengubah kedalam bentuk Base64	Base64 encoder/decoder Plaintext Hirawan 10364016 ciphertext	Menghasilkan <i>ciphertext</i> Base64		Valid
4.	Memasukkan <i>ciphertext</i> Base64 dan mengubah kebentuk <i>plaintext</i>	Base64 encoder/decoder Plaintext SGRiYXdhbnAxMDM2NDAxNg... ciphertext	Menghasilkan <i>Plaintext</i>		Valid

5.	Memasukkan <i>plaintext</i> ASCII untuk dienkripsi		error		Invalid
6.	Memasukkan file gambar (jpg, png, gif)		Menghasilk an enkripsi Base64		Valid

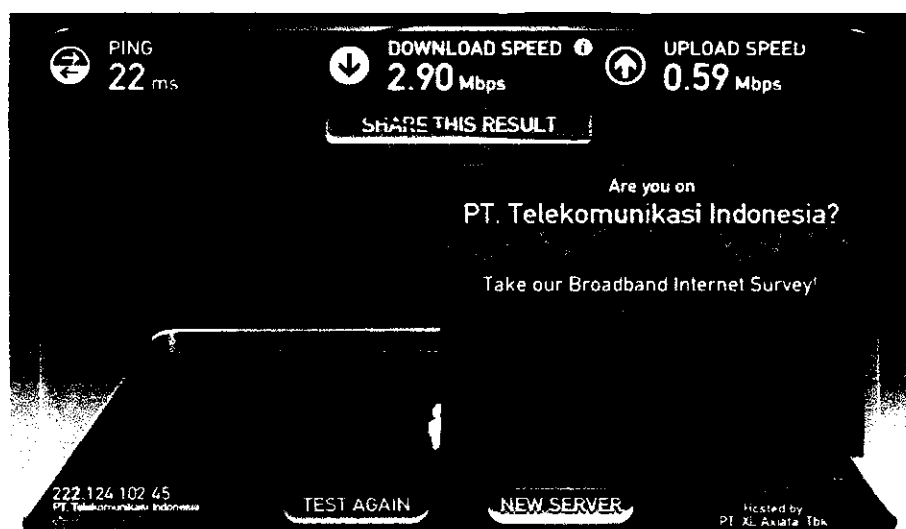
4.3.2. Uji Coba Enkripsi BASE64 File Gambar

File gambar yang telah dienkripsi kedalam Base64 kemudian akan di uji cobakan kedalam demo *website* <http://myta2015.zz.mu/> dan dilakukan penganalisaan dengan menggunakan *pingdom website speed test* dan *webpagetest*” untuk didapatkan hasil kecepatan *loading website* dengan meng-*upload file* gambar yang sudah dienkripsi ke dalam Base64.



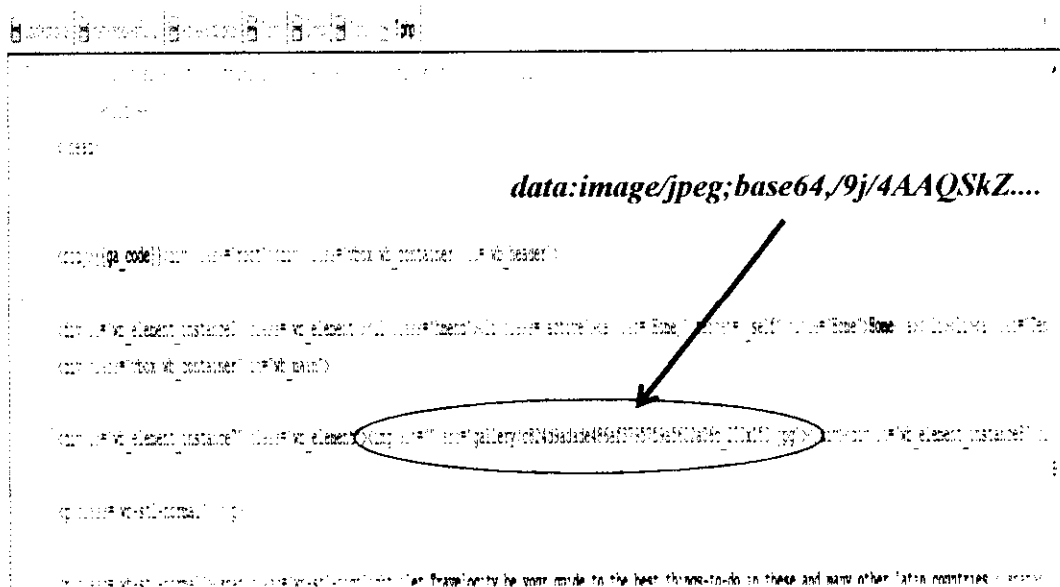
Gambar 4.8. Halaman Demo *website*

Hal pertama yang dilakukan penulis sebelum melakukan tes pada demo *website* adalah mengukur kecepatan dari koneksi internet dalam hal ini menggunakan jaringan telkom *speedy*. Untuk mengetahui kecepatan *download* dan kecepatan *upload* pada saat dilakukan tes pada koneksi internet telkom *speedy* menggunakan *speedtest.net* dan didapatkan kecepatan download 2,90 Mbps dan kecepatan upload 0,59 Mbps. Seperti gambar dibawah ini:



Gambar 4.9. Kecepatan Koneksi *Internet*

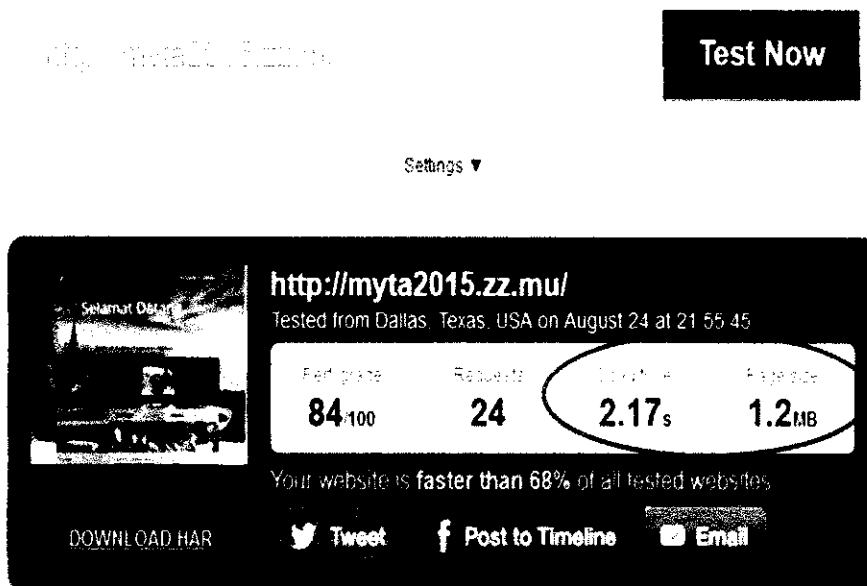
Setelah didapatkan kecepatan koneksi jaringan internet barulah dilakukan tes perbandingan kecepatan sebelum dan sesudah menggunakan enkripsi Base64 pada *file* gambar yang di-*upload* pada *website*. *skrip* yang di gantikan didalam *file* PHP dengan enkripsi Base64 adalah *skrip* untuk memasukkan gambar kedalam *website* “*img src = ” ”*” seperti pada gambar berikut:



The image shows a browser's developer tools window with the 'Elements' tab selected. It displays a snippet of HTML code. A line of code is circled in black, and an arrow points to it from the text 'data:image/jpeg;base64,/9j/4AAQSkZ....' located above the code. The circled code is: ``. The surrounding code includes various JavaScript functions and attributes like 'ga_code', 'vbox', and 'vcontainer'.

Gambar 4.10. Memasukkan *Skrip* Enkripsi Base64 *File* Gambar

Hasil pengujian pengukuran kecepatan loading pada ukuran *file* gambar sebelum di ubah ke dalam bentuk enkripsi Base64 menggunakan *pingdom website speed test* adalah *load time* 2,17 detik ukuran halaman *page size* 1,2 MB.



Gambar 4.11. Tes Kecepatan Pingdom File Gambar (jpg, png, gif)

Tes berikutnya masih pada halaman web yang sama menggunakan *webpagetest* didapatkan data seperti pada gambar dibawah ini:

Resource	Content Type	Request Details				Time to First Byte	Content Download	Bytes Downloaded
		Request Start	DNS Lookup	Initial Connection	SSL Negotiation			
1 http://myta2015.zz.mu/	text/html	0.129 s	145 ms	114 ms	-	157 ms	0 ms	21.1 KB
2 http://myta2015.zz.mu/js/bootstrap.min.js	text/css	0.446 s	-	-	-	113 ms	532 ms	12.1 KB
3 http://myta2015.zz.mu/css/bootstrap.min.css	text/css	0.170 s	-	111 ms	-	117 ms	112 ms	13.4 KB
4 http://myta2015.zz.mu/js/jquery.min.js	application/javascript	0.402 s	-	110 ms	-	142 ms	420 ms	81.7 KB
5 http://myta2015.zz.mu/js/bootstrap.min.js	text/css	0.402 s	-	110 ms	-	111 ms	133 ms	21.3 KB
6 http://myta2015.zz.mu/js/bootstrap.min.js	application/javascript	0.173 s	-	103 ms	-	208 ms	147 ms	29.3 KB
7 http://myta2015.zz.mu/js/bootstrap.min.js	application/javascript	0.173 s	-	104 ms	-	119 ms	114 ms	2.4 KB
8 http://myta2015.zz.mu/js/bootstrap.min.js	text/css	0.020 s	-	-	-	105 ms	110 ms	21.7 KB
9 http://myta2015.zz.mu/js/bootstrap.min.js	text/css	0.024 s	-	-	-	105 ms	110 ms	27.0 KB
10 http://myta2015.zz.mu/js/bootstrap.min.js	image/png	0.174 s	-	-	-	110 ms	110 ms	2.7 KB
11 http://myta2015.zz.mu/js/bootstrap.min.js	image/png	0.174 s	-	-	-	103 ms	103 ms	13.3 KB
12 http://myta2015.zz.mu/js/bootstrap.min.js	image/png	0.174 s	-	-	-	107 ms	107 ms	13.0 KB
13 http://myta2015.zz.mu/js/bootstrap.min.js	image/png	0.174 s	-	-	-	221 ms	210 ms	45.3 KB
14 http://myta2015.zz.mu/js/bootstrap.min.js	image/png	0.174 s	-	-	-	158 ms	148 ms	14.0 KB
15 http://myta2015.zz.mu/js/bootstrap.min.js	image/png	0.174 s	-	-	-	158 ms	148 ms	14.0 KB
16 http://myta2015.zz.mu/js/bootstrap.min.js	image/png	0.174 s	-	-	-	140 ms	-	1.7 KB
17 http://myta2015.zz.mu/js/bootstrap.min.js	image/png	0.174 s	-	-	-	140 ms	-	1.3 KB
18 http://myta2015.zz.mu/js/bootstrap.min.js	image/png	0.469 s	-	-	-	104 ms	140 ms	53.4 KB
19 http://myta2015.zz.mu/js/bootstrap.min.js	text/html	0.442 s	227 ms	45 ms	-	32 ms	-	1.4 KB
20 http://myta2015.zz.mu/js/bootstrap.min.js	image/png	0.05 s	-	-	-	113 ms	99 ms	15.3 KB
21 http://myta2015.zz.mu/js/bootstrap.min.js	image/png	0.057 s	-	-	-	115 ms	117 ms	11.0 KB

Gambar 4.12 Tes Kecepatan Webpagetest File Gambar (jpg, png, gif)

Dari hasil pengukuran kecepatan *loading web* menggunakan *webpagetest* didapatkan waktu rata-rata 1,311 detik dari 12 file gambar.

Tahap selanjutnya adalah tes pengujian *loading web* menggunakan *file* gambar yang telah dienkripsi Base64.

The screenshot shows the Pingdom Website Speed Test interface. At the top, it says "Pingdom Website Speed Test" and "Enter a URL to test the load time of that page, analyze it and find bottlenecks". Below this is a text input field containing "http://myta2015.zz.mu/" and a "Test Now" button. A "Settings" dropdown menu is visible below the input field. The main test results are displayed in a dark-themed box with a white background for the data. The URL being tested is "http://myta2015.zz.mu/" and it was tested from Amsterdam, Netherlands on August 24 at 21:59:37. The results are as follows:

Performance	Progress	Load Time	Response
79/100	25	1.26s	2.6MB

Below the table, it states "Your website is faster than 85% of all tested websites". At the bottom of the results box, there are links for "DOWNLOAD HAR", "Tweet", "Post to Timeline", and "Email".

Gambar 4.13. Tes Kecepatan *Pingdom* enkripsi Base64

Dari data gambar di atas didapatkan data waktu kecepatan *loading* halaman *web* lebih cepat 1,26 detik atau terdapat selisih 0,91 detik dari waktu tes sebelum menggunakan enkripsi Base64. Tetapi didapat data ukuran halaman web 2,6 MB lebih besar dari hasil pengukuran sebelumnya yang bernilai 1,2 MB atau terdapat selisih 1,4 MB. Ini sesuai dengan kesimpulan yang didapat berdasarkan RFC 2045 pada BAB III penulisan.

Resource Type	Resource Name	Size	Status
css	css	21.5 kB	
css	common.css?e=144000000	21.5 kB	
font	font	40.1 kB	
data:image/png;base64	data:image/png;base64,VBQRu0hGooAA	0 B	
data:image/png;base64	data:image/png;base64,VBQRu0hGooAA	0 B	
img	6407509ab72760b30342654a195972e.jpg	635.4 kB	
img	63e70000eac93117e0e9a7175e011ac9	73.1 kB	
img	RuQ3YrTmaVpZmQ6UT8E0z7zEPnyo?	29.1 kB	
data:image/png;base64	data:image/png;base64,9,4AAQSkZURg	0 B	
data:image/png;base64	data:image/png;base64,9,4AAQSkZURg	0 B	
data:image/png;base64	data:image/png;base64,9,4AAQSkZURg	0 B	
data:image/png;base64	data:image/png;base64,9,4AAQSkZURg	0 B	
data:image/png;base64	data:image/png;base64,9,4AAQSkZURg	0 B	
data:image/png;base64	data:image/png;base64,VBQRu0hGooAA	0 B	
data:image/png;base64	data:image/png;base64,VBQRu0hGooAA	0 B	
http://www	http://www.com/examples.getjs	1.4 kB	
data:image/png;base64	data:image/png;base64,9,4AAQSkZURg	0 B	
img	spool0.png	16.6 kB	
http://www	http://www.com/2009/statist/flash	334 B	

Gambar 4.14. Detail Tes Kecepatan *Webpagetest*

Pada tes kedua pada *website* yang sudah menggunakan *file* gambar Base64, *Webpagetest* menampilkan data seperti terlihat pada Gambar 4.14, ukuran *file* gambar dan kecepatan bernilai 0 B dan 0 detik. Ini dikarenakan *file* gambar yang sudah diubah ke bentuk Base64, *file* gambar tidak memiliki alamat *URL*, itu berarti tidak ada permintaan *http request* ke *server*.

Berikut tabel hasil perbandingan kecepatan *loading* halaman *website* dari 10 (sepuluh) kali percobaan menggunakan *pingdom website speed test*.

Tabel 4.3. Perbandingan Kecepatan *Loading File*

No.	Kecepatan Sebelum (detik)	Kecepatan Enkripsi Base64 (detik)
1.	2.17	1.26
2.	1.65	1.63
3.	2.15	1.59
4.	2.13	1.55
5.	1.79	1.42
6.	2.26	1.26
7.	2.15	1.45
8.	1.85	1.63
9.	2.12	1.55
10.	2.11	1.75
Rata-rata	2.038	1.509

Dari data kecepatan waktu loading halaman website yang diperoleh dalam 10 kali percobaan, didapatkan waktu rata-rata sebelum menggunakan *file* gambar yang di enkripsi ke bentuk base64 adalah 2.038 detik dan kecepatan setelah dienkripsi ke dalam bentuk base64 1.509 detik. Artinya terdapat perbedaan kecepatan waktu sebesar 0,529 detik lebih cepat.

Kesimpulan yang didapat, *file* gambar yang dienkripsi ke dalam bentuk Base64 dapat membantu mempercepat *loading* suatu *web* atau *Blog*.

BAB V

PENUTUP

5.1. Simpulan

Berdasarkan rancangan yang telah dibuat dan dilakukan uji coba maka dapat disimpulkan sebagai berikut:

1. Perancangan aplikasi kriptografi algoritma Base64 dapat dengan mudah diterapkan pada bahasa pemrograman berbasis *web* karena didukung banyak *library defaultnya*.
2. Dari aplikasi kriptografi algoritma BASE64 yang dibuat ini, teks dan *file* gambar dapat dengan mudah dikonversi ke dalam bentuk BASE64 dan sudah berbentuk *string* acak yang tidak terbaca oleh seseorang dan hanya orang yang berkepentingan yang dapat mengelola *file* tersebut.
3. Kemudahan operasi dan implementasi penggunaannya sebagai pendukung pengembangan kode *website* yang aman dan ringan. Karena berbentuk kode *string* dan bukan *file*, dari uji coba yang dilakukan dapat dibuktikan *server* dengan cepat bisa menampilkan data 0.529 detik lebih cepat. Berbeda dengan cara memanggil *file* gambar yang kita dapatkan dari sebuah *URL*.

5.2. Saran

Implementasi algoritma BASE64 dapat dikembangkan dan dikombinasikan dengan algoritma-algoritma DES (*Data Encryption Standard*), AES (*Advanced Encryption Standard*) dan algoritma lainnya, tergantung dengan kebutuhan dan seberapa perlu kebutuhan akan kerahasiaan pada teks yang akan diamankan oleh pihak pengguna dan kedepannya dapat dijalankan pada *internet website*.

DAFTAR PUSTAKA

1. Prayudi, Yudi, Idham Halik (2005). *Studi Analisis Algoritma Rivest Code 6 (RC6) Dalam Enkripsi Dekripsi Data*. Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNATI 2005), Yogyakarta.
2. <http://staff.neu.edu.tr/~fahri/cryptography.html>, lectur COM457. Diakses 2015-06-14
3. <https://id.wikipedia.org/wiki/Kriptografi>. diakses 2015-06-14
4. Menezes, A. J.; van Oorschot, P. C.; Vanstone, S. A. (1996) *Handbook of Applied Cryptography*. ISBN 0-8493-8523-7
5. The Base16, Base32, and Base64 Data Encodings. IETF. October 2006. RFC 4648 diakses 2015-06-14
6. ASCII format for Network Interchange. IETF. October 16, 1969. RFC 20 diakses 2015-06-21
7. <https://en.wikipedia.org/wiki/ASCII> diakses 2015-06-21
8. <https://id.wikipedia.org/wiki/Bitmap>. diakses 2015-07-14
9. https://en.wikipedia.org/wiki/Request_for_Comments diakses 2015-06-21
10. MF, Mundazir. (2014). *PHP Tutorial Book For Beginner*. Yogyakarta. ISBN 976-602-244-243-3
11. Enterprise, Jubilee. (2014). *HTML 5 Manual Book*. Jakarta: Penerbit PT Elex Media Komputindo ISBN 978-602-02-5357-2
12. http://www.w3schools.com/js/js_intro.asp/ diakses 2015-07-30
13. <https://keinaturalala.wordpress.com/2013/12/13/metodologi-extreme-programming/> diakses 2015-06-27
14. Multipurpose Internet Mail Extentions (MIME) Format of Internet Messege Bodies. IETF. November 1996. RFC 2045 diakses 2015-06-14

LAMPIRAN



Lembar Konsultasi Bimbingan Skripsi

Teknik Informatika
Institut Sains dan Teknologi Nasional

Nama Lengkap : Himawan Dewabrata
Nim : 10364016
Dosen Pembimbing I : Drs. Syarif Hidayatullah, M.Kom
Dosen Pembimbing II : Siti Madinah L,S.Kom, M.Kom
Judul : Perancangan Aplikasi Kriptografi Algoritma Base64 Berbasis Web



No	Hari/Tanggal Bimbingan	Materi Bimbingan	Paraf Dosen Pembimbing
1	16 April 2015	Bab I	
2	25 April 2015	Penyerahan Perbaikan Bab I, Bab II	
3	4 Juli 2015	Bab III, Revisi Bab II Keperjangan Bab I	
4	15 Agustus 2015	Penyerahan Bab II, III, IV	
5	22 Agustus 2015	Revisi Bab III, IV	
6	24 Agus 2015	Bab V, Revisi Bab IV	
7	26 Agustus 2015	Revisi Bab IV, Program	
8	27 Agustus 2015	Bab IV, V dan Program	
9	28 Agust 2015	Demo Program	
10	31 Agustus 2015	Revisi Bab III Perancangan	

- Bimbingan dimulai pada tanggal :
- Bimbingan diakhiri pada tanggal :

Jakarta, 28 Agustus 2015

Pembimbing I

Drs. Syarif Hidayatullah, M.Kom
NIDN 03.22026401

Pembimbing II

Siti Madinah L,S.Kom,M.Kom
NIDN 0307107201



PT. Dyraftama Indotrimitra

SURAT KETERANGAN

Jakarta, 03 September 2015

Ref No. 001/DIA/030915/14

Perihal : Pengajuan Tempat Reset IT

Yang bertanda tangan dibawah ini:

Nama : Irma Kamili

Jabatan : Direktur

Dengan ini menerangkan bahwa, yang tersebut dibawah ini:

Nama : Himawan Dewabrata

NIK : 00105

Jabatan : Operational & Warehouse

Adalah benar telah melakukan riset IT pada perusahaan kami untuk memenuhi tugas akhir yang bersangkutan dengan judul "Perancangan Aplikasi Kriptografi Algoritma Base64 Berbasis Web" terhitung sejak tanggal 20 April 2015 s/d 21 Agustus 2015 dan yang bersangkutan telah telah melakukan tugasnya dengan baik dan penuh tanggung jawab.

Demikian surat keterangan ini dibuat dengan benar, agar dapat digunakan sebagaimana mestinya.

Hormat Kami,

PT. Dyraftama Indotrimitra

Irma Kamili

Direktur

MGT office: WismaNugra Santana, 13th floor Unit 1311, Jl. Jend. Sudirman Kav 7-8 JktPusat 10220.

OPS office: JL. RSPP No. 15C Cilandak, Jakarta 12430

Phone: +62-21-70493311, Fax:+62-21-7692571

Email: info@dyraftama.com

index.php

```
<?php
require_once('com/config.php');
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title><?php print $site_name . ' | Skripsi Himawan D 10364016' ?></title>
    <meta name="description" content="<?php print $site_description ?>">
    <meta name="keywords" content="<?php print $site_keywords ?>">
    <link rel="stylesheet" href="style/layout.css">
</head>
<body>
    <div id="container">
        <div id="header">
            
            </div>
            <div id="content">
                <div id="services">
                    <?php require_once('com/menu.php'); ?>
                </div>
                <br /><br /><br /><br /><br />
                <br /><br /><br /><br /><br />
                <br /><br /><br /><br /><br />
                <?php require_once('com/footer.php'); ?>
            </div>
        </div>
    </body>
</html>
```

encryption.php

```
<?php
require_once('com/config.php');
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <?php
        if($_GET['algorithm'] == 'binary' || $_GET['algorithm'] == 'base64' ||
$_GET['algorithm'] == 'base64Image'){
            $ok = 1;
            $title = $site_name . ' | Convert ' . $_GET['algorithm'] . ' online';
        }
        else{
            $ok = 0;
            $title = 'Error occured';
        }
    </?php
</head>
```

```

    }

    ?>
    <meta charset="utf-8">
    <title><?php print $title ?></title>
    <meta name="description" content="<?php print $site_description ?>">
    <meta name="keywords" content="<?php print $site_keywords ?>">
    <link rel="stylesheet" href="style/layout.css">
    <script src="com/js/jquery.min.js"></script>
    <script src="com/js/jquery.zclip.js.js"></script>
    <script src="com/js/edBase64Teks.js" ></script>
    <script src="com/js/encodeBase64Image.js" ></script>
    <script>
    $(document).ready(function(){
        $('#copy').zclip({
            path:'com/js/ZeroClipboard.swf',
            copy:function(){return $('#output').val();}
        });
    });
    function reset(){
        document.getElementById("input").value = "";
        document.getElementById("output").value = "";
        document.getElementById("inputimage").value = "";
        document.getElementById("gbr").src = "";
    }
    function SelectAll(id){
        document.getElementById(id).focus();
        document.getElementById(id).select();
    }
    </script>
    <noscript>Maaf!! browser anda tidak mendukung JavaScript!</noscript>
</head>
<body>
    <div id="container">
        <div id="header">
            
            </div>
            <div id="content">
                <?php
                    if(isset($_GET['algorithm'])){
                        if($ok == 1){
                            if(file_exists('com/algorithms/ .
$_GET['algorithm'] . '.php')){
                                require_once('com/algorithms/ .
$_GET['algorithm'] . '.php');
                            }
                            else{
                                print '<br /><h2>Error
occured.</h2>';

```

```

attempt detected.');
```

```

class="green">' . $info['ip'] . '</span><span class="red"> & </span><span class="green">' . $info['ag'] .
'</span><br />';
class="green">' . $info['uri'] . '</span><br />';
$info['message'] . '</span><br /><br />';
logged.<a href="index.php"> Go back</a>';

        $info = errorlog('Possible hacking
        print 'Who? <span
        class="green">' . $info['ip'] . '</span><span class="red"> & </span><span class="green">' . $info['ag'] .
        '</span><br />';
        print 'Where? <span
        print 'What? <span class="red">' .
        print 'Your action has been
        print '<br /><br /><br /><br />';
    }
}
else{
    print '<br /><h2>Error occurred.</h2>';
    $info = errorlog('Possible hacking attempt
    print 'Who? <span class="green">' .
    $info['ip'] . '</span><span class="red"> & </span><span class="green">' . $info['ag'] . '</span><br />';
    print 'Where? <span class="green">' .
    print 'What? <span class="red">' .
    print 'Your action has been logged.<a
    href="index.php"> Go back</a>';
    print '<br /><br /><br /><br />';
}
}
else{
    print '<br /><h2>Error occurred.</h2>';
    $info = errorlog('Possible hacking attempt
    print 'Who? <span class="green">' . $info['ip'] .
    '</span><span class="red"> & </span><span class="green">' . $info['ag'] . '</span><br />';
    print 'Where? <span class="green">' . $info['uri'] .
    '</span><br />';
    print 'What? <span class="red">' . $info['message']
    print 'Your action has been logged.<a
    href="index.php"> Go back</a>';
    print '<br /><br /><br /><br />';
}
}
?>
</div>
</div>
</body>
</html>

```

menu.php

```
<a href="encryption.php?algorithm=base64Image">
    <div class="service">
        <div class="text">
            Base64 Gambar
        </div>
    </div>
</a>
</br></br></br>
<a href="encryption.php?algorithm=base64">
    <div class="service">
        <div class="text">
            Base64 Teks
        </div>
    </div>
</a>
```

base64Image.php

```
<h1>Base64 encoder Image</h1>
<h2>Image</h2>
<div id="input_wrapper">
    <br />
    <img id="gbr" />
</div>

<h2> ciphertext</h2>
<div id="input_wrapper">
    <textarea id="output" onClick="SelectAll('output');" readonly></textarea>
</div>

<div id="options">
    <div class="service" id="copy">
        <div class="text">Copy</div>
    </div>
    <div class="service" onClick="window.history.go(-1);">
        <div class="text">Back</div>
    </div>
    <div class="service">
        <input type="file" class="text" accept="image/*" name="gambar" id="inputimage"
onchange="encodeImageFileAsURL();" />
    </div>
</div>
<br /><br /><br /><br /><?php require_once('com/footer.php'); ?>
```

base64.php

```
<h1>Base64 encoder/decoder</h1>
<h2>Plaintext</h2>
<div id="input_wrapper">
    <textarea id="input"></textarea>
</div>
```

```

<h2>ciphertext</h2>
<div id="input_wrapper">
  <textarea id="output" onClick="SelectAll('output');" readonly></textarea>
</div>

<div id="options">
  <div class="service" onClick="base64Encode();">
    <div class="text">Encode</div>
  </div>
  <div class="service" onClick="base64Decode();">
    <div class="text">Decode</div>
  </div>
  <div class="service" id="copy">
    <div class="text">Copy</div>
  </div>
  <div class="service" onClick="reset();">
    <div class="text">Reset</div>
  </div>
  <div class="service" onClick="window.history.go(-1);">
    <div class="text">Back</div>
  </div>
</div>
<br /><br /><br /><br /><?php require_once('com/footer.php'); ?>

```

encode Base64Image.js

```

function encodeImageFileAsURL(){

  var fileGambar = document.getElementById("inputimage").files;
  if (fileGambar.length > 0)
  {
    var fileToLoad = fileGambar[0];

    var fileReader = new FileReader();

    fileReader.onload = function(encodeFileGambar) {
      var srcData = encodeFileGambar.target.result; // <--- data: base64

      var newImage = document.createElement('img');
      newImage.src = srcData;

      document.getElementById("gbr").innerHTML = newImage.outerHTML;
      document.getElementById("output").value = newImage.outerHTML;

    }
    fileReader.readAsDataURL(fileToLoad);
  }
}

```

edbase64Teks.js

```
function base64Encode(){
    text = document.getElementById("input").value.trim();
    if (/([\u0000-\u00ff])/i.test(text)){
        alert("Maaf!! ini bukan karakter ASCII");
    }
    else{
        var CHARS =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";
        var out = "", i = 0, len = text.length, c1, c2, c3;
        while (i < len) {
            c1 = text.charCodeAtAt(i++) & 0xff;
            if (i == len) {
                out += CHARS.charAt(c1 >> 2);
                out += CHARS.charAt((c1 & 0x3) << 4);
                out += "==";
                break;
            }
            c2 = text.charCodeAtAt(i++);
            if (i == len) {
                out += CHARS.charAt(c1 >> 2);
                out += CHARS.charAt(((c1 & 0x3) << 4) | ((c2 & 0xF0) >> 4));
                out += CHARS.charAt((c2 & 0xF) << 2);
                out += "=";
                break;
            }
            c3 = text.charCodeAtAt(i++);
            out += CHARS.charAt(c1 >> 2);
            out += CHARS.charAt(((c1 & 0x3) << 4) | ((c2 & 0xF0) >> 4));
            out += CHARS.charAt(((c2 & 0xF) << 2) | ((c3 & 0xC0) >> 6));
            out += CHARS.charAt(c3 & 0x3F);
        }
        document.getElementById("output").value = out;
    }
}
// ini algoritma mambalikan nilai BASE64 Decode
```

```
function base64Decode(){
    text = document.getElementById("input").value.trim();

    var CHARS =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";
    cur, prev, digitNum,
    i=0,
    result = [];
    text = text.replace(/=/g, "");
    while(i < text.length){
        cur = CHARS.indexOf(text.charAt(i));
        digitNum = i % 4;
```

```
switch(digitNum){
  case 1:
    result.push(String.fromCharCode(prev << 2 | cur >> 4));
    break;
  case 2:
    result.push(String.fromCharCode((prev & 0x0f) << 4 | cur >> 2));
    break;
  case 3:
    result.push(String.fromCharCode((prev & 3) << 6 | cur));
    break;
}
prev = cur;
i++;
}
document.getElementById("output").value = result.join("");
}
```

layout.css

```
body{
    /*background-image: url("images/BG1.jpg");
    background-position: center top;
    background-repeat: repeat;*/
    font-family: Arial, Helvetica, sans-serif;
    color:#FF7519;
}
a { text-decoration:none }
a:link {color:#6F6F77}
a:visited {color:#6F6F77}
a:hover {color:#B8B8B8;}
a:active {color:#B8B8B8}

.border{
    width:50px;
    height: 45px;
}
#container {
    border-radius: 4px;
    background: url(images/outer-wrapper.png) 0 0 repeat;
    background-color: rgba(0, 0, 0, .2);
    width:900px;
    height:auto;
    margin: 0 auto;
    overflow: auto;
    box-shadow: inset 0 0 0 1px rgba(255, 255, 255, 0.06), 0 0 0 1px rgba(0, 0, 0, 0.9), inset 0 0 5px
    rgba(0, 0, 0, 0.4), 0 3px 5px rgba(0, 0, 0, 0.5), 0 2px 0 1px rgba(0, 0, 0, 0.5);
}
#header{
    height:185px;
}
#header img{
    border-style: none;
}
#content{
    height:auto;text-align: center;
}
#services{
    margin-left:20px;
    margin-top:20px;
}
.text{
    text-align: center;
    margin-top:5px;
}
#intro{text-align: center;padding-top: 5px;}
.service{
    background: rgba(255, 255, 255, .06);cursor: pointer;
    border-radius: 4px;
```

```

float:left;
margin-left: 5px;
margin-top: 5px;
width:138px;
height:45px;
text-align:center;
font-size: 18px;
color:#FF7519;border:1px solid rgba(255, 255, 255, .04);
-webkit-transition: all 1s ease-in-out;
-moz-transition: all 1s ease-in-out;
-o-transition: all 1s ease-in-out;
transition: all 1s ease-in-out;
}
.service:hover{
background: rgba(0, 161, 250, 0.1);box-shadow: 0px 0px 10px rgba(0, 161, 250, 0.5);
text-shadow: 0 0 5px rgba(0, 161, 250, .5);-webkit-transition: all 1s ease-in-out;
-moz-transition: all 1s ease-in-out;
-o-transition: all 1s ease-in-out;
transition: all 1s ease-in-out;border: 1px solid #008bc8;}
.service .text{
margin-top: 10px;
}
}
#input_wrapper{
width:98%;
height:125px;
margin-left: 7px;
margin-top:5px;
}
#input_wrapper textarea{
float:left;
width:860px;
height:125px;
border: 1px solid rgba(255, 255, 255, .04);
font-family: Arial, Verdana, Helvetica, sans-serif;
font-size: 19px;
color: #ffffff;
text-shadow: -1px -1px 1px rgba(0, 0, 0, .2);
padding: 5px 12px 5px;
background-color: rgba(255, 255, 255, .05);
border-radius: 6px;
box-shadow: 0 5px 15px rgba(0, 0, 0, .5), 0 5px 5px rgba(0, 0, 0, .5), 0 0px 35px rgba(0, 0, 0, .6)inset;
outline: none;-webkit-transition: all 1s ease-in-out;-moz-transition: all 1s ease-in-out;-o-transition: all
1s ease-in-out;transition: all 1s ease-in-out;
}
}
#input_wrapper img{
float:center;
max-width:150px;
height:100px;
}
}
#input_wrapper textarea:hover{

```

```
border: 1px solid #008bc8;box-shadow: 0px 0px 10px rgba(0, 161, 250, 0.5);
background: rgba(0, 161, 250, 0.05);box-shadow: 0px 0px 10px rgba(0, 161, 250, 0.5);-webkit-transition:
all 1s ease-in-out;-moz-transition: all 1s ease-in-out;-o-transition: all 1s ease-in-out;transition: all 1s ease-
in-out;
}
#options{
    margin-left: 7px;
    margin-top: 25px;
}
.red{
    color:red;
}
h1 {text-align: left;color: #F3EBFB;text-shadow: 0 0 5px rgba(0, 161, 250, 0.4);margin: 14px 10px 0;}
h2 {text-align: left;color: #F3EBFB;text-shadow: 0 0 5px rgba(0, 161, 250, 0.4);margin: 14px 10px 0;}
h3 {text-align: left;color: #F3EBFB;text-shadow: 0 0 5px rgba(0, 161, 250, 0.4);margin: 14px 0px 0;}
.green{
    color:green;
}
#footer_line{
    margin-left: 10px;
    width: 98%;
    border: 1px solid #FF7519;
    height: 0px;
}
#footer{
    margin-left: 10px;
    width: 98%;
}
}
```