

WEB SERVER MANAGEMENT

MODUL KOMPRESI BERKAS

PROYEK AKHIR II

Oleh :

| | |
|----------------------------|-------------------|
| Zaki A Agha | 3311211031 |
| Reza Hidayatulloh A | 3311211018 |
| Nur Setyo Utomo | 3311211027 |
| Siti Chomsatun SSA | 3311211009 |

Disusun untuk memenuhi syarat kelulusan matakuliah Proyek Akhir II



PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
BATAM
2014

LEMBAR PENGESAHAN

Batam, Juni 2014

Pembimbing,

Tri Ramadhani Ardjo, S.ST

NIK. 109066

KATA PENGANTAR

Dengan rahmat dan rasa syukur penulis haturkan kepada Tuhan Yang Maha Esa atas karunia-Nya penulis dapat menyelesaikan laporan Proyek Akhir II dengan baik.

Penulisan laporan Proyek Akhir II ini dimaksudkan untuk persyaratan kelulusan matakuliah Proyek Akhir II. Untuk memenuhi persyaratan tersebut maka penulis membuat "Web Server Management Modul Kompresi Berkas" guna untuk memudahkan pengguna dalam melakukan manajemen berkas pada server khususnya kompresi, ekstrak dan unggah berkas.

Pada kesempatan ini, penulis mengucapkan terima kasih kepada :

1. Ibu Evaliata Br. Sembiring, S.Kom selaku Dosen pengampu matakuliah Proyek Akhir II
2. Bapak Tri Ramadhani Ardjo, S.ST selaku Dosen Pembimbing Proyek Akhir II. Terima kasih atas bimbingan dan motivasi yang Bapak berikan selama penyelesaian Proyek Akhir II.
3. Kedua orang tua atas dukungan dan doanya.
4. Teman-teman satu tim Proyek Akhir II atas kerja samanya.

Penulis juga menyadari masih terdapat banyak kekurangan dalam penyusunan Proyek Akhir II ini. Penulis mengharapkan kritik dan saran yang membangun dari pihak lain.

Semoga Proyek Akhir II ini bermanfaat bagi pembaca, khususnya bagi yang ingin mengembangkan analisis serupa.

Batam, Juni 2014

Penulis

ABSTRAK

Server bertanggung-jawab memberikan layanan *request* dari *client*. Lingkungan kerja pada server adalah lingkungan *text mode*. Dimana lingkungan *text mode* adalah lingkungan yang menggunakan terminal. Bekerja di lingkungan *text mode* membutuhkan pengetahuan lebih, terutama tentang perintah-perintah yang berjalan di lingkungan tersebut. Salah satu contoh dalam melakukan manajemen berkas pada server.

Manajemen berkas adalah tindakan yang dilakukan untuk mengatur berkas-berkas yang tersimpan di server. Sehingga dengan adanya manajemen berkas server dapat bekerja dengan baik tanpa adanya masalah yang memungkinkan terjadinya sebuah server gagal berfungsi (*down*). Selama ini proses manajemen berkas menggunakan lingkungan *text mode* sering kali sulit digunakan karena pengguna menghafal perintah-perintah yang berjalan di lingkungan tersebut. Maka, dibuatlah suatu panel manajemen berkas guna untuk membantu pengguna dalam melakukan manajemen berkas dalam hal ini ruang lingkupnya melakukan kompresi dan ekstrak berkas.

Kompresi berkas adalah proses memampatkan ukuran berkas sehingga membutuhkan penyimpanan yang lebih kecil dan mempersingkat dalam melakukan transfer data.

Kata Kunci: Web Server, Manajemen Server, Kompresi berkas, Ekstrak berkas, ZIP.

DAFTAR ISI

| | |
|--|------|
| LEMBAR PENGESAHAN | ii |
| KATA PENGANTAR | iii |
| ABSTRAK | iv |
| DAFTAR ISI..... | v |
| DAFTAR GAMBAR | vii |
| DAFTAR TABEL..... | viii |
| Bab I PENDAHULUAN | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 2 |
| 1.3 Batasan Masalah | 2 |
| 1.4 Tujuan Penelitian | 2 |
| 1.5 Sistematika Penulisan | 2 |
| Bab II LANDASAN TEORI | 4 |
| 2.1 Web Server | 4 |
| 2.1.1 PHP | 4 |
| 2.2 Antarmuka Pemakai | 5 |
| 2.3 Manajemen Berkas | 7 |
| 2.4 Kompresi | 8 |
| 2.4.1 Teknik Kompresi Data | 8 |
| 2.4.2 Defenisi Algoritma..... | 9 |
| 2.4.3 Format Kompresi | 10 |
| Bab III ANALISIS DAN PERANCANGAN | 18 |
| 3.1 Deskripsi Umum Sistem..... | 18 |
| 3.2 Diagram Use Case | 19 |
| 3.3 Skenario Use Case | 19 |
| 3.3.1 Use Case Kompresi Berkas..... | 19 |
| 3.3.2 Use Case Ekstrak Berkas | 19 |

| | | |
|--------|---|----|
| 3.4 | Sequence Diagram | 20 |
| 3.4.1 | Sequence Diagram Kompresi berkas | 20 |
| 3.4.2 | Sequence Diagram Ekstrak Berkas | 20 |
| 3.5 | Class Diagram..... | 21 |
| Bab IV | IMPLEMENTASI DAN PENGUJIAN | 22 |
| 4.1 | Struktur Direktori dan Deskripsi Berkas | 22 |
| 4.2 | Implementasi Antarmuka | 23 |
| 4.2.1 | Halaman Utama..... | 23 |
| 4.2.2 | Kompres Berkas | 24 |
| 4.2.3 | Ekstrak Berkas | 25 |
| 4.3 | Pengujian | 25 |
| 4.3.1 | Tujuan Pengujian | 25 |
| 4.3.2 | Metode Pengujian | 26 |
| 4.3.3 | Rencana Pengujian..... | 26 |
| 4.3.4 | Pembahasan dan Hasil Pengujian..... | 26 |
| Bab V | KESIMPULAN DAN SARAN | 27 |
| 5.1 | Kesimpulan..... | 27 |
| 5.2 | Saran | 27 |
| | DAFTAR PUSTAKA | 28 |
| | LAMPIRAN..... | 29 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 1 Struktur Zip | 16 |
| Gambar 2 Alur Sistem..... | 18 |
| Gambar 3 Diagram Use Case | 19 |
| Gambar 4 Sequence Diagram Kompresi Berkas..... | 20 |
| Gambar 5 Sequence Diagram Ekstrak berkas | 20 |
| Gambar 6 Class Diagram | 21 |
| Gambar 7 Antarmuka Halaman Utama..... | 23 |
| Gambar 8 Antarmuka Kompres Berkas | 24 |
| Gambar 9 Antarmuka Ekstrak Berkas..... | 25 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 1 Daftar Direktori dan Berkas | 22 |
| Tabel 2 Tabel Pengujian Menggunakan Metode <i>Black-Box</i> | 29 |

Bab I PENDAHULUAN

1.1 Latar Belakang

Server merupakan sebuah perangkat komputer yang mempunyai tanggungjawab melayani seluruh permintaan dari client. Lingkungan kerja pada server adalah lingkungan kerja *text mode* khususnya server linux. Lingkungan *text mode* adalah lingkungan kerja yang hanya menggunakan terminal. Secara global pada lingkungan ini untuk melakukan konfigurasi, pengguna harus mengetikkan perintah-perintah untuk dieksekusi. Salah satu contoh ketika akan melakukan manajemen berkas pada server.

Selama ini proses manajemen berkas pada server yang menggunakan lingkungan *text mode* sering kali sulit digunakan. Karena, pertama pengguna harus mengetikkan perintah sesuai ejaan, tata bahasa, dan tanda baca yang tepat. Kedua pengguna harus mengetahui perintah-perintah yang digunakan pada lingkungan *text mode*. Salah satu contoh untuk melakukan kompresi berkas ke format .zip menggunakan perintah “*zip -r <nama berkas.zip yang ingin dibuat> <berkas yang akan di kompres>*” dan untuk ekstrak berkas format .zip menggunakan perintah “*unzip nama-berkas.zip*”. Perintah-perintah tersebut terkadang tidak dimengerti dan membingungkan bagi pengguna khususnya pengguna baru. Hal tersebut merupakan salah satu kekurangan pada manajemen berkas dalam hal memberi kemudahan bagi pengguna khususnya pengguna baru yang belum memahami perintah-perintah pada lingkungan kerja *text mode*. Berdasarkan permasalahan tersebut, maka di bangunlah sebuah panel manajemen berkas yang bisa membantu pengguna dalam melakukan manajemen pada server dalam hal ini ruang lingkupnya melakukan kompresi dan ekstrak berkas pada server.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas maka dapat dirumuskan yang menjadi pokok pembahasan yaitu:

1. Bagaimana cara melakukan manajemen berkas pada server?
2. Bagaimana cara membangun modul kompresi berkas berbasis web?
3. Bagaimana melakukan kompresi berkas dan ekstrak berkas menggunakan modul kompresi berkas berbasis web?

1.3 Batasan Masalah

Adapun batasan masalah dalam penulisan Proyek Akhir II ini meliputi:

1. Modul ini hanya dapat melakukan kompresi berkas format .zip.
2. Modul ini hanya dapat melakukan ekstrak berkas format .zip.

1.4 Tujuan Penelitian

Tujuan dari Proyek Akhir II ini adalah sebagai berikut:

1. Dapat membangun modul kompresi berkas berbasis web.
2. Dapat membantu pengguna dalam melakukan kompresi berkas dan ekstrak berkas pada server.
3. Dapat melakukan unggah dan ekstrak berkas otomatis.

1.5 Sistematika Penulisan

Sistematika penulisan laporan ini adalah:

Bab I : Pendahuluan berisi tentang Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan dan Sistematika Penulisan.

Bab II : Landasan Teori berisi tentang penjelasan Web Server Management, web interface dan penjelasan beberapa aplikasi penunjang

Bab III : Analisis dan Perancangan berisi tentang deskripsi umum sistem, Use Case Diagram, Skenario Use Case, Sequence Diagram dan Class Diagram.

Bab IV : Implementasi dan Pengujian setelah dilakukan perancangan maka tahap selanjutnya adalah implementasi dan pengujian aplikasi. Implementasi akan menghasilkan aplikasi yang dapat dijalankan di lingkungan operasional. untuk mengetahui apakah aplikasi tersebut dapat melakukan fungsi sesuai deskripsi perencanaan maka perlu dilakukan pengujian.

Bab V : Kesimpulan berisi tentang kesimpulan yang merupakan rangkuman dari hasil analisis kinerja pada bagian sebelumnya serta saran-saran pengembangan dari penelitian yang dibuat dan aspek yang belum terselesaikan.

Bab II LANDASAN TEORI

2.1 Web Server

Server untuk web seringkali disebut Web Server. Jadi, Web Server adalah tempat diletakkannya berkas-berkas web. Kata Web Server sendiri dapat mengacu pada dua hal, yang sebenarnya sama, hanya saja ruang lingkupnya berbeda:

1. Program komputer yang bertanggung-jawab untuk menerima permintaan HTTP dari *client (browser)* kemudian meresponnya dengan respon HTTP dengan disertai konten lainnya. Misalnya dokumen HTML dan objek yang berkaitan seperti gambar, video, dan *file script* lainnya.
2. Komputer yang menjalankan program server. Tapi pengertian ini lebih luas dan mengacu pada keseluruhan komputer tidak hanya program didalamnya. Arti Web Server ini mencakupi sistem operasi, program aplikasi dan *hardware*.

2.1.1 PHP

PHP ditinjau dari jenis bahasanya merupakan bahasa script seperti HTML. Jadi, *source code* dari program PHP tetap dalam bentuk teks dan langsung di *parse* oleh *software* PHP di *server* tanpa harus diubah ke bentuk lainnya terlebih dahulu seperti *file executable* dalam pemrograman *desktop*.

PHP punya banyak fitur yang membuat *desainer* dan *programmer* membuat halaman web dengan lebih mudah. PHP adalah singkatan dari *PHP Hypertext Preprocessor*. Pertama kali dikembangkan oleh *programmer* bernama Ramus Ledorf, PHP awalnya adalah singkatan dari *Personal Home Page Tools*. Namun, setelah dikembangkan oleh Zeev Suraski dan Andi Gutmans dan fiturnya bertambah, PHP diubah singkatannya menjadi yang sekarang ini.

Ditinjau dari segi sintak bahasanya PHP mirip dengan C. Secara keseluruhan, PHP lebih sederhana dibandingkan dengan C karena PHP tidak menggunakan konsep yang susah seperti C. selain itu PHP tidak seperti C yang tidak memiliki fitur pemrograman *low level* untuk langsung mengakses hardware.

Beberapa kelebihan PHP dari bahasa pemrograman web :

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web Server yang mendukung PHP dapat ditemukan dimana-mana dari mulai Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan pengembang yang siap membantu dalam pengembangannya.
4. Dalam sisi pemaham, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah sistem.

2.2 Antarmuka Pemakai

Antarmuka Pemakai merupakan mekanisme komunikasi antara pengguna dengan sistem. Antarmuka pemakai dapat menerima informasi dari pengguna dan memberikan informasi kepada pengguna untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan suatu solusi.

Contoh konsep luas antarmuka pemakai mencakup aspek interaktif sistem operasi komputer, perkakas tangan, operator kontrol mesin berat dan proses kontrol. Pertimbangan desain yang berlaku saat membuat antarmuka pemakai berkaitan dengan ergonomik dan psikologi.

Antarmuka pemakai yang ada untuk berbagai sistem, dan menyediakan cara :

- *Input*, memungkinkan pengguna untuk memanipulasi sistem.
- *Output*, memungkinkan sistem untuk menunjukkan efek manipulasi pengguna.

Antarmuka pemakai ada dua jenis, yaitu :

1. *Graphical User Interface (GUI)* : Menggunakan unsur-unsur multimedia (seperti gambar, suara, video) untuk berinteraksi dengan pengguna.
2. *Text-Based* : Menggunakan rumus yang sudah ditentukan untuk memberikan perintah.

Saat ini antarmuka yang banyak digunakan dalam *software* adalah *GUI* (*Graphical User Interface*). GUI memberikan keuntungan seperti:

1. Mudah dipelajari oleh pengguna yang pengalaman dalam menggunakan komputer cukup minim.
2. Berpindah dari satu layar ke layar yang lain tanpa kehilangan informasi.
3. Akses penuh pada layar dengan segera untuk beberapa macam tugas/keperluan.

Ada 5 tipe utama interaksi pengguna:

1. Pengoperasian secara langsung : interaksi langsung dengan objek pada layar. Misalnya hapus berkas dengan memasukkannya ke *trash*. Contoh: *Video games*. Kelebihan : Waktu pembelajaran user sangat singkat, *feedback* langsung diberikan pada tiap aksi sehingga kesalahan terdeteksi dan diperbaiki dengan cepat. Kekurangan : antarmuka tipe ini rumit dan memerlukan banyak fasilitas pada sistem komputer, cocok untuk penggambaran secara visual untuk satu operasi atau objek.
2. Pilihan berbentuk menu : Memilih perintah dari daftar yang disediakan. Misalnya saat klik kanan dan memilih aksi yang dikehendaki. Kelebihan : User tidak perlu ingat nama perintah. Pengetikan minimal. Kesalahan rendah.

Kekurangan : Tidak ada logika AND atau OR. Perlu ada struktur menu jika banyak pilihan. Menu dianggap lambat oleh *expert user* dibanding *command language*.

3. Pengisian form : Mengisi area-area pada form. Contoh : *Stock control*. Kelebihan : Masukan data yang sederhana. Mudah dipelajari Kekurangan : Memerlukan banyak tempat di layar. Harus menyesuaikan dengan form manual dan kebiasaan pengguna.
4. Perintah tertulis : Menuliskan perintah yang sudah ditentukan pada program. Contoh: *operating system*. Kelebihan : Perintah diketikan langsung pada system. Misal UNIX, DOS command. Bisa diterapkan pada terminal yang murah. Kombinasi perintah bisa dilakukan. Misal kompres berkas dan ekstrak berkas. Kekurangan : Perintah harus dipelajari dan diingat cara penggunaannya, tidak cocok untuk user biasa. Kesalahan pakai perintah sering terjadi. Perlu ada sistem pemulihan kesalahan. Kemampuan mengetik perlu.
5. Perintah dengan bahasa alami : Menggunakan bahasa alami untuk mendapatkan hasil. Contoh: *search engine* di Internet. Kelebihan: Perintah dalam bentuk bahasa alami, dengan kosa kata yang terbatas, misalnya kata kunci yang kita tentukan untuk dicari oleh *search engine*. Ada kebebasan menggunakan kata-kata. Kekurangan: Tidak semua sistem cocok gunakan ini. Jika digunakan maka akan memerlukan banyak pengetikan.

2.3 Manajemen Berkas

Manajemen berkas adalah tindakan yang dilakukan untuk mengatur berkas-berkas yang tersimpan di server. Sehingga dengan adanya manajemen berkas server dapat bekerja dengan baik tanpa adanya masalah yang memungkinkan terjadinya sebuah server gagal berfungsi (*down*).

Dalam manajemen server ada beberapa hal yang perlu diketahui diantaranya adalah *hardware* jaringan apa saja yang digunakan, arsitektur jaringan komputer

yang dipilih beserta kelemahan dan kelebihanannya serta layanan apa saja yang digunakan oleh server yang ingin di kelola. Tanpa adanya informasi-informasi tersebut seorang administrator server akan kesulitan dalam mengelolanya.

Dalam sebuah server semua perangkat seperti sistem operasi, *hardware*, aplikasi dan jaringan merupakan elemen yang sangat penting dan mutlak harus ada, karena tanpa adanya salah satu dari komponen tersebut maka server tidak akan dapat bekerja.

Fungsi Manajemen Berkas

- a. Manajemen berkas berguna untuk pengalamatan data pada media penyimpanan (Harddisk, UFD, CDROM, dsb), memanipulasi berkas salin, ubah, delete dan sebagainya.
- b. Dapat mengurangi resiko kehilangan berkas yang dikarenakan: terhapus secara tidak disengaja, tertimpa berkas baru, tersimpan dimana saja, dan hal lain yang tidak kita inginkan

2.4 Kompresi

Kompresi adalah sebuah cara untuk memadatkan data sehingga hanya memerlukan ruangan penyimpanan lebih kecil sehingga lebih efisien dalam menyimpannya atau mempersingkat waktu pertukaran data tersebut.

2.4.1 Teknik Kompresi Data

Kompresi data ada dua jenis :

1. Kompresi data tanpa kehilangan (*Lossles*)

Teknik ini mampu memadatkan data dan mengembalikannya sama persis seperti semula. Tidak ada informasi yang hilang atau harus dikurangi dalam proses untuk mengurangi ukuran besar. Biasanya algoritma pemadatan data jenis ini

menggunakan prinsip kelebihan statistik supaya data bisa disimpan dengan lebih ringkas.

Pemadatan tanpa kehilangan memiliki batas rendah di mana berkas tidak bisa dipadatkan lebih jauh lagi. Teorem Shannon menunjukkan bahwa pemadatan data tidak bisa menghasilkan kadar kode yang lebih rendah daripada Entropi Shannon berkas, tanpa menyebabkan kehilangan informasi. Maka, apabila suatu berkas sudah dipadatkan (misalnya, berkas gambar disimpan di berkas .zip), berkas .zip tersebut sudah tidak bisa dipadatkan lagi.

2. Kompresi data berkehilangan (Lossy)

Teknik ini, data hasil dekompresi tidak sama dengan data sebelum kompresi namun sudah "cukup" untuk digunakan. contoh Mp3, streaming media, Jpeg, Mpeg, dan WMA. Kelebihannya ukuran berkas lebih kecil dibanding teknik kompresi data tanpa kehilangan namun masih tetap memenuhi syarat untuk digunakan.

Biasanya teknik ini membuang bagian-bagian data yang sebenarnya tidak begitu berguna. Misalnya terdapat gambar asli berukuran 12,249 *bytes*, kemudian dilakukan kompresi dengan Jpeg kualitas 30 dan berukuran 1,869 *bytes* berarti gambar tersebut 85% lebih kecil dan *ratio* kompresi 15%.

2.4.2 Defenisi Algoritma

Defenisi Algoritma :

1. Algoritma adalah urutan langkah – langkah berhingga untuk memecahkan masalah logika atau matematika
2. Algoritma adalah logika, metode dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan
3. Algoritma adalah urutan langkah – langkah logis penyelesaian masalah yang disusun secara sistematis dan logis.

2.4.3 Format Kompresi

Contoh format kompresi berkas:

1. Berkas RAR

RAR merupakan arsip berkas yang berisi satu atau lebih berkas yang dikompresi dengan *RAR compression*. Jenis berkas ini umumnya menggunakan rasio kompresi yang lebih tinggi dibandingkan dengan kompresi ZIP. *RAR compressor* dapat menciptakan *Spanned* atau *multi-volume*, dan arsip yang terbagi menjadi beberapa berkas kompresi. Berkas ini secara khas memiliki ekstensi *.rar*. Nama RAR mengacu pada Roshal Archive yang merupakan nama belakang Eugene Roushal, sang pengembang algoritma kompresi RAR.

Keunggulan kompresi berkas RAR ini adalah berkas dapat di *uncompressed* atau di ekstrak kembali seperti berkas aslinya tanpa adanya kekhawatiran kehilangan data. Berkas hasil kompresi RAR yang cenderung berukuran lebih kecil sering kali digunakan untuk menyimpan atau mentransfer berkas lintas internet.

2. Berkas ZIP

Zip adalah berkas format yang digunakan untuk kompresi data. Sebuah berkas zip berisi satu atau lebih berkas yang telah dikompresi, untuk mengurangi ukuran berkas, ataupun disimpan. Format berkas zip melakukan kompresi dengan cara menggunakan algoritma kompresi.

Format ini awalnya diciptakan pada tahun 1989 oleh Phil Katz, dan pertama kali diimplementasikan dalam utilitas PKZip PKWARE itu, sebagai pengganti atau penyempurna format kompresi ARC sebelumnya diciptakan oleh Thom Henderson.

Format zip sekarang didukung oleh banyak utilitas perangkat lunak lain dari PKZip. Microsoft telah termasuk Sistem Operasi yang mendukung zip (dengan nama "*compressed folders*") dalam versi Microsoft Windows sejak tahun 1998.

Apple juga mendukung zip di Mac OS X 10.3 (melalui BOMArchiveHelper, sekarang *Archive utility*) dan kemudian, berkembang bersama dengan format kompresi lainnya.

Berkas Zip umumnya menggunakan ekstensi berkas "zip." Atau "ZIP." Dan jenis media MIME aplikasi / zip. Zip digunakan sebagai format berkas base oleh banyak program, biasanya di bawah nama yang berbeda sesuai dengan *software developer* asal (7-Zip, WinZip, dll).

Format ZIP adalah yang paling populer. Sebagai contoh, kebanyakan arsip di Internet adalah arsip berformat ZIP. Jika Anda akan mengirimkan suatu arsip RAR ke seseorang, mungkin saja teman Anda tidak mempunyai WinRAR untuk mengekstrak muatan yang ada dalam arsip RAR, hal ini bisa saja membuat Anda untuk memilih format ZIP. Pada sisi lain, Anda dapat mengirimkan suatu arsip yang berjenis *self-extracting* (dapat mengekstrak dirinya sendiri). Arsip *self-extracting* memiliki sedikit ukuran yang lebih besar, tetapi dapat mengekstrak dirinya tanpa bantuan program apa pun.

Keuntungan ZIP yang lain adalah kecepatan. Proses penciptaan arsip ZIP lebih cepat dari arsip RAR.

a. Desain Format Berkas Zip

Zip merupakan format arsip sederhana yang menyimpan beberapa berkas. Zip memungkinkan berkas yang terkandung untuk dikompresi menggunakan metode yang berbeda, serta hanya menyimpan berkas tanpa kompresi. Setiap berkas disimpan secara terpisah, memungkinkan berkas yang berbeda dalam arsip yang sama yang akan dikompresi menggunakan metode yang berbeda.

Direktori ditempatkan di akhir sebuah berkas zip. Ini mengidentifikasi berkas apa yang di zip dan mengidentifikasi di mana dalam zip berkas berada. Hal ini memungkinkan pembaca zip ke memuat daftar berkas tanpa membaca seluruh arsip zip. Zip arsip juga bisa memasukkan data tambahan yang tidak terkait

dengan arsip zip. Hal ini memungkinkan untuk zip arsip untuk dibuat menjadi *self-extracting* arsip, aplikasi yang terkandung dekompresi data mereka, dengan memasukkan kode program dalam arsip zip dan menandai berkas sebagai *executable* (yaitu, dengan ekstensi exe.). Di sisi lain, juga memungkinkan untuk sebuah berkas berbahaya, seperti berkas gambar GIF, untuk menyembunyikan kode berbahaya dengan membuat berkas arsip zip.

Format zip menggunakan algoritma 32-bit CRC dan mencakup dua salinan dari struktur direktori dari arsip untuk memberikan perlindungan lebih besar terhadap kehilangan data.

b. Metode Kompresi (DEFLATE)

Deflate adalah Algoritma kompresi bersifat data lossless yang menggunakan kombinasi dari algoritma LZ77 dan Huffman coding. Algoritma asli seperti yang dirancang oleh Katz telah dipatenkan sebagai US patent 5051745 dan diberikan hak untuk PKWARE.

a) Stream format

Sebuah aliran Deflate terdiri dari serangkaian blok. Setiap blok didahului oleh sebuah header 3-bit:

1 bit : sebagai bit penanda dan berada di akhir streams format:

1 : ini adalah blok terakhir di streams format

0 : ada blok lagi untuk memproses setelah yang satu ini.

2 bit: Encoding metode yang digunakan untuk jenis blok:

00 : bagian yang disimpan / baku / literal, antara 0 dan 65.535 byte

01 : static Huffman compressed block, menggunakan pohon Huffman

10 : compressed block complete dengan tabel Huffman

11 : tidak digunakan.

Sebagian besar blok akan berakhir menjadi dikodekan menggunakan metode 10, pengkodean Huffman dinamis, yang menghasilkan sebuah pohon Huffman dioptimalkan disesuaikan untuk setiap blok data individual.

Kompresi dicapai melalui dua langkah :

1. Pencocokan dan penggantian string duplikat dengan pointer
2. Mengganti simbol dengan yang baru, simbol tertimbang berdasarkan frekuensi penggunaannya.

b) Duplicate string elimination

Dalam blok yang sedang dikompresi, jika terlihat suatu *byte* yang ganda, maka metode *back* referensi dimasukkan, menghubungkan ke lokasi sebelumnya bahwa string identik sebagai gantinya. Metode *back* referensi dapat dibuat di sejumlah blok, selama jarak muncul dalam 32 kB terakhir dari data terkompresi *decode* (disebut *sliding window*).

c) Bit Reduction

Tahap kedua terdiri dari kompresi menggantikan simbol yang umum digunakan dengan representasi simbol pendek dan kurang umum digunakan dengan representasi ulang. Metode yang digunakan adalah pengkodean Huffman yang menciptakan pohon *unprefixed non-overlapping interval*, di mana panjang setiap urutan berbanding terbalik dengan kemungkinan bahwa simbol yang perlu dikodekan. Lebih besar kemungkinan simbol harus dikodekan, maka akan semakin pendek urutan bitnya.

Sebuah pohon dibuat yang berisi ruang untuk 288 simbol:

0-255 : mewakili byte literal / simbol 0-255.

256 : ujung blok - pengolahan berhenti jika blok terakhir, jika tidak mulai memproses blok berikutnya.

257-285: dikombinasikan dengan ekstra-bit, panjang pertandingan 3-258 byte.

286, 287: tidak digunakan, bagian pendiam dan ilegal tapi masih pohon. Sebuah kode panjang pertandingan akan selalu diikuti dengan kode jarak jauh. Berdasarkan pada jarak kode dibaca, lebih "ekstra" bit dapat dibaca dalam rangka untuk menghasilkan jarak akhir.

Pohon Jarak berisi ruang untuk 32 simbol:

0-3: jarak 1-4

4-5: jarak 5-8, 1 bit tambahan

6-7: jarak 9-16, 2 bit ekstra

8-9: jarak 17-32, 3 bit ekstra

...

26-27: jarak 8,193-16,384, 12 bit ekstra

28-29: jarak 16,385-32,768, 13 bit ekstra

30-31: tidak digunakan, bagian pendiam dan ilegal tapi masih pohon.

Perhatikan bahwa untuk simbol pertandingan jarak 2-29, jumlah bit ekstra dapat dihitung sebagai : $22/2 - 1$

d) Encoder/compressor

Selama tahap kompresi, *encoder* yang memilih jumlah waktu yang dihabiskan untuk mencari pencocokan string. Implementasi referensi *zlib* / *gzip* memungkinkan pengguna untuk memilih dari skala geser mungkin mengakibatkan tingkat kompresi dengan kecepatan *encoding*. Pilihan berkisar dari -0 (jangan mencoba kompresi, hanya menyimpan terkompresi) ke -9 mewakili kemampuan maksimum dari implementasi referensi dalam *zlib* / *gzip*.

Encoders Deflate lainnya telah diproduksi, yang semuanya juga akan menghasilkan sebuah bitstream kompatibel mampu didekompresi oleh *decoder* Deflate yang ada. Implementasi yang berbeda mungkin akan menghasilkan variasi pada akhir dikodekan bit-stream diproduksi.

c. Struktur Zip

Sebuah berkas zip adalah diidentifikasi dengan adanya direktori pusat yang terletak di ujung struktur dalam rangka untuk memungkinkan menambahkan berkas baru. Direktori pusat menyimpan daftar nama-nama entri (berkas atau direktori) disimpan dalam berkas zip, bersama dengan metadata lain tentang entri, dan *offset* ke berkas zip, menunjuk ke entri data aktual. Hal ini memungkinkan daftar berkas arsip yang akan dilakukan relatif cepat, karena seluruh arsip tidak harus dibaca untuk melihat daftar berkas. Entri dalam berkas zip juga mencakup informasi ini untuk redundansi. Urutan entri berkas dalam direktori tidak perlu bertepatan dengan urutan entri berkas dalam arsip.

Setiap entri diperkenalkan oleh sebuah *header* lokal dengan informasi tentang berkas seperti ukuran, komentar berkas dan nama berkas, diikuti oleh opsional "Extra" bidang data, dan kemudian, mungkin terkompresi berkas data dienkripsi mungkin. "Ekstra" bidang data adalah kunci untuk diperpanjang dari format zip. "Ekstra" bidang dieksploitasi untuk mendukung format Zip64, WinZip-kompatibel enkripsi AES, atribut berkas, dan lebih tinggi resolusi NTFS atau Unix cap waktu berkas. Ekstensi lain yang mungkin melalui bidang "Extra". Zip alat yang diperlukan oleh spesifikasi untuk mengabaikan bidang ekstra mereka tidak mengenali.

Format zip menggunakan spesifik 4-byte "*signatures*" untuk menunjukkan berbagai struktur dalam berkas. Setiap entri berkas ditandai dengan tanda tangan khusus. Awal direktori pusat ditandai dengan tanda tangan yang berbeda, dan setiap entri dalam direktori pusat ditandai dengan tanda tangan lain khususnya 4-byte.

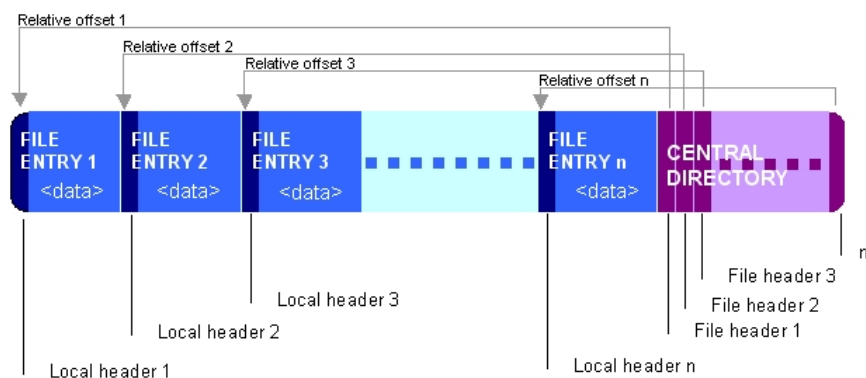
Tidak ada BOF atau EOF penanda dalam spesifikasi zip. Seringkali hal pertama dalam berkas zip adalah entri zip, yang dapat diidentifikasi dengan mudah oleh signature. Tapi itu tidak selalu terjadi bahwa berkas zip dimulai dengan entri zip, dan tidak diperlukan oleh spesifikasi pos.

Tools yang benar membaca arsip zip harus memindai *signatures* dari berbagai bidang, zip direktori pusat. Mereka tidak harus memindai entri karena hanya menentukan direktori mana berkas yang sepotong dimulai. Pemindaian dapat menyebabkan positif palsu, sebagai format data lainnya memungkinkan untuk menjadi antara potongan.

Spesifikasi juga mendukung arsip rar menyebar di beberapa *file system*. Awalnya ditujukan untuk penyimpanan berkas zip besar di beberapa disk 1,44 MB floppy disk, fitur ini sekarang digunakan untuk mengirim arsip zip di bagian melalui email, atau melalui mengangkut lain atau removable media.

FAT filesystem DOS memiliki resolusi timestamp dari hanya dua detik; zip berkas catatan meniru ini. Akibatnya, *built-in timestamp* resolusi berkas dalam sebuah arsip zip adalah hanya dua detik, meskipun bidang tambahan dapat digunakan untuk menyimpan cap waktu lebih akurat.

Pada bulan September 2007, PKZip merilis sebuah revisi dari spesifikasi zip yang berisi ketentuan untuk menyimpan nama berkas menggunakan UTF-8, akhirnya menambahkan kompatibilitas Unicode untuk zip.



Gambar 1 Struktur Zip

d. Kombinasi dengan format lain

Format berkas zip memungkinkan untuk memberi komentar yang berisi semua data pada akhir berkas setelah direktori pusat. Selain itu, karena direktori pusat menentukan *offset* setiap berkas dalam arsip yang berkaitan untuk menjalankan, ada kemungkinan dalam praktek untuk masuknya berkas pertama harus mulai pada *offset* lain dari nol.

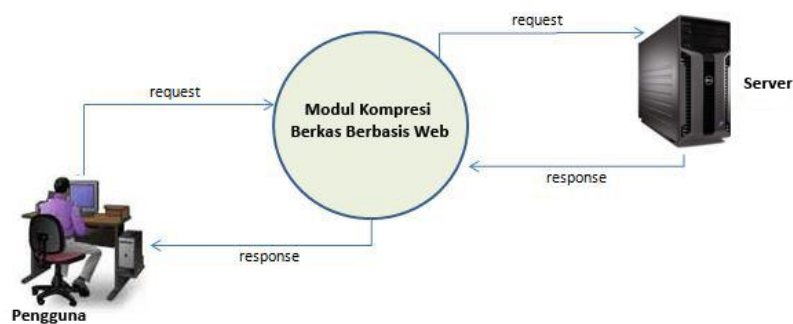
Hal ini memungkinkan data acak akan terjadi di berkas baik sebelum dan setelah data, untuk arsip masih dapat dibaca oleh aplikasi zip. Sebuah efek samping dari hal ini adalah mungkin untuk title sebuah berkas yang kedua bekerja sebagai arsip zip dan format lain, asalkan format lain mentolerir data yang acak di ujungnya, awal, atau tengah. *Self-extracting* arsip (SFX), dari bentuk didukung dengan WinZip dan DotNetZip, mengambil keuntungan dari ini. Exe berkas yang sesuai dengan spesifikasi AppNote.txt PKZip dan bisa dibaca oleh alat zip yang sesuai.

Properti dari format zip, dan dari format JAR yang merupakan varian dari zip, dapat dimanfaatkan untuk menyembunyikan kelompok Java yang berbahaya dalam berkas yang tampaknya tidak berbahaya, seperti gambar GIF yang di upload ke web. GIFAR dapat disebut juga sebagai sebuah serangan efektif terhadap aplikasi web seperti Facebook.

Bab III ANALISIS DAN PERANCANGAN

3.1 Deskripsi Umum Sistem

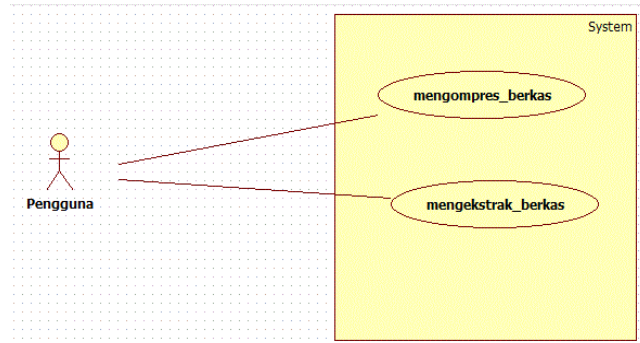
Deskripsi umum modul kompresi berkas berbasis web direpresentasikan pada gambar 2.



Gambar 2 Alur Sistem

Keterangan : modul kompresi berkas berbasis web merupakan panel manajemen berkas yang menyediakan layanan untuk melakukan kompres dan ekstrak berkas pada server. Modul ini hanya terbatas pada berkas arsip format .zip.

3.2 Diagram Use Case



Gambar 3 Diagram Use Case

3.3 Skenario Use Case

3.3.1 Use Case Kompresi Berkas

Nama use case : mengompres_berkas

Aktor : Pengguna

Kondisi Awal : Aktor ingin mengompres berkas ke format .zip

Kondisi Akhir : Aktor telah mengompres berkas ke format .zip

Tujuan : Agar pengguna bisa melakukan kompresi berkas ke format .zip pada server.

3.3.2 Use Case Ekstrak Berkas

Nama use case : mengekstrak_berkas

Aktor : Pengguna

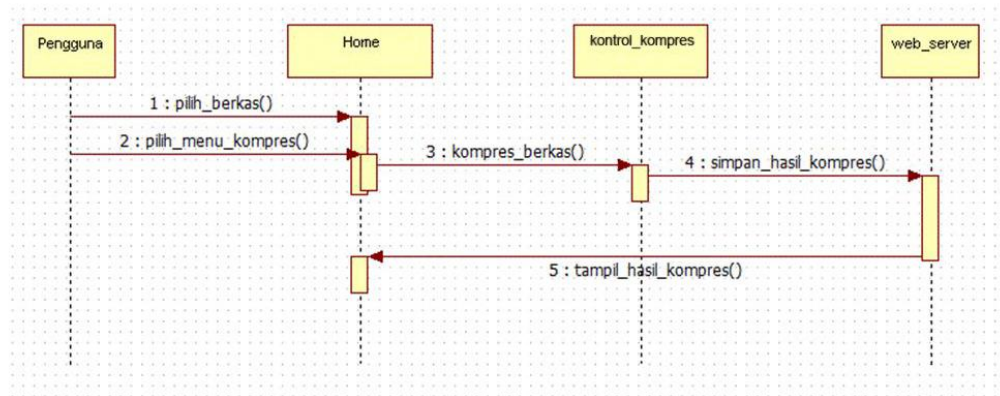
Kondisi Awal : Aktor ingin mengekstrak berkas yang telah terkompres dalam format .zip

Kondisi Akhir : Aktor telah mengekstrak berkas yang telah terkompres dalam format .zip

Tujuan : Agar pengguna bisa melakukan ekstrak berkas yang terkompres dalam format .zip pada server.

3.4 Sequence Diagram

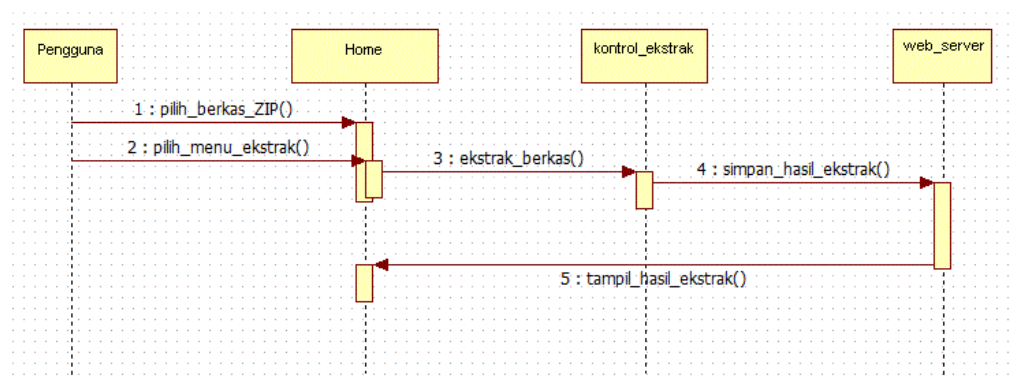
3.4.1 Sequence Diagram Kompresi berkas



Gambar 4 Sequence Diagram Kompresi Berkas

Keterangan : pengguna ingin melakukan kompresi berkas pada server. Pengguna memilih berkas dan memilih menu kompres. Proses dikelola kontrol kompres. Berkas hasil kompresi di simpan di web server. Kemudian ditampilkan di home.

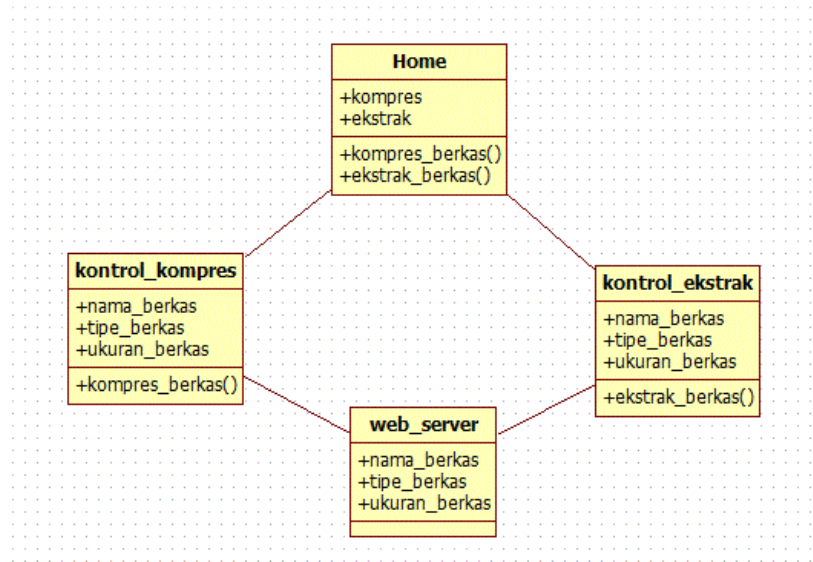
3.4.2 Sequence Diagram Ekstrak Berkas



Gambar 5 Sequence Diagram Ekstrak berkas

Keterangan : pengguna ingin melakukan ekstrak berkas pada server. Pengguna memilih berkas arsip zip dan memilih menu ekstrak pada home. Proses dikelola control ekstrak. Berkas hasil ekstrak di simpan di web server. Kemudian ditampilkan di home.

3.5 Class Diagram



Gambar 6 Class Diagram

Keterangan: pengguna melakukan kompres dan ekstrak berkas dengan memilih menu pada pada home. Proses dikelola kontrol kompres dan kontrol ekstrak. Berkas hasil kompres dan ekstrak akan di simpan di web server kemudian di tampilkan pada home.

Bab IV IMPLEMENTASI DAN PENGUJIAN

4.1 Struktur Direktori dan Deskripsi Berkas

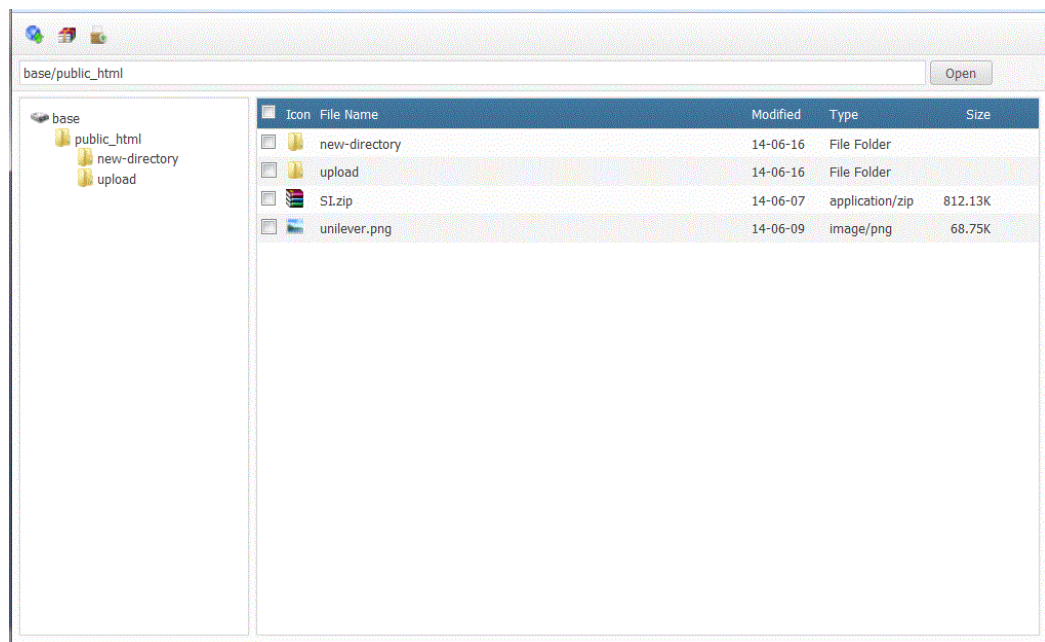
Tabel 1 Daftar Direktori dan Berkas

| Nama Direktori | Nama Berkas | Keterangan |
|-----------------------|-------------------------|---|
| Data | | Untuk menyimpan berkas |
| Script | | berisi berkas-berkas desain yang digunakan dalam membangun modul kompresi berkas berbasis web |
| Style | | untuk membuat tampilan modul kompresi berkas berbasis web menjadi lebih teratur dan bagus |
| | conf.php | Konfigurasi direktori penyimpanan modul kompresi berkas berbasis web |
| | functions.php | Membuat pohon direktori dan tipe berkas |
| | index.php | Tampilan halaman utama modul kompresi berkas berbasis web |
| | tool-file-operation.php | Menangani operasi kompres dan ekstrak berkas |
| | tool-load-dir.php | Mengambil daftar direktori |
| | tool-load-file.php | Mengambil daftar berkas |
| | tool-upload-file.php | Untuk upload berkas |

4.2 Implementasi Antarmuka

4.2.1 Halaman Utama

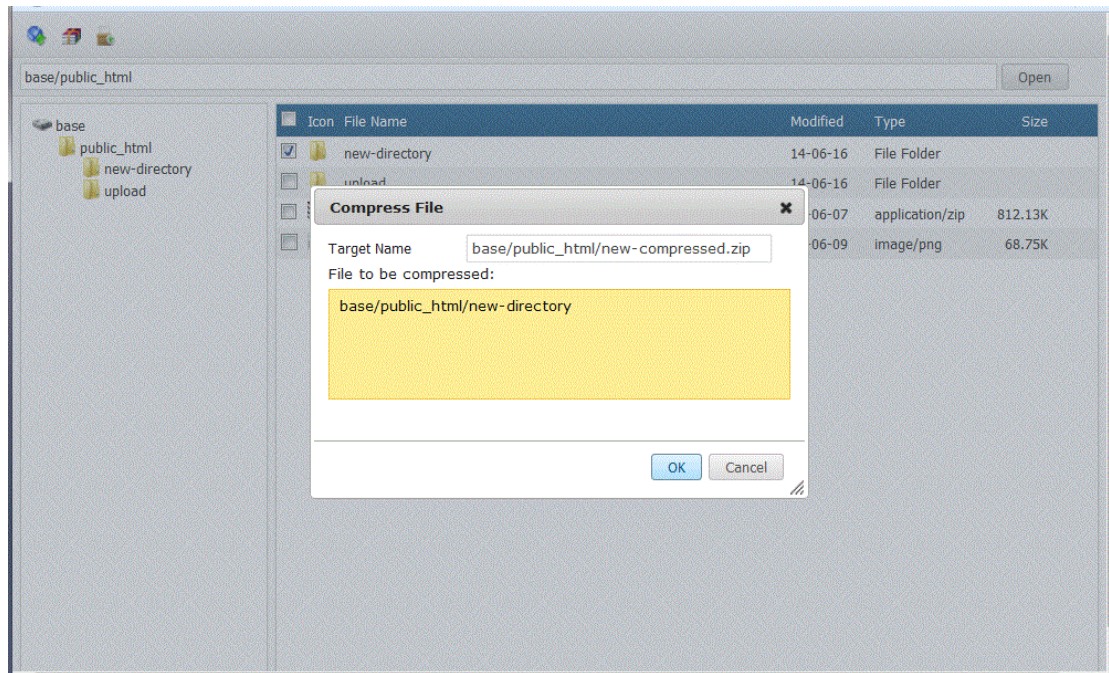
Antarmuka halaman utama merupakan halaman awal ketika pengguna membuka modul kompresi berkas berbasis web. Dalam antarmuka ini tersedia menu utama, daftar direktori dan daftar berkas.



Gambar 7 Antarmuka Halaman Utama

4.2.2 Kompres Berkas

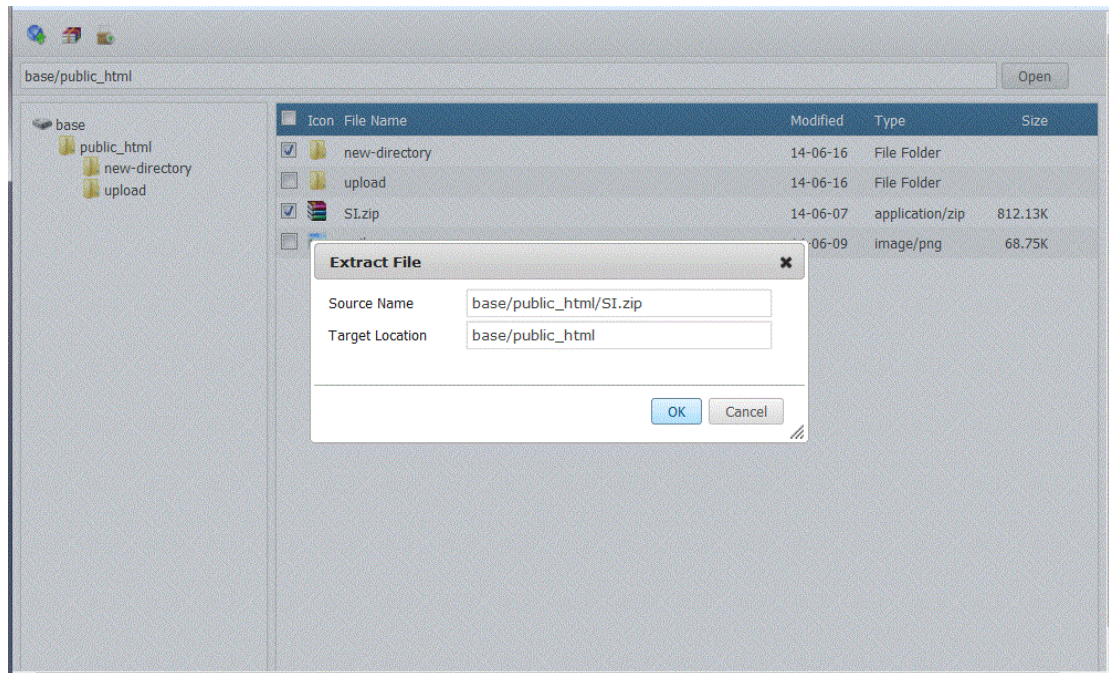
Antarmuka kompres berkas digunakan untuk melakukan kompres berkas pada server. Dalam Antarmuka ini dilengkapi dengan layanan untuk member nama berkas hasil kompresi dan nama direktori tujuan.



Gambar 8 Antarmuka Kompres Berkas

4.2.3 Ekstrak Berkas

Antarmuka ekstrak berkas digunakan untuk melakukan ekstrak berkas pada server. Dalam Antarmuka ini dilengkapi dengan layanan untuk menentukan direktori tujuan hasil ekstrak berkas.



Gambar 9 Antarmuka Ekstrak Berkas

4.3 Pengujian

4.3.1 Tujuan Pengujian

Pengujian adalah proses menjalankan dan mengevaluasi sebuah perangkat lunak secara manual maupun otomatis untuk menguji apakah perangkat lunak sudah memenuhi persyaratan atau belum. Dan untuk menentukan perbedaan antara hasil yang diharapkan dengan hasil sebenarnya.

4.3.2 Metode Pengujian

Pengujian modul kompresi berkas berbasis web ini menggunakan metode *Black Box*. Pengujian *Black Box* adalah pengujian terhadap fungsionalitas sistem tanpa memperhatikan struktur logika internal sistem. Pengujian *Black Box* juga merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada sistem dan kemudian keluaran dari sistem dicek apakah telah sesuai dengan yang diharapkan.

4.3.3 Rencana Pengujian

Pengujian yang akan dilakukan antara lain :

1. Pengujian frame, yaitu pengujian ini difokuskan pada suatu frame dari modul kompresi berkas berbasis web.
2. Pengujian modul kompresi berkas berbasis web menggunakan data uji berupa berkas.

4.3.4 Pembahasan dan Hasil Pengujian

Pengujian telah dilaksanakan dengan metode *Black Box* dan hasilnya dapat dilihat pada lampiran hasil pengujian. Berdasarkan dari hasil pengujian dapat diambil kesimpulan bahwa Modul Kompresi Berbasis Web ini secara fungsional telah memberikan hasil dari setiap proses sesuai yang diharapkan.

Bab V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis dan pengembangan modul kompresi berkas berbasis web maka dapat diambil kesimpulan bahwa:

1. Untuk menjalankan modul ini diperlukan aplikasi pendukung yaitu Xampp untuk Windows dan Lampp untuk Linux.
2. Modul ini dibangun bertujuan untuk membantu pengguna dalam melakukan manajemen berkas pada server. Khususnya dalam melakukan kompres dan ekstrak berkas pada server.

5.2 Saran

Modul ini masih perlu dikembangkan khususnya dalam hal kompres dan ekstrak berkas format tertentu. Misalnya format Rar, 7Zip, Gzip dan lain sebagainya. Guna memperbaiki kinerja modul kompresi berkas berbasis web ini.

DAFTAR PUSTAKA

1. Bowo, Eri, 2014, *Cpanel : Panduan untuk Web Master*, Jasakom, Jakarta
2. Stendy B. sakur, Penerbit Andi, PHP 5 Pemograman Berorientasi Object, 2010.
3. Dwi Nurhayati, ST, MT, Oky, *Kompresi Data*,
http://eprints.undip.ac.id/18380/1/Kompresi_Data_pert6.pdf? , diakses 23
Maret 2014
4. http://reza_chan.staff.gunadarma.ac.id/.../KONSEP+USER+INTERFACE.ppt
diakses 19 April 2014
5. <http://tiksmanka.webs.com/materi/Pengertian%20manajemen%20berkas.pdf>
diakses 27 April 2014
6. <http://www.kamshory.com/?page=kams-file-manager> 04 Mei 2014

LAMPIRAN

Tabel 2 Tabel Pengujian Menggunakan Metode *Black-Box*

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | Penguji |
|------------|------------------------------------|------------------|--|------------------------|----------------|
| 1. | Melakukan kompresi dengan data uji | berkas | Sistem akan melakukan kompresi berkas dan menampilkan hasil kompresi | Sesuai harapan | Yohan N |
| 2. | Melakukan ekstrak dengan data uji | Berkas arsip | Sistem akan melakukan kompresi berkas jika data uji berupa berkas zip dan menolak jika buka berkas arsip zip. Kemudian sistem menampilkan hasil ekstrak. | Sesuai harapan | Yohan N |