

APLIKASI SERVER WEB DENGAN KOMPRESI GZIP

Febryo Ponco Sulistyio

Binus University, School of Computer Science

*Kemanggisan, Jl. Kh. Syahdan No.9, RT.6/RW.12, Palmerah, Kota Jakarta Barat, Daerah Khusus Ibukota
Jakarta 11480*

Email : fsulistyio@binus.edu

ABSTRAK

Menciptakan suatu efisiensi dalam komunikasi dalam web maka ukuran file yang diminta harus dimampatkan oleh server sebelum mengirimkan kembali ke tanggapan ke klien karena semakin kecil ukuran akan ringan kinerja dari peramban selain dapat melakukan pemampatan server juga haruslah mampu melayani multi-klien. Pemampatan Gzip adalah pemampatan yang digunakan untuk server web Apajeh Ponco sebagai pemampatan sumber daya. Pemampatan Gzip dipilih dikarenakan hampir semua peramban web dapat menerima sumber daya yang dimampatkan dengan Gzip Agar server web Apajeh Ponco mampu menerima permintaan dan mengirimkan tanggapan dari dan ke peramban maka server web menggunakan komponen Windows Socket (Winsock). Dari hasil yang didapatkan aplikasi server web Apajeh Ponco ini mampu melayani multi-klien dan memampatkan sumberdaya yang diminta oleh klien tetapi belum mampu mendukung script PHP. Pemampatan Gzip lebih sesuai untuk terhadap berkas halaman web (.html dan .php), berkas dokumen (.docx) dan juga cocok untuk berkas-berkas executable hal ini ditunjukkan oleh rasio pemampatan dari berkas-berkas tersebut lebih tinggi daripada pemampatan untuk audio, video dan gambar.

Kata Kunci : Server Web, Gzip, Windows Socket (Winsock)

PENDAHULUAN

Website merupakan media penyajian informasi yang populer saat ini. Web menyajikan informasi menggunakan hypertext markup language (HTML) sehingga informasi dapat ditampilkan dengan berbagai format data seperti teks, hiperteks, dokumen format portabel (PDF) dan gambar bahkan audio/video, serta dapat diakses menggunakan berbagai jenis aplikasi klien.

Kepopuleran web tersebut tidak lepas dari peran server web sebagai aplikasi yang menyediakan layanan pengiriman data berbasis hypertext transfer protocol (HTTP). Server web yang dikenal juga sebagai server HTTP adalah komputer-komputer yang telah diatur dan dikonfigurasi untuk menyimpan sumber daya agar dapat diakses oleh klien yang ingin mengakses halaman web tersebut (menggunakan peramban) (Iman dan Mambrasar, 2007).

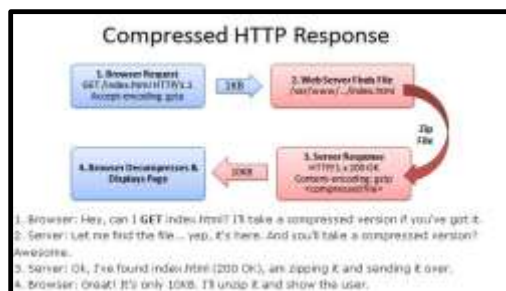
Server web dikembangkan mendukung berbagai fitur pelengkap seperti pengiriman data termampatkan, penggunaan kompresi (pemampatan) sangat penting dalam komunikasi data antara server web dengan klien sehingga dapat menghemat bandwidth dan mempercepat waktu unduh. Kecepatan transfer data di internet tergantung dengan bandwidth yang dimiliki semakin lebar bandwidth maka akan semakin cepat tentu semakin mahal harga yang harus dibayar, tentu harga menjadi hal yang sangat dipertimbangkan, tentu bandwidth haruslah efisien penggunaannya.

Kebanyakan server web menggunakan kompresi gzip karena dapat mengurangi ukuran sumber daya hingga 70% tanpa menghilangkan konten sehingga mengurangi traffic antara sever web dan peramban (Higgins dan Khaw, 2012).

Penulis menganalogikan mengapa perlunya kompresi seperti pada Gambar 1 dan Gambar 2.



Gambar 1 Respon HTTP Tanpa Dimampatkan



Gambar 2 Respon HTTP yang Dimampatkan

Maka dari itu, penulis membuat sebuah aplikasi server web yang mampu memampatkan respon dari server ke klien sehingga dapat meningkatkan efisiensi komunikasi data via web. Aplikasi ini dibangun dengan menggunakan Visual Basic 6.0 dikarenakan di Visual Basic 6.0 sudah ada komponen Winsock yang berfungsi sebagai socket yang menangani permintaan dari klien. Maksud dari pemampatan secara mutlak adalah setiap sumber daya yang diminta dengan menggunakan server web Apajeh Ponco akan langsung dimampatkan apapun format sumber dayanya baik format yang sudah maupun yang belum termampatkan.

STUDI LITERATUR

2.1. Komunikasi Data

Data di komputer memiliki ukuran dalam penyebutannya. Data terkecil di komputer disebut dengan bit. Bit merupakan singkatan dari Binary Digit. Kata Binary sendiri diambil dari nama Binary Number System (Sistem Bilangan Biner). Bit hampir selalu digunakan sebagai satuan terkecil dalam penyimpanan dan komunikasi informasi. Selain itu bit juga satuan ukuran, yaitu kapasitas informasi dari sebuah digit biner. Bit-bit tersebut selanjutnya dirangkai dan rangkaian tersebut diberi kode lagi yang disebut dengan character (Wahyudi, 2008).

Komunikasi data adalah merupakan bagian dari teknologi komunikasi yang secara khusus berkenaan dengan transmisi atau pemindahan data dan informasi di antara komputer dan piranti-piranti yang lain dalam bentuk digital yang dikirimkan melalui media komunikasi data. Data berarti informasi yang disajikan oleh kode digital. Komunikasi data merupakan bagian penting dari suatu sistem informasi karena merupakan pendukung penyediaan infrastruktur yang memungkinkan komputer-komputer dapat berkomunikasi satu sama lain (Suryadi, 2003).

2.2. Model Komunikasi Data

Sistem komunikasi jaringan komputer dapat dianggap identik dengan jaringan komunikasi data yang memungkinkan terjadinya pertukaran data, minimal antar dua entitas. Model komunikasi yang umum adalah seperti yang ditunjukkan oleh Gambar

- Sumber (Source), merupakan pihak yang menghasilkan data yang dikirim.
- Pengirim (Transmitter), yaitu alat yang mengubah data menjadi sinyal yang bisa dikirimkan.
- Sistem Pengiriman (Transmission System), yaitu tempat yang menyalurkan data.
- Penerima (Receiver), yaitu alat yang mengubah sinyal yang diterima menjadi data.
- Tujuan (Destination), merupakan pihak yang mengambil data yang masuk.

2.3. Server Web

Server web yang dikenal juga sebagai server HTTP adalah komputer-komputer yang telah diatur dan dikonfigurasi untuk menyimpan sumber daya agar dapat diakses oleh klien yang ingin mengakses halaman web tersebut (menggunakan peramban) (Iman dan Mambrasar, 2008).

2.3.1. Cara Kerja Server Web

Server web harus dapat mengenali tiap request yang datang, mengenali permintaan terhadap sebuah sumber daya, menguraikan, mengelolanya, dan mengirimkannya kembali ke pengguna (Newman, 2002). Server web memecah operasinya sehingga dapat melayani permintaan baru walaupun permintaan sebelumnya belum selesai diselesaikan. Model seperti ini merupakan model multi proses yang umumnya banyak digunakan pada server web (Iman dan Mambrasar, 2008).

1. Server Dijalankan

Ketika sebuah server web dijalankan, server web akan memeriksa konfigurasinya. Jika ditemukan kesalahan pada pemeriksaan konfigurasi server web, maka server web akan menghentikan proses tersebut sampai pada bagian kesalahan konfigurasi. Beberapa konfigurasi yang akan diperiksa oleh server web antara lain, alamat host, dan nomor port yang akan ditangani oleh server web tersebut, alamat document root, hak akses dan lain sebagainya.

2. Inisialisasi Modul

Setelah itu server web akan memeriksa konfigurasi setiap modul yang akan digunakan.

3. Inisialisasi Anak

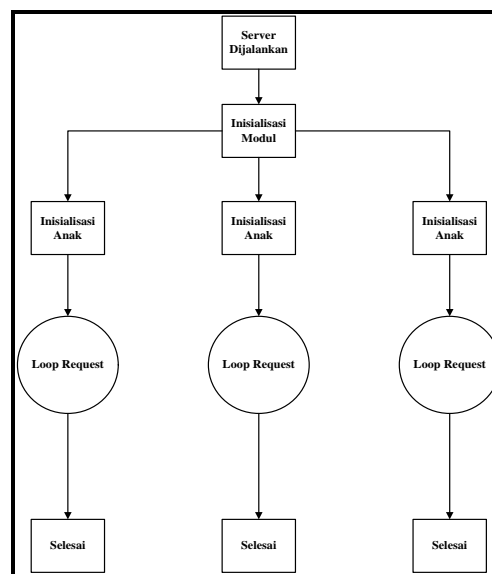
Server web akan menangani setiap permintaan sebagai anak permintaan yang di bawah kendali proses utama. Anak permintaan akan dibentuk apabila ada permintaan baru disinilah pemrograman multisoocket sangat penting. Pada proses ini server web akan membuat replika dari socket utama yang akan ditandai dengan indeks tertentu untuk menangani koneksi lanjutan server.

4. Loop Request

Proses ini merupakan proses penanganan terhadap permintaan klien yang akan dilakukan sampai pengguna selesai mengirimkan permintaan.

5. Selesai

Setelah loop request selesai, maka proses anak yang menanganinya akan dimatikan oleh proses utama dan akan dipanggil kembali apabila anak proses lainnya.



Gambar 3. Model Multi Proses Server Web

Server web menunggu permintaan dari klien yang menggunakan peramban seperti Internet Explorer, Mozilla Firefox, Opera dan peramban lainnya. Jika ada permintaan dari browser, maka server web akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke browser. Server web untuk berkomunikasi dengan kliennya (peramban) mempunyai protokol sendiri yaitu Hypertext Transfer Protocol (HTTP) Dengan protokol ini, komunikasi antara server web dengan kliennya dapat saling dimengerti.

2.4. Pemrograman Soket

Windows menangani soket untuk protokol TCP/IP melalui spesifikasi Winsock API (Dumas,1995). Winsock API adalah sekumpulan fungsi yang mengimplementasikan antarmuka soket yang telah dipopulerkan oleh Berkeley Software Distribution of UNIX. Winsock menambahkan ekstensi khusus Windows, pada implementasi Socket Berkeley, untuk mendukung sistem operasi Windows yang menggunakan arsitektur terkendali-pesan (Message-Driven). Pemrograman Winsock dapat dilakukan dengan beberapa bahasa pemrograman berbasis Windows, antara lain Visual Basic, Delphi, dan Visual C++.

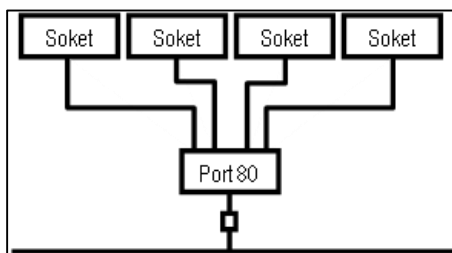
Terdapat dua cara pendekatan yang dapat diambil ketika melakukan pemrograman Windows Socket, yaitu mengkode secara langsung pada API, atau menggunakan sebuah custom control (VBX atau OCX) yang menyediakan sebuah antarmuka pada library dengan pengesetan property dan menanggapi event (Anindito, 2000).

Sebuah custom control menyediakan antarmuka pemrograman yang lebih natural, dan memungkinkan untuk menghindari banyak kecenderungan kesalahan umum dalam pemrograman socket. Dengan memberi kontrol pada sebuah form, mengeset beberapa property dan menanggapi event, pemrogram dapat secara cepat dan mudah menulis sebuah aplikasi internet. Ada banyak custom control yang beredar, antara lain SocketWrench/VB dari Catalyst Software dan Microsoft Winsock Control. (Anindito, 2000) .

2.3.2 Windows Socket

Hubungan antara TCP dan proses lokal adalah port, yaitu suatu mekanisme yang memberikan kemampuan pada proses untuk mengirimkan aliran data ke proses yang tepat (Heywood,1996).

Komunikasi antar komputer melalui jaringan menggunakan TCP/IP mengacu pada hubungan dua komputer. Hubungan ini, jika didasarkan pada protokol TCP, mempunyai implikasi ketika banyak koneksi ke pelayanan komputer yang sama diperlukan. Hubungan tersebut harus mampu menangani banyak hubungan. Inilah yang menimbulkan konsep dari socket. Ketika terjadi sebuah hubungan, komputer menerima hubungan pada port dan menandai hubungan dengan sebuah nomor socket khusus. Dengan demikian, hubungan lain dapat meneruskan untuk menghubungi port yang sama, namun akan ditandai dengan nomor socket yang berbeda. Setiap protokol menggunakan sebuah port TCP yang unik bagi pelayanannya. Dengan demikian, banyak pelayanan yang dapat berjalan pada sebuah komputer karena mendengarkan suatu hubungan pada port yang berbeda (Purbo, 2005).



Gambar 4. Multisocket

Windows Socket memiliki tiga tipe socket, yaitu stream socket, datagram socket, dan raw socket. Socket dapat diatur untuk menyediakan sebuah pelayanan yang handal (reliable), pelayanan stream berorientasi hubungan (connection oriented stream service); pelayanan yang tidak handal, pelayanan

tanpa hubungan (connectionless datagram service); atau raw socket yang biasanya digunakan untuk mengirim pesan ICMP lewat jaringan.

2.4 Pemampatan Data

Kompresi data adalah proses yang mengkonversi sebuah masukan berupa aliran data (the source atau data asli mentah) menjadi suatu aliran data lain (the output, aliran bit, atau aliran sudah dikompres) yang memiliki ukuran lebih kecil. Aliran data (stream) dapat berupa sebuah file atau buffer pada memori. Data dalam konteks kompresi data melingkupi segala bentuk digital dari informasi, yang dapat diproses oleh sebuah program komputer. Bentuk dari informasi tersebut secara luas dapat diklasifikasikan sebagai teks, suara, gambar dan video.(Salomon, 2007).

Konsep utama dalam kompresi terletak pada eliminasi redundansi. Hal ini secara tidak sadar dimanfaatkan dalam kehidupan sehari-hari, contohnya adalah penggunaan singkatan dalam tulisan. Kita sudah tidak asing dengan penggunaan “&” sebagai pengganti kata “dan”, “yg” sebagai pengganti kata “yang” dan “cth” sebagai pengganti kata “contoh”. Permasalahan dalam kompresi data adalah bagaimana menemukan metode yang efisien untuk menghilangkan redundansi dari berbagai tipe data serta metode untuk membentuk kembali wujud semula.(Pu, 2006).

Teknik Kompresi Data dapat dibagi menjadi dua kategori besar, yaitu (Widihiarta, 2008):

1. Lossy Compression

Lossy compression menyebabkan adanya perubahan data dibandingkan sebelum dilakukan proses kompresi. Sebagai gantinya lossy compression memberikan derajat kompresi lebih tinggi. Tipe ini cocok untuk kompresi file suara digital dan gambar digital. File suara dan gambar secara alamiah masih bisa digunakan walaupun tidak berada pada kondisi yang sama sebelum dilakukan kompresi.

2. Lossless Compression

berbasis dictionary adalah algoritma varian Lempel-Ziv (LZ), Deflate dan lain-lain.

2.5 Gzip

Pemampatan Gzip bekerja dengan mencari string yang sama dalam sebuah file teks, dan menggantikan mereka string sementara untuk membuat ukuran file yang lebih kecil secara keseluruhan. Ini bentuk kompresi sangat cocok untuk web karena HTML dan CSS file biasanya mengandung banyak string

berulang, seperti spasi, tag, dan definisi gaya lainnya (Google Developer).

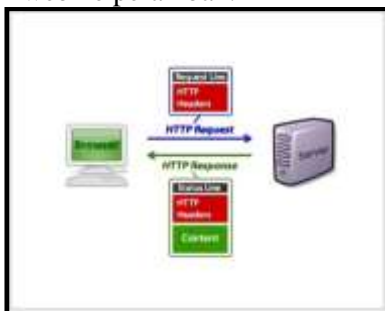
Algoritma kompresi yang bersifat lossless seperti GZIP tidak membuang sedikitpun informasi yang dimiliki oleh berkas asal. Algoritma kompres yang bersifat lossy umumnya digunakan untuk memampatkan berkas-berkas gambar, video ataupun suara, hal ini menimbang perubahan (penghilangan) sedikit pada berkas asal tidak akan menimbulkan efek yang mampu ditangkap oleh indra manusia. Sedangkan algoritma kompresi yang bersifat lossless umumnya digunakan untuk berkas teks atau binary (executable). Hal ini mengingat perubahan yang kecil pada berkas yang dikompresi akan memberi pengaruh besar pada berkas hasil kompresi saat didekompresi ulang. Misalnya pada suatu berkas program computer (source code), perubahan yang terjadi walaupun sedikit akan berakibat pada kesalahan kode program tersebut saat di kompilasi setelah di dekompresi (Salomon, 2007).

METHODOLOGI

3.1. Analisis Masalah

Pada umumnya orang menginginkan informasi yang diinginkan didapatkan dengan cepat dan tepat apalagi begitupula dalam penyebaran informasi di internet seperti yang sudah dijelaskan pada latar belakang mengenai pentingnya pemampatan dan mengapa sumber daya harus dimampatkan, pada bab ini penulis menguraikan lebih detail mengenai hal-hal tersebut dan bagaimana perancangannya.

Komunikasi antara server web dan peramban dilakukan dengan saling membaca header (header permintaan dan header tanggapan), header permintaan adalah header permintaan yang dikirimkan peramban ke server web sedangkan header tanggapan adalah header yang dikirimkan oleh server web ke peramban.



Gambar 5. Header Permintaan dan Tanggapan

3.2. Analisis Kebutuhan

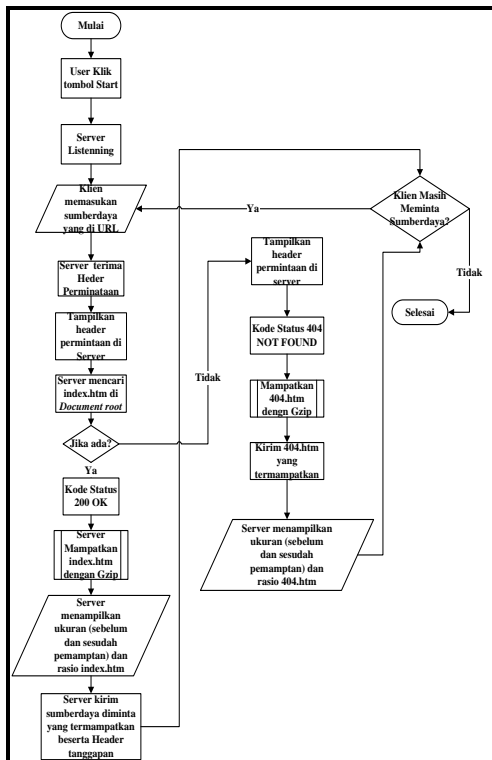
Untuk mengetahui keberhasilan dari aplikasi server web yang penulis rancang bangun maka output yang akan ditampilkan adalah ukuran data, penghitungan rasio dan waktu kompresi sebelum dan sesudah pemampatan. Ukuran data yang akan ditampilkan dalam aplikasi ini adalah ukuran data sebelum dan sesudah dimampatkan dan akan dilihat validasi keberhasilan pemampatannya menggunakan sebuah kakas Fiddler2 yang nantinya berfungsi untuk membuktikan apakah Apajeh Ponco dapat melakukan pemampatan dan ukuran sebelum dan sesudah pemampatannya sama dengan ukuran yang muncul sebelum dan sesudah pemampatan pada Fiddler2.

Setelah mendapatkan ukuran sebelum dan sesudah dimampatkan maka penulis menghitung dan menampilkan rasio pada Apajeh Ponco sehingga dapat mengetahui efisiensi dari pemampatan yang digunakan oleh server web semakin besar nilai presentase rasio maka semakin efisien penggunaan pemampatan terhadap suatu data. Rumus menghitung rasio pemampatan menurut David Salomon adalah :

$$\text{Rasio} = \frac{\text{ukuran sebelum dimampatkan} - \text{ukuran setelah dimampatkan}}{\text{ukuran sebelum dimampatkan}} * 100\%$$

3.3 Cara Kerja Sistem User Meminta Sumber daya (Dilihat dari Sisi Server)

Keluarannya adalah yang tampil di aplikasi Apajeh Ponco yang berupa ukuran sumber daya yang diminta sebelum dan sesudah serta rasionya, sehingga dapat diketahui bahwa hasilnya tersebut nantinya akan dicocokkan dengan kakas Fiddler2 apakah ukurannya sesuai dengan kakas tersebut baik yang sebelum (*no compression*) dan sesudah dimampatkan dengan Gzip (*Gzip Compression*).

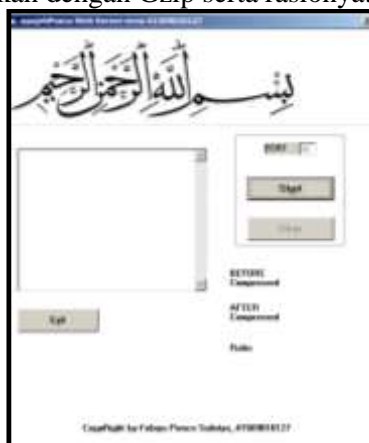


Gambar 6. Diagram Alir Cara Kerja Sistem User Meminta Sumber daya (Dilihat dari Sisi Server)

HASIL DAN DISKUSI

4.1 Implementasi Program dan Antarmuka

Pada Aplikasi ini dapat menampilkan header permintaan dari peramban dan menampilkan ra ukuran sumber daya sebelum dan sesudah dimampatkan dengan Gzip serta rasionya.



Gambar 7. Tampilan Antar Muka Pengguna Aplikasi Server Web Apajeh Ponco

4.2 Hasil Pengujian

Dari hasil yang didapatkan dari pengujian, maka penulis menganalisis kinerja dari server web Apajeh Ponco sebagai berikut :

Tabel 1 Frekuensi Distribusi

No	FileType	Before Compression (Kb)	After Compression (Kb)	%	Category
1.	Webpage	454	316	30,40	Very Good
2.	Video	14.703.324	13.269.192	9,75	Bad
3.	Audio	92.244	49.371	46,48	Very Good
4.	Image				
5.	Binary	483.328	264.407	45,29	Very Good
5.	Dokumen	13.333	10.530	21,02	Average

1. Tombol-tombol pada Aplikasi server web Apajeh Ponco berjalan dengan baik sesuai dengan perancangan. Dikarenakan setiap pengujian dengan kejadian mengklik tombol semua hasil yang diharapkan berjalan sukses.
2. Aplikasi server web Apajeh Ponco ini melayani lebih dari satu klien pada saat bersamaan. Dikarenakan pada saat pengujian pada localhost dengan peramban Mozilla Firefox dan Internet Explorer tidak terjadi galat apapun dan sumber daya yang diminta dapat ditampikan pada peramban atau diunduh oleh aplikasi pengunduh dari peramban atau aplikasi pengunduh lainnya.
3. Aplikasi server web Apajeh Ponco ini mampu memampatkan setiap sumber daya yang diminta oleh klien apapun ekstensinya dan ukuran pemampatannya sesuai dengan apa yang ada pada kakas Fiddler2. Hal ini membuktikan bahwa pemampatan Gzip pada Server web Apajeh Ponco memang benar sesuai dengan yang diharapkan.
4. Server web Apajeh Ponco mengirimkan status dan header tanggapan kepada peramban. Hasil pemampatan setiap permintaan dari setiap pengujian tidak terdapat perbedaan. Hal ini menunjukkan bahwa pemampatan tidak terpengaruh oleh tipe pengujian baik secara topologi jaringan secara *localhost*, *ad-hoc* maupun infrastruktur.
5. Seperti yang disebutkan sebelumnya bahwa cara kerja Gzip yaitu “pemampatan gzip bekerja dengan mencari *string* yang sama dalam sebuah file teks, dan mengganti mereka

string sementara untuk membuat ukuran file yang lebih kecil secara keseluruhan. Bentuk kompresi sangat cocok untuk web karena HTML dan CSS file biasanya mengandung banyak *string* berulang, seperti spasi, tag, dan definisi gaya (Google Developer).” Dari hasil pengujian pemampatan Gzip bekerja lebih baik untuk halaman web, berkas berjenis *executable*, dan juga untuk berkas dokumen. Hal tersebut ditunjukkan dengan nilai rasio yang tinggi daripada berkas lainnya. Ini dikarenakan didalam halaman web, program *executable* yang notabene dirancang dengan sebuah bahasa pemrograman tertentu yang pastinya berisi listing program dimana akan ada *string* yang berulang serta dalam dokumen kata pastilah banyak *string* yang berulang.

6. Setiap file yang sudah termampatkan apabila dimampatkan kembali dengan server web apajeh ponco dikarenakan file tersebut sudah terkompresi dengan pemampatan *lossy* maka apabila dimampatkan kembali dengan gzip yang notabene adalah pemampatan bertipe *loseless* maka rasionya menjadi rendah.
7. Apabila tipe file tersebut belum termampatkan, maka apabila dimampatkan dengan server web Apajeh Ponco maka rasionya menjadi tinggi.
8. Untuk berkas berjenis gambar dengan format jpg rasionya lebih kecil dikarenakan format tersebut sudah dimampatkan sehingga hasil rasionya kecil saat dimampatkan dengan Gzip.
9. Untuk berkas seperti dokumen, audio dan video dapat diputar langsung oleh peramban tergantung *plug-in* nya pada PC yang dijadikan server dapat diputar langsung karena pluginnya sudah terpasang sedangkan pada pengujian *ad-hoc* PC klien tidak memiliki *plug-in* sehingga peramban akan melempar kepada sistem operasi untuk mengunduh dan membuka berkas tersebut dengan aplikasi yang sesuai kecepatan muatnya tergantung spesifikasi dari PC yang digunakan.
10. Server web Apajeh Ponco belum mampu menangani script php walaupuns sudah bisa memuat file berformat php tetapi yang dimuat hanya tag-tag HTML-nya bukan script PHP-nya maka dari itu penulis membuat dua buah pengujian pada PHP satu untuk biodata.php, berkas berformat PHP namun tidak memiliki script PHP dan juga faktorial.php
11. Untuk pengujian pada Android, semua berkas dokumen, audio, video, dan paket android (.apk) dapat diunduh namun setelah diunduh

tidak dapat dibuka ini dikarenakan pemampatan Gzip pada Apajeh Ponco ini tidak dapat dikenali oleh aplikasi lain pada Android. Sehingga server web Apajeh Ponco ini hanya bisa bekerja dengan baik untuk klien sesama Windows.

KESIMPULAN

Setelah didapatkan hasil pengujian dan analisis hasil pengujiannya maka penulis membuat beberapa kesimpulan terkait aplikasi server web Apajeh Ponco sebagai berikut :

1. Server web Apajeh Ponco ini mampu melayani multi-klien dan mampu memampatkan setiap sumber daya yang diminta dari klien dengan pemampatan Gzip.
2. Terlihat dari rasio yang rendah dari hasil pengujian, pemampatan Gzip memang terbukti cocok untuk halaman web dilihat dari hasil pengujian, berkas dengan format html dan php rasionya ternyata memang tinggi dan begitu pula untuk berkas dokumen kata dan program *executable*.
3. Terlihat pula dari rasio pemampatan setiap file, pemampatan pada server web Apajeh yang menggunakan Gzip mampu memampatkan dengan baik apabila file yang diminta bukanlah file yang termampatkan.
4. Hasil pemampatan dari server web Apajeh Ponco ini belum mampu diterapkan dengan baik kepada perangkat telepon seluler dengan sistem operasi Android. Berkas yang dimampatkan oleh Gzip tidak dapat dibuka oleh aplikasi-aplikasi yang berjalan pada sistem operasi Android.

Setelah memberikan kesimpulan dari aplikasi server web Apajeh Ponco, penulis juga memberikan saran untuk pengembangan aplikasi server web Apajeh Ponco selanjutnya.

1. Server web Apajeh Ponco belum mampu menangani *script* PHP untuk pengembangan selanjutnya aplikasi ini dapat membuat sebuah server web yang mampu melakukan pemampatan dan juga mampu menangani *script* PHP.
2. Server web Apajeh Ponco ini juga belum ada keamanan jaringan, diharapkan kedepannya akan ada yang membahas masalah keamanan dari server web Apajeh Ponco.
3. Dikarenakan belum terdapatnya manajemen pengguna dalam server web Apajeh Ponco, kedepannya diharapkan akan ada

- pembahasan mengenai manajemen pengguna sehingga ada batasan antara pengguna biasa dengan administrator server web Apajeh Ponco ini.
4. Aplikasi server web Apajeh Ponco ini dirancang-bangun dengan menggunakan Visual Basic 6.0, peneliti berikutnya bisa menggunakan bahasa pemrograman yang terbaru seperti Java, Visual Basic.Net, Delphi dan lain sebagainya.
 5. Diharapkan dapat dikembangkan mampu melayani multi sistem operasi dan dapat bekerja dengan baik diseleruh klien apapun sistem operasinya.
 6. Server web Apajeh Ponco ini belum mampu membedakan tipe data diharapkan nantinya ada pengembangan dari aplikasi ini yang mampu menggolongkan apabila file dalam format yang sudah termampatkan tidak perlu dimampatkan sehingga rasionya untuk setiap file yang diminta mejadi tetap tinggi.
- Purbo, Onno W. 2010. Buku Pegangan Internet Wireless dan Hotspot. Jakarta : PT. Elex Media Komputindo
- Rahmat, Hapid, Damiri, Dhani Johar, Susanto, Ate. 2012. *Implementasi Kompresi Data Pada Jaringan Komputer Menggunakan Algoritma ZLIB*. Jurnal Algoritma STT Garut.
- Salomon, David, 2007. *Data Compression* . Computer Science Department, California State University
- Sihombing, Wira Nov Kurnia. 2010. *Implementasi Algoritma Kompresi Deflate Pada Data*. Skripsi Jurusan Ilmu Komputer Universitas Sumatera Utara.
- Sommerville, Ian. 2004. *Software Engineering*. Jakarta : Erlangga

DAFTAR PUSTAKA

- Aswan. 2012. *Kumpulan Program Kreatif Visual Basic*. Bandung : Penerbit Informatika.
- Higgins and Khaw. 2011. *How Gzip Works*. Google Developers
- GZIP, <http://www.GZIP.org>, diakses terakhir tanggal 22 September 2006.
- Iman, Erik dan Mambarasar, Yusuf. 2008. *Membuat Aplikasi Web Server Dengan Winsock*. Yogyakarta : Andi
- Moroney Laurence et.all .2006. *Pro ASP .NET 2.0 in VB 2005*, New York : Apress
- Mengyi Pu, Ida. 2006. "Fundamental Data Compression", Linacre Ho.Oxford.
- Sutanta, Edhy. 2005. *Pengantar Teknologi Informasi*. Yogyakarta : Graha Ilmu.
- Wicaksono, Soetam Rizky (2006). *Kompresi Dokumen On-The-Fly Dalam Aplikasi Berbasis Web Menggunakan Kelas Gzip*. Seminar Nasional Sistem dan Informatika 2006. Bali.
- www.planetsourcecode.com/vb/scripts/ShowCode.asp?txtCodeId=64920&lngWId=1. (Diakses pada tanggal 25 Januari 2013).
- Zam, Efvy. 2011. *Buku Sakti Hacker*. Jakarta : Mediakita.