BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Setelah tahap analisis dan perancangan sistem, tahap selanjutnya adalah tahap implementasi. Tujuan dari tahap implementasi ini adalah untuk memastikan perangkat lunak yang dibuat dapat berjalan secara sesuai dengan yang diinginkan. Setelah implementasi maka dilakukan pengujian sistem, dimana akan terlihat kekurangan-kekurangan pada perangkat lunak untuk selanjutnya diadakan pengembangan sistem. Berikut ini adalah implementasi sistem untuk pemampatan data teks dan *citra digital* dengan metode *Lemple Ziv Storer Symanski* dan *Rice Coding*.

4.2 Spesifikasi Perangkat Keras

Dalam pengujian program, maka diperlukan perangkat pendukung implementasi yang didukung oleh spesifikasi *hardware* dan *software*.

4.2.1 Spesifikasi perangkat keras

Spesifikasi terendah dari perangkat keras yang dapat digunakan untuk mengimplementasikan dan pengujian sistem ini adalah sebagai berikut:

a. *Processor* : *Intel Pentium* IV 1.3 *GHz*

b. *Memory* : 128 MB DDR

c. *Harddisk* : 80 GB

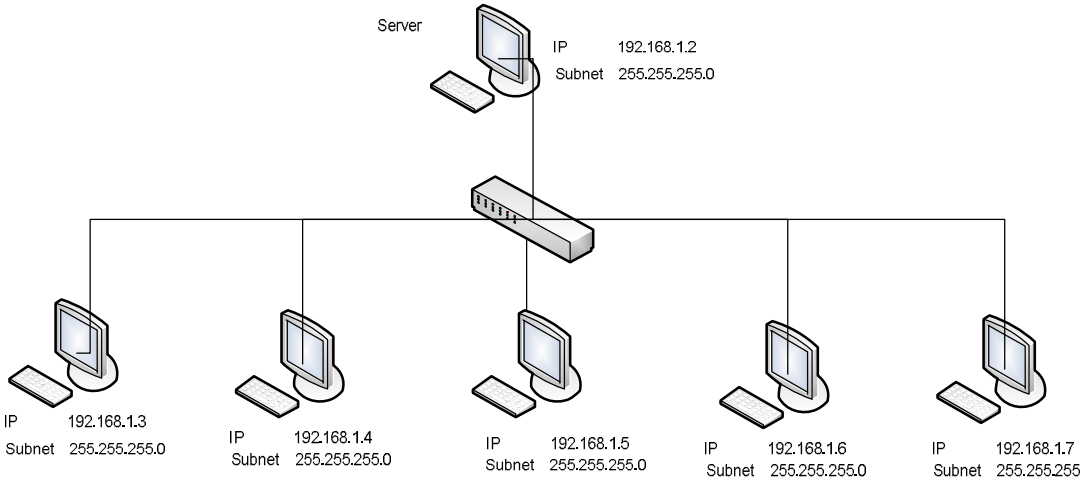
d. *VGA on Board* : 32 MB

4.2.2 Spesifikasi perangkat lunak

Perangkat lunak untuk pemampatan data menggunakan metode *Lemple Ziv Storer Symanski* dan *Rice Coding*. ini dapat di implementasikan menggunakan bahasa pemograman *Delphi* 7.0. Dan Sistem Operasi yang digunakan *Windows XP Professional*.

4.3 Topologi Jaringan

Topologi jaringan yang digunakan uji coba aplikasi ini adalah topologi *Star*. Pemilihan topologi ini karena dalam jaringan dihubungkan ke sebuah konsentrator dengan menggunakan jalur yang berbeda-beda, sehingga jika salah satu komputer mengalami gangguan, jaringan tidak akan terpengaruh. Komunikasi di dalam jaringan diatur oleh *konsentrator*, berupa *hub* maupun *switch*. Skema jaringan dapat dilihat pada gambar 4.1.



Gambar 4.1 Skema jaringan *Client-Server* yang digunakan pada uji coba aplikasi

Pada uji coba aplikasi, *range IP address* yang digunakan pada jaringan adalah 192.168.1.2 sampai 192.168.1.7 dengan *Subnet Mask* 255.255.255.0. Jumlah komputer yang digunakan adalah 6 unit, dimana 1 unit komputer berfungsi sebagai *server* yang berfungsi untuk mendekompresi *File* dan mengirim file hasil *dekompresi*. Sedangkan 5 unit komputer lainnya berfungsi sebagai *client* yang dapat melakukan proses kompresi.

4.4 Simulasi Program

Pada bagian ini kita akan membandingkan kedua algoritma *Lemple Ziv Storer Symanski* dan algoritma *Rice Coding.*. Media yang digunakan sebagai pembanding yaitu dengan menggunakan *File* teks dan *citra digital.* Tetapi tidak semua *File* dapat dimampatkan, karena pada batasan masalah telah disebutkan sebelumnya bahwa *File* yang akan dimampatkan hanya yang berekstensi untuk *File* teks: *.doc, .docx . pdf, .rtf* . Sedangkan *File citra* : *.bmp, .jpg*.

Setelah dilakukan proses kompresi, kita bisa membandingkan hasil akhirnya yaitu dengan melihat dari segi kapasitasnya, Rasio yang diberikan, Persentasi pemampatannya serta waktu yang digunakan untuk melakukan satu kali pengiriman kompresi *File*. Karena ini menggunakan algoritma *Losseless* maka data hasil kompresinya tidak bisa langsung kita lihat karena algoritma ini mirip dengan metode *WinZip*. Untuk melihat kembali data yang dikompres, kita bisa melakukan proses *dekompres.*

4.5 Penggunaan Program Kompresi

Dalam pengujian, langkah-langkah yang akan dilakukan untuk membandingkan kedua algoritma, yaitu:

1. Aplikasi yang digunakan *server*

    1. Jalankan *Program_Kompresi_Citra.exe.* Maka akan tampil:



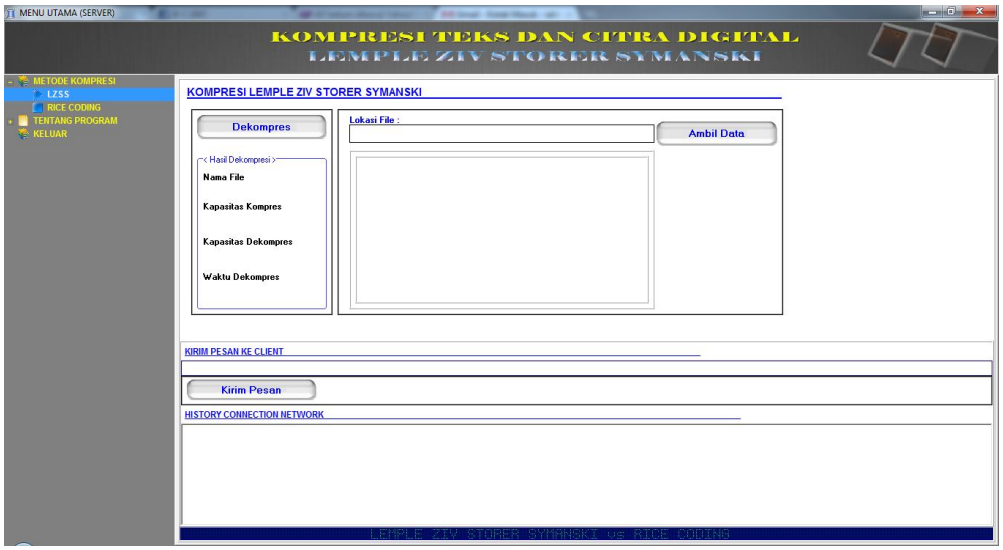
Gambar 4.2 Tampilan *Loading* aplikasi

2. Setelah tampilan *splash* maka akan masuk ke menu utama *Server*. Di dalam menu *server* terdapat di sebelah kiri yaitu Menu: Metode, Tentang Program, dan Keluar.



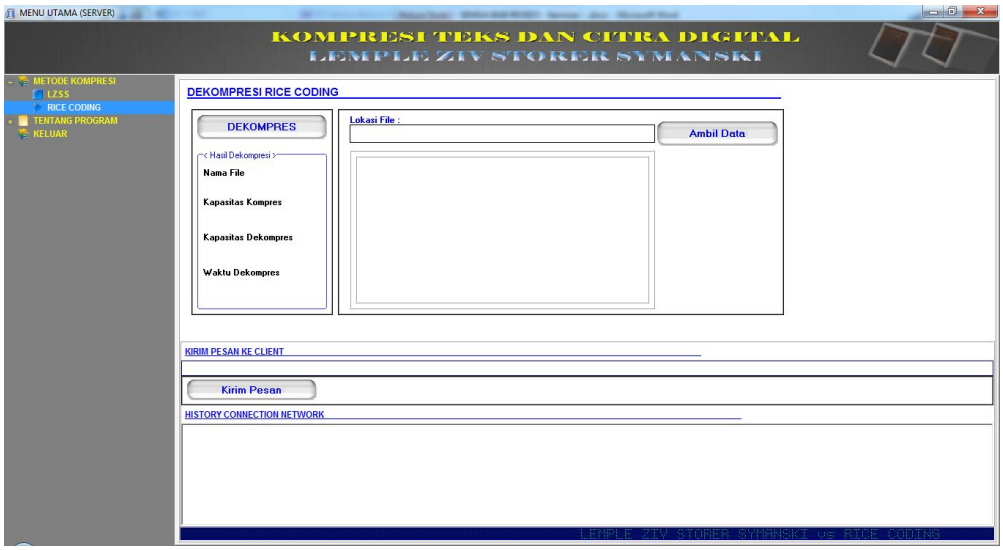
Gambar 4.3 Tampilan Menu Utama aplikasi

3. Setelah mengatur IP maka untuk masuk menu Metode LZSS akan tampil seperti dibawah ini. Menu metode LZSS yang ada di *server* berfungsi untuk memampatkan *File* dan mengirim *File* hasil pemampatan yang akan dikirm melalui jaringan ke *client.*



Gambar 4.4 Tampilan Menu metode LZSS (Dekompresi)

4. Kemudian jika ingin masuk ke menu Metode *Rice Coding*. Maka akan tampil seperti dibawah ini. Menu metode *Rice Coding* pada *server* berfungsi hanya untuk proses pemampatan dan mengirim *File* hasil pemampatan melalui jaringan ke *client*.



Gambar 4.5 Tampilan Menu metode *Rice Coding* (Dekompresi)

5. Menu selanjutnya yaitu menu bantuan yang berfungsi untuk membimbing *user* dalam menggunakan program aplikasi ini.

2. Aplikasi yang digunakan *Client*

1. Jalankan *Program_Kompresi_Citra.exe*. Maka akan tampil:



Gambar 4.6 Tampilan *Loading* aplikasi *Client*

2. Setelah muncul tampilan awal *Splash*, maka akan masuk ke tampilan Menu utama Pada *Client*.



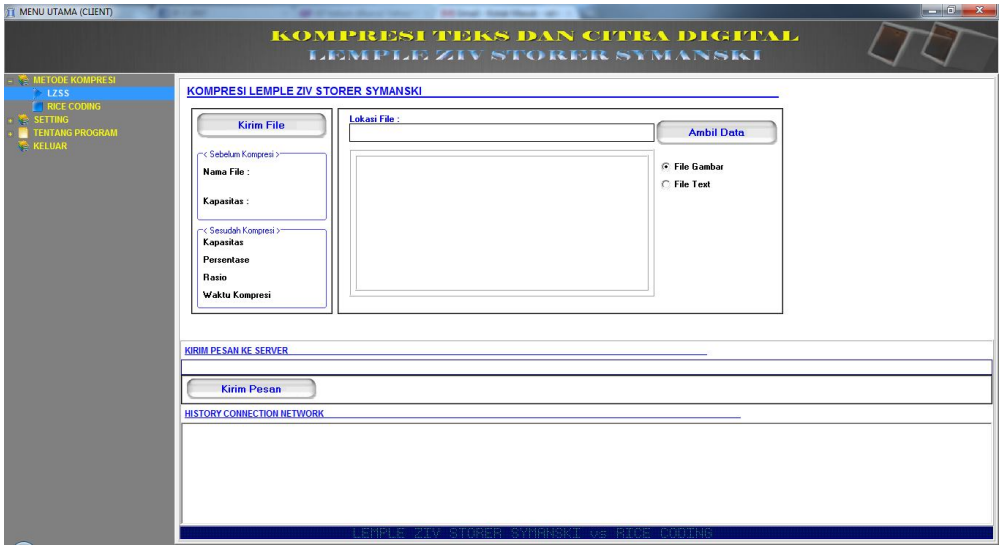
Gambar 4.7 Tampilan Menu utama *Client*

3. Sebelum Menerima data yang telah dikirimkan dari *server* maka terlebih dahulu *setting Ip* ke *server* untuk mengkoneksikan jaringan yang terhubung antara *client* maupun *server*.



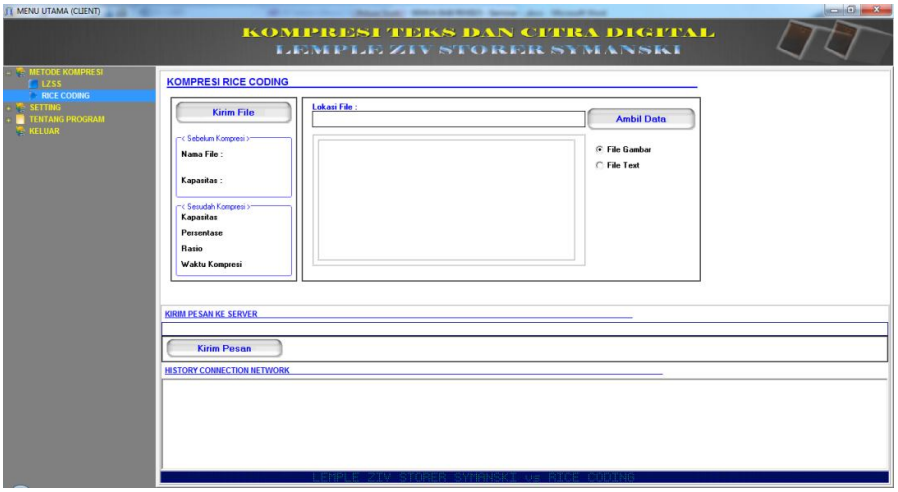
Gambar 4.8 Tampilan *Setting* IP ke *Server*

4. Menu Metode LZSS yang ada di *client* berfungsi untuk mendekompresi *File* hasil pemampatan yang telah dilakukan oleh *server* melalui jaringan.



Gambar 4.9 Tampilan Menu metode LZSS (Kompresi)

5. Setelah itu jika ingin masuk ke menu Metode *Rice Coding* pada *client*. Maka akan tampil seperti dibawah ini. Menu metode *Rice Coding* berfungsi hanya untuk mendekompresi *File* hasil kiriman dari *server* melalui jaringan yang terkoneksi.



Gambar 4.10 Tampilan Menu metode *Rice Coding* (Kompresi)

6. Menu selanjutnya yaitu menu bantuan yang berfungsi untuk membimbing user dalam menggunakan program aplikasi ini.



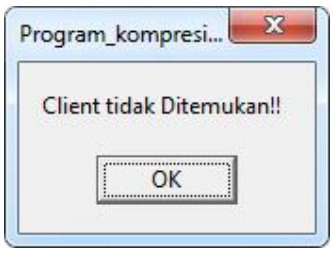
Gambar 4.11 Menu Bantuan

7. Pesan ini akan tampil apabila anda belum mengisikan alamat IP *address*



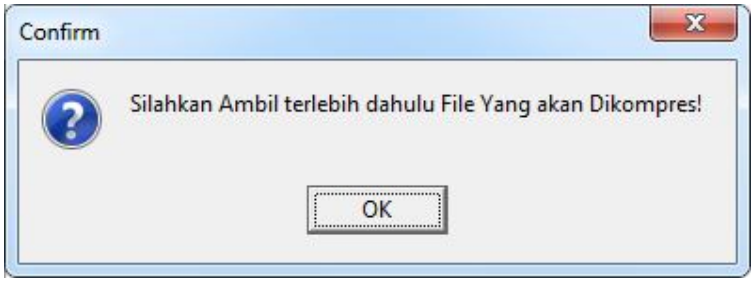
Gambar 4.12 Pesan Belum mengisi alamat IP Address

8. Pesan di bawah ini akan tampil pada saat mengirimkan *File* dan *client* belum terkoneksi ke *server*.



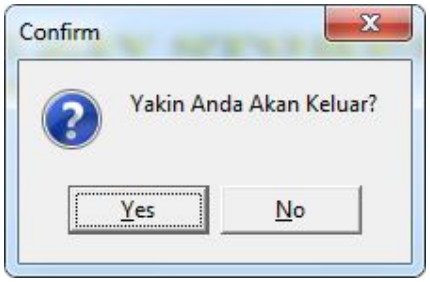
Gambar 4.13 Pesan *error File* akan dikirim *client* belum terkoneksi

9. Jika *File* akan dikompres masih kosong dan belum di inputkan maka akan tampil pesan dibawah ini.



Gambar 4.14 Pesan *File* belum di inputkan

10. Jika ingin keluar dari aplikasi maka akan tampil pesan berikut:



Gambar 4.15 Pesan keluar

4.6 Pengujian Data

Metode pengujian menggunakan metode *black box,* karena metode ini merupakan metode pendekatan terhadap pengguna serta pengujiannya diturunkan dari spesifikasi program atau komponen. Pengujian yang akan dilakukan hanya sebatas pada proses kompresi data teks dan citra *digital* , kemudian proses dekompres serta melakukan perbandingan hasil kompresinya.

4.6.1 Pengujian Kompresi dengan Metode *Black Box*

Tabel 4.1 Pengujian Kompresi dengan algoritma LZSS dan *Rice Coding*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Gambar Asli	*File* terkompresi dengan baik dengan hasil ukuran *File* terkompresi lebih kecil dibanding ukuran *File* asli.	Untuk ukuran *File* hasil kompresi, tergantung pada tipe *File*. Ukuran *File* hasil kompresi lebih kecil dibandingkan dengan ukuran *File* asli, sesuai yang diharapkan.	Diterima
Waktu	Waktu kompresi tidak terlalu lama	Waktu kompresi tergantung dari ukuran gambar, apabila ukuran gambar kecil maka waktu yang dibutuhkan tidak terlalu lama dibandingkan dengan gambar yang	Diterima

		berukuran besar, sesuai yang diharapkan.	
Rasio	Rasio kompresi besar	Apabila rasio kompresi lebih besar maka ukuran *File* hasil kompresi lebih kecil, sesuai dengan yang diharapkan	Diterima

Tabel 4.2 Pengujian Kompresi Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
File input : kosong	Akan ditampilakan pesan kesalahan pada kompresi.	Ditampilkan pesan kesalahan kompresi.	Diterima

4.6.2 Pengujian Dekompresi dengan Metode *Black Box*

Tabel 4.3 Pengujian Dekompresi dengan algoritma LZSS dan *Rice Coding*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Gambar hasil kompresi	*File* yang telah terkompresi dapat dikembalikan seperti semula sehingga gambar dapat dilihat kembali dengan kapasitas yang sama	*File* terkompresi dapat didekompres dengan baik sehingga gambar dapat dilihat kembali dengan kapasitas seperti semula.	Diterima
Waktu	Waktu dekompresi tidak terlalu lama	Waktu dekompresi tergantung dari ukuran gambar hasil kompresinya, apabila ukuran gambar kecil maka waktu yang dibutuhkan tidak terlalu lama dibandingkan dengan gambar yang berukuran besar, sesuai yang	Diterima

		diharapkan.	
Kapasitas	Kapasitasnya seperti gambar asli	Apabila kapasitas gambar hasil dekompres sama dengan citra asli maka telah berhasil.	Diterima

Tabel 4.4 Pengujian Dekompresi dengan Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Tidak ada *File* yang ditampilkan di *list view*	Akan ditampilakan pesan kesalahan pada dekompresi.	Ditampilkan pesan kesalahan dekompresi.	Diterima

4.6.3 Pengujian Data yang di Uji

Pengujian yang dilakukan berdasarkan *File* teks dan citra yang ekstensinya berbeda. *File-file* yang akan di ujicoba *berekstensi* yaitu :

a. *File* teks : *doc,docx, pdf, dan rtf.*

b. *File* gambar : *jpg, bmp.*

Dalam pengujian ini ada 7 jenis *File* yang masing-masing *berekstensi* berbeda. Pemilihan jenis *file* di tersebut dikarenakan jenis jenis *File* yang sering digunakan dalam pengolahan data. Pengujian tersebut

yang telah dirancang selanjutnya di ujicoba dan dilakukan analisis terhadap kinerja algoritma, yaitu:

Seluruh *File* dengan ekstensi yang berbeda dikompres dengan menggunakan LZSS. Untuk pengujian ini di catat ukuran semula, ukuran *File* termampatkan serta rasio pemampatannya. Perhitungan rasio pemampatan dilakukan dengan membandingkan ukuran *File* termampatkan dengan ukuran semula (*Held, 1991*). Perhitungan rasio pemampatan didefinisikan dengan :

Rasio Pemampatan = SK/HK

SK = Ukuran *File* sebelum pemampatan

HK = Ukuran *File* hasil pemampatan

Berikut merupakan hasil pengujian data dengan aplikasi yang di bangun :

4.6.3.1 Pengujian Dengan Membandingkan Hasil Kompresi *File* Teks

1. *File* Teks dengan menggunakan metode LZSS

Tabel 4.5 Hasil Kompresi *File* Teks Menggunakan Metode *LZSS*

Jenis *File* Teks	Ukuran *File* Asli (*Byte*)	Ukuran file pecahan (Byte) ke-						Total Ukuran File Pecahan	Ratio (%)	Persentase (%)	Waktu
		I	II	III	IV	V	VI				
Pdf	579717	70000	98000	10600	10100	10200	77000	554000	1.02	97.68	00:00:03:750
Doc	139264	53000	15000	0	0	0	0	68000	2.00	50.00	00:00:00:990
Rtf	73550	24000	0	0	0	0	0	24000	2.99	33.41	00:00:02:622

Docx	16384	16000	0	0	0	0	0	16000	1.00	100.00	00:00:00:133

2. *File* Teks dengan menggunakan metode *Rice Coding*

Tabel 4.6 Hasil Kompresi *File* Teks Menggunakan Metode *Rice Coding*

Jenis *File* Teks	Ukuran *File* Asli (*Byte*)	Ukuran file pecahan (Byte) ke-						Total Ukuran File Pecahan	Ratio (%)	Persentase (%)	Waktu
		I	II	III	IV	V	VI				
Pdf	579717	98000	98000	98000	98000	98000	78000	568000	0.99	100.00	00:00:01:310
Doc	139264	92000	24000	0	0	0	0	116000	1.21	82.35	00:00:00:359
Rtf	73550	67000	0	0	0	0	0	67000	1.09	91.89	00:00:00:889
Docx	16384	15300	0	0	0	0	0	15300	1.07	93.75	00:00:00:041

4.6.3.2 Pengujian Dengan Membandingkan Hasil Kompresi *File* Citra

1. *File* Citra dengan menggunakan metode LZSS

Tabel 4.7 Hasil Kompresi *File* Citra Menggunakan Metode LZSS

Jenis *File* Teks	Ukuran *File* Asli (*Byte*)	Ukuran file pecahan (Byte) ke-						Total Ukuran File Pecahan	Ratio (%)	Persentase (%)	Waktu
		I	II	III	IV	V	VI				
Bmp	228536	91000	96000	24000	0	0	0	211000	1.06	94.54	00:00:01:520
Jpg	524817	92000	94000	98000	98000	93000	32000	507000	1.03	96.78	00:00:03:412

2. *File Citra* dengan menggunakan metode *Rice Coding*

Tabel 4.8 Hasil Kompresi *File* Citra Menggunakan Metode *Rice Coding*

Jenis *File* Teks	Ukuran *File* Asli (*Byte*)	Ukuran file pecahan (Byte) ke-						Total Ukuran File Pecahan	Ratio (%)	Persentase (%)	Waktu
		I	II	III	IV	V	VI				
Bmp	228536	98000	98000	28000	0	0	0	224000	1.00	99.80	00:00:00:531
Jpg	524817	98000	98000	98000	98000	98000	28000	518000	1.01	98.73	00:00:01:170

4.6.3.3 Pengujian Dengan Membandingkan Hasil Dekompresi *File* Teks

1. *File* Teks dengan menggunakan metode LZSS

Tabel 4.9 Hasil Dekompresi *File* Teks Menggunakan Metode *LZSS*

Nama dan Ukuran *File* (*Byte*)			Perubahan Informasi		
Jenis *File* Teks	Kapasitas Total Ukuran file Pecahan	Kapasitas Dekompres	Ya	Tidak	Rusak
Pdf	554000	579716		√	
Doc	68000	139263		√	
Rtf	24000	394214		√	
Docx	16000	16384		√	

2. *File* Teks dengan menggunakan metode *Rice Coding*

Tabel 4.10 Hasil Dekompresi *File* Teks Menggunakan Metode *Rice Coding*

Nama dan Ukuran *File* (*Byte*)			**Perubahan Informasi**		
Jenis *File* Teks	**Kapasitas Total ukuran File pecahan**	**Kapasitas Dekompres**	**Ya**	**Tidak**	**Rusak**
Pdf	568000	579717		√	
Doc	116000	139264		√	
Rtf	67000	394214		√	
Docx	15300	16384		√	

4.6.3.4 Pengujian Dengan Membandingkan Hasil Dekompresi *File* Citra

1. *File* Citra dengan menggunakan metode LZSS

Tabel 4.11 Hasil Dekompresi *File* Citra Menggunakan Metode *LZSS*

Nama dan Ukuran *File* (*Byte*)			**Perubahan Informasi**		
Jenis *File* Citra	**Kapasitas Total File Pecahan**	**Kapasitas Dekompres**	**Ya**	**Tidak**	**Rusak**
bmp	211000	235571		√	

jpg	507000	524817		√	

2. *File* Citra dengan menggunakan metode *Rice Coding*

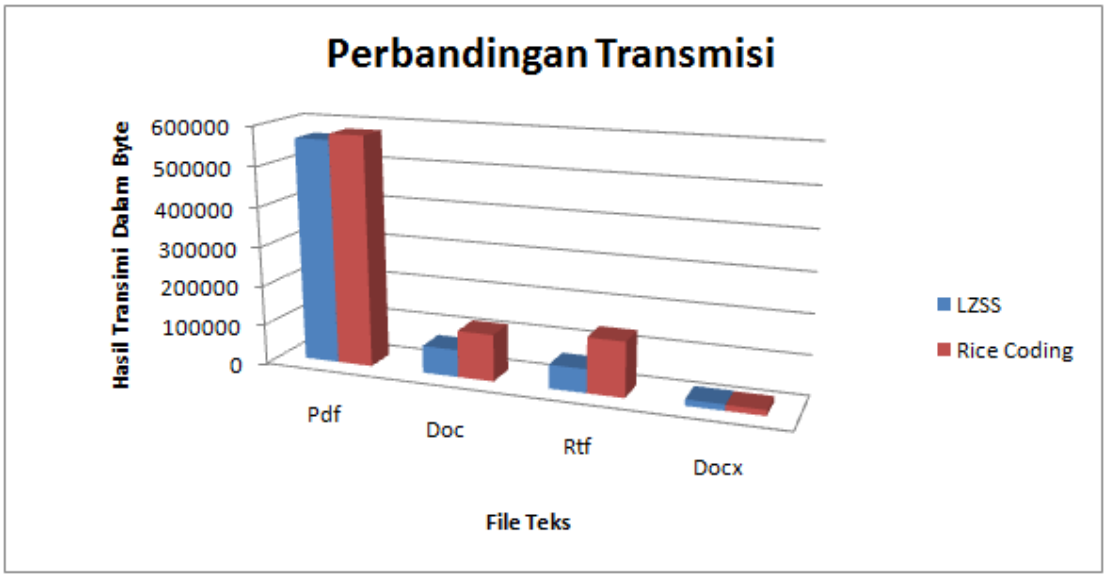
Tabel 4.12 Hasil Dekompresi *File* Citra Menggunakan Metode *Rice Coding*

Nama dan Ukuran *File* (*Byte*)			**Perubahan Informasi**		
Jenis *File* Citra	**Kapasitas Total File Pecahan**	**Kapasitas Dekompres**	**Ya**	**Tidak**	**Rusak**
bmp	224000	228536		√	
jpg	518000	524817		√	

4.6.4 Implementasi Dan Pengujian *File* Teks Berdasarkan *Grafik*

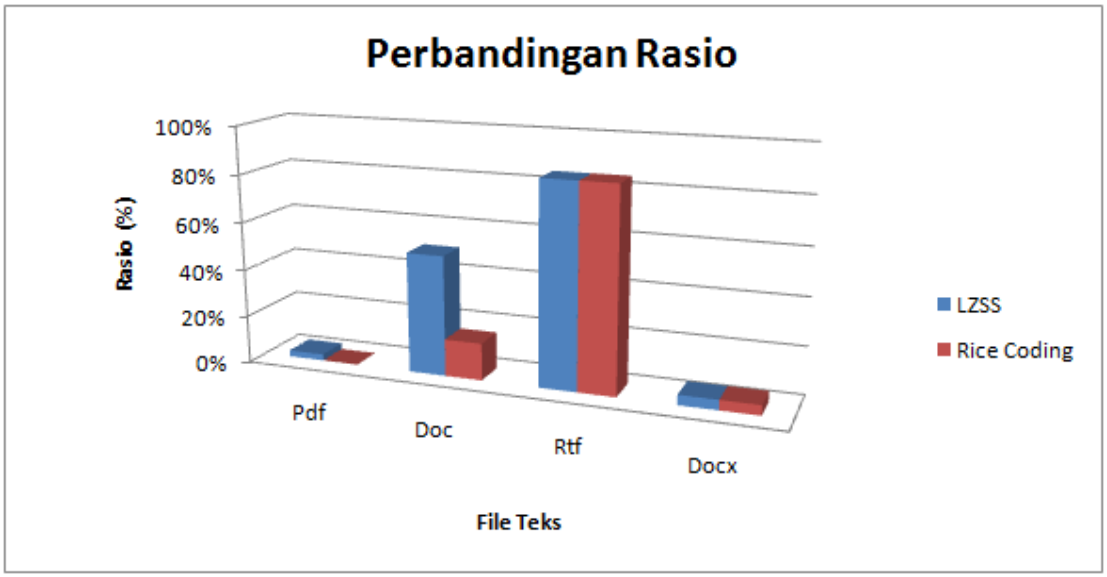
Setelah dilakukan pengujian, langkah selanjutnya adalah membandingkan hasil *file* teks yang telah diperoleh dalam bentuk grafik. Grafik tersebut terdiri dari:

4.6.4.1 Grafik Perbandingan Hasil Transmisi Teks



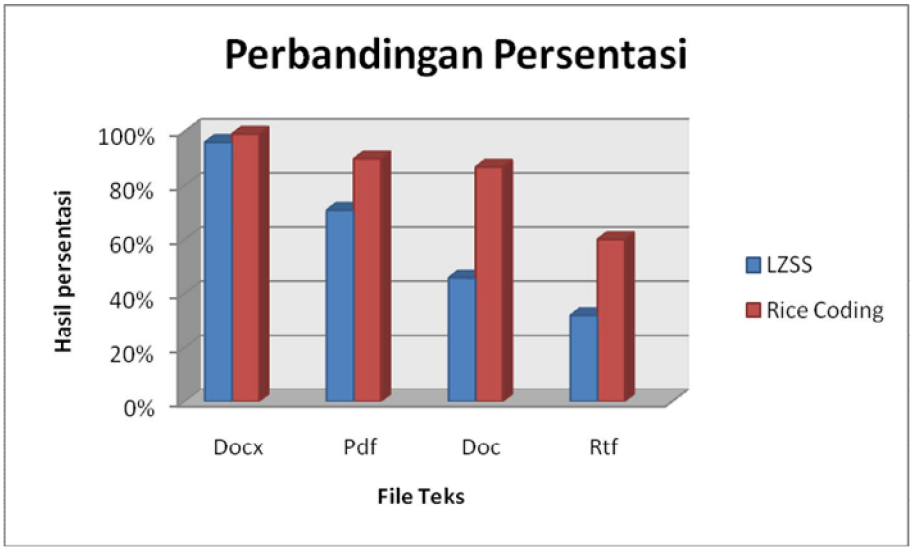
Gambar 4.16 Hasil Perbandingan Transmisi *File* Teks antara Metode LZSS dan *Rice Coding*

4.6.4.2 *Grafik* Perbandingan *Rasio* Teks



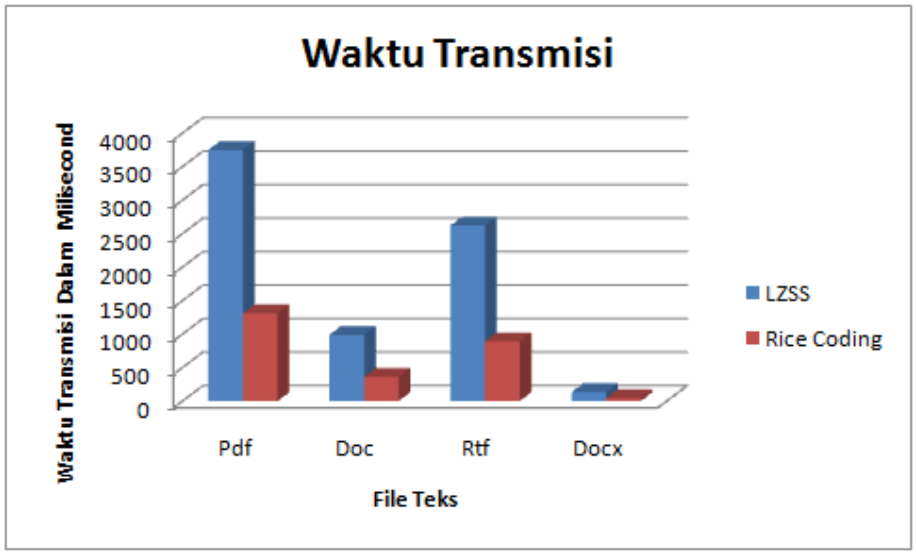
Gambar 4.17 Hasil Perbandingan Rasio *File* Teks antara Metode LZSS dan *Rice Coding*

4.6.4.3 *Grafik* Perbandingan Persentasi Teks



Gambar 4.18 Hasil Perbandingan Persentasi *File* Teks antara Metode LZSS dan *Rice Coding*

4.6.4.4 *Grafik* perbandingan Waktu Transmisi Teks

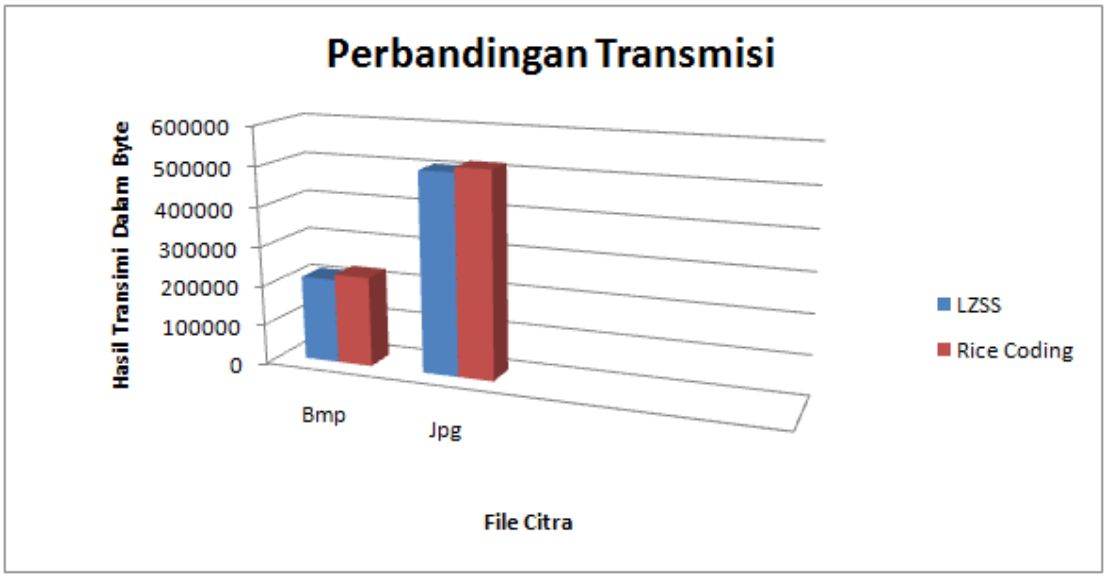


Gambar 4.19 Hasil Perbandingan Waktu *File* Teks antara Metode LZSS dan *Rice Coding*

4.6.5 Implementasi Dan Pengujian *File Citra* Berdasarkan *Grafik*

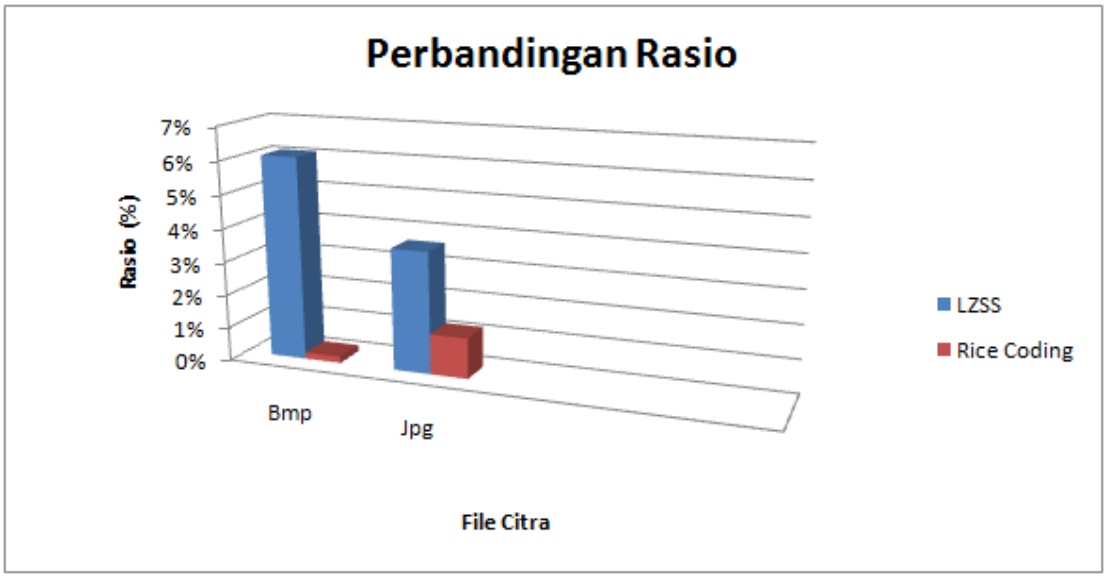
Analisis dan pengujian selanjutnya akan dilanjutkan pada *file* citra yaitu dengan membandingkan hasil *file* citra yang telah diperoleh dalam bentuk *grafik.* Grafik tersebut terdiri dari:

4.6.5.1 Grafik Perbandingan Hasil Transmisi Citra



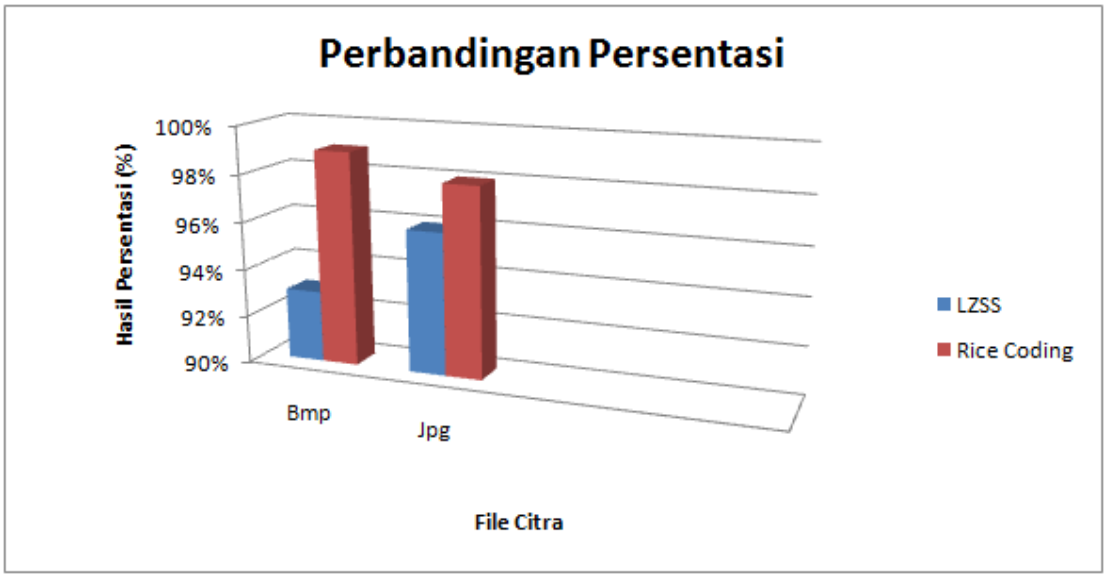
Gambar 4.20 Hasil Perbandingan Transmisi *File* Citra antara Metode LZSS dan *Rice Coding*

4.6.5.2 Grafik Perbandingan Rasio Citra



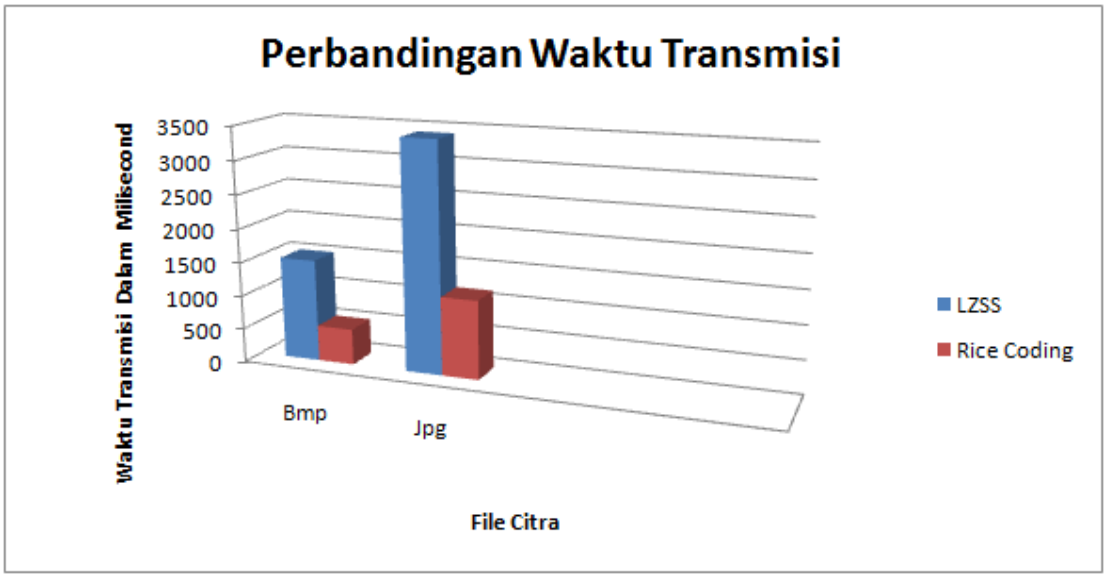
Gambar 4.21 Hasil Perbandingan Rasio *File* Citra antara Metode LZSS dan *Rice Coding*

4.6.5.3 *Grafik* Perbandingan Persentasi Citra



Gambar 4.22 Hasil Perbandingan Persentasi *File* Citra antara Metode LZSS dan *Rice Coding*

4.6.5.4 *Grafik* perbandingan Waktu Transmisi Citra



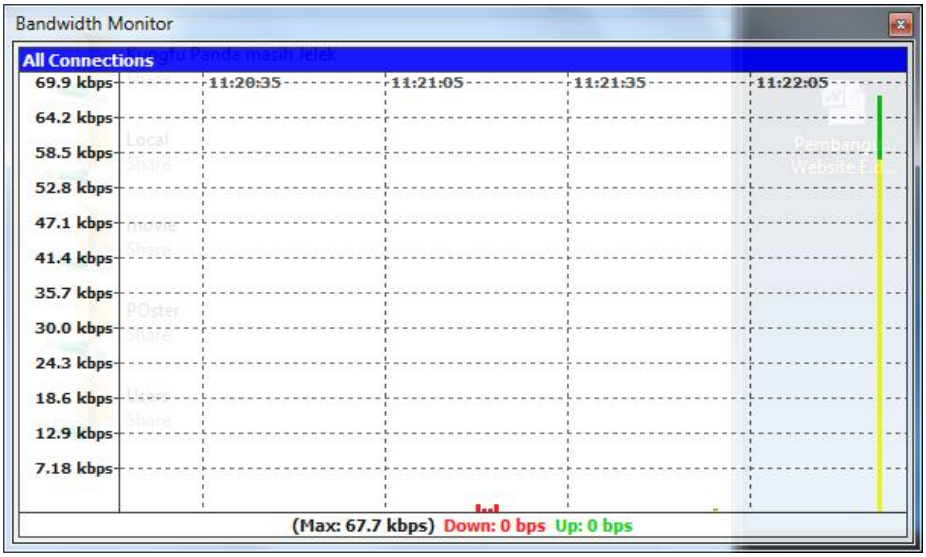
Gambar 4.23 Hasil Perbandingan Waktu Transmisi *File* Citra antara Metode LZSS dan *Rice Coding*

4.6.6 Grafik Pengujian Terhadap Penggunaan *Bandwidth*

Setiap proses *upload* (kompresi) dan *download* (dekompresi), baik yang melalui proses kompresi, dekompresi dan tanpa proses kompresi dan dekompresi tentu memerlukan *bandwidth* disetiap prosesnya. Berikut rata-rata penggunaan *bandwidth* yang digunakan.

1. *Grafik Bandwidth* Dalam Keadaan Normal

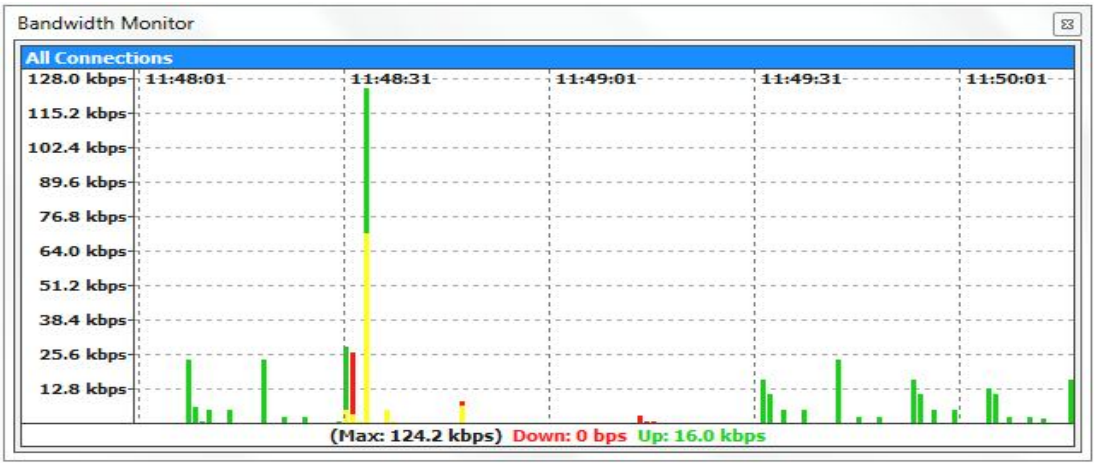
Sebelum melakukan *upload* (kompresi) dan *download* (dekompresi), kita harus mengetahui terlebih dahulu keadaan normal *bandwidth*. Grafik normal *bandwidth* dapat dilihat pada gambar 4.24. Dimana *bandwidth* melakukan proses *download* tarhadap *server* dengan rata-rata 67.7 kbps.



Gambar 4.24 *Grafik Bandwidth* dalam keadaan Normal

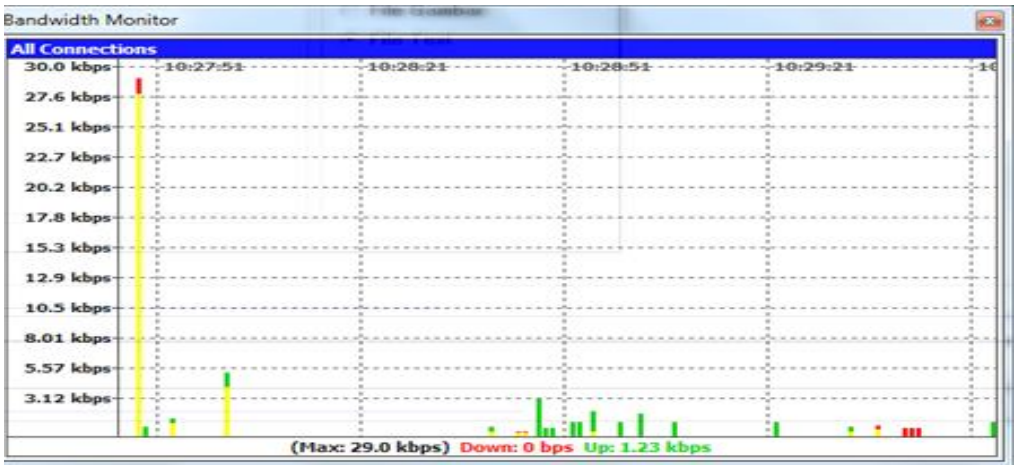
2. *Grafik Bandwidth* Pada Proses *Upload (kompresi)*

Pada proses *upload file* tanpa menggunakan aplikasi, rata-rata menggunakan *bandwidth* 124.2 kbps. Dimana proses ini mengghabiskan kemampuan *bandwidth* sekitar 16.0 kbps. Grafik dapat dilihat pada gambar 4.25.



Gambar 4.25 *Grafik Bandwidth* Proses *Upload file* Tanpa aplikasi

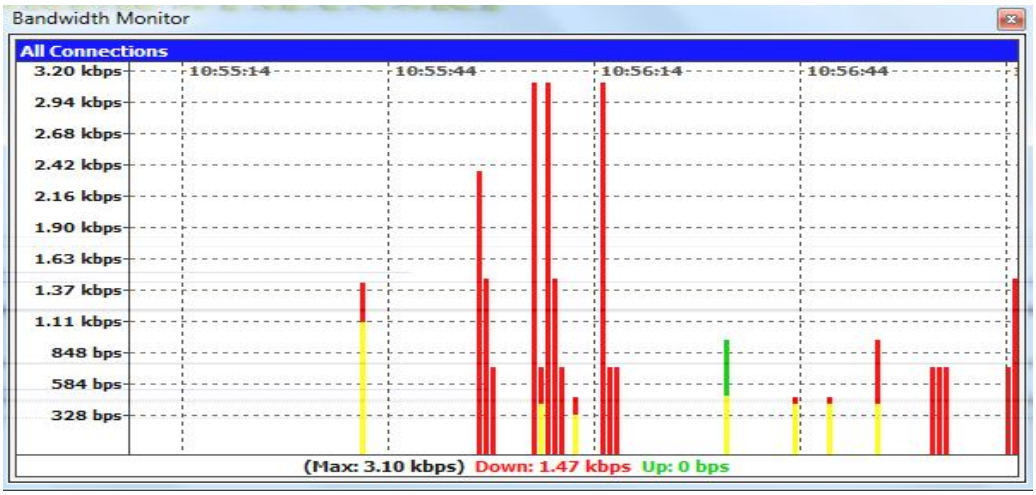
Sedangkan proses *upload file* dengan menggunakan aplikasi, rata-rata menggunakan *bandwidth* 29.0 kbps. Dimana proses ini menggunakan *bandwidth* yang relatif kecil. Grafik dapat dilihat pada gambar 4.26.



Gambar 4.26 *Grafik Bandwidth* Proses *Upload file* dengan aplikasi

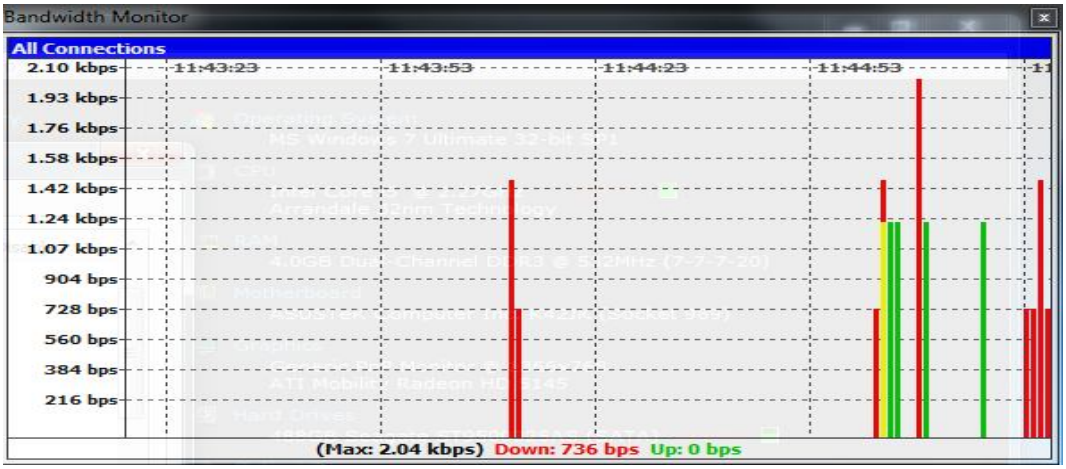
3. *Grafik Bandwidth* Pada Proses *Download (dekompresi)*

Pada proses *download file* tanpa menggunakan aplikasi, rata-rata menggunakan *bandwidth* 3.10 kbps. Dimana proses ini hampir menggunakan kemampuan maksimal *bandwidth*. Grafik dapat dilihat pada gambar 4.27.



Gambar 4.27 *Grafik Bandwidth* Proses *Download file* Tanpa aplikasi

Sedangkan proses *download file* dengan menggunakan aplikasi, rata-rata menggunakan *bandwidth* 2.04 kbps. Dimana proses ini hampir menggunakan kemampuan maksimal *bandwidth*, namun relatif lebih kecil dibandingkan dengan tanpa menggunakan aplikasi. Grafik dapat dilihat pada gambar 4.28.



Gambar 4.28 *Grafik Bandwidth* Proses *Download file* dengan aplikasi

4. Grafik *Bandwidth* Pada Proses *Upload* dan *Download* (kompresi dan dekompresi)

Pada proses *upload* dan *download file* tanpa menggunakan aplikasi, rata-rata menggunakan *bandwidth* 3.10 kbps. Dimana proses ini hampir menggunakan kemampuan maksimal *bandwidth*. Grafik dapat dilihat pada gambar 4.24.

Sedangkan proses *upload* dan *download file* dengan menggunakan aplikasi, rata-rata menggunakan *bandwidth* 3.10 kbps pada proses *download* 1.44 kbps dan 1.54 kbps pada proses *upload*. Grafik dapat dilihat pada gambar 4.29.



Gambar 4.29 *Grafik Bandwidth* Proses *Upload* dan *Download file* dengan aplikasi