BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi jaringan komputer dewasa ini berkembang semakin cepat seiring dengan kebutuhan masyarakat akan layanan yang memanfaatkan jaringan komputer. Pada sistem jaringan komputer,protokol merupakan suatu bagian yang paling penting. *Protocol* jaringan yang umum digunakan adalah *Internet Protocol* (IP). Versi yang telah kita kenal selama ini adalah IP versi 4 yang merupakan desain awal yang berharga. Akan tetapi *Internet Protocol* yang sudah kita kenal lebih dari 25 tahun ini kurang mengantisipasi faktor-faktor yang tak terduga seperti keterbatasan jumlah alamat yang ditawarkan. Karena kebutuhan akan internet semakin banyak maka diciptakanlah *Internet Protocol Version* 6 atau IPv6 (Rafiudin, 2007:1-3).

Perkembangan yang pesat dibidang teknologi aplikasi berbasis web sangat membantu dalam kemudahan serta kecepatan pengiriman, penyampaian dan penerimaan informasi sehingga banyak sistem yang menggunakan teknologi aplikasi web tersebut. Aplikasi web telah berkembang pesat dari segi ukuran, bahasa yang digunakan dan kompleksitasnya. Aplikasi web yang selama ini yang kenal masih menggunakan IPv4. Dikarenakan IPv6 masih dalam proses transisi maka perlu dianalisis apakah ada pengaruh kinerja aplikasi web dengan menggunakan IPv6 yang dilakukan berdasarkan *user experience.*

1.2 Rumusan Masalah.

Perancangan sistem ini digunakan untuk membangun dan melihat kinerja berbagai aplikasi web dengan menggunakan IPv6 berdasarkan *user experience* dengan *client* yang menggunakan IPv6. Hal yang akan diamati yaitu *loading time* pengaksesan halaman web.

1.3 Batasan Masalah

1. Aplikasi web yang akan digunakan antara lain *blog*, video *streaming*, web *shopping cart*, web portal berita standar, dan web jejaring sosial.
2. Menggunakan sebuah komputer sebagai web *server* yang memakai sistem operasi linux.
3. Mengamati dan memberikan hasil *loading time* pada masing-masing aplikasi web tanpa dibatasi bandwidth dengan menggunakan IPv6 serta *buffering time* pada video *streaming*.
4. Menggunakan *router dedicated* sebagai *router* dan *bandwidth Management*
5. Aplikasi web dan *browser* yang dipakai menggunakan HTML versi 4.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah menganalisis kinerja aplikasi web dari sisi *loading time* dengan menggunakan IPv6 sehingga kita dapat mengetahui bagaimana kinerja masing-masing aplikasi web yang memiliki karakteristik yang berbeda-beda tersebut.

1.5 Manfaat Penelitian

Melalui hasil analisis dengan mengakses aplikasi web dengan IPv6, pengguna dapat mengetahui aplikasi web yang seperti apa yang memiliki kinerja yang baik dari sisi *loading time*.

1.6 Metode Penelitian

1.6.1 Studi Pusatka

Studi pustaka digunakan untuk menggali informasi yang terkait dengan penelitian, yaitu melalui buku-buku dan internet terkait dengan penelitian, pemelihan perangkat lunak yang digunakan serta pencarian referensi dalam analisis kinerja aplikasi web.

1.6.2 Implementasi dan Pembangunan Sistem

Analisis perbandingan penggunaan IPv4 dan IPv6 dalam aplikasi web disusun berdasarkan perolehan studi pustaka yang meliputi :

a. Desain Arsitektur Jaringan

   Tahap ini merupakan perancangan arsitektur jaringan komputer yang akan digunakan untuk melakukan analisis kinerja berbagai aplikasi web yang menggunakan IPv6.

b. Pengadaan Perangkat Keras

   Tahap ini merupakan tahap pengadaan perangkat keras berupa *mikrotik* RB750, komputer *client* dan komputer *server* yang nantinya sebagai web server.

c. Pembangunan Sistem

   Tahap ini merupakan tahap instalasi dan konfigurasi web *server*, sistem operasi Linux, konfigurasi *mikrotik routerboard*, dan pemberian alamat IPv6.

d. Metode Analisis

   Tahap ini merupukan penentuan jenis statistik yang digunakan untuk menganalisis, menentukan jenis pengujian, dan menentukan hipotesis.

e. Analisis Kinerja Aplikasi Web

   Tahap ini merupakan tahap analisis kinerja aplikasi web tersebut berdasarkan topologi yang dipakai.

f. Pengambilan Kesimpulan

   Tahap ini merupakan tahap analisis berdasarkan dari data yang sudah terkumpul untuk dilihat hasilnya. Kemudian dari hasil analisis tersebut dapat diambil kesimpulan bagaiman kinerja pada masing-masing web yang memiliki karakteristik yang berbeda-beda tersebut.

1.7 Hipotesis

Karena kinerja aplikasi web tergantung pada konten-konten yang ada di dalam aplikasi web, maka web multimedia lah yang memiliki *loading time* yang lebih lama dari pada web yang lainnya dikarekan aplikasi web tersebut berisi video-video.

1.8 Sistematika Penulisan

BAB I Pendahuluan

Membahas tentang latar belakang masalah, batasan masalah, rumusan masalah, tujuan penelitian, manfaat penelitian dan metodologi yang diangkat menjadi materi laporan tugas akhir Studi Komparasi Kinerja Aplikasi Web pada IPv6.

BAB II Landasan Teori

Membahas dasar-dasar teori yang digunakan dalam perancangan dan pembangunan sebuah web *server*, DNS, *router* sebagai *gateway* yang kemudian akan dianalisis bagaimana kinerja berbagai aplikasi web dengan menggunakan IPv6

BAB III Metodologi

Berisi uraian tentang gambaran umum tentang sistem, analisis kabutuhan perangakat keras dan perangkat lunak yang digunakan untuk membantu penyelesaian tugas akhir, metode analisis yang akan dipakai, dan pembangunan sistem.

BAB IV Hasil dan Pembahasan

Bagian ini memuat dokumentasi pengujian terhadap aplikasi web, yaitu dengan melakukan analisis terhadap kinerja aplikasi web dan kemudian mengambil kesimpulan dari data.

BAB V Penutup

Memuat kesimpulan-kesimpulan dari seluruh proses analisis, saran-saran yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan selama melakukan analisis kinerja aplikasi web pada IPv6.

BAB II

LANDASAN TEORI

2.1 Jaringan Komputer

2.1.1 Pengertian Jaringan Komputer

Jaringan komputer adalah dua atau lebih komputer yang saling terhubung melalui media perantara sehingga dapat berbagi sumber daya (*resources*).Media perantara ini bisa berupa media kabel ataupun media tanpa kabel (nirkabel).

Berdasarkan fungsinya jaringan komputer dapat dibagi menjadi dua jenis, yaitu :

1. *Client - Server*

Client server adalah jaringan komputer yang salah satu (boleh lebih) komputer difungsikan sebagai *server* bagi komputer lain. *Server* melayani komputer lain yang disebut *client*. Layanan yang diberikan bisa berupa web, *e-mail*, file, atau lainnya. *Client server* banyak dipakai pada internet, namun LAN atau jaringa lain pun bisa mengimplementasikan *client server*. Hal ini sangat bergantung pada kebutuhan masing-masing (Sofana, 2008:6).

2. *Peer to Peer*

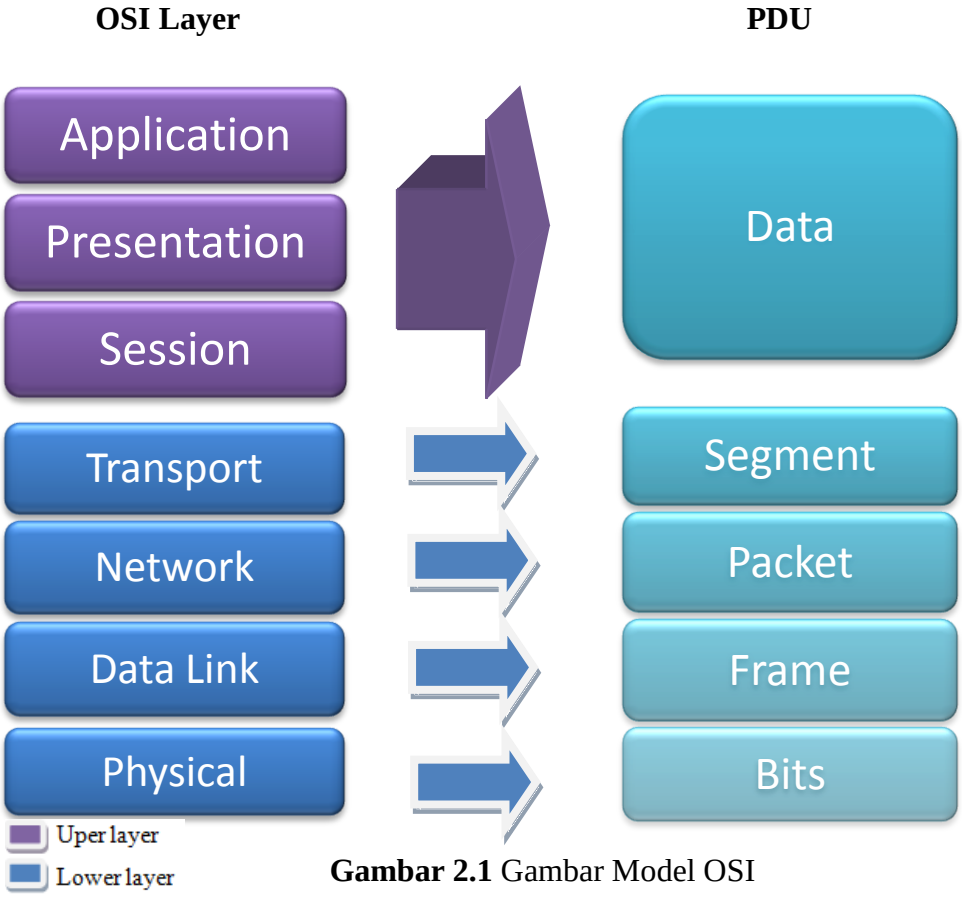
Jaringan *peer to peer* (biasanya disingkat P2P) adalah satu atau lebih komputer yang saling terhubung, baik media kabel atau nirkabel dan tiap komputer dapat berkomunikasi secara langsung. Pada jaringan P2P, sebuah komputer dapat menjadi *client* sekaligus *server* pada saat yang bersamaan dan tidak ada autentifikasi secara terpusat, autentifikasi dapat diatur disetiap node yang memberikan layanan (sisjarkom, 2011:4).

2.1.2 Model OSI

Model OSI dibuat untuk mengatasi berbagai kendala *internetworking* akibat perbedaan arsitektur dan protokol jaringan. Dahulu, komunikasi antarkomputer dari *vendor* yang berbeda sangat sulit dilakukan. Masing-masing *vendor* menggunakan

protokol dan format data yang berbeda-beda. Sehingga *International Organization for Standardization* (ISO) membuat sebuah arsitektur komunikasi yang dikenal sebagai *Open System Interconnection* (OSI) model yang mendefinisikan standar untuk menghubungkan komputer-komputer dari *vendor* yang berbeda.

OSI model terdiri dari tujuh layer dan secara umum dibagi menjadi dua kelompok yaitu *uper layer* (*application layer*) dan *lower layer* (*data transport layer)*. Layer yang tergolong *uper layer* mendefinisikan bagaimana aplikasi pada sebuah *host* akan berkomunikasi dengan *user* dan *host* lainnya. Sedangkan *lower layer* mendefinisikan bagaimana data terkirim dari *host* satu ke *host* lainnya. Proses komunikasi dan pertukaran informasi dari tiap-tiap layer menggunakan sebuah protokolyang disebut sebagai *Protocol Data Unit* (PDU).



Gambar 2.1 Gambar Model OSI

Secara umum, fungsi dari ke-tujuh OSI model diatas adalah sebagai berikut :

7 (*Application Layer*)

Berfungsi sebagai antarmuka (penghubung) aplikasi dengan fungsionalitas jaringan, mengatur bagaimana aplikasi dapat mengakses jaringan, dan kemudian membuat pesan-pesan kesalahan. Pada layer inilah sesungguhnya *user* berinterakasi dengan jaringan. Contoh protokol yang berada pada lapisan ini adalah FTP, *telnet*, SMTP, HTTP, POP3 dan NFS (Sofana,2008:81).

6 (*Presentation Layer*)

Berfungsi untuk mentranslasikan data yang hendak di transmisikan oleh aplikasi dalam format yang dapat ditransmisikan melalui jaringan.

5 (*Session Layer*)

Berfungsi untuk mendefinisikan bagaimana koneksi dimulai, dipelihara, dan diakhiri.

4 (*Transport Layer*)

Berfungsi untuk memecah data menjadi paket-paket data serta memberikan nomor urut setiap paket sehingga dapat disusun kembeli setelah diterima. Paket yang diterima dengan sukses akan deberi tanda (*acknowledgement*). Sedangkan paket yang rusak atau hilang di tengah jalan akan dikirim ulang. Contoh protokol yang digunakan TCP dan UDP (Sofana,2008:82).

3 (*Network*)

Berfungsi mendefinisikan alamat-alamat IP, membuat *header* untuk paket-paket, dan melakukan *routing* melalui *internetworking* dengan menggunakan *router* dan *switch* layer-3.

2 (*Data Link*)

Berfungsi untuk menentukan bagaimana bit-bit data dikelompokkan menjadi frame. Pada level ini terjadi *error correction*, *flow control*, pengalamatan perangkat keras (MAC *Address*), dan menentukan bagaimana perangkat-perangkat jaringan seperti *bridge* dan *switch* layer 2 beroperasi (Sofana,2008:82).

1 (*Physical*)

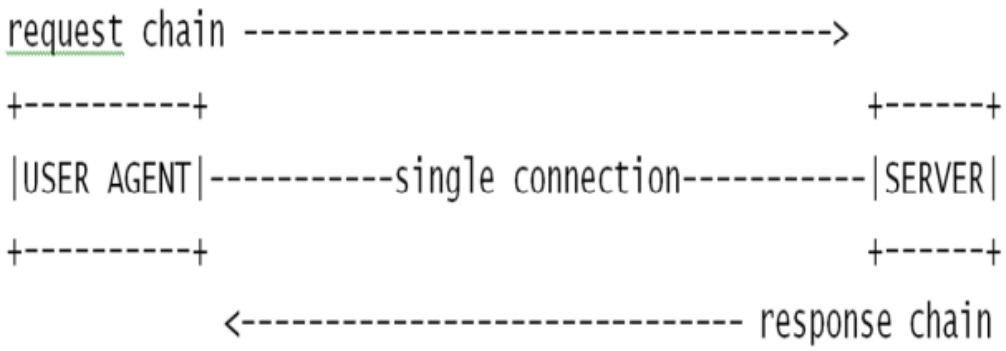
Berfungsi untuk mendefinisikan media transmisi jaringan, metode pensinyalan, sinkronisasi bit, arsitektur jaringan, topologi jaringan, dan pengabelan. Selain itu level ini juga mendefinisikan bagaimana *Network Interface Card* (NIC) berinteraksi dengan media *wire* atau *wireless* (Sofana,2008:83).

2.2 Web *Server*

Setiap kali sebuah web *browser* berhubungan dengan suatu situs web, sebetulnya ia terhubung dengan sebuah web *server*. *Server* tersebut mendengarkan request pada jaringan dan memberikan jawaban berupa data tertentu kepada *client* atau pengirim permintaan. Web *server* atau HTTP adalah sebuah program yang melayani koneksi HTTP (*Hyper Text Transfer Protokol*). Web *server* bekerja berdasarkan *request-response*, yaitu kerika HTTP *client* (misalnya web *browser*) membangun koneksi dengan *server* dengan mengirimkan *request* (permintaan) kepada *server*, maka *server* akan merespon dengan mengolah permintaan tersebut kemudian mengirimkan data sesuai yang diminta oleh HTTP *client*. Format data yang dikirimkan oleh HTTP *server* mengunakan format SGML (*Standard General Markup Languange*), data ini akan ditampilkan oleh HTTP *client* (*browser*) sesuai dengan fasilitas yang dimilikinya. Semisal data yang dikirimkan berupa data gambar, sedangkan browser yang dipakai hanya mampu menampilkan data berupa teks (misal, *lynx* dan *links*) maka *browser* tidak akan mampu menampilkan data tersebut secara sempurna, atau hanya akan ditampilkan alternatifnya saja. Koneksi ke web *server* dilakukan melalui protokol yang dikenal dengan HTTP (*Hypertext Transfer Protokol*) yang secara *default* menggunakan *port* 80 (nilai *port* dapat diubah).

HTTP merupakan protokol yang bekerja pada lapisan aplikasi (*application layer*) dan secara sederhana dapat didefinisikan sebagai sekumpulan aturan untuk tukar menukar data pada *world wide web* (www). Ide dasar dari HTTP adalah bagaimana sebuah file dapat berisi suatu referensi pada file yang lain dengan

sebuah transfer permintaan file yang bersangkutan. Sebenarnya HTTP juga dapat digunakan sebagai *generic protocol* untuk melakukan komunikasi antara *user agents*, *proxy*, dan *gateway* dengan protokol internet lain seperti SMTP, FTP, Gopher, NNTP dan lain sebagainya. Gambar di bawah ini mengilustrasikan secara sederhana bagan *alur request-response* antara *user agent* dengan *server*.



Gambar 2.2 Alur *Request-response* Antara *User Agent* dengan *Server*.

2.3 Aplikasi Web

Dalam rekayasa perangkat lunak, aplikasi web adalah suatu aplikasi yang diakses menggunakan penjelajah web melalui jaringan seperti internet atau intranet.Ia juga merupakan suatu aplikasi perangkat lunak komputer yang dikodekan dalam bahasa yang didukung penjelajah web (seperti HTML, JavaScript, AJAX, Java, dan lain-lain). Aplikasi Web dapat dibagi menjadi dua jenis yaitu aplikasi web statis dan dinamis.Web statis dibentuk dengan menggunakan HTML.Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus menerus untuk mengikuti setiap perkembangan yang terjadi.Kelemahan ini diatasi oleh model aplikasi web dinamis. Pada aplikasi web dinamis, perubahan informasi dalam halaman web dilakukan tanpa perubahan program tetapi melalui perubahan data. Sebagai implementasi, aplikasi web dapat dikoneksikan ke basis data sehingga perubahan informasi dapat dilakukan oleh operator dan tidak menjadi tanggung jawab dari *webmaster*.

2.4 DNS (*Domain Name System*)

Domain Name System adalah distribusi *database* yang digunakan untuk memetakan nama komputer ke alamat IP (*forward mapping*) dan dari alamat IP ke nama komputer (*reverse mapping*). Untuk melakukan komunikasi dan mengakses komputer yang terhubung dalam jaringan, sebetulnya cukup menggunakan IP *address* dari masing-masing komputer saja. Tetapi akan menjadi masalah ketika banyak sekali komputer yang terhubung dalam jaringan. Akan sangat sulit tentunya apabila kita harus mengingat IP *address* dari masing-masing komputer. Itulah mengapa dibuat teknik penerjemahan nama komputer ke IP *address* atau sebaliknya.

Agar lebih mudah diingat, maka IP *address* harus dipetakan ke dalam sebuah nama. Sebagai contoh : komputer dengan IP *address* 202.162.33.10 memiliki nama www.facebook.com, IP *address* 203.162.32.1 adalah www.google.com dan lain sebagainya. Sehingga, jika kita akan menghubungi sebuah komputer di internet, misalnya www.facebook.com maka kita cukup menggunakan nama komputer tersebut. Komputer saling berkomunikasi menggunakan IP *address* bukan menggunakan nama komputer. Untuk itulah diperlukan sebuah mekanisme agar nama komputer tersebut dapat diterjemahkan ke dalam IP *address*, salah satunya adalah dengan memanfaatkan layanan DNS *server*.

2.5 IP *Address*

IP *Address* atau alamat IP adalah pengalamatan untuk mengidentifikasi *interface* jaringan pada *host* dari suatu mesin. IP *address* terdiri atas dua versi yaitu IPv4 dan IPv6.

2.5.1 IPv4 (IP versi 4)

IP versi 4 berupa sekelompok bilangan biner 32 bit yang dibagi menjadi 4 bagian yang masing-masing bagian terdiri dari 8 bit. Jumlah alamat yang bisa dibentuk oleh IP versi 4 ini adalah sebanyak 2^{32} atau setara dengan 4.294.967.296 buah. Untuk memudahkan pembacaan dan penulisan, IP *address* telah dikonversikan ke bilangan desimal yang dipisahkan oleh tanda titik atau *dotted-decimal format*. Apabila setiap

segmen dikonversikan ke bilangan desimal berarti nilai yang mungkin antara 0 hingga 255.Contoh IP versi 4 adalah **01000100 10000001 11111111 00000001** Jika dikonversikan ke bilangan desimal menjadi **68.128.255.1**.

Untuk memudahkan pengaturan IP *address* seluruh komputer pengguna jaringan internet, dibentuklah suatu badan yang mengatur pembagian IP *address*. Badan tersebut bernama InterNIC (*Internet Network Information Center*). InterNIC membagi IP *address* menjadi beberapa kelas :

1. **Kelas A**

   Format : 0xxxxxxx.yyyyyyyy.yyyyyyyy.yyyyyyyy
   Netmask default : 255.0.0.0
   Kisaran : 0.0.0.0 – 127.255.255.255
   Jumlah *host* : 16,777,214

2. **Kelas B**

   Format : 10xxxxxx.xxxxxxxx.yyyyyyyy.yyyyyyyy
   Subnetmask default : 255.255.0.0
   Kisaran : 128.0.0.0 – 191.255.255.255
   Jumlah *host* : 65,532

3. **Kelas C**

   Format : 110xxxxx.xxxxxxxx.xxxxxxxx.yyyyyyyy
   Subnetmask default : 255.255.255.0
   Kisaran : 192.0.0.0 – 223.255.255.255
   Jumlah *host* : 254

4. **Kelas D**

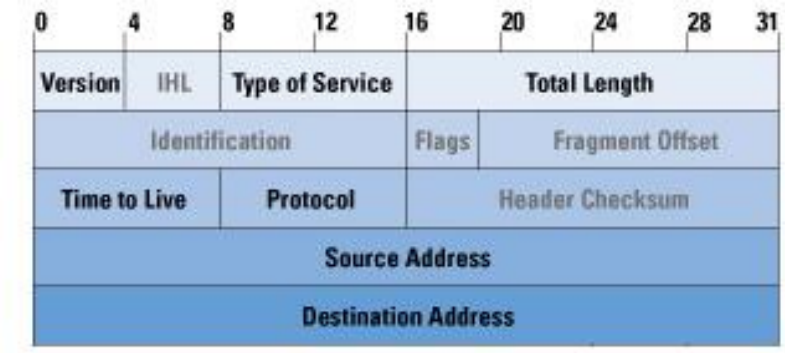
   Selain tiga kelas di atas, ada dua kelas lagi yang ditunjukkan untuk pemakaian khusus, yakni kelas D dan E. Jika empat bit pertama dalah 1110, maka IP *address* termasuk dalam kelas D. IP *address* kelas D digunakan untuk *multicast address* yakni sebuah komputer yang memakai bersama suatu aplikasi. Salah satu penggunaan *multicast address* yang sedang berkembang di

dunia internet adalah suatu aplikasi *real-time video conference* yang melibatkan dua *host*. Pada IP *address* kelasD tidak dikenal bit-bit *network* dan *host* (Sofana, 2008:107).

5. **Kelas E**

   IP *address* kelas E masih bersifat percobaan. Jika empat bit pertama adalah 1111 maka IP *address* termasuk dalam kategori kelas E. Pemakaian IP *address* kelas E dicadangkan untuk keperluan ekseperimental (Sofana, 2008:107-108).

IP *address* kelas A, B, dan C hanya digunakan untuk *host-host* di LAN. Kita bebas menggunakan IP *address* ketiga kelas tersebut untuk keperluan jaringan lokal. Inilah yang disebut IP *address* privat. IP versi 4 memeiliki bentuk *client* seperti gambar 2.3



Gambar 2.3 *Header* IPv4

Header IPv4 akan dijelaskan lebih rinci pada tabel 2.1 (Rafiudin,2005:26)

Tabel 2.1 Rincian *Header* IPv4

Field	Keterangan
Version	Mengindikasikan versi IP. Oleh karena ini IPv4 maka nilainya diset 4.
Internet Header Length (IHL)	Merupakan panjang *header* internet (dalam 32 bit kata). *Size field* ini 4 bit.
Type of Service(TOS)	Layanan service yang diharapkan. *Size field* ini 8 bit yang

	terdiri atas bit-bit untuk prioritas, waktu *delay*, keluaran, dan karakteristik reliabilitas.
Total Length	Panjang total paket IPv4. *Size field* ini 16 bit.
Inditification	Nilai yang ditetapkan pengirim paket untuk membantu *reassembly* fragmen datagram. *Size field* ini 16 bit.
Flags	Flag-flag untuk proses fragmentasi. *Size field* ini 3 bit, tetapi hanya 2 yang ditetapkan untuk penggunaan saat ini.
Fragment Offset	Mengindikasikan posisi *fragment*. *Size field* ini 13 bit.
Time to Live (TTL)	Jumlah link maksimum di mana paket IPv4 berjalan sebelum dibuang.
Protocol	Mengidentifikasi *upper-layer protocol*. *Size field* ini 8 bit.
Header Checksum	Pengecekan error. *Size field* ini 16 bit.
Source Address	Menyimpan IP *addresshost* pengirim. *Size field* ini 32 bit.
Destination Address	Menyimpan IP *addresshost* tujuan. *Size field* ini 32 bit.
Options	Menyimpan satu atau lebih opsi IPv4.

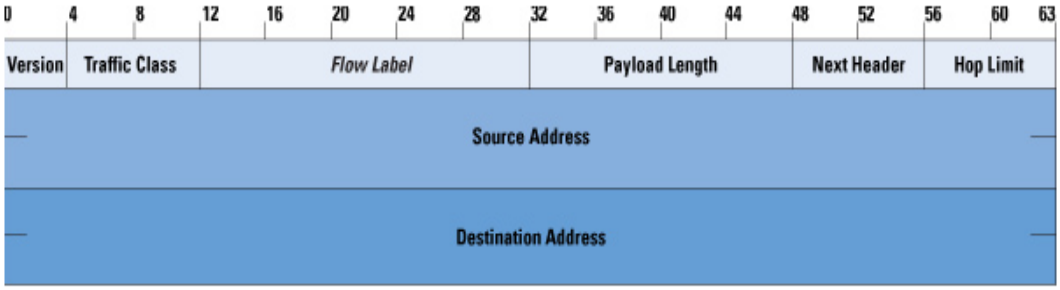
2.5.2 IPv6 (IP Versi 6)

IPv6 terkadang disebut *Next Generation Internet Protocol* IPng. Pertama ditawarkan oleh IPng *Area Directers* dari *Internet Engineering Task Force* (IETF) pada pertemuan tanggal 25 Juli 1994 sebagaimana didokumentasikan dalam RFC 1752, "*The Recommendation for the* IP *Next Generation Protocol*" (Rafiudin, 2007:1).

Perbedaan yang kontras antara IPv4 dan IPv6 adalah spasi *address*-nya.*Address-address* IPv6 berukuran jauh lebih besar dibandingkan *address* IPv4. Ukuran sebuah IPv6 sebesar 128 bit. Dengan spasi *address* sebasar 128 bit, jumlah *address* yang akan terbentuk sebanyak 2^{128} atau setara dengan $3.4x10^{38}$. Untuk IPv6, *address* sebesar 128 bit dibagi kedalam 8 blok, masing-masing blok berisi 16 bit yang

dikonversi ke 4-digit nomor heksadesimal dan dipisahkan dengan tanda titik dua (":"). Contoh *address* dari IPv6 adalah **3ffe:ffff:100:f101:0:0:0:1**.

Sekarang kita perhatikan elemen-elemen *header* IPv6. Ditemukan bahwa *header* IPv6 lebih simpel dibandingkan IPv4



Gambar 2.4 *Header* IPv6

Header IPv4 akan dijelaskan lebih rinci pada tabel 2.2 (Rafiudin,2005:29)

Tabel 2.2 Rincian *Header* IPv6

Field	Keterangan
Version	Mengindikasikan versi IP. Oleh karena ini IPv6 maka nilainya diset 6. *Size field* ini 4 bit.
Traffic class	Mengindikasikan kelas prioritas paket. *Size field* ini 8 bit.
Flow Label	Digunakan pengirim untuk memeberi etiket rangkaian paket-paket di mana ia meminta penanganan khusus oleh router-router IPv6. *Size field* ini 20 bit.
Payload Length	Mengindikasikan panjang data yang dibawa setelah IP *header*. *Size field* ini 16 bit.
Next *Header*	Mengindikasikan tipe *header* berikutnya. *Size field* ini 8 bit.
Hop Limit	Mengindikasikan jumlah *link* maksimum dimana paket IPv6 berjalan sebelum dibuang. *Size field* ini 8 bit.
Source Address	Menyimpan IP *address* (Ipv6 *address*) *host* pengirim. *Size field* ini 128 bit.
Destination Address	Menyimpan IP *address* (Ipv6 *address*) *host* tujuan. *Size field*

	ini 128 bit.

Ada tiga tipe IPv6 *addressing*, yaitu:

1. *Unicast Address*

   Unicast *address* merupakan jenis IP *address* yang digunakan untuk identifikasi sebuah interface saja. *Unicast address* akan menghasilkan beberapa kelompok *address*, yaitu :

   1. *Link Local Unicast Address*

      Merupakan jenis alamat yang mengizinkan sebuah komputer dapat berkomunikasi dengan komputer lainnya dalam satu subnet. Link local unicast *address* dapat dikenali jika bit-bit awalnya adalah 1111111010 atau prefix FE80::/64.

   2. *Site Local Unicast Address*

      Merupakan jenis alamat yang mengizinkan sebuah komputer dapat berkomunikasi dengan komputer lainnya delam sebuah intranet. *Site local unicast address* dapat dikenali jika bit-bit awalnya adalah 1111111011 atau prefix FEC0::/48.

   3. *Global Unicast Address*

      Dapat dianalogikan dengan public IP *address* pada IPv4. Alamat ini secara global dirujuk oleh *host-host* di internet dengan menggunakan proses *routing*. Global *unicast address* dapat dikenali jika bit-bit awal alah 001 atau prefix 2000::/3

2. *Multicast Address*

   Merupakan komunikasi-komunikasi diantara *host* tunggal dangan multi penerima. Multicast diimplementasikan dalam *one-to-many communication*. Cukup mudah mengenali *multicast address* pada IPv6 karena selalu dimulai dengan “FF”.

3. *Anycast Address*

   Anycast address ditujukan untuk multi *interface*. Paket-paket yang dialamatkan ke a*nycast address* diteruskan oleh infrastruktur *routing* ke salah satu *interface* terdekat. *Anycast* harus tidak ditujukan untuk *host-host* IPv6, melainkan hanya untuk *router-router* IPv6.

2.6 Parameter Kinerja Aplikasi Web

Parameter yang digunakan untuk mengukur kinerja aplikasi web *loading time*. *Load time* adalah waktu yang dibutuhkan untuk memuat konten atau sesuatu lain. Dalam bidang web, *loading time* adalah waktu yang dibutuhkan untuk memuat seluruh konten yang dimilikinya. *Loading time* web yang baik adalah 8 detik (Andhy,2012, Cara Optimalkan Kecepatan Loading Time,(On-line),http://andhyzer-zone.blogspot.com/2012/01/cara-optimalkan-kecepatan-loading-time.html,(diakses tanggal 12 Mei 2012).

BAB III
METODOLOGI

3.1 Gambaran Umum Tentang Sistem

Jaringan yang digunakan adalah jaringan lokal, gambar di bawah ini merupakan desain topologi jaringan yang digunakan untuk analisis kinerja aplikasi web



Gambar 3.1 Desain Topologi Jaringan Kinerja Aplikasi Web.

Topologi yang digunakan untuk pengujian menggunakan dua buah komputer, satu komputer digunakan sebagai web *server* dan satu komputer digunakan sebagai *client*. Dua buah komputer tersebut dihubungkan oleh sebuah *mikrotik routerboard*. Penulis menggunakan satu buah *router* karena penulis tidak menggunakan proses *routing* didalamnya.

Mikrotik routerboard digunakan sebagai *bandwidth management*. *Mikrotik* inilah yang nantinya akan memberikan aturan besar *bandwidth* yang akan digunakan. *Mikrotik* memiliki dua buah *interface* yaitu:

1. Ether1, *interface* yang terhubung dengan web *server*dengan alamat IPv6 statik.
2. Ether2, *interface* yang terhubung dengan jaringan lokal dengan alamat IPv6 statik

Web *server* terhubung dengan dua buah komputer *client* melalui *mikrotik routerboard*. Komputer ini menggunakan satu *interface* yaitu eth1 *interface* yang terhubung dengan jaringan lokal. Untuk komputer *client* sendiri masing-masing memiliki satu buah *interface*.

Tabel 3.1 Alokasi alamat IP

Nama Perangkat	Interface	Alamat IPv6
Web *Server*	eth1	2010::1/64
Mikrotik Routerboard	ether1	2010::2/64
	ether2	2011::2/64
Client	eth2	2011::1/64

3.2 Kebutuhan Perangkat Keras

Berdasarkan gambaran umum tentang sistem perangkat keras yang dibutuhkan adalah sebagai berikut:

1. *Mikrotik Routerboard*

   Adalah *platform hardware* yang dibuat oleh *mikrotik*. Terdapat RouterOS didalam perangkat ini. *Router* ini digunakan sebagai *gateway* menuju komputer *server* dan sebagai *bandwidth* management. Seri *routerboard* yang digunakan adalah RB 750 yang memiliki 5 buah *port ethernet* 10/100, dengan prosesor baru *Atheros* 400MHz. Sudah termasuk dengan lisensi level 4 dan adaptor 12V. Berikut detail spesifikasi RB 750 :

   a. Main storage 64 MB
   b. RAM 32 MB
   c. Switch chip 1
   d. Power jack 10-28V
   e. POE input 10-28V
   f. Dimensi 113x89x28mm
   g. Temperatur -40C .. +55C

2. Komputer *Server*

   Di dalam komputer ini terdapat web *server* dan Spesifikasi yang digunakan adalah :

   a. Prosesor 1.6 GHz
   b. Memori (RAM) 512 MB
   c. *Harddisk* dengan kapasitas 80 GB

3. Komputer *client*

   a. *Prosesor core2 duo* 2.10 GHz
   b. Memori (RAM) 2 GB
   c. *Harddisk* 250 GB

3.3 Kebutuhan Perangkat Lunak

1. *Mikrotik Router*OS™

   *MikroTik Router*OS™ adalah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer manjadi *Router network* yang handal, mencakup berbagai fitur yang dibuat untuk ip *network* dan jaringan *wireless*, cocok digunakan oleh ISP dan *provider hotspot*. Kita dapat mengunduh file *image mikrotik Router*OS dari website resmi mikrotik, www.mikrotik.com. Namun, file image ini merupakan versi *trial mikrotik* yang hanya dapat dalam waktu 24 jam saja. Untuk dapat menggunakannya secara *full time*, anda harus membeli lisensi *key* dengan catatan satu lisensi hanya untuk satu *harddisk*.

2. CentOS 5.7

   Community Enterprise Operating System (CentOS) 5.7 digunakan sebagai sistem operasi pada komputer *server*.

3. Windows 7

   Sistem operasi inilah yang akan digunakan oleh komputer *client*.

4. *Winbox*

   Winbox adalah sebuah utilitas yang digunakan untuk melakukan *remote* ke *server mikrotik* kita dalam mode GUI. Jika untuk mengkonfigurasi *mikrotik*

dalam *text mode* melalui PC itu sendiri, maka untuk mode GUI yang menggunakan *winbox* ini kita mengkonfigurasi *mikrotik* melalui komputer *client*. *Winbox* dapat dijalankan pada sistem operasi windows dan macOS dengan bantuan *wine*.

5. *Lampp*

   Perangkat lunak bebas yang digunakan sebagai web *server* pada sistem operasi linux yang dapat melayani halaman web yang dinamis. Lampp terdiri atas program *Apache* HTTP *Server*, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan *Perl*. Program ini tersedia dalam GNU *General Public License.*

6. Aplikasi Web

   Aplikasi web yang digunakan dalam penelitian ini antara lain :

   a. CMS *wordpress*

      Wordpress adalah sebuah aplikasi *open source* yang sangat populer digunakan sebagai mesin blog. Wordpress dibangun dengan bahasa pemrograman PHP dan basis data. Selain sebagai blog, wordpress juga mulai digunakan sebagai CMS (*Content Management System*) karen kemampuannya untuk dimodifikasi dan digunakan sebagai kebutuhan. Web ini berisi tulisan-tulisan.

   b. TomatoCart

      Adalah suatu CMS untuk *shopping cart*. Aplikasi ini bersifat *open source* yang sangat power full dan gratis karena semua yang diperlukan untuk toko *online* sudah tersedia di dalamnya, antara lain keranjang belanja, ongkos kirim, pajak, report penjualan, report user *online*, pembayaran via online (*credit card, paypal, american express*), diskon khusus, dan lain-lain. Isi dari web ini adalah berupa gambar-gambar produk yang dijual.

c. *Elgg*

Adalah aplikasi jejaring sosial yang bersifat *open source* yang menyediakan komponen individu dan organisasi yang dibutuhkan untuk menciptakan lingkungan sosial. *Elgg* didirikan pada tahun 2004 oleh Ben Werdmuller von Elgg dan David Tosh. *Elgg* bebas untuk diunduh dan digunakan. Elg berlisensi di bawah GNU *General Public License* (GPL) dan diterbitkan oleh *Free Software Fondation*. *Elgg* berjalan pada LAMP (Linux, Apache, MySQL, dan PHP). Isi aplikasi web ini sama seperti web jejaring sosial biasanya, yaitu tulisan-tulisan beserta foto-foto sebagai identitas pengguna.

d. i-Tube

Itube merupakan website multimedia yang berisi video *streaming* yang diciptakan oleh mahasiswa Teknik Informatika UII angkatan 2009 bernama Widi widayat. Isi dari web ini adalah full video.

e. SEIP

Adalah suatu web portal berita standar yang dibuat oleh Timur Sahadewa mahasiswa angkatan 2009 jurusan Teknik Informatika UII. Web ini berisi informasi berupa tulisan-tulisan berita tentang SEIP.

7. *Wireshark* 1.6.6

Aplikasi ini digunakan untuk menghitung *loading time* pada saat pengaksesan web.

3.4 Pembangunan Sistem

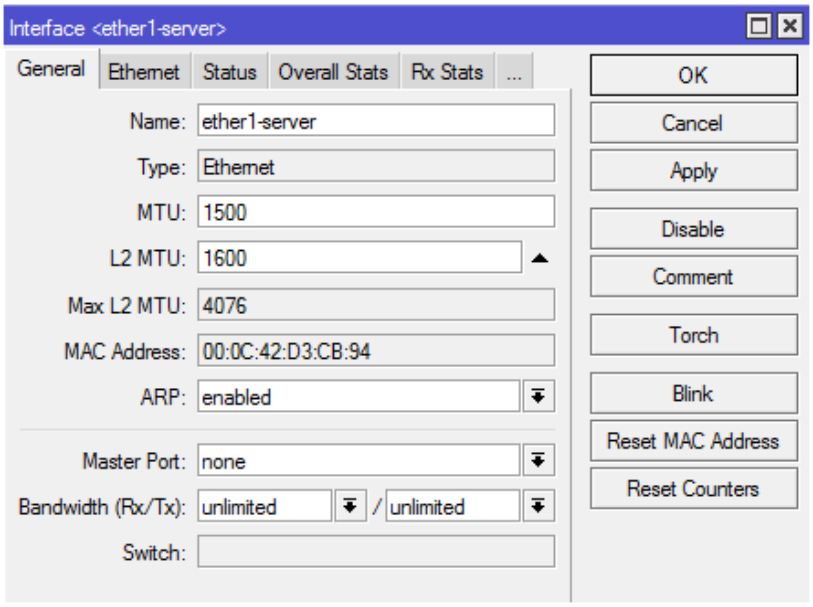
Langkah selanjutnya adalah pembangunan sistem yaitu yang membahas tentang instalasi perangkat lunak serta konfigurasi pada komputer *server* dan *client*.

3.4.1 Konfigurasi *Mikrotik Routerboard* RB750

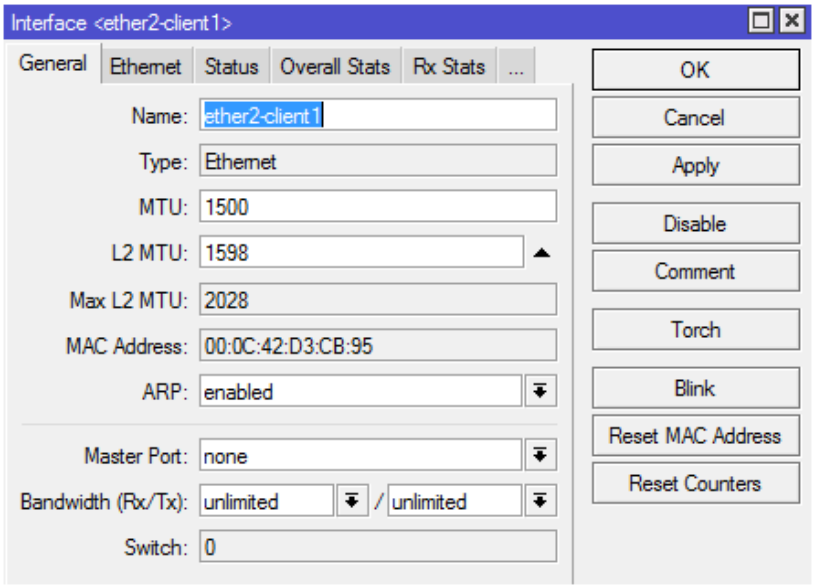
Ada beberapa hal yang harus dikonfigurasi pada *Routerboard* :

1. Konfigurasi *Interface*

   Mendifinisikan nama *interface* yang digunakan. Konfigurasi seperti gambar di bawah ini.

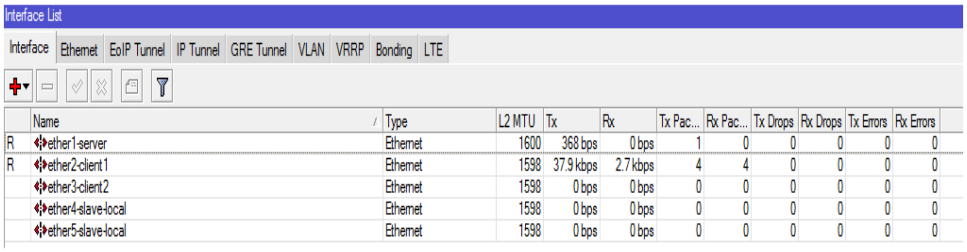


Gambar 3.2 Konfigurasi *Interface ether1*



Gambar 3.3 Konfigurasi *Interface ether2*

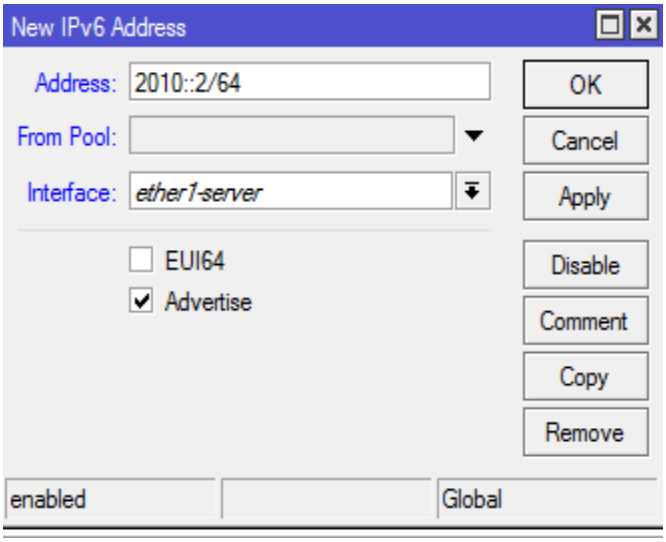
Maka hasil konfigurasi kedua *interface* tersebut adalah seperti gambar di bawah ini



Gambar 3.4 *Interface List*

2. Konfigurasi Alamat IP Pada *Routerboard*

   Konfigurasi di bawah ini merupakan konfigurasi **Ipv6** yaitu dengan cara meng-klik tab IPv6 – *addresses*, kemudian tambah alamat maka akan muncul tampilan seperti gambar 3.5 di bawah ini.



Gambar 3.5 Konfigurasi Alamat IPv6

Keterangan:

Address : Alamat koneksi menuju web *server* (2010::2/64).

Interface : *Interface* yang digunakan (*ether*1-*server*) untuk koneksi menuju web *server*.

Untuk konfigurasi alamat IP menuju *client*, lakukan konfigurasi sama seperti seperti konfigurasi alamat menuju web *server*. Ganti alamat dengan 2011::1/64 dan *interface ether*2-*client*1.

3.4.2 Konfigurasi IP *Address* Komputer

Berikut ini adalah cara konfigurasi IP *address* statik pada sistem operasi linux CentOS dan Windows. Berikut ini adalah konfigurasi untuk IPv4.

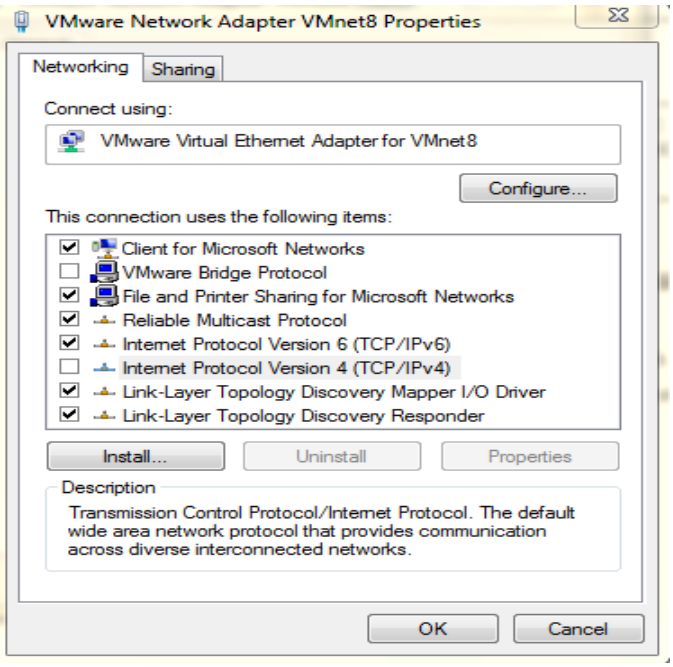
1. Konfigurasi AlamatIP versi 4 pada CentOS

```
# ifconfig eth0 [alamat ipv4/netmask]
# route add default gw [alamat gw ipv4]
```

Untuk versi 6 nya seperti berikut

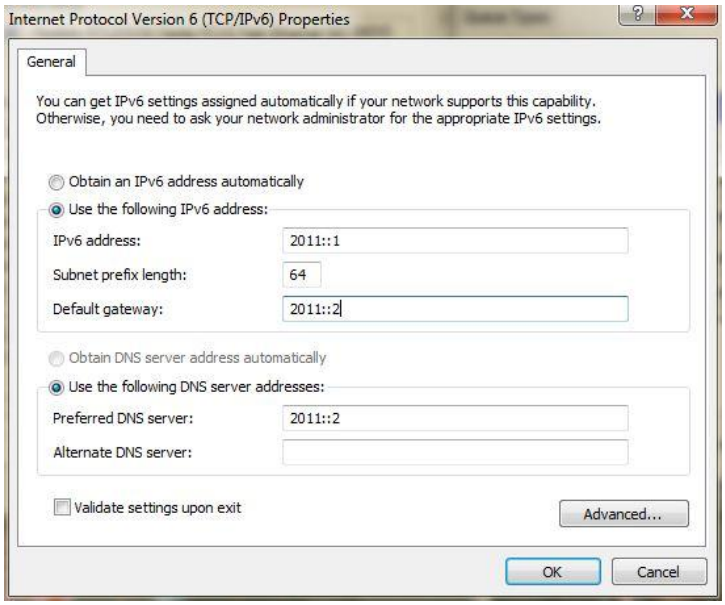
```
# ifconfig eth0 inet6 add [alamat ipv6/prefix]
# route --inet6 add default gw [alamat gw ipv6]
```

2. Konfigurasi Alamat IP pada Windows
   a. Klik menu *control panel* - *Network and Sharing Center*, klik *View status*, kemudian klik *properties*.
   b. Setelah itu ada pilihan versi IP yang akan kita gunakan. Karena pada penelitian ini menggunakan IPv6, maka *uncheck* pada IPv4. Untuk lebih jelasnya dapat dilihat pada gambar 3.6.



Gambar 3.6 *Network Connection Properties*

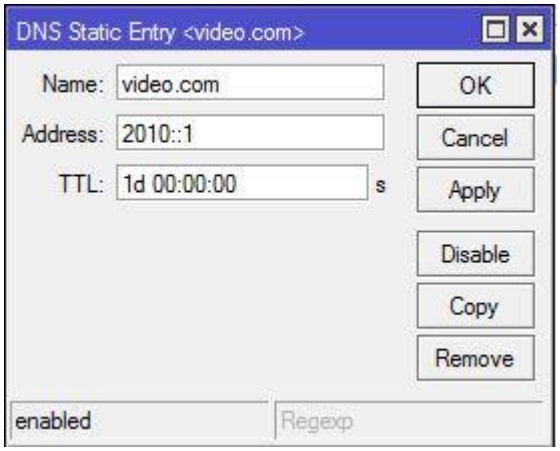
c. Kemudian isikan IP *address* yang kita gunakan, *subnet prefix length*, *gateway*, dan DNS *server* seperti gambar 3.7 dibawah ini.



Gambar 3.7 Konfigurasi Alamat IPv6

3.4.3 Konfigurasi DNS

Konfigurasi DNS dilakukan pada *mikrotik routerboard* dengan cara meng-klik tab IP – DNS – *Static*, kemudian tambahkan alamat maka akan muncul tampilan seperti gambar 3.8 di bawah ini



Gambar 3.8 DNS IPv6

3.4.4 Konfigurasi Web *Server*

Web *server* yang dipakai adalah *Xampp-Linux*-1.7.4. Cara menginstall *xampp* linux adalah menggunakan perintah berikut

```
# tar -xzvf xampp-linux-1.7.4 -C /opt
```

Kemudian setelah ekstraksi selesai jalankan *service xampp* dengan perintah berikut

```
# /opt/lampp/lampp start
```

Setelah itu lakukan konfigurasi pada file /opt/lampp/etc/httpd.conf seperti gambar 3.9 berikut.



Gambar 3.9 Konfigurasi httpd.conf

Keterangan :

ServerName : nama *server* yang dikirim.

ServerAlias : nama lain *server* yang dikirim.

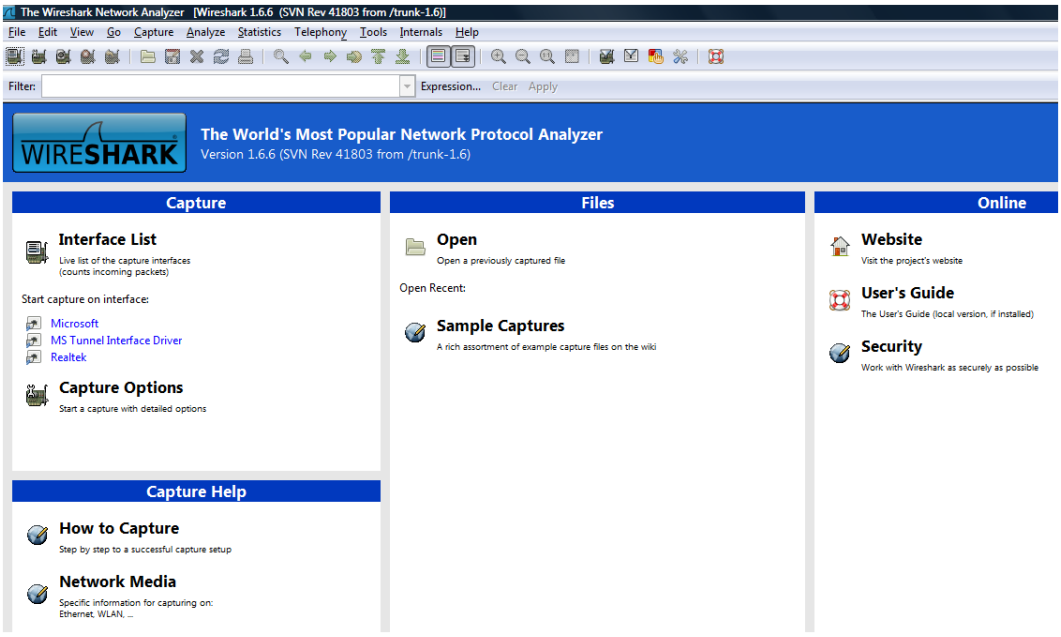
DocumentRoot : letak direktori web yang akan diterbitkan sebagai web *page*.

Dan pada bagian **Listen** ganti dengan

```
Listen [2010::1]:80
```

3.4.5 Installasi *Wireshark*

Klik dua kali pada *wireshark.exe* kemudian ikuti langkah instalasi. Setelah instalasi selesai maka akan muncul tampilan awal seperti gambar 3.10 berikut ini.



Gambar 3.10 Tampilan Awal *Wireshark*

3.4.6 Web *Stress Tools*

Aplikasi ini digunakan untuk melihat berapa ukuran aplikasi web yang kita load. Untuk hasil aplikasi yang kita *load*, maka dapat di lihat pada gambar 3.11 di bawah ini

User No.	Clicks	Hits	Errors	Avg. Click Time [ms]	Bytes	kbit/s	Cookies
1	1	1	0	21.765	136.871	50,31	

Gambar 3.11 Ukuran Aplikasi Web

Bytes merupakan ukuran web yang di *load*.

3.5 Pengambilan Data

Setelah melakukan instalasi dan konfigurasi maka langkah selanjutnya adalah melakukan pengujian. Pengujian dilakukan dalam beberapa kondisi, yaitu :

1. Komputer *client* menggunakan Windows7 tanpa dibatasi *bandwidth.*

2. Pengujian tiap-tiap *bandwidth* dilakukan sebanyak 10 kali percobaan dengan menggunaka *browser firefox* 4.0. Kemudian *loading time* dalam mengakses web satu halaman penuh dengan menggunakan *wireshark*.

3.6 Metode Analisis

Tahap ini merupakan tahap bagaimana cara pengujian data yang didapat. Pada analisis kali ini menggunakan statistis deksriptif dan induktif serta memberikan hipotesis. Yang diuji adalah waktu dalam mengakses satu halaman penuh aplikasi web yang dipakai.

3.6.1 Statistik Deskriptif

Statistik deskriptif yaitu berkenaan dengan mengumpulkan data, mengolah data, menyajikan data (dalam bentuk tabel, grafik, maupun diagram), dan menganalisis data.

3.6.2 Statistik Induktif

Statistik induktif yaitu berkenaan dengan mengumpulkan data, mengolah data, menyajikan data, menganalisis data, dan kemudian mengambil kesimpulan. Hasil analisis yang di dapat akan di uji dengan *One Way Annova* yaitu dikarenakan yang dibandingkan lebih dari 2 data dan datanya terdistribusi normal. *One Way Annova* juga digunakan untuk mengetahui apakah ada perbedaan antar *loading time* aplikasi web yang satu dengan yang lainnya.

3.6.3 Hipotesis

Hipotesis merupakan dugaan sementara yang dipakai dalam penelitian ini. Karena kinerja aplikasi web tergantung pada konten-konten yang ada di dalam aplikasi web, maka web multimedia lah yang memiliki *loading time* yang lebih lama dari pada web yang lainnya dikarekan aplikasi web tersebut berisi video-video.

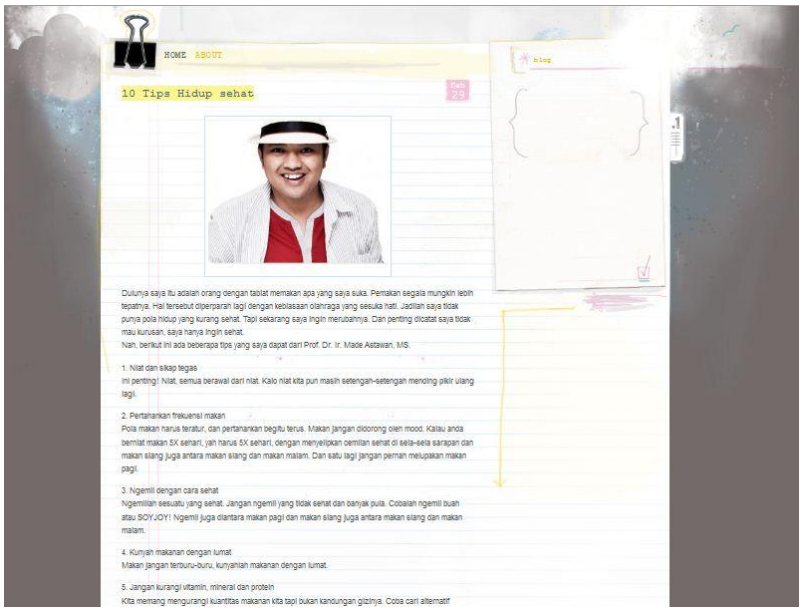
BAB IV
HASIL DAN PEMBAHASAN

Bab ini menampilkan hasil pengujian yang telah dilakukan. Agar lebih mudah dimengerti, bab ini dibagi menjadi beberapa subbab sesuai dengan aplikasi web yang digunakan dalam tugas akhir ini.

4.1 Hasil Pengujian Berdasarkan Aplikasi Web

4.1.1 *Blog*

Gambar berikut ini merupakan gambar aplikasi web *blog*.



Gambar 4.1 *Blog*

Berikut ini merupakan hasil *loading time* dengan sepuluh kali percobaan pada aplikasi web blog yang memiliki ukuran sebesar 136,871 KB.

Tabel 4.1 *Loading Time Blog*

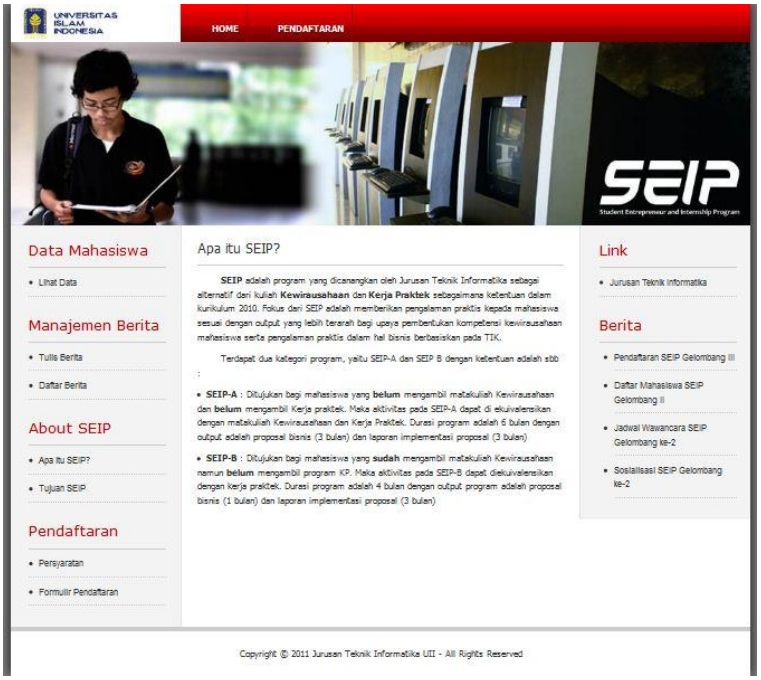
Uji ke	Loading Time (ms)
1	452.09
2	1055.911
3	804.683
4	836.219

5	1053.671
6	857.605
7	760.973
8	831.435
9	802.196
10	830.388
Rata-rata	828.5171

Dari hasil sepuluh kali percobaan, maka aplikasi web *blog* memiliki rata-rata *loading time* sebesar 828,5171 ms.

4.1.2 Portal Berita

Gambar berikut ini merupakan gambar aplikasi web portal berita.



Gambar 4.2 Portal Berita

Berikut ini merupakan hasil *loading time* dengan sepuluh kali percobaan pada aplikasi web portal berita yang memiliki ukuran sebesar 12,09 KB.

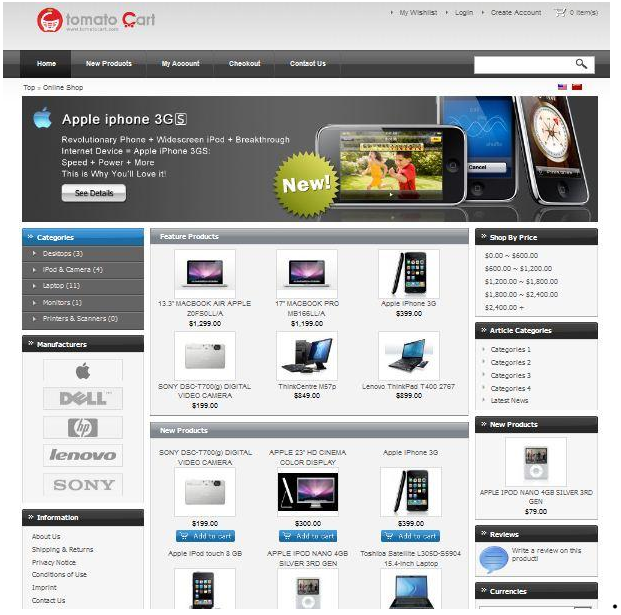
Tabel 4.2 *Loading Time* Portal Berita

Uji ke	Loading Time (ms)
1	643.382
2	95.087
3	268.237
4	270.966
5	300.012
6	103.1
7	819.219
8	257.774
9	226.617
10	306.499
Rata-rata	329.0893

Dari hasil sepuluh kali percobaan, maka aplikasi web portal berita memiliki rata-rata *loading time* sebesar 329,0893 ms.

4.1.3 *Shopping Cart*

Gambar berikut ini merupakan gambar aplikasi web *shopping cart*



Gambar 4.3 *Shopping Cart*

Berikut ini merupakan hasil *loading time* dengan sepuluh kali percobaan pada aplikasi web *shopping cart* yang memiliki ukuran sebesar 27,195 KB.

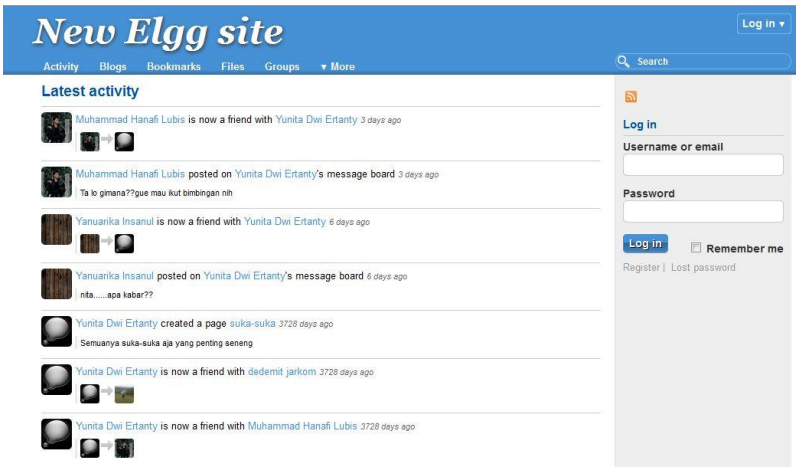
Tabel 4.3 *Loading Time Shopping Cart*

Uji ke	Loading Time (ms)
1	861.536
2	716.934
3	769.362
4	507.519
5	510.312
6	502.958
7	533.28
8	766.544
9	834.709
10	527.732
Rata-rata	653.0886

Dari hasil sepuluh kali percobaan, maka aplikasi *Shopping Cart* memiliki rata-rata loading time sebesar 653,0886 ms.

4.1.4 *Social Networking*

Gambar berikut ini merupakan gambar aplikasi web *social networking*



Gambar 4.4 *Social Networking*

Berikut ini merupakan hasil *loading time* dengan sepuluh kali percobaan pada aplikasi web *social networking* yang memiliki ukuran sebesar 39,870 KB.

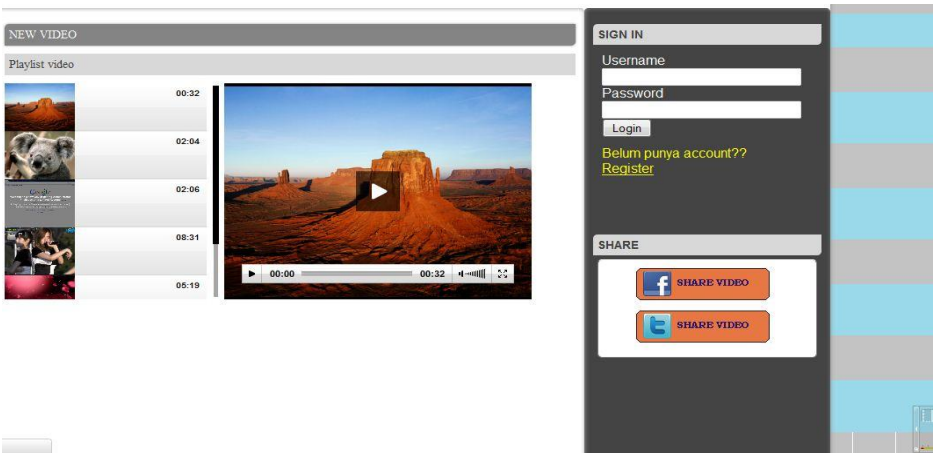
Tabel 4.4 *Loading Time Social Networking*

Uji ke	Loading Time (ms)
1	743.143
2	775.108
3	776.686
4	766.099
5	756.141
6	723.597
7	772.97
8	764.843
9	757.104
10	762.752
Rata-rata	759.8443

Dari hasil sepuluh kali percobaan, maka aplikasi *social networking* memiliki rata-rata *loading time* sebesar 759,8443 ms.

4.1.5 Multimedia

Gambar berikut ini merupakan gambar aplikasi web portal berita.



Gambar 4.5 Multimedia

Berikut ini merupakan hasil *loading time* dengan sepuluh kali percobaan pada aplikasi web multimedia yang memiliki ukuran sebesar 13,98 KB.

Tabel 4.5 *Loading Time* Multimedia

Uji ke	Loading Time (ms)
1	351.824
2	355.152
3	393.092
4	388.766
5	359.507
6	400.46
7	369.302
8	395.836
9	354.522
10	365.422
Rata-rata	373.3883

Dari hasil sepuluh kali percobaan, maka aplikasi multimedia memiliki rata-rata *loading time* sebesar 373,3883 ms.

4.1.6 *Buffering Time* Video

Berikut ini merupakan hasil *buffering time* pada video *streaming* berukuran 42703 KB, ekstensi mp4, resolusi 1280x720, dan durasi 04:01 dengan sepuluh kali percobaan.

Tabel 4.6 *Buffering Time* video

Uji ke	Loading Time (ms)
1	31971.9
2	41976.729
3	31805.359
4	38881.307
5	28544.087
6	33095.065

7	34737.82
8	30377.23
9	27647.99
10	31.079.752
Rata-rata	33226.39

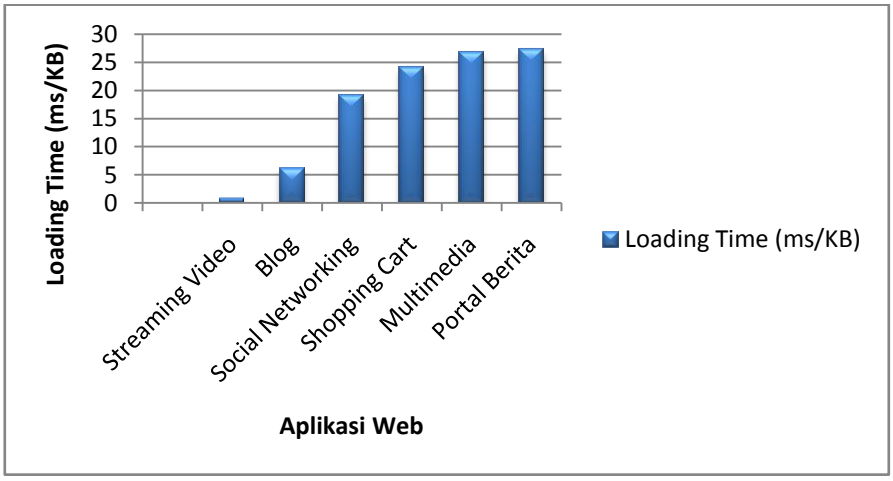
Dari hasil sepuluh kali percobaan, maka *buffering time* pada video *streaming* memiliki rata-rata sebesar 33226.39 ms.

4.2. Penyajian Data

Setelah menyajikan data berdasarkan aplikasi web, maka penulis menyajikan dengan diagram. Penyajian diagram berikut ini merupakan hasil dari rata-rata *loading time* dibagi dengan besar aplikasi web.

Tabel 4.7 Rata-rata *Loading Time* Aplikasi Web

Aplikasi Web	Loading Time (s)	Ukuran Web (KB)	Loading Time/ukuran web (ms/KB)
Blog	828.5171	136.871	6.0533
Portal Berita	329.0893	12.09	27.2200
Shopping Cart	653.0886	27.196	24.0150
Social Networking	759.8443	39.87	19.0580
Multimedia	373.3883	13.98	26.7087
Streaming Video	33011.7239	42703	0.7731



Gambar 4.6 Diagram *Loading Time* Aplikasi Web

Dari hasil dapat kita lihat bahwa *loading* tercepat terdapat pada video *streaming* yaitu sebesar 0.7731 ms/KB. Kemudian pada urutan ke dua ada aplikasi web *blog*. *Loading time* pada aplikasi *blog* sebesar 6,0533 ms/KB. Urutan ke tiga adalah aplikasi *social networking* yaitu sebesar 19,0580 ms/KB, urutan ke empat adalah aplikasi *shopping cart* sebesar 24,0150 ms/KB, urutan ke lima adalah aplikasi Multimedia sebesar 26,7087 ms/KB, dan yang paling memiliki *loading time* terlama yaitu portal berita sebesar 27,2200 ms/KB.

Setelah data disajikan, agar lebih terlihat jelas apakah ada perbedaan dan kesamaan *loading time* aplikasi web dari rata-rata *loading time* yang ada maka data tersebut di uji dengan pengujian statistika menggunakan metode komparasi *One Way Annova*. Metode komparasi tersebut akan memberikan hasil berdasarkan rata-rata *loading time* dari hasil pengujian.

Dan berikut ini merupakan hasil pengujian statistika dengan menggunakan komparasi *One Way Annova*.

Tabel 4.8 Uji Distribusi Normal

		data
N		60
Normal Parameters[a]	Mean	1.727304E1
	Std. Deviation	1.2967885E1
Most Extreme Differences	Absolute	.160
	Positive	.124
	Negative	-.160
Kolmogorov-Smirnov Z		1.241
Asymp. Sig. (2-tailed)		.092

a. Test distribution is Normal.

Tabel di atas merupakan tabel hasil uji normalitas data *loading time* dari 6 aplikasi web. Karena hasilnya (0.092) >0.05 maka data terdistribusi normal sehingga kita dapat melanjutkan ke uji *One Way Annova*.

Tabel 4.9 Uji *One Way Annova*

data					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	6410.473	5	1282.095	19.717	.000
Within Groups	3511.324	54	65.025		
Total	9921.797	59			

Tabel di atas merupakan tabel hasil uji *One Way Annova* data *loading time* dari 6 aplikasi web. Karena nilai p (0.000) < 0.05 maka ada minimal 1 rata-rata *loading time* aplikasi web yang tidak sama. Untuk melihat aplikasi mana saja yang memiliki rata-rata yang berbeda maka di uji dengan *Post Hoc*.

Tabel 4.10 *Post Hoc*

					95% Confidence Interval	
(I) aplikasi	(J) aplikasi	Mean Difference (I-J)	Std. Error	Sig.	Lower Bound	Upper Bound
blog	portal_berita	-21.1666900*	3.6062313E0	.000	-28.396748	-13.936632
	shopping_cart	-17.9617600*	3.6062313E0	.000	-25.191818	-10.731702
	social_networking	-13.0048000*	3.6062313E0	.001	-20.234858	-5.774742
	multimedia	-20.6554800*	3.6062313E0	.000	-27.885538	-13.425422
	streaming_video	5.4700300	3.6062313E0	.135	-1.760028	12.700088
portal_berita	blog	21.1666900*	3.6062313E0	.000	13.936632	28.396748
	shopping_cart	3.2049300	3.6062313E0	.378	-4.025128	10.434988
	social_networking	8.1618900*	3.6062313E0	.028	.931832	15.391948
	multimedia	.5112100	3.6062313E0	.888	-6.718848	7.741268
	streaming_video	26.6367200*	3.6062313E0	.000	19.406662	33.866778
shopping_cart	blog	17.9617600*	3.6062313E0	.000	10.731702	25.191818
	portal_berita	-3.2049300	3.6062313E0	.378	-10.434988	4.025128
	social_networking	4.9569600	3.6062313E0	.175	-2.273098	12.187018
	multimedia	-2.6937200	3.6062313E0	.458	-9.923778	4.536338
	streaming_video	23.4317900*	3.6062313E0	.000	16.201732	30.661848
social_networking	blog	13.0048000*	3.6062313E0	.001	5.774742	20.234858
	portal_berita	-8.1618900*	3.6062313E0	.028	-15.391948	-.931832
	shopping_cart	-4.9569600	3.6062313E0	.175	-12.187018	2.273098
	multimedia	-7.6506800*	3.6062313E0	.038	-14.880738	-.420622
	streaming_video	18.4748300*	3.6062313E0	.000	11.244772	25.704888
multimedia	blog	20.6554800*	3.6062313E0	.000	13.425422	27.885538
	portal_berita	-.5112100	3.6062313E0	.888	-7.741268	6.718848

	shopping_cart	2.6937200	3.6062313E0	.458	-4.536338	9.923778
	social_networking	7.6506800*	3.6062313E0	.038	.420622	14.880738
	streaming_video	26.1255100*	3.6062313E0	.000	18.895452	33.355568
streaming_video	blog	-5.4700300	3.6062313E0	.135	-12.700088	1.760028
	portal_berita	-26.6367200*	3.6062313E0	.000	-33.866778	-19.406662
	shopping_cart	-23.4317900*	3.6062313E0	.000	-30.661848	-16.201732
	social_networking	-18.4748300*	3.6062313E0	.000	-25.704888	-11.244772
	multimedia	-26.1255100*	3.6062313E0	.000	-33.355568	-18.895452

Dari hasil di atas, maka ada tiga kondisi:

1. Jika nilai *lower bound* ke nilai *upper bound* tidak lewat nol dan positif (+), maka (I)aplikasi > (J)aplikasi.
2. Jika nilai *lower bound* ke nilai *upper bound* tidak lewat nol dan negatif (-), maka (I)aplikasi < (J)aplikasi.
3. Jika nilai *lower bound* ke nilai *upper bound* lewat nol, maka (I)aplikasi = (J)aplikasi.

Maka dari hasil *Post Hoc* dapat disimpulkan bahwa *loading time* portal berita sama dengan *loading time shopping cart* sama dengan *loading time* multimedia. Dan ketiga aplikasi web tersebut *loading time* nya lebih besar dari web *blog*. Dan web *blog loading time* nya sama dengan video *streaming*.

Dari semua pengujian setiap aplikasi web memiliki *loading time* yang berbeda-beda. Hal tersebut dikarenakan ukuran dan konten-konten yang ada pada aplikasi web tersebut berbeda-beda. Gambar-gambar yang ada juga ekstensinya berbeda-beda serta *script* yang di pakai tiap masing-masing aplikasi web berbeda-beda.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis dari masing-masing aplikasi web dengan parameter *loading time,* maka dapat ditarik kesimpulan sebagai berikut:

1. Kecepatan *loading time* tergantung pada ukuran aplikasi web dan konten-konten yang ada di dalamnya.
2. Jenis *script* dan tipe gambar dapat berpengaruh dalam kecepatan *loading time*.

5.2 Saran

Berdasarkan keterbatasan yang terjadi dalam penelitian ini, maka ada beberapa saran yang perlu diperhatikan, yaitu:

1. Sebaiknya dalam penelitian membandingkan kinerja aplikasi web menggunakan *hardware* yang memiliki spesifikasi yang lebih besar.
2. Sebaiknya dalam penelitian selanjutnya dapat membandingkan kinerja aplikasi web dengan menggunakan internet protokol yang berbeda ditambah dengan implementasi *tunneling*.
3. Sebaiknya dilakukan penelitian lebih detail lagi bagaiman kontribusi masing-masing teks, gambar, dan video pada *loading time*.

DAFTAR PUSTAKA

Anonim,2006-2011,Red Hat/CentOs IPv6 Network Configuration,(On-line) http://www.cyberciti.biz/faq/rhel-redhat-fedora-centos-ipv6-network-configuration/, (diakses tanggal 15 Januari 2012).

Anonim,2011,How to Prepare IPv6 World Day, (On-line), http://ipv6.redhat.com/howto, (diakses tanggal 15 Januari 2012).

Andhy,2012, Cara Optimalkan Kecepatan Loading Time,(On-line),http://andhyzer-zone.blogspot.com/2012/01/cara-optimalkan-kecepatan-loading-time.html,(diakses tanggal 12 Mei 2012).

Laboratorium Sistem dan Jaringan Komputer.2011.Gambar 2.2 Alur Request-response Antara *User Agent* dengan *Server*. Yogyakarta, Universitas Islam Indonesia.

Laboratorium Sistem dan Jaringan Komputer.2011.Modul Praktikum Jaringan Komputer. Yogyakarta, Universitas Islam Indonesia.

Rafiudin,Rahmat.2005.IPV6 Addressing.Jakarta, PT Elex Media Komputindo.

LAMPIRAN

1. Lembar Penilaian Presentasi Kemajuan Tugas Akhir
2. CD