ANALISA DAN IMPLEMENTASI IPV 6 TUNNEL BROKER UNTUK INTERKONEKSI ANTARA IPV6 DAN IPV4

Reko Artondo (L2F301465)
Jurusan Teknik Elektro Fakultas Teknik
Universitas Diponegoro
artondo@telkom.net

ABSTRAK

Sejak pelepasan IPv4 pada awal tahun 90-an, pengalamatan 32 bit IP lama kelamaan akan habis. Dalam kurung waktu yang tidak lama untuk menambah pengalamatan IP telah datang yang disebut dengan IPv6 atau IP Next Generation.

IPv6 sebagai standard baru harus mampu berinterkoneksi dengan IPv4 yang sudah umum digunakan. Karena pada dasarnya IPv6 tidak kompatibel dengan IPv4 maka diperlukan suatu mekanisme tertentu agar IPv6 ini dapat berinterkoneksi dengan IPv4. Mekanisme dasar dari IPv4 ke IPv6 diantaranya, Dual IP Stack, Tunneling, Translasi protokol. Sistem tunneling adalah sistem yang menghubungkan antara jaringan IPv4 ke jaringan IPv6 dengan menggunakan tunnel broker.

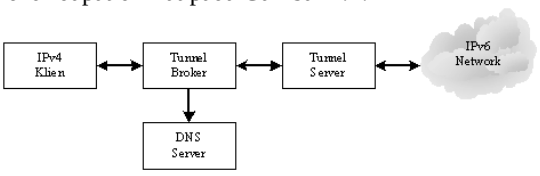
Pada tugas akhir ini akan dibuat IPv6 tunnel broker untuk interkoneksi antara IPv6 dan IPv4 yang meliputi perancangan jaringan IPv4, IPv6 dan PC sebagai tunnel broker, tunnel server dan DNS server. Selanjutnya diuji dan dianalisa sehingga diperoleh hasil suatu sistem yang dapat digunakan untuk komunikasi data dengan jaringan IPv4,IPv6 atau gabungan keduanya.

I. Pendahuluan

Pada sistem pengalamatan internet protokol versi 4 (IPv4), jumlah alamat yang terdapat hanya sebesar 4,295 x 10^9 alamat. Jumlah ini sangat tidak sesuai dengan jumlah perangkat yang harus diberikan alamat dalam jaringan internet. Internet protokol versi 6 (IPv6) atau sering disebut sebagai *IP Next Generation* (IPng) merupakan standard baru dalam bidang internet yang mampu meningkatkan kapasitas pengalamatan yang ada pada IPv4. Peningkatan kapasitas alamat tersebut terjadi karena pengunaan format alamat 128 bit, sehingga alamat total yang dapat disediakan lebih dari 3 x 10^{38} alamat. Selain itu masih ada banyak kelebihan lain IPv6 dibandingkan IPv4 yang sudah umum digunakan. Karena pada dasarnya IPv6 tidak kompatibel dengan IPv4 maka memerlukan suatu mekanisme transisi dasar dari IPv4 ke IPv6 yang ada diantaranya :

1. *Dual IP Stack*
2. *Tunneling*
3. Translasi Protokol

Dari ketiga mekanisme diatas, mekanisme *tunneling* dipakai sebagai obyek penilitian, karena pada dasarnya hanya menyediakan saluran (*tunnel*) bagi paket IPv6 melalui IPv4. Salah satu sistem *tunneling* yaitu IPv6 *tunnel broker* dimana *tunnel* diaktifkan secara otomatis oleh *tunnel broker* kepada *dual stack host* IPv6/IPv4 yang terisolasi dengan jaringan IPv6 yang lain agar bisa berhubungan dengan jaringan IPv6 tersebut, melalui jaringan IPv4 yang sudah ada. Arsitektur IPv6 *tunnel broker* dapat dilihat pada Gambar 1.1:



Gambar 1.1 Arsitektur Tunnel Broker

Tunnel broker merupakan tempat koneksi *tunnel clien* untuk melakukan registrasi dan aktivasi *tunnel*, dengan menggunakan jaringan IPv4. Sedangkan *tunnel server* merupakan *router dual stack* (IPv6 dan IPv4) yang terhubung ke baik jaringan internet IPv4 maupun jaringan IPv6.

Dalam mekanisme *tunneling* paket data yang dikirimkan oleh suatu *host* akan menjadi lebih besar karena adanya proses enkapsulasi (pembungkusan) paket. Untuk pengiriman paket yang lebih besar maka dibutuhkan waktu pengiriman yang lebih besar juga.

1.2 Tujuan

Tujuan dari Tugas Akhir ini adalah untuk merancang dan mengimplementasikan IPv6 *tunnel broker* untuk interkoneksi antara jaringan IPv6 dengan IPv4, dengan cara menyediakan koneksi antara *host* dan jaringan IPv6 yang saling terisolasi melalui jaringan IPv4 yang sudah ada.

1.3 Batasan Masalah

Interkoneksi antara IPv4 dan IPv6 dalam bentuk mekanisme *tunneling* yaitu menyediakan koneksi IPv6 melalui IPv4 (IPv6 *over* IPv4) dengan sistem IPv6 *tunnel broker*.

II. KONSEP IPV4, IPV6 DAN TUNNEL BROKER

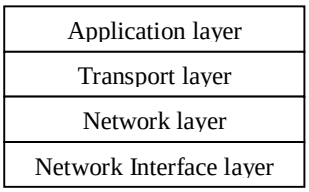
Pada bab II akan dibahas teori mengenai *internet protocol* (IP) dari versi 4 sampai dengan 6 dan sistem *tunneling* agar 2 jaringan berbeda bisa berhubungan. Mekanisme kerja dari *tunnel broker* beserta *tunnel* server dan DNS server.

2.1 TCP dan IP

TCP dan IP merupakan salah satu standar protokol yang dirancang untuk melakukan fungsi-fungsi komunikasi data dalam jaringan internet. TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dalam komunikasi data. Dengan prinsip ini maka tugas masing-masing protokol menjadi jelas dan sederhana, sehingga mudah untuk diimplementasikan di seluruh perangkat dan perangkat lunak jaringan dan juga mudah dalam melakukan proses *trouble shooting*. Karena beberapa kelebihan yang dimiliki protokol TCP/IP ini,

maka saat ini TCP/IP lebih banyak digunakan dengan standard protokol yang lain [5].

Arsitektur TCP/IP dapat dimodelkan dalam empat lapisan TCP/IP, yaitu *network interface layer*, *network layer*, *transport layer* dan *application layer*.



Gambar 2.1 Arsitektur Protokol TCP/IP

Dalam proses pengiriman data antar layer, setiap layer akan menganggap informasi yang datang dari layer sebelumnya sebagai data, sehingga ia akan menambahkan informasi miliknya pada data tersebut. Begitu juga sebaliknya, jika ia menerima data yang dianggap *valid* maka ia akan melepas informasi tersebut.

Network interface layer merupakan lapisan terbawah yang bertanggung jawab untuk mengirim dan menerima data ke dan dari media fisik. Oleh karena protokol dalam layer ini harus mampu merubah bit-bit informasi menjadi sinyal listrik. Contoh dari protokol dalam layer ini adalah PPP, SLIP dan Ethernet [5]. PPP (Point to Point Protocol) adalah protocol yang biasa dipakai pada komunikasi router to router dan host to network diatas jaringan *asynchronous* dan *synchronous*. SLIP (Serial Line in Protocol) adalah protocol sebelum PPP dimana teknik enkapsulasinya lebih sederhana dari PPP. Ethernet adalah standard IEEE 802.3 untuk komunikasi dua komputer atau lebih, Ethernet menggunakan CSMA/CD (Carrier Sense Multiple Access/Collusion Detection) yaitu metode agar tidak saling menggirimkan informasi secara bersamaan. Setiap ethernet card mempunyai 48 bit sebagai alamatnya.

Internet layer merupakan protokol yang bertanggung jawab dalam proses pengiriman paket ke alamat yang tepat dan bersifat *unreliable* dan *connectionless*. Pada layer ini terdapat tiga macam protokol yaitu IP, ARP dan ICMP [5]. *Internet protocol* berfungsi untuk menyampaikan paket data ke alamat yang tepat. ARP (*Address Resolution Protocol*) ialah protokol yang digunakan untuk menemukan alamat hardware dari LAN card.

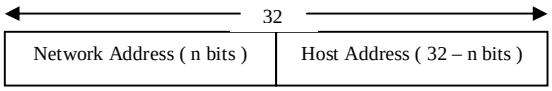
Transport layer merupakan protokol yang bertugas untuk mengadakan hubungan dan mengatur transportasi data antara dua host/komputer. Protokol dalam lapisan ini, yaitu TCP dan UDP. TCP (*Transmission Control Protocol*) bersifat *reliable* dan *connection oriented*, Sedangkan UDP (*Unit Datagram Protocol*) bersifat *connectionless* dan *unreliable* [5].

Application layer, merupakan lapisan teratas yang berisi semua aplikasi berbasis TCP & IP dan berhubungan langsung dengan pemakai. Aplikasi tersebut misalnya FTP, HTTP dan Telnet [5]. FTP (*File Transfer Protokol*) adalah program aplikasi untuk mentransfer file antara clien & server. HTTP (*Hyper Text Transfer Protocol*) adalah program aplikasi yang digunakan untuk menterjemahkan alamat IP menjadi susunan huruf yang dipisahkan dengan tanda "." misal http://www.undip.ac.id Dari beberapa macam protokol yang ada dalam TCP & IP, protokol IP merupakan inti dari protokol TCP & IP. Seluruh data yang berasal dari lapisan diatas IP harus dilewatkan, diolah oleh protokol IP dan kemudian dikirimkan sebagai paket IP ke tujuan. Dalam melakukan pengiriman paket, protokol IP bersifat *unreliable*, *connectionless* dan *datagram delivery service*. Saat ini terdapat dua versi dari protokol IPv4 (32 bit) dan IPv6 (128 bit). *Unreliable* berarti protokol IP tidak menjamin *datagram* yang dikirim pasti sampai di tujuan. Protokol IP hanya berusaha sebaik mungkin untuk membawa *datagram* sampai ke tujuan. *Connectionless* berarti dalam mengirim paket ke tujuan tidak ada perjanjian terlebih dahulu (*handshake*). *Datagram delivery service* berarti paket data yang dikirim *independent* terhadap paket data yang lain. Akibatnya jalur yang ditempuh oleh masing-masing paket berbeda satu dengan lainnya.

2.2 Internet Protokol versi 4 (IPv4)

Model pengalamatan dalam IPv4 menggunakan 32 bit bilangan biner. Namun untuk mempermudah penulisannya maka setiap delapan bit biner diwakili oleh satu segmen bilangan oktet, sehingga setiap alamat akan memiliki empat buah segmen dari 0.0.0.0 sampai dengan 255.255.255.255 misalnya 202.152.254.254 sehingga total alamat sebesar 2^{32}.

Alamat IPv4 dibagi menjadi dua bagian yaitu alamat jaringan (*network address*) dan alamat komputer (*host address*). *Network address* digunakan untuk menunjukkan di jaringan mana komputer berada, sedangakan "*host address*" menunjukkan komputer tersebut dalam jaringannya tersebut.

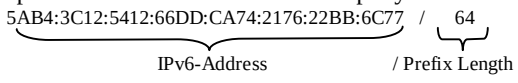


Gambar 2.2 Sistem Pengalamatan IPv4

Untuk meningkatkan efisiensi dan mempermudah administrasi jaringan, maka dalam suatu jaringan yang besar perlu dibagi-bagi ke dalam jaringan yang lebih kecil. Konsep ini sering disebut dengan *subnetwork* / *subnetting* [5].

2.3 Internet Protokol Versi6 (IPv6)

Pada dasarnya IPv6 dikembangkan untuk mengantisipasi kelangakaan IP *address* yang disediakan oleh IPv4. Karena IPv6 ini tidak lagi menggunakan 32 bit biner tetapi 128 bit biner, sehingga alamat yang mampu disediakan yaitu 2^{128} atau sebesar 3×10^{38} alamat. Selain itu juga dilakukan perubahan dalam penulisannya yaitu 128 bit alamat dipisahkan menjadi masing-masing 16 bit yang tiap bagian dipisahkan dengan ":" dan dituliskan dengan bilangan hexadesimal. Untuk mengetahui letak *subnet* dari alamat tersebut maka penulisan alamat IPv6 harus mempunyai format :



Dimana 64 merupakan jumlah bit yang menunjukkan alamat *subnet*nya yaitu [10]

5Ab4:3C12:5412:66DD::/64

2.3.2 Kelebihan IPv6

IPv6 sebagai hasil dari pengembangan IPv4 memiliki berbagai kelebihan, yaitu :

1. Perluasan ruang alamat karena menggunakan 128 bit.
2. Adanya pembagian jenis alamat *unicast, multicast* dan *anycast*
3. Memiliki *header* yang lebih mudah dan efisien
4. Adanya sistem penemuan *node* tetangga, konfigurasi alamat otomatis dan struktur *routing* bertingkat.
5. Mampu berinterkoneksi dengan IPv4

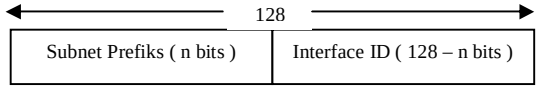
2.3.3 Pengalokasian Alamat pada IPv6

Pada IPv6 pengalokasian alamat dilakukan berdasarkan IPv6-*format prefiks* pada alamat IP-nya yang unik untuk setiap alamat. Ada beberapa *prefiks* IPv6 yang belum digunakan untuk keperluan masa datang.

2.3.4 Jenis-jenis alamat dalam IPv6

2.3.4.1 Alamat Unicast

Alamat *unicast* yaitu alamat yang menunjuk pada sebuah antarmuka atau *host*, digunakan untuk komunikasi satu lawan satu. Alamat *unicast* ini dibentuk dengan menambahakan *subnet prefiks* dengan *interface identifier* (64 bit).

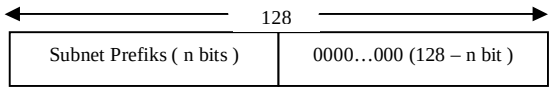


Gambar 2.4 Sistem Pengalamatan Unicast

Pada alamat *unicast* ada yang bersifat global (misalnya untuk *provider*), tetapi ada juga yang bersifat *local*, baik *link local address* maupun *site local address*. *Link local address* adalah alamat yang digunakan di dalam satu *link* yaitu jaringan lokal yang saling terhubung dalam satu *level*. Sedangkan *site local address* setara dengan alamat privat, yang dipakai terbatas di dalam satu *site* sehingga tidak dapat digunakan untuk mengirimkan alamat diluar *site* ini [6].

2.3.4.2 Alamat Anycast

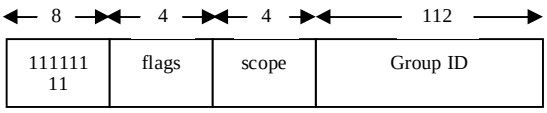
Alamat *anycast* yaitu alamat yang menunjukkan beberapa antarmuka (biasanya pada *node* yang berbeda). Paket yang dikirim ke alamat ini akan dikirim ke salah satu antarmuka yang paling dekat oleh *router*. Alamat ini berguna untuk beberapa *server* yang memberikan pelayanan yang sama, pada waktu ada permintaan layanan tersebut. *Anycast address* tidak mendapat alokasi *subnet / prefiks* yang sama maka alamat tersebut sudah merupakan alamat *anycast*.



Gambar 2.5 Sistem Pengalamatan Anycast

2.3.3.3 Alamat Multicast

Alamat *multicast* yaitu alamat yang menunjukkan beberapa antarmuka (biasanya untuk *node* yang berbeda). Paket yang dikirimkan ke alamat ini maka akan dikirimkan seluruh antarmuka yang ditunjukkan oleh alamat tersebut.



Gambar 2.6 Sistem Pengalamatan Multicast

Suatu host atau antarmuka dapat diberikan lebih dari satu jenis alamat, misalnya memiliki *unicast address*, *link local address* dan *anycast address*.

2.4 Mekanisme Transisi IPv4 ke IPv6

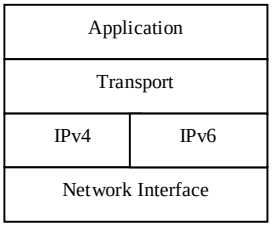
Pada dasarnya IPv4 dan IPv6 tidak kompatibel, sehingga memerlukan suatu mekanisme transisi dari IPv4 ke dalam IPv6. mekanisme transisi tersebut mempunyai dua tujuan utama, yaitu :

1. Membuat agar terminal IPv6 dapat berkomunikasi dengan terminal IPv4.
2. Melewatkan paket IPv6 melalui jaringan IPv4 yang sudah ada.

Ada beberapa mekanisme transisi yang kita kenal antara lain *Dual IP Stack*, *Tunneling* dan *Protokol Translator*.

2.4.1 Dual IP Stack IPv6/IPv4

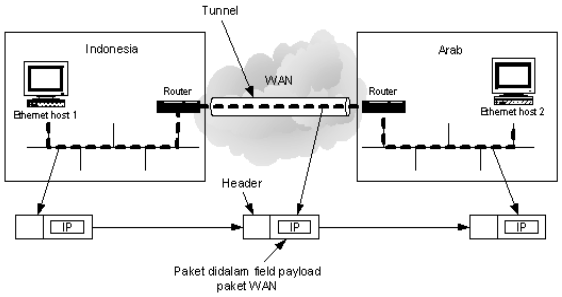
Dual IP stack adalah mekanisme yang mendukung untuk kedua protokol baik IPv6 maupun IPv4 untuk *host* dan *router*.



Gambar 2.8 Dual IP Stack IPv4/IPv6

2.4.2 Tunneling

Agar 2 jaringan yang berbeda bisa berhubungan diperlukan penanganan khusus, yang secara aplikasi merupakan hal yang sangat sulit. Secara umum *tunneling* dapat dianalogikan dengan 2 buah *host*. 1 *host* sumber dan 1 *host* tujuan dan merupakan jaringan yang berjenis sama, akan tetapi terdapat jaringan yang berbeda yang terletak diantaranya. Sebagai contoh, seperti yang disajikan pada Gambar 2.9.

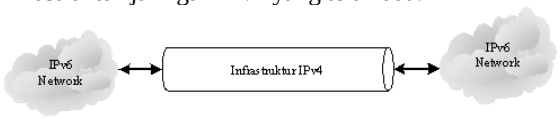


Gambar 2.9 Tunneling suatu paket dari Indonesia ke Arab

Dengan melihat Gambar 2.9 diatas permasalahan dengan 2 jaringan berbeda dapat diselesaikan dengan teknik *tunneling*. Dalam pengiriman paket IP ke *host* 2, *host* 1 membuat paket yang berisi alamat IP *host* 2, menyisipkannya ke *frame ethernet* yang dialamatkan ke *router* Indonesia dan menaruhnya pada *ethernet*. Pada saat *router* mendapatkan *frame*, *router* tersebut

menghapus paket IP dan menyisipkannya ke *field payload*. Paket *network layer* WAN kemudian mengalamatkannya ke alamat *router* WAN di Arab. Ketika paket tiba di Arab, *router* Arab menghapus paket IP dan mengirimkannya ke *host* 2 pada *frame ethernet* [5].

Dalam Tugas Akhir ini mekanisme tunneling yaitu mekanisme melewatkan paket IPv6 melalui jaringan IPv4 yang telah ada, tanpa merubah infrastruktur jaringan IPv4 yang telah ada.

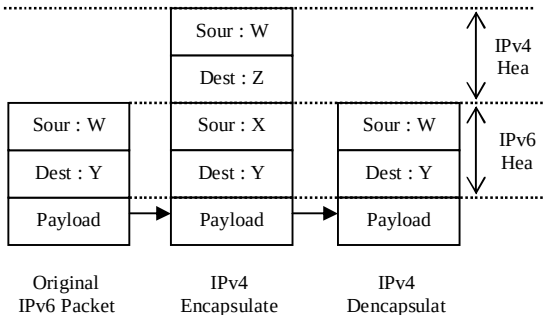


Gambar 2.10 Mekanisme Transisi Tunneling

Ada beberapa mekanisme tunneling, yaitu :

1. *6over4*, dimana paket IPv6 dapat secara otomatis dienkapsulasi melalui jaringan IPv4 dengan menggunakan IP *multicast* [7].
2. *6to4*, dimana alamat IPv6 dibuat berdasarkan alamat IPv4 atau sering disebut dengan IPv4-*compatible* IPv6 *compatible address*.
3. IPv6 *Tunnel Broker*, yang menyediakan *server* tersendiri untuk menkonfigurasi *tunnel* secara otomatis bagi klien IPv4, sehingga dapat terhubung dengan jaringan *backbone* IPv6.
4. DSTM (*Dual Stack Transition Mechanism*), yaitu *Dual Stack* IP dimana alokasi IPv4 dilakukan secara otomatis, penggunaan IPv4 *over* IPv6 untuk pengiriman melalui IPv6 sebelum tersambung ke jaringan IPv4 [4].

Mekanisme *tunneling* ini dilakukan dengan cara mengenkapsulasi paket IPv6 dengan *header* IPv4, kemudian paket tersebut langsung dikirimkan ke jaringan IPv4. enkapsulasi dilakukan oleh pengirim (misalnya R1), dan penerima (misalnya R2) melakukan proses sebaliknya yaitu de-enkapsulasi.



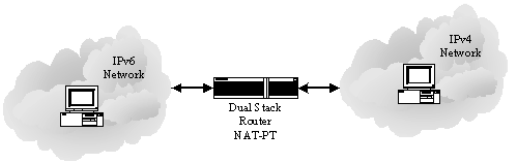
Gambar 2.11 Proses enkapsulasi pada mekanisme transisi Tunneling

2.4.3 Translasi Protokol

Mekanisme ini dilakukan dengan cara menerjemahkan protokol IPv4 ke IPv6 dan sebaliknya. Ada beberapa metode translasi protokol, yaitu :

1. ALG (*Application Level Gateway*), yaitu *host* IPv6 hanya berkomunikasi dengan IPv4 melalui sebuah *Dual Stack Proxy*.

2.NAT-PT (*Network Address translator Protocol Translator*), yaitu metode yang memungkinkan untuk melakukan translasi alamat dan protokol IPv6 dari/ke IPv4 pada level IP [2].



Gambar 2.12 Mekanisme Transisi Translasi Protokol

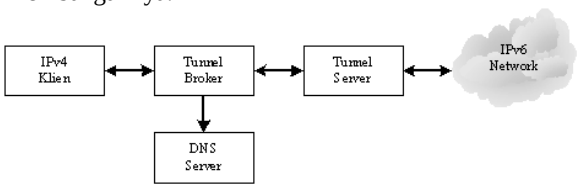
3. BIS (*Bump In Stock*), Yaitu mekanisme yang membolehkan aplikasi pada IPv4 berkomunikasi dengan host IPv6
4. SOCK *Gateway*, yaitu menerima koneksi *enchanced* SOCKS dan meneruskannya ke jaringan IPv4 atau IPv6.

2.5 IPv6 Tunnel Broker

IPv6 *Tunnel Broker* merupakan salah satu mekanisme transisi dari IPv4 ke IPv6 dengan cara menyediakan konfigurasi secara otomatis untuk melakukan *Tunneling* IPv6 melalui IPv4 kepada user IPv4 yang terhubung ke jaringan internet. Jadi IPv6 *tunnel broker* dapat dianalogikan seperti ISP dengan IPv6 yang menyediakan koneksi IPv6 kepada user yang telah terhubung ke internet dengan IPv4

2.5.1 Arsitektur Tunnel Broker

Arsitektur dari *tunnel broker* dimodelkan berdasarkan elemen-elemen fungsional yang membangunnya.



Gambar 2.13 Arsitektur IPv6 Tunnel Broker

2.5.1.1 Tunnel Broker

Tunnel broker merupakan tempat koneksi user IPv4 untuk melakukan proses registrasi dan aktifasi tunnel. Tunnel broker bertugas untuk mengatur pembentukan, modifikasi dan pembubaran tunnel sesuai dengan permintaan user. Dalam prakteknya *tunnel broker* dapat membagi beban jaringan kepada beberapa tunnel server, dengan cara mengirimkan konfigurasi kepada *tunnel server* yang bersangkutan pada saat *tunnel* tersebut dibentuk, dimodifikasi ataupun dihapus. Selain itu *tunnel broker* juga berkewajiban untuk mendaftarkan alamat IPv6 user dan memasukkannya dalam DNS server.

Tunnel broker harus mendukung IPv4 tetapi tidak harus mendukung IPv6, karena *Tunnel Broker* berhubungan langsung dengan IPv4 dan hubungan *tunnel broker* dan *tunnel server* dapat berupa IPv6 maupun IPv4.

2.5.1.2 Tunnel Server

Tunnel server merupakan *router dual stack* (IPv4 dan IPv6) yang terhubung dengan jaringan IPv6. *Tunnel server* bertugas menerima seluruh konfigurasi yang dikirim oleh *tunnel broker* pada saat pembangunan, modifikasi dan pembubaran tunnel disisi *server*.

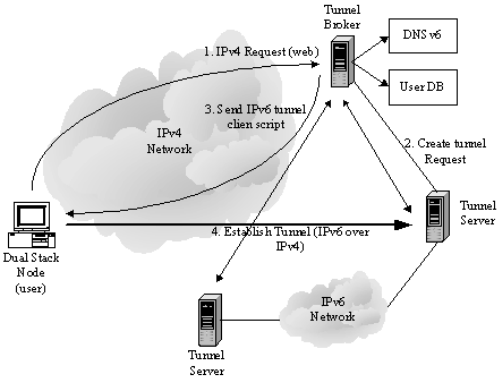
2.5.1.3 DNS (Domain Name Service) Server

DNS *Server* ini bertugas untuk meterjemahkan (*resolve*) dari nama domain ke alamat IP atau

sebaliknya dari pemakai yang telah membentuk *tunnel*. *Server* ini harus mendukung IPv6, karena *domain* yang kita gunakan merupakan jaringan IPv6.

2.5.2 Mekanisme Kerja IPv6 Tunnel Broker

Mekanisme kerja dari IPv6 *Tunnel Broker* dapat disajikan dalam Gambar 2.14.



Gambar 2.14 Cara Kerja IPv6 Tunnel Broker

1. *user* menghubungi *tunnel broker* dan dilanjutkan dengan prosedur registrasi (misalnya dengan mengisi *form* pada *web*), kemudian *user* akan diberi hak untuk mengakses layanan *tunnel*.
2. *user* menghubungi kembali *tunnel broker*, dan setelah ada proses authentifikasi *user* tersebut memberikan informasi tentang konfigurasi dari *host*-nya (alamat IP, *Operating System* dan perangkat lunak pendukung IPv6).
3. *Tunnel Broker* kemudian mengkonfigurasikan *tunnel* di sisi jaringan (*tunnel server*) dan DNS *Server*.
4. Kemudian *user* akan diberikan skrip aktifasi tunnel pada sisi *user*. Jika proses ini berhasil maka *user* telah terhubung ke jaringan IPv6 melalui *tunnel server* yang telah ditentukan tunnel broker.
5. *user* dapat meminta modifikasi dan pembubaran *tunnel* dengan mengakses *tunnel broker* lagi.

Paparan diatas adalah mekanisme IPv6 *tunnel broker* pada saat pembuatan tunnel dan pembubaran tunnel.

III. PERANCANGAN DAN IMPLEMENTASI

Perancangan dan implementasi IPv6 *tunnel broker* disini untuk mengetahui seberapa handal sistem tunnel broker ini untuk diterapkan pada jaringan yang sudah terpasang IPv4 dan membentuk jaringan IPv6 baru tanpa merubah jaringan IPv4 yang sudah ada, dan kedua jaringan tersebut dapat berhubungan seperti dalam satu jaringan. Untuk mengetahui cara kerja dan performansi dari IPv6 *tunnel broker* maka perlu dilakukan perancangan dan implementasi jaringan *tunneling* itu sendiri. Ada banyak pengimplementasian jaringan *tunneling* IPv6 *over* IPv4, diantaranya IPv6 *tunnel broker*, dimana *tunnel* akan diaktifkan dan di non-aktifkan secara otomatis berdasarkan permintaan *user* melalui *web*. Ada beberapa langkah yang akan dilakukan dalam merancang dan mengimplementasikan IPv6 *tunnel broker*, yaitu :

1. Implementasi perangkat lunak *tunnel broker*
2. Implementasi IPv6 *Tunnel Broker* (*tunnel broker, tunnel server*, DNS *Server*)
3. Implementasi *Host* IPv6
4. Implementasi *tunnel klien* (*dual stack* IPv6/IPv4)
5. Pengujian jaringan *tunneling*

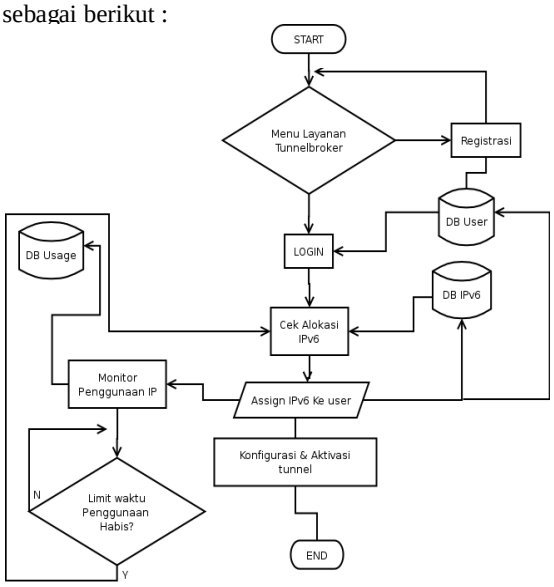
3.1 Perangkat Lunak Tunnel Broker

Perangkat lunak tunnel broker ini bersifat *free open sourc*e artinya bisa didapatkan secara gratis dan dapat disebarluaskan secara bebas yang tentunya apabila ada perubahan, harus diberitahukan ke forum *tunnel broker*.

3.1.1 Script PHP

PHP (*Hypertext Preprocessor*) yaitu skrip tambahan pada skrip HTML yang bersifat *server side* (dijalankan disisi *server*). Hampir semua sintaks PHP diambil dari bahasa C, Java, dan Perl. PHP sangat handal karena mendukung berbagai sistem bahasa basis data (MySQL,postgresql, oracle, dsb), dan mampu menghubungkan layanan dengan protokol yang berbeda-beda (HTTP,FTP,IMAP,POP3, dsb).

Skrip PHP dalam Tugas Akhir ini digunakan sebagai antarmuka antara klien dengan *tunnel broker* melalui *web*. Oleh karena itu skrip-skrip PHP ini harus dapat melakukan fungsi-fungsi dari *tunnel broker* itu sendiri, seperti aktivasi *tunnel*, pencatatan *user*, penghapusan *tunnel*, dan sebagainya. Secara umum *flowchart* dari perangkat lunak *tunnel broker* ini dapat digambarkan sebagai berikut :



Gambar 3.1 Flowchart perangkat lunak tunnel broker

Mengacu pada *flowchart* diatas keperluan perangkat lunak untuk *server tunnel broker* dengan menggunakan skrip PHP diantaranya :

1. registrasi.php, yaitu skrip untuk melakukan pendaftaran user baru dengan cara memasukan data user tersebut ke dalam tabel user pada basis data *tunnel*. Data ini berfungsi untuk authenfikasi *user* yang akan mengaktifkan dan menonaktifkan *tunnel*.
2. login.php, yaitu skrip untuk melakukan pengecekan kode *login* dan *password* dari *user* yang ingin menggunakan *tunnel broker*. Pengecekan berdasarkan pada tabel *user* pada basis data *tunnel*.
3. tunnel.php, yaitu skrip untuk melakukan

pembentukan *tunnel* antara *host* IPv4 *dual stack* dengan host IPv6.

3.1.2 Perancangan Basis Data

Dalam perancangan *tunnel broker* ini dibutuhkan basis data untuk menyimpan data user. Hal ini bertujuan agar setiap dapat dimonitoring. MySQL yaitu salah satu sistem menajemen basis data yang terstruktur, dimana data disimpan dalam tabel-tabel yang berbeda, dan tidak mengumpulkannya dalam satu ruangan penyimpanan (basis data) yang besar, tetapi terpisah-pisah. Setiap tabel dari basis data tersebut dapat saling dihubungkan berdasarkan keinginan pengguna. MySQL mempunyai beberapa kelebihan disbanding dengan system basis data lainnya, diantaranya *multiuser*, mendukung *record* yang besar dan waktu eksekusi perintah yang sangat cepat.

Dalam tunnel broker ini dibutuhkan dua jenis basis data, ayitu admin dan tunnel. Basis data admin digunakan untuk keperluan administrasi tunnel, yang didalamnya terdapat tabel admin dengan data sebagai berikut :

1. Dbuser : user untuk mengakses basis data.
2. DbIPv6 : password user untuk mengakes basis data.
3. Dbusage : pengunaan tunnel yang aktif.

Sedangkan basis data tunnel digunakan untuk mencatat data-data yang dibutuhkan untuk proses registrasi tunnel adalah sebagai berikut :

1. Nama_login :kode login untuk authenfikasi user yang akan mengaktifkan tunnel
3. Password : password dari kode login
4. Password2 : password dari kode login
5. Nama_depan : nama depan dari user pengguna tunnel
6. Nama_belakang : nama belakang dari user pengunna tunnel
7. Instansi : Instansi dari user
8. Alamat : alamat dari user
9. Negara : negara dari user
10. E-Mail : Alamat E-mail dari user

3.2 Implementasi IPv6 Tunnel Broker

IPv6 *Tunnel Broker* terdiri dari *tunnel broker*, *tunnel server* dan DNS *server*. Masing-masing *server* ini diimplementasikan secara terpisah maupun menyatu. Dalam tugas akhir ini ketiga *server* ini diimplementasikan dalam sebuah *host* saja, karena lebih mudah implementasinya dan karena keterbatasan perangkat. Namun dalam implementasinya antara ketiga *server* diatas tetap dibedakan, karena masing-masing mempunyai fungsi-fungsi yang berbeda-beda walaupun hanya pada level perangkat lunak.

3.2.1 Implementasi Tunnel Broker

Tunnel broker merupakan tempat koneksi klien IPv4 untuk melakukan registrasi dan aktivasi *tunnel* melalui *web*. Dalam implementasi *tunnel broker* ini hanya perlu dilakukan instalasi *webserver Apache* untuk memberikan layanan registrasi *tunnel* kepada *user* melalui *web*. Dari *web* inilah skrip-skrip baik PHP dan BASH akan dijalankan untuk memberikan perintah kepada *tunnel server*. Untuk menjamin keamanan data *user* maupun *tunnel* maka diimplementasikan juga SSL (*Secure Socket Layer*).

3.2.2 Implementasi Tunnel Server

Tunnel server merupakan *router dual stack* (IPv6 dan IPv4) yang terhubung ke jaringan internet (IPv4) dan jaringan IPv6, sehingga untuk mengimplementasikannya perlu diimplementasikan terlebih dahulu *host dual stac*k (IPv6/IPv4). Kamudian perlu diaktifkan *option tunnel* pada konfigurasi IPv6 dari host tersebut.

3.2.3 Implementasi DNS Server

DNS (*Domain Name Service*) server merupakan server yang menyediakan layanan *resolving* (menerjemahkan) nama domain ke IP *Address* dan sebaliknya. Untuk mengimplementasikan DNS *server* ini dibutuhkan beberapa perangkat lunak, yaitu :

a) Caching-nameserver
b) Bind-utils
c) Bind

Perangkat lunak di atas merupakan perangkat lunak DNS yang sudah mendukung pengalamatan IPv6. Ada beberapa perbedaan antara implementasi DNS *server* IPv4 dan IPv6, yaitu :

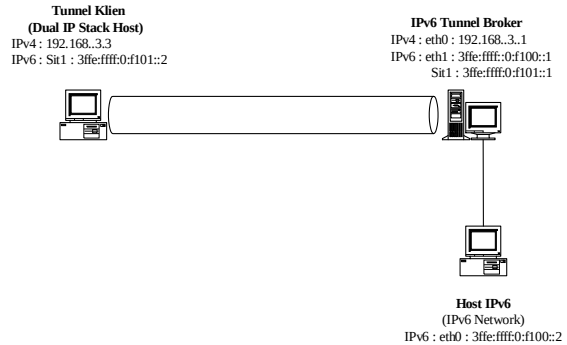
a) Perlu penambahan *zone* baru untuk IPv6
b) Option A (*address*) IPv4 pada file zona berubah menjadi AAAA IPv6

3.4. Perancangan dan Implementasi Jaringan IPv6 Tunnel Broker

Untuk menguji jaringan IPv6 *tunnel broker* diperlukan suatu perancangan arsitektur jaringan yang dapat menggambarkan hal-hal sebagai berikut :

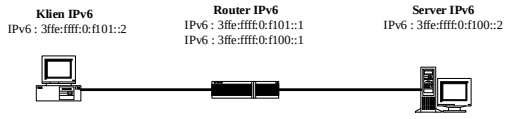
1. Adanya jaringan IPv6 saja yang terpisah dari klien
2. Adanya jaringan IPv4 saja untuk memastikan adanya *tunneling* IPv6 dalam IPv4 dalam jaringan tersebut
3. Adanya IPv6 *tunnel broker* yang merupakan penghubung antara klien IPv6 yang terpisah dengan jaringan IPv6

Dalam tugas akhir ini jaringan yang diharapkan dapat mengambarkan hal-hal seperti diatas, dan juga dapat digunakan untuk mengukur performansi jaringan adalah sebagai berikut :



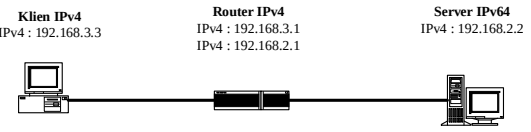
Gambar 3.1 Perancangan Jaringan Tunnel Broker

Selain itu perlu diimplementasikan juga suatu jaringan IPv6 sebagai pembanding jaringan IPv6 *tunnel broker* diatas dalam pengiriman paket ICMP, HTTP dan FTP. Jaringan tersebut dapat digambarkan sebagai berikut :



Gambar 3.2 Perancangan Pengukuran Jaringan IPv6

Untuk pengukuran jaringan IPv4 digunakan jaringan seperti diatas, dengan cara mengganti seluruh alamat menjadi alamat IPv4.



Gambar 3.3 Perancangan Pengukuran Jaringan IPv4

3.4.2 Perancangan dan Implementasi Pengujian Tunnel Broker

Ada beberapa parameter yang akan diuji dalam jaringan *tunnel broker* ini untuk menunjukkan interkoneksi antara IPv4 dan IPv6, yaitu :

1. Pembentukan *tunnel* antara *tunnel* klien dengan *tunnel server* oleh *tunnel broker*, dengan cara melihat antarmuka *tunnel* dan alamat IPv6 pada masing-masing *host*.
2. Koneksi IPv6 melalui IPv4 (tunneling) antara klien dengan IPv6 tunnel broker yang dilakukan dengan mengirimkan paket ICMP.
3. Koneksi antara tunnel klien dengan *host* IPv6 yang dilakukan dengan mengirimkan paket protokol ICMP, FTP dan HTTP.

IV. ANALISA HASIL PENGUJIAN DAN PENGUKURAN IPv6 TUNNEL BROKER

Pengujian dan pengukuran dilakukan untuk mengetahui unjuk kerja dari sistem yang telah di rancang dan di implementasikan. Dari hasil pengujian dan pengukuran tersebut kemudian di analisa dan dibandingkan dengan teori yang telah diperoleh. Analisa tersebut meliputi analisa interkoneksi jaringan, enkapsulasi paket, dan performansi jaringan.

4.1 Analisa Pengujian Interkoneksi Tunneling

Pengujian interkoneksi tunneling ini dibagi menjadi beberapa tahap yaitu pengujian pembentukan *tunnel* oleh *tunnel broker*, interkoneksi antara *tunnel broker* dan *tunnel* klien dan interkoneksi antara *tunnel* klien dengan *host* IPv6.

4.1.1 Pengujian Pembentukan Tunnel Oleh Tunnel Broker

Untuk menguji keberhasilan dari *tunnel broker* dalam mengimplementasikan *tunnel* untuk klien dapat dilihat pada antarmuka jaringan pada *tunnel server* maupun klien. Untuk rotoc operasi linux dapat dilakukan dengan menggunakan perintah *ifconfig* dan *ip tunnel*.

Untuk perintah *ifconfig eth0* akan menghasilkan keluaran sebagai berikut :

```
test6in4  Link encap:IPv6-in-IPv4
          inet6 addr: 3ffe:ffff:0:f100::2/64 Scope:Global
          inet6 addr: fe80::c0a8:302/128 Scope:Link
          UP POINTOPOINT RUNNING NOARP  MTU:1480  Metric:1
          RX packets:239 errors:0 dropped:0 overruns:0 frame:0
          TX packets:304 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:25171 (24.5 Kb)  TX bytes:42647 (41.6 Kb)

eth0      Link encap:Ethernet  HWaddr 00:40:F4:8D:3E:74
          inet6 addr: 3ffe:ffff:0:f100::3/64 Scope:Global
          inet6 addr: fe80::240:f4ff:fe8d:3e74/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:100671 errors:0 dropped:0 overruns:0 frame:0
          TX packets:63195 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:134573425 (128.3 MiB)  TX bytes:8114560 (7.7 MiB)
          Interrupt:10 Base address:0xe000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:135 errors:0 dropped:0 overruns:0 frame:0
          TX packets:135 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:12812 (12.5 KiB)  TX bytes:12812 (12.5 KiB)
```

Untuk perintah *ip tunnel* menghasilkan keluaran sebagai berikut :

```
[root@localhost root]# ip tunnel list
sit0: ipv6/ip  remote any  local any  ttl 64  nopmtudisc
test6in4: ipv6/ip  remote 192.168.2.2  local 192.168.3.2  ttl inherit
```

Sedangkan untuk menguji keberhasilan aktivasi *tunnel* dapat terlihat pada antarmuka jaringan dari klien tersebut dengan perintah yang sama, sehingga akan menghasilkan keluaran yang hampir sama.

4.1.2 Pengujian Interkoneksi

Pengujian ini dilakukan dengan cara mengirimkan paket rotocol ICMP yang dilakukan dengan aplikasi program *ping6*. Pada aplikasi *ping6* apabila interkoneksi *tunneling* sudah terbentuk maka akan menghasilkan keluaran sebagai berikut :

```
C:\Documents and Settings\reko>ping6
3ffe:ffff:0:f100::1
Pinging 3ffe:ffff:0:f100::1
from 3ffe:ffff:0:f100::1 with 32 bytes
of data:
Reply from 3ffe:ffff:0:f100::1:
bytes=32 time<1ms
Reply from 3ffe:ffff:0:f100::1:
bytes=32 time<1ms
Reply from 3ffe:ffff:0:f100::1:
bytes=32 time<1ms
Reply from 3ffe:ffff:0:f100::1:
bytes=32 time<1ms
Ping statistics for
3ffe:ffff:0:f100::1:
    Packets: Sent = 4, Received = 4,
Lost = 0 (0% loss),
Approximate round trip times in milli-
seconds:
        Minimum = 0ms, Maximum = 0ms,
              Average = 0ms
```

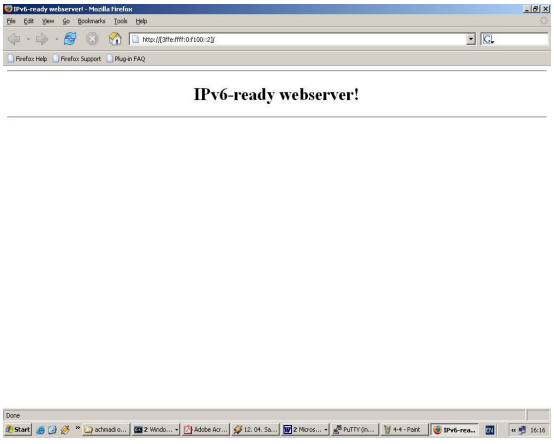
4.1.2.2 Interkoneksi Antara Tunnel Klien dan Host IPv6

Pengujian ini dilakukan seperti pengujian pada point 4.1.2.1 hanya saja *host* tujuan merupakan host IPv6. Selain itu pengujian juga dilakukan untuk paket dengan rotocol FTP (program aplikasi ftp klien) dan HTTP (program aplikasi *web browser*).

```
C:\Documents and Settings\reko>ping6
3ffe:ffff:0:f100::2
Pinging 3ffe:ffff:0:f100::2
from 3ffe:ffff:0:f100::1 with 32 bytes
of data:
Reply from 3ffe:ffff:0:f100::2:
bytes=32 time<1ms
Reply from 3ffe:ffff:0:f100::2:
bytes=32 time<1ms
Reply from 3ffe:ffff:0:f100::2:
bytes=32 time<1ms
Reply from 3ffe:ffff:0:f100::2:
bytes=32 time<1ms
Ping statistics for
3ffe:ffff:0:f100::2:
Packets: Sent = 4, Received = 4, Lost =
0 (0% loss),
Approximate round trip times in milli-
seconds:
       Minimum = 0ms, Maximum = 0ms,
             Average = 0ms
```

```
C:\Documents and Settings\reko>ftp
3ffe:ffff:0:f100::2
Connected to 3ffe:ffff:0:f100::2.
220---------- Welcome to Pure-FTPd
[privsep] [TLS] ----------
220-You are user number 1 of 50
allowed.
220-Local time is now 21:15. Server
port: 21.
220-This is a private system - No
anonymous login
220 You will be disconnected after 15
minutes of inactivity.
User (3ffe:ffff:0:f100::2☹none)):
knoppix
331 User knoppix OK. Password required
Password:
230-User knoppix has group access to:
knoppix  usb      users    games
230- staff    video    dip      audio
sudo     tape     floppy   cdrom
230- voice    fax      dialout  lp
tty
230 OK. Current directory is
/home/knoppix
ftp>
```

Hasil diatas adalah merupakan pengujian interkoneksi antara jaringan Ipv4 dengan jaringan Ipv6 yang terkoneksi dengan *tunnel broker*. Terlihat diatas pengujian dengan ping6 dimana 3ffe:ffff:0:f100::2 adalah *host* tujuan yang menggunakan alamat Ipv6. Dengan adanya kalimat *reply* berarti hubungan komunikasi kedua klien tersebut sukses. Apabila pengujian dengan menggunakan ping6 sukses, pengujian dengan menggunakan aplikasi FTP juga dapat dilakukan dengan *host* yang sama.



Gambar 4.1 Hasil Pengujian program HTTP

4.1.3 Enkapsulasi Paket IPv6 oleh IPv4

Pada proses *tunneling* IPv6 *over* IPv4 terjadi pembungkusan (enkapsulasi) paket IPv6 oleh IPv4. Untuk mengamati proses ini dapat dilakukan dengan cara mengambil (*capturing*) paket yang melewati perangkat *router* IPv4 dengan program *cold. Cold* merupakan program untuk menganalisa paket dalam jaringan dengan cara memecah *frame* data untuk mendapatkan struktur rotocol dan informasi tentang rotoc transportasi data.

Pengamatan dilakukan dengan cara mengirimkan paket dari *tunnel* klien ke *tunnel broker* baik paket IPv6 maupun peket hasil enkapsulasi *tunneling* IPv6 *over* IPv4, kemudian kita jalankan program *cold* pada router IPv4. rotocol yang digunakan dalam pengujian ini yaitu rotocol ICMP, FTP dan HTTP.

Keluaran yang dihasilkan oleh program tersebut dapat dilihat pada lampiran 2, untuk berbagai macam paket yang berbeda.

Paket yang berhasil diambil akan memiliki informasi sebagai berikut :

[PKT] menunjukkan bahwa adanya paket baru, yang disertai dengan nomor *frame* dan jumlah total byte dari paket tersebut.

[MAC] menunjukkan alamat perangkat keras (MAC address) penerima dan pengirim termasuk karakteristik alamat tersebut, serta jenis protokol yang dibawa dalam paket ini.

[IP Frame]menunjukkan *header* dari IPv6 yang mempunyai fungsi sama dengan *header* IPv4, tetapi dari struktrur headernya ada beberapa informasi pada IPv4 yang dihilangkan, dimodifikasi ataupun ditambahkan.

[ICMPv6] menunjukkan bahwa paket yang dikirimkan merupakan protokol ICMP versi 6.

[TCP Frame] menunjukkan *header* pada lapis transport yaitu TCP. *Header* ini berisi informasi yang digunakan oleh lapisan transport untuk menjamin paket sampai di tujuan.

Dari hasil pengamatan keluaran program maka dapat digambarkan struktur paket dari masing-masing pengujian :

4. Aplikasi FTP melalui jaringan IPv4

MAC	IPv4 Header	TCP Header	DATA

4. Aplikasi FTP melalui jaringan tunneling IPv6 over IPv4

MAC	IPv4	IPv6	TCP	Data

4. Aplikasi program ping melalui jaringan IPv4

MAC Header	IPv4 Header	ICMP Packet

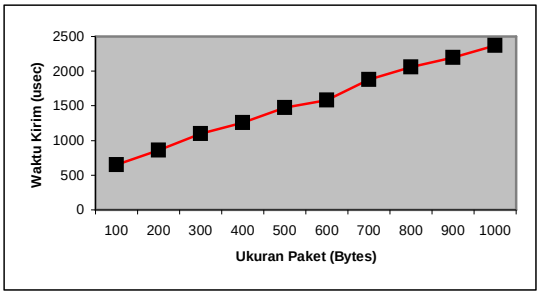
4. Aplikasi program ping melalui jaringan tunneling IPv6 over IPv4

MAC	IPv4 Header	IPv6 Header	ICMP

Dari gambar di atas dapat disimpulkan bahwa pada jaringan *tunneling* IPv6 *over* IPv4 terjadi proses enkapsulasi paket, dimana paket data IPv6 dibungkus (dienkapsulasi) oleh IPv4 *header*, sehingga paket hasil enkapsulasi ini dapat langsung dikirimkan melalui jaringan IPv4 yang telah ada. Tetapi pada proses ini menyebabkan IP *header* menjadi lebih besar karena terdiri dari *header* IPv6 dan IPv4, sehingga data (*actual payload*) akan berkurang, sehingga akan menyebabkan turunnya performansi jaringan.

4.2 Analisa pengukuran performansi interkoneksi tunneling

Pengukuran pengiriman paket ICMP dilakukan dengan menggunakan program *ping6*. keluaran program ini memberikan informasi tentang waktu yang diperlukan oleh sebuah paket ICMP untuk diterima kembali oleh pengirim setelah dibalas oleh penerima. Dalam kondisi jaringan normal, waktu tersebut akan memiliki nilai yang sama untuk semua paket ICMP yang dikrimkan karena paket tersebut menggunakan jalur yang sama.



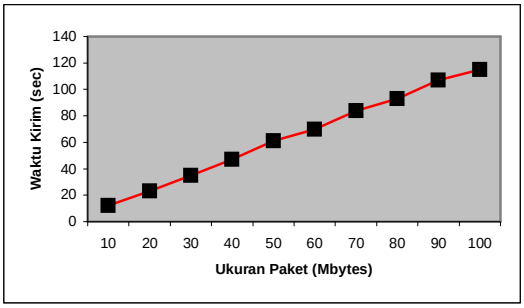
Gambar 4.2 Performansi aplikasi program ping untuk jaringan tunnel IPv6 over IPv4

Dalam pengukuran ini kita menggunakan beberapa ukuran paket ICMP 100 sampai 1000 *bytes*. Dari grafik diatas menunjukkan bahwa pertambahan waktu kirim berbanding lurus dengan pertambahan ukuran paket ICMP. Sehingga dapat disimpulkan bahwa untuk paket ICMP yang lebih besar membutuhkan waktu kirim yang lebih lama.

4.2.2 Pengukuran waktu pengiriman paket dengan protokol FTP

Pengukuran ini dilakukan dengan menggunakan program FTP klien untuk mengambil beberapa file yang ukuran filenya berbeda, maka waktu transfer yang dibutuhkan juga besar.

Dari hasil pengukuran terlihat bahwa kenaikan waktu transfer berbanding linier dengan kenaikan ukuran data yang dikirimkan. Untuk penanganan paket yang besar protokol TCP mempunyai mekanisme menghindari kemacetan (*congestion avoidance*) dan mengontrol dan mengirimkan kembali paket yang rusak (*error detection and retransmission*), tetapi pada kenyataannya mekanisme kurang berpengaruh dalam besarnya waktu kirim.



Gambar 4.3 Performansi Aplikasi FTP untuk jaringan Tunnel IPv6 over IPv4

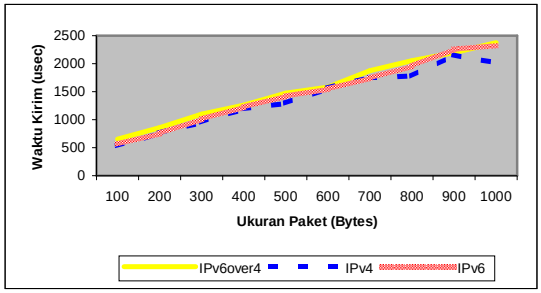
Hal ini disebabkan karena paket yang ada di jaringan sebagian besar berasal dari pengukuran ini. Sehingga kemungkinan adanya kemacetan dan kerusakan paket sangat kecil.

4.3 Perbandingan performansi jaringan antara IPv4, jaringan IPv6 dan jaringan *tunneling* IPv6 over IPv4

Implementasi IPv6 *stack* pada *host* akan melakukan perubahan pada lapisan TCP/IP, dimana ukuran alamat pada IP *header* berubah dari 32 bit menjadi 128 bit dan *header checksum* dihilangkan pada layer IP. Pada Tugas Akhir ini akan dilihat perbandingan performansi jaringan karena adanya perubahan tersebut untuk protokol FTP dan ICMP. Jaringan yang akan meliputi jaringan *tunneling* IPv6 *over* IPv4, jaringan IPv4 dan jaringan IPv6.

4.3.1 Paket dengan protokol ICMP

Pada pengukuran dengan menggunakan protokol ICMP (program *ping*) diperoleh hasil perbandingan antara jaringan IPv6, jaringan IPv4 dan jaringan *tunneling* IPv6 *over* IPv4 seperti terlihat pada gambar 4.9.



Gambar 4.4 Perbandingan performansi jaringan untuk aplikasi ping

Dari grafik hasil perbandingan terlihat bahwa waktu transfer untuk seluruh jaringan akan naik jika ukuran *file* juga naik. Untuk ukuran *file* yang besar maka perbedaan waktu transfer yang dibutuhkan untuk jaringan *tunneling* IPv6 *over* IPv4 lebih besar dari jaringan IPv6, dan untuk jaringan IPv6 lebih besar dari jaringan IPv4, karena ukuran paket *tunneling* IPv6 *over* IPv4 lebih besar dari paket IPv6, dan paket IPv6 lebih besar dari paket IPv4.

4.3.2 Paket dengan menggunakan protokol FTP

Pada pengiriman data yang besar melalui *ethernet* ke jaringan, data dibagi menjadi paket-paket yang besarnya tergantung dari MTU (*Maximum Transfer Unit*) dari *ethernet* itu sendiri. *Real ethernet* memiliki MTU sebesar 1514 bytes termasuk *ethernet header* sebesar 14 bytes, sehingga beban *ethernet* tersebut sebesar 1500 *bytes*. Besar MTU ini sama untuk setiap data yang akan melewati jaringan IPv6, jaringan IPv4, dan jaringan IPv6 over IPv4. karena *header* untuk masing-masing jaringan berbeda maka beban aktual (*actual payload*) setiap paket menjadi berbeda-beda.

Penambahan *header* ini akan mengakibatkan penurunan performansi jaringan sebesar :

$$\eta = \frac{\text{Jumlah penambahan header}}{\text{Jumlah actual payload}}$$

Tabel 4.1 Perbandingan TCP Payload untuk jaringan IPv6, jaringan IPv4 dan Jaringan Tunneling IPv6 over IPv4

Real Ethernet 14 bytes Ethernet Payload	1514 bytes Ethernet Header 1500 bytes		
	IPv4 (bytes)	IPv6 (bytes)	IPv6 over IPv4 (bytes)
IP Header IP Payload TCP Header TCP Payload Options Actual Payload	20 1480 20 1460 xxxx yyyy	40 1460 20 1440 xxxx yyyy	60 1440 20 1420 xxxx yyyy

Actual Payload pada protokol FTP menggunakan TCP *payload* karena protokol FTP menggunakan protokol TCP untuk proses transfer datanya.

Pengaruh perubahan IP Header terhadap payloadnya sebagai berikut :

1. *Tunneling* IPv6 *over* IPv4 terhadap IPv4

Pada kasus terjadi penambahan *header* sebanyak 40 bytes, dan actual *payload* IPv4 sebesar 1460, sehingga performansinya akan turun sebesar :

$$\eta = \frac{40}{1460} = 0{,}0273926027$$

$$= 2{,}74\ \%$$

2. *Tunneling* IPv6 *over* IPv4 terhadap IPv6
   Pada kasus ini terjadi penambahan *header* sebesar 20 bytes, dan *actual payload* IPv6 sebesar 1440, sehingga performansinya akan turun sebesar :

$$\eta = \frac{40}{1460} = 0{,}0138888888$$

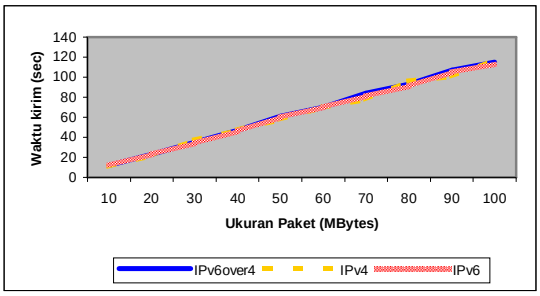
$$= 1{,}39\ \%$$

3. IPv6 terhadap IPv4
   Pada kasus ini terjadi penambahan *header* sebesar 20 bytes, dan *actual payload* IPv4 sebesar 1460, sehingga performansinya akan turun sebesar :

$$\eta = \frac{40}{1460} = 0{,}0136986301$$

$$= 1{,}37\ \%$$

Hasil pengukuran dengan protokol FTP diperoleh perbandingan antara jaringan IPv6, IPv4 dan *tunneling* IPv6 *over* IPv4 seperti pada gambar 4.11



Gambar 4.5 Perbandingan performansi jaringan untuk Aplikasi FTP

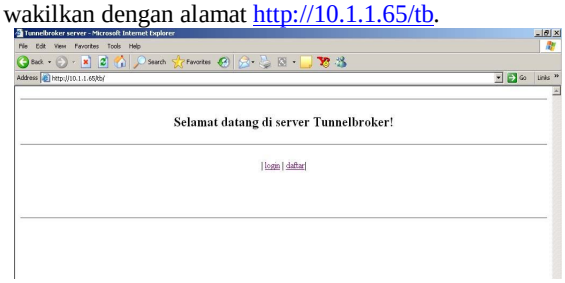
Grafik di atas menunjukkan hasil yang hampir sama dengan perbandingan performansi jaringan untuk aplikasi program *ping*. Dimana untuk ukuran file yang semakin besar membutuhkan waktu transfer yang semakin besar. Karena ukuran paket *tunneling* IPv6 *over* IPv4 memiliki ukuran yang lebih besar karena adanya penambahan *header* sehingga waktu transfernya juga lebih besar. Begitu juga dengan ukuran peket IPv6 jika dibandingkan dengan IPv4 mempunyai ukuran paket yang lebih besar, sehingga waktu transfernya juga lebih besar.

Dari hasil perhitungan penurunan performansi jaringan diatas terlihat adanya perbedaan yang cukup berarti. Hal ini disebabkan beberapa hal yaitu :

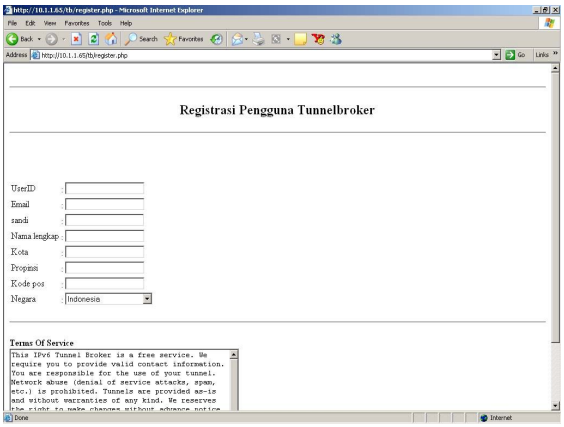
1. Pada perhitungan *Ethernet over* IP, *option* TCP *header* kita asumsikan nol, sedangkan pada kenyataannya aplikasi FTP *option* TCP *Header* belum tentu nol.
2. Adanya protokol yang berbeda yaitu IPv6 dan IPv4 dalam satu host (dual IP Stack) maka akan mempengaruhi performansi dalam *host* tersebut, sehingga mempengaruhi dalam pengiriman data dari/ke jaringan.

4.4 Tampilan *web Tunnel Broker*

Pada pertama kali klien ingin berhubungan dengan jaringan IPv6 yang dilakukannya adalah menuliskan alamat *web* yang telah menyediakan koneksi ke jaringan IPv6, dalam Tugas Akhir ini di wakilkan dengan alamat http://10.1.1.65/tb.

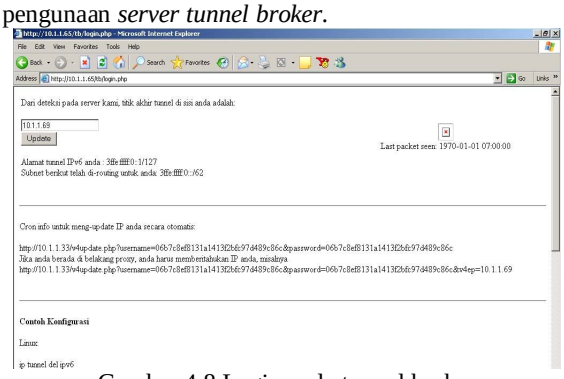


Gambar 4.6 Tampilan utama web tunnel broker



Gambar 4.7 Daftar pada tunnel broker

Setelah melihat tampilan utama dari *web tunnel broker*, klien dihadapkan pada 2 pilihan. Pilihan pertama apabila klien telah mempunyai *account* maka pilihan klien bisa *login*. Dan apabila klien belum mempunyai *account* maka tidak lain pilihan hanya daftar. Pada proses daftar, klien diminta untuk mengisi data-data yang diperlukan untuk keperluan administrasi pengunaan *server tunnel broker*.



Gambar 4.8 Login pada tunnel broker

Pada proses *login*, *tunnel broker* akan mendeteksi titik terakhirnya. Pendeteksian titik terakhir ini berupa *internet address* (IP) dari klien. Setelah *tunnel broker* telah mendapatkan titik terakhir ini maka *tunnel broker* akan memberikan alamat IPv6-nya. Selanjutnya pada sisi klien men-*download script* yang ada pada tampilan *web* untuk merubah *ethernet* nya

menjadi *dual stack*. *Dual stack* disini akan menghubungkan klien yang berada pada jaringan IPv4 dan berkomunikasi dengan jaringan IPv6.

V. PENUTUP

5.1 Kesimpulan

Dari hasil perancangan, implementasi dan analisa tentang IPv6 *Tunnel Broker* ada beberapa kesimpulan yang dapat dikemukakan, yaitu :

1. IPv6 *tunnel broker* merupakan mekanisme transisi IPv4 ke IPv6 yang Sangat penting untuk melakukan pengujian terhadap jaringan IPv6, karena proses pembentukan tunnel Sangat mudah bagi pemakai.
2. Pada IPv6 *Tunnel Broker* seluruh proses aktivasi, monitoring dan penghapusan tunnel hanya dilakukan oleh *tunnel broker* dengan menggunakan perangkat lunak berbasis *web*.
3. Pada *tunneling* IPv6 *over* IPv4 terjadi proses enkapsulasi pakat IPv6 dengan paket IPv4, sehingga paket IPv6 tersebut diperlakukan seperti paket IPv4 yang lain. Dengan demikian dengan mudah paket tersebut dapat dikirimkan melalui jaringan IPv4.
4. Penambahan IP Header secara tidak langsung akan mempengaruhi performansi jaringan, karena semakin besar IP *Header data actual* setiap MTU menjadi berkurang, sehingga jumlah paket semakin banyak dan membutuhkan waktu yang lebih lama.
5. Untuk paket dengan protokol ICMP penurunan performansi untuk jaringan IPv6 dibandingkan dengan jaringan IPv4 sebesar 5,40 %, jaringan *tunnel* IPv6 *over* IPv4 terhadap IPv6 sebesar 2,73 %. Dan jaringan *tunneling* terhadap IPv4 sebesar 13,51 %.
6. Untuk paket dengan protokol FTP penurunan performansi untuk jaringan *tunnel* dibandingkan dengan IPv6 sebesar 2,74 %, jaringan *tunnel* dengan IPv6 over IPv4 terhadap IPv6 1,39 %. Dan jaringan tunneling terhadap IPv4 sebesar 1,37 %.

5.2 Saran

1. Sebagian besar tunnel broker telah diiplementasikan untuk system operasi berbasis UNIX, maka untuk penelitian berikutnya dapat dilakukan desain dan implementasi tunnel broker untuk system operasi berbasis windows.
2. Dalam IPv6 tunnel broker host yang berfungsi sebagai tunnel broker, tunnel server dan DNS server tidak terbatas dalam satu computer server untuk satu tunnel broker bisa lebih dari satu buah server
3. Dari berbagai mekanisme transisi dari IPv4 ke IPv6 yang ada dapat dilakukan perbandingan dan pemilihan mekanisme mana yang mempunyai penurunan performansi yang paling kecil.

DAFTAR PUSTAKA

[1]. Ettikan Kandasamy, “IPv6 Dual Stack Transition Technique Performance Analysis: KAME on FreeBSD as the case”,NTT MSC,Cyberjaya, Malaysia.

[2]. Ettikan Kandasamy, “Application Performance Analysis in Transition Mechanism from IPv4 to IPv6”, Research & Business Development Department, Malaysia.

[3]. Guardini, Ivano, “Migrating From IPv4 to IPv6:planning an effective IPv6 transition”, Global IP Summit 2000.

[4]. “IPv6 Transition Mechanism”, ETRI, Korea.

[5]. Onno W. Purbo, “TCP/IP”,Elex Media Komputindo, 2000.

[6]. RFC 2119, “IP Version 6 Addresing Architecture”, Internet Society, 1998.

[7]. RFC 2529. “Transmission of IPv6 over IPv4 Domain Without Explicit Tunnel”,Internet Society,1999.

[8]. RFC 3053, “IPv6 Tunnel Broker”, Internet Society, 2001.

[9]. S. Tannembaum, Andrew, “Jaringan Komputer”, Prentice Hall Indonesia, 2000.

[10]. Yan Riyanto, “IPv6 : Internet Protokol Generasi Berikut”, Majalah Elektro edisi 9,1997.

Menyetujui makalah Tugas Akhir ini pada tanggal