Halaman Judul

LAPORAN PENELITIAN

ANALISIS KECEPATAN TRANSFER DATA POINT TO POINT TUNNELING PROTOCOL

oleh
W A G I T O, S.T., M.T.
NIDN : 0522126901
NPP : 961080

Mendapat Bantuan Biaya Penelitian dari Puslitbang dan PPM
Semester Ganjil 2013/2014

Sekolah Tinggi Manajemen Informatika dan Komputer
AKAKOM YOGYAKARTA
Tahun 2014

HALAMAN PENGESAHAN

1.	a. Judul Penelitian	: Analisis Kecepatan Transfer Data Point To Point Tunneling Protocol
	b. Bidang Ilmu	: Jaringan Komputer
	c. Kategori	: Implementasi Jaringan Komputer
2.	Ketua Peneliti	
	a. Nama	: Wagito, S.T., M.T.
	b. NIDN	: 0522126901
	c. NPP	: 961080
	d. Pangkat/Golongan	: Pembina Tk 1 / IV B
	e. Jabatan Fungsional	: Lektor Kepala
	f. Jurusan/Prodi	: Teknik Informatika
	g. Alamat Institusi	: Jalan Raya Janti Karang Jambe, Yogyakarta
5.	Waktu Penelitian	: 6 bulan
6.	Biaya Penelitian	:

Yogyakarta, Agustus 2014
Mengetahui

Ketua Prodi

Febri Nova Lenti, S.Si., M.T.
NPP. 961079

Ketua Peneliti

Wagito, S.T., M.T.
NPP. 961080

Menyetujui

Kepala Puslitbang dan PPM
STMIK AKAKOM

Dra. Syamsu Windarti, M.T., Apt
NIP. 19660710 199303 2 001

Kata Pengantar

Puji syukur saya panjatkan ke hadirat Allah S.W.T. karena hanya dengan rahmat dan hidayah-Nya. Berkat pertolongan dan tuntunan-Nya serta dengan berbagai usaha akhirnya penelitian ini berhasil diselesaikan dengan baik.

Penelitian yang berjudul *Analisis Kecepatan Transfer Data Point To Point Tunneling Protocol* dilakukan untuk meneliti pengaruh enkripsi dan kompresi pada transfer *file* melalui kanal VPN PPTP. Kanal VPN PPTP dibentuk antara dua router yang terhubung melalui jaringan Internet. Router yang dipakai adalah Mikrotik. Sebagai server digunakan Mandriva 2010. Protokol transfer data yang dipakai adalah FTP.

Penulis menyadari bahwa hasil penelitian ini masih banyak kekurangannya, sehingga kritik dan saran yang membangun untuk lebih mengembangkan hasilnya sangat diharapkan. Semoga hasil penelitian ini bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Penulis

Daftar Isi

Daftar Gambar

ABSTRAK

Virtual Private Network (VPN) menggunakan *Point to Point Tunneling Protocol* (PPTP) dapat digunakan untuk integrasi jaringan. Jaringan yang terletak pada lokasi berbeda dapat dilakukan integrasi melewati jaringan publik Internet. Kemampuan penyatuan jaringan ini tidak terbatas pada lokasi yang berjauhan. Yang diperlukan adalah bahwa masing-masing jaringan terhubung melalui jaringan publik. Integrasi jaringan dilakukan menggunakan fitur kanal tersembunyi.

Penerapan VPN melibatkan aspek penyembunyian data dan kompresi data. Aspek penyembunyian data berkaitan dengan enkripsi data ketika data ditransfer melalui kanal VPN. Aspek kompresi berkaitan dengan pengurangan ukuran data ketika ditransfer melewati kanal VPN. Proses enkripsi dan kompresi data akan memengaruhi kecepatan transfer data secara keseluruhan. Proses enkripsi menyebabkan transfer data tertunda beberapa saat. Proses kompresi pada satu sisi menyebabkan berkurangnya ukuran data yang ditransfer. Namun proses kompresi memerlukan waktu tersendiri yang berpengaruh pada proses transfer data secara keseluruhan.

Hasil penelitian menunjukkan bahwa enkripsi dan kompresi menyebabkan penurunan kecepatan transfer data melalui kanal VPN PPTP. Salah satu metode kompresi menggunakan use-vj-compression tidak secara signifikan menyebabkan penurunan kecepatan transfer data.

Kata kunci: kecepatan, transfer, VPN, PPTP

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Keberadaan jaringan komputer sangat membantu dalam proses transfer data dari suatu komputer ke komputer lain. Transfer data dapat terjadi antar komputer dalam institusi maupun antar komputer antar institusi bahkan antar komputer yang tidak terbatas institusi. Jaringan komputer sudah menjadi kebutuhan yang sangat penting bagi kehidupan manusia.

Ukuran jaringan komputer sangat bervariasi mulai dari jaringan personal sampai jaringan yang sangat luas tidak terbatas geografi. Ukuran jaringan komputer yang diperlukan sangat tergantung pada besar atau kecilnya institusi. Institusi yang kecil mungkin cukup menggunakan jaringan komputer yang sederhana. Institusi yang punya tempat yang kecil, tentunya hanya memerlukan jaringan komputer yang kecil. Institusi yang punya tempat yang luas dan terdiri dari beberapa lokasi, memerlukan jaringan komputer yang lebih luas.

Suatu masalah timbul apabila antar lokasi pada institusi letaknya cukup jauh. Apabila jarak lokasi institusi sudah di luar jangkau jaringan kabel, jaringan *wireless* dapat dipakai untuk menyatukan lokasi-lokasi tersebut. Jaringan *wireless* punya jangkau yang lebih jauh dibanding jaringan kabel, namun punya

keterbatasan jarak jangkau. Penggunaan jaringan *wireless* juga memerlukan investasi untuk membeli peralatan dan sarana pemasangan jaringan *wireless*. Penggunaan jaringan *wireless* juga memerlukan suatu lokasi yang tidak terhalang antar lokasinya.

Apabila institusi punya beberapa lokasi yang berjarak sangat jauh, maka permasalahan penyatuan jaringan menjadi sesuatu yang rumit. Jaraknya sangat jauh pengertiannya adalah jarak di luar jarak kemampuan media transmisi jaringan. Jarak yang sangat jauh juga punya pengertian apabila jaringan disatukan memerlukan biaya yang mahal.

Pada saat ini jaringan Internet punya penggunaan yang sangat luas. Penggunaan jaringan Internet bagi institusi umumnya hanya sekadar mengakses halaman situs, mengirim *e-mail*, melakukan *chat* atau melakukan komunikasi pada media sosial. Bagi institusi tertentu, jaringan Internet hanya merupakan fasilitas sarana hiburan. Namun bagi institusi lain, Internet menjadi tulang punggung bagi kelancaran kegiatannya atau bahkan menjadi sarana menghasilkan keuntungan.

Kemampuan jaringan komputer dalam kaitannya dengan penyatuan jaringan adalah membuat suatu VPN (*Virtual Private Network*). VPN tidak sekadar menghubungkan dua lokasi yang sangat jauh, namun punya kemampuan untuk menyembunyikan data yang dikirimkan. VPN dapat dilewatkan pada suatu jaringan publik seperti Internet. VPN membuat semacam saluran rahasia (*tunnel*) melintasi jaringan publik.

Penerapan VPN dalam sistem jaringan memerlukan protokol khusus. Beberapa protokol dapat digunakan untuk penerapan VPN, salah satunya adalah PPTP (*Point-to-Point Tunneling Protocol*). PPTP adalah suatu protokol jaringan yang memungkinkan pengiriman data secara aman dari klien yang terpisah jauh kepada *server* dengan cara membuat VPN melalui jaringan data berbasis TCP/IP.

Penerapan VPN menggunakan PPTP melibatkan aspek penyembunyian data dan kompresi data. Aspek penyembunyian data berkaitan dengan enkripsi data ketika data ditransmisikan melalui kanal VPN. Aspek kompresi berkaitan dengan pengurangan ukuran data ketika ditransmisikan melewati kanal VPN. Proses enkripsi dan kompresi data akan memengaruhi kecepatan transfer data secara keseluruhan.

Pada penelitian ini dicoba untuk mengamati bagaimana pengaruh proses enkripsi dan kompresi terhadap kecepatan transfer *file* melalui kanal VPN menggunakan protokol PPTP. Proses enkripsi menyebabkan transfer data tertunda beberapa saat. Namun seberapa pengaruhnya terhadap kecepatan transfer data melalui kanal VPN PPTP perlu diteliti lebih lanjut. Proses kompresi pada satu sisi menyebabkan berkurangnya ukuran data yang ditransfer. Namun proses kompresi memerlukan waktu tersendiri yang menyebabkan proses transfer data tertunda secara keseluruhan. Dengan demikian, perlu juga diteliti lebih lanjut seberapa pengaruh kompresi terhadap kecepatan transfer data melalui kanal VPN PPTP.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian adalah bagaimana pengaruh proses enkripsi dan kompresi terhadap kecepatan transfer *file* melalui kanal VPN menggunakan protokol PPTP.

1.3 Batasan Masalah

Batasan yang perlu diperhatikan dalam kaitan dengan kemungkinan masalah yang muncul penelitian adalah:

- penelitian dititikberatkan pada metode pengukuran kecepatan transfer data melalui kanal VPN PPTP,
- kecepatan transfer data yang diteliti adalah kecepatan transfer *download* dan *upload*.
- pengaruh enkripsi pada proses transfer data melalui kanal VPN PPTP.
- pengaruh kompresi pada proses transfer data melalui kanal VPN PPTP
- tidak membicarakan tentang algoritme enkripsi maupun kompresi yang dipakai, namun meneliti pengaruhnya pada kecepatan transfer data melalui kanal VPN PPTP.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dari hasil penelitian yang dilakukan adalah sebagai berikut.

- Membuat metode pengukuran kecepatan transfer data melalui kanal VPN PPTP.

- Mengamati pengaruh enkripsi data terhadap kecepatan transfer data melalui kanal VPN PPTP.
- Mengamati pengaruh kompresi data terhadap kecepatan transfer data melalui kanal VPN PPTP

1.5 Manfaat Penelitian

Manfaat penelitian berkaitan dengan enkripsi data adalah bisa melihat seberapa besar pengaruhnya pada kecepatan transfer data melalui kanal VPN PPTP. Selanjutnya pengaruh enkripsi dibandingkan dengan sisi keamanan yang didapat. Manfaat penelitian yang berkaitan dengan kompresi data terhadap kecepatan transfer data dapat digunakan untuk meningkatkan kinerja kanal VPN PPTP.

1.6 Target Luaran

Hasil penelitian direncanakan dilakukan publikasi dan seminasi pada kegiatan ilmiah. Kegiatan ilmiah yang diharapkan bisa diikuti adalah seminar nasional yang berkaitan dengan jaringan komputer.

BAB 2 TINJAUAN PUSTAKA

Beberapa penelitian berkaitan dengan integrasi jaringan komputer pernah dilakukan. Beberapa penelitian berkaitan dengan integrasi jaringan yang pernah dilakukan antara lain sebagai berikut.

Penelitian yang dilakukan Muhammad Muslich dan Fatah Yasin berjudul *Virtual Private Network* Berbasis *IP Security* Dengan Linux *Free Secure Wide Area Network* mengisyaratkan pengaruh enkripsi pada penerapan *Virtual Private Network* berbasis *IP Secure* (IPSec). Tujuan penelitian ini untuk menerapkan *Virtual Private Network* berbasis IPSec. Penelitian ini belum membahas lebih lanjut tentang pengaruh enkripsi data .

Pengaruh kompresi header data pernah diteliti oleh Agung Sediyono dan Alitalia Rahman dalam penelitian berjudul Pengaruh Kompresi *Header* Mikrotik Pada Transfer *Rate* Di Jaringan VPN PPTP. Tujuan dari penelitian ini adalah untuk mengetahui seberapa signifikan manfaat kompresi *header* yang ada di Mikrotik untuk tujuan peningkatan kinerja VPN PPTP. Apabila diketahui karakteristik transfer data, maka dapat diambil kebijakan yang tepat dalam implementasi.

Pada studi kasus yang dilakukan oleh Nova Rusydi Setyawan pada tahun 2011 dengan judul “Implementasi VLAN *Trunk Protocol* (VTP) melalui *Ethernet*

over Internet Protocol (EoIP) *Tunnel* pada Mikrotik RouterOS" membahas penggunaan *Ethernet over Internet Protocol* pada sistem operasi Mikrotik RouterOS untuk melewatkan VLAN Trunk Protocol. Protokol yang dipakai adalah EoIP yang merupakan salah satu bentuk protokol *tunneling*.

Protokol EoIP memungkinkan pembentukan saluran khusus (tunnel) Ethernet antara dua *router* di atas hubungan IP. Antarmuka EoIP muncul di atas antarmuka Ethernet. Pada penelitian dilakukan kombinasi dengan fitur VLAN untuk konfigurasi jaringan dengan memanfaatkan Mikrotik RouterOS™ sebagai peralatan utama. Fitur VLAN hanya mempermudah dalam konfigurasi jaringan.

Penelitian Dedy Cahyadi program studi Ilmu Komputer 2010, FMIPA Universitas Mulawarman dengan judul "Pemanfaatan Fitur *Tunneling* Menggunakan *Virtual Interface* EoIP di Mikrotik RouterOS untuk koneksi *Bridging* Antar Kantor Melalui Jaringan ADSL Telkom Speedy" yang membahas tentang pemanfaatan koneksi EoIP dengan Mikrotik RouterOS untuk Integrasi antar kantor melalui jaringan ADSL Telkom Speedy, sehingga jaringan antar kantor menjadi satu jaringan.

Salah satu fitur yang bisa dikembangkan dari protokol EoIP adalah pembentukan jembatan *bridge*. Ketika fungsi *bridge* pada router diaktifkan, semua lalu-lintas Ethernet (protokol Ethernet) akan dilewatkan pada *bridge*. *Bridge* berlaku seperti antarmuka dan kabel fisik Ethernet antara dua *router*. Sebagai saluran publik, pada penelitian ini digunakan jaringan *broadband* ADSL Telkom Speedy.

Penelitian dari Nanda Pramudya, Universitas Duta Wacana tahun 2009 tentang "Implementasi dan Analisis *Point-to-Point Tunneling Protocol* Serta Ethernet Over Internet Protocol Sebagai Metode Untuk Membuat *Virtual Private Network*" yang membahas tentang implementasi dan analisis *Point-to-Point Tunneling Protocol* dan *Ethernet Over Internet Protocol* digunakan sebagai VPN.

Pada penelitian ini juga dimanfaatkan protokol EoIP untuk integrasi jaringan. Sebagai saluran digunakan protokol PPTP. Dengan demikian, pada penelitian ini menggabungkan protokol VPN PPTP dengan protokol EoIP. Namun pada penelitian ini tidak mengaktifkan fitur *bridge*.

Penelitian Kukuh Prasetyo dari Institut Teknologi TELKOM tentang "Analisis Performasi Pada Penggunaan IPsec dan PPTP Untuk *Internet Protocol Television* (IPTV)" yang membahas perbandingan di antara ke dua protokol IPsec dan PPTP dengan parameter yang dibandingkan dan diuji yaitu : pengaruh autentikasi, enkripsi dan enkapsulasi yang berbeda diantara dua protokol tersebut terhadap IPTV.

Penelitian tersebut membandingkan penggunaan protokol PPTP dan IPsec (IP *secure*). Protokol PPTP sebetulnya sudah menentukan sebagai saluran khusus untuk integrasi jaringan. IPsec dapat diaktifkan pada Mikrotik RouterOS™. Secara bawaan fitur ini tidak aktif. Implementasi diuji pengaruhnya terhadap IPTV. Penggunaan IPsec pada satu sisi bisa digunakan untuk meningkatkan keamanan, namun pada sisi lain akan menambah beban pekerjaan *router*. Dal hal ini bisa saja memengaruhi kinerja sistem jaringan.

Pemanfaatan VLAN hanya memudahkan dalam konfigurasi jaringan namun sebetulnya menurunkan kinerja jaringan. Sebetulnya VLAN secara dasar sistem operasi hanya memanfaatkan kemampuan IP alias pada satu antarmuka fisik. IP alias berupa antarmuka virtual. Dengan demikian VPN sebetulnya cukup membebani kerja router.

Protokol EoIP memungkinkan pembentukan saluran khusus (*tunnel*) Ethernet antara dua router di atas hubungan IP. Antarmuka EoIP muncul di atas antarmuka Ethernet. Protokol PPTP memungkinkan pembentukan saluran pada integrasi jaringan. Apabila protokol EoIP digabungkan dengan protokol PPTP, maka yang terjadi adalah pembentukan saluran di dalam saluran. Hal demikian mungkin meningkatkan keamanan, namun jelas membebani router.

Pada penelitian sebelumnya "Implementasi VPN PPTP Untuk Integrasi Jaringan" dicoba dirancang integrasi dua jaringan dengan memanfaatkan protokol PPTP. Salah satu yang diteliti pada penelitian ini adalah konsekuensi integrasi jaringan pada beberapa aspek implementasi jaringan yaitu pertukaran data antar jaringan menggunakan *share*, pemendekan jalur *routing*, koneksi HTTP dan koneksi basisdata.

Pada penerapan PPTP terdapat beberapa variabel yang perlu diteliti lebih pengaruhnya pada implementasi aplikasi jaringan komputer. Pada PPTP terdapat variabel yang dapat diatur yaitu kompresi data dan enkripsi data. Pengaruh kompresi dan enkripsi belum diteliti lebih lanjut berkaitan dengan pertukaran data antar jaringan yang dilakukan integrasi menggunakan metode PPTP. Pada

penelitian ini dicoba diteliti bagaimana pengaruh kompresi data dan enkripsi pada aspek pertukaran data.

BAB 3 TEORI

3.1 PPTP

Point-to-Point Tunneling Protocol (PPTP) adalah sebuah metode untuk menerapkan VPN. PPTP menggunakan saluran kontrol atas TCP dan kanal operasi GRE untuk membungkus paket PPP.

Spesifikasi PPTP tidak menggambarkan enkripsi atau autentikasi fitur dan bergantung pada *Point-to-Point Protocol* yang disalurkan untuk menerapkan fungsi keamanan. Namun, yang paling umum, implementasi pengiriman PPTP pada keluarga produk Microsoft Windows menerapkan berbagai tingkat autentikasi dan enkripsi asli sebagai fitur standar dari *stack* PPTP Windows. Tujuan penggunaan protokol ini adalah untuk memberikan tingkat keamanan dan tingkat akses *remote* sebanding dengan produk khas VPN.

Spesifikasi untuk PPTP diterbitkan pada bulan Juli 1999 dalam bentuk RFC 2637 (The Internet Society, 1999) dan dikembangkan oleh konsorsium vendor yang dibentuk oleh Microsoft, Ascend Communications (sekarang bagian dari Alcatel-Lucent), 3Com, dan lain-lain. PPTP belum diusulkan atau disahkan sebagai standar oleh *Internet Engineering Task Force*.

Deskripsi Protokol

Kanal PPTP dipakai oleh komunikasi *peer* pada TCP port 1723. Koneksi TCP ini kemudian digunakan untuk memulai dan mengelola sebuah kanal teman GRE kedua yang sama.

Format paket PPTP GRE non standar, mencakup *field* pengakuan tambahan menggantikan *field routing* khas pada *header* GRE. Namun, seperti dalam koneksi GRE normal, paket modifikasi GRE tersebut secara langsung dikemas menjadi paket-paket IP dan dipandang sebagai protokol IP nomor 47. Kanal GRE yang digunakan untuk membawa kemasan paket PPP, memungkinkan membentuk kanal dari sembarang protokol yang dapat dibawa dalam PPP, termasuk IP, NetBEUI dan IPX. Dalam penerapan Microsoft, lalu-lintas kanal PPP dapat dilakukan autentikasi dengan PAP, CHAP, MS-CHAP v1/v2.

Implementasi

PPTP adalah protokol VPN pertama yang didukung oleh Microsoft Dial-up Networking. Semua rilis Microsoft Windows sejak Windows 95 OSR2 dibundel dengan klien PPTP, meski pun dibatasi hanya 2 koneksi bersamaan *outbound*. Microsoft Windows Mobile 2003 dan yang lebih tinggi juga mendukung protokol PPTP. Layanan *routing* dan *remote access* untuk Microsoft Windows berisi server PPTP. Penerapan Microsoft menggunakan DES tunggal pada protokol autentikasi MS-CHAP yang banyak ditemukan, cocok untuk kebutuhan perlindungan data. (Marsh, R., 2012)

Windows Vista dan yang lebih tinggi mendukung penggunaan PEAP dengan PPTP. Mekanisme autentikasi yang didukung adalah PEAPv0 / EAP-MSCHAPv2 (*password*) dan PEAP-TLS (*smartcard* dan sertifikat). Windows Vista menghilangkan dukungan untuk menggunakan protokol MSCHAP-v1 untuk autentikasi koneksi akses *remote*. (Anonim, 2012)

Dukungan *server-side* Linux untuk PPTP disediakan oleh *daemon poptop* dan modul *kernel* untuk PPP dan MPPE. *Client-side* Linux untuk penerapan PPTP muncul pada tahun 1997, (Ananian, S., 2013) tetapi implementasi secara luas *server-side* Linux PPTP pertama dikembangkan oleh Matthew Ramsay pada tahun 1999 (Ramsay, M., 2000) dan awalnya didistribusikan di bawah GNU GPL oleh Moreton Bay. Namun, distribusi Linux awalnya tidak memiliki dukungan penuh PPTP karena MPPE dilindungi paten. Dukungan penuh MPPE ditambahkan ke *kernel* Linux dalam rilis 2.6.14 pada 28 Oktober, 2005. SuSE Linux 10 adalah distribusi Linux pertama yang menyediakan klien PPTP yang lengkap. Ada juga ACCEL-PPP - PPTP/L2TP/PPPoE server untuk Linux yang mendukung PPTP dalam mode *kernel*.

Keamanan

PPTP telah menjadi subjek dari banyak analisis keamanan dan kerentanan keamanan yang serius telah ditemukan dalam protokol. Kerentanan yang diketahui berhubungan dengan protokol autentikasi yang mendasari potokol PPP yang digunakan, desain protokol MPPE serta integrasi antara MPPE dan

autentikasi PPP untuk sesi pembangkitan kunci (Schmidt, J., 2012).

3.2 PPTP Pada Mikrotik

PPTP adalah kanal yang aman untuk mengangkut lalu-lintas IP menggunakan PPP. PPTP merangkum PPP dalam garis virtual yang berjalan di atas IP. PPTP menggabungkan PPP dan MPPE (*Microsoft Point to Point Encryption*) untuk membuat link terenkripsi. Tujuan protokol ini adalah untuk membuat koneksi yang aman dikelola dengan baik antar router serta antara router dan klien PPTP (klien tersedia untuk dan/atau termasuk di hampir semua OS termasuk Windows). (Mikrotik, 2008)

Multilink PPP (MP) didukung untuk menyediakan MRRU (kemampuan untuk mengirimkan paket berukuran penuh 1500 dan lebih besar) dan *bridging* lebih dari PPP *link* (menggunakan *Bridge Control Protocol* (BCP) yang memungkinkan pengiriman *frame* Ethernet stadar di atas PPP *link*) . Dengan cara ini adalah mungkin untuk setup *bridging* tanpa EoIP. *Bridge* harus memiliki alamat MAC yang dapat diatur secara administrasi atau antarmuka Ethernet-like di dalamnya, seperti PPP *link* tidak memiliki alamat MAC. (Mikrotik, 2008)

PPTP termasuk PPP autentikasi dan perhitungan untuk setiap koneksi PPTP. Autentikasi dan perhitungan dari masing-masing sambungan penuh dapat dilakukan melalui klien RADIUS atau lokal. PPTP mendukung enkripsi MPPE 40bit RC4 dan MPPE 128bit RC4. (Mikrotik, 2008)

Lalu-lintas PPTP menggunakan port TCP 1723 dan IP GRE (*Generic*

Routing Encapsulation, IP protokol ID 47), seperti yang diberikan oleh *Internet Assigned Numbers Authority* (IANA). PPTP dapat digunakan dengan *Firewall* dan *router* dengan mengizinkan lalu-lintas paket pada port TCP 1723 dan lalu- lintas pada protokol 47 yang akan disalurkan melalui *Firewall* atau *router*.

PPTP Client

Sub-menu: `/interface pptp-client`

Properti-propert untuk Klien PPTP

- ***add-default-route*** (*yes* | *no*; Default: **no**)
- ***allow*** (*mschap2*|*mschap1*|*chap*|*pap*; Default: **mschap2, mschap1, chap, pap**)
- ***connect-to*** (*IP*; Default:)
- ***default-route-distance*** (*byte [0..255]*; Default: **1**)
- ***dial-on-demand*** (*yes* | *no*; Default: **no**)
- ***disabled*** (*yes* | *no*; Default: **yes**)
- ***keepalive-timeout*** (*integer*; Default: **60**)
- ***max-mru*** (*integer*; Default: **1460**)
- ***max-mtu*** (*integer*; Default: **1460**)
- ***mrru*** (*disabled* | *integer*; Default: **disabled**)
- ***name*** (*string*; Default:)
- ***password*** (*string*; Default: **""**)
- ***profile*** (*name*; Default: **default-encryption**)

- ***user*** (*string*; Default:)

PPTP Server

Sub-menu: `/interface pptp-server`

Sub-menu ini menunjukkan antarmuka untuk setiap klien PPTP yang terhubung. Sebuah antarmuka dibuat untuk setiap terowongan didirikan untuk server yang diberikan. Ada dua jenis antarmuka dalam konfigurasi PPTP server sebagai berikut.(Mikrotik, 2008)

- Antarmuka statis ditambahkan secara administratif apabila ada kebutuhan untuk referensi nama antarmuka tertentu (dalam aturan *Firewall* atau di tempat lain) yang diciptakan untuk pengguna tertentu.
- Antarmuka dinamis ditambahkan ke daftar ini secara otomatis setiap kali pengguna terhubung dan *username* yang tidak cocok dengan entri statis yang ada (atau dalam hal entri aktif sudah, karena tidak mungkin ada dua antarmuka terowongan terpisah direferensikan dengan nama yang sama).

Antarmuka dinamis muncul ketika pengguna menghubungkan dan menghilang setelah pengguna terputus, sehingga tidak mungkin untuk referensi terowongan dibuat untuk itu digunakan dalam konfigurasi router (misalnya, dalam *Firewall*), jadi jika pengguna perlu aturan tegas untuk *user* tersebut, sebaiknya dibuat entri statis entri apabila tidak aman untuk menggunakan konfigurasi dinamis.

Properti-propert untuk Server PPTP

- ***authentication*** (*pap*|*chap*|*mschap1*|*mschap2*; Default: **mschap1,mschap2**)
- ***default-profile*** (*name*; Default: **default-encryption**)
- ***enabled*** (*yes* | *no*; Default: **no**)
- ***keepalive-timeout*** (*time*; Default: **30**)
- ***max-mru*** (*integer*; Default: **1460**)
- ***max-mtu*** (*integer*; Default: **1460**)
- ***mrru*** (*disabled* | *integer*; Default: **disabled**)

3.3 LFTP

LFTP adalah program transfer *file* yang memungkinkan FTP, HTTP dan koneksi canggih lain ke host yang berbeda. Jika situs sudah ditentukan, maka LFTP akan terhubung ke situs lain menggunakan sambungan yang telah telah dibentuk dengan perintah terbuka. (Lukyanov, A.V., 2014)

LFTP dapat menangani beberapa metode akses berkas - FTP, FTPS, HTTP, HTTPS, HFTP, FISH, SFTP dan berkas (HTTPS dan FTPS hanya tersedia bila LFTP dikompilasi dengan GNU TLS atau pustaka OpenSSL). Metode dapat ditentukan menggunakan perintah, misalnya http: //www.us.kernel.org/pub/linux. HFTP adalah protokol *ftp-over-http-proxy*. Protokol ini dapat digunakan secara otomatis sebagai pengganti FTP jika ftp:proxy diatur ke http://Proxy[:port]. FISH adalah protokol yang bekerja di atas suatu koneksi ssh pada akun Unix. SFTP

adalah protokol yang diterapkan dalam SSH2 sebagai sub sistem SFTP. (Lukyanov, A.V., 2014)

Selain protokol FTP-like, LFTP memiliki dukungan untuk protokol BitTorrent sebagai perintah *torrent*. *Seeding* juga didukung pada protokol ini.

Setiap operasi pada LFTP dapat diandalkan, yaitu kesalahan non-fatal dapat ditangani dengan benar dan operasi diulang. Jika *download* istirahat, maka akan dimulai dari titik secara otomatis. Bahkan jika FTP Server tidak mendukung perintah REST, LFTP akan mencoba untuk mengambil *file* dari awal sampai *file* ditransfer sepenuhnya. (Lukyanov, A.V., 2014)

LFTP memiliki sintaksis perintah *shell-like* yang memungkinkan menjalankan beberapa perintah secara paralel pada proses latar (&). Hal ini juga memungkinkan untuk mengelompokkan perintah dalam tanda () dan melakukan eksekusi pada latar belakang. Semua pekerjaan latar dapat dieksekusi dalam proses tunggal. Pekerjaan latar depan dapat dibawa ke latar belakang dengan perintah ^Z (c-z) dan kembali dengan perintah *wait* (atau fg yang merupakan alias untuk *wait*). Untuk melihat daftar pekerjaan yang berjalan, digunakan perintah *jobs*. Beberapa perintah memungkinkan pengarahan *output* (cat, ls, dan sebagainya) ke *file* atau melalui pipa menggunakan perintah eksternal. Perintah dapat dieksekusi secara kondisional berdasarkan status penghentian perintah sebelumnya (&&, ||).

Jika keluar LFTP sebelum semua pekerjaan selesai, maka LFTP akan berpindah ke mode *nohup* pada latar secara otomatis. Hal yang sama terjadi pada

modem ketika *hangup* atau ketika *xterm* ditutup.

LFTP telah *built-in mirror* yang dapat melakukan *download* atau memperbarui seluruh pohon direktori. Ada juga reverse mirror (mirror -R) yang melakukan *upload* atau update pada pohon direktori pada server. *Mirror* juga dapat melakukan sinkronisasi direktori antara dua server *remote*, menggunakan FXP apabila tersedia.

Ada perintah *at* untuk memulai pekerjaan di waktu yang ditentukan dalam konteks perintah *queue* untuk mengantrikan perintah untuk dieksekusi secara sekuensial untuk server tersebut.

Pada saat *startup*, LFTP melakukan eksekusi /etc/lftp.conf dan kemudian ~ /.lftprc dan ~/.lftp/rc. Pengguna dapat menempatkan alias dan perintah *set* pada *file* tersebut. Beberapa pengguna lebih suka melihat protokol *debug* secara penuh. Perintah *debug* digunakan untuk mengaktifkan *debug*. Perintah *debug* 3 digunakan untuk melihat pesan *greeting* dan kesalahan. (Lukyanov, A.V., 2014)

LFTP memiliki sejumlah variabel yang dapat diatur. Untuk melihat semua variabel dan nilai-nilainya digunakan perintah *set -a*. untuk melihat daftar default digunakan perintah *set -d*. Nama variabel dapat disingkat dan prefiks dapat dihilangkan asalkan tidak menjadikan ambigu.

Jika LFTP dikompilasi dengan OpenSSL (configure –with-openssl), maka LFTP mendukung perangkat lunak yang dikembangkan oleh OpenSSL Project untuk digunakan dalam OpenSSL *Toolkit*. (http://www.openssl.org/)

3.4 Wput

Wput adalah utilitas gratis yang mampu meng-*upload file* ke ftp-*server*. Wput berkemampuan non-interaktif dan latar belakang. Hal ini dapat meng-*upload file* atau seluruh direktori dan dimaksudkan untuk menjadi klien yang bagus bahkan untuk koneksi yang tidak stabil dan karena itu wput akan mencoba untuk meng-*upload* ulang *file*, jika koneksi putus.(Fritsch, H. , 2014)

Wput mendukung fitur melanjutkan, sehingga secara otomatis terus meng-*upload* dari titik di mana *upload* sebelumnya berhenti, yang berarti pengguna dapat mematikan Wput kapan saja dan akan (jika *server* FTP *remote* mendukung, yang paling mungkin terjadi) menyelesaikan *file upload* secara parsial. (Fritsch, H. , 2014)

Wput mendukung koneksi melalui proxy, yang memungkinkan pengguna untuk menggunakannya dalam lingkungan yang hanya dapat mengakses Internet melalui proxy atau untuk menyediakan anonimitas dengan menyembunyikan alamat IP pengguna ke *server*. Untuk proxy SOCKSv5, Wput mendukung juga modus mendengarkan, memungkinkan pengguna untuk menggunakan *port-mode* FTP melalui *proxy*. Fitur ini berguna apabila FTP jauh berada di belakang *Firewall* atau *Gateway*. (Fritsch, H. , 2014)

Wput mendukung *timestamping*, sehingga akan (dalam kasus yang ideal dan jika *timestamping* diaktifkan) hanya meng-*upload file*, yang lebih baru dari *file remote*.

Tingkat *upload* Wput dapat dibatasi, sehingga Wput tidak akan

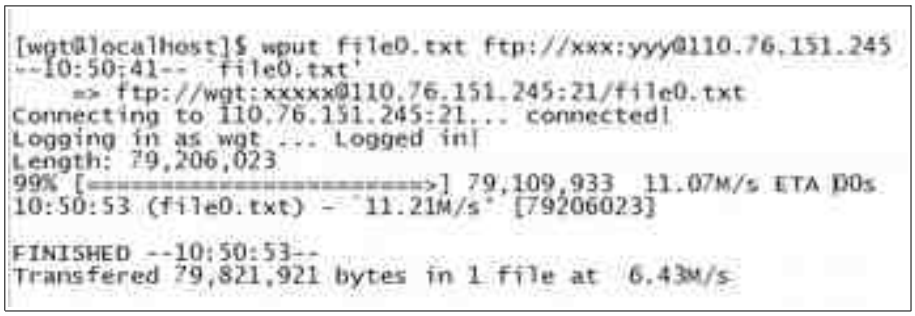
menghabiskan semua *bandwidth* yang tersedia. Wput pertama membaca URL dari baris perintah, dan asosiasi *file* pertama dengan URL pertama, *file* kedua dengan URL kedua an seterusnya, kemudian mengirimkan kombinasi berkas /URL yang sudah lengkap. Dalam situasi yang mana lebih dari satu URL dari *file* yang ditentukan, Wput mencoba menebak nama *file* lokal dari URL. Setelah itu, Wput menggunakan --input *file* (jika ada) dan membaca URL menggunakan skema yang sama. Jika ada masih tersisa *file*, tetapi tidak ada URL yang ditetapkan, Wput menggunakan URL terakhir yang diketahui untuk masing-masing *file*.

Jadi pengguna dapat menentukan misalnya satu URL dan membaca semua nama *file* dari sebuah *file*. Bisa juga digunakan Wput * txt ftp://host, untuk mentransfer semua * txt-*file*. Supaya aman, dianjurkan untuk memasok *file* sebelum URL.

Jika Wput memiliki URL tanpa nama *file* yang sesuai, Wput mencoba menebak lokasi *file* pada lokal. misalnya pengguna menggunakan perintah Wput ftp://host/direktori/path/file Wput akan melihat keluar untuk /direktori/path/file. Jika tidak ditemukan, Wput mencari ./directory/path/file, ./path/file dan ./file. (Fritsch, H. , 2014)

Suatu fitur Wput yang sangat bermanfaat dalam penelitian ini adalah kemampuannya untuk menampilkan informasi kecepatan *upload* ketika proses transfer*file* sudah selesai. Informasi kecepatan *upload* ditampilkan baris terakhir. Pengguna dapat mengambil informasi ini secara terprogram menggunakan sekrip Bash pada sistem operasi Linux. Contoh tampilan informasi ketika Wput

melakukan proses *upload* ditampilkan dalam Gambar 2.1.



Gambar 2.1 Tampilan Wput

Pada Gambar 2.1 ditampilkan kecepatan *upload file* sebesar 6,4 M/s (MegaByte/detik). Beberapa tampilan dimodifikasi untuk keamanan, terutama *username* dan *password*.

3.5 Wget

GNU Wget adalah utilitas gratis untuk di-*download* non-interaktif *file* dari Web, mendukung protokol HTTP, HTTPS dan FTP serta pengambilan melalui proxy HTTP. (Niksic, H., 2014)

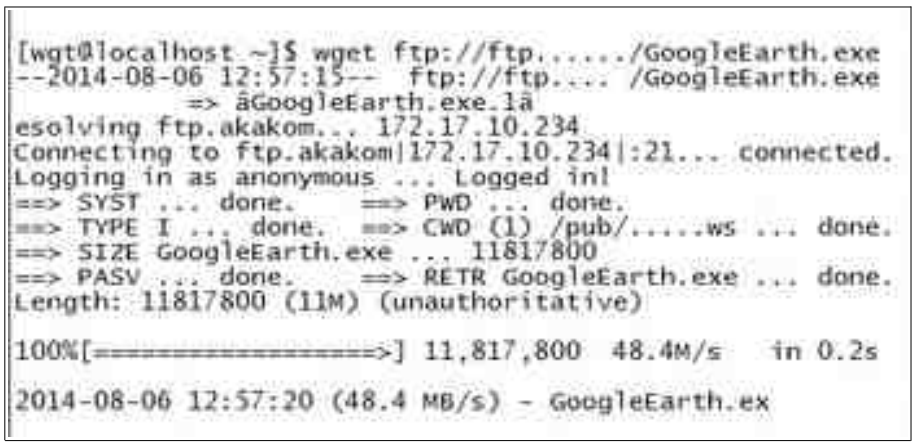
Wget adalah non-interaktif, yang berarti bahwa Wget dapat bekerja di latar belakang, sementara pengguna tidak login. Hal ini memungkinkan pengguna untuk memulai pengambilan suatu dan memutuskan sambungan dari sistem, membiarkan Wget menyelesaikan pekerjaan. (Niksic, H., 2014)

Sebaliknya, sebagian besar browser Web memerlukan kehadiran peran pengguna, yang dapat menjadi penghalang besar ketika melakukan transfer

banyak data. Wget dapat mengikuti link pada halaman HTML, XHTML, dan CSS untuk membuat versi lokal itus web jarak jauh, sepenuhnya menciptakan struktur direktori dari situs asli. Hal ini kadang-kadang disebut sebagai *recursive downloading.* Wget dapat diarahkan untuk mengubah link dalam *download file* untuk menunjuk pada *file* lokal, untuk dilihat secara *offline*. (Niksic, H., 2014)

Wget telah dirancang untuk ketahanan melalui koneksi jaringan lambat atau tidak stabil jika *download* gagal karena masalah jaringan, akan terus mencoba kembali sampai seluruh *file* diambil. Jika *server* mendukung *regetting*, Wget akan memberi instruksi kepada server untuk melanjutkan *download*.

Suatu fitur Wget yang sangat bermanfaat dalam penelitian ini adalah kemampuannya untuk menampilkan informasi kecepatan *download* ketika proses transfer *file* sudah selesai. Informasi kecepatan *download* ditampilkan baris terakhir dalam tanda kurung (). Pengguna dapat mengambil informasi ini secara terprogram menggunakan sekrip Bash pada sistem operasi Linux. Tampilan informasi ketika Wget melakukan proses *download* ditunjukkan pada Gambar 2.2.



Gambar 2.2 Tampilan Wget

Pada Gambar 2.2 ditampilkan kecepatan *download file* yang dicapai adalah sebesar 48,4 MB/s (MegaByte/detik). Beberapa tampilan disembunyikan untuk keamanan, terutama *username* dan *password*.

3.6 Winbox

Konfigurasi Mikrotik RouterOS™ dapat dilakukan dengan dua cara yaitu: melakukan *login* ke *server* Mikrotik menggunakan utilitas Telnet atau SSH dan menggunakan utilitas Winbox. Konfigurasi dengan cara *login* ke *server* Mikrotik dilakukan dengan cara memberi perintah-perintah tertentu dari *shell* Mikrotik. Konfigurasi dengan cara ini cukup sulit bagi pengguna pemula. Winbox merupakan utilitas yang disediakan Mikrotik untuk menangani konfigurasi secara visual. (Mikrotik, 2008)

Semua pengaturan Mikrotik hampir seluruhnya disediakan secara visual oleh Winbox. Utilitas Winbox menyediakan banyak menu antara lain *Interfaces*, *Wireless*, *Bridge*, *Mesh*, PPP, IP, *Routing*, *Port*, *System*, Terminal dan sebagainya. Menu penting yang berkaitan dengan penelitian ini adalah IP *Address* dan IP *Routes*. Menu IP *Address* berkaitan dengan pemberian alamat IP pada antarmuka *router*. IP *Routes* berkaitan dengan penyusunan tabel *routing* pada *router*. (Mikrotik, 2008)

Selain itu, Winbox juga menyediakan terminal untuk melakukan

konfigurasi menggunakan *shell* dan pengaturan lain yang belum disediakan pada fasilitas visual. Salah satu fasilitas yang membuat Winbox sangat fleksibel adalah ketersediaan pembuatan sekrip program. Dengan sekrip ini, Mikrotik RouterOS™ dapat dikendalikan secara terprogram.

BAB 4 METODE PENELITIAN

4.1 Bahan Penelitian

Bahan yang digunakan dalam penelitian berupa *file* dengan bermacam-macam tipe dan ukuran. Tipe *file* yang dipakai sebagai contoh dipakai dalam proses transfer data. *File* yang dipakai berupa *file* teks dan biner. *File* teks yang dipakai berupa *file rich text format*. *File* biner yang dipakai berupa *file* hasil kompresi. *File* yang sudah dilakukan kompresi dalam bentuk kompresi gz.

Ukuran *file* yang dipakai dalam penelitian meliputi bermacam-macam ukuran. Ukuran *file* dipilih sedemikian, sehingga waktu yang diperlukan untuk proses transfer cukup untuk diukur. Ukuran *file* bervariasi dari puluhan sampai ratusan Mega Byte. Sebelumnya dilakukan uji coba awal untuk pemilihan dan penentuan seberapa kira-kira ukuran *file* yang sesuai.

4.2 Alat

Alat yang digunakan dalam penelitian ini berupa perangkat keras dan perangkat lunak. Perangkat lunak yang digunakan dalam penelitian adalah sebagai berikut.

- Sistem Operasi Linux Mandriva 2010.

- Perangkat lunak virtualisasi Virtual Box.
- Perangkat lunak *download* wget 1.12.
- Perangkat lunak *upload* wput 0.6.
- *Server* FTP Proftpd 1.3.3
- Traceroute
- Ping
- Grep
- Awk

Perangkat keras yang digunakan dalam penelitian adalah komputer personal dengan spesifikasi sebagai berikut.

- Prosesor Intel Atom N570.
- RAM 2GB.
- Tipe sistem 32 bit.

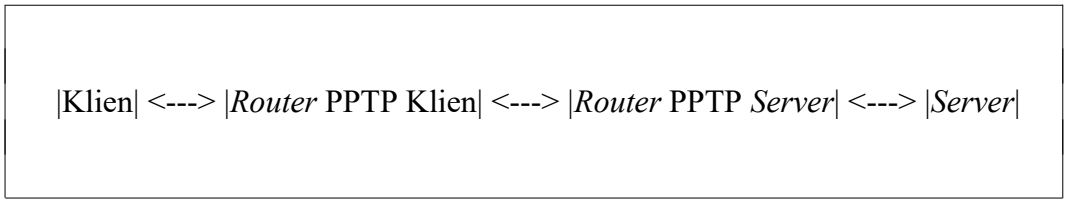
4.3 Jalan Penelitian

Penelitian dilakukan dalam dua tahap yaitu mulai perancangan konfigurasi perangkat keras dan perancangan menggunakan perangkat lunak Virtual Box. Perancangan konfigurasi perangkat keras untuk mengidentifikasi seluruh perangkat keras yang terlibat dalam sistem jaringan serta konfigurasi pada masing-masing mesin. Perancangan pada Virtual Box dimaksudkan untuk memudahkan proses percobaan perhitungan kecepatan *upload* dan *download*.

Penggunaan Virtual Box dapat mengabaikan keruwetan akibat pemasangan kabel yang tidak sempurna.

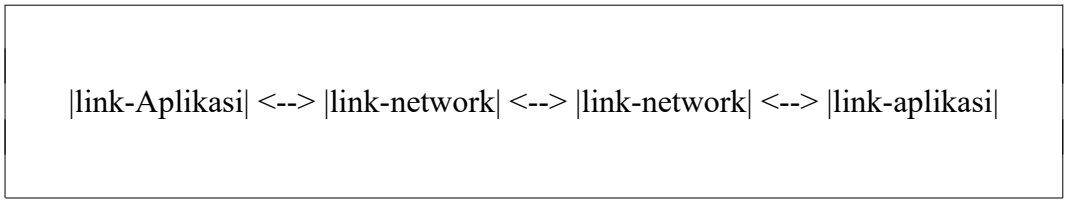
4.3.1 Rancangan Hardware

Mesin yang terlibat dalam penelitian meliputi satu mesin Klien, satu mesin *Server* dan dua mesin *Router*. Diagram jaringan untuk penelitian ditunjukkan pada Gambar 4.1.



Gambar 4.1 Diagram Jaringan

Model TCP/IP digunakan untuk menjelaskan komunikasi data yang berkaitan dengan diagram jaringan tersebut. Secara skematis, model TCP/IP untuk menjelaskan komunikasi data antara dua mesin yang terhubung menggunakan jaringan komputer ditampilkan pada Gambar 4.2.



Gambar 4.2 Model Komunikasi Data TCP/IP

Pada masing-masing mesin, jumlah *layer* yang dialami oleh paket data berbeda-beda. Pada mesin Klien dan *Server*, terdapat empat *layer* yang dialami oleh paket data. *Layer* tersebut adalah *link*, network, *transport* dan *application*. Pada mesin Router, paket data melewati dua *layer*. Layer tersebut adalah *link* dan *network*.

Tahapan-tahapan yang dilakukan pada penelitian untuk pengukura pengaruh enkripsi dan kompresi pada kecepatan transfer data dapat diuraikan sebagai berikut.

- Menyiapkan mesin *Router*
- Menyiapkan mesin *Server*
- Menyiapkan mesin Klien
- Konfigurasi Mandriva 2010 pada mesin *Router*, *Server* dan Klien
- Konfigurasi *server* PPTP pada mesin *Server*
- Konfigurasi *server* FTP pada mesin *Server*
- Menyiapkan *file* percobaan pada mesin *Server*
- Konfigurasi klien PPTP pada mesin Klien
- Instalasi perangkat lunak *download*
- Instalasi perangkat lunak *upload*
- Percobaan *download file* dengan berbagai macam ukuran
- Percobaan *download file* dengan berbagai macam tipe
- Percobaan *upload file* dengan berbagai macam ukuran
- Percobaan *upload file* dengan berbagai macam tipe

- Analisis hasil percobaan transfer *file*.

4.3.2 Rancangan Virtual Box

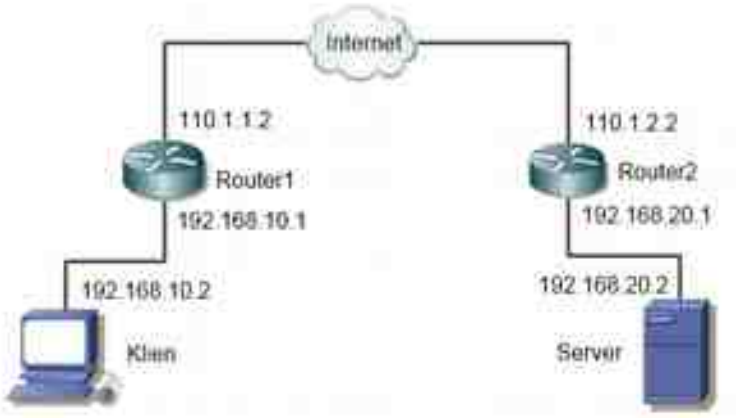
Untuk memudahkan konfigurasi jaringan dan pengukuran kecepatan transfer data digunakan perangkat lunak Virtual Box. Penggunaan Virtual Box dapat mengabaikan *layer* 1 (*link*) TCP/IP dalam perhitungan kecepatan transfer data. Dengan demikian, semua kerumitan perhitungan kecepatan transfer data yang dipengaruhi faktor pengkabelan dapat diabaikan.

Pengabaian faktor pada *layer link* menyebabkan hasil kecepatan transfer data yang terukur hanya dipengaruhi oleh *layer* 2 (*network*), *layer* 3 (*transport*) dan *layer* 4 (*application*). *Layer network, transport dan application* tersebut berkaitan dengan pengaruh perangkat lunak pada kecepatan transfer data. Perangkat lunak yang berpengaruh dalam hal ini adalah sistem operasi dan program aplikasi.

BAB 5 IMPLEMENTASI DAN PEMBAHASAN

5.1 Implementasi

Konfigurasi jaringan yang diperlukan dalam penelitian ini melibatkan satu Server, satu klien, dua Router dan dua sambungan ke jaringan Internet. Server berlaku sebagai server FTP yang melayani proses *download* dan *upload file*. Klien berlaku sebagai tempat untuk mengukur kecepatan *download* dan *upload*. Router berlaku sebagai penyedia layanan PPTP. Jaringan Internet berlaku sebagai media untuk pembentukan kanal PPTP. Diagram jaringan yang dipakai dapat dilihat pada Gambar 5.1 sebagai berikut.



Gambar 5.1 Rancangan Jaringan

5.1.1 Konfigurasi Klien

Komputer Klien digunakan untuk mengukur kecepatan *download* dan *upload file*. Komputer Klien menggunakan sistem operasi Linux distro Mandriva 2010. Pada komputer Klien dipasang perangkat lunak wget dan wput. Perangkat lunak wget digunakan untuk proses *download* sedangkan wput digunakan untuk proses *upload*. Perangkat lunak wget dan wput dapat digunakan untuk proses transfer *file* secara *commnad line* dan dapat menampilkan kecepatan transfer *file* yang digunakan. Dengan demikian, dua perangkat lunak ini dapat digabungkan dalam suatu sekrip program untuk memudahkan pengambilan data.

Komputer Klien terhubung dengan jaringan Internet melalui Router1. Komputer Klien diberi alamat IP 192.168.20.2/24, alamat Gateway 192.168.20.1 dan alamat DNS 8.8.8.8. Pemberian alamat IP pada komputer Klien diatur pada *file* /etc/sysconfig/network-scripts/ifcfg-eth0 yang isinya adalah sebagai berikut.

```
DEVICE=eth0
BOOTPROTO=static
IPADDR=192.168.20.2
NETMASK=255.255.255.0
GATEWAY=192.168.20.1
ONBOOT=yes
METRIC=10
MII_NOT_SUPPORTED=no
USERCTL=no
DNS1=8.8.8.8
RESOLV_MODS=no
IPV6INIT=no
IPV6TO4INIT=no
ACCOUNTING=no
```

Alamat gateway merupakan alamat IP antarmuka pada Router1 yang berhadapan dengan jaringan yang meliputi komputer Klien. Alamat DNS yang dipakai adalah

alamat DNS milik Google.com.

Hasil konfigurasi alamat IP adalah sebagai berikut.

```
# ifconfig eth0

eth0 Link encap:Ethernet  HWaddr 08:00:27:68:50:A7
     inet addr:192.168.20.2 Bcast:192.168.20.255 Mask:255.255.255.0
     inet6 addr: fe80::a00:27ff:fe68:50a7/64 Scope:Link
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:227 errors:0 dropped:0 overruns:0 frame:0
     TX packets:143 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:1000
     RX bytes:21708 (21.1 KiB)  TX bytes:19401 (18.9 KiB)
```

Hasil disunting untuk menghilangkan Shell Linux [root@localhost] dan merapikan tampilan tanpa mengurangi inti isinya.

Hasil konfigurasi alamat Gateway dapat dilihat pada tampilan tabel *routing* komputer Klien sebagai berikut.

```
# route -n

Kernel IP routing table
Destination   Gateway       Genmask         Flags Metric Ref  Use  Iface
192.168.20.0  0.0.0.0       255.255.255.0   U     10     0      0  eth0
169.254.0.0   0.0.0.0       255.255.0.0     U     10     0      0  eth0
0.0.0.0       192.168.20.1  0.0.0.0         UG    10     0      0  eth0
```

5.1.2 Konfigurasi *Server*

Komputer server disiapkan untuk melayani permintaan *download* dan *upload* dari komputer Klien. Asumsi yang dipakai adalah komputer Server terletak pada jaringan yang berbeda dan terhubung ke jaringan Internet melalui Router2. Dengan demikian komputer Server tidak dapat diakses secara langsung oleh komputer Klien.

Komputer server menggunakan sistem operasi Linux distro Mandriva 2010. Komputer Server diberi alamat IP 192.168.10.2/24, alamat Gateway

192.168.10.1 dan alamat DNS 8.8.8.8. Hasil konfigurasi alamat IP adalah sebagai berikut.

```
# ifconfig eth0

eth0   Link encap:Ethernet  HWaddr 08:00:27:5C:A1:27
       inet addr:192.168.20.2  Bcast:192.168.20.255  Mask:255.255.255.0
       inet6 addr: fe80::a00:27ff:fe5c:a127/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:174 errors:0 dropped:0 overruns:0 frame:0
       TX packets:107 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:17906 (17.4 KiB)  TX bytes:12905 (12.6 KiB)
```

Hasil konfigurasi alamat Gateway dapat dilihat pada tampilan tabel *routing* komputer Klien sebagai berikut.

```
# route -n

Kernel IP routing table
Destination   Gateway       Genmask         Flags Metric Ref   Use Iface
192.168.20.0  0.0.0.0       255.255.255.0   U     5      0     0   eth0
169.254.0.0   0.0.0.0       255.255.0.0     U     5      0     0   eth0
0.0.0.0       192.168.20.1  0.0.0.0         UG    5      0     0   eth0
```

Supaya kerja Server lebih ringan, mode teks dipakai sebagai mode kerja pada Server. Seluruh layanan bekerja pada mode teks dan bekerja secara latar. Salah satu layanan yang disiapkan pada Server adalah layanan *download* dan *upload file*. Komputer Server melayani permintaan *download* dan *upload* menggunakan protokol FTP. Layanan permintaan layanan FTP ditangani oleh perangkat lunak ProFTPD. Konfigurasi ProFTPD dituliskan dalam *file* /etc/proftpd.conf. Isi *file* tersebut dapat dilihat pada Lampiran.

5.1.3 Konfigurasi Router1

Router1 digunakan sebagai Router penghubung dari jaringan pertama ke

jaringan Internet. Router1 juga difungsikan sebagai klien PPTP untuk membentuk kanal aman antara Router1 dan Router2. Router1 menggunakan sistem operasi Mikrotik Router OS 2.52 yang merupakan sistem operasi khusus untuk mesin router.

Antarmuka jaringan yang menghadap jaringan lokal yaitu ether1 diberi alamat IP 192.168.10.1/24. Alamat IP ini merupakan alamat IP Gateway seluruh komputer yang ada pada jaringan lokal termasuk komputer Klien. Antarmuka jaringan yang menghadap jaringan Internet diberi alamat IP 110.1.1.2/24. Alamat Gateway Router1 adalah 110.1.1.1 yang merupakan alamat IP pada router milik Provider Internet. Konfigurasi jaringan dilakukan dengan beberapa perintah berikut.

```
> ip address add address=192.168.10.1/24 interface=ether1
> ip address add address=110.1.1.2/24 interface=ether2
> ip route add dst-address=0.0.0.0/0 gateway=110.1.1.1
```

Perintah tersebut sudah disunting dengan menghilangkan tanda Shell Mikrotik [admin@MikroTik] untuk menghemat penulisan dan merapikan tampilan tanpa mengurangi inti isinya.

Hasil konfigurasi alamat IP dan tabel *routing* dapat dilihat pada tampilan berikut.

```
> ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   192.168.10.1/24    192.168.10.0    ether1
 1   110.1.1.2/24       110.1.1.0       ether2

> ip route print
Flags: X - disabled, A - active, D - dynamic,
```

```
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 A S  0.0.0.0/0                          110.1.1.1                 1
 1 ADC  110.1.1.0/24       110.1.1.2       ether2                    0
 2 ADC  192.168.10.0/24    192.168.10.1    ether1                    0
```

Router1 juga berlaku sebagai mesin Firewall antara jaringan lokal dan jaringan Internet. Fungsi Firewall dilaksanakan oleh Router1 menggunakan salah satu fungsi NAT (Network Address Translation) yang disebut Masquerade. Pada Router1 diberi perintah sebagai berikut.

```
> ip firewall nat add chain=srcnat out-interface=ether2 action=masquerade
```

Maksud perintah tersebut adalah bahwa seluruh paket data yang melintasi Router1 yang keluar melalui antarmuka ether2 akan mengalami proses NAT dengan aksi Masquerade. Hasil konfigurasi Firewall pada Router1 dapat dilihat pada tampilan sebagai berikut.

```
> ip firewall nat print
Flags: X - disabled, I - invalid, D - dynamic
 0   chain=srcnat action=masquerade to-addresses=0.0.0.0 out-
interface=ether2
```

5.1.4 Konfigurasi Router2

Router2 digunakan sebagai Router penghubung dari jaringan lokal kedua ke jaringan Internet. Router2 juga difungsikan sebagai server PPTP. Hubungan server dan klien PPTP akan membentuk kanal aman antara Router1 dan Router2. Router2 menggunakan sistem operasi Mikrotik Router OS seperti pada Router1.

Antarmuka jaringan yang menghadap jaringan lokal yaitu ether1 diberi alamat IP 192.168.20.1/24. Alamat IP ini merupakan alamat IP Gateway seluruh

komputer yang ada pada jaringan lokal kedua termasuk komputer Server. Antarmuka jaringan yang menghadap jaringan Internet yaitu ether2 diberi alamat IP 110.1.2.2/24. Alamat Gateway Router2 adalah 110.1.2.1 yang merupakan alamat IP pada router milik Provider Internet. Konfigurasi jaringan dilakukan dengan beberapa perintah berikut.

```
> ip address add address=192.168.20.1/24 interface=ether1
> ip address add address=110.1.2.2/24 interface=ether2
> ip route add dst-address=0.0.0.0/0 gateway=110.1.2.1
```

Hasil konfigurasi alamat IP dan tabel *routing* dapat dilihat pada tampilan berikut.

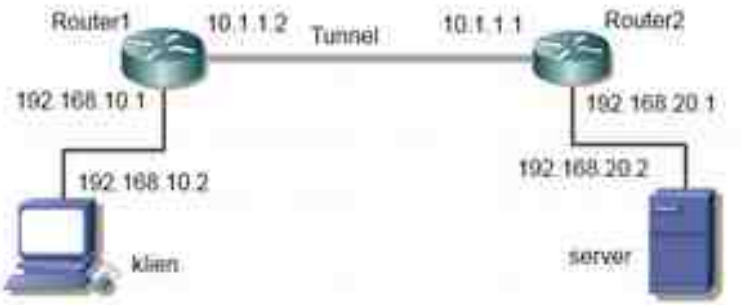
```
> ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   110.1.2.2/24       110.1.2.0       ether2
 1   192.168.20.1/24    192.168.20.0    ether1

> ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 A S  0.0.0.0/0                          110.1.2.1                 1
 1 ADC  110.1.2.0/24       110.1.2.2       ether2                    0
 2 ADC  192.168.20.0/24    192.168.20.1    ether1                    0
```

Router2 juga berlaku sebagai mesin Firewall antara jaringan lokal kedua dan jaringan Internet. Fungsi Firewall dilaksanakan oleh Router1 menggunakan salah satu fungsi NAT yang disebut Masquerade seperti pada Router1. Pada Router2 diberi perintah yang sama seperti pada Router1. Hasil konfigurasi Firewall pada Router2 juga sama dengan tampilan konfigurasi NAT pada Router1.

5.1.5 Konfigurasi Kanal PPTP

Sampai pada tahap konfigurasi NAT pada masing-masing router, komputer Klien dan komputer Server sudah dapat terhubung ke jaringan Internet. Namun demikian kedua komputer belum dapat terhubung secara langsung. Dua jaringan dimaksudkan supaya dapat terhubung secara langsung seperti tidak melalui jaringan Internet.

Salah satu cara untuk menghubungkan dua jaringan melalui jaringan Internet adalah membentuk kanal VPN (Virtual Private Network). Salah satu kanal VPN yang dapat dibentuk adalah PPTP. Secara diagram kanal PPTP dapat ditunjukkan pada Gambar 5.2 berikut.



Gambar 5.2 Rancangan Jaringan Menggunakan VPN PPTP

Router2 berlaku sebagai server PPTP sedangkan Router1 berlaku sebagai klien PPTP. Antara Router1 dan Router2 akan membentuk kanal (*tunnel*) aman. Kanal ini memungkinkan komputer Server dan Klien dapat terhubung secara

langsung.

Router2 dikonfigurasi sebagai server PPTP menggunakan perintah sebagai berikut.

```
> interface pptp-server server set enabled=yes
```

Hasil konfigurasi server PPTP dapat dilihat pada tampilan berikut.

```
> interface pptp-server server print

            enabled: yes
            max-mtu: 1460
            max-mru: 1460
               mrru: disabled
     authentication: mschap1,mschap2
  keepalive-timeout: 30
    default-profile: default-encryption
```

Supaya *server* PPTP lebih aman, ditetapkan *username* dan *password* untuk menghubungi *server*. Untuk menetapkan *username* dan *password* digunakan perintah berikut.

```
> ppp secret add name="pptp-server01" service=pptp password="yyy" local-
address=10.1.1.1 remote-address=10.1.1.2 disabled=no
```

Dengan menggunakan perintah di atas, berarti ditetapkan *username* adalah pptp-server01 dan *password* adalah yyy. *Password* dituliskan berbeda dengan yang sesungguhnya untuk keamanan sistem. Pada perintah tersebut juga ditetapkan alamat IP privat yang ditetapkan pada *server* PPTP 10.1.1.1, sedangkan alamat IP privat untuk klien PPTP adalah 10.1.1.2. *Username* ditentukan dengan pengarah *name*, *password* ditentukan dengan pengarah *password*, alamat IP privat *server* PPTP ditentukan dengan pengarah *local-address*, sedangkan alamat IP privat klien PPTP ditentukan dengan pengarah *remote-address*. Hasil konfigurasi

server PPTP dapat dilihat sebagai berikut.

```
> ppp secret print

Flags: X - disabled
#  NAME          SERVICE CALLER-ID   PASSWORD    PROFILE  REMOTE-ADDRESS
0  pptp-server01 pptp                password    default  10.1.1.2
```

Router1 dikonfigurasi sebagai klien PPTP. Untuk menjalankan klien PPTP diperlukan beberapa parameter yaitu alamat IP *server* PPTP, *username* dan *password*. Untuk menjalankan klien PPTP digunakan perintah sebagai berikut.

```
> interface pptp-client add name=pptp-client01 connect-to=110.1.2.2
user="pptp-server01" password="yyy" disabled=no
```

Maksud perintah tersebut adalah menjalankan klien PPTP untuk menghubungkan diri pada *server* PPTP yang beralamat di 110.1.2.2 menggunakan *username* pptp-server01 serta *password* yang dipakai adalah yyy. Apabila klien berhasil menghubungi *server* PPTP, maka hasilnya dapat dilihat menggunakan perintah sebagai berikut.

```
> interface pptp-client print

Flags: X - disabled, R - running
 0    name="pptp-client01" max-mtu=1460 max-mru=1460 mrru=disabled
connect-to=110.1.2.2 user="pptp-server01" password="password"
profile=default-encryption
      add-default-route=no dial-on-demand=no
allow=pap,chap,mschap1,mschap2
```

Setelah *server* dan klien PPTP berhasil terhubung, muncul antarmuka baru pada Router1 dan Router2 berkaitan dengan hubungan PPTP. Antarmuka ini punya alamat sesuai pengaturan yang sudah ditetapkan. Pada *server* PPTP (Router2), konfigurasi alamat IP dapat dilihat dengan perintah sebagai berikut.

```
> ip address print
```

```
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   110.1.2.2/24       110.1.2.0       ether2
 1   192.168.20.1/24    192.168.20.0    ether1
 2 D 10.1.1.1/32        10.1.1.2        <pptp-pptp-server01>
```

Pada Router2 terdapat tambahan antarmuka bernama pptp-pptp-server01. Antarmuka ini muncul karena adanya hubungan VPN PPTP. Pada tampilan tersebut dapat dilihat, antarmuka ini punya alamat IP 10.1.1.1 sesuai dengan yang ditetapkan.

Selain alamat IP, muncul juga tambahan tabel *routing* berkaitan dengan hubungan VPN PPTP. Isi tabel *routing* pada Router2 dapat dilihat dengan perintah sebagai berikut.

```
> ip route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r
- rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 A S  0.0.0.0/0                          110.1.2.1                 1
 1 ADC  10.1.1.2/32        10.1.1.1        <pptp-pptp-serv...        0
 2 ADC  110.1.2.0/24       110.1.2.2       ether2                    0
 3 ADC  192.168.20.0/24    192.168.20.1    ether1                    0
```

Pada hasil tampilan dapat dilihat terdapat tambahan satu rute menuju jaringan 10.1.1.2/32 dengan *flag* ADC. Rute menuju jaringan ini dilewatkan pada antarmuka pptp-pptp-server01. Rute ini berkaitan dengan jaringan yang dipakai untuk menghubungkan dua *router* menggunakan kanal PPTP.

Pada Router1 juga terdapat tambahan antarmuka berkaitan dengan hubungan VPN PPTP. Konfigurasi alamat IP pada Router1 dapat dilihat menggunakan perintah sebagai berikut.

```
> ip address print
```

```
Flags: X - disabled, I - invalid, D - dynamic
 #    ADDRESS            NETWORK         INTERFACE
 0    192.168.10.1/24    192.168.10.0    ether1
 1    110.1.1.2/24       110.1.1.0       ether2
 2 D  10.1.1.2/32        10.1.1.1        pptp-client01
```

Pada Router1 terdapat tambahan antarmuka bernama pptp-client01. Antarmuka ini berkaitan dengan hubungan VPN PPTP yang sedang terjadi. Alamat IP antarmuka ini adalah 10.1.1.2 sesuai dengan yang ditetapkan.

Pada Router1 juga terdapat tambahan isi tabel *routing*. Untuk melihat ini tabel *routing* pada Router1 digunakan perintah sebagai berikut.

```
> ip route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r
- rip, b - bgp, o - ospf, m - mme, B - blackhole, U - unreachable, P -
prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 A S  0.0.0.0/0                          110.1.1.1                 1
 1 ADC  10.1.1.1/32        10.1.1.2        pptp-client01             0
 2 ADC  110.1.1.0/24       110.1.1.2       ether2                    0
 3 ADC  192.168.10.0/24    192.168.10.1    ether1                    0
```

Pada tampilan tersebut dapat dilihat isi tabel *routing* secara keseluruhan. Tambahan isi tabel *routing* berkaitan dengan tambahan rute menuju jaringan 10.1.1.1/32. Rute menuju jaringan ini dilewatkan pada antarmuka pptp-client01. Jaringan ini merupakan jaringan *Point-to-Point*.

Sampai pada langkah tersebut, hubungan antara Router1 dan Router2 sudah dapat berjalan menggunakan kanal PPTP. Namun demikian, hubungan ini baru terjadi antar *router*. Hubungan antara komputer Klien dan Server belum dapat terjadi. Hal demikian terjadi karena jaringan komputer Klien belum dikenal oleh komputer Server dan demikian juga sebaliknya.

Alamat jaringan pada jaringan komputer Klien dan Server perlu saling

dikenalkan satu sama lain. Pada Router1 perlu ditambahkan satu data rute menggunakan perintah berikut.

```
> ip route add dst-address=192.168.20.0/24 gateway=10.1.1.1
```

Isi tabel *routing* pada Router1 menjadi sebagai berikut.

```
> ip route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r
- rip, b - bgp, o - ospf, m - mme, B - blackhole, U - unreachable, P -
prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 A S  0.0.0.0/0                          110.1.1.1                 1
 1 ADC  10.1.1.1/32        10.1.1.2        pptp-client01             0
 2 ADC  110.1.1.0/24       110.1.1.2       ether2                    0
 3 ADC  192.168.10.0/24    192.168.10.1    ether1                    0
 4 A S  192.168.20.0/24                    10.1.1.1                  1
```

Pada tampilan tersebut dapat dilihat adanya tambahan rute pada tabel *routing*. Rute tersebut diberi kode AS (*Active Static*) yang maksudnya rute tersebut dalam keadaan aktif dan ditambahkan secara statis.

Pada Router2 sebagai server PPTP juga perlu ditambahkan rute menggunakan perintah berikut.

```
> ip route add dst-address=192.168.10.0/24 gateway=10.1.1.2
```

Isi tabel *routing* pada Router2 menjadi sebagai berikut.

```
> ip route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r
- rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 A S  0.0.0.0/0                          110.1.2.1                 1
 1 ADC  10.1.1.2/32        10.1.1.1        <pptp-pptp-serv...        0
 2 ADC  110.1.2.0/24       110.1.2.2       ether2                    0
 3 A S  192.168.10.0/24                    10.1.1.2                  1
 4 ADC  192.168.20.0/24    192.168.20.1    ether1                    0
```

Pada tampilan tersebut dapat dilihat adanya tambahan data rute pada tabel

routing. Rute tersebut diberi kode AS (*Active Static*) yang mana dst-address 192.168.10.0/24 dilewatkan pada gateway 10.1.1.2 yaitu antarmuka PPTP pada Router1.

Sampai tahap ini, komputer Klien dan Server sudah berhasil dihubungkan secara langsung. Langkah berikutnya adalah uji coba kecepatan transfer *file*.

5.2 Pembahasan

Uji coba transfer data dilakukan pada dua jenis *file* yaitu *file* teks dan *file* gz. *File* teks mewakili transfer data untuk *file* yang belum mengalami kompresi, sedangkan *file* gz mewakili transfer data yang sudah mengalami kompresi. Ukuran *file* dipilih yang cukup besar. Sedemikian sehingga kecepatan transfer data dapat diamati secara lebih baik. Transfer data dilakukan secara *upload* maupun *download* menggunakan protokol khusus transfer data yaitu FTP.

Daftar *file* dan ukuran yang dipakai untuk uji coba transfer *file* dapat dilihat pada Tabel 5.1. Tabel tersebut berisi nama *file* dan ukuran *file* yang digunakan untuk percobaan.

Tabel 5.1 Daftar *File*

No	*file*	*Size*	. .	No	*file*	*Size*
1	file0.txt	79206023		6	file1.gz	17097953
3	file1.txt	158412046		7	file2.gz	34186353
3	file2.txt	316824092		8	file3.gz	68377785
4	file3.txt	633648184		9	file4.gz	136767033
5	file4.txt	1267296368		10	file5.gz	273535718

. Selanjutnya dilakukan uji coba hubungan antara Klien dan Server baik

menggunakan uji ping maupun traceroute. Hasil uji ping dari Klien ke Server adalah sebagai berikut.

```
# ping 192.168.10.2 -c 4

PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=62 time=8.54 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=62 time=5.83 ms
64 bytes from 192.168.10.2: icmp_seq=3 ttl=62 time=5.79 ms
64 bytes from 192.168.10.2: icmp_seq=4 ttl=62 time=6.11 ms

--- 192.168.10.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3011ms
rtt min/avg/max/mdev = 5.794/6.571/8.545/1.149 ms
```

Hasil uji ping menunjukkan bahwa komputer Klien dan Server sudah terhubung secara betul dan berfungsi secara baik.

Hasil uji traceroute dari komputer Klien menuju komputer Server dapat dilihat sebagai berikut.

```
# traceroute 192.168.10.2 -n

traceroute to 192.168.10.2 (192.168.10.2), 30 hops max, 60 byte packets
 1  192.168.20.1  1.898 ms  1.826 ms  1.771 ms
 2  10.1.1.2  6.572 ms  9.511 ms  6.090 ms
 3  192.168.10.2  9.314 ms  10.244 ms  10.206 ms
```

Hasil uji traceroute menunjukkan bahwa rute yang dilalui paket data sudah betuk yaitu melalui kanal PPTP. Kanal PPTP ditandai dengan alamat IP 10.1.1.2 yaitu alamat IP pada Router1.

Pada penelitian akan diuji pengaruh enkripsi dan kompresi pada proses transfer data melalui kanal PPTP. Untuk uji coba kecepatan transfer data antara komputer Server dan Klien perlu disiapkan parameter yang berkaitan dengan pengaturan enkripsi dan kompresi. Parameter ini berkaitan dengan proses transfer data melalui kanal PPTP apakah melalui proses enkripsi atau kompresi.

Mikrotik menyediakan satu opsi enkirpsi yaitu use-encryption dan dua opsi kompresi. Opsi kompresi yang disediakan adalah use-compression dan use-vj-compress. Pengaturan opsi-opsi tersebut dituliskan dalam bentuk profil. Pada Router1 dan Router2 diatur empat macam profil yaitu noAll, encrypt, compress, dan vjcompress. Masing-asing mencerminkan transfer data tanpa enkripsi maupun kompresi, transfer data terenkrepsi, transfer data terkompresi dan transfer data terkompresi menggunakan vjcompress.

Untuk membuat profil-profil tersebut, pada Router1 dan Router2 diberi perintah sebagai berikut.

```
> ppp profile add name=noAll change-tcp-mss=default use-mpls=default use-
encryption=no use-vj-compression=no use-compression=no

> ppp profile add name=encrypt change-tcp-mss=default use-mpls=default
use-encryption=yes use-vj-compression=no use-compression=no

> ppp profile add name=compress change-tcp-mss=default use-mpls=default
use-encryption=no use-vj-compression=no use-compression=yes

> ppp profile add name=vj_compress change-tcp-mss=default use-
mpls=default use-encryption=no use-vj-compression=yes use-compression=no
```

Profil-profil tersebut diterapkan pada masing-masing router secara bergantian tetapi sama. Karena kombinasinya cukup banyak, perlu dibuat sekrip program untuk mengambil data kecepatan transfer. Sekrip program dituliskan dalam sekrip Shell Bash sistem operasi Linux. Sekrip program yang digunakan untuk percobaan untuk *upload file* adalah sebagai berikut.

```
pptp_server="110.1.2.2"
pptp_client="110.1.1.2"
server="192.168.20.2"

aprofile="noAll encrypt compress vjcompress encrypt_compress
encrypt_vjcompress"
afile="file0.txt file1.txt file2.txt file3.txt file4.txt file1.gz
file2.gz file3.gz file4.gz"
```

```
#variasi profile server
for profile in `echo $aprofile`
do
  dev="data_upload-$profile.csv"
  #dev="/dev/stdout"

  echo `date` > $dev

  ssh $pptp_server -l admin "interface pptp-server server set default-
profile=$profile; system reboot"
  ssh $pptp_client -l admin "interface pptp-client set pptp-client01
profile=$profile; system reboot"
  sleep 40

  #variasi tipe file
  for file in `echo $afile`
  do
    echo "$profile;$file;"
    echo -n "$profile;$file;" >>$dev

    #jumlah experimen
    for exp in  {1..10}
    do
      #hapus file
      lftp -e "mrm -f *.txt; mrm -f *.gz; bye" -u xxx,yyy $server \
      >/dev/null
      #upload file & membaca rate
      rate=`wput $file ftp://xxx:yyy@$server 2>&1 | \
      grep Transfered | awk '$0=$2' FS="at " RS="M/s"`

      echo -n "$rate;" >>$dev
    done;
    echo >>$dev

  done
done
```

Sekrip program tersebut sedikit dimodifikasi dari program sesungguhnya. Hal demikian untuk menjaga keamanan terutama untuk menyembunyikan username dan password. Username yang dituliskan dalam program adalah xxx, sedangkan password yang dituliskan adalah yyy.

Sekrip program yang digunakan untuk percobaan *download file* adalah sebagai berikut.

```
pptp_server="110.1.2.2"
pptp_client="110.1.1.2"
server="192.168.20.2"

afile="file0.txt file1.txt file2.txt file3.txt file4.txt file1.gz
file2.gz file3.gz file4.gz file5.gz"
aprofile="noAll encrypt compress vjcompress encrypt_compress
encrypt_vjcompress"
```

```
#variasi profile server
for profile in `echo $aprofile`
do
  dev="data_download-$profile.csv"
  #dev="/dev/stdout"

  echo `date` >$dev

  ssh $pptp_server -l admin \
  "interface pptp-server server set default-profile=$profile;\
     system reboot"
  ssh $pptp_client -l admin \
  "interface pptp-client set pptp-client01 profile=$profile; \
     system reboot"
  sleep 40

  #variasi tipe file
  for file in `echo $afile`
  do
    echo "$profile;$file;"
    echo -n "$profile;$file;" >>$dev

    #jumlah experimen
    for exp in  {1..10}
    do
      #hapus file
      rm -f *.txt >/dev/null
      rm -f *.gz >/dev/null

      #download file & membaca rate
      rate=`wget ftp://xxx:yyy@$server/$file 2>&1 | \
      grep saved | awk '$0=$2' FS="(" RS=")"`

      echo -n "$rate;" >>$dev
    done;

    echo >>$dev

  done
done
```

Sekrip program tersebut juga sedikit dimodifikasi dari program sesungguhnya untuk menyembunyikan username dan password.

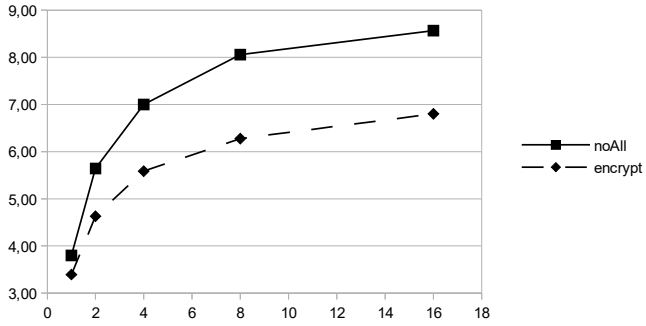
5.2.1 Percobaan *Upload File*

Pada percobaan *upload file* dicoba dilakukan *upload* terhadap *file* teks (belum mengalami kompresi) dan *file* gz (sudah mengalami kompresi). Masing-asing sebanyak lima *file* dengan ukuran yang berbeda. Tiap-tiap *file* dilakukan

pengambilan data sebanyak sepuluh kali. Hasil mentah pengambilan data dapat dilihat pada Lampiran.

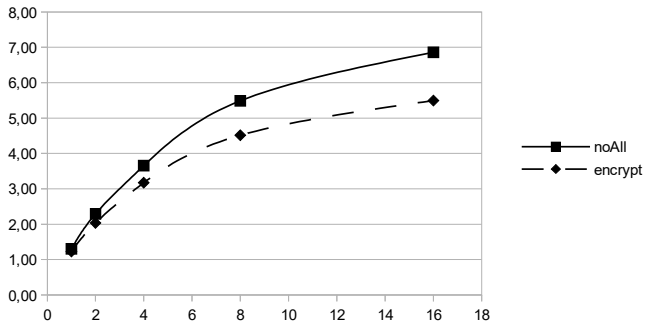
Pengaruh Enkripsi

Pengaruh enkripsi pada *upload* data melalui kanal PPTP dapat dilihat pada Gambar 5.3 dan Gambar 5.4. Gambar 5.3 menunjukkan pengaruh enkripsi pada proses *upload file* teks dari klien menuju server. Proses *upload file* dilakukan menggunakan protokol FTP.



Gambar 5.3 Pengaruh Enkripsi *Upload File* Teks

Gambar 5.4 menunjukkan pengaruh enkripsi pada *upload file* terkompresi gz dari klien menuju server menggunakan protokol FTP.



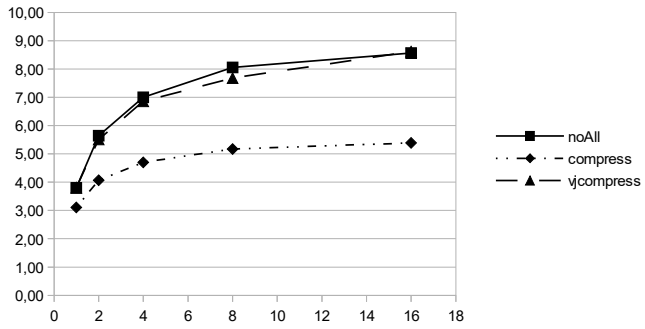
Gambar 5.4 Pengaruh Enkripsi *Upload File* Gz

Pengaruh enkripsi pada *upload file* disebabkan karena adanya proses enkripsi sebelum paket data dikirimkan dari klien PPTP dan proses dekripsi pada saat saat paket data diterima oleh server PPTP. Proses enkripsi dan dekripsi memerlukan komputasi tertentu yang tentunya memerlukan waktu tertentu. Karena adanya dua proses tersebut, waktu pengiriman menjadi semakin besar. Secara keseluruhan kecepatan *upload* data menjadi lebih rendah.

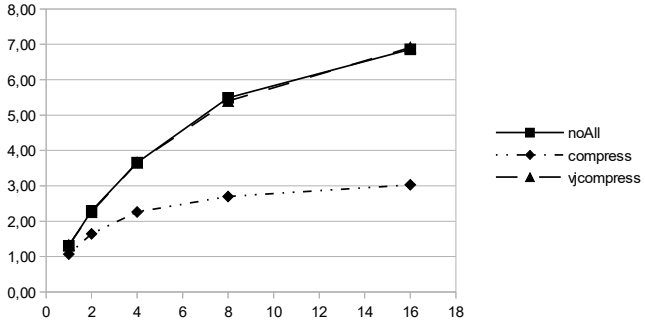
Proses enkripsi pada *upload file* dapat menurunkan kecepatan, namun proses enkripsi tentunya akan meningkatkan keamanan data. Kalau yang dipertimbangkan hanya kecepatan *upload*, proses *upload* data tidak perlu proses enkripsi. Namun kalau isi keamanan sangat dipertimbangkan, diperlukan proses enkripsi. Keamanan bisa diabaikan apabila data dikirimkan melalui jaringan yang aman, misalnya jaringan lokal. Jika *upload file* dilewatkan jaringan yang tidak aman (misalnya jaringan Internet), maka proses enkripsi perlu dilakukan.

Pengaruh Kompresi

Pengaruh kompresi pada *upload file* ditunjukkan pada Gambar 5.5 dan Gambar 5.6. Gambar 5.5 menunjukkan pengaruh proses kompresi pada *upload file* teks dari komputer klien menuju komputer server, sedangkan Gambar 5.6 menunjukkan pengaruh proses kompresi pada *upload file* gz.



Gambar 5.5 Pengaruh Kompresi *Upload File* Teks



Gambar 5.6 Pengaruh Kompresi *Upload File* Gz

Pengaruh kompresi pada *upload file* disebabkan adanya proses kompresi paket data pada saat pengiriman data dari klien dan proses dekompresi paket data pada saat data diterima oleh server. Proses kompresi pada klien PPTP dan

dekompresi server PPTP memerlukan waktu komputasi tertentu yang secara keseluruhan menurunkan kecepatan *upload file*.

Pada penelitian digunakan dua metode kompresi melalui pengaturan parameter use-compression dan use-vj-compression. Penelitian tidak membahas lebih detail berkaitan dengan dua jenis metode kompresi paket data tersebut. Penelitian hanya menggunakan perbedaan menggunakan pengaturan parameter PPTP. Perbedaan mendasar dua metode tersebut adalah bahwa use-vj-compression hanya melakukan kompresi pada header paket data.

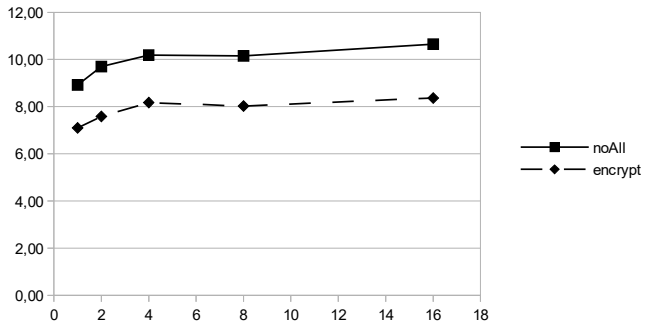
Pengaturan use-vj-compression tidak begitu berpengaruh pada proses *upload* data secara keseluruhan, sedangkan use-compression cukup mempengaruhi kecepatan *upload file*. Kedua parameter punya pengaruh yang mirip baik pada *upload file* teks maupun *upload file* gz. Meski pun *upload* dilakukan pada *file* gz, pengaruh kompresi tetap terjadi. Hal demikian terjadi karena proses kompresi dilakukan pada paket data.

5.2.2 Percobaan *Download File*

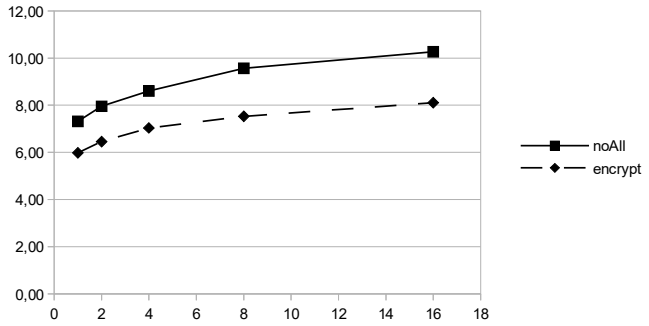
Pada percobaan *download file* dicoba dilakukan *download* terhadap *file* teks (belum mengalami kompresi) dan *file* gz (sudah mengalami kompresi). Masing-asing sebanyak lima *file* dengan ukuran yang berbeda. Tiap-tiap *file* dilakukan pengambilan data sebanyak sepuluh kali. Hasil mentah pengambilan data dapat dilihat pada Lampiran.

Pengaruh Enkripsi

Pengaruh enkripsi pada *download file* melalui kanal PPTP dapat dilihat pada Gambar 5.7 dan Gambar 5.8.

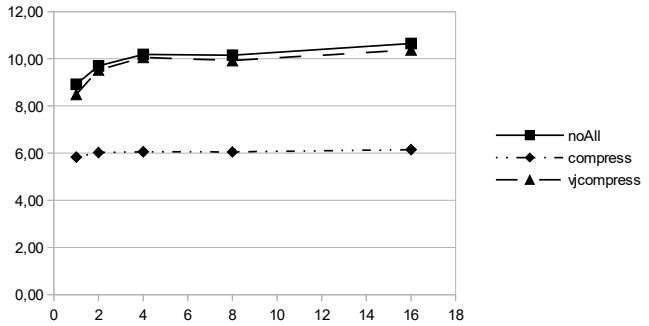


Gambar 5.7 Pengaruh Enkripsi *Download File* Teks

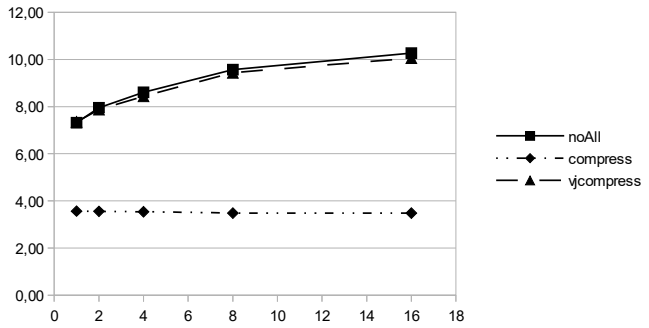


Gambar 5.8 Pengaruh Enkripsi *Download File* Gz

Gambar 5.7 menunjukkan pengaruh enkripsi pada proses *download file* teks dari komputer server menuju komputer klien. Protokol yang dipakai untuk proses *download* adalah FTP. Gambar 5.8 menunjukkan pengaruh enkripsi pada proses *download file* gz dari komputer server ke komputer klien.



Gambar 5.9 Pengaruh Kompresi *Download File* Teks



Gambar 5.10 Pengaruh Kompresi *Download File* Gz

Pengaruh kompresi pada *download file* disebabkan karena adanya proses kompresi sebelum paket data dikirimkan dari server PPTP dan proses dekompresi pada saat saat paket data diterima oleh klien PPTP. Proses kompresi dan dekompresi memerlukan waktu komputasi tertentu. Karena adanya dua proses tersebut, waktu *download file* menjadi semakin besar. Secara keseluruhan kecepatan *download file* menjadi lebih rendah.

Proses kompresi pada *download file* dapat menurunkan kecepatan.

Penurunan kecepatan *download file* tidak signifikan pada penggunaan parameter use-vj-compression. Penggunaan parameter use-compression cukup berpengaruh pada kecepatan download *file*. Pada penggunaan kanal PPTP tidak disarankan menggunakan parameter user-compression. Jika diinginkan mengunakan kompresi, disarankan menggunakan use-vj-compression.

Pada percobaan *upload* dan *download*, secara umum kecepatan *download* lebih tinggi. Dengan demikian apabila transfer *file* dapat dilakukan menggunakan *upload* atau *download*, disarankan menggunakan proses *download*. Kasus ini bisa terjadi apabila, pengguna punya hak akses baik pada server maupun pada klien. Selain itu biasanya kasus ini hanya bisa dilakukan menggunakan protokol FTP atau SFTP (*Secure File Transfer Protocol*).

BAB 6 KESIMPULAN

6.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil pembahasan dan percobaan dalam penelitian ini adalah sebagai berikut.

- Pada penelitian berhasil disusun metode untuk mengamati kecepatan transfer *file* melalui kanal VPN PPTP menggunakan kombinasi perangkat lunak Wget, Wput, LFTP dan utilitas dasar sistem operasi Linux.
- Proses enkripsi pada transfer *file* melalui kanal VPN PPTP menyebabkan penurunan kecepatan transfer *file*.
- Proses kompresi menggunakan parameter use-compression pada transfer *file* melalui kanal VPN PPTP menyebabkan penurunan kecepatan transfer *file*.
- Proses kompresi menggunakan parameter use-vj-compression pada transfer *file* melalui kanal VPN PPTP tidak signifikan menyebabkan penurunan kecepatan transfer *file*.
- Apabila proses transfer *file* dapat dilakukan dua arah, lebih baik menggunakan *download file*.

6.1 Saran

Saran yang diajukan untuk pengembangan dan penelitian lebih lanjut dari penelitian ini adalah sebagai berikut.

- Perlu diketahui pengaruh enkripsi dan kompresi pada transfer data melalui protokol selain FTP. Namun mungkin agak sulit pengamatan untuk proses *upload file*.
- Perlu diteliti lebih lanjut, perbandingan kecepatan transfer *file* melalui beberapa protokol.
- Perlu diteliti pengaruh penggunaan beberapa mesin *router* yang menggunakan pemroses yang lebih baik, sedemikian sehingga proses enkripsi dan kompresi berjalan lebih cepat.

Daftar Pustaka

Agung Sediyono dan Alitalia Rahma, 2008, *Pengaruh Kompresi Header Mikrotik Pada Transfer Rate Di Jaringan Vpn PPTP*, Konferensi Nasional Sistem dan Informatika

Ananian, S., 2013, *PPTP-linux: a PPTP client for Linux*, http://cscott.net/Projects/PPTP/

Anonim, 2012, *The MS-CHAP version 1 authentication protocol has been deprecated in Windows Vista*, http://support.microsoft.com

Dedy Cahyadi, 2010, *Pemanfaatan Fitur Tunneling Menggunakan Virtual Interface EoIP di Mikrotik RouterOS untuk koneksi Bridging Antar Kantor Melalui Jaringan ADSL* Telkom Speedy

Fritsch, H. , 2014, *wput − A wget−like ftp−uploader,* http://wput.sourceforge.net/wput.1.html

Kukuh Prasetyo, 2010, *Analisis Performasi Pada Penggunaan IPsec dan PPTP Untuk Internet Protocol Television (IPTV)*

Lukyanov, A.V., 2014, *lftp - Sophisticated file transfer program*, http://www.manpagez.com/man/1/lftp/

Mikrotik, 2008*, MikroTik RouterOS™ v3.0, Reference Manual,* http://www.mikrotik.com/testdocs/ros/3.0

Muhammad Muslich, Fatah Yasin, 2009, *Virtual Private Network Berbasis IP Security Dengan Linux Free Secure Wide Area Network*, Simposium Nasional RAPI VIII 2009

Nanda Pramudya, 2009, *Implementasi dan Analisis Point-to-Point Tunneling Protocol Serta Ethernet Over Internet Protocol Sebagai Metode Untuk Membuat Virtual Private Network*

Niksic, H., 2014, *Wget - The non-interactive network downloader*, http://www.manpagez.com/man/1/wget/

Nova Rusdy Setyawan, 2011, *Implementasi VLAN Trunk Protocol(VTP) melalui Ethernet over Internet Protocol (EoIP)Tunnel pada Mikrotik RouterOS*, 17 April 2012 pkl 13.15 AM

Ramsay, M., 2000, *PoPToP, a Secure and Free VPN Solution*, http://http://www.linuxjournal.com/node/3965/print

Ray, Marsh, 2012, *MS PPTP MPPE only as secure as *single* DES*. Retrieved April 3, 2012.

Schmidt, J., 2012, *A death blow for PPTP*, http://www.h-online.com/security/features/A-death-blow-for-PPTP-1716768.html

The Internet Society, 1999, *Point-to-Point Tunneling Protocol (PPTP)*, http://tools.ietf.org/html/rfc2637
Wagito, 2012, *Implementasi VPN PPTP Untuk Integrasi Jaringan*