

SKRIPSI
IMPLEMENTASI LOAD BALANCING DUA ISP
MENGGUNAKAN MIKROTIK

(Studi Kasus : Laboratorium Komputer SMK PGRI Bekasi)



Oleh :
ANDRI DWI UTOMO
106091002904

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI
SYARIF HIDAYATULLAH
JAKARTA
2011

PENGESAHAN UJIAN

Skripsi yang berjudul "IMPLEMENTASI LOAD BALANCING DUA ISP MENGGUNAKAN MIKROTIK (Studi Kasus : Laboratorium Komputer SMK PGRI Bekasi)" yang ditulis oleh Andri Dwi Utomo, NIM 106091002904 telah diuji dan dinyatakan lulus dalam sidang Munaqosyah Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta pada tanggal 26 Mei 2011. Skripsi ini telah diterima sebagai salah satu syarat memperoleh gelar Sarjana Strata Satu (S1) Program Studi Teknik Informatika.

Menyetujui,

Penguji I

Penguji II

Herlino Nanang, MT

NIP. 19731209 200501 1 002

Pembimbing I

Arini, MT

NIP. 19760131 200901 2 001

Pembimbing II

Victor Amrizal, M.Kom

NIP. 150411288

Andrew Fiade, M.Kom

NIP. 19820811 200912 1 004

Mengetahui,

Dekan Fakultas Sains dan Teknologi

Ketua Program Studi Teknik Informatika

DR. Syopiansyah Jaya Putra, M.Sis

NIP. 19680117 200112 1 001

Yusuf Durachman, MSc, M.IT

NIP. 19710522 200604 1 002

**IMPLEMENTASI LOAD BALANCING DUA ISP
MENGUNAKAN MIKROTIK
(Studi Kasus : Laboratorium Komputer SMK PGRI Bekasi)**

Skripsi

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar
Sarjana Komputer

Fakultas Sains dan Teknologi

Universitas Islam Negeri Syarif Hidayatullah Jakarta

Oleh

Andri Dwi Utomo

106091002904

Menyetujui,

Pembimbing I

Pembimbing II

Victor Amrizal, M.Kom
NIP. 150411288

Andrew Fiade, M.Kom
NIP. 19820811 200912 1 004

Mengetahui,
Ketua Program Studi Teknik Informatika

Yusuf Durachman, MSc, M.IT
NIP. 19710522 200604 1 002

PERNYATAAN

DENGAN INI SAYA MENYATAKAN BAHWA SKRIPSI INI BENAR-BENAR HASIL KARYA SENDIRI YANG BELUM PERNAH DIAJUKAN SEBAGAI SKRIPSI ATAU KARYA ILMIAH PADA PERGURUAN TINGGI ATAU LEMBAGA MANAPUN.

Jakarta, Mei 2011

Andri Dwi Utomo
NIM. 106091002904

ABSTRAK

Andri Dwi Utomo, *Implementasi Load Balancing Dua ISP Menggunakan Mikrotik* (Studi Kasus : Laboratorium Komputer SMK PGRI Bekasi). (Dibawah bimbingan **Victor Amrizal** dan **Andrew Fiade**).

Kebutuhan akan akses internet saat ini sangat tinggi, baik untuk mencari informasi, artikel maupun pengetahuan terbaru. Banyak sekolah yang telah mengintegrasikan jaringan internet kedalam proses belajar-mengajar. Itu diharapkan agar siswa dapat dengan mudah mencari materi dan memahami pelajaran. Salah satunya ialah SMK PGRI Bekasi yaitu sebuah instansi pendidikan yang telah menjadikan Teknik Komputer Jaringan sebagai salah satu kejuruan yang ada di sekolah tersebut. Dan hampir setiap proses belajar-mengajar disana juga memerlukan koneksi internet untuk memudahkan siswa mencari materi pembelajaran. Maka daripada itu, SMK PGRI Bekasi menginginkan suatu koneksi internet yang stabil dan handal. Oleh karena itu timbul solusi untuk menggunakan dua ISP dan menjadikan mikrotik sebagai *load balancer*. Mekanismenya yaitu mikrotik akan menandai paket yang ingin mengakses internet, lalu memilih jalur ISP mana yang akan dilewatinya dan menyetarakan beban pada kedua ISP tersebut. Berdasarkan metode pengembangan sistem yang digunakan, yaitu *Network Development Life Cycle* (NDLC), maka sebelum menentukan metode *load balancing* yang akan digunakan, penulis melakukan analisis terhadap kondisi *traffic* jaringan yaitu dengan *memonitoring* untuk mendapatkan *log-log* yang berada di jaringan. Pemilihan *Nth load balancing* dikarenakan metode tersebut memenuhi kriteria karena dapat meningkatkan kecepatan koneksi dan membagi beban pada kedua *gateway* agar tidak terjadi *overload*. Lalu penulis menerapkan pula teknik *fail over*, yaitu dimana jika salah satu koneksi *gateway* sedang terputus, maka *gateway* lainnya otomatis akan menjadi *backup* yang akan menopang semua *traffic* jaringan.

Kata kunci : Koneksi internet ganda, *Nth load balancing*, Mikrotik.

Jumlah halaman : xviii + 92 halaman + 2 lampiran.

Jumlah Pustaka : 16 Sumber (2005 – 2010).

KATA PENGANTAR

Bismillahirrahmanirrahiim.

Segala puji dan syukur penulis panjatkan kepada Allah SWT yang Maha Kuasa, yang telah memberikan berkah dan anugrah-Nya kepada penulis sehingga mampu menyelesaikan skripsi ini dengan sebaik-baiknya. Shalawat serta salam tak lupa juga penulis haturkan kepada junjungan kita Nabi Besar Muhammad SAW.

Skripsi ini penulis buat sebagai syarat kelulusan dalam menempuh pendidikan jenjang Strata-1 (S1) di Universitas Islam Negeri Syarif Hidayatullah Jakarta. Selain itu juga penulis berharap penelitian ini dapat dipergunakan dengan baik oleh semua pihak yang membutuhkan, sehingga perkembangan ilmu pengetahuan dan teknologi khususnya di Program Studi Teknik Informatika, Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta dapat lebih maju.

Pada kesempatan ini penulis juga mengucapkan terima kasih kepada pihak-pihak yang telah membantu penulis menyelesaikan skripsi ini:

1. Bapak Dr. Syopiansyah Jaya Putra, M.Sis selaku Dekan Fakultas Sains dan Teknologi.
2. Bapak Yusuf Durrachman, M.Sc, MIT selaku Ketua Program Studi Teknik Informatika.
3. Ibu Viva Arifin, MMSI selaku Sekretaris Program Studi Teknik Informatika.

4. Bapak Victor Amrizal M.Kom selaku Pembimbing I serta Bapak Andrew Fiade M.Kom selaku pembimbing II yang selalu sabar dan rela meluangkan waktunya untuk mendukung dan membimbing penulis dalam menyelesaikan skripsi ini.
5. Kedua orang tua penulis, Ayahanda Sutarto dan Ibunda Karni Harnani, yang selalu memberikan kasih sayang, dukungan moril dan semangat untuk menjadikan penulis sebagai orang yang lebih baik. Serta keluarga besar penulis yang tidak bisa penulis sebutkan satu per satu.
6. Dosen-dosen Fakultas Sains dan Teknologi yang telah mengajarkan kepada penulis berbagai ilmu yang dapat penulis terapkan dalam penulisan skripsi ini dan dalam kehidupan penulis.
7. Sahabat terbaik, Taufik Nurhartantrio S.Kom yang selama ini telah membantu penulis dalam setiap akademik perkuliahan.
8. Sahabat-sahabat terbaik dari Alumni SMA Negeri 58 Jakarta yang selalu *sharing* berbagai hal sehingga menjadikan penulis pribadi yang baik.
9. Seluruh sahabat-sahabat Teknik Informatika Angkatan 2006 yang sama-sama berjuang dalam masa perkuliahan ini. Terimakasih *sharing* dan informasi yang telah diberikan.
10. Sahabat-sahabat Anggota kelompok KKN 78 yang selalu menjaga silaturahmi dan berbagi pengalaman yang telah membuat penulis bersemangat menyusul untuk menyelesaikan studi ini.

11. Seluruh pihak yang telah membantu dan namanya tidak dapat disebutkan satu per satu. Terima kasih yang sebesar-besarnya atas dukungan dan motivasinya. Semoga Allah SWT membalas segala kebaikan yang telah diberikan kepada penulis.

Tidak ada manusia yang sempurna, penulis mengetahui dan menyadari kemampuan penulis masih jauh dari kesempurnaan, oleh karena itu penulis mengharapkan saran dan kritik yang dapat membangun skripsi ini lebih baik lagi.

Jakarta, Mei 2011

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN PEMBIMBING	ii
HALAMAN PENGESAHAN UJIAN	iii
HALAMAN PERNYATAAN	iv
ABSTRAK	v
KATA PENGANTAR	vi
DAFTAR ISI	ix
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
DAFTAR LAMPIRAN	xviii

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan dan Manfaat Penelitian	4
1.4.1 Tujuan	4
1.4.2 Manfaat	4
1.5 Metodologi Penelitian	5
1.5.1 Metode Pengumpulan Data	6
1.5.2 Metode Pengembangan Sistem	6

1.6	Sistematika Penulisan	8
-----	-----------------------------	---

BAB II LANDASAN TEORI

2.1	Implementasi	10
2.2	<i>Load Balancing</i>	10
2.2.1	<i>Static Route</i> dengan <i>Address List</i>	12
2.2.2	<i>Equal Cost Multi Path</i> (ECMP)	10
2.2.3	Nth	13
2.2.4	<i>Per Connection Classifier</i> (PCC)	15
2.3	<i>Internet Service Provider</i> (ISP)	16
2.3.1	Indosat M2	17
2.3.2	Smart	18
2.4	Jaringan Komputer	19
2.5	Bentuk Jaringan	21
2.5.1	<i>Local Area Network</i> (LAN)	21
2.5.2	<i>Metropolitan Area Network</i> (MAN)	22
2.5.3	<i>Wide Area Network</i> (WAN)	23
2.6	Topologi Jaringan	24
2.6.1	Topologi <i>Star</i>	25
2.7	Perangkat Jaringan	26
2.7.1	<i>Personal Computer</i> (PC)	26
2.7.2	<i>Network Internet Card</i>	27
2.7.3	<i>Switch</i>	27
2.7.4	<i>Router</i>	27

2.7.5	<i>Modem Router</i>	28
2.7.6	<i>3G Wireless N Router</i>	29
2.8	<i>Model OSI Layer</i>	30
2.8.1	<i>Layer 1 – Physical Layer</i>	31
2.8.2	<i>Layer 2 – Data Link Layer</i>	31
2.8.3	<i>Layer 3 – Network Layer</i>	32
2.8.4	<i>Layer 4 – Transport Layer</i>	32
2.8.5	<i>Layer 5 – Session Layer</i>	32
2.8.6	<i>Layer 6 – Persentation Layer</i> .	32
2.8.7	<i>Layer 7 – Application Layer</i>	33
2.9	<i>Transmission Control Protocol/Internet Protocol (TCP/IP)</i> ...	34
2.9.1	<i>Hypertext Transfer Protocol (HTTP)</i>	35
2.9.2	<i>Hypertext Transfer Protocol Secure (HTTPS)</i>	35
2.9.3	<i>Secure Socket Layer (SSL)</i>	36
2.9.4	<i>Domain Name Service (DNS)</i>	36
2.9.5	<i>Simple Network Management Protocol (SNMP)</i>	37
2.10	<i>IP Address</i>	37
2.10.1	<i>Format Alamat IPv4</i>	38
2.10.2	<i>Kelas Alamat IP</i>	38
2.11	<i>Subnetting</i>	40
2.12	<i>Network Address Translation (NAT)</i>	40
2.13	<i>Routing</i>	42
2.13.1	<i>Static Route</i>	43

2.14	<i>Microsoft Visio</i>	43
2.15	<i>VMware Workstation</i>	44
2.16	Mikrotik	45
2.17	Winbox	48
2.18	<i>Monitoring Jaringan</i>	49
2.18.1	<i>Torch</i>	49
2.18.2	<i>Interface List</i>	49
2.18.3	<i>Speedtest.net</i>	50
2.18.4	<i>Webproxy Log</i>	50
2.19	Studi Literatur	50
BAB III METODOLOGI PENELITIAN		
3.1	Metode Pengumpulan Data	53
3.19.1	Studi Pustaka	53
3.19.2	Studi Lapangan	53
3.19.3	Studi Literatur	54
3.2	Metode Pengembangan Sistem	54
3.2.1	<i>Analysis</i>	55
3.2.2	<i>Design</i>	55
3.2.3	<i>Simulation Prototyping</i>	56
3.2.4	<i>Implementation</i>	56
3.2.5	<i>Monitoring</i>	57
3.2.6	<i>Management</i>	57
3.3	Kerangka Berpikir	58

BAB IV PEMBAHASAN

4.1	Profil Sekolah Menengah Kejuruan PGRI Bekasi	60
4.1.1	Visi	60
4.1.2	Misi	60
4.2	<i>Analysis</i>	61
4.2.1	Analisa Sistem Berjalan	62
4.2.2	Spesifikasi <i>Software</i> dan <i>Hardware</i>	68
4.3	<i>Design</i>	69
4.3.1	Perancangan Fisik	70
4.4	<i>Simulation Prototyping</i>	71
4.5	<i>Implementation</i>	74
4.5.1	Implementasi Topologi Jaringan	75
4.5.2	Inisialisasi <i>Interface</i> Mikrotik	75
4.5.3	Pemberian Alamat IP	75
4.5.4	Konfigurasi <i>Mangle</i>	78
4.5.5	Konfigurasi <i>Routing</i>	81
4.5.6	Konfigurasi NAT	82
4.6	<i>Monitoring</i>	83
4.6.1	Pengujian Efektifitas Penyetaraan Beban	84
4.6.2	Pengujian Performa <i>Load Balancing</i>	86
4.7	<i>Management</i>	87
4.7.1	Membuat Pengaturan <i>Fail Over</i>	88
4.7.2	Mengganti <i>Username</i> dan <i>Password</i>	88

BAB IV PENUTUP

4.1 Kesimpulan 91

4.2 Saran 92

DAFTAR PUSTAKA 93

DAFTAR LAMPIRAN



DAFTAR GAMBAR

Gambar 1.1 Metodologi penelitian NDLC	7
Gambar 2.1 <i>Load balancing</i> dengan dual ISP	11
Gambar 2.2 <i>Nth load balancing</i>	15
Gambar 2.3 Hasil pengujian ISP IM2	18
Gambar 2.4 Hasil pengujian ISP Smart	19
Gambar 2.5 Pemanfaatan printer pada LAN	20
Gambar 2.6 Koneksi internet pada LAN	21
Gambar 2.7 <i>Local Area Network</i>	22
Gambar 2.8 <i>Metropolitan Area Network</i>	23
Gambar 2.9 <i>Wide Area Network</i>	24
Gambar 2.10 Topologi star	26
Gambar 2.11 Modem Router ZTE MF608	29
Gambar 2.12 TP-Link TL-MR3420	30
Gambar 2.13 Tugas-tugas OSI Layer	34
Gambar 2.14 Format IPv4	38
Gambar 2.15 Kelas-kelas IP Address	39
Gambar 2.16 Mikrotik RouterBoard 750	47
Gambar 3.1 Metodologi penelitian NDLC	54
Gambar 3.2 Kerangka berpikir	59
Gambar 4.1 Topologi jaringan laboratorium komputer	62
Gambar 4.2 Konfigurasi <i>Webproxy log</i>	64

Gambar 4.3 Grafik 10 <i>website</i> terbanyak yang diakses	64
Gambar 4.4 Aplikasi <i>monitoring Torch</i>	65
Gambar 4.5 Desain topologi jaringan dengan <i>load balancing</i>	70
Gambar 4.6 VMware workstation ver 7.0	73
Gambar 4.7 Alur proses pengiriman paket pada Nth <i>load balancing</i>	74
Gambar 4.8 Konfigurasi PC <i>client</i>	77
Gambar 4.9 Konfigurasi modem <i>router</i>	78
Gambar 4.10 Grafik koneksi pada tiap <i>gateway</i> ISP	84
Gambar 4.11 Grafik perbandingan penyebaran paket data	85
Gambar 4.12 Teknik <i>Fail over</i>	89
Gambar 4.13 Mengganti <i>password</i> di mikrotik	89
Gambar 4.14 Mengganti <i>password</i> di modem <i>router</i>	90

DAFTAR TABEL

Tabel 2.1 Perbedaan masing-masing metode <i>load balancing</i>	16
Tabel 2.2 Perbandingan LAN, MAN dan WAN	24
Tabel 2.3 Rangkuman Kelas di IPv4	40
Tabel 2.4 Menu navigasi mikrotik	47
Tabel 4.1 Daftar <i>traffic</i> dari <i>tool torch</i>	66
Tabel 4.2 Spesifikasi <i>software</i>	68
Tabel 4.3 Spesifikasi <i>hardware</i>	68
Tabel 4.4 Tabel <i>IP address</i>	71
Tabel 4.4 Tabel perbandingan penyebaran paket data	85
Tabel 4.5 Pengujian sebelum implementasi <i>load balancing</i>	86
Tabel 4.6 Pengujian setelah implementasi <i>load balancing</i>	87



DAFTAR LAMPIRAN

Lampiran A Surat SK Dosen Pembimbing

Lampiran B Surat Keterangan Riset

Lampiran C Simulasi di *VMware Workstation*

Lampiran B Gambar Pengujian Kecepatan Koneksi

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kebutuhan akan akses internet saat ini sangat tinggi, baik untuk mencari informasi, artikel maupun pengetahuan terbaru. Banyak sekolah yang telah mengintegrasikan jaringan internet kedalam proses belajar-mengajar. Itu diharapkan agar siswa dapat dengan mudah mencari materi dan memahami pelajaran.

Seiring dengan bertambahnya pengguna internet tersebut, agar jaringan benar-benar optimal, selain pengaturan *IP address* perlu juga dilakukan pengaturan *routing*. *Device* yang digunakan untuk proses *routing* disebut *router*. Namun karena harga dari *router* relatif mahal, oleh karena itu ada alternatif *hardware* lain yaitu Mikrotik. Mikrotik *RouterOS* merupakan sistem operasi yang mampu membuat komputer menjadi *router* atau sering disebut *PC Router*. Sistem operasi tersebut mencakup berbagai fitur lengkap untuk *wireline* dan *wireless*, antara lain adalah *bandwidth management*, *proxy server*, *hotspot*, *load balancing* dan sebagainya.

Untuk saat ini topologi dari suatu jaringan LAN umumnya menggunakan topologi *star* (bintang). Dimana karakteristik dari topologi *star* (bintang) yaitu mempunyai banyak *client* yang dihubungkan ke satu *server* atau *switch*. Oleh karena itu tidak menutup kemungkinan, *client* akan mengalami kepadatan jalur transfer data atau akses yang tidak lancar.

Di Indonesia sendiri para penyedia layanan internet atau yang lebih dikenal dengan ISP (*Internet Service Provider*), tidak dapat menyediakan layanan internet yang murah dan handal. Banyak kelebihan dan kekurangan antara satu *provider* dengan *provider* lainnya. Terlebih untuk suatu daerah yang tidak cukup terjangkau oleh *provider* tersebut.

SMK PGRI Bekasi adalah salah satu instansi pendidikan yang telah menjadikan Teknik Komputer Jaringan sebagai salah satu kejuruan yang ada di sekolah tersebut. Dan hampir setiap proses belajar-mengajar juga memerlukan koneksi internet untuk memudahkan siswa mencari materi pembelajaran. Maka daripada itu, SMK PGRI Bekasi menginginkan suatu koneksi internet yang stabil dan handal. Akan tetapi daerahnya yang kurang terjangkau oleh jaringan *line telephone* dari Telkom Speedy dan juga BTS (*Base Transceiver Station*) dari *provider-provider* lainnya.

Oleh karena itu timbul solusi untuk menggunakan dua ISP dan menjadikan mikrotik tersebut sebagai *load balancer*. Dan diharapkan juga Mikrotik dapat mengoptimalkan pembagian *bandwidth* pada setiap *client* yang ingin mengakses internet. Mekanismenya yaitu mikrotik akan menandai paket yang ingin mengakses internet, lalu menyetarakan beban pada kedua ISP dan akan memilih jalur ISP mana yang akan dilewatinya. Dalam tugas akhir ini penulis mengambil judul **“Implementasi Load Balancing Dua ISP Menggunakan Mikrotik (Studi Kasus : Laboratorium Komputer SMK PGRI Bekasi)”**.

1.2 Perumusan Masalah

Berdasarkan latar belakang tersebut maka dalam penelitian ini merumuskan masalah sebagai berikut :

1. Bagaimana solusi alternatif untuk pemberian *bandwidth* yang optimal pada jaringan LAN di Laboratorium SMK PGRI Bekasi?
2. Apakah penggunaan fitur *load balancing* yang akan menyetarakan beban *traffic* pada kedua koneksi internet?
3. Bagaimana penggunaan teknik *fail over* jika salah satu jalur koneksi terjadi masalah seperti terputus atau mati?

1.3 Batasan Masalah

Penulis membatasi masalah yang ada pada penelitian ini pada beberapa hal berikut :

1. Penulis akan menganalisa dan merancang LAN dengan topologi *star* menggunakan Mikrotik RB750 sebagai *load balancer*.
2. Setelah melakukan tahapan analisa, penulis akan menentukan metode *load balancing* yang tepat dengan menggunakan Mikrotik RouterOS.
3. Jumlah koneksi internet yang akan di *load balancing* menggunakan 2 ISP dari *provider* yang berbeda.
4. Melakukan *monitoring* dan *capture traffic* pada LAN tersebut.
5. Optimasi jaringan dilihat dari *monitoring traffic* jaringan dari penggunaan jalur koneksi internet agar tidak ada salah satu koneksi yang mengalami *overload*.

1.4 Tujuan dan Manfaat Penelitian

1.4.1 Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Mengimplementasikan fitur *load balancing* pada mikrotik RouterOS agar dapat lebih efektif dalam meratakan beban *traffic* pada kedua jalur koneksi internet.
2. Menentukan metode *load balancing* yang tepat sesuai dengan karakteristik jaringan.
3. Mengetahui cara kerja *load balancing* pada mikrotik agar dapat menyetarakan beban di kedua koneksi internet.
4. Sistem ini dibuat untuk menyelesaikan masalah kecepatan akses internet yang menggunakan dua koneksi internet.

1.4.2 Manfaat Penelitian

1. Bagi penulis

- a. Menerapkan ilmu yang diperoleh di bangku kuliah.
- b. Memahami bagaimana teori, konsep dan praktek tentang mikrotik dan salah satu fitur *load balancing*.
- c. Bertambahnya wawasan dan pengalaman penulis tentang ilmu jaringan dan hal lainnya yang berkaitan dengan metodologi penulisan tugas akhir ini.

2. Bagi Instansi Pendidikan

- a. Sebagai salah satu alternatif cara yang dilakukan dalam memperoleh koneksi internet yang handal.

- b. Memberikan pilihan solusi penerapan *load balancing* dalam pembagian beban *traffic* jaringan internet.
- c. Dengan akses internet yang memadai dapat mempermudah para guru dan siswa dalam proses belajar mengajar.

3. Bagi masyarakat umum

- a. Dapat memberikan kontribusi pemikiran tentang teknologi informasi yang bermanfaat bagi masyarakat pada umumnya dan *civitas* akademik kampus UIN Syarif Hidayatullah Jakarta pada khususnya.
- b. Menjadi referensi literatur mengenai “Implementasi *load balancing* dua ISP menggunakan mikrotik” di perpustakaan universitas.

1.5 Metodologi Penelitian

Metode yang digunakan penulis dalam penulisan bagian metodologi penelitian dibagi menjadi dua, yaitu metode pengumpulan data dan metode pengembangan sistem.

1.5.1 Metode Pengumpulan Data

Merupakan metode yang digunakan penulis dalam melakukan penelitian dan menjadikannya informasi yang akan digunakan untuk mengetahui permasalahan yang dihadapi.

1. Studi Pustaka

Studi pustaka merupakan pengumpulan bahan-bahan yang berkaitan dengan judul skripsi melalui membaca buku-buku dari perpustakaan dan mencari referensi artikel serta *ebook* dari *internet*.

2. Observasi

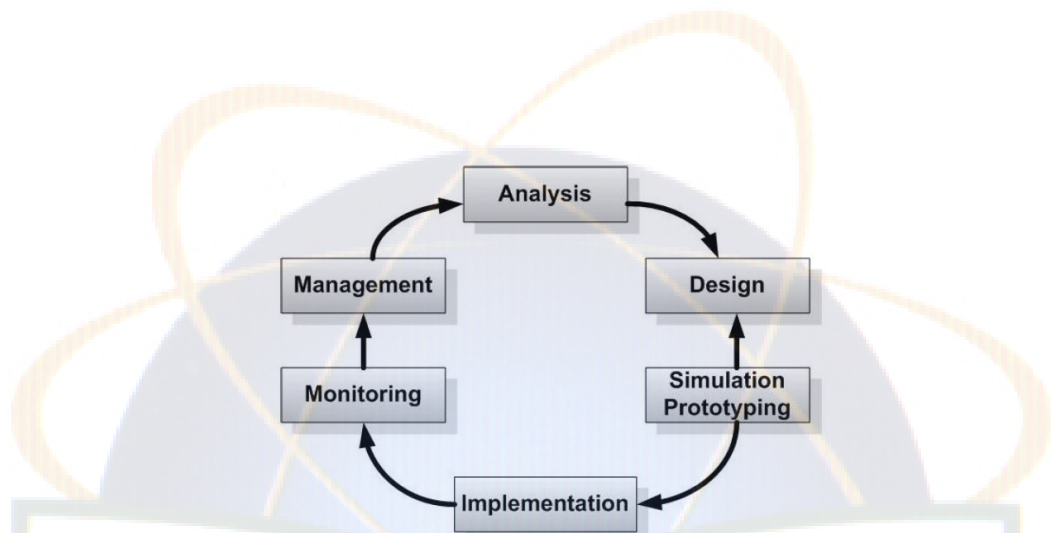
Pengamatan langsung ke lapangan (observasi) yang dilakukan oleh penulis, tempat penelitian yaitu pada Laboratorium Komputer SMK PGRI Kota Bekasi, Jalan Makrik, Rawalumbu, Bekasi Timur.

3. Studi Literatur

Pada tahap ini penulis dalam melakukan penelitian akan menjadikan bahan studi literatur sejenis sebagai acuan untuk mengetahui kekurangan maupun kelebihan dari sistem yang telah dibuat sebelumnya.

1.5.2 Metode Pengembangan Sistem

Penelitian menggunakan *Network Development Life Cycle* (NDLC) sebagai acuan dalam membuat tugas akhir ini. pada gambar 1.1 merupakan siklus dari NDLC.



Gambar 1.1 Metodologi penelitian NDLC.
(Sumber: Deris Stiawan, 2009)

Berikut adalah penjelasan dari masing-masing tahap dalam *Network Development Life Cycle* (NDLC) :

1. **Analysis** : Melakukan analisa pada sistem yang telah berjalan, lalu menentukan metode *load balancing* yang tepat dan selanjutnya mengumpulkan data bagaimana membuat sebuah jaringan dengan menggunakan mikrotik sebagai *load balancer* dua ISP.
2. **Design** : Bentuk desain dalam melakukan perancangan adalah topologi jaringan *star* yang melakukan pembagian beban *bandwidth* yang adil serta proses penentuan jalur *traffic* antara dua buah ISP yang tersedia.
3. **Simulation Prototyping** : Bentuk simulasinya adalah dengan menggunakan *virtual machine* yang berdasarkan rancangan penelitian dimana terdapat user, mikrotik sebagai *load balancer*, dan dua buah modem dari ISP yang berbeda.

4. **Implementations** : Pada tahap ini menerapkan semua yang telah direcanakan dan didesain untuk diuji apakah berhasil atau tidak topologi jaringan yang telah dirancang sebelumnya.

5. **Monitoring** : Dalam tahap ini perlunya pemantauan terhadap *traffic* di jaringan agar berjalan sesuai dengan analisa awal dan pemantauan terhadap infrastruktur *hardware*.

6. **Management** : Tahap ini memerlukan perhatian khusus terhadap kebijakan yang perlu dibuat untuk mengatur dan membuat sistem agar dapat berjalan dengan baik.

1.5 Sistematika Penulisan

Untuk memudahkan dalam pemahaman tugas akhir ini maka penulis menyusun sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini membahas secara singkat teori yang diperlukan dalam penelitian skripsi.

BAB III METODE PENELITIAN


Bab ini akan membahas metode penelitian yang digunakan dalam perancangan sistem.

BAB IV PERANCANGAN DAN ANALISA SISTEM

Bab ini akan membahas tentang perancangan serta analisa dengan pengujian terhadap jaringan yang dibuat menggunakan parameter yang telah ditentukan.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan dari hasil uji coba serta analisa yang dilakukan serta saran-saran yang dibutuhkan untuk pengembangan lebih lanjut.



BAB II

LANDASAN TEORI

2.1 Implementasi

Implementasi adalah pelaksanaan dan penerapan, dimana pelaksanaan berarti proses, cara, perbuatan melaksanakan. Sedangkan penerapan berarti pemasangan, pengenalan, perihal mempratekkan (Zul Fajri, 2004).

Implementasi bermuara pada aktivitas, adanya aksi, tindakan atau mekanisme suatu sistem. Ungkapan mekanisme mengandung arti bahwa implementasi bukan sekedar aktivitas, tetapi suatu kegiatan yang terencana dan dilakukan secara sungguh-sungguh berdasarkan acuan norma tertentu untuk mencapai tujuan kegiatan (Nurman dan Usman, 2004).

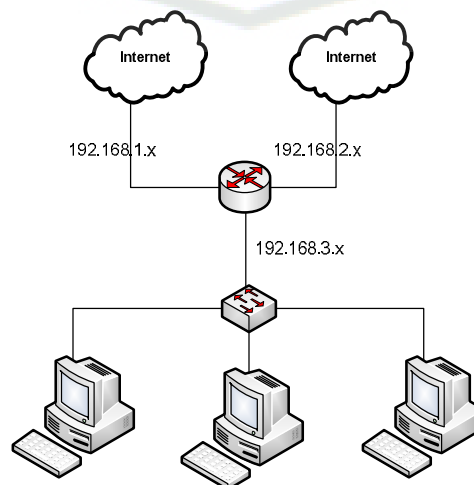
2.2 Load Balancing

Load balancing adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi. (dewobroto, 2009).

Dengan mempunyai banyak *link* maka optimalisasi utilitas sumber daya, *throughput*, atau *response time* akan semakin baik karena mempunyai lebih dari satu *link* yang bisa saling mem-*backup* pada saat

salah satu *link* koneksi *down* dan menjadi cepat pada saat *network* normal memerlukan realibilitas tinggi yang memerlukan 100 % koneksi *uptime* dan yang menginginkan *upstream* yang berbeda dan dibuat saling *mem-backup* (setiawan, 2009).

Selama ini banyak yang beranggapan salah, bahwa dengan menggunakan *load balancing* dua jalur koneksi, maka besar *bandwidth* yang akan didapatkan menjadi dua kali lipat dari *bandwidth* sebelum menggunakan *loadbalance* (akumulasi dari kedua *bandwidth* tersebut). Hal ini perlu diperjelas dahulu, bahwa *loadbalance* tidak akan menambah besar *bandwidth* yang diperoleh, tetapi hanya bertugas untuk membagi trafik dari kedua *bandwidth* tersebut agar dapat terpakai secara seimbang. Bahwa dalam penggunaan *load balancing* tidaklah seperti rumus matematika $1+1=2$ akan tetapi $1+1=1+1$.



Gambar 2.1 Load balancing dengan dual ISP.

Sumber: (<http://mikrotik.co.id>)

Dalam sistem *load balancing*, proses pembagian bebannya memiliki teknik dan algoritma tersendiri. Pada perangkat *load balancing* yang kompleks biasanya disediakan bermacam-macam algoritma pembagian beban ini. Tujuannya adalah untuk menyesuaikan pembagian beban dengan karakteristik dari *server-server* yang ada di belakangnya.

Dalam *load balancing* di mikrotik, hal-hal yang perlu diperhatikan dalam mengatur *load balancing* adalah *static route*, *policy route*, *firewall mangle* dan *firewall src-nat*. *Firewall mangle* adalah penandaan paket sebelum masuk routing. sedangkan *static route* dan *policy route* mengatur mengatur *uplink flow*, yaitu mengenai kebijakan *routing* atau rute jalur yang akan dilalui paket yang telah ditandai.

Ada berbagai metode *load balancing*, antara lain *static route* dengan *address list*, *Equal Cost Multi Path* (ECMP), *Nth* dan *Per Connection Classifier* (PCC). Setiap metode *load balancing* tersebut memiliki kekurangan maupun kelebihan tersendiri, namun lebih dari hal itu, yang paling terpenting dalam menentukan metode *load balancing* apa yang akan digunakan adalah harus terlebih dahulu mengerti karakteristik dari jaringan yang akan diimplementasikan. Berikut ini adalah sedikit pengertian dari masing-masing metode *load balancing* dan disertakan pula kekurangan maupun kelebihanannya.

2.2.1 *Static Route dengan Address List*

Static route dengan *address list* adalah metode *load balancing* yang mengelompokkan suatu *range IP address* untuk

diatur agar dapat melewati salah satu *gateway* dengan menggunakan *static routing*.

Metode ini sering digunakan pada warnet yang membedakan PC untuk *browsing* dan PC untuk *game online*. Mikrotik akan menentukan jalur *gateway* yang dipakai dengan membedakan *src-address* pada paket data.

2.2.2 Equal Cost Multi Path (ECMP)

Equal Cost Multi Path (ECMP) adalah pemilihan jalur keluar secara bergantian pada *gateway*. Contoh jika ada dua *gateway* dia akan melewati kedua *gateway* tersebut dengan beban yang sama (*equal cost*) pada masing-masing *gateway*.

Nilai dari *equal cost* dapat pula didefinisikan secara asimetris atau tidak seimbang pada saat *routing*. Ini dikarenakan jika diantara kedua ISP memiliki kecepatan koneksi yang berbeda jauh. Contoh jika kedua *gateway* mempunyai kecepatan koneksi sebesar 1 Mbps dan 3 Mbp, maka pada saat konfigurasi *routing* akan menjadi “*ip route add dst-address=0.0.0.0/0 gateway=10.10.0.1, 10.10.0.2, 10.10.0.2, 10.10.0.2 check-gateway=ping*” yang diartikan bahwa *gateway* pertama dan kedua berbanding 1:3.

2.2.3 Nth

Nth bukanlah sebuah singkatan, melainkan Nth adalah sebuah integer (bilangan ke-N). Nth menggunakan algoritma *round*

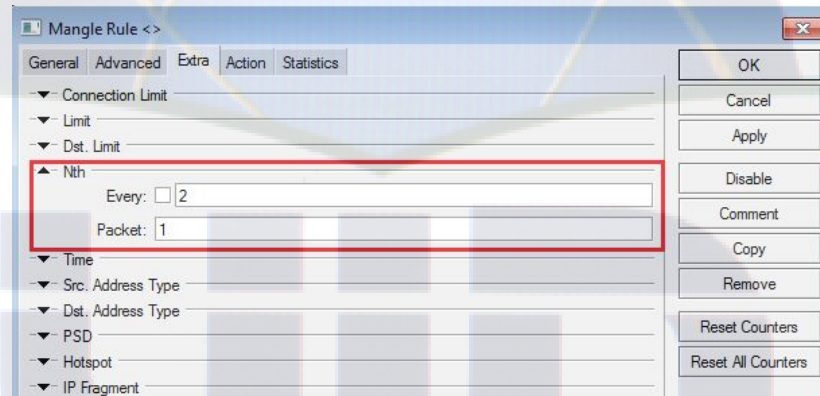
robin yang menentukan pembagian pemecahan *connection* yang akan di-*mangle* ke rute yang dibuat untuk *load balancing*.

Pada dasarnya koneksi yang masuk ke proses di *router* akan menjadi satu arus yang sama, walaupun mereka datang dari interface yang berbeda. Maka pada saat menerapkan metode Nth, tentunya akan memberikan batasan ke *router* untuk hanya memproses koneksi dari sumber tertentu saja. Ketika *router* telah membuat semacam antrian baru untuk batasan yang kita berikan diatas, baru proses Nth dimulai.

Di dalam Nth terdapat variabel yang harus dimengerti, yaitu

1. *Every*: Angka *every* adalah jumlah kelompok yang ingin dihasilkan. Jadi bila administrator ingin membagi alur koneksi yang ada menjadi 2 kelompok yang nantinya akan di *load balance* ke 2 koneksi yang ada, maka angka *every* = 2.
2. *Packet*: Angka *packet* adalah jumlah koneksi yang akan ditandai atau di-*mangle*. Jika ingin membuat 2 kelompok, tentunya harus membuat 2 *mangle rules*. Pada *rules* tersebut, angka untuk *Every* haruslah sama, namun untuk angka *packet* harus berubah. Untuk 2 kelompok, berarti angka *packet* untuk 2 *rules* tersebut adalah 1 dan 2.
3. *Counter*: *Counter* atau disebut penghitung atau pencacah biner. Mulai dari mikrotik versi 3.x nilai *counter* tidak didefinisikan langsung oleh administrator. Setiap *rules* memiliki *counter*

sendiri. Ketika *rules* menerima paket, maka *counter* untuk aturan saat itu akan otomatis bertambah satu. Dan jika nilai *counter* sama dengan nilai “every”, maka paket akan dicocokkan dan *counter* akan diatur ke nilai awal.



Gambar 2.2 Nth load balancing

2.2.4 Per Connection Classifier (PCC)

Per Connection Classifier (PCC) merupakan metode yang menspesifikasikan suatu paket menuju ke *gateway* koneksi tertentu. PCC mengelompokkan trafik koneksi yang melalui atau keluar masuk *router* menjadi beberapa kelompok. Pengelompokan ini bisa dibedakan berdasarkan *src-address*, *dst-address*, *src-port* dan atau *dst-port*. Mikrotik akan mengingat-ingat jalur *gateway* yang telah dilewati diawal trafik koneksi, sehingga pada paket-paket data selanjutnya yang masih berkaitan akan dilewatkan pada jalur *gateway* yang sama dengan paket data sebelumnya yang sudah dikirim.

Tabel 2.1 Perbedaan masing-masing metode load balancing

Metode	Kelebihan	Kekurangan
<i>Static Route</i> dengan <i>Address List</i>	<ul style="list-style-type: none"> dapat membuat topologi jaringan yang sederhana. tidak ada diskoneksi pada <i>client</i> yang disebabkan perpindahan <i>gateway</i> karena <i>load balancing</i> 	<ul style="list-style-type: none"> dapat terjadi <i>overload</i> jika yang aktif hanya pada <i>client-client</i> pada salah satu <i>address list</i> saja.
ECMP	<ul style="list-style-type: none"> dapat membagi beban jaringan berdasarkan perbandingan kecepatan diantara 2 ISP 	<ul style="list-style-type: none"> sering terjadinya diskoneksi yang disebabkan oleh <i>routing table</i> yang merestart secara otomatis setiap 10 menit
Nth	<ul style="list-style-type: none"> dapat membagi penyebaran paket data yang merata pada masing-masing <i>gateway</i> 	<ul style="list-style-type: none"> kemungkinan terjadi terputusnya koneksi yang disebabkan perpindahan <i>gateway</i> karena <i>load balancing</i>
PCC	<ul style="list-style-type: none"> mampu menspesifikasi <i>gateway</i> untuk tiap paket data yang masih berhubungan dengan data yang sebelumnya sudah dilewatkan pada salah satu <i>gateway</i> 	<ul style="list-style-type: none"> dapat terjadinya <i>overload</i> pada salah satu <i>gateway</i> yang disebabkan pengaksesan situs yang sama

2.3 Internet Service Provider (ISP)

Internet Service Provider adalah perusahaan atau badan yang menyelenggarakan jasa sambungan internet dan jasa lainnya yang berhubungan. Kebanyakan perusahaan telepon merupakan penyelenggara jasa Internet. Mereka menyediakan jasa seperti hubungan ke internet, pendaftaran nama *domain*, dan *hosting*.

Berdasarkan catatan whois ARIN dan APNIC, protokol Internet (IP) pertama dari Indonesia, UI-NETLAB (192.41.206/24) didaftarkan oleh Universitas Indonesia pada 24 Juni 1988. Di sekitar tahun 1994 mulai

beroperasi IndoNet yang dipimpin oleh Sanjaya. IndoNet merupakan ISP komersial pertama Indonesia. Pada waktu itu masih sedikit sekali pengguna Internet di Indonesia. Sambungan awal ke Internet dilakukan menggunakan dial-up oleh IndoNet. Akses awal di IndoNet mula-mula memakai mode teks dengan *shell account*, *browser lynx* dan *email client pine* pada *server AIX*.

Mulai 1995 beberapa BBS di Indonesia seperti Clarissa menyediakan jasa akses telnet ke luar negeri. Dengan memakai *remote browser Lynx* di AS, maka pemakai internet di Indonesia bisa akses internet.

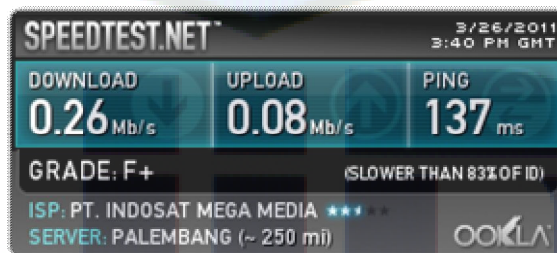
2.3.1 Indosat M2

Indosat Mega Media (IndosatM2) adalah sebuah perusahaan yang dimiliki sepenuhnya oleh PT Indosat Tbk, penyelenggara jasa telekomunikasi di Indonesia, beroperasi secara penuh sejak tahun 2000 untuk membangun dan menerapkan jasa dan produk berbasis IP, internet dan multimedia di Indonesia. (www.indosatm2.com)

IndosatM2 melayani empat segmen pelanggan: korporasi (besar, menengah, dan kecil), pemerintah, institusi, residensial dan perorangan. Layanan korporasi dan institusi meliputi jaringan Virtual Private Network (VPN), sambungan langsung ke backbone internet internasional, serta penyedia layanan multimedia.

Dalam melakukan penelitian, penulis menggunakan paket *Broom Unlimited*. Akses internet prabayar *unlimited* dengan kuota yang melalui jaringan 3.5G dari IM2 dengan metode pembayaran tetap setiap bulannya sebesar Rp 200.000. *User* akan mendapatkan

bandwidth up to 256 Kbps dan quota sebesar 5 GB. Kemudian jika melebihi batas quota, maka setelah itu kecepatan akses akan turun menjadi lebih rendah dari 64 Kbps sampai volume pemakaian yang tidak terbatas. Berikut ini pada gambar 2.3 adalah hasil dari www.speedtest.net mengenai kecepatan *download*, *upload rate* dan waktu *ping* dari ISP IM2.



Gambar 2.3 Hasil Pengujian ISP IM2

Sumber: (www.speedtest.net)

2.3.2 Smart

Smart Telecom yang merupakan bagian dari Sinar Mas Group, adalah perusahaan yang bergerak di bidang jasa telekomunikasi dan berfungsi sebagai operator selular. Smart menggunakan teknologi CDMA dengan frekuensi 1900 MHz.

Selain itu Smart juga memiliki layanan akses data nirkabel kecepatan tinggi yang berbasis teknologi EVDO Rev. A. dan pada awal Januari 2010 Smart juga meluncurkan layanan EVDO–Rev. B secara komersial yang pertama di dunia. (www.smart-telecom.co.id)

Dalam melakukan penelitian, penulis menggunakan paket *Silver*. Akses internet prabayar *unlimited* tanpa quota melalui teknologi CDMA EVDO Rev A. dengan kecepatan *download up to*

512 Kbps dan *upload up to* 128 Kbps dengan metode pembayaran tetap setiap bulannya sebesar Rp 75.000. Berikut ini pada gambar 2.4 merupakan hasil dari www.speedtest.net mengenai kecepatan *download*, *upload rate* dan waktu *ping* dari ISP Smart.



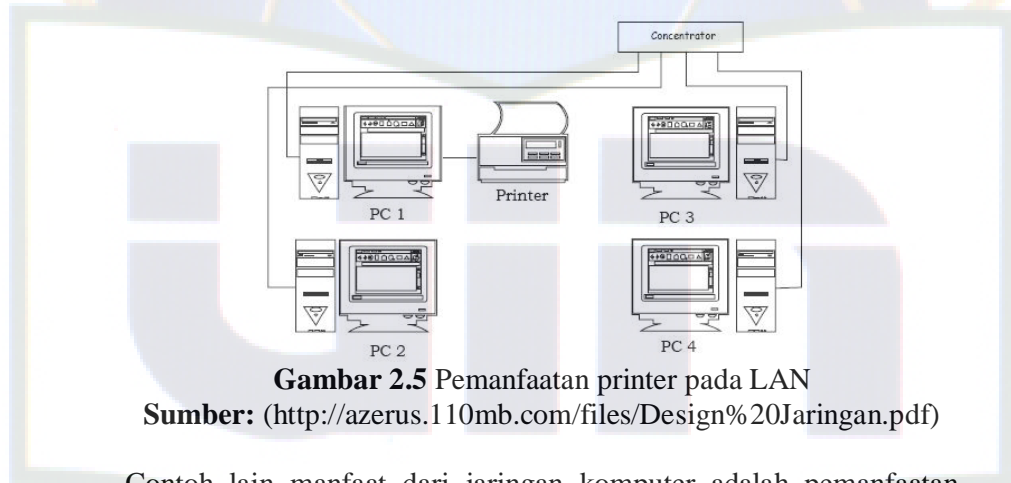
Gambar 2.4 Hasil pengujian ISP Smart
Sumber: (www.speedtest.net)

2.4 Jaringan Komputer

Jaringan komputer adalah kumpulan dua atau lebih komputer yang saling berhubungan untuk melakukan komunikasi data. Komunikasi data yang bisa dilakukan melalui jaringan komputer dapat berupa data teks, gambar, video, dan suara. Untuk membangun sebuah jaringan komputer harus diperhatikan tentang situasi dan kondisi organisasi yang akan membangun jaringan tersebut, misalnya struktur bangunan, jangkauan, kecepatan akses, biaya operasional, dan sebagainya (Syafrizal, 2007).

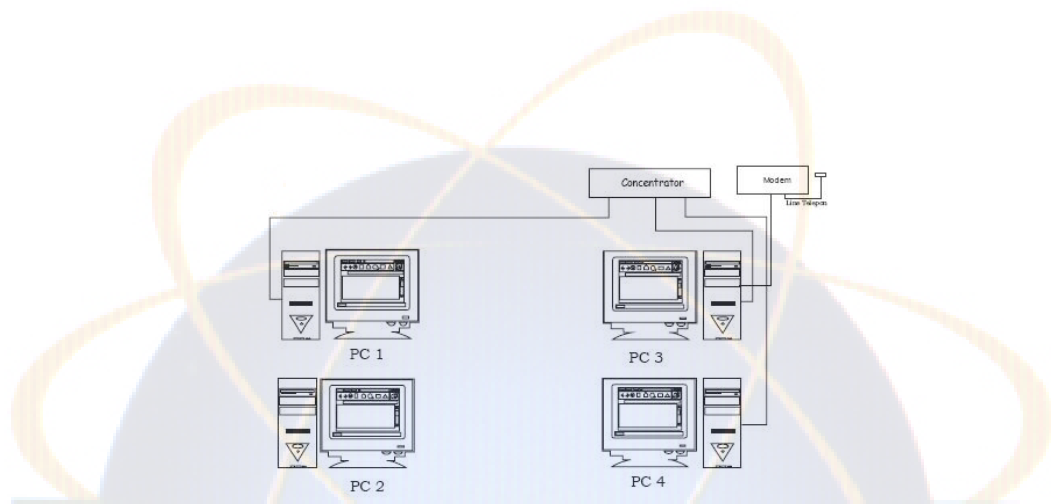
Komputer yang bersifat *stand alone* atau berdiri sendiri mempunyai banyak keterbatasan. Adanya jaringan komputer akan membuat komputer dapat melakukan banyak hal dan dapat membantu efisiensi dan efektivitas dalam dunia kerja. Contoh sederhananya saja, dengan adanya jaringan komputer, maka tidak perlu lagi 1 (satu) komputer memiliki 1 (satu) printer,

tetapi dengan 1 (satu) printer saja dapat digunakan oleh beberapa komputer secara bersama tanpa harus memindahkan printer tersebut setiap kali akan mencetak. Gambar 2.5 menjelaskan bahwa cukup menggunakan satu printer saja untuk beberapa unit komputer. Printer tersebut dapat dipasang pada komputer mana saja yang terhubung jaringan komputer tersebut.



Contoh lain manfaat dari jaringan komputer adalah pemanfaatan internet secara bersama sehingga tidak perlu lagi untuk 1 (satu) unit komputer dengan 1 (satu) modem atau dengan 1 (satu) *line* telepon.

Dengan adanya LAN, cukup dengan 1 (satu) modem dan 1 (satu) *line* telepon saja yang terpasang pada 1 (satu) komputer dapat digunakan koneksi internet untuk beberapa unit komputer. Kita dapat lihat pada gambar 2.6 dengan 1 (satu) modem dan 1 (satu) *line* telepon saja, maka semua komputer dapat terkoneksi ke internet.



Gambar 2.6 Koneksi internet pada LAN

Sumber: (<http://azerus.110mb.com/files/Design%20Jaringan.pdf>)

Sisi kecil lain yang dapat dilihat manfaat adanya jaringan komputer adalah pertukaran data, sehingga tidak perlu lagi menggunakan media penyimpan seperti disket ataupun flash disk dalam melakukan pemindahan data dari satu komputer ke komputer lain. Perkembangan lain dari jaringan komputer dari sisi bisnis adalah memanfaatkan pelaporan data dan *monitoring* perkembangan bisnis, seperti controlling peningkatan target penjualan barang yang dapat dilakukan dari luar kota bahkan luar negeri dengan memanfaatkan jaringan komputer.

2.5 Bentuk Jaringan

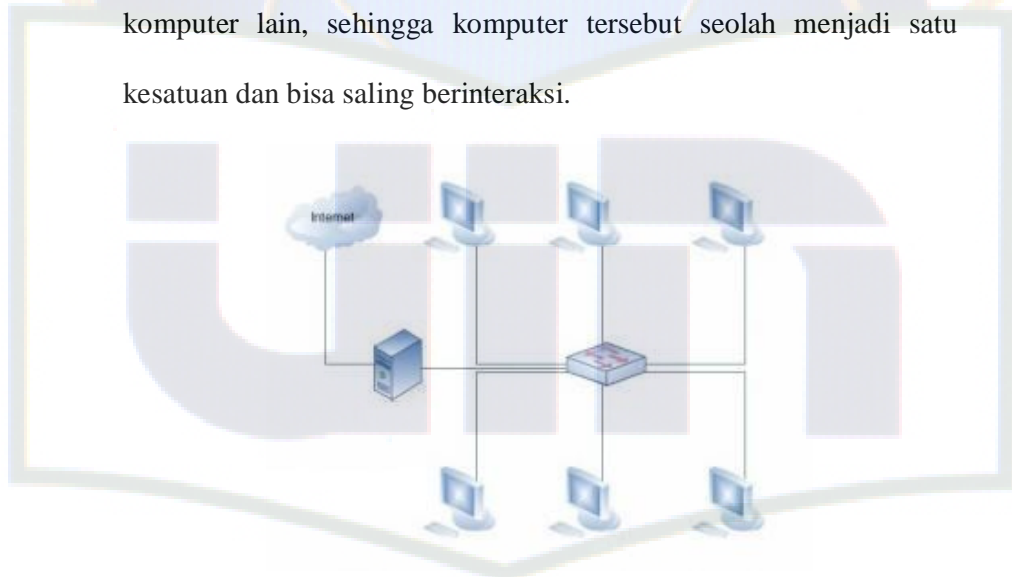
Jaringan komputer dapat dibagi menjadi 3 (tiga) bagian jika dilihat dari sisi geografis. Adapun bagian-bagian tersebut adalah sebagai berikut.

2.5.1 *Local Area Network* (LAN)

LAN merupakan jaringan komputer dengan ruang lingkup terbatas, meliputi lokasi seperti gedung, kampus, kantor, atau pabrik. Tipe ini banyak digunakan untuk perkantoran, bisnis, laboratorium,

dan sebagainya dengan skala kecil seperti warnet, rental komputer, laboratorium komputer, dan sebagainya.

Sebuah LAN dapat dibangun dengan minimal 2 (dua) komputer dengan spesifikasi (kapasitas) komputer rendah sekalipun. Adanya LAN akan menjadikan komputer terhubung dengan komputer lain, sehingga komputer tersebut seolah menjadi satu kesatuan dan bisa saling berinteraksi.



Gambar 2.7 *Local Area Network*

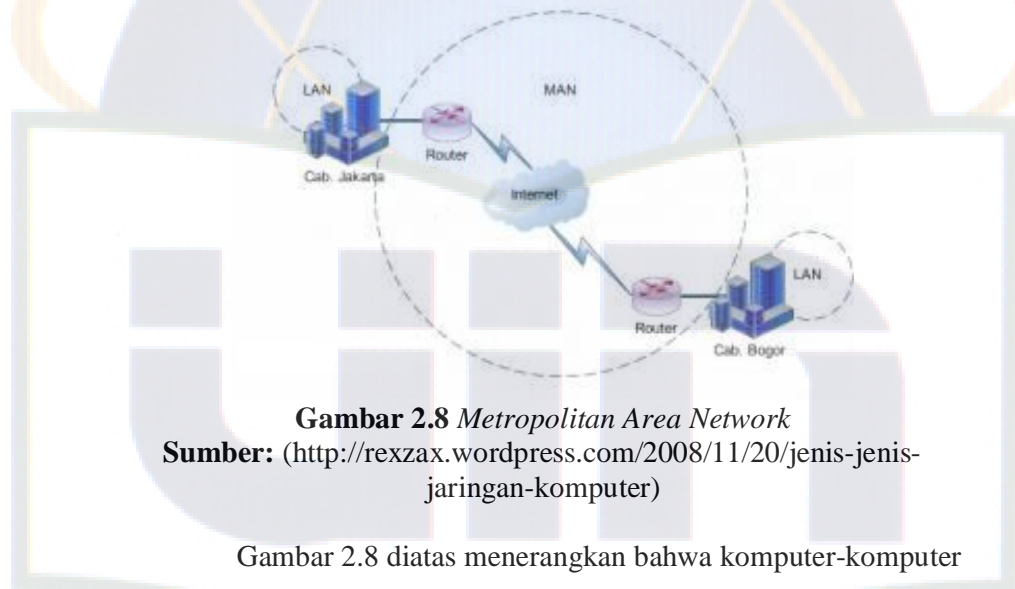
Sumber: (<http://rexzax.wordpress.com/2008/11/20/jenis-jenis-jaringan-komputer>)

2.5.2 Metropolitan Area Network (MAN)

Metropolitan Area Network (MAN), adalah sebuah jaringan yang menggunakan teknologi yang sama dengan LAN, hanya saja ukurannya biasanya lebih luas daripada LAN (Syafriyazal, 2007).

Jenis jaringan komputer ini adalah jaringan komputer yang memungkinkan jarak yang cukup jauh. Tipe ini digunakan untuk membangun jaringan komputer antargedung, dalam satu kota, atau antarkota yang berada pada jangkauannya. Jaringan ini biasanya

digunakan oleh perusahaan-perusahaan besar seperti perbankan, BUMN, perusahaan penjualan motor, dan lain-lain. Simulasinya dapat kita lihat pada gambar 2.8.

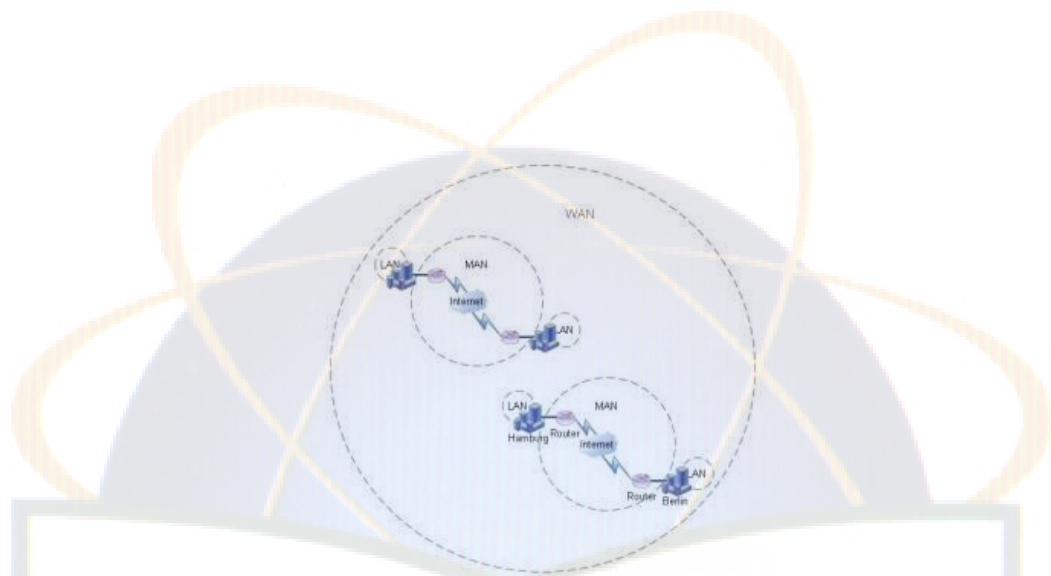


Gambar 2.8 *Metropolitan Area Network*
Sumber: (<http://rexzax.wordpress.com/2008/11/20/jenis-jenis-jaringan-komputer>)

Gambar 2.8 diatas menerangkan bahwa komputer-komputer yang berada di kantor pada sebuah kota tertentu dapat terhubung dengan komputer yang berada di kota lain.

2.5.3 *Wide Area Network (WAN)*

Jaringan jenis ini merupakan jaringan terbesar karena mencakup radius antarnegara bahkan benua tanpa batasan geografis seperti jenis jaringan yang lain. WAN terdiri dari kumpulan LAN, MAN dan mesin-mesin yang bertujuan untuk menjalankan program aplikasi pemakai (Syafriyazal, 2007).



Gambar 2.9 Wide Area Network

Sumber: (<http://rexzax.wordpress.com/2008/11/20/jenis-jenis-jaringan-komputer>)

Berikut ini adalah tabel perbandingan LAN, MAN dan WAN.

Tabel 2.2 Perbandingan LAN, MAN, dan WAN

Jarak antarkomputer	Lokasi/area	Jenis jaringan
1 - 10m	Ruangan	LAN
100 - <1km	Gedung perkantoran	
1-10km	Kota	MAN
>10 - <100 km	Kabupaten, provinsi	
≥ 100 km	Negara	WAN
≥ 1000 km	Benua	
≥ 10000 km	Planet	Internet

Sumber: (Syahrizal, 2007)

2.6 Topologi Jaringan

Topologi menggambarkan metode yang digunakan untuk melakukan pengabelan secara fisik dari suatu jaringan. Topologi jaringan adalah susunan atau pemetaan interkoneksi antara *node*, dari suatu jaringan, baik secara fisik (*real*) dan logis (*virtual*). (Sopandi, 2008)

Jaringan komputer memiliki banyak jenis topologi, namun ada 4 (empat) jenis topologi yang umum digunakan adalah Topologi *Bus (Linier)*, Topologi *Ring*, Topologi *Tree* dan Topologi *Star*. Namun karena penulis hanya menggunakan Topologi *Star*, maka yang dijelaskan hanya topologi tersebut.

2.6.1 Topologi *Star*

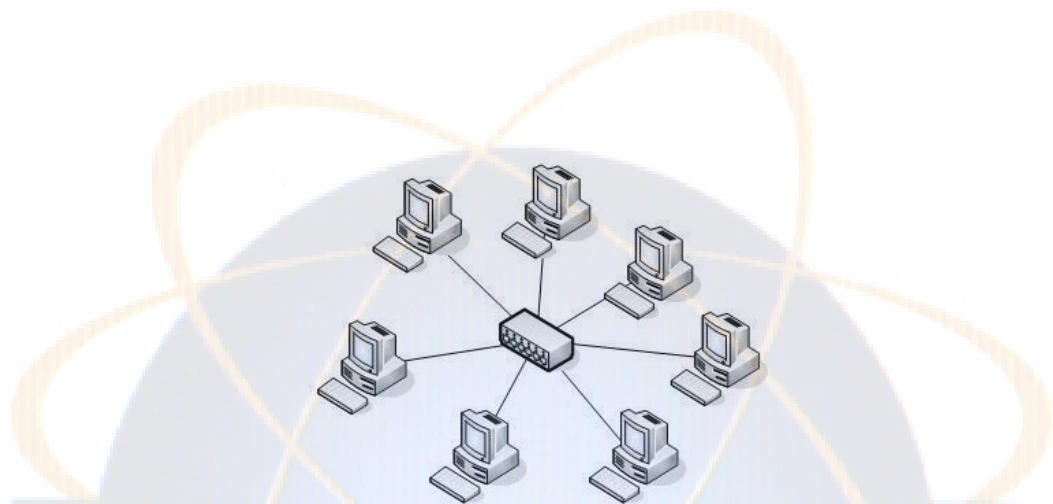
Seperti namanya topologi ini berbentuk seperti bintang, masing-masing komputer dalam jaringan terhubung dengan pusat (sentral). Terminal pusat tersebut bertindak sebagai pengatur dan pengendali semua komunikasi data. Terminal inilah yang menyediakan jalur komunikasi khusus pada komputer yang akan berkomunikasi, yang berupa hub. Hub merupakan alat yang menyediakan lokasi terpusat, di mana semua kabel UTP terpasang.

Kelemahan topologi *star*, di antaranya:

- a. Kesulitan perawatan jika ukuran besar.
- b. Jarak terbatas, sering terjadi tabrakan pada lalu lintas padat.

Sedangkan kelebihan dari topologi *star*, yakni:

- a. Keamanan data tinggi.
- b. Kemudahan pemasangan kabel dan penanganan masalah.
- c. Penambahan terminal yang mudah.



Gambar 2.10 Topologi *star*

Sumber: (<http://banghillman.blogspot.com/2010/06/konsep-dasar-teknologi-informasi>)

2.7 Perangkat Jaringan

Perangkat Keras (*hardware*) adalah semua bagian fisik komputer, dan dibedakan dengan data yang berada di dalamnya atau yang beroperasi di dalamnya, dan dibedakan dengan perangkat lunak (*software*) yang menyediakan instruksi untuk perangkat keras dalam menyelesaikan tugasnya (Sopandi, 2008).

Secara umum, perangkat keras yang dibutuhkan untuk membangun sebuah jaringan komputer yaitu : Komputer, *Network Internet Card* (NIC), *Switch*, dan segala sesuatu yang berhubungan dengan koneksi jaringan seperti: *Bridges*, *Router* dan lainnya yang dibutuhkan untuk proses transformasi data di dalam jaringan.

2.7.1 *Personal Computer* (PC)

Personal computer atau PC merupakan perangkat utama dalam suatu jaringan komputer. PC ini lah yang akan bekerja mengirim dan mengakses data dalam jaringan. Kemampuan suatu PC sangat menentukan sekali unjuk kerja dari jaringan. Semakin

tinggi kemampuan suatu PC maka akses yang dilakukan pun akan semakin cepat (Sugeng, 2006).

2.7.2 Network Internet Card (NIC)

Kartu jaringan merupakan perangkat yang menyediakan media untuk menghubungkan antarkomputer. Kebanyakan kartu jaringan adalah kartu internal, yaitu kartu jaringan yang dipasang pada slot ekspansi di dalam komputer. Kartu Jaringan umumnya telah menyediakan port koneksi untuk kabel koaksial ataupun kabel *twisted pair* (Sugeng, 2006).

2.7.3 Switch

Switch adalah sebuah alat jaringan yang melakukan *bridging* transparan (penghubung segementasi banyak jaringan dengan *forwarding* berdasarkan *MAC Address*).

Switch jaringan dapat digunakan sebagai penghubung komputer atau *router* pada satu area yang terbatas, *switch* juga bekerja pada lapisan *data link*, cara kerja *switch* hampir sama seperti *bridge*, tetapi *switch* memiliki sejumlah *port* sehingga sering dinamakan *multiport bridge* (Sugeng, 2006).

2.7.4 Router

Router adalah sebuah alat jaringan komputer yang mengirimkan paket data melalui sebuah jaringan atau internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai

routing. Proses *routing* terjadi pada lapisan 3 (*Network Layer* seperti *Internet Protocol*) dari tujuh lapisan OSI.

Router berfungsi sebagai penghubung antar dua atau lebih jaringan untuk meneruskan data dari satu jaringan ke jaringan lainnya. *Router* berbeda dengan *switch*. *Switch* merupakan penghubung beberapa alat untuk membentuk suatu *Local Area Network* (LAN). (Sugeng, 2006)

Router sangat banyak digunakan dalam jaringan berbasis teknologi protokol TCP/IP, dan *router* jenis itu disebut juga dengan *IP Router*. Internet merupakan contoh utama dari sebuah jaringan yang memiliki banyak *router* IP. *Router* dapat digunakan untuk menghubungkan banyak jaringan kecil ke sebuah jaringan yang lebih besar, yang disebut dengan *internetwork*, atau untuk membagi sebuah jaringan besar ke dalam beberapa *subnetwork* untuk meningkatkan kinerja dan juga mempermudah manajemennya.

Router juga kadang digunakan untuk mengoneksikan dua buah jaringan yang menggunakan media yang berbeda, seperti halnya *router wireless* yang pada umumnya selain ia dapat menghubungkan komputer dengan menggunakan radio, ia juga mendukung penghubungan komputer dengan kabel UTP.

2.7.5 Modem Router

Modem Router adalah piranti *router* yang sudah *include* dengan modem HSDPA. Teknologi ini mungkin saja sudah lama

dikembangkan, namun di Indonesia sendiri baru diperkenalkan awal tahun 2011. (<http://tekno.kompas.com>)

Salah satu contoh paket *bundling modem router* dari Indosat IM2, salah satu ISP di Indonesia, mengeluarkan produk *Modem Router ZTE MF608* yang memiliki fitur yang lumayan banyak, dikarenakan *modem* tersebut sudah terdapat 4 *port Ethernet*, *Wifi*, *RUIM enable*, dan lain sebagainya.



Gambar 2.11 Modem Router ZTE MF608
Sumber: (<http://www.indosatm2.com>)

2.7.6 3G Wireless N Router

Perangkat keras ini sebenarnya sama dengan *modem router*, tetapi perbedaan yang mendasar adalah bahwa perangkat ini menggunakan *modem usb* tambahan sebagai koneksi internet. Sebagai contohnya ialah TP-LINK MR3420, yang mempunyai slot *usb* dan *port ADSL*. *Port USB* ini dapat digunakan untuk *modem usb* GSM maupun CDMA.

Kendala dari perangkat keras ini adalah bahwa tidak semua *modem usb* akan terdeteksi, jadi *firmware* dari perangkat ini harus

di-update terlebih dahulu. *Firmware* terbaru dapat di-download di <http://www.tp-link.com/support/download.asp>.



Gambar 2.12 TP-Link TL-MR3420
Sumber: (www.tp-link.com)

2.8 Model OSI Layer

Supaya komputer dapat mengirimkan informasi ke komputer lain, dan dapat menerima dan mengerti informasi, harus ada aturan atau standar untuk proses komunikasi tersebut. Standar ini meyakinkan kita bahwa beberapa jenis produk dan perangkat dapat berkomunikasi dengan perangkat lain yang berbeda melewati beberapa jaringan. Pembakuan standard ini disebut "MODEL". (Sutanta, 2005)

Referensi model jaringan yang sering digunakan adalah *Open System Interconnection* (OSI), yang diperkenalkan oleh *International Standard Organisation* (ISO). Referensi model jaringan OSI membagi jaringan komputer menjadi 7 lapisan, setiap lapisan hanya mengatur beberapa layanan dan protokol yang dapat bekerja pada tiap lapisan agar mempermudah pembuatan program untuk jaringan sehingga rapih dan tidak berantakan serta sulit didefinisikan. Berikut ini adalah penjelasan dari setiap OSI layer.

2.8.1 Layer 1 – *Physical Layer*

Lapisan paling bawah dalam model OSI adalah *physical layer*. Lapisan ini mengatur bagaimana sinkronisasi pengiriman dan penerimaan data, spesifikasi mekanik, elektrik dan *interface* antarterminal.

Lapisan ini juga berfungsi juga untuk mendefinisikan media transmisi jaringan, arsitek jaringan, topologi jaringan dan pengabelan. Selain itu, *level* ini juga mendefinisikan bagaimana *Network Interface Card* (NIC) dapat berinteraksi baik dengan media kabel maupun radio (Sutanta, 2005).

2.8.2 Layer 2 – *Data Link Layer*

Diatas *Physical layer* adalah *Data-link layer*, lapisan dimana data dipersiapkan untuk dikirimkan melalui jaringan, pada lapisan ini paket data di kapsulasi dalam sebuah *frame* (*bundle* dari data biner) sebelum dikirimkan. Protokol pada lapisan ini membantu dalam hal pengalamatan (*addressing*) dan pendeteksian kesalahan dari data yang dikirimkan.

Data-link layer terdiri dari dua *sublayer* yaitu; *sublayer Logical Link Control* (LLC) dan *sublayer Media Access Control* (MAC). *Sublayer* LLC adalah antarmuka antara protokol *network layer* dengan metode pengaksesan media misalnya Ethernet atau Token Ring. *Sublayer* MAC menangani koneksi ke media fisik seperti *twisted-pair* atau pengkabelan koaksial (Sutanta, 2005).

2.8.3 Layer 3 – Network Layer

Berikutnya adalah *Network layer* yang bertanggung jawab dalam hal *routing* dari paket-paket data yang didasarkan pada *logical address* dari paket-paket data tersebut. *Network layer* memotong-motong data dan menyusunnya kembali jika diperlukan, ia mengirim paket-paket data dari sumber ke tujuan (Sutanta, 2005).

2.8.4 Layer 4 – Transport Layer

Diatas *Network layer* ada *Transport layer*, lapisan ini menjamin diterimanya paket data yang dikirim. *Transport layer* juga dapat membentuk sebuah sambungan dan mengirim *acknowledgment* ketika paket data diterima (Sutanta, 2005).

2.8.5 Layer 5 – Session Layer

Berikutnya adalah *Session layer*, lapisan ini bertugas untuk mengontrol “dialog” selama komunikasi berlangsung, lapisan ini bertanggung jawab dalam hal bagaimana membentuk sambungan, bagaimana menggunakan sambungan tersebut, dan bagaimana memutuskan sambungan yang terbentuk setelah sebuah sesi komunikasi selesai. *Session layer* juga menambahkan *control header* pada paket data selama pertukaran data terjadi (Sutanta, 2005).

2.8.6 Layer 6 – Presentation Layer

Presentation layer adalah lapisan yang berada dibawah *application layer* dan diatas *session layer*. Lapisan ini menambahkan struktur pada paket data yang akan dikirimkan. Tugas utama lapisan

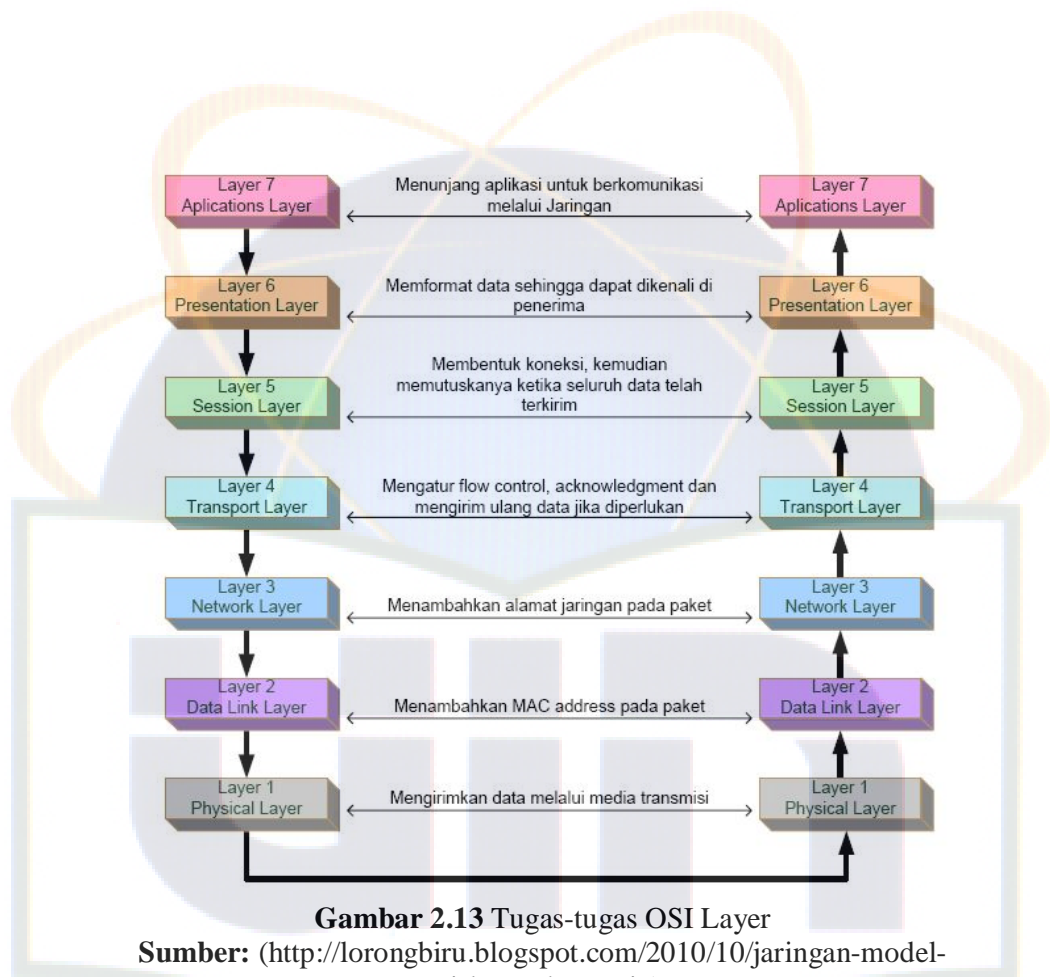
ini adalah untuk meyakinkan bahwa data atau informasi terkirim dengan bahasa atau *syntax* yang dapat dipahami oleh *host* yang dituju. Protokol pada *presentation layer* dapat menerjemahkan data kedalam bahasa atau *syntax* yang dapat dimengerti dan kemudian mengkompresi atau mengenkripsi data sebelum menyampaikan data ke *session layer* (Sutanta, 2005).

2.8.7 Layer 7 – Application Layer

Lapisan paling tinggi dari model OSI adalah *application layer*, seluruh layer dibawahnya bekerja untuk lapisan ini, tugas dari *application layer* adalah mengatur komunikasi antar aplikasi.

Pembentukan paket dimulai dari layer teratas model OSI. *Application layer* mengirimkan data ke *presentation layer*, di *presentation layer* data ditambahkan *header* dan atau *tailer* kemudian dikirim ke lapisan dibawahnya, pada lapisan dibawahnya pun demikian, data ditambahkan *header* dan atau *tailer* kemudian dikirimkan ke layer dibawahnya lagi, terus demikian sampai ke *physical layer*. Di *physical layer* data dikirimkan melalui media transmisi ke *host* tujuan (Sutanta, 2005).

Proses pengiriman paket dari *layer* ke *layer* ini disebut dengan “*peer-layer communication*”. Berikut ini pada gambar 2.13 adalah ilustrasi dari *peer-layer communication*.



2.9 Transmission Control Protocol/Internet Protocol (TCP/IP)

TCP/IP adalah sekumpulan protokol yang didesain untuk melakukan fungsi-fungsi komunikasi data pada Internet. TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dari komunikasi data. Protokol ini merupakan komunikasi utama dalam internet serta intranet. Protokol ini memungkinkan sistem apapun yang terhubung kedalamnya bisa berkomunikasi dengan sistem lain tanpa harus mempedulikan bagaimana *remote system* yang lain tersebut bekerja. Protokol ini dikembangkan pada tahun 1969 oleh DARPA (*Defence Advanced Research Project Agency*)

mendanai riset dan pembuatan paket *switching* eksperimental yang diberi nama ARPANET (Sopandi, 2008).

TCP/IP menggunakan model *client-server* dalam berkomunikasi, dimana komputer *user (client)* meminta kepada komputer lain dan akan disediakan *service* tersebut oleh komputer *server*. Beberapa layanan dan *utility* dari TCP/IP meliputi sebagai berikut.

2.9.1 HyperText Transfer Protocol (HTTP)

HTTP adalah protokol yang dipakai untuk mayoritas komunikasi *World Wide Web (WWW)*. *Windows* menghadirkan *Internet Explorer* sebagai client HTTP dan *Internet Information Service (IIS)* sebagai server HTTP. Pada umumnya *port* HTTPS adalah 80 (Sopandi, 2008).

2.9.2 Hypertext Transfer Protocol Secure (HTTPS)

HTTPS adalah versi aman dari HTTP, protokol komunikasi dari *World Wide Web*. Ditemukan oleh *Netscape Communications Corporation* untuk menyediakan autentikasi dan komunikasi tersandi dan penggunaan dalam komersi elektrik. Selain menggunakan komunikasi plain text, HTTPS menyandikan data sesi menggunakan protokol SSL (*Secure Socket layer*) atau protokol TLS (*Transport Layer Security*). Kedua protokol tersebut memberikan perlindungan yang memadai dari serangan *eavesdroppers*, dan *man in the middle attacks*. Pada umumnya *port* HTTPS adalah 443 (Sopandi, 2008).

2.9.3 *Secure Socket Layer (SSL)*

SSL adalah protokol yang digunakan untuk *browsing website* secara aman. SSL bertindak sebagai protokol yang mengamankan komunikasi antara *client* dan *server*. Protokol ini memfasilitasi penggunaan enkripsi untuk data yang rahasia dan membantu integritas informasi yang dipertukarkan antara *website* dan *web browser*. Pada umumnya *port* SSL sama dengan *port* HTTPS yaitu 443 (Sopandi, 2008).

2.9.4 *Domain Name Service (DNS)*

DNS merupakan seperangkat protokol dan layanan pada suatu jaringan TCP/IP yang membolehkan para pemakai jaringan untuk mempergunakan nama-nama hierarki yang sudah dikenal ketika meletakkan *host* ketimbang harus mengingat dan memakai alamat IP-nya. DNS sangat banyak dipakai di internet dan pada kebanyakan perusahaan pribadi dewasa ini. Saat memakai *web browser*, aplikasi *telnet*, utiliti FTP, atau utiliti TCP/IP mirip lainnya di Internet, maka *user* mungkin sedang memakai sebuah *server* DNS. Pada umumnya *port* DNS adalah 53 (Sopandi, 2008).

1. *Google public DNS*

Google public DNS adalah alamat DNS yang disediakan oleh *google* secara gratis. Ketika menggunakan *google public* DNS pada PC, otomatis alamat DNS dari ISP yang dipakai akan digantikan dengan alamat *google public*

DNS. Alamat *IP address* dari *google public* DNS adalah 8.8.8.8 dan 8.8.4.4.

Alamat DNS tersebut dapat diakses di seluruh dunia. Menjanjikan kecepatan *browser* lebih cepat ke *server google*. Tetapi *google* tidak menceritakan dari mana akses *server* DNS mereka yang begitu dekat dengan pusat komputer *google*.

2.9.5 Simple Network Management Protocol (SNMP)

SNMP memungkinkan seorang *user* untuk mengelola *node* jaringan seperti *server*, *workstation*, *router*, *bridge*, dan *switch* dari *host* sentral. SNMP dapat dipakai untuk mengkonfigurasi *device* yang jauh, memantau untuk kerja jaringan, mendeteksi kesalahan jaringan atau akses yang tidak cocok, dan mengaudit pemakaian jaringan. (Sopandi, 2008)

2.10 IP Address

IP Address (Alamat IP) adalah identitas khusus yang digunakan untuk memberikan tanda atau alamat pada sebuah paket data atau pada suatu sistem komputer. Konsep dasar pengalamatan (*IP Address*) di internet adalah awalan (*prefix*) pada *IP Address* dapat digunakan sebagai dasar pengambilan keputusan dalam pemilihan rute paket data ke alamat tujuan. Sebelum memasuki aspek-aspek lebih jauh tentang *IP Address*,

penting untuk mengerti lebih dahulu beberapa hal fundamental dari IP Address itu sendiri (Sopandi, 2008).

2.10.1 Format Alamat IPv4

Untuk IPv4 menggunakan bilangan 32 Bit yang dipisahkan oleh tanda pemisah berupa tanda titik pada setiap bitnya. Tiap 8 bit ini disebut sebagai oktet. Pengalamatan IP berupa nomor 32 bit tersebut terdiri dari *network ID* dan *host ID*. *Network ID (NetID)* menunjukkan nomor jaringan sedangkan *host ID* mengidentifikasi host dalam satu jaringan. (Sopandi, 2008). Contoh bentuk IP Address sebagai berikut.

Desimal	167	205	206	100
Biner	10100111	11001101	11001110	01100100

Gambar 2.14 Format IPv4

Sumber: (<http://1tkj-smkn5makassar.blogspot.com/2010/11/ip-address.html>)

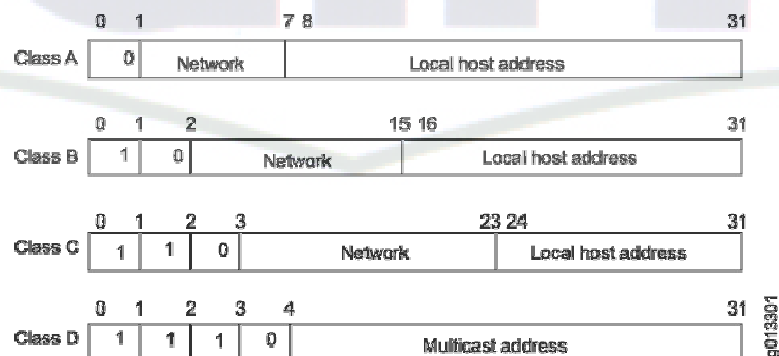
2.10.2 Kelas Alamat IP

Jika dilihat dari bentuknya, IP address terdiri dari 4 buah bilangan oktat (8 bit). Nilai terbesar dari bilangan biner 8 bit yaitu $255 = (2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0)$. Jumlah keseluruhan IP Address adalah sebanyak $255 \times 255 \times 255 \times 255$.

IP Address sebanyak inilah yang harus dibagikan keseluruhan pengguna jaringan internet di seluruh dunia. Untuk mempermudah membagikannya, IP Address dikelompokan dalam kelas-kelas, hal ini dilakukan untuk memudahkan pendistribusian pendaftaran IP

Address (Sopandi, 2008). Kelas-kelas IP *address* yang umum digunakan terdiri dari:

- a. Kelas A: 1–126, dengan batasan 1.0.0.0– 126.255.255.254
 Dengan *Broadcast Address* 126.255.255.255
Subnet Mask-nya : 255.0.0.0
- b. Kelas B: 128–191, dengan batasan 128.0.0.0– 191.255.255.254
 Dengan *Broadcast Address* 191.255.255.255
Subnet Mask-nya : 255.255.0.0
- c. Kelas C: 192–223, dengan batasan 192.0.0.0– 223.255.255.254
 Dengan *Broadcast Address* 223.255.255.255
Subnet Mask-nya : 255.255.255.0



Gambar 2.15 Kelas-kelas IP *address*

Sumber: (http://www.juniper.net/techpubs/en_US/junose10.3/information-products/topic-collections/swconfig-ip-ipv6/id-68448.html)

Dalam pemberian IP *Address* juga harus perdasarkan aturan-aturan dasar, seperti *Network ID* dan *Host ID* tidak boleh bernilai 0 (nol) atau 255 (semua bit di *set* 1). *Network ID* juga tidak boleh bernilai 127, karena secara *default* sudah digunakan sebagai IP Lokal (*loopback IP Address*).

Tabel 2.3 Rangkuman Kelas di IPv4

Kelas	<i>Network ID</i>	<i>Host ID</i>	<i>Default Subnet Mask</i>
A	xxx.0.0.1	xxx.255.255.254	255.0.0.0
B	xxx.xxx.0.1	xxx.xxx.255.254	255.255.0.0
C	xxx.xxx.xxx.1	xxx.xxx.xxx.254	255.255.255.0

Sumber: (Sopandi, 2008)

2.11 *Subnetting*

Sebuah jaringan dapat dipecahkan menjadi beberapa jaringan baru, proses ini disebut *subnetting*. Tujuan dari *subnetting* yaitu untuk mereduksi trafik jaringan, mengoptimalkan *performance* jaringan, memudahkan dalam manajemen jaringan dan mengefektifkan jaringan yang dibatasi area geografis luas. Melalui *subnetting* sebuah alamat jaringan (*network address*) tunggal dipecah menjadi *subnetwork* atau disingkat *subnet*. Sebagai contoh *network* 192.168.10.0, 168.20.0 dan 192.168.30.0 merupakan *subnet network* tunggal 192.168.8.0. *Subnet address* dibuat dengan meminjamkan *bit* porsi *host* dan menjadikannya sebagai *subnet*. Jumlah *bit* yang dipinjam bervariasi tergantung pada nilai *subnet mask* (Sopandi, 2008).

2.12 *Network Address Translation (NAT)*

Network Address Translation atau yang lebih biasa disebut dengan NAT adalah suatu metode untuk menghubungkan lebih dari satu komputer ke jaringan internet dengan menggunakan satu alamat IP *Public*.

Banyaknya penggunaan metode ini disebabkan karena ketersediaan alamat IP yang terbatas, kebutuhan akan keamanan (*security*), dan kemudahan serta fleksibilitas dalam administrasi jaringan.

Saat ini, protokol IP yang banyak digunakan adalah IP versi 4 (IPv4). Dengan panjang alamat 4 *byte* berarti terdapat $2^{32} = 4.294.967.296$ alamat IP yang tersedia. Jumlah ini secara teoritis adalah jumlah komputer yang dapat langsung koneksi ke internet. Karena keterbatasan inilah sebagian besar ISP (*Internet Service Provider*) hanya akan mengalokasikan satu alamat untuk satu pengguna dan alamat ini bersifat dinamik, dalam arti alamat IP yang diberikan akan berbeda setiap kali user melakukan koneksi ke internet (Nugroho, 2005). Hal ini akan menyulitkan untuk bisnis golongan menengah ke bawah. Di satu sisi mereka membutuhkan banyak komputer yang terkoneksi ke internet, akan tetapi di sisi lain hanya tersedia satu alamat IP yang berarti hanya ada satu komputer yang bisa terkoneksi ke internet. Hal ini bisa diatasi dengan metode NAT. Dengan NAT *gateway* yang dijalankan di salah satu komputer, satu alamat IP tersebut dapat dibagi ke beberapa komputer yang lain dan mereka bisa melakukan koneksi ke internet secara bersamaan.

Dengan NAT, suatu jaringan yang besar dapat dipecah-pecah menjadi jaringan yang lebih kecil. Bagian-bagian kecil tersebut masing-masing memiliki satu alamat IP, sehingga dapat menambahkan atau mengurangi jumlah komputer tanpa mempengaruhi jaringan secara keseluruhan.

Dalam mikrotik, NAT dikenal juga sebagai masquerade yaitu fasilitas *router* untuk meneruskan paket dari IP asal dan atau ke IP tujuan. Terdapat dua jenis NAT, yaitu *Source NAT* (srcnat) dan *Destination NAT* (dstnat).

2.13 Routing

Routing (perutean) merupakan sebuah proses untuk meneruskan paket-paket jaringan dari satu jaringan ke jaringan lainnya melalui sebuah *internetwork*. *Routing* juga dapat merujuk kepada sebuah metode penggabungan beberapa jaringan sehingga paket-paket data dapat hinggap dari satu jaringan ke jaringan selanjutnya (Nugroho, 2005).

Untuk melakukan hal ini, digunakanlah sebuah perangkat jaringan yang disebut sebagai *router*. *Router* tersebut akan menerima paket yang ditujukan ke jaringan di luar jaringan yang pertama, dan akan meneruskan paket yang diterima kepada *router* lainnya hingga sampai kepada tujuannya. Jadi *router* berfungsi sebagai penghubung antar dua atau lebih jaringan untuk meneruskan data dari satu jaringan ke jaringan lainnya.

Jika *routing* yang digunakan adalah statis, maka konfigurasinya harus dilakukan secara manual, *administrator* jaringan harus memasukkan atau menghapus rute statis jika terjadi perubahan topologi. Pada jaringan skala besar, jika tetap menggunakan *routing* statis, maka akan sangat membuang waktu *administrator* jaringan untuk melakukan *update table routing*. Karena itu *routing statis* hanya digunakan untuk jaringan skala

kecil. Sedangkan routing dinamis lebih tepat digunakan di jaringan skala besar. (Lammle, 2004)

2.13.1 Static Route

Suatu *static route* adalah suatu mekanisme *routing* yang tergantung dengan *routing table* dengan konfigurasi manual. Dalam skala jaringan yang kecil yang mungkin terdiri dari dua atau tiga *router* saja, pemakaian *static route* lebih umum dipakai. *Static route* haruslah di konfigurasi secara manual dan di-maintenance secara terpisah karena tidak melakukan pertukaran informasi *routing table* secara dinamis dengan *router-router* lainnya.

Suatu *static route* akan berfungsi sempurna jika *routing table* berisi suatu perutean untuk setiap jaringan di dalam *internetwork* yang mana dikonfigurasi secara manual oleh *administrator* jaringan. Setiap host pada jaringan harus dikonfigurasi untuk mengarah kepada *default router* atau *default gateway* agar cocok dengan IP address dari *interface local router*, dimana *router* memeriksa *routing table* dan menentukan *route* yang mana untuk meneruskan paket. (Lammle, 2004)

2.14 Microsoft Visio

Microsoft Visio (atau sering disebut Visio) adalah sebuah program aplikasi komputer yang sering digunakan untuk membuat diagram, diagram alir (flowchart), brainstorm, dan skema jaringan yang dirilis oleh

Microsoft Corporation. Aplikasi ini menggunakan grafik vektor untuk membuat diagram-diagramnya.

Visio aslinya bukanlah buatan Microsoft Corporation, melainkan buatan Visio Corporation, yang diakuisisi oleh Microsoft pada tahun 2000. Versi yang telah menggunakan nama Microsoft Visio adalah Visio 2002, Visio 2003, dan Visio 2007 yang merupakan versi terbaru. Visio 2007 Standard dan Professional menawarkan antarmuka pengguna yang sama, tapi seri Professional menawarkan lebih banyak pilihan template untuk pembuatan diagram yang lebih lanjut dan juga penataan letak (layout). Selain itu, edisi Professional juga memudahkan pengguna untuk mengoneksikan diagram-diagram buatan mereka terhadap beberapa sumber data dan juga menampilkan informasi secara visual dengan menggunakan grafik. (http://id.wikipedia.org/wiki/Microsoft_Visio)

2.15 VMware Workstation

Virtual mesin hampir selalu menjadi tempat bagi instalasi sistem operasi. Sistem operasi ini biasanya disebut dengan *Guest Operating System*. Instruksi-intruksi untuk sebuah virtual mesin biasanya dilewatkan langsung ke *hardware* fisik yang memungkinkan lingkungan virtual untuk beroperasi lebih cepat dan lebih efisien daripada sekedar emulasi, walaupun lebih kompleks intruksi-intruksi harus ditangkap dan diterjemahkan supaya memastikan kompatibilitas yang tepat dan memastikan proses abstraksi dengan *hardware* fisik.

VMware Workstation adalah salah satu jenis aplikasi yang membuat mesin virtual. *Software* ini diperkenalkan oleh perusahaan VMware pada tahun 1999 dengan menggunakan *virtual technology*, dimana sejumlah *virtual machine* dengan sistem operasinya sendiri-sendiri dapat sekaligus bekerja pada satu hardware komputer saja. Setiap *virtual machine* yang dibuat oleh VMware tersebut bekerja seolah-olah merupakan komputer biasa yang berdiri sendiri. (Wijaya, 2008)

2.16 Mikrotik

Mikrotik RouterOS merupakan sistem operasi jaringan (*network operating system*) yang banyak digunakan oleh *Internet Service Provider* untuk keperluan *firewall* atau *router* yang handal yang dilengkapi dengan berbagai fitur dan *tool*, baik untuk jaringan kabel maupun jaringan *wireless* (Saputro & Kustanto, 2008: 56).

Sebagian orang beranggapan bahwa *router* yang baik hanyalah *router* yang bermerek. Padahal, *router* sebenarnya juga bisa dibuat dengan menggunakan komputer, dan menginstal perangkat lunak yang sesuai. Salah satu perangkat lunak yang bisa difungsikan menjadi sebuah *router* adalah Mikrotik.

Mikrotik dibuat oleh MikroTiks sebuah perusahaan di kota Riga, Latvia. Dengan nama merek dagang Mikrotik mulai didirikan tahun 1995 yang pada awalnya ditujukan untuk perusahaan jasa layanan Internet (PJI) atau *Internet Service Provider* (ISP) yang melayani pelanggannya

menggunakan teknologi nirkabel atau *wireless* (Saputro & Kustanto, 2008).

Mikrotik dapat digunakan dalam 2 tipe, yaitu dalam bentuk perangkat keras dan perangkat lunak, dimana keduanya terpasang secara sinkron agar dapat bekerja dengan baik. Dalam bentuk perangkat keras, mikrotik biasanya sudah diinstalasi (*built in*) pada suatu *board* tertentu, sedangkan dalam bentuk perangkat lunak mikrotik merupakan satu *distro linux* yang memang dikhususkan untuk fungsi *router*. (Herlambang & Catur, 2008)

MikroTik *RouterOS* adalah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer menjadi *router network* yang handal, mencakup berbagai fitur yang dibuat untuk *IP network* dan jaringan *wireless*, cocok digunakan oleh ISP dan *provider hotspot*.

Sebagai perangkat lunak *router*, cukup banyak fungsi yang bisa dilakukan dengan Mikrotik, mulai dari *Quality of Services* (pengaturan *bandwidth*), *firewall*, *hotspot gateway*, *web proxy*, *DNS cache*, hingga penggunaan *Virtual Private Network* (VPN). Fasilitas pemantauan seperti *watchdog* dan *netwatch* juga tersedia. Salah satu keunggulan lainnya adalah adanya aplikasi pengaturan yang tidak lagi hanya berbasis teks, tetapi juga berbasis grafis. (Herlambang & Catur, 2008)

Mikrotik *RouterOS* hadir dalam berbagai level. Tiap level memiliki kemampuannya masing-masing, mulai dari level 3, hingga level 6. Secara singkat, level 3 digunakan untuk *router* berinterface ethernet, level 4 untuk

wireless client atau serial interface, level 5 untuk wireless AP, dan level 6 tidak mempunyai limitasi apapun. Untuk aplikasi hotspot, bisa digunakan level 4 (200 pengguna), level 5 (500 pengguna) dan level 6 (tidak terbatas). (<http://www.mikrotik.co.id>)



Gambar 2.16 Mikrotik RouterBoard 750

Sumber: (<http://www.mikrotik.co.id>)

Karena berbasis Linux, Mikrotik mengadopsi pula tampilan hitam putih dalam mode teks (*shell*). Berikut ini adalah beberapa contoh menu navigasi mikrotik yang dijelaskan pada tabel 2.4.

Tabel 2.4 Menu Navigasi Mikrotik

Perintah	Keterangan.
command [enter]	Menjalankan perintah.
?	Melihat daftar perintah beserta keterangannya.
Tab	1. Melengkapi baris perintah. 2. Melihat daftar perintah selanjutnya yang terkait dengan perintah tersebut.
/	Berpindah pada level teratas (root).
/command	Menjalankan perintah yang terdapat pada level atas.
..	Naik satu level di atasnya.
.. ..	Naik dua level di atasnya.

↑ atau ↓	Melihat history dari perintah yang telah dijalankan.
Ctrl + i	Sama dengan fungsi tab.
Ctrl + p	Sama dengan fungsi ↑.
Ctrl + c	Menghentikan proses.
Ctrl + m	Sama dengan fungsi tombol Enter
Ctrl + n	Menghapus baris perintah.
Ctrl + d	Logout/keluar dari sistem.

Sumber: (Herlambang & Catur, 2008)

2.17 Winbox

Winbox adalah sebuah utiliti yang digunakan untuk melakukan remote ke *server* mikrotik dalam mode GUI. Jika untuk mengkonfigurasi mikrotik dalam *text mode* melalui PC itu sendiri, maka untuk mode GUI yang menggunakan winbox ini kita mengkonfigurasi mikrotik melalui komputer *client*.

Mengkonfigurasi mikrotik melalui winbox ini lebih banyak digunakan karena selain penggunaannya yang mudah kita juga tidak harus menghafal perintah-perintah *console*.

Untuk mendapatkan winbox dapat mengunduhnya atau bisa juga mendapatkan di mikrotik. Yaitu dengan cara buka browser komputer klien yang telah tersambung dengan mikrotik, lalu tuliskan di address bar <http://ipaddressrouter/winbox/winbox.exe>. Atau bisa mengunduhnya di website <http://www.mikrotik.co.id/download.php>.

2.18 *Monitoring Jaringan*

Dalam suatu jaringan komputer, terdapat berbagai komputer yang terhubung. Setiap komputer memiliki *resource*, masing-masing dan saling berbagi *resource* satu sama lain. Kadang saat jaringan menjadi sangat lambat, administrator harus mencari tahu apakah penyebab begitu lambatnya jaringan tersebut. Itulah sebabnya administrator membutuhkan untuk melakukan *monitoring* jaringan, supaya lalu lintas data yang ada dalam jaringan dapat terpantau dengan baik. Dalam *memonitoring* jaringan dapat menggunakan *software* khusus untuk *monitoring* jaringan.

2.18.1 *Torch*

Torch adalah sebuah *tool* pada mikrotik yang digunakan untuk *memonitoring* trafik yang berjalan pada *interface* (antarmuka) secara realtime (Herlambang & Catur, 2008).

Tool ini secara umum dapat mengetahui nama protokol, asal alamat IP, tujuan alamat IP, *port* yang digunakan baik asal maupun tujuan *port* dan dapat juga mengetahui data *rate* dari aktifitas penggunaan *bandwidth*.

2.18.2 *Interface list*

Tool ini hampir sama dengan *torch* yang memuat informasi *traffic* pada suatu *interface* secara *realtime*. Yang membedakannya ialah *tool* ini hanya memuat informasi tentang jumlah maupun besar paket data yang melewati salah satu *interface* di mikrotik.

2.18.3 *Speedtest.net*

Speedtest.net adalah sebuah *tools online* untuk mengukur performa dari suatu koneksi *broadband*. Pada akhir dari tiap tes, akan ditampilkan nilai kecepatan *bandwith download* dan *upload*. Tes dapat dilakukan sepenuhnya di dalam *browser* melalui HTTP, dengan catatan *browser* tersebut sudah terinstall Adobe Flash 10 atau versi terbaru.

2.18.4 *WebProxy log*

WebProxy Log package terdiri dari *WebProxy Log* dan *WebProxy Log Catcher* yang ditujukan untuk menerima dan mengimport data dari *log web-proxy* pada mikrotik, lalu membuat laporan dari data tersebut. Aplikasi ini bukanlah aplikasi resmi yang dibuat atau didukung oleh perusahaan mikrotik. Aplikasi ini bersifat *freeware* dan dibuat untuk digunakan dengan *Windows*.

2.19 Studi Literatur

Studi literatur adalah segala usaha yang dilakukan oleh peneliti untuk menghimpun informasi yang relevan dengan topik atau masalah yang akan atau sedang diteliti. Studi literatur merupakan langkah yang penting sekali dalam metode ilmiah untuk mencari sumber data sekunder yang akan mendukung penelitian dan untuk mengetahui sampai ke mana ilmu yang berhubungan dengan penelitian telah berkembang, sampai kemana terdapat kesimpulan dan degeneralisasi yang pernah dibuat.

Studi literatur dapat diartikan sebagai suatu langkah untuk memperoleh informasi dari penelitian terdahulu yang harus dikerjakan, tanpa memperdulikan apakah sebuah penelitian menggunakan data primer atau data sekunder, apakah penelitian tersebut menggunakan penelitian lapangan ataupun laboratorium atau didalam museum.

Dalam tugas akhir ini, penulis menemukan 3 buah studi literatur yang akan digunakan sebagai pembandingan, yaitu antara lain:

1. *Koneksi Internet Ganda dengan Load Balancing Menggunakan Unix Mikrotik pada PT. Marina Buana Asia* oleh Alam Dari Hendarto, Hendarsyah Febryan, dan Dimas Ganjar Romadhan tahun 2008. Sistem ini dibuat untuk untuk menyelesaikan masalah kecepatan akses internet yang telah menggunakan dua koneksi internet. Pada penelitian ini, metode *load balancing* yang digunakan ialah *Simple Queue*, yaitu pemisahan *queue* untuk trafik internet internasional dan trafik internet lokal. Alur kerjanya adalah membuat *routing-mark* dari *dst-address-list=nice*, yaitu paket paket dari *client* yang menuju *IP address* yang terdaftar pada *address-list* “nice”. Jika ada paket yang tidak diproses oleh *routing-mark* “iix” maka otomatis melewati *primary gateway* dan itu bisa dibilang koneksi internasional, karena untuk paket IIX diproses oleh *mangle* dan *rules routing-mark*.
2. *Perancangan, Implementasi Perouter, Firewall, Gateway dan Load Balancing di Mikrotik* oleh Ilham Budi Wijaya tahun 2009. Pada penelitian ini penulis memanfaatkan teknologi *PC Router*, yaitu PC

yang di-*install* dengan Mikrotik RouterOS. Pemanfaatan teknologi PC Router dirasa akan lebih efisien dibanding dengan penggunaan Router yang harganya relatif mahal. Di penelitian ini penulis memanfaatkan banyak fitur dari mikrotik dalam membangun sebuah jaringan. Metode *load balancing* yang dipakai dalam penelitian ini adalah *static route* dengan *address list*, yaitu pemisahan jalur trafik yang dipakai berdasarkan pengelompokan IP address dari *client*.

3. *Implementasi Load balancing dengan Metode Per Connection Classifier (PCC) untuk Optimasi Penggunaan Aplikasi Realtime* oleh Maulana Yusuf pada tahun 2010. Penelitian ini dilakukan untuk mengoptimalkan penggunaan aplikasi *realtime* agar tidak terjadi diskoneksi karena proses *load balancing* pada sistem yang berjalan.

Pada penelitian ini penulis memanfaatkan teknologi PC Router. Metode *load balancing* yang dipakai adalah menggunakan metode *Per Connection Classifier* (PCC), yaitu mengelompokkan trafik koneksi yang melalui atau keluar masuk *router* menjadi beberapa kelompok. Pengelompokan itu berdasarkan *src-address*, *dst-address*, *src-port* dan atau *dst-port*. Setelah itu penulis membandingkan sistem *load balancing* yang telah berjalan dengan sistem *load balancing* PCC yang akan dibangun. Diharapkan setelah menggunakan metode PCC, akan terjadi optimalisasi terhadap aplikasi *real time*.

BAB III

METODE PENELITIAN

3.1 Metode Pengumpulan Data

Metode pengumpulan data merupakan tahapan proses penelitian dalam pengembangan sistem. Penulis membutuhkan data yang tepat agar penelitian berlangsung sesuai dengan perumusan masalah yang sudah ditentukan. Metode pengumpulan data yang penulis lakukan yaitu studi pustaka, observasi dan studi literatur.

3.1.1 Studi Pustaka

Dalam tahapan ini yaitu melakukan pengumpulan bahan-bahan yang berkaitan dengan judul skripsi, melalui membaca buku-buku dari perpustakaan dan mencari referensi artikel serta *ebook* dari *internet*. Secara lengkapnya judul buku dan *website* dapat dilihat pada daftar pustaka.

3.1.2 Studi Lapangan

Penulis melakukan pengamatan langsung ke lapangan (observasi). Tahap ini diperlukan dalam penerapan sistem yang akan dibangun, dengan tujuan untuk memperoleh informasi mengenai sistem yang akan penulis kembangkan dan dengan ketersediaan alat jaringan yang telah ada.

Tempat pelaksanaan penelitian yaitu pada SMK PGRI BEKASI, Jalan Makrik, Rawalumbu, Bekasi Timur. Dan waktu pelaksanaan yaitu selama bulan Januari hingga bulan Maret 2011.

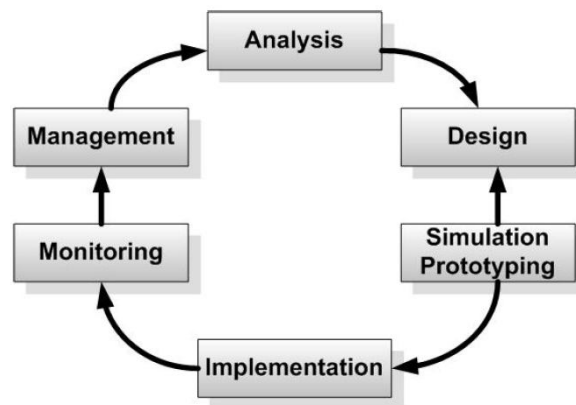
3.1.3 Studi Literatur

Pada tahap ini penulis dalam melakukan perbandingan dan sebagai acuan pembelajaran dan simulasi ujian. Maka perlu dilakukan pengamatan terhadap penelitian sejenis yang telah dilakukan. Lebih jelasnya dapat dilihat pada sub bab 2.19.

3.2 Metode Pengembangan Sistem

Penulis melakukan pendekatan pengembangan sistem dengan menggunakan metode *Network Development Life Cycle* (NDLC) untuk mengimplementasikan konsep *load balancing* pada sebuah jaringan yang mempunyai perumusan masalah yang telah dibahas di bab 1.

NDLC mempunyai beberapa alur kerja dalam mengembangkan suatu sistem jaringan, yang dijelaskan pada gambar 3.1.



Gambar 3.1 Metodologi penelitian NDLC
(Sumber: Deris Stiawan, 2009)

Berikut adalah penjelasan dari masing-masing tahap dalam *Network Development Life Cycle* (NDLC):

3.2.1 Analysis

Tahap awal ini dilakukan analisa kebutuhan, analisa permasalahan yang muncul, analisa keinginan *user*, dan analisa topologi jaringan yang sudah ada saat ini (Deris Stiawan, 2009).

Bisa dibilang pada tahap ini adalah pengumpulan data-data yang dibutuhkan untuk mengetahui perumusan masalah dan cara menyelesaikan masalah tersebut. Dalam hal ini yaitu mengidentifikasi sistem yang berjalan, lalu mengerti kekurangan dalam sistem tersebut dan mencoba untuk menganalisa suatu pengembangan sistem seperti apa yang cocok untuk diterapkan di sistem tersebut. Penjelasan tahapan ini terdapat pada sub bab 4.2.

3.2.2 Design

Dari data-data yang didapatkan sebelumnya, tahap desain ini akan membuat gambar desain topologi jaringan interkoneksi yang akan dibangun, diharapkan dengan gambar ini akan memberikan gambaran seutuhnya dari kebutuhan yang ada.

Desain ini dapat berupa desain struktur topologi, desain alur proses, desain tata *layout* perkabelan, dan sebagainya yang akan memberikan gambaran jelas tentang *project* yang akan dibangun. Dalam penelitian ini, penulis menggunakan aplikasi *Microsoft*

Office Visio untuk membuat desain jaringan yang sudah ada dan yang akan dibuat. Penjelasan tahapan ini terdapat pada sub bab 4.3.

3.2.3 *Simulation Prototype*

Beberapa pengembang jaringan akan membuat dalam bentuk simulasi dengan bantuan *tools* khusus di bidang *network* seperti *Boson*, *Packet Tracert*, *Netsim* dan sebagainya. Hal ini dimaksudkan untuk melihat kinerja awal dari *network* yang akan dibangun dan sebagai bahan presentasi dan *sharing* dengan pengembang jaringan lainnya. Namun karena keterbatasan perangkat lunak simulasi ini, maka penulis hanya menggunakan alat bantu program simulator *VMware Workstation* versi 7 karena dapat membuat suatu *virtual machine* yang seolah-olah mempunyai fisik dan fungsi yang sama dengan sistem nyata dan *Microsoft Office Visio* untuk membangun skema topologi yang akan dibuat dan diagram alur kerja dari *load balancing*. Penjelasan dari tahapan ini terdapat pada sub bab 4.4.

3.2.4 *Implementation*

Di tahapan ini akan memakan waktu lebih lama dari tahapan sebelumnya. Dalam tahap implementasi, penulis menerapkan semua yang telah direncanakan dan dirancang sebelumnya. Pada tahapan inilah akan terlihat bagaimana sistem *load balancing* yang akan dibangun akan memberikan pengaruh

terhadap sistem yang telah ada. Penjelasan dari tahapan ini terdapat pada sub bab 4.5.

3.2.5 *Monitoring*

Setelah implementasi, tahapan *monitoring* merupakan tahapan yang penting agar jaringan komputer dan komunikasi dapat berjalan sesuai dengan keinginan dan tujuan awal dari penulis pada tahap awal analisis. Penulis akan menggunakan *tool-tool* yang ada di mikrotik yang berfungsi untuk *memonitor* lalu lintas data dengan membuat grafik dan meng-*capture* untuk mengukur besar penyebaran paket pada tiap-tiap ISP. Lalu dengan menggunakan aplikasi online yaitu di www.speedtest.net untuk mengukur kecepatan *bandwidth*. Kemudian membandingkan dengan sistem sebelum dan sesudah diterapkan *load balancing* di jaringan tersebut. Penjelasan dari tahapan ini terdapat pada sub bab 4.6.

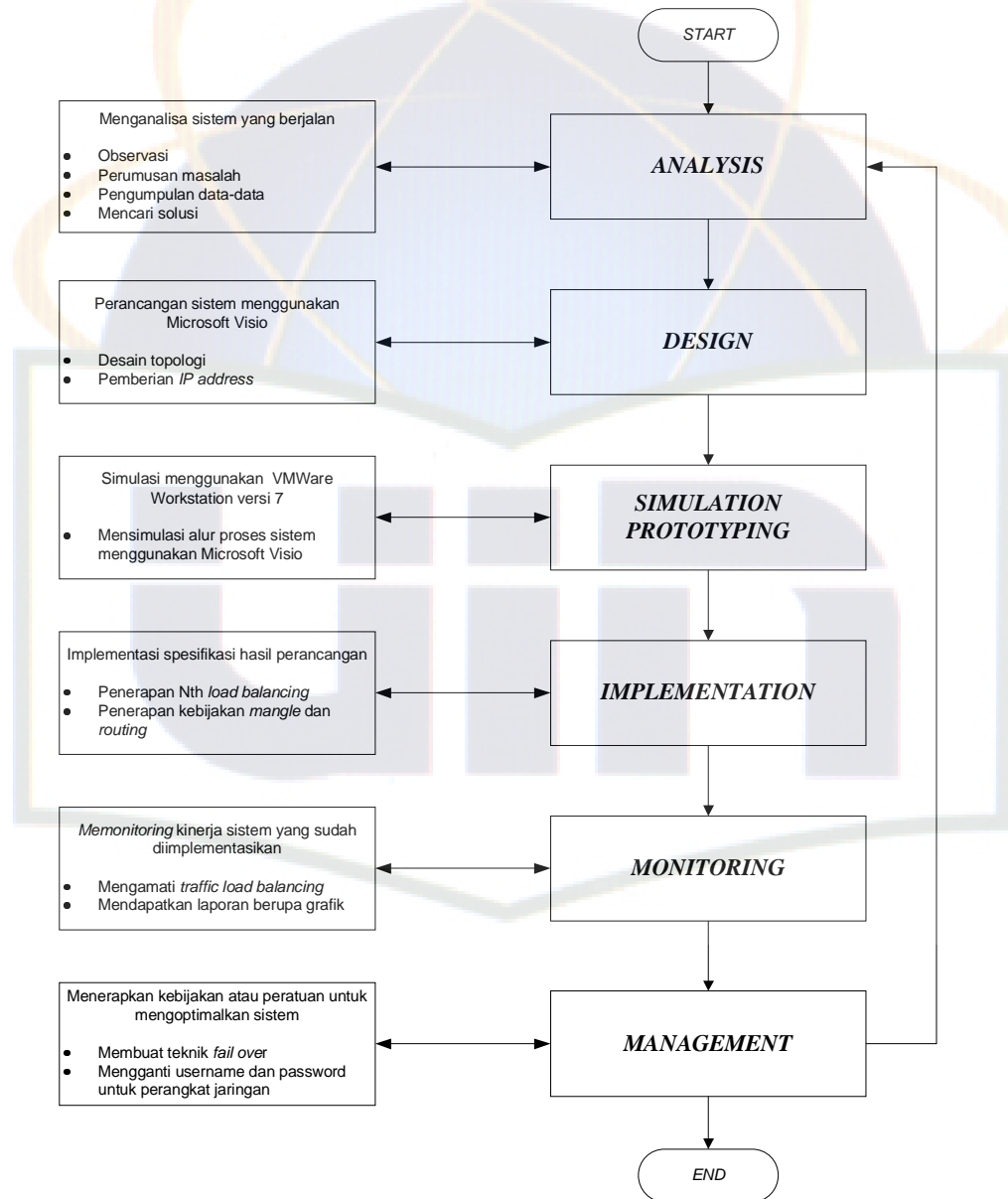
3.2.6 *Management*

Di manajemen atau pengaturan, salah satu yang menjadi perhatian khusus adalah masalah kebijakan, yaitu dalam hal aktivitas, pemeliharaan dan pengelolaan dikategorikan pada tahap ini. Kebijakan perlu dibuat untuk membuat dan mengatur agar sistem yang telah dibangun dan berjalan dengan baik dapat berlangsung lama dan unsur *reliability* terjaga. Penjelasan dari tahapan ini terdapat pada sub bab 4.7.

3.3 Kerangka Berpikir

Dalam melakukan penelitian ini, penulis melakukan tahapan-tahapan kegiatan dengan mengikuti rencana kegiatan yang tertuang dalam kerangka berpikir meliputi metode pengumpulan data dan metode pengembangan sistem. Kerangka berpikir penelitian ini terdapat pada gambar 3.2.





Gambar 3.2 Kerangka berpikir

BAB IV

PEMBAHASAN

Pada bab ini penulis akan menjelaskan secara detail dan terperinci proses implementasi *Nth load balancing* dengan menerapkan metode penelitian yang telah diuraikan pada bab 3. Pada bab tersebut telah dibahas bahwa model pengembangan sistem yang penulis akan gunakan adalah *Network Development Life Cycle* (NDLC).

Sebelum membahas tahapan pengembangan sistem, akan dijelaskan terlebih dahulu profil umum Sekolah Menengah Kejuruan PGRI Bekasi sebagai tempat penelitian.

4.1 Profil Sekolah Menengah Kejuruan PGRI Bekasi

SMK PGRI Bekasi adalah Sekolah Menengah Kejuruan yang didirikan tahun 2008 di Rawalumbu, Bekasi Timur.

Sekolah ini bertujuan untuk membentuk manusia yang bertakwa, berakhlak mulia, cerdas, kreatif, inovatif, mandiri, mampu berdaya saing tinggi dalam persaingan global, sehat jasmani dan rohani.

4.1.1 Visi

Menjadi SMK yang terkemuka dibidang teknologi informasi dan jaringan berbasis pembentukan karakter professional, mandiri dan menjunjung tinggi nilai-nilai spiritual.

4.1.2 Misi

1. Mendidik siswa dibidang keahlian Teknik Komputer Jaringan

2. Melakukan proses pembelajaran dengan 3 azaz utama yaitu professional, mandiri dan inovatif.
3. Mengharapkan nilai-nilai spiritual dalam proses pembelajaran.
4. Menghasilkan lulusan yang professional dan tangguh serta berakhlak mulia.

4.2 *Analysis*

Dalam perkembangannya, internet telah menjadi salah satu teknologi utama dalam penyampaian informasi. Kebutuhan manusia terhadap informasi telah menjadikan internet suatu sarana untuk mendapatkan informasi yang aktual dan *real time* dari suatu peristiwa. Semakin tingginya permintaan terhadap informasi yang ada di internet akan berbanding lurus dengan tingginya pengiriman paket-paket data yang ada. Dan hal ini juga akan mempengaruhi tingkat kecepatan pengaksesan data melalui internet. Para penyedia layanan internet mulai mendistribusikan produknya dengan kelebihan maupun kekurangan layanan internet tersebut.

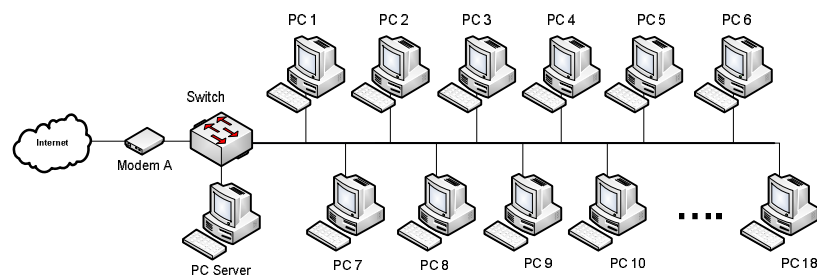
Oleh karena itu, pada tahap ini penulis mencoba menganalisa dan merumuskan sebuah masalah yang terjadi di laboratorium komputer SMK PGRI Bekasi mengenai jaringan komputer disana. Kurangnya sinyal koneksi internet menyebabkan proses belajar-mengajar menjadi terganggu. Maka daripada itu, penulis mencoba menerapkan *multiconnection* internet dan membagi beban *traffic* jaringan secara adil di kedua ISP dan juga

sebagai *backup* apabila salah satu koneksi tersebut dalam keadaan mati. Teknik ini dikenal dengan sebutan *load balancing*.

Dalam tahapan ini, penulis menganalisa sistem yang sedang berjalan dan mengetahui karakteristik dari jaringan yang ada, yaitu dengan mencatat *log* maupun meng-*capture port-port* apa yang sering digunakan didalam jaringan. Ini berguna agar dalam penerapan *load balancing* dapat berjalan secara optimal dan efektif. Pada tahap *analysis* ini penulis menjalani beberapa fase yaitu analisa sistem yang berjalan, analisa mengenai spesifikasi *software* dan *hardware* yang dibutuhkan.

4.2.1 Analisa Sistem berjalan

Untuk mendapatkan hasil yang akurat, pada tahap ini penulis melakukan observasi lapangan. Saat ini, teknologi yang digunakan di laboratorium komputer SMK PGRI Bekasi sangat sederhana. jaringan yang ada terdiri dari 1 PC untuk *server* dan 18 *client*. Gambar 4.1 adalah contoh topologi jaringan laboratorium komputer SMK PGRI Bekasi.



Gambar 4.1 Topologi jaringan laboratorium komputer

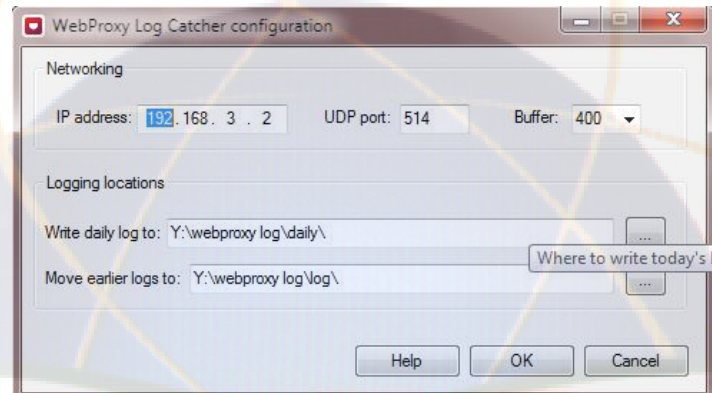
Pada tahap selanjutnya, penulis menggunakan aplikasi *torch* dan *webproxy log* untuk menganalisa dan *memonitoring* jaringan yang dilakukan selama tujuh hari. Aplikasi *webproxy log* yaitu untuk melakukan pencatatan *log-log* mengenai *website-website* yang sering dikunjungi, sedangkan *torch* untuk mengetahui *port-port* yang sering digunakan. Ini sangat diperlukan untuk menentukan metode *load balancing* yang dapat bekerja secara optimal dan efektif.

Dengan memanfaatkan *logging* yang ada di mikrotik, aplikasi tambahan seperti *webproxy log* akan menangkap *log-log* yang melewati mikrotik. Berikut ini adalah perintah di mikrotik untuk mengaktifkan *logging*.

```
/system logging action
add bsd-syslog=no name=WebProxyLog \
    remote=192.168.3.2:514 src-address=0.0.0.0 \
    syslog-facility=daemon syslog-severity=auto \
    target=remote

/system logging
add action=WebProxyLog disabled=no prefix=proxy \
    topics=web-proxy,!debug
```

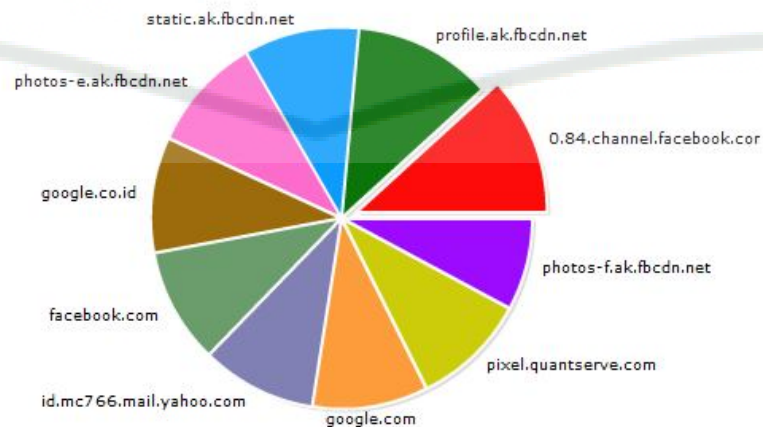
Setelah melakukan settingan di mikrotik, aplikasi *webproxy log* juga harus dikonfigurasi dengan memasukkan *IP address* dan *port* yang akan digunakan untuk menangkap *log*.



Gambar 4.2 Konfigurasi *Webproxy log*

Berikut ini adalah hasil dari *log-log* yang berhasil ditangkap dalam kurun waktu tujuh hari, lalu dijadikan laporan *top 10 servers* yang sering diakses *client*.

Top 10 servers

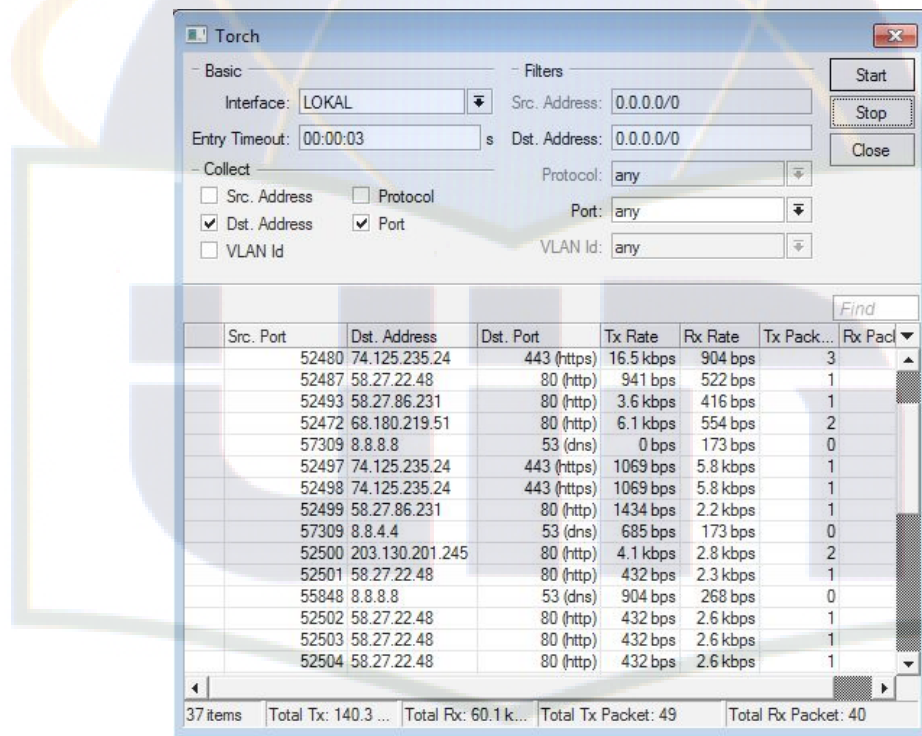


Report generated by WebProxy Log 1.5.0.4

Gambar 4.3 Grafik 10 *website* terbanyak yang diakses

Setelah mengetahui *top 10 server* yang sering diakses *client*, penulis juga menggunakan *tool* dari mikrotik, yaitu *torch*. *Tool* ini akan menangkap info-info mengenai *IP address* dan *port-port* yang

sering diakses oleh *client*. *Torch* dapat digunakan pada aplikasi winbox, yaitu dengan pilih menu *tools* lalu pilih *Torch*.berikut ini adalah tampilan dari *tool torch*.



Gambar 4.4 Aplikasi *monitoring Torch*

Setelah melakukan *monitoring* menggunakan *tool torch* dan aplikasi *webproxy log*, penulis mendapatkan data-data penting untuk melakukan pemilihan metode *load balancing* yang tepat digunakan pada jaringan laboratorium komputer SMK PGRI Bekasi.

Untuk memperjelas data-data yang diperoleh, penulis mengkonversinya menjadi tabel. berikut ini tabel 4.1 adalah laporan yang diambil dari persentase banyaknya koneksi yang terjadi.

Tabel 4.1 Daftar *traffic* dari tool *Torch*

No	IP Address	Port	Website	Persentase
1	74.125.71.147	80	google.com	70%
2	208.80.152.2	80	wikipedia.com	
3	68.180.206.184	80	yahoo.com	
4	121.101.158.176	80	yahoo.com	
5	209.85.175.138	80	google.com	
6	69.63.189.39	80	facebook.com	
7	121.101.158.176	80	yahoo.com	
8	66.220.158.25	80	facebook.com	
9	76.74.254.123	80	wordpress.com	
10	124.108.120.244	80	yahoo.com	
11	112.78.131.5	80	kaskus.us	
12	66.220.146.29	80	facebook.com	
13	209.85.175.138	80	google.com	
14	75.101.142.23	80	twitter.com	
15	66.163.169.186	443	/login.yahoo.com/	20%
16	118.215.101.227	443	-	
17	98.137.130.27	443	-	
18	199.59.148.139	443	-	
19	98.136.48.112	5050	-	10%
20	69.63.180.44	5050	-	

Dari sebagian *log* alamat IP yang diperoleh aplikasi *torch*, banyak *IP address* yang tidak dapat dimuat didalam *browser*. Ini dikarenakan pada suatu halaman *website* terdapat bermacam-macam isi konten yang beralamatkan *IP address* berbeda yang tidak dapat diakses melalui *browser*.

Dengan melihat daftar *traffic* tersebut, bisa diambil kesimpulan bahwa *port* 80 (HTTP) merupakan *port* yang paling

banyak digunakan. Selebihnya *port* 443 (HTTPS) maupun *port* 5050 (*port* untuk *messenger*). Dengan hasil perbandingan yang diperoleh, maka diketahui metode *load balancing* yang tepat untuk jaringan ini adalah menggunakan metode *Nth load balancing*.

Pemilihan metode *Nth load balancing* dikarenakan kelebihan dari metode ini adalah dapat membagi *traffic* jaringan secara adil dan meminimalisir terjadinya *overload* pada salah satu koneksi ISP. Sedangkan kekurangannya yaitu sering terjadi diskoneksi untuk aplikasi *realtime* dikarenakan perpindahan *gateway* dapat diminimalkan dengan dibuat peraturan tambahan.

Yaitu dengan cara hanya *port* yang banyak dipakai yaitu *port* 80 (HTTP) yang hanya akan di *load balancing*. Selebihnya *port-port* lain akan menggunakan *default gateway*, ini dikarenakan untuk *port* 443 (HTTPS) atau *port* 5050 (*messaging*) tidak boleh terjadi perpindahan *gateway*, karena akan menyebabkan terbacanya *IP address* yang berbeda oleh *server* dan dapat menyebabkan terputusnya sesi koneksi antara *server* dan *client*.

Improvisasi dalam penerapan *Nth load balancing* dapat digunakan pada mikrotik, yaitu pada saat pembuatan *connection mark* ditambahkan peraturan yang hanya akan menandai koneksi yang menggunakan *port* 80 (HTTP), lalu pada *routing mark*, *router* akan menentukan jalur yang terbaik sesuai urutan antrian agar tidak terjadi *overload* pada salah satu koneksi ISP.

4.2.2 Spesifikasi *Software* dan *Hardware*

Setelah mengetahui metode *load balancing* yang akan diimplementasikan, fase selanjutnya yaitu menganalisa dan menentukan *software* dan *hardware* apa saja yang dibutuhkan dalam membangun sistem Nth *load balancing*. Berikut ini pada tabel 4.2 akan dijelaskan spesifikasi *software* dan pada tabel 4.3 dijelaskan pula spesifikasi *hardware* yang dibutuhkan dalam implementasi Nth *load balancing*.

Tabel 4.2 Spesifikasi *Software*

No	Software	Keterangan
1	Mikrotik RouterOS ver. 4.11	Sistem operasi untuk Mikrotik
2	Microsoft Windows XP SP2	Sistem operasi untuk <i>Admin</i> dan <i>Client</i>
3	Mikrotik Winbox Loader v2.2.18	Software GUI untuk mikrotik

Tabel 4.3 Spesifikasi *Hardware*

No	Perangkat	Jumlah	Spesifikasi Unit
1	Mikrotik RB750	1	CPU: AR7240 300 MHz CPU Memory: 32MB DDR SDRAM Data storage: 64MB Ethernet: 5 ports Dimensions: 113x89x28mm Weight: 130g

2	PC <i>Server/admin</i>	1	Intel Pentium 4 Memory RAM 2 GB Harddisk 3,5" IDE 160 GB DVD-RW Monitor LCD 15"
3	PC <i>Client</i>	18	Intel Pentium 4 Memory RAM 1 GB Harddisk 3,5" IDE 80 GB Monitor LCD 15"
4	Modem <i>Router</i> ZTE MF608	1	Menggunakan ISP Indosat IM2 paket broom dengan kecepatan mencapai 256 Kbps
5	Wireless N <i>Router</i> TP-MR3420	1	Menggunakan ISP Smart paket silver unlimited dengan kecepatan mencapai 512 Kbps

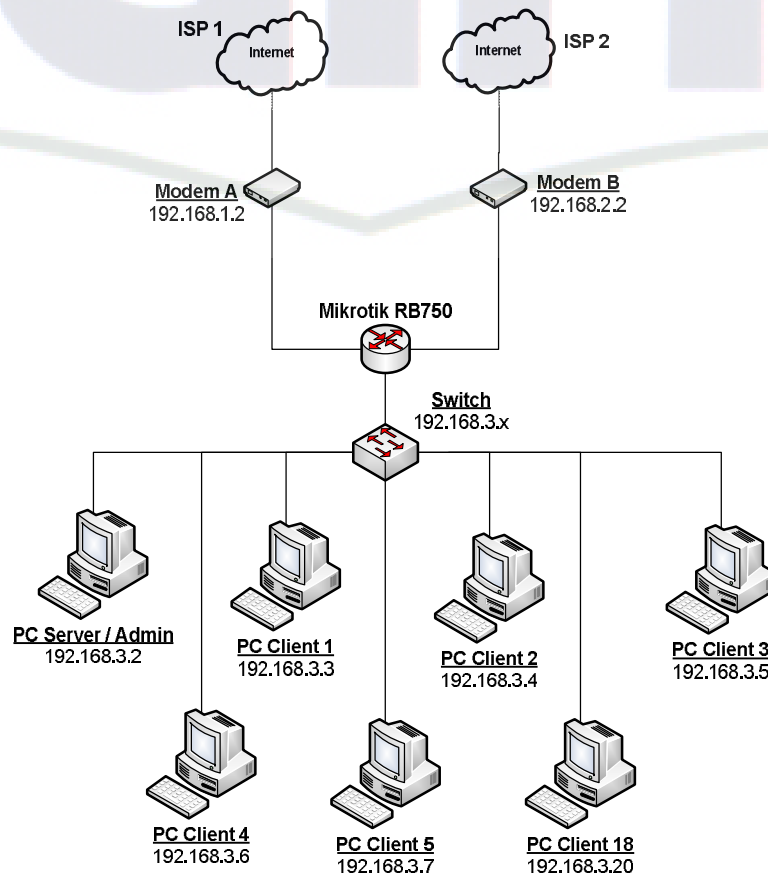
4.3 *Design*

Pada tahap analisis, penulis telah mendapatkan rincian spesifikasi dari sistem yang telah berjalan. Dan di tahap perancangan ini adalah tahapan yang bertujuan untuk mengatasi permasalahan yang ada. Dari data-data yang didapatkan dari tahap analisis, tahap perancangan ini akan membuat rancangan topologi yang akan dibangun termasuk rincian yang

akan dibutuhkan dalam mengimplementasikan *Nth load balancing* ke dalam sistem jaringan di laboratorium komputer SMK PGRI Bekasi.

4.3.1 Perancangan Fisik

Perancangan fisik merupakan perancangan sebuah struktur jaringan yang berhubungan dengan peralatan yang akan digunakan dan pembentukan sebuah topologi jaringan. Ini dimaksud agar dalam pengimplementasiian *Nth load balancing* akan mudah dipahami dan dapat digunakan untuk *troubleshooting* jaringan. Gambar 4.5 adalah topologi jaringan yang akan dibangun dengan 1 PC sebagai *server/admin* dan 18 PC sebagai *client*.



Gambar 4.5 Desain topologi jaringan dengan *load balancing*

Tabel 4.4 Tabel *IP address*

Perangkat	Interface	IP address	Gateway
Mikrotik RB750	ISP-1 (Ether2)	192.168.1.1	192.168.1.2
	ISP-2 (Ether3)	192.168.2.1	192.168.2.2
	Lokal (Ether4)	192.168.3.1	-
<i>Switch</i>	<i>Ethernet</i>	-	-
PC Server / Admin	<i>Ethernet</i>	192.168.3.2	192.168.3.1
PC Client 1	<i>Ethernet</i>	192.168.3.3	192.168.3.1
PC Client 2	<i>Ethernet</i>	192.168.3.4	192.168.3.1
PC Client 3	<i>Ethernet</i>	192.168.3.5	192.168.3.1
PC Client 4	<i>Ethernet</i>	192.168.3.6	192.168.3.1
PC Client 5	<i>Ethernet</i>	192.168.3.7	192.168.3.1
PC Client 18	<i>Ethernet</i>	192.168.3.20	192.168.3.1

Terdapat *interface* yang ada pada sisi *router* yaitu dengan penjelasan sebagai berikut:

1. *Interface* ISP-1: merupakan *interface* yang terkoneksi dengan jaringan yang menuju modem A atau *gateway* ISP-1.
2. *Interface* ISP-2: merupakan *interface* yang terkoneksi dengan jaringan yang menuju modem B atau *gateway* ISP-2.
3. *Interface* Lokal: merupakan *interface* yang terkoneksi dengan jaringan lokal yang menghubungkan *client* dengan *router*.

4.4 *Simulation Prototyping*

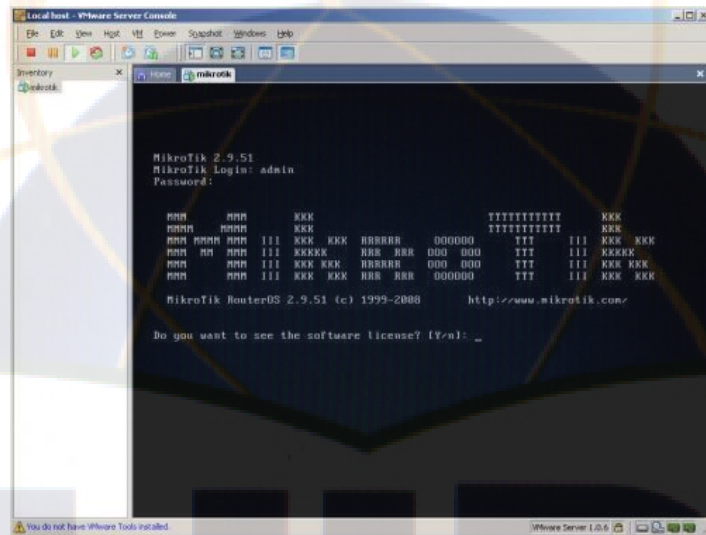
Sebelum melakukan tahap implementasi secara utuh, penulis melakukan tahap simulasi atau uji coba terhadap sistem yang telah dirancang dengan menggunakan simulator. Tahap simulasi bertujuan untuk

melihat kinerja awal dari jaringan yang akan dibangun, memperkecil resiko kegagalan dan akan dijadikan sebagai bahan pertimbangan sebelum jaringan benar-benar akan diimplementasikan.

Ada beberapa *tool* yang penulis gunakan untuk mensimulasikan sistem yang akan dibangun sebagai *prototype* karena kemampuannya dalam mempresentasikan topologi jaringan dan memberikan status layaknya sebuah sistem nyata. Adapun tujuan dari dibuatnya *prototype* oleh penulis antara lain:

1. Memprediksi apa saja hal-hal yang harus diperhatikan dalam proses implementasi pada lingkungan jaringan yang sesungguhnya.
2. Memperkecil resiko kegagalan saat proses pembangunan dan implementasi sistem pada lingkungan jaringan yang sesungguhnya.
3. Menjamin bahwa kesalahan yang terjadi pada saat proses perancangan, pembangunan dan implementasi tidak mengganggu dan mempengaruhi lingkungan sistem yang nyata.

Pada tahap ini, penulis menggunakan *VMware Workstation ver 7.0* untuk membuat mesin virtual, dan akan menginstall Mikrotik *RouterOS* kedalam mesin virtual tersebut. Gambar 4.6 adalah salah satu tampilan dari aplikasi *VMware Workstation*, sedangkan proses instalasi mikrotik *RoterOS* yang akan diinstall di mesin virtual akan dijelaskan di lampiran.



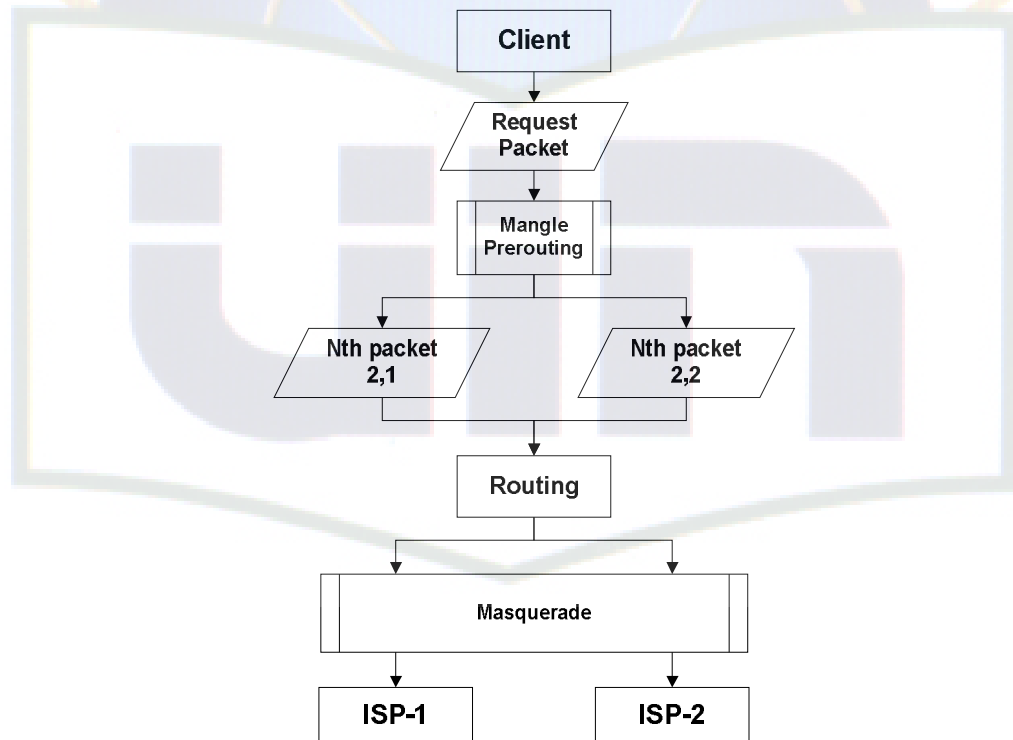
Gambar 4.6 VMware workstation ver 7.0

Untuk menjelaskan proses sistem yang akan dibangun, penulis akan membuat simulasi yang akan menggambarkan proses dari sistem yang akan dibangun yaitu sebagai berikut:

1. Sistem terdiri dari 3 kelompok jaringan atau subnet yaitu kelompok jaringan antara lokal dan *router*, kelompok jaringan antara *router* dan ISP-1 dan kelompok jaringan antara *router* dan ISP-2.
2. Sistem akan memproses semua data dari *client* yang menuju ke akses internet pada sisi *router* dimana akan terjadinya proses *mangle* berdasarkan urutan paket Nth, lalu proses *routing* paket yang akan diarahkan melalui ISP-1 atau ISP-2. Seperti digambarkan pada gambar 4.7.
3. Paket data dari *client* yang masuk ke *router* akan ditandai dengan *connection mark* pada tahapan *mangle* berdasarkan urutan Nth yang dibuat. Lalu setiap tanda di paket Nth tersebut

akan diberikan *routing mark* yang akan menentukan jalur mana yang harus dilaluinya.

4. Pada tahapan *masquerade*, *IP address* dari data yang akan di-*forward*, sebelumnya akan ditranslasikan dengan *IP address* dari *interface* ISP yang digunakan menjadi *gateway*.



Gambar 4.7 Alur proses pengiriman paket pada Nth *load balancing*

4.5 Implementation

Setelah semua tahapan simulasi telah berhasil, maka langkah selanjutnya adalah tahapan implementasi. Tahapan ini mengacu pada tahapan desain yang telah dibuat dan disimulasikan. Berikut ini adalah proses-proses yang dilakukan pada tahapan implementasi.

4.5.1 Implementasi Topologi jaringan

Hal yang pertama kali penulis lakukan adalah mengumpulkan dan memasang seluruh *hardware* yang diperlukan dalam mengimplementasikan *load balancing*, sesuai dengan rancangan topologi yang telah penulis buat di tahapan desain. Setelah itu barulah penulis melakukan konfigurasi pada *hardware*.

4.5.2 Inisialisasi *Interface* Mikrotik

Inisialisasi *interface* berguna untuk memudahkan penulis dalam melakukan pengembangan sistem dengan cara memberikan nama pada masing-masing *interface* sesuai dengan fungsinya. Perintah-perintah yang dilakukan adalah sebagai berikut.

```
/interface Ethernet
Set 1 comment="" disables=no name="ISP-1"
Set 2 comment="" disables=no name="ISP-2"
Set 3 comment="" disables=no name="LOKAL"
```

Perintah “*set 1*” merupakan perintah untuk mengkonfigurasi *interface ether 2* yang terdapat pada mikrotik dengan perintah menghidupkan *interface* dan memberi nama *interface* yaitu ISP-1. Begitu pula untuk perintah-perintah selanjutnya.

4.5.3 Pemberian Alamat *IP address*

Pada tahap ini akan dilakukan pemberian alamat *IP address* pada tiap *interface* yang ada pada laboratorium komputer SMK

PGRI Bekasi, baik pada modem *router*, mikrotik maupun dari sisi *client*.

1. Pada mikrotik

Untuk melakukan pemberian *IP address* pada *router* dengan menggunakan perintah sebagai berikut.

```
/ip address
add address=192.168.1.1/24 network=192.168.1.0 \
    broadcast=192.168.1.255 comment="" \
    disabled=no interface=ISP-1
add address=192.168.2.1/24 network=192.168.2.0 \
    broadcast=192.168.2.255 comment="" \
    disabled=no interface=ISP-2
add address=192.168.3.1/24 network=192.168.3.0 \
    broadcast=192.168.3.255 comment="" \
    disabled=no interface=LOKAL
```

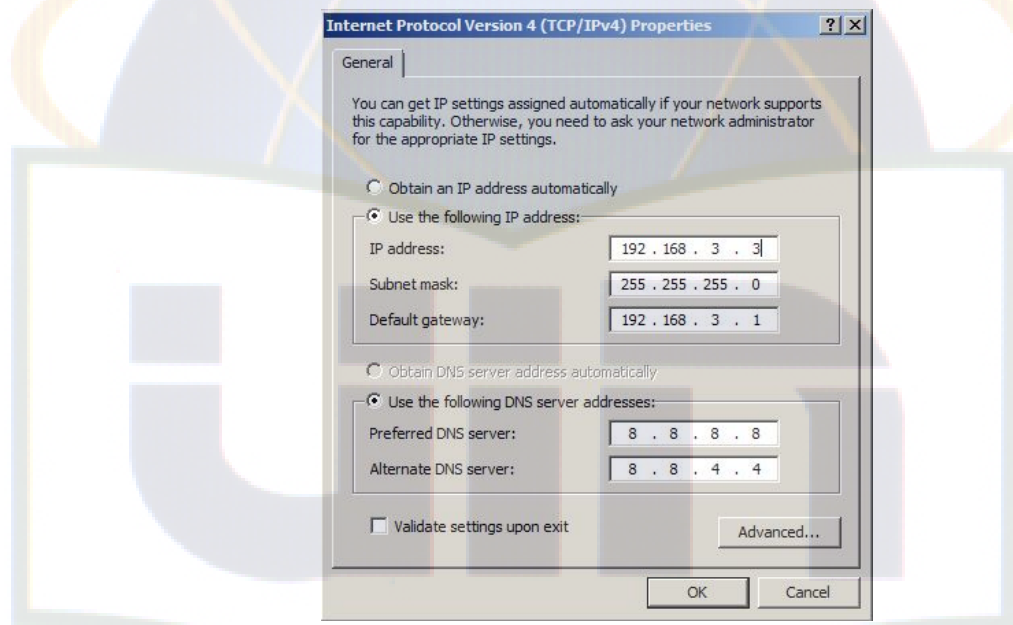
Baris pertama berarti memerintahkan untuk memberikan *interface* ISP-1 dengan *IP address* 192.168.1.1 dengan *subnetting* /24 yaitu 255.255.255.0. begitu pula dengan keterangan perintah-perintah selanjutnya

2. Pada *client*

Untuk memberikan *IP address* pada sisi *client* yaitu dengan cara sebagai berikut.

- a) klik *Start Menu* > *Control Panel* > *Network Connection*.
- b) pilih dan klik kanan pada *Local Area Connection* > *Properties*.
- c) pilih *Internet Protocol (TCP/IP)* > *Properties*.

- d) masukan *IP address* dengan *range* 192.168.3.2-192.168.3.254 dengan subnet mask 255.255.255.0 dan *gateway* 192.168.3.1.



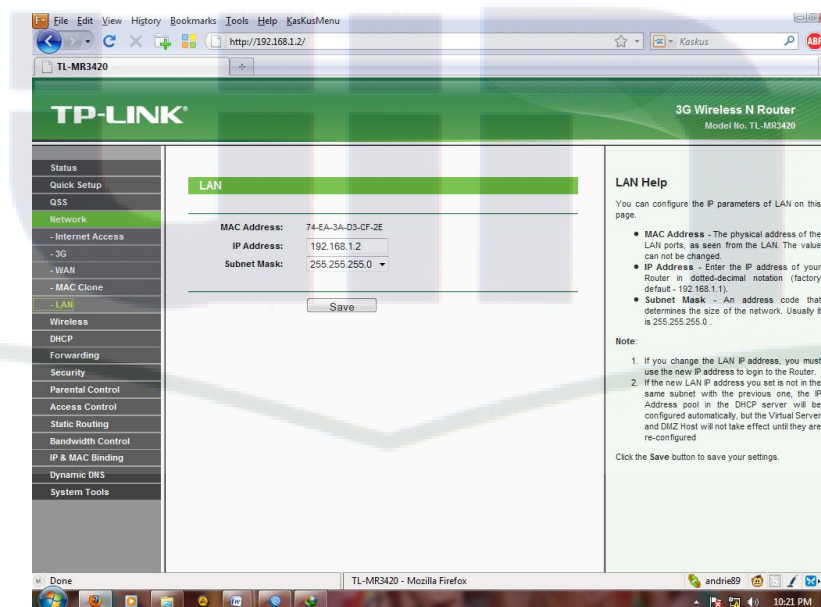
Gambar 4.8 Konfigurasi PC *client*

3. Pada modem *router*

Untuk mengkonfigurasi *IP address* pada modem *router* ZTE MF608 yaitu dengan cara sebagai berikut.

- a) Hubungkan secara *peer to peer* antara modem dengan 1 PC menggunakan kabel LAN *straight*.
- b) Samakan *network ID* antara PC dengan modem. Dalam hal ini *IP address default* untuk setiap modem *router* adalah 192.168.1.1/24. Berarti buat *IP address* untuk PC dengan *network id* 192.168.1.x/24.

- c) Buka browser pada PC, ketikkan *IP address* modem pada *address bar*.
- d) Ganti *IP address* pada *port ethernet 1* yang akan digunakan untuk menghubungkan dengan mikrotik dengan *IP address* 192.168.1.2/24.
- e) Lakukan hal yang sama dengan modem kedua dengan *IP* 192.168.2.2/24.



Gambar 4.9 Konfigurasi modem *router*

4.5.4 Konfigurasi *Mangle*

Mangle adalah tahapan dimana paket data yang datang dari suatu *interface* tertentu akan diproses. Fungsi dari aturan yang ada *dimangle* adalah untuk menandai paket agar dapat diarahkan sesuai dengan *rule routing* yang ada. Di tahap ini penulis akan menerapkan aturan *mangle* dari metode *Nth load balancing*.

Sebelum masuk ke tahap itu, penulis akan menjelaskan teori dasar mengenai variabel yang ada di Nth.

1. *Every*: Angka *every* adalah jumlah kelompok yang ingin dihasilkan. Jadi bila administrator ingin membagi alur koneksi yang ada menjadi 2 kelompok yang nantinya akan di *load balance* ke 2 koneksi yang ada, maka angka *every* = 2.

2. *Packet*: Angka *packet* adalah jumlah koneksi yang akan ditandai atau di-*mangle*. Jika ingin membuat 2 kelompok, tentunya harus membuat 2 *mangle rules*. Pada *rules* tersebut, angka untuk *every* haruslah sama, namun untuk angka *packet* harus berubah. Untuk 2 kelompok, berarti angka *packet* untuk 2 *rules* tersebut adalah 1 dan 2.

3. *Counter*: *Counter* atau disebut penghitung atau pencacah biner. Mulai dari mikrotik versi 3.x nilai *counter* tidak didefinisikan langsung oleh administrator. Setiap *rules* memiliki *counter* sendiri. Ketika *rules* menerima paket, maka *counter* untuk aturan saat itu akan otomatis bertambah satu. Dan jika nilai *counter* telah sama dengan nilai *every*, maka paket akan dicocokkan dan *counter* akan diatur ke nilai awal.

Keterangan diatas dimaksudkan agar untuk mudah memahami penentuan nilai dari variabel yang ada di Nth *load balancing*. Berikut ini adalah perintah-perintah yang ada pada tahapan *mangle*.

```

/ip firewall mangle
add action=mark-connection chain=prerouting \
    comment="" connection-state=new disabled=no \
    dst-port=80 in-interface=LOKAL \
    new-connection-mark=lb_1 nth=2,1 protocol=tcp \
    passthrough=yes \

add action=mark-routing chain=prerouting comment="" \
    connection-mark=lb_1 disabled=no \
    in-interface=LOKAL new-routing-mark=route_1 \
    passthrough=no

```

Perintah awal diatas adalah bentuk penandaan sebelum paket data masuk ke dalam kebijakan *routing* (*prerouting*). Dimana *interface* “LOKAL” dengan *destination port* 80 diberikan *connection mark* dengan nama “lb_1” dengan nilai *nth* yaitu 2,1 yang berarti nilai *every*=2 dan nilai *packet*=1. Lalu ditambah dengan perintah *passthrough=yes* yaitu *command* pada baris awal akan diteruskan ke *rule* baris berikutnya.

Pada baris kedua, dijelaskan bahwa paket data yang berada di *interface* “LOKAL” dengan atribut *connection mark* “lb_1” akan ditandai *routing mark* dengan nama “route_1”.

Setelah selesai *rule* yang pertama, dilanjutkan pada *rule* kedua.

Command yang digunakan juga sama pada *rule* pertama.

```

/ip firewall mangle
add action=mark-connection chain=prerouting \
    comment="" connection-state=new disabled=no \
    dst-port=80 in-interface=LOKAL \
    new-connection-mark=lb_2 nth=2,2 protocol=tcp \
    passthrough=yes \

add action=mark-routing chain=prerouting comment="" \
    connection-mark=lb_2 disabled=no \
    in-interface=LOKAL new-routing-mark=route_2 \
    passthrough=no

```

Sama seperti yang sebelumnya, *interface* “LOKAL” dengan *destination port* 80 diberikan *connection mark* kali ini dengan nama “lb_2” dengan nilai *nth* yaitu 2,2 yang berarti nilai *every*=2 dan nilai

packet=2. Lalu ditambah dengan perintah *passthrough=yes* yaitu *command* pada baris awal akan diteruskan ke *rule* baris berikutnya.

Pada baris kedua, dijelaskan bahwa paket data yang berada di *interface* “LOKAL” dengan atribut *connection mark* “lb_2” akan ditandai *routing mark* dengan nama “route_2”.

Perintah-perintah yang telah dibuat diatas adalah untuk penandaan pada antrian paket data yang berada di *interface* LOKAL. Setiap paket data akan diberi tanda oleh Nth secara berurutan dan berulang-ulang. Ini menyebabkan metode ini dapat membagi *traffic* jaringan secara merata dan tidak terjadi *overload*.

Namun disini penulis hanya menerapkan *rule* tersebut hanya untuk *client* yang ingin mengakses *port* 80 saja. Ini dimaksud agar aplikasi seperti *yahoo messenger* maupun *traffic* HTTPS yang menggunakan *port* 5050 dan *port* 443 tidak mengalami diskoneksi yang disebabkan oleh *load balancing* yang akan berbeda-beda *IP address*. Maka aplikasi tersebut akan didefinisikan hanya menggunakan satu *default routing* tanpa harus melalui proses *load balancing*.

4.5.5 Konfigurasi Routing

Untuk meneruskan paket yang telah ditandai pada proses *mangle*, maka harus dibuat aturan baru pada *routing* tabel agar dapat melewatkan paket data tersebut ke *gateway* ISP yang sesuai dengan *marking* paket yang dibuat pada tahapan *mangle*. Berikut ini untuk membuat aturan pada *routing table*.

```

/ip route
add comment="" disabled=no distance=1 \
  dst-address=0.0.0.0/0 gateway=192.168.1.2 \
  routing-mark=route_1 scope=255 target-scope=10
add comment="" disabled=no distance=1 \
  dst-address=0.0.0.0/0 gateway=192.168.2.2 \
  routing-mark=route_2 scope=255 target-scope=10

```

Pada baris pertama diperintahkan untuk setiap *routing mark* dengan nama “route_1” akan selalu melalui *gateway* 192.168.1.2. dan untuk setiap *routing mark* dengan nama “route_2” akan selalu melalui *gateway* 192.168.2.2.

```

/ip route
add check-gateway=ping comment="default gateway"
  disabled=no distance=1 dst-address=0.0.0.0/0 \
  gateway=192.168.2.2 scope=255 target-scope=10

```

Perintah selanjutnya yaitu dengan menambahkan lagi *gateway* 192.168.2.2 tanpa *routing mark*, dan *gateway* ini dijadikan sebagai *default gateway* untuk semua *traffic* yang tidak melalui proses *load balancing*. Ini dimaksud untuk mencegah aplikasi *Yahoo Messenger* atau semua *traffic* HTTPS maupun SSL mengalami diskoneksi. Karena menggunakan satu *gateway*, maka aplikasi tersebut tetap menggunakan satu *IP Public* tanpa berubah-ubah dikarenakan proses *load balancing*.

4.5.6 Konfigurasi NAT

Konfigurasi terakhir dalam *Nth load balancing* adalah NAT atau dalam mikrotik lebih dikenal dengan *masquerade*. Ini berfungsi agar dapat mengubah alamat sumber paket yaitu alamat *client* yang memiliki *IP address private* agar dapat dikenali di

internet yaitu dengan cara mentranslasikannya menjadi *IP address public*. Berikut ini adalah perintahnya.

```
/ip firewall nat
add action=masquerade chain=srcnat comment="" \
    disabled=no out-interface=ISP-1
add action=masquerade chain=srcnat comment="" \
    disabled=no out-interface=ISP-2
```

Dengan perintah diatas, *router* akan melakukan *masquerade* terhadap paket data yang berasal dari “srcnat” atau dari alamat *client*. Lalu setelah itu akan paket akan dilempar ke *gateway* sesuai dari tujuan paket tersebut.

Selanjutnya ada perintah terakhir yang harus dikonfigurasi pada mikrotik agar *client* dari *interface* LOKAL dapat melakukan ping terhadap kedua *gateway*. Perintah ini dimaksud yaitu melakukan *action redirect* untuk semua *protocol* ICMP tanpa harus melalui proses *load balancing*.

```
/ip firewall nat
add action=redirect chain=dstnat comment="" \
    disabled=no in-interface=LOKAL protocol=icmp
```

4.6 Monitoring

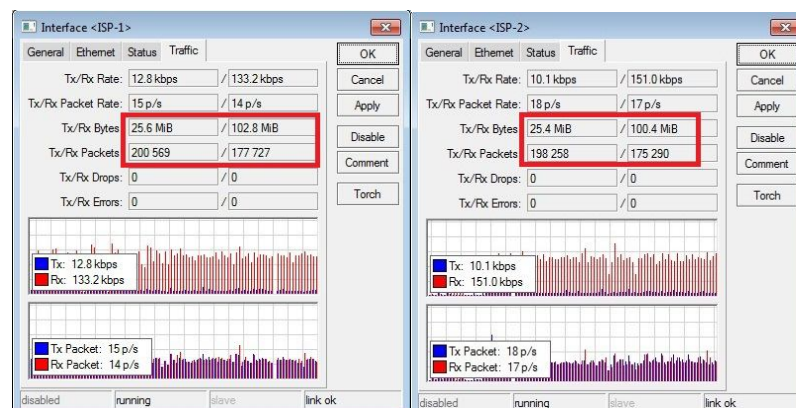
Setelah Nth *load balancing* berhasil diimplementasikan di jaringan laboratorium komputer SMK PGRI Bekasi, lalu tahap selanjutnya pada metode pengembangan sistem NDLC ialah tahapan *monitoring*. Pada tahap ini penulis akan melakukan pengujian sejauh mana sistem yang telah dibangun berjalan dalam mengoptimalisasi kinerja dari jaringan tersebut

dengan menggunakan *Nth load balancing*. Untuk melakukan *monitoring* sistem ini penulis menggunakan beberapa *tool* baik yang terdapat pada winbox maupun aplikasi *online* seperti www.speedtest.net untuk mengetahui hasil kecepatan koneksi apakah hasil dari implementasi dari *Nth load balancing* memberikan hasil yang optimal.

Pengujian sistem ini dilakukan dengan kondisi *client* aktif sebanyak 18 *user* dan dilakukan dalam 2 fase, yaitu pengujian dalam pembagian beban pada tiap ISP dan pengujian dalam mengetahui optimalisasi kecepatan koneksi internet yang telah dicapai setelah diimplementasikan *Nth load balancing*.

4.6.1 Pengujian Efektifitas Penyetaraan Beban pada Gateway ISP

Pada tahap ini penulis *memonitoring* sistem jaringan dengan menggunakan aplikasi atau *tools* yang ada pada winbox. Hasil dari *monitoring* dapat dilihat pada menu *interface list*. Berikut ini adalah hasil *monitoring*nya:

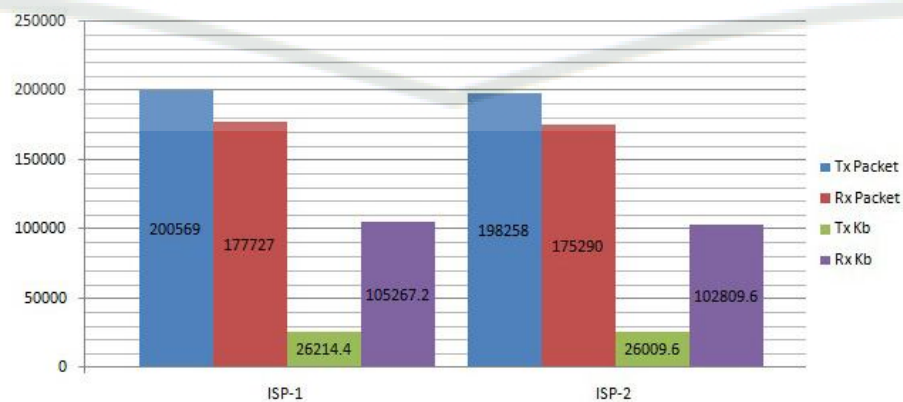


Gambar 4.10 Grafik koneksi pada tiap gateway ISP

Parameter yang dilihat dari kedua *traffic* di *interface* ini adalah besar rata-rata penyebaran dari tiap-tiap *gateway* ISP. Pada *interface* ISP-1 dan *interface* ISP-2 terlihat besar *packet* dan *bytes* yang telah dilewati. Hasil dari monitoring tersebut lalu penulis masukan kedalam tabel dan dibuatkan grafik untuk memperjelas perbandingan dari masing-masing *interface*.

Tabel 4.4 Tabel perbandingan penyebaran paket data

<i>Gateway</i>	Jumlah Paket (packet)		Ukuran Paket (Kb)	
	Tx	Rx	Tx	Rx
ISP-1	200569	177727	26214.4	105267.2
ISP-2	198258	175290	26009.6	102809.6



Gambar 4.11 Grafik perbandingan penyebaran paket data

Berdasarkan tabel 4.4 dan gambar 4.11, terlihat jika *load balancing* telah berhasil menyebarkan *packet* dan *bytes* yang hampir sama di kedua *interface*.

4.6.2 Pengujian Performa *Load balancing*

Pada tahap ini penulis akan menguji kualitas dari koneksi yang telah dibangun dengan menggunakan aplikasi berbasis *online* yaitu www.speedtest.net. dengan pengujian ini akan diketahui *grade* dari kualitas *bandwidth* yang dihasilkan. Selain itu, informasi yang didapat ialah besar *ping*, *download* dan *upload speed*.

Pada tahap ini akan penulis akan membandingkan kecepatan *bandwidth* antara ISP IM2 dan Smart sebelum dilakukan *Nth load balancing* lalu akan membandingkan dengan kedua ISP tersebut yang telah diimplementasikan *Nth load balancing*.

Pengujian akan dilakukan lima kali uji coba pada *server* yang sama, lalu data-data yang diterima akan dibuatkan tabel perbandingan. Adapun keterangan dari hasil pengujian, lebih jelas dapat dilihat pada lampiran.

Tabel 4.5 Pengujian sebelum implementasi *load balancing*

Pengujian	IM2				SMART			
	Ping (ms)	Download (Mbps)	Upload (Mbps)	Grade	Ping (ms)	Download (Mbps)	Upload (Mbps)	Grade
1	148	0.26	0.05	F	442	0.20	0.07	F
2	331	0.22	0.08	F	392	0.15	0.05	F
3	223	0.26	0.06	F	474	0.16	0.06	F
4	143	0.17	0.10	F+	578	0.13	0.09	F
5	235	0.25	0.07	F+	286	0.21	0.07	F

Tabel 4.6 Pengujian setelah implementasi *load balancing*

Pengujian	Ping (ms)	Download (Mbps)	Upload (Mbps)	Grade
1	454	0.22	0.09	F+
2	429	0.30	0.05	F
3	160	0.27	0.07	F+
4	401	0.31	0.05	F
5	164	0.30	0.05	F

Dari tabel 4.5 dan tabel 4.6 diketahui perbandingan kualitas koneksi dari sebelum dan sesudah diimplementasikan Nth *load balancing*. Walaupun tidak terlalu mendapatkan perubahan yang signifikan, namun masih terdapat perbaikan kualitas *bandwidth* setelah mengimplementasikan Nth *load balancing*. Ini dapat dijelaskan bahwa *load balancing* adalah teknik menyeimbangkan koneksi diantara kedua ISP, bukan untuk menyatukannya. *Load balancing* bukan berarti $1+1=2$ namun $1+1=1+1$.

4.7 Management

Di tahapan ini merupakan tahapan pengaturan, salah satu yang menjadi perhatian khusus adalah masalah kebijakan, yaitu dalam hal aktivitas, pemeliharaan dan pengelolaan dikategorikan pada tahap ini. Kebijakan perlu dibuat untuk membuat dan mengatur agar sistem yang telah dibangun dan berjalan dengan baik dapat berlangsung lama dan unsur

reliability terjaga. Dalam hal ini penulis menambahkan kebijakan agar sistem Nth *load balancing* berjalan dengan optimal, yaitu antara lain.

4.7.1 Membuat pengaturan “fail over”.

Pengertian dari *fail over* dalam *load balancing* adalah dimana salah satu koneksi *gateway* sedang diskoneksi, maka *gateway* lainnya otomatis akan menjadi *default gateway* yang menopang semua *traffic* jaringan. Berikut ini adalah perintahnya.

```
/ip route
add check-gateway=ping comment="" disabled=no \
    distance=2 dst-address=0.0.0.0/0 \
    gateway=192.168.1.2 scope=255 target-scope=10
```

Kunci utama dari perintah *fail over* ini terdapat pada pendefinisian “distance=2”. Pada awal perintah *routing*, *gateway* diberikan “distance=1”. Ini dimaksudkan agar *routing* akan selalu mendahulukan nilai *distance* yang terkecil terlebih dahulu. Lalu perintah “add check gateway=ping” berarti *gateway* akan selalu di cek dengan cara melakukan ping, apakah dalam keadaan hidup atau diskonek. Jika *gateway* 192.168.2.2 tidak *mereplay*, maka *router* akan menganggap *gateway* tersebut dalam keadaan *down* dan akan menjadikan *gateway* 192.168.1.2 sebagai *gateway* dengan koneksi tunggal. Kondisi ini juga berlaku sebaliknya. Berikut ini pada gambar 4.12 adalah tampilannya jika salah satu koneksi dalam keadaan mati.

Route List

	Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
AS	0.0.0.0/0	192.168.1.1 reachable ISP-1	1	route_1	
S	0.0.0.0/0	192.168.2.2 unreachable	1	route_2	
... default gateway					
S	0.0.0.0/0	192.168.2.2 unreachable	1		
AS	0.0.0.0/0	192.168.1.1 reachable ISP-1	2		
DAC	192.168.1.0/24	ISP-1 reachable	0		192.168.1.2
DAC	192.168.2.0/24	ISP-2 unreachable	0		192.168.2.1
DAC	192.168.3.0/24	LOKAL reachable	0		192.168.3.1

Interface List

	Name	Type	L2 MTU	Tx	Rx	Tx Pac.	Rx Pac.	Tx Drops	Rx Drops
R	ISP-1	Ethernet	1524	315.4 kbps	466.3 kbps	135	101	0	0
R	ISP-2	Ethernet	1524	0 bps	0 bps	0	0	0	0
R	LOKAL	Ethernet	1524	474.9 kbps	291.9 kbps	142	132	0	0
X	ether1	Ethernet	1526	0 bps	0 bps	0	0	0	0
X	ether5	Ethernet	1524	0 bps	0 bps	0	0	0	0

Gambar 4.12 Teknik *fail over*

4.7.2 Mengganti *Username* dan *Password* untuk Perangkat Jaringan

Pengubahan *username* dan *password* dimaksud agar *client* tidak dapat mengakses perangkat jaringan dengan menggunakan *username* dan *password* yang umum. Disini penulis akan mengubah *username* dan *password* dari mikrotik dan modem *router*.

1. Pada Mikrotik

- a) Buka winbox lalu *login* dengan *user*: “admin” dan tanpa *password*.
- b) Masuk ke menu *system > password*.
- c) Isikan *password* baru yaitu “admin123”

Password

Old Password:

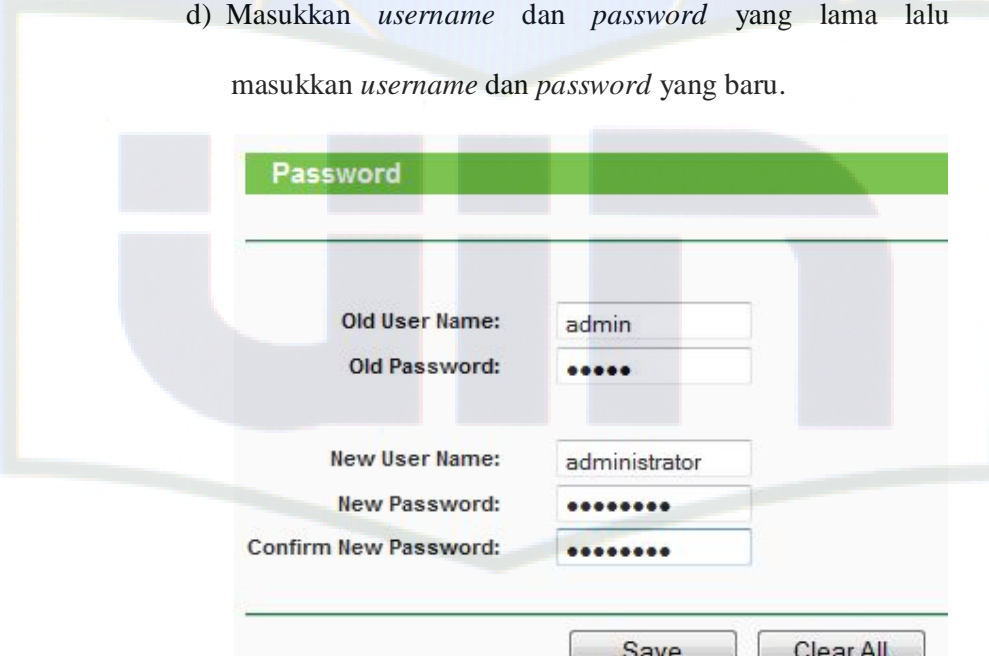
New Password:

Confirm Password:

OK Cancel

Gambar 4.13 Mengganti *password* di mikrotik

2. Pada Modem *router*
 - a) Buka *browser* dan masukkan *IP address* dari modem *router* yaitu 192.168.1.2.
 - b) Masukkan *username*: “admin” dan *password*: “admin”.
 - c) Masuk ke menu *System Tools* > *Password*.
 - d) Masukkan *username* dan *password* yang lama lalu masukkan *username* dan *password* yang baru.



The screenshot shows a web interface for changing the password of a modem router. At the top, there is a green header bar with the word "Password" in white. Below this, the interface is divided into two main sections. The first section, labeled "Old User Name:" and "Old Password:", contains two input fields. The "Old User Name" field is filled with the text "admin", and the "Old Password" field is filled with seven dots. The second section, labeled "New User Name:", "New Password:", and "Confirm New Password:", contains three input fields. The "New User Name" field is filled with the text "administrator", the "New Password" field is filled with seven dots, and the "Confirm New Password" field is also filled with seven dots. At the bottom right of the form, there are two buttons: "Save" and "Clear All".

Gambar 4.14 Mengganti *password* di modem *router*

BAB V

PENUTUP

Setelah melakukan serangkaian penelitian, maka pada bab ini penulis akan menguraikan kesimpulan yang dapat diambil dari rangkaian penelitian tersebut. Selain kesimpulan, penulis juga memberikan saran yang akan bermanfaat bagi pihak-pihak yang akan melanjutkan pengembangan penelitian ini.

5.1 KESIMPULAN

Kesimpulan yang diperoleh setelah melakukan tahapan-tahapan pada penelitian adalah sebagai berikut:

1. Penerapan *Nth load balancing* telah memberikan *bandwidth* yang optimal, namun *load balancing* tidak dapat mengakumulasi besar *bandwidth* kedua koneksi, karena teknik *load balancing* bukan berarti $1+1=2$ melainkan $1+1=1+1$.
2. Penerapan *Nth load balancing* telah membagi beban *traffic* secara seimbang pada ISP 1 dan ISP 2 pada laboratorium komputer SMK PGRI Bekasi.
3. Penerapan teknik *fail over* dapat menjadikannya salah satu *gateway* sebagai koneksi tunggal jika *gateway* yang lain dalam keadaan mati.

5.2 SARAN

Berdasarkan kesimpulan-kesimpulan yang telah dikemukakan, dapat diajukan beberapa saran untuk pengembangan lebih lanjut, antara lain:

1. Untuk meningkatkan kecepatan dan penghematan bandwidth, dapat menambahkan *external proxy server* dengan menggunakan *Squid*.
2. Memfungsikan fitur lain dari mikrotik yaitu *bandwidth management* agar dapat membagi secara rata *bandwidth* sesuai jumlah client yang aktif.
3. Dalam pemilihan ISP, diusahakan yang memiliki kualitas *bandwidth* dan *connection speed* yang hampir sama agar dalam *browsing* tidak terjadi koneksi yang lambat dikarenakan *response time* yang berbeda pada tiap ISP.

DAFTAR PUSTAKA

- Dewobroto, Pujo. 2009. *Load Balance menggunakan Metode PCC*. [Online].
Tersedia: http://www.mikrotik.co.id/artikel_lihat.php?id=34
- Gu, Paul. 2008. *Improved Load Balancing over Multiple Gateways*. [Online].
Tersedia:
http://wiki.mikrotik.com/wiki/Improved_Load_Balancing_over_Multiple_Gateways
- Herlambang, M. Linto dan Azis Catur L. 2008. *Panduan Lengkap Menguasai Router Masa Depan Menggunakan Mikrotik RouterOS*. Yogyakarta: ANDI.
- Megis, Janis dan Riyadi, Valens. 2010. *Load Balance with Masquerade Network on RouterOS*. [Online]. Tersedia:
<http://mum.mikrotik.com/presentations/PL10/balancing.pdf>
- Nugroho, Bunafit. 2005. *Instalasi & Konfigurasi Jaringan Windows dan Linux*. Yogyakarta: ANDI.
- Saputro, Daniel T. & Kustanto. 2008. *Membangun Server Internet dengan Mikrotik*. Yogyakarta: Gava Media.
- Sopandi, Dede. 2008. *Instalasi dan konfigurasi Jaringan Komputer*. Bandung: Informatika Bandung.
- Stiawan, Deris. 2009. *Fundamental Internetworking Development & Design Life Cycle*. [Online]. Tersedia:
http://deris.unsri.ac.id/materi/jarkom/network_development_cycles.pdf

Stiawan, Deris. 2009. *Mengenal Wan dan Solusi Load Balancing*. [Online].

Tersedia:

http://deris.unsri.ac.id/materi/jarkom/Solusi%20Load%20Balancing_web.pdf

Sugeng, Winarno. 2006. *Jaringan Komputer dengan TCP/IP*. Bandung:

Informatika Bandung.

Sutanta, Edhy. 2005. *Pengantar Teknologi Informasi*. Yogyakarta: GRAHA

ILMU.

Syafrizal, Melwin. 2007. *Pengantar Jaringan Komputer*. Yogyakarta: ANDI.

Wardhana, Asoka. 2006 *Modul Basic mikrotik Router OS*. [Online]. Tersedia:

http://mhs.stiki.ac.id/05113940/Basic_Mikrotik_Router_OS.pdf

Wijaya, Hendra. 2008. *Belajar Sendiri VMware Workstation*. Jakarta: Elex Media

Komputindo

Zul Fajri, EM dan Arilia Senja, Ratu. 2004. *Kamus Lengkap Bahasa Indonesia*



**KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI (UIN)
SYARIF HIDAYATULLAH JAKARTA
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Ir. H. Juanda No. 95 Ciputat 15412 Indonesia
Telp.: (62-21) 7493606, 7493547 Fax.: (62-21) 7493315

Email : uinjkt@cabi.net.id
Website : http://www.fst.uinjkt.ac.id

Nomor : Un..01/F9/TL.00/2649/2010
Lamp : -
Hal : Bimbingan Skripsi

Jakarta, 12 Nopember 2010

Kepada Yth Sdr
1. Victor Amrizal, M.Kom
2. Andrew Fiade, M.Kom

Assalamu'alaikum Wr. Wb.

Dengan ini diharapkan kesediaan saudara untuk menjadi Pembimbing I/II/
(Materi/Teknis)* penulisan Skripsi Mahasiswa :

Nama : Andri Dwi Utomo
NIM : 106091002904
Program Studi : Teknik Informatika
Judul Skripsi : **"Implementasi Manajemen Bandwith dan Load Balancing Dual
ISP Dengan Menggunakan Mikrotik "**

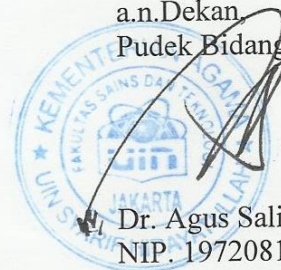
Judul tersebut telah disetujui oleh program studi yang bersangkutan pada tanggal 13 Juni 2010 dengan outline, abstraksi dan daftar pustaka terlampir. Bimbingan skripsi ini diharapkan selesai dalam waktu 6 (enam) bulan.

Apabila judul tersebut terjadi perubahan setelah dikonsultasikan dengan pembimbing harap segera dilaporkan ke Program Studi.

Demikian atas kesediaan saudara kami ucapkan terima kasih.

Wassalamu'alaikum Wr. Wb.

a.n.Dekan
Pudek Bidang Akademik



Dr. Agus Salim, M.Si
NIP. 19720816 199903 1 003

Tembusan :
1. Dekan (sebagai laporan)
2. Ketua Program Studi
3. Mahasiswa yang bersangkutan
* coret yang tidak perlu



YAYASAN PEMBINA LEMBAGA PENDIDIKAN DASAR DAN MENENGAH
PGRI PROPINSI JAWA BARAT

SMK PGRI RAWALUMBU KOTA BEKASI

Jl. Makrik Rt. 007 Rw. 04 Bojong Rawalumbu – Bekasi, Telp. 021- 683 624 04, 021-824 358 14
Jl. Raya Narogong Km. 5 ASRAMA YONIF 202 - BEKASI Telp. (021) 98278750, 68362404

SURAT KETERANGAN

Nomor. 143 / SATDIK-SMK / II.5 / G.11

Yang bertanda tangan di bawah ini, Kepala SMK PGRI Rawalumbu Kota Bekasi, menerangkan bahwa:

Nama : Andri Dwi Utomo
TTL : Jakarta, 13 Februari 1989
NIM : 106091002904
Jurusan : Teknik Informatika
Universitas : Universitas Islam Negeri Syarif Hidayatullah Jakarta

Telah melaksanakan riset di SMK PGRI Bekasi mulai tanggal 3 s.d 31 Januari 2011 dengan judul **"Implementasi Load Balancing Dua ISP menggunakan Mikrotik (Studi Kasus: Laboratorium Komputer SMK PGRI Bekasi)"**

Demikian surat keterangan ini kami berikan, agar dapat dipergunakan sebagaimana mestinya.



Bekasi, 04 Februari 2011
Kepala SMK PGRI Rawalumbu

G. Sutardi, M.M.Pd.

Catatan:

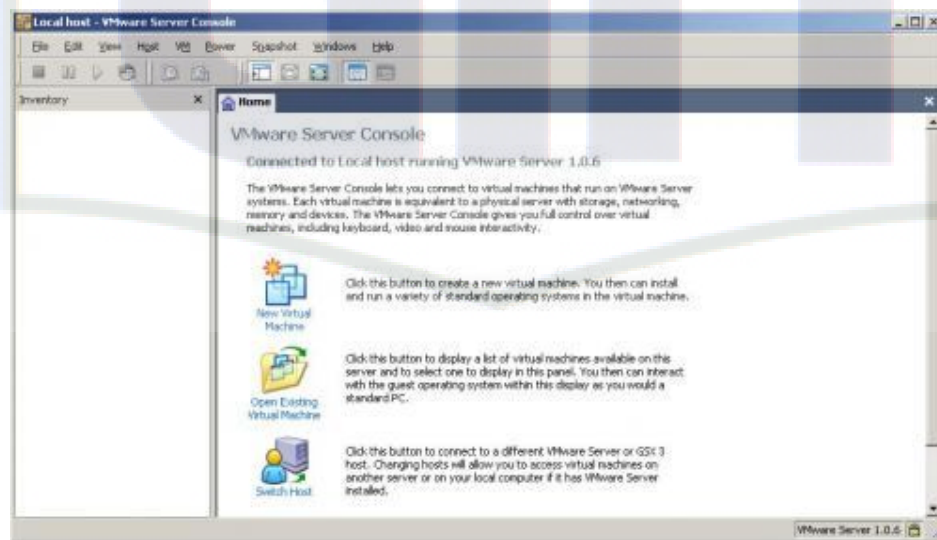
Fotokopi surat penelitian agar diserahkan
sekolah sebagai arsip

LAMPIRAN C

Simulasi di *VMware Workstation*

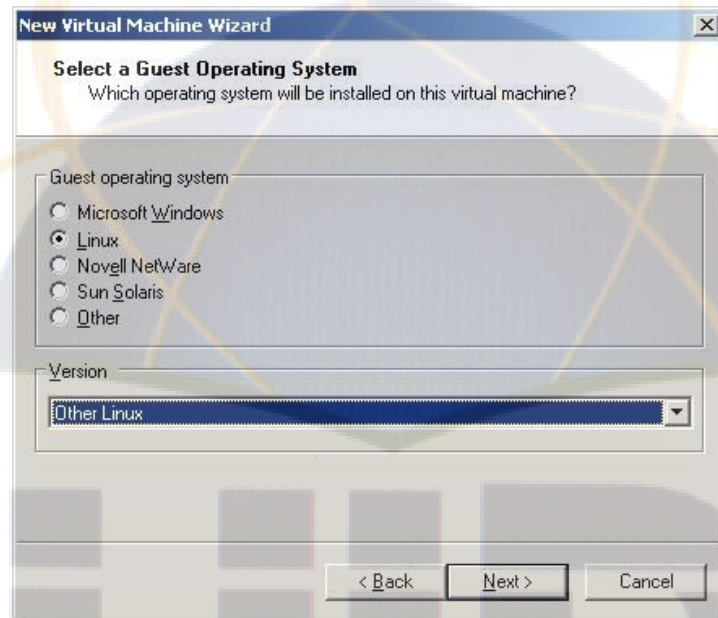
Pada simulasi ini penulis menginstall Mikrotik RouterOS pada mesin virtual menggunakan aplikasi *VMware Workstation*. Berikut ini adalah tahapan-tahapannya:

1. *Download file image* Mikrotik RouterOS. *File* tersebut bisa didapat di alamat berikut <http://www.mikrotik.co.id/getfile.php?nf=mikrotik-3.13.iso>.
2. Buka aplikasi *VMware Workstation*. Kemudian pilih *New Virtual machine*.



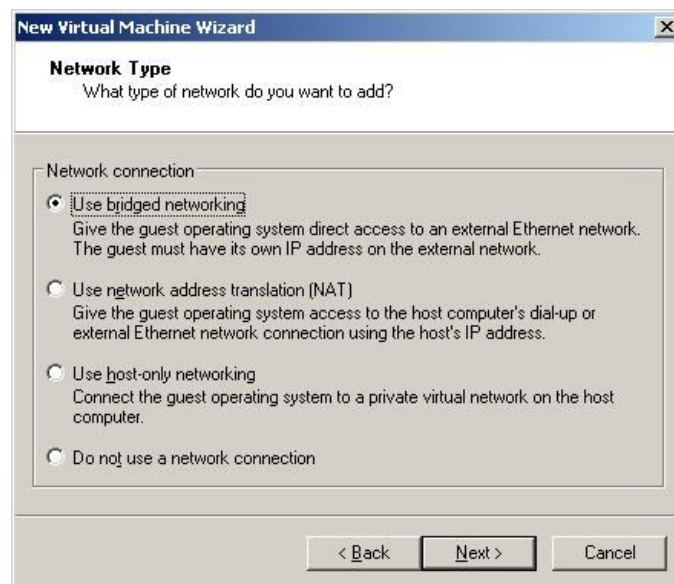
Gambar: Tampilan awal *VMware Workstation*

3. Dalam menu *virtual machine configuration*, pilih opsi *typical*.
4. Setelah itu, untuk menu *guest operating system*, pilih sistem operasi linux karena mikrotik merupakan OS turunan dari linux. Dan untuk versi pilih *other linux*.



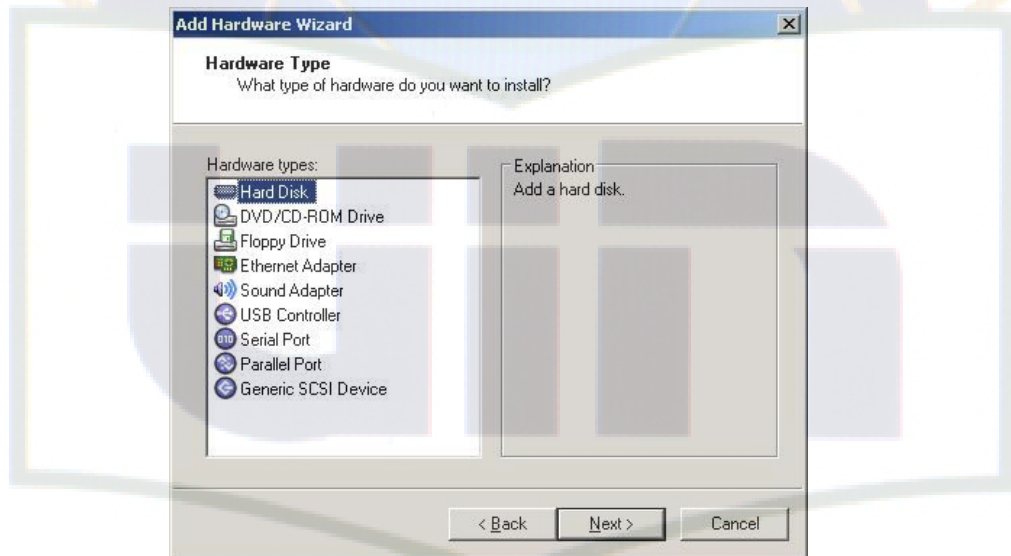
Gambar: Pemilihan *operating system*

5. Lalu beri nama dan lokasi tempat *virtual machine* akan disimpan.
6. Pada menu *network connection* pilih *bridge networking*. Karena penulis akan menghubungkan mikrotik di *virtual machine* dengan OS di PC yang terinstall *VMware Workstation*.



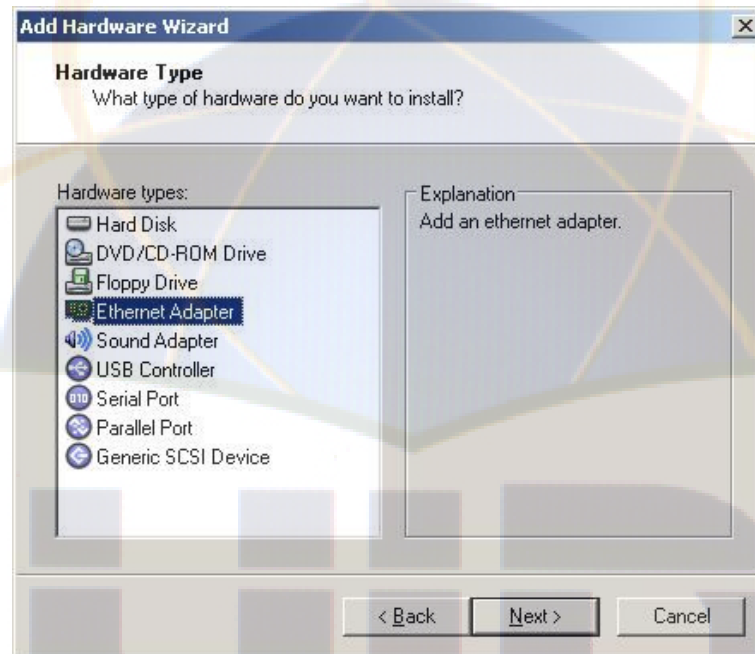
Gambar: Tipe *network connection*

7. Berikutnya harus ada pengaturan tambahan untuk *hardware* dan CD-Room. Secara *default*, pengaturan *harddisk* ini menggunakan tipe SCSI. Namun dalam pembuatan mikrotik, hanya dapat menggunakan *harddisk* IDE. Maka penulis menghapus *harddisk* tipe SCSI dan menambahkan *harddisk* IDE, yaitu dengan memilih menu *add hardware*.



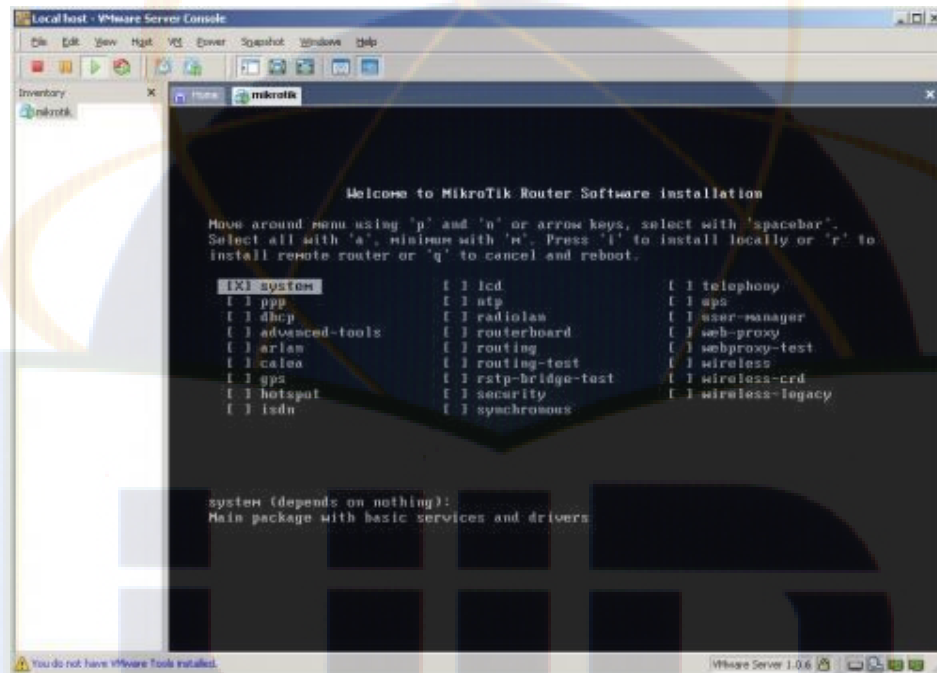
Gambar: Penambahan *harddisk* IDE

8. Supaya dapat berfungsi sebagai *router*, penulis menambahkan *hardware* lagi berupa *ethernet adapter*. Dan untuk *network connection* pilih *bridge networking*.



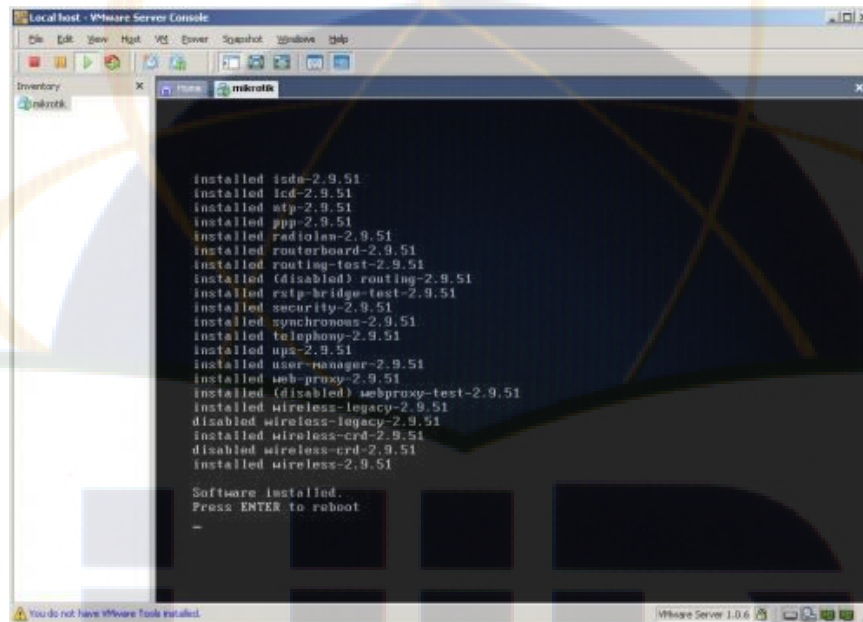
Gambar: Penambahan *ethernet adapter*

9. Untuk pengaturan mesin virtual untuk mikrotik RouterOS sudah selesai. Kali ini penulis akan memerintahkan untuk melakukan *booting* dari file Mikrotik.iso yang sudah *download* pertama kali, kemudian tekan tombol *play*. Berikut ini adalah tampilan jika berhasil *booting* dari *file image* mikrotik RouterOS.



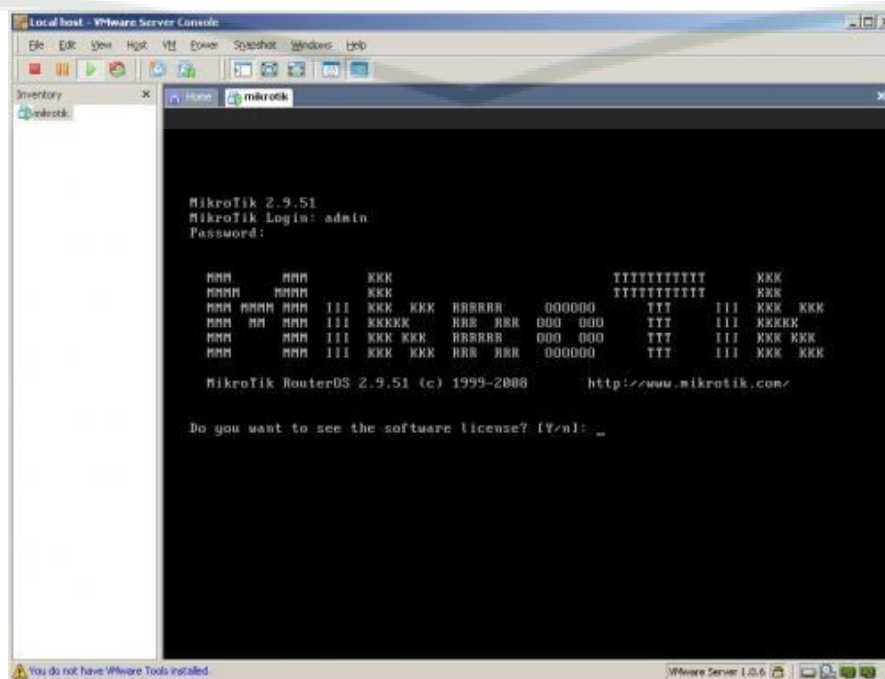
Gambar: Tampilan awal instalasi mikrotik

10. Gambar diatas adalah awal instalasi mikrotik. Pada tahap ini adalah pemilihan paket apa saja yang akan diinstall pada mikrotik yang akan dibuat. Untuk memilih satu per satu paket yang akan diinstall yaitu dengan menggunakan tombol spasi. Atau untuk menginstall semua paket bisa menekan tombol “a”. selanjutnya tekan tombol “i” untuk menginstall paket-paket yang telah dipilih.



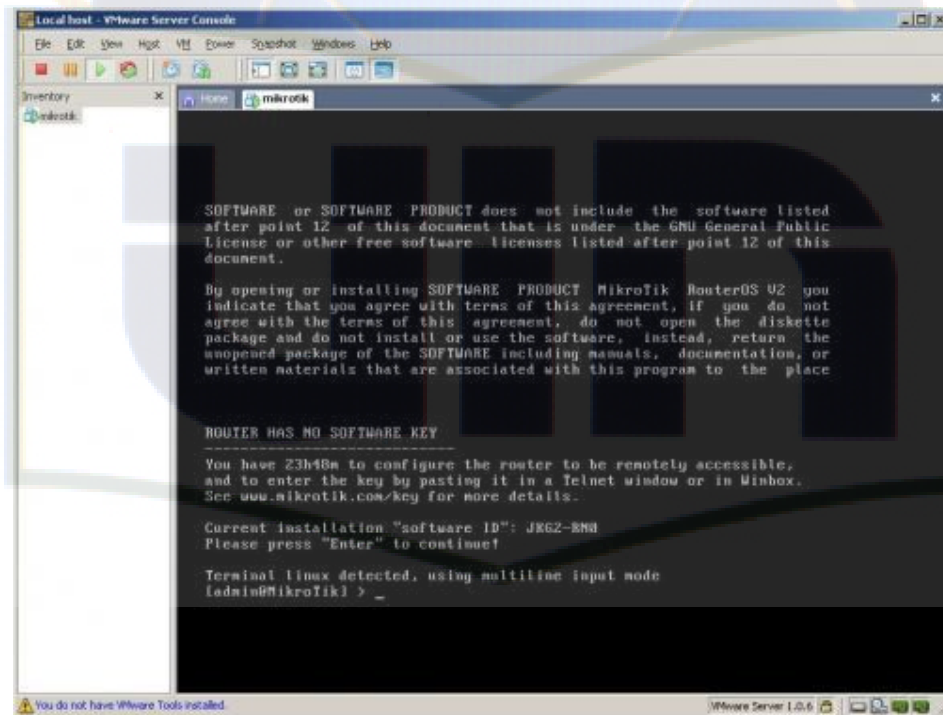
Gambar: Proses instalasi paket

11. Tunggu sekitar 15 menit instalasi akan selesai dan meminta *reboot*. Setelah *reboot* akan tampil seperti di bawah ini.



Gambar: login mikrotik

12. Login dengan user “admin” dan *password* dibiarkan kosong. kemudian tekan ‘Y’ untuk membaca lisensi dan tekan ‘q’ untuk quit dari lisensi.
13. Instalasi mikrotik RouterOS di *VMware Workstation* telah berhasil dan dapat langsung dikonfigurasi sesuai yang diinginkan untuk tahapan *simulation prototyping*.

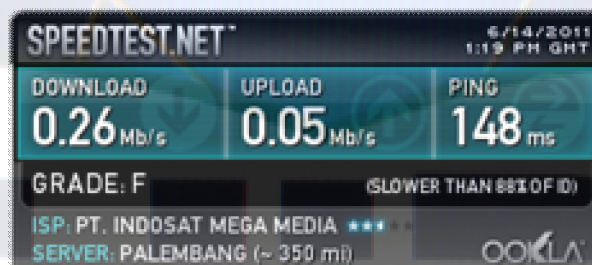


Gambar: tampilan *line console* mikrotik

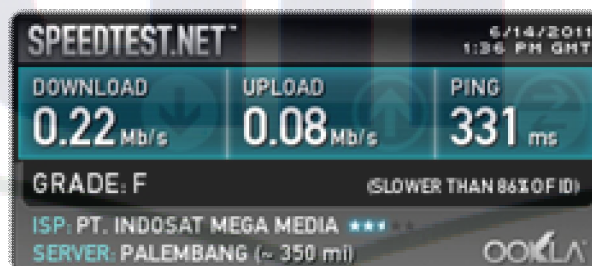
LAMPIRAN C

Gambar Pengujian Kecepatan Koneksi

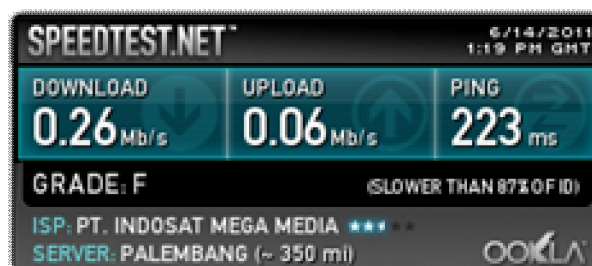
1. Pengujian kecepatan koneksi ISP IM2 sebelum diterapkan *load balancing*.



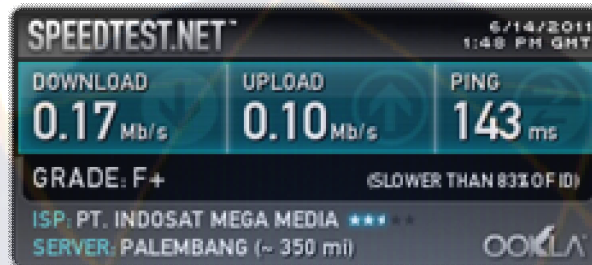
Gambar: Pengujian pertama ISP IM2



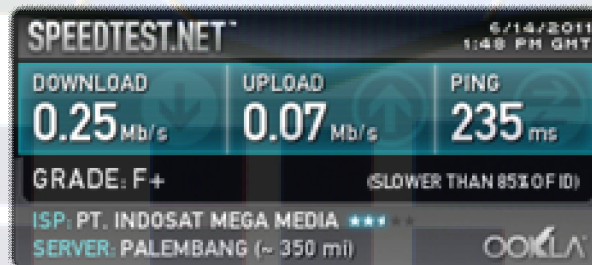
Gambar: Pengujian kedua ISP IM2



Gambar: Pengujian ketiga ISP IM2



Gambar: Pengujian keempat ISP IM2



Gambar: Pengujian kelima ISP IM2

2. Pengujian kecepatan koneksi ISP Smart sebelum diterapkan *load balancing*.



Gambar: Pengujian pertama ISP Smart



Gambar: Pengujian kedua ISP Smart



Gambar: Pengujian ketiga ISP Smart



Gambar: Pengujian keempat ISP Smart

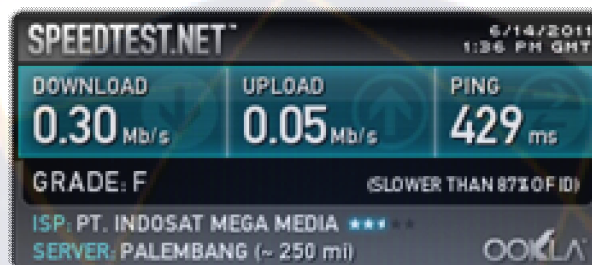


Gambar: Pengujian kelima ISP Smart

3. Pengujian kecepatan koneksi setelah diimplementasikan *load balancing*.



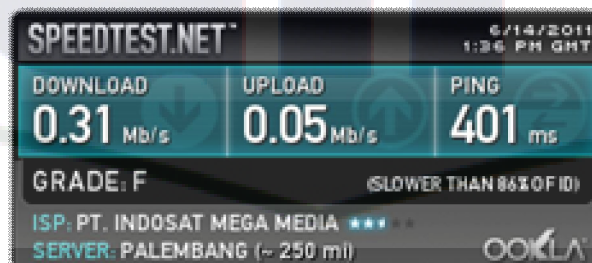
Gambar: Pengujian pertama *load balancing*



Gambar: Pengujian kedua *load balancing*



Gambar: Pengujian ketiga *load balancing*



Gambar: Pengujian keempat *load balancing*



Gambar: Pengujian kelima *load balancing*