TUGAS AKHIR - TE 141599

Implementasi *Border Gateway Protocol* (BGP) pada *Test Bed* IDREN (*Indonesian Research Education Network*)

Umi Faridah
NRP 0711 1645 000 021

Dosen Pembimbing
Dr. Ir. Achmad Affandi, DEA.
Ir. Djoko Suprajitno Rahardjo, M.T.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018

TUGAS AKHIR - TE 141599

Implementasi *Border Gateway Protocol* (BGP) pada *Test Bed* IDREN (*Indonesian Research Education Network*)

Umi Faridah
NRP 0711 1645 000 021

Dosen Pembimbing
Dr. Ir. Achmad Affandi, DEA.
Ir. Djoko Suprajitno Rahardjo, M.T.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018

---Halaman ini sengaja dikosongkan---

PERNYATAAN KEASLIAN
TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi keseluruhan tugas akhir saya dengan judul "**Implementasi *Border Gateway Protocol* (BGP) pada *Test Bed* IDREN (*Indonesian Research Education Network*).**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2018

Umi Faridah
NRP. 07111645000021

---Halaman ini sengaja dikosongkan---

Implementasi *Border Gateway Protocol* (BGP) pada *Test Bed* IDREN (*Indonesian Research Education Network*)

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Teknik Telekomunikasi
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I,	Dosen Pembimbing II,
Dr. Ir. Achmad Affandi, DEA.	Ir. Djoko Suprajitno Rahardjo, MT.
NIP. 196510141990021001	NIP. 195506221987011001

KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI ELEKTRO
DEPARTEMEN TEKNIK ELEKTRO

SURABAYA
JULI 2018

---Halaman ini sengaja dikosongkan---

IMPLEMENTASI *BORDER GATEWAY PROTOCOL* PADA *TEST BED* IDREN (*INDONESIAN RESEARCH EDUCATION NETWORK*)

Nama Mahasiswa : Umi Faridah
NRP : 07111645000021
Dosen Pembimbing I : Dr. Ir. Achmad Affandi, DEA.

Dosen Pembimbing II : Ir. Djoko Suprajitno Rahardjo, MT.

ABSTRAK

Indonesia memiliki *pivate network* yang bernama IdREN. Untuk dapat terhubung dengan jaringan IdREN, tiap perguruan tinggi harus mendaftar dan melakukan konfigurasi *peering* jaringan. Tidak semua perguruan tinggi memiliki *Autonomous System Number* dan terkadang dalam konfigurasi *peering* jaringan terdapat kesalahan konfigurasi sehingga perguruan tinggi yang ingin bergabung dengan jaringan IdREN belum bisa terhubung.

Pada tugas akhir ini dibuat *test bed* IdREN dengan menggunakan router mikrotik RB951-2n dan menggunakan *routing protocol* BGP. Untuk menghubungan perguruan tinggi yang tidak memiliki ASN, digunakan *routing protocol* internal BGP sedangkan untuk menghubungan perguruan tinggi yang memiliki ASN, digunakan *routing protocol* eksternal BGP. Atribut *nexthop* pada BGP digunakan untuk menentukan alamat *nexthop* langsung yang harus digunakan untuk meneruskan paket transit ke tujuan agar tidak terjadi *looping. Bidirectional Forwarding Detection* (BFD) digunakan untuk mendeteksi kesalahan dalam jalur dua arah pada internal BGP dengan cepat.

Dari hasil tabel *routing* pada *test bed* IdREN yang telah didapat, diketahui bahwa *test bed* IdREN yang telah dibuat dapat diakses oleh masing-masing perguruan tinggi dengan tanda flag DAb pada tabel *routing*-nya. Dari hasil konfigurasi dan pengujian menggunakan kabel *ethernet* cat5 yang dilakukan, diperoleh nilai *throughput* saat ITS melakukan *download file* ke *server* IdREN ialah 98.9 Mbps. Saat IAIN

Ponorogo melakukan *download file* ke *server* IdREN ialah 92.4 Mbps dan saat UGM mengakses *file* ke *server* IdREN ialah 98.9 Mbps.

Kata kunci: BGP, atribut *nexthop*, BFD, *Test bed* IdREN

IMPLEMENTATION OF BORDER GATEWAY PROTOCOL AT IDREN (INDONESIAN RESEARCH EDUCATION NETWORK) TEST BED.

Nama Mahasiswa : Umi Faridah
NRP : 07111645000021
Dosen Pembimbing I : Dr. Ir. Achmad Affandi, DEA.

Dosen Pembimbing II : Ir. Djoko Suprajitno Rahardjo, MT.

ABSTRACT

Indonesia has a pivate network called IdREN. In order to connect with the IdrEN network, each college must register and configure network peering. Not all colleges have an Autonomous System Number and sometimes in peering network configuration there is a configuration error so that the college who wants to join the IdrEN network can not connect yet.

In this final project is made test bed IdREN by using router mikrotik RB951-2n and using BGP routing protocol. To connect universities that do not have ASN, BGP internal routing protocol is used whereas to connect universities with ASN, BGP external routing protocol is used. The nexthop attribute in BGP is used to specify the direct nexthop address that should be used to forward the transit packet to the destination in order to avoid looping. Bidirectional Forwarding Detection (BFD) is used to detect errors in two-way lanes on BGP internals quickly.

From the results of the routing table on IdREN test bed that has been obtained, it is known that the test bed IdREN that has been made can be accessed by each college with the flag DAb on its routing table. From the results of configuration and testing using cat5 ethernet cable is done, obtained throughput value when ITS download file to IdREN server is 98.9 Mbps. When IAIN Ponorogo downloaded the file to the IdREN

server it is 92.4 Mbps and when UGM accesses the file to IdREN server it is 98.9 Mbps.

Keywords: BGP, nexthop attribute, BFD, Test bed IdREN

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga tugas akhir ini dapat terselesaikan tepat watu.

Tugas akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Sarjana pada Bidang Studi Telekomunikasi Multimedia, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember yang berjudul: "Implementasi *Border Gateway Protocol* pada *Test Bed* IdREN (*Indonesian Research Education Network*)". Pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

1. Orang tua saya yang telah memberikan dukungan dan doa dalam menyelesaikan Tugas Akhir ini
2. Bapak Dr. Ir. Achmad Affandi, DEA. dan Bapak Ir. Djoko Suprajitno Rahardjo, MT. selaku dosen pembimbing yang telah banyak membantu dan membagikan ilmu selama penelitian tugas akhir ini
3. Bapak Raga, Mas Fikry yang telah memberikan wawasan tentang IdREN
4. Dosen dan Teman-teman dari bidang studi Telekomunikasi Multimedia yang telah memberikan banyak wawasan dan *sharing* ilmu hingga saya dapat memahami mata kuliah sampai kelulusan
5. Pihak-pihak lain yang belum bisa penulis sebutkan satu per satu yang ikut membantu dalam penyelesaian tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan, oleh karena itu saran dan masukan sangat diharapkan untuk perbaikan di masa yang akan datang. Semoga tugas akhir ini bermanfaat bagi pembaca dan masyarakat pada umumnya.

Surabaya, Juli 2018

Umi Faridah

---Halaman ini sengaja dikosongkan---

DAFTAR ISI

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

---Halaman ini sengaja dikosongkan---

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Peneliti Indonesia maupun pelajar yang berada di luar negeri membutuhkan internet untuk bekerja sama dengan komunitas yang ada di Indonesia. Akan tetapi pada saat itu saluran komunikasi masih tergolong mahal. Dengan adanya hal tersebut, mereka mencoba untuk menemukan teknologi yang memungkinkan orang Indonesia agar terhubung ke internet. Pada tahun 2006, terdapat TEIN (*Trans Eurasia Information Networks*) yang menghubungkan NREN (*National Research Education Network*) di Eropa dan Amerika Serikat. Pada tahun yang sama juga, Indonesia memiliki jaringan dengan nama INHERENT (*Indonesia Higher Education Network*) yang menghubungkan 32 Universitas di Indonesia dan infrastrukturnya didukung oleh pemerintah. Pada tahun 2013, INHERENT sudah tidak aktif [1].

Berdasarkan UU nomor 12 Tahun 2012 Pendidikan Tinggi pasal 79 ayat 5 yang berisi bahwa pemerintah mengembangkan jejaring antar perguruan tinggi dengan memanfaatkan teknologi informasi, maka dibentuklah IdREN. IdREN merupakan bagian dari REN (*Research and Education Networks*) yang terhubung melalui TEIN. IdREN termasuk *private network* yang menghubungkan antar institusi riset dan pendidikan di Indonesia. IdREN memungkinkan institusi pendidikan di Indonesia untuk berbagi pakai sumber daya pembelajaran yang dimiliki melalui jalur yang lebih aman. Sumber daya tersebut berupa bahan perkuliahan, bahan pustaka, *software, network access* dan *journal online*. Perguruan Tinggi yang terhubung ialah ITB, UGM, UI, ITS dan UB. Jaringan IdREN didukung oleh PT. Telekomunikasi Indonesia, Tbk. Layanan dan aplikasi yang ada pada IdREN dapat diakses oleh *network* yang tergabung. IdREN mampu mendukung data dengan kecepatan tinggi dan menyediakan saluran khusus *untuk research sharing* antar Perguruan Tinggi.

Agar IdREN dapat terhubung dengan jaringan *research* & *education* di berbagai Negara, maka diperlukan *routing protocol* yang tepat. *Routing protocol* BGP dapat melakukan komunikasi internet dalam skala besar. BGP merupakan *routing protocol* yang disepakati untuk komunikasi internet antar *Internet Service Provider* (ISP) seluruh dunia. Setiap negara atau organisasi memiliki *Autonomous System Number* (AS) sebagai identitas untuk saling bertukar *routing*. *Backbone* IdREN sendiri melalui PT. Telekomunikasi Indonesia, Tbk telah memiliki skala jaringan *Wide Area Network* (WAN) sehingga *routing protocol* BGP digunakan untuk melakukan pertukaran informasi *routing* komunikasi internet dalam skala besar.

Untuk dapat terhubung dengan jaringan IdREN, tiap perguruan tinggi harus mendaftar dan melakukan konfigurasi *peering* jaringan. Tidak semua perguruan tinggi memiliki *Autonomous Number* dan terkadang dalam konfigurasi *peering* jaringan terdapat kesalahan konfigurasi sehingga perguruan tinggi yang ingin bergabung dengan jaringan IdREN belum bisa terhubung.

Berdasarkan uraian di atas, maka dibutuhkan konfigurasi *routing protocol* yang tepat agar perguruan tinggi yang ingin bergabung mendapatkan hak akses yang sama. Oleh karena itu, pada tugas akhir ini, akan dibuat *test bed* jaringan IdREN yang dapat untuk menghubungkan antar perguruan tinggi dengan *network-network address* yang memiliki *Autonomous System Number* yang berbeda dan yang sama sehingga dapat terkoneksi. Untuk menghubungkan perguruan tinggi dengan ASN yang sama dan yang berbeda digunakan *protocol routing* BGP (internal BGP dan eksternal BGP).

1.2 Rumusan Masalah

Permasalahan dalam tugas akhir ini adalah:

1. Diperlukan konfigurasi *peering* jaringan yang tepat untuk terhubung ke jaringan IdREN

1.3 Tujuan

Tujuan dari hasil penelitian tugas akhir ini adalah sebagai berikut:

1. Membuat *test bed* jaringan IdREN dengan menerapkan *routing protocol* BGP (internal BGP dan eksternal BGP)
2. Mengetahui kinerja *routing* BGP pada *test bed* IdREN.

1.4 Batasan Masalah

Adapun batasan masalah dalam penelitian tugas akhir ini, yaitu:

1. Tugas Akhir ini dibuat hanya sebagai *test bed* tanpa diimplementasi pada jaringan sebenarnya.
2. *Routing protocol* yang digunakan ialah *Border Gateway Protocol* (internal BGP dan eksternal BGP).
3. Konfigurasi dilakukan pada *testbed* GATE-ITS, GATE-UGM dan universitas yang terhubung dengan IdREN melalui GATE-ITS dan GATE UGM.

1.5 Metodologi

Metodelogi yang digunakan dalam Penulisan Tugas Akhir ini adalah sebagai berikut:

1 Studi Literatur

Melakukan pembelajaran mengenai literatur yang akan dibutuhkan untuk tugas akhir ini. Mencari, membaca dan mempelajari sumber dari jurnal, buku dan internet yang berhubungan dengan penerapan *routing protocol* BGP pada suatu jaringan.

2 Simulasi dan Perancangan

Perangkat yang digunakan untuk simulasi jaringan ialah mikrotik routerBOARD 9512n. Pada tahap ini dilakukan konfigurasi *router* untuk membuat sebuah *test bed* IdREN dengan menggunakan BGP sebagai *protocol routing*-nya.

3 Penulisan Buku TA.

Setelah *test bed* IdREN yang menggunakan *routing protocol* BGP sudah selesai dibuat, langkah selanjutnya ialah melakukan uji coba dengan cara meneliti setiap bagian dari sistem yang telah dibuat. Jika belum terhubung, maka langkah selanjutnya ialah mencari solusi

untuk mengatasi hal tersebut yakni dengan cara mengecek kembali konfigurasi yang telah dilakukan.

1.6 Relevansi

Diharapkan hasil pengimplementasian *Border Gateway Protocol* pada *test bed* IdREN ini dapat digunakan oleh pihak-pihak yang ingin melakukan *peering* jaringan untuk terhubung ke jaringan IdREN.

1.7 Sistematika Penulisan

Penulisan tugas akhir ini terdiri dari lima bab pembahasan sebagai berikut:

BAB I : PENDAHULUAN
Menguraikan latar belakang penelitian, rumusan masalah yang diangkat, tujuan, metode penelitian, sistematika penulisan, dan relevansi.

BAB II : TEORI PENUNJANG
Bab ini membahas tentang BGP dan IdREN.

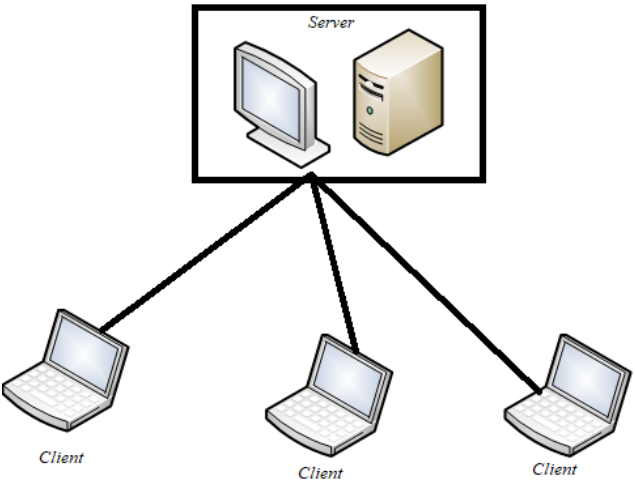
BAB III : PERENCANAAN DAN PEMBUATAN ALAT
Pada bab ini dijelaskan tentang rancangan sistem IdREN menggunakan *routing protocol* BGP (internal BGP dan eksternal BGP). Alat yang digunakan ialah mikrotik routerBOARD 9512n dan perangkat lunak yang digunakan winbox dan iperf.

BAB IV : HASIL SIMULASI DAN ANALISA DATA
Bab ini menjelaskan hasil *ping, traceroute* , *bandwidth* dan *throughput* pada *testbed* IdREN yang telah dibuat.

BAB 2
TINJAUAN PUSTAKA

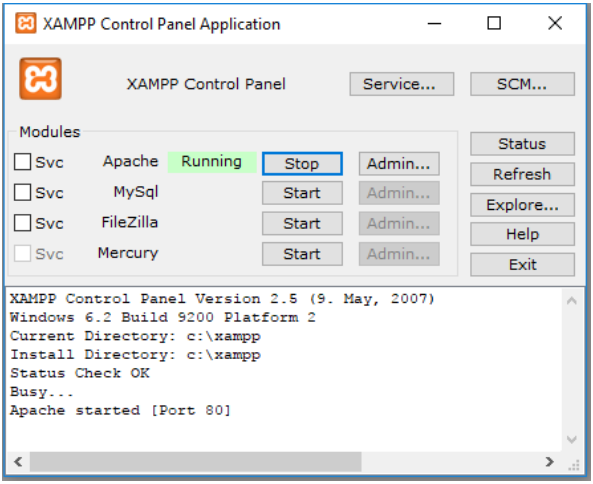
2.1 Jaringan Komputer

Jaringan komputer merupakan suatu sistem yang terdiri dari dua buah komputer atau lebih dan terdistribusi. Perangkat ini disebut sebagai elemen jaringan yang berperan sebagai *transmitter*. Elemen jaringan akan berkomunikasi satu sama lain melalui suatu medium dan menggunakan protokol tertentu yang disetujui oleh kedua perangkat dalam jaringan agar dapat saling berkomunikasi [2,3]. Berdasarkan fungsinya, jaringan dikelompokkan menjadi dua jenis yaitu jaringan *client server* dan jaringan *peer to peer*. Pada jaringan *client server*, terdapat satu komputer yang berfungsi sebagai *server* yang mengatur sistem dalam jaringan, sedangkan komputer lainnya bertindak sebagai *client*. Sedangkan jaringan *peer to peer* tidak memiliki *server* pusat yang mengatur *client-client*. Pada jaringan *peer to peer*, semua komputer bertindak sebagai *server* untuk komputer yang lain. Sehingga semua komputer bisa sebagai *server* sekaligus juga *cient*. Jaringan *client server* dapat dilihat seperti Gambar 2.1.



Gambar 2.1 Jaringan Komputer *Client Server*

Server merupakan sistem komputer yang mampu menyediakan suatu jenis layanan yang sudah ditentukan didalam sebuah jaringan tersebut. Sebagian dari kita menyebut *server* ini dengan sebutan *web server* yakni sebuah perangkat komputer yang bertindak sebagai *server* dan telah didukung dengan perangkat *processor* yang memiliki sifat *scalable* serta mempunyai kapasitas *Random Access Memory* (RAM) yang besar. Salah satu contoh *web serv*er adalah XAMPP. XAMPP ialah perangkat lunak bebas yang mendukung banyak sistem operasi dan *merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server* yang berdiri sendiri (*localhost*). XAMPP terdiri atas program *Apache* HTTP *Server*, MySQL *databas*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan *Perl*. Nama XAMPP merupakan singkatan dari X yang terdiri dari empat sistem operasi apapun yakni *Apache*, MySQL, PHP dan *Perl*. Program ini tersedia dalam *General Public License* (GNU) dan bersifat bebas. XAMPP merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman web yang *dinamis*. Gambar 2.2 adalah Gambar XAMPP *Control Panel Aplication Version* 2.5.



Gambar 2.2 XAMPP *Control Panel Aplication Version* 2.5

2.2 Perangkat Keras Jaringan Komputer

Perangkat keras jaringan komputer merupakan perangkat yang digunakan untuk mengubungkan dua atau lebih komputer yang ada di dalam jaringan komputer agar setiap komputer yang terhubung dapat saling berbagi data, *file*, dan sumber daya yang lain. Seperti halnya komputer, sebuah jaringan komputer bisa beroperasi dengan didukung oleh *software* dan *hardware*. Perangkat keras jaringan komputer antara lain:

1 *Network Interface Card* (NIC) ialah sebuah kartu yang berfungsi sebagai jembatan dari komputer ke sebuah jaringan komputer. NIC biasa disebut sebagai *network adapter*. Setiap NIC memiliki alamat yang disebut MAC *address* bersifat statis dan dapat diubah oleh penggunanya.

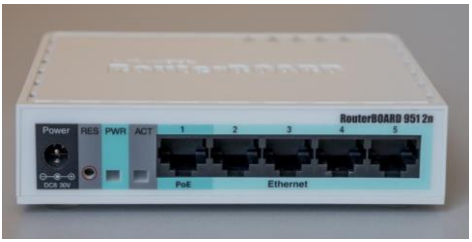
2 *Repeater* merupakan sebuah peralatan jaringan yang berfungsi untuk menangkap sinyal dan mentransmisikan kembali sinyal tersebut dengan kekuatan yang lebih tinggi sehingga sinyal tersebut dapat menempuh jarak yang lebih jauh.

3 *Hub* merupakan *central connection point* pada suatu jaringan. *Hub* tidak memiliki fasilitas *routing* sehingga semua data yang datang akan di-*broadcast* ke semua perangkat yang terhubung padanya. *Hub* ada dua macam yaitu *active hub* dan *passive hub*. *Active hub* juga bertugas sebagai *repeater* sedangkan *passive hub* hanya berfungsi untuk mentransmisikan sinyal ke jaringan.

4 *Bridge* merupakan sebuah komponen jaringan yang digunakan untuk memperluas jaringan atau membuat sebuah segmen jaringan. *Bridge* dapat digunakan untuk menggabungkan dua buah arsitektur jaringan yang berbeda misalnya antara *Token Ring* dan *Ethernet*. *Bridge* tidak melakukan konversi terhadap protokol sehingga agar dua segmen jaringan yang dikoneksikan ke *bridge* tersebut dapat terkoneksi, kedua jaringan tersebut harus mempunyai protokol jaringan yang sama.

5 *Router* berfungsi untuk menghubungkan *network* yang satu dengan *network* yang lain dan memilih jalur yang terbaik (*routing*) untuk mengirimkan paket data yang datang dari satu *port* ke *port* yang dituju paket data tersebut. *Router* mengirimkan paket data berdasarkan alamat IP. *Router* merupakan sebuah alat (*dedicated*) atau berupa aplikasi yang berfungsi untuk memutuskan pada titik manakah paket data harus diteruskan. *Router* biasanya terletak pada *gateway* suatu jaringan. *Router* dapat menghubungkan dua jaringan

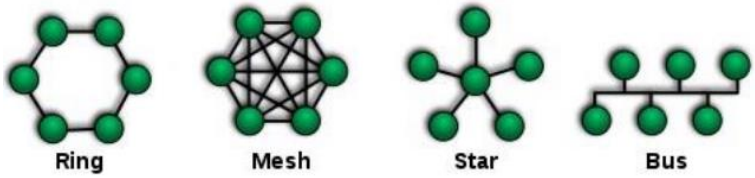
yang berbeda dengan *subnet* yang berbeda. *Router* memiliki *routing table*. *Routing table* merupakan sebuah daftar dari rute yang tersedia dan mampu memilih rute terbaik untuk sebuah paket data. Gambar 2.3 merupakan Gambar *Router* Mikrotik RB951-2n. Mikrotik *routerBOARD* RB951-2n memiliki semua kebutuhan router dan gateway untuk personal dan kantor. Memiliki 5 buah *port ethernet*, 1 buah *access point embedded* 2,4 GHz dan *antenna embedded* 1,5 dbi. Mikrotik *routerBOARD* RB951-2n juga sudah dilengkapi *power adaptor*.



Gambar 2.3 Mikrotik *routerBOARD* RB951-2n

2.3 Topologi Jaringan

Topologi jaringan ialah suatu cara untuk menghubungkan komputer satu dengan komputer yang lain sehingga membentuk suatu jaringan seperti topologi *bus*, *ring*, *star* dan *mesh* seperti yang terlihat pada Gambar 2.4.



Gambar 2.4 Topologi Jaringan [2]

1. Topologi *Bus*

Topologi ini topologi yang berbentuk seperti bus. Sepanjang kabel yang terbentang seperti bus tersebut memiliki *node-node* yang saling

terhubung dengan komputer lain. Sinyal yang lewat dalam kabel ini hanya satu arah [2,3].

2. Topologi *Ring*

Topologi jaringan yang berupa lingkaran tertutup seperti lingkaran dan berisi *node-node* yang terhubung. Sinyal yang lewat dalam kabel ini mengalir secara dua arah [2,3].

3. Topologi *Star*

Pada topologi ini tiap *node* yang ingin terhubung dengan *node* lain harus melalui *central node* seperti *switch* atau *hub*. Jika terdapat salah satu *node* terputus, maka *node* lainnya tidak akan terpengaruh [2,3].

4. Topologi *Mesh*

Pada topologi ini seluruh *node* yang ada saling terhubung sehingga jika terdapat satu *node* yang mati terdapat jalur alternatif lain yang dapat dilewati. Topologi ini sangat cocok untuk kondisi jaringan yang memiliki *traffic* tinggi [2,3].

2.4 Tipe Jaringan Berdasarkan Area

Jaringan komputer dapat dibagi menjadi 4 kategori berdasarkan luas area antara lain: *Personal Area Network, Local Area Network, Metropolitan Area Network dan Wide Area Network* [4].

1. *Personal Area Network* (PAN)

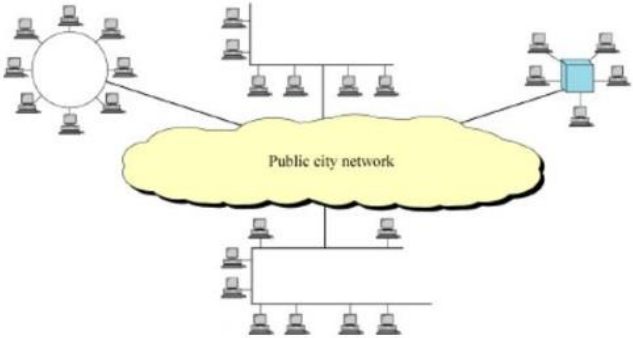
   Personal Area Network merupakan jaringan komputer dalam jarak sangat dekat dan yang diorganisir oleh satu orang dalam suatu bangunan, di dalam kantor maupun di rumah. Biasanya WAN terdiri dari satu komputer, telepon, konsol *video game*, atau perangkat lain dan dihubungkan melalui *bus* yang ada pada komputer, seperti USB dan *Firewire*. Jika terdapat lebih dari satu orang yang menggunakan jaringan PAN dalam satu rumah, maka jaringan tersebut menjadi *home area network* (HAN).

2. *Local Area Network* (LAN)

   Local Area Network (LAN) merupakan jaringan komputer yang dengan skala kecil seperti dalam suatu gedung atau ruangan dalam sekolah. Biasanya jangkauan area LAN hanya mencapai 200 meter. LAN sangat berguna untuk saling berbagi *resource* dalam suatu gedung seperti berbagi *printer*. LAN dapat dibangun dengan peralatan yang tidak terlalu mahal seperti *hub* dan kabel *ethernet*. Jika suatu LAN dihubungkan dengan beberapa LAN yang lain, maka akan terbentuk suatu jaringan yang lebih besar disebut *Metropolitan Area Network* (MAN).

3. *Metropolitan Area Network* (MAN)

   Metropolitan Area Network (MAN) merupakan jaringan komputer yang terdiri dari beberapa jaringan LAN. Biasanya jaringan MAN yang dibangun terdiri dari beberapa jaringan antar gedung yang terhubung seperti pada Gambar 2.5.



Gambar 2.5 MAN [5]

4. *Wide Area Network* (WAN)

   Wide Area Network (WAN) merupakan jaringan yang memiliki skala sangat luas, bisa mencakup jaringan antar negara bahkan hingga benua. WAN biasanya sudah terhubung menggunakan media *nirkabel* seperti satelit atau kabel serat optik. Salah satu contoh WAN ialah IdREN yang merupakan *National Research and Education Network* (NREN) milik Indonesia.

2.5 IPv4

IPv4 merupakan sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP yang menggunakan protokol IP versi 4. Panjang IPv4 adalah 32-bit dan secara teoritis dapat mengalamati 4.294.967.296 *host* komputer yang ada di seluruh dunia. Alamat IPv4 pada awalnya ialah sederet bilangan biner sepanjang 32 bit yang digunakan untuk mengidentifikasikan *host* yang ada pada jaringan. Alamat IPv4 ini diberikan secara unik pada masing-masing komputer atau *host* yang terhubung ke internet. Prinsip kerja IPv4 adalah *packet* yang membawa data dimuati alamat IP dari komputer pengirim data kepada alamat IP pada komputer yang akan dituju, kemudian data tersebut dikirim ke jaringan. Packet ini kemudian dikirim dari *router* satu ke *router* yang lain dengan berpedoman pada alamat IP tersebut menuju ke

komputer yang dituju. Seluruh komputer atau *host* yang tersambung ke internet dibedakan hanya berdasarkan alamat IPv4 ini. Oleh karena itu, komputer atau host yang terhubung tidak boleh memiliki alamat IP yang sama jika terhubung ke jaringan internet [6,7].

2.5.1 Representasi Alamat IPv4

Alamat-alamat IPv4 panjangnya 32 bit dan dibagi menjadi dua identifikasi yakni sebagai berikut [7,8]:

1. Bagian identifikasi pada *network* ID menunjukkan identitas jaringan komputer tempat *host-host* atau komputer dihubungkan.
2. Bagian identifikasi pada *host* ID memberikan suatu pengenal yang unik pada setiap host atau komputer pada suatu jaringan komputer.

Alamat IPv4 biasanya ditulis dalam notasi desimal bertitik (*dotted-desimal notation*), yang dibagi ke dalam empat buah oktet dan memiliki ukuran 8 bit. Karena setiap oktet berukuran 8 bit, maka nilainya ialah mulai dari 0 hingga 255. Pengalamatan IPv4 menggunakan 32 bit dan setiap bit dipisahkan dengan notasi titik. Contoh notasi pengalamatan IPv4 ialah sebagai berikut:

FFFFFFFF.FFFFFFFF.FFFFFFFF.FFFFFFF

Nilai F dirubah menjadi nilai biner yakni 1 dan 0 seperti 11000000.10101000.00000010.00000011. Sehingga jika dirubah dalam desimal menjadi 192.168.2.3.

2.5.2 Jenis-jenis Alamat IPv4

Alamat IPv4 terbagi menjadi tiga jenis antara lain:

1. Alamat *Unicast* merupakan alamat IPv4 yang ditentukan untuk sebuah *interface* jaringan yang dihubungkan ke sebuah *internetwork* IP. Alamat *unicast* biasanya digunakan dalam komunikasi *point-to-point* atau *one-to-one*. Setiap *interface* jaringan yang menggunakan protokol TCP/IP harus diidentifikasikan dengan menggunakan sebuah alamat logis yang unik. Alamat logis unik ini biasa disebut dengan alamat *unicast*. Alamat *unicast* disebut sebagai alamat logis karena alamat ini merupakan alamat yang diterapkan pada lapisan jaringan dalam DARPA *Reference Model* dan tidak memiliki hubungan yang langsung dengan alamat yang digunakan pada lapisan *interface* jaringan dalam DARPA *Reference Model*. Alamat unicast dapat ditetapkan ke sebuah *host* dengan *interface* jaringan dengan menggunakan teknologi *ethernet* yang memiliki

alamat MAC sepanjang 48-*bit*. Alamat *unicast* inilah yang digunakan oleh semua *host* TCP/IP agar dapat saling terhubung. Komponen alamat *unicast* terbagi menjadi dua jenis yakni host identifier dan network identifier. Alamat *unicast* menggunakan kelas A, B, dan C dari kelas-kelas alamat IP versi 4 sehingga ruang alamatnya adalah dari 1.x.x.x hingga 223.x.x.x. Sebuah alamat *unicast* dibedakan dengan alamat yang lain dengan menggunakan skema *subnet mask*. Jika ada sebuah jaringan kecil tidak yang terkoneksi ke internet, semua alamat IP dalam ruangan kelas alamat *unicast* dapat digunakan. Jika koneksi dilakukan secara langsung yakni dengan menggunakan teknik *routing* atau secara tidak langsung yakni dengan menggunakan *proxy server*, maka ada dua jenis alamat yang dapat digunakan di dalam internet, yaitu:

a. *Public address* merupakan alamat-alamat yang telah ditetapkan oleh InterNIC dan berisi beberapa buah *network identifier* yang telah dijamin unik sehingga tidak ada dua *host* yang menggunakan alamat yang sama jika jaringan kecil tersebut telah terhubung ke internet.

   Intranet-intranet pribadi yang tidak dikoneksikan ke internet dapat menggunakan alamat publik yang telah ditetapkan oleh InterNIC. Jika sebuah organisasi selanjutnya memutuskan untuk menghubungkan intranetnya ke internet, skema alamat yang digunakan mungkin mengandung alamat-alamat yang mungkin telah ditetapkan oleh InterNIC atau organisasi lainnya. Alamat-alamat tersebut dapat menjadi konflik antara satu dan lainnya, sehingga disebut juga dengan *illegal address*, yang tidak dapat dihubungi oleh *host* lain.

b. *Private address* merupakan alamat yang setiap *node* IP-nya membutuhkan sebuah alamat IP yang secara global unik terhadap *internetwork* IP. *Private address* digunakan untuk *host-host* di dalam sebuah organisasi yang tidak membutuhkan akses langsung ke internet. *Private address* ditentukan di dalam RFC 1918 dan didefinisikan di dalam tiga blok alamat yakni:

   - 10.0.0.0/8

     Private network 10.0.0.0/8 merupakan sebuah *network identifier* kelas A yang mengizinkan alamat

IP yang *valid* dari 10.0.0.1 hingga alamat 10.255.255.254. *Private network* 10.0.0.0/8 memiliki 24 *bit host* yang dapat digunakan untuk skema *subnetting* di dalam sebuah organisasi privat.

- 172.16.0.0/12

  Private network 172.16.0.0/12 merupakan sebuah blok dari 16 *network identifier* kelas B atau sebuah ruangan alamat dengan memiliki 20 *bit* sebagai *host identifier* yang dapat digunakan dengan menggunakan skema *subnetting* di dalam sebuah organisasi privat. *Private network* 172.16.0.0/12 mengizinkan alamat-alamat IP yang valid dari 172.16.0.1 hingga 172.31.255.254.
- 192.168.0.0/16

  Private network 192.168.0.0/16 dapat merupakan blok dari 256 *network identifier* kelas C dan memiliki 16 *bit* sebagai *host identifier* yang dapat digunakan dengan menggunakan skema *subnetting* apapun di dalam sebuah organisasi privat. *Private network* 192.168.0.0/16 dapat mendukung alamat-alamat IP yang valid mulai dari 192.168.0.1 hingga 192.168.255.254.

  Kemudian ada juga sebuah ruang alamat yang digunakan untuk alamat IP privat dalam beberapa sistem operasi yakni 169.254.0.0/16. Alamat jaringan ini dapat digunakan sebagai alamat privat karena *Internet Assigned Numbers Authority* (IANA) memang mengalokasikan IP ini untuk tidak digunakan dalam IP *public*. Alamat IP ini adalah dari 169.254.0.1 hingga 169.254.255.254 dengan *subnet mask* 255.255.0.0. Alamat ini digunakan sebagai alamat IP privat otomatis dalam Windows yang disebut dengan *Automatic Private Internet Protocol Addressing* (APIPA).

  Karena alamat-alamat IP di dalam *private network* tidak akan ditetapkan oleh *Internet Network Information Center* (InterNIC) atau badan lainnya yang memiliki otoritas sebagai *public network*, maka

tidak akan pernah ada rute yang menuju *private network* tersebut di dalam *router* Internet. Oleh karena itu, semua lalu lintas dari sebuah *host* yang menggunakan sebuah *private network* harus mengirim *request* tersebut ke sebuah *gateway* seperti *proxy server*yang memiliki sebuah *public network* yang *valid* atau memiliki *private network* yang sudah ditranslasikan ke dalam sebuah alamat IP publik yang valid dengan menggunakan *Network Address Translator* (NAT) sebelum dikirimkan ke internet.

2. Alamat *Broadcast* merupakan alamat IPv4 yang didesain agar dapat diproses oleh setiap *node* IP dalam segmen jaringan yang sama. Alamat *broadcast* digunakan dalam komunikasi *one-to-everyone*. Jika sebuah *host* pengirim yang hendak mengirimkan paket data dengan tujuan alamat *broadcast*, maka semua *node* yang terdapat pada segmen jaringan tersebut akan menerima paket tersebut dan memprosesnya. Alamat IP *broadcast* hanya dapat digunakan sebagai alamat tujuan saja sehingga tidak dapat digunakan sebagai alamat sumber. Ada empat buah jenis alamat IP *broadcast* yaitu:
    a. N*etwork broadcast* IP versi 4 adalah alamat yang dibentuk dengan cara mengatur semua *bit host* menjadi 1 dalam sebuah alamat yang menggunakan kelas. Misalnya jika NetID-nya ialah 131.107.0.0/16 maka alamat *broadcast*-nya adalah 131.107.255.255. Alamat *network broadcast* digunakan untuk mengirimkan sebuah paket untuk semua *host* yang terdapat di dalam sebuah *network* yang berbasis kelas. *Router* tidak akan meneruskan paket-paket yang ditujukan dengan alamat *network broadcast*. *Network broadcast* tidak terdapat di dalam sebuah *network* yang tidak menggunakan kelas alamat IP.
    b. *Subnet broadcast* adalah alamat yang dibentuk dengan cara mengatur semua *bit host* menjadi 1 dalam sebuah alamat yang tidak menggunakan kelas. Misalnya jika NetID-nya ialah 131.107.26.0/24 maka alamat *broadcast*-nya ialah 131.107.26.255. Alamat *subnet broadcast* digunakan untuk mengirimkan sebuah paket ke semua *host* dalam sebuah jaringan yang telah dibagi dengan cara *subnetting* atau *supernettin*g. *Router* tidak akan meneruskan paket-paket yang

ditujukan dengan alamat *subnet broadcast*. Alamat *subnet broadcast* tidak terdapat di dalam sebuah *network* yang menggunakan kelas alamat IP.

c. *All-subnets-directed broadcast* adalah alamat *broadcast* yang dibentuk dengan mengatur semua *bit-bit network identifier* yang asli yang berbasis kelas menjadi 1 untuk sebuah *network* dengan alamat tak berkelas. Sebuah paket *network* yang dialamatkan ke alamat ini akan disampaikan ke semua *host* dalam semua *subnet* yang dibentuk dari *network identifer* yang berbasis kelas yang asli. Contoh untuk alamat *all-subnets-directed broadcast* adalah pada saat sebuah *network identifier* memiliki alamat 131.107.26.0/24, maka *alamat all-subnets-directed broadcast*-nya adalah 131.107.255.255. Dengan kata lain, alamat ini merupakan alamat jaringan *broadcast* dari *network identifier* alamat berbasis kelas yang asli. Pada contoh tersebut, alamat 131.107.26.0/24 merupakan alamat kelas B yang secara default memiliki *network identifer* 16 sehingga alamat *all-subnets-directed broadcast*-nya adalah 131.107.255.255. Semua *host* dari sebuah *network* dengan alamat tidak berkelas akan mendengarkan dan memproses paket-paket yang dialamatkan ke alamat ini. RFC 922 mengharuskan *router* IP untuk meneruskan paket yang di-*broadcast* ke alamat ini ke semua *subnet* dalam *network* berkelas yang asli. Dengan banyaknya alamat *network identifier* yang tidak berkelas, maka alamat ini pun tidak relevan lagi dengan perkembangan *network*. Menurut RFC 1812, penggunaan *address* jenis ini telah ditinggalkan.

3. *Limited Broadcast* merupakan alamat yang dibentuk dengan mengatur semua 32 bit alamat IP versi 4 menjadi 1 (11111111111111111111111111111111). *Address* ini digunakan ketika sebuah *node* IP harus melakukan penyampaian data secara *one-to-everyone* di dalam sebuah jaringan lokal tetapi belum mengetahui *network identifier*-nya. Contoh penggunaan *limited broadcast* adalah ketika proses konfigurasi alamat secara otomatis dengan menggunakan *Boot Protocol* (BOOTP) atau *Dynamic Host Configuration Protocol* (DHCP). Dengan DHCP, sebuah klien DHCP harus menggunakan alamat *limited broadcast* untuk semua lalu lintas yang dikirimkan hingga *server* DHCP memberikan sewaan alamat IP kepadanya. Semua *host* yang berbasis kelas atau

tanpa kelas akan mendengarkan dan memproses paket jaringan yang dialamatkan ke alamat ini. Meskipun kelihatannya dengan menggunakan alamat *limited broadcast.*

4. Alamat *Multicast* merupakan alamat IPv4 yang khusus didesain agar diproses oleh satu atau beberapa *node* dalam segmen jaringan yang sama maupun berbeda dan ditetapkan oleh IANA. Alamat biasanya *multicast* digunakan dalam komunikasi *one-to-many*. Alamat IP *multicast* merupakan alamat yang digunakan untuk menyampaikan satu paket kepada banyak penerima. Pada intranet yang memiliki alamat *multicast* IPv4, sebuah paket yang ditujukan ke sebuah alamat *multicast* akan diteruskan oleh *router* ke dalam subjaringan. Di dalam subjaringan tersebut terdapat *host-host* yang sedang berada dalam kondisi "*listening*" terhadap lalu lintas jaringan yang dikirimkan ke alamat *multicast* tersebut. Sehingga alamat *multicast* menjadi cara yang efisien untuk mengirimkan paket data dari satu sumber ke beberapa tujuan untuk beberapa jenis komunikasi. Alamat *multicast* didefinisikan dalam RFC 1112. Alamat *multicast* IP versi 4 didefinisikan dalam ruang alamat kelas D yakni 224.0.0.0/4 yang berkisar dari 224.0.0.0 hingga 224.255.255.255. Prefiks alamat 224.0.0.0/24 yakni dari alamat 224.0.0.0 hingga 224.0.0.255 tidak dapat digunakan karena dicadangkan untuk digunakan oleh lalu lintas *multicast* yang berada pada *subnet* lokal.

2.5.3 Pembagian Kelas pada IPv4

Pembagian kelas pada IPv4 disebutkan dalam RFC 791. IPv4 dibagi ke dalam lima kelas berdasarkan oktet pertamanya seperti terlihat pada Tabel 2.1.

Tabel 2.1. Pembagian Kelas pada IPv4

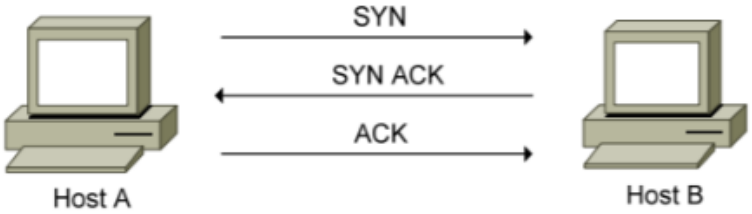
Kelas Alamat IP	Oktet Pertama (biner)	Oktet Pertama (desimal)	Digunakan Oleh
Kelas A	0xxx xxxx	1-127	Alamat *unicast* dengan skala jaringan yang besar
Kelas B	10xx xxxx	128-191	Alamat *unicast* dengan skala jaringan menengah hingga besar

Kelas C	110x xxxx	192-223	Alamat *unicast* dengan skala jaringan kecil
Kelas D	1110 xxxx	224-239	Alamat *multicast*
Kelas E	1111 xxxx	240-255	Untuk percobaan

Dari Tabel 2.1 terllihat bahwa alamat untuk kelas A digunakan untuk jaringan dengan skala yang besar. Nomor urut *bit* tertinggi pada alamat IP versi 4 kelas A selalu di-*set* dengan nilai 0. Tujuh bit berikutnya digunakan untuk melengkapi oktet pertama dan akan membuat sebuah *network identifier*. 24 bit selanjutnya atau tiga oktet terakhir merepresentasikan *host identifier*. Hal inilah yang membuat kelas A dapat digunakan pada 126 *network* dan memiliki 16.777.214 *host* pada tiap jaringannya. Alamat yang memiliki nilai oktet awal 127 tidak diizinkan karena digunakan untuk mekanisme *Interprocess Communication* (IPC) di dalam mesin. Alamat versi 4 pada kelas B digunakan untuk jaringan dengan skala menengah hingga besar. Dua *bit* pertama pada oktet pertama alamat IP kelas B selalu di-*set* ke bilangan biner 10. Kemudian 14 *bit* selanjutnya akan membuat sebuah *network identifier*. 16 *bit* berikutnya yakni pada dua oktet terakhir merepresentasikan *host identifier*. Kelas B dapat memiliki hingga 16.384 *network* dan memiliki 65.534 *host* pada tiap *network*-nya. Alamat IP versi 4 kelas C biasanya digunakan untuk jaringan dengan skala kecil. Tiga *bit* pertama pada *oktet* pertama alamat kelas C selalu di-*set* ke nilai biner 110. Untuk 21 bit selanjutnya akan membentuk sebuah *network identifier* dan 8 bit sisanya akan merepresentasikan *host identifier*. Hal ini membuat IP kelas C memiliki total *network* sebanyak 2.097.152 buah dan 254 *host* untuk setiap *network*-nya. Alamat IP versi 4 kelas D disediakan hanya untuk alamat-alamat *IP* dengan jenis *multicast*. Empat *bit* pertama pada IP kelas D selalu di-*set* ke bilangan biner 1110. Kemudian untuk 28 *bit* sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali *host*. Alamat IP versi 4 kelas E disediakan sebagai alamat untuk percobaan dan dicadangkan untuk digunakan pada masa depan. Empat bit pertama pada IP versi 4 kelas E ini di-*set* dengan bilangan biner 1111. Untuk 28 bit sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali *host*.

2.6 TCP dan UDP

Transmission Control Protocol (TCP) merupakan salah satu protokol yang ada dalam lapisan *transport* dan memiliki sifat *connection oriented*. *Connection oriented* yang dimaksud artinya memerlukan terbangunnya koneksi antara kedua *host* sebelum terjadinya komunikasi data [9]. TCP memiliki metode pembangunan koneksi yang bernama 3-*way handshak*e. Gambar 2.6 merupakan mekanisme kerja dari 3-way handshake:

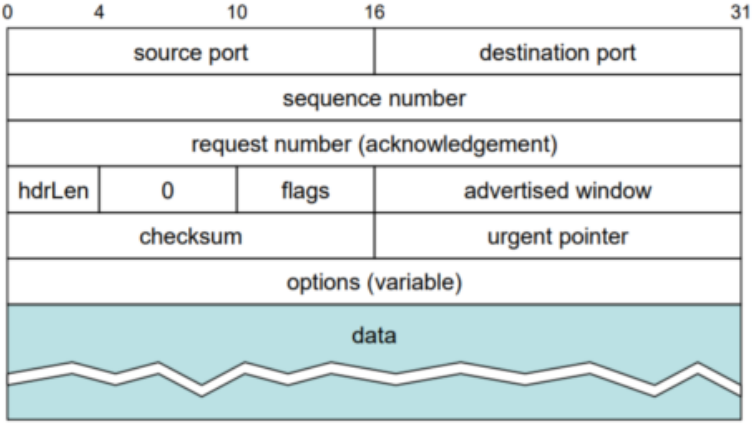


Gambar 2.6 Mekanisme 3-*way handshake* [9]

Berdasarkan Gambar 2.6, mekanisme 3-*way handshake* secara berurutan ialah:

1. Langkah pertama ialah *host* A akan mengirimkan sebuah pesan SYN (*synchronize*) kepada *host* B sebagai penanda bahwa akan dibangun sebuah koneksi.
2. Langkah kedua ialah *host* B akan merespon dengan mengirimkan pesan ACK (*acknowledgement*) sekaligus pesan SYN milik *host* B. Kedua pesan tersebut akan dikombinasikan menjadi SYN+ACK.
3. Langkah ketiga ialah *host* A akan menyelesaikan proses 3-*way handshake* dengan mengirimkan pesan ACK.

Pada header TCP terdapat 11 *fields* yang berbeda. Gambar 2.7 merupakan ilustrasi *header* TCP.



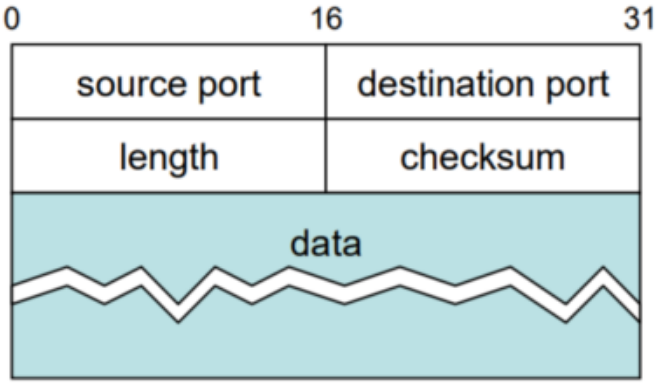
Gambar 2.7 *Header* TCP [10]

Berikut adalah penjelasan dari Gambar 2.7:

1. *Source port*: Memiliki panjang 16 bit dan berisi sumber asal *port* paket TCP.
2. *Destination port*: Memiliki panjang 16 bit dan berisi tujuan *port* paket TCP akan dikirim.
3. *Sequence number*: Memiliki panjang 32 bit yang berisi *sequence number*.
4. *Request number (acknowledgement)*: Memiliki panjang 32 bit yang berisi *acknowledgment number*.
5. *hdrLen (Header Length)*: Memiliki panjang 4 bit dan sering disebut data *offset* yang menandakan dimulainya isi data pada segmen TCP.
6. *Reserved* (0): Memiliki panjang 6 bit dan tidak ada isinya.
7. *Flags*: Memiliki panjang 6 bit dan sering disebut sebagai *control flag*. *Flag* tersebut berisi URG, ACK, SYN, PSH, RST dan FIN.
8. *Advertised window*: Memiliki panjang 16 bit dan berfungsi sebagai *flow control*.
9. *Checksum*: Memiliki panjang 16 bit dan berfungsi sebagai deteksi galat.
10. *Urgent pointer*: Memiliki panjang 16 bit dan akan berisi jika yang dikirimkan ialah *flag* URG sebagai penanda bahwa data urgent telah berakhir.

11. *Options*: Berisi variabel atau *padding* yang berfungsi untuk memastikan agar data yang dikirim selalu sepanjang 32 bit.

User Datagram Protocol (UDP) merupakan salah satu protokol yang ada pada lapisan *transport* dan bersifat *connectionless* sehingga tidak memerlukan adanya koneksi antara kedua host sebelum terjadinya komunikasi data [9]. UDP merupakan protokol yang sederhana dan tidak perlu ada mekanisme pembangunan koneksi seperti 3-*way handshake*, flow control ataupun mengetahui bahwa data yang dikirimkan telah diterima. UDP hanya bertugas mengirimkan paket saja. Oleh karena itu, UDP memiliki sifat *unreliable* jika dibandingkan dengan TCP. Gambar 2.8 merupakan header dari UDP:



Gambar 2.8 *Header* UDP [10]

Berdasarkan Gambar 2.8, UDP memiliki 4 *field* di dalam *header*-nya antara lain [10]:

1 *Source port*: Memiliki panjang 16 bit yang berisi sumber asal *port* UDP
2 *Destination port*: Memiliki panjang 16 bit yang berisi tujuan *port* UDP akan dikirim
3 *Length*: Deretan angka yang memiliki panjang 16 bit. Deretan angka tersebut merepresentasikan panjang *byte* dari datagram UDP yang dikirim
4 *Checksum*: Memiliki panjang 16 bit dan berfungsi sebagai deteksi galat

2.7 Port TCP dan UDP

Pada lapisan *transport*, protokol TCP dan UDP merupakan protokol yang paling sering digunakan di internet. Kedua protokol ini diperkenalkan istilah *port* sebagai entitas yang bersifat *logical* seperti alamat IP yang digunakan untuk identitas dari tiap sesi-sesi komunikasi yang terjadi di lapisan *transport* [11]. *Port* terdiri dari 16 bit angka dan port dibagi menjadi 3 kategori berdasarkan penomorannya antara lain:

1. *Well-known port* yang biasa dikenal sebagai *system port* merupakan nomor *port* yang ditetapkan oleh *Internet Assigned Numbers Authority* (IANA), nomor *port* yang termasuk dalam kategori *well-known port* dimulai dari 0 hingga 1023.
2. *Registered port* yang juga dikenal sebagai *user port* juga ditetapkan oleh IANA, nomor *port* yang masuk ke dalam kategori *registered port* dimulai dari 1024 hingga 4915.
3. *Dynamically assigned port* biasanya digunakan untuk privat. *Port* kategori *dynamically assigned port* disediakan sebagai nomor alternatif atau pengganti bagi *port* yang sudah ada. Penggunaan nomor *port* ini tujuannya adalah untuk keamanan.

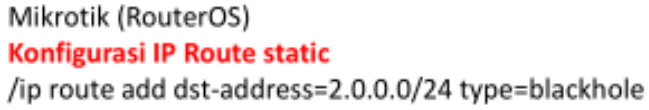
2.8 *Routing*

Pada suatu jaringan, terdapat lalu lintas paket data yang berpindah dari titik asal ke titik tujuan. Paket tersebut akan melewati jalur yang terbaik untuk mencapai tujuan. Oleh karena itu, dibutuhkan suatu teknik agar paket-paket tersebut dapat sampai ke tujuan dengan jalur terbaik. Teknik tersebut biasa disebut *routing*. Metode *routing* terbagi menjadi dua yaitu *routing* statis dan *routing* dinamis [2].

2.8.1 *Routing* Statis

Routing statis merupakan mekanisme *routing* yang dikonfigurasi secara manual. *Routing* statis merupakan metode *routing* yang digunakan pada kondisi jaringan yang statis. Jaringan statis merupakan jaringan yang hanya terjadi sedikit atau bahkan tidak ada perubahan di dalamnya. Karena konfigurasi routing statis harus dilakukan secara manual satu per satu oleh administrator jaringan agar tiap perangkat bisa saling terhubung, maka itu metode ini lebih sering digunakan pada jaringan berskala kecil. *Routing* statis merupakan metode *routing* paling aman karena dilakukan langsung oleh administrator sehingga paling dapat percaya dibandingkan

dengan protokol *routing* yang lain. Hal ini dapat dilihat dari nilai *administrative distance* sebesar 1 [2]. Gambar 2.9 merupakan contoh konfigurasi statis pada mikrotik.



Gambar 2.9 Konfigurasi Statis pada Mikrotik [12]

Konfigurasi statis digunakan saat konfigurasi internal BGP tidak dapat menghubungkan antara satu *router* dengan *router* lain yang tidak terhubung secara langsung. Syarat agar *router* yang terkonfigurasi dengan internal BGP dapat terhubung satu sama lain adalah menggunakan topologi *full mesh.* Antar *router* yang terhubung dengan internal BGP tidak dapat meneruskan informasi *routing* dari satu internal BGP *peer* pada internal BGP *peer* yang lain jika tidak menggunakan topologi *full mesh*. Karena *prefix* harus dikirim langsung oleh pengirim informasi dan diterima langsung oleh penerima informasi. Oleh karena itu, diperlukan konfigurasi *routing* statis untuk menghubungkan *router* yang dikonfigurasi dengan internal BGP.

2.8.2 *Routing Dinamis*

Routing dinamis memiliki karakteristik yang berlawanan dengan *routing* statis yakni memudahkan konfigurasi pada kondisi jaringan yang memiliki skala luas dan yang bersifat dinamis. Dinamis dalam hal ini ialah sering terjadi perubahan jaringan. Perubahan jaringan tersebut dapat berupa penambahan ataupun pengurangan perangkat. Kelebihan *routing* dinamis lainnya adalah metode ini tidak harus dikonfigurasi satu per satu oleh administrator jaringan. Administrator jaringan cukup mengkonfigurasi di titik-titik tertentu dan sisanya akan ada algoritma khusus dari *routing* dinamis yang akan menukar informasi *routing* sesuai dengan protokol yang digunakan [13]. Beberapa contoh algoritma yang banyak diterapkan adalah sebagai berikut:

1. *Distance Vector Protocol*

   Distance vector protocol merupakan protokol *routing* yang paling tua. Protokol ini menggunakan beberapa pendekatan algoritma dalam penerapannya antara lain algoritma Bellman-Ford yang digunakan oleh *Routing Information Protocol* (RIP

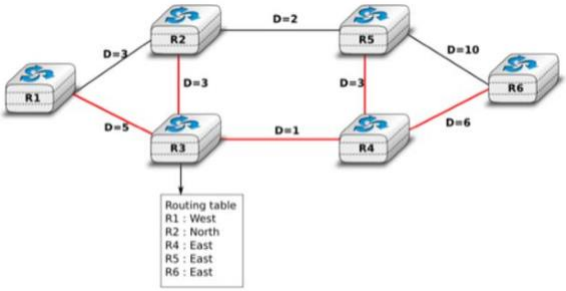
dan RIPv2) dan *Diffusing Update Algorithm* (DUAL) yang digunakan oleh protokol *routing* Cisco yaitu *Enhanced Interior Gateway Protocol* (EIGRP) [14]. Protokol ini menggunakan jarak dan arah sebagai acuan dalam melakukan pengambilan keputusan jalur yang akan dilewati. Beberapa karakteristik *distance vector protocol* ini adalah [15]:

a. Protokol ini akan mengirimkan *advertisement* untuk mempertahankan tabel *routing*-nya dengan *interval* waktu yang tetap yakni 30 detik untuk RIP dan 90 detik untuk EIGRP walaupun tidak ada perubahan dalam *table routing*.
b. Protokol ini dipercaya oleh tetangganya karena informasi tabel *routing* yang diterima sangat bergantung pada *router* tetangga atau yang dikenal dengan istilah *routing by rumor* sehingga setiap *update* tabel *routing* yang diberikan akan selalu diterima. Sehingga akan ada kemungkinan terjadinya *infinite loop* jika suatu saat terdapat kesalahan pada isi tabel *routing*.
c. Saat melakukan *update*, *distance vector protocol* ini akan selalu mengirimkan seluruh tabel *routing* yang dimilikinya. Oleh karena itu, *bandwidth* yang dibutuhkan sangat besar karena ditambah dengan *update* yang dilakukan secara *broadcast*.
d. Ketika terdapat *interface* yang mati atau baru dan terdapat network tambahan, maka *distance vector protocol* akan langsung melakukan *update* (*triggered update*).

2. *Link State Protocol*

Link state protocol seperti *Intermediate System to Intermediate System* (IS-IS) dan *Open Shortest Path First* (OSPF) sangat mengandalkan tiap *router* yang ada di dalam suatu jaringan untuk melakukan *advertise* dari kondisi hubungan di tiap sambungan. *Advertise* tersebut akan menghasilkan sebuah peta topologi jaringan yang utuh atau yang biasa disebut dengan *shortest path tree* [14]. Pada *link state protocol*, penentuan rute adalah dengan cara melihat titik yang memiliki total nilai *cost* terendah. Nilai *cost* merupakan nilai yang diatur oleh administrator

jaringan untuk mencapai suatu *router*. Contoh pemilihan jalur pada *link state protocol* ditunjukkan pada Gambar 2.10 berikut:



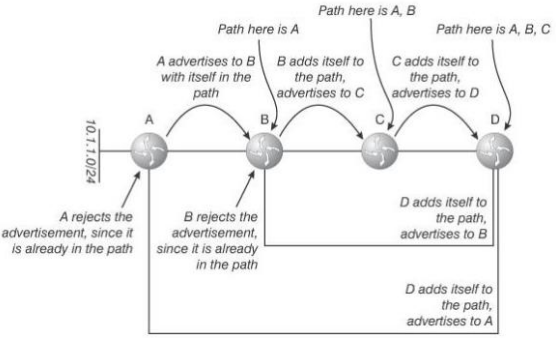
Gambar 2.10 Pemilihan jalur *link state protocol* [16]

Pada Gambar 2.10, terlihat bahwa rute menuju R4 melalui R2 akan lebih dipilih daripada rute melalui R3 karena total *cost* melalui R2 sebesar 25 sedangkan melalui R3 sebesar 40. Cara kerja protokol ini adalah dengan cara membuat *Link State Packet* (LSP) yang berisi informasi tentang jaringan yang terhubung secara langsung (*directly connected*) seperti *neighbor* ID, *link type* dan *bandwidth* oleh tiap *router* dan akan dikirimkan ke seluruh tetangga dari *router* tersebut hingga seluruh *router* menerima LSP dari masing-masing *router* yang ada pada jaringan. Berdasarkan data LSP yang sudah didapatkan, tiap *router* akan membuat peta topologi dari jaringannya sehingga dalam pengiriman paket dapat diketahui rute yang memiliki *cost* paling kecil diantara yang lain. Penyebaran LSP hanya dilakukan pada saat *startup* oleh *router* dan pada saat terdapat perubahan dalam topologi jaringan sehingga lebih efisien dibandingkan dengan *protokol distance vector*.

2. *Path Vector Protocol*

Path vector protocol merupakan protokol terbaru jika dibandingkan dengan dua protokol sebelumnya yakni *distance vector* dan *link state*. Protokol ini merupakan pengembangan dari protokol *distance vector* yang memiliki beberapa kelemahan seperti *looping*, *infinite count* dan informasi tabel routing yang tidak akurat. Protokol ini menjamin tidak akan terjadi *looping* dengan cara merekam tiap *hop* dari *routing advertisement* yang

melewati jaringan [17]. Penjelasan cara kerja dari *path vector* dapat dilihat pada Gambar 2.11.



Gambar 2.11 *Path vector protocol* [17]

Pada Gambar 2.11 terlihat bahwa *router* A akan mengirimkan *advertisement* kepada *router* B. *Advertisement* tersebut berfungsi untuk memberitahukan bahwa ia telah terhubung dengan jaringan 10.1.1.0/24. Ketika *router* B sudah menerima informasi tersebut, ia akan menambahkan dirinya ke dalam *advertisement* dari *router* A dan mengirimkannya ke *router* C. Kemudian router C akan melakukan hal yang sama seperti yang dilakukan oleh *router* B lalu mengirimkan *advertisement* ke *router* D hingga jaringan 10.1.1.0/24 dapat tercapai oleh *router* D. Namun pada saat *router* D ingin mengirimkan *advertisement* ke *router* A untuk memberitahukan bahwa dirinya sudah terhubung, akan ditolak oleh *router* A karena di dalam isi *advertisement* tersebut sudah terdapat *router* A. Hal tersebut juga berlaku saat *router* D akan mengirimkan *advertisement* ke *router* B dan *router* C. Hal ini dilakukan untuk menghindari terjadinya *looping* [17]. *Path vector* sering digunakan sebagai algoritma pemilihan *router* oleh protokol *routing* jenis *Exterior Gateway Protocol* (EGP) seperti BGP yang pertukaran informasi *routing*-nya dilakukan antar *Autonomous System* (AS).

2.9 *Administrative Distance*

Administrative distance (AD) merupakan suatu parameter yang digunakan oleh *router* dalam menentukan rute yang akan dilewatkan jika terdapat dua pilihan rute atau lebih yang berasal dari protokol *routing*

yang berbeda [18]. *Router* akan memilih informasi *routing* yang memiliki AD paling kecil. Semakin kecil nilai AD yang dimiliki oleh protokol *routing*, maka akan semakin dipercaya (*most trustworthy*). Jika terdapat dua informasi *routing* yang berasal dari eksternal BGP dengan nilai AD 20 dan informasi routing lainnya berasal dari internal BGP dengan nilai AD 200, maka *router* akan menggunakan informasi *routing* yang berasal dari eksternal BGP karena nilai AD-nya lebih kecil. Jika terdapat nilai AD sebesar 255, maka *router* akan mengabaikan informasi yang berasal dari sumber tersebut dan tidak akan menyimpannya di tabel *routing*. Nilai AD ditunjukkan pada Tabel 2.2.

Tabel 2.2 Nilai *administrative distance* tiap protokol *routing* [18]

Protokol *Routing*	**Nilai AD**
Connected (terhubung secara langsung)	0
Static	1
Enhanced Interior Gateway Routing Protocol (EIGRP)	5
External Border Gateway Protocol (eBGP)	20
Internal EIGRP	90
IGRP	100
OSPF	110
Intermediate System-to-Intermediate System (IS-IS)	115
Routing Information Protocol (RIP)	120
Exterior Gateway Protocol (EGP)	140
On Demand Routing (ODR)	160
External EIGRP	170
Internal Border Gateway Protocol (iBGP)	200
*Unknown**	255

2.10 *Border Gateway Protocol* (BGP)

BGP merupakan salah satu jenis protokol *routing* yang ada pada dunia komunikasi data. BGP mampu melakukan pengumpulan rute, pertukaran rute dan menentukan rute terbaik menuju ke sebuah lokasi dalam suatu jaringan. Ruting protokol BGP dilengkapi dengan algoritma yang pintar untuk mencari jalan terbaik. BGP termasuk dalam kategori ruting protokol jenis *Exterior Gateway Protocol* (EGP) [19]. Standar saat ini untuk *routing* antar *domain* adalah protokol BGP4. BGP merupakan satu-satunya protokol *routing* internet yang digunakan untuk menjaga

konektivitas antar AS. BGP adalah *path vector* di mana setiap router memilih rute terbaik ke tujuan [20, 21, 22].

Fungsi utama BGP adalah untuk mempertukarkan *network reachability information* antar BGP menggunakan TCP *port* 179 dalam proses pengiriman paket.

BGP merupakan protokol *routing* yang dapat menghubungkan antar AS yang sama maupun AS yang berbeda dalam skala *area* yang besar sehingga cocok digunakan untuk konfigurasi pada IdREN karena universitas yang terhubung ke dalam IdREN memiliki administrasi teknis yang berbeda dan rencana pengembangan IdREN ialah dapat terhubung dengan jaringan *research* & *education* di berbagai Negara. BGP memainkan peran yang penting dalam pembentukan komunikasi di internet. BGP memungkinkan terjadinya pertukaran informasi pada jaringan antar AS. BGP merupakan pengembangan dari *Exterior Gateway Protocol* (EGP) yang dulunya merupakan satu-satunya protokol *routing* pada jaringan antar AS.

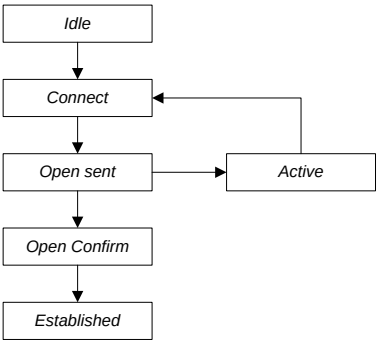
Tabel 2.3 merupakan tabel perbandingan BGP dengan protokol *routing* lainnya:

Tabel 2.3 Perbandingan BGP dengan protokol *routing* lain [2]

Protokol Routing	Skala	Lingkup Kerja	Multihoming	Kompabilitas
BGP	Jaringan skala besar	Antar *Autonomous System*	Mendukung	Semua perangkat
OSPF	Jaringan skala menengah	Dalam *Autonomous System*	Tidak mendukung	Semua perangkat
RIP	Jaringan skala menengah	Dalam *Autonomous System*	Tidak mendukung	Semua perangkat
EIGRP	Jaringan skala menengah	Dalam *Autonomous System*	Tidak mendukung	Hanya perangkat Cisco

2.10.1 *BGP States*

Ketika BGP membentuk *neighborship*, ada beberapa langkah yang dilalui seperti pada Gambar 2.12.



Gambar 2.12 BGP *States*

Neighbour State:
Iddle : *neighbour* tidak merespon
Active : mencoba untuk terhubung
Connect : TCP *session established*
Open Sent : *Open message* dikirim
Open Confirm : *Response* diterima
Establish : BGP terhubung dengan *neighbour*

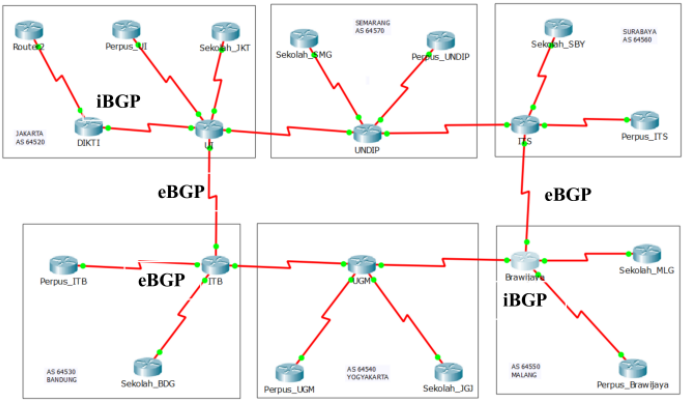
2.10.2 *Operasi BGP*

Seperti yang telah disebutkan sebelumnya bahwa fungsi utama BGP ialah untuk mempertukarkan *network reachability information* antar BGP menggunakan TCP *port* 179 dalam proses pengiriman paket. Informasi *routing* ini dipertukarkan dengan membangun sebuah sesi pada koneksi TCP antar router dengan konfigurasi BGP. Setelah sesi terbangun

dengan kondisi *established*, semua rute terbaik diumumkan oleh BGP *speaker* ke BGP *router* tetangga seperti pada Gambar 2.13.

Gambar 2.13 Operasi dasar BGP [23]

BGP sendiri terbagi menjadi dua jenis yaitu internal BGP (iBGP) dan eksternal BGP (eBGP). Gambar 2.14 merupakan ilustrasi dari iBGP dan eBGP.



Gambar 2.14 Internal BGP dan Eksternal BGP [24]

Pada Gambar 2.14, internal BGP berperan dalam pertukaran informasi *routing* dengan AS yang sama sedangkan eksternal BGP

berperan dalam pertukaran informasi dengan AS yang berbeda. Untuk menghindari *routing loops*, dalam satu AS koneksi antar BGP router dengan konfigurasi internal BGP diterapkan topologi *full mesh* [23].

Pada suatu jaringan yang besar dan memiliki banyak koneksi interdomain, topologi *full mesh* tidak *scalable* karena harus ada $\frac{n-1}{2}n$ internal BGP *session* [23]. Untuk itu solusi yang dapat dilakukan adalah menggunakan konfigurasi statis [12].

Setelah semua rute terbaik diumumkan oleh BGP *speaker* ke BGP router tetangga, langkah selanjutnya adalah BGP menangani kestabilan tabel *routing* yang dimiliki. Apabila ada perubahan tabel *routing*, hanya informasi *update* yang diumumkan ke BGP *peer*-nya. BGP tidak mensyaratkan *refresh* tabel *routing* secara berkala karena kemampuan *route refresh* sehingga perubahan *policy local* dapat langsung diterapkan dengan benar tanpa perlu mereset sesi BGP [23].

Ada empat jenis message yang dapat dipertukarkan antar *router* BGP yakni [23]:

1. *OPEN* : digunakan untuk membangun sesi BGP antar dua *router*
2. *UPDATE* : berisi *reachability information.* UPDATE dapat berisi informasi *prefix* yang ingin diumumkan ataupun menarik kembali (*withdraw*) informasi *prefix* yang telah diumumkan.
3. *NOTIFICATION* : digunakan untuk mengakhiri sesi BGP karena terjadi *error*
4. *KEEPALIVES* : digunakan sebagai tanda bahwa sesi BGP tetap berlangsung meskipun pesan UPDATE atau NOTIFICATION tidak diterima dalam periode waktu tertentu

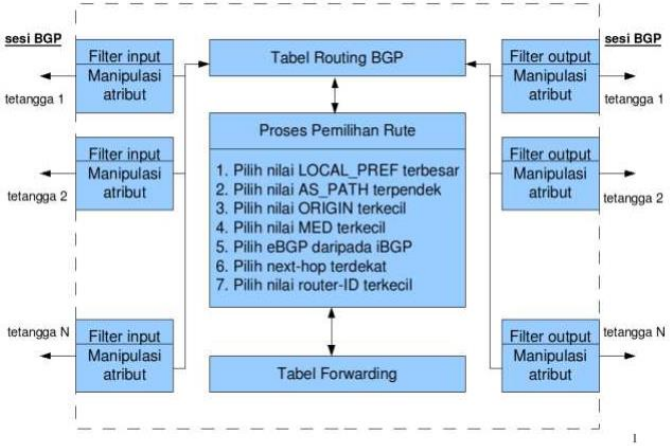
Ketika *router* dari BGP *speaker* mengumumkan suatu *prefix* ke tetangganya, hal ini berarti bahwa router penerima dapat mencapai *prefix* tersebut dengan cara meneruskan trafik menuju ke pengirim yang meng-*advertise prefix* tersebut. Apabila *router* mengirim informasi tidak mencapai *prefix* itu lagi atau tidak mingin membawa trafik menuju tujuan *prefix* tadi, *router* ini akan mengirimkan pesan *UPDATE* ke *router* tetangganya yang mengatakan bahwa rute menuju *prefix* tadi dihapus (*withdrawn*). Hal ini berarti setiap kali sebuah *router* mengganti rute terbaiknya, maka ia harus memberitahukan perubahan ini kepada setiap *router* tetannga yang telah diberitahu tentang rute ini sebelumnya. Selain meng-*advertise prefix*, BGP juga mengizinkan informasi lain untuk di-*advertise* ke BGP *router* tetangga. Informasi ini disebut atribut. Atribut

merupakan salah satu pertimbangan dalam proses pemilihan rute terbaik dari semua rute yang diterima untuk suatu *prefix* tertentu. Dalam proses inilah operator jaringan dapat melakukan manipulasi terhadap suatu rute dengan cara merubah atribut yang terkait dengan *prefix* rute tadi. Atribut BGP ini dapat dibagi menjadi empat jenis yaitu: *well_known mandatory*, *well_known discretionary*, *optional transitive* dan *optional non transitive* seperti pada Tabel 2.4. Atribut yang *transitive* dapat diteruskan ke AS tetangga sedangkan atribut yang *non-transitive* tidak dapat diteruskan ke AS tetangga [23].

Tabel 2.4 Atribut pada BGP [23]

Atribut	Jenis Atribut	Fungsi
ORIGIN	*well_known mandatory*	Mendefiniskan asal jalur informasi
AS_PATH	*well_known mandatory*	Berisi rangkaian segmen jalur AS
NEXT_HOP	*well_known mandatory*	Mendefinisikan alamat IP (*unicast*) yang digunakan sebagai *next hop* untuk suatu tujuan
MULTI_EXIT_DISC	*optional non transitive*	Membedakan titik masuk yang banyak menuju ke suatu AS
LOCAL_PREF	*well_known discretionary*	Menunjukkan tingkatan pemilihan rute
COMMUNITY	*optional transitive*	Menandai rute pelanggan, rute milik *peer*, maupun rute pada suatu area geografis tertentu

Proses pemilihan rute terbaik pada BGP terdiri dari empat komponen utama seperti pada Gambar 2.15.



Gambar 2.15 Pemilihan Jalur Terbaik [23]

Pada Gambar 2.15, untuk memilih jalur terbaik, terdapat empat komponen. Komponen pertama adalah *filter input* dan *output* yang dapat dikonfigurasikan untuk setiap sesi BGP. Fungsi dari *filter* rute adalah untuk menolak rute yang diterima atau dikirim oleh *router* atau dapat juga digunakan untuk memanipulasi atribut rute tersebut. Misalnya *filter* diterapkan hanya untuk menerima rute yang memiliki AS *path* berisi sekumpulan AS terpercaya. Proses *filter* ini dilakukan oleh operator jaringan. Komponen kedua adalah tabel *routing* BGP. Tabel *routing* ini berisi semua rute yang diterima oleh *router* dan lolos dari proses *filter input*. Atribut dari *router* ini disimpan dalam tabel *routing* dan mungkin telah di-*update* oleh *filter input*. Komponen ketiga adalah *decision process*. Dalam proses ini terjadi pemilihan rute terbaik dari rute-rute yang disimpan dalam tabel *routing* untuk setiap *prefix* tujuan. Ketika sebuah rute dipilih sebagai rute terbaik untuk suatu *prefix* tujuan, rute tersebut kemudian diinstall di tabel *forwarding* dan diumumkan ke *router* tetangga. Tabel *forwarding* merupakan komponen keempat dari *router*. Untuk setiap paket yang diterima, tabel *forwarding* akan dilihat kemudian ditentukan *outgoing interface* yang harus digunakan untuk meneruskan paket ke tujuan [23].

Proses pemilihan rute terbaik suatu *prefix* tujuan melibatkan kriteria pemilihan yang sering disebut dengan *best path selection algorithm*. Urutan pemilihan rute terbaik ialah [23]:

1. Memilih rute dengan nilai *LOCAL_PREF* yang terbesar. Atribut ini digunakan untuk memilih rute yang lebih diprioritaskan dibanding rute lain untuk *prefix* tujuan yang sama.
2. Memilih rute dengan AS_*PATH* terpendek. Ketika informasi *routing* dipropagasikan dalam jaringan, ASN akan ditambahkan dalam *AS_PATH*. Dengan memilih *AS_PATH* terpendek, BGP mengasumsikan semakin pendek *AS_PATH* maka nilai *delay* yang diperoleh akan semakin kecil.
3. Memilih rute dengan nilai *ORIGIN* terkecil. Atribut ini mengidentifikasikan bagaimana suatu originating AS mengetahui tentang suatu rute. Nilai *ORIGIN* ini dapat berupa IGP, EGP maupun *INCOMPLETE*.
4. Memilih rute dengan nilai MED terkecil. Rute tanpa tanpa atribut MED dianngap memiliki MED terendah. MED digunakan untuk memilih *egress point* dalam *domain* lokal. Aplikasi MED yang paling sering digunakan adalah *clod potato routing* yang bertujuan untuk membawa trafik selama mungkin berada dalam domain lokal sebelum diteruskan ke domain tetangga.
5. Memilih rute eksternal BGP dibanding rute internal BGP. Hal ini digunakan agar paket dapat segera mungkin meninggalkan *domain (hot potato routing).*
6. Memilih rute dengan *nexthop* terdekat yakni rute yang memiliki IGP *cost* paling kecil menuju *egress point*. Hal ini bertujuan agar paket dapat sebera mungkin meninggalkan *domain (hot potato routing).* Hal ini dicapai dengan cara mengarahkan trafik ke *border router* terdekat berdasarkan nilai IGP cost.
7. Pemilihan rute terbaik selanjutnya adalah memilih rute dengan alamat IP paling kecil atau rute yang berumur lebih lama.

Urutan pemilihan rute dari nomor 2 hingga 7 disebut aturan *tie-breaking* dari proses keputusan dalam BGP. Algoritma *tie-breaking* ini dimulai degan menganggap semua rute menuju tujuan adalah sama dan sebanding kemudian memilih rute yang akan dihapus dari pertimbangan.

2.11 *Autonomous System*

Autonomous System (AS) merupakan sekumpulan perangkat jaringan atau *router-router* yang memiliki administrasi teknis seperti kebijakan *routing* dan alamat *prefix* yang sama. AS biasanya dimiliki oleh

satu *Internet Service Provider* (ISP) maupun instansi atau perusahaan besar untuk terhubung dengan ISP atau instansi lain. AS menggunakan *Interior Gateway Protocol* (IGP) dalam merutekan paket-paket yang akan dikirim di dalam AS tersebut [25]. AS direpresentasikan menggunakan nomor-nomor yang disebut *Autonomous System Number* (ASN) antara 1 sampai 65.535 dan telah ditentukan oleh *Internet Assigned Numbers Authority* (IANA) kepemilikan dari tiap nomor tersebut [26]. Pada IdREN dari 50 universitas yang telah terhubung, ada 30 jaringan universitas yang sudah memiliki ASN sendiri. ASN dari IdREN ialah 64302. Daftar ASN beberapa peguruan tinggi yang sudah tergabung dengan IdREN dan tercatat pada situs https://bgp.he.net/AS64302#_peers ditunjukkan pada Tabel 2.5.

Tabel 2.5. AS Tiap Universitas

No.	*Description*	*Peer*
1	INHERENT	AS18007
2	Institut Teknologi Sepuluh Nopember	AS38331
3	Universitas Gadjah Mada	AS45705
4	Universitas Nusa Cendana	AS136059
5	Bogor *Agricultural University*	AS17553
6	Universitas Muhammadiyah Malang	AS46057
7	Institut Seni Indonesia Surakarta	AS136866
8	Universitas Malikussaleh	AS137299
9	UIN AR-RANIRY	AS137300
10	Universitas Hang Tuah	AS137327

2.12 *National Research Education Network* (NREN)

Research and Education Network (REN) merupakan jaringan komunikasi data yang memiliki kecepatan tinggi dan didedikasikan untuk kebutuhan komunitas pendidikan dan penelitian. REN membuat peneliti, tenaga pengajar, dan mahasiswa untuk saling berbagi informasi penelitian secara digital dengan sangat cepat dan dapat bekerja bersama secara efisien [27]. Sedangkan *National Research and Education Network* (NREN) merupakan infrastruktur REN yang dimiliki oleh suatu negara. Pengelolaan NREN dikelompokan ke dalam regional-regional dan diatur oleh satu organisasi khusus seperti GEANT yang merupakan kumpulan NREN di kawasan Eropa, RedCLARA yang merupakan kumpulan NREN di kawasan Amerika Latin, TEIN3 yang merupakan kumpulan NREN di

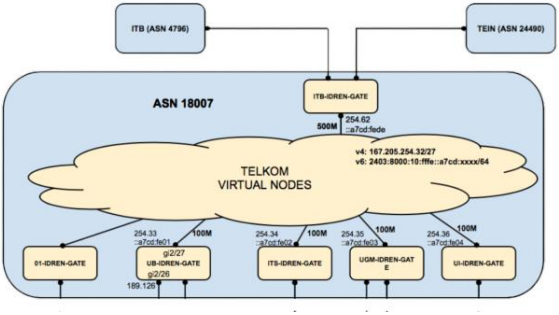
kawasan Asia Pasifik, CAREN yang merupakan kumpulan NREN di kawasan Asia Tengah, EUMEDCONNECT2 yang merupakan kumpulan NREN di kawasan Afrika Utara dan Timur Tengah dan UbuntuNet Alliance yang merupakan kumpulan NREN di kawasan Afrika Selatan dan Afrika Timur. Pengelolaan NREN berdasarkan regional ini bertujuan agar tiap-tiap NREN dapat saling berbagi informasi penelitian di dalam satu regional dan nantinya pengelolaan dalam regional akan memudahkan pembagian informasi antar regional. Pengelolaan REN berada di tiga tingkatan sebagai berikut: tingkatan pertama di lingkup negara atau NREN, tingkatan kedua ialah tingkat regional dan tingkat ketiga ialah tingkat global. NREN bukan digunakan untuk organisasi *profit* karena biasanya dikembangkan melalui pendanaan negara sehingga pengelolaan tiap regionalnya dilakukan oleh organisasi nonprofit seperti DANTE yang merupakan organisasi nonprofit mengelola GEANT. GEAT merupakan jaringan yang menghubungkan lebih dari 40 juta pengguna di lebih dari 40 negara di Eropa [27].

Pada penerapannya, NREN menghubungkan antar institusi penelitian dan pendidikan, universitas bahkan sekolah dan museum pada sebuah negara yang umumnya memiliki kecepatan sangat tinggi yang digunakan sebagai sarana pertukaran data penelitian dan pendidikan. NREN juga dapat berkembang ke sektor pemerintahan dan kesehatan. NREN juga biasanya digunakan sebagai tempat uji coba protokol atau teknologi baru yang telah ditemukan sebelum diimplementasikan ke publik [2].

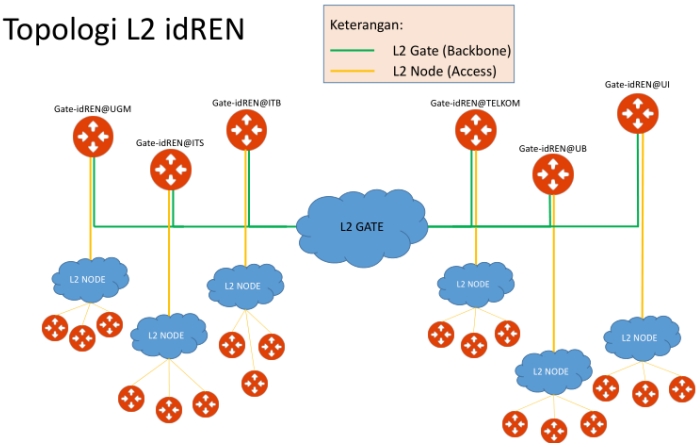
2.13 IdREN

IdREN merupakan NREN milik Indonesia yang dibentuk pada tahun 2015 sebagai upaya untuk meningkatkan kualitas pendidikan dan penelitian dari perguruan tinggi di Indonesia. Sebelumnya, Indonesia pernah memiliki NREN yang bernama *Indonesia Higher Education Network* (INHERENT) namun telah berhenti beroperasi sejak tahun 2013. IdREN merupakan sebuah *network* tertutup (*National Closed User*) bagi *research sharing* antar perguruan tinggi dengan Inisiatif 3S (*Single Network, Sharing and Collaboration & Sustainable Platform*) dengan

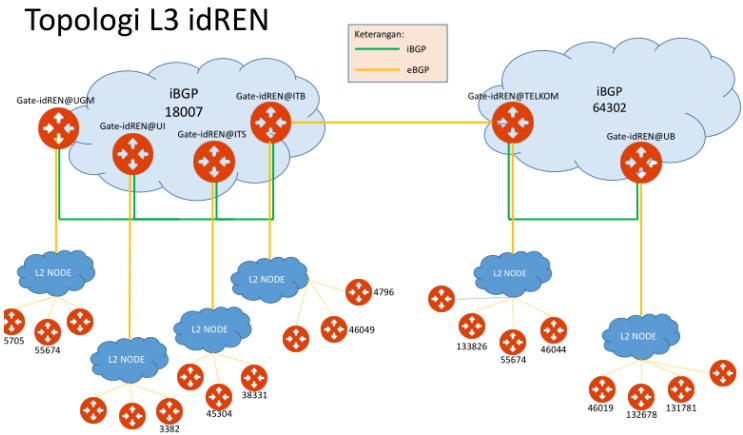
kolaborasi bersama pihak pemerintah, industri, media dan komunitas. Fungsi dari IdREN itu sendiri adalah sebagai *platform research* antar perguruan tinggi dan sebagai jembatan antara institusi pendidikan tinggi, pemerintah, industri, media dan komunitas guna meningkatkan daya saing nasional di kancah global. IdREN dibentuk atas inisiasi dari lima perguruan tinggi yaitu UGM, UI, ITB, ITS dan UB. Kelima perguruan tinggi tersebut berfungsi sebagai *gateway* yang menghubungkan universitas yang ingin bergabung dengan IdREN. PT. Telekomunikasi Indonesia, Tbk. sebagai penyedia fasilitas telekomunikasi menyediakan jalur IdREN di layer dua. *Gateway* tersebut akan menjadi titik yang akan menghubungkan universitas lain yang akan bergabung dengan IdREN. Perguruan tinggi yang ingin bergabung terlebih dahulu mendaftar ke pihak yang mengelola IdREN kemudian setelah di terima permohonan untuk bergabungnya, perguruan tinggi melakukan konfigurasi jaringan dengan *gateway* terdekat. IdREN memungkinkan institusi pendidikan di Indonesia untuk berbagi pakai sumber daya pembelajaran yang dimiliki melalui jalur yang lebih aman. Sumber daya tersebut berupa bahan perkuliahan, bahan pustaka, *software*, *network access*, dan *journal online*. Untuk saat ini, jalur IdREN yang disediakan oleh PT. Telekomunikasi Indonesia, Tbk adalah sebesar 20 Mbps. Gambar infrastruktur jaringan IdREN, topologi *real* L2 oleh PT. Telekomunikasi Indonesia, Tbk. Menggunakan *virtual* LAN dan L3 menggunakan konfigurasi BGP terlihat pada Gambar 2.16, Gambar 2.17 dan Gambar 2.18.



Gambar 2.16 Infrastruktur IdREN saat ini [28]



Gambar 2.17 Topologi L2 IdREN [29]



Gambar 2.18 Topologi L3 IdREN [29]

Untuk rincian alokasi IPv4 IdREN 103.78.232.0/22 adalah sebagai berikut [29]:

- 103.78.232.0/24 – Cadangan
- 103.78.233.0/24 – Cadangan
- 103.78.234.0/24 – *Network* Akses:
  - 103.78.234.0/27 – UI-IDREN
  - 103.78.234.32/27 – ITB-IDREN
  - 103.78.234.64/27 – ITS-IDREN
  - 103.78.234.96/27 – UGM-IDREN
  - 103.78.234.128/27 – UB-IDREN
  - 103.78.234.160/27 – 01-IDREN
  - 103.78.234.192/27 – Cadangan
  - 103.78.234.224/27 – Cadangan
- 103.78.235.0/24
  - 103.78.235.0/25 – *Apps* dan *Service* Jaringan
  - 103.78.235.128/25 – *Network Backbone*

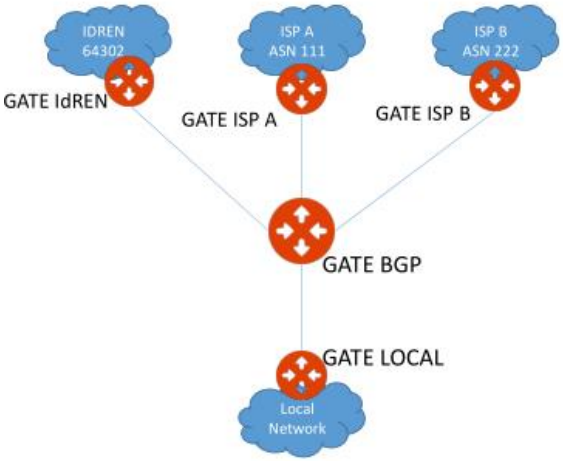
Kemudian untuk rincian alokasi IPv6 IdREN 2001:df6:5a00::/48 adalah sebagai berikut [29]:

- *Network Access:*
- 2001:DF6:5A00::A601:FE01/64 – UI-IDREN.
- 2001:DF6:5A00::A602:FE01/64 – ITB-IDREN.
- 2001:DF6:5A00::A603:FE01/64 – UGM-IDREN.
- 2001:DF6:5A00::A604:FE01/64 – UB-IDREN.
- 2001:DF6:5A00::A605:FE01/64 – 01-IDREN.

- *Network Backbone:*
- 2001:DF6:5A00::A7CD:FE01/64

2.13.1 *Topologi Node IdREN yang Tidak Memiliki ASN*

Topologi *node* IdREN yang tidak memiliki ASN dijelaskan pada Gambar 2.19.



Gambar 2.19 Topologi *node* IdREN yang tidak memiliki ASN [29]

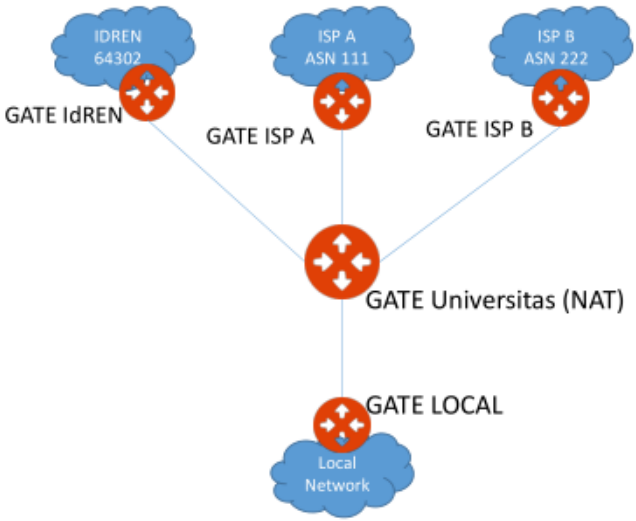
Berdasarkan Gambar 2.19, *rule* BGP-nya adalah sebagai berikut:

1. *Filter prefix* yang di-*advertise* – Universitas *Prefix* (/24)
2. Nilai *local preference Gate* IdREN > ISP A & B. Contoh:
   • IdREN: 500
   • ISP A: 300
   • ISP B: 200

3. Panjang *as-path* ASN yang di *advertise* ke *Gate* IdREN < ISP A & B
   - IdREN: 46019
   - ISP A: 46019 46019
   - ISP B: 46019 46019

2.13.2 *Topologi Node IdREN yang Memiliki ASN*

Topologi *node* IdREN yang memiliki ASN dijelaskan pada Gambar 2.20.



Gambar 2.20 Topologi *node* IdREN yang memiliki ASN [29]

Berdasarkan Gambar 2.20, *rule* BGP-nya adalah sebagai berikut:
1. *Filter prefix* yang akan di-*advertise* – *prefix* Universitas.
2. Nilai *local preference Gate* IdREN > ISP A & B. Contoh:
   • IdREN: 500
   • ISP A: 300
   • ISP B: 200

3. Panjang *as-path* ASN yang di *advertise* ke *Gate* IdREN < ISP A & B
   - IdREN: 46019
   - ISP A: 46019 46019
   - ISP B: 46019 46019

2.13.3 *Konfigurasi IdREN secara Real pada Mikrotik (RouterOS)*

Konfigurasi IdREN secara *real* pada Mkrotik terdiri dari konfigurasi BGP *Session*, IP *Route static*, Konfigurasi *Prefix* dan *Route Map (Filter Prefix advertise)* seperti dibawah ini [29]:

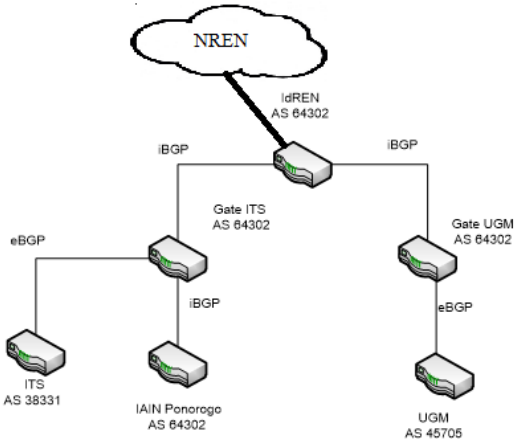
Konfigurasi BGP *Session*
/routing bgp instance set default as=65530
/routing bgp peer add remote-address=1.1.1.1
remote-as=65520 out-filter=AS65530-bgp-out
in-filter=AS65520-bgp-in nexthop-choice=force-
self comment="ANYNET“

Mikrotik (RouterOS)
Konfigurasi IP *Route static*
/ip route add dst-address=2.0.0.0/24 type=blackhole

Konfigurasi *Prefix* dan *Route Map (Filter Prefix advertise)*
/routing filter add chain=AS65530-bgp-out prefix=11.0.0.0/22
action=accept
/routing filter add chain=AS65530-bgp-out action=discard
Route Map Local Preference dan as-path prepend
/routing filter add chain=AS46019-bgp-in set-bgp-local-pref=500
/routing filter add chain=AS46019-bgp-out set-bgp-prepend=2

2.13.4 *Prinsip Kerja Test Bed IdREN*

Test bed IdREN yang dibuat secara keseluruhan menggunakan Mikrotik *routerBOARD* RB951-2n. Kabel yang digunakan untuk menghubungkannya ialah kabel *ethernet* cat5 dengan kecepatan maksimal 100 Mbps. Kabel UTP cat5 merupakan kabel UTP dengan standar yang diciptakan pada tahun 2001 oleh TIA/EIA-568-B. Kabel UTP cat5 hanya dapat melakukan transmisi data sebesar 100 Mbit/s, kapasitas maksimum ini sama dengan kapasitas kemampuan *ethernet* dalam mengirimkan signal data 100BASE-TX pada era tahun 2001. *Test bed* IdREN menggunakan enam buah *router* mikrotik dan dikonfigurasi dengan internal BGP, eksternal BGP dan konfigurasi statis pada internal BGP yang tidak bertetangga secara langsung seperti pada Gambar 2.21. MikroTik *RouterOS* mendukung BGP Versi 4, sebagaimana didefinisikan dalam RFC 4271.



Gambar 2.21 Prinsip kerja *Test bed* IdREN

Konfigurasi BGP pada Gambar 2.21 menggunakan atribut *BEXT-HOP*. Pada *Request For Comment* (RFC) 4271, *NEXT_HOP* merupakan atribut *well-known mandatory* yang mendefinisikan IP *address* dari sebuah *router* yang harus digunakan sebagai hop berikutnya ke tujuan yang tercantum dalam pesan *UPDATE*. Biasanya, atribut NEXT_HOP dipilih sehingga jalur yang terpendek akan diambil. Atribut NEXT_HOP digunakan oleh BGP *speaker* untuk menentukan *interface* keluar yang sebenarnya dan alamat *NEXT-HOP* langsung yang harus digunakan untuk meneruskan paket transit ke tujuan. Alamat *NEXT-HOP* langsung ditentukan dengan melakukan operasi pencarian rute rekursif untuk alamat IP di *NEXT_HOP* atribut dan menggunakan isi dari *routing table*.

Nexthop-choice yang ada pada saat konfigurasi *peer* BGP memengaruhi pemilihan atribut *NEXT_HOP* yang keluar karena *nexthop* yang di-*set* di *filter* selalu didahulukan. *NEXT_HOP* tidak diubah pada refleksi rute. Pilihan *NEXT_HOP* pada mikrotik ada tiga antara lain [30]:

- *Default* ialah pilih *NEXT_HOP* secara *default* yang seperti yang dijelaskan dalam RFC 4271.
- *Force-self* ialah pilihan *NEXT_HOP* yang selalu menggunakan *local address* dari *interface* yang digunakan untuk terhubung ke *peer* sebagai *NEXT_HOP*.
- *Propagate* ialah pilihan *NEXT_HOP* yang mencoba untuk menyebarkan lebih lanjut *NEXT_HOP* yang diterima yaitu jika rute

memiliki atribut BGP *NEXT_HOP*, maka digunakan sebagai *NEXT_HOP*. Jika tidak, maka kembali ke *default.*

Berdasarkan hasil wawancara dengan Bapak Raga, konfigurasi *test bed* IdREN ini menggunakan pengaturan *nexthop-choice=force-self* yang berfungsi agar *local address* dari *interface* yang digunakan terhubung ke *peer*. *Action* yang digunakan pada pengaturan *filter* untuk pengaturan *NEXT-HOP* ialah *passthrough* agar *prefix* yang di-*advertise* melewati *hop* yang telah diatur pada *NEXT-HOP*. Konfigurasi *prefix* dan *route map (filter prefix advertise)* ialah dengan melakukan konfigurasi pada *out-filter. Out-filter* yang dipilih ialah nama dari *routing filter chain* yang diterapkan ke informasi perutean keluar. IP yang diatur pada *out-filter* merupakan alamat IP pada *gateway* yang keluar dari BGP *speake*r menuju *neighbor* tujuan. Konfigurasi IP *route static* digunakan pada *test bed* ini karena saat konfigurasi internal BGP tidak menggunakan topologi *full mesh.*

Konfigurasi peer pada internal BGP juga menggunakan protokol *Bidirectional Forwarding Detection* (BFD) untuk deteksi kesalahan dalam jalur dua arah pada internal BGP dengan cepat. BFD merupakan *low-overhead* dan *short-duration protocol* yang berfungsi untuk mendeteksi kesalahan dalam jalur dua arah dan tertulis dalam RFC 5880. BFD pada dasarnya adalah protokol halo yang berfungsi untuk memeriksa kemampuan jangkauan tetangga dua arah. BFD menyediakan dukungan deteksi kegagalan tautan dalam sub-detik. BFD bukan protokol *routing* yang spesifik. BFD tidak seperti protokol *hello timer* atau semacamnya. Paket BFD *Control* ditransmisikan dalam paket UDP dengan *port* tujuan 3784. *Source port* berada di kisaran 49152 hingga 65535. Dan paket BFD *Echo* dienkapsulasi dalam paket UDP dengan port tujuan 3785 [31].

2.14 Performa dan Kualitas pada Jaringan

Dalam mengukur performa dan kualitas pada suatu jaringan digunakan beberapa parameter diantaranya adalah *bandwidth, throughput dan latency* [32]. Selain itu terdapat juga *jitter* dan *packet loss*. Berikut penjelasannya [2]:

1. *Bandwidth* merupakan besarnya kapasitas data yang dapat dibawa oleh suatu medium penghantar data atau jaringan. Salah satu tool yang dapat digunakan untuk mengukur *throughput bandwidth* dalam sebuah *link network* ialah *Iperf*.

2. *Throughput* ialah perhitungan seberapa banyak data yang benar-benar terkirim per satuan waktu dalam suatu jaringan, kanal, atau *interface*. *Throughput* secara teori mirip dengan *bandwidth* tapi saat diterapkan secara langsung *bandwidth* yang dihasilkan tidak sesuai dengan yang diatur inilah yang disebut sebagai *throughput*.
3. Latency *ialah* waktu yang dibutuhkan saat proses pengiriman data dalam suatu jaringan atau kanal komunikasi. Perhitungan dimulai saat proses permintaan data dibuat hingga data tersebut sampai ke tujuan. Semakin kecil nilai *latency* maka kualitas jaringan yang semakin baik.
4. *Packet Loss* merupakan jumlah paket yang gagal dalam proses pengiriman paket. Semakin kecil *packet loss,* maka kualitas jaringan tersebut semakin baik.
5. *Jitter* merupakan variasi *delay*. Hal ini terjadi karena adannya panjang antrian dalam suatu pengolahan data dan penyusunan kembali paket data di akhir pengiriman akibat kegagalan sebelumnya [33].

BAB 3
PERANCANGAN DAN REALISASI ALAT

3.1 Sumber Data

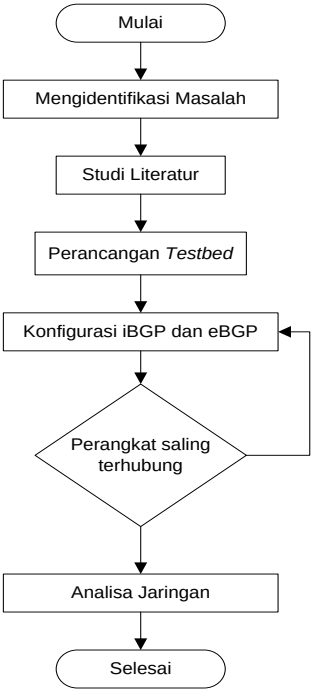
Tugas akhir ini menggunakan sumber data sebagai berikut:

1. Informasi dan studi literatur seperti buku tentang IPv4 dan juga prisip BGP. Jurnal tentang prinsip kerja BGP, makalah tentang BGP dan situs-situs internet tentang IdREN dan BGP yang berhubungan dengan tugas akhir.
2. Hasil diskusi dengan dosen pembimbing dan pihak-pihak yang ahli yakni Bapak Raga di ITS dan Bapak Fikry alumni UGM jurusan Teknologi informasi lulus tahun 2017 terkait topik tugas akhir yakni BGP dan IdREN.

3.2 Diagram Alir Penelitian Secara Keseluruhan

Penelitian dimulai dengan langkah pertama yakni mengindentifiksi masalah yang melatarbelakangi analisis dari pembuatan *test bed* ini. Setelah masalah teridentifikasi, proses berikutnya ialah melakukan studi literatur mengenai tugas akhir ini sebagai solusi dari masalah-masalah yang ada. Kemudian itu dilanjutkan dengan memulai pelaksanaan tugas akhir ini yakni perancangan *test bed*. *Test bed* pada tugas akhir ini dilakukan dengan menggunakan mikrotik RB951-2n. Pada tahap awal pelaksanaan, dirancang terlebih dahulu topologi dari IdREN berdasarkan literatur yang telah didapatkan dan dilengkapi dengan konfigurasi-konfigurasi dari tiap perangkat yang dibutuhkan. Selanjutnya masuk ke proses *routing* untuk membentuk koneksi di tiap perangkat dengan cara melakukan konfigurasi internal BGP dan eksternal BGP. Konfigurasi internal BGP dan eksternal BGP dengan cara menambahkan pengaturan atribut dari BGP yaitu *nexthop*. Jika koneksi belum terbangun dengan kondisi *established*, maka dilakukan pengecekan kembali konfigurasi yang ada hingga koneksi terbentuk. Setelah koneksi terbentuk, dilakukan pengecekan pemilihan jalur yang dilewati melalui *traceroute* dan pengujian *throughput* saat melakukan *download* ke *server*. Setelah itu dilakukan analisis mengenai performa dari hasil proses routing internal BGP dan eksternal BGP yang telah dilakukan. Setelah pelaksanaan selesai, dilakukan penarikan kesimpulan dan penulisan laporan. Diagram

alir penelitian secara keseluruhan yang dibuat ditunjukkan pada Gambar 3.1.

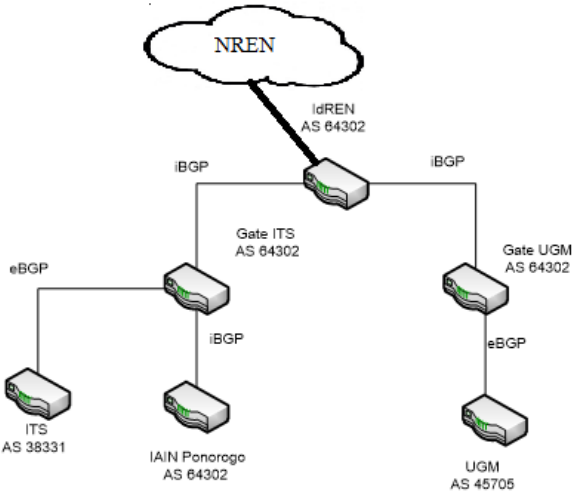


Gambar 3.1 Diagram Alir Penelitian Secara Keseluruhan

3.3 Perancangan Sistem

Perancangan sistem pada tugas akhir ini yaitu merancang *Test bed* jaringan IdREN yang terhubung ke *GATE*-ITS dan *GATE*-UGM terlebih dahulu. *Test bed* jaringan IdREN ini terdiri dari *GATE*-ITS, ITS, IAIN Ponorogo, *GATE*-UGM dan UGM seperti pada Gambar 3.1. IP yang digunakan pada *test bed* ini ialah IPv4. *Test bed* ini tidak terhubung dengan internet sehingga IP yang digunakan dapat berupa IP publik maupun IP privat. Karena *test bed* ini dibuat untuk mensimulasikan jaringan yang sebenarnya, maka IP yang digunakan ialah IP publik. Jaringan utama IdREN dengan *GATE*-ITS menggunakan *routing protocol* internal BGP. Jaringan di bawah *GATE*-ITS yakni ITS menggunakan

routing protocol eksternal BGP sedangkan IAIN Ponorogo menggunakan *routing protocol* internal BGP karena IAIN Ponorogo belum memiliki ASN. Kemudian jaringan utama IdREN dengan *GATE*-UGM menggunakan *routing protocol* internal BGP. Jaringan di bawah *GATE*-UGM yakni UGM menggunakan *routing protocol* eksternal BGP. Untuk konfigurasi antara IAIN Ponorogo dan IdREN menggunakan konfigurasi statis karena topologi yang digunakan saat konfigurasi internal BGP tidak menggunakan topologi *full mesh*. Pada tugas akhir ini hanya menggunakan tiga area kampus dikarenakan ketersediaan *router* mikrotik yang terdapat di laboratorium B301. *Test bed* Jaringan IdREN dengan *GATE*-ITS *dan GATE*-UGM ditunjukkan pada Gambar 3.2.



Gambar 3.2 *Testbed* Jaringan IdREN dengan *GATE*-ITS dan *GATE*-UGM

3.4 Perancangan *Testbed* Jaringan IdREN dengan *GATE*-ITS dan *GATE*-UGM

Test bed jaringan IdREN yang akan dirancang ialah jaringan *GATE*-ITS dan universitas yang terhubung dengan *GATE*-ITS. Metodologi yang digunakan di *Test bed* jaringan IdREN yaitu menggunakan *routing protocol* BGP (internal BGP dan eksternal BGP). Internal BGP digunakan untuk menghubungkan *GATE*-IdREN dengan *GATE*-ITS dan *GATE*-ITS dengan IAIN Ponorogo karena memiliki ASN yang sama. Eksternal BGP

digunakan untuk menghubungkan *GATE*-ITS dengan ITS karena memiliki ASN yang berbeda. ASN, *Router* ID, dan *Pefix* dari tiap *router* ditunjukkan pada Tabel 3.1.

Tabel 3.1 ASN, *Router* ID, dan *Pefix* dari tiap *Router*

Router	**ASN**	***Router* ID**	**Prefix**
GATE-IdREN	64302	103.78.235.254	103.78.235.128/25
GATE-ITS	64302	103.78.235.251	103.78.235.128/25
ITS	38331	202.46.129.246	202.46.129.244/30
IAIN Ponorogo	64302	103.78.234.66	103.78.234.66/27
GATE-UGM	64302	103.78.232.2	103.78.232.0/25
UGM	45705	202.43.93.246	202.43.93.0/24

Jalur dari *GATE*-IdREN dan *GATE*-ITS dikonfigurasi menggunakan iBGP karena memiliki ASN yang sama yakni 64302. Jalur dari *GATE*-ITS dan ITS dikonfigurasi menggunakan eBGP karena memiliki ASN yang berbeda yakni 64302 untuk *GATE*-ITS dan 38331 untuk ITS. Jalur dari *GATE*-ITS dan IAIN Ponorogo dikonfigurasi menggunakan iBGP karena memiliki ASN yang sama yakni 64302. Jalur dari *GATE*-IdREN dan *GATE*-UGM dikonfigurasi menggunakan iBGP karena memiliki ASN yang sama yakni 64302. Kemudian untuk jalur dari *GATE*-UGM dan UGM dikonfigurasi menggunakan eBGP karena memiliki ASN yang berbeda yakni 64302 untuk GATE-UGM dan 45705 untuk UGM. Sedangkan Untuk pengalamatan tiap *router* yang terhubung dapat dilihat pada Tabel 3.2.

Tabel 3.2 Pengalamatan tiap *Router*

Router	***Address Router* ID**	***Remote Address***
GATE-IdREN (*SERVER*)	103.78.235.254/25	103.78.235.251
GATE-ITS	103.78.235.251/25	103.78.235.254
ITS	202.46.129.246/30	202.46.129.245
IAIN Ponorogo	103.78.234.66/30	103.78.234.65
GATE-UGM	103.78.233.2/25	103.78.233.1
UGM	202.43.93.246/24	203.43.93.247

Kemudian untuk pengalamatan IP *server* dan *client* pada tiap *router* yang terhubung dapat dilihat pada Tabel 3.3.

Tabel 3.3 Pengalamatan IP *Server* dan *Client*

Router	*Gateway* ke Laptop	IP Laptop
GATE-IdREN (*SERVER*)	101.203.173.1/24	101.203.173.2/24
GATE-ITS	101.203.168.1/24	101.203.168.2/24
ITS	101.203.170.1/24	101.203.170.2/24
IAIN Ponorogo	101.203.169.1/24	101.203.169.2/24
GATE-UGM	101.203.171.1/24	101.203.171.2/24
UGM	101.203.174.1/24	101.203.174.2/24

3.5 Konfigurasi BGP *Session*

Pada saat konfigurasi BGP *Session*, hal yang pertama dilakukan adalah konfigurasi BGP *Instance*. Pada kolom *name* diisi nama *session*-nya. Kemudian diisi AS *number*-nya. Langkah selanjutnya ialah menulis *router-id*. Hasil konfigurasi BGP *Instance* seperti berikut:

```
name="idren" as=64302 router-id=103.78.234.66 redistribute-
connected=no redistribute-static=no redistribute-rip=no
redistribute-ospf=no redistribute-other-bgp=no out-filter="" client-
to-client-reflection=no ignore-as-path-len=no routing-table=""
```

Langkah selanjutnya ialah konfigurasi pada *routing filter* untuk digunakan pada saat konfigurasi *peer* iBGP. Kofigurasinya ialah menuliskan nama *chain*, kemudian memilih *action passthrough* agar *prefix* yang di-*advertise* melewati *hop* yang telah diatur pada *NEXT-HOP*. Kemudian mengatur *set out next hop* agar tidak terjadi looping karena berdasarkan RFC 4271, *NEXT_HOP* bertugas mendefinisikan IP *address* dari sebuah *router* yang harus digunakan sebagai *hop* berikutnya ke tujuan yang tercantum dalam pesan *UPDATE*. Atribut *NEXT_HOP* digunakan oleh BGP *speaker* untuk menentukan *interface* keluar yang sebenarnya dan alamat *NEXT-HOP* langsung yang harus digunakan untuk meneruskan paket transit ke tujuan.

Hasil konfigurasinya sebagai berikut:

```
chain=NEXTHOP-ITS GATE invert-match=no action=passthrough
    set-out-nexthop=103.78.234.66 set-bgp-prepend-path=""
```

Langkah selanjutnya ialah konfigurasi pada *peer*. Untuk iBGP, konfigurasinya ialah menulis nama, *instance*, *remote address*, *remote* AS, *nexthop-choice*, *out filter* dan *use* BFD. *Nexthop-choice=force-self* berfungsi agar *local address* dari *interface* yang digunakan terhubung ke *peer*. *Out-filter* yang dipilih ialah nama dari *routing filter chain* yang diterapkan ke informasi perutean keluar. IP yang diatur pada *out-filter* merupakan alamat IP pada *gateway* yang keluar dari BGP *speaker* menuju *neighbor* tujuan. Konfigurasi *peer* menggunakan BFD sesuai dengan RFC 5880 yang berfungsi untuk mendeteksi kesalahan dalam jalur dua arah pada internal BGP dengan cepat. Hal tersebut dilakukan karena iBGP rawan terjadi looping. Hasil konfigurasinya sebagai berikut:

```
name="peer to gate ITS" instance=idren remote
address=103.78.234.65 remote-as=64302 tcp-md5-key=""
nexthop-choice=force-self multihop=no  route-reflect=no hold-
time=3m ttl=default in-filter="" out-filter=NEXTHOP-ITS GATE
address-families=ip default-originate=never remove-private-as=no
as-override=no passive=no use-bfd=yes remote-
id=103.78.235.251 local-address=103.78.234.66 uptime=28m57s
prefix-count=9 updates-sent=5 updates-received=14 withdrawn-
sent=0 withdrawn-received=1 remote-hold-time=3m used-hold-
time=3m used-keepalive-time=1m refresh-capability=yes as4-
capability=yes state=established
```

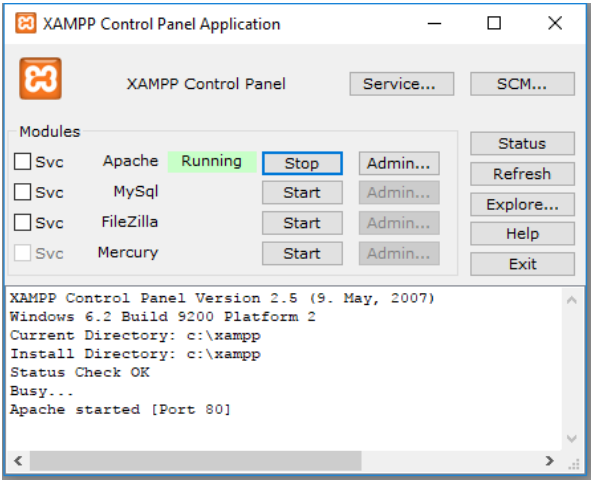
Untuk eBGP, konfigurasinya hanya menulis nama, *instance*, *remote address*, *remote* AS dan *nexthop-choice*.

Langkah selanjutnya ialah konfigurasi pada *network* yang di-*advertise*. *Network* yang di-*advertise* adalah *network* yang terhubung ke *gateway* BGP *speaker*. Setelah semua konfigurasi selesai, pada *session peer* akan muncul *state established* yang menandakan bahwa BGP

speaker telah terhubung dengan *neighbor*-nya. Untuk berubah status dari *idle* menuju *established* membutuhkan waktu konvergensi.

3.6 Konfigurasi *Server*

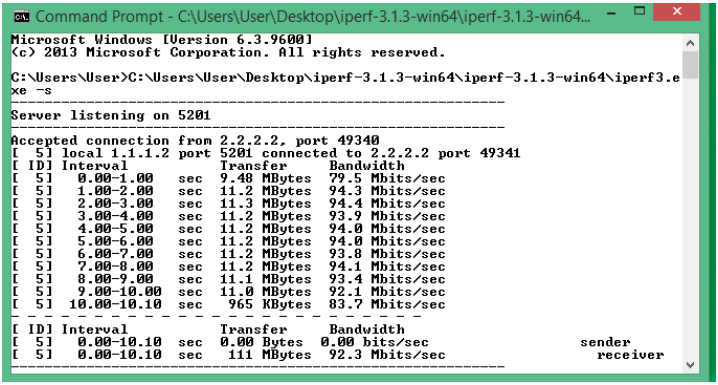
Aplikasi *server* yang digunakan dalam tugas akhir ini adalah XAMPP dan *Iperf*. Konfigurasi XAMPP dilakukan dengan cara meng-*install* XAMPP pada Windows. Setelah itu, memasukkan *file* yang akan di-*download* oleh *client* pada C:\xampp\htdocs. File dapat langsung dimasukkan di folder tersebut dan bisa juga dengan cara membuat *folder* baru dan memasukkan *file* ke *folder* baru tersebut. Langkah selanjutnya adalah menjalankan XAMPP dengan melakukan *running* pada Apache. Apache merupakan sebuah nama web server yang bertanggung jawab pada request-response HTTP dan logging informasi secara *detail*. Gambar XAMPP saat melakukan *running Apache* ditunjukkan pada Gambar 3.3.



Gambar 3.3 XAMPP saat melakukan *running Apache*

Konfigurasi Iperf ialah dengan cara melakukan install Iperf yang telah di-*download* pada situs https://iperf.fr/iperf-download.php. Instalasi *iperf* dapat dilakukan pada Windows. Untuk menjadi *server*, langkah pertama ialah membuka cmd kemudian *file iperf* yang telah didownload ditempatkan pada *desktop*. Setelah jendela cmd terbuka, maka *file* yang ada pada *folder iperf* tersebut ditarik ke cmd. Untuk melakukan pengukuran, iperf yang ter-*install* harus *point to point*. Baik disisi *server*

maupun *client*. Untuk melakukan konfigurasi laptop sebagai *server*, setelah *file iperf* dimasukkan ke cmd, maka hanya perlu ditambahkan –s seperti pada Gambar 3.4.



Gambar 3.4 Konfigurasi *Iperf* sebagai *Server*

3.7 Peralatan Pendukung

Untuk mendukung proses implementasi dan pengujian penelitian ini dibutuhkan peralatan pendukung seperti perangkat keras dan perangkat lunak yang digunakan. Alat yang digunakan dalam penelitian ini terdiri dari dua jenis, yaitu perangkat lunak dan perangkat keras.

3.7.1 Perangkat Lunak

Penelitian ini menggunakan tiga perangkat lunak diantaranya *Winbox*, XAMPP dan *Iperf.*

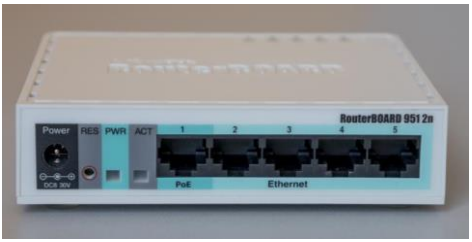
1. *Winbox* merupakan sebuah software yang di gunakan untuk me-*remote* sebuah *server* mikrotik kedalam mode GUI (*Graphical User Interface*) melalui *operating system windows*.
2. XAMPP merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi dan merupakan kompilasi dari beberapa program. Fungsinya ialah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache* HTTP *Server*, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan *Perl*. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, MySQL, PHP dan

Perl. Program ini tersedia dalam *General Public License* (GNU) dan bebas. XAMPP merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis.

3. *Iperf* merupakan perangkat lunak yang akan dipasang pada *host* dengan OS *windows* dan bisa digunakan sebagai *client* maupun *server* untuk mengetes *bandwith* antar *host.*

3.7.2 Perangkat Keras

Untuk meenjalankan *winbox*, diperlukan mikrotik. Mikrotik yang penulis gunakan ialah RB951-2n yang ada di lab B301. Mikrotik routerBOARD RB951-2n memiliki semua kebutuhan *router* dan *gateway* untuk personal dan kantor. Memiliki 5 buah *port ethernet*, 1 buah *access point embedded* 2,4 GHz, *antenna embedded* 1,5 dbi. Sudah termasuk *power adaptor*. Mikrotik routerBOARD RB951-2n terlihat pada Gambar 3.5.



Gambar 3.5 Mikrotik routerBOARD RB951-2n

Perangkat keras yang digunakan dalam penelitian ini adalah laptop PC di LAB B301. Adapun spesifikasi laptop milik penulis ditunjukkan pada Tabel 3.4.

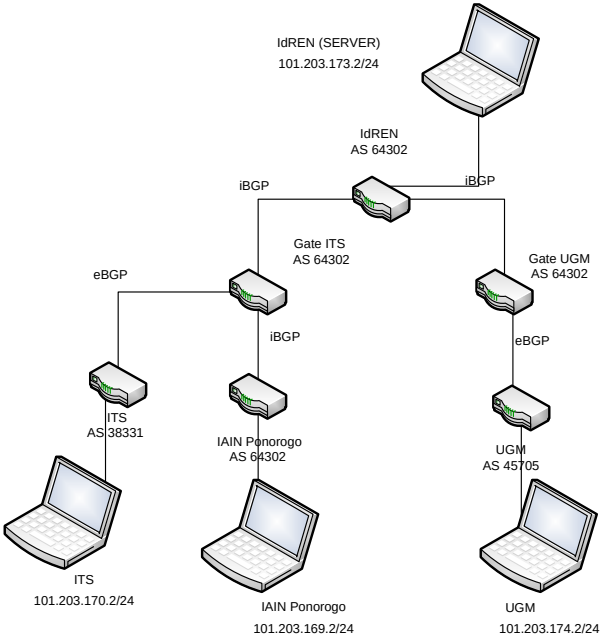
Tabel 3.4 Spesifikasi Laptop

Processor	Intel® Core™ i3-6006U CPU @ 2.00GHz 1.99GHz
Installed Memory (RAM)	4.00GB
System type	64-*Bit Operating System*
Hard disk	500 GB SATA 5400rpm

3.8 Rancangan Pengujian

Pada penelitian ini dilakukan pengujian dalam pemilihan jalur *routing* dengan cara *traceroute* serta melakukan pengujian besarnya *throughput* yang diperoleh saat melakukan *download* ke *server*. Pengujian *throughput* dilakukan tiga kali seperti pada Gambar 3.6:

1. Pengujian *throughput* dari ITS ke *server* IdREN
2. Pengujian *throughput* dari IAIN Ponorogo ke *server* IdREN
3. Pengujian *throughput* dari UGM ke *server* IdREN



Gambar 3.6 *Testbed* Jaringan IdREN dengan *GATE*-ITS dan *GATE*-UGM

BAB 4
PENGUJIAN DAN ANALISA

Bab ini membahas mengenai hasil pengujian dari sistem yang telah dirancang pada bab 3.

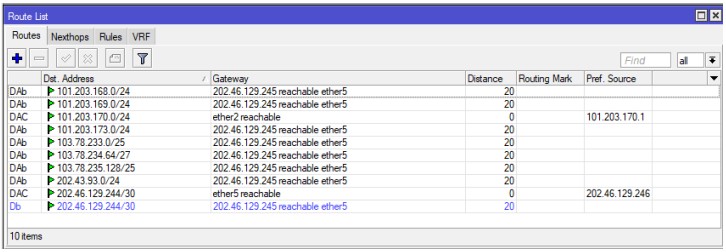
4.1 Hasil *Routing Table* pada ITS

Pada *test bed* ini, dilakukan pengecekan konfigurasi yang telah dilakukan pada BGP *speaker* yakni *router* ITS dengan ASN 38331. Setelah dilakukan konfigurasi BGP pada BGP *session* dan mengatur bagian *nexthop-choice=force-self* serta dilakukan konfigurasi IP *prefix* pada bagian *network*, maka diperoleh *routing table* seperti pada Gambar 4.1. Konfigurasi *nexthop-choice=force-self* berfungsi agar IP *address* dari *interface* yang digunakan terhubung ke *peer* sedangkan konfigurasi IP *prefix* berfungsi agar saat *router* ITS akan mengirimkan *advertisement* kepada *router* yang lain, *router* yang lain mengetahui bahwa *router* ITS telah terhubung dengan *network* 101.203.170.0/24 dan 202.46.129.244/30. Ketika *router* lain sudah menerima informasi tersebut, *router* lain akan menambahkan dirinya ke dalam *advertisement* dari *router* ITS dan mengirimkannya ke *router* selanjutnya. Namun pada saat *router* lain ingin mengirimkan *advertisement* ke *router* ITS untuk memberitahukan bahwa dirinya sudah terhubung, akan ditolak oleh *router* ITS karena di dalam isi *advertisement* tersebut sudah terdapat *router* ITS. Hal ini dilakukan untuk menghindari terjadinya *looping*.

Pada Gambar 4.1 terlihat bahwa *prefix* 101.203.170.0/24 memiliki *flag Dynamic Active Connected* (DAC). Hal ini menunjukkan bahwa laptop yang terhubung dengan *router* ITS yang memiliki ASN 38331 pada *ether2* sudah terhubung dengan baik. Kemudian untuk *prefix* 101.203.173.0/24 memiliki *flag Dynamic Active bgp* (DAb). Hal ini menunjukkan bahwa laptop yang terhubung dengan *router* ITS pada *ether2* sudah terhubung dengan *server* IdREN yang memiliki ASN 64302 menggunakan konfigurasi BGP. *Gateway* yang dilalui oleh ITS ialah 202.46.129.245 yakni *gateway* milik *GATE*-ITS yang terhubung ke *router* IdREN. Jalur yang dilalui ITS agar sampai ke *server* IdREN ialah melewati *prefix* 202.46.129.244/30 dengan *administrative distance* (AD) sebesar 20 karena konfigurasi dari ITS ke GATE-ITS menggunakan konfigurasi eksternal BGP.

Untuk *prefix* 101.203.169.0/24 milik IAIN Ponorogo juga memiliki *flag Dynamic Active bgp* (DAb). Hal ini menunjukkan bahwa laptop yang

terhubung dengan *router* ITS pada *ether2* sudah terhubung dengan IAIN Ponorogo menggunakan konfigurasi BGP. *Gateway* yang dilalui oleh ITS ialah 202.46.129.245 yakni *gateway* milik *GATE*-ITS yang terhubung ke *router* IAIN Ponorogo. Jalur yang dilalui ITS agar sampai ke IAIN Ponorogo ialah melewati *prefix* 202.46.129.244/30 dengan AD sebesar 20 karena konfigurasi dari ITS ke GATE-ITS menggunakan konfigurasi eksternal BGP.



Gambar 4.1 Hasil *routing table* pada ITS

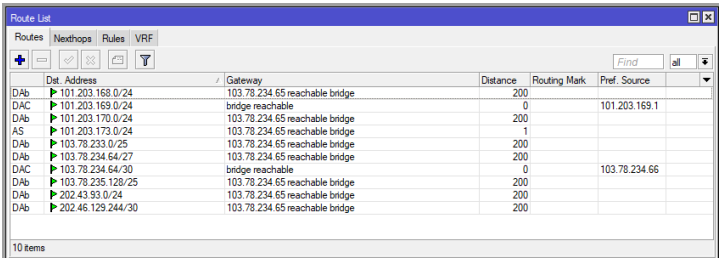
4.2 Hasil *Routing Table* pada IAIN Ponorogo

Pada *test bed* ini, dilakukan pengecekan konfigurasi yang telah dilakukan. Gambar 4.2 menunjukkan hasil dari *routing table* pada router IAIN Ponorogo yang telah dikonfigurasi. Setelah dilakukan konfigurasi BGP pada BGP *session* dan mengatur bagian *nexthop-choice=force-self*, *out-filter*, penggunaan BFD dan dilakukan konfigurasi IP *prefix* pada bagian *network*, maka diperoleh *routing table* seperti pada Gambar 4.2. Konfigurasi *nexthop-choice=force-self* berfungsi agar IP *address* dari *interface* yang digunakan terhubung ke *peer*. Konfigurasi *out filter set out next hop* sesuai dengan RFC 4271, bertugas mendefinisikan IP *address* dari sebuah *router* yang harus digunakan sebagai *hop* berikutnya ke tujuan yang tercantum dalam pesan *UPDATE*. Atribut *NEXT_HOP* digunakan oleh BGP *speaker* untuk menentukan *interface* keluar yang sebenarnya dan alamat *NEXT-HOP* langsung yang harus digunakan untuk meneruskan paket transit ke tujuan agar tidak terjadi *looping*. Konfigurasi *peer* menggunakan BFD sesuai dengan RFC 5880 berfungsi untuk mendeteksi kesalahan dalam jalur dua arah pada internal BGP dengan cepat. Sedangkan konfigurasi IP *prefix* berfungsi agar saat *router* IAIN Ponorogo akan mengirimkan *advertisement* kepada *router* yang lain, *router* yang lain mengetahui bahwa *router* IAIN Ponorogo telah terhubung dengan *network* 101.203.169.0/24 dan 103.78.234.64/27.

Ketika *router* lain sudah menerima informasi tersebut, *router* lain akan menambahkan dirinya ke dalam *advertisement* dari *router* IAIN Ponorogo dan mengirimkannya ke *router* selanjutnya. Namun pada saat *router* lain ingin mengirimkan *advertisement* ke *router* IAIN Ponorogo untuk memberitahukan bahwa dirinya sudah terhubung, akan ditolak oleh *router* IAIN Ponorogo karena di dalam isi *advertisement* tersebut sudah terdapat *router* IAIN Ponorogo. Hal ini dilakukan untuk menghindari terjadinya *looping*.

Pada Gambar 4.2 terlihat bahwa *prefix* 101.203.169.0/24 memiliki *flag Dynamic Active Connected* (DAC). Hal ini menunjukkan bahwa laptop yang terhubung dengan router IAIN Ponorogo pada *ether2* sudah terhubung dengan baik. Kemudian untuk *prefix* 101.203.173.0/24 memiliki *flag Active Static (AS).* Hal ini menunjukkan bahwa laptop yang terhubung dengan *router* ITS pada *ether2* sudah terhubung dengan *server* IdREN menggunakan konfigurasi statis karena IAIN Ponorogo yang tidak memiliki ASN dan menggunakan konfigurasi internal BGP tidak terhubung langsung dengan IdREN. *Peering* internal BGP harus menggunakan topologi *full mesh* antar router. Hal ini karena antar *router* yang terhubung dengan internal BGP tidak dapat meneruskan informasi *routing* dari satu internal BGP *peer* pada internal BGP *peer* yang lain. *Prefix* harus dikirim langsung oleh pengirim informasi dan diterima langsung oleh penerima informasi. *Gateway* yang dilalui oleh IAIN Ponorogo ialah 103.78.234.65 yakni *gateway* milik *GATE*-ITS yang terhubung ke *router* IdREN. Jalur yang dilalui IAIN Ponorogo agar sampai ke *server* IdREN ialah melewati *prefix* 103.78.243.64/27 dengan AD sebesar 200 karena konfigurasi dari IAIN Ponorogo ke GATE-ITS menggunakan konfigurasi internal BGP.

Untuk *prefix* 101.203.170.0/24 milik ITS juga memiliki *flag Dynamic Active bgp (DAb).* Hal ini menunjukkan bahwa laptop yang terhubung dengan *router* IAIN Ponorogo pada *ether2* sudah terhubung dengan ITS menggunakan konfigurasi BGP. *Gateway* yang dilalui oleh IAIN Ponorogo ialah 103.78.234.65 yakni *gateway* milik *GATE*-ITS yang terhubung ke *router* ITS. Jalur yang dilalui IAIN Ponorogo agar sampai ke ITS ialah melewati *prefix* 103.78.243.64/27 dengan AD sebesar 200 karena konfigurasi dari IAIN Ponorogo ke *GATE*-ITS menggunakan konfigurasi internal BGP.



Gambar 4.2 Hasil *routing table* pada IAIN Ponorogo

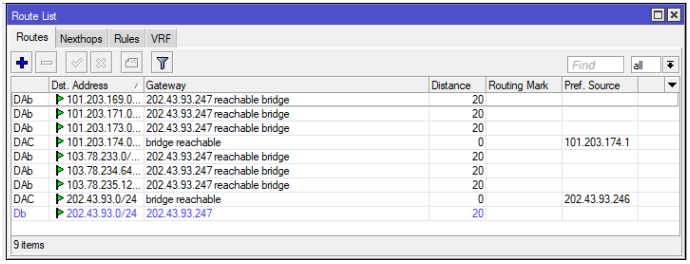
4.3 Hasil *Routing Table* pada UGM

Pada *test bed* ini, dilakukan pengecekan konfigurasi yang telah dilakukan pada BGP *speaker* yakni *router* UGM dengan ASN 45705. Setelah dilakukan konfigurasi BGP pada BGP *session* dan mengatur bagian *nexthop-choice=force-self* serta dilakukan konfigurasi IP *prefix* pada bagian *network*, maka diperoleh *routing table* seperti pada Gambar 4.3. Konfigurasi *nexthop-choice=force-self* berfungsi agar IP *address* dari *interface* yang digunakan terhubung ke *peer* sedangkan konfigurasi IP *prefix* berfungsi agar saat *router* UGM akan mengirimkan *advertisement* kepada *router* yang lain, *router* yang lain mengetahui bahwa *router* UGM telah terhubung dengan *network* 101.203.174.0/24 dan 202.43.93.0/24. Ketika *router* lain sudah menerima informasi tersebut, *router* lain akan menambahkan dirinya ke dalam *advertisement* dari *router* UGM dan mengirimkannya ke *router* selanjutnya. Namun pada saat *router* lain ingin mengirimkan *advertisement* ke *router* UGM untuk memberitahukan bahwa dirinya sudah terhubung, akan ditolak oleh *router* UGM karena di dalam isi *advertisement* tersebut sudah terdapat *router* UGM. Hal ini dilakukan untuk menghindari terjadinya *looping*.

Gambar 4.3 menunjukkan hasil dari *routing table* pada *router* UGM yang telah dikonfigurasi. Pada Gambar 4.3 terlihat bahwa *prefix* 101.203.174.0/24 memiliki *flag Dynamic Active Connected* (DAC). Hal ini menunjukkan bahwa laptop yang terhubung dengan router UGM pada *ether3* sudah terhubung dengan baik. Kemudian untuk *prefix* 101.203.173.0/24 memiliki *flag Dynamic Active bgp (DAb).* Hal ini menunjukkan bahwa laptop yang terhubung dengan *router* UGM pada *ether3* sudah terhubung dengan *server* IdREN menggunakan konfigurasi BGP. *Gateway* yang dilalui oleh UGM ialah 202.43.93.247 yakni *gateway* milik *GATE*-UGM yang terhubung ke *router* IdREN. Jalur yang dilalui UGM agar sampai ke *server* IdREN ialah melewati *prefix*

202.43.93.0/24 dengan AD sebesar 20 karena konfigurasi dari UGM ke *GATE*-UGM menggunakan konfigurasi eksternal BGP.

Untuk *prefix* 101.203.169.0/24 milik IAIN Ponorogo juga memiliki *flag Dynamic Active bgp (DAb).* Hal ini menunjukkan bahwa laptop yang terhubung dengan *router* UGM pada *ether2* sudah terhubung dengan IAIN Ponorogo menggunakan konfigurasi BGP. *Gateway* yang dilalui oleh UGM ialah 202.43.93.247 yakni *gateway* milik *GATE*-UGM yang terhubung ke *router* IdREN. Jalur yang dilalui UGM agar sampai ke IAIN Ponorogo ialah melewati *prefix* 202.43.93.0/24 dengan AD sebesar 20 karena konfigurasi dari UGM ke *GATE*-UGM menggunakan konfigurasi eksternal BGP.



Gambar 4.3 Hasil *routing table* pada UGM

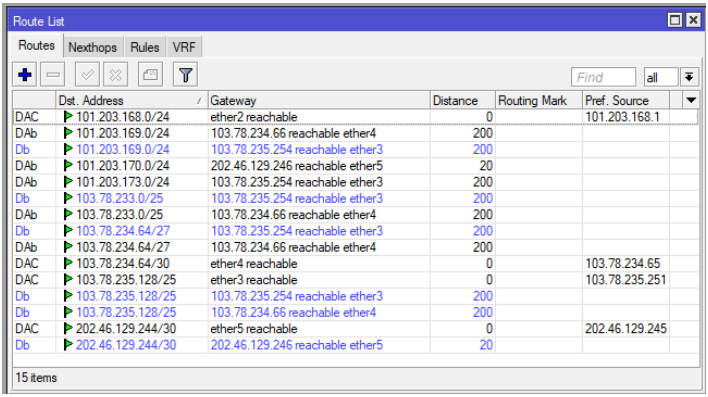
4.4 Hasil *Routing Table* pada *GATE*-ITS

Pada *test bed* ini, dilakukan pengecekan konfigurasi yang telah dilakukan. Gambar 4.4 menunjukkan hasil dari *routing table* pada router *GATE*-ITS yang telah dikonfigurasi. Pada Gambar 4.4 terlihat bahwa *prefix* 101.203.168.0/24 memiliki *flag Dynamic Active Connected* (DAC). Hal ini menunjukkan bahwa laptop yang terhubung dengan router *GATE*-ITS pada *ether2* sudah terhubung dengan baik. Kemudian untuk *prefix* 101.203.169.0/24 memiliki *flag Dynamic Active bgp (DAb).* Hal ini menunjukkan bahwa laptop yang terhubung dengan *router GATE*-ITS pada *ether2* sudah terhubung dengan laptop pada *router* IAIN Ponorogo menggunakan konfigurasi internal BGP. *Gateway* yang dilalui oleh *GATE*-ITS ialah 103.78.234.66 yakni *gateway* milik *router* IAIN Ponorogo. Jalur yang dilalui *GATE*-ITS agar sampai ke *router* IAIN Ponorogo ialah melewati *prefix* 103.78.234.64/27 dengan AD sebesar 200 karena konfigurasi dari *GATE*-ITS ke IAIN Ponorogo menggunakan konfigurasi internal BGP.

Untuk *prefix* 101.203.170.0/24 milik ITS juga memiliki *flag Dynamic Active bgp (DAb).* Hal ini menunjukkan bahwa laptop yang

terhubung dengan *router GATE*-ITS pada *ether2* sudah terhubung dengan ITS menggunakan konfigurasi BGP. *Gateway* yang dilalui oleh *GATE*-ITS ialah 202.46.129.246 yakni *gateway* milik ITS. Jalur yang dilalui *GATE*-ITS agar sampai ke ITS ialah melewati *prefix* 202.46.129.244/30 dengan AD sebesar 20 karena konfigurasi dari *GATE*-ITS ke ITS menggunakan konfigurasi eksternal BGP.

Untuk *prefix* 101.203.173.0/24 milik *server* IdREN juga memiliki *flag Dynamic Active bgp (DAb).* Hal ini menunjukkan bahwa laptop yang terhubung dengan *router GATE*-ITS pada *ether2* sudah terhubung dengan *server* IdREN menggunakan konfigurasi BGP. *Gateway* yang dilalui oleh *GATE*-ITS ialah 103.78.235.254 yakni *gateway* milik IdREN. Jalur yang dilalui *GATE*-ITS agar sampai ke *server* IdREN ialah melewati *prefix* 103.78.235.128/25 dengan AD sebesar 200 karena konfigurasi dari *GATE*-ITS ke *server* IdREN menggunakan konfigurasi internal BGP.



Gambar 4.4 Hasil *routing table* pada *GATE*-ITS

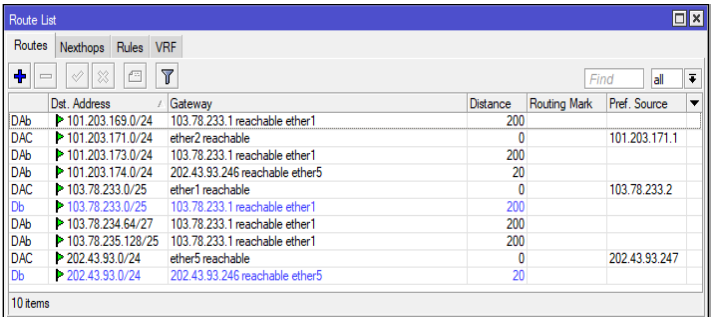
4.5 Hasil *Routing Table* pada *GATE*-UGM

Pada *test bed* ini, dilakukan pengecekan konfigurasi yang telah dilakukan. Gambar 4.5 menunjukkan hasil dari *routing table* pada router *GATE*-UGM yang telah dikonfigurasi. Pada Gambar 4.5 terlihat bahwa *prefix* 101.203.171.0/24 memiliki *flag Dynamic Active Connected* (DAC). Hal ini menunjukkan bahwa laptop yang terhubung dengan router *GATE*-UGM pada *ether2* sudah terhubung dengan baik. Kemudian untuk *prefix* 101.203.169.0/24 memiliki *flag Dynamic Active bgp (DAb).* Hal ini menunjukkan bahwa laptop yang terhubung dengan *router GATE*-UGM

pada *ether2* sudah terhubung dengan laptop pada *router* IAIN Ponorogo menggunakan konfigurasi internal BGP. *Gateway* yang dilalui oleh *GATE*-UGM ialah 103.78.233.1 yakni *gateway* milik *router* IdREN. Jalur yang dilalui *GATE*-UGM agar sampai ke *router* IAIN Ponorogo ialah melewati *prefix* 103.78.233.0/25 dengan AD sebesar 200 karena konfigurasi dari *GATE*-UGM ke IdREN menggunakan konfigurasi internal BGP.

Untuk *prefix* 101.203.173.0/24 milik *server* IdREN juga memiliki *flag Dynamic Active bgp (DAb).* Hal ini menunjukkan bahwa laptop yang terhubung dengan *router GATE*-UGM pada *ether2* sudah terhubung dengan *server* IdREN menggunakan konfigurasi BGP. *Gateway* yang dilalui oleh *GATE*-UGM ialah 103.78.233.1 yakni *gateway* milik IdREN. Jalur yang dilalui *GATE*-UGM agar sampai ke *server* IdREN ialah melewati *prefix* 103.78.233.0/25 dengan AD sebesar 200 karena konfigurasi dari *GATE*-UGM ke *server* IdREN menggunakan konfigurasi internal BGP.

Untuk *prefix* 101.203.174.0/24 milik UGM juga memiliki *flag Dynamic Active bgp (DAb).* Hal ini menunjukkan bahwa laptop yang terhubung dengan *router GATE*-UGM pada *ether2* sudah terhubung dengan UGM menggunakan konfigurasi BGP. *Gateway* yang dilalui oleh *GATE*-UGM ialah 202.43.93.246 yakni *gateway* milik UGM. Jalur yang dilalui *GATE*-UGM agar sampai ke UGM ialah melewati *prefix* 202.43.93.0/24 dengan AD sebesar 20 karena konfigurasi dari *GATE*-UGM ke UGM menggunakan konfigurasi eksternal BGP.



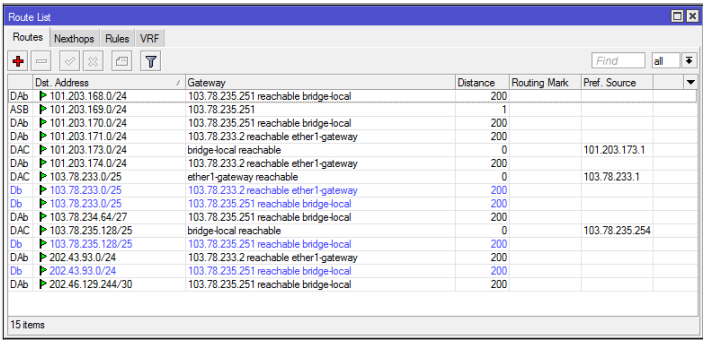
Gambar 4.5 Hasil *routing table* pada *GATE*-UGM

4.6 Hasil *Routing Table* pada IdREN

Pada *test bed* ini, dilakukan pengecekan konfigurasi yang telah dilakukan. Gambar 4.6 menunjukkan hasil dari *routing table* pada router IdREN yang telah dikonfigurasi. Pada Gambar 4.6 terlihat bahwa *prefix* 101.203.173.0/24 memiliki *flag Dynamic Active Connected* (DAC). Hal ini menunjukkan bahwa laptop yang terhubung dengan router IdREN pada *ether2* sudah terhubung dengan baik. Kemudian untuk *prefix* 101.203.168.0/24 memiliki *flag Dynamic Active bgp* (DAb). Hal ini menunjukkan bahwa laptop yang terhubung dengan *router* IdREN pada *ether2* sudah terhubung dengan laptop pada *router GATE*-ITS menggunakan konfigurasi internal BGP. *Gateway* yang dilalui oleh IdREN ialah 103.78.235.251 yakni *gateway* milik *router GATE*-ITS. Jalur yang dilalui IdREN agar sampai ke *router GATE*-ITS ialah melewati *prefix* 103.78.235.128/25 dengan AD sebesar 200 karena konfigurasi dari *GATE*-UGM ke IdREN menggunakan konfigurasi internal BGP.

Untuk *prefix* 101.203.173.0/24 milik *server* IdREN juga memiliki *flag Dynamic Active bgp (DAb).* Hal ini menunjukkan bahwa laptop yang terhubung dengan *router GATE*-UGM pada *ether2* sudah terhubung dengan *server* IdREN menggunakan konfigurasi BGP. *Gateway* yang dilalui oleh *GATE*-UGM ialah 103.78.233.1 yakni *gateway* milik IdREN. Jalur yang dilalui *GATE*-UGM agar sampai ke *server* IdREN ialah melewati *prefix* 103.78.233.0/25 dengan AD sebesar 200 karena konfigurasi dari *GATE*-UGM ke *server* IdREN menggunakan konfigurasi internal BGP.

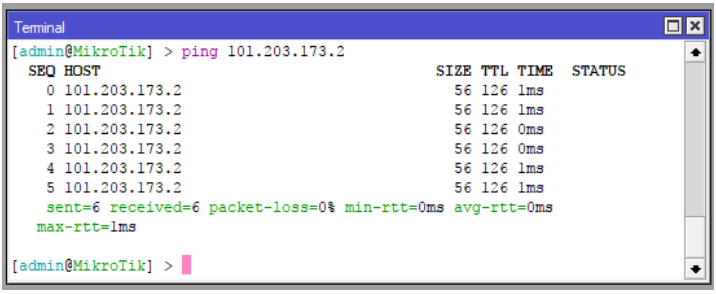
Untuk *prefix* 101.203.174.0/24 milik UGM juga memiliki *flag Dynamic Active bgp (DAb).* Hal ini menunjukkan bahwa laptop yang terhubung dengan *router GATE*-UGM pada *ether2* sudah terhubung dengan UGM menggunakan konfigurasi BGP. *Gateway* yang dilalui oleh *GATE*-UGM ialah 202.43.93.246 yakni *gateway* milik UGM. Jalur yang dilalui *GATE*-UGM agar sampai ke UGM ialah melewati *prefix* 202.43.93.0/24 dengan AD sebesar 20 karena konfigurasi dari *GATE*-UGM ke UGM menggunakan konfigurasi eksternal BGP.



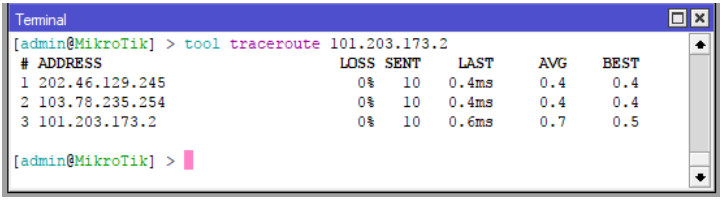
Gambar 4.6 Hasil *routing table* pada IdREN

4.7 Hasil *Ping* dan *Traceroute* dari ITS ke *Server* IdREN

Pada *test bed* ini, dilakukan pengujian *ping* dan *traceroute* dari ITS ke *server* IdREN. Pengujian *ping* dilakukan untuk mengetahui bahwa ITS sudah terhubung dengan *server* IdREN. Pengujian *traceroute* dilakukan untuk mengetahui jalur yang ditempuh untuk mencapai *server*. Hasil *ping* dan *traceroute* yang diperoleh ditunjukkan pada Gambar 4.7 dan Gambar 4.8.



Gambar 4.7 Hasil *ping* dari ITS ke *server* IdREN

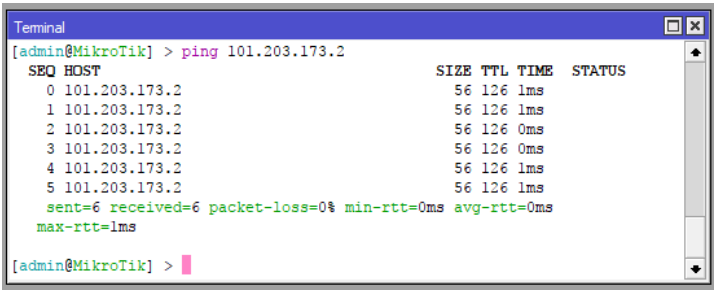


Gambar 4.8 Hasil *traceroute* dari ITS ke *server* IdREN

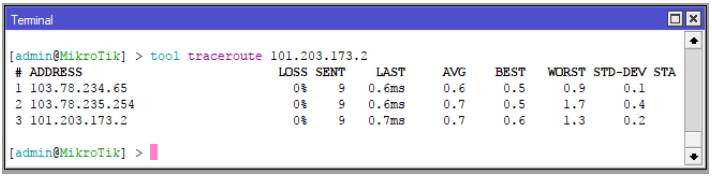
Dari hasil pengujian *traceroute* dapat diketahui bahwa saat ITS ingin terhubung dengan *server* IdREN, *gateway* yang dilalui adalah 202.46.129.245 milik *GATE*-ITS kemudian *gateway* 103.78.235.254 milik IdREN dan 101.203.173.2 yang merupakan IP milik *server* IdREN. Rute tersebut diperoleh dengan konfigurasi eksternal BGP dari ITS menuju *GATE*-ITS dan internal BGP dari *GATE*-ITS IdREN.

4.8 Hasil *Ping* dan *Traceroute* dari IAIN ke *Server* IdREN

Pada *test bed* ini, dilakukan pengujian *ping* dan *traceroute* dari IAIN Ponorogo ke *server* IdREN. Hasil *ping* dan *traceroute* yang diperoleh ditunjukkan pada Gambar 4.9 dan Gambar 4.10.



Gambar 4.9 Hasil *ping* dari IAIN Ponorogo ke *server* IdREN

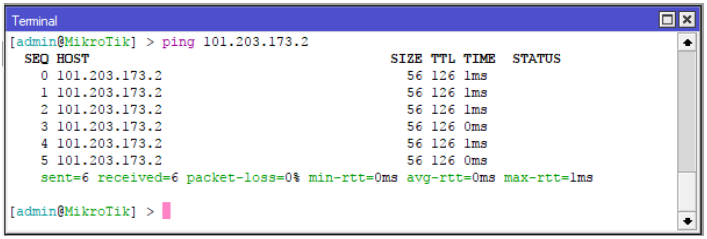


Gambar 4.10 Hasil *traceroute* dari IAIN Ponorogo ke *server* IdREN

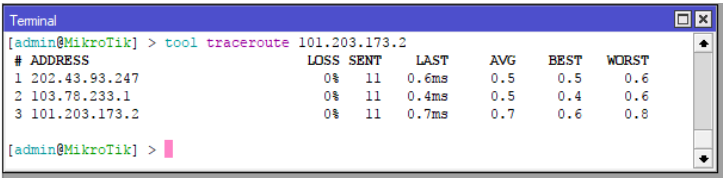
Dari hasil pengujian *traceroute* dapat diketahui bahwa saat IAIN Ponorogo ingin terhubung dengan *server* IdREN, *gateway* yang dilalui adalah 103.78.234.56 milik *GATE*-ITS kemudian *gateway* 103.78.235.254 milik IdREN dan 101.203.173.2 yang merupakan IP milik *server* IdREN. Rute tersebut diperoleh dengan konfigurasi internal BGP dari IAIN Ponorogo menuju *GATE*-ITS dan internal BGP dari *GATE*-ITS IdREN.

4.9 Hasil *Ping* dan *Traceroute* dari UGM ke *Server* IdREN

Pada *test bed* ini, dilakukan pengujian *ping* dan *traceroute* dari UGM ke *server* IdREN. Hasil *ping* dan *traceroute* yang diperoleh ditunjukkan pada Gambar 4.11 dan Gambar 4.12.



Gambar 4.11 Hasil *ping* dari UGM ke *server* IdREN

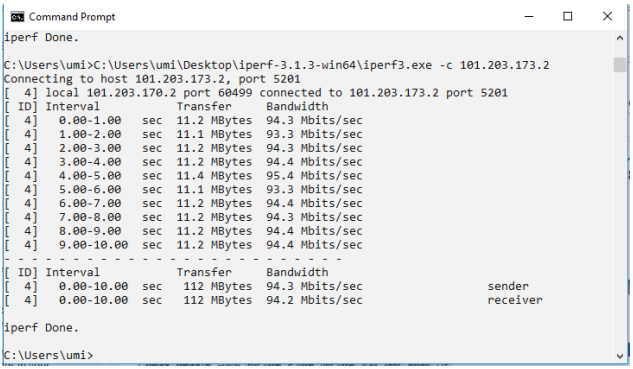


Gambar 4.12 Hasil *traceroute* dari UGM ke *server* IdREN

Dari hasil pengujian *traceroute* dapat diketahui bahwa saat UGM ingin terhubung dengan *server* IdREN, *gateway* yang dilalui adalah 202.43.93.247 milik *GATE*-UGM kemudian *gateway* 103.78.235.254 milik IdREN dan 101.203.173.2 yang merupakan IP milik *server* IdREN. Rute tersebut diperoleh dengan konfigurasi eksternal BGP dari UGM menuju *GATE*-UGM dan internal BGP dari *GATE*-ITS IdREN.

4.10 Hasil *Test Bandwidth* ITS dan *Server* IdREN menggunakan *Iperf*

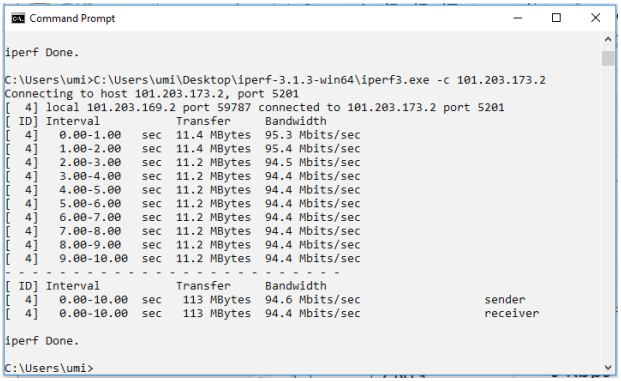
Pada *test bed* ini, dilakukan pengetesan *bandwidth* yang terkirim dari *server* IdREN ke *client* ITS pada saat kondisi trafik kosong atau tidak ada data yang lewat. Pengujian *bandwidth* ini menggunakan kabel *ethernet* cat5. Berdasarkan TIA/EIA-568-B, kabel UTP cat5 hanya dapat melakukan transmisi data sebesar 100 Mbps. Hasil *test bandwidth* yang diperoleh ialah 92.4 Mbps seperti yang ditunjukkan pada Gambar 4.13.



Gambar 4.13 Hasil *test bandwidth* ITS dan *server* IdREN menggunakan *Iperf*

4.11 Hasil *Test Bandwidth* IAIN Ponorogo dan *Server* IdREN menggunakan *Iperf*

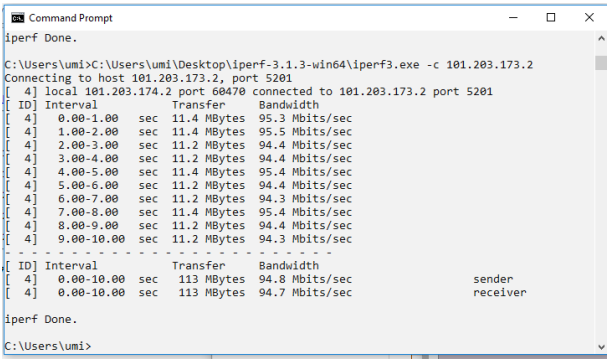
Pada *test bed* ini, dilakukan pengetesan *bandwidth* yang terkirim dari *server* IdREN ke *client* IAIN Ponorogo pada saat kondisi trafik kosong atau tidak ada data yang lewat. Pengujian *bandwidth* ini menggunakan kabel *ethernet* cat5. Berdasarkan TIA/EIA-568-B, kabel UTP cat5 hanya dapat melakukan transmisi data sebesar 100 Mbps. Hasil *test bandwidth* yang diperoleh ialah 94.4 Mbps seperti yang ditunjukkan pada Gambar 4.14.



Gambar 4.14 Hasil *test bandwidth* IAIN Ponorogo dan *server* IdREN menggunakan *Iperf*

4.12 Hasil *Test Bandwidth* UGM dan *Server* IdREN menggunakan *Iperf*

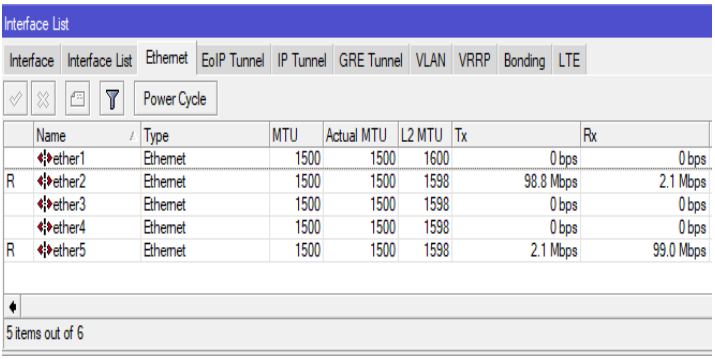
Pada *test bed* ini, dilakukan pengetesan *bandwidth* yang terkirim dari *server* IdREN ke *client* UGM pada saat kondisi trafik kosong atau tidak ada data yang lewat. Pengujian *bandwidth* ini menggunakan kabel *ethernet* cat5. Berdasarkan TIA/EIA-568-B, kabel UTP cat5 hanya dapat melakukan transmisi data sebesar 100 Mbps. Hasil *test bandwidth* yang diperoleh ialah 94.7 Mbps seperti yang ditunjukkan pada Gambar 4.15.



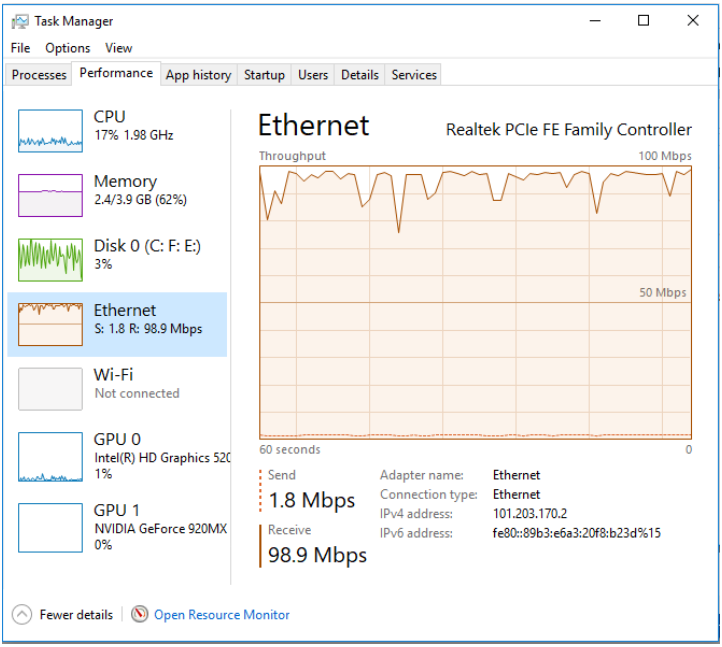
Gambar 4.15 Hasil *test bandwidth* UGM dan *server* IdREN menggunakan *Iperf*

4.13 Hasil Pengujian *Throughput* saat ITS mengakses *file* ke *Server* IdREN

Pada *test bed* ini, dilakukan pengujian *throughput* saat ITS mengakses *file* ke *server* IdREN. Pengujian *throughput* ini menggunakan kabel *ethernet* cat5. Berdasarkan TIA/EIA-568-B, kabel UTP cat5 hanya dapat melakukan transmisi data sebesar 100 Mbps. Nilai *throughput* yang diperoleh ditunjukkan pada Gambar 4.16 yakni 98.8 Mbps terlihat pada *interface* mikrotik *ether2* yang terhubung langsung pada laptop. Sedangkan nilai *throughput* pada *ether5* yang yang merupakan jalur ITS dengan *GATE*-ITS ialah 99.0 Mbps. Kemudian untuk nilai *throughput* yang ada pada *task manager* ialah 98.9 Mbps seperti yang terlihat pada Gambar 4.17.



Gambar 4.16 Hasil pengujian *throughput* pada *Interface* Mikrotik saat ITS mengakses *file* ke *Server* IdREN



Gambar 4.17 Hasil pengujian *throughput* pada *Task Manager* saat ITS mengakses *file* ke *Server* IdREN

4.14 Hasil Pengukuran *Througput* Tiap Perguruan Tinggi

Hasil pengukuran *throughput* tiap perguruan tinggi saat tiap perguruan tinggi mengakses *fil*e ke *server* IdREN. Pengujian *throughput* ini menggunakan kabel *ethernet* cat5. Berdasarkan TIA/EIA-568-B, kabel UTP cat5 hanya dapat melakukan transmisi data sebesar 100 Mbps. Nilai *throughput* yang diperoleh ditunjukkan pada Tabel 4.1.

Tabel 4.1 Hasil Pengukuran Throughput Tiap Perguruan Tinggi

Nama Perguruan Tinggi	Nilai *Throughput*
UGM	98.9 Mbps
ITS	98.9 Mbps
IAIN Ponorogo	92.4 Mbps

---Halaman ini sengaja dikosongkan---

BAB 5
PENUTUP

5.1 Kesimpulan

Pada pembuatan *test bed* IdREN menggunakan *Border Gateway Protocol*, didapatkan beberapa kesimpulan antara lain:

1. Saat melakukan konfigurasi BGP, diperlukan konfigurasi *nexthop-choice=force-self* yang berfungsi agar IP *address* dari *interface* yang digunakan terhubung ke *peer*.
2. Untuk melakukan konfigurasi internal BGP, diperlukan atribut *NEXT-HOP* sesuai dengan sesuai dengan RFC 4271. Atribut *NEXT-HOP* bertugas untuk mendefinisikan IP *address* dari sebuah *router* yang harus digunakan sebagai *hop* berikutnya ke tujuan yang tercantum dalam pesan *UPDATE*. Atribut *NEXT_HOP* digunakan oleh BGP *speaker* untuk menentukan *interface* keluar yang sebenarnya dan alamat *NEXT-HOP* langsung yang harus digunakan untuk meneruskan paket transit ke tujuan agar tidak terjadi *looping*.
3. Saat melakukan konfigurasi internal BGP, diperlukan konfigurasi *peer* menggunakan BFD sesuai dengan RFC 5880 yang berfungsi untuk mendeteksi kesalahan dalam jalur dua arah pada internal BGP dengan cepat.
4. Saat melakukan konfigurasi BGP, diperlukan konfigurasi IP *prefix* yang berfungsi agar saat BGP *speaker* akan mengirimkan *advertisement* kepada *neighbor*-nya, *neighbor* tersebut akan mengetahui bahwa BGP *speaker* telah terhubung dengan *network* yang di-*advertise.*
5. Konfigurasi statis digunakan saat topologi internal BGP tidak menggunakan topologi *full mesh.*
6. *Throughput* yang diperoleh saat UGM mengakses *file* ke *server* IdREN ialah 98.9 Mbps, saat ITS mengakses *file* ke *server* IdREN ialah 98.9 Mbps dan IAIN Ponorogo mengakses *file* ke *server* IdREN ialah 92.4 Mbps. Pengujian *throughput* menggunakan kabel *ethernet* cat5. Berdasarkan TIA/EIA-568-B, kabel UTP cat5 hanya dapat melakukan transmisi data sebesar 100 Mbps.
7. Berdasarkan tabel *routing* yang diperoleh dan *traceroute* yang dilakukan, sistem yang dibuat sudah berjalan dengan baik.

5.2 Saran

1. Diperlukan penelitian lebih lanjut dengan menggunakan atribut BGP yang lain
2. Diperlukan penelitian lebih lanjut dengan menggunakan perangkat lain seperti Cisco

DAFTAR PUSTAKA

[1] https://www.slideshare.net/IDNOG/23-idnog03-affan-basalamah-itb-ahmad-basuki-unibraw-overview-of-indonesia-research-and-education-network-inherent

[2] Ardian, Fikry., "Simulasi Routing Policy pada Border Gateway Protocol dalam Jaringan Indonesian Research and Education Network (IdREN) Mempergunakan GNS3". Fakultas Teknik Universitas Gadjah Mada, Yogyakarta, 2017.

[3] Kizza, Joseph Migga., "Guide to Computer Network Security". Springer, London, 2015.

[4] Jyoti., Saini, Himanshi., "A Study on Networks and Comparison of Wired , Wireless and Optical Networks," vol. 5, Issue 3, pp. 3801–3809, March, 2017.

[5] Syafrizal, M., "Pengantar Jaringan Komputer", Yogyakarta, 2005.

[6] Sugeng, W.," Jaringan Komputer dengan TCP/IP", Informatika Bandung, 2006.

[7] Ulfa, Maria., Sobri, Muhammad., Seprina, Iin. "Analisis Perbandingan Ipv4 dan Ipv6 dalam Membangun Sebuah Jaringan", Prosiding SNIT pp. A-343, 2014.

[8] Yani, A.," Utility Jaringan Panduan Mengoptimalkan Jaringan Komputer Berbasis Windows", PT. Kawan Pustaka, Tangerang, 2006.

[9] A. Balchunas, "TCP and UDP v1.21," pp. 1–13, 2012.108

[10] S. Mneimneh, "Computer Networks UDP and TCP," 2014.

[11] "RFC 6335: Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry," pp. 1–33, 2011.

[12] alan@ub.ac.id, "IdREN 2017", 2017.

[13] Balchunas, A., "Static vs Dynamic Routing v1.21," <URL: http://www.routeralley.com/guides/static_dynamic_routing.pdf>, 2007 [Diakses: 03-Mei-2018].

[14] White, R., McPherson, D., and Sangli, S., "Practical BGP", 1st ed. Addison-Wesley Professional, 2004.

[15] Doyle, J., "Dynamic Routing Protocol," <URL: http://www.ciscopress.com/articles/article.asp?p=24090&seqNum=3>, 2001. [Diakses: 03-Mei-2018].

[16] Bonaventure, O., Vanbever, L., Van den Shriek, V., Saucez, D., and Hoerdt, M., “Computer Networking: Principles, Protocols and Practice-Link State Routing”.
[17] White, R., McPherson, D., Sangli, S., “Practical BGP”, 1st ed. Addison-Wesley Professional, 2004.
[18] Cisco System, I., “What Is Administrative Distance?”, <URL:https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/15986-admin-distance.html>, 2013.
[19] Alfisyahrin, Yusnia., <URL:https://yusniaalfisyahrin.wordpress .com/2012/12/07/tentang-routing-dinamis/>, 2012 [Diakses: 03-Desember-2017].
[20] Rekhter, Y., and Li, T., “A Border Gateway Protocol”, RFC 1771 (BGP version 4), March 1995.
[21] Stewart, W., “BGP4: Inter-Domain Routing in the Internet”, Addison-Wesley, 1998.
[22] Halabi, B., “Internet Routing Architectures” Cisco Press, 1997.
[23] Bramantyo, Adhi S., "Optimasi Interdomain Routing dengan BGP pada Stub-Multihomed Autonomous System", Teknik Elektro, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, Bandung, 2007.
[24] Hafiz, M. R., “Simulasi Border Gateway Protocol (BGP) pada Ipv6 dalam Proses Pemilihan Jalur Routing,” Skripsi, 2015.
[25] Y. Rekhter, T. Li, and S. Hares, “RFC 4271: A Border Gateway Protocol 4 (BGP-4)”, 2006.
[26] IANA, “Autonomous System (AS) Numbers,” <URL: http://www.iana.org/assignments/as-numbers/as-numbers.xhtml>, 2016. [Diakses: 08-Maret-2018]
[27] Hashem, M. M. A., “Understanding the Research and Education Networks,” 2012.
[28] Dwiharyanto, Y., and Ediansyah, P., “Indonesian Updates INHERENT to IdREN,” 2016.
[29] alan@ub.ac.id, “IdREN 2017”, 2017.
[30] https://wiki.mikrotik.com/wiki/Manual:Routing/BGP
[31] https://wiki.mikrotik.com/wiki/Manual:Routing/BFD
[32] Kozierok, C. M., “The TCP/IP Guide - Performance Measurements: Speed, Bandwidth, Throughput and Latency,” URL:http://www.tcpipguide.com/free/t_PerformanceMeasurements SpeedBandwidthThroughputand.htm., 2015.

[33] Cormen, T., Leiserson, C., Rivest, R., and Stein, C., “Introduction to Algorithms, 2nd Edition”, Cambridge, Massachusetts: McGraw-Hill Book Company, 2001.

---Halaman ini sengaja dikosongkan---

LAMPIRAN A

Departemen Teknik Elektro
Fakultas Teknologi Elektro - ITS

TE 141599 TUGAS AKHIR – 4 SKS

09 FEB 2018

Nama Mahasiswa	: Umi Faridah
Nomor Pokok	: 07111645000021
Bidang Studi	: Telekomunikasi Multimedia
Tugas Diberikan	: Semester Genap 2017/ 2018
Dosen Pembimbing	: 1. Dr. Ir. Achmad Affandi, DEA. 2. Ir. Djoko Suprajitno Rahardjo, MT.

Judul Tugas Akhir : **Implementasi *Border Gateway Protocol* pada *Test Bed* IdREN (*Indonesian Research Education Network*)**
(Impementation of Border Gateway Protocol at IdREN (Indonesian Research Education Network) Test Bed)

Uraian Tugas Akhir :

Indonesia memiliki jaringan yang bernama IdREN. IdREN merupakan *private network* antar institusi riset dan pendidikan di Indonesia. Layanan dan aplikasi yang ada pada IdREN dapat diakses oleh *network* yang tergabung. IdREN memberi kemudahan pada institusi pendidikan di Indonesia untuk berbagi pakai sumber daya pembelajaran yang dimiliki melalui jalur yang lebih aman. Perguruan tinggi lain juga dapat memperoleh hak akses yang sama dengan cara terhubung ke *network* yang telah terhubung dengan IdREN. Tidak semua pihak mendapatkan hak akses untuk hal tersebut. Oleh karena itu, perlu adanya *filter* agar pihak yang tidak memiliki hak akses tidak dapat mengakses data tersebut. *Border Gateway Protocol* merupakan *routing protokol* jenis *Exterior Gateway Protokol* (EGP). *Routing* BGP berfungsi untuk menghubungkan antara *network-network address* yang memiliki *Autonomous System Number* yang berbeda sehingga dapat terkoneksi. BGP berfungsi untuk mengontrol serta mengatur trafik dari sumber berbeda di dalam *network multihome.* Pada Tugas akhir ini akan dibuat *test bed* jaringan IdREN yang menggunakan implementasi BGP untuk mencegah pihak yang tidak memiliki hak akses pada IdREN mengakses sumber daya pembelajaran yang ada.

Dosen Pembimbing I,

Dr. Ir. Achmad Affandi, DEA.
NIP. 196510141990021001

Dosen Pembimbing II,

Ir. Djoko Suprajitno Rahardjo, M.T.
NIP. 195506221987011001

Mengetahui,
Ketua Program Studi S1,

Dedet C. Riawan, ST. M.Eng. Ph.D.
NIP. 197311192000031001

Menyetujui,
Kepala Laboratorium Jaringan Telekomunikasi

Dr. Ir. Achmad Affandi, DEA.
NIP. 196510141990021001

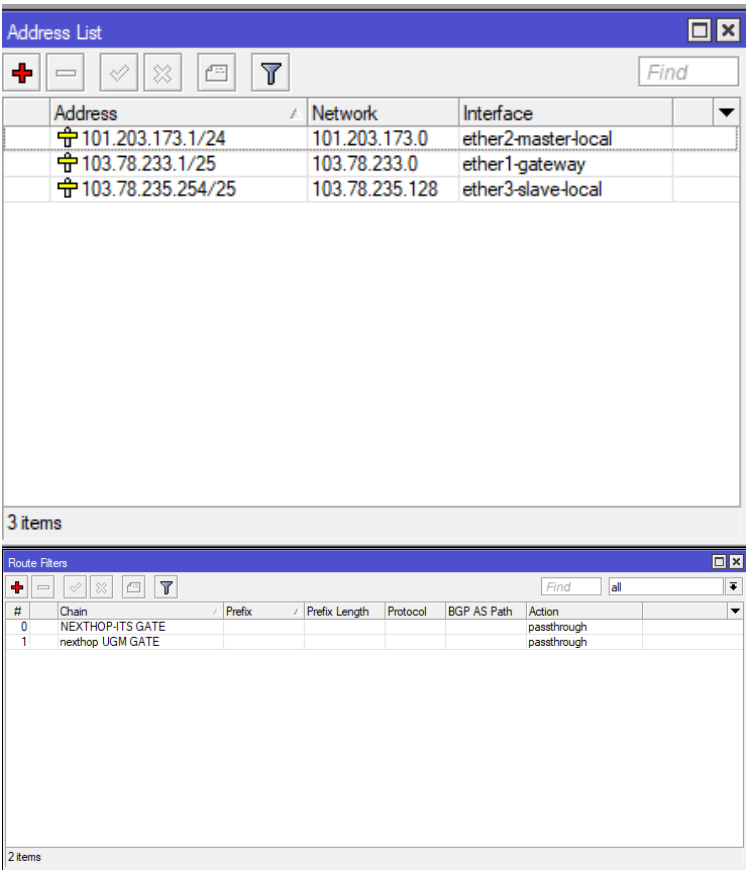
---Halaman ini sengaja dikosongkan---

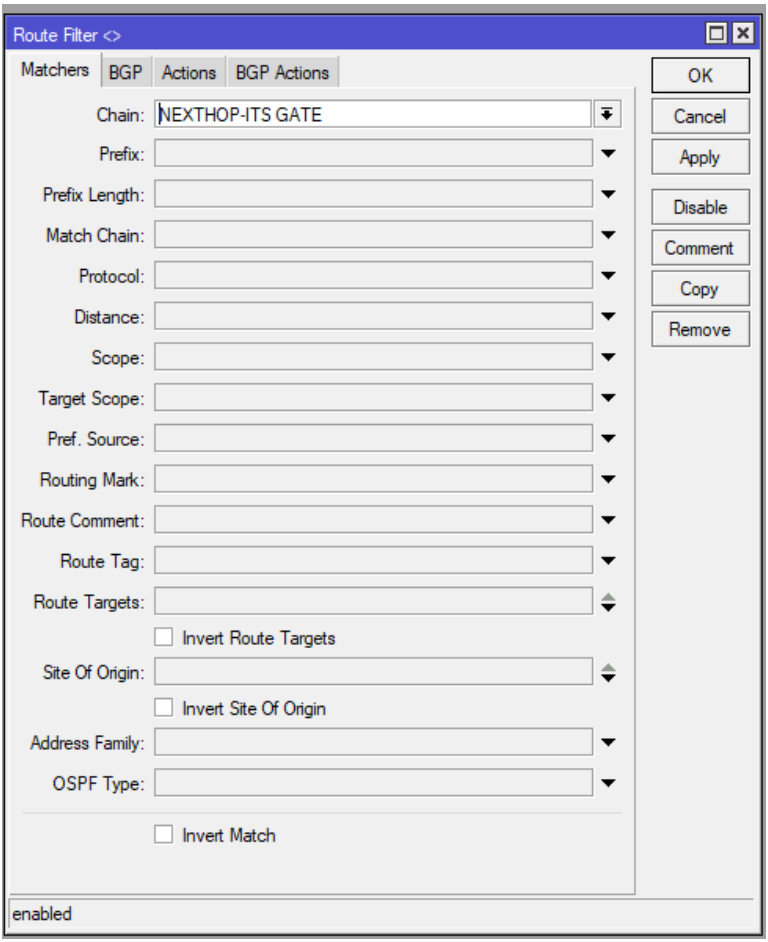
LAMPIRAN B

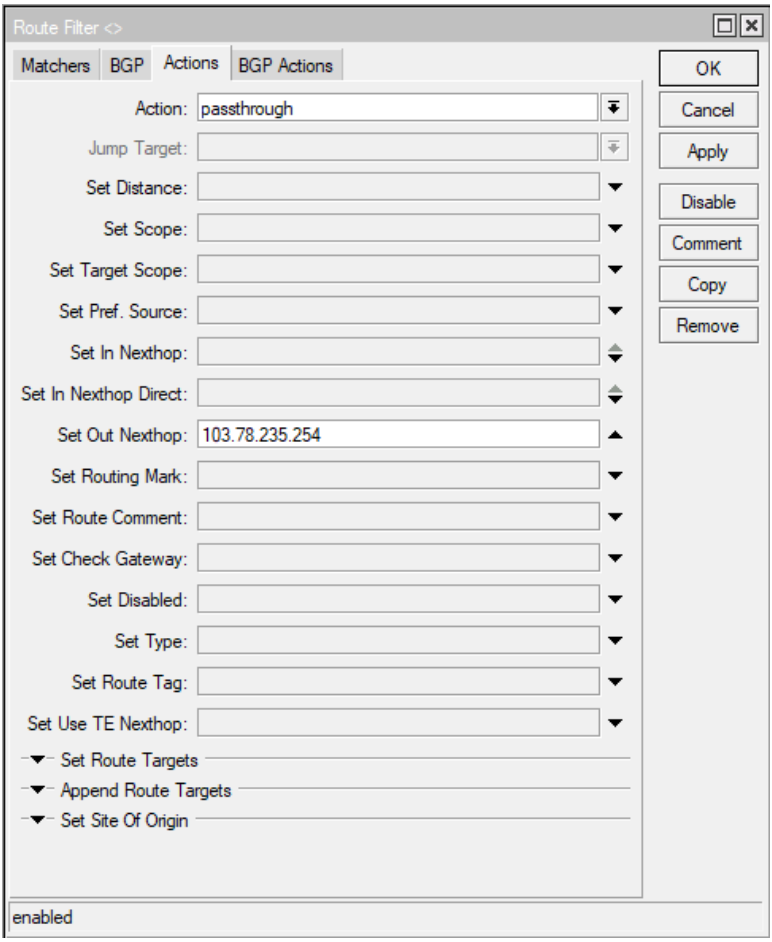
Konfigurasi pada *router* IdREN

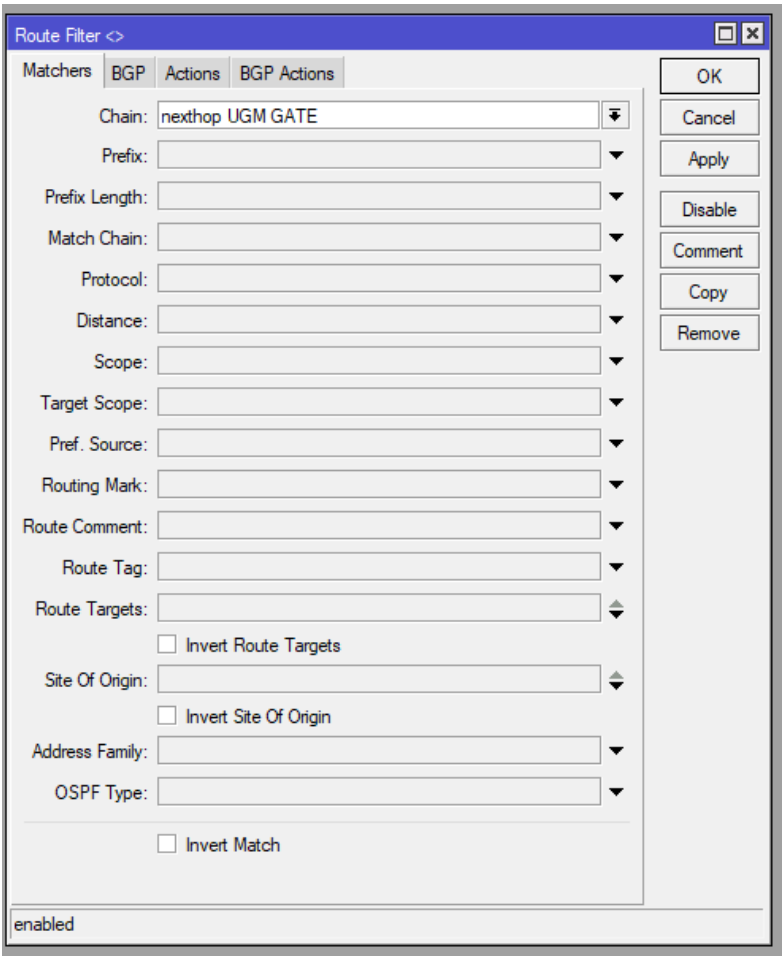
```
[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
 #    NAME                TYPE            MTU L2MTU  MAX-L2MTU
 0  R  ether1-gateway          ether          1500  1600       4076
 1  R  ether2-master-local      ether          1500  1598       2028
 2  R  ether3-slave-local       ether          1500  1598       2028
 3     ether4-slave-local       ether          1500  1598       2028

[admin@MikroTik] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   103.78.235.254/25  103.78.235.128  ether3-slave-local
 1   101.203.173.1/24   101.203.173.0   ether2-master-local
 2   103.78.233.1/25    103.78.233.0    ether1-gateway
[admin@MikroTik] > routing bgp instance print
Flags: * - default, X - disabled
 0 *X name="default" as=65530 router-id=0.0.0.0
     redistribute-connected=no redistribute-static=no
     redistribute-rip=no redistribute-ospf=no
     redistribute-other-bgp=no out-filter=""
     client-to-client-reflection=yes ignore-as-path-len=no

[admin@MikroTik] > routing bgp peer print
Flags: X - disabled, E - established
 #   INSTANCE    REMOTE-ADDRESS                  REMOTE-
AS
 0 E idren       103.78.235.251                    64302
 1 E idren       103.78.233.2                     64302
[admin@MikroTik] >
[admin@MikroTik] > routing bgp peer print
Flags: X - disabled, E - established
 #   INSTANCE    REMOTE-ADDRESS                  REMOTE-
AS
 0 E idren       103.78.235.251                    64302
 1 E idren       103.78.233.2                     64302
```

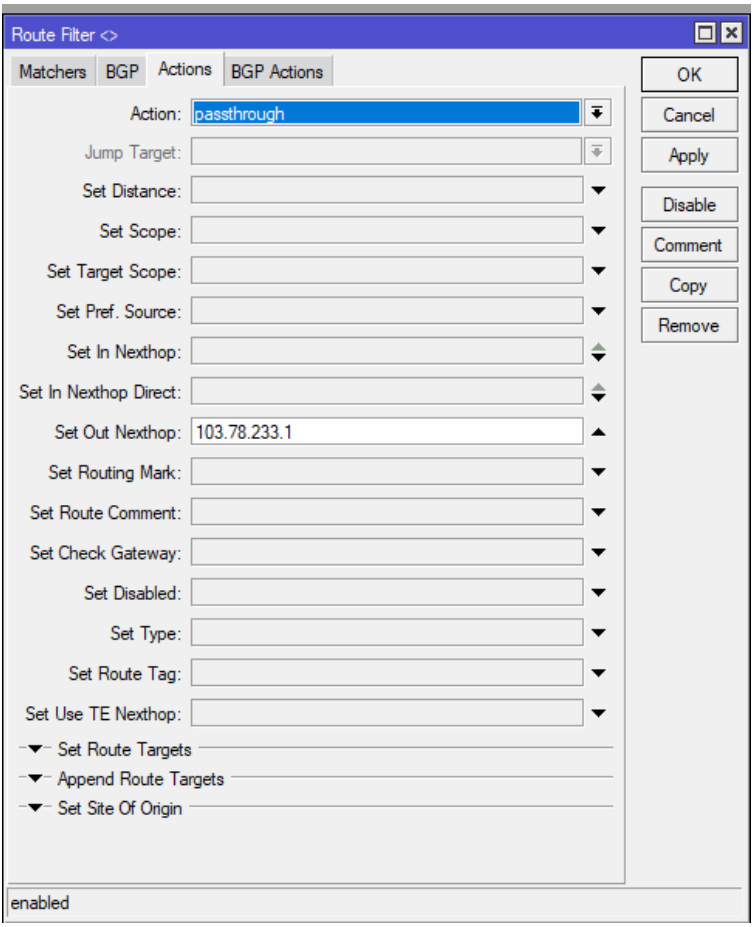
```
[admin@MikroTik] > routing bgp peer print status
Flags: X - disabled, E - established
 0 E name="peer to ITS gate" instance=idren
    remote-address=103.78.235.251 remote-as=64302 tcp-md5-key=""
    nexthop-choice=force-self multihop=no route-reflect=no
    hold-time=3m ttl=default in-filter="" out-filter=NEXTHOP-ITS
GATE
    address-families=ip default-originate=never remove-private-as=no
    as-override=no passive=no use-bfd=yes remote-id=103.78.235.251
    local-address=103.78.235.254 uptime=7h7m18s prefix-count=8
    updates-sent=14 updates-received=81469 withdrawn-sent=1
    withdrawn-received=6 remote-hold-time=3m used-hold-time=3m
    used-keepalive-time=1m refresh-capability=yes as4-capability=yes
    state=established

 1 E name="peer to UGM GATE" instance=idren remote-
address=103.78.233.2
    remote-as=64302 tcp-md5-key="" nexthop-choice=force-self
    multihop=no route-reflect=no hold-time=3m ttl=default
    in-filter="" out-filter=nexthop UGM GATE address-families=ip
    default-originate=never remove-private-as=no as-override=no
    passive=no use-bfd=yes remote-id=103.78.233.2
    local-address=103.78.233.1 uptime=7h6m55s prefix-count=4
    updates-sent=12 updates-received=83139 withdrawn-sent=1
    withdrawn-received=0 remote-hold-time=3m used-hold-time=3m
    used-keepalive-time=1m refresh-capability=yes as4-capability=yes
    state=established
```

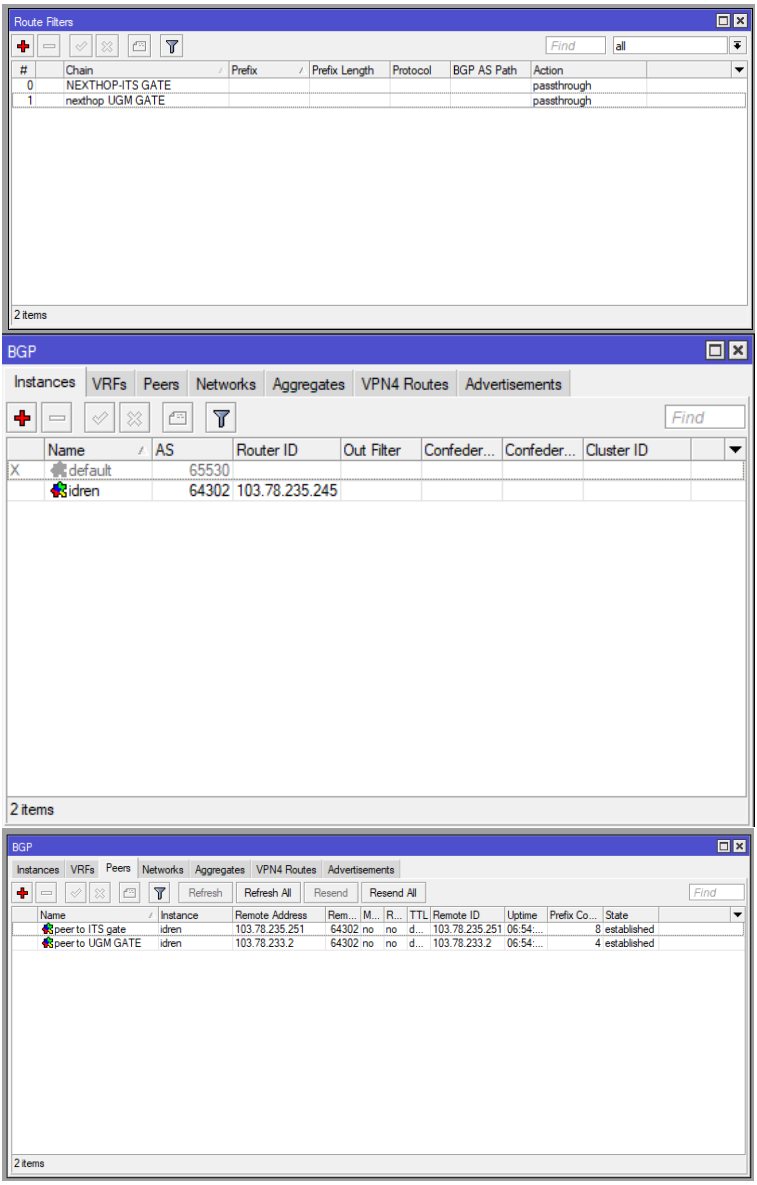

Address List
Find
Address
Network
Interface
101.203.173.1/24
101.203.173.0
ether2-master-local
103.78.233.1/25
103.78.233.0
ether1-gateway
103.78.235.254/25
103.78.235.128
ether3-slave-local
3 items
Route Filters
Find
all
#
Chain
Prefix
Prefix Length
Protocol
BGP AS Path
Action
0
NEXTHOP-ITS GATE
passthrough
1
nexthop UGM GATE
passthrough
2 items

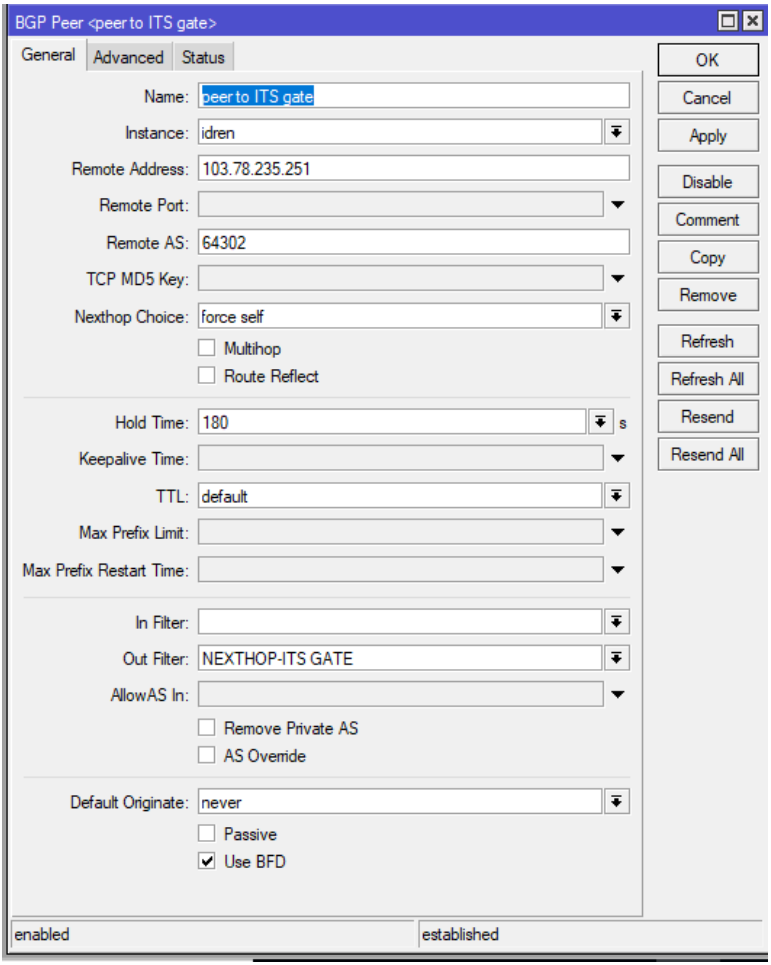

Route Filter <>
Matchers
BGP
Actions
BGP Actions
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Chain:
NEXTHOP-ITS GATE
Prefix:
Prefix Length:
Match Chain:
Protocol:
Distance:
Scope:
Target Scope:
Pref. Source:
Routing Mark:
Route Comment:
Route Tag:
Route Targets:
Invert Route Targets
Site Of Origin:
Invert Site Of Origin
Address Family:
OSPF Type:
Invert Match
enabled

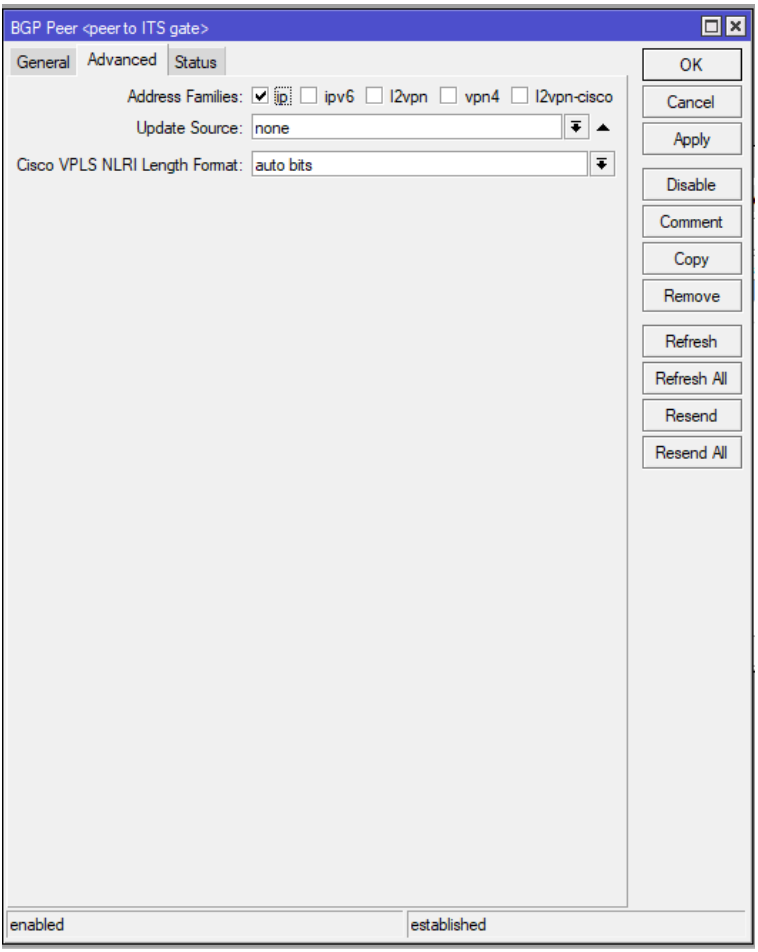

Route Filter <>
Matchers
BGP
Actions
BGP Actions
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Action:
passthrough
Jump Target:
Set Distance:
Set Scope:
Set Target Scope:
Set Pref. Source:
Set In Nexthop:
Set In Nexthop Direct:
Set Out Nexthop:
103.78.235.254
Set Routing Mark:
Set Route Comment:
Set Check Gateway:
Set Disabled:
Set Type:
Set Route Tag:
Set Use TE Nexthop:
Set Route Targets
Append Route Targets
Set Site Of Origin
enabled

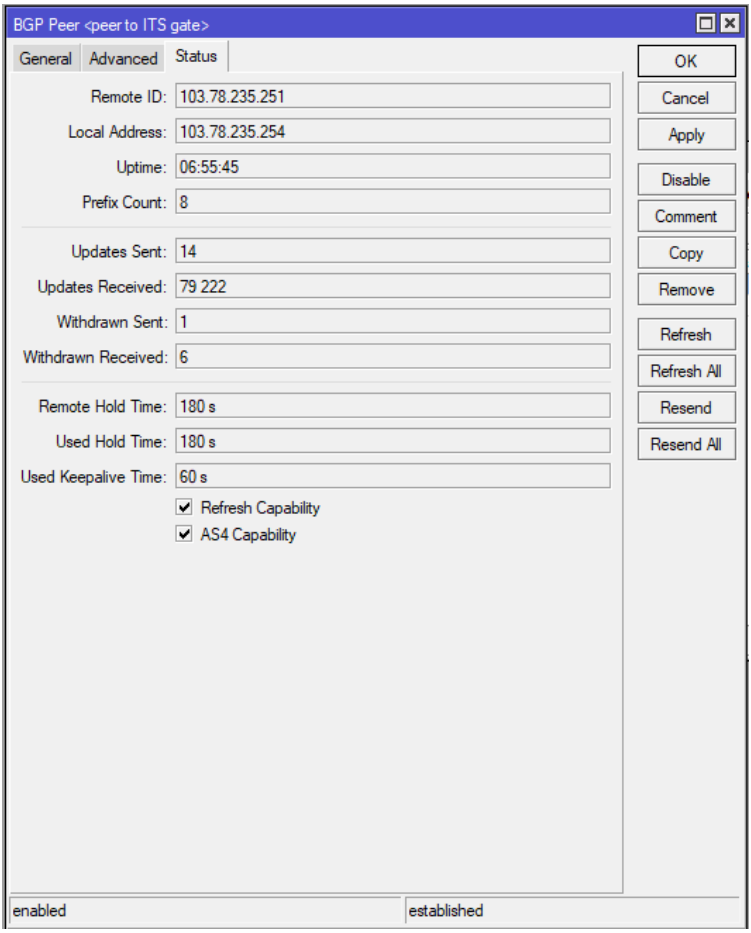

Route Filter <>
Matchers
BGP
Actions
BGP Actions
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Chain:
nexthop UGM GATE
Prefix:
Prefix Length:
Match Chain:
Protocol:
Distance:
Scope:
Target Scope:
Pref. Source:
Routing Mark:
Route Comment:
Route Tag:
Route Targets:
Invert Route Targets
Site Of Origin:
Invert Site Of Origin
Address Family:
OSPF Type:
Invert Match
enabled

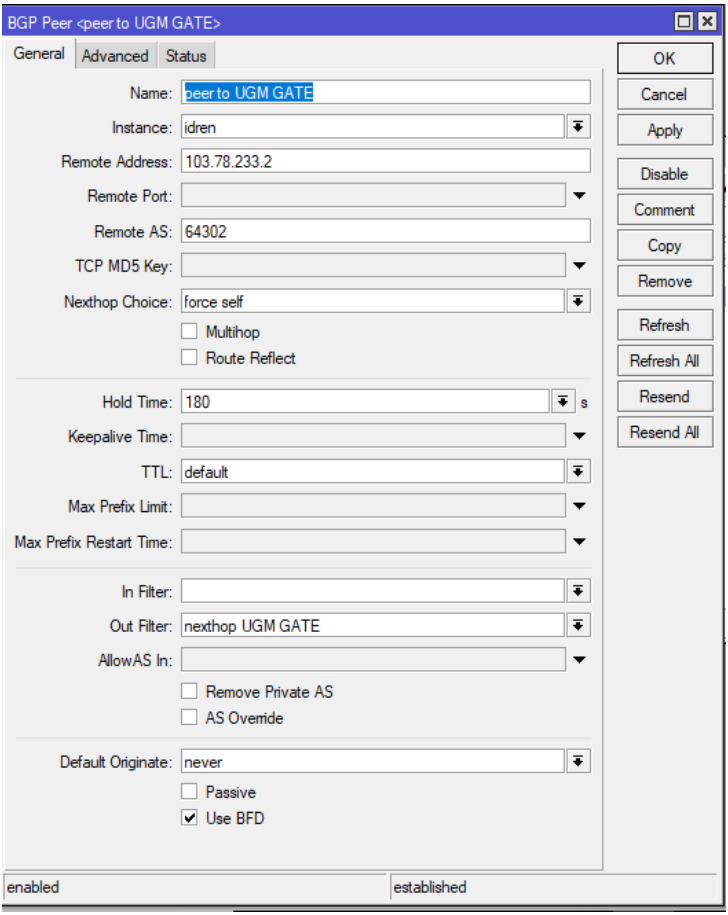

Route Filter <>
Matchers
BGP
Actions
BGP Actions
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Action: passthrough
Jump Target:
Set Distance:
Set Scope:
Set Target Scope:
Set Pref. Source:
Set In Nexthop:
Set In Nexthop Direct:
Set Out Nexthop: 103.78.233.1
Set Routing Mark:
Set Route Comment:
Set Check Gateway:
Set Disabled:
Set Type:
Set Route Tag:
Set Use TE Nexthop:
Set Route Targets
Append Route Targets
Set Site Of Origin
enabled

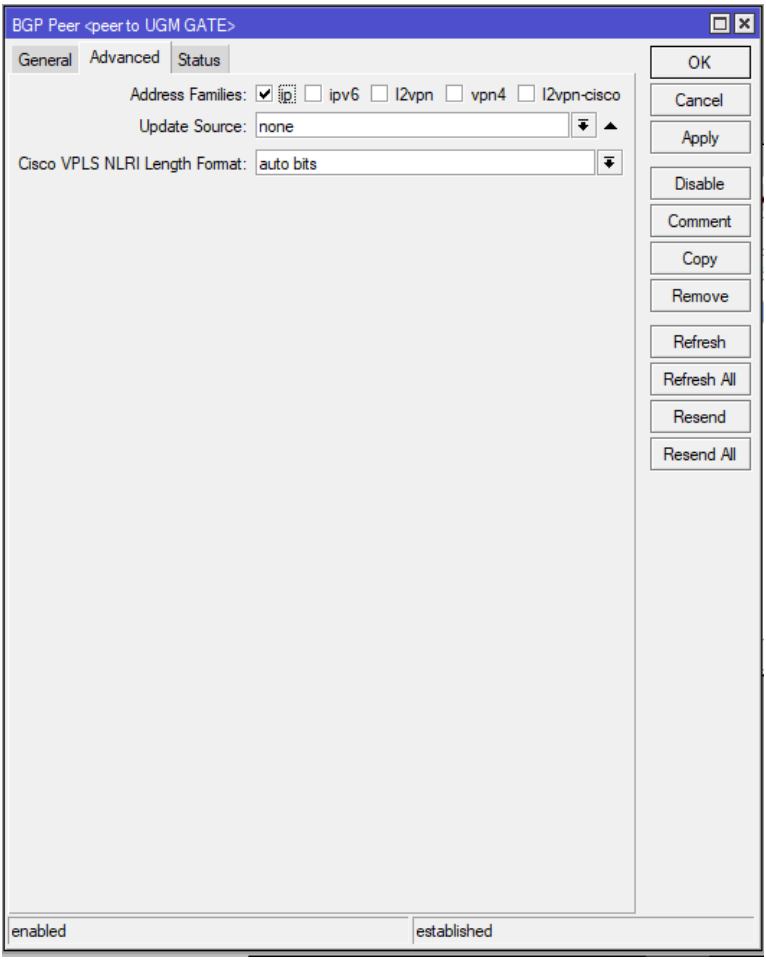

Route Filters
Find
all
Chain Prefix Prefix Length Protocol BGP AS Path Action
0 NEXTHOP-ITS GATE passthrough
1 nexthop UGM GATE passthrough
2 items
BGP
Instances VRFs Peers Networks Aggregates VPN4 Routes Advertisements
Find
Name AS Router ID Out Filter Confeder... Confeder... Cluster ID
X default 65530
idren 64302 103.78.235.245
2 items
BGP
Instances VRFs Peers Networks Aggregates VPN4 Routes Advertisements
Refresh Refresh All Resend Resend All
Find
Name Instance Remote Address Rem... M... R... TTL Remote ID Uptime Prefix Co... State
peer to ITS gate idren 103.78.235.251 64302 no no d... 103.78.235.251 06:54:... 8 established
peer to UGM GATE idren 103.78.233.2 64302 no no d... 103.78.233.2 06:54:... 4 established
2 items

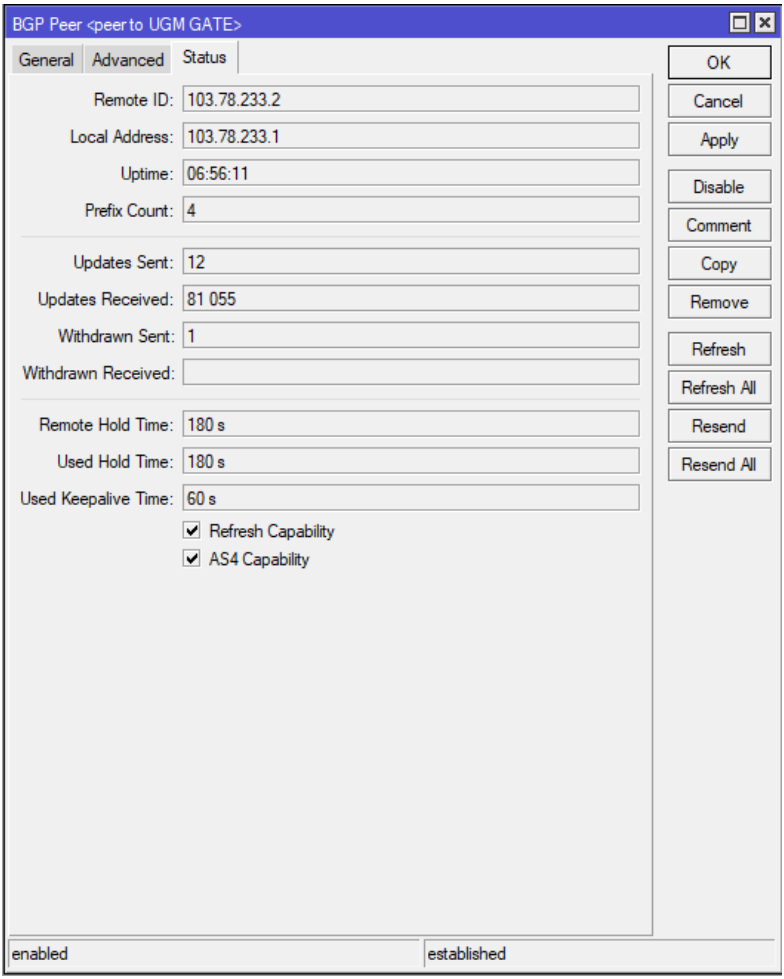

BGP Peer <peer to ITS gate>
General
Advanced
Status
Name: peer to ITS gate
Instance: idren
Remote Address: 103.78.235.251
Remote Port:
Remote AS: 64302
TCP MD5 Key:
Nexthop Choice: force self
Multihop
Route Reflect
Hold Time: 180 s
Keepalive Time:
TTL: default
Max Prefix Limit:
Max Prefix Restart Time:
In Filter:
Out Filter: NEXTHOP-ITS GATE
AllowAS In:
Remove Private AS
AS Override
Default Originate: never
Passive
Use BFD
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Refresh
Refresh All
Resend
Resend All
enabled
established

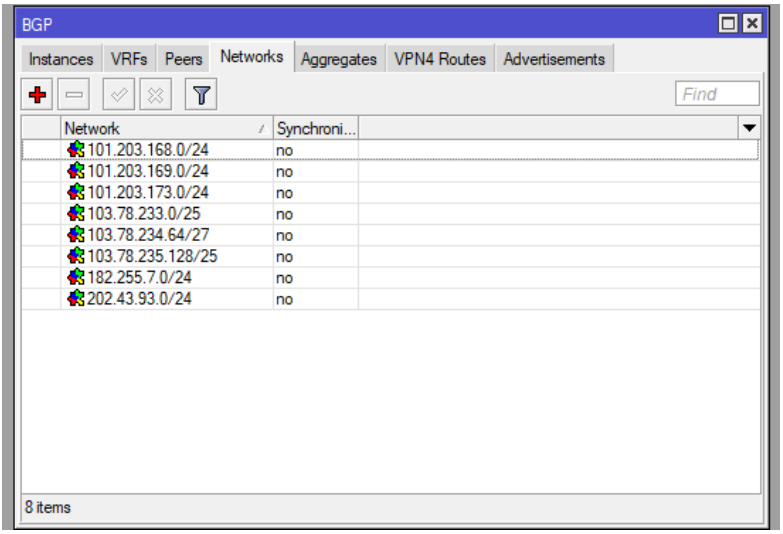

BGP Peer <peer to ITS gate>
General
Advanced
Status
Address Families:
ip
ipv6
l2vpn
vpn4
l2vpn-cisco
Update Source:
none
Cisco VPLS NLRI Length Format:
auto bits
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Refresh
Refresh All
Resend
Resend All
enabled
established

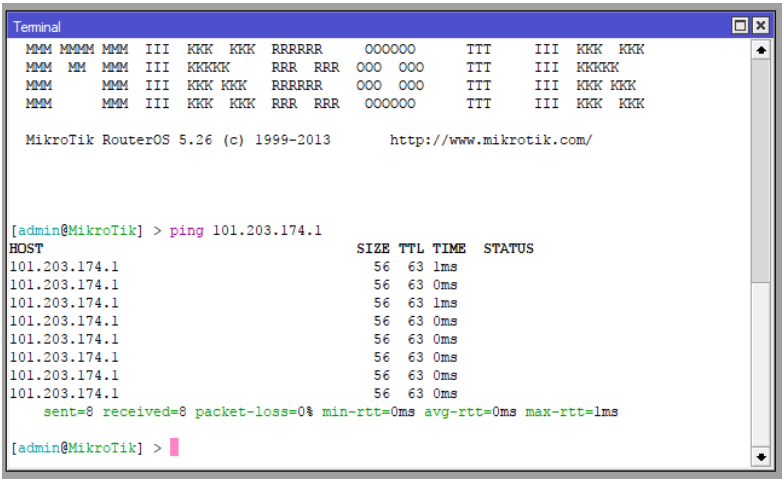

BGP Peer <peer to ITS gate>
General
Advanced
Status
Remote ID: 103.78.235.251
Local Address: 103.78.235.254
Uptime: 06:55:45
Prefix Count: 8
Updates Sent: 14
Updates Received: 79 222
Withdrawn Sent: 1
Withdrawn Received: 6
Remote Hold Time: 180 s
Used Hold Time: 180 s
Used Keepalive Time: 60 s
Refresh Capability
AS4 Capability
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Refresh
Refresh All
Resend
Resend All
enabled
established

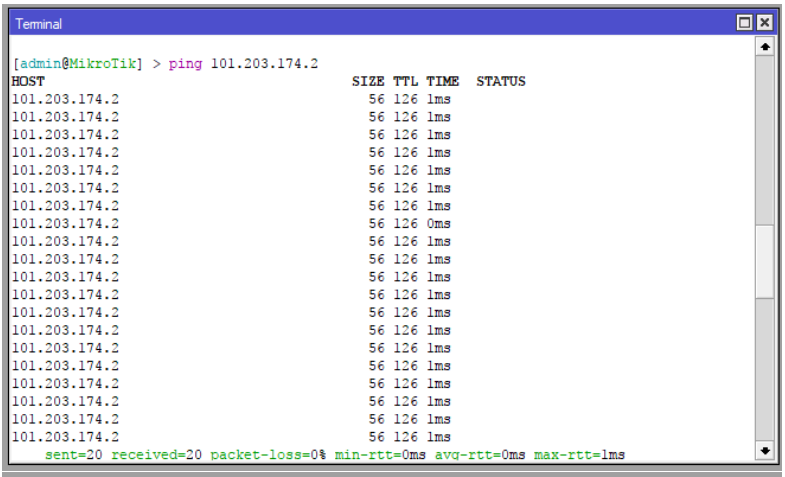

BGP Peer <peer to UGM GATE>
General
Advanced
Status
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Refresh
Refresh All
Resend
Resend All
Name: peer to UGM GATE
Instance: idren
Remote Address: 103.78.233.2
Remote Port:
Remote AS: 64302
TCP MD5 Key:
Nexthop Choice: force self
Multihop
Route Reflect
Hold Time: 180 s
Keepalive Time:
TTL: default
Max Prefix Limit:
Max Prefix Restart Time:
In Filter:
Out Filter: nexthop UGM GATE
AllowAS In:
Remove Private AS
AS Override
Default Originate: never
Passive
Use BFD
enabled
established

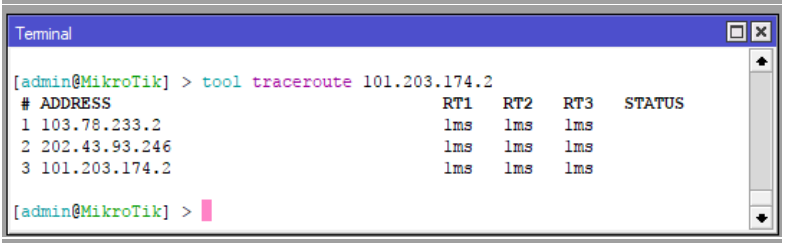

BGP Peer <peer to UGM GATE>
General
Advanced
Status
Address Families:
ip
ipv6
l2vpn
vpn4
l2vpn-cisco
Update Source:
none
Cisco VPLS NLRI Length Format:
auto bits
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Refresh
Refresh All
Resend
Resend All
enabled
established

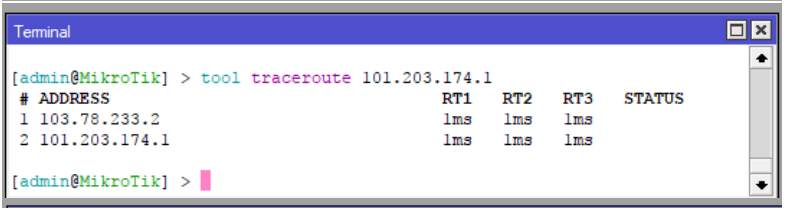

BGP Peer <peer to UGM GATE>
General
Advanced
Status
Remote ID: 103.78.233.2
Local Address: 103.78.233.1
Uptime: 06:56:11
Prefix Count: 4
Updates Sent: 12
Updates Received: 81 055
Withdrawn Sent: 1
Withdrawn Received:
Remote Hold Time: 180 s
Used Hold Time: 180 s
Used Keepalive Time: 60 s
Refresh Capability
AS4 Capability
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Refresh
Refresh All
Resend
Resend All
enabled
established

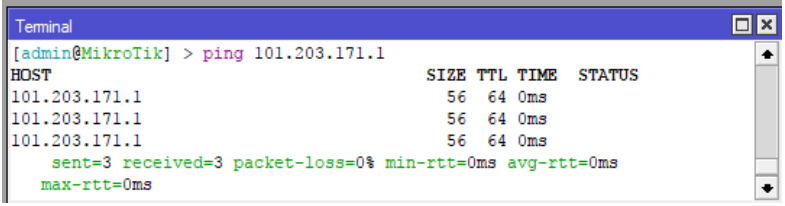

BGP
Instances
VRFs
Peers
Networks
Aggregates
VPN4 Routes
Advertisements
Find
Network
Synchroni...
101.203.168.0/24 no
101.203.169.0/24 no
101.203.173.0/24 no
103.78.233.0/25 no
103.78.234.64/27 no
103.78.235.128/25 no
182.255.7.0/24 no
202.43.93.0/24 no
8 items

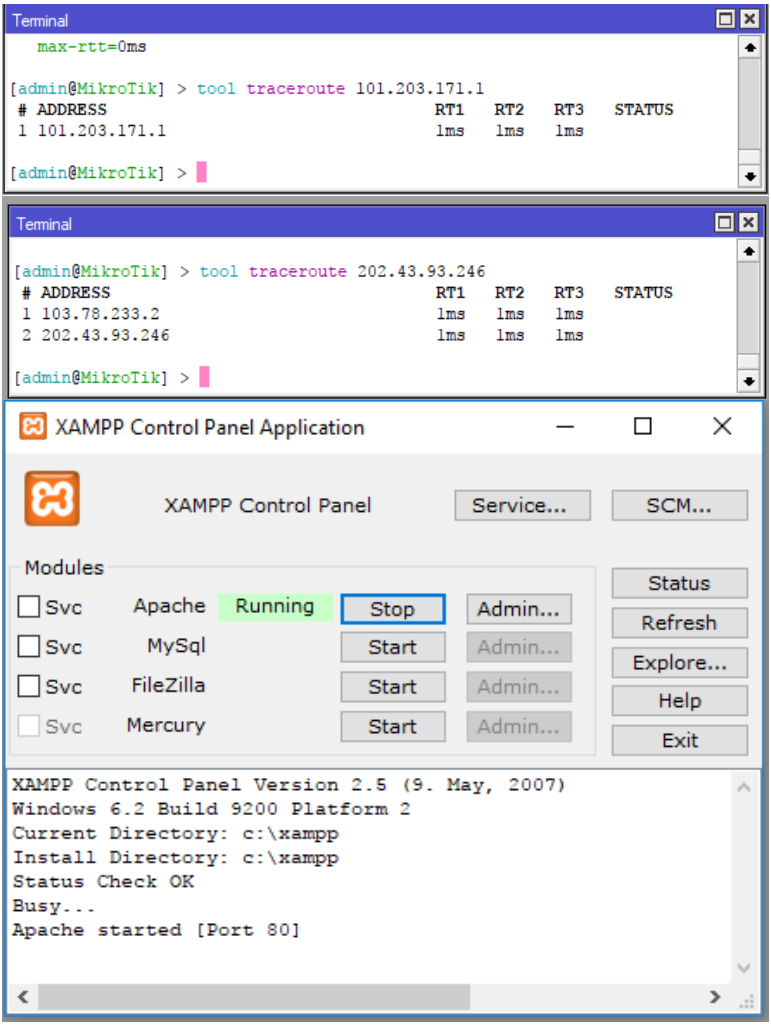

Terminal
MMM MMMM MMM III KKK KKK RRRRRR OOOOOO TTT III KKK KKK
MMM MM MMM III KKKKK RRR RRR OOO OOO TTT III KKKKK
MMM MMM III KKK KKK RRRRRR OOO OOO TTT III KKK KKK
MMM MMM III KKK KKK RRR RRR OOOOOO TTT III KKK KKK
MikroTik RouterOS 5.26 (c) 1999-2013 http://www.mikrotik.com/
[admin@MikroTik] > ping 101.203.174.1
HOST SIZE TTL TIME STATUS
101.203.174.1 56 63 1ms
101.203.174.1 56 63 0ms
101.203.174.1 56 63 1ms
101.203.174.1 56 63 0ms
101.203.174.1 56 63 0ms
101.203.174.1 56 63 0ms
101.203.174.1 56 63 0ms
101.203.174.1 56 63 0ms
sent=8 received=8 packet-loss=0% min-rtt=0ms avg-rtt=0ms max-rtt=1ms
[admin@MikroTik] >

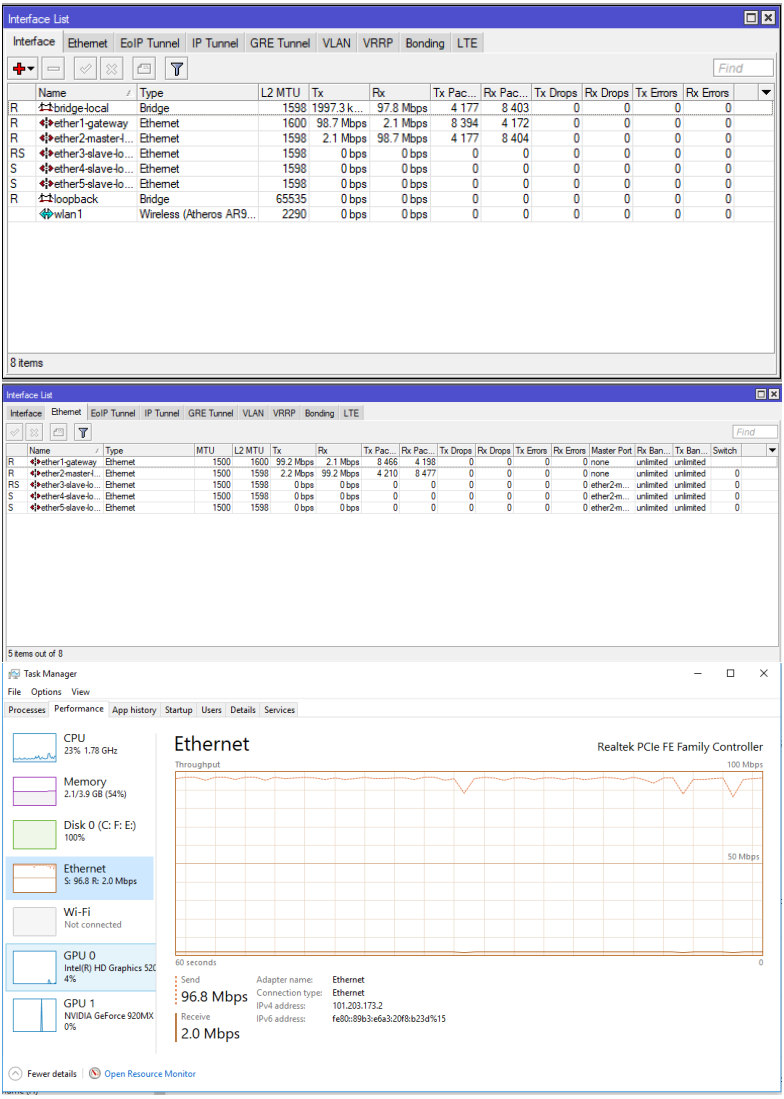

Terminal
[admin@MikroTik] > ping 101.203.174.2
HOST SIZE TTL TIME STATUS
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 0ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
101.203.174.2 56 126 1ms
sent=20 received=20 packet-loss=0% min-rtt=0ms avg-rtt=0ms max-rtt=1ms


Terminal
[admin@MikroTik] > tool traceroute 101.203.174.2
ADDRESS RT1 RT2 RT3 STATUS
1 103.78.233.2 1ms 1ms 1ms
2 202.43.93.246 1ms 1ms 1ms
3 101.203.174.2 1ms 1ms 1ms
[admin@MikroTik] >


Terminal
[admin@MikroTik] > tool traceroute 101.203.174.1
ADDRESS RT1 RT2 RT3 STATUS
1 103.78.233.2 1ms 1ms 1ms
2 101.203.174.1 1ms 1ms 1ms
[admin@MikroTik] >


Terminal
[admin@MikroTik] > ping 101.203.171.1
HOST SIZE TTL TIME STATUS
101.203.171.1 56 64 0ms
101.203.171.1 56 64 0ms
101.203.171.1 56 64 0ms
sent=3 received=3 packet-loss=0% min-rtt=0ms avg-rtt=0ms
max-rtt=0ms


Terminal
max-rtt=0ms
[admin@MikroTik] > tool traceroute 101.203.171.1
ADDRESS RT1 RT2 RT3 STATUS
1 101.203.171.1 1ms 1ms 1ms
[admin@MikroTik] >
Terminal
[admin@MikroTik] > tool traceroute 202.43.93.246
ADDRESS RT1 RT2 RT3 STATUS
1 103.78.233.2 1ms 1ms 1ms
2 202.43.93.246 1ms 1ms 1ms
[admin@MikroTik] >
XAMPP Control Panel Application
XAMPP Control Panel
Service...
SCM...
Modules
Svc Apache Running Stop Admin...
Svc MySql Start Admin...
Svc FileZilla Start Admin...
Svc Mercury Start Admin...
Status
Refresh
Explore...
Help
Exit
XAMPP Control Panel Version 2.5 (9. May, 2007)
Windows 6.2 Build 9200 Platform 2
Current Directory: c:\xampp
Install Directory: c:\xampp
Status Check OK
Busy...
Apache started [Port 80]


Interface List
Interface
Ethernet
EoIP Tunnel
IP Tunnel
GRE Tunnel
VLAN
VRRP
Bonding
LTE
Find
Name
Type
L2 MTU
Tx
Rx
Tx Pac...
Rx Pac...
Tx Drops
Rx Drops
Tx Errors
Rx Errors
R bridge-local Bridge 1598 1997.3 k... 97.8 Mbps 4 177 8 403 0 0 0 0
R ether1-gateway Ethernet 1600 98.7 Mbps 2.1 Mbps 8 394 4 172 0 0 0 0
R ether2-master-l... Ethernet 1598 2.1 Mbps 98.7 Mbps 4 177 8 404 0 0 0 0
RS ether3-slave-lo... Ethernet 1598 0 bps 0 bps 0 0 0 0 0 0
S ether4-slave-lo... Ethernet 1598 0 bps 0 bps 0 0 0 0 0 0
S ether5-slave-lo... Ethernet 1598 0 bps 0 bps 0 0 0 0 0 0
R loopback Bridge 65535 0 bps 0 bps 0 0 0 0 0 0
wlan1 Wireless (Atheros AR9... 2290 0 bps 0 bps 0 0 0 0 0 0
8 items
Interface List
Interface
Ethernet
EoIP Tunnel
IP Tunnel
GRE Tunnel
VLAN
VRRP
Bonding
LTE
Find
Name
Type
MTU
L2 MTU
Tx
Rx
Tx Pac...
Rx Pac...
Tx Drops
Rx Drops
Tx Errors
Rx Errors
Master Port
Rx Ban...
Tx Ban...
Switch
R ether1-gateway Ethernet 1500 1600 99.2 Mbps 2.1 Mbps 8 466 4 198 0 0 0 0 none unlimited unlimited
R ether2-master-l... Ethernet 1500 1598 2.2 Mbps 99.2 Mbps 4 210 8 477 0 0 0 0 none unlimited unlimited 0
RS ether3-slave-lo... Ethernet 1500 1598 0 bps 0 bps 0 0 0 0 0 0 ether2-m... unlimited unlimited 0
S ether4-slave-lo... Ethernet 1500 1598 0 bps 0 bps 0 0 0 0 0 0 ether2-m... unlimited unlimited 0
S ether5-slave-lo... Ethernet 1500 1598 0 bps 0 bps 0 0 0 0 0 0 ether2-m... unlimited unlimited 0
5 items out of 8
Task Manager
File
Options
View
Processes
Performance
App history
Startup
Users
Details
Services
CPU
23% 1.78 GHz
Memory
2.1/3.9 GB (54%)
Disk 0 (C: F: E:)
100%
Ethernet
S: 96.8 R: 2.0 Mbps
Wi-Fi
Not connected
GPU 0
Intel(R) HD Graphics 520
4%
GPU 1
NVIDIA GeForce 920MX
0%
Ethernet
Realtek PCIe FE Family Controller
Throughput
100 Mbps
50 Mbps
60 seconds
0
Send
96.8 Mbps
Receive
2.0 Mbps
Adapter name: Ethernet
Connection type: Ethernet
IPv4 address: 101.203.173.2
IPv6 address: fe80::89b3:e6a3:20f8:b23d%15
Fewer details
Open Resource Monitor

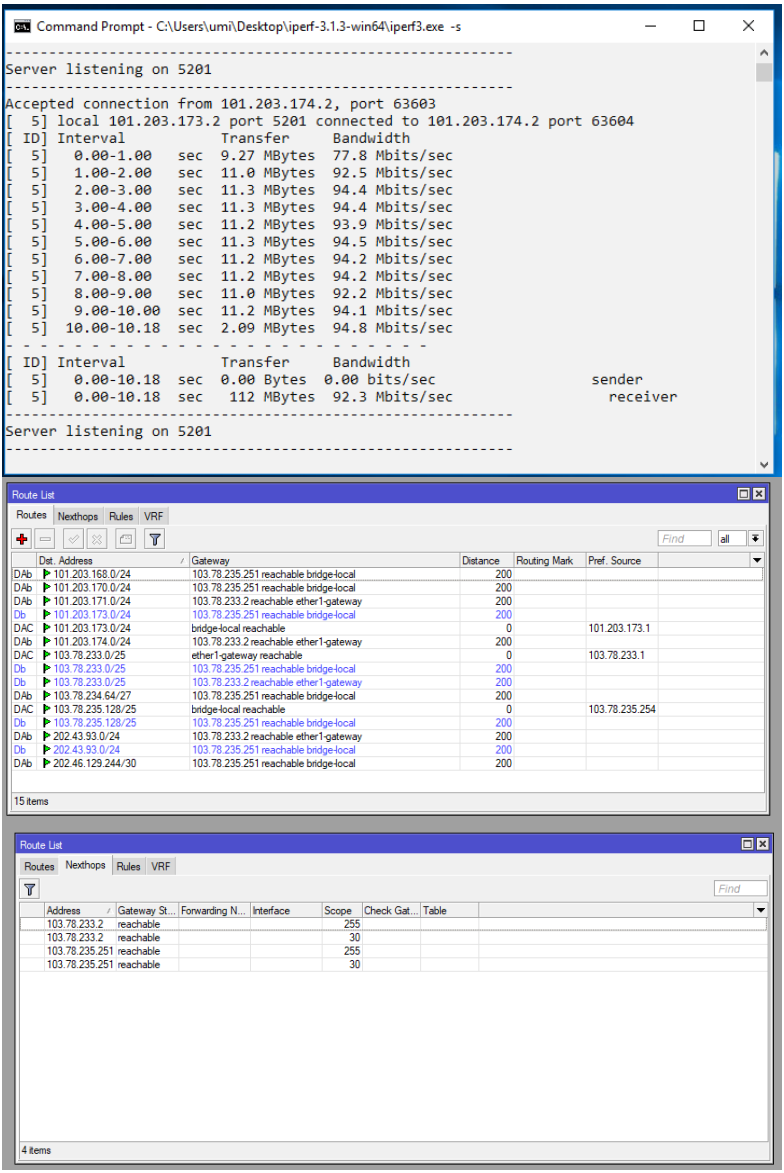
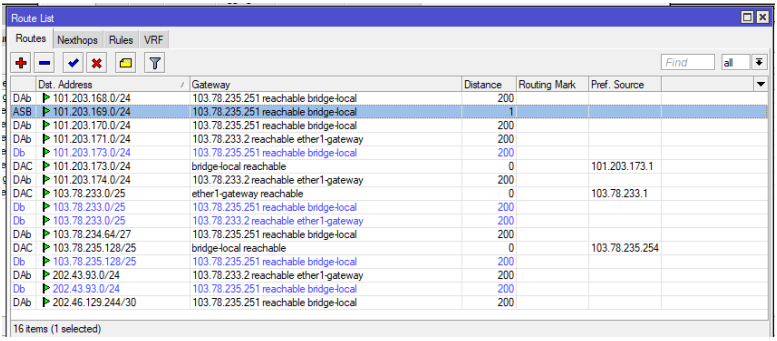

Command Prompt - C:\Users\umi\Desktop\iperf-3.1.3-win64\iperf3.exe -s
Server listening on 5201
Accepted connection from 101.203.174.2, port 63603
[5] local 101.203.173.2 port 5201 connected to 101.203.174.2 port 63604
[ID] Interval Transfer Bandwidth
[5] 0.00-1.00 sec 9.27 MBytes 77.8 Mbits/sec
[5] 1.00-2.00 sec 11.0 MBytes 92.5 Mbits/sec
[5] 2.00-3.00 sec 11.3 MBytes 94.4 Mbits/sec
[5] 3.00-4.00 sec 11.3 MBytes 94.4 Mbits/sec
[5] 4.00-5.00 sec 11.2 MBytes 93.9 Mbits/sec
[5] 5.00-6.00 sec 11.3 MBytes 94.5 Mbits/sec
[5] 6.00-7.00 sec 11.2 MBytes 94.2 Mbits/sec
[5] 7.00-8.00 sec 11.2 MBytes 94.2 Mbits/sec
[5] 8.00-9.00 sec 11.0 MBytes 92.2 Mbits/sec
[5] 9.00-10.00 sec 11.2 MBytes 94.1 Mbits/sec
[5] 10.00-10.18 sec 2.09 MBytes 94.8 Mbits/sec
[ID] Interval Transfer Bandwidth
[5] 0.00-10.18 sec 0.00 Bytes 0.00 bits/sec sender
[5] 0.00-10.18 sec 112 MBytes 92.3 Mbits/sec receiver
Server listening on 5201
Route List
Routes Nexthops Rules VRF
Find all
Dst. Address Gateway Distance Routing Mark Pref. Source
DAb 101.203.168.0/24 103.78.235.251 reachable bridge-local 200
DAb 101.203.170.0/24 103.78.235.251 reachable bridge-local 200
DAb 101.203.171.0/24 103.78.233.2 reachable ether1-gateway 200
Db 101.203.173.0/24 103.78.235.251 reachable bridge-local 200
DAC 101.203.173.0/24 bridge-local reachable 0 101.203.173.1
DAb 101.203.174.0/24 103.78.233.2 reachable ether1-gateway 200
DAC 103.78.233.0/25 ether1-gateway reachable 0 103.78.233.1
Db 103.78.233.0/25 103.78.235.251 reachable bridge-local 200
Db 103.78.233.0/25 103.78.233.2 reachable ether1-gateway 200
DAb 103.78.234.64/27 103.78.235.251 reachable bridge-local 200
DAC 103.78.235.128/25 bridge-local reachable 0 103.78.235.254
Db 103.78.235.128/25 103.78.235.251 reachable bridge-local 200
DAb 202.43.93.0/24 103.78.233.2 reachable ether1-gateway 200
Db 202.43.93.0/24 103.78.235.251 reachable bridge-local 200
DAb 202.46.129.244/30 103.78.235.251 reachable bridge-local 200
15 items
Route List
Routes Nexthops Rules VRF
Find
Address Gateway St... Forwarding N... Interface Scope Check Gat... Table
103.78.233.2 reachable 255
103.78.233.2 reachable 30
103.78.235.251 reachable 255
103.78.235.251 reachable 30
4 items

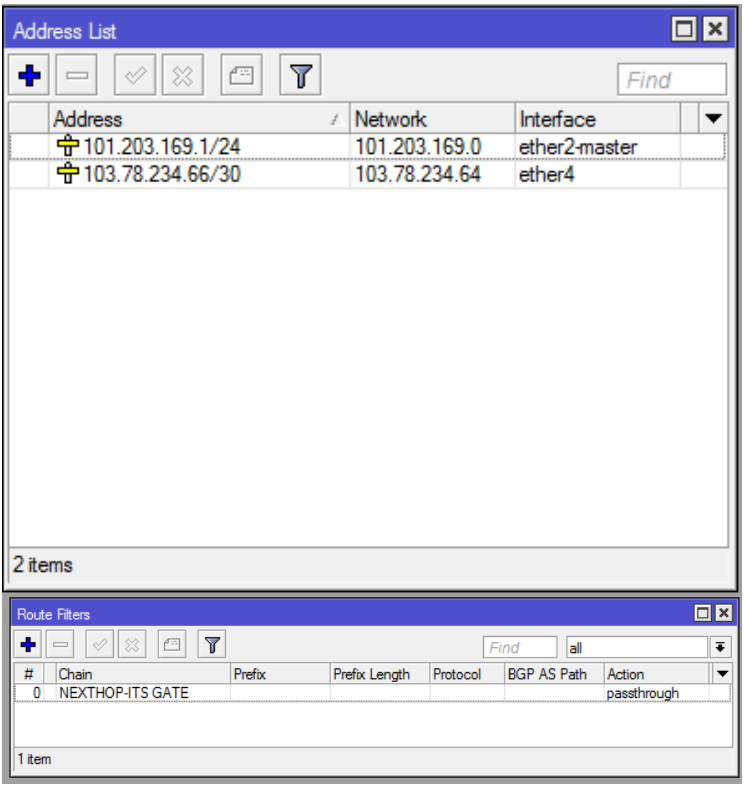
***Table route* IdREN to IAIN PONOROGO**

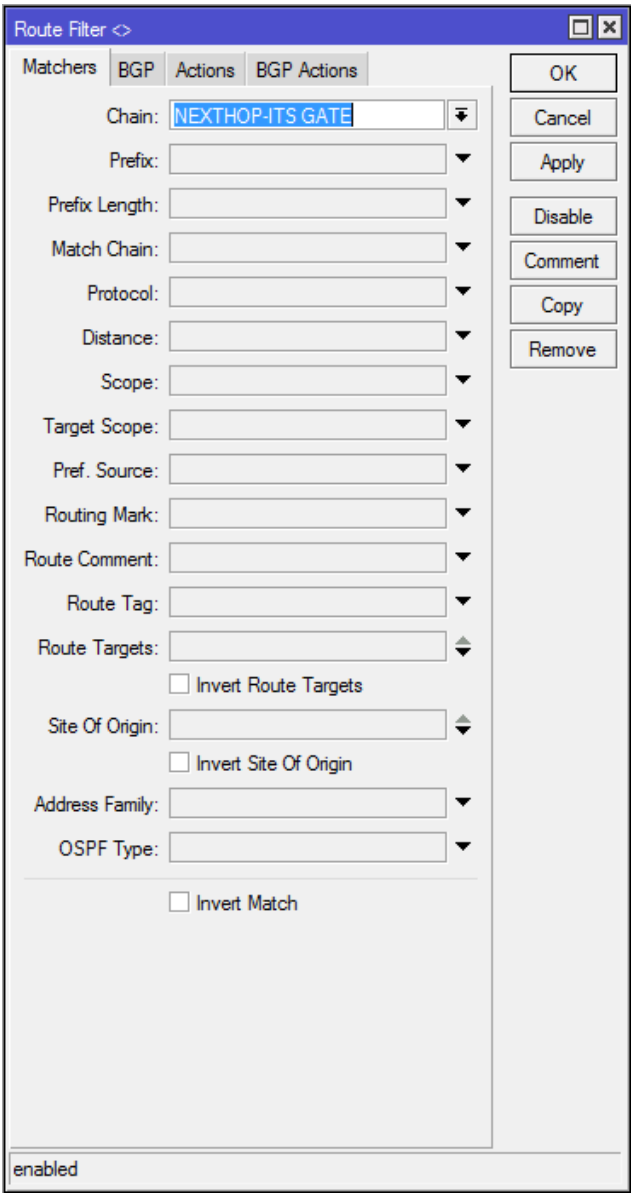


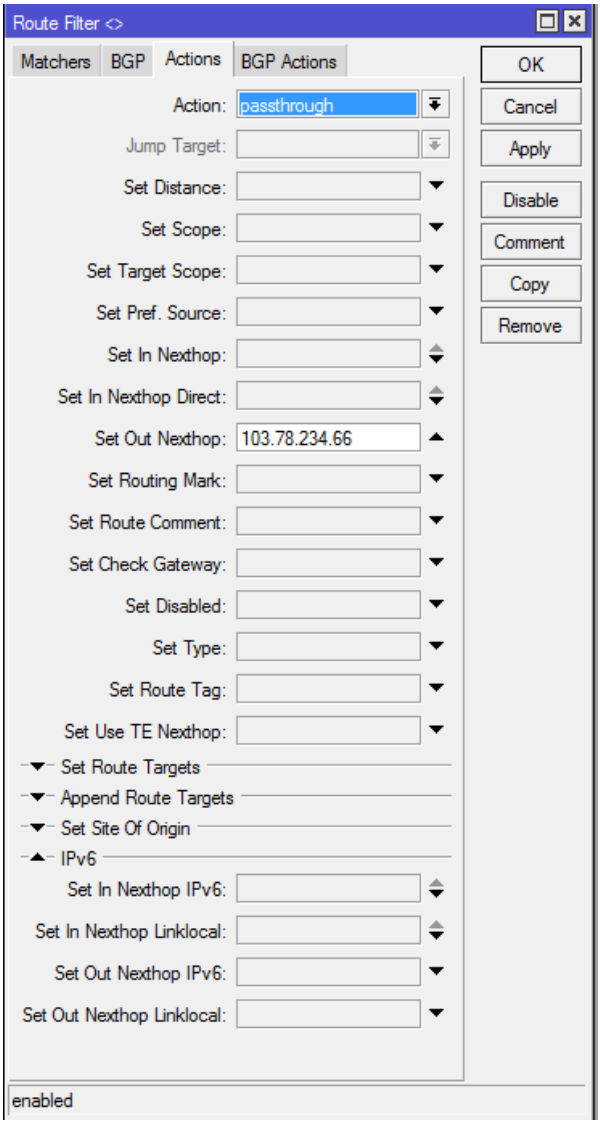
Konfigurasi IAIN Ponorogo

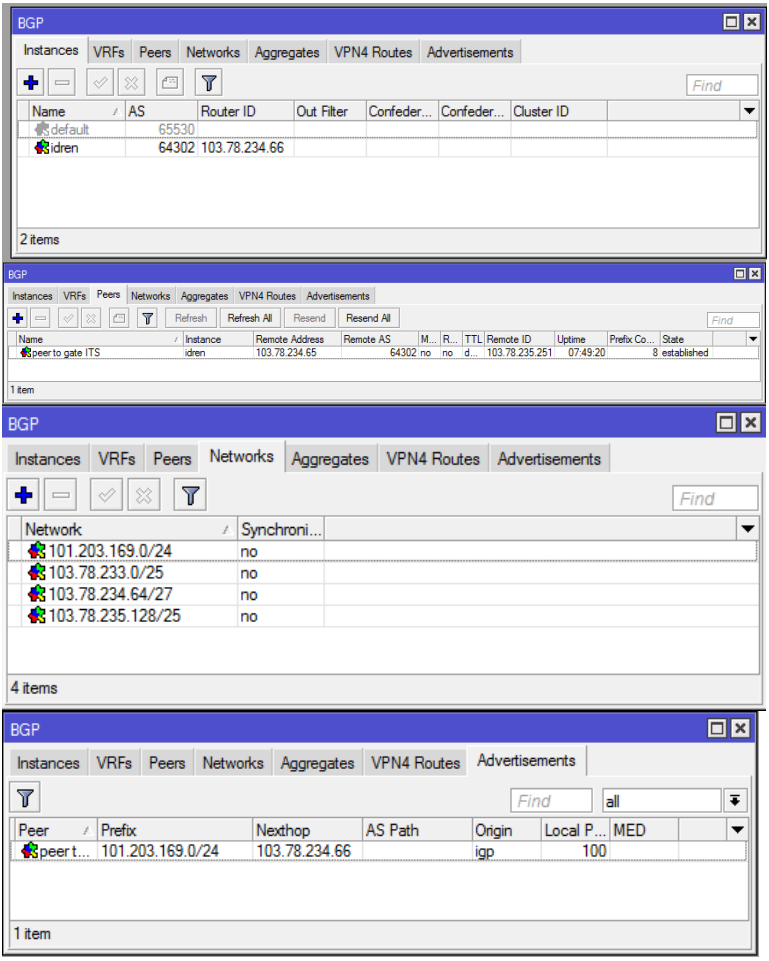
```
[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
 #    NAME                TYPE       ACTUAL-MTU L2MTU
MAX-L2MTU
 0    ether1              ether          1500  1600      4076
 1  RS ether2-master          ether          1500  1598      2028
 2   S ether3             ether          1500  1598      2028
 3  RS ether4              ether          1500  1598      2028
 4   S ether5             ether          1500  1598      2028
 5   S wlan1               wlan           1500  1600      2290
 6  R  ;;; defconf
      bridge              bridge         1500  1598
[admin@MikroTik] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK        INTERFACE
 0   101.203.169.1/24   101.203.169.0  ether2-master
 1   103.78.234.66/30   103.78.234.64  ether4
[admin@MikroTik] > routing bgp peer print
Flags: X - disabled, E - established
 #   INSTANCE        REMOTE-ADDRESS
REMOTE-AS
 0 E idren           103.78.234.65                 64302
```

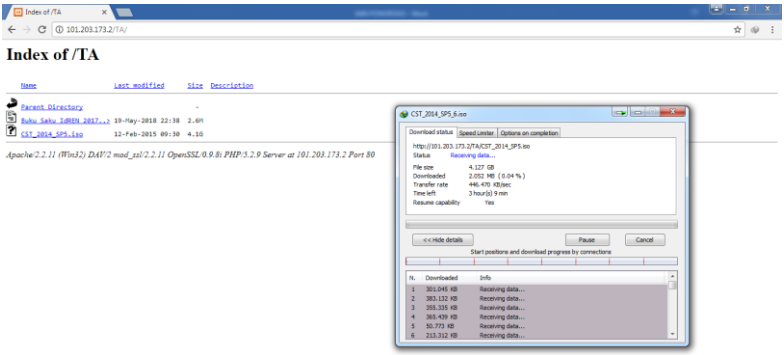
```
[admin@MikroTik] > routing bgp instance print
Flags: * - default, X - disabled
 0 *X name="default" as=65530 router-id=0.0.0.0 redistribute-
connected=no
      redistribute-static=no redistribute-rip=no redistribute-ospf=no
      redistribute-other-bgp=no out-filter="" client-to-client-
reflection=yes
      ignore-as-path-len=no routing-table=""

 1    name="idren" as=64302 router-id=103.78.234.66 redistribute-
connected=no
      redistribute-static=no redistribute-rip=no redistribute-ospf=no
      redistribute-other-bgp=no out-filter="" client-to-client-reflection=no
      ignore-as-path-len=no routing-table=""
[admin@MikroTik] > routing filter print
Flags: X - disabled
 0   chain=NEXTHOP-ITS GATE invert-match=no action=passthrough
     set-out-nexthop=103.78.234.66 set-bgp-prepend-path=""
[admin@MikroTik] > routing bgp peer print status
Flags: X - disabled, E - established
 0 E name="peer to gate ITS" instance=idren remote-
address=103.78.234.65
     remote-as=64302 tcp-md5-key="" nexthop-choice=force-self
multihop=no
     route-reflect=no hold-time=3m ttl=default in-filter=""
     out-filter=NEXTHOP-ITS GATE address-families=ip default-
originate=never
     remove-private-as=no as-override=no passive=no use-bfd=yes
     remote-id=103.78.235.251 local-address=103.78.234.66
uptime=7h46m15s
     prefix-count=8 updates-sent=7 updates-received=89145 withdrawn-
sent=1
     withdrawn-received=6 remote-hold-time=3m used-hold-time=3m
     used-keepalive-time=1m refresh-capability=yes as4-capability=yes
     state=established
```


Address List
Find
Address
Network
Interface
101.203.169.1/24
101.203.169.0
ether2-master
103.78.234.66/30
103.78.234.64
ether4
2 items
Route Filters
Find
all
#
Chain
Prefix
Prefix Length
Protocol
BGP AS Path
Action
0
NEXTHOP-ITS GATE
passthrough
1 item

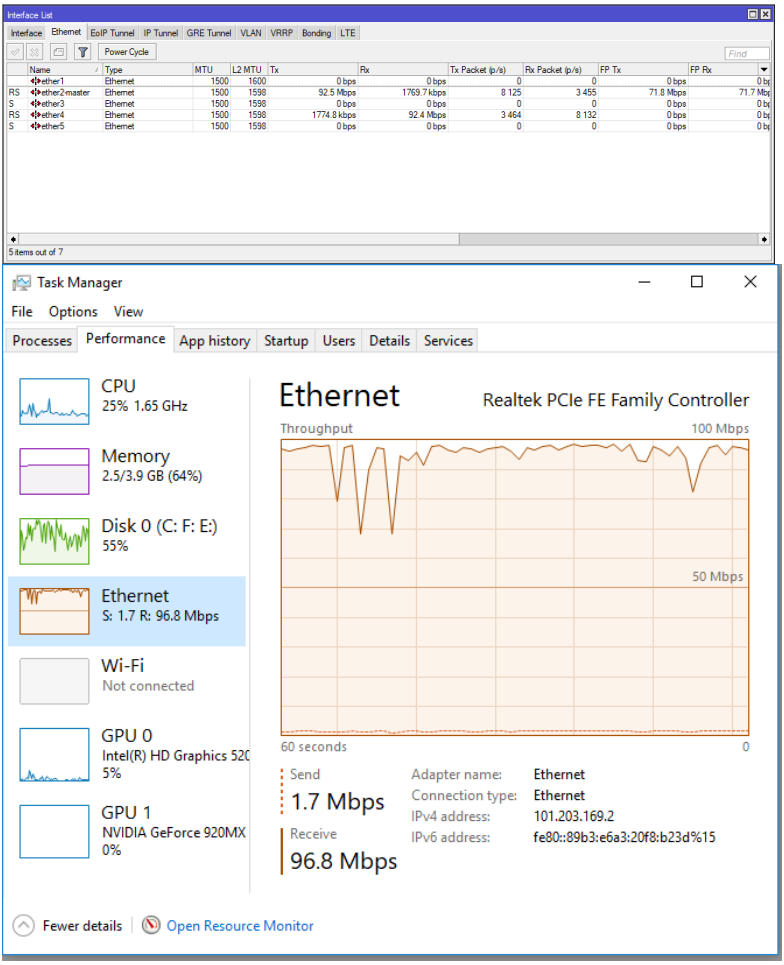

Route Filter <>
Matchers
BGP
Actions
BGP Actions
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Chain:
NEXTHOP-ITS GATE
Prefix:
Prefix Length:
Match Chain:
Protocol:
Distance:
Scope:
Target Scope:
Pref. Source:
Routing Mark:
Route Comment:
Route Tag:
Route Targets:
Invert Route Targets
Site Of Origin:
Invert Site Of Origin
Address Family:
OSPF Type:
Invert Match
enabled


Route Filter <>
Matchers
BGP
Actions
BGP Actions
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Action: passthrough
Jump Target:
Set Distance:
Set Scope:
Set Target Scope:
Set Pref. Source:
Set In Nexthop:
Set In Nexthop Direct:
Set Out Nexthop: 103.78.234.66
Set Routing Mark:
Set Route Comment:
Set Check Gateway:
Set Disabled:
Set Type:
Set Route Tag:
Set Use TE Nexthop:
Set Route Targets
Append Route Targets
Set Site Of Origin
IPv6
Set In Nexthop IPv6:
Set In Nexthop Linklocal:
Set Out Nexthop IPv6:
Set Out Nexthop Linklocal:
enabled


BGP
Instances
VRFs
Peers
Networks
Aggregates
VPN4 Routes
Advertisements
Find
Name
AS
Router ID
Out Filter
Confeder...
Confeder...
Cluster ID
default
65530
idren
64302
103.78.234.66
2 items
BGP
Instances
VRFs
Peers
Networks
Aggregates
VPN4 Routes
Advertisements
Refresh
Refresh All
Resend
Resend All
Find
Name
Instance
Remote Address
Remote AS
M...
R...
TTL
Remote ID
Uptime
Prefix Co...
State
peer to gate ITS
idren
103.78.234.65
64302
no
no
d...
103.78.235.251
07:49:20
8
established
1 item
BGP
Instances
VRFs
Peers
Networks
Aggregates
VPN4 Routes
Advertisements
Find
Network
Synchroni...
101.203.169.0/24
no
103.78.233.0/25
no
103.78.234.64/27
no
103.78.235.128/25
no
4 items
BGP
Instances
VRFs
Peers
Networks
Aggregates
VPN4 Routes
Advertisements
Find
all
Peer
Prefix
Nexthop
AS Path
Origin
Local P...
MED
peer t...
101.203.169.0/24
103.78.234.66
igp
100
1 item


Index of /TA
101.203.173.2/TA/
Index of /TA
Name
Last modified
Size
Description
Parent Directory
Buku Saku IUREN 2017...2
19-May-2018 22:38
2.6M
CST_2014_SP5.iso
12-Feb-2015 09:30
4.1G
Apache/2.2.11 (Win32) DAV/2 mod_ssl/2.2.11 OpenSSL/0.9.8i PHP/5.2.9 Server at 101.203.173.2 Port 80
CST_2014_SP5.iso
Download status
Speed Limiter
Options on completion
http://101.203.173.2/TA/CST_2014_SP5.iso
Status
Receiving data...
File size
4.127 GB
Downloaded
2.052 MB (0.04 %)
Transfer rate
446.470 KB/sec
Time left
3 hour(s) 9 min
Resume capability
Yes
<< Hide details
Pause
Cancel
Start positions and download progress by connections
N.
Downloaded
Info
1
301.045 KB
Receiving data...
2
383.132 KB
Receiving data...
3
355.335 KB
Receiving data...
4
365.439 KB
Receiving data...
5
50.771 KB
Receiving data...
6
213.312 KB
Receiving data...


10:03 PM
Index of /TA
101.203.173.2/TA/
Index of /TA
Name
Last modified
Size
Description
Parent Directory
Buku Saku IUREN 2017...2
19-May-2018 22:38
2.6M
CST_2014_SP5.iso
12-Feb-2015 09:30
4.1G
Apache/2.2.11 (Win32) DAV/2 mod_ssl/2.2.11 OpenSSL/0.9.8i PHP/5.2.9 Server at 101.203.173.2 Port 80
10:03 PM



Konfigurasi UGM

[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
 # NAME TYPE ACTUAL-MTU L2MTU

```
 0    ether1                    ether          1500  1600
 1  S ether2                    ether          1500  1598
 2 RS ether3                    ether          1500  1598
 3  S ether4                    ether          1500  1598
 4 RS ether5                    ether          1500  1598
 5  S wlan1                     wlan           1500  1600
[admin@MikroTik] > routing bgp instance print
Flags: * - default, X - disabled
 0 *X name="default" as=65530 router-id=0.0.0.0 redistribute-
connected=n>
     redistribute-static=no redistribute-rip=no redistribute-ospf=no
     redistribute-other-bgp=no out-filter=""
     client-to-client-reflection=yes ignore-as-path-len=no
     routing-table=""
 1    name="UGM" as=45705 router-id=202.43.93.246
     redistribute-connected=no redistribute-static=no
     redistribute-rip=no redistribute-ospf=no
[admin@MikroTik] > routing bgp peer print
Flags: X - disabled, E - established
 #   INSTANCE     REMOTE-ADDRESS                REMOTE-
AS
 0 E UGM          202.43.93.247                 64302
[admin@MikroTik] > routing bgp peer print status
Flags: X - disabled, E - established
 0 E name="peer to UGM GATE" instance=UGM remote-
address=202.43.93.247
    remote-as=64302 tcp-md5-key="" nexthop-choice=force-self
    multihop=no route-reflect=no hold-time=3m ttl=default in-filter=""
    out-filter="" address-families=ip default-originate=never
    remove-private-as=no as-override=no passive=no use-bfd=no
    remote-id=103.78.233.2 local-address=202.43.93.246
uptime=6h47m41s
    prefix-count=7 updates-sent=79716 updates-received=79721
    withdrawn-sent=0 withdrawn-received=1 remote-hold-time=3m
```

used-hold-time=3m used-keepalive-time=1m refresh-capability=yes as4-capability=yes state=established

```
[admin@MikroTik] > routing filter print
Flags: X - disabled
 0   chain=NEXTHOP-IDREN invert-match=no action=passthrough
     set-out-nexthop=202.43.93.246 set-bgp-prepend-path=""
```

Konfigurasi *GATE*-ITS

```
[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
 #    NAME                TYPE       ACTUAL-MTU L2MTU
 0    ether1              ether          1500  1600
 1  R  ether2              ether          1500  1598
 2  R  ether3              ether          1500  1598
 3  R  ether4              ether          1500  1598
 4  R  ether5              ether          1500  1598
 5  X  wlan1               wlan           1500  1600
[admin@MikroTik] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   101.203.168.1/24   101.203.168.0   ether2
 1   103.78.235.251/25  103.78.235.128  ether3
 2   103.78.234.65/30   103.78.234.64   ether4
 3   202.46.129.245/30  202.46.129.244  ether5
[admin@MikroTik] > routing bgp instance print
Flags: * - default, X - disabled
 0 *X name="default" as=65530 router-id=0.0.0.0 redistribute-
connected=no
     redistribute-static=no redistribute-rip=no redistribute-ospf=no
     redistribute-other-bgp=no out-filter=""
     client-to-client-reflection=yes ignore-as-path-len=no
     routing-table=""
```

```
 1   name="ITS GATE " as=64302 router-id=103.78.235.251
     redistribute-connected=no redistribute-static=no
     redistribute-rip=no redistribute-ospf=no
     redistribute-other-bgp=no out-filter=""
     client-to-client-reflection=no ignore-as-path-len=no
     routing-table=""
[admin@MikroTik] > routing bgp peer print
Flags: X - disabled, E - established
 #   INSTANCE     REMOTE-ADDRESS                  REMOTE-
AS
 0 E ITS GATE     103.78.234.66                   64302
 1 E ITS GATE     202.46.129.246                  38331
 2 E ITS GATE     103.78.235.254                  64302
[admin@MikroTik] > routing bgp peer print status
Flags: X - disabled, E - established
 0 E name="peer to IAIN PONOROGO" instance=ITS GATE
     remote-address=103.78.234.66 remote-as=64302 tcp-md5-key=""
     nexthop-choice=force-self multihop=no route-reflect=no
     hold-time=3m ttl=default in-filter=""
     out-filter=nexthop IAIN PONOROGO address-families=ip
     default-originate=never remove-private-as=no as-override=no
     passive=no use-bfd=yes remote-id=103.78.234.66
     local-address=103.78.234.65 uptime=7h32m32s prefix-count=4
     updates-sent=88196 updates-received=10 withdrawn-sent=1
     withdrawn-received=1 remote-hold-time=3m used-hold-time=3m
     used-keepalive-time=1m refresh-capability=yes as4-capability=yes
     state=established

 1 E name="peer to ITS" instance=ITS GATE  remote-
address=202.46.129.246
     remote-as=38331 tcp-md5-key="" nexthop-choice=force-self
     multihop=no route-reflect=no hold-time=3m ttl=default in-filter=""
     out-filter="" address-families=ip default-originate=never
     remove-private-as=no as-override=no passive=no use-bfd=no
```

remote-id=202.46.129.246 local-address=202.46.129.245
uptime=7h31m55s prefix-count=2 updates-sent=88202
updates-received=88194 withdrawn-sent=1 withdrawn-received=1
remote-hold-time=3m used-hold-time=3m used-keepalive-time=1m
refresh-capability=yes as4-capability=yes state=established

2 E name="peer to idren" instance=ITS GATE
remote-address=103.78.235.254 remote-as=64302 tcp-md5-key=""
nexthop-choice=force-self multihop=no route-reflect=no
hold-time=3m ttl=default in-filter="" out-filter=nexthop to IDREN

[admin@MikroTik] > routing filter print
Flags: X - disabled
0 chain=nexthop to IDREN invert-match=no action=passthrough
set-out-nexthop=103.78.235.251 set-bgp-prepend-path=""

1 chain=nexthop IAIN PONOROGO invert-match=no
action=passthrough
set-out-nexthop=103.78.234.65 set-bgp-prepend-path=""

Konfigurasi ITS

[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
NAME TYPE ACTUAL-MTU L2MTU
0 ether1 ether 1500 1600
1 R ether2 ether 1500 1598
2 ether3 ether 1500 1598
3 ether4 ether 1500 1598
4 R ether5 ether 1500 1598
5 X wlan1 wlan 1500 1600
[admin@MikroTik] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
ADDRESS NETWORK INTERFACE

```
 0   101.203.170.1/24   101.203.170.0   ether2
 1   202.46.129.246/30  202.46.129.244  ether5
[admin@MikroTik] > routing bgp instance print
Flags: * - default, X - disabled
 0 *X name="default" as=65530 router-id=0.0.0.0 redistribute-
connected=no
      redistribute-static=no redistribute-rip=no redistribute-ospf=no
      redistribute-other-bgp=no out-filter=""
      client-to-client-reflection=yes ignore-as-path-len=no
      routing-table=""

 1    name="ITS" as=38331 router-id=202.46.129.246
      redistribute-connected=no redistribute-static=no
      redistribute-rip=no redistribute-ospf=no
      redistribute-other-bgp=no out-filter=""
      client-to-client-reflection=no ignore-as-path-len=no
      routing-table=""
[admin@MikroTik] > routing bgp peer print
Flags: X - disabled, E - established
 #   INSTANCE     REMOTE-ADDRESS                  REMOTE-
AS
 0 E ITS          202.46.129.245                  64302
[admin@MikroTik] > routing filter print
Flags: X - disabled
[admin@MikroTik] > routing bgp peer print status
Flags: X - disabled, E - established
 0 E name="peer to ITS GATE" instance=ITS remote-
address=202.46.129.245
    remote-as=64302 tcp-md5-key="" nexthop-choice=force-self
    multihop=no route-reflect=no hold-time=3m ttl=default in-filter=""
    out-filter="" address-families=ip default-originate=never
    remove-private-as=no as-override=no passive=no use-bfd=no
    remote-id=103.78.235.251 local-address=202.46.129.246
    uptime=6h57m27s prefix-count=8 updates-sent=81459
```

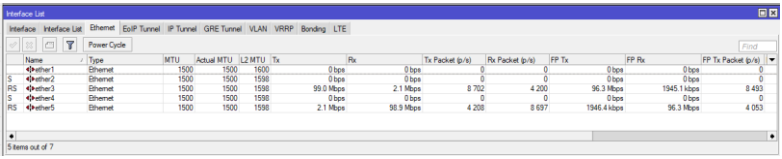
updates-received=81466 withdrawn-sent=1 withdrawn-received=1
remote-hold-time=3m used-hold-time=3m used-keepalive-time=1m
refresh-capability=yes as4-capability=yes state=established

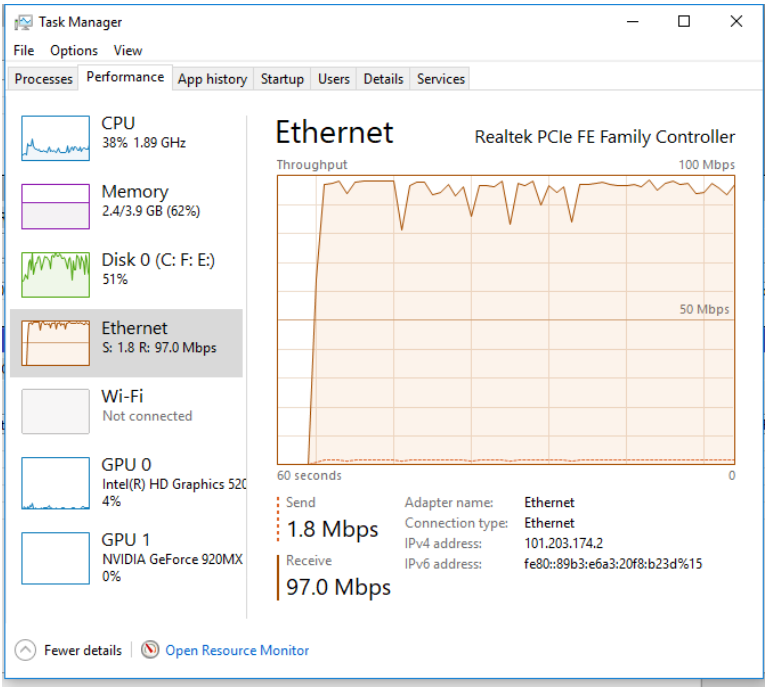
Konfigurasi *GATE*-UGM

```
[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
 #    NAME                TYPE       ACTUAL-MTU L2MTU
 0  R  ether1              ether          1500  1600
 1  R  ether2              ether          1500  1598
 2     ether3              ether          1500  1598
 3     ether4              ether          1500  1598
 4  R  ether5              ether
 5  X  wlan1               wlan

 [admin@MikroTik] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   101.203.171.1/24   101.203.171.0   ether2
 1   103.78.233.2/25    103.78.233.0    ether1
 2   202.43.93.247/24   202.43.93.0     ether5
[admin@MikroTik] > routing bgp instance print
Flags: * - default, X - disabled
 0 *X name="default" as=65530 router-id=0.0.0.0 redistrib
     redistribute-static=no redistribute-rip=no redistri
     redistribute-other-bgp=no out-filter=""
     client-to-client-reflection=yes ignore-as-path-len=
     routing-table=""

[admin@MikroTik] > routing bgp peer print
Flags: X - disabled, E - established
 #   INSTANCE     REMOTE-ADDRESS
 0 E UGM GATE     103.78.233.1
 1 E UGM GATE     202.43.93.246
```

```
[admin@MikroTik] > routing bgp peer print status
Flags: X - disabled, E - established
 0 E name="peer to idren" instance=UGM GATE remote-addres
     remote-as=64302 tcp-md5-key="" nexthop-choice=force-
     multihop=no route-reflect=no hold-time=3m ttl=defaul
     out-filter=NEXTHOP-IDREN address-families=ip
     default-originate=never remove-private-as=no as-over

[admin@MikroTik] > routing filter print
Flags: X - disabled
 0   chain=NEXTHOP-IDREN invert-match=no action=passthrou
     set-out-nexthop=103.78.233.2 set-bgp-prepend-path=""

 1   chain=nexthop ITS invert-match=no action=passthrough
     set-out-nexthop=203.189.122.194 set-bgp-prepend-path
-- [Q quit|D dump|down]
```

Hasil *throughput* saat UGM melakukan *download file* ke *server* IdREN




Task Manager
File Options View
Processes Performance App history Startup Users Details Services
CPU
38% 1.89 GHz
Memory
2.4/3.9 GB (62%)
Disk 0 (C: F: E:)
51%
Ethernet
S: 1.8 R: 97.0 Mbps
Wi-Fi
Not connected
GPU 0
Intel(R) HD Graphics 520
4%
GPU 1
NVIDIA GeForce 920MX
0%
Ethernet
Realtek PCIe FE Family Controller
Throughput
100 Mbps
50 Mbps
60 seconds
0
Send
1.8 Mbps
Receive
97.0 Mbps
Adapter name: Ethernet
Connection type: Ethernet
IPv4 address: 101.203.174.2
IPv6 address: fe80::89b3:e6a3:20f8:b23d%15
Fewer details
Open Resource Monitor

---Halaman ini sengaja dikosongkan---

RIWAYAT PENULIS

Penulis adalah lulusan dari SMA Negeri 1 Sidayu Gresik, kemudian melanjutkan studinya di D3 Jurusan Teknik Telekomunikasi PENS di tahun 2012 dan sekarang di ITS bidang studi Telekomunikasi Multimedia 2016.

Nama	:	Umi Faridah
NRP	:	07111645000021
Tempat, tanggal Lahir	:	Gresik, 8 April 1994
Alamat	:	RT.13 RW.IV No.69 Campurejo Panceng Gresik 61156
No. Telp	:	081252857596
E-mail	:	umifaridahsubhan@gmail.com