

IMPLEMENTASI ALGORITMA DIJKSTRA PADA PROTOKOL ROUTING OPEN SHORTEST PATH FIRST DENGAN MENGGUNAKAN SIMULASI GNS3

Siti Yusmalinda¹, Wydyanto², Devi Udariansyah³

Mahasiswi Universitas Bina Darma¹, Dosen Universitas Bina Darma^{2,3}

Jalan Jendral Ahmad Yani No.3, Palembang 30264

E-Mail : Sitiyusmalinda23@gmail.com¹, wydyanto@binadarma.ac.id², devi.udariansyah@binadarma.ac.id³

ABSTRACT: A process in a computer network that is very important for data communications can run well called routing. In the routing process requires a routing algorithm that serves to direct and determine the best path or alternative path if there is a path that is broken (down) to be passed the data packets from the source node to the destination node. There are many routing algorithms used in computer networks, and one of them is the Dijkstra Algorithm which is also known as link-state algorithm. In this research will discuss about work system Dijkstra Algorithm on OSPF routing protocol. For the manufacture of Dijkstra Algorithm simulation, using GNS3 software, while to calculate the quality of the network through QoS which aims to know the management of bandwidth, packet loss, and response time. Measurement of parameters that m using the packet monitoring application that is Axenetools by means of sending ICMP packets on each client is determined by time, after which time the researcher finished observing the number of packets received during the delivery of the ICMP packet. With this research is expected to be a reference about routing algorithms in general and Dijkstra Algorithm in particular.

Keywords: Routing, Routing Algorithm, Dijkstra Algorithm (link-state), QoS, Axenetools, GNS3.

ABSTRACT: Sebuah proses dalam jaringan komputer yang sangat penting agar komunikasi data dapat berjalan dengan baik disebut dengan *routing*. Di dalam proses *routing* tersebut memerlukan algoritma *routing* yang berfungsi untuk mengarahkan dan menentukan jalur terbaik atau jalur *alternative* bila ada jalur yang terputus (*down*) yang akan dilalui paket data dari simpul sumber ke simpul tujuan. Ada banyak algoritma *routing* yang digunakan dalam jaringan komputer, dan salah satunya adalah Algoritma *Dijkstra* yang juga dikenal dengan algoritma *link-state*. Dalam penelitian ini akan membahas tentang sistem kerja Algoritma *Dijkstra* pada protokol *routing OSPF*. Untuk pembuatan simulasi Algoritma *Dijkstra*, menggunakan *software GNS3*, sedangkan untuk menghitung kualitas jaringan yang dilalui menggunakan *QoS* yang bertujuan untuk mengetahui pengelolaan *bandwidth*, *packet loss*, dan *response time*. Pengukuran parameter yang m menggunakan aplikasi monitoring paket yaitu *Axenetools* yaitu dengan cara melakukan pengiriman paket ICMP pada masing-masing *client* ditentukan dengan waktu, setelah waktu tersebut habis peneliti melakukan pengamatan jumlah paket yang diterima selama pengiriman paket ICMP tersebut. Dengan penelitian ini diharapkan dapat menjadi referensi mengenai algoritma *routing* pada umumnya dan Algoritma *Dijkstra* pada khususnya.

Kata Kunci : Routing, Algoritma Routing, Algoritma *Dijkstra* (link-state), QoS, Axenetools, GNS3.

1. PENDAHULUAN

Dunia teknologi informasi pada saat ini semakin berkembang sangat pesat dengan menggunakan teknologi informasi berbasis jaringan komputer akan mempermudah user dalam melakukan pertukaran informasi, akan tetapi dalam melakukan pertukaran informasi tersebut perlu adanya perancangan jaringan komputer terlebih dahulu agar jaringan yang diterapkan nantinya terarah sesuai dengan kebutuhan, untuk melakukan perencanaan itu aplikasi GNS3 adalah salah satu aplikasi simulator jaringan komputer yang dapat merancang topologi jaringan yang lebih kompleks

dibandingkan dengan simulator lainnya. Program ini dapat dijalankan di berbagai sistem operasi, seperti *Windows*, *Linux*, atau *Mac OS X*.

Fitur-fitur yang didukung GNS3 antara lain, desain jaringan kualitas tinggi dan topologi jaringan yang kompleks, mengemulasikan berbagai platform *Cisco IOS router*, *IPS*, *PIX* dan *ASA firewall*, *JUNOS*, Simulasi *Ethernet* sederhana, *ATM* dan *Frame Relay switch*.

Dalam hal melakukan transper data agar paket yang dikirim sesuai rute yang telah ditentukan. Algoritma ditemukan oleh Edsger W. Dijkstra dan di publikasi pada tahun 1959. Algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau

untuk mencapai tujuan tertentu. Dalam dunia komputer, Algoritma sangat berperan penting dalam pembangunan suatu *software*. Algoritma bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalnya titik menggambarkan gedung dan garis menggambarkan jalan, maka algoritma *Dijkstra* melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.

Algoritma *Dijkstra* merupakan salah satu varian bentuk algoritma yang sering di gunakan dalam pemecahan persoalan terkait masalah optimasi pencarian lintasan terpendek sebuah lintasan yang mempunyai panjang minimum dari verteks a ke z dalam *graph* berbobot, bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh *node* negative. (Novandi, 2007). Namun jika terjadi demikian, maka penyelesaian yang diberikan adalah infinity (Tak Hingga). Pada algoritma *Dijkstra*, *node* digunakan karena algoritma *Dijkstra* menggunakan *graph* berarah untuk penentuan rute listasan terpendek. Untuk menghitung routing table, setiap *router* menggunakan algoritma *Dijkstra* untuk *link state database*-nya. Algoritma *Dijkstra* menghitung jalur tersingkat diantara dua titik pada jaringan dengan menggunakan grafik yang tersusun dari *node* dan *edge*. Algoritma membagi *node-node* menjadi dua pasang yaitu sementara dan permanen. Algoritma memilih *node-node*, membuatnya sementara, memeriksanya, dan jika mereka melampaui kriteria, buat *node-node* tersebut secara permanen.

Router Protocol dapat mengatur proses dalam pencarian jalur komunikasi atau *routing* yang disebut dengan *protocol routing*. Dengan adanya *protocol routing*, penyampaian data dari satu komputer ke komputer lain menjadi cepat dan efisien. Untuk menghitung kualitas jaringan yang dilalui menggunakan *Quality of Service (QoS)*.

Quality of Service (QoS) merupakan teknologi yang memungkinkan administrator jaringan untuk menangani berbagai efek dari terjadinya kongesti pada lalu lintas aliran paket dari berbagai layanan untuk memanfaatkan sumber daya jaringan secara optimal, dibandingkan dengan menambah kapasitas fisik jaringan tersebut. Flanagan dkk (2003) mendefinisikan bahwa *QoS* adalah teknik untuk mengelola *bandwidth*, *packet loss*, dan *response time* untuk aliran paket dalam jaringan. Tujuan dari mekanisme *QoS* adalah mempengaruhi setidaknya satu diantara parameter dasar *QoS* yang telah ditentukan.

Dalam *routing protocol* pencarian jalur komunikasi yang digunakan pada jaringan TCP/IP saat ini adalah salah satunya *protocol routing Open Shortest Path First (OSPF)*.

OSPF atau *Open Shortest Path First* merupakan sebuah *routing* protokol berjenis *IGRP (Interior Gateway Routing Protocol)* yang hanya dapat bekerja dalam jaringan internal suatu organisasi atau perusahaan. Jaringan internal maksudnya adalah jaringan di mana seseorang masih memiliki hak untuk menggunakan, mengatur, dan memodifikasinya. Pada *OSPF* dikenal kondisi *adjacency* antar *router*. Sebelum *router-router* tersebut bertukar informasi *routing*, maka sebuah *router* harus terlebih dahulu mencapai kondisi *adjacency* dengan *router* tetangganya. *Router-router* tidak akan bertukar informasi *routing* jika kondisi *adjacency* belum tercapai. (towidjojo, 2013). Untuk mencapai kondisi *convergence*, *router-router* yang menjalankan *OSPF (Open Shortest Path First)* akan terlebih dahulu mengumpulkan informasi mengenai status dari *directly connected network* yang dimilikinya. Inilah yang mendasari nama *link state routing protocol*, karena *router-router* selalu berusaha mengumpulkan status (*state*) dari setiap *interface* yang dimiliki (*link*). (towidjojo, 2013). Di dalam *OSPF*, proses *routing* yang merupakan implementasi dari graf berarah, masalah yang biasanya muncul adalah ketika rute terputus maka *routing* ini dapat melakukan perubahan kembali agar jaringan dapat tetap terjaga atau efisien/efektif. Oleh karena itu penulis berkeinginan melakukan pengujian terhadap algoritma *dijkstra* untuk penentuan jalur terbaik atau rute terpendek agar paket sampai ketujuan.

Berdasarkan latar belakang tersebut untuk menerapkannya maka penulis tertarik untuk melakukan penelitian rute yang dipilih oleh algoritma *Dijkstra* pada *protokol routing OSPF* dengan menggunakan simulasi GNS3 adapun judul penelitian yang di angkat adalah “Implementasi Algoritma *Dijkstra* Pada Protokol *Routing Open Shortest Path First* Dengan Menggunakan Simulasi GNS3”.

2. Landasan Teori

A. Implementasi

Menurut Usman (2002) Implementasi adalah bermuara pada aktivitas, aksi, tindakan, atau adanya mekanisme suatu sistem. Implementasi bukan sekedar aktivitas, tetapi suatu kegiatan yang terencana dan untuk mencapai tujuan kegiatan.

Pengertian implementasi yang dikemukakan di atas, dapat dikatakan bahwa implementasi adalah bukan sekedar aktivitas, tetapi suatu kegiatan yang terencana dan dilakukan secara sungguh-sungguh berdasarkan acuan standar tertentu untuk mencapai tujuan. Oleh karena itu implementasi tidak berdiri sendiri tetapi dipengaruhi oleh objek berikutnya.

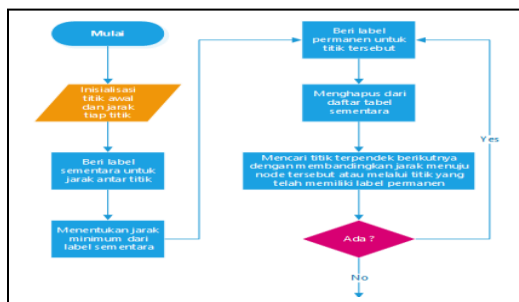
B. Simulasi

Simulasi adalah sebagai suatu model sistem dimana komponennya di presentasikan oleh prosesor-prosesor aritmatika dan logika yang di jalankan komputer untuk memperkirakan sifat-sifat dinamis sistem tersebut. (Emshoff dan Simun, 1970). Menurut Shannon (1975) simulasi adalah merupakan proses perencanaan model dari sistem nyata yang dilanjutkan dengan pelaksanaan eksperimen terhadap model untuk mempelajari perilaku sistem atau evaluasi strategi. Menurut Banks dan Carson (1984) simulasi adalah tiruan sistem nyata yang dikerjakan secara manual atau komputer, yang kemudian di observasi dan disimpulkan untuk mempelajari karakteristik sistem. Menurut Khosnevis (1994) simulasi merupakan proses aplikasi membangun model dari sistem nyata atau usulan sistem, melakukan eksperimen dengan model tersebut untuk menjelaskan perilaku sistem, mempelajari kinerja sistem atau untuk membangun sistem baru sesuai dengan kinerja yang diinginkan

Jadi dapat disimpulkan bahwa simulasi yaitu sebelum mekakukan percobaan, atau melakukan penelitian perlu adanya perencanaan dan perancangan terlebih dahulu agar tindakan yang akan dilakukan tidak menyimpang dan sesuai dengan yang diharapkan.

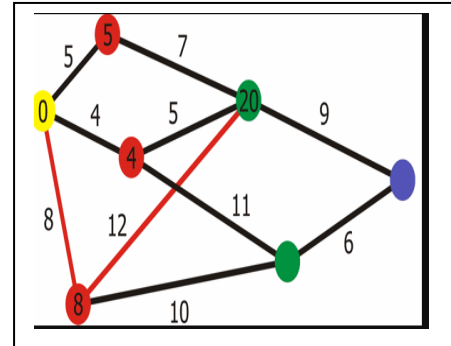
C. Algoritma Dijkstra

Algoritma *Dijkstra* ditemukan oleh seorang ilmuwan komputer berkebangsaan belanda bernama Edsger W. Dijkstra dan di publikasikan pada tahun 1959. Algoritma ini merupakan sebuah algoritma rakus (*greedy algorithm*) yang dipakai dalam memecahkan permasalahan terkait masalah optimasi pencarian lintasan terpendek sebuah lintasan yang mempunyai panjang minimum dari verteks *a* ke *z* dalam graph berbobot, bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh *node* negatif. Namun jika terjadi demikian, maka penyelesaian yang diberikan adalah infinity (Tak Hingga). Pada algoritma *Dijkstra*, *node* digunakan karena algoritma *Dijkstra* menggunakan *graph* berarah untuk penentuan rute listasan terpendek. Berikut *flowchart* algoritma *Dijkstra* dapat dilihat pada gambar 2.1



Gambar 2.1 Pseudo Code dan Flowchart Algoritma Dijkstra

Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalnya titik menggambarkan gedung dan garis menggambarkan jalan, maka algoritma *Dijkstra* melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.



Gambar 2.2 Implementasi Dijkstra

Pertama-tama tentukan titik mana yang akan menjadi *node* awal, lalu beri bobot jarak pada *node* pertama ke *node* terdekat satu persatu, *Dijkstra* akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap inilah rutan logika dari algoritma *Dijkstra*:

- 1) Beri nilai bobot (jarak) untuk ssetiap titik ke titik lainnya, lalu set nilai 0 pada *node* awal dan nilai tak hingga terhadap *node* lain (belum terisi).
- 2) Set semua *node* “belum terjamah” dan set *node* awal sebagai “*node* keberangkatan”.
- 3) Dari no keberangkatan, pertimbangkan *node* tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke node C berjarak 2, maka jarak C melewati B menjadi $6+2=8$. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
- 4) Saat kita selesai mempertimbangkan setiap jarak terhadap *node* tetangga, tandai *node* yang telah terjamah sebagai “*node* terjamah”, *node* terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
- 5) Set “*node* belum terjamah” dengan jarak terkecil (dari *node* keberangkatan) sebagai “*node* keberangkatan” selanjutnya dan lanjutkan dengan kembali ke step 3.

D. Protokol Jaringan

Menurut stinke (2003), agar dapat saling berkomunikasi satu sama lain, komputer-komputer yang terhubung dalam suatu jingan harus mempunyai satu set peraturan yang sama. Peraturan-peraturan tersebut, disebut dengan protokol. Seperti halnya dua orang yang berlainan bangsa, maka untuk berkomunikasi diperlukan penerjemah (*interpreter*) atau satu bahasa yang dimengerti kedua belah pihak. Untuk itu, maka badan dunia yang menangani masalah standarisasi *ISO (International Standardization Organization)* membuat aturan baku yang dikenal dengan nama model referensi *OSI (Open System Interconnection)*.

E. OSI Reference Model

OSI Refence Model adalah model konseptual yang terdiri dari tujuh lapis, masing-masing menetapkan fungsi jaringan tertentu. *OSI Reference Model* menggambarkan bagaimana informasi dari satu aplikasi komputer diangkat melintasi jaringan ke aplikasi yang sama (atau serupa) di komputer lain. (Castelli, 2003).

F. Quality Of Service (QoS)

Menurut Ningsih dkk (2004) *QoS* adalah kemampuan sebuah jaringan untuk menyediakan layanan yang lebih baik lagi bagi layanan trafik yang melewatinya. *QoS (Quality of Service)* : “*the collective effect of service performance which determines the degree of satisfaction of a user of the service*”. *International Telecommunication Union (ITU 1998, X.641)*. [1] Berdasarkan beberapa definisi diatas, dapat disimpulkan *QoS (Quality of Service)* adalah kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan *bandwith*, mengatasi *jitter* dan *delay*. Parameter *QoS* adalah *latency*, *jitter*, *packet loss*, *throughput*, *MOS*, *echo cancellation* dan *PDD*

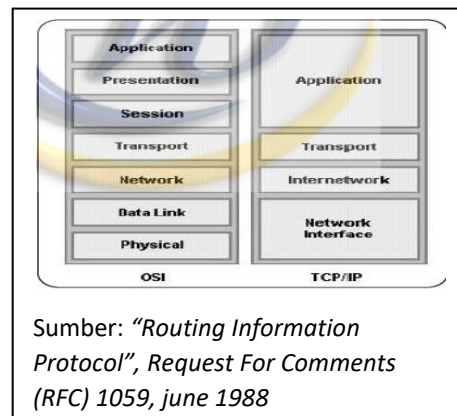
G. TCP/IP

Menurut Kozierok (2005), *Transmission Control Protokol/Internet Protocol (TCP/IP)* adalah satu set standar aturan komunikasi data yang digunakan dalam proses transfer data dari satu komputer ke komputer lain di dalam jaringan komputer tanpa melihan perbedaan jenis *hardware*. Model *TCP/IP* merupakan hasil eksperimen dan pengembangan terhadap *ARPANET*, sebuah *packet-switching network* milik Departemen Pertahanan Amerika Serikat. Model ini disebut sebagai *Internet Protocol Suite*.

H. Perbandingan Umum Model OSI dengan TCP/IP

perbedaan antara model *OSI* dengan model *TCP/IP*:

- 1) Implementasi model *OSI* menekankan pada penyediaan layanan transfer data yang *reliable*, sementara *TCP/IP* memperlakukan *reliability* sebagai masalah *end-to-end*.
- 2) Setiap *layer* pada *OSI* mendeteksi dan menangani kesalahan pada semua data yang dikirimkan. *layer Transport* pada *OSI* memeriksa *reliability* di *source-to-destinatio*.
- 3) pada *TCP/IP*, *control reliability* dikonsentrasikan pada *Layer Transport*. *Layer Transport* menangani semua kesalahan yang terdeteksi dan memulihkannya. *Layer Transport TCP/IP* menggunakan *checksum*, *acknowledgment*, dan *timeout* untuk mengontrol transmisi dan menyediakan verifikasi *end-to-end*.



Gambar 2.5 Perbandingan OSI DAN TCP/IP

3. Metodologi Penelitian

Metode yang akan digunakan pada penelitian ini adalah *Action Research* atau penelitian tindakan merupakan salah satu bentuk rancangan penelitian, dalam penelitian tindakan peneliti mendeskripsikan dan menjelaskan suatu situasi sosial pada waktu yang bersamaan dengan melakukan perubahan atau intervensi dengan tujuan perbaikan atau partisipasi. *Action research* dalam pandangan tradisional adalah suatu kerangka penelitian pemecahan masalah, dimana terjadi kolaborasi antara peneliti dan *client* dalam mencapai tujuan dikemukakan oleh Kurt Lewin, (1973), dan disitasi Sulaksana (2004), sedangkan pendapat Davison, Martinsons & Kock (2004), menyebutkan penelitian tindakan, sebagai sebuah metode penelitian, didirikan atas asumsi bahwa teori dan praktik dapat secara tertutup diintegrasikan dengan pembelajaran dari hasil

intervensi yang direncanakan setelah diagnosis yang rinci terhadap konteks masalahnya.

Menurut BOPTN (2013:1) *Action Research* merupakan kegiatan penelitian yang difokuskan secara *kolaboratif* untuk menyelesaikan permasalahan yang ada, dimana permasalahan lebih kepada masyarakat secara langsung maupun tidak langsung.

A. Metode Pengumpulan Data

Adapun metode pengumpulan data yang digunakan dalam menyelesaikan penelitian Tugas Akhir ini, antara lain :

- 1) Pengamatan (*Observasi*)
Merupakan metode pengumpulan data dengan cara mengadakan tinjauan secara langsung ke objek yang detile. Maka penulis melakukan pengamatan secara langsung ke Amik Sigma.
- 2) Studi Pustaka
Mengumpulkan data dengan cara mencari dan mempelajari data-data dari buku-buku ataupun dari referensi lain yang berhubungan dengan penulisan laporan penelitian proposal.
- 3) Pengujian (*Testing*)
Metode ini mengadakan pengujian terhadap objek yang akan diteliti dengan melakukan pengendalian terhadap unsur-unsur penting ke dalam situasi yang sedemikian rupa untuk mengetahui sistem kerja algoritma *Dijkstra* pada protokol *routing OSPF*.

B. Penelitian Tindakan (*Action Research*)

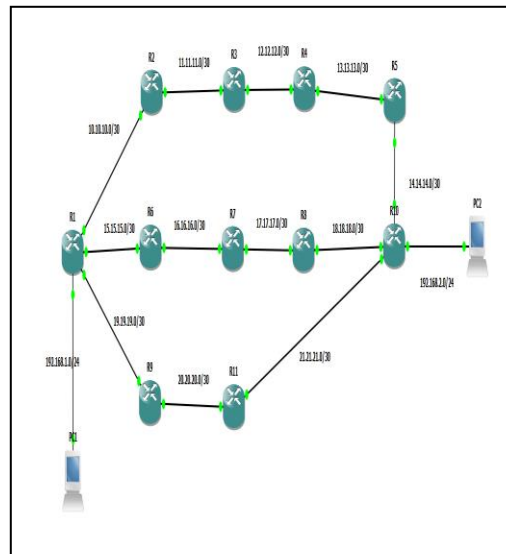
- 1) Melakukan Diagnosa (*Diagnosing*)
Pada tahap ini peneliti melakukan diagnosa kebutuhan perangkat yang diperlukan dalam pengujian implementasi algoritma *Dijkstra* untuk pencarian jalur tercepat atau rute terbaik untuk sampai ke tujuan. perangkat yang digunakan untuk melakukan implementasi algoritma *Dijkstra* pada protokol *routing OSPF* sangat mempengaruhi dalam penelitian ini sebab apabila perangkat yang digunakan tidak efektif maka pengujian tidak akan berjalan dengan baik.

- 2) Membuat rencana tindakan (*Action Planning*)

Pada tahap ini peneliti memahami pokok permasalahan dan kemudian akan dilanjutkan penyusunan rencana tindakan yang tetap untuk menyelesaikan masalah yang ada,peneliti akan melakukan perancangan jaringan *OSPF*.

- 3) Melakukan tindakan (*Action Taking*)

Peneliti akan mengimplementasikan rencana tindakan yang telah dibuat dengan harapan dapat menyelesaikan masalah. Perencanaan topologi jaringan yang akan diterapkan untuk melakukan penelitian yang mengangkat judul mengenai Implementasi Algoritma *Dijkstra* Pada Protokol *Routing Open Shortest Path First* Dengan Menggunakan Simulasi GNS3 guna mengetahui route terbaik yang dipilih oleh *routing OSPF* nantinya, peneliti menetapkan topologi jaringan sebagai Berikut :



Gambar 3.1 Topologi Jaringan

Dari gambar diatas 3.1 Topologi Jaringan. Dapat kita lihat dalam melakukan penelitian ini peneliti melakukan perancangan jaringan yang dapat menentukan *route* terbaik, peneliti menentukan *bandwidth* dan *node* yang berbeda pada masing-masing *route*, hal ini dilakukan agar paket yang akan berjalan nantinya dapat perhitungan sendiri dengan cara melihat *bandwidth* dan banyaknya *node* yang akan dilalui.

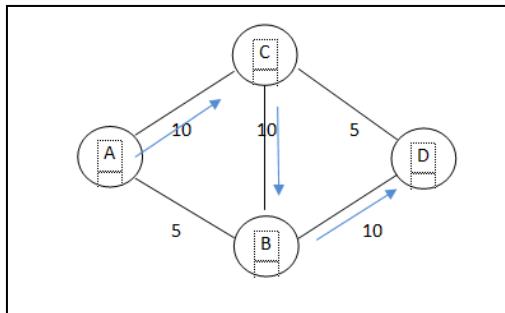
Pada penelitian ini peneliti melakukan perancangan jaringan dengan menentukan tiga *route* yang dapat dilalui paket nantinya, *route* yang telah ditentukan adalah sebagai berikut :

1. *Route* pertama terdapat empat *node* dan masing-masing *bandwidth* ditentukan sebesar 3000MB.
2. *Route* kedua terdapat tiga *node* dan masing-masing *bandwidth* ditentukan sebesar 2000MB.
3. *Route* ketiga terdapat dua *node* dan masing-masing *bandwidth* ditentukan sebesar 1000MB.

Menjalankan tahapan-tahapan dalam mengkonfigurasi *routing* protokol *OSPF* dengan simulasi dengan perangkat lunak pada PC yaitu GNS3 dan melakukan pengujian terhadap *routing* protokol dengan parameter yang telah ditentukan. Setelah *bandwidth* telah ditentukan, peneliti melakukan pengukuran *quality of service* pada jaringan yang memiliki besaran *bandwidth* yang berbeda, seperti yang telah kita dibahas pada bab sebelumnya peneliti nantinya akan melakukan pengukuran mengenai pemakaian *bandwidth*, *packet loss* dan *response time*, tahapan dalam melakukan penelitian adalah dengan cara menentukan besaran *byte*, menganalisa besaran paket yang diterima, dan paket yang gagal.

4) Karakteristik Algoritma *Dijkstra*

Algoritma *Dijkstra* adalah algoritma bertujuan untuk menemukan jarak terpendek berdasarkan bobot terkecil dari satu titik ketitik lainnya yang dianggap bernilai non-negatif. Algoritma *Dijkstra* dapat melakukan pengembangan pencarian bobot terkecil dari satu step ke step selanjutnya, dan dapat menentukan *route* sendiri jika jarak awal bernilai negatif.



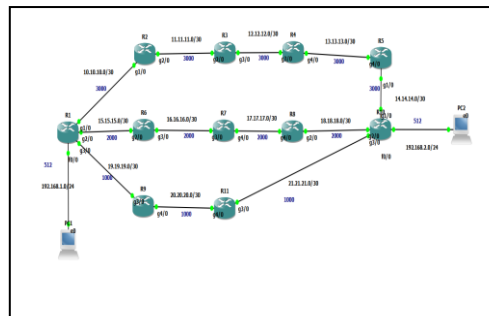
Gambar 3.2 Algoritma *Dijkstra*

Algoritma *Dijkstra* adalah algoritma yang dapat memilih *route* nya sendiri dengan cara merekam kondisi lalu lintas pada graf yang bernilai aktif, kemudian algoritma ini melakukan eksekusi lalu *route* itu sendiri. Dalam hal menentukan *route* lintas jaringan, *routing OSPF* adalah salah satu solusi dalam jaringan yang dapat merekam, eksekusi rute dan melakukan transfer paket dengan menggunakan konsep yang telah diterapkan. Agar paket yang ditransfer sampai ketitik tujuan. Pada gambar 3.2 dapat dilihat jika paket yang akan dikirim dari *node* A menuju *node* D jalur terdekat adalah A,B,D atau A,C,D, tetapi dengan menggunakan *routing OSPF*, *routing* ini akan memilih *route* A,C,B,D dengan ketentuan *route* tersebut dianggap bernilai Non-Negatif.

4. HASIL DAN PEMBAHASAN

A. Hasil Penelitian

Penelitian ini dilakukan dan diterapkan pada aplikasi simulasi GNS3, dengan menggunakan metode *Action Research* dimana metode yang digunakan adalah dengan cara melakukan tindakan dan perangkat yang dibutuhkan telah tersedia lengkap penelitian ini akhirnya mendapatkan hasil yang sesuai dengan perencanaan sebelumnya. Berikut adalah hasil dari perancangan topologi jaringan komputer yang telah di buat menggunakan simulator GNS3.



Gambar 4.1 Hasil Topologi Jaringan

1) Melakukan Evaluasi (*Evaluating*)

Dari hasil perancangan jaringan sesuai topologi diatas peneliti akan melakukan evaluasi terhadap hasil konfigurasi *routing* dengan cara menganalisa *route* jika ada yang terputus, menganalisa *configurasi* apakah sudah berjalan, setelah itu peneliti akan menganalisa *quality of service* pada jaringan yang telah dirancang, parameter yang menjadi tolak ukur dalam penelitian ini adalah menghitung *bandwidth*, *packet loss* ditentukan waktu selama 5 menit, dengan cara tiga kali pengujian dan masing-masing pengujian membedakan besaran pemakaian *byte*. Dengan cara membedakan pemakaian *byte* pada masing-masing pengujian peneliti nantinya berharap mendapatkan data yang sesuai dengan diharapkan.

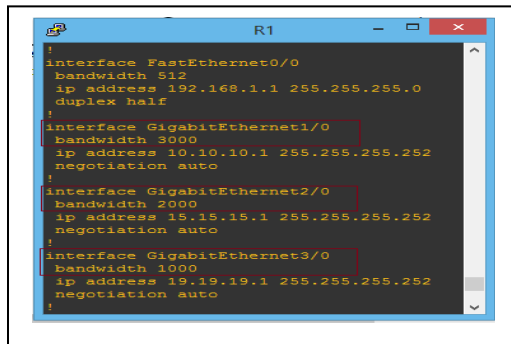
2) Algoritma *Dijkstra* pada Protokol Routing *OSPF*

Dalam melakukan Penerapan algoritma *Dijkstra* pada protokol *routing OSPF* yang bertujuan dapat menentukan *route* terbaik dalam melakukan pertukaran informasi dalam jaringan komputer, algoritma *Dijkstra* dapat memilih rutenya sendiri dengan cara merekam kondisi jaringan yang akan dilewatinya, setelah kondisi jaringan telah diketahui algoritma *Dijkstra* akan memilih rute terbaik dengan menggunakan strategi *greedy* dimana setiap langkah dipilih *edge* dengan bobot terkecil yang dapat menghubungkan sebuah simpul yang sudah terpilih. Algoritma ini

memiliki keunggulan tersendiri jika pada saat melakukan transfer data jaringan tersebut mengalami RTO (*Request Time Out*) maka algoritma *Dijkstra* akan memilih sendiri rute yang akan dilalui agar paket tersebut sampai ketitik tujuan.

3) Menentukan *Bandwidth*

Dalam jaringan komputer *bandwidth* adalah suatu nilai yang digunakan untuk melakukan transfer data, perhitungan yang digunakan oleh *bandwidth* dalam melakukan transfer data adalah *bit/secon* atau sering disebut dengan singkatan bps antar *client* dan *server*. Pada penelitian ini penulis merancang jaringan dengan menentukan sebanyak tiga rute lalu lintas jaringan yang dapat dilalui paket, dimana pada masing masing route dibedakan menjadi tiga tingkatan kapasitas *bandwidth* diantaranya *bandwidth* yang ditentukan adalah sebesar 3000MB,2000MB,1000MB, pada tiap jaringan yang berikut adalah gambar *bandwidth* yang telah di input pada masing-masing *interface* pada router.

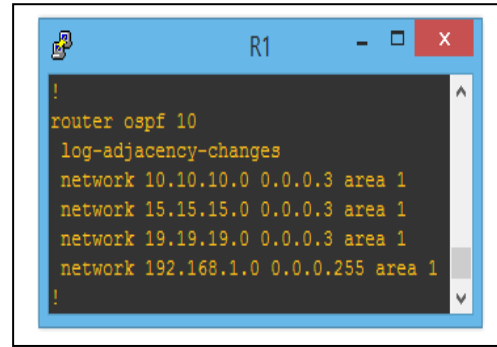


Gambar 4.2 *Bandwidth* di Interface

4) Konfigurasi *Routing OSPF*

Protokol *routing OSPF* yang terletak pada area *Link-State* dimana protokol dinamis ini diciptakan untuk jaringan tingkat metropolitan area dan protokol *routing OSPF* ini dapat mengatur, menjaga, mendistribusi suatu jaringan untuk mengetahui informasi *routing* antar jaringan dan apabila jaringan tersebut terdapat kegagalan route, maka protokol *routing* ini akan melakukan perubahan *route* dalam waktu singkat.

Pada penelitian ini peneliti melakukan konfigurasi *routing OSPF* pada masing masing router dengan menentukan id *routing* 10, dan *network address* menggunakan satu area. Berikut adalah gambar konfigurasi *routing OSPF* yang terdapat pada R1.

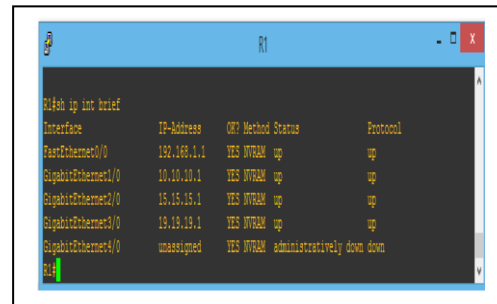


Gambar 4.3 Konfigurasi *Routing OSPF*

Dari gambar diatas Gambar 4.3 peneliti menentukan id *routing* 10, dan *network address* menggunakan satu area.

5) Protokol Aktif

Setelah konfigurasi pada tiap router telah dibuat dari pemasangan *ip address* pada masing-masing *interface*, pemasangan *bandwidth* pada masing-masing *interface*, dan pemasangan *routing* pada tiap-tiap router, peneliti melakukan pengecekan pada tiap route yang telah aktif pada protokol *routing*. Berikut adalah gambar Protokol jaringan komputer yang telah dirancang dan telah siap untuk melakukan operasi.

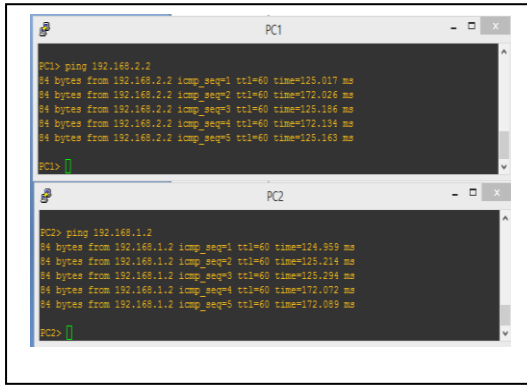


Gambar 4.4 Protokol Aktif

Dari hasil konfigurasi *interface* diatas *interface* yang setatusnya telah UP, menyatakan bahwa *interface* telah siap untuk digunakan.

6) Monitoring Paket & Route

Setelah konfigurasi pada tiap router telah diselesaikain penulis melakukan pengiriman paket *Internet Control Message Protokol (ICMP)* hal ini guna mengetahui paket yang akan di transfer akan sampai pada tujuan yang telah ditentukan. Agar dapat mengetahui hal tersebut peneliti melakukan *ping* antar *client* yang terhubung dalam jaringan, berikut adalah gambar dari pengujian koneksi dengan mengirim paket ICMP pada masing-masing pc *client* :



Gambar 4.5 Monitoring Paket

Pada gambar diatas Gambar 4.5 peneliti melakukan pengiriman paket ICMP, hal ini dilakukan untuk mengetahui jika terdapat kegagalan koneksi pada jaringan, pada gambar diatas saat paket dari PC 1 ke PC 2 melakukan transper data tidak terdapat paket yang gagal, hal ini menunjukan bahwa jaringan yang telah dikonfigurasi dengan menggunakan *routing OSPF* sudah bisa digunakan.

B. Pembahasan

1) Pengukuran Jaringan

Setelah perancangan topologi jaringan dan konfigurasi telah diselesaikan dengan baik peneliti melakukan pengukuran kualitas jaringan yang menggunakan protokol *routing OSPF*. dalam melakukan pengukuran jaringan ini peneliti menggunakan aplikasi monitoring paket yaitu *Axenettols*, dari aplikasi ini data yang diambil pada penelitian ini adalah dengan cara melakukan pengiriman paket ICMP pada masing-masing *client* ditentukan dengan waktu, setelah waktu tersebut habis peneliti melakukan pengamatan jumlah paket yang diterima selama pengiriman paket ICMP tersebut. Berikut adalah pengukuran yang dilakukan :

1. Time

Pada penelitian ini untuk mengetahui jumlah *bandwidth* dan *packet loss* peneliti melakukan penentuan waktu yang di ukur selama paket berjalan, waktu yang ditentukan adalah selama 5 menit, setelah waktu tersebut habis paneliti melakukan stop paket transper dan melihat jumlah terima, jumlah rata-rata *bandwidth* yang digunakan, dan peneliti melihat jumlah paket data yang ditransper tidak sampai ke tujuan. Pada penelitian ini peneliti melakukan analisa paket sebanyak tiga kali dengan menggunakan *byte* yang berbeda, dan waktu transper data yang sama.

2. Bandwidth

Dalam melakukan pengukuran paket data, *bandwidth* adalah sebuah penentuan atau suatu nilai transper data yang dihitung dalam bps antara server dan *client* dalam waktu tertentu, pada penelitian ini *bandwidth* yang digunakan pada masing-masing route di bedakan menjadi tiga yaitu, 3000,2000,1000MB. pada penelitian ini peneliti melakukan pengukuran *bandwidth* dengan menggunakan aplikasi monitoring jaringan "*axenettols*" dengan menggunakan aplikasi ini maka *bandwidth* yang digunakan oleh protokol *routing* dapat dilihat dengan detail. Berikut adalah tabel pengukuran *bandwidth* yang didapat :

Tabel 4.3 Tabel *Bandwidth*

No	Byte	Bandwidth			Time
		Average	Minimum	Maximum	
1	10000 Byte	177 942	54 268	329 339	00:05:00
2	30000 Byte	263 197	91 204	534 479	00:05:00
3	60000 Byte	306 364	121 344	573 544	00:05:00

Dari tabel *bandwidth* di atas dapat di jelaskan bahwa pengiriman data yang dilakukan pada masing-masing *route* yang memiliki besaran *bandwidth* yang berbeda, yaitu 3000MB, 2000MB, 1000MB, maka di dapatkan hasil yaitu:

- Pada pengujian yang pertama dengan melakukan pengukuran menggunakan besaran *bandwidth* 10000 Byte dengan waktu yang di tentukan selama 5 menit maka didapatkan hasil, *Average* (rata-rata) data yang dikirim yaitu sebesar 177 942, *Minimum* data yang dikirim sebesar 54 268, *Maximum* data yang dikirm sebesar 329 339.
- Pada pengujian kedua dengan melakukan pengukuran menggunakan besaran *bandwidth* 30000 Byte dengan waktu yang di tentukan selama 5 menit maka didapatkan hasil, *Average* (rata-rata) data yang dikirim yaitu sebesar 263 197, *Minimum* data yang dikirim sebesar 91 204, *Maximum* data yang dikirim sebesar 534 479.
- Pada pengujian ketiga dengan melakukan pengukuran menggunakan besaran *bandwidth* 60000 Byte dengan waktu yang di tentukan selama 5 menit maka didapatkan hasil, *Average* (rata-rata) data yang dikirim yaitu sebesar 306 364, *Minimum* data yang dikirim sebesar 121 344, *Maximum* data yang dikirim sebesar 573 544.

3. Packet Loss

Packet loss adalah definisi kegagalan transmisi paket data yang tidak mencapai ke jarak tempuh. Pada penelitian ini peneliti melakukan pengambilan data dengan melihat jumlah paket yang terkirim, jumlah paket yang diterima, dan jumlah paket yang gagal, untuk mendapatkan data ini peneliti melakukan hal yang sama seperti pengambilan data *bandwidth* yaitu dengan membedakan *byte* dan memakan waktu yang sama. Berikut adalah tabel *packet loss* yang didapatkan dengan menggunakan aplikasi monitoring jaringan *axenettols*.

Tabel 4.4 Packet Loss

No	Byte	Paket			Time
		Send	Received	Lost	
1	10000 Byte	300	300 (100%)	0 (0%)	00:05:00
2	30000 Byte	299	299(100%)	0 (0%)	00:05:00
3	60000 Byte	300	76 (25%)	244 (75%)	00:05:00

Dari tabel *packet loss* di atas dapat di jelaskan bahwa pengiriman data yang dilakukan pada masing-masing *route* yang memiliki besaran *bandwidth* yang berbeda, yaitu 3000MB, 2000MB, 1000MB, maka di dapatkan hasil yaitu:

- Pada pengujian pertama dengan melakukan pengukuran menggunakan besaran *bandwidth* 10000 Byte dengan waktu yang di tentukan selama 5 menit maka didapatkan hasil, *Send* (kirim) data yang terkirim yaitu sebesar 300, *Received* (terima) data yang diterima sebesar 300 (100%), *Loss* atau data yang tidak terkirim sebesar 0 (0%). Jadi dari penjelasan tersebut, pengiriman data dengan menggunakan besaran *bandwidth* sebesar 10000 Byte tidak mengalami *packet loss*.
- Pada pengujian kedua dengan melakukan pengukuran menggunakan besaran *bandwidth* 30000 Byte dengan waktu yang di tentukan selama 5 menit maka didapatkan hasil, *Send* (kirim) data yang terkirim yaitu sebesar 299, *Received* (terima) data yang diterima sebesar 299 (100%), *Loss* atau data yang tidak terkirim sebesar 0 (0%). Jadi dari penjelasan tersebut, pengiriman data dengan menggunakan besaran *bandwidth* sebesar 20000 Byte masih tidak mengalami *packet loss*.
- Pada pengujian ketiga dengan melakukan pengukuran menggunakan besaran *bandwidth* 60000 Byte dengan waktu yang di tentukan selama 5 menit maka didapatkan hasil, *Send* (kirim) data yang terkirim yaitu

sebesar 300, *Received* (terima) data yang diterima sebesar 76 (25%), *Loss* atau data yang tidak terkirim sebesar 244 (75%) Jadi dari penjelasan tersebut, pengiriman data dengan menggunakan besaran *bandwidth* sebesar 60000 Byte telah mengalami *packet loss* sebesar 244 (75%).

4. Response Time

Response time adalah waktu yang diberikan oleh *interface* pada saat *user request*, mengirim permintaan ke komputer. Secara umum pengguna menginginkan waktu tanggapan sependek-pendeknya tetapi waktu tidak dapat ditentukan karena waktu tanggap yang lambat dapat mempengaruhi kinerja pengguna.

Tabel 4.5 Response Time

No	Byte	Response Time				Time
		Last	Average	Min	Max	
1	10000 Byte	66	112	59	247	00:05:00
2	30000 Byte	86	245	82	615	00:05:00
3	60000 Byte	107	295	70	927	00:05:00

Dari tabel diatas dapat dilihat masing masing paket yang berjalan menggunakan besaran *byte* yang berbeda-beda dan dalam kurun waktu yang ditentukan mendapatkan hasil yang sesuai yaitu

- Dalam pengujian pertama dengan pengiriman paket ICMP menggunakan besaran 10000 Byte dengan waktu yang di tentukan selama 5 menit maka didapatkan hasil, Last (terkirim) sebesar 66, Average (terima) sebesar 112, Minimum paket yang di kirim sebesar 59, maximum paket yang dikirim sebesar 247.
- Dalam pengujian kedua dengan pengiriman paket ICMP menggunakan besaran 30000 Byte dengan waktu yang di tentukan selama 5 menit maka didapatkan hasil, last (terkirim) sebesar 86, Average (terima) sebesar 245, minimum paket yang di kirim sebesar 82, maximum paket yang dikirim sebesar 615.
- Dalam pengujian ketiga dengan pengiriman paket ICMP menggunakan besaran 60000 Byte dengan waktu yang di tentukan selama 5 menit maka didapatkan hasil, last (terkirim) sebesar 107, Average (terima) sebesar 295, minimum paket yang di kirim sebesar 70, maximum paket yang dikirim sebesar 927.

DAFTAR PUSTAKA

Hal ini dilakukan agar peneliti dapat mengetahui seberapa banyak paket yang mampu melewati route pada jaringan agar paket sampai ke tujuan, dengan cara melakukan pengiriman paket ICMP dan membedakan *byte* pada waktu yang sama peneliti melakukan monitoring paket dengan menggunakan aplikasi *axenettols* maka peneliti dapat melihat proses yang berjalan pada saat itu.

5. KESIMPULAN

Setelah penelitian mendapatkan hasil yang sesuai dengan yang diharapkan peneliti dapat menyimpulkan hasil dari penelitian yang mengangkat judul "Implementasi Algoritma *Dijkstra* pada Protokol *Routing OSPF* dengan menggunakan simulasi *GNS3*" adapun kesimpulan yang dapat dipaparkan oleh peneliti adalah sebagai berikut :

1. Protokol *Routing Dinamis OSPF* adalah suatu protokol yang dapat menentukan route terbaik berdasarkan nilai *bandwidth* pada masing-masing *interface* disetiap router,
2. Protokol *routing OSPF* dapat melakukan perubahan topologi apabila terdapat kegagalan *link*, perubahan dapat dilakukan dalam hitungan detik,
3. Algoritma *Dijkstra* adalah sebuah algoritma yang dapat memilih rute dengan bobot terkecil untuk sampai ke semua titik yang dituju, algoritma ini lah yang digunakan *routing OSPF* dalam melakukan transper data.

Alfred. V. Aho, John E. Hopcroft, Jeffrey D.Ullman. 1995, *Data Structures and Algorithms*, Addison-Wesley Publishing Company.

Andrew S. Tanenbaum 1997. *Jaringan Komputer Edisi Bahasa Indonesia dari Computer Networks 3e*, Jilid 2, Prenhallindo: Jakarta.

Definisi protokol dalam jaringan menurut para ahli, diakses dari : <http://repository.widyatama.ac.id/xmlui/bitstream/handle/123456789/3465/Bab%202.pdf?sequence=6> (diakses pada tanggal 30 Maret 2017 pukul 10:37 PM).

Definisi router menurut para ahli, diakses dari: <http://globallavebookx.blogspot.co.id/2015/02/pengertian-router-menurut-ahli.html> diakses pada tanggal 30 Maret 2017 pukul 10:39 PM).

Muhammad Ghufron Mahfudhi. 2010. "Penerapan Algoritma *Dijkstra* pada Link State Routing Protocol untuk Mencari Jalur Terpendek". <http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2010/2011/Makalah2010/MakalahStima2010-092.pdf> (diakses pada tanggal 12 september 2017).

Novandi, Raden Aprian Diaz. 2007. "Studi Implementasi Algoritma *Dijkstra* Pada Protokol Perutean Open Shortest Path First (OSPF)"