

**PROSES PENGIRIMAN PAKET DATA BERBASIS
SIMULATOR NS3**

Diajukan Sebagai Salah Satu Syarat Untuk memperoleh Gelar Sarjana Jurusan
Teknik Informatika



Oleh:

Nama : Dedek

No. Mahasiswa : 09523328

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAMINDONESIA**

2017

LEMBAR PENGESAHAN DOSEN PEMBIMBING
PROSES PENGIRIMAN PAKET DATA BERBASIS SIMULATOR
NS3

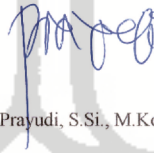
TUGAS AKHIR

Oleh :

Nama : Dedek
NIM : 09523328

Yogyakarta, 14 Januari 2017

Pembimbing,



(Yudi Prayudi, S.Si., M.Kom.)



**HALAMAN PENGESAHAN DOSEN PENGUJI
PROSES PENGIRIMAN PAKET DATA BERBASIS SIMULATOR
NS3**

TUGAS AKHIR

Oleh :
Nama : Dedek
NIM : 09523328

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta,
Tim Penguji,

Yudi Prayudi, S.Si., M.Kom.

Ketua

Syarif Hidayat, S.Kom., M.I.T

Anggota 1

Ari Sujarwo, S.Kom., M.I.T

Anggota 2

Mengetahui,

Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia



(Dr. Drs. Inam Djati Widodo, M.Eng.Sc.)

**LEMBAR PERNYATAAN KEASLIAN
TUGAS AKHIR**

Saya yang bertanda tangan dibawah ini,

Nama : Dedek

No. Mahasiswa: 09523328

Menyatakan bahwa seluruh komponen dan isi dalam laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana semestinya.

Yogyakarta, 30 Agustus 2016



HALAMAN PERSEMBAHAN

Puji syukur penulis panjatkan kehadirat Allah SWT, sang pencipta dan sang pengatur langit dan bumi beserta isinya, hidayah, serta kasih sayang-Nya kepada penulis sehingga penulis dapat menyelesaikan tugas akhir ini. Tak lupa pula shalawat dan salam penulis panjatkan kepada Rasulullah muhammad SAW yang telah diutus ke bumi sebagai lentara bagi hati manusia, Nabi yang telah membawa manusia dari zaman kebodohan menuju zaman yang penuh dengan pengetahuan yang luar biasa seperti saat ini.

Dengan mengucapkan syukur Alhamdulillah, kupersembahkan karya Tugas Akhirku ini kepada orang-orang yang kusayangi:

Ibuku: Salmah

Baakku: Dahniar

Abangku:

Sahril dan Sahrizal

Kakakku:

Dahlina, Mardiah, Meisitoh

Adikku:

Dedy

Terima Kasih untuk semua dukungan, semangat, do'a, cinta dan kasih sayangnya.

HALAMAN MOTTO

“Gunakan kehidupan yang singkat di bumi ini dengan menggunakan pedoman Al-Quran dan juga sunnah Rosulullah S.A.W, Jadilah bermanfa’at bagi banyak orang”



KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT, karena berkat rahmat dan karunia-Nyasehingga penulis dapat menyelesaikan karya ilmiah yang berjudul: “proses pengiriman paket data pada jaringanpada software *network simulator* NS-3” ini.Penyusunan karya ilmiah ini merupakan salah satu syarat untuk memperoleh gelar Sarjana dalam program studi Teknik Informatika Universitas Islam Indonesia Yogyakarta.Dalam penyusunan karya ilmiah ini, berbagai pihak telah banyak memberikandorongan, bantuan serta masukan sehingga dalam kesempatan ini penulismenyampaikan terima kasih yang sebesar-besarnya kepada: Ketua Program Studi, Yudi Prayudi, S.Si., M.Kom., selakupembimbing utama yang telah banyak memberikan pengetahuan, bimbingan dan doarestunya.

1. Dr. Ir. Harsoyo, Msc. Selaku Rektor Universitas Islam Indonesia.
2. Dr. Drs. Imam Djati Widodo, M.Eng.Sc. Selaku Dekan Fakultas Teknologi Industri.
3. Hendrik, ST,. M.Eng Selaku Ketua Jurusan informatika.

Kepada semua pihak yang membantu terlaksananya karya ilmiah ini, terima kasih atas dukungan dan doanya selama ini.Penulis menyadari bahwa penulisan karya ilmiah ini masih banyak kekurangan, olehkarenanya kritik dan saran sangat penulis harapkan guna menyempurnakan penulisanini.

Akhir kata penulis mengucapkan banyak terima kasih dan semoga karya ilmiah inidapat berguna bagi kita semua.

Yogyakarta, 30 Agustus 2016

Penulis

ABSTRAK

Percepatan dalam pengembangan dunia teknologi dan informasi semakin berkembang pesat secara signifikan, hal ini semakin linear dengan semakin bertumbuhnya kebutuhan akan infrastruktur jaringan yang menjadi tulang punggung bagi proses pertanggung jawaban atas kebutuhan informasi dan teknologi dewasa ini. Salah satu bentuk perkembangan dunia teknologi dan informasi dewasa ini salah satunya ditunjukkan dengan banyaknya perkembangan teknologi jaringan yang memberikan kelebihan serta kekurangannya untuk implementasi komunikasi data. Tujuan dari penelitian ini adalah untuk menggambarkan proses pengiriman paket data pada jaringan menggunakan software *network simulator* NS-3. Metode yang digunakan dalam penelitian ini terdiri dari beberapa tahap, diantaranya adalah studi literature yang diwujudkan dengan membaca buku, jurnal, artikel, yang berhubungan dengan NS-3, kemudian perancangan sistem serta implementasi dan pemrograman modul-modul. Pemilihan simulator cukup penting dan cukup sulit karena simulator banyak tersedia dan tidak semua simulator dapat digunakan dengan mudah untuk mengimplementasikan model terbaru jaringan komputer. Implementasi model teknologi jaringan terbaru yang merupakan modifikasi dari konsep atau *library* yang telah ada lebih cocok menggunakan simulator NS3. Namun NS3 cenderung akan menyulitkan bilamana digunakan untuk implementasi model jaringan terbaru yang belum memiliki komponen untuk dipergunakan direkomendasikan menggunakan selain NS3, walaupun untuk pengguna tingkat advance NS3 merupakan tool simulasi yang sangat bisa dicustomize dengan kemampuan yang powerful dan fungsional.

Kata kunci: Paket Data, Simulasi NS-3

TAKRIR

Network Jaringan

User Pengguna

Aplication Aplikasi

Channel Saluran/Kanal

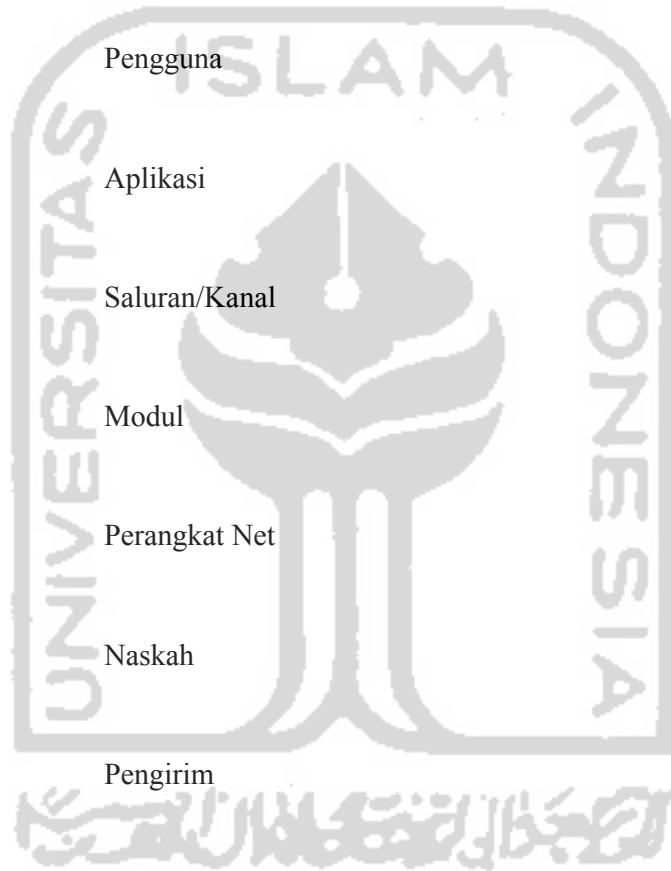
Module Modul

Net Device Perangkat Net

Script Naskah

Sender Pengirim

Resiver Penerima

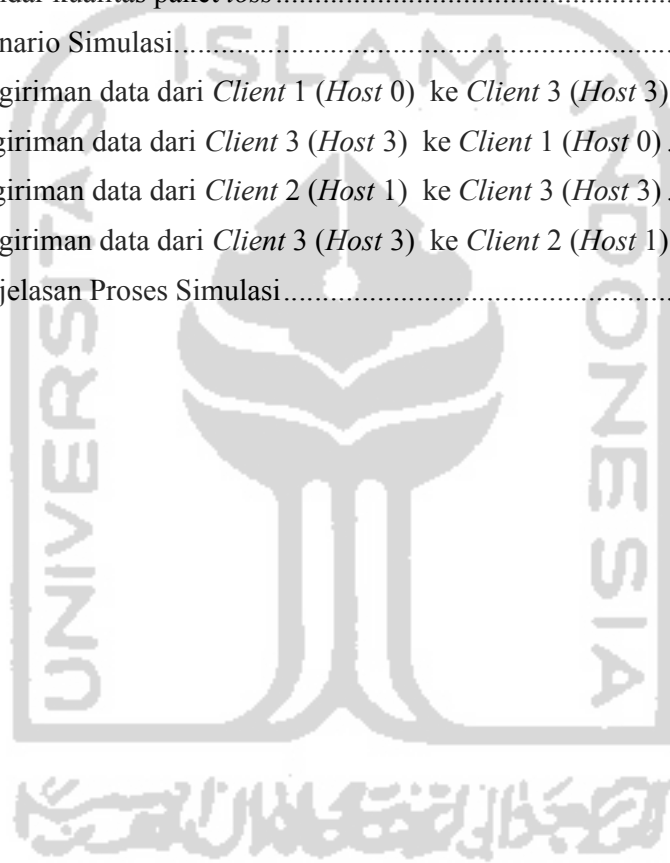


DAFTAR ISI

HALAMAN JUDUL	i	
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii	
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii	
LEMBAR PERNYATAAN KEASLIAN.....	iv	
HALAMAN PERSEMBAHAN.....	v	
HALAMAN MOTTO.....	vi	
KATA PENGANTAR.....	vii	
ABSTRAK.....	viii	
TAKRIR.....	ix	
DAFTAR ISI.....	x	
BAB I PENDAHULUAN		
1.1 Latar Belakang.....	1	
1.2 Rumusan Masalah.....	3	
1.3 Batasan Masalah.....	4	
1.4 Tujuan Penelitian.....	4	
1.5 Manfaat Penelitian.....	4	
1.6 Metode Penelitian.....	5	
1.7 Sistematika Penelitian.....	5	
BAB II LANDASAN TEORI.....		7
2.1. Penelitian Terdahulu.....	7	
2.2. Network Simulator.....	9	
2.3. Dasar Simulasi NS-3.....	18	
2.4. Parameter Pengiriman Paket.....	20	
BAB III METODOLOGI DAN PERANCANGAN.....		22
3.1 Garis Besar Penelitian.....	22	
3.2 Tools.....	23	
3.3 Rancangan Simulasi.....	25	
3.4 Rancangan Analisa Data.....	26	
BAB IV HASIL DAN PEMBAHASAN.....		28
4.1 Persiapan Sistem.....	28	
4.2 Simulasi Pengiriman Paket Data.....	31	
4.3 Hasil Simulasi.....	34	
4.4 Konfigurasi Pembuatan Topologi.....	43	
4.5 Karakteristik Pengiriman.....	44	
4.6 Pemahaman Proses.....	45	
BAB V PENUTUP		
5.1 Kesimpulan.....	47	
5.2 Saran.....	48	
DAFTAR PUSTAKA.....	49	
LAMPIRAN.....	50	

DAFTAR TABEL

Tabel 2.1 Rangkuman beberapa peneliti.....	9
Tabel 2.2 Standar kualitas <i>delay</i>	20
Tabel 2.3 Standar kualitas <i>jitter</i>	20
Tabel 2.4 Standar kualitas paket <i>loss</i>	21
Tabel 4.1 Skenario Simulasi.....	32
Tabel 4.2 Pengiriman data dari <i>Client 1 (Host 0)</i> ke <i>Client 3 (Host 3)</i>	38
Tabel 4.3 Pengiriman data dari <i>Client 3 (Host 3)</i> ke <i>Client 1 (Host 0)</i>	38
Tabel 4.4 Pengiriman data dari <i>Client 2 (Host 1)</i> ke <i>Client 3 (Host 3)</i>	38
Tabel 4.5 Pengiriman data dari <i>Client 3 (Host 3)</i> ke <i>Client 2 (Host 1)</i>	39
Tabel 4.6 Penjelasan Proses Simulasi.....	46



DAFTAR GAMBAR

Gambar 1.1 Ilustrasi Permasalahan Penelitian	3
Gambar 2.1 Prosedur Simulasi Ns-3.....	12
Gambar 2.2 Simple model topologi.....	18
Gambar 2.3 Rumus <i>Throughput</i>	19
Gambar 2.4 Rumus <i>Delay</i>	20
Gambar 2.5 Rumus <i>Jitter</i>	20
Gambar 2.6 Rumus <i>Packet loss</i>	21
Gambar 3.1 Ilustrasi Langkah Penelitian.....	22
Gambar 3.2 Topologi Skenario.....	25
Gambar 4.1 Persiapan Sistem.....	28
Gambar 4.2 Direktori Ns-3.....	29
Gambar 4.3 <i>Distclean</i> Ns-3	29
Gambar 4.4 Output dari eksekusi <i>simulator</i>	30
Gambar 4.5 Proses eksekusi test <i>simulator</i>	31
Gambar 4.6 <i>Source Code</i> program	35
Gambar 4.7 <i>Output</i> dari Alur Data dan <i>Interface</i> Sebuah <i>Router</i>	35
Gambar 4.8 Contoh <i>Output</i> dari pengiriman data	36
Gambar 4.9 <i>Output</i> Nilai Parameter pengiriman dari <i>Client</i> 1 ke <i>Client</i> 3... 37	37
Gambar 4.10 <i>Output</i> Nilai Parameter dari <i>Client</i> 3 ke <i>Client</i> 1	37
Gambar 4.11 Grafik Perubahan Nilai Parameter <i>Throughput</i>	40
Gambar 4.12 Grafik Perubahan Nilai Parameter <i>Delay</i>	41
Gambar 4.13 Grafik Perubahan Nilai Parameter <i>Jitter</i>	42
Gambar 4.14 Grafik Perubahan Nilai Parameter <i>Packet loss</i>	43

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan pesat dari komputer dan teknologi informasi tidak lepas dari perkembangan teknologi jaringan komputer. Dalam hal ini, jaringan komputer adalah sekumpulan komputer dan perangkat jaringan lainnya yang saling berhubungan satu sama lain sesuai dengan bentuk topologi yang dipilih (Sukmaaji & Rianto, 2008). Jaringan komputer dibangun untuk membawa informasi secara tepat tanpa adanya kesalahan dari sisi pengirim (*transmitter*) maupun sisi penerima (*receiver*) melalui media komunikasi.

Pengiriman dan penerimaan paket data dalam sebuah jaringan adalah merupakan salah satu aspek penting dari teknologi jaringan. Pada prinsipnya hal yang terkait dengan mekanisme dan proses pengiriman paket data dari satu perangkat ke perangkat lainnya tidak akan terlihat serta tidak dapat diamati dengan cermat. Teknologi jaringan telah mampu menjalankan proses pengiriman dan penerimaan paket data sedemikian cepatnya. Padahal sebenarnya, apabila dilakukan pengamatan dengan baik, maka sebuah paket data yang terkirim dari sebuah perangkat komputer kepada perangkat komputer lainnya dalam sebuah jaringan dapat ditelusuri prosesnya. Pada prinsipnya sebuah data yang dikirim dari suatu perangkat akan mengalami serangkaian proses sebelum mencapai tujuannya atau perangkat penerimanya, baik yang berbentuk komputer ataupun perangkat jaringan lainnya.

Memahami proses pengiriman paket data dalam sebuah system jaringan komputer memerlukan pendekatan dari berbagai sudut pandang. Memahami sistem layer protocol dalam konsep 7 layer OSI, arsitektur jaringan TCP/IP, proses enkapsulasi dan dekapulasi adalah beberapa hal yang dipahami terlebih dahulu sebelum memahami proses pengiriman paket data. Sayangnya penjelasan tentang paket data dengan pendekatan diatas termasuk materi yang sulit untuk dipahami. Hal ini sejalan dengan pendapat dari (Irawan, 2011) yang menyimpulkan bahwa materi dan penjelasan tentang paket data termasuk materi yang sulit untuk dipahami bagi mereka yang sedang mempelajari materi jaringan

computer. Hal ini sejalan juga dengan uraian dari (Sulistyo, 2009), tentang modul-modul paket data yang mengatakan bahwa penjelasan tentang materi paket data adalah sesuatu yang cukup kompleks untuk dipahami.

Untuk itu, maka perlu adanya cara lain yang dilakukan untuk memahami proses pengiriman paket data, salah satunya adalah melalui pendekatan simulasi. Pendekatan simulasi ini selain untuk memahami proses pengiriman paket data, juga dapat digunakan untuk memahami berbagai aspek lainnya dari mekanisme pengiriman paket data. Pendekatan simulasi juga akan memberikan sudut pandang lain dari pemahaman pengiriman paket data melalui kemampuan visualisasi dan *capturing data*. Sayangnya penggunaan sejumlah *tools* simulasi untuk membantu menjelaskan proses pengiriman paket data tersebut belum banyak dilakukan.

Beberapa peneliti telah mencoba untuk menjelaskan proses pengiriman paket data melalui pendekatan simulasi. Antara lain adalah (Yousefi, 2015) yang melakukan simulasi pengiriman paket sensor node menggunakan NS-2. Peneliti lainnya adalah (Alborz, Keyvani, Nikolic, & Trajkovic, 2000) melalui simulasi pengiriman paket data menggunakan Opnet. Untuk melengkapi penelitian yang pernah dilakukan tersebut, maka pada penelitian tugas akhir ini akan dilakukan simulasi pengiriman paket data dengan simulator NS-3.

Khusus terkait dengan simulasi, menurut (Khan, Bilal, & Othman, 2012) untuk proses pengiriman paket data, terdapat banyak *tools* yang dapat digunakan untuk kepentingan itu, diantaranya adalah: ns-2, ns-3, OMNET++, SWAN, OPNET, Jist, dan GloMoSiM. Dalam penelitian tentang komparasi beberapa *tools open source* untuk kepentingan simulasi jaringan, maka berdasarkan beberapa parameter seperti penggunaan CPU, *memory*, waktu komputasi dan skalabilitas, maka (Khan et al., 2012) menyimpulkan bahwa NS-3 merupakan *tools* yang paling direkomendasikan untuk kepentingan simulasi jaringan. Hal ini juga sejalan pula dengan kesimpulan dari (Weing, Lehn, & Wehrle, 2009) yang membandingkan kinerja 3 *tools* simulator yaitu ns-3, OMNeT++ dan JiST dengan rekomendasi pada penggunaan NS-3 sebagai *tools* yang disarankan.

Dengan pertimbangan informasi tersebut, maka penggunaan simulator NS-3 untuk mensimulasikan proses pengiriman data pada sebuah jaringan adalah sebuah tantangan penelitian yang perlu untuk didalami. Dalam hal ini, kesulitan dalam hal memahami proses pengiriman paket data kemudian rekomenadasi penggunaan simulasi melalui tools NS-3 akan menjadi solusi untuk memberikan pemahaman tentang bagaimana mekanisme pengiriman paket data tersebut. Melalui simulasi diharapkan dapat diketahui pula beberapa karakteristik lainnya dari proses pengiriman paket data tersebut.

Untuk itu, maka hal yang kemudian menjadi permasalahan dalam penelitian tugas akhir ini adalah bagaimanakah membangun rancangan simulasi pengiriman paket data dengan menggunakan tools NS-3, kemudian berdasarkan simulasi yang telah dibuat, bagaimanakah karakteristik yang didapat dari proses pengiriman paket data tersebut. Melalui dua hal tersebut maka hal-hal terkait dengan pengiriman paket data akan dapat dijelaskan lebih lanjut. Secara umum ilustrasi dari permasalahan dalam penelitian ini adalah sebagaimana terlihat pada Gambar 1



Gambar 1.1 Ilustrasi Permasalahan Penelitian

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang masalah diatas maka rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut:

- a. Untuk mendapatkan pemahaman yang lebih tentang pengiriman paket data, maka salah satu solusinya adalah menggunakan pendekatan simulasi. Salah satu *tools* solusi yang banyak di rekomendasikan adalah menggunakan NS-3, untuk itu maka permasalahan yang ingin diselesaikan

dalam penelitian ini adalah bagaimanakah membangun simulasi untuk pengiriman paket data menggunakan simulator NS-3.

- b. Setelah dilakukan proses simulasi maka akan didapat sejumlah informasi yang dapat menjelaskan karakteristik proses pengiriman paket data. Untuk itu maka permasalahan berikutnya dari penelitian ini adalah mengenali lebih lanjut bagaimanakah karakteristik pengiriman paket yang didapat dari simulasi menggunakan NS-3.

1.3 Batasan Masalah

Dalam menjalankan penelitian ini terdapat beberapa batasan masalah, yaitu:

- a. *Tools* yang digunakan dalam penelitian ini *Network Simulator* NS-3 versi 3.26 yang diunduh dari alamat <https://www.nsnam.org/ns-3-26/>
- b. Karakteristik paket data yang akan diamati adalah pada parameter sebagai *packet loss, jitter, packet delay* dan *throughput*.

1.4 Tujuan Penelitian

Berdasarkan pemaparan rumusan masalah diatas maka tujuan penelitian ini adalah sebagai berikut:

- a. Mendapatkan rancangan simulasi untuk pengiriman paket menggunakan NS-3.
- b. Mengetahui karakteristik pengiriman paket yang didapat dari simulasi menggunakan NS-3.

1.5 Manfaat Penelitian

Manfaat dari penelitian adalah:

- a. Dapat memahami persiapan, perancangan uji coba atau simulasi paket data pada suatu jaringan dengan menggunakan NS3.
- b. Dapat mengetahui beberapa karakteristik dari proses pengiriman paket data yang dapat dijadikan sebagai bahan untuk memberikan penjelasan atau penerangan seputar materi pengiriman paket data.

1.6 Metode Penelitian

Dalam menyelesaikan tugas akhir ini tahapan-tahapan yang akan dilalui diantaranya adalah:

- a. Studi literatur seperti membaca buku, jurnal, artikel, yang berhubungan dengan NS-3.
- b. Perancangan sistem melakukan perancangan topologi jaringan serta skenario pengiriman paket data dari sumber ke tujuan pada topologi yang sudah dirancang sebelumnya.
- c. Persiapan instalasi NS3 dan Implementasi serta pemrograman modul-modul.
- d. Pengujian dan evaluasi
- e. Kesimpulan dan saran yaitu membuat kesimpulan dari seluruh tahapan yang telah dilakukan.

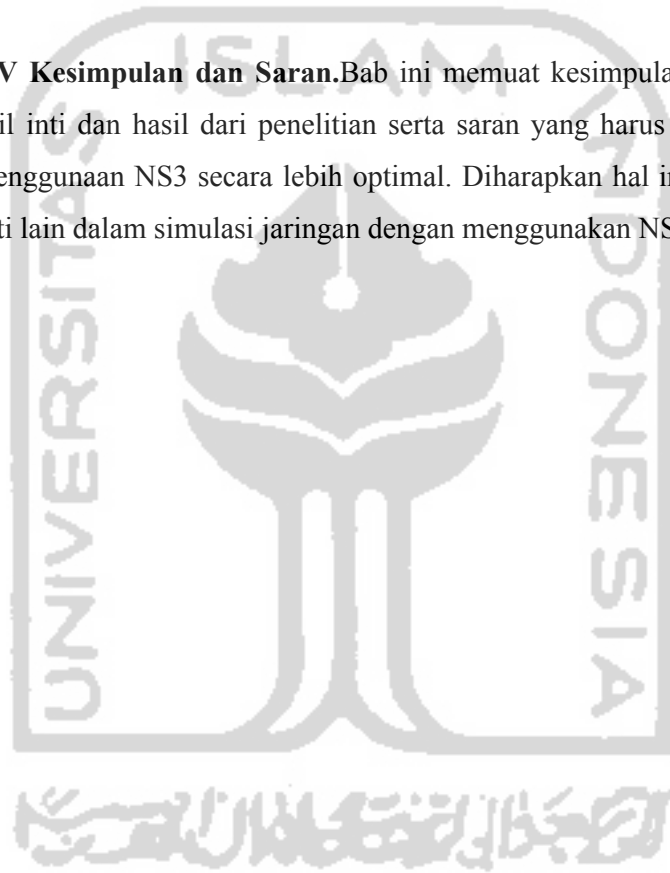
1.7 Sistematika Penulisan

Laporan tugas akhir ini disusun menjadi lima bab. Masing-masing bab dibagi ke dalam beberapa *sub* bab yang merupakan pokok dari bab bersangkutan, sebagai berikut:

- **BAB I Pendahuluan.**Bab ini terdiri dari Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, *Review* Penelitian, Metode Penelitian, dan Sistematika Penulisan Laporan tugas akhir.
- **BAB II Landasan Teori.**Bab ini berisi teori-teori yang berhubungan dengan topik yang akan dibahas dalam melakukan analisis, perancangan, dan implementasi alat simulasi NS3.
- **BAB III Metodologi dan Perancangan.**Bab ini berisikan tahap penggunaan program Python dalam mengeksekusi simulasi serta persiapan serta pengoptimalan modul yang dibutuhkan dalam NS3 agar dapat

menjalankan simulasi jaringan yang diinginkan melalui *platform* simulator NS-3 berdasarkan skenario yang ditentukan.

- **BAB IV Implementasi Hasil dan Analisis.**Bab ini berisikan implementasi penggunaan NS3 mulai dari tahapan persiapan instalasi hingga tahapan akhir berupa eksekusi untuk uji coba berjalannya simulasi NS3.
- **BAB V Kesimpulan dan Saran.**Bab ini memuat kesimpulan yang akan diambil inti dan hasil dari penelitian serta saran yang harus diperhatikan dari penggunaan NS3 secara lebih optimal. Diharapkan hal ini membantu peneliti lain dalam simulasi jaringan dengan menggunakan NS3.



BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian lainnya seputar pengiriman paket data telah dilakukan oleh (Sulistyo, 2009). Dalam penelitiannya tersebut diungkapkan bahwa jaminan pengiriman data melalui jaringan komputer menjadi sesuatu yang sangat penting. Salah satu jaminan yang dibutuhkan oleh user (pemakai) adalah bahwa data yang dikirimkan ke komputer tujuan sampai dengan baik. Selain itu, terdapat fasilitas yang memberikan keterangan terhadap status pengiriman data kepada user terutama bila pengiriman gagal dilakukan. Untuk itu dibutuhkan sebuah mekanisme yang berfungsi untuk melakukan kontrol pada saat proses pengiriman data. Protokol UDP merupakan protokol yang bersifat *connectionless* dan *unreliable* dalam proses pengiriman data. Untuk itu dengan menggunakan metode *command/request* dapat dilakukan perancangan sistem reliabilitas transmisi data pada protokol udp (*user datagram protocol*) (Sulistyo, 2009).

Penelitian berikutnya dilakukan oleh (Yuliansyah, 2016). Dalam penelitiannya ini dilakukan proses perancangan dan prototipe untuk pengiriman data secara wireless menggunakan mikrokontroler arduino uno dan modul ESP8266. Data yang terkirim disimpan di dalam database komputer server. Sementara itu (Azmi, 2014) yang mengungkapkan bahwa dalam teknik komunikasi data harus mencakup kemampuan untuk mendeteksi jika terjadi kesalahan. Teknik komunikasi digital memungkinkan adanya deteksi kesalahan dan koreksi kesalahan. Berdasarkan banyak kesalahan bit, maka error dapat dibedakan atas kesalahan 1 bit kesalahan ledakan. Kesalahan bit juga dapat ditunjukkan oleh jarak Hamming antara pesan yang terdapat pada sisi pengirim dan pada sisi penerima.

Berikut merupakan penelitian-penelitian terdahulu yang hampir sama dengan penelitian yang akan peneliti lakukan, diantaranya adalah sebagai berikut:

1. Penelitian oleh (Irawan, 2011) yang menemukan bahwa Jaringan MANET bisa disimulasikan menggunakan NS-3 terutama karena dukungan C++ yang memberikan kemudahan untuk menspesifikasikan protokol jaringan MANET dengan lebih detail.
2. Penelitian yang kedua oleh (Khasanah, 2014) hasil penelitian yang didapatkan dari perancangan sistem berupa adanya hotspot login, meningkatnya kecepatan akses internet dengan efisiensi waktu yang diperoleh 36 detik, serta mengoptimalkan kinerja jaringan dengan mengalokasikan alamat IP menggunakan metode subnetting. Dalam penelitian dilakukan kajian kinerja jaringan dengan melihat parameter Quality of Service (QoS) yaitu delay, packet loss, dan throughput.
3. Penelitian selanjutnya dari (Chaudhary, 2012) yang kemudian mendapatkan hasil bahwa NS-3 adalah sebuah proyek open source yang aktif dan model pengembangan sumber terbuka, beberapa simulator fitur yang dirancang untuk membantu penelitian Internet saat ini, pembangunan berbasis masyarakat dan pemeliharaan model, berusaha untuk menghindari beberapa masalah dengan ns-2, seperti sebagai interoperabilitas dan kopling antara model-model, kekurangan manajemen memori, debugging split bahasa objek.

Dalam tugas akhir ini, penelitian yang akan dilakukan adalah melakukan simulasi dengan menggunakan alat simulasi NS3 dengan fokus percobaan pengiriman paket data. Dari sejumlah penelitian yang telah dilakukan oleh (Irawan, 2011), (Khasanah, 2014) dan (Chaudhary, 2012), terlihat bahwa penelitian tentang pengiriman paket data yang dilakukan sifatnya adalah UDP (*user datagram protocol*). Hal ini dikarenakan UDP merupakan protokol transport yang dirancang bukan untuk difokuskan pada masalah reliabilitas transportasi data, sehingga tidak membutuhkan buffer lokal untuk menyimpan data yang dikirimkan maupun yang dipesan. Sementara hal terkait dengan simulasi

pengiriman paket data masih belum dilakukan. Sehingga penelitian yang akan dilakukan dalam tugas akhir ini akan mencoba untuk melakukan penelitian yang berbeda dalam peneliti sebelumnya dalam hal *simulator* pengiriman paket data.

Tabel 2.1. Rangkuman beberapa penelitian sejenis.

No	Peneliti	Fokus Simulasi	Teknik	Parameter Simulasi
1	(Irawan, 2011)	MANET	Pengamatan <i>Throughput</i>	Perhitungan <i>Throughput</i>
2	(Khasanah, 2014)	HOTSPOT	Hotspot Login	QoS
3	(Chaudhary, 2012)	NS-3 dan NS-2	Ns-2	Sebagai <i>interoperabilitas</i>
4	(Prawira, 2015)	MPLS	Routing Open Shortest Path First(OSPF)	QoS
	TUGAS AKHIR	Paket Data	Rekayasa trafik (<i>traffic engineering</i> , TE)	Proses pemilihan saluran data traffic

Berdasarkan rangkuman penelitian pada Tabel 2.1 tersebut, maka yang membedakan dari penelitian yang akan dilakukan pada tugas akhir ini dengan penelitian lainnya pada referensi tersebut adalah pada focus Pengiriman paket bersifat dengan menggunakan teknik Rekayasa trafik serta penggunaan parameter Proses pemilihan saluran data *traffic*.

2.2 Network Simulator (NS3)

Simulator NS-3 adalah sebuah network simulator peristiwa yang memiliki ciri tersendiri yang ditargetkan secara utama untuk tujuan riset dan pendidikan. Proyek NS-3, dimulai pada tahun 2006, adalah sebuah proyek open source yang diatur oleh komunitas peneliti dan pengembang. NS-3 bukan extension dari NS-2, melainkan sebuah simulator yang baru. Kedua simulator ditulis menggunakan bahasa pemrograman C++, tetapi NS-3 tidak menyokong API milik NS-2. NS-3 membolehkan peneliti untuk mempelajari protokol-protokol

Internet dan sistem berskala besar dalam lingkungan yang terkontrol. NS-3 merupakan sebuah simulator jaringan yang sering digunakan untuk simulasi protokol routing diantara yang simulator lainnya, dan juga sering digunakan untuk riset mengenai ad-hoc networking, dan mendukung protokol jaringan yang populer, serta menyediakan hasil simulasi untuk jaringan kabel maupun nirkabel. NS-3 juga cukup populer di kalangan peneliti karena berbasis open source serta menyediakan dokumentasi penelitian dari penelitian sebelumnya secara online pada website pengembang NS-3.

Menurut (Irawan, 2011) NS-3 merupakan simulasi network yang memungkinkan network programmer untuk membuat simulasi dari jaringan yang akan dibangun dan saling berbagi code dari simulasi dan implementasi dari protokol yang dijalankan. NS-3 bahkan mampu memproses paket dalam ukuran besar. NS-3 adalah program simulasi dari sebuah jaringan *open source*, yang umumnya ditujukan untuk penggunaan percobaan peneliti, keperluan pendidikan dan praktisi. NS-3 dibuat dengan menggunakan bahasa pemrograman C++ dan juga menggunakan script python. Simulator NS-3 adalah sebuah network simulator peristiwa yang memiliki ciri tersendiri yang ditargetkan secara utama untuk tujuan riset dan pendidikan. Proyek NS-3, dimulai pada tahun 2006, adalah sebuah proyek open source yang diaturoleh komunitas peneliti dan pengembang (Ramadhan, 2011). Pada software simulasi jaringan ini, NS-3 mempunyai beberapa fitur yang dapat dimanfaatkan untuk memodelkan dan menguji berbagai model jaringan.

Dengan NS-3, proses pengiriman paket data disimulasikan dengan membuat dua skenario. Pembuatan topologi, node dan protokol yang digunakan untuk proses pengiriman paket data sudah didukung oleh NS-3. Dengan NS-3 maka dapat menambahkan fungsi-fungsi baru didalam core NS-3 karena NS-3 bersifat open source. NS-3 dikembangkan menggunakan bahasa C++ di lapisan inti dan script python. Fitur-fitur NS-3 di antaranya adalah sistem atribut NS-3 terdokumentasi dengan baik. Setiap objek NS-3 memiliki seperangkat atribut (*name, type, initial value*) dan NS-3 selaras dengan sistem nyata. Model node yang lebih seperti komputer nyata, dukungan utama antarmuka seperti socket API dan IP

atau perangkat driver antarmuka (di Linux) serta NS-3 juga telah meng-update model-model (memuat campuran model baru dan ported model). NS-3 juga terintegrasi dengan *software/tools*lain seperti *wireshark* untuk melihat *trace output*. Representasi hasil data simulasi pada NS-3 dapat ditampilkan dalam bentuk grafik, sehingga memudahkan untuk menganalisa dan mengevaluasi hasil terhadap suatu model jaringan WIFI, CDMA, GSM, MPLS dan lainnya.

Sebagai alat simulasi jaringan, NS-3 mempunyai model-model untuk semua elemen jaringan yang terdapat pada jaringan real. Element-element jaringan tersebut adalah:

1. *Node*

Dalam jargon internet, perangkat komputer yang terhubung ke jaringan disebut host atau terkadang end-system. Dalam NS-3 abstraksi perangkat komputasi dasar atau komputer disebut node. Abstraksi ini diwakili dalam C++ oleh kelas node. Kelas node menyediakan metode untuk mengelola representasi perangkat komputasi di simulasi. Kelas node menyediakan metode untuk mengelola representasi perangkat komputasi di simulasi.

2. Aplikasi

Dalam NS-3 abstraksi dasar untuk program pengguna yang menghasilkan beberapa kegiatan yang akan disimulasikan adalah aplikasi. Abstraksi ini diwakili dalam C++ oleh kelas Application. Kelas Application menyediakan metode untuk mengelola representasi versi NS-3 pada aplikasi-aplikasi level user dalam simulasi. Pengembang diharapkan untuk mengkhususkan kelas Application dalam pengertian pemrograman berorientasi obyek untuk membuat aplikasi baru.

3. *Channel*

Media dimana aliran data dalam jaringan mengalir disebut channel. Dalam dunia simulasi NS-3, seseorang menghubungkan sebuah node ke objek yang mewakili sebuah saluran komunikasi. Di NS-3 abstraksi komunikasi dasar *subnetwork* disebut channel dan diwakili di C++ oleh kelas channel.

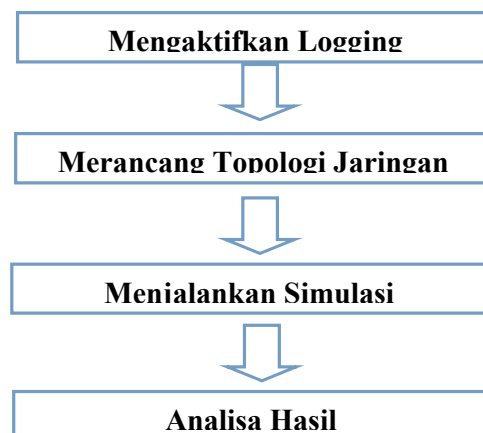
4. *Net Device*

Untuk terhubung dengan jaringan, komputer harus memiliki perangkat keras yang disebut dengan peripheral card. Peripheral card tersebut diimplementasikan beberapa fungsi jaringan, sehingga disebut *NetworkInterface Cards* (NICs). NIC tidak akan berfungsi tanpa sebuah software driver untuk mengontrol perangkat keras tersebut. Pada Unix (atau Linux), sebuah peripheral hardware disebut sebagai device. Device dikontrol menggunakan device driver, dan NIC dikontrol menggunakan network device driver yang disebut dengan net device. Di NS-3, netdevice meliputi baik software driver dan simulasi hardware. Sebuah net device diinstalasi pada sebuah node agar memungkinkan node untuk berkomunikasi dengan node lainnya dengan simulasi melalui channels. Abstraksi netdevice direpresentasikan dengan C++ oleh kelas NetDevice. Kelas NetDevice menyediakan metode untuk mengatur koneksi ke objek node dan channel.

5. *Topology Helpers.*

Dalam sebuah jaringan simulasi besar akan diperlukan banyak koneksi untuk mengatur antaranode, NetDevice serta channel. NS-3 menyediakan apa yang disebut objek Topology Helpers untuk mengatur simulasi-simulasi jaringan semudah mungkin.

Simulasi NS-3 mempunyai 3 prosedur yaitu: Mengaktifkan Logging, merancang topologi jaringan, menjalankan simulasi serta melakukan analisa hasil. Gambar 2.1 menunjukkan alur dasar prosedur simulasi jaringan dengan menggunakan NS-3. Berikut ini deskripsi dari masing-masing tahapnya. Untuk dapat menggunakan NS-3, hal yang harus dilakukan adalah mengunduh dan memasang NS-3.26 ke dalam sistem komputer.



Gambar 2.1Prosedur Simulasi NS-3

Berdasarkan prosedur simulasi pada Gambar 2.1, maka dapat dijelaskan prosedur lengkap simulasi dengan NS-3 sebagai berikut ini :

- 1) Mengaktifkan logging. Langkah ini bertujuan untuk mengaktifkan fitur logging yang ada di NS-3 yang dapat ditampilkan pada konsol / terminal Ubuntu dan juga dapat dicatat pada logging file packet capture yang dapat dibaca oleh aplikasi Wireshark untuk melakukan trace pada paket yang mengalir dalam simulasi yang dibuat.
- 2) Merancang topologi jaringan. Perancangan topologi difungsikan serta dimaksudkan untuk memperjelas alur dari proses pengiriman paket data yang ada. Sehingga kemudian topologi ini kemudian menjadi bahan untuk dimasukkan dalam simulasi pengiriman paket data pada NS3.

Untuk mengaktifkan modul-modul yang ada di NS-3 terdapat 5 tahapan sebagai berikut:

- a. Pembuatan *Node*. Node pada NS-3.26 merupakan abstraksi dari end-system atau lebih sering dikenal dengan host pada suatu jaringan komputer. Abstraksi ini diwakili dalam C++ oleh kelas node yang menyediakan metode untuk mengelola representasi perangkat komputasi di simulasi. Node merepresentasikan komputer yang akan ditambahkan sesuatu seperti protokol, aplikasi, dan peripheral card. Topology helper *NodeContainer* menyediakan sebuah cara yang mudah untuk membuat, mengontrol, dan mengakses Node apapun yang telah dibuat untuk dapat menjalankan sebuah simulasi. Langkah selanjutnya dalam pembuatan sebuah topologi jaringan adalah dengan menghubungkan node yang telah dibuat pada sebuah jaringan. Bentuk

paling sederhana dari sebuah jaringan yang dapat dibuat adalah sebuah jalur point-to-point antara dua node.

- b. Penentuan Topology Helper. Dalam sebuah jaringan simulasi besar akan diperlukan banyak koneksi untuk mengatur node, Net Device serta channel. NS-3 Dev menyediakan apa yang disebut objek topology helpers untuk mengatur simulasi–simulasi jaringan semudah mungkin. Dalam membuat topologi point-to-point, *topologyhelperPointToPointHelper* diperlukan untuk membuat jalur untuk menghubungkan kedua node. Istilah yang akan digunakan dalam hal ini adalah Net Device dan Channel. Dalam dunia nyata, istilah tersebut sesuai dengan peripheral card dan kabel jaringan. Umumnya, kedua hal ini berhubungan erat dan tidak dapat ditukarkan satu sama lain, sebagai contoh menggunakan perangkat Ethernet tetapi dengan channel wireless. Oleh karena itu, pada pembuatan jalur point-to-point akan menggunakan *PointToPointHelper* untuk mengkonfigurasi dan menghubungkan objek NS-3 *PointToPointNetDevice* dan *pointToPointChannel* yang ditunjukkan pada kode di bawah ini.

```
PointToPointHelper pointToPoint;
pointToPoint.setDeviceAttribute
    ("DataRate", StringValue
        ("5Mbps"));
pointToPoint.setChannelAttribute ("Delay",
    StringValue
        ("2ms"));
```

Pada baris pertama diinisialisasikan objek *PointToPointHelper*. Lalu pada baris berikutnya kode di atas memberitahu objek *PointToPointHelper* untuk menggunakan nilai “5Mbps” (lima megabit per detik) sebagai “*DataRate*” ketika membuat sebuah objek *PointToPointNetDevice*. Kata “*DataRate*” merupakan atribut dari *PointToPointNetDevice*. Lalu pada baris ketiga, *PointToPointHelper*

diminta untuk memakai nilai “2ms” (dua mili detik) sebagai besarnya *delay* transmisi dari setiap channel *point-to-point* yang dibuat.

- c. Pemasangan *Net Device*. Untuk terhubung dengan jaringan, komputer harus memiliki perangkat keras yang di sebut dengan peripheral card. Peripheral card tersebut di implementasikan beberapa fungsi jaringan, sehingga disebut *Network InterfaceCards* (NICs). NIC tidak akan berfungsi tanpa sebuah software driver untuk mengontrol perangkat keras tersebut. Pada Unix (atau Linux), sebuah peripheral hardware disebut sebagai device. Device dikontrol menggunakan *device driver*, dan NIC dikontrol menggunakan network device driver yang disebut dengan netdevice. Di NS-3 Dev, net Device meliputi baik software driver dan simulasi hardware. Sebuah *net device* di instalasi pada sebuah node agar memungkinkan node untuk berkomunikasi dengan node lainnya dengan simulasi melalui channels. Abstraksi net device di representasikan dengan C++ oleh kelas *NetDevice*. Kelas *NetDevice* menyediakan metode untuk mengatur koneksi keobjek node dan channel. Untuk menampung objek *NetDevice* yang akan dibuat, digunakanlah *NetDeviceContainer*, seperti halnya pada *NodeContainer* untuk menampung node yang telah dibuat. Metode install pada *PointToPointHelper* memiliki parameter yaitu *NodeContainer*. *PointToPointNetDevice* akan terbentuk dan tersimpan pada *NetDeviceContainer* sebanyak jumlah node yang ada dalam *NodeContainer*, dalam hal ini sebanyak dua buah karena untuk topologi point-to-point. Sebuah *PointToPointChannel* telah terbentuk dan dua buah *PointToPointNetDevice* telah terpasang. Kedua device akan terkonfigurasi untuk mengirimkan data dengan kecepatan sebesar lima megabit per detik melalui channel yang telah terbentuk yang memiliki delay sebesar dua milidetik.
- d. Pemasangan Protocol Stack. Setelah node dan device terkonfigurasi dengan baik, langkah berikutnya yaitu memasang protokol pada node

yang telah dibuat. `Internet StackHelper` adalah sebuah topology helper yang berfungsi untuk memasang protokol Internet pada point-to-point netdevice. Metode `Install` memiliki parameter yakni `NodeContainer`. Ketika di jalankan, akan diinstall stack Internet seperti TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*), IP (*Internet Protocol*), dan sebagainya pada setiap node yang ada dalam `NodeContainer`.

- e. Penentuan alamat IP. Berikutnya node yang telah dibuat akan dipasangkan alamat IP untuk dapat berkomunikasi antar node satu sama lain. Topology helper `IPv4AddressHelper` ini berfungsi untuk mengatur pengalokasian dari alamat IP. Berikut ini kode yang bertujuan untuk menetapkan alamat IP pada node yang telah dibuat.

```
IPv4AddressHelper address;
address.SetBase ("10.1.1.0",
                "255.255.255.0");
```

Pada baris pertama mendeklarasikan sebuah objek address helper yang memberitahukan agar alamat IP yang dialokasikan adalah dengan jaringan 10.1.1.0 dan menggunakan subnet mask 255.255.255.0. Secara default alamat yang dialokasikan akan dimulai dari satu dan akan bertambah secara statik, maka alamat pertama yang dialokasikan dalam hal ini adalah 10.1.1.1, diikuti dengan 10.1.1.2, dan seterusnya. Sistem NS-3 dapat mengingat seluruh alamat IP yang telah dialokasikan sehingga bila secara tidak sengaja menetapkan alamat IP yang sama maka akan menimbulkan fatal error.

Objek `IPv4Interface` dibutuhkan untuk mengasosiasikan antara alamat IP dengan device. `IPv4InterfaceContainer` di atas berfungsi untuk menampung daftar dari objek `IPv4Interface` sebagai referensi yang mungkin akan dipakai dilain kesempatan. Sekarang, sebuah jaringan

point-to-point sudah terbentuk dengan baik. Yang diperlukan selanjutnya adalah aplikasi untuk menghasilkan aliran data.

3) Menjalankan simulasi.

NS3 (Network Simulator generation 3), Simulator ialah berbasis open source atau kode terbuka dalam pengembangannya yang dikembangkan oleh proyek VINT (*Virtual Inter Network Testbed*). Karena sistem operasi yang digunakan dalam penelitian ini adalah berbasis open source yaitu ubuntu 15.10 LLTS, maka dalam instalasinya NS3 membutuhkan beberapa komponen yang diperlukan, maka komponen yang diperlukan tersebut harus dipilih ulang.

Sedangkan NS3 yang digunakan adalah versi allinone-3.26, yang merupakan gabungan komponen, sehingga tidak perlu menginstal komponen satu demi satu. Hal yang harus dilakukan adalah instalasi patch karena masih terdapat kesalahan pada modul instalasi Network Animator (NAM) dari versi NS3 yang digunakan. Hal ini baru akan diperbaiki pada versi berikutnya yaitu versi 3.26. Bahasa pemrograman yang digunakan oleh NS3 adalah Tcl/Tk, tetapi untuk pengembangan model dilakukan dengan objek dan bahasa pemrograman C++. Di samping itu harus dibuat pula antarmuka untuk menghubungkan model yang dibuat dalam C++ dengan rutin simulasi yang dibuat dalam Tcl/Tk. Pada NS2 sudah terdapat contoh atau library untuk algoritma Flooding yang akan dibandingkan.

Khusus untuk *Wireless Sensor Network* (WSN), pada versi 3.26 sudah terdapat modul tambahan yang mendukung untuk simulasi IEEE 802.15.4, termasuk energy model. NS3 lebih bersifat text based, tetapi sudah dilengkapi dengan fasilitas animasi hasil simulasi. Karena sudah populer, NS3 memiliki banyak pengguna sehingga lebih banyak modul atau komponen yang tersedia. Dari sisi dokumentasi, NS3 memiliki dokumentasi yang lengkap dan sudah ada buku panduan yang diterbitkan selain dari petunjuk pengguna. Hal ini disebabkan NS3 sudah lebih lama dan lebih populer. Hal ini pulalah yang membuat mailing list NS3 lebih

aktif atau memiliki lebih banyak pengguna. Hal yang sebaliknya terjadi untuk dokumentasi komponen atau model yang telah dikembangkan sebelumnya.

Proses selanjutnya adalah menjalankan simulasi dengan mengeksekusi baris perintah yang sudah dibuat dalam listing program atau aplikasi dalam ekstensi `c++` yang sudah dibuat

Untuk me `Simulator::Run ();` bal seperti berikut:

```

Simulator::Run ();
Simulator::Destroy ();

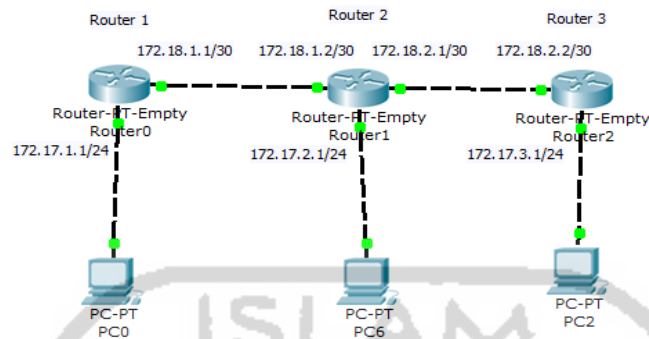
```

4) Analisa Hasil Simulasi

Dari hasil simuasli yang telah dibuat, akan mempersentasikan pengiriman paket data berupa parameter paket *delay*, paket *loss*, dan *throughput*.

2.3 Dasar Simulasi NS3

Simulasi jaringan yang dibuat dengan menggunakan NS-3 mengikuti alur model tersebut. Dimana paket yang dihasilkan oleh Application akan melewati berbagai susunan protokol sebelum dikirimkan melalui Channel yang merupakan sebuah media yang menjadi tempat mengalirnya data dalam suatu jaringan. Dalam NS-3 Dev abstraksi komunikasi dasar subnetwork disebut channel dan diwakili di C++ oleh kelas `channel`, oleh `NetDevice`. `Application`, `Protocol stack`, dan `NetDevice` tersebut terdapat dalam sebuah `Node` yang telah dibuat. Lalu paket tersebut akan dikirimkan ke node yang dituju, yang pertama-tama akan diterima oleh `NetDevice` node yang dituju, kemudian melewati lapisan protokol, dan akan dibaca dan ditampilkan isi dari paket tersebut oleh `application` juga. Channel berfungsi sebagai media perantara yang menjembatani antara node yang satu dengan yang lain. Untuk memperjelas teori diatas, bisa diberikan sebuah contoh pengembangan model simulasi jaringan LAN yang menggunakan protokol CSMA/CD. Topologi jaringan LAN ini bisa diberikan contoh pada Gambar 2.2.



Gambar 2.2 Simple Model Topology

Sumber: Irfani, (2010)

NetDevice : router – router

Channel : Media/ kabel yang digunakan, untuk setting parameter-parameter Datarate dan Delay

Nodes : host – host

Protocol Stack: IP Address Application : UDP atau TCP sebagai format pengiriman paket dan terdapat parameter-parameter seperti MaxPacket, Interval, PacketSize.

Ketika setting IP pada Device Router (Layer 3) dan Nodes maka akan secara otomatis MAC Address terbentuk (Layer 2). Jika semua Atribut terpenuhi maka jaringan akan saling terkoneksi dan menampilkan hasil output timing dari sumber ke tujuan yang dipengaruhi parameter-parameter yang ada di Channel dan Application.

2.4 Parameter Pengiriman Paket

Memasukkan parameter-parameter pengiriman paket data yaitu Throughput, Delay, Jitter dan Packet Loss ke dalam source code topologi NS-3 untuk mendapatkan nilai-nilai dari 4 parameter tersebut. Nilai parameter-parameter yang didapat, dipengaruhi oleh parameter awal yang dimasukkan. Untuk mendapatkan nilai parameter-parameter tersebut, memasukkan rumus-rumus dari ke 4 parameter tersebut berdasarkan jurnal yang didapatkan. Masing-masing

parameter tersebut memiliki rumusnya sendiri dengan parameter dan variable yang berbeda. Berikut rumus-rumus dari ke 4 parameter tersebut (Darmawan & Alif & Basuki, 2013):

Throughput

Berikut rumusan untuk mendapatkan nilai dari parameter *Throughput* :

$$\text{Throughput} = \frac{Pr}{1 \text{ det } ik} \text{ pkt / det } \dots 0 \leq t \leq T$$

Gambar 2.3 Rumus Throughput

Pr = Paket yang diterima (paket)

T = Waktu simulasi (detik)

t = Waktu pengambilan sampel (detik)

Delay

Berikut rumusan untuk mendapatkan nilai dari parameter *Delay* :

$$\text{Delay} = \left(\frac{Tr - Ts}{Pr} \right) \text{ det } ik \dots 0 \leq t \leq T$$

Gambar 2.4 Rumus Delay

Tr = Waktu penerimaan paket (detik)

Ts = Waktu pengiriman paket (detik)

Pr = Paket yang diterima (paket)

T = Waktu simulasi (detik)

t = Waktu pengambilan sampel (detik)

Tabel 2.2 Standar Kualitas untuk Delay

Nilai Delay	Kualitas
0-150 ms	Baik
150-400 ms	Cukup, masih dapat diterima
> 400 ms	Buruk, tidak dapat diterima

Jitter

Berikut rumusan untuk mendapatkan nilai dari parameter *Jitter* :

$$jitter(i) = (R_{i+1} - S_{i+1}) - (R_i - S_i)$$

Gambar 2.5 Rumus Jitter

Terdapat kualitas dari *jitter* yang dapat dikategorikan dalam beberapa kategori berdasarkan ITU-T yaitu sebagai berikut :

Tabel 2.3 Standar Kualitas ITU-T untuk Jitter

Nilai Jitter	Kualitas	Keterangan
0-20 ms	Baik	Ri = <i>Received Time</i> Si = <i>Sent Time</i>
20-50 ms	Cukup	
>50 ms	Buruk	

Packet Loss

Berikut rumusan untuk mendapatkan nilai dari parameter *Packet Loss* :

$$PacketLoss = \left(\frac{Pd}{Ps} \right) \times 100\% \dots 0 \leq t \leq T$$

Gambar 2.6 Rumus Packet Loss

Pd = Paket yang mengalami *drop* (paket)

Ps = Paket yang dikirim (paket)

T = Waktu simulasi (detik)

t = Waktu pengambilan sampel (detik)

Terdapat juga standar kualitas dari *packet loss* yang dapat dikategorikan dalam beberapa kategori berdasarkan ITU-T yaitu sebagai berikut :

Tabel 2.4 Standar Kualitas ITU-T untuk Packet Loss

Packet Loss	Kualitas
0 – 0.5 %	SangatBaik
0.5 – 1.5 %	Baik

BAB III

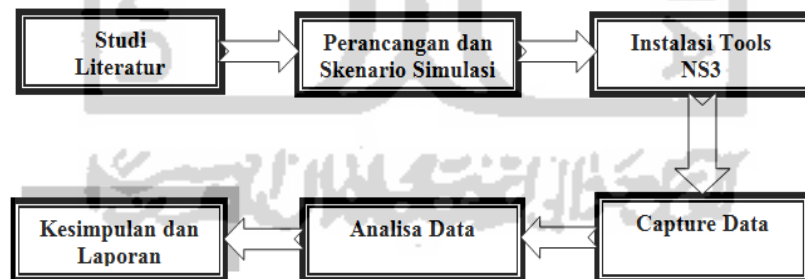
METODOLOGI DAN PERANCANGAN

3.1 Garis Besar Penelitian

Penelitian ini dimaksudkan untuk mendapatkan gambaran tentang kinerja dari proses pengiriman paket menggunakan Simulator Ns-3. Penelitian ini secara tidak langsung akan membantu mereka yang ingin mendalami lebih lanjut tentang jaringan komputer khususnya mengenai proses pengiriman paket data.

Berdasarkan ilustrasi pada (Gambar 1.1), maka untuk mendapatkan pemahaman yang lebih tentang pengiriman paket data, maka salah satu solusinya adalah menggunakan pendekatan simulasi. Salah satu tools solusi yang banyak direkomendasikan adalah menggunakan NS-3, untuk itu maka permasalahan yang ingin diselesaikan dalam penelitian ini adalah bagaimanakah membangun simulasi untuk pengiriman paket data menggunakan NS-3. Permasalahan lainnya adalah untuk mengenali lebih lanjut bagaimanakah karakteristik pengiriman paket yang didapat dari simulasi menggunakan NS-3.

Permasalahan tersebut diatas akan diselesaikan dengan menggunakan urutan metodologi sebagaimana pada Gambar 3.1



Gambar 3.1 Ilustrasi Langkah Penelitian Tuags Akhir Pengiriman Paket Simulator Ns-3

Berdasarkan uraian Gambar 3.1 maka langkah penelitian yang akan dilaksanakan adalah sbb:

1. Studi literatur: Melakukan sejumlah studi literature terkait dengan materi Pengiriman paket menggunakan Simulator Ns-3 dan tools untuk implementansinya.
2. Perancangan dan skenario simulasi: Mengimplementasikan perancangan simulator agar dapat melakukan pengiriman paket data.
3. Instalasi tools Ns-3 bertujuan untuk mengaktifkan modul-modul yang akan digunakan pada penelitian.
4. Capture data. Melakukan pengambilan data yang diperlukan untuk kepentingan monitoring jaringan.
5. Analisis data. Melakukan analisa dari data yang didapat untuk menjawab permasalahan yang muncul dalam penelitian ini.
6. Kesimpulan laporan. Membuat laporan akhir untuk mendokumentasikan semua kegiatan penelitian yang telah dilakukan.

3.2 Tools yang Digunakan

1. Ubuntu 15.10 LTS

Sistem Operasi Ubuntu merupakan salah satu distribusi Linux yang berbasis Debian dan didistribusikan sebagai software bebas. Ubuntu 15.04 LTS merupakan versi yang terbaru dari ubuntu. Sistem operasi ini yang nantinya akan digunakan untuk menjalankan beberapa aplikasi seperti NS3 dan NetAnim yang berfungsi untuk menjalankan simulasi dan membuat animasi model jaringan.

Dalam penelitian ini alasan penggunaan Ubuntudikarenakan Linux memiliki gcc++ compiler dan menjadi platform yang direkomendasikan oleh NS3. Selain itu, terdapat beberapa aspek yang ada pada NS3 bergantung pada platform UNIX khususnya Linux. Contohnya emulation or TapBridge features yang tidak terdapat pada platform Windows.

2. *Sublime Text 3*

Sebuah syntax editor yang menggunakan Python API dikembangkan oleh Jon Skinner seorang programmer dari Australia. Sublime text mempunyai fitur plugin tambahan yang sangat memudahkan. Desain pada Sublime Text yang simpel dan bagus sehingga terkesan elegan untuk sebuah syntax editor. Sublime Text juga memiliki kemudahan dalam auto complete dan auto correct dalam beberapa bahasa pemrograman salah satunya adalah C++.

3. NS-3 (*Network Simulator*)

NS3 adalah aplikasi simulasi jaringan dengan fokus untuk kegiatan penelitian dan pendidikan yang bersifat *Open Source*. NS3 membuat sebuah simulasi yang hampir mirip dengan kejadian nyatanya yang bersifat *realtime* dengan mendukung protokol jaringan yang populer, serta menyediakan hasil simulasi untuk jaringan kabel maupun nirkabel.

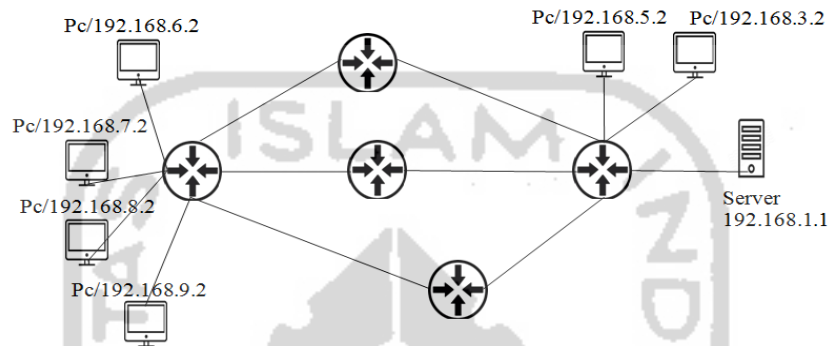
Pada simulasi pengiriman paket ini digunakan NS-3 versi ns-3.26 yang telah stabil pada bulan oktober 2016. NS3 hampir setiap 3 bulan merilis versi baru aplikasinya. Ketika suatu versi di rilis maka dokumentasi dan contoh simulasi yang ada pada Doxygen disesuaikan dengan versi yang dirilis. Sehingga ada kemungkinan contoh simulasi atau dokumentasi pada versi yang lama tidak berlaku atau tidak dapat digunakan pada versi yang baru. Begitu juga sebaliknya untuk contoh simulasi atau dokumentasi pada versi yang baru tidak dapat digunakan pada versi yang lama. Hal ini dikarenakan terdapat beberapa module baru yang diperbaiki, dikembangkan, dan ditambahkan fitur-fiturnya.

4. NetAnim

NetAnim adalah sebuah pembuat animasi offline yang dibuat berdasarkan pada Qt toolkit. NetAnim membuat animasi dalam bentuk XML berdasarkan pada trace file yang dihimpun selama simulasi berlangsung. NetAnim dikemas menjadi satu dengan NS3 sehingga ketika

menginstall NS3 otomatis menginstall NetAnim.NetAnim digunakan untuk melihat simulasi jaringan dalam bentuk animasi.

3.3 Rancangan simulasi pengiriman paket



Gambar 3.2 Topologi Untuk Skenario Simulasi

Pada Skenario simulasi akan dilakukan proses pengiriman paket data yang dibuat dengan mengirim paket data dari satu server pengiriman, dengan IP adalah 192.168.1.1, kemudian digunakan 5 destinasi yang berbeda dan jalur yang berbeda dari 10 total jumlah destinasi. Skenario ini akan menampilkan setiap perjalanan paket dari awal paket dikirim sampai paket sampai ke tujuannya dengan label yang telah ditentukan. Pada skenario ini juga akan ditentukan size pada setiap paket yang dikirim untuk mengetahui pengaruh waktu ketika beberapa paket dikirim secara bersamaan dengan tujuan yang berbeda beda, nantinya disetiap paket yang melewati router paket akan di buktikan melalui terminal aplikasi NS-3.

Pada skenario ini, server yang di sebelah kiri adalah source, dan server tersebut akan mengirim paket ke 6 destinasi dari 9 destinasi yang ada. Keenam destinasi tersebut adalah dengan IP : 192.168.3.2, 192.168.5.2, 192.168.6.2, 192.168.7.2, 192.168.8.2, 192.168.9.2. Kemudian ditetapkan paket yang dikirim sebanyak 4 buah paket. Dari keenam tujuan paket yang dikirimkan, 5 destinasi memiliki paket yang sama yaitu 1024 byte ke alamat berikut ini: 192.168.3.2, 192.168.6.2, 192.168.7.2, 192.168.8.2,

192.168.9.2, sedangkan satu paketnya berisi paket 128 byte yang dikirimkan ke alamat IP 192.168.5.2. Paket akan dikirim dari source ke destinasi dengan interval waktu 0.01 - 2 detik.

Perjalanan paket dari source ke masing-masing destinasi ditentukan jalur yang sudah ditentukan. Paket yang di generate oleh NS-3 memiliki keluaran output berupa file pcap. Dari file inilah bisa dilihat paket - paket yang diterima oleh host-host yang diinginkan. Dari hasil tangkapan wireshark ini bisa lihat berapa paket yang diterima oleh host yang menjadi destinasi, berapa waktu pengirimannya dari source, dan keterangan tentang protokol yang digunakan. Diharapkan dari skenario ini bisa melihat dan menilai bagaimana modul ini bekerja. Hal ini dapat dilihat berdasarkan jumlah perbandingan paket yang dikirim dengan paket yang diterima.

3.4 Rancangan Analisa Data

Pada penelitian ini akan dicatat beberapa hal yang terkait ujicoba untuk memperoleh hasil pengiriman paket data dari skenario yang telah ditentukan.

3.4.1 Persiapan Sistem Pengiriman Paket Data

- Tahap pertama yang akan dilakukan adalah tahap instalasi NS-3 allinone versi 3.26 serta modul-modul yang dibutuhkan. Kemudian ketikkan di konsole perintah hg <http://code.nsnam.org/ns-3-allinone> yang akan otomatis meng-clone kode sumber file ns-3 yang akan dipergunakan untuk mensimulasikan paket data.
- Instalasi librari boos-devel yang akan dipergunakan untuk menjalankan module openflow integration, module openflow ini dibutuhkan agar sistem dapat berjalan dengan optimal.

3.4.2 Persiapan Karakteristik Pengiriman Paket Data

- Paket loss Kehilangan paket ketika terjadi *peak load* dan *congestion* (kemacetan transmisi paket akibat padatnya traffic yang harus dilayani) dalam batas waktu tertentu. Paket loss maksimum yang direkomendasi oleh ITU (*International Telecommunication Union*) adalah 1 %
- Paket Delay. Waktu yang dibutuhkan untuk mengirimkan data dari sumber (pengirim) ke tujuan (penerima). Delay maksimum yang direkomendasikan oleh ITU (*International Telecommunication Union*) untuk aplikasi suara adalah 150 ms, dan yang masih bisa diterima pengguna adalah 250 ms.
- Throughput. Aspek utama throughput yaitu berkisar pada ketersediaan bandwidth yang cukup untuk suatu aplikasi. Hal ini menentukan besarnya trafik yang dapat diperoleh aplikasi saat melewati jaringan. Aspek penting lainnya adalah error (berhubungan dengan link error rate) dan loss (berhubungan dengan kapasitas *buffer*).
- *Jitter*. Merupakan variasi delay yang terjadi akibat adanya selisih waktu atau interval antar kedatangan paket di penerima. *Jitter* maksimum yang direkomendasi oleh ITU adalah 75 ms.

BAB IV

HASIL DAN PEMBAHASAN

Pada Bab IV ini akan dijelaskan mengenai hasil dari skenario yang telah ditentukan pada Bab III sebelumnya. Hasil yang didapat akan dianalisis dan dibahas sehingga diharapkan menghasilkan suatu jawaban atas masalah yang ada pada Bab I, selanjutnya dapat digunakan sebagai acuan dalam menentukan kesimpulan dan rencana kedepannya.

Pada penelitian tugas akhir ini, simulasi NS-3 yang dijalankan menggunakan NS-3 versi development yaitu versi 3.26 serta berjalan pada sistem operasi Linux Ubuntu versi kernel 15.10 LTS 32-bit.

4.1 Persiapan Sistem

Tahapan pertama yang akan dilakukan adalah tahap instalasi NS-3 allinone versi 3.26 serta module. Setelah jendela aplikasi konsole terbuka maka akan mengetikkan perintah `hg clone http://code.nsnam.org/ns-3-allinone` yang akan mengclone kode sumber file ns-3 yang akan dipergunakan untuk melakukan simulasi pengiriman paket.

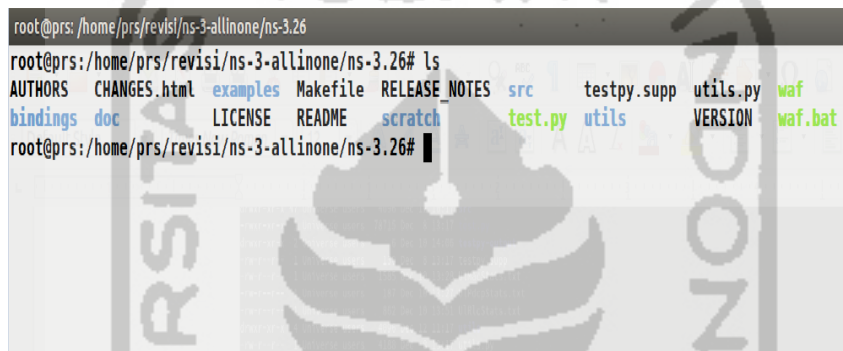
```
root@prs:/home/prs/revisi/ns-3-allinone
# Get PyBindGen
#
Required pybindgen version: 0.17.0.post57-nga6376f2
Trying to fetch pybindgen; this will fail if no network connection is available. Hit Ctrl-C to skip.
-> git clone https://github.com/gjcarneiro/pybindgen.git pybindgen
Cloning into 'pybindgen'...
remote: Counting objects: 5916, done.
remote: Total 5916 (delta 0), reused 0 (delta 0), pack-reused 5916
Receiving objects: 100% (5916/5916), 10.93 MiB | 1.11 MiB/s, done.
Resolving deltas: 100% (4033/4033), done.
Checking connectivity... done.
Fetch was successful.
-> git checkout a6376f2 -q
-> python setup.py clean
Traceback (most recent call last):
  File "setup.py", line 2, in <module>
    from setuptools import setup
ImportError: No module named setuptools
*** Did not fetch pybindgen (Command ['python', 'setup.py', 'clean'] exited with code 1); python bindings will not be available.
#
# Get NetAnim
#
Required NetAnim version: netanim-3.107
Retrieving NetAnim from http://code.nsnam.org/netanim
-> hg clone http://code.nsnam.org/netanim netanim
requesting all changes
adding changesets
adding manifests
adding file changes
added 298 changesets with 1585 changes to 228 files
updating to branch default
195 files updated, 0 files merged, 0 files removed, 0 files unresolved
#
# Get bake
#
Retrieving bake from http://code.nsnam.org/bake
-> hg clone http://code.nsnam.org/bake
destination directory: bake
requesting all changes
adding changesets
adding manifests
adding file changes
added 374 changesets with 842 changes to 63 files
updating to branch default
45 files updated, 0 files merged, 0 files removed, 0 files unresolved
root@prs:/home/prs/revisi/ns-3-allinone#
```

Gambar 4.1.Perintah untuk clone kode sumber file ns- ns-3 allinone 3.26

Langkah selanjutnya setelah proses clone selesai akan dilanjutkan dengan instalasi librari boos-devel yang akan dipergunakan untuk menjalankan module

openflow integration, module openflow ini dibutuhkan agar pengiriman paket dapat berjalan dengan optimal. Maka akan menyetikkan perintah `sudo apt-get install boost-devel` pada konsol / terminal.

Setelah instalasi ini selesai maka akan masuk ke direktori `ns-3.26`, dan akan melakukan eksekusi beberapa perintah program. Perintah pertama adalah dengan menyetikkan perintah `ls` pada terminal untuk mengetahui isi dari direktori secara keseluruhan.



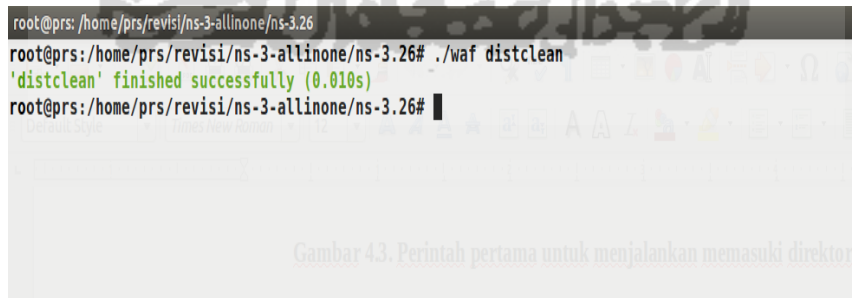
```

root@prs:/home/prs/revisi/ns-3-allinone/ns-3.26
root@prs:/home/prs/revisi/ns-3-allinone/ns-3.26# ls
AUTHORS  CHANGES.html  examples  Makefile  RELEASE_NOTES  src      testpy.suppl  utils.py  waf
bindings doc              LICENSE   README    scratch        test.py  utils         VERSION   waf.bat
root@prs:/home/prs/revisi/ns-3-allinone/ns-3.26#

```

Gambar 4.2.Perintah pertama untuk menjalankan memasuki direktori NS-3.26

Proses simulasi dijalankan pada direktori `NS-3.26` ini sehingga untuk selanjutnya akan mengeksekusi perintah `sudo ./waf distclean` (sama dengan perintah untuk `make distclean`), perintah ini akan memerintahkan program untuk membersihkan dari temporary file yang tidak diperlukan pada `waf`.



```

root@prs:/home/prs/revisi/ns-3-allinone/ns-3.26
root@prs:/home/prs/revisi/ns-3-allinone/ns-3.26# ./waf distclean
'distclean' finished successfully (0.010s)
root@prs:/home/prs/revisi/ns-3-allinone/ns-3.26#

```

Gambar 4.3.*distclean* NS-3.26

Setelah eksekusi perintah diatas menunjukkan output yang diinginkan, selanjutnya akan mengetikkan perintah `sudo ./waf --build-profile=optimized --enable-examples --enable-tests configure` yang berguna untuk membuat profile baru bagi simulasi serta untuk mengaktifkan modul test dan sample serta mengkonfigurasi awal lingkungan program.

Eksekusi perintah pada Waf dari direktori lokal, maka waf akan dikonfigurasi ulang serta membuat profile baru sebagai super user (sudo) yang akan memberikan *priveledge* sistem total saat membangun sistem simulasi serta melakukan pengecekan ulang untuk berbagai dependensi module yang dibutuhkan pada simulasi dalam profile yang dibuat. Selanjutnya perintah diatas akan memunculkan keluaran sebagaimana berikut:



```

root@prs: /home/prs/revisi/ns-3-allinone/ns-3.26# sudo ./waf --build-profile=optimized --enable-examples --enable-tests configure
re
Setting top to          : /home/prs/revisi/ns-3-allinone/ns-3.26
Setting out to         : /home/prs/revisi/ns-3-allinone/ns-3.26/build
Checking for 'gcc' (C compiler)      : /usr/bin/gcc
Checking for 'cc' version              : 5.2.3
Checking for 'g++' (C++ compiler)    : /usr/bin/g++
Checking for compilation flag -march=native support : ok
Checking for program 'python'        : /usr/local/bin/python
Checking for python version           : (2, 7, 5, 'final', 0)
python-config
Asking python-config for pyembed '--cflags --libs --ldflags' flags : yes
Testing pyembed configuration        : Could not build a python embedded interpreter
Checking boost includes              : lib boost
Checking boost libs                  : ok
Checking for boost linkage            : ok
Checking for click location          : not found
Checking for program 'pkg-config'     : /usr/bin/pkg-config
Checking for 'gtk+-2.0' >= 2.12      : yes
Checking for 'libxml-2.0' >= 2.7     : yes
Checking for type uint128_t          : not found
Checking for type __uint128_t        : not found
Checking high precision implementation : cairo 128-bit integer (default)
Checking for header stdint.h          : yes
Checking for header inttypes.h       : yes
Checking for header sys/inttypes.h   : not found
Checking for header sys/types.h      : yes
Checking for header sys/stat.h       : yes
Checking for header signal.h         : yes
Checking for header dirent.h         : yes
Checking for header stdlib.h          : yes
Checking for header pthread.h        : yes
Checking for header stdint.h          : yes
Checking for header inttypes.h       : yes
Checking for header sys/inttypes.h   : not found
Checking for library rt               : yes
Checking for header sys/ioctl.h       : yes
Checking for header net/if.h          : yes
Checking for header net/ethernet.h    : yes
Checking for header linux/if_tun.h   : yes
Checking for header netpacket/packet.h : yes
Checking for NSC location             : not found
Checking for Openflow location        : not found
Checking for 'sqlite3'               : yes
Checking for header linux/if_tun.h   : yes
Checking for program 'sudo'           : /usr/bin/sudo
Checking for program 'valgrind'       : /usr/bin/valgrind
Checking for 'gsl'                    : yes

```

Gambar Program 4.4. Output dari eksekusi perintah `sudo ./waf --build-profile=optimized --enable-examples --enable-tests configure`

Dari keluaran perintah `sudo ./waf --build-profile=optimized --enable-examples --enable-tests configure`, terlihat beberapa pilihan modul ns-3 tidak diaktifkan secara default atau memerlukan dukungan dari sistem yang mendasari untuk bekerja dengan baik (modul yang tidak diaktifkan secara default dan memerlukan proses instalasi lebih lanjut). Misalnya, untuk mengaktifkan integrasi openflow dengan NS3, maka module openflow harus didownload serta diinstalasi serta dikonfigurasi agar dapat terintegrasi secara sempurna pada NS3. Jika module

openflow ini tidak ditemukan, maka keluaran eksekusi perintah selanjutnya yang membutuhkan dukungan modul openflow tidak akan bisa dijalankan.

Proses persiapan sistem dalam hal ini adalah program simulator NS-3 versi NS-3.26 akan melalui tahap persiapan akhir, selanjutnya di layar konsole akan menjalankan perintah `./test.py core`. Perintah ini akan mengeksekusi program ns-3 untuk menjalankan sejumlah unit tes pada waf, eksekusi perintah ini akan memakan waktu cukup lama tergantung kecepatan prosesor serta kapasitas memori fisik yang ada pada pc/laptop yang digunakan, selanjutnya perintah ini akan menghasilkan proses eksekusi perintah sebagaimana terlihat pada Gambar 4.5 dibawah ini.

```

root@prs: /home/prs/revisi/ns-3-allinone/ns-3.26# ls
AUTHORS  CHANGES.html  LICENSE  RELEASE_NOTES  test.py  utils.py  waf  wscript
bindings doc  Makefile  scratch  testpy supp  utils.pyc  waf.bat  utils.py
build    examples  README  src          utils  waf-tools  waf-tools.pyc

root@prs: /home/prs/revisi/ns-3-allinone/ns-3.26# ./test.py
Waf: Entering directory '/home/prs/revisi/ns-3-allinone/ns-3.26/build'
[ 1/2482] Compiling install ns3-header: ns3/antenna-model.h
[ 2/2482] Compiling install ns3-header: ns3/isotropic-antenna-model.h
[ 3/2482] Compiling install ns3-header: ns3/parabolic-antenna-model.h
[ 4/2482] Compiling install ns3-header: ns3/cosine-antenna-model.h
[ 5/2482] Compiling install ns3-header: ns3/aodv-packet.h
[ 6/2482] Compiling install ns3-header: ns3/aodv-neighbor.h
[ 7/2482] Compiling install ns3-header: ns3/aodv-routing-protocol.h
[ 8/2482] Compiling install ns3-header: ns3/aodv-rtable.h
[ 9/2482] Compiling install ns3-header: ns3/aodv-id-cache.h
[10/2482] Compiling install ns3-header: ns3/aodv-dpd.h
[11/2482] Compiling install ns3-header: ns3/udp-client.h
[12/2482] Compiling install ns3-header: ns3/bulk-send-application.h
[13/2482] Compiling install ns3-header: ns3/on-off-helper.h
[14/2482] Compiling install ns3-header: ns3/udp-client-server-helper.h
[15/2482] Compiling install ns3-header: ns3/udp-trace-client.h
[16/2482] Compiling install ns3-header: ns3/udp-echo-server.h
[17/2482] Compiling install ns3-header: ns3/packet-sink.h
[18/2482] Compiling install ns3-header: ns3/bulk-send-helper.h
[19/2482] Compiling install ns3-header: ns3/udp-server.h
[20/2482] Compiling install ns3-header: ns3/packet-sink-helper.h
[21/2482] Compiling install ns3-header: ns3/udp-echo-client.h
[22/2482] Compiling install ns3-header: ns3/on-off-application.h
[23/2482] Compiling install ns3-header: ns3/packet-loss-counter.h
[24/2482] Compiling install ns3-header: ns3/udp-echo-helper.h
[25/2482] Compiling install ns3-header: ns3/application-packet-probe.h
[26/2482] Compiling install ns3-header: ns3/seq-ts-header.h
[27/2482] Compiling install ns3-header: ns3/bridge-net-device.h
[28/2482] Compiling install ns3-header: ns3/bridge-channel.h
[29/2482] Compiling install ns3-header: ns3/bridge-helper.h
[30/2482] Compiling install ns3-header: ns3/bridge-buildings-propagation-loss-model.h
[31/2482] Compiling install ns3-header: ns3/oh-buildings-propagation-loss-model.h
[32/2482] Compiling install ns3-header: ns3/building-list.h
[33/2482] Compiling install ns3-header: ns3/building-position-allocator.h
[34/2482] Compiling install ns3-header: ns3/building.h
[35/2482] Compiling install ns3-header: ns3/buildings-propagation-loss-model.h
[36/2482] Compiling install ns3-header: ns3/buildings-helper.h
[37/2482] Compiling install ns3-header: ns3/building-container.h
[38/2482] Compiling install ns3-header: ns3/building-allocator.h
[39/2482] Compiling install ns3-header: ns3/itu-r-1231-propagation-loss-model.h
[40/2482] Compiling install ns3-header: ns3/itu-r-1231-propagation-loss-model.h
[41/2482] Compiling install ns3-header: ns3/itu-r-1231-propagation-loss-model.h
[42/2482] Compiling install ns3-header: ns3/itu-r-1231-propagation-loss-model.h

```

Gambar 4.5. Proses dari eksekusi perintah `sudo ./test.py`

4.2 Skenario Simulasi Pengiriman Paket Data

Simulasi menggunakan simulator NS3 versi 3.26 ini, Dalam pengujian untuk menjalankan simulasi ini menggunakan 4 *router* yang memiliki peranannya masing-masing. *Router* R0, R1, dan R2 adalah jalur utama dalam proses transmisi data. Lalu R3 adalah jalur transmisi cadangan antar data dalam topologi. Terdapat juga 3 *client* dan 1 buah *server* yang dapat saling bertukar data sehingga membentuk sebuah jaringan. Setelah semua komponen tersebut dihubungkan, maka penulis akan memonitor lalu-lintas (*traffic*) jaringan. Jika *traffic* jaringan dalam topologi sudah sesuai dengan yang diinginkan, maka langkah selanjutnya,

memasukkan parameter-parameter ke dalam simulator NS-3 untuk melakukan pengujian terhadap jaringan yang telah dibuat sebelumnya. Parameter-parameter yang akan diuji berupa *jitter*, *delay*, *packet loss* dan *throughput*. Setelah mendapatkan hasil dari setiap parameter-parameter QoS tersebut dengan memanipulasi *data rate* serta *packet size*, penulis akan membandingkan hasil-hasil dari parameter tersebut, sehingga mendapatkan perbedaan yang terjadi pada nilai-nilai ke empat parameter tersebut dengan *data rate* dan *packet size* yang berbeda. Pada proses tersebut akan diadakan 16 kali simulasi dimana rincian tentang simulasi tersebut dapat dilihat pada tabel berikut :

Tabel 4.1 Skenario Simulasi

Jumlah Simulasi	Source IP	Destination IP	Deskripsi
4 kali dengan data rate 32Kbps, 64Kbps, 128Kbps, 256Kbps	192.168.1.1	192.168.3.1	Mencari pengaruh <i>data rate</i> terhadap <i>throughput</i> dan <i>delay</i> .
4 kali dengan data rate 32Kbps, 64Kbps, 128Kbps, 256Kbps	192.168.3.1	192.168.1.1	Mencari pengaruh <i>data rate</i> terhadap <i>throughput</i> dan <i>delay</i> .
4 kali dengan data rate 32Kbps, 64Kbps, 128Kbps, 256Kbps	192.168.2.1	192.168.3.1	Mencari pengaruh <i>data rate</i> terhadap <i>throughput</i> dan <i>delay</i> .
4 kali dengan data rate 32Kbps, 64Kbps, 128Kbps, 256Kbps	192.168.3.1	192.168.2.1	Mencari pengaruh <i>data rate</i> terhadap <i>throughput</i> dan <i>delay</i> .

Agar pengiriman data antar *client* dan *server* menjadi sempurna dalam topologi yang penulis bangun, maka diperlukan 6 aliran data dalam topologi. Berikut proses 6 aliran data yang akan disimulasikan :

1. Dari Client 1 (Host 0) ke Server akan dilakukan pengiriman data yang akan di lanjutkan server ke Client 2 (Host 1) ataupun ke Client 3 (Host 3)
2. Dari Client 2 (Host 1) ke Server akan dilakukan pengiriman data yang akan di lanjutkan server ke Client 1 (Host 0) ataupun ke Client 3 (Host 3)
3. Dari Client 3 (Host 3) ke Server akan dilakukan pengiriman data yang akan di lanjutkan server ke Client 1 (Host 0) ataupun ke Client 2 (Host 1)

4. Dari Server ke Client 1 (Host 0) akan dilakukan pengiriman data, dimana sumber data berasal dari Client 2 (Host 1) ataupun dari Client 3 (Host 3)
5. Dari Server ke Client 2 (Host 1) akan dilakukan pengiriman data, dimana sumber data berasal dari Client 1 (Host 0) ataupun dari Client 3 (Host 3)
6. Dari Server ke Client 3 (Host 3) akan dilakukan pengiriman data, dimana sumber data berasal dari Client 1 (Host 0) ataupun dari Client 2 (Host 1)

Berdasarkan skenario topologi diatas, selanjutnya akan dibangun implementasi topologinya ke dalam simulator NS-3. Topologi yang telah dibangun akan di *build* dalam NS-3 untuk mendapatkan hasil output jalur topologi yang telah dibuat. Hasil output tersebut kemudian dievaluasi kembali, untuk memvalidasi apakah jalur-jalur didalam topologi yang telah dibuat sudah sesuai dengan yang diinginkan maupun direncanakan ataukah belum memenuhi desain sistem sebelumnya.

Setelah topologi telah sesuai dengan yang diinginkan/ rencanakan, maka selanjutnya adalah memasukkan parameter-parameter yang dapat mempengaruhi nilai akhir dari pengiriman paket data tersebut. Pada pengujian untuk mendapatkan nilai throughput dan delay, parameter-parameter awal yang akan digunakan yaitu berupa parameter delay awal sebesar 1ms(default), serta packet size sebesar 40bytes. Waktu pengiriman paket dimulai dari detik ke 2. Lalu akan membandingkan nilai – nilai throughput dan delay dengan 4 jenis data rate (bandwidth) dengan interval 32Kbps, 64Kbps, 128Kbps, 256Kbps. Interval yang dipilih tersebut disesuaikan dengan standar yang diberikan oleh ITU-T (*Internasional Telecommunications Union*).

Pada pengujian selanjutnya untuk mendapatkan nilai jitter dan packet loss, parameter-parameter awal yang akan digunakan yaitu berupa parameter delay awal sebesar 1ms(default). Waktu pengiriman paket dimulai dari detik ke 2. Lalu akan membandingkan nilai – nilai jitter dan packet loss dengan 4 nilai packet size yang berbeda yaitu 40 bytes, 80 bytes, 120 bytes dan 160 bytes. Pada pengujian ini akan disimulasikan dengan 2 nilai data rate yang berbeda yaitu 128Kbps dan

256Kbps. Nilai-nilai tersebut dimasukkan ke dalam source code topologi yang sudah dibangun di dalam NS-3.

Berdasarkan rumus yang dikemukakan oleh (Rahmadita, Hadi, Politeknik, Negeri, & Telekomunikasi, n.d.), tentang Throughput, Delay, Jitter, Packet Loss maka implementasi dalam NS3, dapat dituangkan kedalam *source codes* sehingga nantinya akan didapatkan detail output *traffic* jaringan serta output dari hasil pengiriman paket tersebut:

- a. *Ps Packets* : (Ps packets adalah jumlah paket yang dikirim).
- b. *Pr Packets* : (Pr Packets (adalah jumlah paket yang diterima).
- c. *Ps Bytes* : (Ps Bytes adalah ukuran dari paket yang dikirim),
- d. *Pr Bytes* : (Pr Bytes (adalah ukuran dari paket yang diterima).
- e. *Throughput*: *bps* (Throughput (adalah besarnya nilai throughput (bps)).
- f. *Average delay*: *ms* (Average delay adalah rata-rata besaran nilai delay (ms)).
- g. *Average jitter*:*ms*(Average jitter: adalah rata-rata besaran nilai jitter (ms)).
- h. *Average received packet size*: *byte* (Average received packet size adalah rata-rata ukuran paket yang diterima (byte)).
- i. *Actual Packets lost Pd*: ...(packetsActual Packets lost Pd adalah jumlah paket yang mengalami drop (packets)).
- j. *Actual Packet loss*: % (Actual Packet loss adalah persentase paket yang hilang (%)).

4.3 Hasil Simulasi

Berikut ini adalah hasil pengujian menggunakan program simulasi jaringan NS-3 yang sudah di bangun sesuai dengan skenario yang telah dibahas, parameter awal berupa kapasitas *Data rate*64bps, *delay* 1ms, *packet size* 40bytes:

```

63
64 NodeContainer hosts;
65 NodeContainer routers;
66 NodeContainer routers2;
67
68 PointToPointHelper pointToPoint;
69 Ipv4AddressHelper address;
70 NetDeviceContainer devices;
71 InternetStackHelper internet;
72 MplsNetworkConfigurator network;
73
74 hosts.Create (4);
75 internet.Install (hosts);
76 routers = network.CreateAndInstall (4);
77
78 pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("64Kbps"));
79 pointToPoint.SetChannelAttribute ("Delay", StringValue ("1ms"));
80
81 // Hosts applications
82 uint16_t port = 5000;
83 UdpEchoServerHelper server (port);
84 ApplicationContainer apps = server.Install (hosts.Get(2));
85 apps.Start (Seconds (2.0));
86 apps.Stop (Seconds (12.0));
87
88 UdpEchoClientHelper client (Ipv4Address ("192.168.4.2"), port);
89 client.SetAttribute ("MaxPackets", IntegerValue (5));
90 client.SetAttribute ("Interval", TimeValue (MilliSeconds (0.01)));
91 client.SetAttribute ("PacketSize", IntegerValue (40));
92 apps = client.Install (hosts.Get (0));
93 apps.Start (Seconds (2.0));
94 apps.Stop (Seconds (4.0));
95

```

Gambar 4.6 Source Code Program untuk Inisiasi Parameter Awal

Dari hasil penentuan nilai parameter awal diatas, akan ditampilkan nilai-nilai parameter yang dicari seperti 2 gambar dibawah ini :

4.3.1 Traffic Monitoring

Hasil output jalur jaringan dalam topologi layanan.

Client 1 (Host 0) mengirim data sebesar 40bytes ke *client 3 (Host 3)*, melewati *server* 192.168.4.2 dan sebaliknya :

```

"build" finished successfully (2.677s)
NetworkDiscoverer: found link from node4:dev3 00:00:00:00:00:09 to node5:dev2 00:00:00:00:00:0a, next-hop: 10.1.1.2
NetworkDiscoverer: found link from node4:dev4 00:00:00:00:00:10 to node7:dev2 00:00:00:00:00:0f, next-hop: 10.1.5.1
NetworkDiscoverer: found link from nodes:dev2 00:00:00:00:00:08 to node4:dev3 00:00:00:00:00:09, next-hop: 10.1.1.1
NetworkDiscoverer: found link from nodes:dev3 00:00:00:00:00:0b to node6:dev2 00:00:00:00:00:0c, next-hop: 10.1.3.2
NetworkDiscoverer: found link from node6:dev2 00:00:00:00:00:0c to node5:dev3 00:00:00:00:00:0b, next-hop: 10.1.3.1
NetworkDiscoverer: found link from node6:dev3 00:00:00:00:00:0d to node7:dev3 00:00:00:00:00:0e, next-hop: 10.1.4.2
NetworkDiscoverer: found link from node7:dev1 00:00:00:00:00:0e to node6:dev3 00:00:00:00:00:0d, next-hop: 10.1.4.1
NetworkDiscoverer: found link from node7:dev2 00:00:00:00:00:0f to node4:dev4 00:00:00:00:00:10, next-hop: 10.1.5.2
Node 4 interfaces:
dev0 Type ns3::LoopbackNetDevice Hwaddr 00:00:00:00:00:00
      ln et addr:127.0.0.1 Bcast:127.255.255.255 Mask:255.0.0.0
      UP BROADCAST MTU:65535
dev1 Type ns3::PointToPointNetDevice Hwaddr 00:00:00:00:00:02
      ln et addr:192.168.1.2 Bcast:192.168.1.255 Mask:255.255.255.0
      mpls:disabled Interface:0
      UP PPP MULTICAST BROADCAST MTU:1500
dev2 Type ns3::PointToPointNetDevice Hwaddr 00:00:00:00:00:04
      ln et addr:192.168.2.2 Bcast:192.168.2.255 Mask:255.255.255.0
      mpls:disabled Interface:1
      UP PPP MULTICAST BROADCAST MTU:1500
dev3 Type ns3::PointToPointNetDevice Hwaddr 00:00:00:00:00:09
      ln et addr:10.1.1.1 Bcast:10.1.1.255 Mask:255.255.255.0
      mpls:enabled Interface:2
      UP PPP MULTICAST BROADCAST MTU:1500
dev4 Type ns3::PointToPointNetDevice Hwaddr 00:00:00:00:00:10
      ln et addr:10.1.5.2 Bcast:10.1.5.255 Mask:255.255.255.0
      mpls:enabled Interface:3
      UP PPP MULTICAST BROADCAST MTU:1500
dev5 Type ns3::PointToPointNetDevice Hwaddr 00:00:00:00:00:12
      mpls:disabled Interface:4
      UP PPP MULTICAST BROADCAST MTU:1500

```

Gambar 4.7 Output dari Alur Data dan Interface Sebuah Router

Gambar 4.8 di atas, terdapat *Network Discoverer* yang menjelaskan jalur-jalur yang terdapat pada topologi jaringan, pada Gambar 3.3. Jalur-jalur tersebut akan digunakan *client* dan *server* untuk saling melakukan pengiriman paket.

Selain *Network Discoverer* yang ditampilkan, terdapat juga contoh *interface* dari *router 0* yang menunjukkan rute atau jalur yang dapat dilewati paket.

```

UP PPP MULTICAST BROADCAST MTU:1500
Node 7 Interfaces:
dev0 Type ns3::LoopbackNetDevice HWaddr: 00:00:00:00:00:00
      Inet addr:127.0.0.1 Bcast:127.255.255.255 Mask:255.0.0.0
      UP BROADCAST MTU:65535
dev1 Type ns3::PointToPointNetDevice HWaddr: 00:00:00:00:00:0a
      Inet addr:192.168.4.2 Bcast:192.168.4.255 Mask:255.255.255.0
      npl(enabled) interface:0
      UP PPP MULTICAST BROADCAST MTU:1500
dev2 Type ns3::PointToPointNetDevice HWaddr: 00:00:00:00:00:0b
      Inet addr:10.1.5.1 Bcast:10.1.5.255 Mask:255.255.255.0
      npl(enabled) interface:1
      UP PPP MULTICAST BROADCAST MTU:1500

At time 2s client sent 40 bytes to 192.168.4.2 port 5060
At time 2s client sent 40 bytes to 192.168.4.2 port 5060
At time 2s client sent 40 bytes to 192.168.4.2 port 5060
2.00537 [node 4] Classification of the received packet (ldev 1 lvs 0x0 DSCP Default ECN Not-ECT LLL 04 04 protocol 17 offset (bytes) 0 flags [
none]) length: 00 192.168.1.1 -> 192.168.4.2
2.00537 [node 4] Found suitable entry -- Ftn0 (lnet src:192.168.1.1/32 & lnet dst:192.168.4.2/32), nhlfe(s): [default policy] (swap,100 nexthop
10.1.1.2)
2.00537 [node 4] Search of the suitable nhlfe for Ftn0 (lnet src:192.168.1.1/32 & lnet dst:192.168.4.2/32), nhlfe(s): [default policy] (swap,100
nexthop 10.1.1.2)
2.00537 [node 4] nhlfe 0 swap,100 nexthop 10.1.1.2 -- selected (*)
2.00537 [node 4] Sending labeled packet via lfi dev2 hwaddr: 00:00:00:00:00:00
2.011 [node 5] packet from 81.80.80.80:5060:45 received on node 4
2.011 [node 5] Stack top label:100 ttl:63
2.011 [node 5] Searching of label mapping for label 100 lfi dev2
2.011 [node 5] found suitable entry -- ltn0 label 100, nhlfe(s): [default policy] (pop)
2.011 [node 5] Search of the suitable nhlfe for ltn0 label 100, nhlfe(s): [default policy] (pop)
2.011 [node 5] nhlfe 0 pop selected (*)
2.011 [node 5] Stack top entry -- ip4 based forwarding must be used
At time 2.01637s server received 40 bytes from 192.168.1.1 port 49153

```

Gambar 4.8 Contoh *Output* dari Pengiriman Data

Gambar 4.9 diatas menunjukkan salah satu simulasi yang dilakukan dalam pengiriman paket dari *client 1 (Host 0)* ke *client 3 (Host 3)* yang melewati *server*. Disebutkan dalam gambar tersebut waktu yang diperlukan dari proses pengiriman hingga penerimaan paket oleh *server*.

4.3.2 Hasil Perbandingan Parameter

Setelah jalur topologi yang dibangun sudah sesuai dengan yang diinginkan, maka langkah selanjutnya adalah memberikan hasil *output* nilai-nilai parameter yang diuji.

Dari hasil penentuan nilai parameter yang sudah ditentukan, akan ditampilkan nilai-nilai parameter tentang pengiriman paket data seperti dibawah ini:


```

At time 2.13825s client received 40 bytes from 192.168.4.2 port 5060
Flow 4 ( 192.168.1.1 ->192.168.4.2 )
Ps Packets: 5
Pr Packets: 5
Ps Bytes: 340
Pr Bytes: 340
Throughput: 38857.1 bps
Average delay: 64.75 ms
Average jitter: 17.5 ms
Average received packet size: 68 byte
Actual Packets lost Pd: 0 packets
Actual Packet loss: 0 %

Flow 5 ( 192.168.2.1 ->192.168.4.2 )
Ps Packets: 5
Pr Packets: 0
Ps Bytes: 340
Pr Bytes: 0
Throughput: -nan bps
Average delay: -nan ms
Average jitter: 0 ms
Average received packet size: -nan byte
Actual Packets lost Pd: 5 packets
Actual Packet loss: 100 %

Flow 6 ( 192.168.3.1 ->192.168.4.2 )
Ps Packets: 5
Pr Packets: 5
Ps Bytes: 340
Pr Bytes: 340
Throughput: 38857.1 bps
Average delay: 73.5 ms
Average jitter: 17.5 ms
Average received packet size: 68 byte
Actual Packets lost Pd: 0 packets
Actual Packet loss: 0 %

```

Gambar 4.9 Output Nilai Parameter pengiriman dari *Client 1* ke *Client 3*

Output dari pengiriman paket ini menampilkan sejumlah parameter didapat berupa Throughput, *delay*, jitter dan juga *packet lost* dari hasil penentuan *source code* parameter awal pada Gambar 4.6, Kemudian sejumlah data tersebut untuk mencari perbandingan dari nilai parameter yang didapat.

```

Flow 7 ( 192.168.4.2 ->192.168.1.1 )
Ps Packets: 5
Pr Packets: 5
Ps Bytes: 340
Pr Bytes: 340
Throughput: 38857.1 bps
Average delay: 29.75 ms
Average jitter: 0 ms
Average received packet size: 68 byte
Actual Packets lost Pd: 0 packets
Actual Packet loss: 0 %

Flow 8 ( 192.168.4.2 ->192.168.3.1 )
Ps Packets: 5
Pr Packets: 5
Ps Bytes: 340
Pr Bytes: 340
Throughput: 38857.1 bps
Average delay: 29.75 ms
Average jitter: 0 ms
Average received packet size: 68 byte
Actual Packets lost Pd: 0 packets
Actual Packet loss: 0 %

```

Gambar 4.10 Output Nilai Parameter dari *Client 3* ke *Client 1*

Hasil *Output* dari pengiriman paket yang di peroleh menampilkan sejumlah parameter didapat seperti *Throughput*, *delay*, *jitter* dan juga *packet lost* dari hasil penentuan *source code* parameter awal pada Gambar 4.6 , Kemudian akan membandingkan nilai-nilai parameter yang di dapat

Parameter-parameter yang didapat dari hasil-hasil *output* tersebut yaitu *throughput* (dalam satuan bps), *delay* (dalam satuan ms), *jitter* (dalam satuan ms), dan *packet loss* (dalam persentase). Nilai dari parameter *throughput* dan *delay* akan dibandingkan berdasarkan *data rate* yang diberikan dalam topologi.

Berikut tabel perbandingan nilai antara parameter – parameter tersebut:

- a. Pengiriman data dari *Client 1 (Host 0)* ke *Client 3 (Host 3)*

Tabel 4.2 Perbandingan Parameter dari *Client 1 (Host 0)* ke *Client 3 (Host 3)* Dengan Perubahan *Data Rate*(dalam satuan Kbps)

Data Rate	32	64	128	256
Throughput	19428	38857	77714	155429
Delay	126.5	64.75	33.87	18.43
Jitter	35	17.5	8.75	4.3
Packet Loss	0 %	0 %	0 %	0 %

- b. Pengiriman data dari *Client 3 (Host 3)* ke *Client 1 (Host 0)*

Tabel 4.3 Perbandingan Parameter dari *Client 3 (Host 3)* ke *Client 1 (Host 0)* Dengan Perubahan *Data Rate*(dalam satuan Kbps)

Data Rate	32	64	128	256
Throughput	19428	38857	77714	155429
Delay	144	73.5	38	20.62
Jitter	35	17.5	8.75	4.3
Packet Loss	0 %	0 %	0 %	0 %

- c. Pengiriman data dari *Client 2 (Host 1)* ke *Client 3 (Host 3)*

Tabel 4.4 Perbandingan Parameter dari *Client 2 (Host 1)* ke *Client 3 (Host 3)* Dengan Perubahan *Data Rate*(dalam satuan Kbps)

Data Rate	32	64	128	256
Throughput	19428	38857	77714	155429
Delay	126.5	64.75	33.87	18.43
Jitter	35	17.5	8.75	4.3
Packet Loss	0 %	0 %	0 %	0 %

- d. Pengiriman data dari *Client 3 (Host 3)* ke *Client 2 (Host 1)*

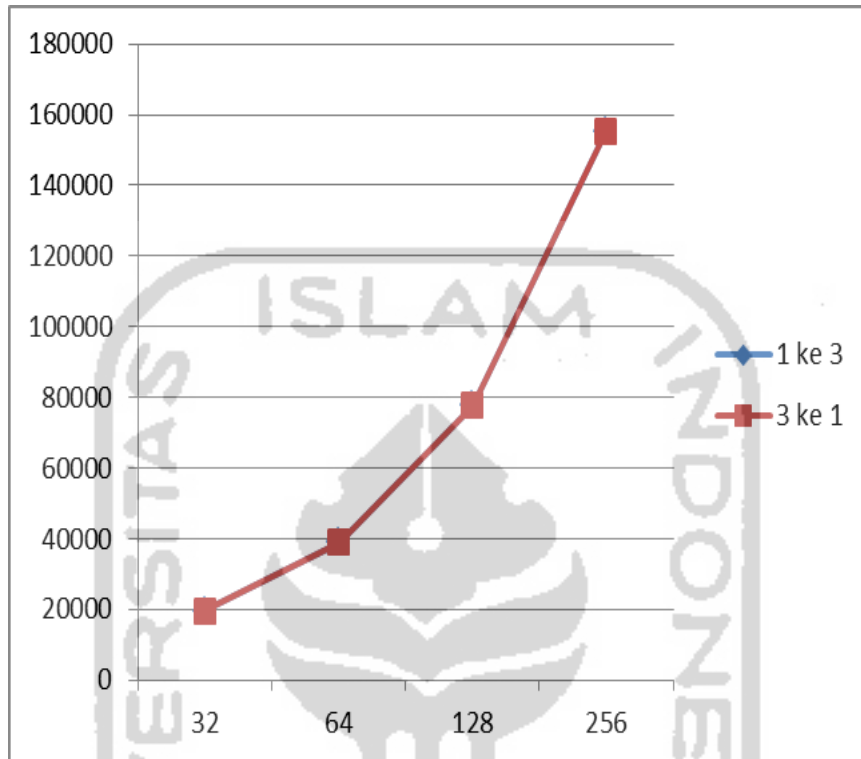
Tabel 4.5 Perbandingan Parameter dari *Client 3 (Host 3)* ke *Client 2 (Host 1)*
Dengan Perubahan *Data Rate*(dalam satuan Kbps)

Data Rate	32	64	128	256
Throughput	19428	38857	77714	155429
Delay	144	73.5	38	20.62
Jitter	35	17.5	8.75	4.3
Packet Loss	0 %	0 %	0 %	0 %

Hasil perbandingan nilai parameter dari *Client 1 (Host 0)* ke *Client 3 (Host 3)* serta sebaliknya dan juga *Client 2 (Host 1)* ke *Client 3 (Host 3)* serta sebaliknya, didapatkan bahwa nilai *throughput* terus meningkat berdasarkan *data rate* yang diberikan. Untuk nilai *delay*, terdapat penurunan nilai seiring dengan meningkatnya *data rate* yang diberikan. Terlihat juga disini bahwa pengiriman dari *client* yang berbeda tetap memiliki nilai - nilai parameter yang sama di antara keduanya karena jalur yang dilewati tidak berbeda dalam proses pengirimannya.

Berikut akan diberikan juga grafik dari parameter *throughput* dan *delay* yang diuji dengan *data rate* yang berbeda tersebut :

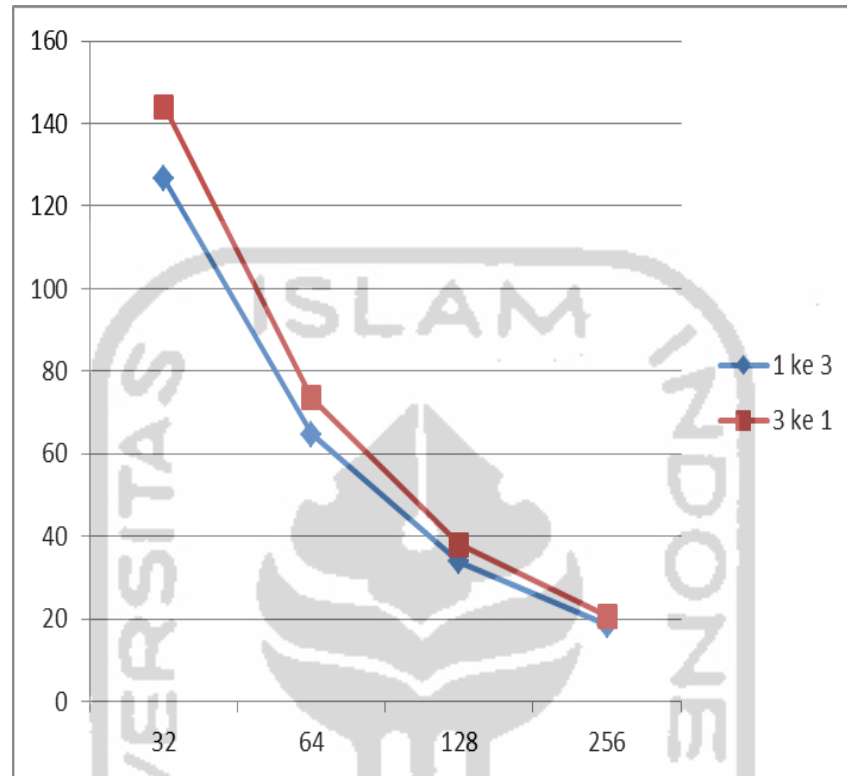
1. *Throughput*



Gambar 4.11 Grafik Perubahan Nilai Parameter *Throughput*

Dari grafik pada Gambar 4.11, pengiriman dari *client 1 ke client 3*, dan *client 3 ke client 1* mempunyai kualitas yang sama, dimana jika kualitas *throughput* meningkat, maka kualitas parameter tersebut akan memberikan kualitas layanan yang baik, dan apabila kualitas parameter *throughput* menurun, akan menyebabkan penurunan kualitas pada jaringan.

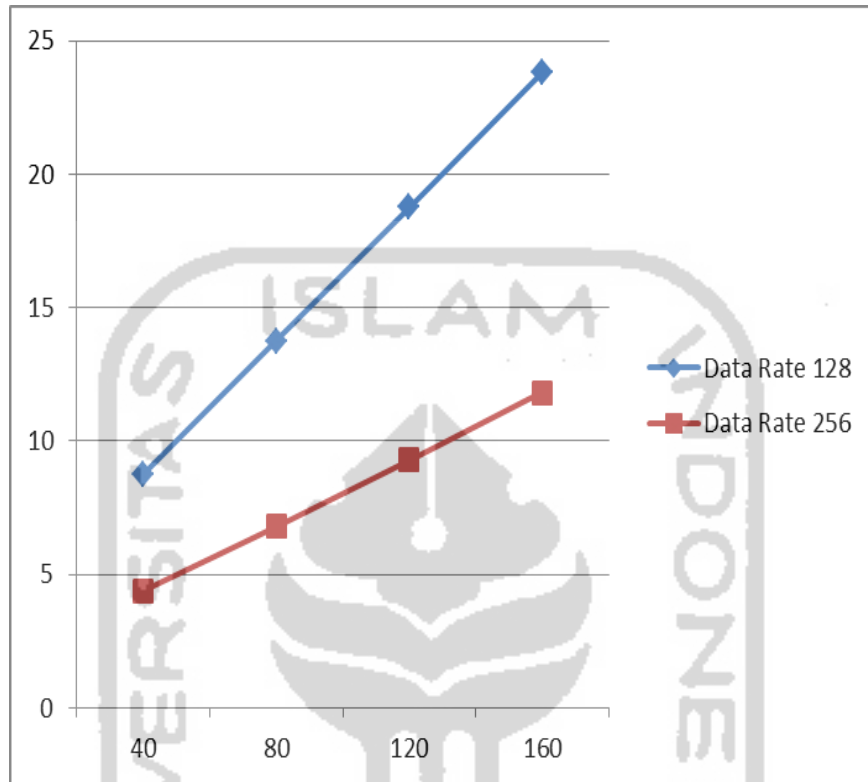
2. *Delay*



Gambar 4.12 Grafik Perubahan Nilai Parameter *Delay*

Dari grafik pada Gambar 4.12, pengiriman paket data dari *client 3* menuju *client 1* menggambarkan penurunan nilai *delay* yang lebih drastis dibandingkan pengiriman paket dari *client 1* ke *client 3*. Pengiriman paket data dari *client 3* ke *client 1* pada awalnya memiliki *delay* yang cukup jauh perbedaannya dengan jalur yang sebaliknya. Namun pada *data rate* terakhir yang diuji yaitu 256Kbps, kedua jalur tersebut memiliki perbedaan nilai *delay* yang tidak berbeda jauh. Grafik diatas menunjukkan bahwa semakin besar nilai *data rate*, semakin kecil pula nilai *delay* yang muncul begitupun sebal

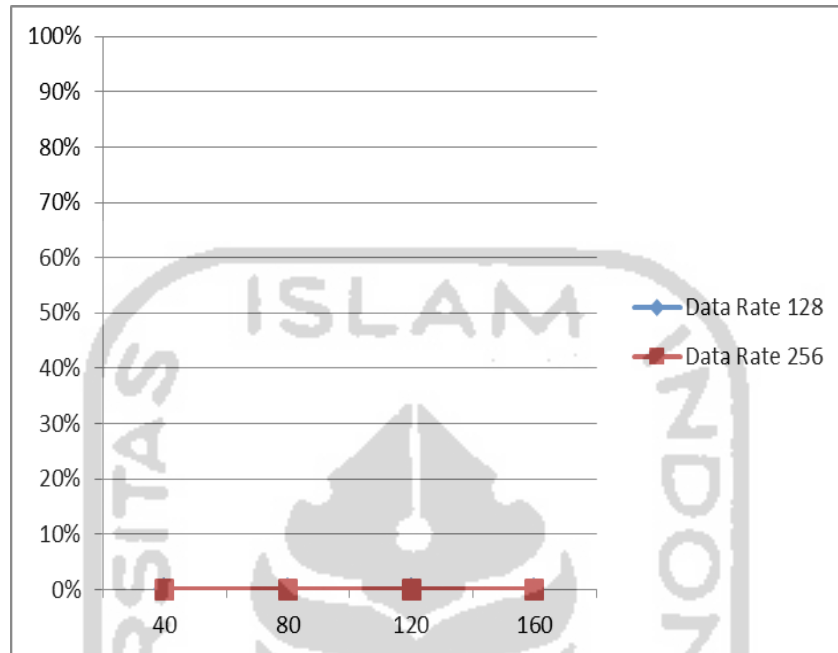
3. *Jitter*



Gambar 4.13 Grafik Perubahan Nilai Parameter *Jitter*

Dari grafik 4.13 pada Gambar 4.13, dapat terlihat bahwa pengiriman paket dari *client* 1 ke *client* 3 dan begitu juga sebaliknya mempunyai nilai *jitter* yang semakin besar sesuai dengan *packet size* yang diberikan. Nilai *jitter* yang didapat pada *data rate* sebesar 128Kbps jika diberikan *packet size* sebesar 160 bytes, nilai *jitter* yang didapat sudah berada diluar kategori Baik menurut standar kualitas ITU-T G.114 yang terdapat pada tabel 4.3 sehingga diperlukan penyesuaian *data rate* hingga mendapatkan nilai *jitter* yang sesuai standar misalnya ke 256Kbps .

4. Packet Loss



Gambar 4.14 Grafik Perubahan Nilai Parameter *Packet Loss*

Dari grafik pada Gambar 4.15, pengiriman paket dari *client 1* menuju *client 3* dan sebaliknya menggambarkan bahwa terdapat nilai parameter *packet loss* yang stabil pada nilai 0 %. Hal tersebut mengartikan bahwa tidak terdapat paket yang hilang dalam proses pengiriman data pada *data rate* yang berbeda. Selain itu, dapat terlihat juga bahwa perbedaan *packet size* yang diberikan tidak mempengaruhi hasil dari *packet loss* pada jaringan.

4.4 Pembuatan Topologi Pengiriman Paket Data

Perintah pertama untuk menjalankan simulasi adalah dengan mengetikkan perintah `ns node-config -network ON` hal ini akan mengeksekusi perintah untuk pembuatan node (LSR0).

```
$ns node-config -network ON
set LSR0 [$ns node]
$ns node-config -network OFF
```

Selanjutnya kita akan mengkonfigurasi agen LDP pada setiap node-node *network* (disini node-node dinamakan dengan LSRi, i disini dimaksudkan sebagai integer).

```
for {set i 0} {$i < n} {incr i} {
  set a LSR$i
  for {set j [expr $i+1]} {$j < 5} {incr j}
  {
    set b LSR$j
    eval $ns LDP-peer $$a $$b
  }
  set m [eval $$a get-module "network"]
  $m enable-reroute "new"
}
```

```
$ns configure-ldp-on-all-network-nodes
```

Listing Program 4.1. Eksekusi Perintah Untuk Mengkonfigurasi Node

Dari listing program diatas, maka pada langkah ke node LDR menunjukkan aktivitas program untuk menset network. Dari Listing di atas mengetahui karakteristik pengiriman paket yang didapat dari simulasi menggunakan NS-3.

4.5 Karakteristik Pengiriman Paket Data

Berdasarkan data yang telah didapat rangkuman umum tentang karakteristik pengiriman paket data dari hasil pengujian *throughput*, *jitter*, *delay*, dan *paket loss* sebagai berikut:

- *Throughput*. hasil perbandingan didapatkan bahwa nilai *throughput* terus meningkat berdasarkan *data rate* yang diberikan berarti kualitas parameter tersebut akan memberikan kualitas layanan yang baik, dan apabila kualitas parameter *throughput* menurun, akan menyebabkan penurunan kualitas pada jaringan.
- *Delay*. terdapat penurunan nilai seiring dengan meningkatnya *data rate* yang diberikan, bahwa semakin besar nilai *data rate*, semakin kecil pula nilai *delay* yang muncul begitupun sebaliknya.

- *Jitter*. nilai yang didapat pada *data rate* sebesar 128Kbps jika diberikan *packet size* sebesar 160 bytes, nilai *jitter* yang didapat sudah berada diluar kategori Baik menurut standar kualitas ITU-T G.114.
- *Packet loss*. Terdapat nilai parameter *packet loss* yang stabil pada nilai 0 %. Hal tersebut mengartikan bahwa tidak terdapat paket yang hilang dalam proses pengiriman data pada *data rate* yang berbeda.

4.6 Pemahaman Proses Pengiriman Paket Data

Melalui mekanisme simulasi proses pengiriman paket data maka secara umum kesulitan dalam memahami tahapan proses dapat diatasi hal ini terlihat dengan adanya langkah proses pengiriman paket data yang dapat memperjelas secara visual bagaimana tahapan-tahapan penting dalam proses simulasi yang tidak dapat sepenuhnya digambarkan melalui sejumlah penjelasan narasi ataupun ilustrasi gambar. Selanjutnya berdasarkan implementasi proses pengiriman paket data yang telah dijalankan melalui pendekatan simulasi NS-3 ini, maka didapat sebuah rangkuman sederhana tentang apa yang dapat dilakukan oleh simulasi untuk memperjelas pemahaman tentang proses pengiriman paket data. Hal ini terlihat dari Tabel 4.5

Table 4.5.Proses Simulasi

No	Topik	Penjelasan lewat media tulis	Penjelasan lewat Simulasi Paket Data
1	<i>Jitter</i>	<i>Jitter</i> atau variasi dalam <i>latency</i> , diakibatkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data.	Untuk memahami <i>jitter</i> lebih lanjut maka dalam simulasi dapat dilakukan dengan cara membandingkan nilai ukur dari <i>packet size</i> dan <i>packet rate</i> seperti pada simulasi (“ <i>DataRate</i> ”, <i>StringValue</i> (“ <i>128Kbps</i> ”)); (“ <i>PacketSize</i> ”, <i>UIntegerValue</i> (160));
2	<i>Packet loss</i>	<i>Packet loss</i> ialah kegagalan transmisi paket data untuk mencapai tujuannya.	Untuk memahami <i>Packet loss</i> lebih lanjut maka hal yang dapat dilakukan dalam simulasi adalah dengan cara membandingkan nilai ukur dari paket data (“ <i>PacketSize</i> ”, <i>UIntegerValue</i> (40));
3	<i>Throughput</i>	<i>Throughput</i> ialah kecepatan (<i>rate</i>) transfer data.	Untuk memahami <i>Throughput</i> maka pada simulasi dapat diukur berdasarkan besarnya data rate yang dimasukkan (“ <i>DataRate</i> ”, <i>StringValue</i> (“ <i>64Kbps</i> ”));
4	<i>Delay</i>	<i>Delay</i> adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik lain.	Untuk mengetahui nilai <i>Delay</i> maka melalui simulator dapat dilakukan dengan memberikan nilai pada source code sebagai berikut (“ <i>Delay</i> ”, <i>StringValue</i> (“ <i>1ms</i> ”));

Dari data pada Tabel 4.5, maka terlihat bahwa penggunaan simulasi untuk menjelaskan proses pengiriman paket data akan jauh lebih memberikan pemahaman yang lebih baik ketika menjelaskan tentang proses pengiriman paket data. Dalam hal ini, desain simulasi dengan NS-3 yang telah dilakukan mampu untuk menjelaskan dengan baik bagian-bagian penting dari proses pengiriman paket data.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan proses penelitian yang telah dilakukan pada penelitian ini, maka didapat beberapa hal sebagai kesimpulan.

- Pada penelitian ini telah berhasil dilakukan simulasi pengiriman paket data menggunakan tools simulator NS3. Untuk dapat melakukan simulasi tersebut maka langkah yang harus dipersiapkan adalah membuat scenario simulasi yang memuat alat IP dan label untuk proses pengiriman paket data serta kemudian menentukan sejumlah parameter pengukuran. Setelah semuanya siap maka kemudian melakukan penambahan dan setting pada *scriptsource code* pada simulator NS3 agar proses simulasi dan pencatatan hasilnya dapat dilakukan dengan baik.
- Secara umum tools NS3 memiliki kemudahan dalam mendukung penjelasan untuk memahami proses pengiriman paket data. Modul NetAnim membuat proses animasi dalam bentuk XML dapat dilakukan berdasarkan pada trace file yang ada, sementara modul sublime text-3 yang sudah menggunakan dukungan API memudahkan dalam proses auto compile dalam *source code* C++. Hal ini akan memudahkan membangun scenario jaringan dan penentuan parameter yang akan diukur dalam proses simulasi.

Selanjutnya berkenaan dengan karakteristik pengiriman paket yang didapat dari simulasi menggunakan NS-3. Maka dapat disimpulkan sbb:

- Berdasarkan proses simulasi yang telah dilakukan maka didapat beberapa karakteristik dari proses pengiriman paket data, yaitu : *Troughput*, *jitter*, *delay* dan juga *packet loss*.

Karakteristik tersebut menjelaskan sejumlah hal tentang *point* penting dari proses pengiriman paket data yaitu: *Troughput*, *jitter*, *delay*, *packet loss*. Dari simulasi menjelaskan dengan baik berdasarkan *Data Rate* yang

diberikan terhadap *throughput* meningkatnya kualitas parameter tersebut akan memberikan kualitas layanan yang baik, dan apabila kualitas parameter *throughput* menurun, akan menyebabkan penurunan kualitas pada jaringan.

Melalui mekanisme simulasi proses pengiriman paket data maka secara umum kesulitan dalam memahami tahapan proses dapat diatasi hal ini terlihat dengan adanya langkah proses pengiriman paket data untuk beberapa parameter tersebut dapat diakomodasi oleh pemilihan perintah dan setting variable yang sesuai yang umumnya sudah terdida dalam NS3.

5.2 Saran

Secara umum, penelitian tentang simulasi jaringan dengan menggunakan software simulasi open source NS3 belumlah begitu banyak, untuk itu masih sangat diperlukan penelitian sejenis dengan membandingkan hasil yang didapat dalam penelitian ini dengan tools simulator lainnya, misalnya: GNS3, OMNET++, SWAN, OPNET, Jist, dan GloMoSiM. Selain itu penggunaan scenario pengiriman paket data yang lebih kompleks akan mendapatkan data yang lebih baik untuk dapat menyimpulkan hal-hal penting dari parameter yang diukur dalam penelitian ini.

DAFTAR PUSTAKA

- Alborz, N., Keyvani, M., Nikolic, M., & Trajkovic, L. (2000). Simulation of packet data networks using opnet. *SIMULATION OF PACKET DATA NETWORKS USING OPNET*, 1–6.
- Azmi, Z. (2014). Deteksi dan Koreksi Kesalahan Pada Komunikasi Data. *Deteksi Dan Koreksi Kesalahan Pada Komunikasi Data*, 13(2), 159–167.
- Chaudhary, R. (2012). A study of comparison of Network Simulator -3 and Network Simulator -2. *International Journal of Computer Science and Information Technologies*, 3(1), 3085–3092.
- Darmawan & Alif & Basuki. (2013). Analisis Qos (Quality of Service) Pada Jaringan Internet (Studi Kasus : Fakultas Teknik Universitas Tanjungpura). *Analisis Qos (Quality of Service)*, 1–6.
- Irawan, D. (2011). Simulasi Model Jaringan Mobile Ad-Hoc (Manet) Dengan Ns-3. *Badan Pengkajian Dan Penerapan Teknologi, Jakarta. Jurnal Konferensi Nasional Sistem Dan Informatika 2011; Bali, November 12, 2011.*, 335–339.
- Khan, A. ur R., Bilal, S. M., & Othman, M. (2012). A Performance Comparison of Network Simulators for Wireless Networks. *2012 IEEE International Conference on Control System, Computing and Engineering*, 34–38. <https://doi.org/10.1109/ICCSCE.2012.6487111>
- Khasanah, F. N. (2014). (2014). Perancangan Dan Simulasi Jaringan Komputer Menggunakan Graphical Network Simulator 3. *Perancangan Dan Simulasi Jaringan Komputer Menggunakan Graphical Network Simulator 3*, 3, 2014.
- Prawira, D. Y. (2015). Analisis Kinerja Jaringan Multiprotocol Label Switching (Mpls) Untuk Layanan Video Streaming. *Analisis Kinerja Jaringan Multiprotocol Label Switching (Mpls) Untuk Layanan Video Streaming*, 13, 30–35.
- Rahmadita, D. A., Hadi, M. Z. S., Politeknik, M., Negeri, E., & Telekomunikasi, J. T. (n.d.). Analisa aplikasi voip pada jaringan berbasis mpls. *Quality*, 1–6.
- Ramadhan, 2011. (2011). Analisa Performa Routing Protocol AODV , OLSR , dan DSDV Menggunakan NS-3 Pada Mobile AD-HOC Network. *Analisa Performa Routing Protocol AODV , OLSR , Dan DSDV Menggunakan NS-3 Pada Mobile AD-HOC Network*.
- Sulistyo, W. (2009). Perancangan Reliabilitas Sistem Trasmisi Data Pada Protokol UDP (User Datagram Protokol). *Perancangan Reliabilitas Sistem Trasmisi Data Pada Protokol UDP (User Datagram Protokol)*, 2009(semnasIF), 74–82.
- Weing, E., Lehn, H., & Wehrle, K. (2009). A performance comparison of recent network simulators - Google Acadêmico. Retrieved from http://scholar.google.com.br/scholar?q=A+performance+comparison+of+recent+network+simulators&hl=pt-BR&as_sdt=0&as_vis=1&oi=scholart&sa=X&ei=w-QmUfWGC-Hm0gHet4HwCg&ved=0CEAQgQMwAA
- Yousefi, F. (2015). How to simulate a node receiving and sending packets through

two different mac protocols in NS2?

Yuliansyah, H. (2016). Uji Kinerja Pengiriman Data Secara Wireless Menggunakan Modul ESP8266 Berbasis Rest Architecture.

D. Saladino, A. P. (2012). A Tool for Multimedia Quality Assessment in NS3: QoE Monitor.

Purbo Onno W, et al. 2002. TCP/IP: Standar, Desain dan Implementasi. Jakarta: PT Elex Media Komputindo.

NS3. (2011), “ns-3: ns-3 Documentation” <https://nsnam.org/documentation/> (diakses 10 November 2015)

NS3. (2011), “ns-3: ns-3 Wiki” <https://www.nsnam.org/wiki/>. Diakses pada tanggal 10 November 2015

The Cisco System Documentation, “Quality of Service Solutions Configuration Guide”, page at <http://www.cisco.com>, 2015. Diakses pada tanggal 10 November 2015.

