

IMPLEMENTASI ALGORITME TCP WESTWOOD+ UNTUK MENGURANGI DAMPAK KONGESTI PADA JARINGAN LTE

SKRIPSI

untuk memenuhi salah satu persyaratan mencapai derajat Sarjana S1



Disusun oleh:

Abdinul Fakhroni

14524098

**Jurusan Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta**

2018

LEMBAR PENGESAHAN
IMPLEMENTASI ALGORITME TCP WESTWOOD+ UNTUK MENGURANGI DAMPAK
KONGESTI PADA JARINGAN LTE

TUGAS AKHIR
ISLAM

Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia

Disusun oleh:

Abdinul Fakhroni
14524098



Yogyakarta, 30 Mei 2018

Menyetujui,

Pembimbing



Ida Nurcahyani, ST., M. Eng.
155240104

LEMBAR PENGESAHAN

SKRIPSI

**IMPLEMENTASI ALGORITME TCP WESTWOOD+ UNTUK MENGURANGI
DAMPAK KONGESTI PADA JARINGAN LTE**

Dipersiapkan dan disusun oleh:

Abdinul Fakhroni

14524098

Telah dipertahankan di depan dewan penguji

Pada tanggal: 20 Juli 2018

Susunan dewan penguji

Ketua Penguji : **Ida Nurcahyani, S.T. M.Eng.,**

Anggota Penguji 1: **Tito Yuwono S.T., M.Sc.,**

Anggota Penguji 2: **Dzata Farahiyah, ST, M.Sc.,**

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana

Tanggal: 25 Juli 2018

Ketua Program Studi/Teknik Elektro



Dr. Eng. Hendra Setiawan, S.T., M.T.

025200526

PERNYATAAN

Dengan ini Saya menyatakan bahwa:

1. Skripsi ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan Saya juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Skripsi yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal penggunaan informasi dan materi Skripsi terkait paten maka akan diskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut diatas.

Yogyakarta, 10 Mei 2018



Abdinul Fakhroni

KATA PENGANTAR



Assalamu'alaykum Warahmatullahi Wabarakatuh,

Alhamdulillahirabbil'alamin, puji dan syukur penulis panjatkan kepada Allah SWT yang Maha Pengasih dan Maha Penyayang, yang telah memberikan rahmat serta karunia-NYA sehingga Tugas Akhir yang berjudul : “IMPLEMENTASI ALGORITME TCP *WESTWOOD*+ UNTUK MENGURANGI DAMPAK KONGESTI PADA JARINGAN LTE” ini dapat diselesaikan dengan baik dan lancar. tak lupa pula shalawat dan salam tercurahkan kepada Rasulullah Muhammad SAW yang menjadi teladan bagi kita.

Tujuan penulisan laporan Tugas Akhir ini sebagai salah satu syarat kelulusan pada Pendidikan Strata Satu (S1) Fakultas Teknologi Industri Jurusan Teknik Elektro Universitas Islam Indonesia selain itu agar dapat bermanfaat bagi para pembaca. Dalam penulisan laporan tugas akhir ini penulis mendapatkan bantuan dari berbagai pihak untuk itu penulis mengucapkan banyak terima kasih atas bantuan dan dukungannya. Penulis mengucapkan terima kasih antara lain kepada :

1. Orang tua penulis atas semua dukungan, semangat, serta doa yang telah mereka berikan.
2. Bapak Dr.Eng. Hendra Setiawan, S.T., M.T selaku Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Ibu Ida Nurcahyani S.T, M.Eng. selaku Dosen Pembimbing Tugas Akhir yang telah mendampingi dan memberikan berbagai masukan dalam penulisan laporan ini.
4. Segenap Dosen Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia yang telah membimbing dan memberikan ilmunya selama penulis duduk di bangku kuliah.
5. Kakak dan Adik penulis yang selalu memberikan semangat untuk mengerjakan Tugas Akhir ini.

6. Teman – teman Teknik Elektro UII pada umumnya dan khususnya angkatan 2014 atas doa dan dukungannya.
7. Pihak – pihak lain yang tidak dapat disebutkan satu persatu, baik secara langsung maupun tidak langsung telah membantu penulis dalam penyelesaian laporan ini yang telah membantu dalam penyelesaian skripsi ini.

Dalam penulisan laporan ini penulis menyadari masih terdapat kekurangan untuk itu penulis memohon maaf dikarenakan keterbatasan yang dimiliki penulis baik dalam segi pengalaman maupun segi pengetahuan, sehingga penulisan laporan tugas akhir ini masih jauh dari kata sempurna. Semoga skripsi ini bisa bermanfaat bagi pembaca dan penggunanya.

Wassalamu'alaykum Warahmatullahi Wabarakatuh.

ARTI LAMBANG DAN SINGKATAN

LTE	: <i>Long Term Evolution</i>
TCP	: <i>Transmission Control Protocol</i>
UE	: <i>User Equipment</i>
OSI	: <i>Open Sytem Interconnection</i>
NS	: <i>Network Simulator</i>
QoS	: <i>Quality of Services</i>
3GPP	: <i>The Third Generation Partnership Project</i>
GSM	: <i>Global System for Mobilecommunication</i>
IP	: <i>Internet Protocol</i>
SLA	: <i>Service Level Agreement</i>
AIMD	: <i>Additive Increase Multiplicative Decrease</i>
AIAD	: <i>Additive Increase Addaptive Decrease</i>
ACK	: <i>Acknowledgement</i>
RTT	: <i>Round Trip Time</i>
HTTPs	: <i>Hypertext Transfer Protocol Secure</i>
SMTP	: <i>Simple Mail Transfer Protocol</i>
FTP	: <i>File Transfer Protocol</i>
UCLA	: <i>Universitas California Los Angeles</i>
sshtres	: <i>slow start threshold</i>
cwnd	: <i>congestion windows</i>
MSN	: <i>Microsoft Network</i>
GPL	: <i>Gnu Public License</i>
UDP	: <i>User Datagram Protocol</i>
OFDMA	: <i>Orthogonal Frequency Division Multiple Access</i>

SC-FDMA	: <i>Single Carrier Frequency Division Multiple Access</i>
FDD	: <i>Frequency Division Duplexing</i>
PLR	: <i>Packet Loss Ratio</i>
OS	: <i>Operating System</i>
TIPHON	: <i>Telecommunications and Internet Protocol Harmonization Over Networks</i>
ETSI	: <i>European Telecommunications Standards Institute</i>

ABSTRAK

Komunikasi seluler 4G atau biasa disebut juga *Long Term Evolution* (LTE) merupakan perkembangan dari teknologi sebelumnya yakni 3G dan 3.5G. Dengan meningkatnya jumlah layanan data yang disebabkan bertambahnya pengguna dan bertambahnya kebutuhan akses data dapat menimbulkan kongesti pada jaringan. Hal ini dapat membuat jaringan menurunkan *bit rate* dan bertambahnya *delay* terhadap transmisi jaringan. Oleh sebab itu dibutuhkan sebuah mekanisme kontrol kongesti untuk menangani atau mengurangi efek kongesti tersebut. Penulis menggunakan mekanisme kontrol kongesti algoritme *Transmission Control Protocol* (TCP). Dalam penelitian ini diimplementasikan jenis algoritme TCP *Westwood+* untuk mengurangi kongesti pada jaringan LTE. Skenario yang digunakan dalam penelitian ini adalah simulasi pada jaringan LTE tanpa algoritme TCP *Westwood+* dan dengan algoritme TCP *Westwood+* dengan jumlah *node* yang digunakan 100, 120, 140, 160, 180, dan 200 *node* dengan waktu simulasi yang digunakan adalah 60 detik untuk masing-masing simulasi. Hasil penelitian ini menunjukkan adanya kenaikan rata-rata untuk *delay*, tanpa TCP *Westwood+* sebesar 0.2706336s dan dengan TCP *Westwood+* sebesar 0.21189611s untuk PLR, tanpa TCP *Westwood+* sebesar 2.181% dan dengan TCP *Westwood+* sebesar 1.76% sedangkan untuk *throughput* sendiri rata-rata penurunannya, tanpa TCP *Westwood+* sebesar 0.004172014 Mbps dan dengan TCP *Westwood+* sebesar 0.00419459 Mbps.

Kata Kunci : LTE, kongesti, TCP *Westwood+*, QoS

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
PERNYATAAN.....	ii
KATA PENGANTAR.....	iv
ARTI LAMBANG DAN SINGKATAN	vi
ABSTRAK	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
BAB 2 TINJAUAN PUSTAKA	4
2.1 Studi Literatur	4
2.2 Tinjauan Teori.....	6
2.2.1 LTE (<i>Long Term Evolution</i>)	6
2.2.2 TCP (<i>Transmission Control Protocol</i>)	6
2.2.3 TCP Westwood+.....	7
2.2.4 Kongesti.....	8
2.2.5 NS-3 (<i>Network Simulator 3</i>).....	9
2.2.6 FTP (<i>File Transfer Protocol</i>).....	9

BAB 3 METODOLOGI	10
3.1 Mekanisme Sistem Perancangan	10
3.2 Studi Literatur	11
3.3 Menentukan Parameter Jaringan.....	12
3.4 Instalasi Perangkat	12
3.5 Menentukan Parameter QoS (<i>Quality of Services</i>) yang dianalisis	13
3.6 Membuat Skenario Program	15
3.7 Menjalankan Simulasi.....	15
3.8 Pengambilan Data	16
3.9 Analisis Hasil Data Simulasi	16
3.10 Representasi Data	16
3.11 Penyusunan Laporan.....	16
BAB 4 HASIL DAN PEMBAHASAN.....	17
4.1 Perbandingan Delay.....	17
4.2 Perbandingan PLR (<i>Packet Loss Ratio</i>).....	19
4.3 Perbandingan Throughput.....	21
BAB 5 KESIMPULAN DAN SARAN.....	23
5.1 Kesimpulan	23
5.2 Saran	24
DAFTAR PUSTAKA	25
LAMPIRAN	29

DAFTAR GAMBAR

Gambar 2.1 Model OSI.....	7
Gambar 2.2 Mekanisme Kontrol Kongesti TCP <i>Westwood+</i> [16].....	8
Gambar 3.1 Topologi Simulasi.....	10
Gambar 3.2 <i>Flowchart</i> Metodologi Penelitian	11
Gambar 4.1 Perbandingan <i>Delay</i>	17
Gambar 4.2 Perbandingan <i>Packet Loss Ratio</i>	19
Gambar 4.3 Perbandingan <i>Throughput</i>	21

DAFTAR TABEL

Tabel 3.1 Parameter Simulasi.....	12
Tabel 3.2 Parameter QoS	15
Tabel 4.1 Kenaikan Rata-Rata <i>Delay</i>	18
Tabel 4.2 Perbedaan <i>Delay</i>	18
Tabel 4.3 Kenaikan Rata-Rata PLR	20
Tabel 4.4 Perbedaan PLR.....	20
Tabel 4.5 Penurunan Rata-Rata <i>Throughput</i>	21
Tabel 4.6 Perbedaan <i>Throughput</i>	22

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Dalam era yang sudah modern ini perkembangan akan jaringan internet sudah sangat cepat. Kebutuhan manusia akan pertukaran data dan informasi semakin meningkat. Maka dari itu jaringan telekomunikasi harus memiliki kualitas yang memadai. Perkembangan sistem teknologi komunikasi bergerak ini dari generasi yang pertama ke generasi berikutnya sangat pesat sekali. Salah satunya teknologi 4G atau *Long Term Evolution* (LTE) merupakan perkembangan dari teknologi sebelumnya yakni 3G dan 3.5G. Teknologi LTE ini merupakan suatu jawaban dari permasalahan meningkatnya kualitas pelayanan kepada pelanggan [1]. LTE adalah suatu teknologi yang menawarkan berbagai keuntungan dalam dunia jaringan bergerak diantaranya kecepatan akses yang cukup tinggi, kapasitas pengguna yang besar, kesederhanaan arsitektur, kemudahan dalam implementasi, serta luasnya pilihan dalam jenis *User Equipment* (UE) yang dapat mengakses LTE [1].

Dengan meningkatnya jumlah layanan data yang disebabkan bertambahnya pengguna dan bertambahnya kebutuhan akses data hal ini menimbulkan kongesti pada jaringan yang dapat membuat jaringan tidak dapat untuk melewati layanan data dan menjadi kurang maksimal. Oleh sebab itu dibutuhkan mekanisme kontrol kongesti untuk menangani kongesti tersebut [2]. Dalam dunia internet *Transmission Control Protocol* (TCP) merupakan penanggung jawab transaksi data yang terjadi dan khususnya dalam hal penanganan kongesti [2]. Dalam pemodelan *Open System Interconnection* (OSI) TCP merupakan protokol yang bekerja pada *layer transport* [3]. *TCP Westwood+* merupakan salah satu jenis TCP yang digunakan dalam penanganan kongesti [4].

Oleh sebab itu melalui implementasi algoritme *TCP Westwood+* pada jaringan LTE diharapkan dapat mengurangi dampak kongesti. Simulasi dilakukan dengan perangkat lunak *Network Simulator-3* (NS-3) pada nilai *Quality of Services* (QoS) yang didapatkan.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini :

1. Bagaimana pengaruh penambahan jumlah pengguna pada jaringan LTE tanpa algoritme TCP *Westwood+* dan dengan algoritme TCP *Westwood+*?
2. Bagaimana mengimplementasikan kontrol kongesti pada jaringan LTE dengan algoritme TCP *Westwood+*?
3. Bagaimana kinerja jaringan LTE setelah diimplementasikan algoritme TCP *Westwood+* untuk menangani kongesti?

1.3 Batasan Masalah

Pada penelitian ini dilakukan batasan-batasan masalah sebagai berikut :

1. Parameter *Quality of Services* (QoS) yang dianalisis adalah *delay*, *Packet Loss Ratio* (PLR), dan *throughput*.
2. Simulasi hanya dilakukan pada jaringan LTE dengan menggunakan perangkat lunak *Network Simulator-3* (NS-3) versi NS-3.27.
3. Menggunakan ruang lingkup 1 *cell*.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Mengurangi kongesti dengan mengimplementasikan jenis algoritma TCP *Westwood+* pada jaringan LTE.
2. Membandingkan dan menganalisis parameter kinerja jaringan *Quality of Services* (QoS) tanpa algoritme TCP *Westwood+* dan dengan algoritme TCP *Westwood+* pada jaringan LTE.

1.5 Manfaat Penelitian

Manfaat penelitian ini adalah :

1. Mengurangi kongesti dengan mengimplementasikan jenis algoritma TCP *Westwood+* pada jaringan LTE.
2. Mengetahui kinerja algoritme TCP *Westwood+* pada jaringan LTE.

BAB 2

TINJAUAN PUSTAKA

2.1 Studi Literatur

M. Fajri Fitrianto, Sukiswo, dan Imam Santoso [5] melakukan penelitian mengenai kinerja TCP BIC untuk pencegahan kongesti pada jaringan LTE. Menggunakan *Network Simulator-2.33* (NS-2.33) dilakukan analisis *throughput* dan *fairness*. Dalam proses simulasi menggunakan jumlah pengguna 5 dan dalam satu jaringan LTE yang sama tetapi dengan variasi *Beta* yang digunakan 0.6 sampai 0.9, dan untuk analisa terhadap *delay end-to-end* dengan menggunakan 6 pengguna aktif digunakan variasi topologi yakni 1 eNodeB, 2 eNodeB dan 2 gateway. Hasil penelitian untuk TCP BIC 0.9 memiliki *throughput* paling tinggi tetapi *fairness* yang rendah, dan untuk TCP BIC 0.6 memiliki *throughput* paling rendah tetapi *fairness* paling tinggi. Untuk analisis *delay end-to-end* diperoleh *delay* dalam topologi 1 eNodeB 4.8ms, sedangkan topologi 2 eNodeB 8.9 ms. *Delay* yang dihasilkan dengan menggunakan TCP BIC tergolong tinggi.

Grace Karlina Permatasari, Sukiswo, dan Imam Santoso [6] dalam penelitiannya melakukan pengujian kinerja TCP *Westwood* untuk pencegahan kongesti pada jaringan LTE dengan menggunakan *Network Simulator 2.33*. Skenario simulasi yang dibagi menjadi dua yakni skenario perhitungan *throughput* dan *fairness* serta perhitungan *delay end-to-end*. Lama simulasi 300s pada model skenario perhitungan *throughput* dan *fairness* menggunakan 5 pengguna, dan untuk model skenario kedua model perhitungan *delay end-to-end* menggunakan 6 pengguna yang *mobile* dengan waktu simulasi yang dipakai selama 50s. Hasil dari simulasi pada analisis *throughput*, maka *throughput* dengan *loss rate* $1e-3$ dan $5e-3$ memiliki perbedaan *throughput* sebesar 0.23007691 Mbps. Hasil analisis yang diperoleh adalah TCP *Westwood* dengan *loss rate* $5e-3$ lebih baik dari TCP *Westwood* dengan *loss rate* $1e-3$ dalam segi *indeks fairness*. Untuk pengujian *delay end-to-end* topologi 1 eNB nilai *delay* lebih kecil dibandingkan dengan topologi 2 eNB dan 2 aGW, maka dari itu *fairness* topologi 1 eNB lebih bagus.

Aria Tanzila Harfad, Sabriansyah Rizqika Akbar, dan Adhitya Bhawiyuga [7] dalam penelitiannya melakukan pengimplementasian mekanisme kontrol kongesti untuk mengatur lalu lintas jaringan saat terjadinya pemakaian berlebih. Penelitian ini membandingkan kinerja empat varian TCP, yakni TCP *Cubic*, TCP *Reno*, TCP *Vegas* dan TCP *Westwood+*. Pengujian analisis *cwnd*, *ssthresh* dan *throughput*. Dalam penelitian ini kami menggunakan perangkat *sender* dan *receiver*. Keduanya dihubungkan dengan menggunakan kabel *Ethernet* yang memiliki koneksi dengan *bandwidth* sebesar 100Mbit/s. Data TCP ditampilkan berupa grafik dengan *Gnuplot* dan hasil dari penelitian ini adalah TCP *Westwood+* yang terbaik dari TCP *Cubic*, TCP *Reno* dan TCP *Vegas* di semua lingkungan *Service Level Agreement (SLA) service provider* dengan mekanisme *Bandwidth Estimation* yang digunakannya.

Widianto Adinugroho, Ida Wahidah, dan R. Rumani [2] dalam penelitiannya melakukan perbandingan performansi kontrol kongesti TCP *Westwood+* dengan TCP *New Reno*. Dalam penelitian ini melakukan pengujian efektifitas terhadap peningkatan kinerja yang dilakukan oleh TCP *Westwood+* kepada TCP *New Reno* yang dilakukan dengan cara mensimulasikannya. Hasil penelitiannya didapatkan bahwa TCP *Westwood+* dapat mampu meningkatkan perolehan *goodput* ketika *Acknowledgement (ACK) compression* sebesar 0.118 Mbps dan *packet loss* dalam *lossy medium* sebesar 27%. TCP *Westwood+* ini juga dapat meningkatkan *fairness* dengan nilai rata-rata 0.02 antar koneksi pada saat terjadi perbedaan *Round Trip Time (RTT)* antar koneksi yang cukup signifikan. TCP *Westwood+* ini juga memberikan *goodput* yang cukup tinggi yakni sebesar 0.98 Mbps dari TCP *New Reno* 0.882 Mbps ketika terjadi variasi *route* dan ukuran paket. TCP *Westwood+* ini merupakan protokol yang *friendly* terhadap TCP *New Reno* dalam segi hal *friendliness* yakni dengan nilai toleransi di atas 0.9.

Dalam penelitian ini, penulis menggunakan jenis algoritme TCP *Westwood+* dalam mengurangi dampak kongesti pada jaringan LTE. Menggunakan *Network Simulator-3 (NS-3)* dilakukan analisis terhadap *delay*, *Packet Loss Ratio (PLR)*, dan *throughput*.

2.2 Tinjauan Teori

2.2.1 LTE (*Long Term Evolution*)

Long Term Evolution atau biasa disingkat LTE merupakan sebuah standar komunikasi akses data nirkabel tingkat tinggi [8]. LTE ini diperkenalkan oleh *Third Generation Partnership Project* (3GPP) yang dirancang untuk meningkatkan kapasitas dan kecepatan jaringan sistem telekomunikasi bergerak dari pendahulunya [9]. Koneksi super cepat inilah merupakan salah satu kelebihan dari LTE selain itu jangkauan LTE pun lebih luas.

Agar LTE menjangkau seluruh wilayah, teknologi ini menggunakan rentang *channel* yang cukup lebar, mulai dari 1.4 MHz sampai 20 MHz [10]. LTE juga memberikan tingkat kapasitas *bit rate downlink* minimal sebesar 100 Mbps dan untuk *uplink* paling sedikit 50 Mbps [7]. Teknologi LTE ini dapat memenuhi regulasi yang telah ditentukan di setiap negara [11].

2.2.2 TCP (*Transmission Control Protocol*)

Transmission Control Protocol (TCP) adalah protokol yang berorientasi koneksi yang menyediakan layanan yang dapat diandalkan untuk aplikasi data [3]. TCP ini biasa dipergunakan apabila protokol lapisan aplikasi membutuhkan layanan transfer data yang bersifat andal, oleh sebab itu layanan tersebut tidak dimiliki oleh protokol lain [12]. TCP mempunyai peran penting untuk bereaksi terhadap *packet loss* dan mengambil tindakan untuk mengurangi kongesti. Oleh karena itu TCP berperan dalam memperbaiki pengiriman data dari suatu *client* ke *server* [13]. Contoh aplikasi TCP adalah *Hypertext Transfer Protocol Secure* (HTTPS), *Simple Mail Transfer Protocol* (SMTP) dan *File Transfer Protocol* (FTP) [14]. Jenis varian TCP memiliki mekanisme kerja masing-masing untuk mengatasi permasalahan ketika terjadi kongesti dan juga berperan penting dalam baik dan buruknya kinerja dalam sebuah jaringan [7]. TCP ini salah satu protokol yang bekerja pada layer *transport* pemodelan *Open System Interconnection* (OSI) [4].

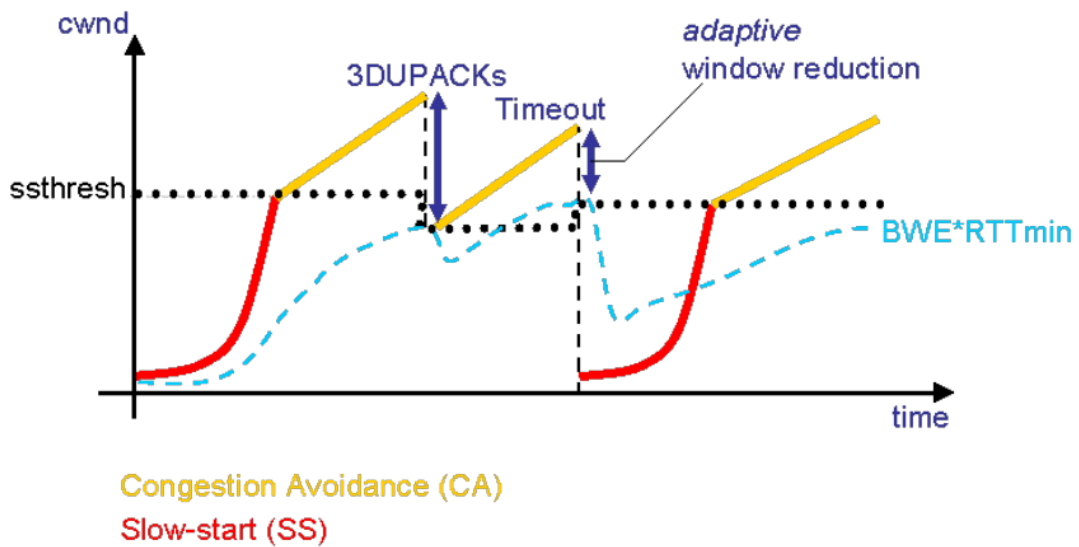
Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical

Gambar 2.1 Model OSI

2.2.3 TCP Westwood+

TCP *Westwood+* adalah salah satu dari berbagai macam varian algoritma TCP yang dikembangkan oleh Saveiro Mascolo beserta kawan-kawannya di *Universitas California, Los Angeles* (UCLA). TCP *Westwood+* pengembangan dari TCP *New Reno* dan hal yang membedakan TCP *Westwood+* dengan TCP *New Reno* adalah pada mekanisme kontrol kongestinya [15].

Additive Increase Multiplicative Decrease (AIMD) pada TCP *New Reno* diganti dengan *Additive Increase Adaptive Decrease* (AIAD) ke TCP *Westwood+* guna mencegah pemotongan *transfer-rate* yang berlebihan ketika terjadi kongesti [2]. TCP *New Reno* menggunakan *Additive Increase Multiplicative Decrease* (AIMD) yaitu dengan menurunkan *slow start threshold* (*sssthres*) dan *congestion windows* (*cwnd*) menjadi setengah pada saat terjadi kongesti sehingga pita yang dipakai menjadi menurun drastis [16]. Sedangkan pada TCP *Westwood+* menggunakan *Additive Increase Adaptive Decrease* (AIAD) pada saat terjadi kongesti paket yang hilang tidak akan direspon atau diturunkan secara ekstrim karena nilai *sssthres* dan *cwnd* akan diestimasi secara adaptif berdasarkan *bandwidth end-to-end* pada jaringan [17]. TCP *Westwood+* ini menggunakan estimasi *bandwidth* dari *Acknowledgment* (ACK) untuk mengatur nilai *congestion windows* (*cwnd*) dan *slow start threshold* (*sssthresh*) [16]. TCP *Westwood+* mempunyai kelebihan dalam hal mengatasi jaringan dengan *loss* tinggi dan *delay* yang panjang dan tingkah laku TCP *Westwood+* ini bersifat adaptif [18]. Pada Gambar 2.2 merupakan mekanisme kontrol kongesti dari TCP *Westwood+*.



Gambar 2.2 Mekanisme Kontrol Kongesti TCP *Westwood+* [16]

2.2.4 Kongesti

Masalah yang sering terjadi pada suatu jaringan adalah kongesti [19]. Menurut *Microsoft Network (MSN) Encarta* kongesti dalam komputasi adalah sebagai kondisi dimana jumlah informasi yang di kirim lebih besar dibandingkan dengan kemampuan jalur komunikasi. [20]. Secara garis besar suatu jaringan dikatakan mengalami kongesti apabila kualitas layanan yang dirasakan oleh *end-host* mengalami penurunan akibat dari beban jaringan meningkat. Hal ini terjadi karena paket-paket yang dikirimkan melalui jaringan mendekati jumlah paket maksimum yang diijinkan dan sumber jaringan dibebani dengan beban yang melebihi batas jaringan itu sendiri [21]. Dengan banyaknya pengguna yang berbagi *bandwidth* namun jumlah permintaan akan *bandwidth* melebihi kapasitas *link* yang tersedia akibatnya performa jaringan akan menurun [21]. Kontrol kongesti difokuskan pada lalu lintas data yang masuk ke sebuah jaringan telekomunikasi dan untuk menghindari kegagalan dalam pengiriman data, dilakukan pengurangan tingkat paket yang akan dikirimkan [21].

2.2.5 NS-3 (*Network Simulator 3*)

Network Simulator 3 (NS-3) adalah sebuah simulator jaringan yang biasanya digunakan dan memiliki ciri tersendiri yang ditargetkan secara utama untuk tujuan riset dan pendidikan. Proyek NS-3 dimulai pada tahun 2006, NS-3 ini bersifat *open source* dibawah *Gnu Public License* (GPL) [22]. Dengan NS-3 ini penelitian mengenai jaringan simulasi menjadi lebih mudah, bahasa yang digunakan adalah bahasa C++ dilapisan inti dan *script python* [23]. Karena bersifat *open source* dengan NS-3 banyak struktur dinamik jaringan komunikasi yang dapat dipelajari dan dengan simulator NS-3 ini simulasi jaringan nirkabel dan protokol seperti algoritme routing, *User Datagram Protocol* (UDP), dan *Transmission Control Protocol* TCP dapat diselesaikan dengan baik [15]. Lingkungan pengembangan NS-3 ini bersifat terbuka dan setiap user dapat berkontribusi untuk mengembangkan dan menambahkan fungsi-fungsi baru didalam NS-3 [24].

2.2.6 FTP (*File Transfer Protocol*)

FTP adalah singkatan dari *File Transfer Protocol*, dimana FTP ini adalah salah satu protokol internet yang digunakan untuk urusan penerimaan dan pengiriman data dalam jaringan komputer, seperti *upload* dan *download* yang dilakukan oleh *FTP client* dan *FTP server* [25]. FTP ini merupakan salah satu contoh aplikasi dari *Transmission Control Protocol* (TCP) karena FTP menggunakan TCP untuk komunikasi data yang dilakukan antara *client* dengan *server* [14]. FTP banyak digunakan karena kemudahannya dalam proses pertukaran data [26].

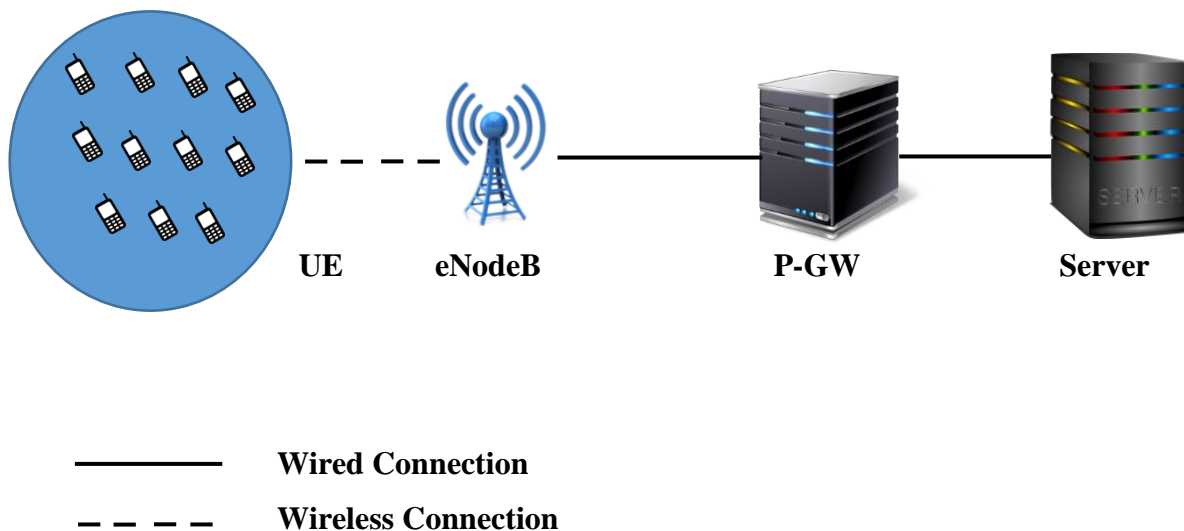
BAB 3

METODOLOGI

3.1 Mekanisme Sistem Perancangan

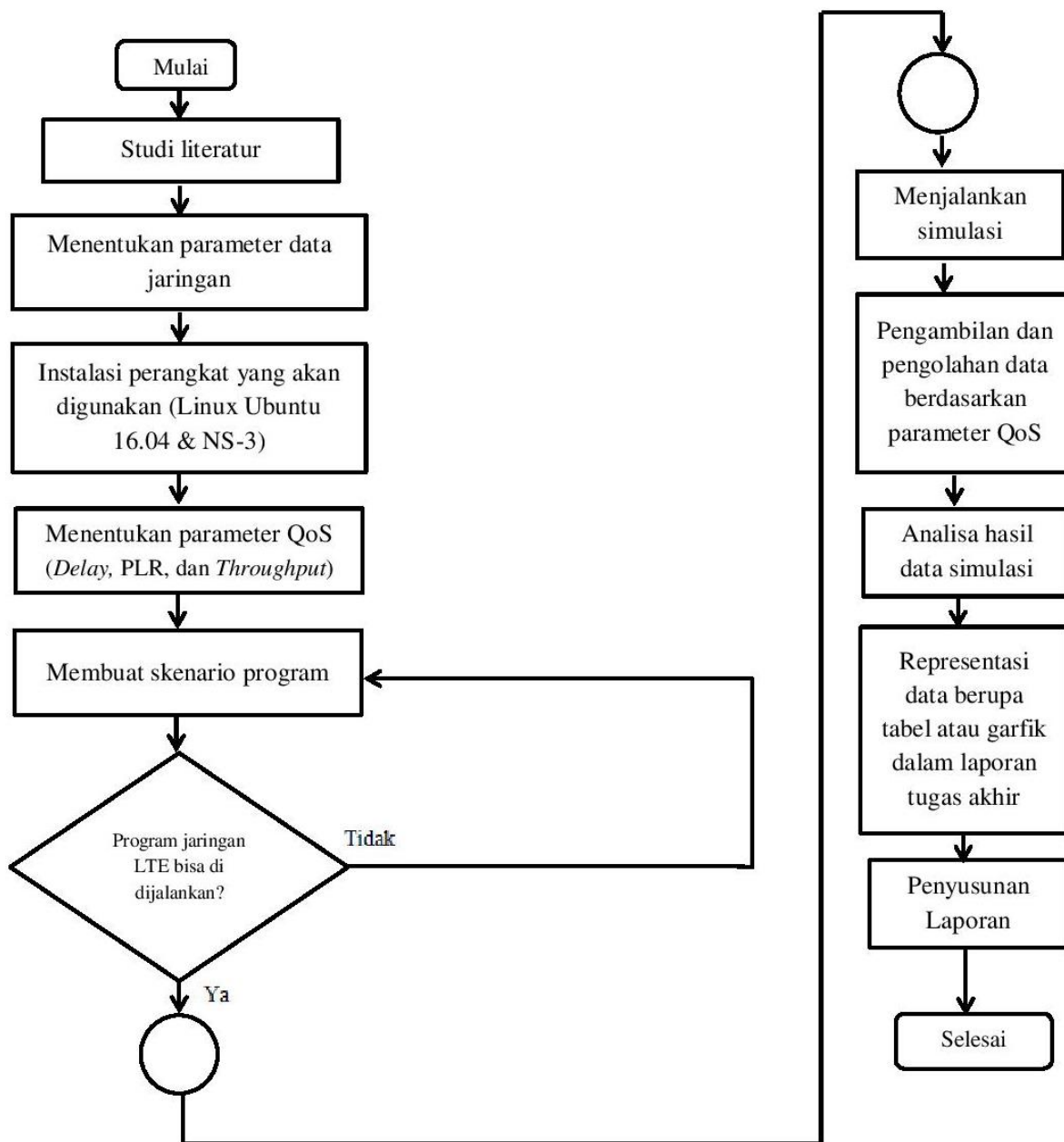
Pada penelitian ini akan mensimulasikan jaringan LTE yang akan diimplementasikan TCP *Westwood+* menggunakan perangkat lunak *Network Simulator-3* (NS-3). Ada tiga pembahasan utama yang akan di analisis yakni : *delay*, *Packet Delivery Ratio* (PLR), dan *throughput*.

Untuk topologi yang digunakan dalam simulasi penelitian ini menggunakan satu jenis topologi saja dengan menggunakan 1 eNodeB, 1 P-GW/gateway dan 1 *server*. Jumlah untuk *User Equipment* (UE) yang digunakan adalah 100, 120, 140, 160, 180, 200 pengguna. Pada Gambar 3.1 menunjukkan topologi simulasi yang akan digunakan dalam penelitian ini.



Gambar 3.1 Topologi Simulasi

Adapun *flowchart* metodologi penelitian secara keseluruhan seperti pada Gambar 3.2 berikut ini.



Gambar 3.2 *Flowchart* Metodologi Penelitian

3.2 Studi Literatur

Dalam proses ini penulis mencari dan mempelajari jurnal-jurnal untuk dijadikan sumber acuan dan referensi dalam melakukan penelitian ini. Dalam penelitian ini parameter-parameter yang diterapkan banyak didapatkan dari jurnal maupun *paper*.

3.3 Menentukan Parameter Jaringan

Pada penelitian ini menentukan parameter-parameter yang akan diterapkan baik itu pada parameter jaringan LTE maupun parameter menjalankan simulasi, dan dapat ditunjukkan pada Tabel 3.1 :

Tabel 3.1 Parameter Simulasi

Parameter	Nilai
Teknologi Akses	<i>Downlink</i> = OFDMA
	<i>Uplink</i> = SC-FDMA
Model Propagasi	COST-231
Model Radio Akses	FDD (<i>Frequency Division Duplexing</i>)
<i>Frequensi</i>	1.8 GHz
<i>Bandwidth</i>	5 MHz
<i>Data Rate</i>	11 Mbps
Jumlah <i>Node</i> Pengguna	100, 120, 140, 160, 180, 200
Posisi <i>Node</i>	<i>Random Waypoint Position</i>
Dimensi Topologi	250m x 250m
Jenis Layanan Data	FTP (<i>File Transfer Protocol</i>)
Durasi Simulasi	60 s
<i>Delay</i> Transmisi	0.01 s
Mekanisme Kontrol Kongesti	Algoritme TCP <i>Westwood+</i>

3.4 Instalasi Perangkat

Penelitian ini membutuhkan *Operating System* (OS) Linux Ubuntu untuk proses simulasi dengan menggunakan *software Network Simulator-3* (NS-3) yang sudah terinstall didalamnya. Versi Ubuntu yang digunakan adalah versi 16.04 dan untuk NS-3 sendiri menggunakan versi NS-3.27.

3.5 Menentukan Parameter QoS (*Quality of Services*) yang dianalisis

Pengukuran dalam kinerja jaringan memerlukan parameter-parameter dan ini yang dinamakan *Quality of service* (QoS) [27]. Kinerja jaringan adalah tingkat dimana pencapaian yang terukur mengenai seberapa baik jaringan untuk mendefinisikan karakteristik dan sifat dari suatu layanan dan kinerja dalam jaringan dapat diukur dengan mengetahui QoS [28]. QoS di definisikan sebagai sebuah cara atau mekanisme yang memungkinkan layanan dapat berjalan sesuai dengan karakteristiknya masing-masing dalam sebuah jaringan [29]. Dengan QoS menunjukkan kemampuan jaringan tersebut menyediakan layanan yang lebih baik bagi trafik yang melewatinya [30]. QoS mengacu dalam kemampuan suatu jaringan tertentu dengan teknologi yang berbeda-beda. Berikut beberapa parameter QoS yang dapat dilihat untuk menentukan kualitas suatu jaringan, yakni [30] :

- ***Delay***

Delay adalah selang waktu antara paket pertama dikirim dengan paket yang terakhir diterima/waktu yang dibutuhkan data untuk menempuh jarak dari asal sampai ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, dan kongesti atau juga waktu proses yang lama. Selain itu adanya antrian atau mengambil rute lain untuk menghindari kemacetan juga dapat mempengaruhi *delay* [31]. Berikut ini persamaan *delay* yang dipakai pada simulasi penelitian ini :

$$Delay = \frac{delaySum}{rxPackets} \quad (3.1)$$

Keterangan :

delaySum = Waktu paket tiba

rxPackets = Paket yang diterima

- **PLR (*Packet Loss Ratio*)**

Packet Loss Ratio (PLR) adalah jumlah paket yang gagal atau hilang selama proses pengiriman dari *node* asal ke *node* tujuan. Paket yang hilang atau gagal sampai disebabkan beberapa faktor, seperti perlemahan kekuatan sinyal saat proses transmisi, paket yang rusak, kegagalan perangkat jaringan, *collision* dan *congestion* pada jaringan. PLR diukur dalam persen (%) [28]. Berikut persamaan *Packet Loss Ratio* yang dipakai pada simulasi penelitian ini :

$$PLR = \frac{txPackets - rxPackets}{txPackets} \times 100\% \quad (3.2)$$

Keterangan :

txPackets = Paket yang dikirim

rxPackets = Paket yang diterima

- ***Throughput***

Throughput adalah laju data aktual persatuan waktu dimana jumlah total kedatangan paket yang berhasil yang diamati pada tujuan selama interval waktu tertentu dibagi dengan durasi *interval* waktu tersebut. *Throughput* sendiri dapat diartikan sebagai kecepatan (*rate*) transfer data efektif [32]. Berikut ini persamaan *throughput* yang dipakai pada simulasi penelitian ini :

$$Throughput = rxBytes / (Time\ last\ RxPacket - Time\ first\ TxPacket) \quad (3.3)$$

Keterangan :

rxBytes = Paket yang diterima (*bytes*)

Time last RxPacket = Waktu terakhir paket diterima

Time first TxPacket = Waktu pertama pengiriman paket

Berikut ini adalah parameter QoS berdasarkan standar *Telecommunications and Internet Protocol Harmonization Over Networks* (TIPHON) dibawah naungan *European Telecommunications Standards Institute* (ETSI) yang ditunjukkan pada Tabel 3.2 :

Tabel 3.2 Parameter QoS [33]

Kategori Degredasi	<i>Packet Loss Ratio</i> (PLR)	Besar <i>Delay</i>	Indeks
Sangat Bagus	0	<150ms	4
Bagus	3%	150 s/d 300ms	3
Sedang	15%	300 s/d 450ms	2
Buruk	25%	>450ms	1

3.6 Membuat Skenario Program

Dalam penelitian ini diterapkan 2 skenario, yakni :

1. Skenario 1 dengan simulasi jaringan LTE tanpa algoritme TCP *Westwood+* dengan waktu simulasi 60s menggunakan jumlah *node* pada percobaan ke-1 100 *node*, percobaan ke-2 120 *node*, percobaan ke-3 140 *node*, percobaan ke-4 160 *node*, percobaan ke-5 180 *node*, dan percobaan ke-6 200 *node*.
2. Untuk skenario 2 simulasi jaringan LTE dengan algoritme TCP *Westwood+* dengan waktu simulasi yang sama yaitu 60s menggunakan jumlah *node* pada percobaan ke-1 100 *node*, percobaan ke-2 120 *node*, percobaan ke-3 140 *node*, percobaan ke-4 160 *node*, percobaan ke-5 180 *node*, dan percobaan ke-6 200 *node*.

3.7 Menjalankan Simulasi

Pada proses ini menjalankan simulasi pada *Network Simulator-3* (NS-3) dengan skenario 1 dan skenario 2 yang sudah ditentukan oleh penulis. Jika simulasi berhasil maka dilanjutkan ke proses selanjutnya yaitu pengambilan data, tetapi jika simulasi dianggap gagal maka dilakukan proses simulasi ulang hingga hasil simulasi yang didapatkan berhasil.

3.8 Pengambilan Data

Ketika proses simulasi dengan skenario 1 dan skenario 2 yang sudah ditentukan oleh penulis sudah selesai, maka penulis mendapatkan data berdasarkan parameter QoS. Data tersebut diambil dan kemudian data tersebut dianalisis.

3.9 Analisis Hasil Data Simulasi

Pada proses ini, penulis mendapatkan hasil simulasi yang sudah dijalankan dengan menggunakan skenario 1 dan skenario 2, maka tahap selanjutnya adalah hasil data simulasi tersebut dianalisis perbandingan kinerja antara skenario 1 dengan skenario 2.

3.10 Representasi Data

Pada tahap representasi data ini, hasil dari simulasi yang sudah dijalankan dengan menggunakan skenario 1 dan skenario 2 yang sudah dianalisis diubah kedalam bentuk grafik dan sebagai gambaran perbandingan kinerja antara skenario 1 dengan skenario 2. Hasil representasi data berupa grafik ini diproses selanjutnya ke penyusunan laporan.

3.11 Penyusunan Laporan

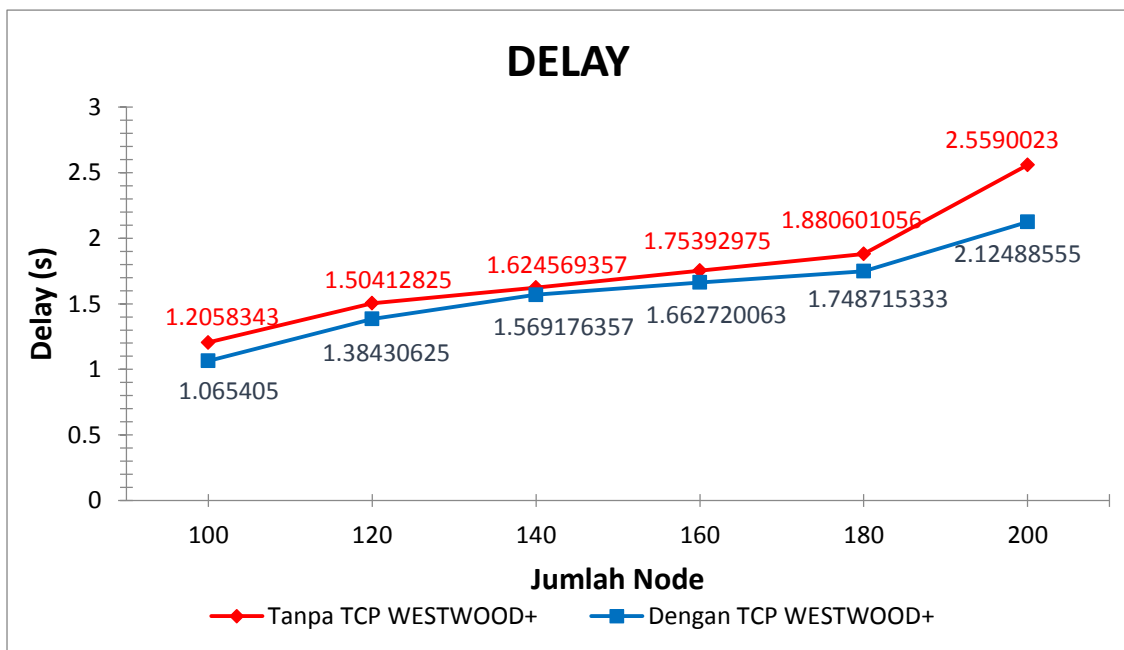
Pada tahapan akhir ini dimana hasil yang didapatkan berupa grafik perbandingan kinerja antara skenario 1 dengan skenario 2 kemudian disusun ke bentuk laporan. Penyusunan laporan ini berisikan awal mula penulis mencari studi literatur hingga mendapatkan hasil simulasi yang sudah dilakukan.

BAB 4

HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk mengetahui perbandingan kinerja antara tanpa implementasi TCP *Westwood+* dan dengan implementasi TCP *Westwood+* guna melihat kinerja kontrol kongesti pada jaringan LTE. Skenario yang akan diujikan adalah melihat perbandingan pada jaringan LTE apabila tanpa mekanisme kontrol kongesti algoritme TCP *Westwood+* dan dengan algoritme TCP *Westwood+* dengan waktu simulasi 60s dengan jumlah *node* disetiap percobaan adalah 100, 120, 140, 160, 180, dan 200 *node*.

4.1 Perbandingan Delay



Gambar 4.1 Perbandingan *Delay*

Dapat dilihat pada Gambar 4.1 *delay* yang dihasilkan tanpa TCP *Westwood+* dan dengan TCP *Westwood+* berbeda, dengan *delay* yang dihasilkan tanpa TCP *Westwood+* lebih besar dibandingkan dengan TCP *Westwood+* dan *delay* yang dihasilkan dari *node* 100 hingga 200 cenderung meningkat. Untuk kenaikan rata-rata *Delay* dapat dilihat dalam Tabel 4.1 dan perbedaan *delay* pada Tabel 4.2.

Tabel 4.1 Kenaikan Rata-Rata *Delay*

Kenaikan Rata-Rata <i>Delay</i>				
Node	Tanpa TCP <i>Westwood+</i> (s)	Dengan TCP <i>Westwood+</i> (s)	Rata – Rata (s)	Rata – Rata (%)
100 Ke- 120	0.29829395	0.31890125	0.3085976	30%
120 Ke- 140	0.120441107	0.184870107	0.152655607	15%
140 Ke- 160	0.129360393	0.093543706	0.11145205	11%
160 Ke- 180	0.126671306	0.08599527	0.106333288	10%
180 Ke- 200	0.678401244	0.376170217	0.527285731	52%
Total Rata-Rata	0.2706336	0.21189611	0.241264855	24%

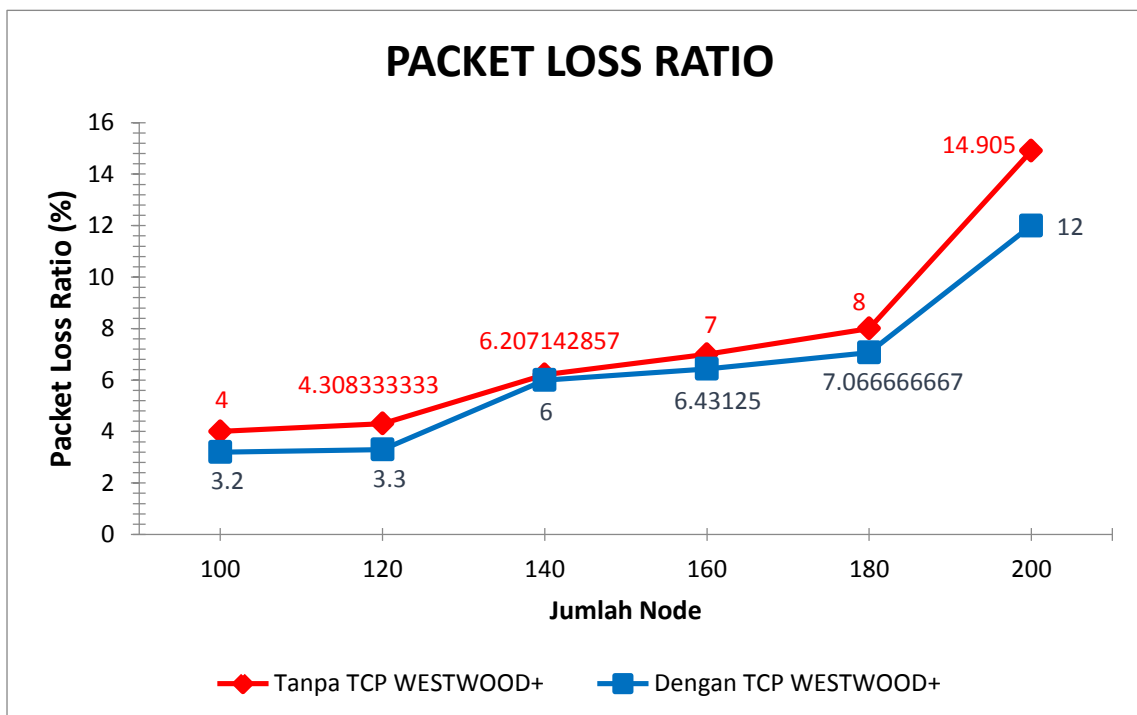
Tabel 4.2 Perbedaan *Delay*

Perbedaan <i>Delay</i> Tanpa TCP <i>Westwood+</i> dan Dengan TCP <i>Westwood+</i>		
Node	Perbedaan (s)	Perbedaan (%)
100	0.1404293	14%
120	0.119822	11%
140	0.055393	5%
160	0.091209687	9%
180	0.131885723	13%
200	0.43411675	43%
Total Rata-Rata	0.162142743	16%

Dari hasil perbandingan *delay* antara tanpa TCP *Westwood+* dan dengan TCP *Westwood+* dapat dilihat dari jumlah *node* 100 *delay* tanpa TCP *Westwood+* lebih besar dibandingkan *delay* dengan TCP *Westwood+* dengan kenaikan awal memang tanpa TCP *Westwood+* lebih kecil dibandingkan dengan TCP *Westwood+* namun ketika dengan penambahan jumlah *node* kenaikan rata-rata *delay* tanpa TCP *Westwood+* semakin besar sedangkan dengan TCP *Westwood+* semakin menurun dan dapat dilihat dari Tabel 4.1. Ini karena teknik kontrol kongesti *Additive Increase Adaptive Decrease* (AIAD) pada TCP *Westwood+* dimana akan menyesuaikan dengan jumlah *node* yang ada dan akan menurunkan *delay* dengan jumlah *node* yang ada. Dapat disimpulkan bahwa tanpa TCP *Westwood+*, *delay* yang dihasilkan lebih besar dibandingkan dengan menggunakan TCP *Westwood+* ini karena pada TCP *Westwood+* sistem *Additive Increase Adaptive Decrease* (AIAD) pada saat terjadi kongesti paket yang hilang tidak akan direspon atau diturunkan secara ekstrim karena nilai *ssthres* dan *cwnd* akan diestimasi

secara adaptif berdasarkan *bandwidth end-to-end* pada jaringan sehingga *delay* menjadi berkurang. *Delay* yang dihasilkan tanpa TCP *Westwood+* lebih besar 0.162142743s dibandingkan *delay* yang dihasilkan dengan TCP *Westwood+*. Namun menurut standar parameter THIPON pada Tabel 3.3 *delay* yang dihasilkan kedua perbandingan pada semua jumlah *node* masih tergolong bagus.

4.2 Perbandingan PLR (*Packet Loss Ratio*)



Gambar 4.2 Perbandingan *Packet Loss Ratio*

Dapat dilihat pada Gambar 4.2 merupakan hasil PLR dari simulasi yang sudah dilakukan tanpa TCP *Westwood+* dan dengan TCP *Westwood+*. Hasil perbandingan PLR tanpa TCP *Westwood+* memiliki PLR yang lebih besar dibandingkan dan dengan TCP *Westwood+* dari jumlah *node* 100 hingga 200. Untuk kenaikan rata-rata PLR tanpa TCP *Westwood+* dan dengan TCP *Westwood+* dapat dilihat dalam Tabel 4.3 dan untuk perbedaan PLR sendiri dapat dilihat dalam Tabel 4.4.

Tabel 4.3 Kenaikan Rata-Rata PLR

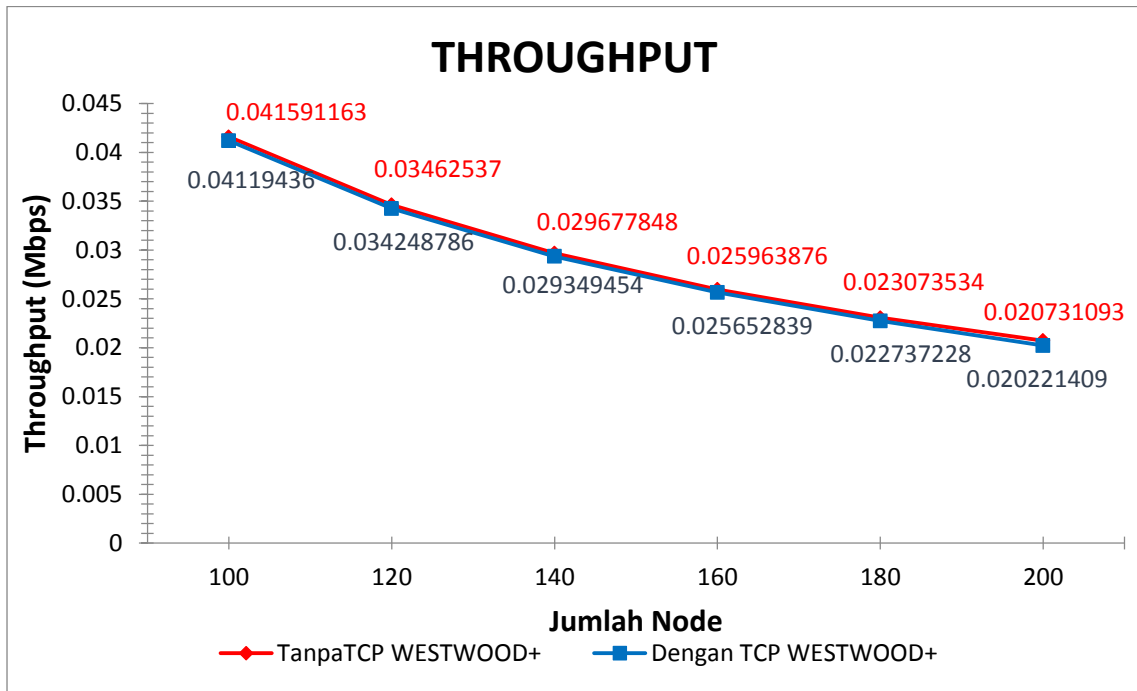
Kenaikan Rata-Rata PLR			
Node	Tanpa TCP <i>Westwood+</i> (%)	Dengan TCP <i>Westwood+</i> (%)	Rata – Rata (%)
100 Ke- 120	0.308333333	0.1	0.204166667
120 Ke- 140	1.898809524	2.7	2.299404762
140 Ke- 160	0.792857143	0.43125	0.612053572
160 Ke- 180	1	0.635416667	0.817708334
180 Ke- 200	6.905	4.933333333	5.919166667
Total Rata-Rata	2.181	1.76	1.9705

Tabel 4.4 Perbedaan PLR

Perbedaan PLR Tanpa TCP <i>Westwood+</i> dan Dengan TCP <i>Westwood+</i>	
Node	Perbedaan (%)
100	0.8
120	1.008333333
140	0.207142857
160	0.56875
180	0.933333333
200	2.905
Total Rata-Rata	1.070426587

Penambahan jumlah *node* sangat mempengaruhi hasil kinerja PLR, dari hasil perbedaan PLR ini pada jumlah *node* 100 ke jumlah *node* 120 tanpa TCP *Westwood+* dan dengan TCP *Westwood+* mengalami peningkatan, kenaikan yang terjadi tanpa TCP *Westwood+* lebih besar dibandingkan dengan TCP *Westwood+*, ini karena sifat TCP *Westwood+* yang bersifat adaptif dimana ketika menaikkan PLR yang lumayan besar TCP *Westwood+* akan menyesuaikan diri dengan jumlah *node* selanjutnya dimana nilai kenaikan rata-rata PLR akan menyesuaikan dengan jumlah *node*. Meskipun demikian perbedaan hasil rata-rata tanpa TCP *Westwood+* memiliki nilai PLR yang lebih tinggi 1.070426587% dibandingkan dengan TCP *Westwood+*, dan menurut standar parameter THIPON pada Tabel 3.4 PLR yang dihasilkan ini tergolong sedang.

4.3 Perbandingan Throughput



Gambar 4.3 Perbandingan *Throughput*

Dapat dilihat pada Gambar 4.3 adalah hasil *throughput* yang telah diperoleh dari hasil percobaan simulasi tanpa TCP *Westwood+* dan dengan TCP *Westwood+*. Hasil yang didapat tanpa TCP *Westwood+* sedikit lebih besar dibandingkan dengan TCP *Westwood+*. Dari jumlah *node* 100 hingga 200 *throughput* yang dihasilkan menurun, ini karena penambahan jumlah *node* yang dilakukan. Untuk melihat penurunan rata-rata *throughput* dapat dilihat dalam Tabel 4.5 dan untuk melihat perbedaan *throughput* dapat dilihat dalam Tabel 4.6.

Tabel 4.5 Penurunan Rata-Rata *Throughput*

Penurunan Rata-Rata <i>Throughput</i>				
Node	Tanpa TCP <i>Westwood+</i> (Mbps)	Dengan TCP <i>Westwood+</i> (Mbps)	Rata – Rata (Mbps)	Rata – Rata (%)
100 Ke- 120	0.006965793	0.006945574	0.006955684	0.6%
120 Ke- 140	0.004947522	0.004899332	0.004923427	0.4%
140 Ke- 160	0.003713972	0.003696615	0.003705294	0.3%
160 Ke- 180	0.002890342	0.002915611	0.002902977	0.2%
180 Ke- 200	0.002342441	0.002515819	0.00242913	0.2%
Total Rata-Rata	0.004172014	0.00419459	0.004183302	0.4%

Tabel 4.6 Perbedaan *Throughput*

Perbedaan <i>Throughput</i> Tanpa TCP <i>Westwood+</i> dan Dengan TCP <i>Westwood+</i>		
Node	Perbedaan (Mbps)	Perbedaan (%)
100	0.000396803	0.039%
120	0.000376584	0.037%
140	0.000328394	0.032%
160	0.000311037	0.031%
180	0.000336306	0.033%
200	0.000509684	0.050%
Total Rata-Rata	0.000376468	0.037%

Dapat disimpulkan dari hasil perbandingan *throughput* dapat dilihat pada grafik hasil *throughput* tanpa TCP *Westwood+* lebih besar dibandingkan dengan TCP *Westwood+* ini karena TCP *Westwood+* mengatasi jaringan dengan *loss* tinggi dan *delay* yang panjang dan TCP *Westwood+* dioptimalkan untuk kehandalan komunikasi bukan pada kecepatan, meskipun perbedaan *throughput* tanpa TCP *Westwood+* dengan TCP *Westwood+* sebesar 0.000376468 Mbps, dan perbedaanya ini tergolong sangat rendah.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan penelitian ini, maka dapat diambil kesimpulan bahwa :

1. Semakin banyak jumlah *node* mempengaruhi kinerja jaringan LTE tanpa TCP *Westwood+* dan dengan TCP *Westwood+*.
2. *Delay* yang dihasilkan tanpa TCP *Westwood+* lebih besar 0.162142743s dibandingkan *delay* yang dihasilkan dengan TCP *Westwood+*. Kenaikan rata-rata pada *delay* tanpa TCP *Westwood+* lebih besar 0.05873749s dibandingkan dengan kenaikan rata-rata dengan TCP *Westwood+*.
3. PLR yang dihasilkan dengan TCP *Westwood+* lebih bagus 1.070426587% daripada PLR yang dihasilkan tanpa TCP *Westwood+*. Kenaikan rata-rata pada PLR tanpa TCP *Westwood+* lebih besar 0.421% dibandingkan dengan kenaikan rata-rata dengan TCP *Westwood+*.
4. Dari grafik *throughput* tanpa TCP *Westwood+* lebih tinggi dari dengan TCP *Westwood+* dengan perbedaan *throughput* tanpa TCP *Westwood+* dengan TCP *Westwood+* sebesar 0.000376468 Mbps, dan perbedaanya ini tergolong sangat rendah. Penurunan rata-rata pada *throughput* tanpa TCP *Westwood+* lebih kecil 0.000022527 Mbps dibandingkan penurunan rata-rata dengan TCP *Westwood+*.
5. Implementasi Algoritme TCP *Westwood+* lebih bagus dibandingkan dengan tanpa TCP *Westwood+* dari segi *loss* dan *delay* walaupun perbedaan pada *throughput* terbilang sangat rendah, dengan menggunakan algoritme TCP *Westwood+* ini juga kongesti dapat dikurangi.

5.2 Saran

Setelah melakukan penelitian ini, penulis memiliki beberapa saran untuk penelitian selanjutnya agar mampu menambah performa dari sistem ini :

1. Untuk mengetahui lebih bagusnya TCP *Westwood+* dapat mengontrol kongesti lebih baik lagi diharapkan penelitian selanjutnya apabila dari penelitian ini dikembangkan, menggunakan *node* yang lebih banyak lagi dan dengan ruang lingkup yang lebih besar juga.
2. Penelitian ini menggunakan jenis trafik *File Transfer Protocol* (FTP) dan untuk kedepannya agar menggunakan jenis trafik yang lebih beragam.

DAFTAR PUSTAKA

- [1] G. S. Mahanani, “Analisis dan Pengujian di Jaringan 3G Dan 4G dalam Layanan Quality Of Services (QOS),” Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga : Yogyakarta, 2016.
- [2] W. Adinugroho, I. Wahidah, and R. Rumani, “Analisis Perbandingan Performansi Kontrol Kongesti TCP Westwood+ Dengan TCP New Reno,” Fakultas Teknik Elektro Universitas Telkom : Bandung, 2008.
- [3] R. A. Hamamreh and M. J. Bawatna, “Protocol for Dynamic Avoiding End-to-End Congestion in MANET,” *Journal of Wireless Networking and Communications*, vol. 4, no. 3, Computer Engineering Department Al-Quds University : Jerusalem, Palestine, pp. 67–75, 2014.
- [4] M. Taruk and H. J. Setyadi, “Analisis Mekanisme Penanganan Kemacetan (Congestion Control) pada Algoritma Varian,” *Konferensi Nasional Ilmu Komputer (KONIK)*, Fakultas Teknologi Informasi Institut Teknologi Surabaya : Surabaya, pp. 0–4, 2016.
- [5] M. F. Fitrianto, “Analisis Kinerja TCP BIC untuk Pencegahan Kongesti pada Jaringan LTE Dengan Menggunakan Network Simulator 2.33,” Fakultas Teknik Universitas Diponegoro : Semarang, pp. 1–8, 2014.
- [6] G. K. Permatasari, Sukiswo, and I. Santoso, “Analisis Kinerja TCP Westwood untuk Pencegahan Kongesti pada Jaringan LTE dengan Menggunakan Network Simulator 2.33 (NS2.33),” vol. 33, no. 4, Fakultas Teknik Universitas Diponegoro : Semarang, 2014.
- [7] A. T. Harfad, S. R. Akbar, and A. Bhawiyuga, “Analisa Kinerja Algoritma TCP Congestion Control Cubic, Reno, Vegas dan Westwood+,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 3, Fakultas Ilmu Komputer Universitas Brawijaya : Malang, pp. 1099–1108, 2018.

- [8] U. Shabrina and W. Broto, "Studi Perkembangan Teknologi 4G – LTE dan WiMAX di Indonesia," *Prosiding Seminar Nasional Fisika (E-Journal) SNF2017*, vol. 9, no. 2, Fakultas Teknik Elektro Universitas Pancasila : Jakarta, pp. 60–65, 2017.
- [9] A. Marhadi, "Perencanaan Jaringan Long Term Evolution (LTE) Frekuensi 1800MHz di Jembatan Suramadu dengan Physical Cell Identity (PCI)," Fakultas Teknik Elektro Universitas Telkom : Surabaya, 2016.
- [10] D. Riyansyah, "Analisa Kelayakan Migrasi BTS 3G Berbasis WCDMA Menuju Jaringan LTE di DKI Jakarta," Fakultas Teknik Universitas Indonesia : Jakarta, 2010.
- [11] D. K. A. Saputro, "Analisis Perencanaan Jaringan LTE di Pita Frekuensi 3500 MHz dengan Mode TDD dan FDD," *Jurnal Telekomunikasi dan Komputer*, Universitas Mercu Buana : Yogyakarta, 2016.
- [12] H. P. Siregar and N. Mubarakah, "Analisis Kinerja Transmission Control Protocol pada Jaringan Wide Area Network," *Jurnal SINGUDA ENSIKOM*, vol. 12, Fakultas Teknik Universitas Sumatera Utara : Medan, pp. 33–38, 2015.
- [13] A. Cahyaningtyas, "Monitoring Layer Transport (Protokol TCP dan UDP) menggunakan Wireshark," *Komunitas eLearning IlmuKomputer.Com*, Fakultas Teknik Telekomunikasi Politeknik Negeri Semarang : Semarang, pp. 1–6, 2007.
- [14] A. Dwiyanakuntoko, "Membandingkan Protokol UDP dan TCP," *Komunitas eLearning IlmuKomputer.Com*, Fakultas Teknik Telekomunikasi Politeknik Negeri Semarang : Semarang, pp. 1–6, 2007.
- [15] R. Satria, "Analisis Perbandingan Kinerja Protokol TCP New Reno dan Westwood+ pada Jaringan WiMAX IEEE 802.16," Fakultas Teknik Universitas Widyatama : Bandung, pp. 1–16, 2013.
- [16] I. D. Octavianty, "Analisa dan Simulasi Perbandingan Algoritma Westwood dan Algoritma Selective Acknowledgment Option pada Sistem Kendali Kongesti Jaringan TCP," Fakultas Teknik Universitas Muhammadiyah Malang : Malang, pp. 10–35, 2012.

- [17] T. Rachmanto, "Simulasi Kinerja Protokol Tcp pada Jaringan WiMAX Menggunakan Network Simulator 2 (NS-2)," vol. 2, Fakultas Teknik Universitas Diponegoro : Semarang, pp. 1–12, 2004.
- [18] H. H. Nuha and F. Arif, "Metode Pengaturan Throughput untuk TCP Westwood+ pada Saluran Bottleneck," no. 1, Pasca Sarjana Teknik Informatika IT Telkom : Bandung, 2015.
- [19] M. Said, "Waktu Tunda Versus Loss pada Mira dan Lmir dalam Menghadapi Kongesti pada Jaringan MPLS," *Jurnal JETC*, vol. 4, no. 2, Universitas Khairun : Ternate, 2010.
- [20] P. H. Rusmin, C. Machbub, and A. Harsoyo, "Sistem Kendali Kongesti Di Internet," *Jurnal MAKARA TEKNOLOGI*, vol. 12, no. 1, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung : Bandung, pp. 19–26, 2008.
- [21] R. Inelia, S. N. Hertiana, and I. Wahidah, "Analisis dan Simulasi Performansi Kendali Kongesti dengan Skema PGMCC," Fakultas Teknik Elektro Universitas Telkom : Bandung, pp. 1–15, 2011.
- [22] P. S. Katkar, "Comparative Study of Network Simulator : NS2 and NS3," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 6, no. 3, Department of Computer Science Shivaji University : Kolhapur, Maharashtra, India, pp. 608–612, 2016.
- [23] R. K. Jha and P. Kharga, "Advanced Open Source Simulator : NS-3," *International Journal of Computer Sciences and Engineering (e-IJCSE: 2347-269)*, no. 12, Dept. of Electronics & Communication Engineering Shri Mata Vaishno Devi University : Katra, Jammu & Kashmir, India, 2015.
- [24] G. P. D. Setyo, L. Vidya, and T. A. Wibowo, "Perancangan dan Emulasi Protokol Routing Epidemic dan Static pada Jaringan DTN," *Journal e-Proceeding of Engineering*, vol. 4, no. 1, Fakultas Teknik Elektro Universitas Telkom : Bandung, pp. 1–15, 2017.
- [25] C. Chung, "An Introduction to FTP," Kinecton, Canada : 2BrightSparks, p. 4, 2014.

- [26] I. W. A. Sapura and C. R. A. Pramatha, "Perancangan FTP (File Transfer Protocol) Melalui SCTP (Stream Control Transmission Protocol) Menggunakan Socket Programming," vol. 1, no. 1, Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Udayana : Bali, 2012.
- [27] N. Jiatmiko and Y. Prayudi, "Simulasi Jaringan MANET Dengan NS3 Untuk Membandingkan Performa Routing Protokol AODV dan DSDV Simulasi Jaringan MANET Dengan NS3 Untuk Membandingkan Performa Routing Protokol AODV dan DSDV," Fakultas Teknologi Industri Universitas Islam Indonesia : Yogyakarta, pp. 44–55, 2015.
- [28] S. A. Endarsa, "Analisa Kinerja Jaringan pada Internet Connection Sharing menggunakan Virtual Access Point dan Real Access Point," Fakultas Sains dan Teknologi Universitas Sanata Dharma : Yogyakarta, p. 108, 2013.
- [29] R. T. Y. Anggraeni and A. Sukmaaji, "Analisis Perbandingan Unjuk Kerja Protokol TCP, UDP, DAN SCTP Menggunakan Simulasi Lalu Lintas Data Multimedia," Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya Kedung Baruk Surabaya : Surabaya, pp. 1–7, 2009.
- [30] H. Fitriawan and A. Wahyudin, "Simulasi Kinerja Jaringan Nirkabel IEEE-802.11a dan IEEE-802.11g Menggunakan NS-2," *Jurnal Rekayasa Elektrika*, vol. 10, no. 4, Fakultas Teknik Universitas Lampung : Lampung, 2013.
- [31] W. Cahyaningtyas and W. Sulistyio, "Analisis Radio Frequency Channel Wireless Fidelity (WiFi) pada Performa Jaringan WiFi FTI UKSW (Studi Kasus Wifi FTI UKSW)," Fakultas Teknologi Informasi Universitas Kristen Satya Wacana : Salatiga, pp. 2–18, 2017.
- [32] Spyridon Marinis Artelaris, "Performance Evaluation of Routing Protocols for Wireless Mesh Networks," *Journal Computer Sciences*, Department of Computer Science Faculty of Technology Linnaeus University : Swedia, 2015.
- [33] ETSI, "Telecommunication and Internet Protocol Harmonization Over Network (TIPHON); General Aspects of Quality of Service (QoS)," *European Telecommunications Standards Institute ETSI 1999*, vol. 2.1.1, Valbonne, France, pp. 1–37, 1999.

LAMPIRAN

Coding Simulasi :

```
using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("LTE_2-cluster");

int main (int argc, char* argv[])
{
std::string phyMode ("DsssRate11Mbps"); //wifi PHY mode
std::string tcpVariant = "TcpWestwoodPlus";
/* TCP variant type. */
double rss = -80; // -dBm, rss=received signal strength
uint32_t nbUE1 = 280; //number of user equipment nodes with doubleinterface WAVE
double simTime = 60; //end of simulation (sec)

//command line parameters:
CommandLine cmd;
cmd.AddValue ("tcpVariant", "Transport protocol to use: TcpNewReno, "
    "TcpHybla, TcpHighSpeed, TcpHtcp, TcpVegas, TcpScalable, TcpVeno, "
    "TcpBic, TcpYeah, TcpIllinois, TcpWestwood, TcpWestwoodPlus, TcpLedbat ", tcpVariant);
cmd.AddValue("simTime", "Total duration of the simulation [sec]",simTime);
cmd.AddValue ("nbUE", "Number of LTE User Equipment devicescluster 1", nbUE1);
cmd.AddValue ("rss", "Received signal strength", rss);
cmd.Parse (argc,argv);

tcpVariant = std::string ("ns3:") + tcpVariant;

// Select TCP variant
if (tcpVariant.compare ("ns3:TcpWestwoodPlus") == 0)
{
// TcpWestwoodPlus is not an actual TypeId name; we need TcpWestwood here
Config::SetDefault ("ns3::TcpL4Protocol::SocketType", TypeIdValue (TcpWestwood::GetTypeId ()));
// the default protocol type in ns3::TcpWestwood is WESTWOOD
Config::SetDefault ("ns3::TcpWestwood::ProtocolType", EnumValue (TcpWestwood::WESTWOODPLUS));
}
else
{
TypeId tcpTid;
NS_ABORT_MSG_UNLESS (TypeId::LookupByNameFailSafe (tcpVariant, &tcpTid), "TypeId " << tcpVariant << " not found");
Config::SetDefault ("ns3::TcpL4Protocol::SocketType", TypeIdValue (TypeId::LookupByName (tcpVariant)));
}

LogComponentEnable("OnOffApplication", LOG_LEVEL_INFO);
std::cout << "1. CREATE LTE AND EPC" << std::endl;
```

```

//CREATE LTE & EPC
//use lte and epc helper
Ptr<LteHelper> lteHelper = CreateObject<LteHelper> ();
Ptr<PointToPointEpcHelper> epcHelper =
CreateObject<PointToPointEpcHelper> ();
lteHelper->SetEpcHelper (epcHelper);

//pathloss Model
lteHelper->SetAttribute ("PathlossModel", StringValue("ns3::FriisPropagationLossModel"));

AodvHelper aodv;
Ipv4StaticRoutingHelper ipv4RoutingHelper;

Ipv4ListRoutingHelper list1;
list1.Add (ipv4RoutingHelper, 10);

Ipv4ListRoutingHelper list2;
list2.Add (aodv, 20);

std::cout << "2. DEFINE PGW/REMOTE HOST NODES, AND CREATE THEINTERNET" << std::endl;

//create PGW node
Ptr<Node> pgw = epcHelper->GetPgwNode ();
//Setting untuk Node tidak bergerak
AnimationInterface::SetConstantPosition(pgw,190,375);

//create a single remoteHost node
NodeContainer remoteHostContainer;
remoteHostContainer.Create (1);
Ptr<Node> remoteHost = remoteHostContainer.Get (0);

//Setting untuk Node tidak bergerak
AnimationInterface::SetConstantPosition(remoteHost,190,500);

//install protocol stacks to the remote Host
InternetStackHelper internet;
internet.SetRoutingHelper (list1); //has an effect only on thenext Install()
internet.Install (remoteHostContainer);
internet.Reset(); //clear the stack

//create a p2p network
PointToPointHelper p2p;
p2p.SetDeviceAttribute ("DataRate", DataRateValue (DataRate("100Gb/s")));
p2p.SetDeviceAttribute ("Mtu", UIntegerValue (1500)); //maximumtransmission unit
p2p.SetChannelAttribute ("Delay", TimeValue (Seconds (0.010)));
//create devices for PGW/remoteHost nodes, and install the p2pnetwork on them
NetDeviceContainer internetDevices = p2p.Install (pgw,remoteHost);

//assign IP addresses to internet Devices (pgw,remote host)
Ipv4AddressHelper ipv4h;
ipv4h.SetBase ("1.0.0.0", "255.0.0.0");
Ipv4InterfaceContainer internetInterfaces = ipv4h.Assign(internetDevices);

//interface 0 is localhost, 1 is the p2p device
//Ipv4Address remoteHostAddr = internetInterfaces.GetAddress (1);
//specify routes so that the remote host can reach the UEs
Ptr<Ipv4StaticRouting> remoteHostStaticRouting = ipv4RoutingHelper.GetStaticRouting (remoteHost->GetObject<Ipv4>());

remoteHostStaticRouting->AddNetworkRouteTo (Ipv4Address("7.0.0.0"), Ipv4Mask ("255.0.0.0"), 1);
//7.0.0.0 is the network base of the UE nodes, 255.0.0.0 the mask,and 1 the port
std::cout << "3. DEFINE THE ENB/UE NODES" << std::endl;

```

```

//DEFINE THE ENB/UE NODES
//create eNodeB and UE nodes
NodeContainer enbNodes;
NodeContainer ueNodes1;

enbNodes.Create (1);
ueNodes1.Create (nbUE1);

//DEFINE LTE MOBILITY MODEL for FIXED position
MobilityHelper enbMobility;
enbMobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
Ptr<ListPositionAllocator> positionAlloc = CreateObject<ListPositionAllocator> ();
positionAlloc = CreateObject<ListPositionAllocator> ();
positionAlloc->Add (Vector (250.0, 250.0, 0.0));

//Install eNodeB position (mobility to be fixed)
enbMobility.SetPositionAllocator(positionAlloc);
enbMobility.Install(enbNodes);

//Create waypoint for UEs Mobility
Ptr<ListPositionAllocator> waypoint1 =CreateObject<ListPositionAllocator> ();
waypoint1 = CreateObject<ListPositionAllocator> ();
waypoint1->Add (Vector (100.0, 0.0, 0.0));
waypoint1->Add (Vector (200.0, 0.0, 0.0));
waypoint1->Add (Vector (100.0, 50.0, 0.0));
waypoint1->Add (Vector (200.0, 50.0, 0.0));

MobilityHelper ueMobility1;

ueMobility1.SetPositionAllocator ("ns3::GridPositionAllocator",
                                "GridWidth", UintegerValue (4),
                                "MinX", DoubleValue (125.0),
                                "MinY", DoubleValue (10.0),
                                "DeltaX", DoubleValue (125.0),
                                "DeltaY", DoubleValue (10.0));
ueMobility1.SetMobilityModel ("ns3::RandomWaypointMobilityModel", "Speed", StringValue
("ns3::UniformRandomVariable[Min=5.0|Max=10.0]"), "Pause", StringValue
("ns3::ConstantRandomVariable[Constant=0.0]"), "PositionAllocator", PointerValue(waypoint1));

//ueMobility1 diinstal pada ueNode1
for (uint32_t i=0; i < nbUE1; i++)
{
    ueMobility1.Install(ueNodes1.Get(i));
}

```

```

//install LTE devices to eNodeB/UEs
NetDeviceContainer enbDevices = lteHelper->InstallEnbDevice(enbNodes);
NetDeviceContainer ueDevices1 = lteHelper->InstallUeDevice(ueNodes1);

//install protocol stack to the UEs
for (uint32_t i = 0; i < nbUE1; i++)
{
    internet.SetRoutingHelper (list1); //internet
    internet.Install (ueNodes1.Get(i));
}
internet.Reset();

Ipv4InterfaceContainer ueInterfaces1;

//assign IP to UEs
ueInterfaces1 = epcHelper->AssignUeIpv4Address (NetDeviceContainer(ueDevices1));

//set the default gateway for the UEs
for (uint32_t u = 0; u < nbUE1; ++u)
{
    Ptr<Node> ueNode = ueNodes1.Get (u);
    Ptr<Ipv4StaticRouting> ueStaticRouting = ipv4RoutingHelper.GetStaticRouting (ueNode->GetObject<Ipv4> ());
    ueStaticRouting->SetDefaultRoute (epcHelper->GetUeDefaultGatewayAddress(), 1);
}

//Pemberian/Assign IPv4 dan internet harus dilakukan sebelum Attach EPS
for (uint32_t i = 0; i < nbUE1; i++)
{
    lteHelper->Attach (ueDevices1.Get(i), enbDevices.Get(0));
}

std::cout << "4. Configure CBR application" << std::endl;
//interface 0 is localhost, 1 is the p2p device
Ipv4Address remoteHostAddr = internetInterfaces.GetAddress (1);
//default values for the cbr application
ApplicationContainer onoffapps;
uint16_t port = 8;
OnOffHelper onoff("ns3::TcpSocketFactory",
    InetSocketAddress(remoteHostAddr, port));
onoff.SetAttribute("OnTime",
    StringValue("ns3::ConstantRandomVariable[Constant=1.0]"));
onoff.SetAttribute("OffTime",
    StringValue("ns3::ConstantRandomVariable[Constant=0.0]"));
onoff.SetAttribute("DataRate", StringValue("10Mbps")); //Megabitsper second
onoff.SetAttribute("PacketSize", StringValue("1024")); //bytes
//install application on the last wifi ad-hoc node
onoffapps.Add(onoff.Install(ueNodes1.Get(nbUE1-1)));
onoffapps.Add(onoff.Install(ueNodes1.Get(nbUE1-2)));
onoffapps.Add(onoff.Install(ueNodes1.Get(nbUE1-3)));
onoffapps.Add(onoff.Install(ueNodes1.Get(nbUE1-4)));
onoffapps.Add(onoff.Install(ueNodes1.Get(nbUE1-5)));
onoffapps.Add(onoff.Install(ueNodes1.Get(nbUE1-6)));
onoffapps.Add(onoff.Install(ueNodes1.Get(nbUE1-7)));
onoffapps.Add(onoff.Install(ueNodes1.Get(nbUE1-8)));
onoffapps.Add(onoff.Install(ueNodes1.Get(nbUE1-9)));
onoffapps.Add(onoff.Install(ueNodes1.Get(nbUE1-10)));

```

```

//set the start/stop time
onoffapps.Start(Seconds(1.0));
onoffapps.Stop(Seconds(simTime));
PacketSinkHelper sink("ns3::TcpSocketFactory",InetSocketAddress(remoteHostAddr, port));
sink.Install(remoteHost);
//p2p.EnablePcapAll("lte-wave");
//phy.EnablePcap ("lte-wave-1", waveDevices1);
//phy.EnablePcap ("lte-wave-2", waveDevices2);
//install FlowMonitor on the last wifiNodes
FlowMonitorHelper flowmon;
Ptr<FlowMonitor> monitor;
monitor = flowmon.Install(ueNodes1);
monitor = flowmon.Install(enbNodes);
monitor = flowmon.Install(remoteHost);

//stop simulation
Simulator::Stop(Seconds(simTime));
//print a start message
std::cout << "Running simulation..." << std::endl;

AnimationInterface anim ("1.xml"); // Mandatory
//Deskripsi Node wave dan UE
for (uint32_t i=0; i < nbUE1; i++)
{
anim.UpdateNodeDescription (ueNodes1.Get(i), "UE"); //b Optional
}

//Setting warna untuk masing2 node
for (uint32_t i=1; i < nbUE1; i++)
{
anim.UpdateNodeColor (ueNodes1.Get(i), 255, 0, 0); // Optional
}

anim.UpdateNodeColor (enbNodes.Get(0), 0, 0, 255); // Optional
anim.UpdateNodeDescription (enbNodes.Get(0), "eNodeB1");
anim.UpdateNodeColor (remoteHostContainer.Get(0), 0, 0, 0);
anim.UpdateNodeColor (epcHelper->GetPgwNode (), 0, 255, 0);
anim.UpdateNodeDescription (remoteHostContainer.Get(0), "RemoteHost");
anim.UpdateNodeDescription (epcHelper->GetPgwNode (), "PGW");

anim.EnablePacketMetadata(true); // Optional
anim.EnableIpv4RouteTracking ("routingtable-LTE-2-clusteraadv.xml",
Seconds (0), Seconds (60), Seconds (0.25)); //Optional

//run simulation
Simulator::Run();
monitor->CheckForLostPackets ();
Ptr<Ipv4FlowClassifier> classifier = DynamicCast<Ipv4FlowClassifier>(flowmon.GetClassifier ());
std::map<FlowId, FlowMonitor::FlowStats> stats = monitor->GetFlowStats();
std::map<FlowId, FlowMonitor::FlowStats>::const_iterator vw =
stats.begin ();
std::cout << "Hitung Parameter " << vw->first << std::endl;
for (std::map<FlowId, FlowMonitor::FlowStats>::const_iterator
i = stats.begin (); i != stats.end (); ++i)
{
std::cout << "running !!!" << "\n";
Ipv4FlowClassifier::FiveTuple t = classifier->FindFlow(i->first);
if ((t.destinationAddress == "1.0.0.2"))
{
std::cout << "Flow number " << i->first<< " (" << t.sourceAddress << " -> " << t.destinationAddress << ") \n";
std::cout << " Tx Bytes: " << i->second.txBytes<< " \n";
std::cout << " Rx Bytes: " << i->second.rxBytes<< " \n";
std::cout << " Throughput: " << i->second.rxBytes* 8.0 / (i->second.timeLastRxPacket.GetSeconds() - i->second.timeFirstTxPacket.GetSeconds())
/1024/1024 << " Mbps\n";
std::cout << " PDR: " << (i->second.rxPackets * 100) / (i->second.txPackets) << " % \n";
std::cout << " PLR: " << (i->second.txPackets * 100 - i->second.rxPackets * 100) / (i->second.txPackets) << " % \n";
if (i->second.rxPackets > 0)
{
std::cout <<"Delay = " << i->second.delaySum.GetSeconds()/ i->second.rxPackets <<"\n";
}
}
}
}

```

```

else
{
//std::cout << "Flow number " << i->first << " (" <<t.sourceAddress << " -> " << t.destinationAddress << ")\n";
//std::cout << " Tx Bytes: " << i->second.txBytes << "\n";
//std::cout << " Rx Bytes: " << i->second.rxBytes << "\n";
//std::cout << " Throughput: " << i->second.rxBytes * 8.0 / (i->second.timeLastRxPacket.GetSeconds() - i->second.timeFirstTxPacket.GetSeconds())
// std::cout << " PDR: " << (i->second.rxPackets * 100) / (i->second.txPackets) << " % \n";
if (i->second.rxPackets > 0)
{
//std::cout <<"Delay = " << i->second.delaySum.GetSeconds()/ i->second.rxPackets <<"\n";
}
}
}
//Tambahan Kode 5 --> menyimpan hasil flow monitor ke file
monitor->SerializeToXmlFile("flow-monitor-LTE-2-cluster.xml",
true, true);
/*
GtkConfigStore config;
config.ConfigureAttributes();
*/
Simulator::Destroy();
//print an end message
std::cout << "Done!!" << std::endl;
return 0;
}

```