

**PERBANDINGAN DAN PENGARUH *HANDOVER* TERHADAP
KINERJA PENJADWALAN PAKET *ROUND ROBIN* DAN
PROPORTIONAL FAIR PADA JARINGAN LTE**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Chandra Yogi Adhitama

NIM: 145150200111080



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

**PERBANDINGAN DAN PENGARUH *HANDOVER* TERHADAP
KINERJA PENJADWALAN PAKET *ROUND ROBIN* DAN
PROPORTIONAL FAIR PADA JARINGAN LTE**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Chandra Yogi Adhitama

NIM: 145150200111080



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PERBANDINGAN DAN PENGARUH *HANDOVER* TERHADAP KINERJA
PENJADWALAN PAKET *ROUND ROBIN* DAN *PROPORTIONAL FAIR*
PADA JARINGAN LTE

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Chandra Yogi Adhitama
NIM: 145150200111080

Skripsi ini telah diuji dan dinyatakan lulus pada
26 Desember 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I


Ir. Primantara Hari Trisnawan, M. Sc.
NIP: 19680912 199403 1 002

Dosen Pembimbing II


Reza Andria Siregar S.T., M.Kom.
NIP: 19790621 200604 1 003

Mengetahui
Ketua Jurusan Teknik Informatika



Ir. Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 31 Desember 2018



Chandra Yogi Adhitama
NIM: 145150200111080

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT karena limpahan rahmat dan karunianya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Analisis Perbandingan dan Pengaruh *Handover* Terhadap Kinerja Penjadwalan Paket *Round Robin* dan *Proportional Fair* pada Jaringan LTE”. Tujuan dalam penulisan skripsi ini adalah untuk memenuhi persyaratan perolehan gelar Sarjana Fakultas Ilmu Komputer Universitas Brawijaya Malang.

Dalam penyusunan skripsi ini tidak lepas dari bantuan dan dukungan dari beberapa pihak. Sehingga penulis mengucapkan terima kasih kepada:

1. Kedua orang tua penulis yaitu Bapak Sugiarto dan Ibu Sunarti, beserta adik penulis Gelia Puspa yang telah memberikan bantuan berupa dukungan moral dan material selama proses pengerjaan skripsi ini.
2. Bapak Ir. Primantara Hari Trisnawan, M. Sc., dan Reza Andria Siregar S.T., M.Kom., selaku Dosen Pembimbing I dan Dosen Pembimbing II yang telah bersabar dalam membimbing, memberi arahan, dan dukungan serta saran selama proses pengerjaan dan penyusunan skripsi ini.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D, selaku Ketua Jurusan Teknik Informatika dan Bapak M. Tanzil Furqon, S.Kom, M.ComSc, selaku Sekretaris Jurusan Teknik Informatika.
4. Teman-teman saya diantaranya Bella Dwi Rahmatulia, Egi Muliandri, Harits Abdurrohman, Imada Nur Afifah, Rahmat Yani, Ratna Tri Utami, Rayza Arfian, Rizky Haryandi Rahman, Silvia Aprilla, dan Wanda Athira Luqyana yang selalu memberikan bantuan berupa doa, tenaga, dukungan dan informasi yang sangat bermanfaat dalam kelancaran pengerjaan skripsi ini. Tidak lupa juga teman-teman yang lain yang tidak bisa penulis sebutkan satu per satu.
5. Semua pihak yang telah membantu dalam proses pengerjaan, penyusunan dan penyelesaian skripsi ini.

Dengan penulisan skripsi ini diharapkan dapat berguna bagi semua pihak. Dalam penulisan skripsi terdapat banyak kekurangan, maka penulis dengan terbuka menerima kritik dan saran yang membangun serta membuat penulisan skripsi ini menjadi lebih bermanfaat.

Malang,

Penulis

Cadhitama91@gmail.com

ABSTRAK

Jaringan *Long Term Evolution* (LTE) merupakan jaringan yang sudah banyak penggunaannya untuk mendapatkan informasi. Karena banyaknya pengguna mobile pada jaringan ini maka diperlukan sebuah algoritme penjadwalan paket untuk menjadwalkan paket-paket yang digunakan untuk para pengguna bisa saling berkomunikasi pada tempat yang berubah-ubah. Penelitian ini menilai kinerja algoritme penjadwalan paket *Round Robin* dan *Proportional Fair* serta pengaruh dari *handover* terhadap keduanya. Penelitian ini dilaksanakan dengan menyimulasikan jaringan LTE pada simulator NS-3 dan akan menghasilkan data-data yang diolah kembali sehingga terbentuk nilai untuk mengukur kinerja dari algoritme penjadwalan paket seperti *throughput*, *packet loss ratio*, *delay*, dan *jitter*. Nilai-nilai tersebut dibandingkan dari hasil penjadwalan menggunakan algoritme *Round Robin* dan *Proportional Fair*. Hasil yang didapatkan menunjukkan bahwa algoritme penjadwalan paket *Proportional Fair* mendapatkan hasil *throughput* lebih tinggi dan persentase paket hilang lebih rendah dibandingkan algoritme penjadwalan paket *Round Robin*. Pada *end-to-end delay* algoritme penjadwalan paket *Proportional Fair* juga mendapatkan hasil rata-rata yang lebih cepat 0,003 detik dibandingkan algoritme penjadwalan paket *Round Robin*. Namun pada hasil *jitter* yang didapat, algoritme penjadwalan paket *Round robin* mendapatkan nilai yang lebih baik pada jumlah *user equipment* (UE) yang banyak. Sedangkan algoritme penjadwalan *Proportional Fair* mendapatkan hasil yang bagus pada *jitter* pada saat jumlah UE yang sedikit. Adanya *handover* terhadap kinerja algoritme penjadwalan tersebut memiliki pengaruh yang sedikit, seperti menurunkan rata-rata *throughput* yang terbentuk dan menambah rata-rata paket yang hilang pada saat pengiriman.

Kata kunci: penjadwalan paket, *Round Robin*, *Proportional Fair*, *handover*, LTE

ABSTRACT

Network Long Term Evolution (LTE) is a network that has many users using it to get information. Because mobile users are currently on this network, a packet scheduling algorithm is needed to schedule packages for ubiquitous communication. This research discuss the performance of the packet scheduler Round Robin and the Proportional Fair and the effect of the handover on that packet schedulers. This research was conducted by simulating an LTE network in the NS-3 simulator and will produce data that is reprocessed so that a value is formed to measure the performance of the packet scheduling algorithm such as throughput, packet loss ratio, delay, and jitter. These data will be analyze by comparing them from the results of the Round Robin scheduling algorithm and the Proportional Fair scheduling algorithm. The Proportional Fair scheduling algorithm shows throughput and packet loss ratio which is better than the Round Robin packet scheduling algorithm. Likewise at the end-to-end delay algorithm for Proportional Fair scheduling packets get average 0,003 seconds faster results than the Round Robin packet scheduling algorithm. In the obtained result, the Round robin scheduling algorithm shows a better jitter result on a large number of user equipment (UE). And the Proportional Fair scheduling algorithm show good result in jitter when there is a small amount of UE. While handovers to the scheduling algorithm have little effect, such as reducing throughput and increasing packet loss during the delivery.

Keywords: packet scheduler, Round Robin, Proportional Fair, handover, LTE

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS.....	iii
KATA PENGANTAR	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan.....	3
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 Kajian Pustaka	5
2.2 LTE	5
2.2.1 Voice over LTE	6
2.3 Evolved Packet Core (EPC).....	6
2.4 Evolved UTRAN Node B	7
2.5 Penjadwalan paket.....	7
2.5.1 Round Robin	7
2.5.2 Proportional Fair.....	8
2.6 Handover	9
2.6.1 Reference Signal Received Power (RSRP)	11
2.6.2 Reference Signal Received Quality (RSRQ)	11
2.7 Sounding Reference Signal	11
2.8 Network Simulator 3	12
BAB 3 METODOLOGI	14
3.1 Studi Literatur	14

3.2 Analisis Kebutuhan	16
3.3 Perancangan	17
3.4 Implementasi	18
3.5 Pengujian dan Analisis	20
3.6 Pengambilan Kesimpulan	20
BAB 4 PERANCANGAN.....	21
4.1 Perancangan Sistem	21
4.1.1 Perancangan Infrastruktur Jaringan	21
4.1.2 Topologi Jaringan	21
4.1.3 Peta Persebaran dan UE pada eNB.....	22
4.2 Perancangan Pengujian	27
4.2.1 Perancangan pengujian terhadap <i>Throughput</i>	29
4.2.2 Perancangan pengujian terhadap Packet Loss Rasio	29
4.2.3 Perancangan pengujian terhadap Rata-rata <i>End-to-End Delay</i> ..	30
4.2.4 Perancangan pengujian terhadap Rata-rata <i>Jitter</i>	30
BAB 5 IMPLEMENTASI	31
5.1 Implementasi Sistem	31
5.1.1 Pembangunan UE dan ENB	32
5.1.2 Pembangunan EPC dan RemoteHost.....	35
5.2 Implementasi Pengujian	36
5.2.1 Implementasi Pergerakan UE	36
5.2.2 Implementasi Algoritme Penjadwalan Paket dan <i>Handover</i>	38
5.2.3 Pembangunan <i>Traffic</i>	38
BAB 6 PENGUJIAN DAN ANALISIS	41
6.1 Pengujian	41
6.2 Hasil Pengujian	43
6.2.1 Hasil Pengujian Terhadap <i>Throughput</i>	43
6.2.2 Hasil Pengujian Terhadap <i>Packet Loss Ratio</i>	45
6.2.3 Hasil Pengujian Terhadap <i>End-to-End Delay</i>	47
6.2.4 Hasil Pengujian Terhadap <i>Jitter</i>	49
6.3 Analisis Terhadap Hasil Pengujian	51
BAB 7 PENUTUP	53

7.1 Kesimpulan 53

7.2 Saran 53

DAFTAR PUSTAKA 54

LAMPIRAN A 55



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	5
Tabel 4.1 Peta persebaran setiap node	23
Tabel 4.2 Tabel Parameter Pengujian	28
Tabel 4.3 Tabel perancangan pengujian <i>throughput</i>	29
Tabel 4.4 Tabel perancangan pengujian <i>packet loss ratio</i>	29
Tabel 4.5 Tabel perancangan pengujian <i>end-to-end delay</i>	30
Tabel 4.6 Tabel perancangan pengujian <i>jitter</i>	30
Tabel 5.1 <i>Pseudocode</i> untuk memasukkan modul pada sistem	31
Tabel 5.2 <i>Pseudocode</i> untuk membuat node UE	32
Tabel 5.3 <i>Pseudocode</i> peletakan UE	33
Tabel 5.4 <i>Pseudocode</i> untuk memberikankan alamat IP pada UE	33
Tabel 5.5 <i>Pseudocode</i> pembuatan eNB	34
Tabel 5.6 <i>Pseudocode</i> untuk menghubungkan UE dan eNB	35
Tabel 5.7 <i>Pseudocode</i> pembuatan EPC	35
Tabel 5.8 <i>Pseudocode</i> pembuatan <i>remoteHost</i>	36
Tabel 5.9 <i>Pseudocode</i> pembangunan Internet	36
Tabel 5.10 <i>Pseudocode</i> implementasi pergerakan UE pada skenario <i>handover</i> ..	37
Tabel 5.11 <i>Pseudocode</i> implementasi pergerakan UE pada skenario <i>non-handover</i>	37
Tabel 5.12 <i>Pseudocode</i> pembangunan <i>Traffic Constant BitRate</i>	39
Tabel 6.1 Hasil <i>Throughput</i> untuk setiap node	43
Tabel 6.2 Hasil <i>Packet loss ratio (%)</i>	45
Tabel 6.3 Hasil <i>end-to-end delay (s)</i> untuk setiap node	47
Tabel 6.4 Hasil pengujian terhadap <i>jitter (s)</i>	49

DAFTAR GAMBAR

Gambar 2.1 Prosedur handover	10
Gambar 2.2 Arsitektur LTE pada NS-3	13
Gambar 3.1 Diagram alir metodologi penelitian	14
Gambar 3.2 Diagram alir perancangan sistem	17
Gambar 4.1 Topologi Jaringan.....	22
Gambar 6.1 Contoh pelaksanaan simulasi	41
Gambar 6.2 Contoh <i>file</i> xml yang terbentuk	42
Gambar 6.3 Contoh <i>file</i> aktivitas UE dan eNB yang terbentuk	42



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Kebutuhan akses data yang semakin meningkat berdampak pada teknologi untuk terus beradaptasi dan berkembang. Pada saat ini teknologi telekomunikasi yang sudah sering digunakan untuk memenuhi kebutuhan tersebut adalah teknologi *Long Term Evolution*. Jaringan *Long Term Evolution* (LTE) atau jaringan pada generasi 4G ini merupakan jaringan telekomunikasi yang menggunakan gelombang radio untuk bertukar data kepada penggunanya, jaringan ini dapat dinilai berkecepatan tinggi karena kecepatan *downlink* pada jaringan LTE ini dapat mencapai 100 Mbps, *uplink* mencapai 50 Mbps dengan *bandwidth* 1.4 MHz-20 Mhz (Panjaitan, 2015). Jaringan LTE ini memperbaiki kekurangan dari jaringan pada generasi sebelumnya yaitu jaringan *High-Speed Downlink Packet Access* (HSDPA) pada teknologi generasi 3,5G yaitu pada *air interface*, jaringan radio, dan jaringan *core* serta menambahkan fitur yang mendukung untuk mengoperasikan fitur *Multimedia Broadcast Multicast Service*. Faktor yang meningkatkan kecepatan pada teknologi jaringan 4G ini adalah teknik modulasi yang digunakan yaitu *Orthogonal Frequency Division Multiple Access* (OFDMA) dan lebar dari *bandwidth* yang digunakan.

Bandwidth merupakan sumber daya atau *resource* yang digunakan untuk dibagikan kepada perangkat-perangkat pengguna. Dikarenakan pengguna yang tidak sedikit dan jumlah *resource* yang terbatas mengakibatkan diperlukannya pembagian *resource* yang efektif kepada setiap penggunanya. Bagian yang digunakan untuk membagi *resource* tersebut terdapat pada bagian *Evolved Universal Terrestrial Radio Access Network* (E-UTRAN) menggunakan algoritme penjadwalan paket. Pada metode penjadwaan paket ini diperlukan paket-paket yang digunakan untuk dikirimkan pada pengaturan skema penjadwalan tersebut. Terdapat berbagai algoritme penjadwalan paket dengan kelebihan dan kekurangan masing-masing algoritme. Untuk membantu kinerja dari OFDMA, E-UTRAN juga dibantu oleh jaringan inti berbasis IP yang disebut *Evolved Packet Core* (EPC) yang bertanggung jawab dalam memenejemen pergerakan, *charging*, otentikasi, dan *setup end-to-end connection* (Swetha, et al., 2014).

Kebutuhan informasi yang ingin didapatkan dengan cepat juga mempengaruhi banyaknya pengguna *mobile phone* dikarenakan dengan *mobile phone* penggunanya dapat mengakses kebutuhan informasi dimana saja dan kapan saja. Penggunaan *mobile phone* dapat mempermudah penggunanya mendapatkan informasi saat dalam keadaan aktif bergerak seperti saat berjalan ataupun saat berkendara. Hal ini dapat menyebabkan terjadinya perpindahan hubungan antara pengguna atau dapat disebut sebagai *user equipment* (UE) dengan *Base Transceiver Station* atau pada jaringan LTE dikenal dengan nama *Evolved UTRAN Node B* (eNB). Perpindahan tersebut menyebabkan adanya prosedur serah terima hubungan UE oleh eNB yang bisa disebut dengan *handover* dikarenakan ruang lingkup dari setiap eNB yang terbatas.

Penanganan setiap eNB terhadap keluar masuknya, ataupun terhubung terputusnya pengguna atau UE inilah yang menjadi pokok permasalahan bagaimana *resource* dialokasikan ke setiap UE. Pengalokasian *resource* ini membutuhkan algoritme penjadwalan paket yang dapat membagikan secara adil dan tidak menyia-nyiaikan *resource* yang ada.

Pada penelitian sebelumnya yang berjudul “Analisa Performansi *Packet Sheduling Downlink Max-Throughput & Proportional Fair* pada Layanan VoLTE” yang dilakukan oleh Aaron Panjaitan (2015), membahas tentang perbandingan antara algoritme penjadwalan paket *Maximum Throughput* dengan algoritme penjadwalan paket *Proportional Fair* untuk menguji kualitas layanan suara pada jaringan LTE. Layanan suara pada jaringan LTE disebut sebagai *Voice over LTE* (VoLTE). Peneliti sebelumnya memilih *Maximum Throughput* dan *Proportional Fair* sebagai penjadwalan paket sebagai metode yang diteliti karena *Maximum Throughput* bertujuan memaksimalkan nilai *Throughput* dalam *cell* dan mengutamakan pengalokasian *resource* pada UE yang memiliki kualitas sinyal *Channel Quality Indicator* (CQI) terbaik, sedangkan penjadwalan paket *Proportional Fair* bertujuan memaksimalkan *Throughput* dengan dalam *cell* sekaligus memberikan keadilan dalam pembagian *throughput* kepada setiap UE. Dari penelitian tersebut didapatkan hasil bahwa penjadwalan paket *Maximum Throughput* memiliki keunggulan pada parameter *throughput* dan *spectral efficiency* dan penjadwalan paket *Proportional Fair* memiliki keunggulan pada parameter *fairness*, *packet loss ratio*, dan *delay*. Namun pada penelitian tersebut terbatas pada eNB atau *cell* yang digunakan yaitu berjumlah 1 buah. Sedangkan pada penelitian lain yang dilakukan oleh Swetha (2014) untuk meneliti kinerja algoritme penjadwalan *Round Robin* yang memberikan kesempatan penjadwalan secara merata dan sama pada setiap pengguna dalam *cell* pada aplikasi *video streaming* dalam jaringan LTE didapatkan hasil bahwa algoritme *Round Robin* mempunyai kinerja yang baik pada *codec rates* yang bernilai tinggi.

Berdasarkan latar belakang tersebut peneliti ingin melanjutkan penelitian sebelumnya dengan menggunakan algoritme *Round Robin* dan algoritme *Proportional Fair* sebagai algoritme penjadwalan paket dan meneliti pengaruh *handover* terhadap kinerja penjadwalan paket pada jaringan LTE tersebut. Penelitian yang akan dilakukan juga akan menambahkan jumlah eNB dari penelitian sebelumnya yang hanya berjumlah 1 buah menjadi 3 buah sehingga UE dapat bergerak bebas dan mendukung terjadinya *handover* pada penelitian. Penelitian dilakukan dengan menyimulasikan jaringan LTE dan menerapkan algoritma penjadwalan paket kedalam simulasi pada simulator NS-3 dengan skenario-skenario yang mendukung. Penelitian ini dilakukan untuk menjadi pertimbangan dalam pemilihan algoritme penjadwalan paket ataupun dapat menjadi evaluasi algoritme penjadwalan paket dalam pengembangannya dimasa depan.

1.2 Rumusan Masalah

1. Bagaimana penerapan algoritme penjadwalan paket *Round Robin* dan *Proportional Fair* pada jaringan LTE?
2. Bagaimana kinerja algoritme penjadwalan paket *Round Robin* dan *Proportional Fair* pada jaringan LTE?
3. Bagaimana pengaruh *handover* terhadap kinerja dari algoritme penjadwalan paket *Round Robin* dan *Proportional Fair*?

1.3 Tujuan

1. Menerapkan algoritme penjadwalan paket pada jaringan LTE.
2. Mengetahui kinerja dari masing-masing algoritme penjadwalan paket
3. Mengetahui pengaruh *handover* terhadap kinerja algoritme penjadwalan paket.

1.4 Manfaat

Manfaat dari penulisan skripsi ini adalah untuk mengetahui kinerja pada scheduling *Round Robin* dan *Proportional Fair* pada jaringan LTE dan diharapkan dapat diimplementasikan pada pengembangan teknologi komunikasi multimedia.

1.5 Batasan Masalah

Adapun batasan masalah dalam penelitian ini agar pembahasan dapat dilakukan secara fokus dan mendapatkan hasil yang sesuai dengan yang diharapkan, antara lain:

1. Simulasi scheduling dilakukan menggunakan perangkat lunak NS-3 yang berjalan pada sistem operasi Ubuntu
2. Fokus analisis berada pada *Quality of Services* dan pengaruh *handover*
3. Jumlah eNodeB (eNB) yang digunakan adalah 3 dengan jarak setiap eNB adalah 500 meter dan diletakkan secara sejajar.
4. *Sounding Reference Signal (SRS) periodicity* sebesar 40 ms.

1.6 Sistematika Pembahasan

Sistematika pembahasan penelitian ini disusun sebagai berikut:

BAB I PENDAHULUAN

Pada bagian pendahuluan, penulis menjelaskan tentang latar belakang, rumusan masalah, tujuan, serta manfaat dari penelitian.

BAB II LANDASAN KEPUSTAKAAN

Bab ini akan diuraikan mengenai dasar teori yang digunakan oleh penulis dalam proses penelitian.

BAB III METODOLOGI PELAKSANAAN

Bab ini menjelaskan mengenai langkah-langkah penulis dalam melaksanakan penelitian meliputi studi literatur, analisis kebutuhan perancangan, pengujian dan analisis hingga mendapatkan kesimpulan.

BAB IV PERANCANGAN

Bab ini berisi tentang spesifikasi tentang kebutuhan untuk membangun simulasi serta skenario yang akan diujikan dalam penelitian.

BAB V IMPLEMENTASI

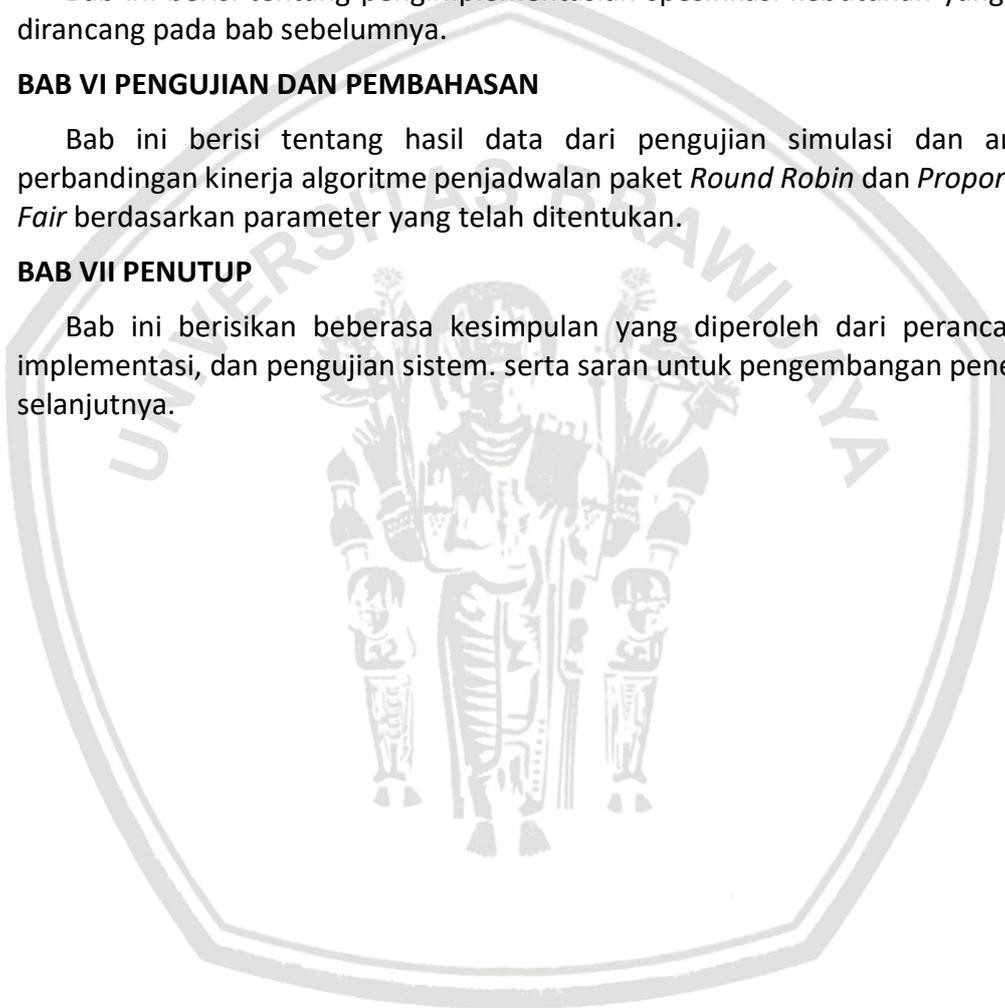
Bab ini berisi tentang pengimplementasian spesifikasi kebutuhan yang telah dirancang pada bab sebelumnya.

BAB VI PENGUJIAN DAN PEMBAHASAN

Bab ini berisi tentang hasil data dari pengujian simulasi dan analisis perbandingan kinerja algoritme penjadwalan paket *Round Robin* dan *Proportional Fair* berdasarkan parameter yang telah ditentukan.

BAB VII PENUTUP

Bab ini berisikan beberapa kesimpulan yang diperoleh dari perancangan, implementasi, dan pengujian sistem. serta saran untuk pengembangan penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada penelitian ini, kajian pustaka diambil dari beberapa penelitian yang sudah pernah dilakukan dan menampilkan persamaan dan perbedaan dari rencana penelitian yang akan dilakukan pada skripsi ini. Persamaan dan perbedaan tersebut dijelaskan pada Tabel 2.1.

Tabel 2.1 Kajian Pustaka

No.	Penulis, Tahun, dan Judul Penelitian	Persamaan	Perbedaan	
			Penelitian sebelumnya	Rencana Penelitian
1.	Aaron Panjaitan, 2015, Analisa Performansi Penjadwalan paket Downlink Max-Throughput & <i>Proportional Fair</i> pada Layanan VoLTE	Menggunakan packet scheduler <i>Proportional Fair</i> pada layanan VoLTE dan parameter skenario simulasi	eNB dan cell terbatas hanya berjumlah 1 dan tidak ada prosedur <i>handover</i>	eNB dan cell berjumlah 3 dengan menambahkan prosedur <i>handover</i>
2.	Swetha, 2014, Performance Study of <i>Round Robin</i> and <i>Proportional Fair</i> Scheduling Algorithm by Emulation for Video Traffic in LTE Network	Penjadwalan paket yang digunakan yaitu <i>Round Robin</i> dan <i>Proportional Fair</i>	UE berjumlah 4 dan tidak ada prosedur <i>handover</i>	UE berjumlah 1 sampai 19 per eNB dan menambahkan prosedur <i>handover</i>

2.2 LTE

Long Term Evolution (LTE) merupakan teknologi jaringan telekomunikasi yang memiliki kecepatan tinggi dengan standar yang telah diterapkan oleh *Third Generation Partnership Project* (3GPP) sebagai penerus teknologi jaringan seluler 3G. LTE menggunakan gelombang radio dalam mentransmisikan data antar UE. Jumlah gelombang radio yang dapat digunakan terbatas oleh jumlah *bandwidth* yang tersedia, oleh karena itu dibutuhkan algoritme penjadwalan transmisi data agar *bandwidth* dapat digunakan dengan optimal dan UE memperoleh layanan dengan kualitas yang maksimal, algoritme tersebut disebut sebagai algoritme penjadwalan paket (Panjaitan, 2015).

Jaringan LTE menggunakan kombinasi pembagian frekuensi multiplexing dan pembagian waktu multiplexing pada downstream channel yang biasa dikenal sebagai *Orthogonal Frequency Division Multiplexing* (OFDM). Setiap node seluler yang aktif pada jaringan LTE dialokasikan satu atau lebih *slot* waktu 0,5 ms dalam satu atau lebih frekuensi saluran. Dengan pengalokasian lebih banyak *slot* waktu ini sebuah node seluler akan dapat mencapai tingkat transmisi yang semakin tinggi. Pengalokasian antar node seluler dapat dilakukan dengan sering pada setiap milidetik. Dengan skema modulasi yang berbeda juga dapat digunakan untuk mengubah *transmission rate*. Inovasi yang digunakan pada jaringan LTE ini adalah penggunaan antena multiple-input multiple-output (MIMO) yang canggih. Kecepatan data maksimum untuk pengguna jaringan LTE mencapai 100Mbps pada arah downstream dan 50 Mbps pada arah upstream ketika menggunakan spektrum nirkabel sebesar 20 MHz

2.2.1 Voice over LTE

VoLTE (*Voice over LTE*) adalah salah satu layanan pada jaringan LTE yang berfokus pada komunikasi suara pada interaksinya dan berbasis IP atau VoIP (*Voice over IP*). Menurut Spirent (2012), meskipun berbeda dari segi teknologi, namun secara konseptual VoIP yang berjalan pada jaringan LTE dapat disebut sebagai VoLTE. VoLTE memiliki nilai standar kualitas MOS (*Mean Opinion Score*) lebih besar atau sama dengan 3,5.

2.3 Evolved Packet Core (EPC)

Evolved packet core atau EPC adalah sebuah sistem arsitektur komunikasi seluler dengan jaringan all-IP pada bagian core network. Jaringan all-IP yang digunakan memungkinkan jaringan LTE dapat berkomunikasi atau terhubung ke jaringan lain yang berbeda-beda. Dengan demikian semua pemrosesan paket IP dikelola pada EPC. Hal ini dapat mempercepat waktu respon pada penjadwalan ataupun re-transmission dan juga dapat meningkatkan latency maupun throughput (Kurose & Ross, 2012).

Tugas utama dari EPC adalah untuk mengelola sumber daya jaringan sehingga dapat menyediakan layanan dengan kualitas tinggi kepada pengguna yang terhubung pada jaringan LTE. Dalam penyediaan layanan tersebut EPC dibagi menjadi beberapa bagian yang memiliki tugas yang berbeda-beda untuk menunjang kualitas layanan. Bagian-bagian tersebut antara lain adalah Serving Gateway (SGW), Packet Data Network Gateway (PGW), dan Mobility Management Entity (MME).

SGW merupakan interface dari EPC yang menghubungkannya kepada eNodeB (eNB). SGW berfungsi sebagai *local Mobility Anchor point* yang menangani permasalahan *handover* UE pada eNB dan mengumpulkan informasi serta statistik yang diperlukan dalam proses charging. SGW juga memiliki tanggung jawab untuk mereplika lalu lintas pengguna jika terjadi gangguan atau *error*. Berbeda dengan SGW, PGW merupakan interface yang menghubungkan antara jaringan LTE dengan jaringan lain yang bukan termasuk jaringan LTE. PGW juga bertanggung

jawab dalam kontrol kebijakan didalam pengalokasian IP prefix. Sedangkan MME merupakan control-plane pada EPC yang bertanggung jawab pada penghubungan atau pelepasan *bearers* pada terminal, penanganan posisi status *idle* menuju transisi aktif, serta penanganan *security keys*.

2.4 Evolved UTRAN Node B

Evolved UTRAN NodeB (eNB) merupakan bentuk evolusi dari NodeB di UTRA UMTS, dan perangkat ini adalah perangkat keras yang terhubung ke jaringan telepon seluler yang berkomunikasi secara langsung dengan *user equipment* (UE). ENodeB bertindak seperti *base transceiver station* (BTS) pada jaringan GSM. Jika NodeB yang terdapat pada jaringan *Global System for Mobile communications* (GSM) elemen pengontrol diatur oleh *Radio Network Controller* maka pada eNB tidak menggunakan elemen pengontrol terpisah seperti NodeB dan menggunakan fungsi kontrol tersendiri dengan demikian dapat menyederhanakan arsitektur dan memungkinkan waktu respon yang lebih rendah (Mahfoudi, et al., 2014).

2.5 Penjadwalan paket

Penjadwalan paket adalah algoritme yang digunakan untuk mengalokasikan *resource block* dalam transmisi data ke *user*, penjadwalan paket bertujuan untuk meningkatkan efisiensi penggunaan *bandwidth*, memberikan layanan data kepada *user* (UE) dengan standar QoS (*Quality of Service*) dan tingkat *fairness* yang maksimal, dengan cara menentukan proses transmisi data berdasarkan hasil penghitungan algoritme yang disebut sebagai nilai *metric*. *Resource block* merupakan blok-blok pada *bandwidth* yang digunakan dalam transmisi data.

2.5.1 Round Robin

Penjadwalan *Round Robin* (RR) merupakan algoritme yang penjadwalannya dilakukan secara bergiliran berdasarkan antrean. Proses yang telah dieksekusi prosessor namun belum selesai dikarenakan waktu yang diberikan untuk memproses aket telah habis akan dikembalikan ke antrean terakhir yang ada pada saat itu sehingga penggiliran untuk eksekusi tersebut berputar hingga seluruh paket dalam antrean selesai diproses.

Algoritme penjadwalan *Round Robin* memiliki prinsip dasar semua antrean dianggap sama sehingga diberi waktu yang disebut *time quantum*, dan jika *time quantum* tersebut habis atau proses selesai maka proses berlanjut pada antrean berikutnya. Algoritme penjadwalan paket *Round Robin* mengabaikan dan tidak memperhitungkan *Channel Quality Information* (CQI) sehingga dapat dikatakan bahwa algoritme ini menawarkan jaminan dengan nilai keadilan yang besar namun menurunkan kinerja *throughput* pada sistem. Penjadwalan *Round Robin* dapat diimplementasikan menjadi dua cara yaitu dengan menggunakan *Time Domain Round Robin* (TDRR) dan *Time and Frequency Domain Round Robin* (TFDRR). Pada TDRR, pengguna pertama yang dijangkau akan dilayani dengan seluruh spektrum frekuensi untuk jangka waktu tertentu (1 TTI) dan tidak menggunakan CQI, kemudian setelah waktu yang digunakan habis maka sumber daya yang digunakan

tersebut dialihkan pada pengguna berikutnya untuk jangka waktu yang lain. Jangka waktu tersebut yang disebut sebagai time quantum. Setelah sumberdaya yang digunakan oleh pengguna pertama dialihkan, pengguna tersebut akan ditempatkan pada akhir antrian tunggu sehingga dapat dilayani kembali pada waktu berikutnya. Sedangkan dalam beberapa penggunaan TFDRR, pengguna diizinkan untuk dijadwalkan dalam satu TTI dalam satu urutan.

Secara spesifik penjadwalan *Round Robin* dapat dimodelkan secara matematis seperti persamaan berikut,

$$user_j = \text{mod}((k - 1), N) + 1 \quad (2.1)$$

Keterangan:

$user_j$ = pengguna ke j

k = urutan pada TTI dimana pengguna akan dijadwalkan

N = jumlah pengguna aktif

$User_j$ akan dijadwalkan pada TTI (*Transmission Time Interval*) ke- k jika urutan pengguna sama dengan nilai dari sisa hasil dari pembagian antara $k - 1$ dan banyaknya jumlah pengguna yang aktif dan ada dalam antrian penjadwalan dan ditambahkan 1 untuk memulai indeks awal penjadwalan dari urutan 1 (bukan 0). Dari persamaan tersebut terlihat bahwa teknik ini independen terhadap kondisi propagasi maupun karakteristik kanal, sehingga tidak menganggap adanya keragaman kondisi multi-user (Suryaman, et al., 2010).

2.5.2 Proportional Fair

Proportional Fair (PF) adalah algoritme penjadwaan paket yang bertujuan memaksimalkan *throughput cell* sekaligus meningkatkan nilai *fairness*. Algoritme ini merupakan algoritme yang umum digunakan pada sistem *multi-user* dengan *Time frequency* yang awalnya diimplementasikan dalam sistem *Time Domain Scheduling* (TDS) dan diadopsi ke LTE untuk mengeksplorasi kemampuan OFDMA dalam sistem TDS dan *Frequency Time Scheduling* (FDS). Penggabungan sistem TDS dan FDS ini memiliki tujuan untuk mencapai *trade-off* yang baik antara *throughput* sistem secara keseluruhan dan tingkat keadilan data diantara pengguna dengan mengeksplorasi keanekaragaman *multi-user*.

Algoritme *Proportional Fair* ini menghitung nilai metrik berdasarkan penghitungan nilai *data rate* dan nilai *Throughput* rata-rata yang diperoleh dari penghitungan metrik terakhir oleh *flow* yang sama (Capozzi, et al., 2013). Algoritme ini juga berdasar pada keseimbangan antara kepentingan untuk memaksimalkan *throughput* total dan kepentingan yang memungkinkan semua pengguna mendapatkan pelayanan yang minimal. Dengan demikian algoritme *Proportional Fair* lebih mengutamakan pengiriman data kepada UE yang memiliki nilai *Throughput* rata-rata terendah.

Algoritme penjadwalan *Proportional Fair* dapat dimodelkan secara matematis menjadi persamaan sebagai berikut,

$$user(j) = \max \frac{rate(j)}{\overline{rate}} \quad (2.2)$$

Keterangan:

$user(j)$ = pengguna dengan nomor urutan ke- j

$rate(j)$ = nilai *transfer rate* pada pengguna dengan nomor urutan ke- j

\overline{rate} = nilai rata-rata *transfer rate*

Penjadwalan ini memilih $user_j$ untuk dijadwalkan berdasarkan rasio antara *transfer rate* $user_j$ terhadap nilai rata-rata *transfer rate*, bagi $user$ yang sedang aktif rasio ini yang kemudian didefinisikan sebagai *preference metric*. Pada kasus khusus seperti jika terdapat $user$ yang memiliki nilai *preference metric* yang sama maka penjadwalan akan menggunakan algoritme *Round Robin*.

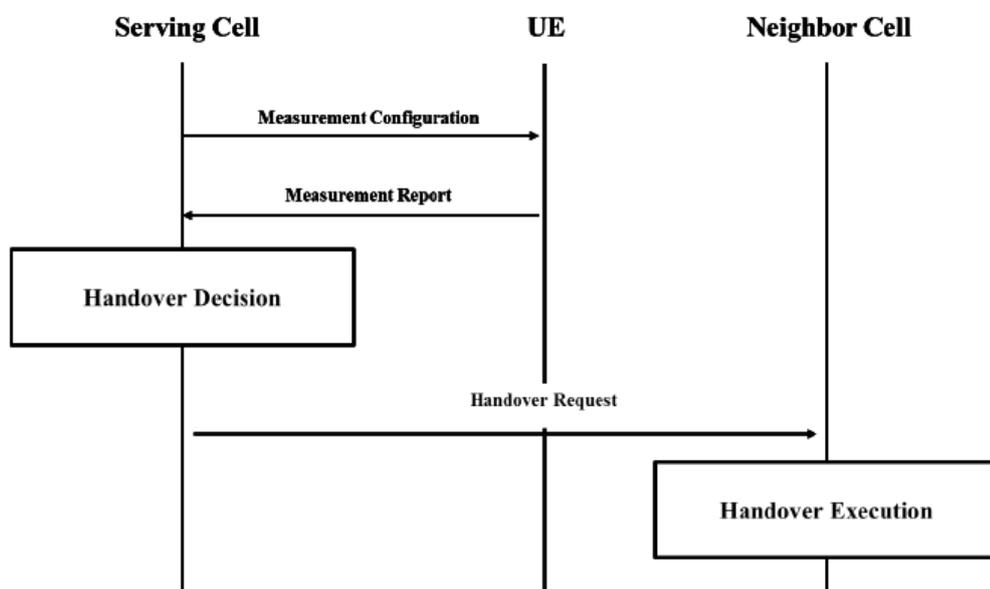
2.6 Handover

Handover adalah proses pengalihan *kanal traffic* secara otomatis pada *Mobile Station* (MS) yang sedang digunakan untuk berkomunikasi tanpa terjadinya pemutusan hubungan. Sederhananya *handover* adalah sebuah 'call' koneksi yang bergerak dari satu sel ke sel lainnya.

Handover terjadi karena kualitas sinyal atau daya sinyal turun. Penurunan level sinyal ini dideteksi dari pengukuran yang dilakukan UE ataupun eNB. Dengan demikian *handover* ditujukan ke *cell* dengan sinyal lebih besar. Selain itu, *handover* dapat terjadi apabila *traffic* dari sel yang dituju sudah penuh. Saat UE melewati *cell*, dialihkan ke '*neighbouring cell*' dengan beban *traffic* yang lebih dua kecil. Terdapat pengukuran yang mempengaruhi terjadinya *handover* yaitu, *Reference Signal Received Power* (RSRP) dan *Reference Signal Received Quality* (RSRQ) (Afroz, 2015).

Pada jaringan downlink LTE, terdapat dua jenis prosedur *handover* untuk UE yang dalam keadaan mode aktif. Mode aktif yang dimaksud menandakan bahwa UE sedang dalam komunikasi atau sedang mengirim/menerima paket di dalam jaringan baik berupa paket suara maupun paket data. Dua prosedur *handover* tersebut adalah S1 dan X2. Prosedur *handover* X2 biasa digunakan dalam *handover* antar eNB yang dimaksudkan untuk menyeimbangkan beban jaringan dan mencegah interferensi. Prosedur ini bisa digunakan jika terdapat interface X2 pada dua eNB yang terhubung atau sedang terjadi proses *handover*. Jika interface X2 tidak ada pada salah satu atau keduanya maka prosedur *handover* kan dialihkan kepada interface S1. Biasanya prosedur *handover* S1 digunakan pada jaringan non-3GPP seperti CDMA2000/HRPD (Lin, et al., 2011).

Terdapat tiga fase pada prosedur *handover* pada interface S1 maupun X2 yaitu fase persiapan, fase eksekusi, dan fase penyelesaian. Pada fase persiapan, UE harus mengirimkan informasi pengukuran secara berkala kepada eNB. Dari informasi yang telah diberikan UE tersebut eNB akan memutuskan atau menentukan eNB tempat UE akan diserahkan. Selain informasi pengukuran tersebut terdapat kriteria lain yang menjadi pertimbangan sebelum akhirnya eNB mengirimkan pesan kontrol ke eNB yang akan dituju untuk mempersiapkan terjadinya *handover*. Fase persiapan terjadi setelah eNB tujuan menerima pesan kontrol tersebut. Selanjutnya eNB tujuan akan mempersiapkan buffer kepada UE yang akan berhubungan. Dan setelah fase persiapan telah selesai, pesan kontrol perintah untuk melakukan *handover* dikirimkan dari eNB ke UE untuk memberitahukan bahwa UE tersebut sedang dialihkan atau diserahkan ke eNB tujuan. Kemudian UE akan memutuskan hubungan dari eNB sumber dan meminta kepada eNB tujuan untuk berhubungan. Di saat yang sama eNB sumber meneruskan semua paket UE kepada eNB tujuan. Paket-paket ini akan dimasukkan pada antrean oleh eNB tujuan kedalam *buffer* yang telah dipersiapkan sebelumnya. Setelah UE berhasil terhubung dengan eNB tujuan, eNB tersebut akan mengirimkan semua paket buffer UE yang diikuti peket-paket lain yang masuk dari gateway eNB tersebut. Setelah semua prosedur *handover* telah selesai maka selanjutnya fase penyelesaian dimuali. Fase ini berlangsung dengan UE mengirimkan pesan yang menunjukkan bahwa proses *handover* telah berhasil dan selesai kepada eNB tujuan yang telah terhubung.



Gambar 2.1 Prosedur handover

Sumber: (Kim, et al., 2014)

2.6.1 Reference Signal Received Power (RSRP)

RSRP merupakan salah satu jenis dari algoritme *handover* dengan metrik yang mempertimbangkan kekuatan sinyal dari sel yang digunakan sebagai masukan dari *cell resection* dan keputusan untuk melakukan *handover*. RSRP didefinisikan sebagai rata-rata daya (dalam watt) dari Elemen sumber daya yang membawa sinyal referensi sel spesifik pada *bandwidth* (Afroz, et al., 2015).

Ukuran RSRP dinyatakan dalam satuan *decibel-milliwatts* (dBm) dan digunakan untuk membuat peringkat diantara kandidat-kandidat sel sesuai kekuatan dari sinyanya yang dipancarkan.

2.6.2 Reference Signal Received Quality (RSRQ)

Pengukuran pada RSRQ menggunakan metrik yang mempertimbangkan kualitas dari sinyal pada sel tertentu. Proses pengambilan keputusan *handover* pada RSRQ mirip dengan RSRP yaitu membuat peringkat atau membandingkan antar kandidat-kandidat sel namun sesuai kualitas dari sinyal yang diberikan sel tersebut. Pengukuran RSRQ dapat dilakukan untuk pengambilan keputusan ulang jika pada metode RSRP dinilai tidak mencukupi (Afroz, et al., 2015). Pengukuran RSRQ dapat didefinisikan dengan persamaan berikut.

$$RSRQ = \frac{(N \cdot RSRP)}{(LTE \text{ carrier RSSI})} \quad (2.3)$$

Dari persamaan tersebut dapat diartikan bahwa RSRQ merupakan rasio RSRP dalam *Received Signal Strength Indicator* (RSSI) dimana N adalah jumlah *Resource Block* dari pengukuran *bandwidth* pada RSSI dalam LTE. RSSI dinyatakan sebagai rata-rata linier dari total daya yang diterima dan yang diamati hanya tanda-tanda OFDM yang membawa tanda referensi UE dari berbagai sumber, termasuk saluran pada *-serving* ataupun *non-serving cell*, gangguan saluran yang berdekatan, dan gangguan termal dalam pengukuran *bandwidth* lebih dari N *Resource Block* (Sesia, et al., 2011).

2.7 Sounding Reference Signal

Sounding Reference Signal (SRS) atau yang biasa dikenal dengan sinyal referensi yang dikirimkan UE untuk eNB. Sinyal ini digunakan oleh eNB untuk mengetahui kualitas saluran dari jalur *uplink* dari setiap subbagian *frequency region*. Pada jaringan LTE, eNB sering mengalokasikan hanya sebagian dari keseluruhan *bandwidth* pada UE tertentu dan pada waktu tertentu. Sehingga perlu untuk mengetahui subbagian *frequency region* yang memiliki kualitas saluran yang baik sehingga UE dapat dialokasikan ke subbagian *frequency region* tersebut.

Pengiriman SRS oleh UE dapat diatur dengan variasi interval tertentu. Variasi interval ini bisa disebut sebagai SRS periodicity dengan macamnya adalah 2, 5, 10, 20, 40, 80, 160, 320 ms atau milidetik. Namun ada juga pilihan untuk UE untuk tidak mengirimkan sinyal referensi tersebut. Jika terdapat beberapa UE dengan interval yang sama dalam pengirimannya maka hal tersebut dapat memungkinkan

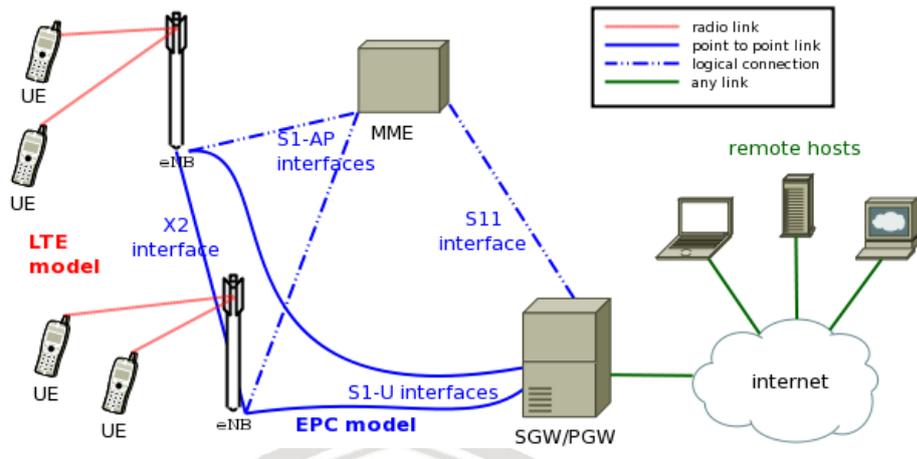
sinyal yang diberikan oleh UE akan saling tumpang tindih ataupun dapat saling menghancurkan sinyal tersebut. Dengan demikian terdapat beberapa pilihan yang memungkinkan untuk menghindari masalah tersebut, yaitu dengan memperbesar interval, mengatur setiap UE untuk mengirim SRS pada jadwal yang berbeda-beda, ataupun mengurangi jumlah UE pada subbagian *frequency region*.

2.8 Network Simulator 3

Network Simulator 3 (NS-3) adalah sebuah simulator diskrit yang menggunakan bahasa pemrograman C++ dalam membangun sistem simulatornya. NS-3 dibangun dengan tujuan untuk dapat mensimulasikan suatu jaringan dalam kebutuhan pengembangan jaringan ataupun dalam kebutuhan akademik. Dalam pengimplementasian kedalam simulator ini, meskipun menggunakan bahasa pemrograman C++ NS-3 juga dapat menjalankan script dari bahasa pemrograman Python. Hal ini dikarenakan NS-3 dapat mengekspor semua API-nya ke dalam Python.

NS-3 mendukung fitur untuk menampilkan atau membantu penggunaanya dalam pengumpulan data hasil keluaran dari simulasi yang telah dilakukan. Fitur ini bernama FlowMonitor. FlowMonitor ini adalah sebuah program yang terintegrasi dalam simulator NS-3 dan berfungsi sebagai aplikasi yang dapat memberikan data hasil berupa statistik mengenai suatu node. Data statistik yang ditunjukkan berupa alamat IP, alamat MAC, tabel routing, jumlah paket yang dikirim maupun yang diterima, jumlah paket yang hilang, waktu paket dikirim maupun yang diterima, dan lain-lain. Data-data tersebut disimpan kedalam suatu *file* dengan format XML dan dapat diakses menggunakan aplikasi NetAnim ataupun aplikasi pengolah data lainnya.

Simulator NS-3 sudah menyiapkan model LTE dengan arsitektur seperti pada Gambar 2.2. Pada model tersebut terdapat beberapa bagian yaitu UE sebagai perangkat mobile, eNB sebagai perangkat perantara yang menghubungkan komunikasi antar UE, dan EPC. EPC sendiri memiliki beberapa bagian yang terdiri dari MME yang bertindak sebagai *control-plane* dan sebagai *authentication* dan *security*, *mobility management*, *managing subscription profile*, dan *service connectivity*, serta SGW/PGW yang bertindak sebagai interface jaringan LTE dengan paket data jaringan lain maupun interface yang menghubungkan EPC ke eNB.



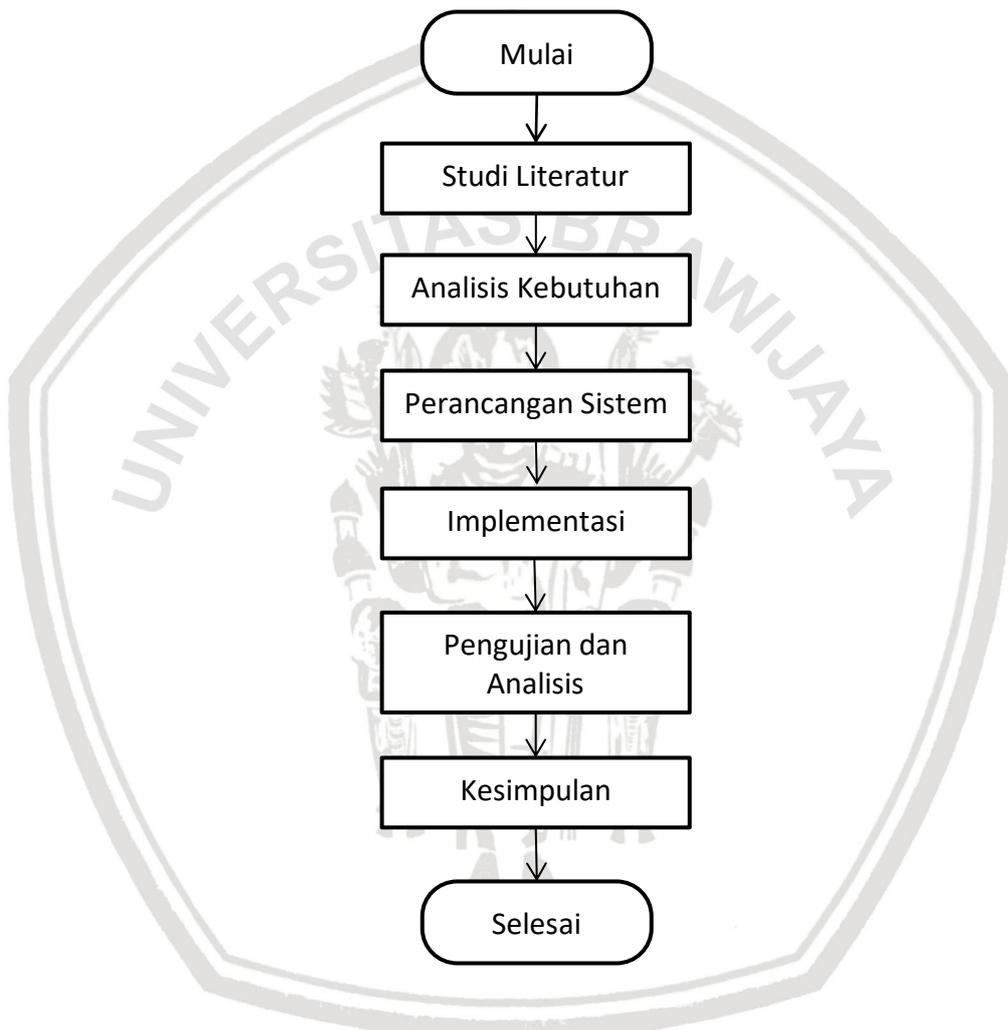
Gambar 2.2 Arsitektur LTE pada NS-3

Sumber: <https://www.nsnam.org/docs/models/html/lte-design.html>



BAB 3 METODOLOGI

Bab ini menguraikan dan menjelaskan langkah-langkah yang akan ditempuh dan digunakan dalam penelitian. Penelitian ini merupakan penelitian yang mengembangkan dan melanjutkan penelitian sebelumnya. Gambar 3.1 merupakan diagram yang berupa tahapan atau langkah-langkah yang dilakukan dalam penelitian yang meliputi studi literatur, analisis kebutuhan, simulasi, pengujian dan analisis, penarikan kesimpulan.



Gambar 3.1 Diagram alir metodologi penelitian

3.1 Studi Literatur

Studi literatur merupakan tahap awal yang dilakukan untuk memahami tentang dasar dari jaringan LTE, layanan VoLTE, serta penjadwalan paket *Round Robin* dan *Proportional Fair* yang akan menjadi dasar dan panduan dalam mensimulasikan kinerja penjadwalan paket tersebut dalam jaringan LTE. Untuk membantu tercapainya penelitian yang akan dilakukan telah dilakukan kajian

terhadap beberapa bidang ilmu serta mempelajari beberapa penelitian yang sudah dilakukan sebelumnya yang masih berkaitan dan menunjang pengembangan penelitian. Diantaranya adalah kajian tentang jaringan LTE, algoritme penjadwalan paket yaitu algoritme *Round Robin* dan algoritme *Proportional Fair*, serta algoritme *handover* dengan menggunakan pengukuran RSRQ.

Kajian terhadap jaringan LTE membahas tentang arsitektur yang digunakan dalam LTE. Terdapat dua bagian penting yang terdapat pada arsitektur LTE yaitu *Evolved UMTS Terrestrial Radio Access Network* (E-UTRAN) dan *Evolved Packet Core* (EPC). E-UTRAN pada penelitian ini lebih dikenal dengan nama *The Evolved UTRAN Node B* (eNB) yang digunakan seperti *base transceiver station* (BTS) pada jaringan GSM. Sedangkan EPC sendiri memiliki beberapa bagian yaitu *Mobility Management Entity* (MME) yang bertindak sebagai *control-plane*, *Serving Gateway* (S-GW) yang bertindak sebagai pengumpul informasi dan statistik yang diperlukan untuk perhitungan algoritme penjadwalan paket ataupun *handover* dalam pergerakan UE terhadap eNB, dan *Packet Data Network Gateway* (PGW) yang bertindak sebagai *interface* antara jaringan LTE dengan paket data pada jaringan lain.

Selanjutnya kajian terhadap algoritme penjadwalan paket. Algoritme penjadwalan paket yang dilakukan pada penelitian ini adalah algoritme *Round Robin* dan algoritme *Proportional Fair* yang keduanya memiliki cara yang berbeda. Pada algoritme penjadwalan paket *Round Robin* terdapat *time quantum* yang digunakan untuk memberikan waktu pada setiap paket yang akan diproses, sedangkan pada algoritme penjadwalan paket *Proportional Fair* untuk membandingkan paket yang akan diproses perlu memerhatikan besarnya *transfer rate* pada setiap node.

Kemudian pada kajian terhadap algoritme *handover* terdapat dua jenis pengukuran yaitu RSRP dan RSRQ. Namun pada penelitian ini hanya menggunakan pengukuran RSRQ dikarenakan pengukuran RSRQ sudah merupakan algoritme pengembangan dari RSRP. Algoritme RSRQ melakukan pemilihan sinyal dengan memilih kualitas sinyal terbaik dari perbandingan kualitas sinyal yang diberikan pada setiap eNB yang tertangkap dan kualitas yang diberikan oleh pengukuran RSRQ didapatkan dari pertimbangan kekuatan sinyal pada RSRP, pengukuran gangguan sinyal seperti *noise*, dan interferensi sinyal.

Kajian selanjutnya adalah kajian tentang simulator yang digunakan pada penelitian ini yaitu Network Simulator 3 (NS-3). NS-3 menggunakan bahasa pemrograman C++ dalam pengimplementasiannya kedalam bentuk kode sumber. Namun NS-3 juga dapat mengenali atau membaca *script* dengan bahasa pemrograman lain yaitu bahasa pemrograman Python. Dalam pengimplementasiannya terhadap jaringan LTE, Simulator NS-3 memiliki modul yang mendukung penelitian pada jaringan LTE dan modul lain yang dapat mendukung dalam pembuatan node sebagai komponen-komponen pada jaringan LTE seperti eNB, UE, dan EPC. Pembuatan eNB dan UE pada NS-3 dapat dilakukan dengan menggunakan bantuan modul-modul lain yang terdapat dalam library NS-

3. Namun dalam pembuatan UE tersebut terdapat batasan dari simulator yang secara default mengatur waktu interval atau SRS *periodicity* yang digunakan sebesar 40 ms sehingga node UE yang akan dibangun terbatas maksimal sebanyak 19 UE per eNB. Kemudian dalam pembangunan node EPC, NS-3 menjadikan fungsi dari SGW dan PGW menjadi satu node. NS-3 juga mendukung pengumpulan data hasil dari pengujian menggunakan aplikasi FlowMonitor.

3.2 Analisis Kebutuhan

Analisis kebutuhan yang dilakukan adalah untuk mendapatkan hasil sesuai dari kebutuhan yang sudah ditetapkan dan dibatasi pada saat mensimulasikan pengujian. Kebutuhan yang diambil berdasarkan kebutuhan fungsional yaitu kebutuhan yang menjadi subjek penelitian dan harus terpenuhi.

Dari studi literatur yang telah dilakukan sebelumnya dapat dirumuskan bahwa dalam penelitian ini dibutuhkan komponen-komponen yang dapat menunjang simulasi yang akan dilakukan. Berikut merupakan kebutuhan fungsional pada perangkat lunak dari penelitian yang akan dilakukan:

1. Sistem operasi Ubuntu 16.04
2. Network Simulator 3
3. Aplikasi FlowMonitor.

Penelitian ini juga dijalankan pada perangkat keras dengan spesifikasi sebagai berikut:

1. Processor : Intel Core i5 3230M
2. RAM : DDR3 4 GB
3. GPU : Intel HD Graphics 4000.

Pada penelitian ini software yang digunakan untuk menjalankan simulasi jaringan LTE dan digunakan untuk mengetahui kinerja penjadwalan paket adalah *Network Simulator 3* (NS-3) yang dijalankan pada sistem operasi Ubuntu 16.04.

Simulator NS-3 memiliki beberapa komponen yang dapat mendukung penelitian ini yang terdiri dari, node, aplikasi jaringan, jenis *handover*, jenis penjadwalan paket, jenis pergerakan, dan paket data. Semua komponen tersebut dapat mendukung dalam perancangan dan pembangunan arsitektur jaringan dalam penelitian yang akan dilakukan pada simulator NS-3 ini. Peran node adalah sebagai sebuah perangkat yang dapat mengirim ataupun menerima suatu paket data, atau dapat dijadikan sebagai perantara yang menghubungkan dua atau lebih perangkat. Paket data yang dibuat dan dikirim merupakan bagian dari komponen aplikasi jaringan. Pada penelitian ini paket data yang digunakan dibuat dengan bantuan dari kelas *UdpClient* yang membuat paket UDP.

Jenis penjadwalan paket yang disediakan pada NS-3 membantu untuk memproses paket-paket data yang masuk. Paket-paket tersebut masuk kedalam suatu daftar peringkat terlebih dahulu untuk memilih paket mana yang kemudian

akan diproses atau dikirimkan sesuai dengan urutan peringkat kepada alamat yang dituju. Jenis penjadwalan paket yang digunakan pada penelitian ini menggunakan algoritme *Round Robin* dan algoritme *Proportional Fair*. Algoritme *Round Robin* berjalan dengan mengurutkan peringkat paket sesuai dengan paket yang lebih dulu masuk dan diberikan *time quantum* sehingga jika *time quantum* tersebut habis atau paket telah selesai diproses maka berlanjut ke paket selanjutnya. Sedangkan algoritme *Proportional Fair* berjalan dengan mengurutkan peringkat paket sesuai nilai CQI setiap paket. Paket dengan nilai CQI tertinggi akan diproses terlebih dahulu pada antrian dan menyusul paket dengan nilai CQI tertinggi setelahnya.

Kemudian untuk jenis pergerakan yang dirancang akan berhubungan terhadap *handover* yang akan digunakan. Pergerakan yang dilakukan UE akan menggunakan pergerakan dengan pergerakan acak dengan area yang dibatasi sedangkan pergerakan dari komponen lain seperti eNB, remoteHost, EPC, dan internet menggunakan jenis pergerakan dengan posisi yang tetap. Dengan pergerakan acak dari UE dan pergerakan posisi tetap dari eNB maka *handover* yang mendukung untuk digunakan adalah *handover* dengan pemicu otomatis jenis RSRQ.

Simulator NS-3 juga memiliki fitur statistik bernama FlowMonitor yang digunakan untuk membaca data-data penting seperti alamat IP, besar paket yang dikirim, besar paket yang diterima, waktu paket mulai dikirim dan mulai diterima, waktu paket selesai dikirim dan selesai diterima, dan lain-lain. Data-data tersebut dibuat secara otomatis oleh fitur FlowMonitor berupa *file* yang berekstensi .xml. Data-data tersebut digunakan untuk menghitung hasil dari kinerja dari algoritme penjadwalan paket dan pengaruh *handover* yang digunakan seperti *throughput*, *packet loss ratio*, *delay*, dan *jitter*.

3.3 Perancangan

Pada tahap perancangan terdapat tiga jenis perancangan yang digunakan yaitu perancangan sistem perancangan pengujian dan perancangan evaluasi. Perancangan sistem dilakukan untuk membentuk infrastruktur jaringan yang akan diuji. Sedangkan perancangan pengujian dilakukan untuk merancang skenario tertentu untuk diterapkan pada pengujian setiap algoritme penjadwalan paket.



Gambar 3.2 Diagram alir perancangan sistem

Perancangan sistem berisi perancangan infrastruktur jaringan serta arsitektur jaringan LTE, topologi jaringan yang akan digunakan, dan peletakan posisi atau persebaran node UE pada setiap eNB. Dimulai dari merancang infrastruktur seperti menentukan komponen-komponen ataupun modul-modul yang akan digunakan serta jumlah dari komponen tersebut.

Perancangan selanjutnya yang dilakukan adalah pembuatan topologi. Topologi yang digunakan untuk menghubungkan antara eNB dan EPC pada penelitian ini adalah topologi yang memungkinkan setiap eNB dapat berhubungan dengan eNB yang berdekatan dan menjadi tetangganya. Hal ini memungkinkan komunikasi antar eNB jika terjadi proses *handover*.

Perancangan selanjutnya adalah jumlah dan peletakan UE maupun eNB. Setiap UE diletakkan secara acak dengan jumlah yang sama pada radius tertentu pada setiap eNB. Peletakan ini dimaksudkan agar tidak adanya *handover* pada awal waktu dan memiliki hubungan dengan eNB yang setara diawal pengujian. Untuk Peletakan eNB yang sejajar ini dibuat dengan tidak terpaku pada perspektif geometri dan sumber daya pada wilayah tertentu.

Kemudian perancangan pergerakan node dan kecepatan node. Untuk menyesuaikan simulasi dengan kenyataan maka node eNB, remote host dan EPC yang dibuat ditempatkan pada posisi yang tetap dan tidak berubah. Lalu untuk membuat data yang bervariasi pergerakan UE akan dilakukan secara acak dengan kecepatan yang sama pada setiap UE.

Perancangan pengujian dilaksanakan melalui beberapa tahapan yaitu dimulai dari merancang durasi simulasi yang akan dilaksanakan, menerapkan mekanisme *handover* kedalam simulasi, mengimplementasikan algoritme penjadwalan paket, mengatur dan membatasi ruang gerak UE, dan merancang jumlah UE yang digunakan pada setiap skenario yang akan dilakukan. Pengujian akan menerapkan beberapa skenario sederhana yang memungkinkan hasil yang didapatkan memenuhi tujuan dari penelitian ini. Skenario yang dirancang akan membedakan hasil dari masing-masing kinerja algoritme penjadwalan paket dengan jumlah dari UE yang berbeda yaitu sebanyak 1 sampai 19 UE per eNB. Skenario selanjutnya akan membedakan pengaruh *handover* terhadap kinerja dari masing-masing algoritme penjadwalan paket dengan jumlah UE yang sama pada skenario sebelumnya yaitu 1 sampai 19 UE per eNB. Perancangan skenario tanpa adanya *handover* dibuat dengan membatasi luas area dari UE untuk bergerak pada masing-masing eNB.

3.4 Implementasi

Tahap implementasi merupakan tahap untuk menerapkan hasil dari perancangan sistem yang telah dibuat kedalam bentuk kode sumber. Pada pengimplementasian ini dilakukan pada simulator NS-3 dengan menggunakan bahasa pemrograman C++.

Pengimplementasian dimulai dari penambahan modul-modul yang akan digunakan untuk membantu simulasi yang akan dilakukan. Modul network yang digunakan memiliki *helper class* NodeContainer dan NetDeviceContainer yang berfungsi untuk membuat node. Kemudian terdapat modul PointToPoint yang digunakan untuk membuat internet ataupun menyambungkan beberapa komponen dengan kabel dan mengatur data rate, *Delay* dan lain-lain pada sambungan tersebut. Kemudian terdapat modul Ite yang memiliki *helper class* yang

membantu untuk menghubungkan UE dengan eNB sehingga UE dapat berkomunikasi dengan jaringan LTE. Terdapat pula modul mobility yang digunakan untuk mengatur pergerakan dari setiap node.

Setelah modul-modul yang dibutuhkan terpasang selanjutnya adalah membangun node yang digunakan sebagai UE dan eNB. Setelah tiap node terbentuk selanjutnya peletakan dan pergerakan tiap node sesuai dengan rancangan topologi dan rancangan peta persebaran UE serta jenis pergerakan dari setiap node. Untuk *helper class* yang digunakan untuk peletakan dan pergerakan node ini adalah *helper class* MobilityHelper dari modul Mobility. Model penempatan yang digunakan pada penelitian ini adalah ConstantPositionMobilityModel untuk posisi dari eNB dan RandomDiscPositionAllocator untuk posisi dari UE. Sedangkan untuk pergerakan dari UE menggunakan RandomWalk2dMobilityModel sehingga UE dapat bergerak bebas dan acak.

Selanjutnya adalah pembangunan remoteHost, EPC, dan Internet. pembangunan remoteHost, EPC, dan Internet tidak terpaku pada peletakan dan yang terpenting adalah fungsi dari komponen-komponen tersebut dapat berjalan dengan baik. Pembangunan EPC diterapkan dengan bantuan dari modul LTE dengan *helper class* LteHelper dengan menerapkan algoritme penjadwalan paket dan algoritme *handover* didalamnya. Pembangunan remoteHost diterapkan melalui bantuan modul Internet. Dan pembangunan Internet dibuat dengan menyambungkan remoteHost tersebut dengan interface EPC yaitu PGW yang dibuat pada saat pembangunan EPC.

Pada pengimplementasian hasil perancangan pengujian, Algoritme penjadwalan paket dan algoritme *handover* dapat diterapkan dengan menggunakan *helper class* LteHelper yang diletakkan pada EPC. Pada penelitian ini algoritme yang digunakan pada penjadwalan paket adalah RrFfMacScheduler untuk algoritme *Round Robin* dan PffFfMacScheduler untuk algoritme ProportionalFair. Algoritme yang dipakai untuk menangani *handover* menggunakan tipe algoritme A2A4RsrqHandoverAlgorithm.

Skenario pengujian dibagi menjadi 2 cabang yaitu terjadinya *handover* dan tidak terjadinya *handover* serta menerapkan kedua algoritme penjadwalan paket kedalam dua cabang tersebut secara terpisah. Pada pengimplementasian terjadinya *handover* ataupun tidak dibedakan dengan batas ruang gerak dari masing-masing UE. Pembatasan ruang UE dapat diatur dalam proses pembangunan node yang digunakan sebagai UE sebelumnya.

Jenis *traffic* yang digunakan untuk berkomunikasi antar UE adalah ConstantBitRate yang menggunakan OnOffApplication dan *helper class* OnOffHelper. OnOffApplication ini dapat membantu untuk membuat packet UDP dengan ukuran dan data rate menjadi konstan. Jenis *traffic* ini kemudian diterapkan pada setiap UE yang dibuat.

Setelah semua infrastruktur dari sistem telah terpenuhi dilanjutkan konfigurasi pada sistem untuk memenuhi atau menyesuaikan dengan skenario pengujian yang akan dilakukan.

3.5 Pengujian dan Analisis

Pada pengujian ini menghasilkan sebuah log dari masing-masing skenario berupa xml *file* yang dihasilkan oleh aplikasi FlowMonitor. Data yang masuk ke dalam log tersebut berupa *timeFirstTxPacket*, *timeFirstRxPacket*, *timeLastTxPacket*, *timeLastRxPacket*, *delaySum*, *jitterSum*, *lastDelay*, *txByte*, *rxByte*, *txPacket*, *rxPacket*, dan *lossPacket* beserta alamat dan port sumber maupun alamat dan port tujuannya. Kemudian dari setiap data tersebut dapat diolah kembali untuk mendapatkan parameter *throughput* sebagai parameter perhitungan bandwidth aktual, *packet loss ratio* sebagai paket yang hilang pada saat komunikasi antar UE, *delay* sebagai parameter lamanya jeda pengiriman untuk sampai pada UE tujuan paket, dan *jitter* sebagai variasi waktu *delay* yang terjadi saat pengiriman ataupun penerimaan paket.

Parameter-parameter tersebut dapat menentukan kualitas dari algoritme penjadwalan paket ketika antar UE sedang dalam kondisi aktif berkomunikasi. Parameter yang terbentuk tersebut akan dianalisis dengan membandingkan hasil dari masing-masing penjadwalan paket dan membandingkan pula antara hasil dari terjadinya ataupun tidak terjadinya *handover*.

3.6 Pengambilan Kesimpulan

Pengambilan kesimpulan dapat dilakukan ketika studi literatur, analisis kebutuhan, perancangan, implementasi, dan pengujian dan analisis telah selesai dilakukan. Kesimpulan dapat diambil berdasarkan hasil yang didapat dari proses pengujian dan analisis serta menjawab rumusan masalah.

BAB 4 PERANCANGAN

4.1 Perancangan Sistem

Sistem yang akan digunakan dibangun dengan simulator NS-3. Komponen-komponen yang wajib ada untuk mendukung kelengkapan infrastruktur jaringan pada NS-3 yaitu berupa eNB, UE, remoteHost, jenis *handover* yang digunakan, dan area pergerakan UE.

4.1.1 Perancangan Infrastruktur Jaringan

Komponen-komponen berupa eNB, UE dan remoteHost dibuat menggunakan node dengan membedakan fungsi dari masing-masing komponen. Node eNB digunakan sebagai perantara komunikasi antar UE dan keduanya dijumpai oleh remoteHost ke dalam jaringan internet. Pada penelitian ini jumlah node eNB yang digunakan sebanyak 3 node dan variasi jumlah node untuk membuat UE adalah sebanyak 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, dan 57 node.

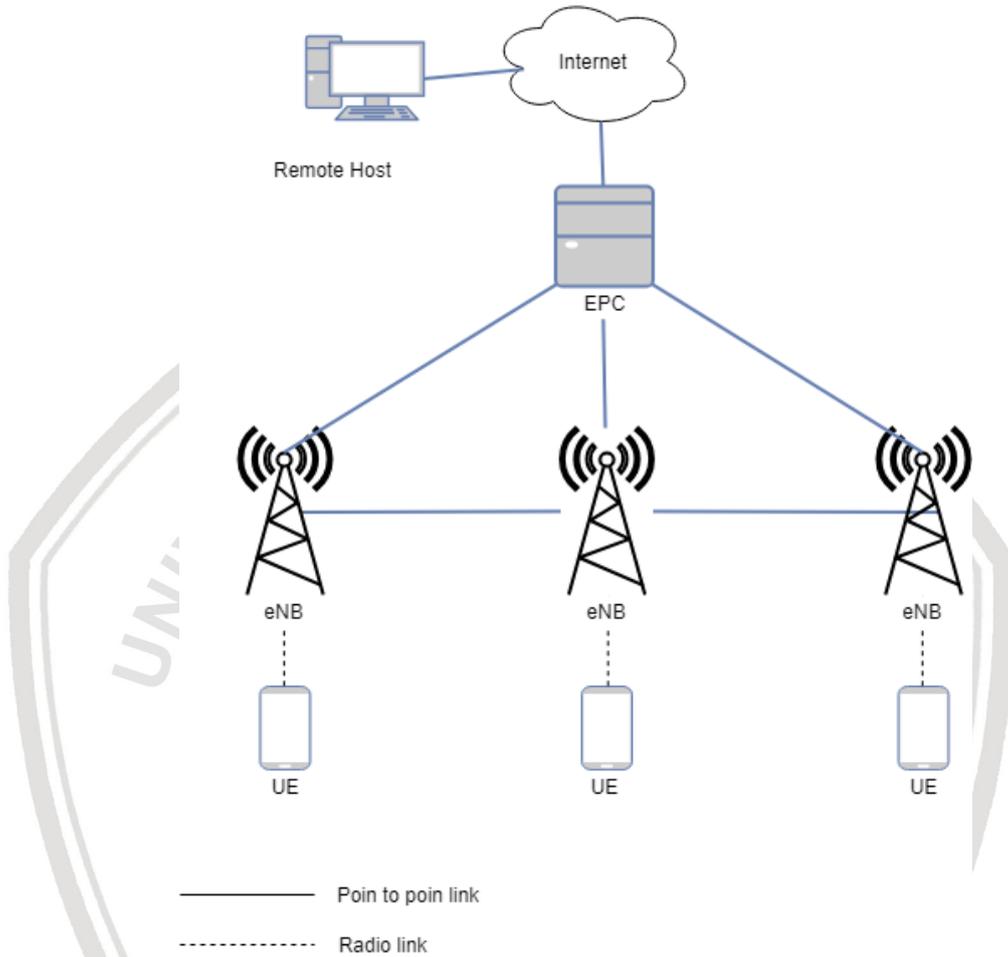
Untuk membentuk komunikasi antar UE diperlukan perantara yang dapat menampung dan menghubungkan antar UE. Pada simulator NS-3 menyediakan NetDevice yang berperan sebagai penghubung sehingga komunikasi antar UE dapat tersambung. Terdapat beragam jenis dari NetDevice namun pada penelitian ini jenis NetDevice yang digunakan adalah PointToPoint. PointToPoint digunakan untuk menyambungkan komponen-komponen pada EPC dan eNB. Setelah node-node tersebut terhubung dilakukan pemberian atribut seperti *data rate*, *mtu*, dan *Delay* dan *value* dari atribut tersebut. Kemudian pemberian alamat IP yang dibantu oleh Ipv4AddressHelper dari modul Network.

Algoritme penjadwalan paket dan *handover* diterapkan dengan bantuan IteHelper pada modul LTE. Selanjutnya menentukan aplikasi jaringan atau jenis *traffic* yang berfungsi untuk menghasilkan paket yang akan dikirim maupun diterima selama berkomunikasi. Terdapat bermacam-macam model yang ditawarkan pada simulator NS-3 seperti BulkSendApplication, OnOffApplication, PacketSink, UdpClientServer, dan UdpEcho. Namun model yang digunakan pada penelitian ini adalah OnOffApplication dengan bantuan UdpClientHelper dan OnOffHelper. UdpClientHelper dan OnOffHelper digunakan untuk membuat node UE bertindak seperti klien yang mengirimkan maupun menerima paket UDP .

4.1.2 Topologi Jaringan

Model topologi jaringan yang akan dibuat diilustrasikan pada Gambar 4.1. Topologi pada penelitian ini dibuat dari adaptasi arsitektur LTE yang telah dijelaskan sebelumnya. UE yang terhubung kepada eNB berjumlah 1 sampai 19 per eNB melalui sinyal radio yang dipancarkan melalui antena-antena UE maupun eNB. Antena yang digunakan UE dan eNB berupa IsotropicAntennaModel. Model antena *isotropic* digunakan karena jangkauan sinyal yang dipancarkan atau diterima menjangkau seluruh arah. Pada topologi yang dibangun, EPC terhubung

ke semua eNB melalui *interface* S1. Hubungan ini dimaksudkan supaya komunikasi pada jaringan LTE dapat terhubung kepada jaringan lain melalui EPC. Sedangkan masing-masing eNB terhubung ke eNB tetangga terdekatnya melalui *interface* X2. Hal ini dimaksudkan untuk mengurangi beban EPC sehingga saat terjadinya *handover* eNB dapat berkomunikasi langsung tanpa harus melewati EPC.



Gambar 4.1 Topologi Jaringan

Peletakan eNB diberikan jarak sejauh 500 meter dari eNB tetangganya. Kemudian UE diletakkan secara berkelompok pada setiap eNB dengan jumlah 1 sampai 19 UE. Peletakan dan pergerakan UE dibatasi dengan panjang area 1500 meter dan lebar area sepanjang 500meter.

4.1.3 Peta Persebaran dan UE pada eNB

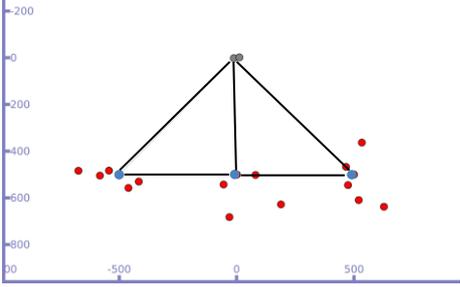
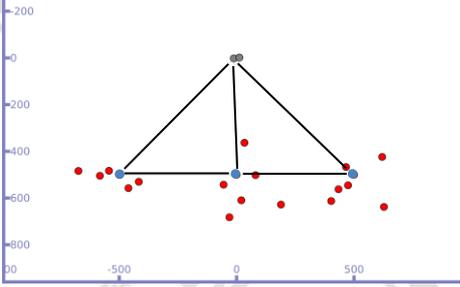
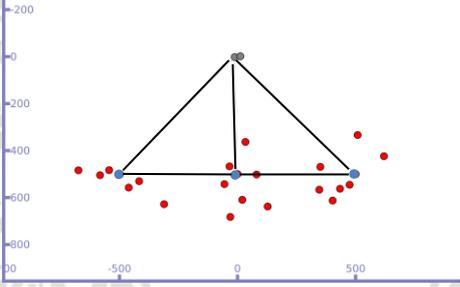
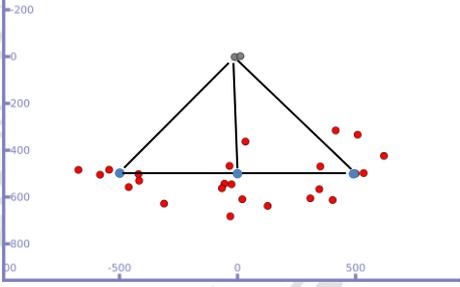
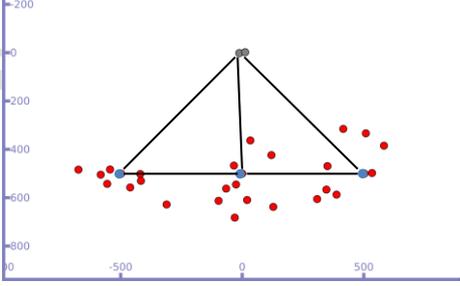
Setiap node ditempatkan dan diatur pergerakannya menggunakan modul yang tersedia pada simulator NS-3 yaitu modul mobility sehingga memudahkan penelitian yang dilakukan. Model yang digunakan adalah model ConstantPositionMobilityModel, RandomDiscPositionAllocator, dan RandomWalk2dMobilityModel.

Penggunaan model ConstantPositionMobilityModel diterapkan pada peletakan eNB sehingga eNB berada posisi yang tidak berubah dan tidak berpindah. Model

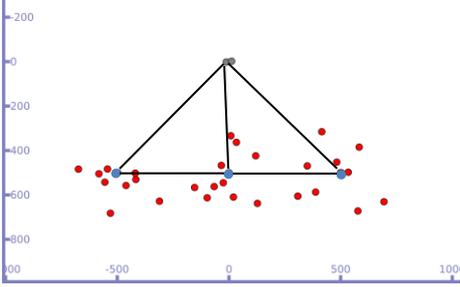
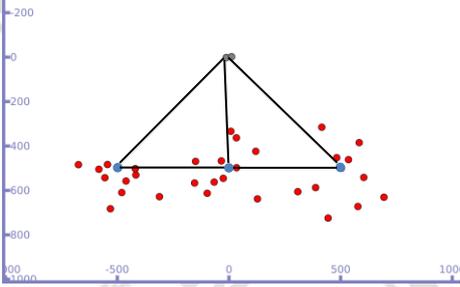
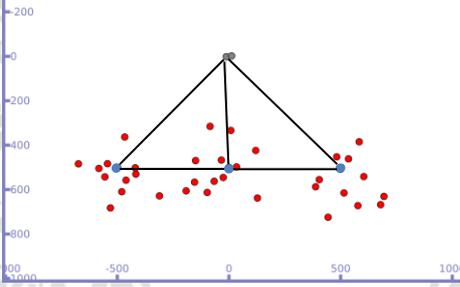
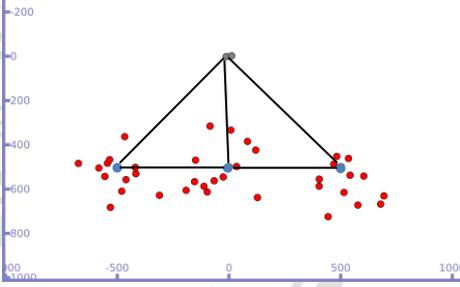
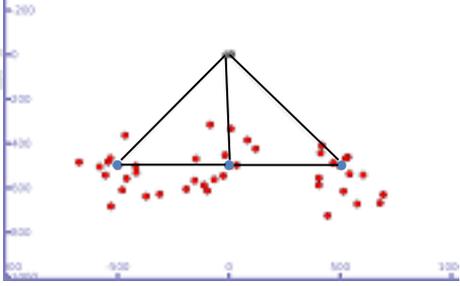
RandomDiscPositionAllocator diterapkan pada UE sehingga pada awal simulasi UE berada pada suatu grup yang berpusat pada satu eNB. Selanjutnya untuk model dari pergerakan UE menggunakan model RandomWalk2dMobilityModel sehingga tiap UE dapat bergerak secara bebas dengan arah yang acak. Variasi dari persebaran UE pada eNB dapat dilihat pada Tabel 4.1, dengan keterangan *node* dengan warna merah adalah UE, *node* dengan warna biru adalah eNB dan *node* dengan warna abu-abu adalah EPC dan remotetHost.

Tabel 4.1 Peta persebaran setiap node

No.	Jumlah UE / eNB	Peta Persebaran
1	1	
2	2	
3	3	
4	4	

No.	Jumlah UE / eNB	Peta Persebaran
5	5	
6	6	
7	7	
8	8	
9	9	



No.	Jumlah UE / eNB	Peta Persebaran
10	10	
11	11	
12	12	
13	13	
14	14	

No.	Jumlah UE / eNB	Peta Persebaran
15	15	
16	16	
17	17	
18	18	
19	19	



4.2 Perancangan Pengujian

Perancangan pengujian yang akan dibuat dapat dilihat pada Tabel 4.2 yang menunjukkan parameter-parameter yang digunakan. Durasi waktu yang akan dilaksanakan untuk melakukan simulasi adalah 600 detik atau 10 menit. Variasi jumlah UE yang diujikan adalah 1 sampai 19 UE per eNB yang ditempatkan secara merata pada setiap eNB. Dalam penelitian ini pergerakan UE dibuat untuk mendukung terjadinya *handover*. Maka jenis pergerakan yang digunakan tiap UE adalah RandomWalk yang memungkinkan tiap UE bergerak ke segala arah dan memungkinkan terjadinya *handover*. Kecepatan yang digunakan UE untuk bergerak adalah 10 meter per detik sehingga memungkinkan UE untuk dapat berpindah hubungan dengan eNB lain atau terjadi proses *handover*.

Semua UE yang telah dibuat tersebut dapat saling terhubung melalui perantara eNB. eNB yang telah dibuat yang diletakkan secara sejajar dan berjumlah 3 eNB dengan jarak antara eNB satu dengan eNB tetangganya adalah 500 meter. UE dan eNB menggunakan jenis antena isotropic. Kekuatan daya yang digunakan tiap eNB pada sinyal yang dipancarkan sebesar 46 Dbm, nilai yang sering diunakan pada sebagian besar penyedia layanan atau produsen (Hamid, et al., 2012).

Protokol yang digunakan adalah *User Datagram Protocol* (UDP). Data yang dikirimkan berupa konten multimedia berupa suara, sehingga jenis *traffic data* yang digunakan adalah *Constant BitRate* (CBR). CBR digunakan agar pada setiap UE menggunakan besaran paket yang sama dan tetap. Hal ini dapat memudahkan dalam pelaksanaan analisis karena nilai *value* dari parameter masukan berawal dari nilai yang sama meskipun hasil yang didapat dapat berbeda-beda.

Pengujian dilakukan dengan menerapkan algoritme penjadwalan paket pada EPC. Paket-paket yang telah dikirimkan akan dijadwalkan sesuai dengan urutan yang ditetapkan oleh tiap-tiap algoritme penjadwalan paket yang diimplementasikan. Algoritme penjadwalan paket yang digunakan adalah algoritme *Round Robin* dan algoritme *Proportional Fair*. Data yang terbentuk dari penerapan algoritme-algoritme tersebut setelah pengujian selesai akan dibandingkan berdasarkan besar *Throhgput*, *Packet Loss Ratio*, rata-rata *End-to-End Delay* dan rata-rata *Jitter*.

Penelitian ini juga menguji pengaruh *handover* terhadap penjadwalan paket yang diterapkan pada sistem. Pengujian ini dilakukan dengan membandingkan hasil dari *Throhgput*, *Packet Loss Ratio*, rata-rata *End-to-End Delay* dan rata-rata *Jitter* pada sistem yang memungkinkan terjadinya *handover* dan yang tidak memperbolehkan UE untuk berpindah eNB. Penerapan tersebut dilakukan dengan membatasi ruang gerak dari UE. Pada skenario yang memungkinkan terjadinya *handover* ruang gerak UE memiliki panjang 1500 meter dan lebar 500 meter. Sedangkan pada skenario yang tidak memperbolehkan *handover* ruang gerak UE pada setiap grup yang berpusat pada satu eNB memiliki panjang dan lebar 500 meter.

Tabel 4.2 Tabel Parameter Pengujian

Parameter	Nilai
Durasi Simulasi	600 detik
Algoritme <i>Handover</i>	RSRQ
Algoritme Penjadwalan Paket	<i>Round Robin</i> dan <i>Proportional Fair</i>
<i>SRS periodicity</i>	40 ms
Ruang Gerak UE	1500 m x 500 m
Jumlah UE	1 sampai 19 UE per eNB
Jenis pergerakan UE	Random Walk
Kecepatan UE	10 mps
Antena UE	Isotropic
Jumlah eNB	3
Jarak eNB	500 m
Kekuatan sinyal eNB	46 Dbm
Antena eNB	Isotropic
Jenis <i>Traffic</i>	CBR

Skenario yang akan digunakan dalam penelitian ini akan digolongkan menjadi 2 yaitu;

1. Simulasi dengan menggunakan *handover*
2. Simulasi dengan tidak menggunakan *handover*

Pada simulasi menggunakan prosedur *handover*, ruang gerak UE akan menggunakan luas 1500 m x 500 m yang didalamnya terdapat 3 eNB sehingga UE dapat bergerak pindah dari eNB satu ke eNB lainnya. Sedangkan pada simulasi dengan tidak menggunakan prosedur *handover*, ruang gerak UE akan menggunakan luas 500 m x 500 m pada setiap kelompok UE di setiap eNB. Hal ini dimaksudkan agar UE tidak dapat berpindah tempat atau tidak dapat melakukan prosedur *handover* pada simulasi.

Setiap simulasi dilakukan 2 pengujian dengan algoritme penjadwalan paket yang berbeda yaitu algoritme penjadwalan paket *Round Robin* dan algoritme penjadwalan paket *Proportional Fair*. Dan pada setiap pengujian akan menggunakan jumlah UE yang berbeda-beda. Dimulai dari UE berjumlah 1 per eNB hingga UE berjumlah 19 per eNB. Sehingga simulasi akan dijalankan sebanyak 76 kali.

4.2.1 Perancangan pengujian terhadap *Throughput*

Pada data hasil yang terbentuk dari penggunaan aplikasi FlowMonitor didapatkan data-data yang rxByte, timeLastRxPacket, dan timeFirstRxPacket. Menurut Carneiro, et al. (2009) data-data tersebut dapat diolah dengan menggunakan Persamaan 4.1 untuk menghasilkan data baru yaitu *throughput* pada tiap komunikasi yang terbentuk dalam satuan *bit per second* (bps).

$$Throughput = \frac{rxBytes \times 8}{timeLastRxPacket - timeFirstRxPacket} \quad (4.1)$$

Throughput-throughput tersebut akan dirataratakan lagi dan kemudian dimasukkan pada tabel perancangan pengujian terhadap *throughput* seperti Tabel 4.3.

Tabel 4.3 Tabel perancangan pengujian *throughput*

UE/eNB	Penjadwalan paket			
	RR	PF	RRHO	PFHO
1				
2				
3				
4				
5				

4.2.2 Perancangan pengujian terhadap *Packet Loss Rasio*

Pada data hasil yang terbentuk dari penggunaan aplikasi FlowMonitor didapatkan data-data yang txPacket dan rxPacket. Menurut Carneiro, et al. (2009) data-data tersebut dapat diolah dengan menggunakan Persamaan 4.2 untuk menghasilkan data baru yaitu *packet loss ratio* pada tiap komunikasi yang terbentuk dalam satuan persen (%).

$$Packet Loss Ratio = \frac{txPackets - rxPackets}{txPackets} \times 100\% \quad (4.2)$$

Rasio-rasio tersebut akan dirataratakan lagi dan kemudian dimasukkan pada tabel perancangan pengujian terhadap *packet loss ratio* seperti Tabel 4.4.

Tabel 4.4 Tabel perancangan pengujian *packet loss ratio*

UE/eNB	Penjadwalan paket			
	RR	PF	RRHO	PFHO
1				
2				
3				
4				
5				

4.2.3 Perancangan pengujian terhadap Rata-rata *End-to-End Delay*

Pada data hasil yang terbentuk dari penggunaan aplikasi FlowMonitor didapatkan data-data yang delaySum dan rxPacket. Menurut Carneiro, et al. (2009) data-data tersebut dapat diolah dengan menggunakan Persamaan 4.3 untuk menghasilkan data baru yaitu *mean delay* pada tiap komunikasi yang terbentuk dalam satuan detik (s).

$$\overline{delay} = \frac{delaySum}{rxPackets} \times \frac{1}{1000000000} \quad (4.3)$$

Mean delay yang terbentuk tersebut akan dirata-ratakan lagi dan kemudian dimasukkan pada tabel perancangan pengujian terhadap rata-rata *end-to-end delay* seperti Tabel 4.5.

Tabel 4.5 Tabel perancangan pengujian *end-to-end delay*

UE/eNB	Penjadwalan paket			
	RR	PF	RRHO	PFHO
1				
2				
3				
4				
5				

4.2.4 Perancangan pengujian terhadap Rata-rata *Jitter*

Pada data hasil yang terbentuk dari penggunaan aplikasi FlowMonitor didapatkan data-data yang JitterSum dan rxPacket. Menurut Carneiro, et al. (2009) data-data tersebut dapat diolah dengan menggunakan Persamaan 4.4 untuk menghasilkan data baru yaitu *mean jitter* pada tiap komunikasi yang terbentuk dalam satuan detik (s).

$$\overline{jitter} = \frac{jitterSum}{rxPackets - 1} \times \frac{1}{1000000000} \quad (4.4)$$

Mean *jitter* yang terbentuk tersebut akan dirata-ratakan lagi dan kemudian dimasukkan pada tabel perancangan pengujian terhadap rata-rata *jitter* seperti Tabel 4.6.

Tabel 4.6 Tabel perancangan pengujian *jitter*

UE/eNB	Penjadwalan paket			
	RR	PF	RRHO	PFHO
1				
2				
3				
4				
5				

BAB 5 IMPLEMENTASI

5.1 Implementasi Sistem

Pengimplementasian perancangan dilakukan dengan menerapkan kode sumber ke dalam simulator NS-3. Penerapan *source code* pada simulator menggunakan bahasa pemrograman C++. Pembangunan infrastruktur yang telah dirancang sebelumnya diimplementasikan dengan membuat *source code* dan memanfaatkan modul-modul yang telah disediakan oleh simulator NS-3.

Penerapan pertama dilakukan dengan memasukkan modul-modul yang mempunyai peran dan kegunaan didalam sistem yang akan dibuat. Modul-modul tersebut dimasukkan dengan melakukan include kedalam kode sumber. Modul-modul yang digunakan dan pemasukannya dalam bentuk *pseudocode* dapat dilihat pada Tabel 5.1. Penerapan *pseudocode* kedalam kode sumber dapat dilihat pada tabel kode sumber Lampiran A pada baris pertama sampai baris ke-9.

Tabel 5.1 *Pseudocode* untuk memasukkan modul pada sistem

1	Masukkan modul Core dari library ns3
2	Masukkan modul Network dari library ns3
3	Masukkan modul Internet dari library ns3
4	Masukkan modul Mobility dari library ns3
5	Masukkan modul LTE dari library ns3
6	Masukkan modul FlowMonitor dari library ns3
7	Masukkan modul Applications dari library ns3
8	Masukkan modul PointtoPoint dari library ns3
9	Masukkan modul ConfigStore dari library ns3

Modul core digunakan pada simulasi ini agar menjalankan ataupun menghentikan simulator dengan menyediakan manajemen waktu simulasi ataupun mengkonversi antar satuan waktu dan dapat menyesuaikan dengan waktu di dunia nyata dengan mengambil waktu dari perangkat yang digunakan untuk menjalankan simulator. Modul core juga dapat digunakan sebagai alat debugging dan digunakan pula mengatur atau mengendalikan atribut-atribut yang digunakan.

Untuk membangun jaringan dapat menggunakan seperti modul network, application, point-to-point, dan internet. Modul network pada simulasi ini digunakan untuk membuat node yang akan diimplementasikan sebagai node UE, eNB, EPC, remoteHost, dan internet. Modul network juga digunakan untuk membantu memberikan peran pada tiap node dengan memberikan interface dan alamat IP kepada node-node yang dibuat. Modul network juga dibantu modul internet yang digunakan untuk mengimplementasikan routing protocol. Pada simulasi ini protocol yang digunakan menggunakan Ipv4. Modul application juga digunakan untuk membantu komunikasi yang akan terbentuk dengan menyediakan jenis *traffic* yang akan digunakan. Modul point-to-point digunakan sebagai penghubung antar perangkat agar dapat tiap perangkat tersebut dapat

berkomunikasi. Modul mobility yang digunakan pada simulasi ini diterapkan ke pada tiap node sehingga tiap node dapat ditentukan lokasinya dan diatur pula pergerakannya. Mobility diterapkan pada node-node yang penting yang diharuskan untuk diatur letak dan pergerakannya seperti UE dan eNB. Module LTE digunakan sebagai jenis jaringan yang digunakan yaitu jaringan LTE. Modul ini juga mengatur node seperti UE, eNB, dan EPC yang memang hanya ada pada jaringan LTE. Modul flowMonitor pada simulasi ini membantu penelitian untuk mendapatkan data statistik. Data-data tersebut akan dihitung kembali untuk mendapatkan parameter perbandingan seperti *Throughput*, *Packet Loss Ratio*, *End-to-End Delay*, dan *Jitter*.

Objek-objek dan nilai atribut dari NS-3 disimpan pada basis data internal tertentu. Modul config-store ini membantu penelitian dengan memungkinkan nilai-nilai tersebut dapat diimpor atau diekspor ke berbagai jenis *file* misalkan *file* dengan format .xml yang dapat digunakan untuk membuat *file* pada modul flowMonitor. Modul ini juga dapat membuat *file* teks mentah yang dapat membantu untuk menampilkan log dari komunikasi yang terbentuk dari simulasi.

5.1.1 Pembangunan UE dan ENB

Setelah modul-modul terpasang maka selanjutnya adalah pembuatan dari node-node yang akan digunakan sebagai UE dan eNB. Pembangunan UE dimulai dengan pembuatan node dengan menggunakan bantuan dari kelas NodeContainer. UE dibuat sebanyak yang telah dirancang sebelumnya yaitu 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, dan 57. Untuk pembuatan UE dapat diimplementasikan dalam bentuk *pseudocode* pada tabel 5.2. Penerapan *pseudocode* untuk membuat node UE dapat dilihat pada baris ke 108, 170 dan 173.

Tabel 5.2 Pseudocode untuk membuat node UE

1	Inisialisasi numberOfUe
2	Buat objek ueNode dari kelas NodeContainer
3	Buat objek ueNode sebanyak numberOfUe

Node UE yang telah berhasil dibuat diletakkan dan diatur pergerakannya untuk memenuhi skenario yang telah direncanakan. Terdapat dua skenario yang membedakan pergerakan dari UE yaitu terjadinya *handover* dan tidak memperbolehkan terjadinya *handover*. Perbedaan penerapan *handover* atau tidaknya adalah pada pembatasan area UE untuk bergerak.

Untuk peletakan UE dilakukan satu demi satu menggunakan perulangan sebanyak jumlah UE yang ditentukan. Peletakan node UE dibuat secara bergrup dengan pusat eNB yang menghubungkan komunikasi antar UE. Untuk setiap grup UE disebar dengan bantuan algoritme RandomDiskPositionAlocator. Algoritme ini membantu meletakkan UE secara acak pada titik koordinat dan radius tertentu.

Peletakan UE pada skenario *handover* ataupun non-*handover* tidak diberikan perbedaan. Pada grup pertama diletakkan pada pusat koordinat $X = -500$ dan $Y =$

500. Pada grup kedua diletakkan pada pusat koordinat $X = 0$ dan $Y = 500$. Dan pada grup ketiga diletakkan pada pusat koordinat $X = 500$ dan $Y = 500$. Radius yang digunakan untuk ketiga grup tersebut sama yaitu sejauh 250 meter. Penerapan pada *pseudocode* dapat dilihat pada Tabel 5.3. Sedangkan penerapannya kedalam kode sumber dapat dilihat pada tabel kode sumber Lampiran A pada baris ke-195 sampai baris ke-224 dan baris ke-240 sampai baris ke-241.

Tabel 5.3 Pseudocode peletakan UE

1	Buat objek ueMobility dari kelas MobilityHelper
2	For i = 0 sampai numberOfUe
3	If nodeUe kurang dari jumlahUe/3
4	Set lokasi ueNode dengan RandomDiskPositionAlocator
5	Set pusat koordinat $X = -500$ dan $Y = 500$
6	Set radius 250 meter
7	Else If nodeUe lebih dari sama dengan jumlahUe/3 dan kurang dari $2*(jumlahUe/3)$
8	Set lokasi ueNode dengan RandomDiskPositionAlocator
9	Set pusat koordinat $X = 0$ dan $Y = 500$
10	Set radius 250 meter
11	Else
12	Set lokasi ueNode dengan RandomDiskPositionAlocator
13	Set pusat koordinat $X = 500$ dan $Y = 500$
14	Set radius 250 meter
15	End If
16	End For
17	Set antena model menjadi IsotropicAntenna
18	Instal perangkat kedalam ueNode

Setelah mengatur peletakan dan pergerakan dari UE hal yang dilakukan adalah memberikan alamat pada masing-masing node UE. Pemberian alamat IP ini dilakukan agar antar UE dapat saling berkomunikasi dengan mengirimkan atau menerima paket dari alamat tujuan dan alamat sumber yang jelas. Pemberian alamat pada node dilakukan dengan melakukan instalasi modul internet pada node UE dan jenis alamat IP yang digunakan adalah ipv4. Penerapan *pseudocode* untuk memberikan alamat IP dapat dilihat pada Tabel 5.4. Sedangkan penerapannya kedalam kode sumber dapat dilihat pada tabel kode sumber Lampiran A pada baris ke-244 sampai baris ke-246.

Tabel 5.4 Pseudocode untuk memberikankan alamat IP pada UE

1	Instal internet pada ueNode
2	Buat objek uelPInterfaces dari kelas Ipv4InterfaceContainer
3	Berikan alamat IP kepada ueNode

Dalam pembangunan eNB terdapat beberapa unsur yang perlu diperhatikan yaitu jarak dan kekuatan sinyal yang akan dipancarkan oleh eNB. Jarak diatur sejauh 500 meter dari eNB satu ke eNB tetangganya. Kekuatan sinyal yang dipancarkan oleh eNB diatur secara default sebesar 46 Dbm. Dan jenis antena yang

digunakan sama dengan jenis antena yang digunakan oleh UE yaitu `IsotropicAntenna`.

Hal pertama yang dilakukan untuk membangun eNB adalah membuat node-nya terlebih dahulu. Node eNB dibuat dengan bantuan kelas `NodeContainer` dan dibuat sebanyak 3 buah. Untuk pergerakan eNB, diatur menggunakan `ConstantPosition`. Pergerakan jenis ini dimaksudkan agar eNB tidak bergerak atau berubah tempat dari posisi semulanya. ENB pertama diletakkan pada posisi koordinat $X = -500$ dan $Y = 500$. ENB kedua diletakkan pada posisi sejauh 500 meter dari eNB pertama yaitu pada koordinat $X = 0$ dan $Y = 500$. Dan eNB ketiga diletakkan sejauh 500 meter dari eNB kedua yaitu pada koordinat $X = 500$ dan $Y = 500$. Setelah posisi dari eNB terbentuk maka selanjutnya penginstalasian antena dan kekuatan sinyal yang dipancarkan. Penerapan pembangunan eNB dalam bentuk *pseudocode* dapat dilihat pada Tabel 5.5. Sedangkan penerapannya kedalam kode sumber dapat dilihat pada tabel kode sumber Lampiran A pada baris ke-109, baris ke-171, baris ke-176 sampai baris ke-189, baris ke-236 sampai baris ke-238.

Tabel 5.5 Pseudocode pembuatan eNB

1	Inisialisasi distance
2	Inisialisasi enbTxPowerDbm
3	Inisialisasi numberOfEnb
4	Buat objek enbNode dari kelas NodeContainer
5	Buat objek enbNode sebanyak numberOfEnb
6	Buat objek enbPositionAlloc
7	For i = 0 sampai numberOfEnb
8	Buat objek enbPosition dari kelas Vector dan set koordinat X dan Y
9	Simpan posisi kedalam objek enbPositionAlloc
10	End for
11	Buat objek enbMobility dari kelas MobilityHelper
12	Set model pergerakan enbMobility dengan ConstantPosition
13	Set posisi dari enb dengan koordinat yang tersimpan dalam enbPositionAlloc
14	Instal enbNode
15	Set TxPower sebesar enbTxPowerDbm
16	Set antena model menjadi IsotropicAntenna
17	Instal perangkat kedalam enbNode

Selanjutnya adalah menghubungkan antara UE dan eNB dengan fungsi `attach` pada pada kode sumber. Penentuan UE dan eNB mana yang dihubungkan diatur dengan membagi grup UE berdasarkan pusat eNBnya. Dengan demikian untuk grup UE pertama akan dihubungkan dengan eNB yang pertama, grup UE yang kedua akan dihubungkan dengan eNB yang kedua dan grup UE yang ketiga akan dihubungkan pada eNB yang ketiga. Penerapan *pseudocode* untuk menghubungkan UE dan eNB dapat dilihat pada Tabel 5.6. Sedangkan penerapannya kedalam kode sumber dapat dilihat pada tabel kode sumber Lampiran A pada baris ke-249 sampai baris ke-259.

Tabel 5.6 Pseudocode untuk menghubungkan UE dan eNB

1	For i = 0 sampai numberOfUe
2	If ueNode kurang dari jumlahUe/3
3	Hubungkan ueNode ke-i ke enbNode ke-0
4	Else If nodeUe lebih dari sama dengan jumlahUe/3 dan kurang dari 2*(jumlahUe/3)
5	Hubungkan ueNode ke-i ke enbNode ke-1
6	Else
7	Hubungkan ueNode ke-i ke enbNode ke-2
8	End If
9	End For

5.1.2 Pembangunan EPC dan RemoteHost

Pembangunan EPC digunakan untuk menerapkan jaringan LTE pada sistem. Pembangunan EPC dimulai dari membuat objek *LteHelper* yang akan dihubungkan atau digunakan sebagai node yang berfungsi sebagai EPC. Penerapan algoritme penjadwalan paket dan algoritme *handover* dimasukkan kedalam EPC. Penerapan algoritme penjadwalan paket dapat dilakukan dengan memasukkan pada kode sumber *RrFfMacScheduler* untuk algoritme *Round Robin* dan *PfFfMacScheduler* untuk algoritme *Proportional Fair*. Dan untuk penerapan algoritme *handover* dapat dilakukan dengan memasukkan *A2A4RsrqHandoverAlgorithm* ke dalam kode sumber untuk menggunakan algoritma RSRQ. Kemudian dibuat node *pgw* sebagai *interface* yang mengatur jaringan dengan jenis yang berbeda. Penerapan *Pseudocode* untuk membangun EPC dapat dilihat pada Tabel 5.7. Sedangkan penerapannya kedalam kode sumber dapat dilihat pada tabel kode sumber Lampiran A pada baris ke-131 sampai baris ke-143.

Tabel 5.7 Pseudocode pembuatan EPC

1	Buat objek <i>LteHelper</i> dari kelas <i>LteHelper</i>
2	Buat objek <i>epcHelper</i> dari kelas <i>PointToPointEpcHelper</i>
3	Set <i>epcHelper</i> kedalam <i>LteHelper</i>
4	Set tipe penjadwalan
5	Set tipe <i>Handover</i>
6	Set <i>ServingCellThreshold</i> menjadi 30
7	Set <i>NeighbourCellOffset</i> menjadi 1
8	Buat Node <i>pgw</i>

Untuk membantu kerja dari sistem LTE dan membangun jaringan internet dibuat remoteHost. Pembuatan remoteHost sebanyak 1 buah dan menginstal internet serta memberikan alamat IP. Pembuatannya remoteHost menggunakan remoteHostContainer. Penerapannya dalam bentuk *pseudocode* dapat dilihat pada tabel 5.8. Sedangkan penerapannya kedalam kode sumber dapat dilihat pada tabel kode sumber Lampiran A pada baris ke- sampai baris ke-.

Tabel 5.8 Pseudocode pembuatan remoteHost

1	Inisialisasi remoteHostContainer
2	Buat Objek remoteHostContainer sebanyak 1
3	Buat node remoteHost dari objek remoteHostContainer
4	Inisialisasi internet
5	Instal internet pada remoteHostContainer

Selanjutnya adalah membangun simulasi untuk seakan-akan terhubung ke internet atau jaringan lain. Pembangunan internet dibuat dengan menghubungkan bagian EPC yang menjembatani jaringan LTE ke jaringan lainnya yaitu PGW dengan remoteHost. Objek yang menghubungkan antara PGW dan remoteHost dibuat dengan bantuan dari kelas PointToPointHelper dengan menambahkan atribut data rate sebesar 100Gb/s, atribut maximum transmission unit (MTU) sebesar 1500 byte, dan propagation *Delay* sebesar 0.01 detik. Pada pembangunan internet ini juga dilakukan pemberian alamat pada remote host dengan alamat ipv4 1.0.0.0 dan mask 255.0.0.0. Penerapannya pada *pseudocode* dapat dilihat pada Tabel 5.9. Sedangkan penerapannya kedalam kode sumber dapat dilihat pada tabel kode sumber Lampiran A pada baris ke-153 sampai baris ke-161.

Tabel 5.9 Pseudocode pembangunan Internet

1	Buat objek p2ph dari kelas PointToPointHelper
2	Set Data Rate sebesar 100Gb/s
3	Set Mtu sebesar 1500 byte
4	Set <i>Delay</i> 0.010 detik
5	Instal dan hubungkan pgw dengan remoteHost dengan objek
6	internetDevice
7	Buat objek ipv4h
8	Set Base ke alamat 1.0.0.0 dan mask 255.0.0.0
9	Terapkan ipv4h ke internetDevice
10	Beri alamat remoteHost

5.2 Implementasi Pengujian

5.2.1 Implementasi Pergerakan UE

Untuk pergerakan UE menggunakan metode Random Walk. Random Walk digunakan agar setiap UE dapat bergerak dengan arah yang acak. Kecepatan gerak

yang digunakan adalah 10m/s. Namun untuk skenario menggunakan *handover* dan tidak terdapat perbedaan pada ruang geraknya. Untuk skenario *handover* area untuk seluruh UE bergerak adalah sama yaitu dengan nilai X minimal ada pada titik -750, X maksimal ada pada titik 750, Y minimal ada pada titik 250 dan Y maksimal ada pada titik 750. Penerapannya pada *pseudocode* dapat dilihat pada Tabel 5.3. Sedangkan pada skenario non-*handover* area gerak UE dibagi sesuai grup masing-masing. Pada grup pertama pergerakan UE dibatasi pada ruang X minimal ada pada titik -750, X maksimal ada pada titik 250, Y minimal ada pada titik 250 dan Y maksimal ada pada titik 750. Pada grup kedua pergerakan UE dibatasi pada ruang X minimal ada pada titik 250, X maksimal ada pada titik 500, Y minimal ada pada titik 250 dan Y maksimal ada pada titik 750. Sedangkan grup ketiga pergerakan UE dibatasi pada ruang X minimal ada pada titik 500, X maksimal ada pada titik 750, Y minimal ada pada titik 250 dan Y maksimal ada pada titik 750. Penerapannya pada *pseudocode* dapat dilihat pada Tabel 5.10. Sedangkan penerapannya kedalam kode sumber dapat dilihat pada tabel kode sumber Lampiran A pada baris ke-225 sampai baris ke-231.

Tabel 5.10 Pseudocode implementasi pergerakan UE pada skenario *handover*

1	Set pergerakan ueNode menjadi RandomWalk
2	Set waktu mulai pergerakan menjadi 0.5 detik setelah simulasi
3	dijalankan
4	Set kecepatan menjadi konstan 10m/s
5	Set area pergerakan UE dengan koordinat X minimal = -750, X maksimal
6	= 750, Y minimal = 250, dan Y maksimal = 750
	Instal ueNode

Penerapan pembatasan ruang area gerak UE dapat dilihat pada Tabel 5.11. Pembatasan ini memungkinkan untuk tidak terjadinya *handover* karena ketika area pada UE terbatas sehingga tidak dapat masuk kedalam sel eNB ataupun menangkap sinyal yang kekuatannya melebihi kekuatan sinyal eNB tempat UE tersebut terhubung.

Tabel 5.11 Pseudocode implementasi pergerakan UE pada skenario non-*handover*

1	For i = 0 sampai numberOfUe
2	If ueNode kurang dari jumlahUe/3
3	Set pergerakan ueNode menjadi RandomWalk
4	Set waktu mulai pergerakan menjadi 0.5 detik setelah simulasi
	dijalankan
5	Set kecepatan menjadi konstan 10m/s
6	Set area pergerakan ueNode dengan koordinat X minimal = -750, X
	maksimal = -250, Y minimal = 250, dan Y maksimal = 750
7	Else If nodeUe lebih dari sama dengan jumlahUe/3 dan kurang dari
	2*(jumlahUe/3)
8	Set pergerakan ueNode menjadi RandomWalk
9	Set waktu mulai pergerakan menjadi 0.5 detik setelah simulasi
	dijalankan

10	Set kecepatan menjadi konstan 10m/s
11	Set area pergerakan ueNode dengan koordinat X minimal = -250, X maksimal = 250, Y minimal = 250, dan Y maksimal = 750
12	Else
13	Set pergerakan ueNode menjadi RandomWalk
14	Set waktu mulai pergerakan menjadi 0.5 detik setelah simulasi dijalankan
15	Set kecepatan menjadi konstan 10m/s
16	Set area pergerakan ueNode dengan koordinat X minimal = 250, X maksimal = 750, Y minimal = 250, dan Y maksimal = 750
17	End If
18	End For
19	Instal ueNode

5.2.2 Implementasi Algoritme Penjadwalan Paket dan *Handover*

Pengimplementasian Algoritme Penjadwalan paket dapat dilakukan dengan memasukkan kode Algoritme penjadwalan paket pada EPC yang telah dibuat karena simulator NS-3 sudah memiliki modul penjadwalan paket untuk diimplementasikan secara langsung tanpa membuat algoritme penjadwalan paket dari awal. Jika algoritme yang akan digunakan adalah algoritme penjadwaan paket Round Robin maka dapat dituliskan pada program dengan kode sumber "lteHelper->SetSchedulerType ("ns3::RrFfMacScheduler");". Sedangkan untuk algoritme penjadwaan Proportional Fair dapat dituliskan pada program dengan kode sumber "lteHelper->SetSchedulerType ("ns3::PffFMacScheduler");".

Untuk pengimplementasian *handover* dapat dilakukan juga dengan langsung memasukkan kode algoritme *handover* pada EPC. Pemasukkan kode tersebut dapat dilakukan dengan menuliskan kode sumber "lteHelper->SetHandoverAlgorithmType ("ns3::A2A4RsrqHandoverAlgorithm");".

Penerapan kode sumber algoritme penjadwalan paket tersebut dapat dilihat pada tabel kode sumber Lampiran A pada baris 134. Dan penerapan kode sumber algoritme *handover* dapat dilihat pada tabel kode sumber Lampiran A pada baris 137.

5.2.3 Pembangunan *Traffic*

Pembangunan *traffic* dengan jenis *Constant BitRate* (CBR) dapat dilakukan dengan menginisialisasika port yang digunakan untuk download dan upload dan port lainnya untuk membantu pembangunan *traffic* ini. Kemudian dibutuhkan pula objek sebagai aplikasi yang digunakan. Dan pada simulasi ini aplikasi yang digunakan adalah sebagai server, client dan aplikasi lain yaitu onOff. Kemudian menginisiaisasi objek packetSink untuk port downlaod, upoad dan lainnya dari kelas packetSinkHelper dan menambahkan Ipv4 kedalamnya. Setelah itu

menambahkan objek-objek tersebut yang diterapkan kepada ueNode dan remoteHost kedalam objek aplikasi serverApps yang sudah dibuat sebelumnya.

Untuk objek client dibuat menggunakan bantuan dari kelas UdpClientHelper. Obje-objek ini dibuat juga dengan mempertimbangkan jenis port yang sudah disebutkan sebelumnya dan diterapkan kepada alamat dari UE dan alamat dari remoteHost. Atribut yang digunakan untuk pembangunan client adalah interval dan maxPacket.

Kemudian membangun *traffic* CBR dengan bantuan onOffHelper dengan membuat objek dan memberikan alamat dan port seperti pembuatan clien sebelumnya dan atribut yang diperlukan adalah OnTime, OffTime, DataRate, PacketSize. Selanjutnya objek yang telah dibuat tersebut ditambahkan kedalam clientApps dan onOffApps yang telah dibuat sebelumnya dan telah diterapkan kedalam ueNode dan remoteHost. Pembangunan *traffic* CBR ini dapat dilihat pada Tabel 5.12. Sedangkan penerapannya kedalam kode sumber dapat dilihat pada tabel kode sumber Lampiran A pada baris ke-264 sampai baris ke-312.

Tabel 5.12 Pseudocode pembangunan Traffic Constant BitRate

1	Inisiaisasi dIPort
2	Inialisasi ulPort
3	Inialisasi otherPort
4	Buat objek clientApps dari kelas ApplicationContainer
5	Buat objek serverApps dari kelas ApplicationContainer
6	Buat objek onOffApp dari kelas ApplicationContainer
7	For u=0 sampai banyakdari ueNode
8	Tambah ulPort dengan 1
9	Tambah otherPort dengan 1
10	Inialisasi dIPacketSinkHelper dengan protokol UDP pada alamat IP apapun dan pada dIPort
11	Inialisasi ulPacketSinkHelper dengan protokol UDP pada alamat IP apapun dan pada ulPort
12	Inialisasi PacketSinkHelper dengan protokol UDP pada alamat IP apapun dan pada otherPort
13	Instal dIPacketSinkHelper pada ueNode dan tambahkan ke serverApps
14	Instal ulPacketSinkHelper pada remoteHost dan tambahkan ke serverApps
15	Instal packetSinkHelper pada ueNode dan tambahkan ke serverApps
16	Buat objek dIClient dari alamat uelpifaces dan port dIPort
17	Set atribut interval dan maxpacket dari dIClient
18	Buat objek ulClient dari alamat remoteHostAddr dan port ulPort
19	Set atribut interval dan maxpacket dari ulClient
20	Buat objek client dari alamat uelpifaces dan port otherPort
21	Set atribut interval dan maxpacket dari client
22	Inialisasi onOffHelper dengan protokol UDP pada alamat IP apapun dan pada ulPort
23	

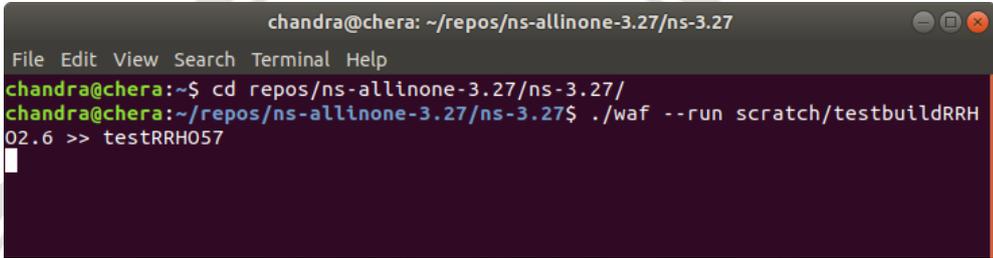
24	Set atribut OnTime, OffTime, DataRate, dan PacketSize pada objek onOffHelper
25	Instal onOffHelper pada ueNode dan tambahkan ke onOffApp
26	Instal dlClient pada remoteHost dan tambahkan ke clinetApps
27	Instal ulClient pada ueNode dan tambahkan ke clinetApps
28	If u+1 kurang dari jumlah ueNode
29	Instal client pada ueNode ke-u+1 dan tambahkan ke clientApps
30	Else
31	Instal client pada ueNode ke-0 dan tambahkan ke clientApps
32	End If
	End For



BAB 6 PENGUJIAN DAN ANALISIS

6.1 Pengujian

Pengujian merupakan tahap setelah kode sumber telah diimplementasikan dari *pseudocode* yang telah dibuat pada bab sebelumnya. Pelaksanaan pengujian dilakukan dengan menjalankan simulasi dari kode sumber yang telah dibuat. Simulasi dijalankan pada NS-3 dengan masuk ke folder ns-3.27 dan menuliskan perintah “`./waf --run`” dan menambahkan folder letak *file* kode sumber beserta nama *file* kode sumber. Untuk menyimpan hasil dari aktivitas UE dan eNB saat simulasi dijalankan dapat dilakukan dengan menambahkan simbol “`>>`” pada perintah di terminal dan diikuti dengan nama *file* yang dibuat. Ketika simulasi berhasil dijalankan maka akan seperti Gambar 6.1.



```
chandra@chera: ~/repos/ns-allinone-3.27/ns-3.27
File Edit View Search Terminal Help
chandra@chera:~$ cd repos/ns-allinone-3.27/ns-3.27/
chandra@chera:~/repos/ns-allinone-3.27/ns-3.27$ ./waf --run scratch/testbuildRRH
02.6 >> testRRH057
```

Gambar 6.1 Contoh pelaksanaan simulasi

Setelah simulasi selesai dijalankan maka akan terbentuk dua buah *file*. Salah satu dari dua *file* tersebut adalah *file* dengan format .xml yang merupakan hasil pengujian dan terdiri dari variabel-variabel seperti *timeFirstTxPacket*, *timeFirstRxPacket*, *timeLastTxPacket*, *timeLastRxPacket*, *delaySum*, *jitterSum*, *lastDelay*, *txByte*, *rxByte*, *txPacket*, *rxPacket*, dan *lossPacket* beserta alamat dan port sumber maupun alamat dan port tujuannya.

File dengan format .xml tersebut dapat digunakan untuk menganalisis hasil dari pengujian yang telah dilakukan. Penganalisisan hasil dapat dilakukan dengan cara menerapkan hasil perancangan analisis pada Bab 4, yaitu prancangan analisis terhadap parameter *throughput*, *packet loss ratio*, *delay*, dan *jitter*. Hasil yang dituliskan pada setiap sel dalam tabel merupakan rata-rata dari setiap parameter yang terbentuk dalam satu kali pengujian simulasi.

Sedangkan *file* lainnya berisi riwayat aktivitas dari UE dan eNB termasuk informasi tentang *handover* yang terjadi jika skenario yang dijalankan adalah skenario dengan adanya *handover*. Informasi tersebut digunakan untuk mengetahui aktifitas UE dimulai dari permintaan hubungan kepada eNB sampai dengan eNB menyatakan bahwa proses menghubungkan komunikasi tersebut telah sukses. Informasi lain yang bisa didapatkan dari *file* tersebut adalah proses terjadinya *handover* dimulai dari fase persiapan sampai pada fase penyelesaian beserta waktu terjadinya fase-fase tersebut. Dua *file* tersebut terlihat seperti pada Gambar 6.2 dan Gambar 6.3.

```

TestBuildRR57HO.xml
~/Documents/RRHO
Open Save
<?xml version="1.0" ?>
<FlowMonitor>
  <FlowStats>
    <Flow flowId="1" timeFirstTxPacket="+1000000.0ns"
timeFirstRxPacket="+142999999.0ns" timeLastTxPacket="+60091000000.0ns"
timeLastRxPacket="+60094099999.0ns" delaySum="+18038893991.0ns"
jitterSum="+741200000.0ns" lastDelay="+30999999.0ns" txBytes="6322520"
rxBytes="6321468" txPackets="6010" rxPackets="6009" lostPackets="1"
timesForwarded="6009">
    </Flow>
    <Flow flowId="2" timeFirstTxPacket="+1000000.0ns"
timeFirstRxPacket="+136999999.0ns" timeLastTxPacket="+60091000000.0ns"
timeLastRxPacket="+60094799999.0ns" delaySum="+200943993991.0ns"
jitterSum="+704100000.0ns" lastDelay="+37999999.0ns" txBytes="6322520"
rxBytes="6321468" txPackets="6010" rxPackets="6009" lostPackets="1"
timesForwarded="6009">
    </Flow>
    <Flow flowId="3" timeFirstTxPacket="+1000000.0ns"
timeFirstRxPacket="+133999999.0ns" timeLastTxPacket="+60091000000.0ns"
timeLastRxPacket="+60094099999.0ns" delaySum="+16648093991.0ns"
jitterSum="+952500000.0ns" lastDelay="+30999999.0ns" txBytes="6322520"
rxBytes="6321468" txPackets="6010" rxPackets="6009" lostPackets="1"
timesForwarded="6009">
    </Flow>
    <Flow flowId="4" timeFirstTxPacket="+1000000.0ns"
timeFirstRxPacket="+139999999.0ns" timeLastTxPacket="+60091000000.0ns"
timeLastRxPacket="+60094799999.0ns" delaySum="+109414996956.0ns"
jitterSum="+507400000.0ns" lastDelay="+37999999.0ns" txBytes="6322520"
rxBytes="3202288" txPackets="6010" rxPackets="3044" lostPackets="2896"
timesForwarded="3044">
    </Flow>
    <Flow flowId="5" timeFirstTxPacket="+1000000.0ns"
timeFirstRxPacket="+140999999.0ns" timeLastTxPacket="+60091000000.0ns"
timeLastRxPacket="+58596799999.0ns" delaySum="+68919997993.0ns"
jitterSum="+273500000.0ns" lastDelay="+57999999.0ns" txBytes="6322520"
rxBytes="2111364" txPackets="6010" rxPackets="2007" lostPackets="3903"
timesForwarded="2007">
    </Flow>
  </FlowStats>
</FlowMonitor>
XML Tab Width: 8 Ln 1, Col 1 INS
  
```

Gambar 6.2 Contoh file xml yang terbentuk

```

testHO57
~/Documents/RRHO/logRR
Open Save
Waf: Entering directory `/home/chandra/repos/ns-allinone-3.27/ns-3.27/build'
[ 962/2010] Compiling scratch/testbuildRRHO2.6.cc
[1941/2010] Linking build/scratch/testbuildRRHO2.6
Waf: Leaving directory `/home/chandra/repos/ns-allinone-3.27/ns-3.27/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (12.413s)
/NodeList/12/DeviceList/0/LteUeRrc/ConnectionEstablished UE IMSI 8:
connected to CellId 1 with RNTI 2 Time: 0.0202143
/NodeList/17/DeviceList/0/LteUeRrc/ConnectionEstablished UE IMSI 13:
connected to CellId 1 with RNTI 3 Time: 0.0202143
/NodeList/21/DeviceList/0/LteUeRrc/ConnectionEstablished UE IMSI 17:
connected to CellId 1 with RNTI 4 Time: 0.0202143
/NodeList/22/DeviceList/0/LteUeRrc/ConnectionEstablished UE IMSI 18:
connected to CellId 1 with RNTI 1 Time: 0.0202143
/NodeList/25/DeviceList/0/LteUeRrc/ConnectionEstablished UE IMSI 21:
connected to CellId 2 with RNTI 2 Time: 0.0202143
/NodeList/28/DeviceList/0/LteUeRrc/ConnectionEstablished UE IMSI 24:
connected to CellId 2 with RNTI 3 Time: 0.0202143
/NodeList/33/DeviceList/0/LteUeRrc/ConnectionEstablished UE IMSI 29:
connected to CellId 2 with RNTI 1 Time: 0.0202143
/NodeList/41/DeviceList/0/LteUeRrc/ConnectionEstablished UE IMSI 37:
connected to CellId 2 with RNTI 4 Time: 0.0202143
/NodeList/47/DeviceList/0/LteUeRrc/ConnectionEstablished UE IMSI 43:
connected to CellId 3 with RNTI 2 Time: 0.0202143
/NodeList/49/DeviceList/0/LteUeRrc/ConnectionEstablished UE IMSI 45:
connected to CellId 3 with RNTI 3 Time: 0.0202143
/NodeList/52/DeviceList/0/LteUeRrc/ConnectionEstablished UE IMSI 48:
connected to CellId 3 with RNTI 1 Time: 0.0202143
/NodeList/57/DeviceList/0/LteUeRrc/ConnectionEstablished UE IMSI 53:
connected to CellId 3 with RNTI 4 Time: 0.0202143
/NodeList/2/DeviceList/0/LteEnbRrc/ConnectionEstablished eNB CellId 1:
successful connection of UE with IMSI 8 RNTI 2 Time: 0.0202143
/NodeList/2/DeviceList/0/LteEnbRrc/ConnectionEstablished eNB CellId 1:
successful connection of UE with IMSI 8 RNTI 2 Time: 0.0202143
/NodeList/2/DeviceList/0/LteEnbRrc/ConnectionEstablished eNB CellId 1:
successful connection of UE with IMSI 13 RNTI 3 Time: 0.0202143
/NodeList/2/DeviceList/0/LteEnbRrc/ConnectionEstablished eNB CellId 1:
successful connection of UE with IMSI 17 RNTI 4 Time: 0.0202143
Plain Text Tab Width: 8 Ln 1, Col 1 INS
  
```

Gambar 6.3 Contoh file aktivitas UE dan eNB yang terbentuk



6.2 Hasil Pengujian

6.2.1 Hasil Pengujian Terhadap *Throughput*

Hasil pengujian terhadap *throughput* dapat dilihat pada Tabel 6.1. Pada tabel tersebut dapat dilihat bahwa *throughput* cenderung terjadi penurunan pada setiap penambahan UE pada masing-masing eNB.

Pada hasil *throughput* tanpa adanya *handover*, algoritme *Round Robin* mengalami penurunan dengan rata-rata 3,58% sedangkan algoritme *Proportional Fair* mengalami penurunan dengan rata-rata sebesar 2,97%. Pada hasil *throughput* dengan adanya *handover* pada simulasi, algoritme *Round Robin* mengalami penurunan dengan rata-rata sebesar 3,73% sedangkan algoritme *Proportional Fair* mengalami penurunan dengan rata-rata 2,92%.

Tabel 6.1 Hasil *Throughput* untuk setiap node

UE/eNB	Penjadwalan paket			
	RR (bps)	PF (bps)	RRHO (bps)	PFHO (bps)
1	84173,821	84173,821	84173,821	84173,821
2	79405,748	79405,806	79405,748	79405,806
3	74771,650	77557,811	74644,941	77917,732
4	74927,500	77758,773	74513,811	75696,715
5	67957,030	69682,631	66680,958	69093,138
6	61908,866	65718,246	61064,084	64890,656
7	60834,485	67009,342	61033,092	67287,781
8	59646,367	65293,254	59555,259	63362,523
9	55727,057	59730,642	54776,954	58746,081
10	55107,534	59074,635	54069,022	58199,567
11	53413,726	57581,028	52005,512	56470,860
12	53535,084	58110,919	52025,560	56935,664
13	49570,859	54630,517	48696,413	53548,502
14	56139,428	61038,729	53799,000	59179,267
15	44020,646	50371,270	44354,817	50399,639
16	45962,122	51399,139	45283,136	51697,998
17	45308,402	51949,698	44354,157	51760,257
18	44123,810	50229,327	42686,786	49789,190
19	41772,631	47238,499	41149,770	48034,093

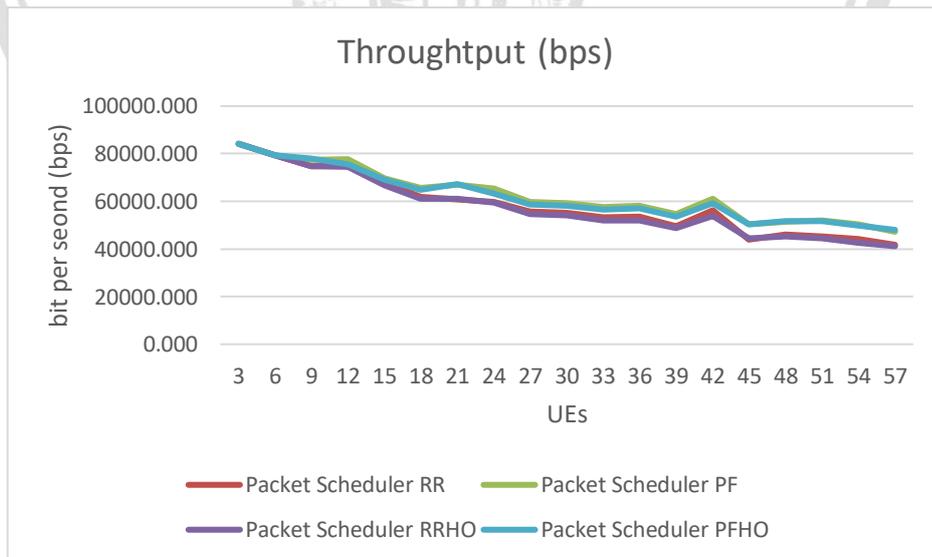
Pada algoritme penjadwalan paket *Round Robin* tanpa terjadi *handover* terlihat bahwa *throughput* cenderung menurun pada setiap penambahan UE, penurunan *throughput* terbesar terjadi pada UE per eNB berjumlah 15 buah dengan hasil *throughput* sebesar 44020,646 bps, yaitu sebesar 21,59% dari *throughput* yang dihasilkan sebelumnya pada saat UE per eNB berjumlah 14 buah dengan hasil *throughput* sebesar 56139,428 bps. Begitu juga pada algoritme penjadwalan paket *Round Robin* pada skenario terjadinya *handover*. Nilai *throughput* cenderung menurun pula pada setiap penambahan UE per eNB.

Penurunan terbesar terjadi pada saat UE per eNB berjumlah 14 dengan hasil *throughput* sebesar 53799,000 bps ke jumlah UE per eNB sebanyak 15 buah dengan hasil *throughput* sebesar 44354,817 bps, yaitu sebesar 17,55%.

Pada algoritme penjadwalan paket *Proportional Fair* tanpa terjadi *handover* terlihat bahwa *throughput* juga cenderung menurun pada setiap penambahan UE, penurunan *throughput* terbesar terjadi pada UE per eNB berjumlah 15 buah dengan hasil *throughput* sebesar 50371,270 bps, yaitu sebesar 17,48% dari *throughput* yang dihasilkan sebelumnya pada saat UE per eNB berjumlah 14 buah dengan hasil *throughput* sebesar 61038,729bps. Begitu juga pada algoritme penjadwalan paket *Proportional Fair* pada skenario terjadinya *handover*. Nilai *throughput* cenderung menurun pula pada setiap penambahan UE per eNB. Penurunan terbesar terjadi pada saat UE per eNB berjumlah 14 dengan hasil *throughput* sebesar 59179,267 bps ke jumlah UE per eNB sebanyak 15 buah dengan hasil *throughput* sebesar 50399,639 bps, yaitu sebesar 14,84%.

Berdasarkan hasil pengujian *throughput* pada Tabel 6.1 tersebut, dapat diketahui bahwa nilai *throughput* akan cenderung menurun dan berbanding terbalik dengan bertambahnya UE dan dapat diketahui pula bahwa algoritme penjadwalan paket *Proportional Fair* lebih unggul dari pada algoritme penjadwalan paket *Round Robin*. Algoritme *Proportional Fair* cenderung memiliki *throughput* lebih tinggi dan rata-rata persentase penurunan lebih rendah dalam penanganan bertambahnya UE pada masing-masing eNB dibandingkan algoritme penjadwalan paket *Round Robin*. Untuk grafik pengujian *throughput* yang terbentuk dapat dilihat pada Gambar 6.1.

Terjadinya *handover* pada pengujian *throughput* juga mempengaruhi kinerja namun tidak terlalu signifikan dari algoritme penjadwalan paket *Round Robin* maupun *Proportional Fair* dengan nilai *throughput* relatif lebih rendah dari penggunaan algoritme penjadwalan paket dengan tanpa terjadi *handover*.



Gambar 6.1 Grafik hasil pengujian terhadap *throughput*

6.2.2 Hasil Pengujian Terhadap *Packet Loss Ratio*

Hasil pengujian terhadap persentase paket yang hilang saat pengiriman atau *packet loss ratio* dapat dilihat pada Tabel 6.2. Pada tabel hasil pengujian *packet loss ratio* tersebut dapat dilihat bahwa persentase paket yang gagal dikirim cenderung meningkat berbanding lurus dengan bertambahnya jumlah UE pada setiap eNB.

Tabel 6.2 Hasil *Packet loss ratio* (%)

UE/eNB	Penjadwalan paket			
	RR (%)	PF (%)	RRHO (%)	PFHO (%)
1	0,017	0,017	0,017	0,017
2	5,679	5,679	5,679	5,679
3	11,187	7,874	13,162	9,353
4	11,002	7,637	13,516	11,543
5	19,648	17,356	22,592	19,738
6	26,666	22,024	29,749	25,087
7	28,261	20,522	30,328	22,886
8	29,202	22,467	31,695	25,501
9	33,906	29,121	36,241	31,664
10	34,613	29,884	36,377	31,537
11	36,575	31,614	38,976	33,614
12	36,461	31,002	38,454	32,646
13	41,206	35,182	43,228	37,270
14	33,349	27,509	36,857	30,635
15	47,831	40,306	47,731	40,296
16	46,920	39,055	46,660	39,086
17	48,434	38,396	48,061	39,097
18	49,609	40,383	49,853	41,243
19	52,729	44,001	52,748	44,197

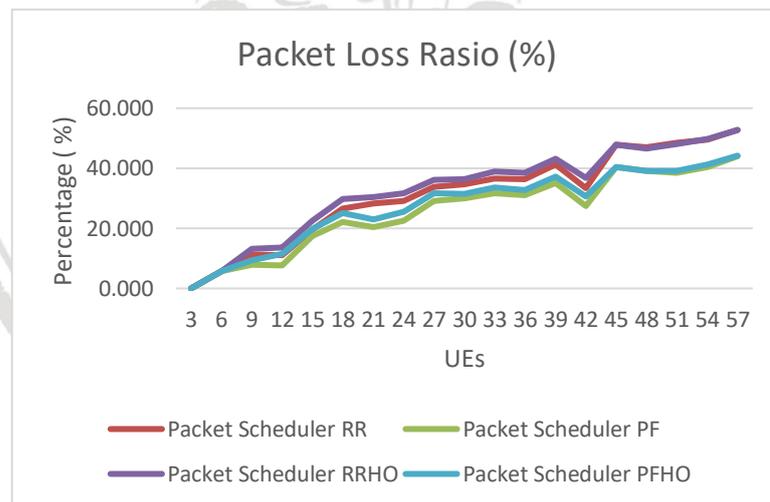
Pada algoritme penjadwalan paket *Round Robin* tanpa terjadi *handover* terlihat bahwa *packet loss ratio* cenderung meningkat pada setiap penambahan UE, peningkatan persentase terbesar terjadi pada UE per eNB berjumlah 15 buah dengan hasil persentase kegagalan pengiriman sebesar 47,831%, dari persentase yang dihasilkan sebelumnya pada saat UE per eNB berjumlah 14 buah dengan hasil sebelumnya sebesar 33,349%, yaitu memiliki selisih sebesar 14,48%. Begitu juga pada algoritme penjadwalan paket *Round Robin* pada skenario terjadinya *handover*. Persentase kegagalan pengiriman paket cenderung meningkat pula pada setiap penambahan UE per eNB. Peningkatan terbesar terjadi pada saat UE per eNB berjumlah 14 dengan persentase sebesar 36,857% ke jumlah UE per eNB sebanyak 15 buah dengan persentase sebesar 47,731%, yaitu memiliki selisih 10,87%.

Pada algoritme penjadwalan paket *Proportional Fair* tanpa terjadi *handover* terlihat bahwa *packet loss ratio* juga cenderung meningkat pada setiap penambahan UE, peningkatan persentase terbesar terjadi pada UE per eNB

berjumlah 15 buah dengan hasil persentase kegagalan pengiriman sebesar 40,306%, dari persentase yang dihasilkan sebelumnya pada saat UE per eNB berjumlah 14 buah dengan hasil sebelumnya sebesar 40,306%, yaitu memiliki selisih sebesar 12,80%. Begitu juga pada algoritme penjadwalan paket *Proportional Fair* pada skenario terjadinya *handover*. Persentase kegagalan pengiriman paket cenderung meningkat pula pada setiap penambahan UE per eNB. Peningkatan terbesar terjadi pada saat UE per eNB berjumlah 14 dengan persentase sebesar 30,635% ke jumlah UE per eNB sebanyak 15 buah dengan persentase sebesar 40,296%, yaitu memiliki selisih 9,66%.

Berdasarkan tabel 6.2 hasil pengujian terhadap *packet loss ratio* tersebut dapat dilihat bahwa persentase paket yang gagal terkirim meningkat di setiap penambahan UE pada setiap eNB dan peningkatan tersebut terjadi pada semua algoritme penjadwalan yang digunakan. Namun algoritme penjadwalan paket *Round Robin* memiliki persentase yang lebih besar daripada algoritme penjadwalan paket *Proportional Fair*. Dapat disimpulkan bahwa algoritme *Proportional Fair* dapat menangani *packet loss* lebih baik dibandingkan penggunaan algoritme penjadwalan paket *Round Robin*. Untuk grafik pengujian terhadap *packet loss ratio* yang terbentuk dapat dilihat pada Gambar 6.2.

Terjadinya *handover* mempengaruhi tingginya persentase dari packet loss pada kinerja kedua algoritme penjadwalan paket yang digunakan. Pada UE berjumlah kurang dari 45 buah, *packet loss ratio* dengan adanya *handover* terlihat lebih tinggi dibandingkan tanpa adanya *handover*. Namun pada saat UE berjumlah 45 buah atau lebih, *handover* tidak terlalu berpengaruh terhadap kinerja kedua algoritma penjadwalan paket yang digunakan.



Gambar 6.2 Grafik Hasil Pengujian *Packet Loss Ratio*

6.2.3 Hasil Pengujian Terhadap *End-to-End Delay*

Hasil pengujian terhadap *end-to-end delay* dapat dilihat pada Tabel 6.3. Pada tabel hasil pengujian *delay* tersebut dapat dilihat bahwa waktu yang *delay* cenderung meningkat dan berbanding lurus dengan bertambahnya jumlah UE pada setiap eNB.

Tabel 6.3 Hasil *end-to-end delay* (s) untuk setiap node

UE/eNB	Penjadwalan paket			
	RR (s)	PF (s)	RRHO (s)	PFHO (s)
1	0,023	0,023	0,023	0,023
2	0,024	0,024	0,024	0,024
3	0,031	0,029	0,030	0,027
4	0,033	0,030	0,029	0,028
5	0,032	0,031	0,032	0,031
6	0,040	0,033	0,036	0,032
7	0,051	0,040	0,044	0,035
8	0,045	0,043	0,047	0,038
9	0,055	0,050	0,051	0,047
10	0,055	0,050	0,054	0,049
11	0,059	0,055	0,056	0,053
12	0,058	0,054	0,056	0,052
13	0,059	0,056	0,058	0,055
14	0,057	0,054	0,056	0,054
15	0,062	0,058	0,059	0,058
16	0,058	0,059	0,059	0,060
17	0,066	0,065	0,065	0,064
18	0,068	0,068	0,067	0,066
19	0,066	0,064	0,064	0,065

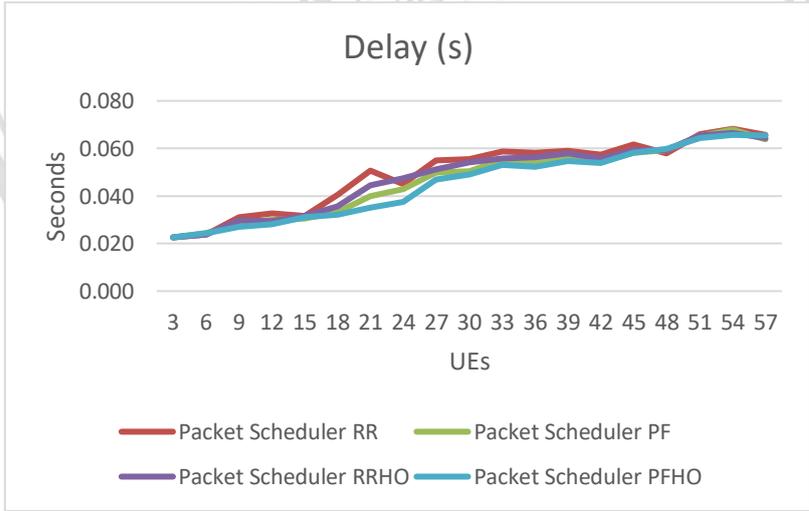
Pada algoritme penjadwalan paket *Round Robin* tanpa terjadi *handover* terlihat bahwa *delay* cenderung meningkat pada setiap penambahan UE, peningkatan waktu *Delay* terbesar terjadi pada UE per eNB berjumlah 7 buah dengan waktu *delay* sebesar 0,051 detik, dari persentase yang dihasilkan sebelumnya pada saat UE per eNB berjumlah 6 buah dengan waktu *delay* sebelumnya sebesar 0,040, yaitu memiliki selisih sebesar 0,011 detik. Begitu juga pada algoritme penjadwalan paket *Round Robin* pada skenario terjadinya *handover*. Waktu *delay* cenderung meningkat pula pada setiap penambahan UE per eNB. Peningkatan terbesar terjadi pada saat UE per eNB berjumlah 6 dengan waktu selama 0,036 detik ke jumlah UE per eNB sebanyak 7 buah dengan waktu selama 0,044 detik, yaitu memiliki selisih 0,08 detik.

Pada algoritme penjadwalan paket *Proportional Fair* tanpa terjadi *handover* terlihat bahwa *delay* cenderung meningkat pada setiap penambahan UE, peningkatan waktu *Delay* terbesar terjadi pada UE per eNB berjumlah 7 buah dengan waktu *delay* sebesar 0,040 detik, dari persentase yang dihasilkan

sebelumnya pada saat UE per eNB berjumlah 6 buah dengan waktu *delay* sebelumnya sebesar 0,033, dan pada UE per eNB berjumlah 9 buah dengan waktu *delay* sebesar 0,050 detik dari persentase yang dihasilkan sebelumnya pada saat UE per eNB berjumlah 8 buah dengan waktu *delay* sebelumnya sebesar 0,043 , yaitu memiliki selisih sebesar 0,007 detik. Begitu juga pada algoritme penjadwalan paket *Proportional Fair* pada skenario terjadinya *handover*. Waktu *delay* cenderung meningkat pula pada setiap penambahan UE per eNB. Peningkatan terbesar terjadi pada saat UE per eNB berjumlah 8 dengan waktu selama 0,038 detik ke jumlah UE per eNB sebanyak 9 buah dengan waktu selama 0,047 detik, yaitu memiliki selisih 0,009 detik.

Berdasarkan tabel 6.3 hasil pengujian terhadap *end-to-end delay* tersebut dapat dilihat bahwa waktu *delay* meningkat di setiap penambahan UE pada setiap eNB dan peningkatan tersebut terjadi pada semua algoritme penjadwalan yang digunakan. Untuk grafik pengujian terhadap *end-to-end delay* yang terbentuk dapat dilihat pada Gambar 6.3.

Terjadinya *handover* tidak terlalu mempengaruhi lama waktu yang dibutuhkan untuk *delay* pada kinerja kedua algoritme penjadwalan paket yang digunakan. Namun rata-rata *delay* yang terbentuk pada hasil simulasi dengan skenario *handover* cenderung lebih rendah 0,002 detik dibandingkan rata-rata *delay* yang dibutuhkan pada skenario tidak terjadinya *handover*. Pada algoritme penjadwalan paket *Round Robin* rata-rata *delay* yang terbentuk dari ke 19 percobaan mendapatkan waktu rata-rata selama 0,050 detik sedangkan pada skenario ini *handover*, algoritme membutuhkan waktu rata-rata selama 0.048 detik untuk *delay*. Kemudian untuk algoritme penjadwalan paket *Proportional Fair* rata-rata *delay* yang terbentuk dari 19 percobaan mendapatkan waktu rata-rata 0,047 detik sedangkan pada skenario *handover*, algoritme ini membutuhkan waktu *Delay* rata-rata selama 0,045 detik.



Gambar 6.3 Grafik Hasil Pengujian *End-to-End Delay*



6.2.4 Hasil Pengujian Terhadap *Jitter*

Hasil pengujian terhadap *jitter* dapat dilihat pada Tabel 6.4. Pada tabel hasil pengujian *jitter* tersebut dapat dilihat bahwa variasi waktu atau *jitter* yang terbentuk cenderung meningkat dan berbanding lurus dengan bertambahnya jumlah UE pada setiap eNB.

Tabel 6.4 Hasil pengujian terhadap *jitter* (s)

UE/eNB	Penjadwalan paket			
	RR (s)	PF (s)	RRHO (s)	PFHO (s)
1	0,000	0,000	0,000	0,000
2	0,000	0,000	0,000	0,000
3	0,002	0,001	0,001	0,001
4	0,004	0,002	0,002	0,001
5	0,004	0,002	0,004	0,003
6	0,007	0,005	0,006	0,003
7	0,009	0,006	0,008	0,005
8	0,010	0,007	0,009	0,006
9	0,012	0,010	0,013	0,011
10	0,015	0,013	0,016	0,013
11	0,018	0,016	0,017	0,017
12	0,015	0,013	0,015	0,014
13	0,020	0,019	0,019	0,018
14	0,019	0,017	0,018	0,018
15	0,021	0,021	0,022	0,022
16	0,018	0,021	0,019	0,021
17	0,021	0,024	0,021	0,024
18	0,020	0,025	0,021	0,024
19	0,019	0,023	0,019	0,025

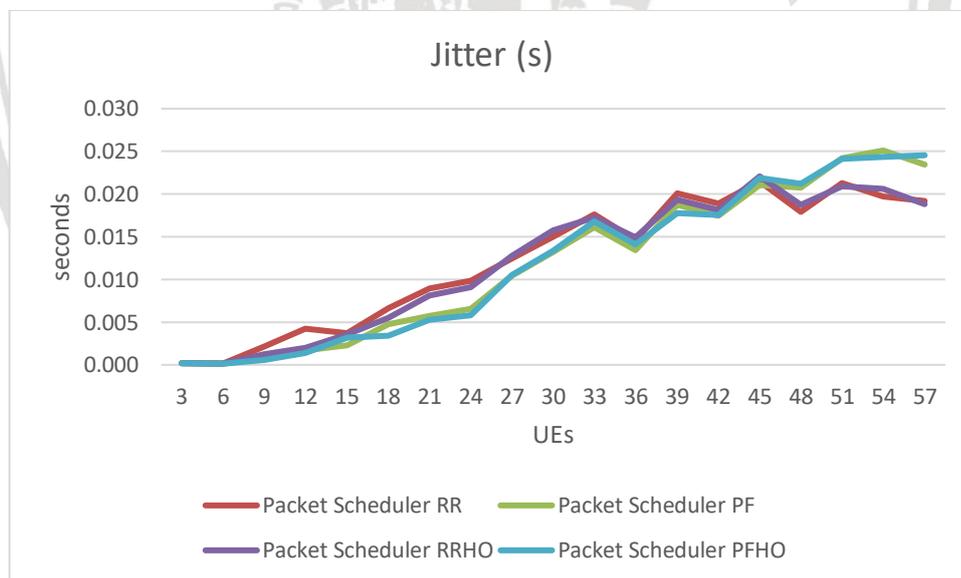
Pada algoritme penjadwalan paket *Round Robin* tanpa terjadi *handover* terlihat bahwa *jitter* cenderung meningkat pada setiap penambahan UE, peningkatan waktu *jitter* terbesar terjadi pada UE per eNB berjumlah 13 buah dengan waktu *jitter* sebesar 0,020 detik, dari waktu yang dihasilkan sebelumnya pada saat UE per eNB berjumlah 12 buah dengan waktu *jitter* sebelumnya sebesar 0,015, yaitu memiliki selisih sebesar 0,005 detik. Begitu juga pada algoritme penjadwalan paket *Round Robin* pada skenario terjadinya *handover*. Waktu *jitter* cenderung meningkat pula pada setiap penambahan UE per eNB. Peningkatan terbesar terjadi memiliki selisih 0,004 detik, yaitu pada penambahan jumlah UE dari 8 UE per eNB ke 9 UE per eNB, pada penambahan jumlah UE dari 12 UE per eNB ke 13 UE per eNB, dan pada penambahan jumlah UE dari 14 UE per eNB ke 15 UE per eNB.

Pada algoritme penjadwalan paket *Proportional Fair* tanpa terjadi *handover* terlihat bahwa *jitter* cenderung meningkat pada setiap penambahan UE, peningkatan waktu *Jitter* terbesar terjadi pada UE per eNB berjumlah 13 buah dengan waktu *jitter* sebesar 0,019 detik, dari waktu yang dihasilkan sebelumnya

pada saat UE per eNB berjumlah 12 buah dengan waktu *jitter* sebelumnya sebesar 0,013, yaitu memiliki selisih sebesar 0,006 detik. Begitu juga pada algoritme penjadwalan paket *Proportional Fair* pada skenario terjadinya *handover*. Waktu *jitter* cenderung meningkat pula pada setiap penambahan UE per eNB. Peningkatan terbesar terjadi memiliki selisih 0,005 detik, yaitu pada penambahan jumlah UE dari 8 UE per eNB dengan *jitter* sebesar 0,006 detik ke 9 UE per eNB dengan *jitter* sebesar 0,011 detik.

Berdasarkan tabel 6.4 hasil pengujian terhadap *jitter* tersebut dapat dilihat bahwa waktu *jitter* meningkat di setiap penambahan UE pada setiap eNB dan peningkatan tersebut terjadi pada semua algoritme penjadwalan yang digunakan. Untuk grafik pengujian terhadap *jitter* yang terbentuk dapat dilihat pada Gambar 6.4. Pada grafik tersebut terlihat ada *crossing point* yang terbentuk pada UE per eNB berjumlah 15. Dapat disimpulkan bahwa algoritme penjadwalan paket *Proportional Fair* lebih unggul pada penanganan pada keadaan UE per eNB kurang dari 15, sedangkan ketika UE per eNB lebih dari 15 algoritme penjadwalan paket *Round Robin* lebih unggul dan memiliki waktu yang lebih rendah untuk pengujian *jitter* dibandingkan dengan algoritme penjadwalan paket *Proportional Fair*.

Terjadinya *handover* tidak terlalu mempengaruhi lama waktu yang terbentuk untuk *jitter* pada kinerja kedua algoritme penjadwalan paket yang digunakan. Waktu yang terbentuk pada skenario *handover* cenderung berhimpitan dengan waktu yang terbentuk pada kenario tanpa adanya *handover*.



Gambar 6.4 Grafik Hasil Pengujian *Jitter*

6.3 Analisis Terhadap Hasil Pengujian

Hasil pengujian terhadap *throughput* dari kedua algoritme penjadwalan paket dipengaruhi oleh jumlah UE yang digunakan, semakin banyak UE maka *throughput* yang didapat akan semakin menurun. Hal ini dikarenakan dalam menjadwalkan pengiriman paket tidak mampu menampung semua paket yang akan dikirimkan sehingga terdapat paket yang hilang atau terbuang. Dari hasil tersebut dapat ditunjukkan pula hasil dari *packet loss ratio* yang didapat. Persentase paket yang hilang atau terbuang meningkat dengan semakin bertambahnya pula UE yang digunakan. Dari hasil tersebut didapatkan bahwa semakin banyak paket yang hilang atau dibuang maka nilai dari *throughput* yang akan didapat akan semakin kecil.

Untuk perbandingan algoritme penjadwalan paket dapat disimpulkan bahwa untuk penanganan UE yang banyak algoritme penjadwalan paket *Proportional Fair* lebih baik dibandingkan algoritme *Round Robin* pada sisi *throughput* dan *packet loss ratio*. Hal ini dikarenakan pada algoritme *Proportional Fair* terdapat pertimbangan untuk memaksimalkan total *throughput* yang digunakan sehingga keberhasilan pengalokasian sumber daya *bandwidth* lebih maksimal. Sedangkan algoritme *Round Robin* mendapatkan nilai *throughput* lebih kecil dan *packet loss ratio* lebih besar karena pada penjadwalannya algoritma ini hanya memberikan *time quantum* atau waktu yang digunakan untuk memproses paket yang membatasi waktu pengiriman sehingga jika paket yang dikirimkan belum terkirim seluruhnya sedangkan waktu sudah habis maka paket tersebut akan dibuang.

Dalam penanganan *delay*, kedua algoritme penjadwalan paket memiliki hasil yang tidak jauh berbeda dan algoritme penjadwalan paket *proportional* lebih unggul dengan hanya memiliki selisih rata-rata 0,003 detik dibandingkan hasil rata-rata yang dihasilkan oleh algoritme penjadwalan paket *Round Robin*. *Delay* pada algoritme *Proportional Fair* dapat mendapatkan nilai yang lebih baik karena pada penjadwalannya paket-paket diurutkan berdasarkan pertimbangan nilai CQI, sehingga paket-paket dikirimkan melalui kualitas saluran yang baik dan mempercepat pengiriman paket untuk sampai ke tujuan.

Sedangkan pada hasil pengujian *jitter* didapatkan bahwa algoritme penjadwalan paket *Round Robin* lebih unggul ketika UE yang digunakan banyak. Hal ini dikarenakan dalam penjadwalan paket, algoritma *Round Robin* menganggap semua paket itu sama sehingga semua paket diberikan *time quantum* atau waktu pemrosesan yang sama, dengan demikian variasi waktu *delay* yang diukur sebagai *Jitter* akan cenderung stabil atau tidak berubah terlalu signifikan pada jumlah UE yang banyak.

Handover yang terjadi ketika simulasi berjalan tidak berpengaruh besar dalam kinerja algoritme penjadwalan paket *Round Robin* maupun *Proportional Fair*. Pada hasil pengujian *throughput handover* hanya mengurangi kinerja dari algoritma penjadwalan paket *Round Robin* dengan rata-rata 0,15% dari skenario pengujian tanpa adanya *handover* sedangkan pada algoritme penjadwalan paket *Proportional Fair* selisih dari penurunan *throughput* hanya sebesar 0,05% dari

skenario pengujian tanpa adanya *handover*. Pada hasil *packet loss ratio handover* hanya menambah persentase kehilangan paket sekitar 1,5% pada algoritma penjadwalan paket *Round Robin* dan sekitar 1,6% pada algoritma penjadwalan paket *Proportional Fair* dari hasil pengujian tanpa adanya *handover*.



BAB 7 PENUTUP

7.1 Kesimpulan

Kesimpulan yang dapat diambil berdasarkan hasil penelitian yang telah dilakukan adalah sebagai berikut:

1. Pengimplementasian algoritme penjadwalan paket dapat dilakukan dengan memasukkan algoritme yang akan digunakan pada EPC. Perbedaan antara algoritme penjadwalan paket *Round Robin* dan *Proportional Fair* terletak pada pertimbangan pada penentuan peringkat paket untuk diproses. Algoritme *Round Robin* memberikan *time quantum* yang sama pada setiap paket untuk diproses secara bergantian sedangkan algoritme *Proportional Fair* mempertimbangkan paket yang mempunyai *throughput* yang lebih kecil untuk diproses.
2. Pada pengujian *throughput*, *packet loss ratio*, dan *end-to-end delay* algoritme penjadwalan paket *Proportional Fair* mendapatkan nilai *throughput* lebih tinggi, persentase *packet loss ratio* lebih rendah, dan mendapat nilai rata-rata delay 0.003 lebih cepat dibandingkan algoritme *Round Robin* sehingga algoritme *Proportional Fair* lebih baik dari algoritme *Round Robin* dalam penanganan *throughput*, *packet loss ratio*, dan *end-to-end delay*. Sedangkan pada pengujian *Jitter*, algoritme *proportional fair* memiliki nilai yang lebih baik dibandingkan algoritme *Round Robin* pada saat UE berjumlah kurang dari 15 UE per eNB sedangkan algoritme *Round Robin* memiliki nilai yang lebih baik ketika UE berjumlah lebih dari 15 UE per eNB.
3. *Handover* memiliki pengaruh yang kecil terhadap kinerja dari algoritme penjadwalan paket *Round Robin* dan *Proportional Fair*.

7.2 Saran

Berikut adalah saran untuk pengembangan penelitian selanjutnya:

1. *SRS periodicity* yang digunakan pada penelitian ini hanya sebesar 40 ms saja. Diharapkan untuk penelitian selanjutnya nilai ini dapat diperbesar sehingga dapat menambah jumlah UE yang digunakan.
2. Peletakan eNB lebih memperhatikan atau diterapkan dengan mempertimbangkan keadaan sumber daya maupun keadaan geografis pada suatu daerah tertentu sehingga simulasi dapat dilakukan berdasarkan kondisi yang nyata.
3. Mempertimbangkan variasi pergerakan UE kecepatan pada manusia saat berkomunikasi menggunakan jaringan LTE, seperti memperlakukan objek ketika UE berkomunikasi didalam sebuah kendaraan ataupun UE berkomunikasi saat berjalan kaki.

DAFTAR PUSTAKA

- Afroz, F. et al., 2015. SINR, RSRP, RSSI and RSRQ Measurements in Long Term Evolution Networks. *International Journal of Wireless & Mobile Networks (IJWMN)*, pp. 113-123.
- Capozzi, F. et al., 2013. Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, pp. 1-23.
- Carneiro, G., Fortuna, P. & Ricardo, M., 2009. FlowMonitor - a network monitoring framework for the Network Simulator 3 (NS-3).
- Hamid, N. I. B., Kawser, M. T. & Hoque, M. A., 2012. Coverage and Capacity Analysis of LTE Radio Network Planning considering Dhaka City. *International Journal of Computer Applications*, Volume 46, pp. 49-56.
- Kim, J., Lee, G. & In, H., 2014. Adaptive Time-to-Trigger Scheme for Optimizing LTE Handover. *International Journal of Control and Automation*.
- Kurose, J. F. & Ross, K. W., 2012. *Computer Networking: A Top-Down Approach (6th Edition)*. s.l.:Pearson.
- Lin, C. C., Sandrasegaran, K., Ramli, H. A. M. & Basukala, R., 2011. Optimized Performance Evaluation of LTE Hard Handover Algorithm with Average RSRP Constraint. *International Journal of Wireless & Mobile Networks (IJWMN)*, pp. 1-16.
- Mahfoudi, M., Bekkali, M. E., Najid, A. & Mazer, S., 2014. High Mobility Evaluation for Voice & Video over LTE.
- Panjaitan, A., 2015. Analisa Performansi Packet Scheduling Downlink Max-Throughput & Proportional Fair pada Layanan VoLTE. *Jurnal Penelitian Informatika*.
- Sesia, S., Toufik, I. & Baker, M., 2011. *LTE—the UMTS Long Term Evolution : From Theory to Practice. 2nd ed.* s.l.:Wiley.
- Suryaman, B., Irawati, I. D. & Mulyana, A., 2010. *Perbandingan Performansi Algoritma Penjadwalan Round Robin, Maximum C/I, dan Proportional Fair dengan Menggunakan HARQ pada Sistem 3GPP LTE*. Bandung, Institut Teknologi Telkom Bandung, pp. 7-14.
- Swetha, Mohankumar, M. N. & Devaraju, J. T., 2014. *Performance Study of Round Robin and Proportional Fair Scheduling Algorithms by Emulation for Video Traffic in LTE Networks*. Bengaluru, ResearchGate, pp. 6-9.