

**SIMULASI MOBILE AD-HOC NETWORK (MANET) UNTUK
MENGANALISIS PERBANDINGAN PERFORMANSI
ROUTING PROTOKOL OLSR DAN DSDV**

TUGAS AKHIR

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



الجامعة الإسلامية
الاندونيسية

Oleh :

Nama : Egi Marudi Samudarji

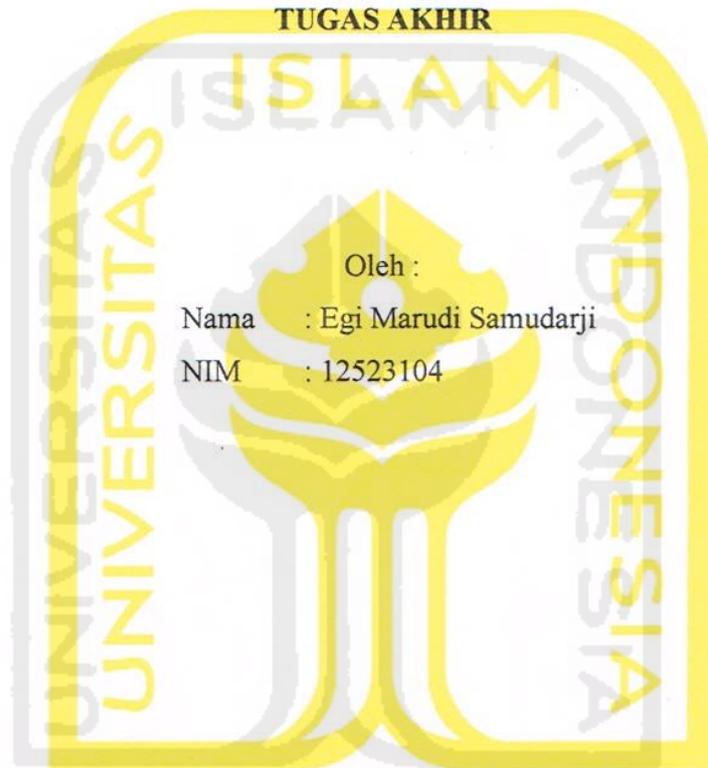
NIM : 12523104

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2017

LEMBAR PENGESAHAN DOSEN PEMBIMBING
SIMULASI MOBILE AD-HOC NETWORK (MANET) UNTUK
MENGANALISIS PERBANDINGAN PERFORMANSI
ROUTING PROTOKOL OLSR DAN DSDV

TUGAS AKHIR



Oleh :

Nama : Egi Marudi Samudarji

NIM : 12523104

Yogyakarta, 15 Maret 2017

Pembimbing,

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

LEMBAR PENGESAHAN DOSEN PENGUJI
SIMULASI MOBILE AD-HOC NETWORK (MANET) UNTUK
MENGANALISIS PERBANDINGAN PERFORMANSI
ROUTING PROTOKOL OLSR DAN DSDV

TUGAS AKHIR

Oleh :

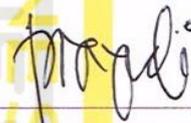
Nama : Egi Marudi Samudarji
NIM : 12523104

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat untuk
Memperoleh Gelar Sarjana Jurusan Teknik Informatika Fakultas Teknologi
Industri Universitas Islam Indonesia
Yogyakarta, April 2017

Tim Penguji

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

Ketua



Fietyata Yudha, S.Kom., M.Kom.

Anggota 1



Elyza Gustri Wahyuni, S.T., M.Cs.

Anggota 2



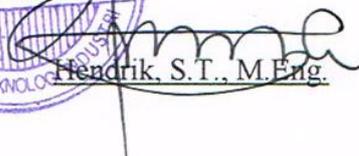
Mengetahui,

Ketua Jurusan Teknik Informatika

Fakultas Teknologi Industri
Universitas Islam Indonesia



Hendrik, S.T., M.Eng.



LEMBAR PERNYATAAN KEASLIAN

TUGAS AKHIR

Saya yang bertanda tangan di bawah ini,

Nama : Egi Marudi Samudarji

NIM : 12523104

Menyatakan bahwa seluruh tulisan dan komponen yang ada di dalam laporan tugas akhir ini diambil dengan menyalin atau meniru pemikiran orang lain. Tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari nanti terbukti bahwa ada beberapa tulisan dan komponen dari tugas akhir ini bukan hasil karya sendiri, maka saya siap menanggung segala resiko dan konsekuensinya.

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

Yogyakarta, 15 Maret 2017



Egi Marudi Samudarji

HALAMAN PERSEMBAHAN

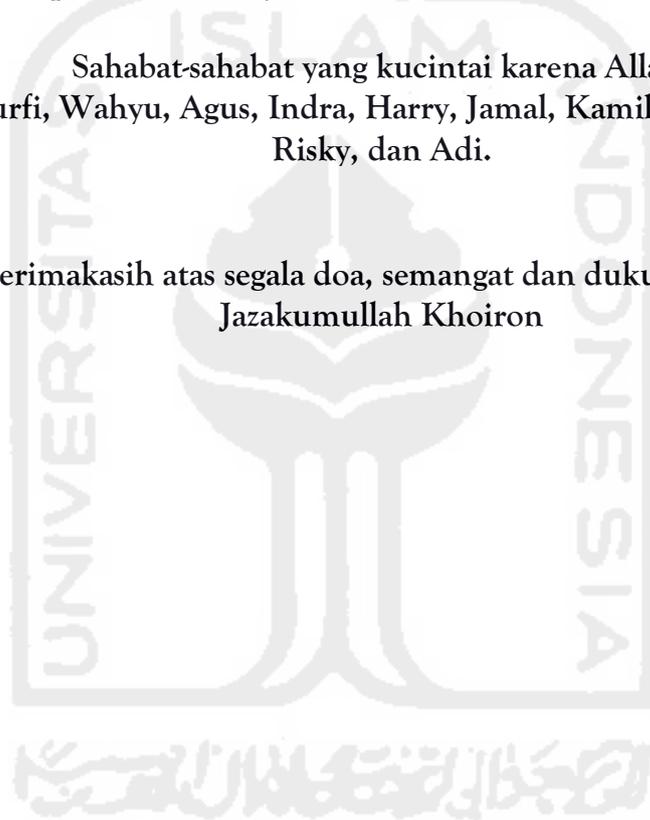
Karya tulis ini saya persembahkan untuk,

Ibu dan Ayah tercinta,
Siti Agustina dan Setyaman, S.Hut.

Kedua kakak dan keponakan tercinta,
Vage Septian, Ria Resty, S.E., dan Muhammad Hanafi Rasyid.

Sahabat-sahabat yang kucintai karena Allah,
Abas, Nurfi, Wahyu, Agus, Indra, Harry, Jamal, Kamil, Dhandi, Ferdy,
Risky, dan Adi.

Terimakasih atas segala doa, semangat dan dukungannya.
Jazakumullah Khoiron



HALAMAN MOTTO

“..Barangsiapa bertakwa kepada Allah niscaya Dia akan membukakan baginya jalan keluar. Dan memberinya rezeki dari arah yang tidak ia duga-duga. Dan barangsiapa yang bertawakkal kepada Allah niscaya Allah akan mencukupkan (keperluan)nya. ..” (QS. At Talaq : 2-3).

“..Sesungguhnya orang yang paling mulia di antara kamu di sisi Allah ialah orang yang paling takwa di antara kamu..” (QS. Al Hujurat: 13)

“Sesungguhnya Allah tidak melihat pada rupa dan harta kalian. Namun yang Allah lihat adalah hati dan amalan kalian.” (HR. Muslim no. 2564)

Seindah-indahnya dunia, tidak ada bandingannya dengan indahnya surga. Gapai surga dengan ridhonya Allah Ta’ala dan gapai ridhonya Allah dengan patuh kepada-Nya dan mengikuti sunnah Rasulullah Shallallahu ‘alaihi wasallam.

KATA PENGANTAR

Assalamu'alaikum. Wr. Wb

Segep rasa syukur saya panjatkan kehadiran Allah Ta'ala, yang telah melimpahkan rahmat dan taufik serta hidayat-Nya, sehingga pada akhirnya saya dapat menyelesaikan Tugas Akhir dan penyusunan laporan ini berjalan dengan baik.

Tugas Akhir ini dibuat sebagai salah satu syarat untuk memperoleh gelar sarjana Teknik Informatika Universitas Islam Indonesia dan juga sebagai salah satu sarana untuk mempraktekkan secara langsung ilmu yang telah diperoleh selama kuliah.

Penulis menyadari bahwa dalam penelitian dan penyusunan laporan Tugas Akhir ini tidak bisa lepas dari kuasa Allah Ta'ala, bimbingan, dorongan, dan bantuan baik material maupun spiritual dari berbagai pihak, oleh karena itu perkenankanlah saya mengucapkan terima kasih kepada:

1. Allah Ta'ala. Segala puja dan puji syukur kami haturkan atas karunia dan kasih sayang-Nya sehingga penulis masih diberi kekuatan, kemampuan dan kesempatan untuk menyelesaikan Tugas Akhir ini dengan baik.
2. Nabi Muhammad Shallallahu 'Alaihi Wasallam sebagai suri teladan dan panutan untuk seluruh umat manusia.
3. Ibu dan Ayah tercinta, yang selalu memberikan doa, dukungan semangat dan dukung materi dari keringat mereka, yang selalu berusaha untuk memenuhi kebutuhan sehari-hari.
4. Kedua kakak tercinta Vage Septian dan Ria Resty Febrina, S.E., yang juga selalu mendoakan, memberi semangat dan dukungan materi.
5. Bapak Nandang Sutrisno, SH., M.Hum., LL.M., Ph.D., selaku Rektor Universitas Islam Indonesia.
6. Bapak Dr., Drs. Imam Djati Widodo, M.Eng.Sc., selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
7. Bapak Hendrik, S.T., M.Eng., selaku Ketua Jurusan Teknik Informatika

Fakultas Teknologi Industri Universitas Islam Indonesia.

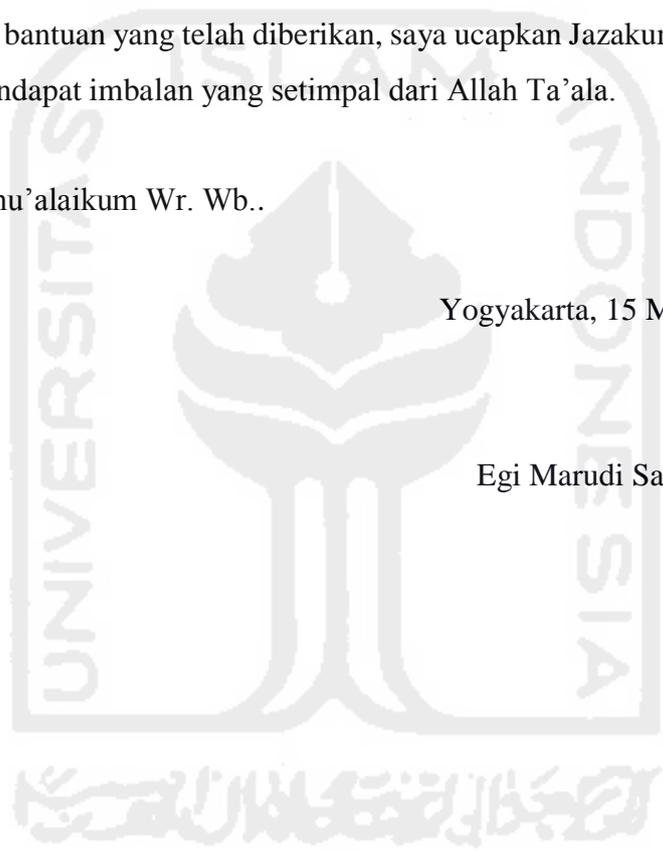
8. Bapak Yudi Prayudi, S.Si., M.Kom., selaku pembimbing Tugas Akhir yang senantiasa memberikan bimbingan demi suksesnya Tugas Akhir ini.
9. Teman-teman dan sahabat yang kucintai karena Allah yang telah memberikan doa, bantuan dan dukungannya.
10. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu, mempermudah dan memperlancar Tugas Akhir ini.

Atas segala bantuan yang telah diberikan, saya ucapkan Jazakumullah Khoiron dan semoga mendapat imbalan yang setimpal dari Allah Ta'ala.

Wassalamu'alaikum Wr. Wb..

Yogyakarta, 15 Maret 2017

Egi Marudi Samudarji



SARI

Mobile Ad Hoc Network (MANET) adalah salah satu jenis teknologi *wireless* yang terdiri dari kumpulan *node-node* (*device*) bergerak dan bersifat dinamis yang bekerja tanpa menggunakan infrastruktur atau administrasi terpusat. MANET membutuhkan routing protokol sebagai sistem pencarian rute paket data di dalam jaringan. Pada penelitian ini diteliti lebih lanjut perbandingan performa dari routing protokol OLSR dan DSDV yang merupakan dari jenis yang sama, yaitu proaktif routing protokol. Perbandingan tersebut nantinya dapat dijadikan referensi untuk memilih routing protokol yang akan digunakan pada suatu kondisi jaringan tertentu. Sebelum pembuatan simulasi MANET dibuat terlebih dahulu rancangan skenario dan simulasi, kemudian dibuat kesimpulan dari hasil simulasi tersebut.

Pada penelitian ini telah dilakukan simulasi MANET untuk menganalisis perbandingan performansi dari routing protokol OLSR dan DSDV menggunakan aplikasi NS3 (*Network Simulator 3*). Pada skenario penambahan *node* didapatkan nilai rata-rata parameter untuk OLSR adalah *throughput* 6,938775510 Kbps, *packet delivery ratio* 96,51%, *packet loss* 3,48%, *delay* 6,8246062 m/s dan untuk DSDV adalah *throughput* 6,095238095 Kbps, *packet delivery ratio* 90,59%, *packet loss* 9,40%, dan *delay* 3,8272005 m/s. Sedangkan pada skenario penambahan ukuran paket data didapatkan nilai rata-rata parameter untuk OLSR adalah *throughput* 6,459183673 Kbps, *packet delivery ratio* 86,99%, *packet loss* 13,07%, *delay* 28,907607 m/s dan untuk DSDV adalah *throughput* 5,132653061 Kbps, *packet delivery ratio* 79,50%, *packet loss* 20,49%, dan *delay* 3,8272005 m/s.

Berdasarkan hasil yang didapat, diketahui OLSR lebih baik dari DSDV dilihat dari skenario penambahan *node* dan penambahan ukuran paket data berdasarkan nilai parameter *throughput*, *packet delivery ratio* dan *packet loss*. Sedangkan DSDV lebih baik dari OLSR pada nilai parameter *delay*. Diketahui juga bahwa DSDV cocok digunakan pada kepadatan jaringan yang kecil. Sedangkan OLSR cocok digunakan pada kepadatan jaringan menengah ke atas.

Kata kunci: MANET, NS3, routing protokol, OLSR, DSDV

TAKARIR

<i>Device</i>	Perangkat
<i>Node</i>	Simpul
<i>Link</i>	Sambungan
<i>Wireless</i>	Jaringan tanpa kabel
<i>Mobile</i>	Dapat bergerak atau dapat digerakkan dengan bebas dan mudah
<i>User</i>	Pengguna
<i>Hop</i>	Lompatan
<i>Traffic</i>	Jumlah banyaknya paket data pada jalur jaringan
<i>Open Source</i>	Program komputer yang lisensinya memberi kebebasan kepada pengguna dalam menjalankan program tersebut
<i>Module</i>	Unit piranti lunak berdiri sendiri yang terdiri dari model
<i>Trace</i>	Jejak rute yang dilewati
<i>File</i>	Berkas
<i>Channel</i>	Penghubung antar node
<i>Cluster</i>	Pengelompokan unit
<i>Broadcast</i>	Sebuah metode pengiriman data, dimana data dikirim ke banyak titik sekaligus, tanpa melakukan pengecekan apakah titik tersebut siap atau tidak, atau tanpa memperhatikan apakah data itu sampai atau tidak.
<i>Real-time</i>	Waktu sebenarnya
<i>Bandwidth</i>	Nilai hitung konsumsi transfer data telekomunikasi yang dihitung dalam satuan bit per detik.

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN DOSEN PEMBIMBING.....	ii
LEMBAR PENGESAHAN DOSEN PENGUJI	iii
LEMBAR PERNYATAAN KEASLIAN	iv
HALAMAN PERSEMBAHAN.....	v
HALAMAN MOTTO	vi
KATA PENGANTAR.....	vii
SARI	ix
TAKARIR.....	x
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xiv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
1.6. Metodologi Penelitian.....	4
1.7. Sistematika Penulisan.....	6
BAB II LANDASAN TEORI	8
2.1. Studi Literatur.....	8
2.1.1. Perbandingan Penelitian	8
2.2. Jaringan Nirkabel (<i>Wireless</i>)	9
2.3. <i>Mobile Ad Hoc Network</i> (MANET)	10
2.4. Routing Protokol.....	11
2.4.1. <i>Table Driven Routing Protocol</i> (Proaktif).....	11
2.5. Bahasa Pemrograman C++	12
2.6. <i>Network Simulator 3</i> (NS3).....	13

2.7. Parameter Kinerja Jaringan	13
2.7.1. <i>Throughput</i>	14
2.7.2. <i>Packet Delivery Ratio</i>	14
2.7.3. <i>Packet Loss</i>	14
2.7.4. <i>Delay</i>	14
BAB III METODOLOGI	15
3.1. Alur Penelitian.....	15
3.2. Analisis Kebutuhan.....	16
3.2.1. Analisis Kebutuhan <i>Hardware</i>	16
3.2.2. Analisis Kebutuhan <i>Software</i>	16
3.3. Perancangan Skenario.....	17
3.3.1. Skenario Penambahan <i>Node</i>	17
3.3.2. Skenario Penambahan Ukuran Paket Data	21
3.4. Perancangan Simulasi.....	21
3.4.1. Parameter Simulasi	21
3.4.2. Tahapan-Tahapan Simulasi.....	24
3.5. <i>Flowchart</i> Program Simulasi MANET.....	26
3.6. Tahapan-Taapan Pengambilan Data.....	29
BAB IV HASIL DAN PEMBAHASAN	30
4.1. Implementasi	30
4.1.1. Script dan Penjelasan Program Simulasi MANET	30
4.1.2. Tahapan-Tahapan Menjalankan Simulasi.....	37
4.1.3. Tahapan Pengambilan Data	38
4.2. Analisis Data.....	41
4.2.1. Hasil Simulasi Untuk Skenario Penambahan <i>Node</i>	42
4.2.2. Hasil Simulasi Untuk Skenario Penambahan Ukuran Paket Data.....	47
BAB V KESIMPULAN DAN SARAN	53
5.1. Kesimpulan.....	53
5.2. Saran	54
DAFTAR PUSTAKA	55

DAFTAR GAMBAR

Gambar 1.1. Tahapan penelitian	4
Gambar 2.1. MANET (globalspec.com, 2008)	10
Gambar 2.2. Routing protokol	11
Gambar 3.1. Alur penelitian.....	15
Gambar 3.2. 20 nodes.....	18
Gambar 3.3. 10 nodes.....	19
Gambar 3.4. 20 nodes.....	20
Gambar 3.5. 40 nodes.....	20
Gambar 3.6. Tahapan-tahapan simulasi	24
Gambar 3.7. <i>Flowchart</i> simulasi MANET.....	27
Gambar 3.8. Tahapan-tahapan pengambilan data.....	29
Gambar 4.1. Memulai simulasi	38
Gambar 4.2. <i>Output file</i>	39
Gambar 4.3. <i>File .csv</i> skenario 20 node OLSR (1)	39
Gambar 4.4. <i>File .csv</i> skenario 20 node OLSR (2).....	40
Gambar 4.5. <i>File .flowmon</i> skenario ukuran paket 64 bytes DSDV.....	41
Gambar 4.6. Animasi simulasi.....	41
Gambar 4.7. Grafik <i>throughput vs nodes</i>	43
Gambar 4.8. Grafik PDR vs <i>nodes</i>	44
Gambar 4.9. Grafik <i>packet loss vs nodes</i>	45
Gambar 4.10. Grafik <i>delay vs nodes</i>	46
Gambar 4.11. Grafik <i>throughput vs packet size</i>	48
Gambar 4.12. Grafik PDR vs <i>packet size</i>	49
Gambar 4.13. Grafik <i>packet loss vs packet size</i>	50
Gambar 4.14. Grafik <i>delay vs packet size</i>	51

DAFTAR TABEL

Tabel 2.1. Perbandingan parameter.....	9
Tabel 3.1. Parameter simulasi	22
Tabel 4.1. Nilai rata-rata <i>throughput</i> pada skenario penambahan <i>node</i>	42
Tabel 4.2. Nilai rata-rata PDR pada skenario penambahan <i>node</i>	44
Tabel 4.3. Nilai rata-rata <i>packet loss</i> pada skenario penambahan <i>node</i>	45
Tabel 4.4. Nilai rata-rata <i>delay</i> pada skenario penambahan <i>node</i>	46
Tabel 4.5. Nilai rata-rata <i>throughput</i> pada skenario penambahan ukuran paket.....	47
Tabel 4.6. Nilai rata-rata PDR pada skenario penambahan ukuran paket	49
Tabel 4.7. Nilai rata-rata <i>packet loss</i> pada skenario penambahan ukuran paket.....	50
Tabel 4.8. Nilai rata-rata <i>delay</i> pada skenario penambahan ukuran paket.....	51

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi informasi terutama teknologi komunikasi saat ini sangat berkembang pesat, dikarenakan kebutuhan manusia yang semakin banyak dan kompleks. Teknologi komunikasi termasuk di dalamnya jaringan komputer yang terbagi menjadi dua yaitu jaringan berkabel dan jaringan nirkabel (*wireless*). Dalam jaringan nirkabel terdapat dua topologi yaitu topologi dengan infrastruktur dan topologi tanpa infrastruktur atau yang disebut dengan *Mobile Ad Hoc Network* (MANET). Jaringan komputer yang tidak memerlukan infrastruktur ini sangat berguna saat dimana infrastruktur tidak memungkinkan untuk dibangun dan saat infrastruktur yang sudah ada tidak dapat digunakan karena bencana alam.

Dalam MANET tidak membutuhkan infrastruktur tetap, setiap *device* atau *node* dalam MANET tidak hanya bertugas mengirim dan menerima paket data, namun dapat juga sebagai router yang dapat meneruskan paket data ke *node* lainnya. Topologi pada MANET berubah-ubah berdasarkan pergerakan yang dilakukan setiap *node*-nya, hal tersebut membuat tabel routing harus terus diperbarui untuk menentukan jalur pengiriman paket data. Routing pada jaringan komputer merupakan hal yang sangat penting, karena tanpa adanya routing pengiriman paket data tidak dapat dilakukan. Dalam hal ini terdapat dua jenis routing protokol dalam MANET yaitu proaktif routing protokol dan reaktif routing protokol.

Pada penelitian ini hanya akan menggunakan dari satu jenis routing protokol saja, yaitu proaktif routing protokol. Proaktif routing protokol adalah algoritma routing yang mendistribusikan tabel routing ke seluruh *node* yang diketahuinya, tabel routing akan diperbarui secara berkala saat terjadi perubahan *link* untuk memilih jalur terbaik.

Setiap routing protokol memiliki karakteristik, kelebihan dan kekurangannya masing-masing, sehingga akan dilakukan perbandingan performa

dari dua routing protokol dari jenis routing protokol proaktif. Protokol yang akan dibandingkan adalah OLSR (*Optimized Link State Routing*) dengan DSDV (*Destination Sequenced Distance Vector Routing*).

Alasan mengapa memilih protokol OLSR dan DSDV untuk membandingkan performansinya karena beberapa hal berikut, yaitu:

1. Kedua protokol ini banyak digunakan pada MANET dan memiliki algoritma yang berbeda, meskipun dari jenis routing protokol yang sama.
2. Dari penelitian sebelumnya yang juga membahas perbandingan performa protokol OLSR dan DSDV (Irfan et al., 2016) didapatkan hasil bahwa routing protokol OLSR memiliki performa yang lebih baik dibandingkan routing protokol DSDV. Pada penelitian tersebut diketahui bahwa routing protokol OLSR lebih baik dibandingkan DSDV berdasarkan parameter *throughput*, *packet delivery ratio*, *delay* dan konsumsi energi dengan sebuah skenario perubahan kecepatan mobilitas.

Sehingga dengan membandingkan protokol OLSR dan DSDV ini harapannya dapat bagaimana performa kinerja dari kedua protokol tersebut. Kemudian dari hasil kinerja protokol yang sudah di dapat, maka dapat diambil kesimpulan berdasarkan *QoS* dan skenario yang sudah di tentukan, mana yang lebih baik performanya dari kedua protokol tersebut dan yang dapat disesuaikan untuk suatu kebutuhan dan kondisi jaringan tertentu. Untuk membanding kedua protokol tersebut, dalam penelitian ini akan membuat simulasi dengan menggunakan *Network Simulator 3* (NS-3).

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka dirumuskan masalah sebagai berikut:

1. Bagaimana membuat rancangan simulasi MANET menggunakan *Network Simulator 3* (NS3) untuk membandingkan performa dari routing protokol OLSR dan DSDV ?
2. Bagaimana menganalisis perbandingan performansi routing protokol OLSR dan DSDV dengan skenario yang ditentukan pada simulasi MANET

berdasarkan parameter *QoS* yaitu *throughput*, *packet delivery ratio*, *packet loss* dan *delay* ?

1.3. Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Simulasi MANET akan dijalankan menggunakan software *Network Simulator 3* (NS3).
2. Simulasi MANET menggunakan *wifi* Standar IEEE 802.11b.
3. Simulasi akan membandingkan 2 jenis protokol routing yaitu OLSR dan DSDV.
4. Skenario simulasi yang digunakan adalah skenario penambahan *node* dan skenario penambahan ukuran paket data.
5. Parameter *QoS* sebagai perbandingan analisis adalah *throughput*, *packet delivery ratio*, *packet loss* dan *delay*.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Membuat simulasi MANET menggunakan *Networks Simulator 3* (NS3) untuk membandingkan performa dari routing protokol OLSR dan DSDV.
2. Menganalisis perbandingan performansi routing protokol OLSR dan DSDV dengan skenario yang ditentukan pada simulasi MANET berdasarkan parameter *QoS* yaitu *throughput*, *packet delivery ratio*, *packet loss* dan *delay*.

1.5. Manfaat Penelitian

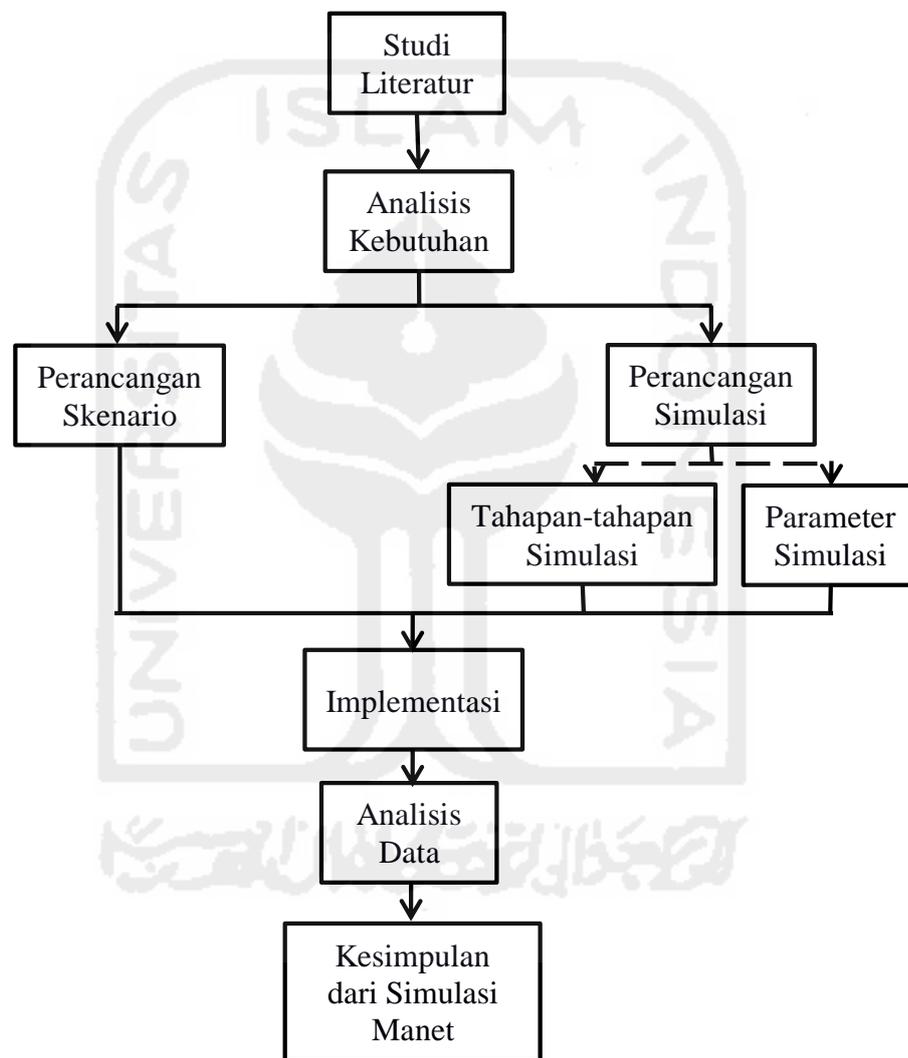
Dari hasil penelitian yang dilakukan, maka akan didapat manfaat yaitu sebagai berikut:

1. Dapat mengetahui performa dan perbandingan kinerja dari kedua routing protokol tersebut.
2. Dapat mengetahui routing protokol yang lebih baik dan yang lebih baik digunakan pada suatu kondisi jaringan tertentu.
3. Membuat rancangan jaringan yang dapat dijadikan referensi saat akan

membuat infrastruktur jaringan yang sebenarnya.

1.6. Metodologi Penelitian

Dalam penelitian ini dilakukan beberapa tahapan agar hasil akhirnya sesuai dengan yang diharapkan. Berikut adalah tahapan pada penelitian ini:



Gambar 1.1. Tahapan penelitian

1. Studi Literatur

Studi literatur adalah tahapan yang melakukan pencarian materi dan referensi yang berkaitan dengan judul penelitian yang akan dikerjakan. Hal tersebut dilakukan agar dapat membantu penulis dalam memahami dari tema yang

diteliti dan mempermudah dalam mengerjakan penelitian ini. Materi dan referensi yang dicari yaitu yang berkaitan dengan simulasi jaringan, *mobile ad hoc network*, routing protokol dan NS3.

2. Analisis Kebutuhan

Analisis kebutuhan adalah tahapan dalam menentukan kebutuhan apa saja yang diperlukan dalam pengerjaan penelitian ini baik itu perangkat keras atau perangkat lunak.

3. Perancangan Skenario

Perancangan skenario dibutuhkan untuk mendukung implementasi simulasi jaringan MANET. Skenario juga dibutuhkan agar data yang dihasilkan dari simulasi lebih detail dan berbeda pada masing-masing kondisi skenario yang telah ditentukan.

4. Perancangan Simulasi

Perancangan simulasi yaitu membuat rancangan algoritma routing protokol yang akan dibandingkan. Pada tahap ini akan merancang beberapa parameter yang akan digunakan dalam simulasi. Kemudian rancangan ini akan dituliskan ke dalam baris kode program dengan menggunakan bahasa pemrograman C++ yang disesuaikan dengan algoritma routing protokol yang telah dirancang sebelumnya. Perancangan simulasi akan terbagi menjadi dua, yaitu perancangan tahapan-tahapan simulasi dan parameter simulasi.

a. Tahapan-tahapan Simulasi

Tahapan-tahapan simulasi ini merupakan tahap yang akan melakukan rancangan tahapan simulasi yang akan dijalankan nantinya. Tahapan-tahapan simulasi yang dibuat ini akan disusun berdasarkan kebutuhan.

b. Parameter Simulasi

Parameter simulasi adalah nilai-nilai yang menjadi patokan dalam menjalankan simulasi. Parameter simulasi ini diimplementasikan dalam bentuk variabel dengan nilai yang selalu tetap pada saat simulasi berjalan dan pada skenario yang berbeda.

5. Implementasi

Implementasi ini dilakukan dengan membuat simulasi MANET yang telah

dirancang sebelumnya dan diimplementasikan ke dalam kode dengan bahasa pemrograman C++. Kemudian membuat skenario pada simulasi tersebut dan dijalankan untuk menghitung beberapa nilai kerja parameter *QoS* yang telah ditetapkan.

6. Analisis Data

Pada tahap analisis data ini melakukan perbandingan dari hasil ukur parameter *QoS* antara routing protokol OLSR dan DSDV yaitu berupa nilai *throughput*, *packet delivery ratio*, *packet loss* dan *delay* yang di dapat dari simulasi skenario yang dijalankan.

7. Kesimpulan dari Simulasi MANET

Pada tahap ini akan diambil kesimpulan dari analisis data yang dihasilkan dari simulasi. Kesimpulan tersebut mencakup bagaimana kinerja routing protokol di semua skenario, bagaimana perbandingan nilai *QoS* antara routing protokol dan routing protokol mana yang lebih baik di kondisi skenario tertentu.

1.7. Sistematika Penulisan

Dalam penulisan laporan penelitian ini terdiri dari lima bab, dengan rincian sebagai berikut:

BAB I PENDAHULUAN

Pada Bab ini berisi gambaran penelitian, seperti latar belakang, batasan masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada Bab ini berisi teori-teori yang berkaitan dengan apa yang akan dibahas mengenai simulasi *mobile ad-hoc network* (MANET) untuk menganalisis perbandingan performansi routing protokol OLSR dan DSDV.

BAB III METODOLOGI

Pada bab ini berisi metodologi yang akan digunakan untuk menyelesaikan

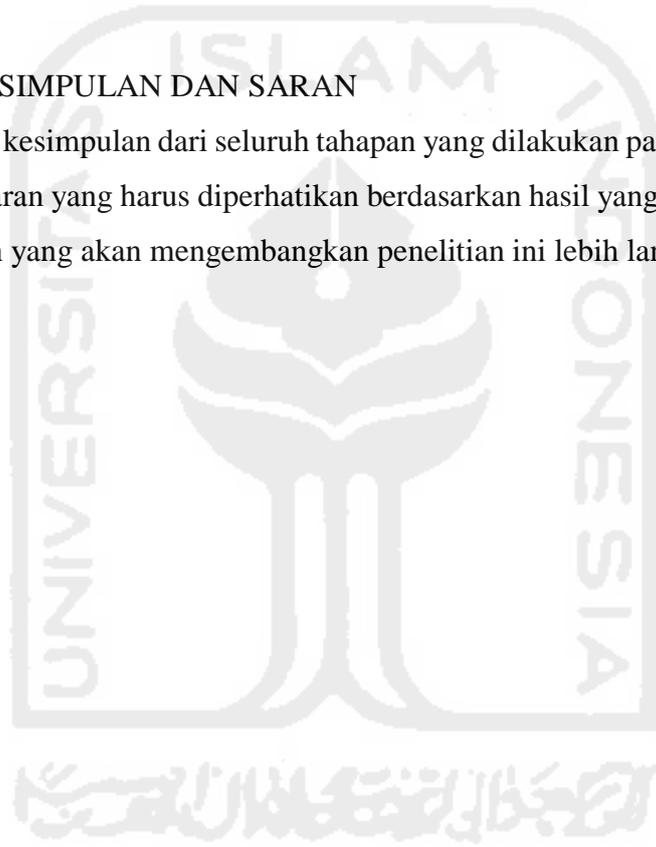
penelitian ini. Bab ini juga berisi analisis kebutuhan penelitian, pembuatan skenario dan perancangan simulasi *mobile ad-hoc network* (MANET) untuk menganalisis perbandingan performansi routing protokol OLSR dan DSDV.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi hasil implementasi simulasi *mobile ad-hoc network* (MANET) dan analisis perbandingan performansi routing protokol OLSR dan DSDV.

BAB V KESIMPULAN DAN SARAN

Merupakan kesimpulan dari seluruh tahapan yang dilakukan pada penelitian ini dan berisikan saran yang harus diperhatikan berdasarkan hasil yang telah didapat untuk peneliti lain yang akan mengembangkan penelitian ini lebih lanjut.



BAB II

LANDASAN TEORI

2.1. Studi Literatur

Pada penelitian sebelumnya yang juga membahas perbandingan performa protokol OLSR dan DSDV, yang berjudul Analisis Perbandingan Kinerja Protokol Routing DSDV dan OLSR Untuk Perubahan Kecepatan Mobilitas pada Standar IEEE 802.11ah (Irfan et al., 2016). Tujuan dari penelitian tersebut adalah untuk mengevaluasi routing protokol OLSR dan DSDV pada standar IEEE 802.11ah dengan menggunakan NS3. Skenario yang digunakan pada penelitian tersebut adalah perubahan kecepatan mobilitas pada setiap *station*, pada skenario ini rentan kecepatan mobilitas yang digunakan adalah 0 - 0,6 m/s, 0,6 - 1,2 m/s, 1,2 - 1,8 m/s, 1,8 - 2,4 m/s, 2,4 - 3 m/s. Penelitian tersebut menganalisis dengan parameter *QoS throughput*, *packet delivery ratio*, *delay*, dan konsumsi energi.

Berdasarkan hasil simulasi pada penelitian tersebut, didapat kesimpulan bahwa routing protokol OLSR memiliki kinerja yang lebih baik dibandingkan routing protokol DSDV pada skenario perubahan kecepatan. Nilai rata-rata *throughput* untuk OLSR adalah 28400 *Bps* sedangkan untuk DSDV adalah 2934 *Bps*. Nilai rata-rata PDR untuk OLSR adalah 14,582% sedangkan untuk DSDV adalah 2,7%. Nilai rata-rata *delay* untuk OLSR adalah 0,04453994 *s* sedangkan untuk DSDV adalah 0,6261986 *s*. Sedangkan rasio perbandingan untuk konsumsi energi antara routing protokol OLSR dan DSDV adalah 1,48% untuk skenario perubahan kecepatan. Diketahui juga pada penelitian tersebut lebih berfokus pada mengevaluasi routing protokol DSDV dan OLSR pada standar IEEE 802.11ah.

2.1.1. Perbandingan Penelitian

Pada penelitian ini akan lebih berfokus untuk membandingkan performa dari routing protokol OLSR dan DSDV secara umum, tidak terfokus pada implementasi terhadap jenis *wireless* tertentu dan juga pada penelitian ini akan menggunakan jenis *wireless* yang berbeda, yaitu standar IEEE 802.11b.

Berdasarkan parameter *throughput*, *packet delivery ratio*, *packet loss* dan *delay* yang akan dijalankan pada dua skenario, yaitu skenario penambahan node dan penambahan ukuran paket data. Beberapa perbandingan parameter antara penelitian sebelumnya dengan penelitian ini, dapat dilihat pada tabel 2.1.

Tabel 2.1. Perbandingan parameter

Parameters	Penelitian Sebelumnya	Penelitian Ini
Routing Protokol	OLSR/DSDV	OLSR/DSDV
<i>Wireless</i>	IEEE 802.11ah	IEEE 802.11b
Parameter <i>QoS</i>	<i>throughput</i> , <i>packet delivery ratio</i> , <i>delay</i> , dan konsumsi energi	<i>throughput</i> , <i>packet delivery ratio</i> , <i>packet loss</i> dan <i>delay</i>
Skenario	<ul style="list-style-type: none"> •Perubahan kecepatan mobilitas pada setiap STA 	<ul style="list-style-type: none"> •Penambahan <i>node</i> •Penambahan ukuran paket data
Transport Layer	UDP	UDP
Data Rate	1,2 Mbps	11 Mbps
Ukuran Paket Data	100 <i>bytes</i>	64 <i>bytes</i> dan 128 <i>bytes</i>

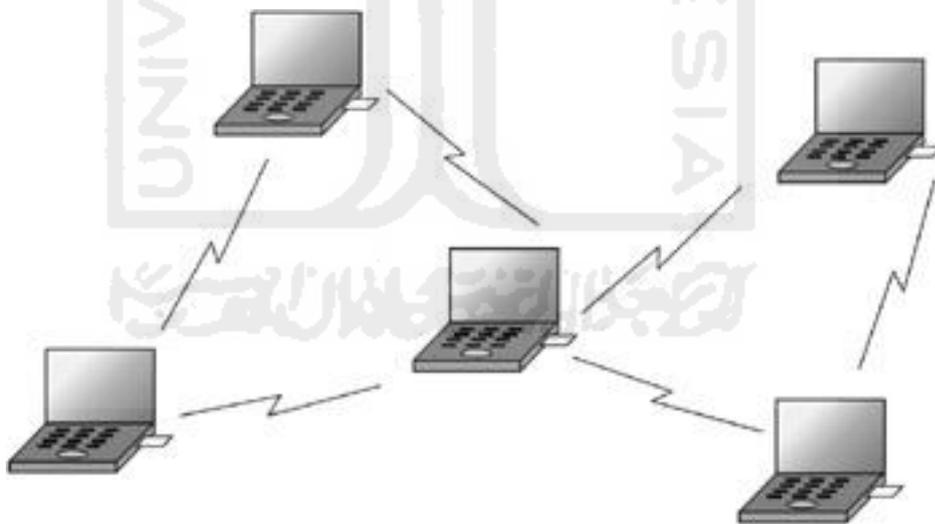
2.2. Jaringan Nirkabel (*Wireless*)

Jaringan nirkabel adalah bidang disiplin yang berkaitan dengan komunikasi antar sistem komputer tanpa menggunakan kabel. Jaringan nirkabel ini sering dipakai untuk jaringan komputer baik pada jarak yang dekat maupun pada jarak jauh (lewat satelit). Bidang ini erat hubungannya dengan bidang telekomunikasi, teknologi informasi, dan teknik komputer. Jenis jaringan yang populer dalam kategori jaringan nirkabel ini meliputi jaringan kawasan lokal nirkabel (*wireless LAN/WLAN*), dan *Wi-Fi*. Jaringan nirkabel biasanya menghubungkan satu sistem komputer dengan sistem yang lain dengan menggunakan beberapa macam media transmisi tanpa kabel, seperti gelombang radio, gelombang mikro, maupun cahaya infra merah (id.wikipedia.org, 2016). Beberapa kategori yang dimiliki jaringan

nirkabel adalah *Wireless Wide Area Networks*, *Wireless Metropolitan Area Networks*, *Wireless Local Area Networks*, *Wireless Personal Area Networks*, *Peer to Peer Network/Ad-Hoc Wireless LAN* (Jiatmiko & Prayudi, 2016).

2.3. *Mobile Ad Hoc Network (MANET)*

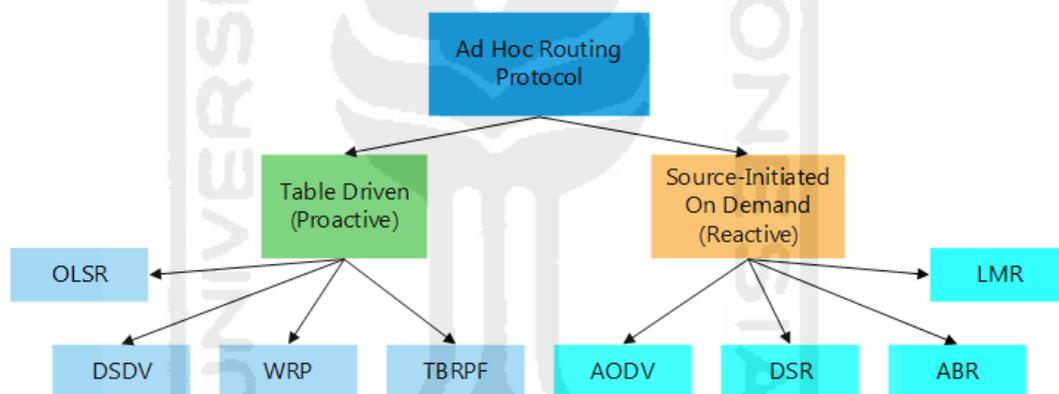
Mobile Ad Hoc Network (MANET) adalah salah satu jenis teknologi *wireless* yang terdiri dari kumpulan *node-node (device)* bergerak dan bersifat dinamis yang bekerja tanpa menggunakan infrastruktur atau administrasi terpusat. *Node* pada MANET dapat bertindak sebagai *user* dan router, *node* bertindak sebagai *user* saat mengirim dan menerima paket dan *node* bertindak sebagai router saat membangun rute dan meneruskan paket ke *node* lainnya. MANET tidak memerlukan dukungan *back-bone* infrastruktur sehingga mudah diimplementasikan dan sangat berguna ketika infrastruktur tidak ada ataupun tidak berfungsi lagi (Mulyanta, 2005). Contoh gambar dari MANET dapat dilihat pada gambar 2.1.



Gambar 2.1. MANET (globalspec.com, 2008)

2.4. Routing Protokol

Routing tak lain adalah untuk menentukan arah paket data dari satu jaringan ke jaringan lain. Penentuan arah ini disebut juga sebagai route, routing dapat diberikan secara dinamis (*dynamic routing*) atau secara statis (*static routing*) (ICT, 2005). Protokol adalah aturan-aturan dan konversi yang menentukan bagaimana sebuah *device* di dalam jaringan komputer untuk saling berkomunikasi. Sehingga routing protokol diperlukan untuk mengatur bagaimana router berkomunikasi antara satu dengan yang lain dalam menyebarkan informasi, yang memungkinkan router untuk memilih rute pada jaringan komputer (Kopp, 1999). Pada jaringan *ad-hoc*, routing protokol dibagi menjadi dua jenis yaitu *Table Driven Routing Protocol* (Proaktif) dan *One-Demand Routing Protocol* (Reaktif).



Gambar 2.2. Routing protokol

2.4.1. Table Driven Routing Protocol (Proaktif)

Pada *Table Driven Routing Protocol* (Proaktif), masing-masing *node* memiliki tabel routing yang lengkap. Artinya sebuah *node* akan mengetahui semua rute ke *node* lain yang berada dalam jaringan tersebut. Setiap *node* akan melakukan pembaruan tabel routing yang dimilikinya secara periodik sehingga perubahan topologi jaringan dapat diketahui setiap interval waktu tersebut. Contoh *table driven routing* adalah OLSR (*Optimized Link State Routing*), DSDV (*Destination Sequenced Distance Vector Routing*), WRP (*Wireless Routing Protocol*) (Sidharta & Widjaja, 2013).

2.4.1.1. OLSR (*Optimized Link State Routing*)

OLSR merupakan routing protokol yang mewarisi algoritma *link state* dan khusus digunakan untuk MANET (Jacquet et al., 2001). Pada OLSR tabel route selalu diperbarui dengan cara mengirim pesan control secara berkala, hal tersebut membuat beban jaringan bertambah. Untuk mengurangi beban jaring OLSR menggunakan mekanisme MPR (*multipoint relay*). Konsep MPR adalah mengurangi pesan broadcast yang memiliki isi informasi yang sama dan mengurangi routing overhead. MPR adalah *node* tetangga yang dipilih oleh suatu *node* dengan spesifikasi tertentu. *Node* ini digunakan untuk memecah paket kemudian setiap *node* MPR memilih sekumpulan node dari tetangga satu *hop*-nya yang digunakan untuk mentransmisikan paket broadcast. Setiap *node* dari MPR memelihara sebuah tabel untuk mengidentifikasi *node* yang akan mengirimkan paket (Mapa, Djanali, & Shiddiqi, 2014). Keuntungan dari penggunaan OLSR adalah dapat mengoptimalkan penggunaan bandwidth yang tersedia (Jacquet et al., 2001).

2.4.1.2. DSDV (*Destination Sequenced Distance Vector Routing*)

DSDV adalah routing protokol konvensional yang diadopsi dari *routing information protocol* (RIP) untuk digunakan pada jaringan *ad hoc*. DSDV menambahkan atribut baru, yaitu *sequence number* untuk setiap entri tabel rute yang masuk. Dengan menggunakan *sequence number*, node dapat membedakan informasi yang kedaluwarsa dan yang baru dengan demikian mencegah pembentukan *routing loops*. Pada DSDV, setiap node di dalam jaringan *ad hoc* harus menyimpan dan menjaga tabel routing yang berisi seluruh alamat tujuan, *metric*, *hop* selanjutnya, dan *sequence number* (He, 2002). Tabel rute pada DSDV akan ditransmisikan dengan cara *broadcast* secara berkala untuk menjaga konsistensi.

2.5. Bahasa Pemrograman C++

Setiap masalah yang akan diselesaikan menggunakan komputer harus dituliskan kedalam sebuah kode program dengan bahasa pemrograman tertentu. Bahasa pemrograman tersebut akan diimplementasikan pada komputer yang akan

menjalankan setiap program yang ditulis dalam satu bahasa. Dari sekian banyak bahasa pemrograman yang ada, pada penelitian ini akan menggunakan bahasa pemrograman C++.

Bjarne Stroustrup pada Bell Labs pertama kali mengembangkan C++ pada awal tahun 1980-an. Untuk mendukung fitur-fitur pada C++, dibangun efisiensi dan sistem support untuk pemrograman tingkat rendah (*low level coding*). Pada C++ ditambahkan konsep-konsep baru seperti *class* dengan sifat-sifatnya seperti *inheritance* dan *overloading*. Selain itu, C++ merupakan bahasa pemrograman yang memiliki sifat *Object Oriented Programming* (OPP) (id.wikipedia.org, n.d.). Untuk menyelesaikan masalah, C++ melakukan langkah pertama dengan mendefinisikan *class-class* yang merupakan *a.-class* yang dibuat sebelumnya sebagai abstraksi dari objek-objek fisik. *Class* tersebut berisi keadaan objek, anggota-anggotanya, dan kemampuan dari objeknya. Setelah beberapa *Class* dibuat, masalah dipecahkan menggunakan *class* (Al Fatta, 2007).

2.6. Network Simulator 3 (NS3)

NS3 adalah sebuah aplikasi simulasi jaringan yang terutama fokus untuk kegiatan penelitian dan pendidikan. NS3 merupakan aplikasi open source yang di bawah lisensi *GNU GPLv2* dan tersedia untuk penelitian dan pengembangan. Simulasi jaringan yang dibuat di NS3 bersifat *real-time* dan hampir sama dengan kejadian nyata (nsgn.org, 2011). Pembuatan simulasi menggunakan NS3 memudahkan untuk menganalisa dan mengevaluasi hasil terhadap suatu model jaringan karena representasi hasil data simulasi dapat ditampilkan dalam bentuk grafik (Jiatmiko & Prayudi, 2016). Aplikasi NS3 ditulis sepenuhnya dalam bahasa C++ dan *Python* sebagai *optional binding*.

2.7. Parameter Kinerja Jaringan

Quality of Service (QoS) adalah parameter yang dibutuhkan untuk mengukur kinerja jaringan. Dengan melihat kinerja jaringan dapat menganalisis konsistensi jaringan, tingkat keberhasilan pengiriman data, waktu *delay* data dan lain-lain. Adapun beberapa parameter yang digunakan sebagai pengukur kinerja

jaringan, sebagai berikut:

2.7.1. *Throughput*

Throughput adalah laju data aktual per satuan waktu. *Throughput* dapat dikatakan sebagai *bandwidth* dalam kondisi sebenarnya (Sidharta & Widjaja, 2013). *Bandwidth* dan *throughput* memiliki perbedaan pada sifat laju datanya. *Bandwidth* bersifat tetap, sementara *throughput* bersifat dinamis atau tidak tetap karena tergantung pada *traffic*. Satuan dari *throughput* adalah Kbps (*Kilo Bit per second*). Rumus untuk menghitung *throughput* adalah:

$$\textit{Throughput} = \frac{\textit{ukuran data yang diterima}}{\textit{waktu pengiriman data}} \quad (2.1)$$

2.7.2. *Packet Delivery Ratio*

Packet Delivery Ratio (PDR) adalah rasio antara banyaknya paket yang diterima oleh tujuan dengan banyaknya paket yang dikirim oleh sumber (Sidharta & Widjaja, 2013). Rumus untuk menghitung PDR adalah:

$$\textit{PDR} = \frac{\textit{paket yang diterima}}{\textit{paket yang dikirim}} \times 100\% \quad (2.2)$$

2.7.3. *Packet Loss*

Packet Loss adalah banyaknya paket yang hilang selama proses pengiriman paket dari *node* asal ke *node* tujuan (Sidharta & Widjaja, 2013). Rumus untuk menghitung *packet loss* adalah:

$$\textit{PL} = \frac{\textit{paket yang dikirim} - \textit{paket yang diterima}}{\textit{paket yang dikirim}} \times 100\% \quad (2.3)$$

2.7.4. *Delay*

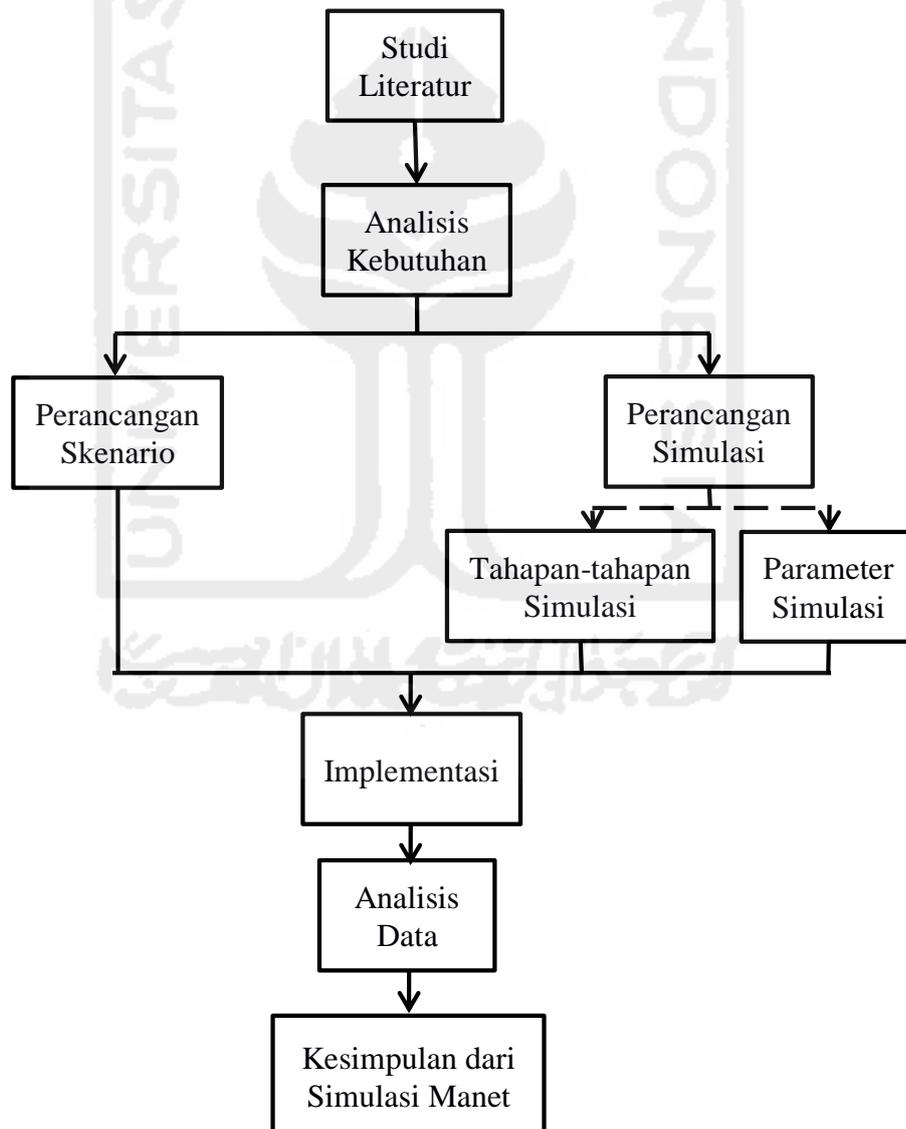
Delay adalah jeda waktu antara paket dikirim dengan diterimanya paket tersebut ke tujuan. Rumus untuk menghitung *delay* adalah:

$$\textit{Delay} = \textit{waktu data diterima} - \textit{waktu data dikirim} \quad (2.4)$$

BAB III METODOLOGI

3.1. Alur Penelitian

Dalam pembuatan simulasi *Mobile Ad-Hoc Network* (MANET), sebelumnya dilakukan tahapan-tahapan untuk menyelesaikannya. Berikut gambaran tahapan-tahapan tersebut dapat dilihat pada diagram alur di bawah ini:



Gambar 3.1. Alur penelitian

3.2. Analisis Kebutuhan

Analisis kebutuhan pada simulasi MANET ini dilihat dari *requirement* dasar yang dibutuhkan dalam menggunakan *script* NS3 dan menganalisis data yang dihasilkan. Hasil analisis kebutuhan dapat dilihat sebagai berikut:

3.2.1. Analisis Kebutuhan *Hardware*

1. Laptop

Sebuah laptop Asus EeePc 1225B, dengan spesifikasi *processor* AMD E-450 dan RAM 2048 MB yang digunakan untuk melakukan simulasi MANET.

3.2.2. Analisis Kebutuhan *Software*

1. Ubuntu 16.10

Sistem operasi Ubuntu ini akan digunakan untuk menjalankan beberapa aplikasi pendukung simulasi MANET seperti NS3 dan NetAnim. Alasan penggunaan Linux Ubuntu pada penelitian ini dikarenakan Linux memiliki *gcc compiler* dan menjadi *platform* yang direkomendasikan oleh NS3.

2. *Network Simulator 3* (NS3)

NS3 adalah aplikasi simulasi jaringan yang target utamanya adalah untuk riset di bidang jaringan dan sarana pembelajaran yang bersifat *Open Source*. NS3 pertama kali di publikasikan pada tahun 2006. Simulasi yang dibuat pada NS3 hampir sama dengan implementasi nyatanya yang bersifat *real-time*. Sehingga hasil yang dihasilkan pada simulasi ini nantinya akan hampir sama dengan yang nyata. Pada penelitian ini menggunakan NS3 versi ns-3.26 yang dirilis pada 3 Oktober 2016. Pada NS3 ketika merilis versi terbaru maka dokumentasi dan contoh simulasi yang terdapat pada Doxygen (*Web Documentation*) akan disesuaikan dengan versi yang dirilis. Sehingga ada kemungkinan simulasi yang berjalan pada versi yang lama tidak dapat digunakan pada versi terbaru dan sebaliknya simulasi yang berjalan pada versi terbaru tidak dapat digunakan pada versi yang lama. Hal ini disebabkan adanya pembaruan, perbaikan dan fitur-fitur terbaru pada

module di versi NS3 yang terbaru.

3. Sublime Text 3

Sublime Text 3 adalah sebuah *syntax editor* yang menggunakan *Python* API dikembangkan oleh Jon Skinner seorang programmer dari Australia. Sublime text mempunyai fitur plugin tambahan yang sangat memudahkan. Desain pada sublime text yang simpel dan bagus sehingga terkesan elegan untuk sebuah *syntax editor*. Sublime text juga memiliki kemudahan dalam *auto complete* dan *auto correct* dalam beberapa bahasa pemrograman salah satunya adalah C++.

4. NetAnim

NetAnim adalah aplikasi pembuatan animasi yang dibuat berdasarkan pada *Qt toolkit*. NetAnim membuat animasi dalam bentuk XML berdasarkan pada *trace file* yang dihimpun selama simulasi berjalan. NetAnim dikemas menjadi satu paket dengan NS3 dalam paket instalasi.

3.3. Perancangan Skenario

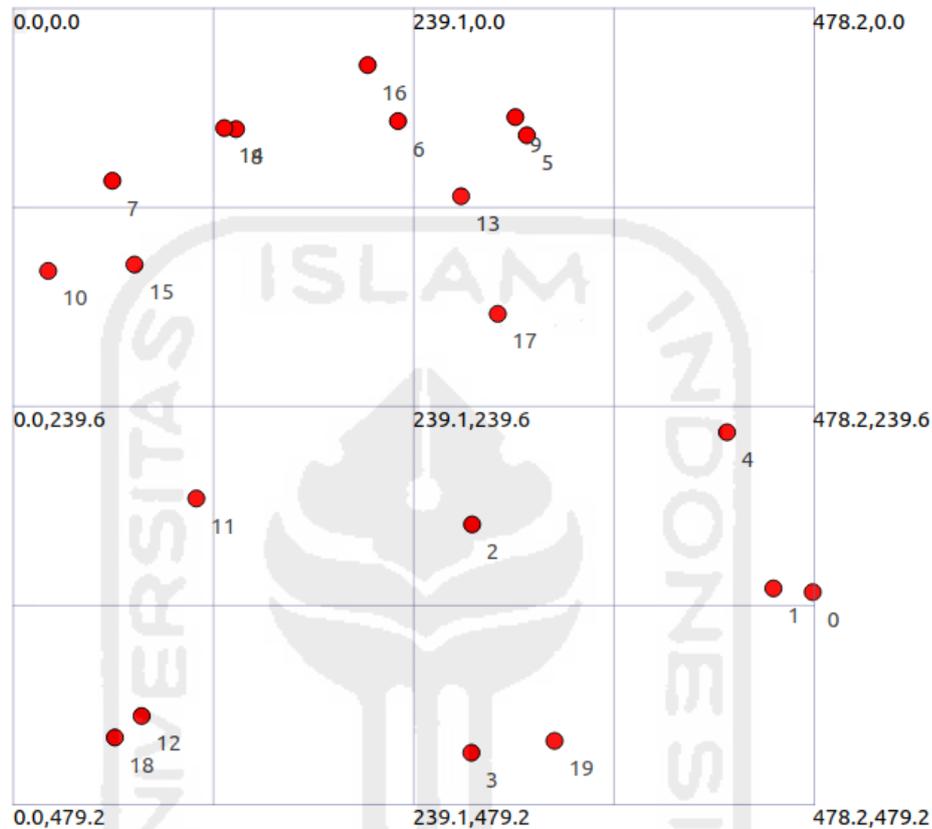
Simulasi pada MANET ini dibutuhkan suatu skenario sebagai pendukung pada saat implementasi. Skenario berguna untuk melihat gambaran dari simulasi yang ingin kita buat dan untuk memberi plot bagaimana pengaruhnya terhadap hasil analisis dari simulasi tersebut. Pada penelitian ini akan dibuat dua buah skenario dengan kondisi tertentu, yaitu:

1. Kondisi dimana luas area yang tetap dengan sebuah jaringan yang mengalami penambahan *node*.
2. Kondisi dimana luas area dan jumlah *node* yang tetap dengan ukuran paket data yang mengalami penambahan.

3.3.1. Skenario Penambahan Node

Pada skenario penambahan node akan dilakukan analisis hasil kerja parameter *QoS* seperti *throughput*, *PDR*, *packet loss* dan *delay* terhadap jaringan yang mengalami penambahan *node*. Kemudian dari hasil analisis parameter *QoS* di kedua protokol routing akan dibandingkan. Berikut contoh gambar yang diambil

dari hasil keluaran NetAnim untuk skenario dengan 20 buah *node*, seperti gambar dibawah:

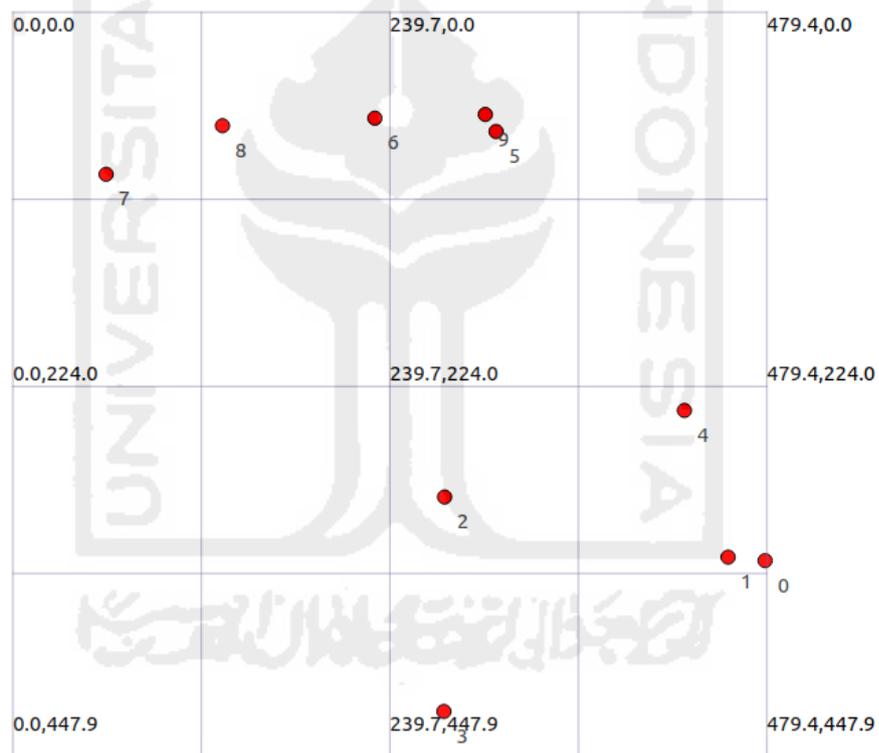


Gambar 3.2. 20 nodes

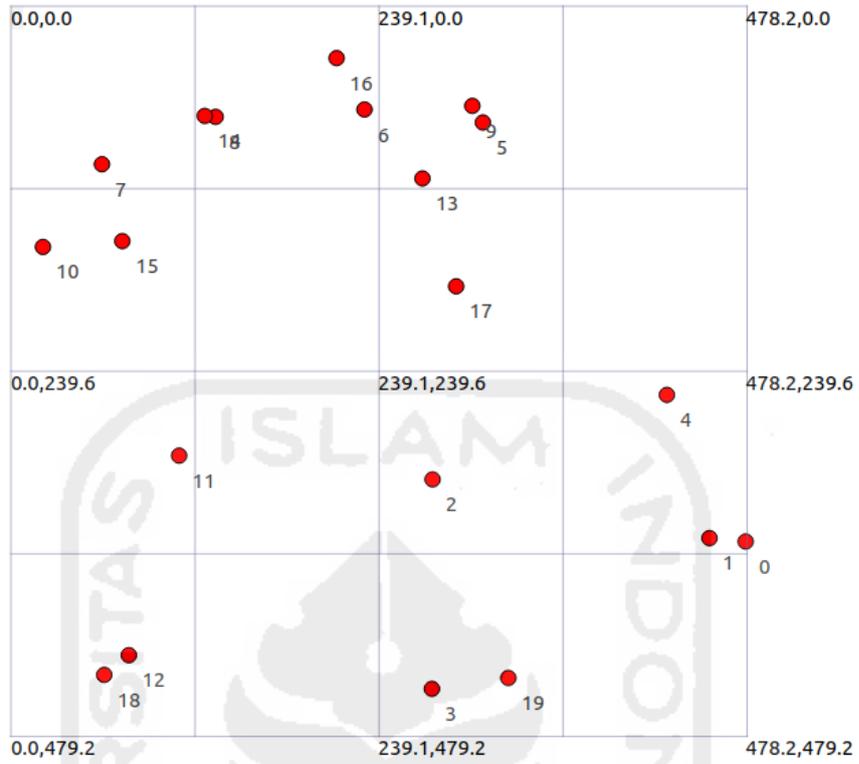
Dari gambar diatas terlihat 20 *node* yang tersebar pada suatu wilayah. *Node* pada wilayah tersebut dimulai dari index ke 0 sampai dengan index ke 19. *Node* index ke 0 memiliki alamat IP 10.1.1.0 dan seterusnya sampai dengan IP 10.1.1.19. Penambahan *node* pada routing protokol OLSR dan DSDV ini sangat mempengaruhi kinerjanya dari protokol tersebut karena membuat padatan pada suatu jaringan. Banyaknya *node* yang ada pada suatu wilayah menjadi salah satu sebab dari terjadinya padatnya suatu jaringan. Dengan luas wilayah yang sama dan semakin banyaknya *node* maka kepadatan jaringan semakin besar. Kepadatan jaringan berpengaruh pada routing tabel yang terbentuk untuk mengirimkan paket data ke suatu *node* dan itu berpengaruh pada proses pengiriman paket data ke semua

node.

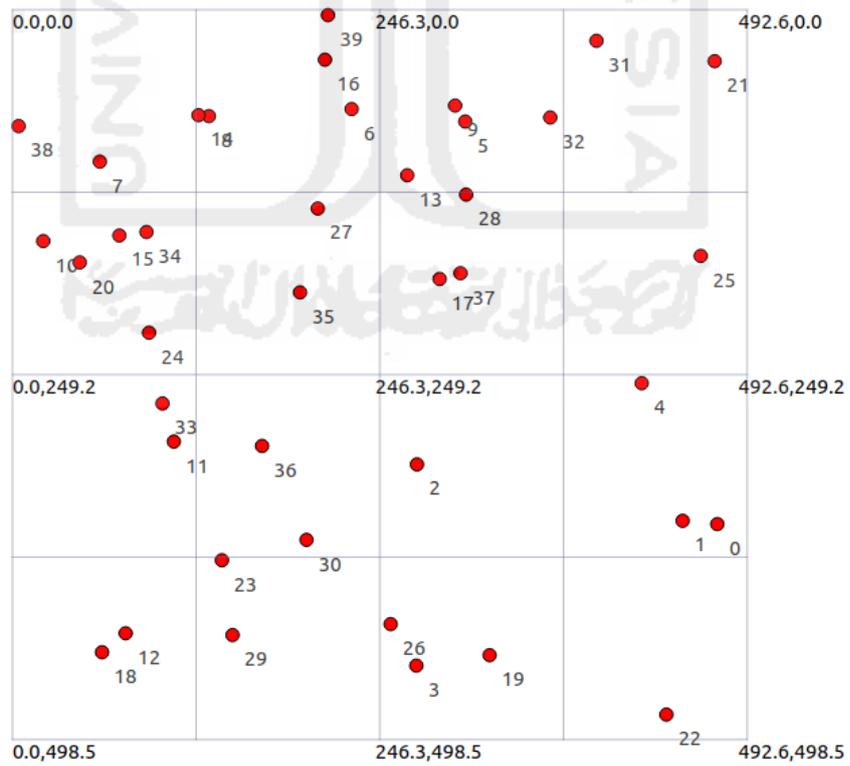
Pada simulasi ini akan dibuat beberapa skenario penambahan *node*, yaitu 10 *node*, 20 *node* dan 40 *node*. Dari penambahan *node* tersebut terbagi 3 *cluster*, yaitu *cluster* kecil untuk 10 *node*, *cluster* sedang untuk 20 *node* dan *cluster* besar untuk 40 *node*. *Node-node* yang berada di suatu wilayah ini akan tersebar secara random. Walaupun tersebar secara *random*, posisi awal *node* antara simulasi satu dengan simulasi berikutnya akan tetap sama. Hal tersebut dikarenakan NS3 memiliki fitur yang nilainya disimpan dalam variabel *streamIndex*. Hal ini dapat dilihat pada gambar dibawah ini:



Gambar 3.3. 10 nodes



Gambar 3.4. 20 nodes



Gambar 3.5. 40 nodes

3.3.2. Skenario Penambahan Ukuran Paket Data

Skenario ini bertujuan untuk menganalisis hasil kerja parameter *QoS* seperti *throughput*, *PDR*, *packet loss* dan *delay* terhadap jaringan yang mengalami penambahan ukuran paket data. Kemudian dari hasil analisis parameter *QoS* di kedua protokol routing akan dibandingkan. Dalam simulasi ini akan digunakan dua ukuran paket data yaitu *64 bytes* dan *128 bytes*. Karena pada skenario ini berfokus pada penambahan ukuran paket data, maka akan dipilih *20 node* yang merupakan jumlah *node* di *cluster* sedang.

3.4. Perancangan Simulasi

Perancangan simulasi yaitu melakukan eksperimen dengan menjalankan beberapa *script* yang tersedia di NS3 untuk membuat rancangan simulasi yang akan digunakan. Pada eksperimen ini dilakukan dengan mengambil referensi dari beberapa penelitian yang telah dilakukan sebelumnya dan dari contoh-contoh yang ada di dalam NS3. Pada beberapa penelitian sebelumnya yang juga membahas tema yang sama dengan penelitian ini, telah menghasilkan beberapa hasil analisis yang dapat menjadi referensi untuk membuat suatu perancangan. Maka beberapa parameter dalam simulasi ini dan skenario yang digunakan berbeda dari penelitian sebelumnya. Terdapat 2 tahapan yang dilakukan pada perancangan ini yaitu menentukan parameter simulasi dan tahapan-tahapan simulasi.

3.4.1. Parameter Simulasi

Parameter simulasi adalah nilai-nilai yang menentukan bagaimana simulasi ini akan berjalan dan nilai-nilainya juga berhubungan dengan beberapa komponen penting di dalam simulasi. Parameter yang berbeda antara simulasi satu dengan yang lainnya akan memberikan hasil keluaran nilai *QoS* yang berbeda. Sehingga nilai parameter pada simulasi ini merupakan variabel yang memiliki nilai konstan selama proses simulasi berjalan dan akan selalu digunakan untuk semua simulasi pada penelitian ini.

Tabel 3.1. Parameter simulasi

Parameter	Nilai
Waktu simulasi	100 detik
Pengiriman paket data	Dimulai pada detik ke 51
Tipe kanal	<i>Wireless</i>
Transport Layer	<i>UDP</i>
Model pergerakan node	<i>Random Waypoint</i>
Kecepatan node	10 m/s
Ukuran paket data	64 <i>bytes</i> dan 128 <i>bytes</i>
Dimensi ruang	500 x 500 m
<i>TX Power</i>	5
Koneksi	4
<i>Pause time</i>	0

1. Waktu simulasi

Waktu simulasi adalah lamanya simulasi yang dijalankan. Waktu simulasi yang dipilih pada penelitian ini adalah 100 detik, lamanya waktu ini dirasa cukup untuk menjalankan simulasi MANET.

2. Pengiriman paket data

Pengiriman paket data adalah waktu dimana node-node melakukan pengiriman paket data. Waktu 50 detik atau setengah dari waktu simulasi dipilih untuk menyediakan waktu bagi routing protokol untuk melakukan *update* routing tabel.

3. Tipe kanal

Tipe kanal adalah media penghubung antar *node*, pada simulasi ini dipilih *wireless* karena OLSR dan DSDV merupakan routing protokol untuk MANET yang bersifat *mobile* atau bergerak secara dinamis.

4. *Traffic sources*

Traffic sources yang digunakan untuk mengirimkan paket data dalam simulasi ini adalah *UDP*.

5. Model pergerakan *node*

Model pergerakan *node* yaitu keadaan *node-node* yang akan bergerak selama simulasi. Dalam simulasi ini digunakan model *random waypoint* yaitu *node-node* yang akan bergerak secara acak. Akan tetapi walaupun *node-node* bergerak secara acak pergerakannya akan tetap sama untuk banyak *node* yang sama pada simulasi yang berbeda.

6. Kecepatan *node*

Kecepatan *node* adalah kecepatan *node* bergerak dari satu posisi ke posisi yang lain. Dalam simulasi ini dipilih kecepatan *node* 10 m/s.

7. Ukuran paket data

Ukuran paket data adalah besar paket yang dikirimkan. Dalam simulasi ini akan menjalankan skenario penambahan ukuran paket data, sehingga akan digunakan 2 ukuran paket data yaitu 64 *bytes* dan 128 *bytes*.

8. Dimensi ruang

Dimensi ruang adalah media tempat berlangsungnya simulasi dan ukurannya dengan satuan m. Dalam simulasi ini digunakan luas 500 x 500 m dirasa cukup untuk menampung 40 *node*.

9. *Tx Power*

Tx power adalah *power* atau daya yang digunakan dalam pengiriman paket data. Dalam simulasi ini dipilih nilai 5 untuk meminimalisir kemungkinan paket data yang hilang pada saat pengiriman.

10. Koneksi

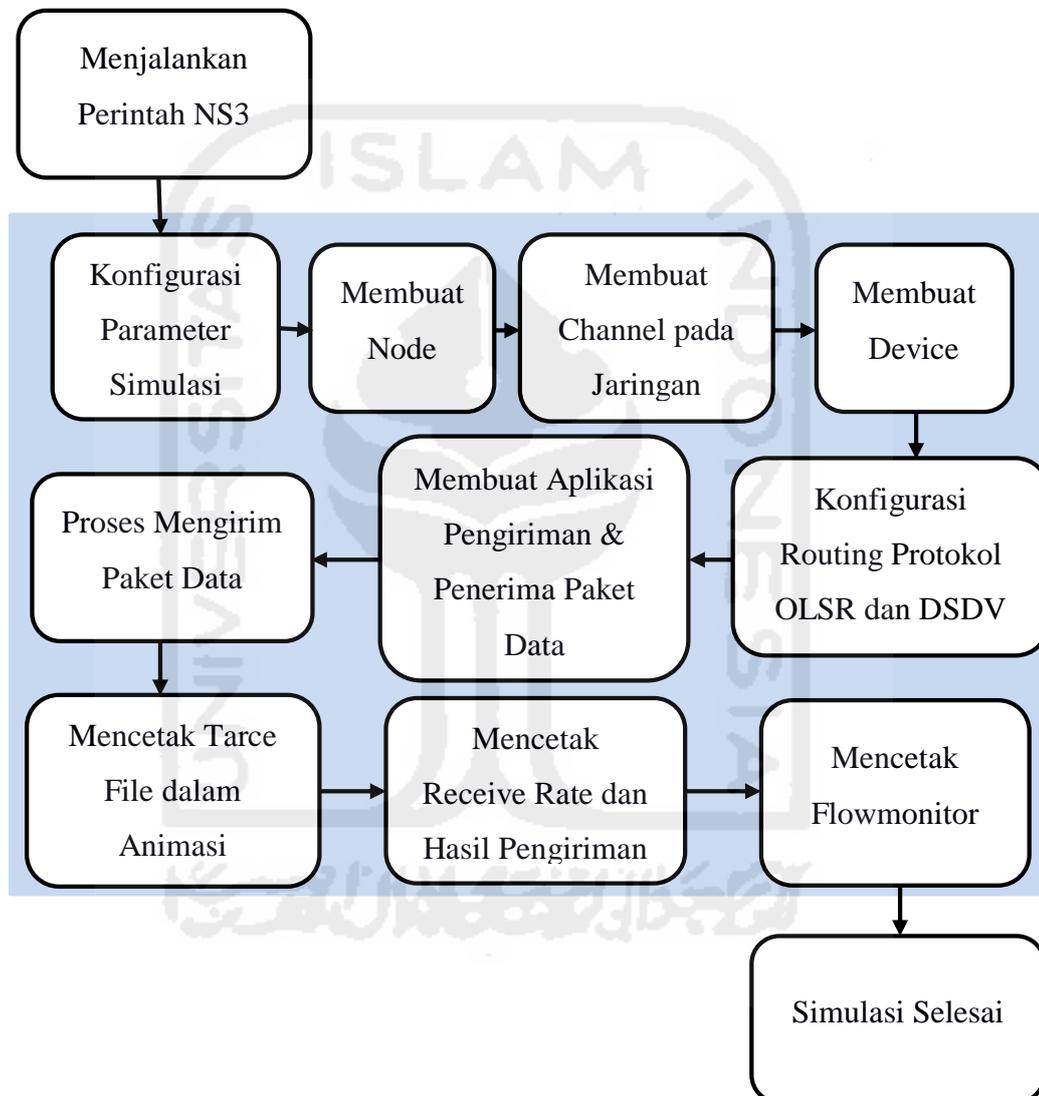
Koneksi adalah hubungan antara 2 *node* atau lebih yang saling berkomunikasi. Dalam penelitian ini koneksi jaringan di set sebanyak 4 koneksi, yang dirasa cukup untuk jaringan MANET 40 *node*.

11. *Pause time*

Pause time adalah waktu dimana *node-node* akan berhenti bergerak secara acak, kemudian *node* bergerak kembali sesuai kecepatan yang telah ditentukan. Dalam simulasi ini *Pause Time* di set 0, agar dari awal *node* terus bergerak dan kecepatan pergerakan *node* tetap sama pada jumlah *node* yang sama dengan simulasi yang berbeda.

3.4.2. Tahapan-Tahapan Simulasi

Pada tahapan-tahapan simulasi ini akan memaparkan poin-poin penting secara garis besar bagaimana langkah-langkah simulasi dijalankan. Berikut ini gambaran tahapan-tahapan simulasi:



Gambar 3.6. Tahapan-tahapan simulasi

1. Menjalankan perintah NS3

Pada tahap pertama, *user* akan menjalankan perintah untuk menjalankan *source script* yang ada di dalam NS3.

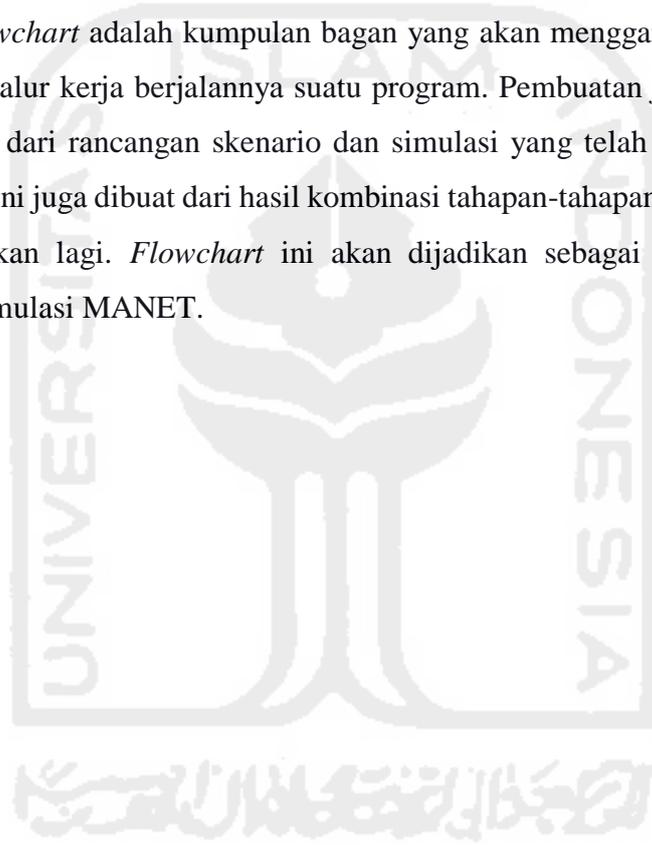
2. Konfigurasi parameter simulasi
Program akan melakukan pengaturan dan memberikan nilai untuk beberapa parameter yang bersifat statis.
3. Membuat *node*
Saat akan menjalankan perintah dengan *script* program terdapat parameter untuk menentukan banyaknya *node* pada simulasi. Setelah *node-node* dibuat pertama kali sesuai dengan yang diinginkan, kemudian dipasangkan aplikasi untuk pengiriman dan penerimaan paket data yang dihubungkan dengan *channel*.
4. Membuat *channel* pada jaringan
Channel dibuat untuk menghubungkan antar *node* dan sebagai media untuk pengiriman, perantaraan, penerimaan paket data.
5. Membuat *device*
Device dibuat untuk menghubungkan *node* dan *channel*. Selain itu, *device* juga berfungsi sebagai tempat *MAC* dan *IP address*.
6. Konfigurasi routing protokol OLSR dan DSDV
Pada tahap ini akan dipilih routing protokol yang akan digunakan dalam simulasi dan juga dilakukan pembuatan *routing table list*.
7. Membuat aplikasi pengiriman dan penerima paket data
Aplikasi berfungsi untuk melakukan pengiriman dan penerimaan paket data selama simulasi berjalan. Setiap *node* yang melakukan pengiriman dan penerimaan paket data akan dipasangkan aplikasi selama simulasi berjalan.
8. Proses mengirim pakat data
Saat simulasi dimulai beberapa *node* akan melakukan *broadcast* untuk membuat routing table kemudian node akan melakukan pengiriman paket data.
9. Mencetak *trace file* dalam animasi
Saat simulasi berjalan, program juga membuat *trace file* yang disimpan dalam bentuk animasi dengan ekstensi *.xml*. *File* animasi tersebut dapat dilihat dan dijalankan menggunakan NetAnim.
10. Mencetak *receive rate* dan detail pengiriman paket

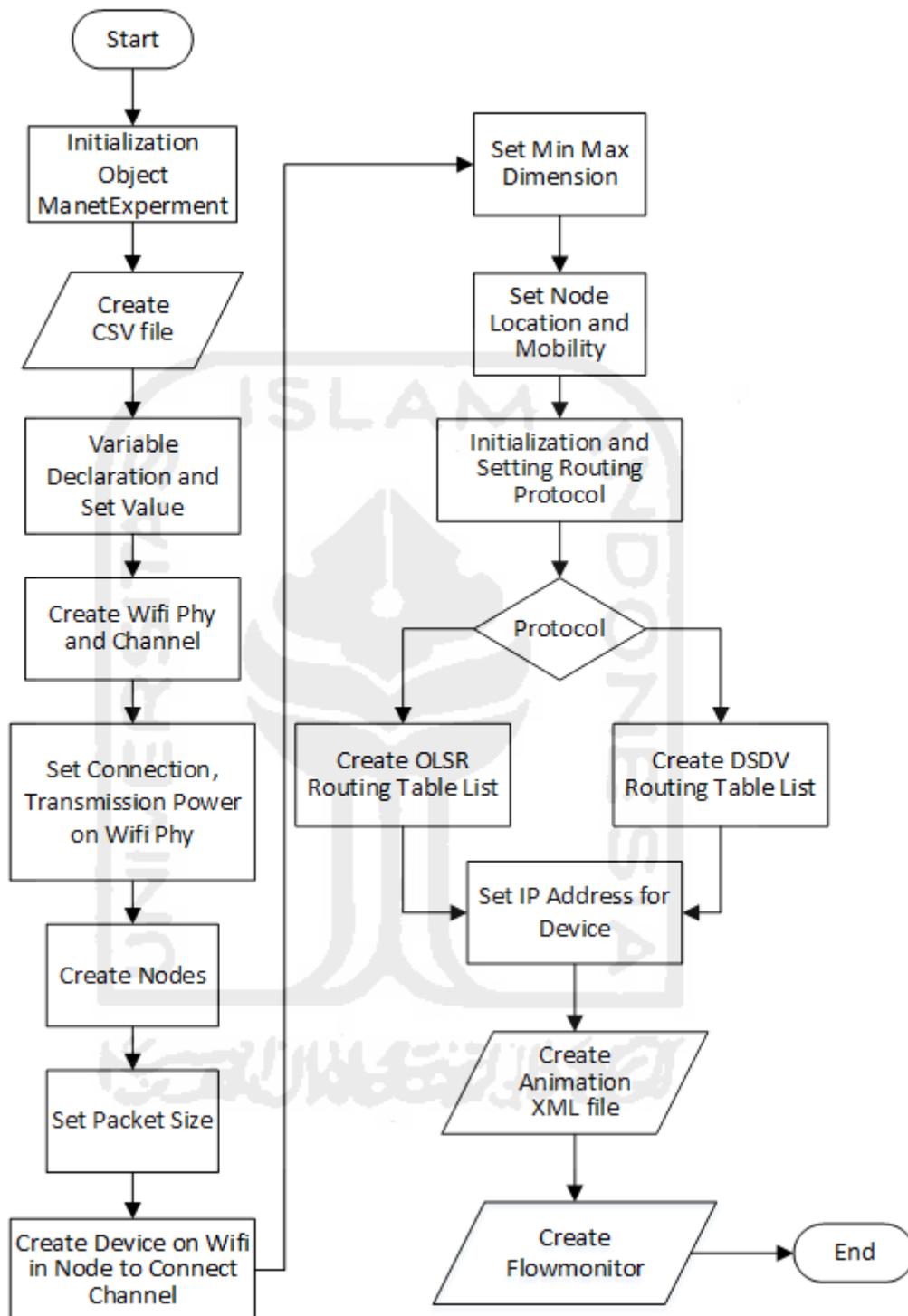
Kemudian, NS3 akan menghasilkan sebuah file berekstensi .csv yang berisi pengiriman data berupa *received rate*. *File* ini akan digunakan untuk menghitung throughput. NS3 juga menyediakan fitur untuk mencatat hasil pengiriman dan *delay* paket data kedalam sebuah *file* berekstensi .flowmon.

11. Simulasi selesai

3.5. **Flowchart Program Simulasi MANET**

Flowchart adalah kumpulan bagan yang akan menggambarkan bagaimana urutan dan alur kerja berjalannya suatu program. Pembuatan *flowchart* ini sangat dipengaruhi dari rancangan skenario dan simulasi yang telah dibuat sebelumnya. *Flowchart* ini juga dibuat dari hasil kombinasi tahapan-tahapan simulasi yang lebih dikembangkan lagi. *Flowchart* ini akan dijadikan sebagai rujukan pembuatan program simulasi MANET.





Gambar 3.7. Flowchart simulasi MANET

Saat ingin menjalankan simulasi maka user harus menuliskan perintah yaitu banyaknya *node* dan jenis routing protokol. Kemudian program akan menginisialisasi sebuah objek dari kelas *ManetExperiment*. Selanjutnya program

akan membuat file .csv dan juga mengisi tabel header di dalamnya. Program kemudian akan menset beberapa nilai default ke dalam beberapa variabel yang ada pada fungsi run.

Program kemudian akan mengkonfigurasi *wifi* yang akan digunakan sebagai media pengiriman dan penerimaan paket data dan juga membuat channel sebagai penghubung antar *node*. Program akan mengatur daya transmisi dan jumlah koneksi pada simulasi. Setelah itu program akan membuat *node* yang akan ditentukan oleh *user*. Selanjutnya program membuat *device*, dimana *device* ini akan di install pada masing-masing *node*, *device* digunakan untuk menghubungkan antara *node* dan *channel*.

Program akan membuat sebuah dimensi wilayah yang akan digunakan sebagai tempat melakukan simulasi. *Node-node* akan ditempatkan di dimensi tersebut dan diatur pergerakannya.

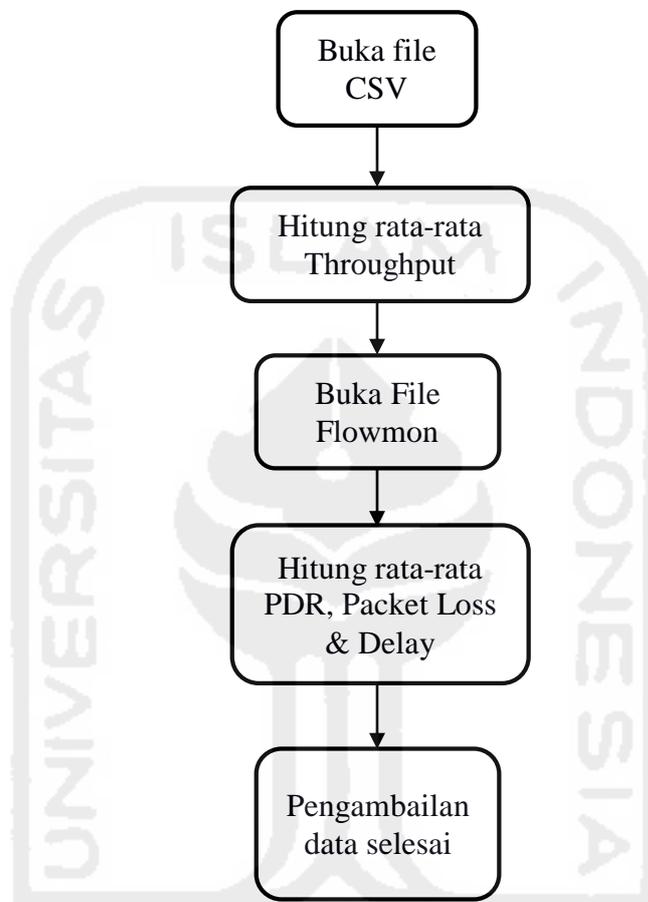
Program akan melakukan inisialisasi routing protokol dan membuat routing tabel berdasarkan protokol yang dipilih saat akan menjalankan perintah simulasi program, kemudian akan melakukan pengaturan range *IP address* dan memberikan *IP* pada masing-masing *node*. Kemudian program akan mengatur kapan aplikasi mulai mengirim dan berhenti mengirim data.

Selanjutnya program akan membuat *trace file* dengan format .tr. Program juga akan mencetak berapa besar bit dan paket data yang diterima oleh *node* tujuan dan dikelompokkan setiap detik. Program akan membuat *file* animasi dalam format .xml yang dihasilkan dari *trace* mobilitas *node*.

Kemudian program akan memasuki bagian pengumpulan data untuk menghitung *delay* dan pengiriman paket. Awalnya program akan menentukan lokasi dan pergerakan *node*, kemudian program akan menentukan *node* tujuan dan sumber yang akan mengirimkan data. Program akan men-generate *traffic* berupa banyaknya paket data yang akan dikirimkan. Program akan melakukan konfigurasi *FlowMonitor* untuk menghitung delay dan pengiriman paket. Kemudian program akan mulai melakukan perhitungan dimulai dari perhitungan paket dikirim, paket diterima, serta minimum, maksimum, total dan rata-rata delay. Lalu program akan mencatat hasil tersebut kedalam *file* yang berekstensi .flowmon. Program selesai

dan simulasi berhenti.

3.6. Tahapan-Taapan Pengambilan Data



Gambar 3.8. Tahapan-tahapan pengambilan data

Untuk melakukan pengambilan data, terdapat 2 *file* yaitu *CSV* dan *flowmon* sebagai rujukan untuk menghitung *throughput*, *PDR*, *packet loss*, dan *delay*. Tahap pertama dalam menganalisis data adalah dengan membuka *file CSV* yang digunakan untuk mencari nilai rata-rata *throughput*.

Kemudian menghitung rata-rata *throughput* dari nilai-nilai yang ada di dalam *file CSV*. Selanjutnya membuka *file flowmon* yang digunakan untuk mencari nilai rata-rata dari *PDR*, *packet loss*, dan *delay*. Kemudian menghitung rata-rata nilai *PDR*, *packet loss* dan *delay*. Setelah menghitung maka akan didapat nilai rata-rata *throughput*, *PDR*, *packet loss* dan *delay*. Pengambilan data selesai.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Implementasi

Pada tahap implementasi ini akan membahas mengenai tahapan-tahapan yang dilakukan dalam pembuatan simulasi MANET routing protokol OLSR dan DSDV. Setelah melakukan perancang sebelumnya, kemudian selanjutnya membuat *script* program, menjalankan program simulasi, dan mengumpulkan data yang dihasilkan dari simulasi. Tahapan-tahapan tersebut adalah tahapan yang dilakukan sebelum menganalisis data dan membuat kesimpulan.

Dalam pembuatan program simulasi ini dibangun berdasarkan rancangan tahapan-tahapan simulasi dan *flowchart* yang telah dibuat sebelumnya. Dalam NS3 menyediakan modul-modul dan contoh-contoh *script* yang digunakan untuk membantu dalam pembuatan program ini. Pada program simulasi ini memanfaatkan contoh *script* manet routing *compare* yang tersedia di NS3 yang kemudian akan ditambahkan *script* yang dibutuhkan.

Setelah simulasi program dibuat sesuai rancangan, kemudian simulasi dijalankan, yang akan menghasilkan beberapa *output*, yang berisi kinerja dari routing protokol selama simulasi berjalan. *Output* dari simulasi ini adalah berupa file .xml, .csv dan .flowmon, kemudian akan digunakan untuk pengambilan data untuk di analisis.

4.1.1. Script dan Penjelasan Program Simulasi MANET

1. Fungsi Utama

Fungsi utama berisi inisialisasi kelas untuk eksperimen, pembuatan file *csv* dan perintah menjalankan simulasi.

```
int
main (int argc, char *argv[])
{
    ManetExperiment experiment;
    std::string CSVfileName = experiment.CommandSetup
(argc, argv);
```

```

std::ofstream out (CSVfileName.c_str ());
out << "SimulationSecond," <<
"ReceiveRate," <<
"PacketsReceived," <<
"NumberOfSinks," <<
"RoutingProtocol," <<
"TransmissionPower" <<
std::endl;
out.close ();
int nSinks = 4;
double txp = 5;
experiment.Run (nSinks, txp, CSVfileName);
}

```

2. Inisialisasi Kelas utama *ManetExperiment*

Kelas ini merupakan kelas utama yang berisi beberapa variabel utama dan fungsi-fungsi yang akan dijalankan saat simulasi.

```

class ManetExperiment
{
public:
    ManetExperiment ();
    void Run (int nSinks, double txp, std::string
CSVfileName);
    std::string CommandSetup (int argc, char **argv);
private:
    Ptr<Socket> SetupPacketReceive (Ipv4Address addr,
Ptr<Node> node);
    void ReceivePacket (Ptr<Socket> socket);
    void CheckThroughput ();
    uint32_t port;
    uint32_t bytesTotal;
    uint32_t packetsReceived;
    std::string m_CSVfileName;
    int m_nSinks;
    std::string m_protocolName;
    double m_txp;
    uint32_t m_protocol;
    uint32_t m_nodesTotal;
    uint32_t m_packetS;
    std::string m_sizeP;
};
ManetExperiment::ManetExperiment ()
: port (9),
  bytesTotal (0),
  packetsReceived (0),
  m_CSVfileName ("simulasi-manet-olsr-dsdv.csv"),
  m_protocol (1),
  m_nodesTotal (20),
  m_packetS (1)
{}

```

3. *Print Received Packet*

Print Received Packet adalah untuk mencetak paket data yang dikirim dan diterima oleh sebuah *node* pada saat simulasi berjalan.

```
static inline std::string
PrintReceivedPacket (Ptr<Socket> socket, Ptr<Packet> packet,
Address senderAddress){
    std::ostringstream oss;
    oss << Simulator::Now ().GetSeconds () << " " << socket-
>GetNode ()->GetId ();
    if (InetSocketAddress::IsMatchingType (senderAddress)){
        InetSocketAddress addr =
InetSocketAddress::ConvertFrom (senderAddress);
        oss << " received one packet from " << addr.GetIpv4
());}
    else{
        oss << " received one packet!";}
    return oss.str ();}
void ManetExperiment::ReceivePacket (Ptr<Socket> socket){
    Ptr<Packet> packet;
    Address senderAddress;
    while ((packet = socket->RecvFrom (senderAddress))){
        bytesTotal += packet->GetSize ();
        packetsReceived += 1;
        NS_LOG_UNCOND (PrintReceivedPacket (socket, packet,
senderAddress));}
}
Ptr<Socket>
ManetExperiment::SetupPacketReceive (Ipv4Address addr,
Ptr<Node> node){
    TypeId tid = TypeId::LookupByName
("ns3::UdpSocketFactory");
    Ptr<Socket> sink = Socket::CreateSocket (node, tid);
    InetSocketAddress local = InetSocketAddress (addr, port);
    sink->Bind (local);
    sink->SetRecvCallback (MakeCallback
(&ManetExperiment::ReceivePacket, this));
    return sink;
}
```

4. Perhitungan *Throughput*

- a. Cek dan cetak *received rate* dan waktu simulasi untuk menghitung *throughput*.

Program akan menghitung berapa besarnya data yang diterima pada masing-masing *node* dalam setiap detiknya selama simulasi berjalan. Kemudian data-data tersebut akan di catat ke dalam file .csv.

```
void ManetExperiment::CheckThroughput (){
    double kbs = (bytesTotal * 8.0) / 1024;
```

```

bytesTotal = 0;
std::ofstream out (m_CSVfileName.c_str (), std::ios::app);
out << (Simulator::Now ()).GetSeconds () << ", "
    << kbs << ", "
    << packetsReceived << ", "
    << m_nSinks << ", "
    << m_protocolName << ", "
    << m_txp << " "
    << std::endl;
out.close ();
packetsReceived = 0;
Simulator::Schedule (Seconds (1.0),
&ManetExperiment::CheckThroughput, this);
}

```

b. Set *sinks* dan *source node*

Program akan menentukan *node* yang akan mengirim paket (sumber) dan *node* yang akan menerima paket (tujuan). Kemudian juga ditentukan kapan *node* akan memulai mengirim paket dan kapan *node* berhenti mengirim paket.

```

for (int i = 0; i < nSinks; i++)
{
    Ptr<Socket> sink = SetupPacketReceive
(adhocInterfaces.GetAddress (i), adhocNodes.Get (i));
    AddressValue remoteAddress (InetSocketAddress
(adhocInterfaces.GetAddress (i), port));
    onoff1.SetAttribute ("Remote", remoteAddress);
    Ptr<UniformRandomVariable> var =
CreateObject<UniformRandomVariable> ();
    ApplicationContainer temp = onoff1.Install
(adhocNodes.Get (i + nSinks));
    temp.Start (Seconds (var->GetValue (50.0, 51.0)));
    temp.Stop (Seconds (TotalTime));
}

```

5. Fungsi *Command Setup*

Fungsi *Command Setup* merupakan fungsi untuk menyimpan nilai parameter yang di masukkan oleh *user* saat menjalankan perintah `./waf` menggunakan terminal. Nilai parameter itu di isi oleh *user* sesuai dengan skenario yang ingin dijalankan, seperti jumlah *node* dan ukuran paket data.

```

std::string ManetExperiment::CommandSetup (int argc, char
**argv)
{
    CommandLine cmd;
    cmd.AddValue ("csv", "The name of the CSV output file
name", m_CSVfileName);
}

```

```

cmd.AddValue ("protocol", "1=OLSR;2=DSDV", m_protocol);
cmd.AddValue ("nodes", "Number of Nodes", m_nodesTotal);
cmd.AddValue ("packet", "Packet Size", m_packetS);
cmd.Parse (argc, argv);
return m_CSVfileName;
}

```

6. Pembuatan *Node*

Program akan membuat *node* terlebih dahulu yang merupakan konsep dari NS3.

```

NodeContainer adhocNodes;
adhocNodes.Create (nodesTotal);

```

7. Inisialisasi *Wifi Settings* dan *Wifi Phy channel*

Wifi settings adalah jenis *wifi* yang digunakan pada simulasi ini. *Wifi* berfungsi sebagai sarana pengirim dan penerima data pada program. Kemudian program membuat *channel* yang berfungsi sebagai penghubung antar *node*.

```

WifiHelper wifi;
wifi.SetStandard (WIFI_PHY_STANDARD_80211b);

YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default
();
YansWifiChannelHelper wifiChannel;
wifiChannel.SetPropagationDelay
("ns3::ConstantSpeedPropagationDelayModel");
wifiChannel.AddPropagationLoss
("ns3::FriisPropagationLossModel");
wifiPhy.SetChannel (wifiChannel.Create ());

```

8. Transmisi *Power*

Transmisi *power* adalah penentu besarnya daya yang akan digunakan untuk mengirim paket.

```

wifiPhy.Set ("TxPowerStart", DoubleValue (txp));
wifiPhy.Set ("TxPowerEnd", DoubleValue (txp));

```

9. Membuat *device*

Pembuatan *device* untuk setiap *node* yang berfungsi sebagai penghubung antara *node* dan *channel*.

```

wifiMac.SetType ("ns3::AdhocWifiMac");
NetDeviceContainer adhocDevices = wifi.Install (wifiPhy,

```

```
wifiMac, adhocNodes);
```

10. Dimensi wilayah simulasi

Program akan membuat dimensi wilayah sebagai tempat simulasi berjalan. Dimensi tersebut di representasikan dengan bentuk axis X dan Y. Satuan dari dimensi tersebut adalah meter. Wilayah simulasi tersebut berbentuk persegi atau persegi panjang sesuai ukuran yang dimasukkan pada X dan Y. Wilayah simulasi dapat dilihat secara visual menggunakan *file* .xml yang dijalankan pada aplikasi NetAnim.

```
ObjectFactory pos;
  pos.SetTypeId ("ns3::RandomRectanglePositionAllocator");
  pos.Set ("X", StringValue
("ns3::UniformRandomVariable[Min=0.0|Max=500.0]"));
  pos.Set ("Y", StringValue
("ns3::UniformRandomVariable[Min=0.0|Max=500.0]"));
```

11. Set posisi awal *node* dan kecepatan *node* menjadi *static*

Pada skenario penambahan *node*, dibutuhkan posisi awal *node* yang tetap sama pada routing protokol OLSR atau DSDV, maupun pada jumlah *node* 20 atau 40, agar validitas dan konsistensi skenario tetap terjamin saat ingin dibandingkan. Sehingga pada langkah ini di konfigurasi *streamIndex* menjadi 0. Kemudian program juga mengatur pergerakan dan kecepatan *node* agar tetap dan konsisten selama simulasi berjalan.

```
Ptr<PositionAllocator> taPositionAlloc = pos.Create ()-
>GetObject<PositionAllocator> ();
  streamIndex += taPositionAlloc->AssignStreams
(streamIndex);
  std::stringstream ssSpeed;
  ssSpeed << "ns3::UniformRandomVariable[Min=0.0|Max=" <<
nodeSpeed << " ]";
  std::stringstream ssPause;
  ssPause << "ns3::ConstantRandomVariable[Constant=" <<
nodePause << " ]";
  mobilityAdhoc.SetMobilityModel
("ns3::RandomWaypointMobilityModel",
  "Speed", StringValue (ssSpeed.str ()),
  "Pause", StringValue (ssPause.str ()),
  "PositionAllocator", PointerValue (taPositionAlloc));
  mobilityAdhoc.SetPositionAllocator (taPositionAlloc);
  mobilityAdhoc.Install (adhocNodes);
  streamIndex += mobilityAdhoc.AssignStreams (adhocNodes,
streamIndex)
```

12. Inisialisasi dan pengaturan routing protokol OLSR dan DSDV

Program akan melakukan inisialisasi routing protokol, kemudian program akan memilih routing protokol OLSR atau DSDV sesuai dengan perintah yang dimasukkan *user*.

```
OlsrHelper olsr;
DsdvHelper dsdv;
Ipv4ListRoutingHelper list;
InternetStackHelper internet;
switch (m_protocol)
{
case 1:
list.Add (olsr, 100);
m_protocolName = "OLSR";
internet.SetRoutingHelper (list);
internet.Install (adhocNodes);
break;
case 2:
list.Add (dsdv, 100);
m_protocolName = "DSDV";
internet.SetRoutingHelper (list);
internet.Install (adhocNodes);
break;
default:
NS_FATAL_ERROR ("No such protocol:" << m_protocol);
}
```

13. Set IP Address

Program membuat *range IP Address*, kemudian akan memberikan *IP* pada setiap *node*.

```
Ipv4AddressHelper addressAdhoc;
addressAdhoc.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer adhocInterfaces;
adhocInterfaces = addressAdhoc.Assign (adhocDevices);
```

14. Perhitungan pengiriman paket menggunakan *FlowMonitor*

Program akan melakukan perhitungan pengiriman paket seperti *delay*, jumlah paket dikirim dan diterima, waktu pengiriman, dan paket yang hilang dengan menggunakan *module FlowMonitor* yang ada di NS3. Setelah program melakukan perhitungan, kemudian hasilnya akan dicetak kedalam sebuah file *.flowmon*.

```
Ptr<FlowMonitor> flowmon;
FlowMonitorHelper flowmonHelper;
```

```

flowmon = flowmonHelper.InstallAll ();
flowmon->SerializeToXmlFile ((tr_name + ".flowmon").c_str(),
false, false);

```

15. Membuat *AnimationInterface*

Setelah melakukan perhitungan paket data dan mencatat pergerakan *node*, kemudian program akan mencetak hasil *trace node* dalam bentuk animasi yang akan disimpan kedalam file *.xml*.

```

AnimationInterface anim ((tr_name + ".xml").c_str());

```

16. Simulasi dimulai dan berhenti

```

Simulator::Stop (Seconds (TotalTime));
Simulator::Run ();
. . .
Simulator::Destroy ();

```

4.1.2. Tahapan-Tahapan Menjalankan Simulasi

Setelah implementasi pembuatan program selesai, kemudian program akan dijalankan dengan perintah-perintah tertentu yang sudah di konfigurasi sebelumnya. Perintah-perintah tersebut akan menyesuaikan dengan skenario simulasi yang akan dijalankan.

1. Sebelum menjalankan program, *user* terlebih dahulu harus masuk ke direktori aplikasi NS3 yang terdapat *file .waf*. Kemudian *user* membuka *terminal* untuk mengetik perintah menjalankan program simulasi, yaitu :

```

./waf --run "manet-routing --protokol=1 --nodes=20 --paket=1
csv=OLSR20Node.csv"

```

Penjelasan:

- ./waf --run* : perintah *default* NS3 untuk menjalankan program
- manet-routing* : nama *file* program simulasi
- protokol* : jenis protokol yang dipilih, 1=OLSR dan 2=DSDV
- nodes* : jumlah *node*
- paket* : ukuran paket data, 1=64 *bytes* dan 2=128 *bytes*

csv : nama *file* .csv.

- Setelah menjalankan perintah eksekusi program, kemudian NS3 akan melakukan simulasi dengan mem-*build* dan meng-*compile* program dan simulasi berjalan, seperti pada gambar 4.1.

```

root@Abu-PC:/home/egi/ns-allinone-3.26/ns-3.26# ./waf --run "scratch/simulasi-ma
net-olsr-dsdy --protocol=1 --nodes=20 --packet=1 --csv=OLSR20Nodes.csv"
Waf: Entering directory `/home/egi/ns-allinone-3.26/ns-3.26/build'
Waf: Leaving directory `/home/egi/ns-allinone-3.26/ns-3.26/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1m21.806s)
50.3174 3 received one packet from 10.1.1.8
50.325 2 received one packet from 10.1.1.7
50.5643 3 received one packet from 10.1.1.8
50.5714 2 received one packet from 10.1.1.7
50.8143 3 received one packet from 10.1.1.8
50.8214 2 received one packet from 10.1.1.7
51.0643 3 received one packet from 10.1.1.8
51.0714 2 received one packet from 10.1.1.7
51.3143 3 received one packet from 10.1.1.8
51.3214 2 received one packet from 10.1.1.7
51.4803 1 received one packet from 10.1.1.6
51.4836 1 received one packet from 10.1.1.6
51.5643 3 received one packet from 10.1.1.8
51.5714 2 received one packet from 10.1.1.7
51.8143 3 received one packet from 10.1.1.8

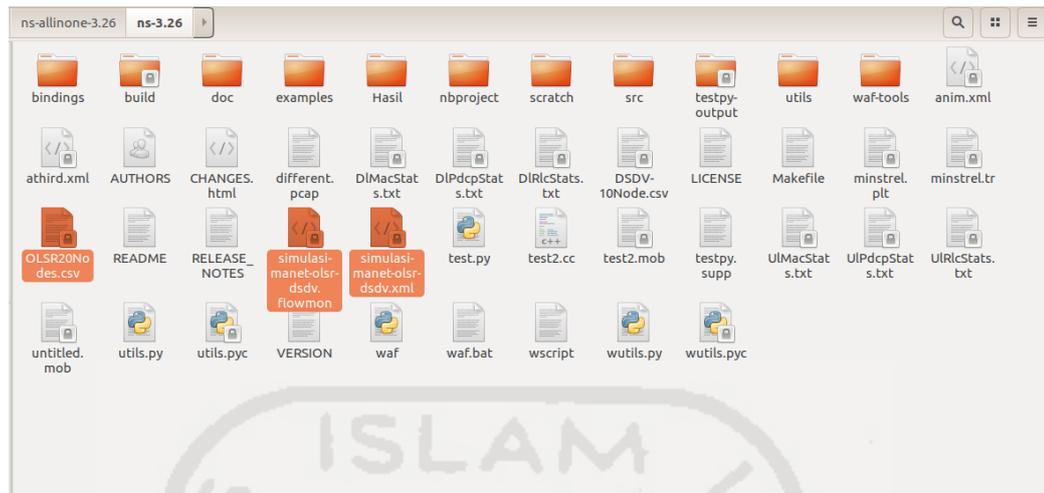
```

Gambar 4.1. Memulai simulasi

- Kemudian simulasi berakhir dan NS3 akan berhenti.

4.1.3. Tahapan Pengambilan Data

Setelah melakukan tahapan-tahapan menjalankan simulasi seperti diatas, NS3 akan menghasilkan output 3 buah *file* yaitu .csv, .flowmon dan .xml seperti pada gambar 4.2. Ketiga *file* tersebut akan digunakan untuk menganalisis data yang akan dijelaskan pada langkah selanjutnya.



Gambar 4.2. Output file

1. File .csv

File ini berisi beberapa informasi, yaitu lamanya waktu simulasi, besar data yang diterima, paket yang diterima, jumlah koneksi, jenis protokol, dan daya transmisi. File .csv ini dibutuhkan saat analisis data untuk menghitung nilai throughput. Untuk menghitung nilai throughput perlu diketahui total paket yang diterima dan lamanya waktu pengiriman paket dihitung dari pertama kali paket dikirim sampai dengan paket yang terakhir diterima. Isi dari file .csv dapat dilihat pada gambar 4.3 dan 4.4.

SimulationSecond	ReceiverRate	PacketsReceived	NumberOfSinks	RoutingProtocol	TransmissionPower
0	0	0	4	OLSR	5
1	0	0	4	OLSR	5
2	0	0	4	OLSR	5
3	0	0	4	OLSR	5
4	0	0	4	OLSR	5
5	0	0	4	OLSR	5
6	0	0	4	OLSR	5
7	0	0	4	OLSR	5
8	0	0	4	OLSR	5
9	0	0	4	OLSR	5
10	0	0	4	OLSR	5
11	0	0	4	OLSR	5
12	0	0	4	OLSR	5
13	0	0	4	OLSR	5
14	0	0	4	OLSR	5
15	0	0	4	OLSR	5
16	0	0	4	OLSR	5
17	0	0	4	OLSR	5
18	0	0	4	OLSR	5
19	0	0	4	OLSR	5
20	0	0	4	OLSR	5
21	0	0	4	OLSR	5
22	0	0	4	OLSR	5
23	0	0	4	OLSR	5
24	0	0	4	OLSR	5

Gambar 4.3. File .csv skenario 20 node OLSR (1)

74	8	16	4 OLSR	5
75	8	16	4 OLSR	5
76	8	16	4 OLSR	5
77	8	16	4 OLSR	5
78	8	16	4 OLSR	5
79	8	16	4 OLSR	5
80	8	16	4 OLSR	5
81	8	16	4 OLSR	5
82	8	16	4 OLSR	5
83	8	16	4 OLSR	5
84	8	16	4 OLSR	5
85	8	16	4 OLSR	5
86	8	16	4 OLSR	5
87	8	16	4 OLSR	5
88	8	16	4 OLSR	5
89	8	16	4 OLSR	5
90	8	16	4 OLSR	5
91	8	16	4 OLSR	5
92	8	16	4 OLSR	5
93	8	16	4 OLSR	5
94	8	16	4 OLSR	5
95	7.5	15	4 OLSR	5
96	8	16	4 OLSR	5
97	8	16	4 OLSR	5
98	8	16	4 OLSR	5
99	8	16	4 OLSR	5

Gambar 4.4. File .csv skenario 20 node OLSR (2)

2. File .flowmon

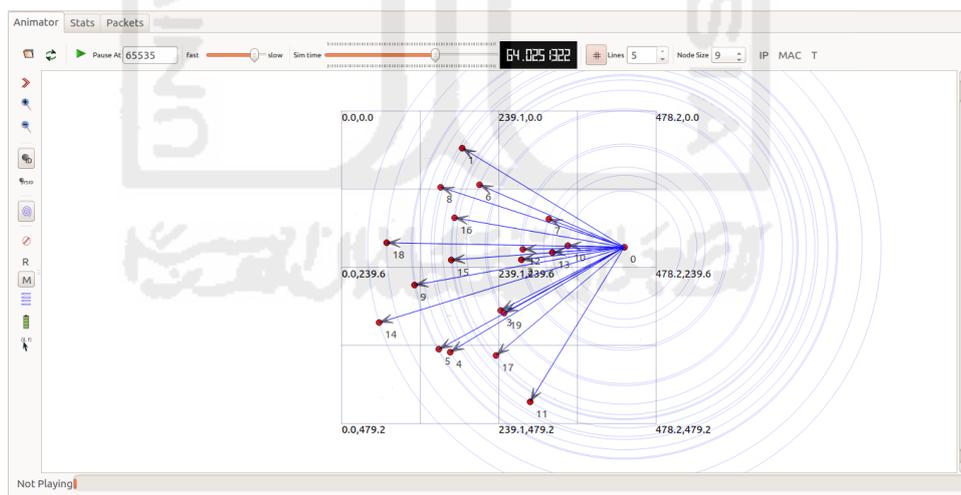
File *FlowMonitor* yang berekstensi .flowmon ini berisi beberapa informasi yang berkaitan dengan nilai-nilai parameter paket data, seperti *delay*, paket yang hilang, paket dikirim dan paket diterima. File .flowmon ini dibutuhkan saat analisis data untuk menghitung PDR (*packet delivery ratio*), *packet loss* dan *delay*. Untuk menghitung PDR dan *packet loss* perlu diketahui paket yang dikirim dapat dilihat pada *txPackets* dan paket yang diterima dapat dilihat pada *rxPackets*. Sedangkan untuk nilai *delay*, pada *FlowMonitor* nilai *delay* secara otomatis telah dihitung, hanya saja perlu menghitung rata-rata delay dari empat koneksi tersebut. File .flowmon ini dapat dibuka menggunakan aplikasi NetAnim. Isi dari file .flowmon dapat dilihat pada gambar 4.5.

Flow Id:1	Flow Id:2	Flow Id:3	Flow Id:4
===== UDP 10.1.1.5/49153---->10.1.1.1/9	===== UDP 10.1.1.8/49153---->10.1.1.4/9	===== UDP 10.1.1.7/49153---->10.1.1.3/9	===== UDP 10.1.1.6/49153---->10.1.1.2/9
Tx bitrate:2.95887kbps Rx bitrate:2.52831kbps Mean delay:1.49905ms Packet Loss ratio:14.5729%	Tx bitrate:2.95887kbps Rx bitrate:2.95907kbps Mean delay:0.815677ms Packet Loss ratio:0%	Tx bitrate:2.95887kbps Rx bitrate:2.95907kbps Mean delay:0.681074ms Packet Loss ratio:0%	Tx bitrate:2.95887kbps Rx bitrate:2.33469kbps Mean delay:5.40655ms Packet Loss ratio:19.4872%
timeFirstTxPacket= 5.02711e+10ns timeFirstRxPacket= 5.02839e+10ns timeLastTxPacket= 9.97711e+10ns timeLastRxPacket= 9.97714e+10ns delaySum= 2.54839e+08ns jitterSum= 1.48064e+08ns lastDelay= 2.54839e+08ns txBytes= 18308 rxBytes= 15640 txPackets= 199 rxPackets= 170 lostPackets= 29 timesForwarded= 60	timeFirstTxPacket= 5.0314e+10ns timeFirstRxPacket= 5.03177e+10ns timeLastTxPacket= 9.9814e+10ns timeLastRxPacket= 9.98143e+10ns delaySum= 1.6232e+08ns jitterSum= 2.07276e+08ns lastDelay= 1.6232e+08ns txBytes= 18308 rxBytes= 18308 txPackets= 199 rxPackets= 199 lostPackets= 0 timesForwarded= 0	timeFirstTxPacket= 5.03211e+10ns timeFirstRxPacket= 5.03248e+10ns timeLastTxPacket= 9.98211e+10ns timeLastRxPacket= 9.98214e+10ns delaySum= 1.35534e+08ns jitterSum= 1.5374e+08ns lastDelay= 1.35534e+08ns txBytes= 18308 rxBytes= 18308 txPackets= 199 rxPackets= 199 lostPackets= 0 timesForwarded= 0	timeFirstTxPacket= 5.04376e+10ns timeFirstRxPacket= 5.04516e+10ns timeLastTxPacket= 9.99376e+10ns timeLastRxPacket= 9.9945e+10ns delaySum= 8.48828e+08ns jitterSum= 1.03828e+09ns lastDelay= 8.48828e+08ns txBytes= 18308 rxBytes= 14444 txPackets= 199 rxPackets= 157 lostPackets= 38 timesForwarded= 0

Gambar 4.5. File .flowmon skenario ukuran paket 64 bytes DSDV

3. File .xml

Simulasi yang sudah dijalankan dapat dilihat prosesnya dalam bentuk animasi, dengan cara membuka *file .xml* menggunakan aplikasi NetAnim. *File .xml* memperlihatkan bagaimana simulasi berjalan mulai dari *broadcast*, proses pengiriman paket data dan pergerakan *node-node*. Animasi dapat dilihat pada gambar 4.6.



Gambar 4.6. Animasi simulasi

4.2. Analisis Data

Pada tahapan analisis data ini akan melakukan perbandingan kinerja routing protokol OLSR dan DSDV setelah melakukan perhitungan parameter *QoS*. Analisis

data dilakukan dengan menghitung nilai-nilai yang terdapat di *file* yang dihasilkan dari simulasi, yaitu *file* .csv dan .flowmon.

Pertama, untuk menghitung *throughput* dengan membuka file .csv, kemudian hitung total dari *recevie rate* dan selisih waktu pertama kali paket dikirim hingga terakhir kali paket diterima. Nilai *throughput* didapat setelah membagi total *recevie rate* dengan waktu pengiriman paket.

Selanjutnya, untuk menghitung PDR, *packet loss* dan *delay* dengan membuka file .flowmon. Untuk menghitung PDR, membagi *rxPackets* dengan *txPackets*, kemudian dirubah dalam bentuk persen dengan cara dikalikan dengan 100%. Untuk menghitung *packet loss*, *txPackets* dikurang *rxPackets* kemudian hasilnya dibagi *txPackets*, kemudian dirubah dalam bentuk persen dengan cara dikalikan dengan 100%. Untuk menghitung *delay* dengan melihat *Mean delay* kemudian dijumlah *Mean delay* di setiap koneksi dan dibagi dengan jumlah koneksi.

4.2.1. Hasil Simulasi Untuk Skenario Penambahan *Node*

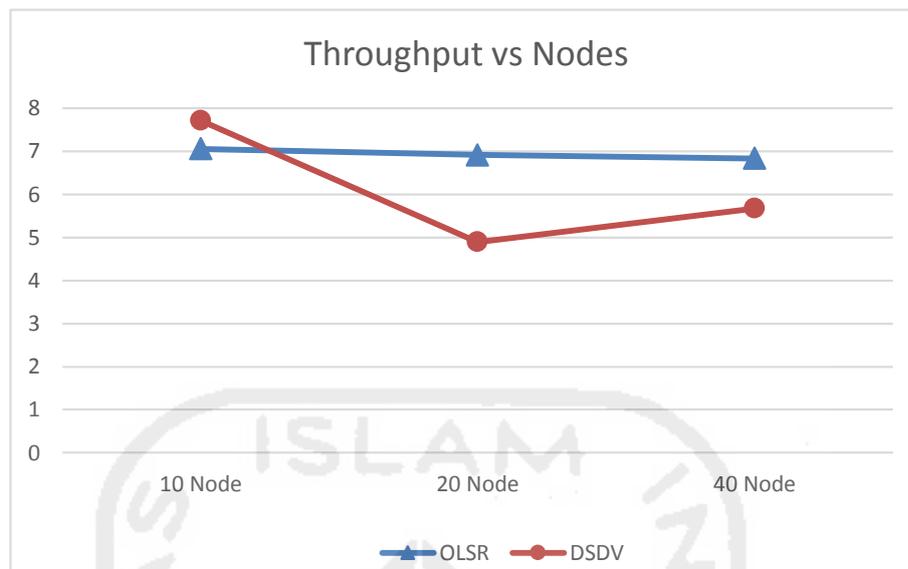
Setelah hasil simulasi skenario penambahan *node* selesai kemudian akan dianalisis yang akan dibagi dalam 4 bagian sesuai dengan parameter *QoS* yang diuji.

4.2.1.1. *Throughput*

Hasil dari perhitungan rata-rata *throughput* pada perbandingan performa routing protokol OLSR dan DSDV yang didapat dari simulasi untuk skenario penambahan *node* yang telah dijalankan, hasil tersebut dapat di lihat pada tabel 4.1 dan gambar 4.7. nilai rata-rata *throughput* ini menggunakan satuan Kbps (*Kilo bit per second*).

Tabel 4.1. Nilai rata-rata *throughput* pada skenario penambahan *node*

Protokol	10 Node	20 Node	40 Node	AVG
OLSR	7,061224489	6,918367346	6,836734693	6,938775510
DSDV	7,714285714	4,897959183	5,673469387	6,095238095



Gambar 4.7. Grafik *throughput vs nodes*

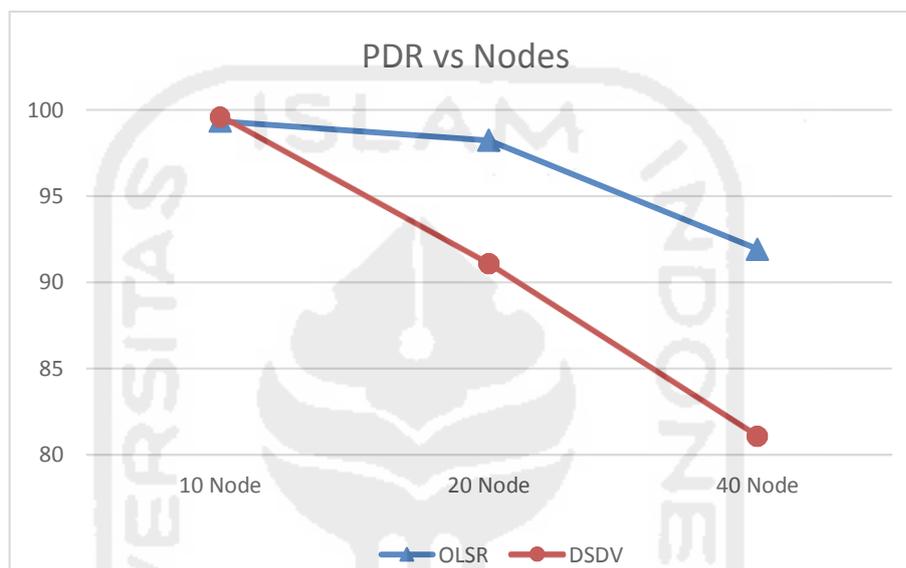
Berdasarkan grafik pada gambar 4.7, dapat dilihat routing protokol OLSR lebih baik dibandingkan dengan DSDV dari nilai performa rata-rata throughput terhadap penambahan node. OLSR menunjukkan nilai rata-rata 6,938775510 sedangkan DSDV menunjukkan nilai rata-rata 6,095238095. Hal ini dikarenakan DSDV menyimpan seluruh informasi ke semua node dan mem-*broadcast* secara berkala untuk memperbarui tabel routing. *Broadcast* yang dilakukan DSDV ini mengakibatkan penggunaan *bandwidth* yang berlebih, yang seharusnya dapat digunakan untuk mengirim paket data. Sedangkan OLSR menggunakan metode MPR dalam memperbarui tabel routing. Jadi dalam OLSR hanya *node* MPR saja yang mengirimkan pesan *broadcast*. Hal tersebut dapat mengurangi penggunaan *bandwidth* yang berlebihan.

4.2.1.2. *Packet Delivery Ratio*

Hasil dari perhitungan rata-rata PDR pada perbandingan performa routing protokol OLSR dan DSDV yang didapat dari simulasi untuk skenario penambahan *node* yang telah dijalankan, hasil tersebut dapat di lihat pada tabel 4.2 dan gambar 4.8. nilai rata-rata PDR ini menggunakan satuan persentase.

Tabel 4.2. Nilai rata-rata PDR pada skenario penambahan *node*

Protokol	10 Node	20 Node	40 Node	AVG
OLSR	99,36%	98,24%	91,92%	96,51%
DSDV	99,62%	91,08%	81,08%	90,59%

**Gambar 4.8.** Grafik PDR vs *nodes*

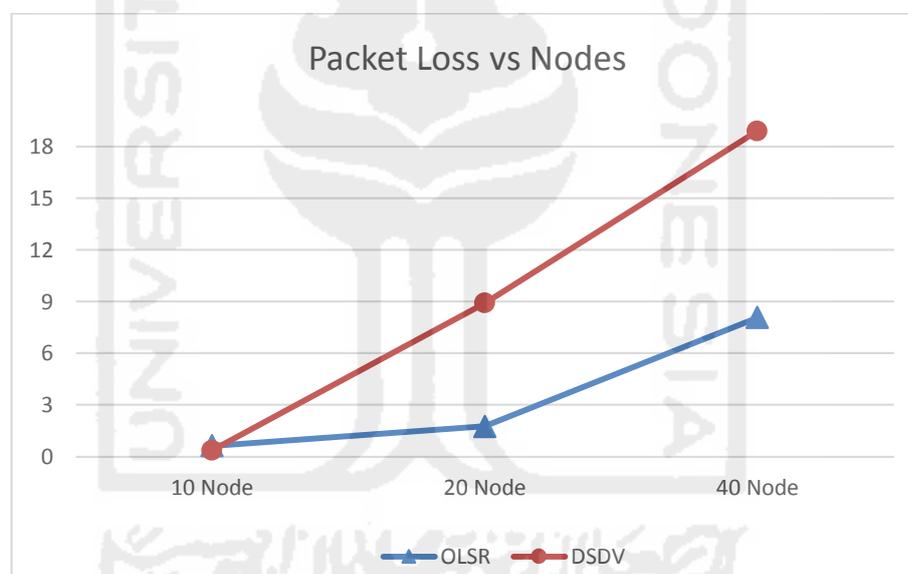
Berdasarkan grafik pada gambar 4.8, dapat dilihat routing protokol OLSR lebih baik dibandingkan dengan DSDV dari nilai performa rata-rata PDR terhadap penambahan node. OLSR menunjukkan nilai rata-rata 96,51% sedangkan DSDV menunjukkan nilai rata-rata 90,59%. Hal ini disebabkan kepadatan *node* yang menyebabkan terbentuknya banyak *hop* dan juga memungkinkan terjadinya perubahan jalur. Pada OLSR hal tersebut dapat teratasi karena memiliki sistem MPR. Sedangkan pada DSDV yang harus memperbarui tabel routing secara berkala. Hal tersebut menyebabkan ketika mengirim paket dan kehilangan jalur maka pengiriman paket harus menunggu hingga proses pembaruan routing tabel selesai untuk menentukan jalur pengiriman.

4.2.1.3. Packet Loss

Hasil dari perhitungan rata-rata *packet loss* pada perbandingan performa routing protokol OLSR dan DSDV yang didapat dari simulasi untuk skenario penambahan *node* yang telah dijalankan, hasil tersebut dapat di lihat pada tabel 4.3 dan gambar 4.9. nilai rata-rata *packet loss* ini menggunakan satuan persentase.

Tabel 4.3. Nilai rata-rata *packet loss* pada skenario penambahan *node*

Protokol	10 Node	20 Node	40 Node	AVG
OLSR	0,63%	1,75%	8,07%	3,48%
DSDV	0,37%	8,91%	18,91%	9,40%



Gambar 4.9. Grafik *packet loss* vs nodes

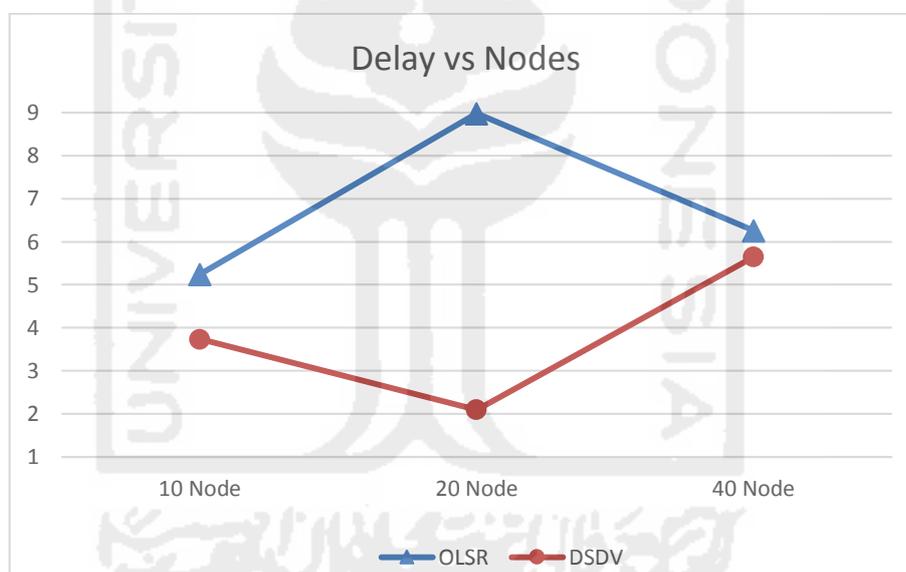
Berdasarkan grafik pada gambar 4.9, dapat dilihat routing protokol OLSR lebih baik dibandingkan dengan DSDV dari nilai performa rata-rata *packet loss* terhadap penambahan *node*. OLSR menunjukkan nilai rata-rata 3,48% sedangkan DSDV menunjukkan nilai rata-rata 9,40%. Sama halnya dengan PDR, hal ini disebabkan kepadatan *node* yang besar, memungkinkan hilangnya paket data saat pengiriman.

4.2.1.4. Delay

Hasil dari perhitungan rata-rata *delay* pada perbandingan performa routing protokol OLSR dan DSDV yang didapat dari simulasi untuk skenario penambahan *node* yang telah dijalankan, hasil tersebut dapat di lihat pada tabel 4.4 dan gambar 4.10. nilai rata-rata *delay* ini menggunakan satuan *milliseconds*.

Tabel 4.4. Nilai rata-rata *delay* pada skenario penambahan *node*

Protokol	10 Node	20 Node	40 Node	AVG
OLSR	5,2439447	8,9745405	6,2553335	6,8246062
DSDV	3,7331015	2,1005877	5,6479125	3,8272005



Gambar 4.10. Grafik *delay vs nodes*

Berdasarkan grafik pada gambar 4.10, dapat dilihat routing protokol DSDV lebih baik dibandingkan dengan OLSR dari nilai performa rata-rata *delay* terhadap penambahan *node*. OLSR menunjukkan nilai rata-rata 6,8246062 sedangkan DSDV menunjukkan nilai rata-rata 3,8272005. Hal ini disebabkan sifat DSDV yang selalu melakukan pembaruan tabel routing secara berkala. Sehingga DSDV dapat dengan cepat menentukan jalur pengiriman, karena sudah terdapat pada tabel routing.

Melihat performa routing protokol berdasarkan nilai *QoS* terhadap skenario penambahan *node*. Diketahui bahwa routing protokol OLSR lebih baik pada nilai rata-rata *throughput*, PDR dan *packet loss* dibandingkan dengan DSDV. Sedangkan routing protokol DSDV lebih baik pada nilai rata-rata *delay* dibandingkan dengan OLSR. Diketahui juga bahwa nilai *throughput*, PDR dan *packet loss* DSDV pada skenario penambahan *node* dengan jumlah *node* yang sedikit, DSDV lebih baik dibandingkan OLSR. DSDV lebih cocok digunakan untuk kepadatan jaringan berskala kecil (Dabungke, Wahidah, & Mulyana, 2009). Sedangkan OLSR cocok digunakan untuk kepadatan jaringan berskala menengah ke atas. Hal tersebut disebabkan sistem MPR yang dimiliki OLSR, yang membuat proses pengiriman paket data lebih cepat.

4.2.2. Hasil Simulasi Untuk Skenario Penambahan Ukuran Paket Data

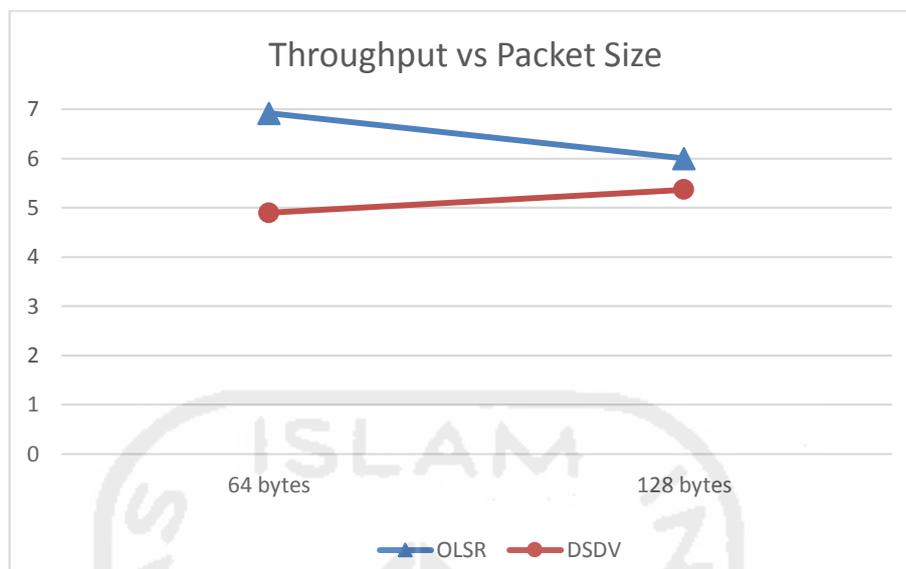
Setelah hasil simulasi skenario penambahan ukuran paket data selesai kemudian akan dianalisis yang akan dibagi dalam 4 bagian sesuai dengan parameter *QoS* yang diuji.

4.2.2.1. *Throughput*

Hasil dari perhitungan rata-rata *throughput* pada perbandingan performa routing protokol OLSR dan DSDV yang didapat dari simulasi untuk skenario penambahan ukuran paket data yang telah dijalankan, hasil tersebut dapat di lihat pada tabel 4.5 dan gambar 4.11. nilai rata-rata *throughput* ini menggunakan satuan Kbps (*Kilo bit per second*).

Tabel 4.5. Nilai rata-rata *throughput* pada skenario penambahan ukuran paket

Protokol	64 bytes	128 bytes	AVG
OLSR	6,918367346	6	6,459183673
DSDV	4,897959183	5,367346938	5,132653061



Gambar 4.11. Grafik *throughput vs packet size*

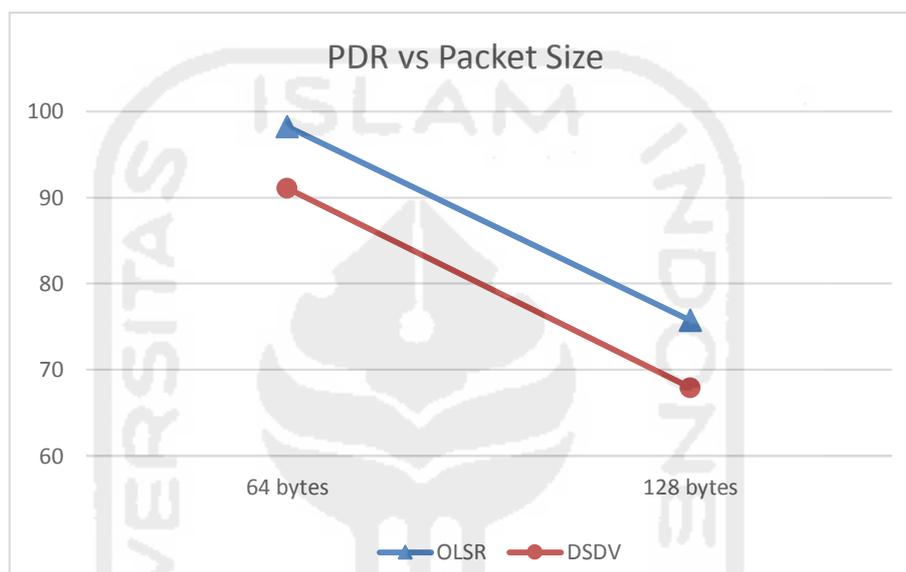
Berdasarkan grafik pada gambar 4.11, dapat dilihat routing protokol OLSR lebih baik dibandingkan dengan DSDV dari nilai performa rata-rata *throughput* terhadap penambahan ukuran paket data. OLSR menunjukkan nilai rata-rata 6,459183673 sedangkan DSDV menunjukkan nilai rata-rata 5,132653061. Hal ini dikarenakan DSDV menyimpan seluruh informasi ke semua *node* dan mem-*broadcast* secara berkala untuk memperbarui tabel routing. *Broadcast* yang dilakukan DSDV ini mengakibatkan penggunaan *bandwidth* yang berlebih, yang seharusnya dapat digunakan untuk mengirim paket data, terlebih lagi ukuran paket yang bertambah. Sedangkan OLSR menggunakan metode MPR dalam memperbarui tabel routing. Jadi dalam OLSR hanya node MPR saja yang mengirimkan pesan *broadcast*. Hal tersebut dapat mengurangi penggunaan *bandwidth* yang berlebihan.

4.2.2.2. *Packet Delivery Ratio*

Hasil dari perhitungan rata-rata PDR pada perbandingan performa routing protokol OLSR dan DSDV yang didapat dari simulasi untuk skenario penambahan ukuran paket data yang telah dijalankan, hasil tersebut dapat di lihat pada tabel 4.6 dan gambar 4.12. Nilai rata-rata PDR ini menggunakan satuan persentase.

Tabel 4.6. Nilai rata-rata PDR pada skenario penambahan ukuran paket

Protokol	64 bytes	128 bytes	AVG
OLSR	98,24%	75,75%	86,99%
DSDV	91,08%	67,92%	79,50%

**Gambar 4.12.** Grafik PDR vs *packet size*

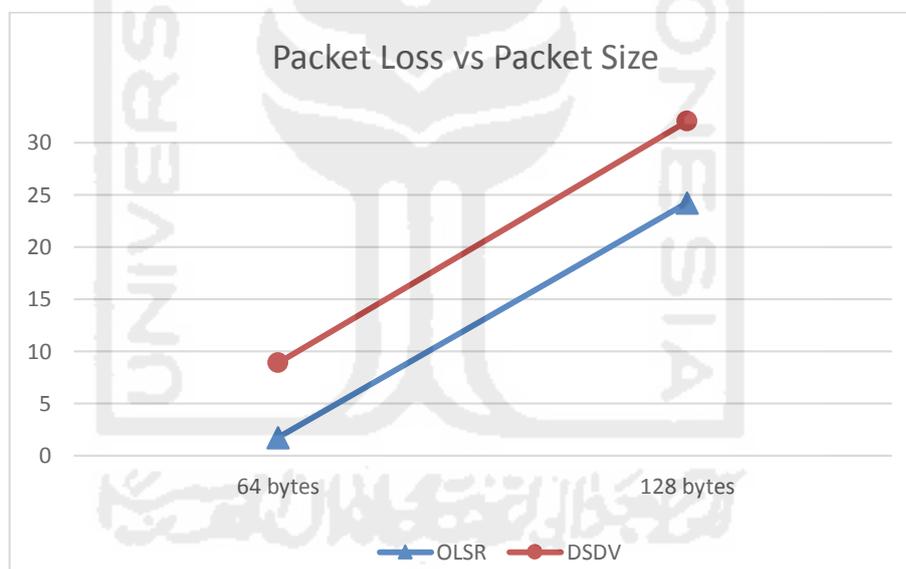
Berdasarkan grafik pada gambar 4.12, dapat dilihat routing protokol OLSR lebih baik dibandingkan dengan DSDV dari nilai performa rata-rata PDR terhadap penambahan ukuran paket data. OLSR menunjukkan nilai rata-rata 86,99% sedangkan DSDV menunjukkan nilai rata-rata 79,50%. Hal ini disebabkan ukuran paket data yang besar berpengaruh pada pengiriman paket. Pada OLSR hal tersebut dapat teratasi karena memiliki sistem MPR sehingga beban paket hanya akan disebar pada *node* MPR saja. Sedangkan pada DSDV yang harus memperbarui tabel routing secara berkala. Hal tersebut menyebabkan ketika mengirim paket dan kehilangan jalur maka pengiriman paket harus menunggu hingga proses pembaruan routing tabel selesai untuk menentukan jalur pengiriman.

4.2.2.3. Packet Loss

Hasil dari perhitungan rata-rata *packet loss* pada perbandingan performa routing protokol OLSR dan DSDV yang didapat dari simulasi untuk skenario penambahan ukuran paket data yang telah dijalankan, hasil tersebut dapat di lihat pada tabel 4.7 dan gambar 4.13. nilai rata-rata *packet loss* ini menggunakan satuan persentase.

Tabel 4.7. Nilai rata-rata *packet loss* pada skenario penambahan ukuran paket

Protokol	64 bytes	128 bytes	AVG
OLSR	1,75%	24,24%	13,07%
DSDV	8,91%	32,07%	20,49%



Gambar 4.13. Grafik *packet loss vs packet size*

Berdasarkan grafik pada gambar 4.13, dapat dilihat routing protokol OLSR lebih baik dibandingkan dengan DSDV dari nilai performa rata-rata *packet loss* terhadap penambahan ukuran paket data. OLSR menunjukkan nilai rata-rata 13,07% sedangkan DSDV menunjukkan nilai rata-rata 20,49%. Sama halnya dengan PDR, hal ini disebabkan ukuran paket yang besar, memungkinkan

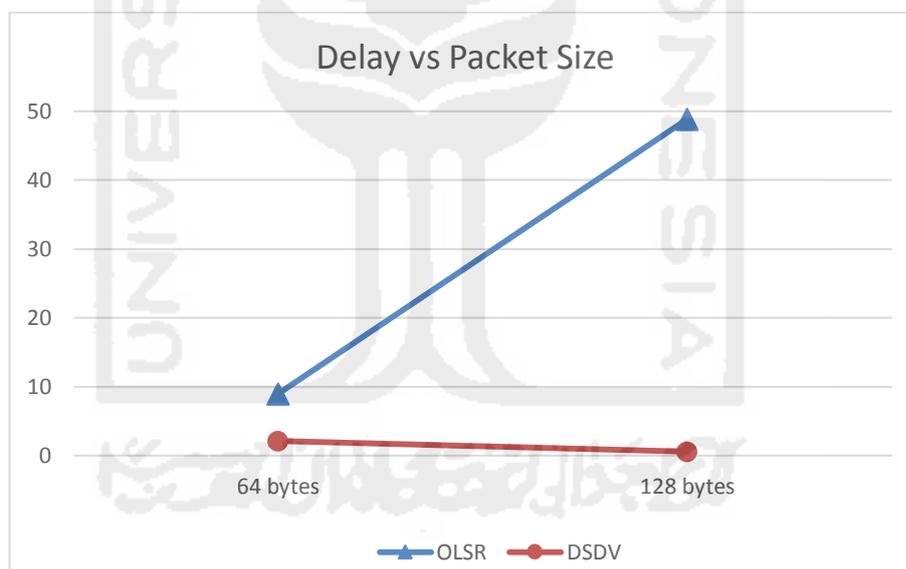
hilangnya paket data saat pengiriman.

4.2.2.4. Delay

Hasil dari perhitungan rata-rata *delay* pada perbandingan performa routing protokol OLSR dan DSDV yang didapat dari simulasi untuk skenario penambahan node yang telah dijalankan, hasil tersebut dapat di lihat pada tabel 4.8 dan gambar 4.14. nilai rata-rata *delay* ini menggunakan satuan *milliseconds*.

Tabel 4.8. Nilai rata-rata *delay* pada skenario penambahan ukuran paket

Protokol	64 bytes	128 bytes	AVG
OLSR	8,974540	48,840675	28,907607
DSDV	2,100585	0,555727	1,328156



Gambar 4.14. Grafik *delay vs packet size*

Berdasarkan grafik pada gambar 4.14, dapat dilihat routing protokol DSDV lebih baik dibandingkan dengan OLSR dari nilai performa rata-rata *delay* terhadap penambahan ukuran paket data. OLSR menunjukkan nilai rata-rata 28,907607 sedangkan DSDV menunjukkan nilai rata-rata 3,8272005. Hal ini disebabkan sifat DSDV yang selalu melakukan pembaruan tabel routing secara berkala. Sehingga

DSDV dapat dengan cepat menentukan jalur pengiriman, karena sudah terdapat pada tabel routing.

Melihat performa routing protokol berdasarkan nilai *QoS* terhadap skenario penambahan ukuran paket data. Diketahui bahwa routing protokol OLSR lebih baik pada nilai rata-rata *throughput*, PDR dan *packet loss* dibandingkan dengan DSDV. Sedangkan routing protokol DSDV lebih baik pada nilai rata-rata *delay* dibandingkan dengan OLSR.



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah menyelesaikan beberapa tahapan pada penelitian ini, maka ditarik kesimpulan sebagai berikut:

1. Simulasi MANET untuk membandingkan performa dari routing protokol OLSR dan DSDV berdasarkan *QoS* dapat diterapkan pada NS3 dengan baik dan dapat dijadikan referensi untuk membuat jaringan yang sebenarnya.
2. Berdasarkan hasil analisis perbandingan performansi routing protokol OLSR dan DSDV dengan skenario yang ditentukan pada simulasi MANET berdasarkan parameter *QoS* yaitu *throughput*, *packet delivery ratio*, *packet loss* dan *delay*, didapatkan beberapa kesimpulan, sebagai berikut :
 - a. Dilihat dari skenario penambahan node berdasarkan nilai parameter *throughput*, *packet delivery ratio* dan *packet loss* OLSR lebih baik dari DSDV. Sedangkan DSDV lebih baik dari OLSR pada nilai parameter *delay*. Kemudian, dilihat dari skenario penambahan ukuran paket data berdasarkan nilai parameter *throughput*, *packet delivery ratio* dan *packet loss* OLSR lebih baik dari DSDV. Sama halnya pada skenario penambahan node, DSDV lebih baik dari OLSR pada nilai parameter *delay*.
 - b. Diketahui juga bahwa DSDV cocok digunakan pada kepadatan jaringan yang kecil. Hal tersebut dapat dibuktikan dengan nilai parameter *throughput*, *packet delivery ratio*, *packet loss* dan *delay* pada skenario penambahan node dengan jumlah *node* 10, DSDV lebih baik dibandingkan OLSR. Sedangkan pada jumlah *node* 20 dan 40 OLSR lebih baik dari DSDV, kecuali pada nilai parameter *delay*. Hal tersebut menunjukkan juga OLSR cocok digunakan pada kepadatan jaringan menengah ke atas.
 - c. Melihat nilai parameter *delay* DSDV pada semua skenario lebih baik dari OLSR. Hal tersebut menunjukkan juga DSDV cocok digunakan pada kondisi dimana membutuhkan kecepatan akses data dengan kepadatan

jaringan yang kecil.

5.2. Saran

Dari hasil penelitian yang telah diselesaikan dan dalam rangka untuk pengembangan penelitian pada bidang yang sama, maka penulis ingin membuat saran untuk penelitian selanjutnya, yaitu:

1. Membuat koneksi yang bersifat statis di semua skenario, yang artinya pasangan *node* yang berkomunikasi ditentukan kita sendiri, tidak secara random ditentukan aplikasi. Dengan begitu akan menghasilkan perbandingan yang lebih baik karena *node* yang berkomunikasi akan sama di semua skenario.
2. Menggunakan *stats module/data collector* pada NS3 untuk menghitung parameter *QoS*. Karena *stats module/data collector* memiliki informasi parameter *QoS* yang lebih detail.
3. Dilakukan penilaian parameter *QoS* yang berbeda untuk membandingkan performa dari routing protokol, seperti *jitter*, *routing overhead* dan lain-lain.

DAFTAR PUSTAKA

- Al Fatta, H. (2007). *Analisis Dan Perancangan Sistem Informasi*. Yogyakarta: Andi.
- Dabungke, B., Wahidah, I., & Mulyana, A. (2009). Evaluasi Performansi Protokol Ruting Dsdv Dan Dsr Pada Jaringan Wireless Mobile Ad Hoc Network (Manet).
- globalspec.com. (2008). Chapter 12_ Mobile Ad Hoc Networks _ Engineering360. Retrieved from <http://www.globalspec.com/reference/81338/203279/chapter-12-mobile-ad-hoc-networks>
- He, G. (2002). Destination-sequenced distance vector (DSDV) protocol. *Networking Laboratory, Helsinki University of Technology*, 1–9. <https://doi.org/10.1145/190314.190336>
- ICT. (2005). Module 20: Static Routing & Dynamic Routing. In *Network Tech Support*.
- id.wikipedia.org. (n.d.). C++ - Wikipedia bahasa Indonesia, ensiklopedia bebas. Retrieved from <https://id.wikipedia.org/wiki/C%2B%2B>
- id.wikipedia.org. (2016). Jaringan nirkabel - Wikipedia bahasa Indonesia, ensiklopedia bebas. Retrieved from https://id.wikipedia.org/wiki/Jaringan_nirkabel
- Irfan, M. D., Perdana, D., & Ridha, M. N. (2016). Analisis Perbandingan Kinerja Protokol Routing DSDV dan OLSR Untuk Perubahan Kecepatan Mobilitas pada Standar IEEE 802.11 ah.
- Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., & Viennot, L. (2001). Optimized link state routing protocol for ad hoc networks. *Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century.*, 62–68. <https://doi.org/10.1109/INMIC.2001.995315>
- Jiatmiko, N., & Prayudi, Y. (2016). Simulasi Jaringan MANET Dengan NS3 Untuk Membandingkan Performa Routing Protokol AODV dan DSDV,

(November 2015).

Kopp, C. (1999). Ad Hoc networking. *NCW 101 - An Introduction to Network Centric Warfare*, 60–63. Retrieved from <http://www.ausairpower.net/SP/NCW-101-4.pdf>

Mapa, F., Djanali, S., & Shiddiqi, A. M. (2014). Optimasi Olsr Routing Protocol Pada Jaringan Wireless Mesh Dengan Adaptive Refreshing Time Interval Dan Enhance Multi Point Relay Selecting Algorithm. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 12(1), 44. <https://doi.org/10.12962/j24068535.v12i1.a46>

Mulyanta, E. S. (2005). *Pengenalan Protokol Jaringan Wireless Komputer*. Yogyakarta: Andi.

nsnam.org. (2011). Documentation « ns-3. Retrieved from <https://www.nsnam.org/documentation/>

Sidharta, Y., & Widjaja, D. (2013). Perbandingan Unjuk Kerja Protokol Routing Ad Hoc on-Demand Distance Vector (Aodv) Dan Dynamic Source Routing (Dsr) Pada Jaringan Manet.