

**ANALISIS *PERFORMANSI* PROTOKOL *ROUTING EPIDEMIC*
DAN *SPRAY AND WAIT* PADA DTN BERDASARKAN
MOBILITAS NODE**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Jodi Prayoga Wahyudwi
NIM: 125150207111048



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

PENGESAHAN

ANALISIS PERFORMANSI PROTOKOL ROUTING EPIDEMIC DAN SPRAY AND WAIT PADA DTN BERDASARKAN MOBILITAS NODE

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Jodi Prayoga Wahyudwi

NIM: 125150207111048

Skripsi ini telah diuji dan dinyatakan lulus pada
15 Januari 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Rakhmadhany Primananda, S.T, M.Kom
NIK: 201609860406 1001

Achmad Basuki, S.T., M.MG., Ph.D
NIP: 19741118 200312 1 002

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP: 197105182003121001

PERNYATAAN ORISINALITAS

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

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

Malang, 4 Januari 2018



Jodi Prayoga Wahyudwi

NIM: 125150207111048

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan kasih dan sayang-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “ANALISIS *PERFORMANSI* PROTOKOL *ROUTING EPIDEMIC* DAN *SPRAY AND WAIT* PADA DTN BERDASARKAN MOBILITAS NODE”

Maksud dari penyusunan skripsi ini adalah untuk memenuhi salah satu syarat dalam memperoleh gelar Sarjana Komputer (S.Kom) pada Jurusan Teknik Informatika, Fakultas Ilmu Komputer , Universitas Brawijaya.

Dalam penyusunan skripsi ini, banyak pihak yang sangat membantu penulis dalam berbagai hal. Oleh karena itu, penulis sampaikan rasa terima kasih yang sedalam-dalamnya kepada :

1. Rakhmadhany Primananda, S.T., M.Kom., selaku dosen pembimbing 1, yang telah bersedia membimbing dan memberi arahan dan motivasi sehingga skripsi ini dapat terselesaikan.
2. Achmad Basuki, S.T., M.MG., Ph.D., selaku dosen pembimbing 2, yang telah memberikan saran dan kritik dalam proses penulisan dan penyusunan skripsi ini.
3. Wayan Firdaus Mahmudy, S.Si., M.T, Ph.D., selaku Dekan Fakultas Ilmu Komputer, Universitas Brawija.
4. Agus Wahyu Widodo, S.Kom., M.Cs., selaku Ketua Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya.
5. Bambang Sudjarwo sebagai ayahanda dan Budi Sri Lestari sebagai ibunda yang merupakan Orang tua tercinta yang telah sangat banyak memberikan doa dan dukungannya kepada penulis baik secara moril maupun materil sehingga skripsi ini dapat diselesaikan penulis.
6. Aliffazrie Wahyutama Yosto sebagai Kakak dan Intaniar Wahyu Trivani sebagai adik tercinta serta keluarga dan kerabat yang senantiasa memberikan doa serta dukungan semangat kepada penulis.
7. Seluruh dosen dan karyawan Fakultas Ilmu Komputer Universitas Brawijaya, atas arahan dan bantuannya sehingga penulis dapat menyelesaikan skripsi ini.
8. Randi Pratama, Kikit Maulana, Fahmi Arief Hidayat, M fadhil Hawari, Boni Saputra, Afdin F, Genjah Amарtha Gora dan semua teman yang telah bersedia memberi bantuan.

9. Sahabat serta rekan-rekan seperjuangan seangkatan jurusan teknik informatika/Illmu computer Fakultas Ilmu Komputer Universitas Brawijaya
10. Semua pihak yang telah banyak membantu dalam penyusunan skripsi ini.

Penulis menyadari didalam penulisan skripsi ini masih terdapat banyak ketidaksesuaian. Penulis masih sadar bahwa masih banyak kesalahan dalam skripsi ini dan kekurangan. Penulis berharap, topik skripsi yang akan dibahas ini dapat menambah ilmu dan wawasan bagi para pembaca dan khususnya bagi mahasiswa Fakultas Ilmu Komputer, Universitas Brawijaya.

Malang, 4 Januari 2018

Penulis

joghamlg@live.com

ABSTRAK

DTN atau *Delay Tolerant Network* adalah jaringan internet yang tidak mempermasalahkan waktu tunda. Pada penelitian ini, analisis *routing* DTN menggunakan *Epidemic* dan *Spray And Wait* berdasarkan mobilitas node. Pada *routing* DTN menggunakan aplikasi *The One simulator* untuk merancang jalur rute dengan kondisi kota yang sebenarnya. Dalam penilaian performa, digunakan tiga parameter uji yaitu *Delivery Probability*, *Overhead Ratio*, *Average Latency*. Pada skenario pertama, *Delivery Probability* tertinggi pada Mobilitas *Open Shortest Path* sebesar 0.9853 pesan, peringkat kedua *Map Based Movement* sebesar 0.9412 pesan, dan terakhir *Map Route Movement* sebesar 0.9265 pesan. Pada skenario kedua, *Delivery Probability* tertinggi pada Mobilitas *Map Based Movement* sebesar 1 pesan, peringkat kedua *Open Shortest Path* dan *Map Route Movement* sebesar dengan nilai yang sama yaitu sebesar 0.9796 pesan. Pada skenario ketiga *Delivery Probability* tertinggi pada Mobilitas *Map Route Movement* sebesar 0.9457 pesan, peringkat kedua *Map Based Movement* sebesar 0.9225 pesan, dan terakhir *Open Shortest Path* sebesar 0.8837 pesan. Pada *Overhead Ratio*, *routing Epidemic* pada semua mobilitas memiliki *Overhead Ratio* yang tinggi sebesar 6.0938, 6.0625, dan 6.9328 pesan dibandingkan dengan *Overhead Ratio* pada *routing Spray And Wait* pada semua Mobilitas sebesar 3.8408, 3.8129, dan 4.3099 pesan. *Average Latency* terendah didapatkan pada Mobilitas *Open Shortest Path* sebesar 72.9538 ms.

Kata kunci: *Delay Tolerant Network*, *Routing Epidemic*, *Routing Spray & Wait*, *The ONE simulator*.

ABSTRACT

DTN or Delay Tolerant Network is an internet network that does not matter the time delay. In this study, DTN routing analysis using Epidemic and Spray And Wait based on node mobility. In DTN routing use The One simulator application to list route path with actual city condition. In the performance appraisal, three test parameters were used: Probability of Submission, Overhead Ratio, Average Latency. In the first scenario, the highest Delivery Probability on the Open Shortest Path Mobility is 0.9853 messages, the second ranked Map by Movement of 0.9412 messages, and the last Route Movement Map of 0.9265 messages. In the second scenario, High Possibility on Mobility of Movement Based Map by 1 message, second rank of Open Shortest Path and Movement Map Map with the same value is 0.9796 message. On the third map of the highest Delivery Probability on Mobility Route Map Movement for 0.9457 messages, the second ranked Map Based Movement of 0.9225 messages, and last Open Shortest Path of 0.8837 messages. In Overhead Ratio, routing Epidemics in all mobility have. Average Lowest Latency on Open Shortest Path Mobility based on 72.9538 ms.

Keywords: Delay Tolerant Network, Routing Epidemic, Routing Spray & Wait, The ONE simulator.

DAFTAR ISI

PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	4
1.4 Manfaat.....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Pembahasan	5
BAB 2 LANDASAN KEPUSTAKAAN	7
2.1 Kajian Pustaka	7
2.2 Dasar Teori.....	12
2.2.1 <i>Delay Tolerant Network (DTN)</i>	12
2.2.1.1 Bundle Layer.....	16
2.2.1.2 Metode Store-and-Forward	17
2.2.1.3 Custody Transfer	17
2.2.2 <i>Node Statis dan Dinamis</i>	18

2.2.3 <i>Routing Protokol Epidemic</i>	18
2.2.3.1 Mekanisme Pengiriman Pesan Pada Protokol Routing Epidemic .	19
2.2.4 <i>Routing Protokol Spray And Wait</i>	20
2.2.5 <i>THE ONE SIMULATOR</i>	22
2.2.6 <i>OpenJUMP</i>	25
2.2.7 Pengujian.....	26
BAB 3 METODOLOGI	28
3.1 Studi Literatur	28
3.2 Perancangan	29
3.2.1 Analisis Kebutuhan.....	29
3.2.2 Perancangan Sistem	29
3.2.2.1 Rancangan Skenario Sistem	30
3.3 Pengujian	30
3.4 Pengumpulan dan Pengambilan Data	31
3.5 Analisis	31
3.6 Kesimpulan.....	31
BAB 4 Perancangan Sistem	32
4.1 Topologi Arsitektur Jaringan.....	32
4.1.1 Pengambilan Data Map Menggunakan <i>OpenJUMP</i>	32
4.1.2 Langkah-Langkah Konfigurasi Aplikasi The One Simulator	36
4.2 Skenario Pengujian	39
4.2.1 Konfigurasi <i>default settings</i>	41
4.3 Pengujian Skenario	43
4.3.1 <i>Map Based Movement</i>	43
4.3.1.1 Pengujian Skenario Pertama	43
4.3.1.2 Pengujian Skenario kedua	45

4.3.1.3 Pengujian Skenario ketiga	47
4.3.2 <i>Open Shortest Path</i>	49
4.3.2.1 Pengujian Skenario Pertama	49
4.3.2.2 Pengujian Skenario kedua	51
4.3.2.3 Pengujian Skenario ketiga	53
4.3.3 <i>Map Route Movement</i>	55
4.3.3.1 Pengujian Skenario Pertama	55
4.3.3.2 Pengujian Skenario kedua	57
4.3.3.3 Pengujian Skenario ketiga	59
BAB 5 PEMBAHASAN.....	62
5.1 Analisis Data.....	62
5.2 Hasil Pengujian.....	62
5.2.1 Delivery Probability	62
5.2.1.1 Map Based Movement.....	62
5.2.1.2 Open Shortest Path.....	64
5.2.1.3 Map Route Movement.....	65
5.2.1.4 Perbandingan Delivery Probability terhadap tipe Mobilitas	67
5.2.1.5 Pengaruh penggunaan tipe mobilitas untuk Delivery Probability pada Protokol Routing:.....	70
5.2.2 <i>Overhead Ratio</i>	71
5.2.2.1 Map Based Movement	71
5.2.2.2 Open Shortest Path	72
5.2.2.3 Map Route Movement	74
5.2.2.4 Perbandingan Overhead Ratio terhadap tipe Mobilitas ..	75
5.2.2.5 Pengaruh penggunaan tipe mobilitas untuk Overhead Ratio pada Protokol Routing:.....	78
5.2.3 <i>Average Latency</i>	79

5.2.3.1 Map Based Movement	79
5.2.3.2 Open Shortest Path	80
5.2.3.3 Map Route Movement	82
5.2.3.4 Perbandingan Average Latency terhadap tipe Mobilitas..	83
5.2.3.5 Pengaruh penggunaan tipe mobilitas untuk Average Latency pada Protokol Routing:	86
5.2.4 Analisis Tiap Skenario.....	87
5.2.4.1 Skenario 1	87
5.2.4.2 Skenario 2	91
5.2.4.3 Skenario 3	95
BAB 6 Penutup	99
6.1 Kesimpulan.....	99
6.2 Saran	101
DAFTAR PUSTAKA.....	102
DAFTAR LAMPIRAN	104

DAFTAR TABEL

Tabel 2.1 Perbedaan Penelitian Sebelumnya Berhubungan Dengan Usulan Topik Skripsi.....	7
Tabel 4.1 Skenario Pengujian Simulasi Pertama.....	40
Tabel 4.2 Skenario Pengujian Simulasi Kedua.....	40
Tabel 4.3 Skenario Pengujian Simulasi Ketiga.....	40
Tabel 4.4 Skenario Pengujian Pertama.....	43
Tabel 4.5 Skenario Pengujian Kedua.....	45
Tabel 4.6 Skenario Pengujian Ketiga.....	47
Tabel 4.7 Skenario Pengujian Pertama.....	49
Tabel 4.8 Skenario Pengujian Kedua.....	51
Tabel 4.9 Skenario Pengujian Ketiga.....	53
Tabel 4.10 Skenario Pengujian Pertama.....	55
Tabel 4.11 Skenario Pengujian Kedua.....	57
Tabel 4.12 Skenario Pengujian Ketiga.....	59
Tabel 5.1 Hasil <i>Delivery Probability</i> Pada Semua Skenario.....	62
Tabel 5.2 Hasil <i>Delivery Probability</i> Pada Semua Skenario.....	64
Tabel 5.3 Hasil <i>Delivery Probability</i> Pada Semua Skenario.....	65
Tabel 5.4 Hasil <i>Overhead Ratio</i> Pada Semua Skenario.....	71
Tabel 5.5 Hasil <i>Overhead Ratio</i> Pada Semua Skenario.....	72
Tabel 5.6 Hasil <i>Overhead Ratio</i> Pada Semua Skenario.....	74
Tabel 5.7 Hasil <i>Average Latency</i> Pada Semua Skenario.....	79
Tabel 5.8 Hasil <i>Average Latency</i> Pada Semua Skenario.....	80
Tabel 5.9 Hasil <i>Average Latency</i> Pada Semua Skenario.....	82
Tabel 5.10 Hasil Skenario 1 (<i>Epidemic</i>).....	87
Tabel 5.11 Hasil Skenario 1 (<i>Spray And Wait</i>).....	88

Tabel 5.12 Hasil Skenario 1 (<i>Epidemic</i>).....	88
Tabel 5.13 Hasil Skenario 1 (<i>Spray And Wait</i>).....	89
Tabel 5.14 Hasil Skenario 1 (<i>Epidemic</i>).....	89
Tabel 5.15 Hasil Skenario 1 (<i>Spray And Wait</i>).....	90
Tabel 5.16 Hasil Skenario 2 (<i>Epidemic</i>).....	91
Tabel 5.17 Hasil Skenario 2 (<i>Spray And Wait</i>).....	92
Tabel 5.18 Hasil Skenario 2 (<i>Epidemic</i>).....	92
Tabel 5.19 Hasil Skenario 2 (<i>Spray And Wait</i>).....	93
Tabel 5.20 Hasil Skenario 2 (<i>Epidemic</i>).....	93
Tabel 5.21 Hasil Skenario 2 (<i>Spray And Wait</i>).....	94
Tabel 5.22 Hasil Skenario 3 (<i>Epidemic</i>).....	95
Tabel 5.23 Hasil Skenario 3 (<i>Spray And Wait</i>).....	96
Tabel 5.24 Hasil Skenario 3 (<i>Epidemic</i>).....	96
Tabel 5.25 Hasil Skenario 3 (<i>Spray And Wait</i>).....	97
Tabel 5.26 Hasil Skenario 3 (<i>Epidemic</i>).....	97
Tabel 5.27 Hasil Skenario 3 (<i>Spray And Wait</i>).....	98

DAFTAR GAMBAR

Gambar 2.1 Lapisan <i>Bundle</i>	16
Gambar 2.2 Komunikasi ACK pada Lapisan <i>Bundle</i> <i>node</i> awal ke <i>node</i> yang lain.....	17
Gambar 2.3 Metode <i>store-and-forward message switching</i> (Warthman, 2003).....	17
Gambar 2.4 Proses Pengiriman dengan <i>Custody Transfer</i> (Warthman, 2003).....	18
Gambar 2.5 Mekanisme Pengiriman Pesan Ke Tujuan Pada Protokol <i>Routing Epidemic</i> (Sumber : Vahdat & D. Becker, 2000).....	19
Gambar 2.6 Contoh <i>Intermittently Connected Mobile Networks (ICMN)</i>	20
Gambar 2.7 Mekanisme Cara Kerja Protokol <i>Routing Spray And Wait</i>	22
Gambar 2.8 Tampilan GUI Pada Aplikasi <i>The One Simulator</i>	23
Gambar 2.9 Gambaran Lingkungan Pada <i>The One Simulator</i>	24
Gambar 2.10 Tampilan Utama Program <i>OpenJUMP</i>	26
Gambar 3.1 Diagram Alir Metodologi.....	28
Gambar 4.1 Tampilan Map Ketiga kota (Blitar, Tulung Agung, dan Trenggalek)	32
Gambar 4.2 Peta setelah diubah format menjadi <i>.wkt</i>	33
Gambar 4.3 Hasil <i>Linestring</i> jalur kota Blitar. <i>wkt</i>	34
Gambar 4. 4 hasil <i>Linestring</i> Jalur kota Tulung Agung. <i>wkt</i>	35
Gambar 4.5 Hasil <i>Linestring</i> Jalur kota Trenggalek. <i>wkt</i>	35
Gambar 4.6 Hasil <i>Linestring ConnectionLines.wkt</i>	36
Gambar 4.7 Setting Java Variabel.....	37
Gambar 4.8 Isi dari folder <i>The One Simulator</i>	37
Gambar 4.9 Proses Compile Aplikasi pada <i>The One Simulator</i>	38
Gambar 4.10 Proses <i>Compile</i> Aplikasi pada <i>The One Simulator</i>	38
Gambar 4.11 Tampilan <i>default setting</i> pada skenario Peta Helsinki.....	39

Gambar 4.12	Settingan Konfigurasi waktu pengiriman Pesan.....	41
Gambar 4.13	Settingan Konfigurasi Jumlah <i>Host Group</i>	41
Gambar 4.14	Settingan Konfigurasi Pada <i>Node</i> Yang Bergerak.....	42
Gambar 4.15	Konfigurasi Pergerakan <i>Node</i>	42
Gambar 4.16	Contoh Settingan Konfigurasi <i>Group</i>	42
Gambar 4.17	konfigurasi Konversi Ukuran Pesan.....	43
Gambar 4.18	Simulasi Pertama Protokol <i>Routing Epidemic</i>	44
Gambar 4.19	Simulasi Pertama Protokol <i>Routing Spray And Wait</i>	45
Gambar 4.20	Simulasi Kedua Protokol <i>Routing Epidemic</i>	46
Gambar 4.21	Simulasi Kedua Protokol <i>Routing Spray And Wait</i>	47
Gambar 4.22	Simulasi Ketiga Protokol <i>Routing Epidemic</i>	48
Gambar 4.23	Simulasi Ketiga Protokol <i>Routing Spray And Wait</i>	49
Gambar 4.24	Simulasi Pertama Protokol <i>Routing Epidemic</i>	50
Gambar 4.25	Simulasi Pertama Protokol <i>Routing Spray And Wait</i>	51
Gambar 4.26	Simulasi Kedua Protokol <i>Routing Epidemic</i>	52
Gambar 4.27	Simulasi Kedua Protokol <i>Routing Spray And Wait</i>	53
Gambar 4.28	Simulasi Ketiga Protokol <i>Routing Epidemic</i>	54
Gambar 4.29	Simulasi Ketiga Protokol <i>Routing Spray And Wait</i>	55
Gambar 4.30	Simulasi Pertama Protokol <i>Routing Epidemic</i>	56
Gambar 4.31	Simulasi Pertama Protokol <i>Routing Spray And Wait</i>	57
Gambar 4.32	Simulasi Kedua Protokol <i>Routing Epidemic</i>	58
Gambar 4.33	Simulasi Kedua Protokol <i>Routing Spray And Wait</i>	59
Gambar 4.34	Simulasi Ketiga Protokol <i>Routing Epidemic</i>	60
Gambar 4.35	Simulasi Ketiga Protokol <i>Routing Spray And Wait</i>	61
Gambar 5.1	Grafik <i>Delivery Probability</i> Pada Semua Skenario Pengujian....	63
Gambar 5.2	Grafik <i>Delivery Probability</i> Pada Semua Skenario Pengujian....	64

Gambar 5.3 Grafik <i>Delivery Probability</i> Pada Semua Skenario Pengujian....	66
Gambar 5.4 Grafik Perbandingan <i>Delivery Probability</i> Skenario 1 Berdasarkan Tipe Mobilitasnya.....	67
Gambar 5.5 Grafik Perbandingan <i>Delivery Probability</i> Skenario 2 Berdasarkan Tipe Mobilitasnya.....	68
Gambar 5.6 Grafik Perbandingan <i>Delivery Probability</i> Skenario 3 Berdasarkan Tipe Mobilitasnya.....	69
Gambar 5.7 Grafik <i>Overhead Ratio</i> Pada Semua Skenario Pengujian.....	71
Gambar 5.8 Grafik <i>Overhead Ratio</i> Pada Semua Skenario Pengujian.....	73
Gambar 5.9 Grafik <i>Overhead Ratio</i> Pada Semua Skenario Pengujian.....	74
Gambar 5.10 Grafik Perbandingan <i>Overhead Ratio</i> Berdasarkan Tipe Mobilitasnya (Skenario 1).....	75
Gambar 5.11 Grafik Perbandingan <i>Overhead Ratio</i> Berdasarkan Tipe Mobilitasnya (Skenario 2).....	76
Gambar 5.12 Grafik Perbandingan <i>Overhead Ratio</i> Berdasarkan Tipe Mobilitasnya (Skenario 3).....	77
Gambar 5.13 Grafik <i>Average Latency</i> Pada Semua Skenario Pengujian.....	79
Gambar 5.14 Grafik <i>Average Latency</i> Pada Semua Skenario Pengujian.....	81
Gambar 5.15 Grafik <i>Average Latency</i> Pada Semua Skenario Pengujian.....	82
Gambar 5.16 Grafik Perbandingan <i>Average Latency</i> Berdasarkan Tipe Mobilitasnya.....	83
Gambar 5.17 Grafik Perbandingan <i>Average Latency</i> Berdasarkan Tipe Mobilitasnya.....	84
Gambar 5.18 Grafik Perbandingan <i>Average Latency</i> Berdasarkan Tipe Mobilitasnya.....	85

DAFTAR LAMPIRAN

Lampiran 1 Konfigurasi <i>Default Settings</i> Pada <i>The One Simulator</i>	104
Lampiran 2 <i>Source Code</i> Protokol <i>Routing Epidemic</i> (<i>EpidemicRouter.java</i>)	110
Lampiran 3 <i>Source Code</i> Pada <i>Routing</i> Protokol <i>Spray And Wait</i> (<i>SprayAndWaitRouter.java</i>).....	113

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Jaringan Internet adalah grup jaringan komputer yang saling terhubung satu sama lain. Internet menyediakan akses jaringan telekomunikasi dan bahan informasi untuk masyarakat. Permasalahan Akses jaringan pada daerah yang memadai di Indonesia belum dapat tersebar di berbagai penjuru daerah, seperti di pegunungan, pantai dan tempat pedesaan yang sulit dijangkau jaringan lainnya yang masih minim akan koneksi jaringan Internet. Salah satu solusi agar tempat – tempat tersebut dapat berkomunikasi untuk mengirimkan informasi berupa pesan di daerah yang sulit dijangkau jaringan dari sumber ke tujuan dengan cepat. Maka, dengan menerapkan cara berkomunikasi melalui komunikasi jarak jauh dengan melakukan suatu pengiriman informasi untuk mengirimkan informasi pesan satu sama lainnya dengan menerapkan teknologi *Delay Tolerant Network* (DTN).

DTN merupakan jaringan yang tidak memperlumahkan waktu tunda. DTN dibuat untuk mengatasi kondisi jaringan dengan *delay* terpanjang, koneksi yang tidak stabil dan tingkat kerusakan yang tinggi (Fall, 2003). Selain itu terdapat fungsi *store-forward* yang memungkinkan kita tetap dapat mengirim data saat muncul kembali jaringan atau koneksi yang sebelumnya terputus atau hilang meskipun dalam proses menunggu koneksi atau jaringan muncul memerlukan waktu yang agak lama (Demmer, M., dkk, 2004). Penggunaan teknologi DTN tidak dapat dipisahkan dengan menerapkan suatu protokol *routing* yang tepat dengan kontur dan situasi pada suatu daerah.

Sarana dapat dibangun dan dimanfaatkan dalam membuat jaringan DTN, untuk memfasilitasi layanan DTN dengan menggunakan beberapa alat-alat pendukung berupa angkutan transportasi dalam mengantarkan sebuah pesan dengan transportasi mobil yang akan diterapkan sebagai perangkat *router* yang bergerak pada DTN (Warthman, 2003). Dalam mengantarkan sebuah pesan dari sumber ke tujuan dengan perangkat *router* DTN dapat menerapkan algoritma *routing* yang tepat untuk dapat menentukan jalur yang lebih cepat dan *optimal*.

Algoritma *Routing* adalah salah satu komponen utama dalam arsitektur DTN, *routing* di DTN menantang karena penundaan dan konektivitas *intermittent* yang tinggi (Bozed, dkk, 2016). Dengan menerapkan sebuah algoritma *routing* yang ada pada DTN sehingga dapat mengatur dan menentukan sebuah jalur jaringan untuk proses pengiriman pesan agar dapat terkirim dari sumber ke tujuan.

Penelitian sebelumnya yang pernah dilakukan sebelumnya oleh (Giwang Rudira Sugiyanto, 2015) yang berjudul “Analisis Protokol *Maxprop* dan *Prophet* pada simulasi jaringan DTN (*Delay Tolerant Network*)”. Penelitian tersebut membahas tentang pengujian perbandingan pada protokol *routing* *Maxprop* dan *Prophet* untuk skenario proses pengiriman pesan yang dilakukan dari sekolah

sampai ke sekolah lainnya yang 1 *node* sekolah akan dijadikan sebagai gateway penghubung untuk akses Internet. Pengiriman pesan dilakukan dengan menggunakan perantara *router* yang bergerak seperti angkutan umum yang akan mengirimkan informasi dari *node* sumber ke *node* tujuan dengan menggunakan simulator *One (Opportunistic Network Environment)* sebagai visualisasi proses pengiriman pesan pada daerah yang sulit dijangkau jaringan di kota Magetan.

Proses skenario yang akan dibuat dengan menggunakan sebuah aplikasi simulator *The One simulator* membutuhkan peta yang mempunyai skala yang akurat dan asli sesuai pada *Real world*, dimana pengambilan gambar peta dapat mudah diperoleh dari website *GIS (Geographic Information System)* seperti *Googlemap* dan *openstreetmap.org* diambil dalam gambar berformat PNG. Data dari gambar map inilah yang kemudian di-digitalisasi menggunakan aplikasi berbasis *GIS* yaitu *OpenJump*. *OpenJump* merupakan sebuah aplikasi berbasis *GIS (Geographic Information System)* yang dapat mengkonversi data dari peta dunia nyata. Data dari hasil *OpenJump* ini akan di digitalisasi menjadi format berjenis *.WKT (well known format)* yang digunakan untuk simulasi pada *ONE Simulator* (Keranen, A., dkk, 2009).

Melakukan pengujian dengan menerapkan model protokol algoritma *routing DTN* yang ada pada *The One simulator* yaitu *Epidemic* dan *Spray And Wait* untuk meneruskan pesan yang akan dikirim dari sumber ke tujuan. Algoritma *Routing Epidemic* merupakan *routing* yang dilakukan dengan cara membanjiri paket data, seperti virus yang menyebarkan *copy* salinan seluruh informasi data ke *node* yang terhubung dengan *node* sumber, dengan syarat bahwa seluruh *node* penerima tidak memiliki salinan informasi sesudahnya (D'Souza & Jose, 2010). Dan sedangkan protokol *routing* pada *Spray And Wait* yaitu proses penyebaran pesan yang telah disalin pada sebuah jaringan berbeda prosesnya dibandingkan dengan protokol *routing Epidemic*. Pada protokol *routing Spray And Wait* terdiri dari 2 fase, fase pertama adalah *fase spray*, yaitu fase dimana untuk setiap pesan yang berasal dari *node* sumber akan disalin Dan Salinan pesan yang telah disalin akan disebar dan diteruskan oleh *node* sumber dan kemungkinan, *node* lain akan menerima salinan pesan yang berbeda. Kemudian pada *fase Wait* adalah fase dimana jika *node* tujuan yang telah di *spray* atau di sebar tidak ditemukan, maka masing – masing *node* yang membawa salinan pesan akan melakukan transmisi langsung yaitu dengan meneruskan pesan langsung ke *node* tujuannya (Spyropoulos, dkk , 2005).

Dengan mengacu pada penelitian terdahulu tersebut maka simulasi dapat digunakan sebagai bahan analisis pengujian dengan menggunakan *The One simulator* dengan menggunakan perantara *node* yang bergerak sebagai *node* pengiriman pesannya, sehingga pada penelitian ini akan didapatkan hasil dari perbandingan 2 protokol *routing DTN* yang akan digunakan sebagai parameter untuk mengetahui solusi dari protokol *routing* pada pertukaran informasi dengan jumlah ukuran pesan yang bervariasi.

Oleh sebab itu, penelitian yang dapat diuji yang akan menjadi bahan judul pada tugas akhir ini yaitu dengan judul “Analisis *Performansi* Protokol *Routing Epidemic* Dan *Spray And Wait* Pada DTN Berdasarkan Mobilitas Node” pada 3 buah simulasi untuk membandingkan *performansi* dari kedua protokol *routing* tersebut. Dan tujuan untuk membandingkan *performansi* dari kedua protokol *routing* tersebut dikarenakan bahwa kedua protokol *routing* tersebut menggunakan mekanisme penyebaran pesan yang sama yaitu berbasis *flooding* (Keranen, A., dkk, 2009). Dan protokol *routing Spray And Wait* muncul dikarenakan untuk mengontrol masalah *flooding* yang tidak terbatas pada protokol *routing Epidemic* yang menyebabkan penyimpanan *Buffer* menjadi *Overflow*, oleh sebab itu protokol *routing Spray And Wait* dalam mengatasi masalah tersebut dengan cara membatasi jumlah salinan pesan atau mengurangi jumlah pesan yang akan dikirim sehingga penyimpanan *Buffer* tidak menjadi *Overflow* sehingga kedua protokol *routing* tersebut cocok untuk dilakukan penelitian untuk dianalisa perbandingan *performansi* kinerja kedua protokol *routing* tersebut (Spyropoulos, 2007). Setelah itu, skenario simulasi pengiriman pesan akan dibuat dengan menggunakan aplikasi *The One simulator* dengan merancang skenario sesuai dengan kondisi kota yang sebenarnya, tetapi dalam hal ini daerah kota yang akan dipilih dalam penelitian ini hanya untuk contoh visualisasi dalam mensimulasikan jaringan DTN. Kota yang dipilih adalah kota blitar, Tulung agung dan Trenggalek dengan menggunakan aplikasi simulasi *The ONE Simulator*. Setelah skenario simulasi pengiriman pesan telah dibuat, kemudian akan dilakukan uji coba dan evaluasi menggunakan *simulator The ONE (Opportunistic Network Environment)* (Keranen, A., dkk, 2009). Dengan menggunakan parameter - parameter pengujian yaitu *Delivery Probability*, *Average Latency (Delay)*, dan *Overhead Ratio* dari ukuran pesan yang bervariasi dari data yang akan digunakan dalam pengujian dan melakukan pengujian terhadap mobilitasnya, sehingga dapat dilakukannya analisis dari parameter yang digunakan dalam penelitian untuk mengetahui *performansi* dari kedua protokol *routing Epidemic* dan *Spray And Wait* dan analisis berdasarkan tipe mobilitasnya yaitu seperti mobilitas *Map Based Movement*, *Open Shortest Path*, dan *Map Route Movement* untuk diuji dengan menerapkan parameter – parameter pengujian yang ada ke beberapa tipe mobilitasnya. Dengan tujuan, akan mengetahui tipe mobilitas terbaik berdasarkan parameter pengujian tersebut yang akan diuji coba dalam penulisan tugas akhir ini.

1.2 Rumusan Masalah

Dari hasil paparan latar belakang, didapatkan 2 poin rumusan masalah yaitu :

Pertanyaan penelitian:

1. Bagaimana hasil analisis *performansi* dari perbandingan kinerja protokol *routing Epidemic* dan *Spray And Wait* pada skenario rute dengan

menggunakan parameter – parameter pengujian berdasarkan beberapa tipe mobilitas yang akan digunakan.

2. Manakah protokol *routing* terbaik diantara *Epidemic* dan *Spray and Wait* berdasarkan analisis pengujian yang telah dilakukan.

1.3 Tujuan

Tujuan dilakukan penelitian ini adalah sebagai berikut :

1. Untuk mendapatkan hasil perbandingan analisis kinerja protokol *routing Epidemic* dan *Spray And Wait* dalam keberhasilan pengiriman paket data pada simulasi jaringan DTN.
2. Untuk dapat mengetahui hasil perbandingan pada setiap skenario rute dalam menggunakan beberapa tipe mobilitas untuk kedua protokol *routing* seperti *Epidemic* dan *Spray And Wait* dalam simulasi Jaringan DTN.
3. Untuk dapat mengetahui kekurangan dan kelebihan yang didapatkan dari hasil pengujian analisis kinerja protokol *routing Epidemic* dan *Spray And Wait* dengan berdasarkan tipe mobilitasnya.

1.4 Manfaat

Pembuatan sistem ini diharapkan dapat bermanfaat untuk :

1. Dapat lebih memahami tentang perbandingan terhadap kinerja protokol *routing Epidemic* dan *Spray And Wait*.
2. Dapat lebih mengetahui hasil akhir analisis untuk penerapan protokol *routing Epidemic* dan *Spray And Wait* berdasarkan tipe mobilitasnya.

1.5 Batasan Masalah

Batasan masalah penelitian dalam skripsi ini yaitu :

1. Membuat skenario pengujian sesuai dengan rute yang ada pada 3 *Node Statis* seperti kota Blitar, Tulung Agung, dan Trenggalek.
2. Menggunakan 6 *Node* yang bergerak (dinamis) sebagai kurir penyampai pesan dari sumber ke tujuan.
3. Menggunakan beberapa tipe Mobilitas seperti *Map Based Movement*, *Open Shortest Path*, dan *Map Route Movement* pada protokol *routing Epidemic* dan *Spray and Wait*.
4. Menganalisis pengujian pada skenario menggunakan simulator berupa *The One Simulator*.

5. Menganalisa *Delivered Message* pada 2 protokol *routing Epidemic* dan *Spray and Wait*.
6. Melakukan uji coba dengan membandingkan beberapa skenario *Delivered Message* pada masing –masing ukuran pesan yang bervariasi yaitu 300 MB, 600 MB, 1,2 GB dengan berdasarkan parameter-parameter pada pengujian seperti *Delivery Probability*, *Average Latency (Delay)*, dan *Overhead Ratio* dan Mobilitasnya.

1.6 Sistematika Pembahasan

Penulisan skripsi ini meliputi beberapa bab sebagai berikut :

Bab I Pendahuluan

Pada bab ini berisi tentang latar belakang, rumusan masalah, tujuan , manfaat, batasan masalah, dan sistematika pembahasan yang terkait dengan penelitian yang diambil yaitu tentang Analisis *Performansi* Protokol *Routing Epidemic* dan *Spray And Wait* Pada DTN Dan Mobilitasnya.

Bab II Landasan Kepustakaan

Bab ini berisi tentang penjelasan teori – teori sebagai referensi yang digunakan sebagai landasan penyusunan topik skripsi yang menggunakan makalah ilmiah, jurnal, dan paper pendukung yang dibuat dalam melakukan penelitian.

Bab III Metodologi

Bab ini berisi metodologi penelitian yang akan dijelaskan tentang bagaimana alur dari tugas akhir yang akan dikerjakan penulis dengan menggambarkan proses alur diagram sehingga pengerjaan tugas akhir dapat terstruktur.

Bab IV Hasil

Bab ini berisi Perancangan skenario yang akan dikerjakan oleh penulis dalam melakukan penelitian untuk mendapatkan sebuah hasil dari pengujian skenario yang nantinya akan berupa data yang akan di analisa.

Bab V Pembahasan

Bab ini berisi penjelasan tentang hasil pengujian yang telah diuji yang akan dijelaskan kedalam sebuah grafik dan nantinya akan dapat menentukan kesimpulan pada bab selanjutnya.

Bab VI Penutup

Bab ini menjelaskan tentang hasil kesimpulan yang telah diuji pada kedua protokol *routing* berdasarkan poin-poin pertanyaan pada rumusan masalah.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada kajian pustaka ini akan menjelaskan mengenai penelitian-penelitian sebelumnya yang relevan dengan judul yang diajukan. Penelitian yang digunakan sebagai kajian dari pustaka adalah “Analisis *Performansi* Protokol *Routing Epidemic* dan *Spray And Wait* Pada DTN Berdasarkan Mobilitas Node”.

Tabel 2.1 Perbedaan Penelitian Sebelumnya Berhubungan Dengan Usulan Topik Skripsi

Nama Penulis , Tahun Judul	Judul Penelitian	Persamaan	Perbedaan	
			Penelitian Dahulu	Rancang Penelitian
Lidya Amalia Rahmania, 2013.	Penerapan <i>Delay Tolerant Network</i> (DTN) untuk Sistem Konsultasi Kesehatan Jarak Jauh Berbasis Web	- Menggunakan proses pengiriman informasi data dan menerima file menggunakan metode DTN(<i>Delay Tolerant Network</i>)	- Penelitian menggunakan DTN yang dibuat untuk Sistem Konsultasi Kesehatan Jarak Jauh yang digunakan untuk tenaga medis di desa dan di kota untuk saling berkonsultasi mengenai masalah kesehatan dengan tujuan meningkatkan ilmu pengetahuan dan meningkatkan	- Penelitian menggunakan DTN yang dibuat untuk sistem pengiriman pesan dengan menggunakan protokol <i>routing Epidemic</i> dan <i>Spray And Wait</i> dengan menggunakan aplikasi Simulator yaitu <i>The One simulator</i> dan <i>OpenJump</i>

Nama Penulis , Tahun Judul	Judul Penelitian	Persamaan	Perbedaan	
			Penelitian Dahulu	Rancang Penelitian
			<p>n mutu pelayanan kesehatan di daerah yang sulit dijangkau jaringan dengan menggunakan aplikasi dtn2 sebagai membantu dalam proses pengiriman pesan.</p> <ul style="list-style-type: none"> - Menggunakan protokol <i>routing static</i> 	<ul style="list-style-type: none"> - Menggunakan protokol <i>routing Epidemic</i> dan <i>Spray And Wait</i>

Nama Penulis , Tahun Judul	Judul Penelitian	Persamaan	Perbedaan	
			Penelitian Dahulu	Rancang Penelitian
Giwang Rudira Sugiyanto, 2015	Analisis Protokol <i>Maxprop</i> dan <i>Prophet</i> pada simulasi jaringan DTN (<i>Delay Tolerant Network</i>)	- Menggunakan Protokol <i>Routing</i> DTN dengan menggunakan aplikasi simulator <i>The One (Opportunistic Network Environment)</i>	- Menguji Kinerja protokol <i>routing</i> DTN seperti <i>Maxprop</i> dan <i>Prophet</i> dengan menggunakan skenario pengujian di kab.Magetan dengan parameter pengujian yaitu <i>delivery probability, average Latency (Delay), average buffer time, dan Overhead Ratio</i>	- Menguji <i>performansi</i> kinerja protokol <i>routing</i> DTN seperti <i>Epidemic</i> dan <i>Spray And Wait</i> dengan menggunakan skenario pengujian di kota Blitar, Tulung Agung, dan Trenggalek dengan parameter pengujian yaitu <i>delivery probability, average Latency (Delay) dan Overhead Ratio</i>

Nama Penulis , Tahun Judul	Judul Penelitian	Persamaan	Perbedaan	
			Penelitian Dahulu	Rancang Penelitian
Siska Permatasari , 2017	Analisis Kinerja Protokol <i>Routing Prophet, Epidemic, Dan Spray And Wait</i> menggunakan <i>The Opportunistic Network Environment Simulator</i>	- Menerapkan dan melakukan simulasi pada Protokol <i>Routing DTN</i> dengan menggunakan aplikasi simulator <i>The One (Opportunistic Network Environment)</i>	- Menguji Kinerja Protokol <i>Routing DTN</i> dengan menggunakan algoritma <i>Prophet, Epidemic</i> dan <i>Spray And Wait</i> dengan parameter pengujian berupa <i>Message Create Report, Message Delivery Report, Message Relay, Delivery Probability report, Average Latency, Overhead Ratio</i>	- Menguji <i>performansi</i> Kinerja Protokol <i>Routing DTN</i> dengan menggunakan <i>Epidemic</i> dan <i>Spray And Wait</i> dengan parameter pengujian berupa <i>Message Create Report, Message Delivery Report, Message Relay, Delivery Probability report, Average Latency, dan Overhead Ratio</i>

Nama Penulis , Tahun Judul	Judul Penelitian	Persamaan	Perbedaan	
			Penelitian Dahulu	Rancang Penelitian
			<ul style="list-style-type: none"> - Menggunakan 5 <i>Node</i> yang bergerak dalam penyampaian informasi pesan - Menggunakan model mobilitas <i>node</i> yaitu <i>Open Shortest Path</i> dalam pengujian terhadap beberapa skenario percobaan. 	<ul style="list-style-type: none"> - Menggunakan 6 <i>Node</i> yang bergerak dalam penyampaian informasi pesan - Menggunakan model mobilitas <i>node</i> yaitu <i>Map Based Movement, OpenShortest Path</i>, dan <i>Map Route Movement</i> dalam pengujian terhadap beberapa skenario percobaan

Dari paparan beberapa penelitian yang telah dijelaskan diatas, maka pada penulisan tugas akhir ini di fokuskan dengan menggunakan 2 protokol *routing Epidemic* dan *Spray And Wait* dengan berdasarkan parameter pengujian berupa *Message Create Report, Message Delivery Report, Message Relay, Delivery Probability report, Average Latency* dan *Overhead Ratio*.

2.2 Dasar Teori

2.2.1 *Delay Tolerant Network (DTN)*

Delay Tolerant Network adalah sebuah arsitektur jaringan komputer yang bertujuan untuk mengatasi permasalahan saat di daerah yang tidak terjangkau jaringan akses Internet. DTN juga merupakan arsitektur jaringan untuk membantu solusi bagi jaringan yang memiliki konektivitas yang terputus-putus, *delay* yang panjang, kecepatan data yang bervariasi dan tingkat akurasi kesalahan yang tinggi. DTN menyediakan model komunikasi *store-carry-forward*, yaitu model komunikasi yang dapat mengirimkan data berupa *bundle* yang dapat disimpan dan diteruskan oleh DTN. Model komunikasi ini bertumpu pada pergerakan *node* untuk mengirimkan data antara *node* yang terpisah secara geografis (Siswanti, 2013). Konsep DTN dapat diterapkan dengan memanfaatkan kendaraan yang berperan sebagai kurir untuk menjangkau daerah yang sulit dijangkau jaringan yang membutuhkan akses jaringan dan mengumpulkan data-data. Selanjutnya kurir bergerak ke tempat yang memiliki akses Internet (Siswanti, 2013). Banyak lokasi di dunia yang tidak dapat terjangkau oleh koneksi jaringan Internet tradisional seperti *fiber optic*, jaringan satellite maupun jaringan gelombang radio. DTN dapat digunakan untuk jaringan yang mempunyai suatu *delay* penundaan yang tinggi, koneksi yang terputus-putus dan konektivitas yang tidak selalu terkoneksi secara *real time* (Fall, 2003). DTN dapat diimplementasikan dengan menggunakan aplikasi *The One Simulator*, Dengan menggunakan fungsi yang ada pada *The One Simulator*, DTN dapat dibangun dan koneksi jaringan Internet dapat dihadirkan di daerah yang sulit dijangkau jaringan meskipun secara tidak langsung (Widhiarti, 2013). Kelebihan sistem yang dimiliki oleh Jaringan DTN dapat melakukan penjadwalan pengiriman data secara otomatis dalam proses pendistribusian data. Karena, jika proses pendistribusian data dilakukan dalam pengiriman data secara manual menggunakan bentuk fisik seperti flashdisk atau harddisk eksternal maka akan rentan akan kehilangan suatu data yang lebih besar.

Kurir sebagai *node* yang bergerak ini digunakan untuk menyebarluaskan data dari satu tempat ke tempat lain dimana koneksi akhir-ke-akhir (*end-to-end*) tidak tersedia. Berdasarkan hal tersebut sistem jaringan ini dapat sebagai solusi alternatif untuk jaringan komunikasi pada daerah yang sulit dijangkau jaringan yang memiliki permasalahan putusnya koneksi saat transfer data masih berlangsung. Jika koneksi terputus, DTN akan menyimpan sebagian data yang sudah berhasil dikirimkan, sehingga saat koneksi tersambung kembali, proses transfer data dapat dilanjutkan tanpa mengulang proses dari awal, sedangkan pada jaringan yang menggunakan protokol TCP/IP, pengiriman data akan gagal apabila koneksi terputus saat proses pengiriman (Siswanti, 2013). Berdasarkan hal tersebut pada penelitian ini menggunakan Aplikasi *The One simulator* sebagai visualisasi pengiriman pesan antar daerah yang sulit dijangkau jaringan dapat dilakukan.

Letak geografis dari daerah pedesaan yang sulit dijangkau jaringan dan tersebar yang jauh dari pusat kota sehingga tidak ada sarana telekomunikasi baik kabel maupun nirkabel yang menjangkau daerah tersebut, sehingga

penyampaian data yang dilakukan untuk daerah-daerah yang sulit dijangkau jaringan mengalami hambatan. Sebenarnya hal tersebut dapat diatasi dengan menyediakan jaringan komunikasi di daerah tersebut salah satunya memanfaatkan koneksi satelit (VSAT) tetapi hal ini membutuhkan biaya berlangganan yang cukup mahal. Selain itu dapat juga disediakan koneksi sms, tetapi hal ini pun menjadi kendala jika di daerah tersebut tidak disediakan jaringan komunikasi atau jaringan yang terputus-putus. Akibatnya data tidak tersampaikan dan dapat juga terjadi kesalahan dalam proses pengiriman transfer data informasi dari sumber ke tujuan.

Delay dapat didefinisikan sebagai selisih waktu antara saat pengiriman blok (sebuah bit atau sebuah paket data) informasi pada sumber dan saat penerimaan blok informasi tersebut di sisi penerima. Secara umum, *delay* yang terjadi dalam jaringan telekomunikasi disebabkan oleh *delay* transmisi dan *delay switching*. *Delay* transmisi tergantung pada jarak yang harus ditempuh antara sumber dengan tujuan dan juga tergantung pada jenis media transmisinya, tetapi tidak tergantung pada teknik pengiriman data yang digunakan. *Delay* transmisi terjadi karena terbatasnya kecepatan sinyal dalam media. *Delay switching* adalah *delay* total yang diperlukan sebuah sel untuk melalui perangkat *switching*, yaitu sejak sel masuk ke *inlet switching* hingga berada di outlet perangkat *switching* tersebut. Kecepatannya bergantung pada kecepatan *internal switch* dan proses-proses penambahan bit informasi yang diperlukan.

Arsitektur DTN (*Delay Tolerant Network*) mencakup konsep jaringan yang terhubung pada satu waktu dimana aktivitas ini terjadi akibat adanya partisi yang terdiri dari lebih dari satu set protokol yang berbeda atau keluarga protokol. Dasar untuk pembetukan arsitektur ini terletak pada *interplanetary* Internet, yang menitik beratkan permasalahan pada komunikasi dengan jarak yang sangat jauh di lingkungan yang tinggi tingkat *delay*nya. Pembahasan DTN ini diharapkan nantinya dapat dimanfaatkan di berbagai lingkungan operasional, termasuk untuk pemecahan permasalahan tersebut yaitu adanya *disruption* (gangguan) dan *disconnection* (pemutusan hubungan) dan permasalahan komunikasi jarak jauh dengan tingkat *delay* yang tinggi.

Jaringan lain juga dapat menerapkan arsitektur seperti ini antara lain jaringan sensor yang dilengkapi dengan konektivitas *intermittent* yang terjadwal, jaringan nirkabel terrestrial yang biasanya tidak dapat melakukan perawatan konektivitas end to end, jaringan satelit dengan *moderate delays* dan *periodic connectivity* dan jaringan bawah air dengan tingkat keterlambatan yang sedang dan gangguan yang diakibatkan oleh faktor lingkungan.

Dalam arsitektur DTN (*Delay Tolerant Network*), didefinisikan bahwa *end-to-end message-oriented overlay* disebut sebagai "*Bundle layer*", dimana *Bundle layer* ini terletak di atas layer transport atau layer lainnya pada suatu jaringan yang terhosted dan dimana di dalamnya terdapat berbagai aplikasi. Perangkat yang mengimplementasikan *Bundle layer* disebut sebagai DTN *nodes*. Bentuk-bentuk *Bundle layer* tersebut, ada yang dijadikan sebagai tempat penyimpanan yang persisten untuk membantu mengurangi adanya gangguan jaringan. Di dalamnya juga termasuk *hop by hop transfer* pengiriman yang dapat diandalkan

dan *optional end-to-end acknowledgement*. Di dalamnya juga mencakup adanya sejumlah *diagnostic* dan management fitur. Untuk *interoperabilitas*, *Bundle* layer menggunakan skema penamaan yang fleksibel (berdasarkan *Uniform Resource Identifiers* [RFC3986]) yang mampu mengenkapsulasi skema penamaan dan pengalamatan yang berbeda di dalam keseluruhan sintaks penamaan yang sama.

Bundle layer menyediakan sejumlah fungsionalitas yang mirip dengan *gateway* layer Internet yang dijelaskan di dalam ARPANET/Internet designs. Namun, *Bundle* layer berbeda dengan ARPANET gateways, karena *Bundle* layer ini merupakan layer-agnostic dan lebih terfokus pada *virtual message forwarding* daripada terfokus pada packet *switching*. Bagaimanapun juga, secara umum keduanya mendukung *interoperabilitas* antara *underlying* protokol yang khusus untuk suatu lingkungan dan protokol-protokol yang khusus untuk lingkungan lainnya, dan juga keduanya mendukung layanan *store-and-forward forwarding* (dengan tetap memanfaatkan penyimpanan *persistent Bundle* layer untuk penyimpanannya dan fungsi *forward*).

Arsitektur DTN juga menyediakan metode umum untuk *interconnecting heterogeneous gateways* atau *proxy* yang menggunakan *store-and-forward message routing* untuk mengatasi berbagai macam gangguan komunikasi. Arsitektur DTN juga menyediakan layanan yang mirip dengan *electronic mail (e-mail)* tetapi dengan adanya peningkatan penamaan, *routing* dan kemampuan *security*. *Nodes* tidak dapat mendukung secara penuh kapabilitas yang diperlukan oleh arsitektur, karena mungkin tidak didukung oleh lapisan aplikasi *proxy* yang bertindak sebagai DTN. Dibentuknya sebuah arsitektur *Delay Tolerant Network* (DTN) didasarkan pada beberapa faktor. Garis besarnya adalah protokol Internet yang ada sekarang (telah ada sebelumnya), tidak dapat bekerja secara maksimal pada beberapa lingkungan.

Protokol Internet yang ada saat ini belum dapat menjangkau semua karakteristik Internet yang dibutuhkan oleh pengguna. Beberapa karakteristik yang belum dapat terpenuhi antara lain (Warthman, 2003) :

- *Continuous, Bidirectional End-to-End path* : Sebuah koneksi dua arah yang selalu tersedia antara sumber dan tujuan untuk mendukung interaksi *end-to-end*.
- *Short round-trips*: *Delay* yang pendek dalam pengiriman paket data dan jaringan yang konsisten.
- *Symmetric data rates* : Kecepatan transmisi data yang konsisten antara sumber dan tujuan.
- *Low error rates* : Tingkat kemungkinan *error* dan korupsi data yang rendah.

Arsitektur DTN dirancang untuk lebih memperlunak asumsi-asumsi tersebut, berikut adalah asumsi-asumsi yang terdapat pada arsitektur DTN (IETF Trust, 2007):

- Menggunakan *variable-length message* (tidak terbatas pada ukuran paket) sebagai *communication abstraction* untuk membantu meningkatkan kemampuan jaringan dalam hal membuat penjadwalan yang baik atau pemilihan *path* apabila dirasa memungkinkan.
- Menggunakan sintaks penamaan yang mendukung berbagai macam penamaan dan pengalamatan untuk meningkatkan *interoperabilitas*.
- Menggunakan penyimpanan dimana jaringan mendukung *store-and-forward operation* pada *multiple path* dan secara potensial memiliki rentang waktu yang lebih panjang (contohnya untuk mendukung berbagai operasi di dalam lingkungan yang banyak terdapat atau tidak terdapat *end to end path*) serta tidak terlalu mementingkan *end-to-end reliability*.
- Menyediakan mekanisme *security* untuk melindungi infrastruktur dari penggunaan yang tidak sah dengan cara menghilangkan *traffic* secepat mungkin.
- Menyediakan *coarse-grained classes of service, delivery options*, dan cara untuk menyampaikan *useful lifetime of data* dengan cepat sehingga memungkinkan jaringan untuk mengirimkan data secara lebih baik dalam menerjemahkan kebutuhan aplikasi.

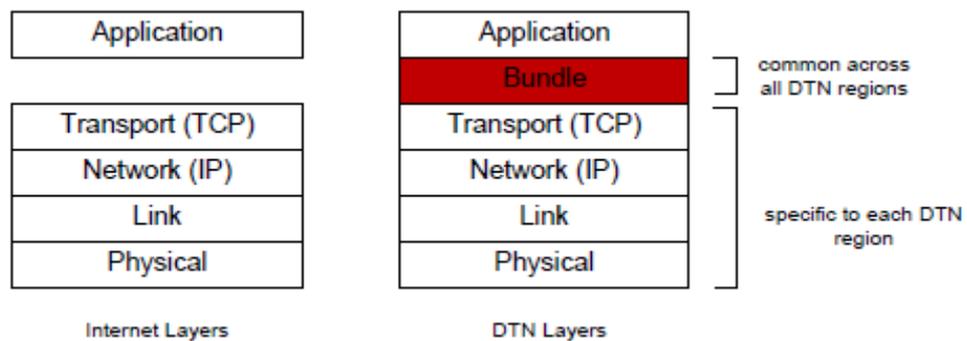
Karakteristik pada dtn antara lain :

- ❖ *Intermittent Connectivity*. Apabila tidak terdapat koneksi *end-to-end* diantara sumber dan tujuan (*network partitioning*), maka komunikasi *end-to-end* dengan menggunakan protokol TCP/IP mustahil dilakukan.
- ❖ *Long or Variable Delay*. Waktu tunda yang panjang adalah efek dari *intermittent connectivity*, waktu tunda perambatan yang panjang antara *node-node* menghasilkan total waktu tunda yang cukup panjang antara sumber dan tujuan. Hal ini menyebabkan protokol IP terhambat dan aplikasi yang membutuhkan sinyal ACK data yang cepat akan terganggu dan tidak bisa berfungsi.
- ❖ *Asymmetric Data Rates*. Internet mendukung dengan laju data yang asimetris. Meskipun terdapat perbedaan laju ada, akan tetapi tidak boleh terlalu besar. Apabila perbedaan laju data terlalu tinggi, maka komunikasi tidak dapat dilakukan. Contoh jaringan yang memiliki laju data asimetris adalah jaringan Internet melalui TV kabel atau *digital subscriber line* (DSL).
- ❖ *High Error Rates*. Kesalahan pengiriman pada bit data akan membutuhkan pengiriman ulang keseluruhan paket yang dapat menyebabkan lalu lintas

jaringan semakin tinggi. Semakin sedikit pengiriman ulang yang perlu dilakukan maka semakin lancar komunikasi TCP/IP yang terjadi.

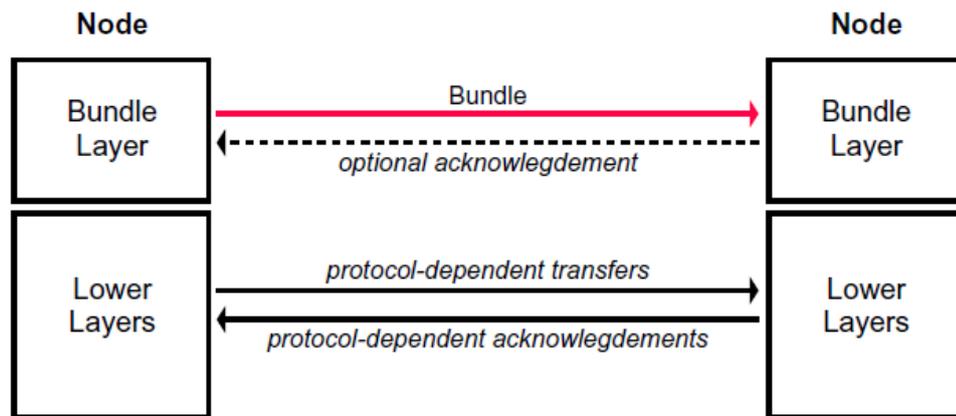
2.2.1.1 Bundle Layer

Bundle merupakan unit dasar data berupa ukuran variabel dan *signal* yang diperlukan untuk melintasi jaringan DTN. Lapisan *Bundle* atau *Bundle* protokol merupakan protokol utama yang digunakan dalam DTN. Lapisan *Bundle* terkait dengan spesifikasi *region* lapisan dibawahnya, sehingga memungkinkan program aplikasi dapat berkomunikasi dengan beberapa *region* lainnya. Lapisan *Bundle* bertugas menyimpan dan meneruskan keseluruhan atau sebagian *Bundle* di antara *node*. Suatu lapisan *Bundle* protokol digunakan pada saat melintasi semua jaringan (*region*) dalam DTN. Sebaliknya, lapisan di bawah lapisan *Bundle* (lapisan *transport*, *network*, *link*, *physical*), dipilih sesuai dengan lingkungan komunikasi pada setiap *region*. Gambar 2.1 menunjukkan letak lapisan *Bundle* pada DTN diperbandingan dengan lapisan Internet.



Gambar 2.1 Lapisan Bundle

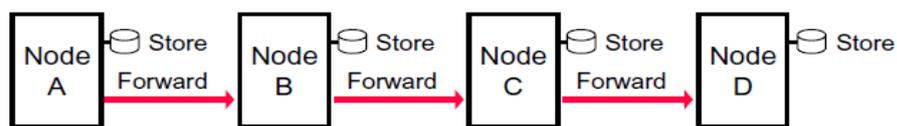
Lapisan *Bundle* berada di antara lapisan *transport* dan *application* dalam lapisan OSI. Sebuah protokol *Bundle* layer digunakan di seluruh jaringan yang tercakup dalam DTN. Pada jaringan yang koneksinya tidak selalu ada dan memiliki waktu tunda yang panjang, protokol yang membutuhkan percakapan seperti TCP, akan membutuhkan waktu yang panjang dan gagal untuk berkomunikasi. Oleh karena itu, DTN menyediakan lapisan *Bundle* yang berkomunikasi dengan lapisan *Bundle* lainnya menggunakan sesi sederhana yang terdapat fasilitas ACK. Diagram pengiriman ACK oleh lapisan *Bundle* dengan lapisan-lapisan yang ada di bawahnya ditunjukkan dalam Gambar 2.2. (Warthman, 2003).



Gambar 2.2 Komunikasi ACK pada Lapisan *Bundle node* awal ke *node* yang lain

2.2.1.2 Metode *Store-and-Forward*

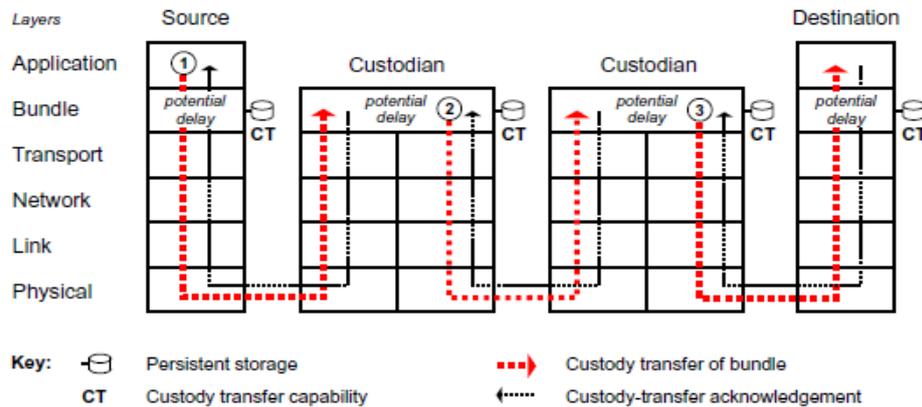
DTN yang memiliki sifat yang tidak menentu pada proses pengiriman data dari *node* awal ke *node* tujuan, memiliki kemungkinan *error* yang tinggi dan kecepatan data yang tidak sama. Dengan menerapkan cara *store-and-forward message switching*. Paket data akan disimpan dan diteruskan seperti yang digunakan pada sistem pos. Keseluruhan pesan atau pecahannya (*fragment*) dipindahkan dari tempat penyimpanan di sebuah titik (*node*) ke titik lain pada jalur antara sumber dan tujuan (Warthman, 2003). Gambar 2.3 dibawah ini yang menerapkan cara metode *store-and-forward message switching*.



Gambar 2.3 Metode *store-and-forward message switching* (Warthman, 2003)

2.2.1.3 *Custody Transfer*

Custody transfer merupakan salah satu metode pengiriman kembali data pada DTN. Hal ini dilakukan untuk menanggulangi permasalahan kesalahan pengiriman data pada DTN. *Custody transfer* hanya dapat dilakukan pada *Bundle layer*, dimana hanya pada *node-node* yang sudah menerima *Bundle*. Mekanisme pengiriman *Bundle* adalah ketika sebuah *node* akan mengirim suatu *Bundle*, maka *node* tersebut akan meminta *node* berikutnya untuk mengirimkan *Bundle acknowledgment*. *Bundle acknowledgment* menyatakan bahwa *node* tersebut telah siap menerima *Bundle* dari *node* sebelumnya. Akan tetapi jika sampai waktu yang ditentukan, *node* pengirim tidak menerima *Bundle acknowledgment*, maka *Bundle* akan dihapus dari *storage* (Warthman, 2003).



Gambar 2.4 Proses Pengiriman dengan *Custody Transfer* (Warthman, 2003)

2.2.2 Node Statis dan Dinamis

Node Statis merupakan sebuah *node* tetap yang berhubungan dengan perangkat jaringan lainnya untuk mengirimkan suatu paket data ke *node* yang lain tanpa berpindah tempat.

Node Dinamis berbeda dengan *node statis*. *Node Dinamis* merupakan sebuah *node* yang memiliki kemampuan mobilitas yang tidak terbatas. *Node dinamis* akan terhubung dengan perangkat jaringan lain ketika telah mencapai suatu *range* area jaringan lainnya yang terhubung dengannya. Berikut ini merupakan contoh penerapan dari *node statis* dan *node dinamis*. Terlihat bahwa link statis bersifat tetap dalam penentuan jalur, Sedangkan link dinamis dapat berubah-ubah atau bergerak dari sumber menuju ke *node* selanjutnya hingga sampai di tujuan.

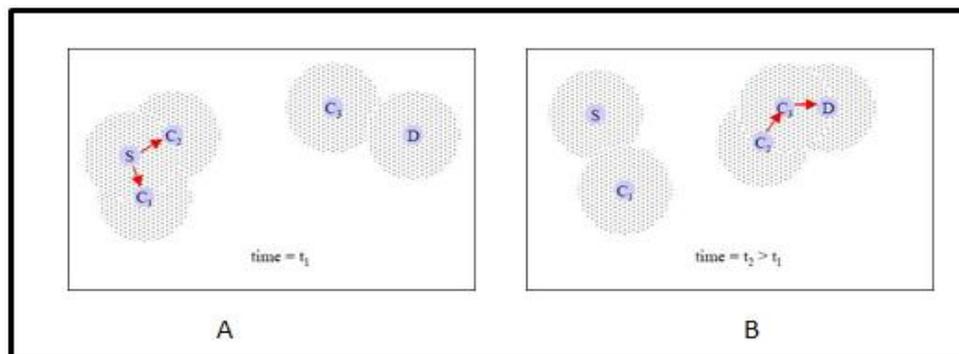
2.2.3 Routing Protokol Epidemic

Algoritma *Routing Protokol Epidemic* yang diterbitkan oleh Vahdat dan Becker et al (2000). Dan diusulkan sebagai algoritma *forwarding* berbasis *Flooding* (Modi & Desai a, 2015). *Epidemic Routing* juga mendistribusikan informasi pesan ke host yang disebut operator bagian-bagian dalam yang terhubung dari jaringan ad hoc. Dengan cara ini, pesan cepat didistribusikan melalui bagian yang terhubung jaringan. *Epidemic Routing* kemudian bergantung pada operator yang datang ke dalam kontak dengan bagian lain yang terhubung jaringan melalui mobilitas simpul. Pada titik ini, pesan menyebar ke *node* yang telah ditambahkan. Untuk menjaga konsistensi data, sinkronisasi database dilakukan dengan menggunakan algoritma *routing Epidemic*. Algoritma *routing Epidemic* merupakan algoritma *routing* yang dilakukan dengan cara membanjiri paket data, seperti virus yang menyebarkan *copy* Salinan seluruh informasi data ke *node* yang terhubung dengan *node* sumber, dengan syarat bahwa seluruh *node* penerima tidak memiliki salinan informasi sesudahnya (D'Souza & Jose , 2010).

Algoritma *Routing Epidemic* juga dapat disebut sebagai *flooding-based forwarding algorithm* (Modi & Desai a, 2015). Hasil akhir tujuan dari *Routing Epidemic* tersebut adalah untuk dapat memaksimalkan tingkat pengiriman pesan, meminimalkan pesan *Latency* dan Meminimalkan total sumber daya *bandwidth* yang digunakan dalam pengiriman pesan (Modi & Desai a, 2015). Jenis *routing* yang seperti ini akan menghasilkan inefisiensi penggunaan sumber daya jaringan. Karena, penyampaian beberapa salinan dari pesan yang sama, *bandwidth* dan mahal dalam hal konsumsi energi dan memori. Disini terlihat bahwa *routing Epidemic* menyediakan penyebaran salinan data tercepat dalam jaringan yang juga dapat menghasilkan suatu waktu pengiriman yang optimal. Namun, *Flooding* dapat menyebabkan sejumlah besar paket kontrol di saluran kontrol, yang dapat mengakibatkan kemacetan jaringan (Modi & Desai a, 2015). Melalui transmisi transitif seperti data pesan yang memiliki probabilitas tinggi akhirnya mencapai tujuan mereka (Vahdat & D. Becker, 2000).

2.2.3.1 Mekanisme Pengiriman Pesan Pada Protokol *Routing Epidemic*

Mekanisme Pengiriman Pesan yang menggambarkan *Epidemic Routing* pada tingkat tinggi, dengan *mobile node* yang direpresentasikan sebagai lingkaran hitam dan jangkauan komunikasi nirkabel mereka ditampilkan sebagai lingkaran bertitik membentang dari sumber (Vahdat & D. Becker, 2000).



Gambar 2.5 Mekanisme Pengiriman Pesan Ke Tujuan Pada Protokol *Routing Epidemic* (Sumber : Vahdat & D. Becker, 2000).

Kasus Pada Gambar 2.5 (A) , sumber S ingin mengirim pesan ke tujuan ke D tetapi tidak ada jalan yang terhubung tersedia dari S ke D. S mengirimkan pesan kepada dua *node* tetangganya yaitu C1 dan C2 dalam komunikasi jarak langsung. Pada beberapa waktu kemudian, seperti yang ditunjukkan pada Gambar 2.5 (B), *node* C2 datang ke dalam jangkauan komunikasi langsung dengan host lain yaitu *node* C3 untuk mengirimkan pesan. *Node* C3 terhubung dalam jaringan secara langsung dengan *node* D dan akhirnya *node* C3 langsung mengirimkan pesan ke tujuan *node* D. Sebuah sumber S ingin mengirimkan pesan ke tujuan tapi tidak ada jalan terhubung tersedia dibagian (a). Operator C1-C3 yang dimanfaatkan untuk transitif menyampaikan pesan ke tujuan di beberapa titik waktu seperti pada (b). Tujuannya adalah untuk menyampaikan pesan dengan probabilitas tinggi bahkan ketika tidak pernah ada *path* sepenuhnya

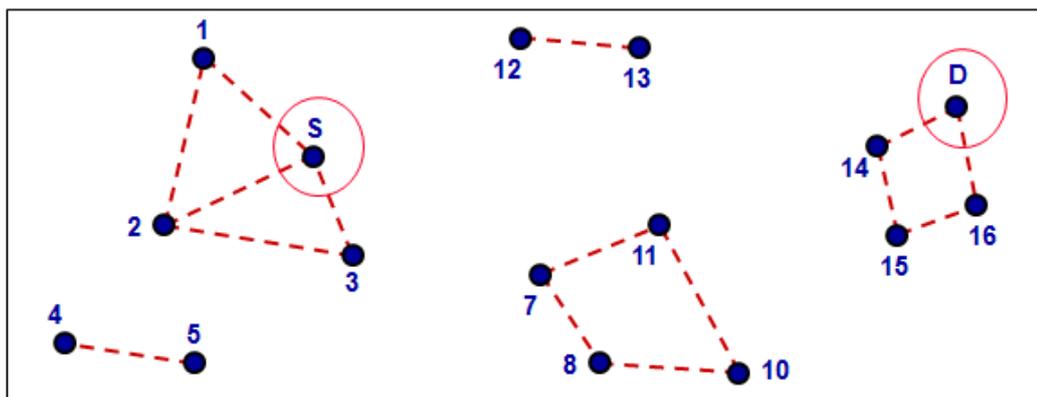
terhubung, Protokol *Routing* dapat melakukan yang lebih baik (Vahdat & D. Becker, 2000).

2.2.4 Routing Protokol *Spray And Wait*

Protokol *Routing Spray And Wait* mengatasi masalah *flooding* yang terjadi pada protokol *routing Epidemic*, tetapi pada *Spray And Wait* berusaha untuk mengontrol jumlah *copy message* untuk mengurangi *cost* pada konsep *flooding* pada *Epidemic*. Awalnya penyebaran *message* dilakukan dengan cara yang sama dengan *Epidemic*. Ketika *message* telah menyebar di setiap *node relay* untuk menjamin bahwa setidaknya salah satu dari *node relay* akan menemukan *node Destination* dengan cepat (dengan probabilitas yang tinggi), ketika *node Destination* tidak ditemukan maka *node relay* dapat melakukan transmisi secara langsung atau *direct transmission* ke *node Destination* (Spyropoulos, 2007).

Spray And Wait mengambil keuntungan dari protokol *routing Epidemic* dengan transmisi *message* lebih cepat, *Delivery Ratio* yang tinggi dan proses *forwarding* ke *Destination* secara langsung. Tujuan dari protokol *routing Spray And Wait* adalah mengontrol *flooding* dengan membatasi atau mengurangi jumlah *copy* (L) yang dibuat dan mengurangi *overhead* di *Epidemic* yang berhubungan dengan jumlah *node* (N).

Intermittently Connected Mobile Networks (ICMN) adalah jalur wireless yang tersebar dimana-mana yang sebagian besar *nodenya* tidak memiliki jalur (*path*) lengkap atau tidak terhubung secara langsung dari *node source* ke *node Destination* atau tidak ada *end-to-end path*.



Gambar 2.6 Contoh *Intermittently Connected Mobile Networks (ICMN)*

(Sumber: Spyropoulos, 2007)

Pada Gambar 2.6 telah dijelaskan bahwa, S adalah *node source* dan D adalah *node Destination*, dalam hal ini tidak ada jalur yang terhubung secara langsung dari *node Source* ke *node Destination*. Dalam keadaan ini semua

protokol konvensional akan gagal dalam hal pengiriman pesan secara langsung. Pada skema *routing* protokol, *node source* mengirimkan *copy message* ke *node Destination* dengan cara mengcopy *message* ke *node* tetangga atau *node "relay"* atau *L message copies* diteruskan oleh *node source* untuk *L relay* yang berbeda.

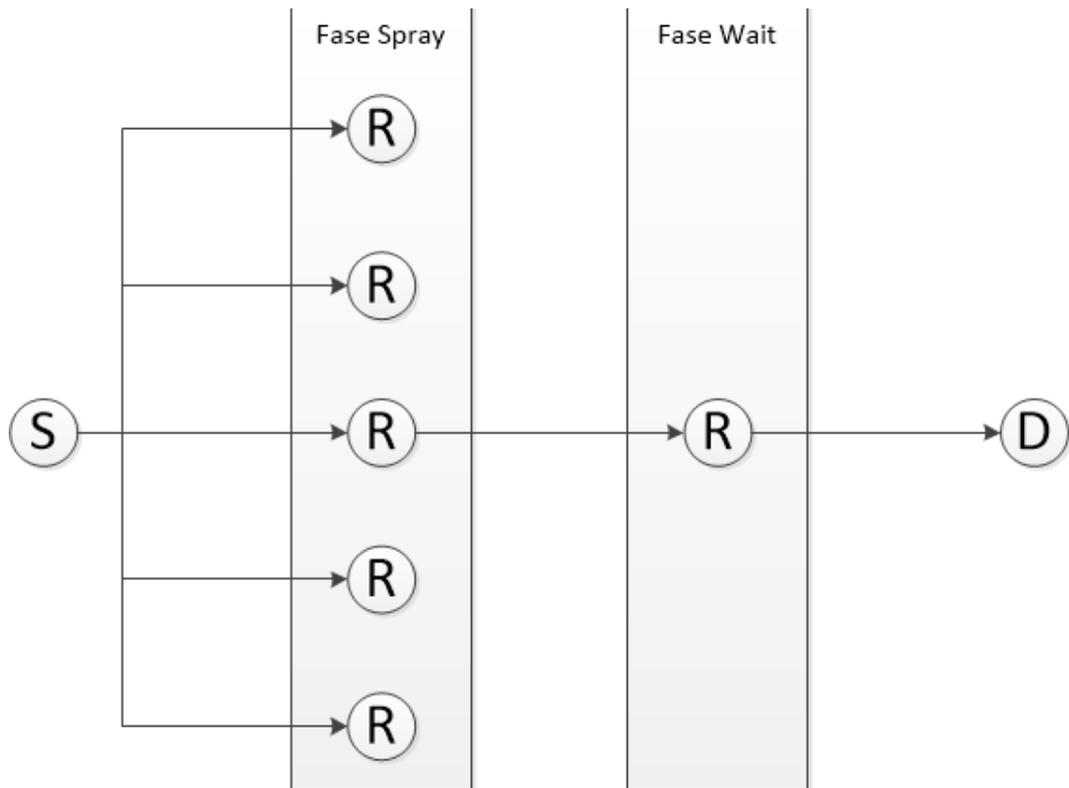
Cara kerja protokol *routing Spray And Wait* terdiri dari dua fase yaitu :

1. Fase *Spray*

Fase yang pertama adalah fase *Spray* dimana *node Source* membuat *L copy* pesan untuk disebar ke *relay node*. Fase *Spray* membatasi *message* yang dicopy untuk meminimalkan penggunaan sumber daya (*resource*) jaringan. Pada fase *Spray*, proses *multi-cast* dilakukan untuk mengirim beberapa *copy message* dari *source* ke *relay node*. Jika *Destination* tidak ditemukan dalam fase *Spray* maka *node* akan memasuki tahap "*wait*" dimana setiap *relay node* yang memiliki *copy message* menunggu sampai *node* tujuan ditemukan untuk mentransmisikan *message*

2. Fase *Wait*

Fase yang kedua adalah fase *Wait*. Jika *node Destination* tidak ditemukan dalam fase *Spray*, maka setiap *relay node* yang membawa *copy L* melakukan transmisi langsung ke *node Destination* yaitu meneruskan *message* hanya untuk *node Destination*. Pada fase *Wait* *node* diperbolehkan untuk menyampaikan *message* ke *node Destination* menggunakan transmisi secara langsung (*direct transmission*) ketika *Time-To-Live* berakhir. Pada fase *Wait* sebuah *node* akan meneruskan *message* ke *relay node* yang lain sampai tersisa satu *message* saja, dan *relay node* yang hanya memegang satu *copy message* akan masuk dalam fase *Wait*. Pada fase ini, *relay node* akan menunggu sampai bertemu *node Destination* dan melakukan transmisi.



Gambar 2.7 Mekanisme Cara Kerja Protokol *Routing Spray And Wait*

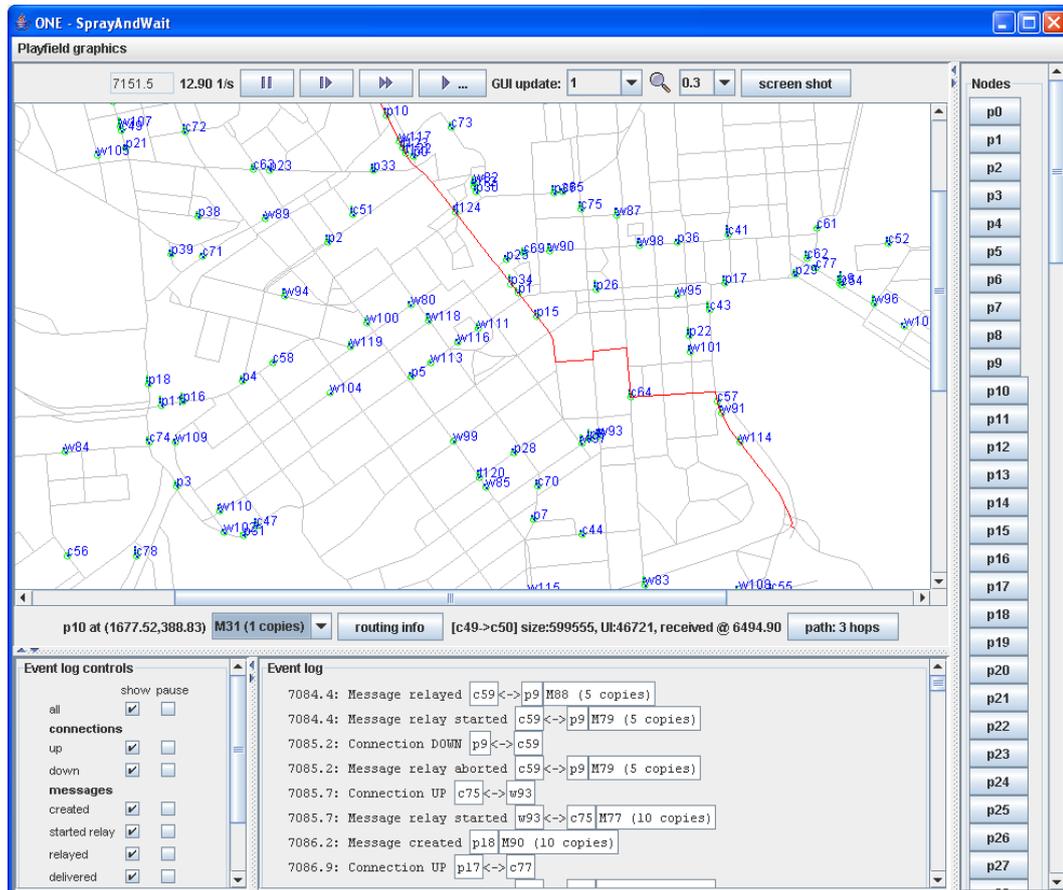
Pada Gambar 2.7 telah dijelaskan bahwa, *node S* yang bertindak sebagai *node Source* akan mengirimkan pesan ke *node D (Destination)* dengan menggunakan cara kerja pertama pada tahap fase *spray* yaitu dengan *node s (Source)* akan mengcopy dan mengirimkan pesan ke *node Relay(R)* atau *node* selanjutnya dengan syarat, *node Relay (R)* Selanjutnya harus berada didalam daerah jangkauan jaringan pada *node S (Source)* maka pesan akan berhasil dikirimkan. Setelah pesan berhasil dikirimkan, selanjutnya akan dilanjutkan ke tahap fase *Wait* dimana *node Relay (R)* pada fase *Spray* akan mengirimkan pesan ke *node Relay (R)* selanjutnya, jika *Node Relay (R)* selanjutnya tidak ditemukan atau tidak dalam jangkauan jaringan, maka *node* akan menunggu hingga *time to live (TTL)* habis, setelah itu pesan langsung dikirimkan ke *node Relay (R)* selanjutnya. Setelah itu *node Relay (R)* terakhir akan dapat mengirimkan ke *node Destination (D)* jika ada koneksi jaringan (Spyropoulos, 2007).

2.2.5 THE ONE SIMULATOR

Aplikasi *The One (Opportunistic Network Environment)* merupakan sebuah simulator untuk simulasi pemodelan jaringan DTN (*Delay Tolerant Network*). *The One simulator* merupakan simulator berbasis *Open Source* yaitu *java*. Fungsi Utama *The One simulator* yaitu memodelkan peta suatu daerah, pergerakan *node*, hubungan antar *node*, *routing* dan penanganan pesan pada simulasi jaringan DTN (keranen, 2008). Pergerakan *node* diimplementasikan menggunakan beberapa model pergerakan. Terdapat beberapa model

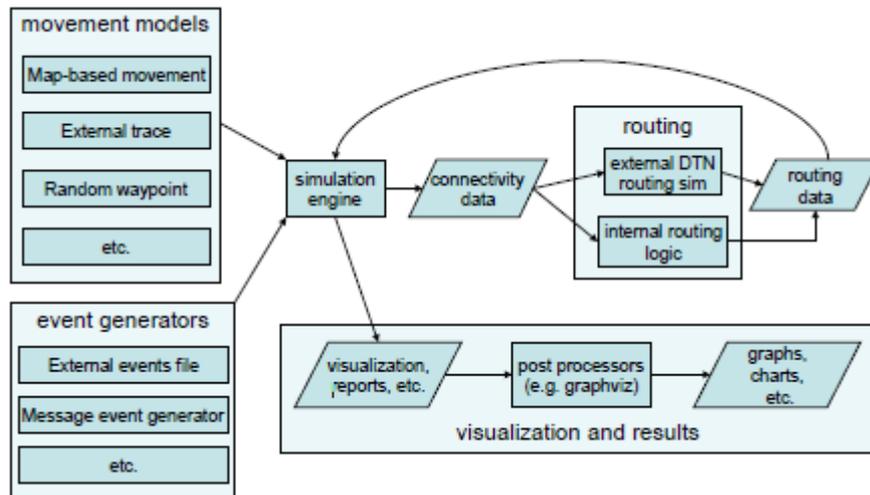
pergerakan yang sudah ada dalam simulator. Fungsi *routing* diterapkan menggunakan modul yang menentukan untuk meneruskan pesan.

Hasil simulasi jaringan DTN didapatkan dari proses simulasi berupa file laporan yang berisi *statistic report*. *The One simulator* memvisualisasikan simulasi dengan melalui *Graphical user interface (GUI)* seperti yang ditunjukkan pada Gambar 2.7 (keranen, 2008).



Gambar 2.8 Tampilan GUI Pada Aplikasi *The One Simulator*

Pada Gambar 2.8, Aplikasi *The One simulator* dapat menampilkan simulasi peta jalur secara nyata, lokasi *node*, jalur, hubungan *node* antar *node*, Jumlah Pesan yang diterima oleh *node* dan semuanya akan divisualisasikan pada jendela utama. Jika pergerakan *node* berbasis *map*, maka semua peta dan jalur *map* akan ditampilkan. Pada Aplikasi *The One simulator* membutuhkan peta atau jalur *node* untuk perpindahan *node*. Peta yang digunakan oleh Aplikasi *The One simulator* mempunyai format ekstensi WKT(*Well Known Text*). Format WKT ini merupakan format yang sering digunakan pada program *Geographic Information System (GIS)*. Program *Editor GIS* yang dapat digunakan yaitu *OpenJump*.



Gambar 2.9 Gambaran Lingkungan Pada *The One Simulator*

(Sumber : Keranen, 2009)

Pada Gambar 2.9 menjelaskan tentang fungsi dari *The One simulator* yaitu untuk memodelkan mobilitas *node*, mengatur koneksi jaringan antar *node*, mengatur *routing*, menampilkan visualisasi peta, dan *message handling*. *Movement model* digunakan untuk mengatur pergerakan *node* yang telah disiapkan pada simulator, *event generators* digunakan sebagai mengatur jumlah banyaknya event dan membuat jumlah pesan untuk simulasi. *Routing* yang digunakan pada *The One simulator* ada dua macam yaitu *external DTN routing sim* dan *internal routing logic* dalam mengatur logika *routing* didalam dan diluar DTN. Setelah itu hasil dan visualisasi dapat diperlihatkan pada *report file*, *post processors*, dan dalam bentuk data berupa grafik atau *charts*.

❖ Pergerakan Node Pada *The One Simulator*

MapBasedMovement adalah model pergerakan yang sudah terintegrasi dengan *The ONE simulator*. Model pergerakan ini mengharuskan *node* bergerak sepanjang jalur yang ditentukan pada sebuah peta (Keranen, 2009). *The One simulator* memiliki tiga jenis pergerakan *MapBasedMovement* yaitu seperti penjelasan dibawah ini :

1. *Random Map-based Movement (MBM)*

Pergerakan *node* jenis ini merupakan yang paling sederhana. *Node* bergerak secara acak namun tetap mengikuti alur pada peta dan menghasilkan data yang kurang akurat.

2. Shortest Path Map Based Movement (SPMBM)

Pergerakan *node* dengan cara memilih satu titik yang dituju secara acak dan bergerak melalui jalur terpendek yang dapat dilalui menuju titik tersebut. Tempat tujuan dipilih secara acak atau berdasarkan *Point of Interest* (POI). POI dipilih berdasarkan tempat-tempat yang sering menjadi destinasi manusia seperti restaurant, tempat rekreasi, tempat perbelanjaan, tempat pengisian bahan bakar dan lain-lain.

3. Routed Map-based Movement (RMB)

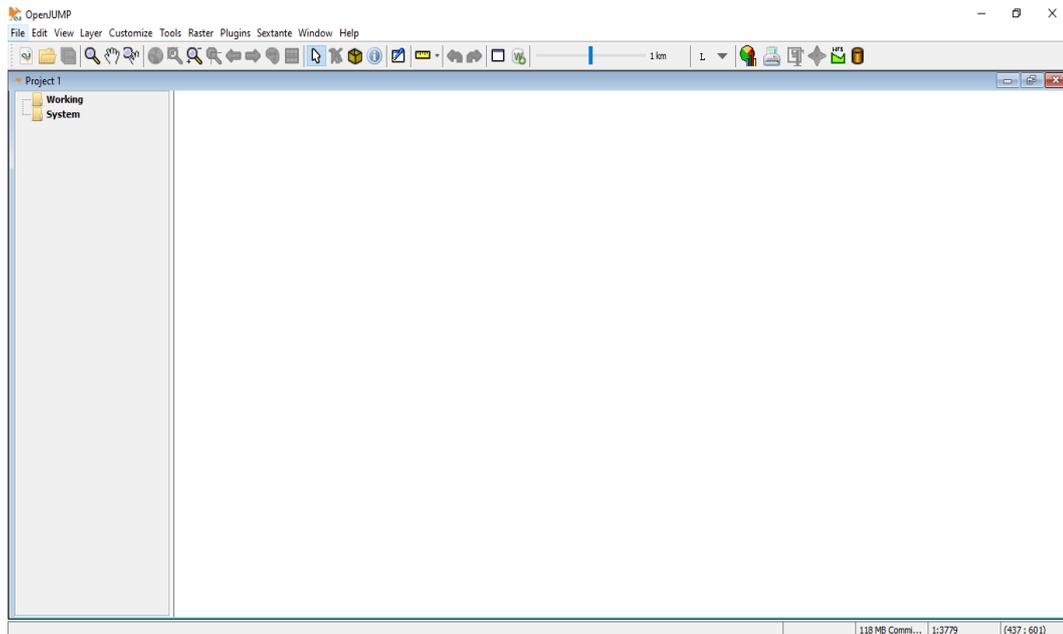
node bergerak pada konstruksi jalan yang *fixed* seperti kereta dan bus.

2.2.6 OpenJUMP

OpenJUMP adalah Sistem Informasi Geografis (GIS), yang telah dikembangkan awalnya oleh kedua Perusahaan yaitu Kanada Solusi Vivid dan Refraksi. Kedua perusahaan melakukan penelitian dengan nama *JUMP*. Nama *JUMP* Adalah kependekan dari *Unified Mapping Platform*. The "J" menunjuk pada bahasa pemrograman yang digunakan yaitu "Java". Makna kata "Open" adalah untuk "Open Source", yang berarti kode *quell* dapat diakses oleh semua orang. *OpenJUMP* mendasari GNU *General Public License* dan dapat di *maintenance* oleh pemrogram di dunia.

➤ Fitur utama *OpenJUMP* adalah:

- GIS berbasis vektor
- mendasari GNU General Public License
<http://www.gnu.org/licenses/licenses.html#GPL>
- didasarkan pada standar Open GIS <http://www.opengeospatial.org/>
- open *source* (kode pemrograman yang mudah diakses)
- dapat dengan mudah diperluas oleh Plugins
- Mudah digunakan pada fungsi editing dan analisis
- mendukung beberapa bahasa (Inggris, Prancis, PortuGIS, Spanyol, Jerman)



Gambar 2.10 Tampilan Utama Program *OpenJUMP*

2.2.7 Pengujian

Pengujian ini menggunakan beberapa parameter pengujian yang akan digunakan sebagai bahan penunjang dalam menganalisa data pada pengujian. Beberapa parameter yang dapat digunakan dalam analisis data yaitu sebagai berikut ini :

1. *Delivery Probability*
2. *Overhead Ratio*
3. *Average Latency*

Kemudian penjelasan lebih rinci mengenai beberapa parameter pengujian yang dapat digunakan dalam tahap pembahasan dan analisis data akan dijelaskan sebagai berikut ini :

1. Delivery Probability

Delivery Probability adalah rasio perbandingan antara jumlah pesan yang dikirim hingga sampai ke *node* tujuan dibagi dengan jumlah pesan yang dihasilkan oleh *node* sumber, salinan duplikat dari pesan yang sama tidak dihitung (Bozed, dkk, 2016).

$$Delivery\ Prob = \frac{Delivery\ Message}{Message\ Create}$$

Persamaan 2.1 perhitungan rumus *delivery probability* (Mehto & Chawla, 2014)

2. Overhead Ratio

Overhead Ratio merupakan berapa banyak pesan berlebihan untuk pengiriman satu pesan hingga sampai ke tujuan. *Overhead* dalam mentransmisikan pesan menghasilkan konsumsi energi tambahan. *Overhead Ratio* mencerminkan *transmission cost* atau biaya transmisi dalam jaringan (Bozed, dkk, 2016). Rumus Perhitungan pada *Overhead Ratio* sebagai berikut :

$$MessageOverhead = \frac{(messageRelay - messageDelivery)}{messageDelivery}$$

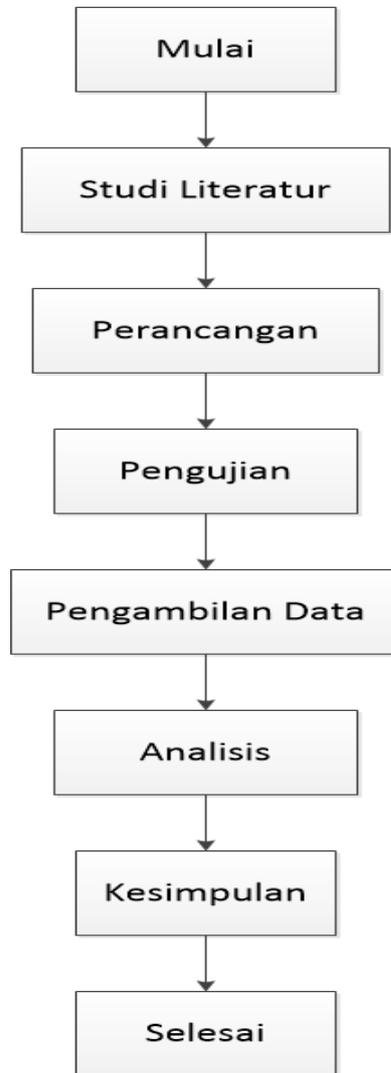
Persamaan 2.2 Rumus Perhitungan Pada *Overhead Ratio* (Barzan, A., dkk, 2013)

3. Average Latency

Average Latency adalah waktu rata-rata yang dibutuhkan oleh sebuah pesan yang terkirim untuk mencapai tujuannya (Barzan, A., dkk, 2013).

BAB 3 METODOLOGI

Pada bab ini menjelaskan tentang tahapan-tahapan yang akan digunakan dalam penelitian, dengan beberapa metode tahapan yang akan digambarkan pada diagram alir dibawah ini.



Gambar 3.1 Diagram Alir Metodologi

3.1 Studi Literatur

Pada studi literatur ini dilakukannya untuk bertujuan mengolah konsep yang direncanakan untuk proses penelitian dalam membangun program ini dan menentukan objek penelitian yang tepat dan akan digunakan sebagai kajian peneliti. Mempelajari teori pendukung dari berbagai macam jurnal dan beberapa

penelitian sebelumnya yang terkait dengan penelitian yang akan dilakukan oleh peneliti. Penelitian yang akan di gunakan mengenai menganalisis protokol *routing* DTN seperti *Epidemic* dan *Spray And Wait* dan aplikasi simulator berupa *The One simulator* yang akan digunakan sebagai studi literatur ini.

3.2 Perancangan

3.2.1 Analisis Kebutuhan

Pada Analisis kebutuhan ini berlaku untuk menentukan apa yang dibutuhkan dan membangun pada program yang dibuat. Tahapan ini menjelaskan tentang batasan kebutuhan pada program. Pada analisis kebutuhan dibagi menjadi dua kebutuhan yaitu *hardware* (Perangkat Keras) dan *software* (Perangkat Lunak).

A. Perangkat Keras

Perangkat Keras dapat membantu pada proses penelitian tugas akhir dengan kemampuannya dalam menjalankan berbagai aplikasi yang mumpuni. Berikut ini adalah spesifikasi perangkat keras yang dibutuhkan dalam mensupport untuk tugas akhir ini :

- ❖ Laptop : Asus S46cm
- ❖ Ram : 8,00 GB
- ❖ Processor : Intel (R) Core(TM) i7-3517U CPU @ 2.40 Ghz
- ❖ Sistem Operasi : Windows 10 Enterprise 64 Bit

B. Perangkat Lunak

Kebutuhan Perangkat Lunak yang akan digunakan adalah menggunakan aplikasi Simulator yaitu *The One* versi 1.4.1. Peneliti melakukan skenario pengujian menggunakan Aplikasi *The One simulator* Untuk Memodelkan Skenario agar dapat divisualisasikan rutenya. *The One simulator* adalah aplikasi yang dapat mengolah data dari map asli dengan ekstensi gambar berupa (.jpg/.png) yang akan di jadikan dengan format ekstensi .WKT (*Well Known Text*) agar dapat dikenali oleh aplikasi simulator *The One* Dan dengan dibantu dengan aplikasi berbasis *GIS* (*Geographic Information System*) seperti *OpenJump* versi 1.9.1 rev.4840 *PLUS* dalam mensupport dalam digitalisasi map. Dan untuk mengambil data map dengan skala yang tepat yang akan digunakan pada *OpenJump* dengan menerapkan website *GIS* yang telah ada seperti *GoogleMap* dan *openstreetmap.org*.

3.2.2 Perancangan Sistem

Perancangan sistem akan menggambarkan keseluruhan sistem secara umum dan bagaimana perancangan sistem dapat dibuat. Perancangan sistem ini terdiri dari rancangan topologi dan skenario sistem yang akan dianalisa. Dibawah ini akan dijelaskan mengenai tahapan-tahapan dalam perancangan sistem berikut ini :

3.2.2.1 Rancangan Skenario Sistem

Penentuan rancangan skenario jaringan yang dipakai pada pengujian ini terdapat dua skenario untuk pengiriman pesan dengan menggunakan dua protokol algoritma *routing* DTN seperti *Epidemic* dan *Spray And Wait* dengan menggunakan tiga ukuran pesan yang bervariasi dari 300 MB, 600 MB, 1,2 GB. Transportasi yang akan digunakan sebagai penyampaian pesan antar kota adalah sebuah mobil sebagai *node* yang bergerak ke daerah kota yang sulit dijangkau jaringan. Dengan menggunakan *node* yang bergerak berjumlah 6 *node* dari dua group yang sudah dibuat pada rancangan skenario sistem dengan memiliki spesifikasi kecepatan tiap *node* sebesar 35 km/jam. Pola mobilitas *node* yang akan digunakan pada skenario sistem adalah *Map Based Movement*, *Open Shortest Path* dan *Map Route Movement*. Rancangan skenario rute sebagai berikut:

1. Skenario satu, Skenario pengiriman pesan jarak dekat pada rute Blitar menuju Tulung Agung.
2. Skenario kedua, Skenario pengiriman pesan jarak jauh pada rute Tulung Agung menuju Trenggalek.
3. Skenario ketiga, Skenario pengiriman pesan jarak berbeda tujuan pada rute Trenggalek ke Tulung Agung, dan Blitar.
4. Skenario Masing- Masing akan dijalankan pada waktu yang sesuai dengan jarak asli perjalanan. Waktu yang digunakan pada masing-masing skenario selama simulasi yaitu 45 menit untuk Blitar ke Tulung Agung , 65 menit untuk Tulung agung ke Trenggalek dan 85 menit untuk Trenggalek ke Tulung Agung dan Blitar.

Pada penelitian ini akan menguji bagaimana kinerja kedua protokol *routing* yang berbeda dapat melakukan sebuah pengiriman pesan antar *node* sehingga dapat ditemukannya solusi protokol *routing* yang *optimal* pada kondisi jaringan yang tidak stabil saat pengiriman pesan berlangsung.

3.3 Pengujian

Pengujian pada penelitian tugas akhir ini dengan menggunakan aplikasi simulator yaitu *The One simulator* untuk mendapatkan hasil yang tepat. Dan akan di analisa datanya untuk dibuat penarikan kesimpulan pada akhir penelitian ini berdasarkan dari beberapa rancangan yang telah dibuat.

Pengujian dilakukan setelah proses perancangan skenario simulasi. Untuk memastikan sistem dapat berjalan dengan baik. Pengujian dilakukan dengan menerapkan waktu yang berbeda-beda yang telah ditentukan kepada setiap skenario dengan waktu yang telah ditentukan pada perancangan sistem yang telah dibuat sebelumnya. Dengan melakukan pengujian terhadap parameter-parameter berdasarkan beberapa tipe mobilitasnya untuk mengetahui kinerja protokol *routing Epidemic* dan *Spray And Wait*, parameter antara lain :

- *Delivery Probability*
- *Average Latency*
- *Overhead Ratio*

3.4 Pengumpulan dan Pengambilan Data

Pengumpulan data dan pengambilan data akan dilakukan setelah pengujian telah dilakukan kemudian akan didapatkan data dari hasil pengujian yang telah digunakan pada aplikasi simulator, setelah itu data – data hasil dari pengujian yang diperoleh akan dimasukkan kedalam suatu tabel data, data tersebut akan dimodifikasi dan ditampilkan menjadi data statistik, Sehingga proses analisis data tersebut dapat dilakukan untuk dianalisa.

3.5 Analisis

Pada bagian analisis ini akan dijelaskan mengenai perbandingan dari beberapa model protokol *routing* yang diuji dan sehingga didapatkan hasil analisis yang tepat dalam penarikan kesimpulan dari beberapa skenario pengujian yang telah dilakukan. Sehingga didapatkan hasil analisis dari pengujian algoritma *routing* yang terbaik dan optimal berdasarkan dari segi kinerja *routing* dan mobilitasnya dalam proses pengiriman data.

3.6 Kesimpulan

Penarikan kesimpulan akan dilakukan ketika pengujian yang dilakukan telah selesai dan dapat diketahui hasil akhir analisa dari perbandingan kedua protokol *routing Epidemic* dan *Spray And Wait* berdasarkan tipe beberapa mobilitasnya.

BAB 4 PERANCANGAN SISTEM

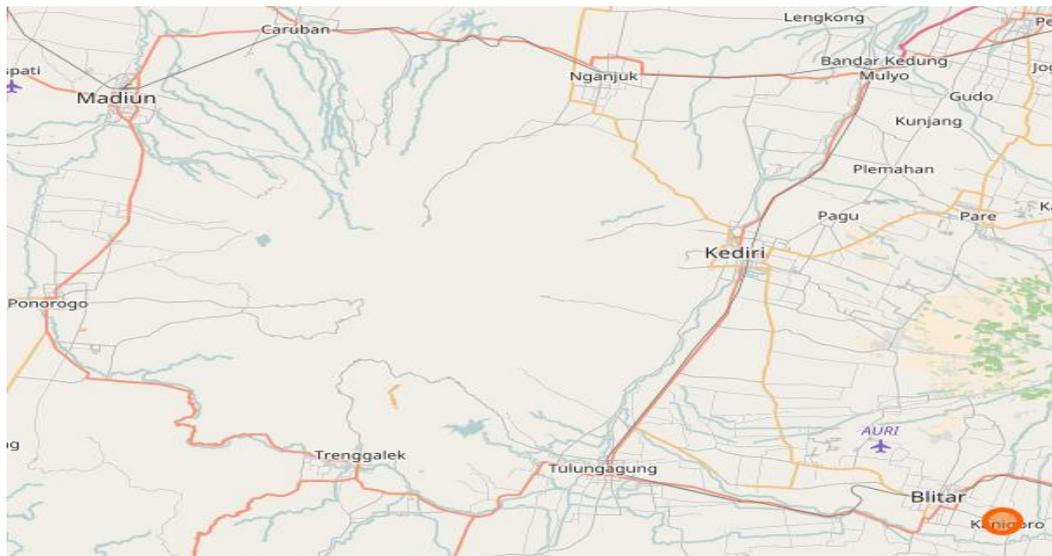
Pada bab ini akan menjelaskan terkait beberapa tahapan yang dilakukan dalam pengujian pada kinerja algoritma *routing* DTN seperti *Epidemic* dan *Spray And Wait* berdasarkan pada perancangan skenario sistem yang telah dibuat sebelumnya.

4.1 Topologi Arsitektur Jaringan

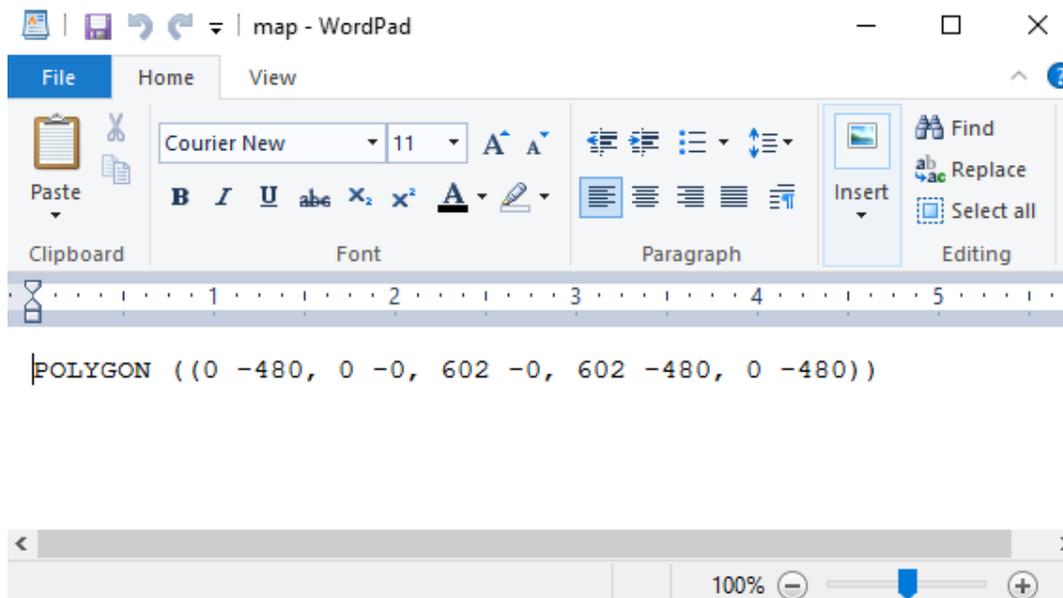
Topologi Arsitektur jaringan yang akan digunakan adalah sebagai berikut :

4.1.1 Pengambilan Data Map Menggunakan *OpenJUMP*

Pengambilan jalur dengan menggunakan peta dari kota Blitar, Tulung Agung, dan Trenggalek pada Gambar 4.1 dengan menerapkan Website GIS seperti *openstreetmap.org*. Pengambilan data peta dari website tersebut dilakukan dengan cara *crop* atau memotong gambar pada peta yang ada di website *openstreetmap.org* tersebut, Kemudian data akan diubah format ekstensi datanya dari bentuk *.png* menjadi bentuk *.wkt* (*Well Known Text*) seperti pada gambar 4.2 dengan menggunakan bantuan aplikasi *OpenJump* agar data rute map tersebut dapat dikenali saat dimasukkan ke dalam aplikasi *One Simulator*.



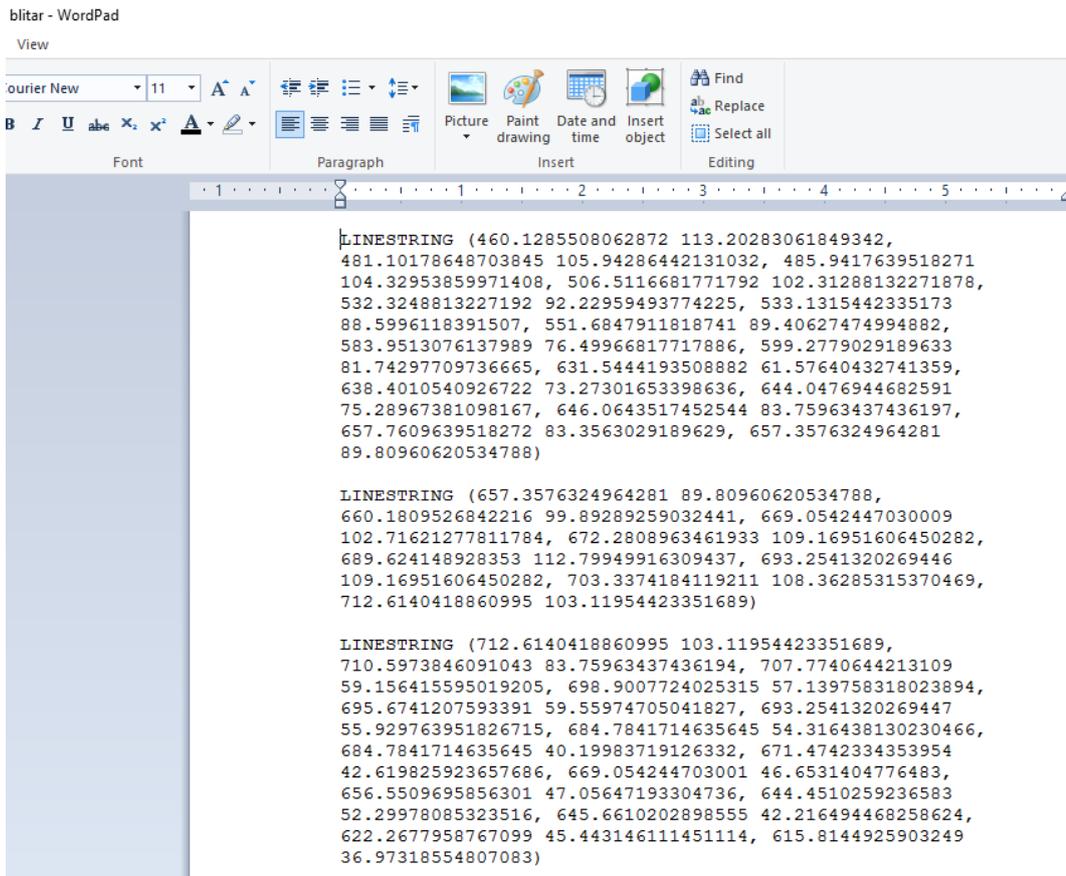
Gambar 4.1 Tampilan Map Ketiga kota (Blitar, Tulung Agung, dan Trenggalek)



Gambar 4.2 Peta setelah diubah format menjadi .wkt

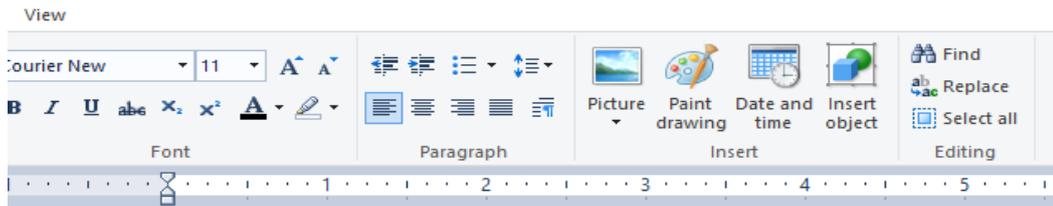
Hasil perubahan peta oleh *OpenJump* yang akan digunakan pada Aplikasi simulator *The one* dapat terlihat pada Gambar 4.2. Kemudian setelah peta telah dibuat pada *OpenJump*, maka akan dilanjutkan dengan proses pembuatan jalur yang akan digunakan sebagai rute perjalanan mobil dan akan dijelaskan pada Gambar 4.3.

Untuk pembuatan jalur map yang akan digunakan sebagai rute perjalanan dari *node* yang bergerak dapat digambarkan kedalam aplikasi *OpenJump* sesuai jalur yang ada pada map, setelah itu hasil data pembuatan jalur akan disimpan ke dalam bentuk format .wkt yang nantinya akan dimasukkan ke dalam aplikasi simulator *The One*. Isi dari data jalur map yang telah dibuat sebelumnya yang bereksistensi format .wkt merupakan sebuah *linestring*. *Linestring* adalah *curve* dengan interpolasi linier antara titik-titik. Hasil isi data *linestring* terlihat pada gambar 4.3, gambar 4.4, gambar 4.5 dan gambar 4.6.

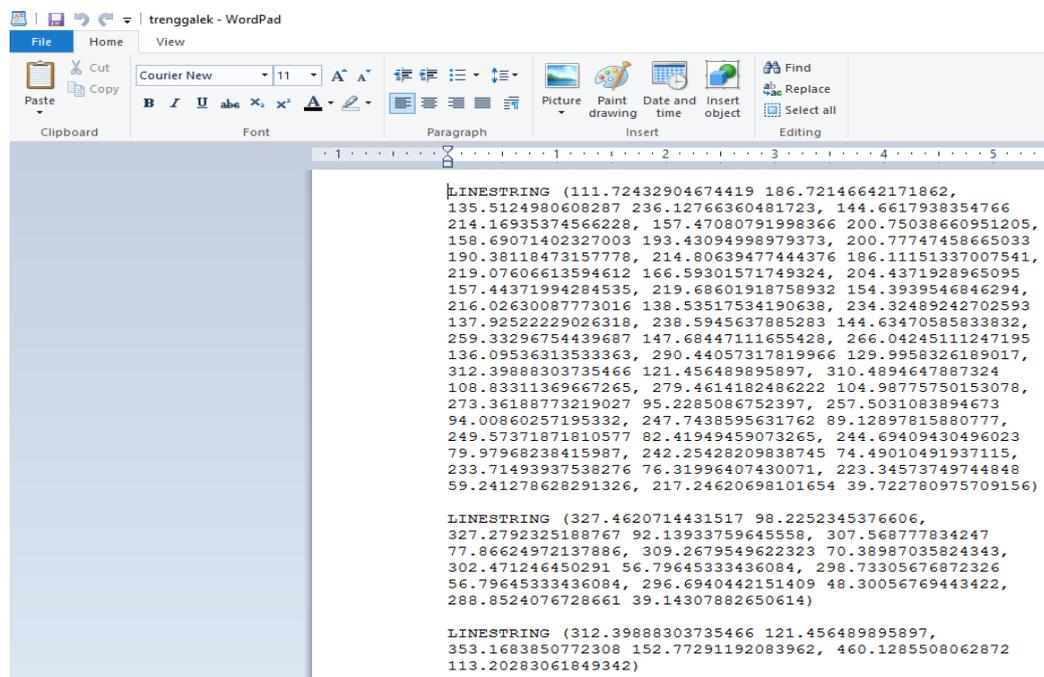


Gambar 4.3 Hasil Linestring jalur kota Blitar.wkt

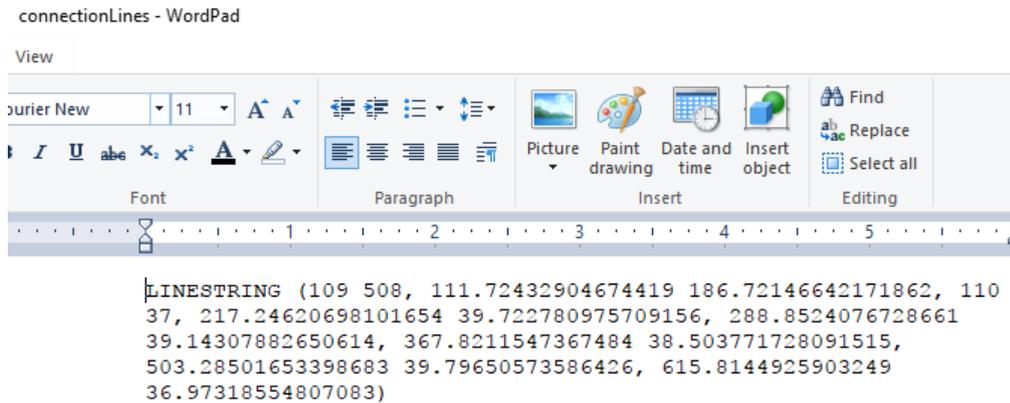
Pada Gambar 4.3 menunjukkan hasil jalur dari rute Blitar menuju Tulung Agung. Pada Gambar 4.4 menunjukkan hasil jalur dari rute Tulung Agung menuju Trenggalek, Gambar 4.5 menunjukkan hasil jalur dari rute Trenggalek, dan Gambar 4.6 sebagai koneksi penghubung tambahan antar jalur dari rute Blitar-Trenggalek.



Gambar 4. 4 hasil *Linestring* Jalur kota Tulung Agung.wkt



Gambar 4.5 Hasil *Linestring* Jalur kota Trenggalek.wkt



Gambar 4.6 Hasil *Linestring ConnectionLines.wkt*

4.1.2 Langkah-Langkah Konfigurasi Aplikasi The One Simulator

A. Instal *The One Simulator*

Langkah utama yang perlu dilakukan adalah dengan cara menginstall aplikasi dari *The One simulator* yang telah didownload dari website <https://akeranen.github.io/the-one/>. Download aplikasi *The One simulator* dengan versi one_1.4.1.rar atau versi keluaran terbaru. Aplikasi *The One simulator* telah mendukung sistem operasi berbasis windows, Linux dan Mac OS X.

Penulis menggunakan aplikasi *The One simulator* dengan versi 1.4.1 yang akan digunakan untuk pengujian skenario. Setelah aplikasi berhasil di download, kemudian ekstrak aplikasi yang berformat one_1.4.1.rar tersebut dan install *The One Simulator*.

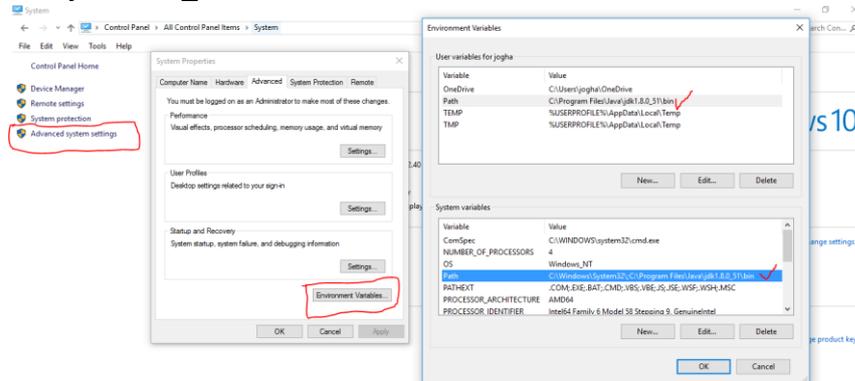
B. Setting variabel java

Untuk dapat menjalankan aplikasi *The One simulator* ada syarat yang dibutuhkan agar dapat berjalannya aplikasi tersebut. Dengan cara mendownload program pendukung seperti JDK (*Java Development Kit*) dengan versi 1.8.0_51 atau versi terbaru. Kemudian install dan setting lokasi *path* JDK yang akan digunakan untuk mensupport dan menjalankan aplikasi *The One simulator*.

Langkah-langkahnya ada pada Gambar 4.7 dengan cara sebagai berikut :

1. Klik kanan pada *My computer* setelah itu klik kiri pada *properties*
2. Klik *Advanced System Settings* kemudian pilih tab *Advanced* setelah itu klik kiri pada *Environment Variables*. Kemudian pilih *variable path* dan edit lokasi *path* dengan memasukkan lokasi JDK yang telah diinstall pada *program files*.

3. Masukkan lokasi penyimpanan path JDK yang akan dipasang. Pada Gambar 4.7 terlihat bahwa penyimpanan *path* JDK terletak pada lokasi C:\Program Files\Java\jdk 1.8.0_51\bin



Gambar 4.7 Setting Java Variabel

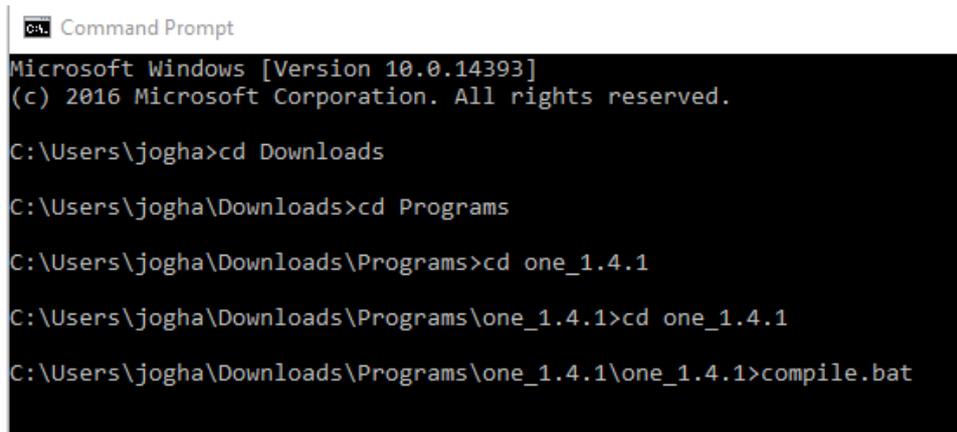
C. Compile dan Running The One simulator

Setelah melakukan setting lokasi *path* jdk, maka kemudian dapat mengcompile Aplikasi *The One Simulator*. Aplikasi simulator ini dibuat dengan menggunakan Bahasa pemrograman *open source* berbasis *java*, sehingga setelah melakukan *setting* dan *compile* aplikasi simulator maka program *The One simulator* siap untuk digunakan untuk simulasi.

Gambar 4.8 Isi dari folder *The One Simulator*

Cara menjalankan Aplikasi *The One simulator* dapat menggunakan dua cara:

- a. Double Klik kiri pada *compile.bat* secara langsung
- b. Dapat menggunakan sebuah terminal *Command Prompt* yang ada di bawaan *windows* kemudian masuk ke lokasi direktori penyimpanan aplikasi *The One simulator* dengan perintah *syntax cd (Change Directory)*. Setelah itu tuliskan perintah *compile.bat* lalu tekan enter. Untuk Lebih detailnya terlihat pada gambar 4.9.



```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\jogha>cd Downloads

C:\Users\jogha\Downloads>cd Programs

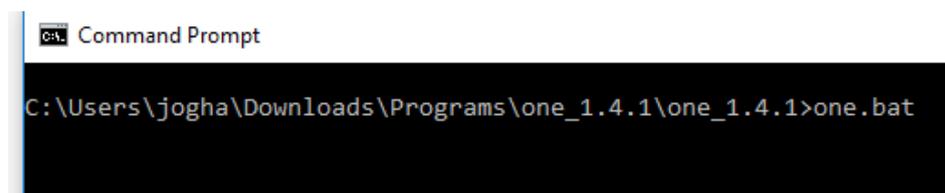
C:\Users\jogha\Downloads\Programs>cd one_1.4.1

C:\Users\jogha\Downloads\Programs\one_1.4.1>cd one_1.4.1

C:\Users\jogha\Downloads\Programs\one_1.4.1\one_1.4.1>compile.bat
```

Gambar 4.9 Proses Compile Aplikasi pada *The One Simulator*

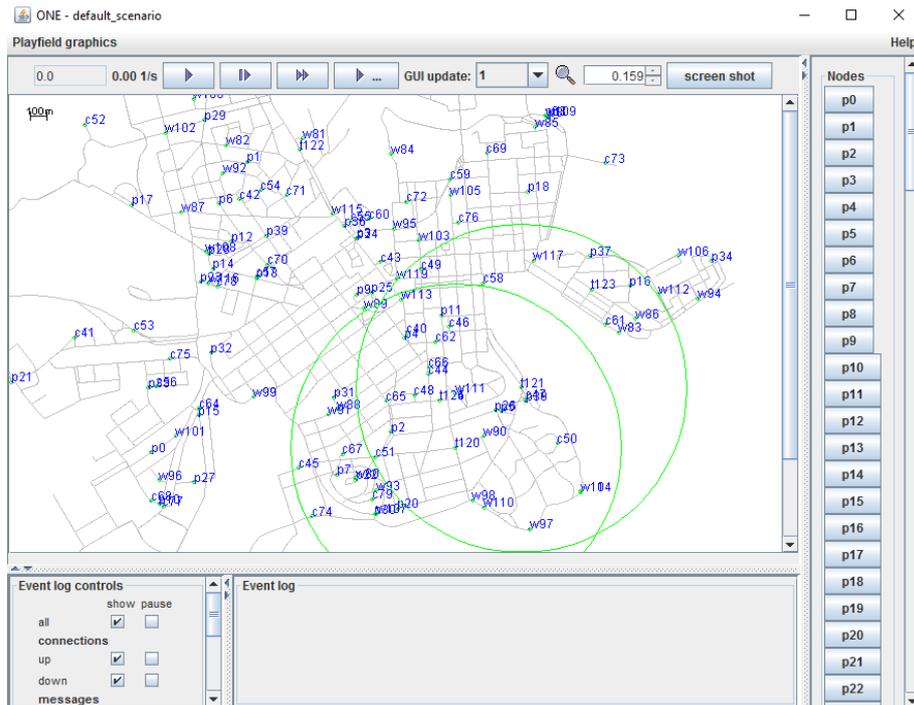
- c. Setelah proses *compile* berhasil dilakukan . Kemudian coba menjalankan aplikasi *The One simulator* dengan perintah *one.bat* pada *command prompt* seperti pada gambar 4.10. Setelah itu jika peta Helsinki berhasil ditampilkan seperti pada gambar 4.11 maka aplikasi *The One simulator* siap untuk dijalankan. Peta Helsinki merupakan contoh tampilan peta yang ada pada bawaan Aplikasi *The One simulator* dengan tampilan *default settings* pada simulator.



```
Command Prompt

C:\Users\jogha\Downloads\Programs\one_1.4.1\one_1.4.1>one.bat
```

Gambar 4.10 Proses *Compile* Aplikasi pada *The One Simulator*



Gambar 4.11 Tampilan *default setting* pada skenario Peta Helsinki

4.2 Skenario Pengujian

Proses pembuatan skenario pengujian pada simulasi dibuat dengan berdasarkan rute perjalanan dari ketiga kota yang kemudian dibagi dengan dua proses skenario. Skenario Pertama dengan rute jarak dekat yaitu rute perjalanan dari Kota Blitar ke Kota Tulung Agung dengan menggunakan transportasi 6 buah mobil dengan masing – masing *node* mengirimkan paket data berupa pesan. Pesan yang dikirimkan *node* memiliki ukuran pesan yang bervariasi yaitu 300 MB, 600 MB, 1,2 GB. Waktu pengiriman pesan sesuai dengan berdasarkan waktu jarak tempuh perjalanan asli selama 45 menit yang ditempuh dari kota blitar menuju Tulung Agung. Skenario kedua dengan rute jarak jauh yaitu rute perjalanan yang ditempuh dari kota Tulung Agung menuju Tulung Agung dengan menggunakan jumlah transportasi *node* yang sama dengan skenario pertama yang berjumlah 6 buah mobil yang masing-masing mempunyai ukuran pesan yang bervariasi yaitu 300 MB, 600 MB, 1,2 GB, dengan waktu perjalanan yang ditempuh sebanyak 65 menit dan Skenario ketiga dengan rute jarak yang berbeda tujuan yaitu rute perjalanan yang ditempuh dari kota Trenggalek ke kota lainnya yaitu Tulung Agung dan Blitar dengan menggunakan transportasi *node* yang sama dengan skenario pertama yang berjumlah 6 buah mobil yang masing – masing mempunyai ukuran pesan yang bervariasi yaitu 300 MB, 600 MB, dan 1,2 GB, dengan waktu perjalanan sebanyak 85 menit. Seluruh skenario pengujian dengan menggunakan waktu jarak tempuh sebesar 46 menit, 65 menit, dan 85 menit yang bertujuan untuk dapat mengetahui hasil perbandingan performansi yang diperoleh dengan menggunakan waktu yang bervariasi yang sesuai waktunya dengan jarak tempuh asli perjalanan dari suatu node sumber ke node

tujuan dan juga akan dapat melihat apakah ada suatu paket yang di drop atau gagal dalam proses pengiriman dengan waktu yang bervariasi. Skenario pengujian dengan menggunakan jumlah ukuran pesan sebesar 300 MB, 600 MB, dan 1,2 GB dengan waktu jarak tempuh dalam simulasi pengujian bertujuan untuk mengukur berapa jumlah pesan yang akan sampai ke tujuan dan apakah dapat mempengaruhi tingkat proses keberhasilan atau *Delivery Probability* pengiriman dengan menggunakan protokol *routing Epidemic* dan *Spray And Wait* berdasarkan beberapa mobilitasnya seperti *Map Based Movement*, *Open Shortest Path*, dan *Map Route Movement* dengan waktu tempuh dan ukuran pesan yang bervariasi. Skenario pengujian simulasi dapat dilihat lebih detail pada tabel 4.1, tabel 4.2 dan tabel 4.3.

Tabel 4.1 Skenario Pengujian Simulasi Pertama

Rute	Jumlah Node	Ukuran Pesan	Waktu Simulasi
1	6	300 MB	45 Menit
		600 MB	
		1,2 GB	

Tabel 4.2 Skenario Pengujian Simulasi Kedua

Rute	Jumlah Node	Ukuran Pesan	Waktu Simulasi
2	6	300 MB	65 Menit
		600 MB	
		1,2 GB	

Tabel 4.3 Skenario Pengujian Simulasi Ketiga

Rute	Jumlah Node	Ukuran Pesan	Waktu Simulasi
3	6	300 MB	85 Menit
		600 MB	
		1,2 GB	

4.2.1 Konfigurasi *default settings*

Hal yang perlu dilakukan dalam proses simulasi dalam pengujian adalah dengan mengkonfigurasi *default settings* sebagai berikut :

A. Setting konfigurasi Skenario Waktu pengiriman Pesan

Konfigurasi waktu pengiriman pesan untuk Skenario dari sumber ke tujuan dapat dilakukan dengan menggunakan sintax yang ada pada *The One simulator* dengan sintax *Scenario.endTime* yang ditunjukkan pada Gambar 4.12. Pada konfigurasi tersebut difungsikan untuk mengatur berapa lama waktu skenario akan berjalan, satuan waktu yang digunakan untuk skenario adalah satuan waktu detik (s). Sehingga waktu simulasi untuk pengiriman pesan dapat diatur sesuai dengan kebutuhan pada skenario simulasi yang telah dibuat.

```
# Skenario waktu Pengiriman
# 43200s == 12h
# 86400s == 24h
# 45 menit = 2700s (SKENARIO UJI 1)
# 65 menit = 3900s (SKENARIO UJI 2)
# 85 menit = 5100s (SKENARIO UJI 3)
Scenario.endTime = 2700
```

Gambar 4.12 Settingan Konfigurasi waktu pengiriman Pesan

B. Setting Konfigurasi Pembuatan Jumlah Group dan *Node* bergerak

Dalam membuat grup yang digunakan dalam proses pengiriman pesan untuk setiap grup dengan menggunakan *syntax Scenario.nrofHostGroups* dapat dilihat pada gambar 4.13.

Perintah *Scenario.nrofHostGroups* merupakan *syntax* untuk membuat jumlah host grup yang akan digunakan pada skenario simulasi yang telah dibuat sebelumnya. Pada skenario ini , grup yang akan di uji berjumlah 4 grup dan setiap satu grup mempunyai *node* yang bergerak atau kurir berjumlah 6 grup yang akan diatur oleh *syntax Group.nrofHosts* , dan perintah *syntax* dapat dilihat pada gambar 4.14. Sehingga settingan konfigurasi untuk *node* yang bergerak pada *Group.nrofHosts* berjumlah 6.

```
!---Banyaknya Jumlah Host Group yang diinisialisasi pada simulator -
Scenario.nrofHostGroups = 4
```

Gambar 4.13 Settingan Konfigurasi Jumlah *Host Group*

```
---jumlah host node yang bergerak----
Group.nrofHosts = 6
```

Gambar 4.14 Settingan Konfigurasi Pada Node Yang Bergerak

C. Konfigurasi Model Mobilitas Node (Pergerakan Node)

Dalam mengkonfigurasi pergerakan keseluruhan *node* menggunakan perintah *syntax Group.movementModel*. Model pergerakan *node* menggunakan tiga pergerakan yaitu *MapBasedMovement*, *Open Shortest Path* dan *Map Route Movement* pada *The One simulator*.

```
#-----Mode pergerakan node pada protokol Routing---
Group.movementModel = MapBasedMovement
Group.movementModel = MapRouteMovement
Group.movementModel = ShortestPathMapBasedMovement
|
```

Gambar 4.15 Konfigurasi Pergerakan Node

D. Inisialisasi group

Hal terpenting dalam proses Inisialisasi *Group* adalah dengan menginisialisasi jumlah *Group* yang akan dibuat dalam simulasi. Kemudian, memberikan sebuah atribut kepada setiap *node* yang berbeda yang tujuannya agar dapat membedakan pergerakan antara *group host* dan *group node*.

```
#skenario rute 1
Group1.groupID= Blt_
Group1.routeFile = data/Mapping/blitar.wkt
Group1.okMaps = 3
Group1.speed = 1, 1|
```

Gambar 4.16 Contoh Settingan Konfigurasi Group

E. Konfigurasi Ukuran pesan pada Simulasi

Dalam Mengkonfigurasi settingan ukuran pesan dapat digunakan pada skenario pengujian dengan mengatur ukuran pesan dengan sesuai kebutuhan pada skenario pengujian.

Pada tahap pengujian, ukuran pesan yang akan digunakan dalam uji coba adalah sebesar 300 MB, 600 MB, 1,2 GB. Untuk dapat mengetahui banyaknya pesan yang telah dibuat dan dapat mengirimkan ke *node* tujuan dengan berdasarkan ukuran pesan yang besar pada tiap masing-masing skenario. Untuk tata cara penulisan *syntax* ukuran pesan terletak pada konfigurasi *default setttings* pada Simulator dengan menggunakan *syntax event.size*.

pada *syntax event.size*, satuan ukuran pesan menggunakan satuan data *KiloByte* (KB). Jika pesan yang dikirimkan dengan satuan ukuran data *MegaByte* (MB) dan *GigaByte* (GB), maka konversi ukuran pesan terlebih dahulu diubah satuannya kedalam ukuran data *kilobyte* (KB).

```
#--Konversi Ukuran Pesan----
#300MB = 300000 kB (uji 1)
#600MB = 600000 kB (uji 2)
#1,2GB = 1200000 KB (uji 3)|
Events1.size = 1200000
```

Gambar 4.17 konfigurasi Konversi Ukuran Pesan

4.3 Pengujian Skenario

Pada Pengujian Skenario akan dilakukan dari tiga tahap perancangan yang telah dibuat sebelumnya berdasarkan beberapa tipe mobilitasnya seperti *Map Based Movement*, *Open Shortest Path*, dan *Map Route Movement* dengan menggunakan aplikasi simulator yaitu *The One Simulator*.

4.3.1 Map Based Movement

4.3.1.1 Pengujian Skenario Pertama

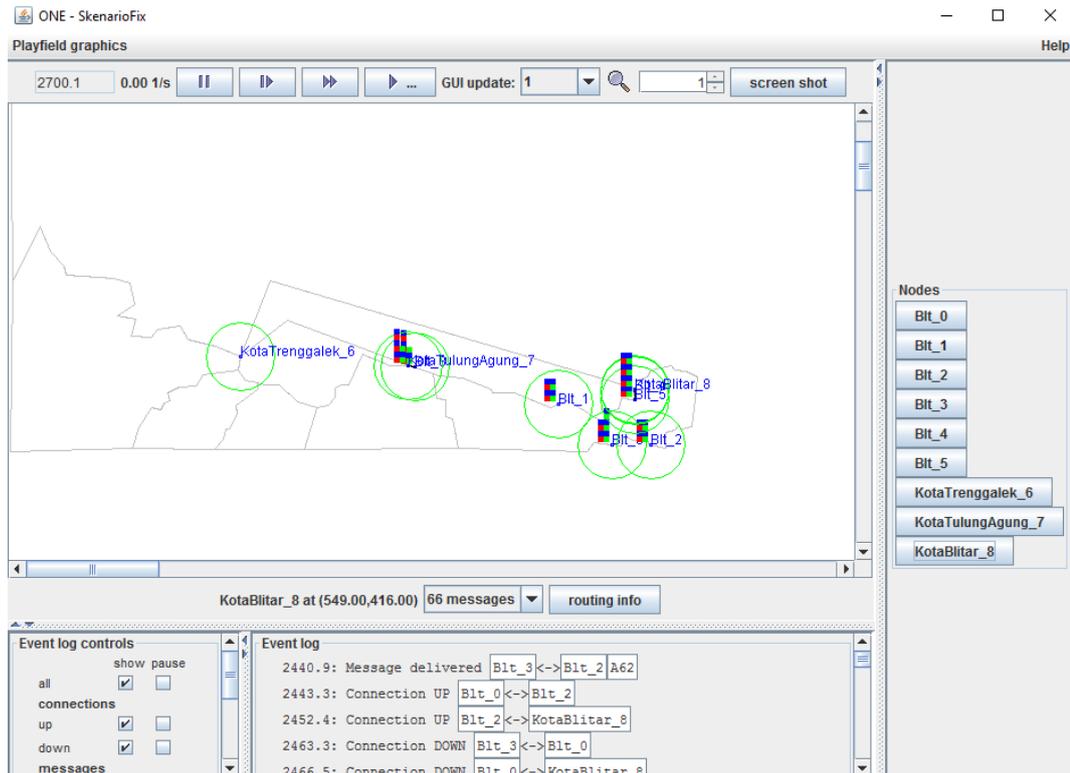
Pengujian pada skenario pertama bertujuan untuk dapat membandingkan *performansi* terhadap kinerja kedua protokol *routing* seperti *Epidemic* dan *Spray And Wait* menggunakan pergerakan node yaitu *Map Based Movement* dengan cara menjalankan skenario pertama sebagai acuan pada parameter pengujiannya. Pengujian yang dilakukan pada skenario pertama dengan waktu pengujian simulasi selama 45 menit sesuai dengan waktu tempuh yang sebenarnya untuk melakukan perjalanan jarak dekat antar kota Blitar ke Tulung Agung.

Tabel 4.4 Skenario Pengujian Pertama

Skenario	Waktu	Protokol <i>Routing</i>	Ukuran Pesan		
1	45 Menit	<i>Epidemic</i>	300 MB	600 MB	1,2 GB
		<i>Spray And Wait</i>	300 MB	600 MB	1,2 GB

A. Pengujian Pertama Pada Protokol *Routing Epidemic*

Pengujian Pertama dengan menggunakan protokol *routing Epidemic* dengan waktu simulasi yang sama selama 45 menit dan menggunakan ukuran pesan 300 MB, 600 MB dan 1,2 GB.

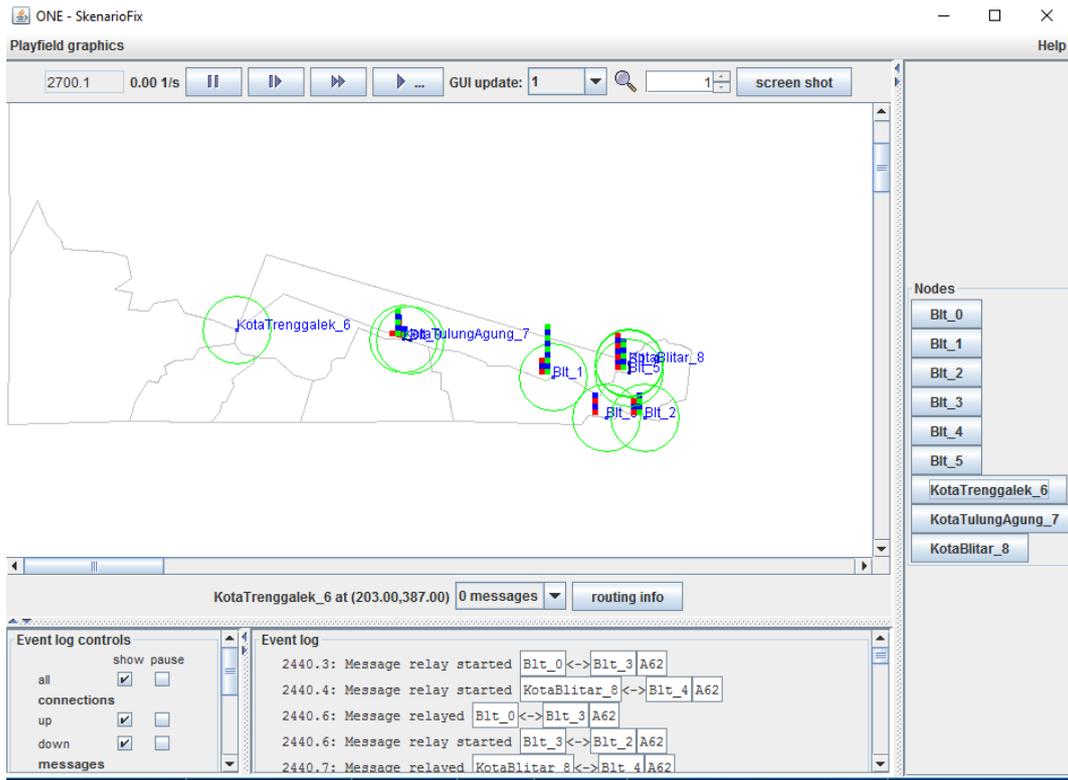


Gambar 4.18 Simulasi Pertama Protokol *Routing Epidemic*

Pada Gambar 4.18 Menunjukkan bahwa tahap simulasi pengujian skenario pertama pada Protokol *Routing Epidemic* telah sukses dilakukan. Tampilan Hasil pada Screenshot Gambar 4.18 terlihat bahwa *node* yang bergerak (kurir) yang bertindak membawa pesan berjumlah 6 buah yaitu, Blt_0, Blt_1, Blt_2, Blt_3, Blt_4, dan Blt_5. Dan *node* statis atau tetap berjumlah 3 buah yaitu , KotaTrenggalek_6, KotaTulungAgung_7, dan KotaBlitar_8.

B. Pengujian Pertama Pada Protokol *Routing Spray And Wait*

Pengujian pada skenario pertama ini dengan menggunakan protokol *routing Spray And Wait* dengan waktu pengujian simulasi yang sama selama 45 menit dengan ukuran pesan 300 MB, 600 MB, 1,2 GB.



Gambar 4.19 Simulasi Pertama Protokol *Routing Spray And Wait*

Pada Gambar 4.19 Menunjukkan bahwa tahap simulasi pengujian skenario pertama pada Protokol *Routing Spray And Wait* telah sukses dilakukan.

4.3.1.2 Pengujian Skenario kedua

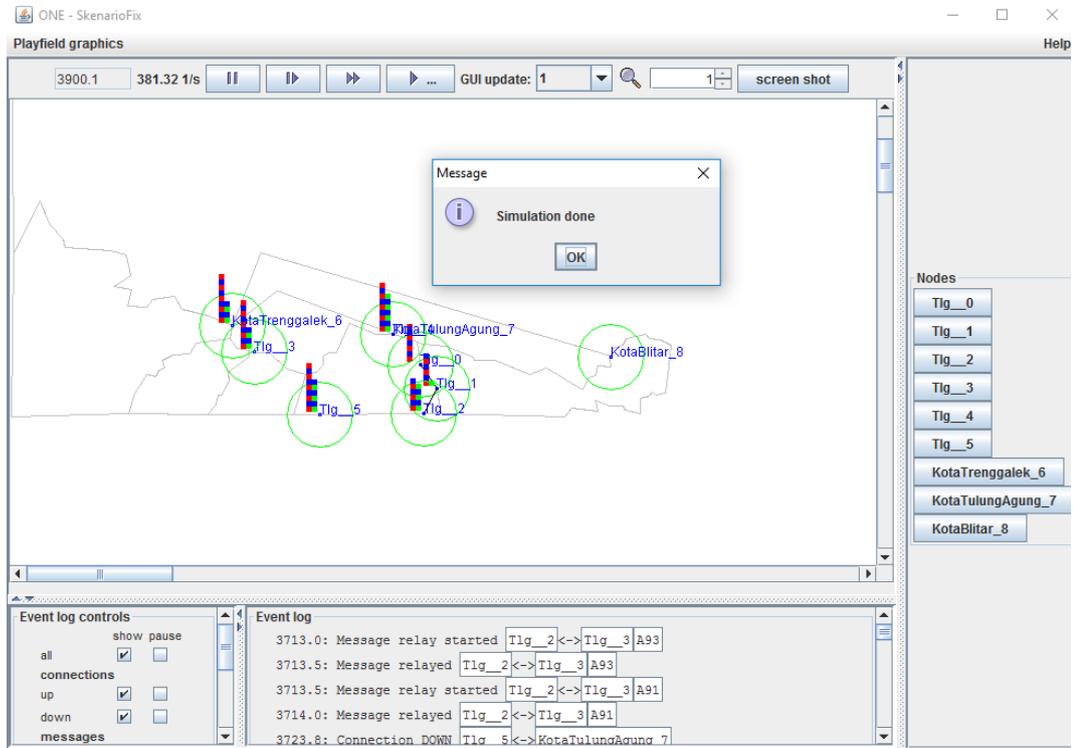
Pengujian pada skenario kedua bertujuan untuk dapat membandingkan *performansi* terhadap kinerja kedua protokol *routing* seperti *Epidemic* dan *Spray And Wait* menggunakan pergerakan node yaitu *Map Based Movement* dengan cara menjalankan skenario kedua sebagai acuan pada parameter pengujiannya. Pengujian yang dilakukan pada skenario kedua dengan waktu pengujian simulasi selama 65 menit sesuai dengan waktu tempuh yang sebenarnya untuk melakukan perjalanan jarak jauh antar kota Tulung Agung ke Trenggalek.

Tabel 4.5 Skenario Pengujian Kedua

Skenario	Waktu	Protokol <i>Routing</i>	Ukuran Pesan		
		2	65 Menit	<i>Epidemic</i>	300 MB
		<i>Spray And Wait</i>	300 MB	600 MB	1,2 GB

A. Pengujian Kedua Pada Protokol *Routing Epidemic*

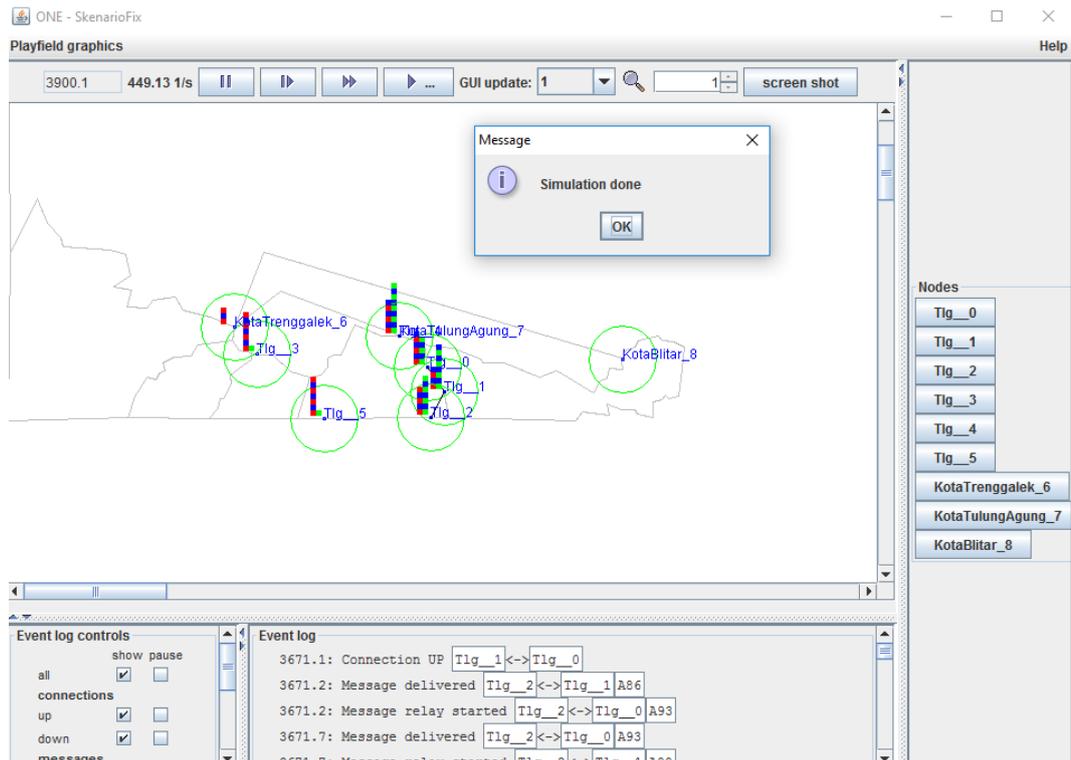
Pengujian Kedua dengan menggunakan protokol *routing Epidemic* dengan waktu simulasi yang sama selama 65 menit dan menggunakan ukuran pesan 300 MB, 600 MB dan 1,2 GB. Hasil Pengujian dapat ditunjukkan pada Gambar 4.20.



Gambar 4.20 Simulasi Kedua Protokol *Routing Epidemic*

B. Pengujian Kedua Pada Protokol *Routing Spray And Wait*

Hasil Pengujian Kedua menggunakan Protokol *Routing Spray And Wait* dapat terlihat pada Gambar 4.21.



Gambar 4.21 Simulasi Kedua Protokol *Routing Spray And Wait*

4.3.1.3 Pengujian Skenario ketiga

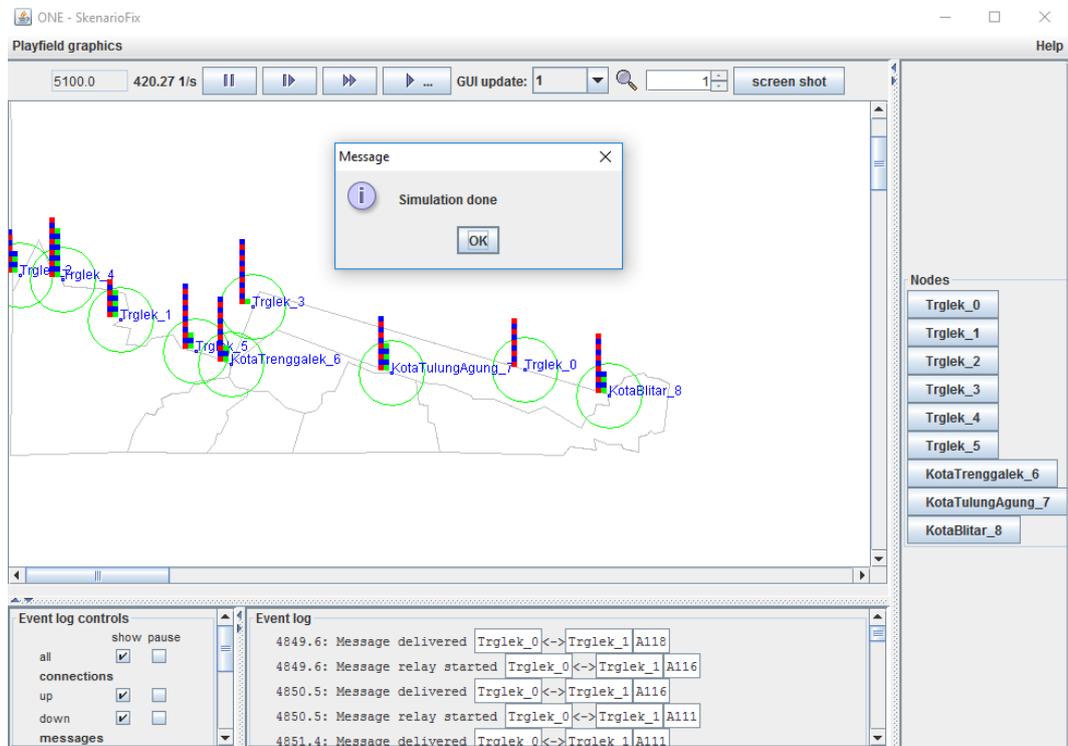
Pengujian pada skenario ketiga bertujuan untuk dapat membandingkan *performansi* terhadap kinerja kedua protokol *routing* seperti *Epidemic* dan *Spray And Wait* menggunakan pergerakan node yaitu *Map Based Movement* dengan cara menjalankan skenario kedua sebagai acuan pada parameter pengujiannya. Pengujian yang dilakukan pada skenario ketiga dengan waktu pengujian simulasi selama 85 menit sesuai dengan waktu tempuh yang sebenarnya untuk melakukan perjalanan jarak berbeda tujuan antar kota Treggalek ke Tulung Agung dan Blitar.

Tabel 4.6 Skenario Pengujian Ketiga

Skenario	Waktu	Protokol <i>Routing</i>	Ukuran Pesan		
			300 MB	600 MB	1,2 GB
3	85 Menit	<i>Epidemic</i>	300 MB	600 MB	1,2 GB
		<i>Spray And Wait</i>	300 MB	600 MB	1,2 GB

A. Pengujian Ketiga Pada Protokol *Routing Epidemic*

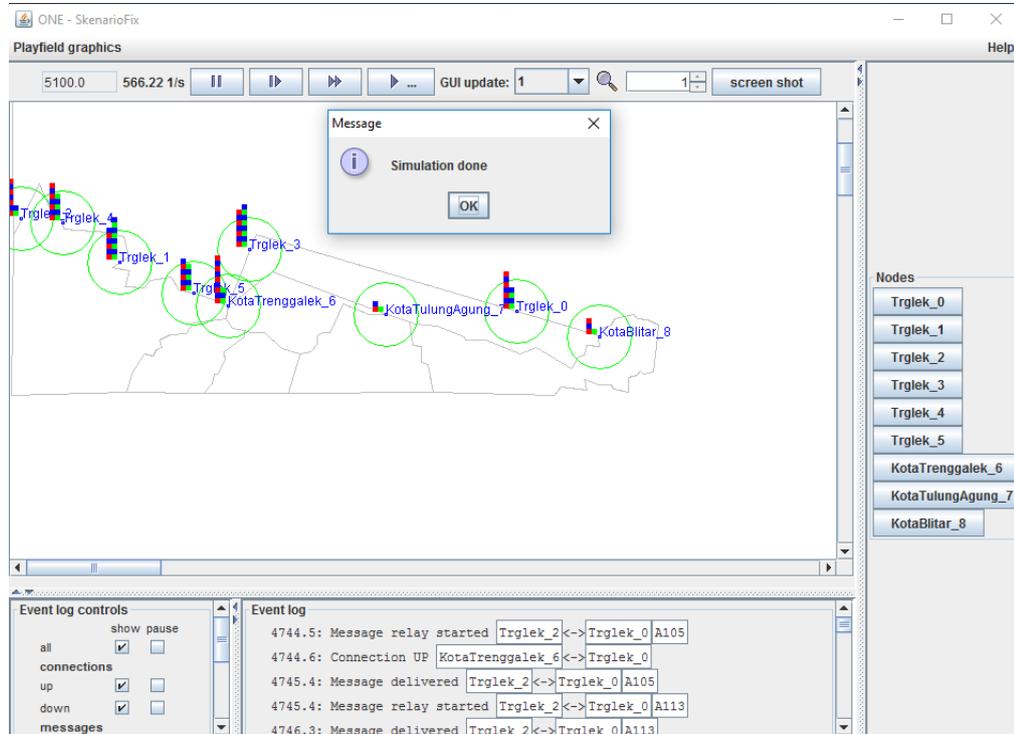
Pengujian ketiga dengan menggunakan protokol *routing Epidemic* dengan waktu simulasi yang sama selama 85 menit dan menggunakan ukuran pesan 300 MB, 600 MB dan 1,2 GB. Hasil Pengujian dapat ditunjukkan pada gambar 4.22



Gambar 4.22 Simulasi Ketiga Protokol *Routing Epidemic*

B. Pengujian Ketiga Pada Protokol *Routing Spray And Wait*

Hasil Pengujian ketiga menggunakan Protokol *Routing Spray And Wait* dapat terlihat pada Gambar 4.23.



Gambar 4.23 Simulasi Ketiga Protokol *Routing Spray And Wait*

4.3.2 Open Shortest Path

4.3.2.1 Pengujian Skenario Pertama

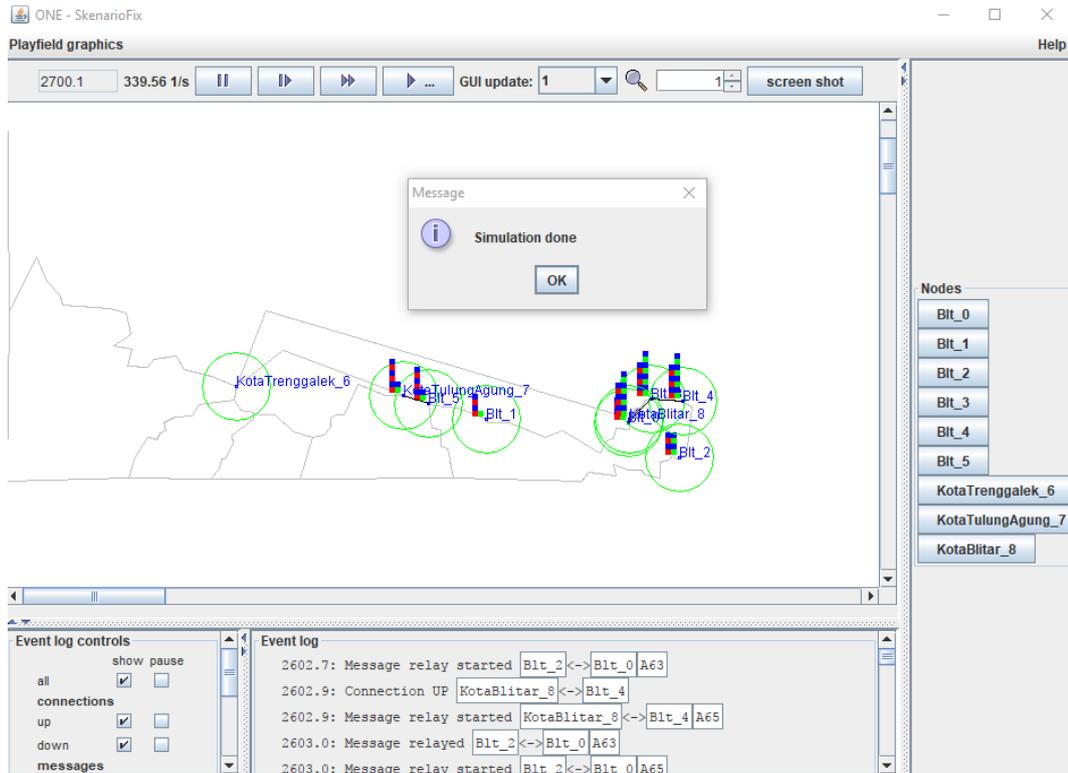
Pengujian pada skenario pertama bertujuan untuk dapat membandingkan *performansi* terhadap kinerja kedua protokol *routing* seperti *Epidemic* dan *Spray And Wait* menggunakan pergerakan node yaitu *Open Shortest Path* dengan cara menjalankan skenario pertama sebagai acuan pada parameter pengujiannya. Pengujian yang dilakukan pada skenario pertama dengan waktu pengujian simulasi selama 45 menit sesuai dengan waktu tempuh yang sebenarnya untuk melakukan perjalanan jarak dekat antar kota Blitar ke Tulung Agung.

Tabel 4.7 Skenario Pengujian Pertama

Skenario	Waktu	Protokol <i>Routing</i>	Ukuran Pesan		
		1	45 Menit	<i>Epidemic</i>	300 MB
		<i>Spray And Wait</i>	300 MB	600 MB	1,2 GB

A. Pengujian Pertama Pada Protokol *Routing Epidemic*

Pengujian Pertama dengan menggunakan protokol *routing Epidemic* dengan waktu simulasi yang sama selama 45 menit dan menggunakan ukuran pesan 300 MB, 600 MB dan 1,2 GB.

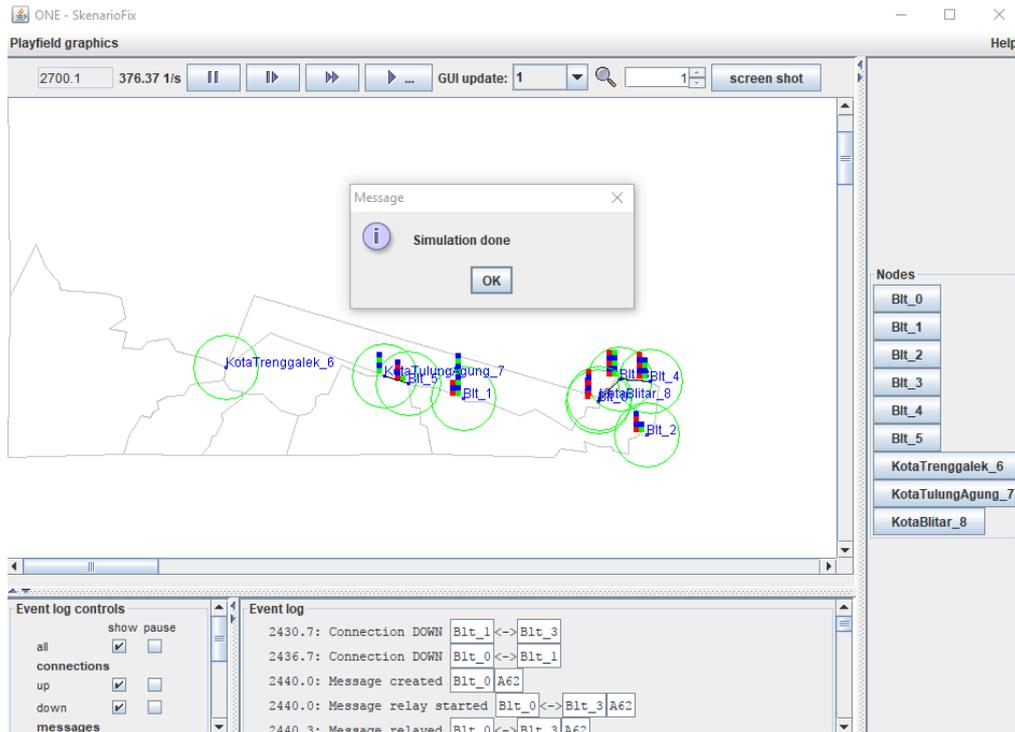


Gambar 4.24 Simulasi Pertama Protokol *Routing Epidemic*

Pada Gambar 4.24 Menunjukkan bahwa tahap simulasi pengujian skenario pertama pada Protokol *Routing Epidemic* telah sukses dilakukan. Tampilan Hasil pada Screenshot Gambar 4.18 terlihat bahwa *node* yang bergerak (kurir) yang bertindak membawa pesan berjumlah 6 buah yaitu, Blt_0, Blt_1, Blt_2, Blt_3, Blt 4, dan Blt_5. Dan *node* statis atau tetap berjumlah 3 buah yaitu , KotaTrenggalek_6, KotaTulungAgung_7, dan KotaBlitar_8.

B. Pengujian Pertama Pada Protokol *Routing Spray And Wait*

Pengujian pertama dengan menggunakan protokol *routing Spray And Wait* dengan waktu pengujian simulasi yang sama selama 45 menit dengan ukuran pesan 300 MB, 600 MB, 1,2 GB.



Gambar 4.25 Simulasi Pertama Protokol *Routing Spray And Wait*

Pada Gambar 4.25 Menunjukkan bahwa tahap simulasi pengujian skenario pertama pada Protokol *Routing Spray And Wait* telah sukses dilakukan.

4.3.2.2 Pengujian Skenario kedua

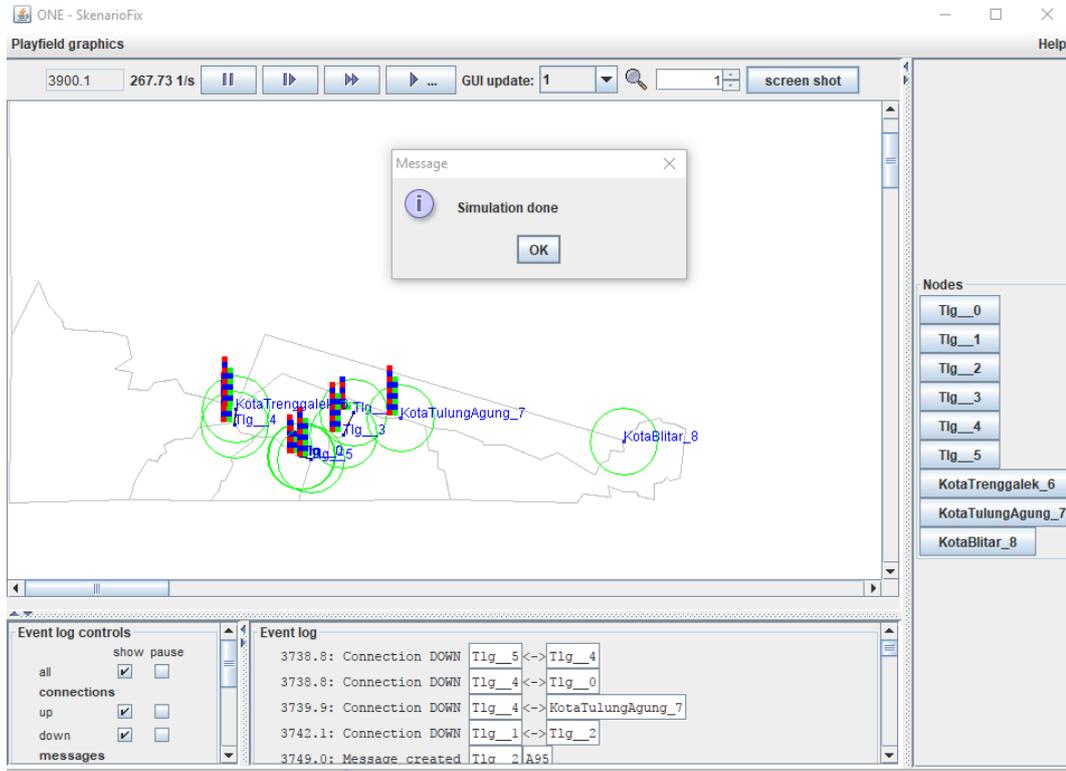
Pengujian pada skenario kedua bertujuan untuk dapat membandingkan *performansi* terhadap kinerja kedua protokol *routing* seperti *Epidemic* dan *Spray And Wait* menggunakan pergerakan node yaitu *Open Shortest Path* dengan cara menjalankan skenario kedua sebagai acuan pada parameter pengujiannya. Pengujian yang dilakukan pada skenario kedua dengan waktu pengujian simulasi selama 65 menit sesuai dengan waktu tempuh yang sebenarnya untuk melakukan perjalanan jarak jauh antar kota Tulung Agung ke Tenggalek.

Tabel 4.8 Skenario Pengujian Kedua

Skenario	Waktu	Protokol <i>Routing</i>	Ukuran Pesan		
2	65 Menit	<i>Epidemic</i>	300 MB	600 MB	1,2 GB
		<i>Spray And Wait</i>	300 MB	600 MB	1,2 GB

A. Pengujian Kedua Pada Protokol *Routing Epidemic*

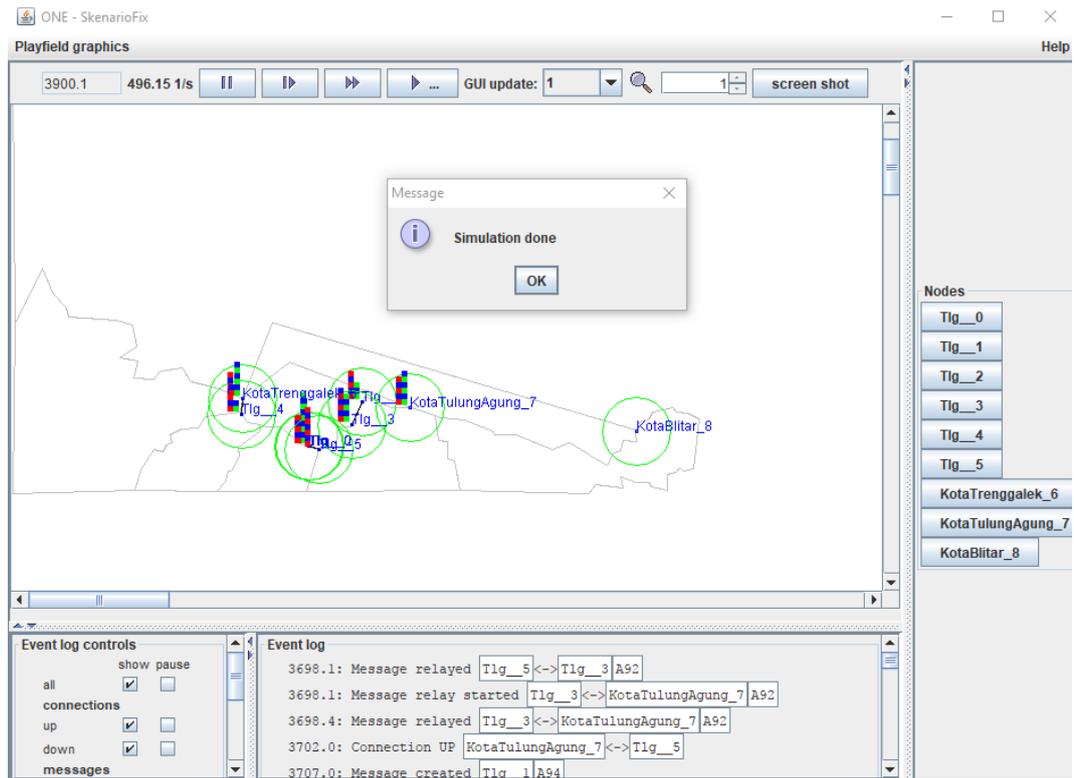
Pengujian Kedua dengan menggunakan protokol *routing Epidemic* dengan waktu simulasi yang sama selama 65 menit dan menggunakan ukuran pesan 300 MB, 600 MB dan 1,2 GB. Hasil Pengujian dapat ditunjukkan pada Gambar 4.26.



Gambar 4.26 Simulasi Kedua Protokol *Routing Epidemic*

B. Pengujian Kedua Pada Protokol *Routing Spray And Wait*

Hasil Pengujian Kedua menggunakan Protokol *Routing Spray And Wait* dapat terlihat pada Gambar 4.27.



Gambar 4.27 Simulasi Kedua Protokol *Routing Spray And Wait*

4.3.2.3 Pengujian Skenario ketiga

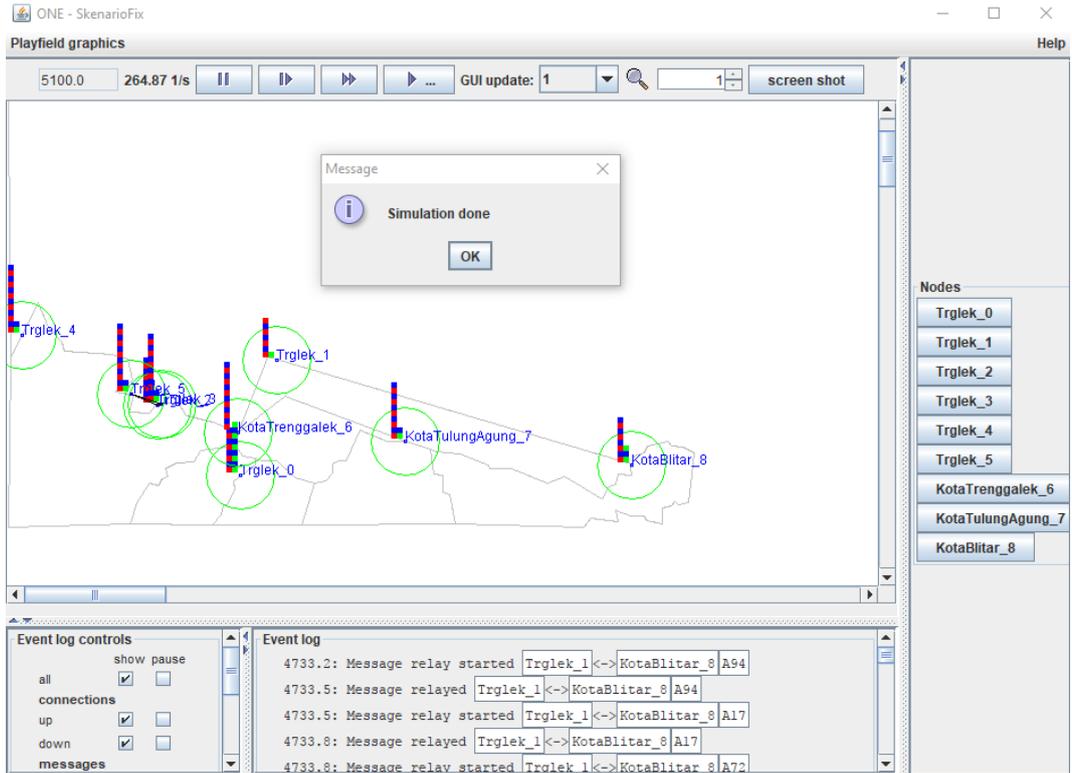
Pengujian pada skenario ketiga bertujuan untuk dapat membandingkan *performansi* terhadap kinerja kedua protokol *routing* seperti *Epidemic* dan *Spray And Wait* menggunakan pergerakan node yaitu *Open Shortest Path* dengan cara menjalankan skenario kedua sebagai acuan pada parameter pengujiannya. Pengujian yang dilakukan pada skenario ketiga dengan waktu pengujian simulasi selama 85 menit sesuai dengan waktu tempuh yang sebenarnya untuk melakukan perjalanan berbeda tujuan antar kota Trenggalek ke Tulung Agung dan Blitar.

Tabel 4.9 Skenario Pengujian Ketiga

Skenario	Waktu	Protokol <i>Routing</i>	Ukuran Pesan		
		3	85 Menit	<i>Epidemic</i>	300 MB
		<i>Spray And Wait</i>	300 MB	600 MB	1,2 GB

A. Pengujian Ketiga Pada Protokol *Routing Epidemic*

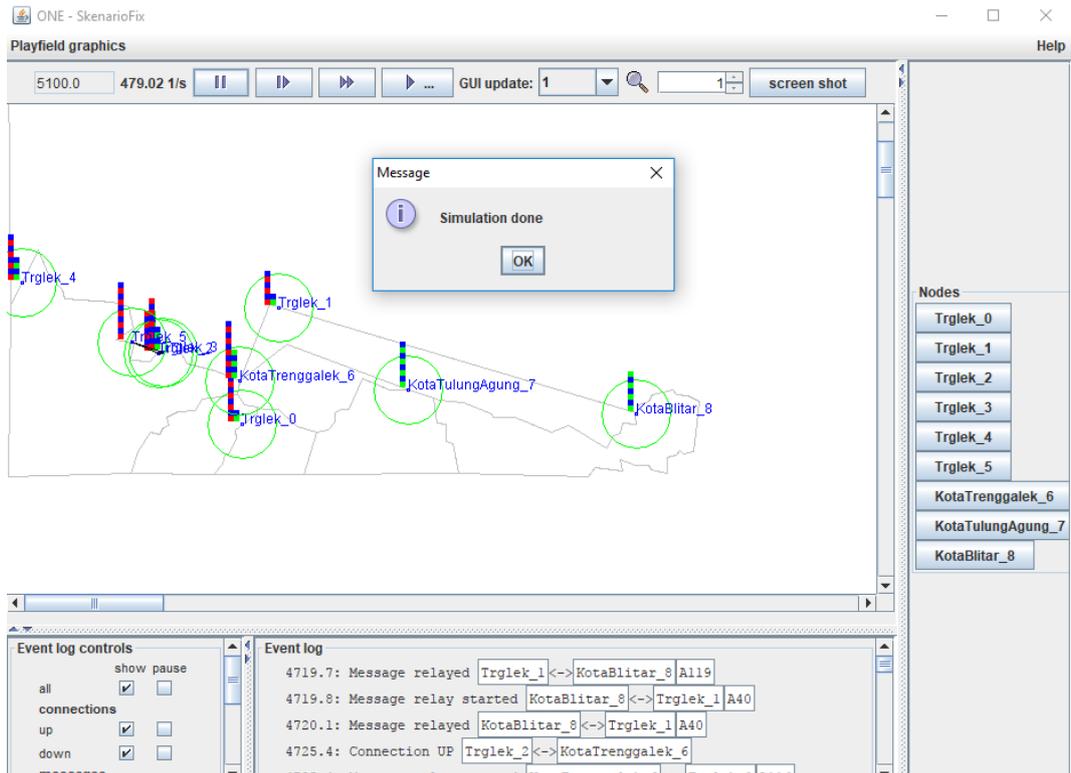
Pengujian ketiga dengan menggunakan protokol *routing Epidemic* dengan waktu simulasi yang sama selama 85 menit dan menggunakan ukuran pesan 300 MB, 600 MB dan 1,2 GB. Hasil Pengujian dapat ditunjukkan pada Gambar 4.28



Gambar 4.28 Simulasi Ketiga Protokol *Routing Epidemic*

B. Pengujian Ketiga Pada Protokol *Routing Spray And Wait*

Hasil Pengujian ketiga menggunakan Protokol *Routing Spray And Wait* dapat terlihat pada Gambar 4.29.



Gambar 4.29 Simulasi Ketiga Protokol Routing Spray And Wait

4.3.3 Map Route Movement

4.3.3.1 Pengujian Skenario Pertama

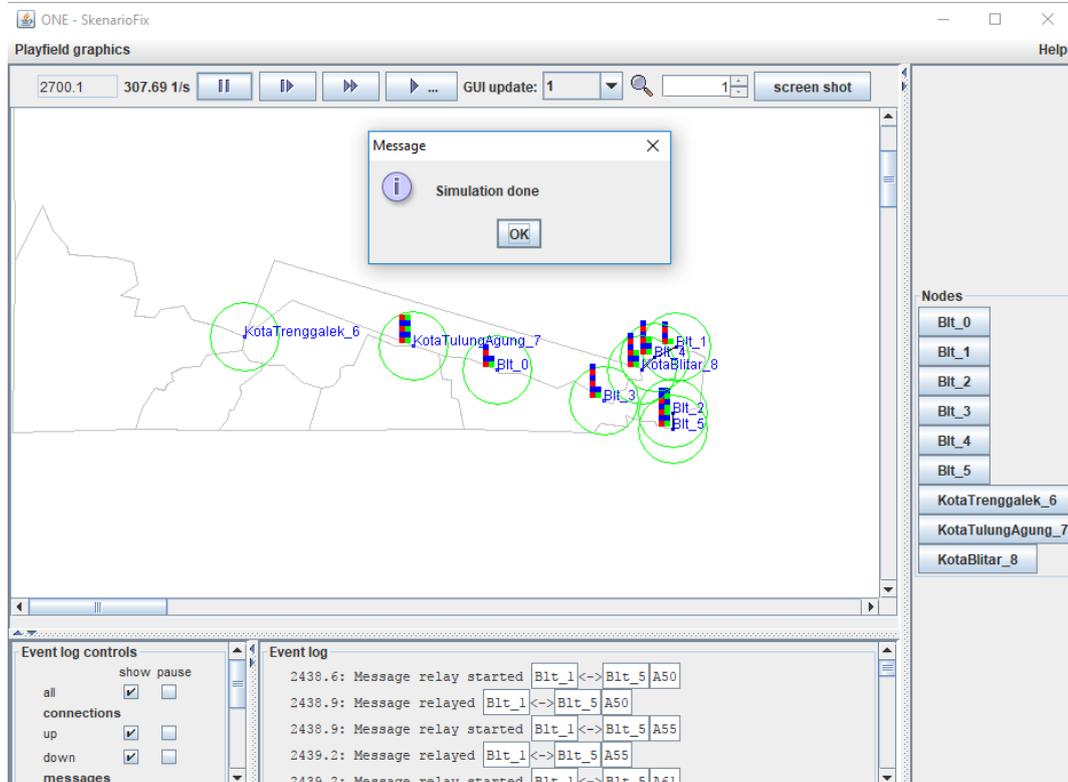
Pengujian pada skenario pertama bertujuan untuk dapat membandingkan *performansi* terhadap kinerja kedua protokol routing seperti *Epidemic* dan *Spray And Wait* menggunakan pergerakan node yaitu *Map Route Movement* dengan cara menjalankan skenario pertama sebagai acuan pada parameter pengujiannya. Pengujian yang dilakukan pada skenario pertama dengan waktu pengujian simulasi selama 45 menit sesuai dengan waktu tempuh yang sebenarnya untuk melakukan perjalanan jarak dekat antar kota Blitar ke Tulung Agung.

Tabel 4.10 Skenario Pengujian Pertama

Skenario	Waktu	Protokol Routing	Ukuran Pesan		
			300 MB	600 MB	1,2 GB
1	45 Menit	<i>Epidemic</i>	300 MB	600 MB	1,2 GB
		Spray And Wait	300 MB	600 MB	1,2 GB

C. Pengujian Pertama Pada Protokol *Routing Epidemic*

Pengujian Pertama dengan menggunakan protokol *routing Epidemic* dengan waktu simulasi yang sama selama 45 menit dan menggunakan ukuran pesan 300 MB, 600 MB dan 1,2 GB.

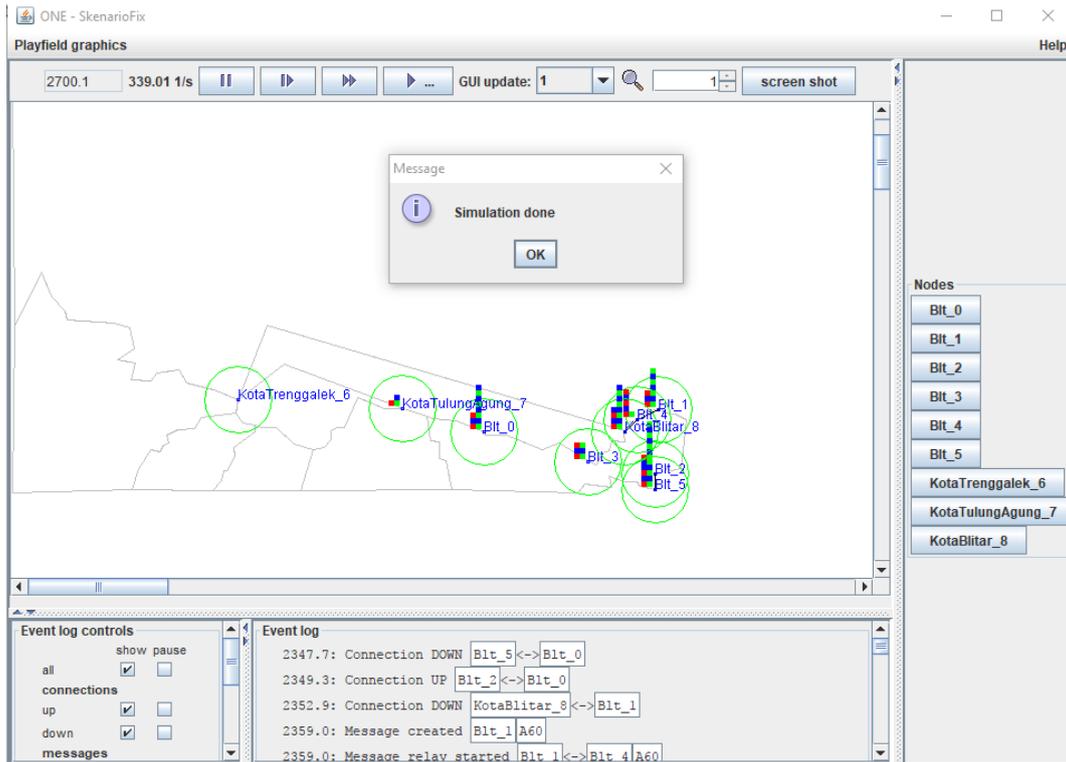


Gambar 4.30 Simulasi Pertama Protokol *Routing Epidemic*

Pada Gambar 4.30 Menunjukkan bahwa tahap simulasi pengujian skenario pertama pada Protokol *Routing Epidemic* telah sukses dilakukan. Tampilan Hasil pada Screenshot Gambar 4.18 terlihat bahwa *node* yang bergerak (kurir) yang bertindak membawa pesan berjumlah 6 buah yaitu, Blt_0, Blt_1, Blt_2, Blt_3, Blt_4, dan Blt_5. Dan *node* statis atau tetap berjumlah 3 buah yaitu , KotaTreggalek_6, KotaTulungAgung_7, dan KotaBlitar_8.

D. Pengujian Pertama Pada Protokol *Routing Spray And Wait*

Pengujian Pertama pada skenario pertama ini dengan menggunakan protokol *routing Spray And Wait* dengan waktu pengujian simulasi yang sama selama 45 menit dengan ukuran pesan 300 MB, 600 MB, 1,2 GB.



Gambar 4.31 Simulasi Pertama Protokol *Routing Spray And Wait*

Pada Gambar 4.31 Menunjukkan bahwa tahap simulasi pengujian skenario pertama pada Protokol *Routing Spray And Wait* telah sukses dilakukan.

4.3.3.2 Pengujian Skenario kedua

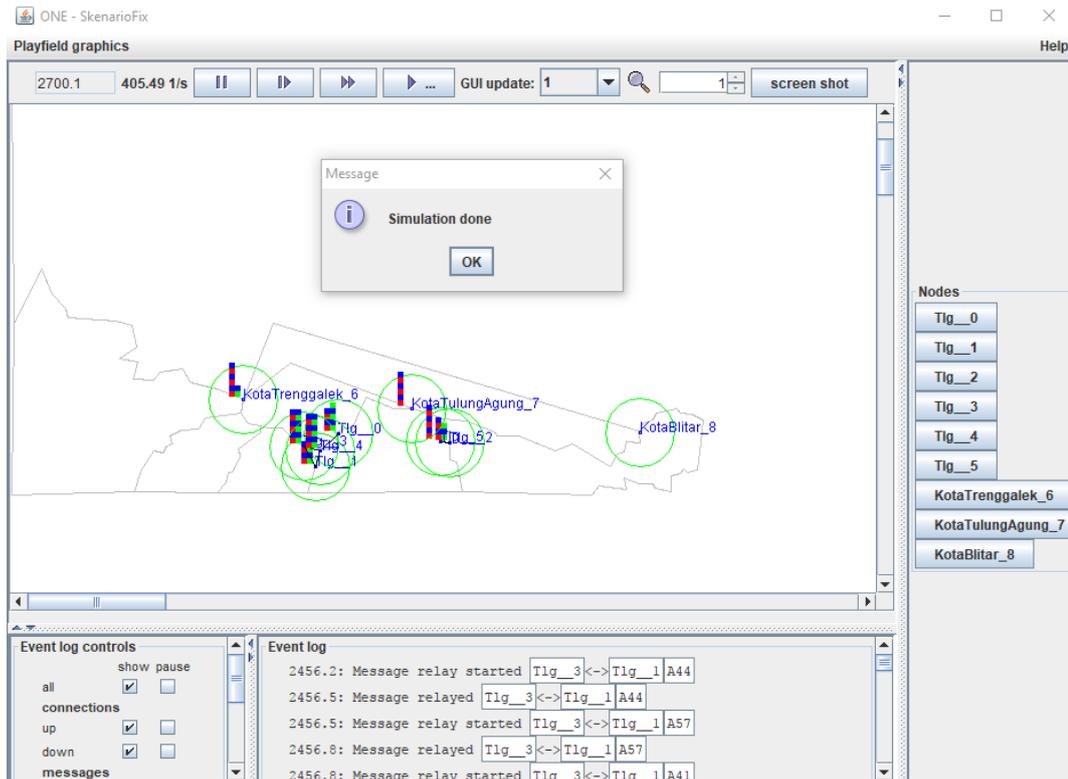
Pengujian pada skenario kedua bertujuan untuk dapat membandingkan *performansi* terhadap kinerja kedua protokol *routing* seperti *Epidemic* dan *Spray And Wait* menggunakan pergerakan node yaitu *Map Route Movement* dengan cara menjalankan skenario kedua sebagai acuan pada parameter pengujiannya. Pengujian yang dilakukan pada skenario kedua dengan waktu pengujian simulasi selama 65 menit sesuai dengan waktu tempuh yang sebenarnya untuk melakukan perjalanan jarak jauh antar kota Tulung Agung ke Trenggalek.

Tabel 4.11 Skenario Pengujian Kedua

Skenario	Waktu	Protokol <i>Routing</i>	Ukuran Pesan		
		<i>Epidemic</i>	300 MB	600 MB	1,2 GB
2	65 Menit	<i>Spray And Wait</i>	300 MB	600 MB	1,2 GB

A. Pengujian Kedua Pada Protokol Routing Epidemic

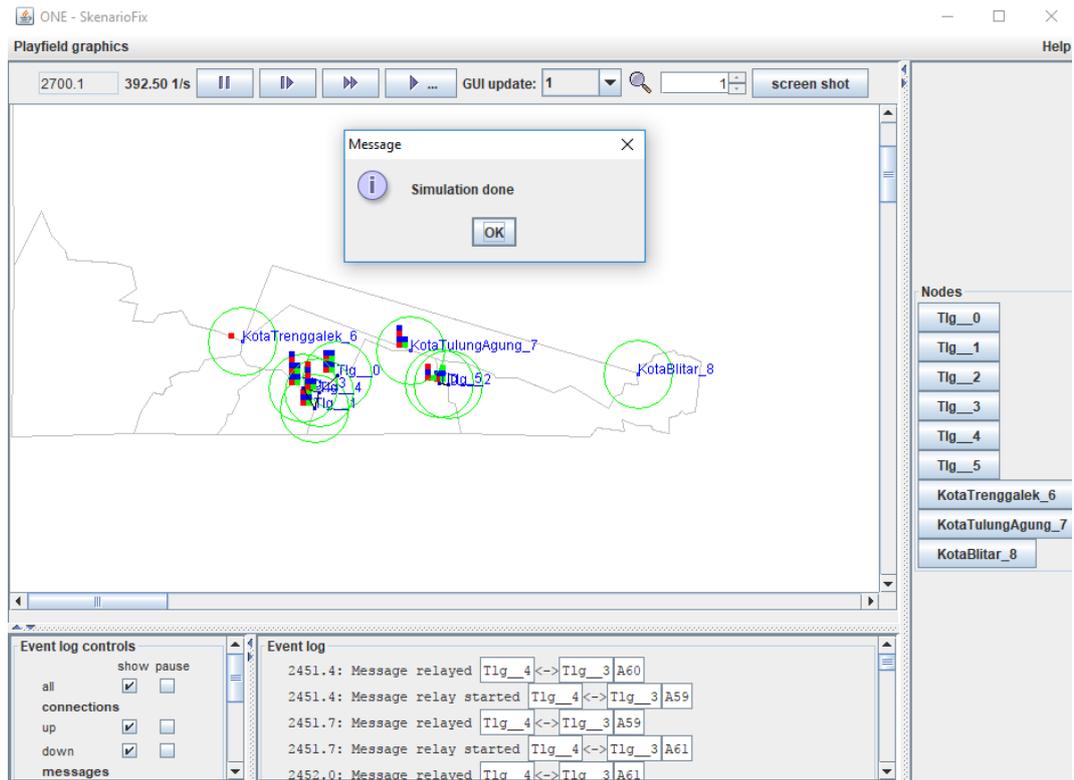
Pengujian Kedua dengan menggunakan protokol *routing Epidemic* dengan waktu simulasi yang sama selama 65 menit dan menggunakan ukuran pesan 300 MB, 600 MB dan 1,2 GB. Hasil Pengujian dapat ditunjukkan pada Gambar 4.32.



Gambar 4.32 Simulasi Kedua Protokol *Routing Epidemic*

B. Pengujian Kedua Pada Protokol Routing Spray And Wait

Hasil Pengujian Kedua menggunakan Protokol *Routing Spray And Wait* dapat terlihat pada Gambar 4.33.



Gambar 4.33 Simulasi Kedua Protokol Routing Spray And Wait

4.3.3.3 Pengujian Skenario ketiga

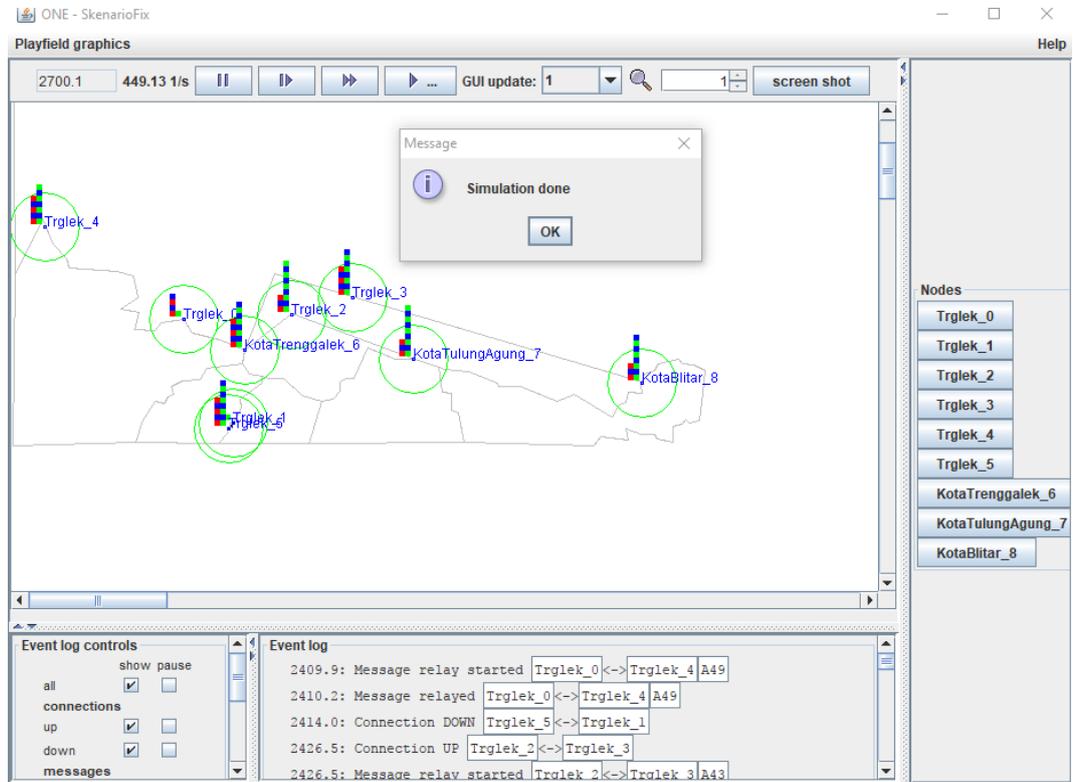
Pengujian pada skenario ketiga bertujuan untuk dapat membandingkan *performansi* terhadap kinerja kedua protokol *routing* seperti *Epidemic* dan *Spray And Wait* menggunakan pergerakan node yaitu *Map Route Movement* dengan cara menjalankan skenario kedua sebagai acuan pada parameter pengujiannya. Pengujian yang dilakukan pada skenario ketiga dengan waktu pengujian simulasi selama 85 menit sesuai dengan waktu tempuh yang sebenarnya untuk melakukan perjalanan berbeda tujuan antar kota Trenggalek ke Tulung Agung dan Blitar.

Tabel 4.12 Skenario Pengujian Ketiga

Skenario	Waktu	Protokol Routing	Ukuran Pesan		
		3	85 Menit	<i>Epidemic</i>	300 MB
		<i>Spray And Wait</i>	300 MB	600 MB	1,2 GB

A. Pengujian Ketiga Pada Protokol Routing Epidemic

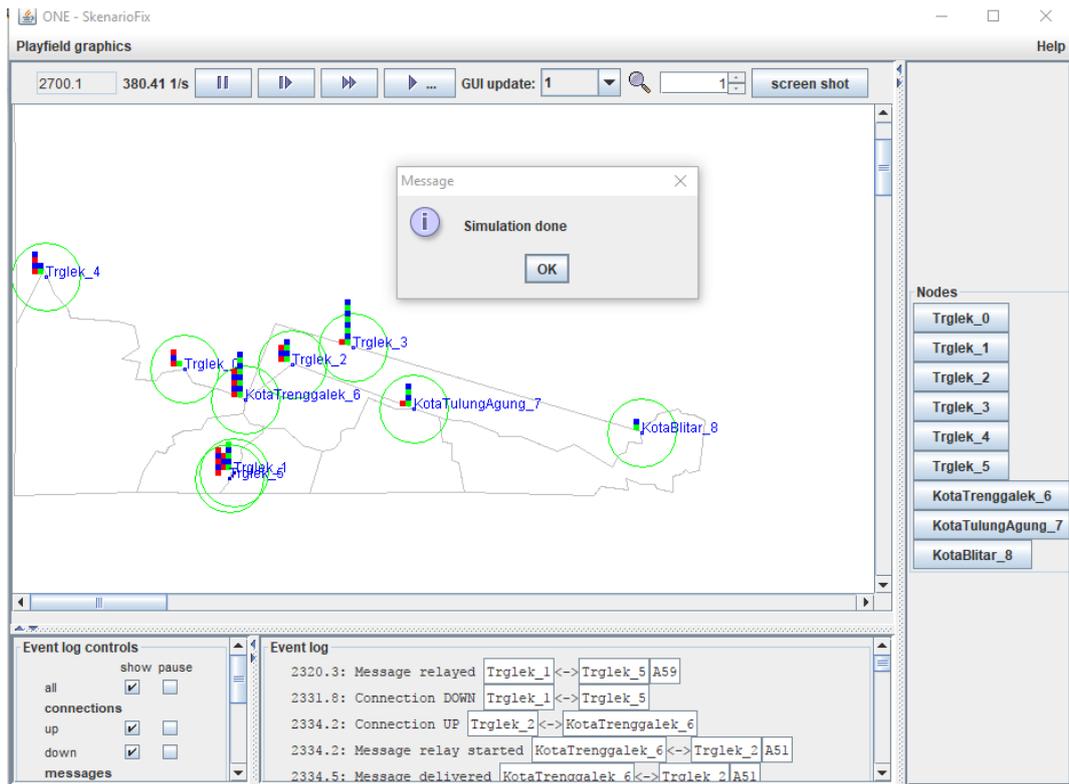
Pengujian ketiga dengan menggunakan protokol *routing Epidemic* dengan waktu simulasi yang sama selama 85 menit dan menggunakan ukuran pesan 300 MB, 600 MB dan 1,2 GB. Hasil Pengujian dapat ditunjukkan pada Gambar 4.34



Gambar 4.34 Simulasi Ketiga Protokol Routing Epidemic

B. Pengujian Ketiga Pada Protokol Routing Spray And Wait

Hasil Pengujian ketiga menggunakan Protokol *Routing Spray And Wait* dapat terlihat pada Gambar 4.35.



Gambar 4.35 Simulasi Ketiga Protokol *Routing Spray And Wait*

BAB 5 PEMBAHASAN

Pada bab pembahasan ini akan membahas mengenai hasil dari pengujian yang telah dilakukan pada tahap perancangan simulasi sebelumnya dan akan menjelaskan secara lebih detail tentang hasil pembahasan, sehingga pada hasil ini dapat dijadikan sebagai penarikan kesimpulan untuk bab selanjutnya.

5.1 Analisis Data

Analisis data akan diperlukan untuk dapat menguji skenario pengujian. Dan pada bagian pembahasan akan dijelaskan bagaimana tahap dalam menganalisa data. Pada bagian analisis data dapat ditentukan menjadi dua tipe yaitu berdasarkan parameter pengujian yang telah ditentukan dan menganalisis data setiap skenario.

5.2 Hasil Pengujian

Hasil pengujian akan dijelaskan dengan menggunakan parameter *Delivery probability*, *Overhead Ratio*, dan *Average Latency* dalam pengiriman pesan untuk seluruh skenario protokol *routing Epidemic* dan *Spray And Wait* berdasarkan tipe mobilitasnya yaitu, *Map Based Movement*, *Open Shortest Path*, dan *Map Route Movement* dan juga pengaruh dari penggunaan mobilitas pada protokol *routing* akan dijelaskan lebih detail dibawah ini :

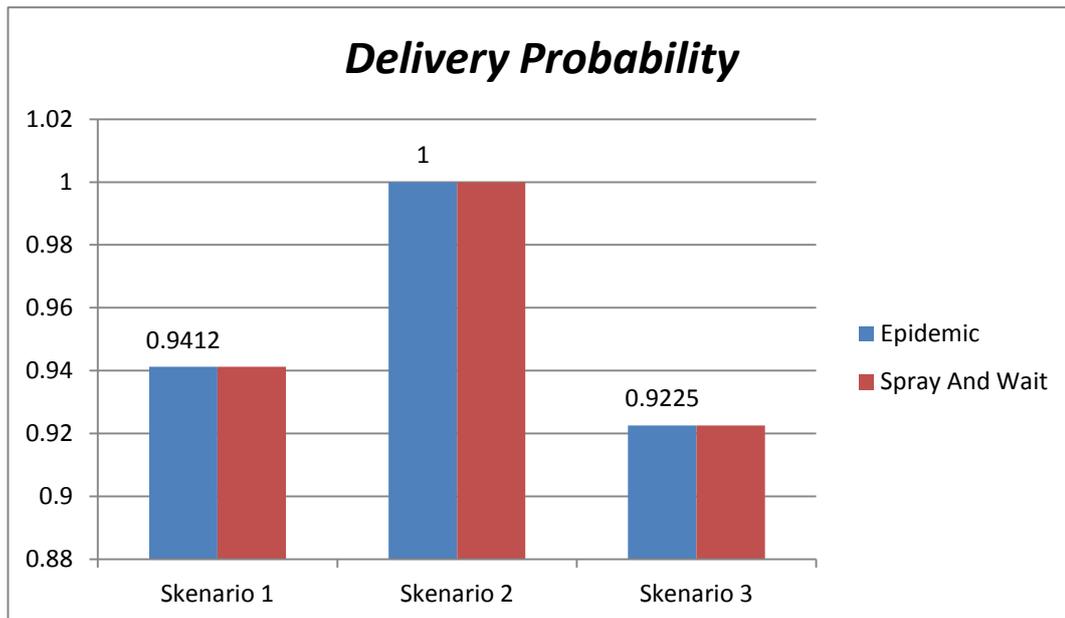
5.2.1 Delivery Probability

Hasil pengujian perbandingan pada parameter *Delivery Probability* menggunakan protokol *routing Epidemic* dan *Spray And Wait* berdasarkan pada tipe mobilitasnya dapat ditunjukkan pada tabel dan grafik dibawah ini :

5.2.1.1 Map Based Movement

Tabel 5.1 Hasil *Delivery Probability* Pada Semua Skenario

Rute	Routing	Delivery Probability
1	<i>Epidemic</i>	0.9412
	<i>Spray And Wait</i>	0.9412
2	<i>Epidemic</i>	1
	<i>Spray And Wait</i>	1
3	<i>Epidemic</i>	0.9225
	<i>Spray And Wait</i>	0.9225



Gambar 5.1 Grafik *Delivery Probability* Pada Semua Skenario Pengujian

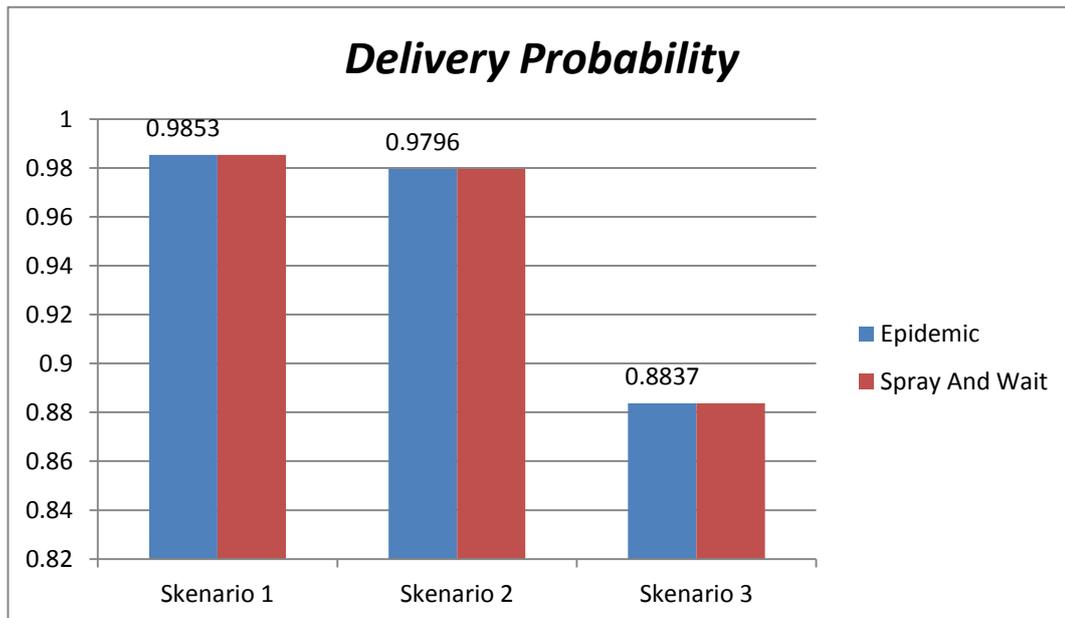
Pada Gambar 5.1 dengan menggunakan Rumus persamaan 2.1 pada bab 2 didapatkan hasil pengujian *Delivery Probability* pengiriman pesan pada skenario 1 berdasarkan mobilitas *Map Based Movement* dengan protokol *routing Epidemic* sebesar 0.9412 pesan pada ukuran pesan 300 MB, 600 MB, dan 1,2 GB dan pada protokol *routing Spray And Wait* sebesar 0.9412 pesan dengan menggunakan ukuran pesan yang sama dengan protokol *routing Epidemic*. Hasil pengujian pada skenario 2 pada *routing Epidemic* sebesar 1 pesan dan pada protokol *routing Spray And Wait* sebesar 1 pesan untuk semua ukuran pesan. Pada Skenario 3 didapatkan hasil pengujian pada protokol *routing Epidemic* sebesar 0.9225 pesan dan hasil pada protokol *routing Spray And Wait* sebesar 0.9225 pesan untuk semua ukuran pesan.

Hasil dan analisis pengujian yang dilakukan terhadap *Delivery Probability* menunjukkan bahwa pengujian yang dilakukan terhadap seluruh skenario menggunakan kedua protokol *routing* seperti *Epidemic* dan *Spray And Wait* mengalami peningkatan pesan pada skenario 2 sebesar 1 pesan atau tingkat *Delivery Probability* pengiriman pesan yang didapatkan mencapai 100% pada seluruh ukuran pesan. Dalam hal ini terlihat bahwa *Delivery Probability* pada setiap skenario pengujian pada kedua protokol *routing Epidemic* dan *Spray And Wait* mengalami peningkatan yang sama. Dari hal itu maka, dapat disimpulkan bahwa nilai probabilitas yang semakin tinggi maka protokol *routing* dapat mengirimkan pesan hingga sampai ke tujuan menjadi lebih banyak.

5.2.1.2 Open Shortest Path

Tabel 5.2 Hasil *Delivery Probability* Pada Semua Skenario

Rute	Routing	<i>Delivery Probability</i>
1	<i>Epidemic</i>	0.9853
	<i>Spray And Wait</i>	0.9853
2	<i>Epidemic</i>	0.9796
	<i>Spray And Wait</i>	0.9796
3	<i>Epidemic</i>	0.8837
	<i>Spray And Wait</i>	0.8837



Gambar 5.2 Grafik *Delivery Probability* Pada Semua Skenario Pengujian

Pada Gambar 5.2 dengan menggunakan Rumus persamaan 2.1 pada bab 2 didapatkan hasil pengujian *Delivery Probability* pengiriman pesan pada skenario 1 berdasarkan mobilitas *Open Shortest Path* dengan protokol routing *Epidemic* sebesar 0.9853 pesan pada ukuran pesan 300 MB, 600 MB, dan 1,2 GB dan pada protokol routing *Spray And Wait* sebesar 0.9853 pesan dengan menggunakan ukuran pesan yang sama dengan protokol routing *Epidemic*. Hasil pengujian pada skenario 2 pada routing *Epidemic* sebesar 0.9796 pesan dan pada

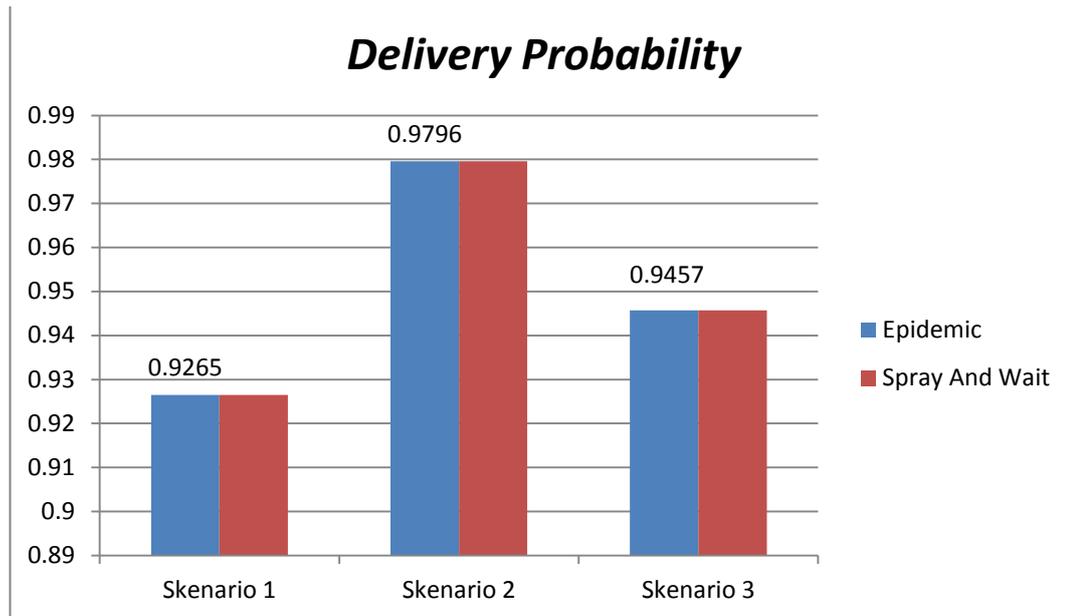
protokol *routing Spray And Wait* sebesar 0.9796 pesan untuk semua ukuran pesan. Pada Skenario 3 didapatkan hasil pengujian pada protokol *routing Epidemic* sebesar 0.8837 pesan dan hasil pada protokol *routing Spray And Wait* sebesar 0.8837 pesan untuk semua ukuran pesan.

Hasil dan analisis pengujian yang dilakukan terhadap *Delivery Probability* menunjukkan bahwa pengujian yang dilakukan terhadap seluruh skenario menggunakan kedua protokol *routing* seperti *Epidemic* dan *Spray And Wait* mengalami peningkatan pesan pada skenario 1 sebesar 0.9853 pesan dan mengalami penurunan pesan pada skenario 3 sebesar 0.8837 pesan. Dalam hal ini terlihat bahwa *Delivery Probability* pada setiap skenario pengujian pada kedua protokol *routing Epidemic* dan *Spray And Wait* mengalami peningkatan yang sama. Dari hal itu maka, dapat disimpulkan bahwa nilai probabilitas yang semakin tinggi maka protokol *routing* dapat mengirimkan pesan hingga sampai ke tujuan menjadi lebih banyak dan nilai probabilitas yang rendah maka protokol *routing* dalam mengirimkan pesan hingga ke tujuan menjadi lebih sedikit.

5.2.1.3 Map Route Movement

Tabel 5.3 Hasil *Delivery Probability* Pada Semua Skenario

Rute	Routing	<i>Delivery Probability</i>
1	Epidemic	0.9265
	Spray And Wait	0.9265
2	Epidemic	0.9796
	Spray And Wait	0.9796
3	Epidemic	0.9457
	Spray And Wait	0.9457



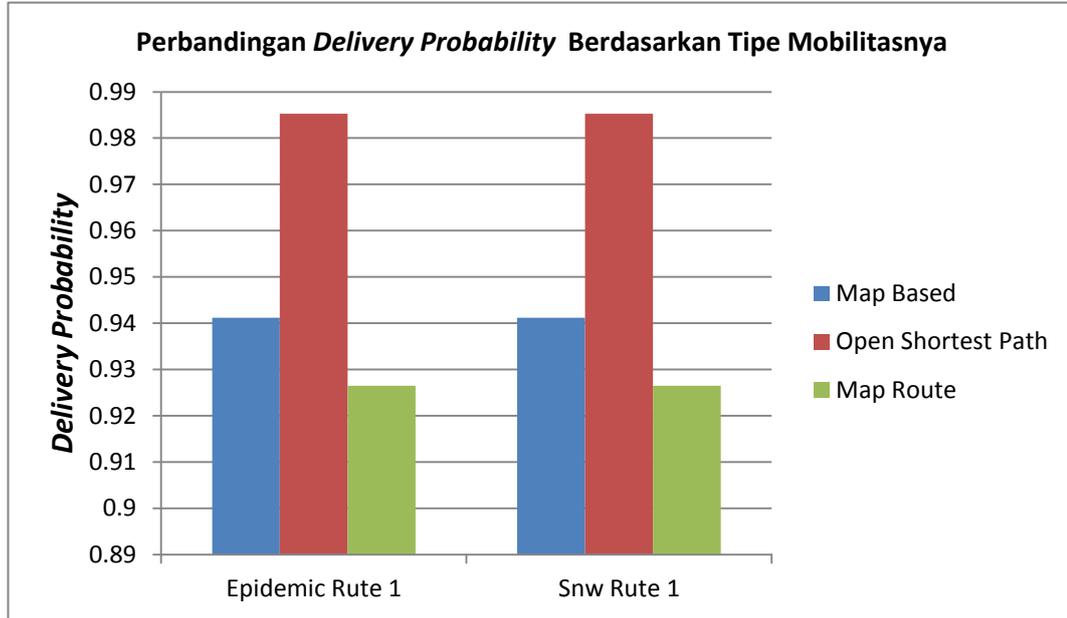
Gambar 5.3 Grafik *Delivery Probability* Pada Semua Skenario Pengujian

Pada Gambar 5.3 dengan menggunakan Rumus persamaan 2.1 pada bab 2 didapatkan hasil pengujian *Delivery Probability* pengiriman pesan pada skenario 1 berdasarkan mobilitas *Map Route* dengan protokol *routing Epidemic* sebesar 0.9265 pesan pada ukuran pesan 300 MB, 600 MB, dan 1,2 GB dan pada protokol *routing Spray And Wait* sebesar 0.9265 pesan dengan menggunakan ukuran pesan yang sama dengan protokol *routing Epidemic*. Hasil pengujian pada skenario 2 pada *routing Epidemic* sebesar 0.9796 pesan dan pada protokol *routing Spray And Wait* sebesar 0.9796 pesan untuk semua ukuran pesan. Pada Skenario 3 didapatkan hasil pengujian pada protokol *routing Epidemic* sebesar 0.9457 pesan dan hasil pada protokol *routing Spray And Wait* sebesar 0.9457 pesan untuk semua ukuran pesan.

Hasil dan analisis pengujian yang dilakukan terhadap *Delivery Probability* menunjukkan bahwa pengujian yang dilakukan terhadap seluruh skenario menggunakan kedua protokol *routing* seperti *Epidemic* dan *Spray And Wait* mengalami peningkatan pesan pada skenario 2 sebesar 0.9796 pesan dan mengalami penurunan pesan pada skenario 1 sebesar 0.9265 pesan. Dalam hal ini terlihat bahwa *Delivery Probability* pada setiap skenario pengujian pada kedua protokol *routing Epidemic* dan *Spray And Wait* mengalami peningkatan yang sama. Dari hal itu maka, dapat disimpulkan bahwa nilai probabilitas yang semakin tinggi maka protokol *routing* dapat mengirimkan pesan hingga sampai ke tujuan menjadi lebih banyak dan nilai probabilitas yang rendah maka protokol *routing* dalam mengirimkan pesan hingga ke tujuan menjadi lebih sedikit.

5.2.1.4 Perbandingan *Delivery Probability* terhadap tipe Mobilitas

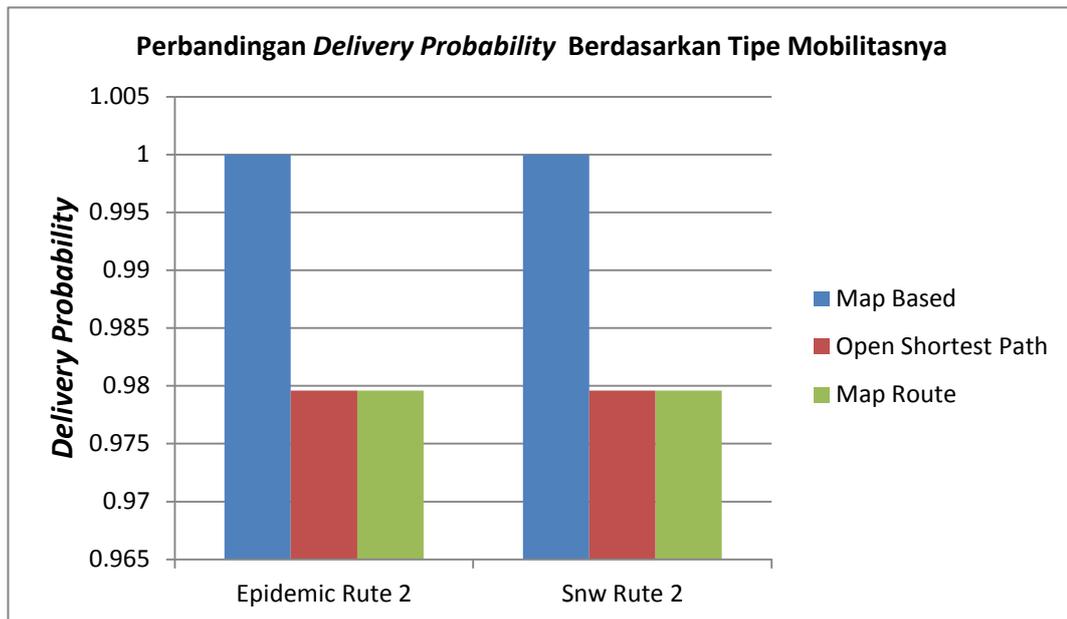
A. Skenario 1



Gambar 5.4 Grafik Perbandingan *Delivery Probability* Skenario 1 Berdasarkan Tipe Mobilitasnya

Pada Gambar 5.4, terlihat bahwa hasil perbandingan *Delivery Probability* protokol *routing Epidemic* dan *Spray And Wait* pada Skenario Rute 1 menggunakan protokol *routing Epidemic* sebesar 0.9412 pesan pada mobilitas *Map Based Movement*, 0.9853 pesan pada mobilitas *Open Shortest Path* dan 0.9265 pesan pada mobilitas *Map Route Movement*. Kemudian untuk protokol *routing Spray And Wait (Snw)* sebesar 0.9412 pesan pada mobilitas *Map Based Movement*, 0.9853 pesan pada mobilitas *Open Shortest Path*, dan 0.9265 pesan pada mobilitas *Map Route Movement*.

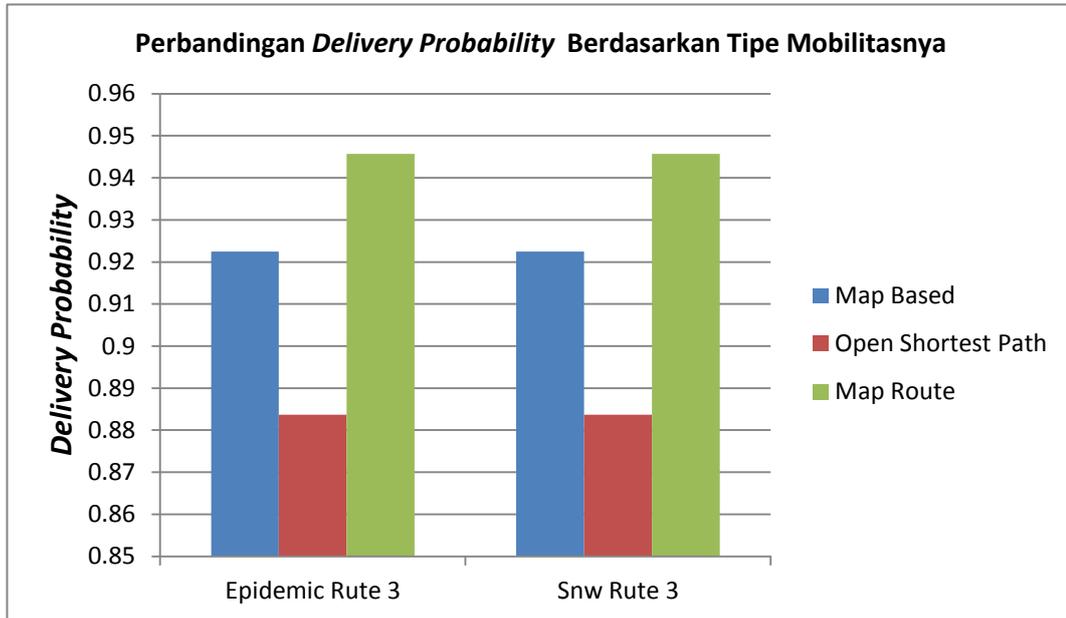
B. Skenario 2



Gambar 5.5 Grafik Perbandingan *Delivery Probability* Skenario 2 Berdasarkan Tipe Mobilitasnya

Pada Gambar 5.5, terlihat bahwa Skenario Rute 2 menggunakan protokol *routing Epidemic* sebesar 1 pesan pada mobilitas *Map Based Movement*, 0.9796 pesan pada mobilitas *Open Shortest Path* dan 0.9796 pesan pada mobilitas *Map Route Movement*. Kemudian untuk protokol *routing Spray And Wait (Snw)* sebesar 1 pesan pada mobilitas *Map Based Movement*, 0.9796 pesan pada mobilitas *Open Shortest Path*, dan 0.9796 pesan pada mobilitas *Map Route Movement*.

C. Skenario 3



Gambar 5.6 Grafik Perbandingan *Delivery Probability* Skenario 3 Berdasarkan Tipe Mobilitasnya

Pada Gambar 5.6, terlihat bahwa Skenario Rute 3 menggunakan protokol *routing Epidemic* sebesar 0.9225 pesan pada mobilitas *Map Based Movement*, 0.8837 pesan pada mobilitas *Open Shortest Path* dan 0.9457 pesan pada mobilitas *Map Route Movement*. Kemudian untuk protokol *routing Spray And Wait (Snw)* sebesar 0.9225 pesan pada mobilitas *Map Based Movement*, 0.8837 pesan pada mobilitas *Open Shortest Path*, dan 0.9457 pesan pada mobilitas *Map Route Movement*.

Berdasarkan dari hasil perbandingan pengujian seluruh skenario yaitu skenario rute 1, skenario rute 2, dan skenario rute 3 dari parameter *Delivery Probability* pada beberapa mobilitas terlihat bahwa, *Delivery Probability* tertinggi ada pada Skenario Rute 2 dengan menggunakan mobilitas *Map Based Movement* sebesar 1 pesan. Dan hasil *Delivery Probability* paling rendah ada pada Mobilitas *Open Shortest Path* pada Skenario Rute 3 sebesar 0.8837 pesan.

Hasil dan Analisis dari pengujian tersebut adalah penggunaan protokol *routing Epidemic* dan *Spray And Wait* untuk tingkat probabilitas pengiriman jumlah pesan yang lebih banyak dari sumber ke tujuan yaitu dengan menggunakan mobilitas *Map Based Movement* dikarenakan mobilitas *Map Based Movement*, node bergerak mengirimkan pesan secara acak namun tetap mengikuti alur pada peta sehingga tingkat probabilitas pengiriman menjadi sangat banyak .

5.2.1.5 Pengaruh penggunaan tipe mobilitas untuk *Delivery Probability* pada Protokol Routing:

- A. Pergerakan node dari *Source* ke *Destination* dengan menggunakan tipe mobilitas *Map Based Movement* mempunyai dampak *Delivery Probability* yang dihasilkan mengalami tingkat probabilitas pengiriman pesan yang lebih banyak dibandingkan dengan tipe mobilitas lainnya seperti mobilitas *Open Shortest Path*, dan *Map Route Movement*. Sehingga tingkat probabilitas pesan yang tinggi disebabkan karena penerapan pada mobilitas *Map Based Movement* dipengaruhi oleh adanya protokol *Routing Epidemic* yang juga berperan dalam mengatur rute perjalanan. Dan pada sifat dari protokol *Routing Epidemic* itu sendiri yang juga membanjiri pesan pada suatu jaringan dan juga jarak tempuh dan waktu tempuh yang berperan didalamnya. Sehingga pergerakan node pada mobilitas ini dapat memberikan suatu pesan ke tujuan menjadi lebih baik.
- B. Pergerakan node dari *Source* ke *Destination* dengan menggunakan tipe mobilitas *Open Shortest Path* mempunyai dampak *Delivery Probability* yang dihasilkan mengalami tingkat probabilitas pengiriman pesan yang rendah atau buruk. Dan dikarenakan oleh mobilitas ini pergerakan nodenya memilih titik yang dituju secara acak dan bergerak melalui jalur terpendek sehingga pengaruh dari mobitas ini untuk *Delivery Probability* yang dihasilkan menjadi tidak maksimal dalam hal proses pengiriman ke tujuan dan juga disebabkan oleh adanya peran jarak tempuh, waktu tempuh, dan penggunaan protokol *Routing* yang digunakan.
- C. Pergerakan node dari *Source* ke *Destination* dengan menggunakan tipe mobilitas *Map Route Movement* mempunyai dampak terhadap *Delivery Probability* yang dihasilkan mengalami tingkat probabilitas pengiriman pesan yang rendah atau tidak terlalu tinggi. Sehingga pada mobilitas ini dalam hal probabilitas rasio pengiriman pesan kurang maksimal. Dikarenakan pergerakan node bergerak pada konstruksi jalan yang tetap.

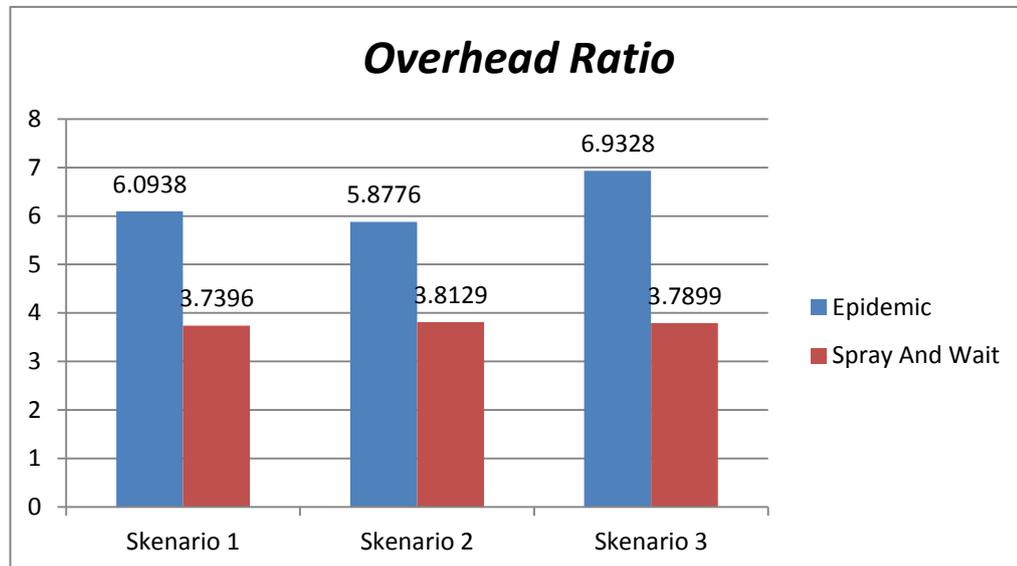
5.2.2 Overhead Ratio

Hasil pengujian pada parameter *Overhead Ratio* dalam pengiriman pesan untuk seluruh skenario protokol *routing Epidemic* dan *Spray And Wait* dapat ditunjukkan pada tabel 5.2 dibawah ini :

5.2.2.1 Map Based Movement

Tabel 5.4 Hasil *Overhead Ratio* Pada Semua Skenario

Rute	Routing	Overhead Ratio
1	<i>Epidemic</i>	6.0938
	<i>Spray And Wait</i>	3.7396
2	<i>Epidemic</i>	5.8776
	<i>Spray And Wait</i>	3.8129
3	<i>Epidemic</i>	6.9328
	<i>Spray And Wait</i>	3.7899



Gambar 5.7 Grafik *Overhead Ratio* Pada Semua Skenario Pengujian

Pada Gambar 5.7, terlihat bahwa hasil pengujian terhadap *Overhead Ratio* untuk semua ukuran pesan 300 MB, 600 MB, dan 1,2 GB menggunakan protokol *routing Epidemic* pada Skenario 1 berdasarkan mobilitas *Map Based Movement* sebesar 6.0938 pesan dan pada protokol *Spray And Wait* sebesar

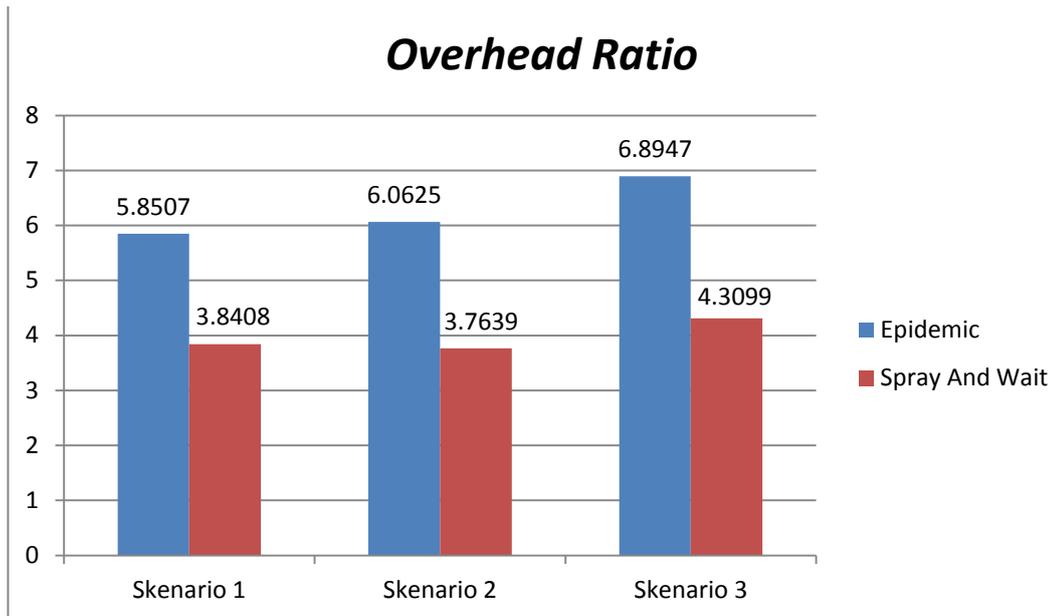
3.7396 pesan. Hasil Pengujian pada Skenario 2 menggunakan protokol *routing Epidemic* sebesar 5.8776 pesan dan pada *routing* protokol *Spray And Wait* sebesar 3.8129 pesan. Kemudian hasil pengujian pada Skenario 3 dengan menggunakan protokol *routing Epidemic* sebesar 6.9328 pesan dan sedangkan pada protokol *routing Spray And Wait* sebesar 3.7899 pesan.

Hasil dan analisis pengujian terhadap *Overhead Ratio* dari seluruh Skenario adalah pengujian pada protokol *routing Epidemic* memiliki *Overhead Ratio* tertinggi daripada protokol *routing Spray And Wait* terhadap ketiga percobaan pengujian Skenario yang telah dilakukan. Hal ini dikarenakan protokol *routing Epidemic* memberikan seluruh pesan yang dilewati oleh jaringan, sehingga protokol *routing* ini menyebabkan terjadinya jumlah *Overhead* yang sangat besar. Dan sedangkan pada *routing* protokol *Spray And Wait* memiliki *Overhead Ratio* yang sangat rendah dibandingkan dengan protokol *routing Epidemic* disebabkan pada proses pengiriman pesan terhadap protokol *routing Spray And Wait* tidak membutuhkan banyak proses yang dilakukan, dimana setiap kali bertemu dengan *node*, maka pesan akan di transfer ke *node* tersebut . Jika Semakin rendah nilai pada *Overhead Ratio*, maka kemampuan kinerja protokol *routing* semakin baik dalam mereplikasi paket (Liu, C., 2009).

5.2.2.2 Open Shortest Path

Tabel 5.5 Hasil *Overhead Ratio* Pada Semua Skenario

Rute	<i>Routing</i>	<i>Overhead Ratio</i>
1	<i>Epidemic</i>	5.8507
	<i>Spray And Wait</i>	3.8408
2	<i>Epidemic</i>	6.0625
	<i>Spray And Wait</i>	3.7639
3	<i>Epidemic</i>	6.8947
	<i>Spray And Wait</i>	4.3099



Gambar 5.8 Grafik *Overhead Ratio* Pada Semua Skenario Pengujian

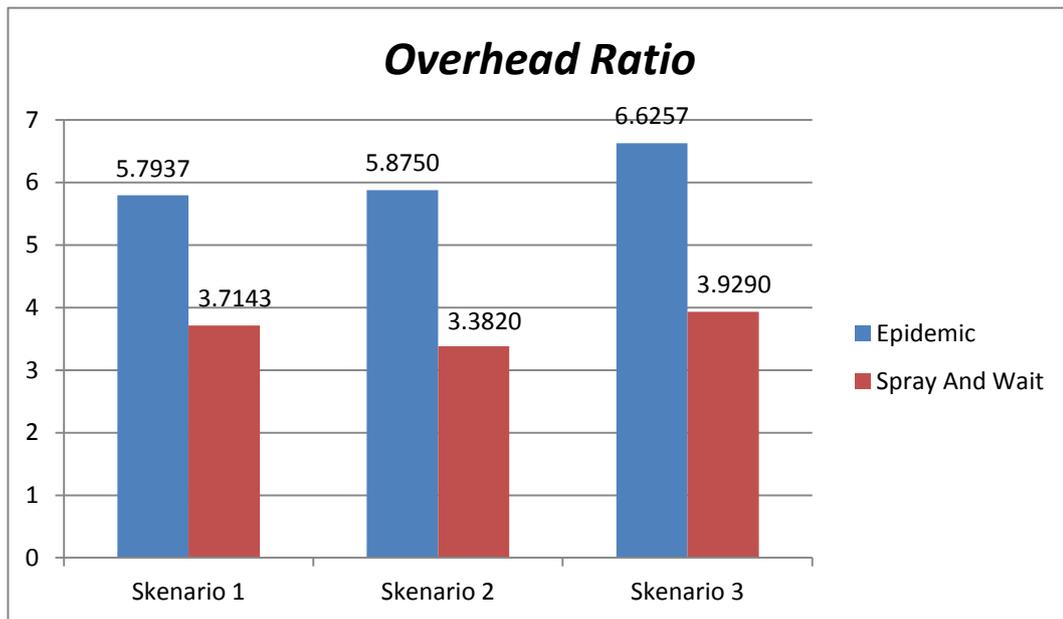
Pada Gambar 5.8, terlihat bahwa hasil pengujian terhadap *Overhead Ratio* untuk semua ukuran pesan 300 MB, 600 MB, dan 1,2 GB menggunakan protokol *routing Epidemic* pada Skenario 1 berdasarkan mobilitas *Open Shortest Path* sebesar 5.8507 pesan dan pada protokol *Spray And Wait* sebesar 3.8408 pesan. Hasil Pengujian pada Skenario 2 menggunakan protokol *routing Epidemic* sebesar 6.0625 pesan dan pada *routing* protokol *Spray And Wait* sebesar 3.7639 pesan. Kemudian hasil pengujian pada Skenario 3 dengan menggunakan protokol *routing Epidemic* sebesar 6.8947 pesan dan sedangkan pada protokol *routing Spray And Wait* sebesar 4.3099 pesan.

Hasil dan analisis pengujian terhadap *Overhead Ratio* dari seluruh Skenario adalah pengujian pada protokol *routing Epidemic* memiliki *Overhead Ratio* tertinggi daripada protokol *routing Spray And Wait* terhadap ketiga percobaan pengujian Skenario yang telah dilakukan. Hal ini dikarenakan protokol *routing Epidemic* memberikan seluruh pesan yang dilewati oleh jaringan, sehingga protokol *routing* ini menyebabkan terjadinya jumlah *Overhead* yang sangat besar. Dan sedangkan pada *routing* protokol *Spray And Wait* memiliki *Overhead Ratio* yang sangat rendah dibandingkan dengan protokol *routing Epidemic* disebabkan pada proses pengiriman pesan terhadap protokol *routing Spray And Wait* tidak membutuhkan banyak proses yang dilakukan, dimana setiap kali bertemu dengan *node*, maka pesan akan di transfer ke *node* tersebut. Jika Semakin rendah nilai pada *Overhead Ratio*, maka kemampuan kinerja protokol *routing* semakin baik dalam mereplikasi paket (Liu, C., 2009).

5.2.2.3 Map Route Movement

Tabel 5.6 Hasil *Overhead Ratio* Pada Semua Skenario

Rute	Routing	Overhead Ratio
1	Epidemic	5.7937
	Spray And Wait	3.7143
2	Epidemic	5.8750
	Spray And Wait	3.3820
3	Epidemic	6.6257
	spray And Wait	3.9290



Gambar 5.9 Grafik *Overhead Ratio* Pada Semua Skenario Pengujian

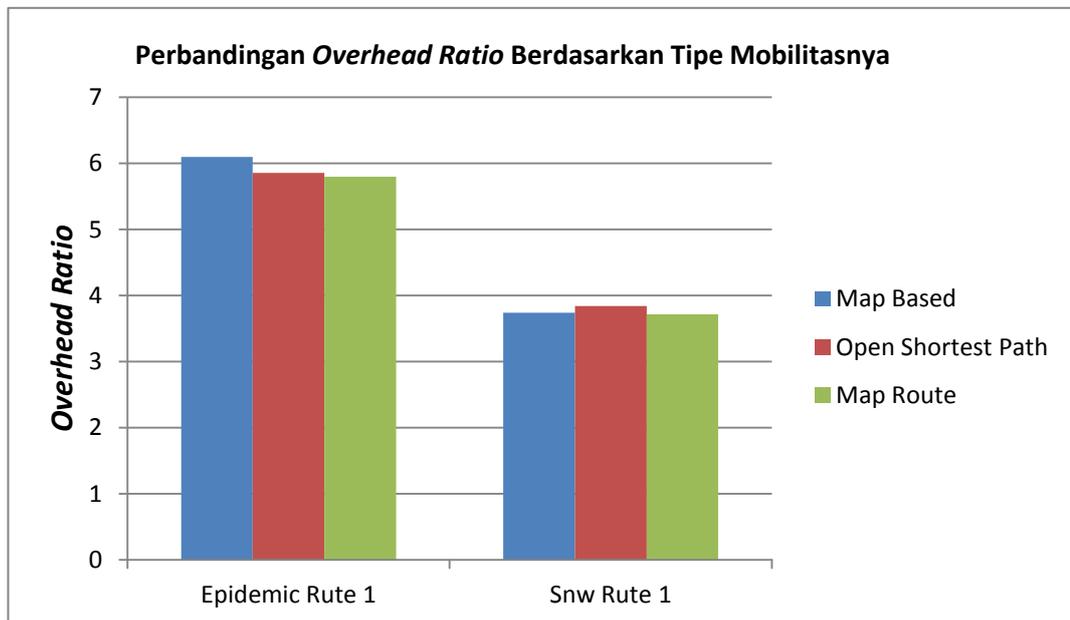
Pada Gambar 5.9, terlihat bahwa hasil pengujian terhadap *Overhead Ratio* untuk semua ukuran pesan 300 MB, 600 MB, dan 1,2 GB menggunakan protokol routing *Epidemic* pada Skenario 1 berdasarkan mobilitas *Map Route Movement* sebesar 5.7937 pesan dan pada protokol *Spray And Wait* sebesar 3.7143 pesan. Hasil Pengujian pada Skenario 2 menggunakan protokol routing *Epidemic* sebesar 5.8750 pesan dan pada routing protokol *Spray And Wait* sebesar 3.3820 pesan. Kemudian hasil pengujian pada Skenario 3 dengan menggunakan protokol routing *Epidemic* sebesar 6.6257 pesan dan sedangkan pada protokol routing *Spray And Wait* sebesar 3.9290 pesan.

Hasil dan analisis pengujian terhadap *Overhead Ratio* dari seluruh Skenario adalah pengujian pada protokol routing *Epidemic* memiliki *Overhead Ratio* tertinggi daripada protokol routing *Spray And Wait* terhadap ketiga percobaan pengujian Skenario yang telah dilakukan. Hal ini dikarenakan protokol routing *Epidemic* memberikan seluruh pesan yang dilewati oleh jaringan, sehingga

protokol *routing* ini menyebabkan terjadinya jumlah *Overhead* yang sangat besar. Dan sedangkan pada *routing* protokol *Spray And Wait* memiliki *Overhead Ratio* yang sangat rendah dibandingkan dengan protokol *routing Epidemic* disebabkan pada proses pengiriman pesan terhadap protokol *routing Spray And Wait* tidak membutuhkan banyak proses yang dilakukan, dimana setiap kali bertemu dengan *node*, maka pesan akan di transfer ke *node* tersebut . Jika Semakin rendah nilai pada *Overhead Ratio*, maka kemampuan kinerja protokol *routing* semakin baik dalam mereplikasi paket (Liu, C., 2009).

5.2.2.4 Perbandingan *Overhead Ratio* terhadap tipe Mobilitas

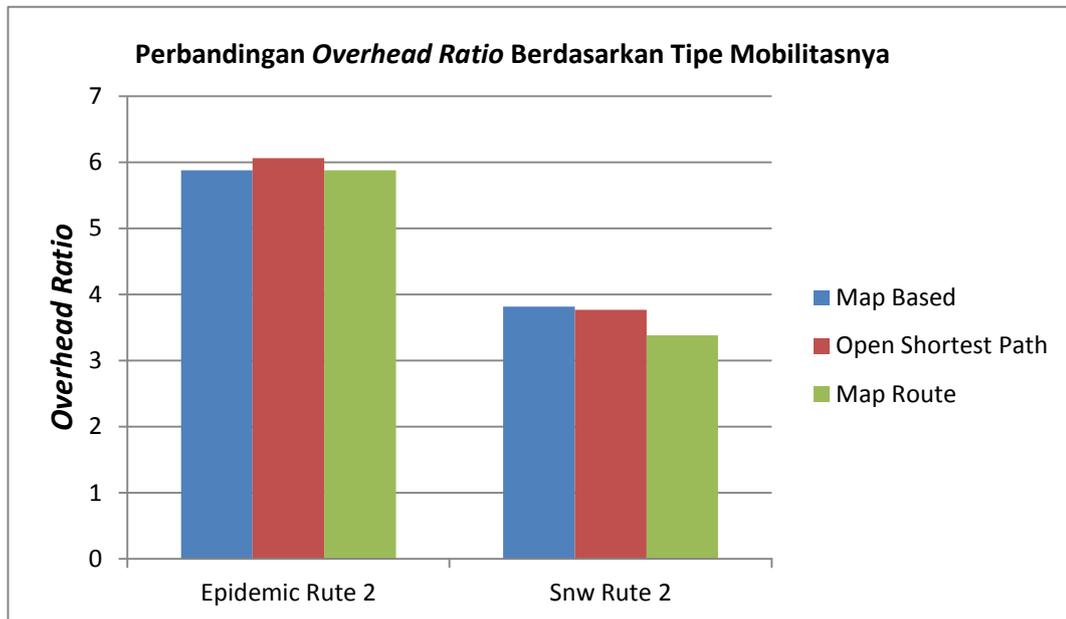
A. Skenario 1



Gambar 5.10 Grafik Perbandingan *Overhead Ratio* Berdasarkan Tipe Mobilitasnya (Skenario 1)

Pada Gambar 5.10, terlihat bahwa hasil perbandingan *Overhead Ratio* pada protokol *routing Epidemic* dan *Spray And Wait* pada setiap Skenario Rute dengan nilai masing – masing pada skenario Rute 1 menggunakan protokol *routing Epidemic* sebesar 6.0938 pesan pada mobilitas *Map Based Movement*, 5.8507 pesan pada mobilitas *Open Shortest Path* dan 5.7937 pesan pada mobilitas *Map Route Movement*. Kemudian untuk protokol *routing Spray And Wait (Snw)* sebesar 3.7396 pesan pada mobilitas *Map Based Movement*, 3.8408 pesan pada mobilitas *Open Shortest Path*, dan 3.7143 pesan pada mobilitas *Map Route Movement*.

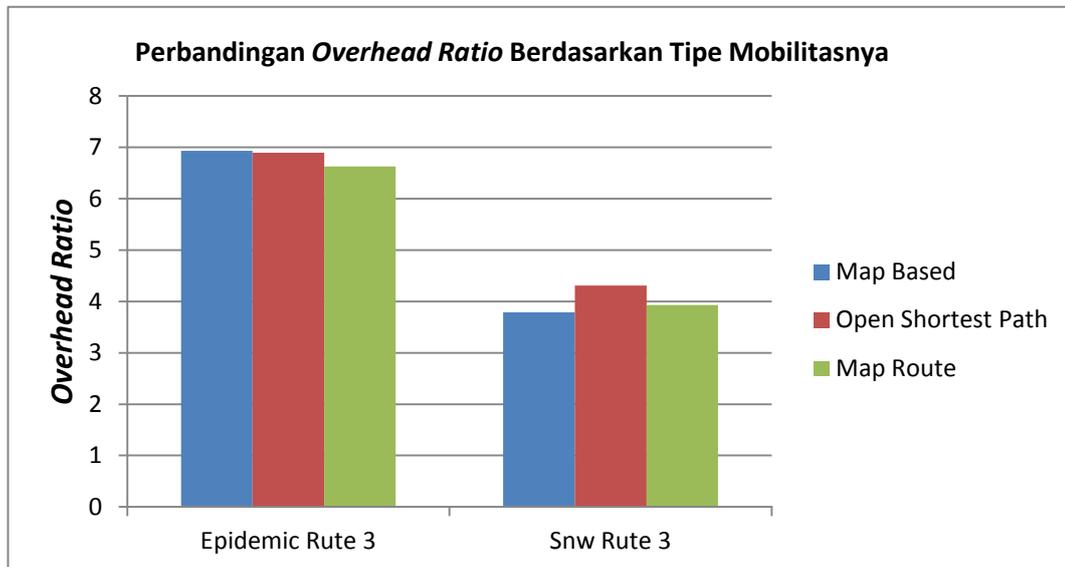
B. Skenario 2



Gambar 5.11 Grafik Perbandingan *Overhead Ratio* Berdasarkan Tipe Mobilitasnya (Skenario 2)

Pada Gambar 5.11, terlihat bahwa *Overhead Ratio* Skenario Rute 2 menggunakan protokol *routing Epidemic* sebesar 5.8776 pesan pada mobilitas *Map Based Movement*, 6.0625 pesan pada mobilitas *Open Shortest Path* dan 5.8750 pesan pada mobilitas *Map Route Movement*. kemudian untuk protokol *routing Spray And Wait (Snw)* sebesar 3.8129 pesan pada mobilitas *Map Based Movement*, 3.7639 pesan pada mobilitas *Open Shortest Path*, dan 3.3820 pesan pada mobilitas *Map Route Movement*.

C. Skenario 3



Gambar 5.12 Grafik Perbandingan *Overhead Ratio* Berdasarkan Tipe Mobilitasnya (Skenario 3)

Pada Skenario Rute 3 menggunakan protokol *routing Epidemic* sebesar 6.9328 pesan pada mobilitas *Map Based Movement*, 6.8947 pesan pada mobilitas *Open Shortest Path* dan 6.6257 pesan pada mobilitas *Map Route Movement*. Kemudian untuk protokol *routing Spray And Wait (Snw)* sebesar 3.7899 pesan pada mobilitas *Map Based Movement*, 4.3099 pesan pada mobilitas *Open Shortest Path*, dan 3.9290 pesan pada mobilitas *Map Route Movement*.

Berdasarkan dari hasil pengujian perbandingan parameter *Overhead Ratio* pada setiap skenario rute 1, skenario rute 2, dan skenario rute 3, pada beberapa mobilitasnya mengalami peningkatan *Overhead Ratio* tertinggi ada pada protokol *routing Epidemic* pada Skenario Rute 3 dengan menggunakan mobilitas *Map Based Movement* sebesar 6.9328 pesan. Dan hasil *Overhead Ratio* paling rendah ada pada Mobilitas *Map Route Movement* pada Skenario Rute 2 dengan menggunakan protokol *routing Spray And Wait (Snw)* sebesar 3.3820 pesan.

Hasil dan Analisis dari pengujian tersebut adalah mobilitas *Map Based Movement* mempunyai tingkat *Overhead Ratio* Paling tinggi dibandingkan tipe mobilitas lainnya seperti *Open Shortest Path* dan *Map Route Movement*. Hal ini disebabkan karena, protokol *routing Epidemic* pada mobilitas *Map Based Movement* menghasilkan jumlah pesan yang berlebihan ketika pesan sampai ditujuan dan membanjiri pesan pada jaringan dikarenakan efek dari penerapan

protokol *routing Epidemic*. Dan oleh sebab itu maka, tipe mobilitas *Map Based Movement* termasuk merupakan mobilitas yang sangat buruk dibandingkan dengan tipe mobilitas lainnya seperti *Open Shortest Path* dan *Map Route Movement*. Sehingga penerapan mobilitas terbaik ada pada mobilitas *Map Route Movement* pada protokol routing *Spray And Wait* dikarenakan mobilitas ini tidak menghasilkan pesan yang sangat berlebihan ketika sampai di tujuan dan membatasi jumlah *copy* pesan.

5.2.2.5 Pengaruh penggunaan tipe mobilitas untuk *Overhead Ratio* pada Protokol Routing:

- A. Pada mobilitas *Map Based Movement* menghasilkan dampak yang sangat buruk pada *Overhead Ratio* dengan menghasilkan tingkat *Overhead* yang sangat tinggi dibandingkan dengan mobilitas lainnya seperti *Open Shortest Path*, dan *Map Route Movement*. Dengan adanya tingkat *Overhead* yang tinggi, mobilitas ini kurang baik dalam hal proses pengiriman pesan karena didasari oleh penerapan protokol *Routing Epidemic* yang juga berperan dalam menghasilkan pesan yang berlebihan atau *Overflow* di tempat tujuan dikarenakan Protokol *Routing* ini memberikan seluruh pesan yang dilewati oleh jaringan dan juga disebabkan oleh atas penggunaan waktu dan jarak tempuh yang digunakan.
- B. Pada pengujian *Overhead Ratio* pada mobilitas *Open Shortest Path* mempunyai pengaruh dampak yang dihasilkan juga mempunyai tingkat jumlah pesan yang dihasilkan hampir sama banyaknya dengan mobilitas *Map Based Movement*. Sehingga pesan yang dihasilkan akan menjadi banyak dan *Overflow* pada suatu jaringan.
- C. Pada pengujian *Overhead Ratio* pada mobilitas *Map Route Movement* mempunyai pengaruh dampak yang dihasilkan menjadi sangat baik dalam hal menghasilkan pesan tidak sangat berlebihan ketika sampai ditujuan. Dalam hal ini protokol routing *Spray And Wait* juga berperan dalam mengatur dan membatasi jumlah salinan pesan pada mobilitas ini.

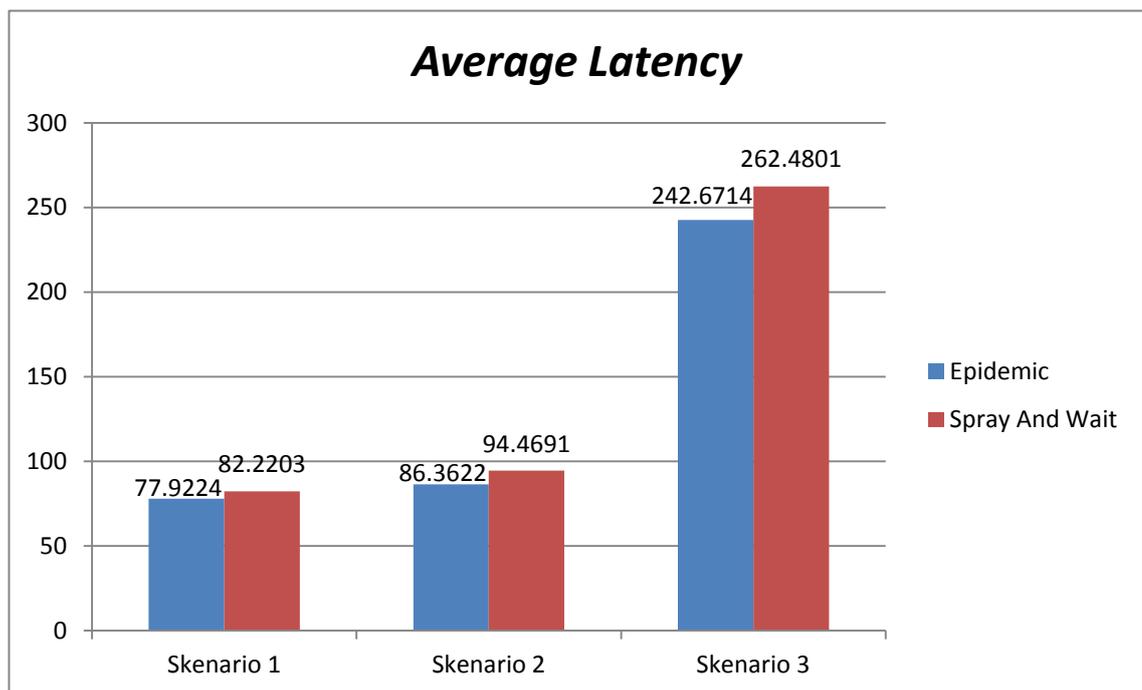
5.2.3 Average Latency

Hasil pengujian *Average Latency* terhadap ketiga skenario dapat ditunjukkan pada tabel-tabel berikut ini :

5.2.3.1 Map Based Movement

Tabel 5.7 Hasil *Average Latency* Pada Semua Skenario

Rute	Routing	Average Latency
1	<i>Epidemic</i>	77.9224
	<i>Spray And Wait</i>	82.2203
2	<i>Epidemic</i>	86.3622
	<i>Spray And Wait</i>	94.4691
3	<i>Epidemic</i>	242.6714
	<i>Spray And Wait</i>	262.4801



Gambar 5.13 Grafik *Average Latency* Pada Semua Skenario Pengujian

Pada Gambar 5.13, menunjukkan bahwa *Average Latency* terhadap ketiga skenario pengujian akan didapatkan sebuah hasil dari percobaan pada kedua protokol *routing* yang telah diujikan. Hasil pada protokol *routing Epidemic* pada

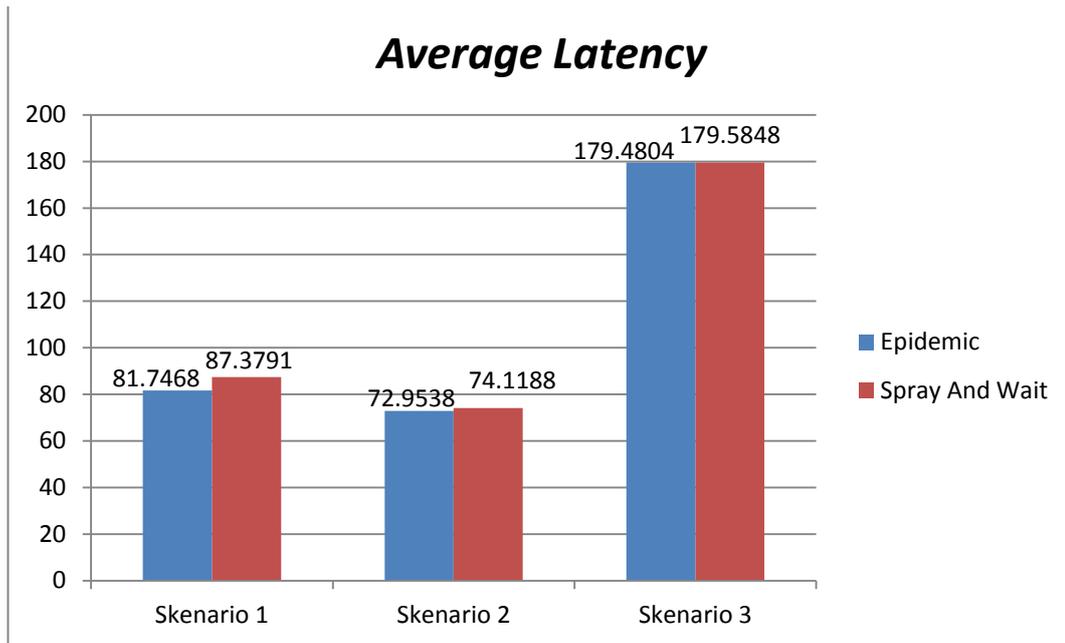
Skenario 1 berdasarkan mobilitas *Map Based Movement* sebesar 77.9224 ms dan pada *routing* protokol *Spray And Wait* sebesar 82.2203 ms. Pada hasil pengujian skenario 2 pada protokol *routing Epidemic* sebesar 86.3622 ms dan pada protokol *routing Spray And Wait* sebesar 94.4691 ms. Dan pada hasil pengujian skenario 3 pada protokol *routing Epidemic* sebesar 242.6714 ms dan pada protokol *routing Spray And Wait* sebesar 262.4801 ms.

Hasil dan analisis dari pengujian terhadap *Average Latency* dari ketiga skenario yang telah dilakukan, bahwa *Average Latency* terendah ada pada protokol *routing Epidemic* dan sedangkan *Average Latency* tertinggi ada pada protokol *routing Spray And Wait*. Hal ini disebabkan pada protokol *routing Spray And Wait* ketika saat *node* ingin mencoba untuk mengirimkan paket, maka *node* yang yang bertindak sebagai pengirim paket akan menunggu konfirmasi sampai paket telah diterima dan pada protokol *routing Spray And Wait* akan membatasi jumlah *copy* pengiriman. Jika semakin rendah *Average Latency* maka pengiriman pesan yang dilakukan akan semakin cepat.

5.2.3.2 Open Shortest Path

Tabel 5.8 Hasil *Average Latency* Pada Semua Skenario

Rute	<i>Routing</i>	<i>Average Latency</i>
1	<i>Epidemic</i>	81.7468
	<i>Spray And Wait</i>	87.3791
2	<i>Epidemic</i>	72.9538
	<i>Spray And Wait</i>	74.1188
3	<i>Epidemic</i>	179.4804
	<i>Spray And Wait</i>	179.5848



Gambar 5.14 Grafik Average Latency Pada Semua Skenario Pengujian

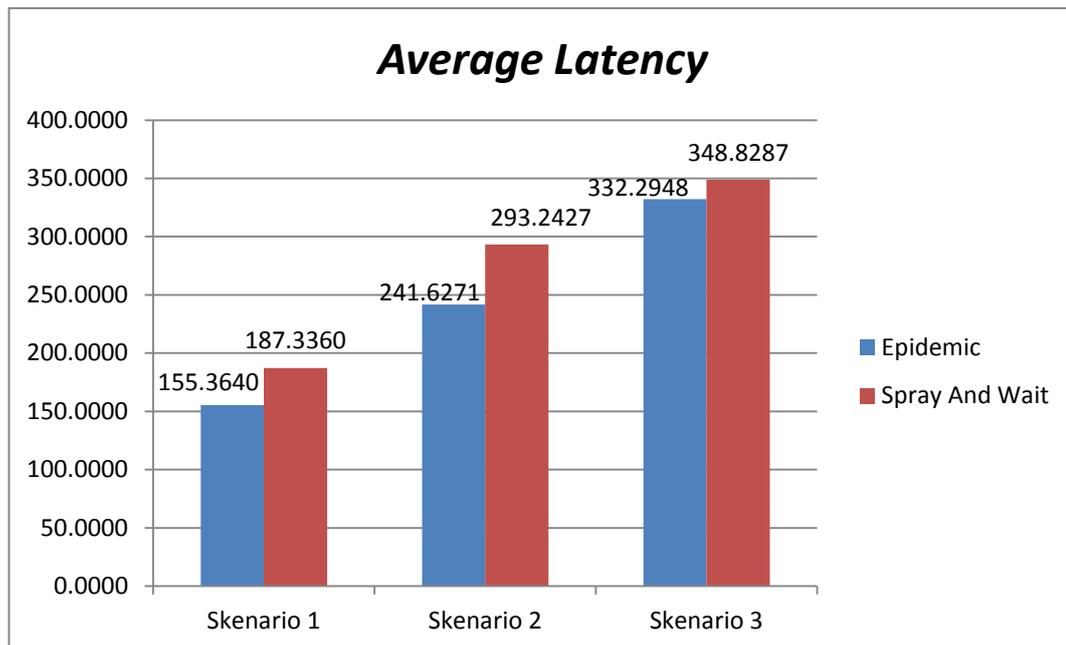
Pada Gambar 5.14, menunjukkan bahwa *Average Latency* terhadap ketiga skenario pengujian akan didapatkan sebuah hasil dari percobaan pada kedua protokol *routing* yang telah diujikan. Hasil pada protokol *routing Epidemic* pada Skenario 1 berdasarkan mobilitas *Open Shortest Path* sebesar 81.7468 ms dan pada *routing* protokol *Spray And Wait* sebesar 87.3791 ms. Pada hasil pengujian skenario 2 pada protokol *routing Epidemic* sebesar 72.9538 ms dan pada protokol *routing Spray And Wait* sebesar 74.1188 ms. Dan pada hasil pengujian skenario 3 pada protokol *routing Epidemic* sebesar 179.4804 ms dan pada protokol *routing Spray And Wait* sebesar 179.5848 ms.

Hasil dan analisis dari pengujian terhadap *Average Latency* dari ketiga skenario yang telah dilakukan, bahwa *Average Latency* terendah ada pada protokol *routing Epidemic* dan sedangkan *Average Latency* tertinggi ada pada protokol *routing Spray And Wait*. Hal ini disebabkan pada protokol *routing Spray And Wait* ketika saat *node* ingin mencoba untuk mengirimkan paket, maka *node* yang yang bertindak sebagai pengirim paket akan menunggu konfirmasi sampai paket telah diterima dan pada protokol *routing Spray And Wait* akan membatasi jumlah *copy* pengiriman. Jika semakin rendah *Average Latency* maka pengiriman pesan yang dilakukan akan semakin cepat.

5.2.3.3 Map Route Movement

Tabel 5.9 Hasil *Average Latency* Pada Semua Skenario

Rute	Routing	Average Latency
1	<i>Epidemic</i>	155.3640
	<i>Spray And Wait</i>	187.3360
2	<i>Epidemic</i>	241.6271
	<i>Spray And Wait</i>	293.2427
3	<i>Epidemic</i>	332.2948
	<i>Spray And Wait</i>	348.8287



Gambar 5.15 Grafik *Average Latency* Pada Semua Skenario Pengujian

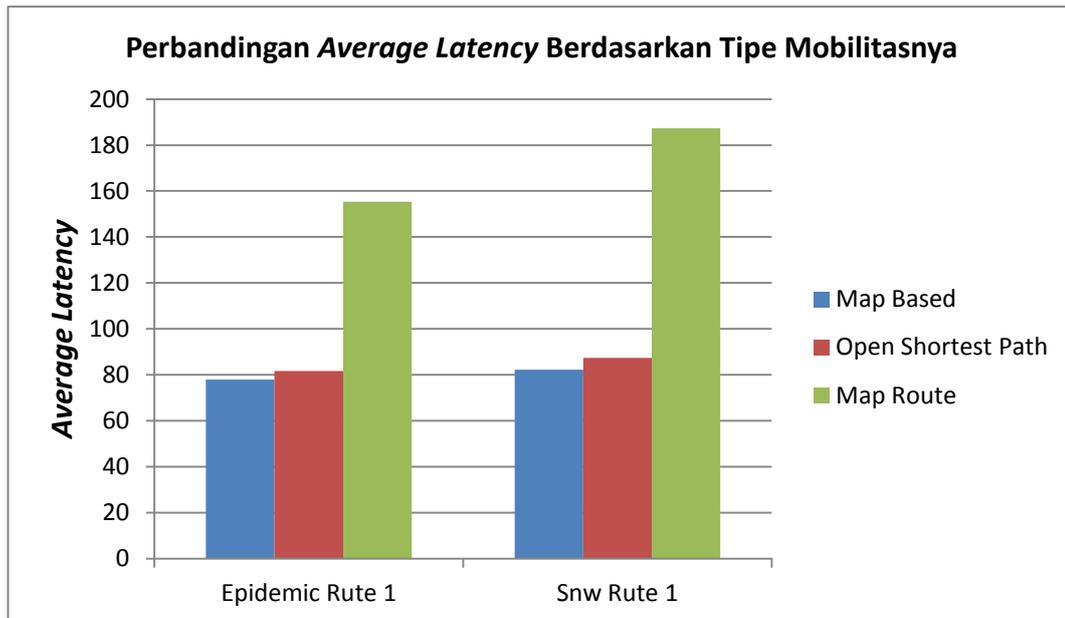
Pada Gambar 5.15, menunjukkan bahwa *Average Latency* terhadap ketiga skenario pengujian akan didapatkan sebuah hasil dari percobaan pada kedua protokol *routing* yang telah diujikan. Hasil pada protokol *routing Epidemic* pada Skenario 1 berdasarkan mobilitas *Map Route Movement* sebesar 155.3640 ms dan pada *routing* protokol *Spray And Wait* sebesar 187.3360 ms. Pada hasil pengujian skenario 2 pada protokol *routing Epidemic* sebesar 241.6271 ms dan pada protokol *routing Spray And Wait* sebesar 293.2427 ms. Dan pada hasil pengujian skenario 3 pada protokol *routing Epidemic* sebesar 332.2948 ms dan pada protokol *routing Spray And Wait* sebesar 348.8287 ms.

Hasil dan analisis dari pengujian terhadap *Average Latency* dari ketiga skenario yang telah dilakukan, bahwa *Average Latency* terendah ada pada protokol *routing Epidemic* dan sedangkan *Average Latency* tertinggi ada pada protokol *routing Spray And Wait*. Hal ini disebabkan pada protokol *routing Spray And Wait* ketika saat *node* ingin mencoba untuk mengirimkan paket, maka *node*

yang bertindak sebagai pengirim paket akan menunggu konfirmasi sampai paket telah diterima dan pada protokol *routing Spray And Wait* akan membatasi jumlah *copy* pengiriman. Jika semakin rendah *Average Latency* maka pengiriman pesan yang dilakukan akan semakin cepat.

5.2.3.4 Perbandingan *Average Latency* terhadap tipe Mobilitas

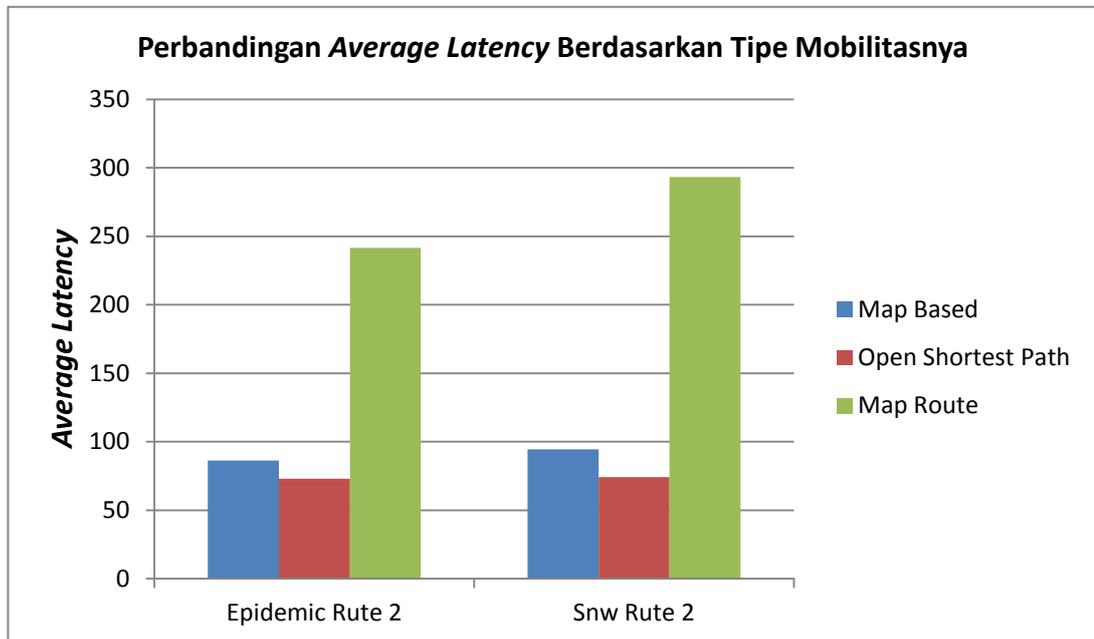
A. Skenario 1



Gambar 5.16 Grafik Perbandingan *Average Latency* Berdasarkan Tipe Mobilitasnya

Pada Gambar 5.16, terlihat bahwa hasil perbandingan *Average Latency* pada protokol *routing Epidemic* dan *Spray And Wait* pada setiap Skenario Rute dengan nilai masing – masing pada skenario Rute 1 menggunakan protokol *routing Epidemic* sebesar 77.9224 ms pada mobilitas *Map Based Movement*, 81.7468 ms pada mobilitas *Open Shortest Path* dan 155.3640 ms pada mobilitas *Map Route Movement*. Kemudian untuk protokol *routing Spray And Wait (Snw)* sebesar 82.2203 ms pada mobilitas *Map Based Movement*, 87.3791 ms pada mobilitas *Open Shortest Path*, dan 187.3360 ms pada mobilitas *Map Route Movement*.

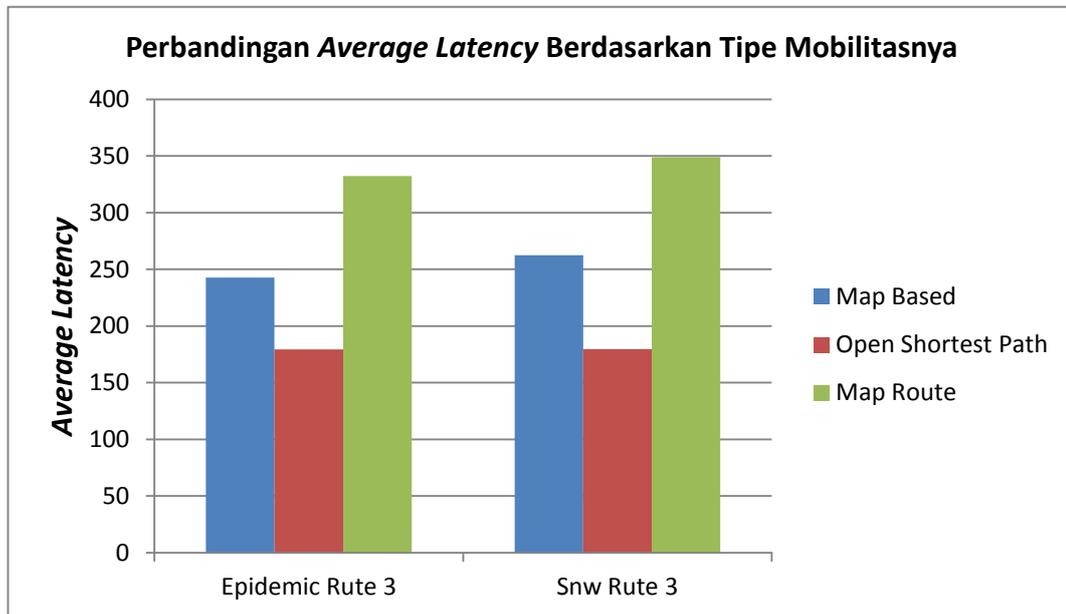
B. Skenario 2



Gambar 5.17 Grafik Perbandingan *Average Latency* Berdasarkan Tipe Mobilitasnya

Pada Gambar 5.17, terlihat bahwa hasil perbandingan Skenario Rute 2 menggunakan protokol *routing Epidemic* sebesar 86.3622 ms pada mobilitas *Map Based Movement*, 72.9538 ms pada mobilitas *Open Shortest Path* dan 241.6271 ms pada mobilitas *Map Route Movement*. Kemudian untuk protokol *routing Spray And Wait (Snw)* sebesar 94.4691 ms pada mobilitas *Map Based Movement*, 74.1188 ms pada mobilitas *Open Shortest Path*, dan 293.2427 ms pada mobilitas *Map Route Movement*.

C. Skenario 3



Gambar 5.18 Grafik Perbandingan *Average Latency* Berdasarkan Tipe Mobilitasnya

Pada Gambar 5.18, terlihat bahwa hasil perbandingan Skenario Rute 3 menggunakan protokol *routing Epidemic* sebesar 242.6714 ms pada mobilitas *Map Based Movement*, 179.4804 ms pada mobilitas *Open Shortest Path* dan 332.2948 ms pada mobilitas *Map Route Movement*. Kemudian untuk protokol *routing Spray And Wait (Snw)* sebesar 262.4801 ms pada mobilitas *Map Based Movement*, 179.5848 ms pada mobilitas *Open Shortest Path*, dan 348.8287 ms pada mobilitas *Map Route Movement*.

Berdasarkan dari hasil pengujian perbandingan seluruh skenario yaitu, skenario rute 1, skenario rute 2, dan skenario rute 3 dengan menggunakan parameter *Average Latency* pada beberapa mobilitas, terlihat bahwa *Average Latency* tertinggi ada pada mobilitas *Map Route Movement* sebesar 348.8287 ms dan sedangkan hasil *Average Latency* terendah ada pada mobilitas *Open Shortest Path* sebesar 72.9538 ms dengan menggunakan protokol *Epidemic* dan *Spray And Wait (Snw)*.

Hasil dan Analisis dari pengujian pada seluruh skenario tersebut adalah mobilitas *Open Shortest Path* merupakan mobilitas yang paling baik dibandingkan dengan mobilitas *Map Based Movement* dan *Map Route Movement* dikarenakan mobilitas *Open Shortest Path* pergerakan nodenya dari *Source* ke *Destination* menggunakan jalur terpendek sehingga pergerakan node akan lebih cepat sampai di tujuan dengan tingkat *Average Latency* yang rendah. Dan sedangkan pada mobilitas *Map Route Movement* merupakan mobilitas yang buruk, dikarenakan mobilitas ini mempunyai tingkat *Average Latency* yang tinggi. Sehingga mobilitas *Map Route Movement* dalam pengiriman pesan dari *Source*

ke *Destination* akan membutuhkan tingkat kecepatan waktu yang lama hingga pesan sampai ke *Destination*.

5.2.3.5 Pengaruh penggunaan tipe mobilitas untuk *Average Latency* pada Protokol *Routing*:

- A. Pada pengujian *Average Latency* yang dihasilkan tidak terlalu signifikan peningkatannya. Sehingga pada mobilitas *Map Based Movement*, akan terlihat hasil yang stabil dibandingkan pada mobilitas lainnya. saat proses pengiriman pesan ke tujuan.
- B. Pada mobilitas *Open Shortest Path* mempunyai pengaruh dampak yang dihasilkan pada tingkat *Average Latency* sangat rendah dibandingkan dengan mobilitas lainnya seperti *Map Based Movement* dan *Map Route Movement*. Hal ini disebabkan pada mobilitas ini karena pergerakan node dengan menggunakan jalur terpendek, maka tingkat kecepatan dalam hal proses pengiriman menjadi lebih cepat dan baik. Sehingga pada mobilitas ini untuk halnya kecepatan lebih baik dari mobilitas *Map Based Movement* dan *Map Route Movement* dan juga disebabkan oleh jarak tempuh yang dilalui dan waktu tempuh yang dibutuhkan hingga sampai ke tujuan. Semakin jauh jarak tempuh dan waktu tempuh pada mobilitas ini maka mobilitas ini lebih tepat digunakan untuk dalam hal kecepatan dalam proses pengiriman pesan akan menjadi lebih cepat sampai di tujuan.
- C. Pada mobilitas *Map Route Movement* mempunyai pengaruh dampak yang dihasilkan pada tingkat *Average Latency* sangat tinggi dan buruk. Sehingga pada mobilitas ini pengiriman pesan ke tujuan akan menjadi sangat lama dan dalam hal kecepatan mobilitas ini menjadi kurang maksimal dalam proses pengiriman pesan ke tujuan. Hal ini disebabkan mobilitas ini pergerakan node hanya mengikuti jalur yang telah ditentukan. Sehingga dalam proses pengiriman pesan ke tujuan dengan menggunakan mobilitas ini kurang baik dalam hal kecepatan.

5.2.4 Analisis Tiap Skenario

Analisis tiap skenario menggunakan protokol *routing Epidemic* dan *Spray And Wait* pada skenario 1, Skenario 2, dan Skenario 3 dengan berdasarkan beberapa tipe Mobilitas yaitu *Map Based Movement*, *Open Shortest Path* dan *Map Route Movement*.

5.2.4.1 Skenario 1

Pada Skenario 1 telah dilakukan selama 45 menit pada simulator *The One* dan dengan menggunakan ukuran pesan yang berbeda yaitu sebesar 300 MB, 600 MB, 1,2 GB. Berikut analisis skenario 1 berdasarkan Mobilitas dibawah ini :

1. Mobilitas (*Map Based Movement*)

a. Analisis Skenario 1 Pada Protokol *Routing Epidemic*

Tabel 5.10 Hasil Skenario 1 (*Epidemic*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Blt_0	46	46	46	138	64	68	454	0.9412	6.0938	77.9224
Blt_1	44	44	44	132						
Blt_2	44	44	44	132						
Blt_3	63	63	63	189						
Blt_4	66	66	66	198						
Blt_5	66	66	66	198						

Pada tabel 5.10 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 1 dengan menggunakan protokol *routing Epidemic* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.10 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 1 pada protokol *routing Epidemic* sebanyak 68 pesan, *message relay* yang dihasilkan sebanyak 454 pesan, *Delivery Probability* yang dihasilkan sebesar 0.9412 pesan, *Overhead Ratio* yang dihasilkan sebesar 6.0938 pesan, dan *Average Latency* yang dihasilkan sebesar 77.9224 ms.

b. Analisis Skenario 1 Pada Protokol *Routing Spray And Wait*

Tabel 5.11 Hasil Skenario 1 (*Spray And Wait*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Blt_0	40	40	39	119	64	68	303	0.9412	3.7396	82.2203
Blt_1	39	40	40	119						
Blt_2	34	34	33	101						
Blt_3	42	42	41	125						
Blt_4	54	53	54	161						
Blt_5	45	45	44	134						

Pada tabel 5.11 telah dijelaskan mengenai hasil pengujian yang dilakukan dari skenario 1 dengan menggunakan protokol *routing Spray And Wait* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.11 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 1 pada protokol *routing Spray And Wait* sebanyak 68 pesan, *message relay* yang dihasilkan sebanyak 303 pesan, *Delivery Probability* yang dihasilkan sebesar 0.9412 pesan, *Overhead Ratio* yang dihasilkan sebesar 3.7396 pesan, dan *Average Latency* yang dihasilkan sebesar 82.2203 ms.

2. Mobilitas (*Open Shortest Path*)

a. Analisis Skenario 1 Pada Protokol *Routing Epidemic*

Tabel 5.12 Hasil Skenario 1 (*Epidemic*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Blt_0	47	47	47	141	67	68	459	0.9853	5.8507	81.7468
Blt_1	41	41	41	123						
Blt_2	44	44	44	132						
Blt_3	68	68	68	204						
Blt_4	68	68	68	204						
Blt_5	62	62	62	186						

Pada tabel 5.12 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 1 dengan menggunakan protokol *routing Epidemic* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.12 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 1 pada protokol *routing Epidemic* sebanyak 68 pesan, *message relay* yang dihasilkan sebanyak 459 pesan, *Delivery Probability* yang dihasilkan sebesar 0.9853 pesan, *Overhead Ratio* yang dihasilkan sebesar 5.8507 pesan, dan *Average Latency* yang dihasilkan sebesar 81.7468 ms.

b. Analisis Skenario 1 Pada Protokol *Routing Spray And Wait*

Tabel 5.13 Hasil Skenario 1 (*Spray And Wait*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Blt_0	40	40	40	120	67	68	324	0.9853	3.8408	87.3791
Blt_1	38	38	38	114						
Blt_2	42	42	43	127						
Blt_3	55	55	55	165						
Blt_4	54	54	57	165						
Blt_5	41	41	40	122						

Pada tabel 5.13 telah dijelaskan mengenai hasil pengujian yang dilakukan dari skenario 1 dengan menggunakan protokol *routing Spray And Wait* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.13 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 1 pada protokol *routing Spray And Wait* sebanyak 68 pesan, *message relay* yang dihasilkan sebanyak 324 pesan, *Delivery Probability* yang dihasilkan sebesar 0.9853 pesan, *Overhead Ratio* yang dihasilkan sebesar 3.8408 pesan, dan *Average Latency* yang dihasilkan sebesar 87.3791 ms.

3. Mobilitas (*Map Route Movement*)

a. Analisis Skenario 1 Pada Protokol *Routing Epidemic*

Tabel 5.14 Hasil Skenario 1 (*Epidemic*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Blt_0	42	42	42	126	63	68	428	0.9265	5.7937	155.3640
Blt_1	41	41	41	123						

Blt_2	44	44	44	132						
Blt_3	62	62	62	186						
Blt_4	63	63	63	189						
Blt_5	63	63	63	189						

Pada tabel 5.14 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 1 dengan menggunakan protokol *routing Epidemic* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.14 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 1 pada protokol *routing Epidemic* sebanyak 68 pesan, *message relay* yang dihasilkan sebanyak 428 pesan, *Delivery Probability* yang dihasilkan sebesar 0.9265 pesan, *Overhead Ratio* yang dihasilkan sebesar 5.7937 pesan, dan *Average Latency* yang dihasilkan sebesar 155.3640 ms.

b. Analisis Skenario 1 Pada Protokol *Routing Spray And Wait*

Tabel 5.15 Hasil Skenario 1 (*Spray And Wait*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	<i>Message Create</i>	<i>Message Relay</i>	<i>Delivery Probability</i>	<i>Overhead Ratio</i>	<i>Average Latency</i>
	300 MB	600 MB	1,2 GB							
Blt_0	38	38	38	114	63	68	297	0.9265	3.7143	187.3360
Blt_1	37	37	37	111						
Blt_2	39	39	39	117						
Blt_3	33	33	33	99						
Blt_4	51	51	49	151						
Blt_5	45	45	45	135						

Pada tabel 5.15 telah dijelaskan mengenai hasil pengujian yang dilakukan dari skenario 1 dengan menggunakan protokol *routing Spray And Wait* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.15 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 1 pada protokol *routing Spray And Wait* sebanyak 68 pesan, *message relay* yang dihasilkan sebanyak 297 pesan, *Delivery Probability* yang dihasilkan sebesar 0.9265 pesan, *Overhead Ratio* yang dihasilkan sebesar 3.7143 pesan, dan *Average Latency* yang dihasilkan sebesar 187.3360 ms.

5.2.4.2 Skenario 2

Pada Skenario 2 telah dilakukan selama 65 menit pada simulator *The One* dan dengan menggunakan ukuran pesan yang bervariasi yaitu sebesar 300 MB, 600 MB, 1,2 GB. Berikut analisis skenario 2 berdasarkan Mobilitas dibawah ini :

1. Mobilitas (*Map Based Movement*)

a. Analisis Skenario 2 Pada Protokol *Routing Epidemic*

Tabel 5.16 Hasil Skenario 2 (*Epidemic*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Tlg_0	70	70	70	210	98	98	674	1	5.8776	86.3622
Tlg_1	61	61	61	183						
Tlg_2	65	65	65	195						
Tlg_3	94	94	94	282						
Tlg_4	97	97	97	291						
Tlg_5	96	96	96	288						

Pada tabel 5.16 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 2 dengan menggunakan protokol *routing Epidemic* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.16 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 2 pada protokol *routing Epidemic* sebanyak 98 pesan, *message relay* yang dihasilkan sebanyak 674 pesan, *Delivery Probability* yang dihasilkan sebesar 1 pesan, *Overhead Ratio* yang dihasilkan sebesar 5.8776 pesan, dan *Average Latency* yang dihasilkan sebesar 86.3622 ms.

b. Analisis Skenario 2 Pada Protokol *Routing Spray And Wait*

Tabel 5.17 Hasil Skenario 2 (*Spray And Wait*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Tlg_0	65	65	64	194	98	98	471	1	3.8129	94.4691
Tlg_1	48	48	50	146						
Tlg_2	57	57	58	172						
Tlg_3	70	71	73	214						
Tlg_4	68	69	69	206						
Tlg_5	70	71	71	212						

Pada tabel 5.17 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 2 dengan menggunakan protokol *routing Spray And Wait* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.17 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 2 pada protokol *routing Epidemic* sebanyak 98 pesan, *Message Relay* yang dihasilkan sebanyak 471 pesan, *Delivery Probability* yang dihasilkan sebesar 1 pesan, *Overhead Ratio* yang dihasilkan sebesar 3.8129 pesan, dan *Average Latency* yang dihasilkan sebesar 94.4691 ms.

2. Mobilitas (*Open Shortest Path*)

a. Analisis Skenario 2 Pada Protokol *Routing Epidemic*

Tabel 5.18 Hasil Skenario 2 (*Epidemic*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Tlg_0	70	70	70	210	96	98	678	0.9796	6.0625	72.9538
Tlg_1	61	61	61	183						
Tlg_2	64	64	64	192						
Tlg_3	97	97	97	291						
Tlg_4	97	97	97	291						
Tlg_5	97	97	97	291						

Pada tabel 5.18 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 2 dengan menggunakan protokol *routing Epidemic* dengan ukuran

pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.18 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 2 pada protokol *routing Epidemic* sebanyak 98 pesan, *message relay* yang dihasilkan sebanyak 678 pesan, *Delivery Probability* yang dihasilkan sebesar 0.9796 pesan, *Overhead Ratio* yang dihasilkan sebesar 6.0625 pesan, dan *Average Latency* yang dihasilkan sebesar 72.9538 ms.

b. Analisis Skenario 2 Pada Protokol *Routing Spray And Wait*

Tabel 5.19 Hasil Skenario 2 (*Spray And Wait*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Tlg_0	59	60	60	179	96	98	457	0.9796	3.7639	74.1188
Tlg_1	53	53	53	159						
Tlg_2	58	58	58	174						
Tlg_3	65	65	65	195						
Tlg_4	72	71	72	215						
Tlg_5	70	70	70	210						

Pada tabel 5.19 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 2 dengan menggunakan protokol *routing Spray And Wait* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.19 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 2 pada protokol *routing Epidemic* sebanyak 98 pesan, *Message Relay* yang dihasilkan sebanyak 457 pesan, *Delivery Probability* yang dihasilkan sebesar 0.9796 pesan, *Overhead Ratio* yang dihasilkan sebesar 3.7639 pesan, dan *Average Latency* yang dihasilkan sebesar 74.1188 ms.

3. Mobilitas (*Map Route Movement*)

a. Analisis Skenario 2 Pada Protokol *Routing Epidemic*

Tabel 5.20 Hasil Skenario 2 (*Epidemic*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Tlg_0	69	69	69	207	96	98	660	0.9796	5.8750	241.6271
Tlg_1	61	61	61	183						
Tlg_2	62	62	62	186						

Tlg_3	97	97	97	291						
Tlg_4	96	96	96	288						
Tlg_5	94	94	94	282						

Pada tabel 5.20 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 2 dengan menggunakan protokol *routing Epidemic* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.20 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 2 pada protokol *routing Epidemic* sebanyak 98 pesan, *message relay* yang dihasilkan sebanyak 660 pesan, *Delivery Probability* yang dihasilkan sebesar 0.9796 pesan, *Overhead Ratio* yang dihasilkan sebesar 5.8750 pesan, dan *Average Latency* yang dihasilkan sebesar 241.6271 ms.

b. Analisis Skenario 2 Pada Protokol *Routing Spray And Wait*

Tabel 5.21 Hasil Skenario 2 (*Spray And Wait*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	<i>Message Create</i>	<i>Message Relay</i>	<i>Delivery Probability</i>	<i>Overhead Ratio</i>	<i>Average Latency</i>
	300 MB	600 MB	1,2 GB							
Tlg_0	67	67	67	201	96	98	420	0.9796	3.3820	293.2427
Tlg_1	44	45	44	133						
Tlg_2	31	31	31	93						
Tlg_3	90	90	87	267						
Tlg_4	72	74	70	216						
Tlg_5	44	44	46	134						

Pada tabel 5.21 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 2 dengan menggunakan protokol *routing Spray And Wait* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.20 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 2 pada protokol *routing Epidemic* sebanyak 98 pesan, *Message Relay* yang dihasilkan sebanyak 420 pesan, *Delivery Probability* yang dihasilkan sebesar 0.9796 pesan, *Overhead Ratio* yang dihasilkan sebesar 3.3820 pesan, dan *Average Latency* yang dihasilkan sebesar 293.2427 ms.

5.2.4.3 Skenario 3

Pada Skenario 3 telah dilakukan selama 85 menit pada simulator *The One* dan dengan menggunakan ukuran pesan yang bervariasi yaitu sebesar 300 MB, 600 MB, 1,2 GB. Berikut analisis skenario 3 berdasarkan Mobilitas dibawah ini :

1. Mobilitas (Map Based Movement)

a. Analisis Skenario 3 Pada Protokol *Routing Epidemic*

Tabel 5.22 Hasil Skenario 3 (*Epidemic*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Trglek_0	90	90	90	270	119	129	944	0.9225	3.7899	242.6714
Trglek_1	75	75	75	225						
Trglek_2	84	84	84	252						
Trglek_3	121	121	121	363						
Trglek_4	119	119	119	357						
Trglek_5	123	123	123	369						

Pada tabel 5.22 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 3 dengan menggunakan protokol *routing Epidemic* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.21 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 3 pada protokol *routing Epidemic* sebanyak 129 pesan, *message relay* yang dihasilkan sebanyak 944 pesan, *Delivery Probability* yang dihasilkan sebesar 0.9225 pesan, *Overhead Ratio* yang dihasilkan sebesar 3.7899 pesan, dan *Average Latency* yang dihasilkan sebesar 242.6714 ms.

b. Analisis Skenario 3 Pada Protokol *Routing Spray And Wait*

Tabel 5.23 Hasil Skenario 3 (*Spray And Wait*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Trglek_0	73	73	74	220	119	129	570	0.9225	3.7899	262.4801
Trglek_1	68	68	68	204						
Trglek_2	72	72	72	216						
Trglek_3	77	76	78	231						
Trglek_4	76	75	75	226						
Trglek_5	69	69	65	203						

Pada tabel 5.23 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 3 dengan menggunakan protokol *routing Spray And Wait* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.22 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 3 pada protokol *routing Spray And Wait* sebanyak 129 pesan, *message relay* yang dihasilkan sebanyak 570 pesan, *Delivery Probability* yang dihasilkan sebesar 0.9225 pesan, *Overhead Ratio* yang dihasilkan sebesar 3.7899 pesan, dan *Average Latency* yang dihasilkan sebesar 262.4801 ms.

2. Mobilitas (*Open Shortest Path*)

a. Analisis Skenario 3 Pada Protokol *Routing Epidemic*

Tabel 5.24 Hasil Skenario 3 (*Epidemic*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Trglek_0	89	89	89	267	114	129	900	0.8837	6.8947	179.4804
Trglek_1	71	71	71	213						
Trglek_2	84	84	84	252						
Trglek_3	122	122	122	366						
Trglek_4	122	122	122	366						
Trglek_5	122	122	122	366						

Pada tabel 5.24 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 3 dengan menggunakan protokol *routing Epidemic* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.23 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 3 pada protokol *routing Epidemic* sebanyak 129 pesan, *message relay* yang dihasilkan sebanyak 900 pesan, *Delivery Probability* yang dihasilkan sebesar 0.8837 pesan, *Overhead Ratio* yang dihasilkan sebesar 6.8947 pesan, dan *Average Latency* yang dihasilkan sebesar 179.4804 ms.

b. Analisis Skenario 3 Pada Protokol *Routing Spray And Wait*

Tabel 5.25 Hasil Skenario 3 (*Spray And Wait*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Trglek_0	82	82	82	246	114	129	605	0.8837	4.3099	179.5848
Trglek_1	62	62	63	187						
Trglek_2	77	77	77	231						
Trglek_3	94	93	94	281						
Trglek_4	84	84	88	256						
Trglek_5	100	99	99	298						

Pada tabel 5.25 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 3 dengan menggunakan protokol *routing Spray And Wait* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.24 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 3 pada protokol *routing Spray And Wait* sebanyak 129 pesan, *message relay* yang dihasilkan sebanyak 605 pesan, *Delivery Probability* yang dihasilkan sebesar 0.8837 pesan, *Overhead Ratio* yang dihasilkan sebesar 4.3099 pesan, dan *Average Latency* yang dihasilkan sebesar 179.5848 ms.

3. Mobilitas (*Map Route Movement*)

a. Analisis Skenario 3 Pada Protokol *Routing Epidemic*

Tabel 5.26 Hasil Skenario 3 (*Epidemic*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	Message Create	Message Relay	Delivery Probability	Overhead Ratio	Average Latency
	300 MB	600 MB	1,2 GB							
Trglek_0	90	90	90	270	122	129	930	0.9457	6.6257	332.2948
Trglek_1	75	75	75	225						

Trglek_2	87	87	87	261						
Trglek_3	122	122	113	357						
Trglek_4	121	121	121	363						
Trglek_5	124	124	124	372						

Pada tabel 5.26 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 3 dengan menggunakan protokol *routing Epidemic* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.25 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 3 pada protokol *routing Epidemic* sebanyak 129 pesan, *message relay* yang dihasilkan sebanyak 930 pesan, *Delivery Probability* yang dihasilkan sebesar 0.9457 pesan, *Overhead Ratio* yang dihasilkan sebesar 6.6257 pesan, dan *Average Latency* yang dihasilkan sebesar 332.2948 ms.

b. Analisis Skenario 3 Pada Protokol *Routing Spray And Wait*

Tabel 5.27 Hasil Skenario 3 (*Spray And Wait*)

Node Bergerak	Ukuran Pesan			Jumlah Pesan	Pesan Terkirim	<i>Message Create</i>	<i>Message Relay</i>	<i>Delivery Probability</i>	<i>Overhead Ratio</i>	<i>Average Latency</i>
	300 MB	600 MB	1,2 GB							
Trglek_0	66	67	66	199	122	129	601	0.9457	3.9290	348.8287
Trglek_1	64	64	63	191						
Trglek_2	78	76	73	227						
Trglek_3	54	53	44	151						
Trglek_4	73	73	72	218						
Trglek_5	101	100	95	296						

Pada tabel 5.27 telah dijelaskan, bahwa hasil pengujian yang dilakukan dari skenario 3 dengan menggunakan protokol *routing Spray And Wait* dengan ukuran pesan sebesar 300 MB, 600 MB, 1,2 GB. Dan setiap *node* yang bergerak memiliki jumlah total pesan keseluruhan dari ketiga ukuran pesan yang telah dijelaskan pada tabel 5.26 dan jumlah pesan sesuai dengan ukuran pesan yang telah ditentukan. Dalam hal ini terlihat bahwa, *Message Create* yang dihasilkan skenario 3 pada protokol *routing Spray And Wait* sebanyak 129 pesan, *message relay* yang dihasilkan sebanyak 601 pesan, *Delivery Probability* yang dihasilkan sebesar 0.9457 pesan, *Overhead Ratio* yang dihasilkan sebesar 3.9290 pesan, dan *Average Latency* yang dihasilkan sebesar 348.8287 ms.

BAB 6 PENUTUP

6.1 Kesimpulan

Dari hasil analisis pengujian yang telah dilakukan didapatkan kesimpulan. Hasil kesimpulan yang telah dilakukan adalah sebagai berikut :

1. Dari hasil pengujian dan analisis yang telah dilakukan didapatkan beberapa poin kesimpulan sebagai berikut :
 - a. Pada pengujian *Delivery Probability* pada skenario 1, 2, dan 3 routing *Epidemic* dan *Spray and Wait* didapatkan perbandingan data yang sama. Hal ini menunjukkan bahwa *Delivery Probability* hanya ditentukan oleh mobilitasnya saja. Pada skenario pertama yang dilakukan selama 45 menit didapatkan *Delivery Probability* tertinggi pada Mobilitas *Open Shortest Path* sebesar 0.9853 pesan, peringkat kedua *Map Based Movement* sebesar 0.9412 pesan, dan terakhir *Map Route Movement* sebesar 0.9265 pesan. Pada skenario kedua yang dilakukan selama 65 menit didapatkan *Delivery Probability* tertinggi pada Mobilitas *Map Based Movement* sebesar 1 pesan, peringkat kedua *Open Shortest Path* dan *Map Route Movement* dengan nilai yang sama sebesar 0.9796 pesan. Pada skenario ketiga yang dilakukan selama 85 menit dengan tujuan yang lebih banyak didapatkan *Delivery Probability* tertinggi pada Mobilitas *Map Route Movement* sebesar 0.9457 pesan, peringkat kedua *Map Based Movement* sebesar 0.9225 pesan, dan terakhir *Open Shortest Path* sebesar 0.8837 pesan. Dari hasil tersebut dapat disimpulkan bahwa pada skenario pertama *Delivery Probabilitas* tertinggi ada pada Mobilitas *Open Shortest Path*, hal ini dikarenakan Mobilitas untuk mendapatkan tujuan dengan waktu yang tidak cukup singkat dibandingkan skenario lainnya dibutuhkan jalur terpendek, berbeda dengan skenario kedua dimana *Delivery Probabilitas* tertinggi ada pada Mobilitas *Map Based Movement* yang membutuhkan waktu lebih untuk menemukan tujuannya, waktu yang ditempuh berbanding lurus dengan jumlah probabilitas pengacakan jalur yang akan ditempuh, semakin lama waktu yang diberikan maka semakin banyak juga kemungkinan yang didapatkan untuk menyampaikan pesan yang disampaikan agar paket data tidak di drop ditengah jalan. Lalu pada skenario ketiga didapatkan hasil dengan nilai *Delivery Probality* tertinggi pada Mobilitas *Map Route Movement* yang dikarenakan tujuan yang ditempuh lebih banyak sehingga menjadikan rute yang tetap dan waktu yang lebih lama menjadikan Mobilitas yang memiliki rute yang tetap lebih baik, selain itu

performa dari *Mobilitas Map Based Movement* menurun dikarenakan harus mengacak jalur terlebih dahulu, sedangkan performa dari *Open Shortest Path* juga menurun dikarenakan harus menentukan jalur tercepat yang harus ditempuh terlebih dahulu sebelum waktu berakhir.

- b. Pada pengujian *Overhead Ratio* didapatkan perbandingan antara *Mobilitas* yang tidak jauh berbeda hasilnya pada setiap protokol *routing*. Sehingga analisis difokuskan pada hasil perbedaan protokol *routing Epidemic* dan *Spray And Wait*. Hasil yang didapatkan pada Gambar 5.8, 5.9, dan 5.10 menunjukkan bahwa protokol *routing Epidemic* pada semua *mobilitas* memiliki *Overhead Ratio* yang tinggi sebesar 6.0938, 6.0625, dan 6.9328 pesan dibandingkan dengan *Overhead Ratio* pada protokol *routing Spray And Wait* pada semua *Mobilitas* sebesar 3.8408, 3.8129, dan 4.3099 pesan. Dari hasil tersebut dapat disimpulkan bahwa protokol *routing Spray And Wait* dapat mengatasi pesan yang berlebihan pada jaringan dibandingkan dengan protokol *routing Epidemic* yang tidak memiliki metode untuk membatasi pesan yang berlebihan. Sehingga menyebabkan tingkat *Overhead Ratio* yang sangat tinggi.
- c. Pada pengujian *Average Latency* didapatkan data yang berbeda baik dari protokol *routing* dan *Mobilitasnya*. Berdasarkan pada gambar 5.12, 5.13, dan 5.14 bisa dilihat bahwa *Mobilitas* dengan tingkat *Average Latency* terendah didapatkan oleh *Mobilitas Open Shortest Path* sebesar 72.9538 ms. Kemudian untuk perbandingan protokol *routing*, *Average Latency* terendah adalah protokol *routing Epidemic* untuk semua tipe *Mobilitasnya* jika dibandingkan dengan protokol *routing Spray And Wait*. Dari hasil tersebut dapat disimpulkan bahwa hal tersebut terjadi karena tipe *Mobilitas Open Shortest Path* memiliki tingkat *Average Latency* terendah karena, dapat menentukan jarak terpendek dibandingkan *Mobilitas* yang lainnya. Kemudian, untuk protokol *routing* dimana protokol *routing Epidemic* memiliki *Average Latency* yang lebih rendah dibandingkan dengan protokol *routing Spray And Wait*, hal itu dikarenakan protokol *routing Epidemic* tidak memiliki control untuk membatasi pesan sehingga menyebabkan data yang akan dikirimkan tidak terbatas, berbeda dengan protokol *routing Spray And Wait* yang harus melakukan pengontrolan data terlebih dahulu sehingga membutuhkan waktu yang lebih banyak dibandingkan dengan protokol *routing Epidemic*.

2. Berdasarkan terhadap pengujian dan analisis yang telah dilakukan oleh peneliti dapat disimpulkan bahwa protokol *routing* dikatakan lebih baik tergantung pada kasus dan kebutuhan tertentu. Protokol *routing Epidemic* tepat untuk kasus dimana dibutuhkan pengiriman data yang cepat tanpa mempertimbangkan kredibilitas atau kualitas data yang akan dikirim dan lingkungan jaringan yang memiliki *bandwith* dan sumber daya yang besar. Sedangkan protokol *routing Spray And Wait* tepat digunakan untuk kasus yang membutuhkan kredibilitas atau kualitas data yang tinggi dan lingkungan jaringan yang memiliki *bandwith* dan sumber daya yang terbatas.

6.2 Saran

Penulisan pada tugas akhir ini tentunya mempunyai banyak kekurangan dan kesempurnaan. Tentunya saran sangat diperlukan untuk penelitian lebih baik kedepannya, saran – saran sebagai berikut :

- a. Pengujian kinerja protokol *routing* DTN dapat dilakukan pada daerah lainnya dengan menggunakan parameter pengujian lainnya agar dapat melihat *performansi* pada protokol *routing* secara lebih detail lagi dan optimal.
- b. Sebaiknya perlu dilakukan pengujian terhadap penambahan *node* agar lebih banyak lagi supaya dapat melihat kinerja protokol *routing* menjadi lebih baik dan optimal dalam pengiriman sebuah pesan.
- c. Perlu melakukan pengujian simulasi terhadap mekanisme manajemen *buffer* pada protokol *routing Epidemic dan Spray And Wait*.

DAFTAR PUSTAKA

- Warthman F., 2003. *Delay-Tolerant Networks (DTNs) A Tutorial v1.1*. Diunduh dari: <http://www.dtnrg.org/docs/tutorials/warthman-1.1.pdf> [Diakses 9 September 2016].
- Siswanti, S. D., 2013. *Pengembangan Sistem Aplikasi Pengiriman Data Daerah Terpencil Berbasis Delay Tolerant Network*, Jurnal Generic, Vol. 8, No. 2.
- Fall, Kevin, 2003. *A Delay-Tolerant Network Architecture for Challenged Internets*. Intel Research, Berkeley.
- D'Souza, R. J., Jose J., 2010. *Routing Approaches in Delay Tolerant Networks: A Survey*, International Journal of Computer Applications (0975 - 8887) Volume 1 – No. 17.
- Trust, T. I., 2007. *Delay-Tolerant Networking Architecture*. [pdf] Tersedia di <http://tools.ietf.org/pdf/rfc4838.pdf> [Diakses 9 September 2016].
- Muslih, Elkaf, R., Nurhendrato, S., 2014. *Desain Pola Struktur Mapping Schema Untuk Sinkronisasi Dan Integrasi Multidatabase Terdistribusi Dalam Mengelola Data Epidemiologi*. Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro.
- Desai, R., Modi A., P., N., 2015. *A Group Based Control Flooding Routing Algorithm for Delay Tolerant Network*, International Journal of Innovative and Emerging Research in Engineering Volume 2, Issue 11.
- Widhiarti, Yuniar Ratna, 2013. *Digital Repository Materi Pembelajaran Bagi Daerah Tertinggal Berbasis Delay Tolerant Network (DTN)*. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya Malang.
- Vahdat, A., Becker, D., 2000. *Epidemic routing for partially connected ad hoc networks*.
- Spyropoulos, T., Psounis, K., Raghavendra, C., S., 2007. *Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks*. IEEE ACM SIGCOMM Workshops on Delay-Tolerant Networking 2007. White Plains, New York, March.
- Keranen A., Ott, J., Karkkainen, T., 2009. *The One simulator for DTN Protocol Evaluation: SIMUTools*, Rome. Italy
- Bozed, K., A., Zerek A., R., Shlibek, A., S., 2016. *Performance Evaluation of Disruption Tolerant Network Routing Protocols in Urban Area Scenario*, 3rd International Conference on Automation, Control, Engineering and Computer Science.

Mehto A., Chawla, M., Ph.D, 2014. *Modified Different Neighbor History Spray And Wait using PROPHET in Delay Tolerant Network*, International Journal of Computer Applications (0975 – 8887) Volume 86 – No 18.

Barzan, A., Bonn´e, B., Quax, p., Lamotte, W., 2013. *A Comparative Simulation of Opportunistic Routing Protocols Using Realistic Mobility Data Obtained from Mass Events*.

Liu, C., 2009. *DESIGN AND IMPLEMENTATION OF EFFICIENT ROUTING PROTOCOLS IN DELAY TOLERANT NETWORKS*.

DAFTAR LAMPIRAN

Lampiran 1 Konfigurasi *Default Settings* Pada *The One Simulator*

```
#
# Default settings for the simulation
#
## Scenario settings
#-----identifikasi skenario-----
# Skenario waktu Pengiriman
# 43200s == 12h
# 86400s == 24h
# 45 menit = 2700s (SKENARIO UJI 1)
# 65 menit = 3900s (SKENARIO UJI 2)
# 85 menit = 5100s (SKENARIO UJI 3)

Scenario.name = SkenarioFix
Scenario.simulateConnections = true
Scenario.updateInterval = 0.1

#Scenario.endTime = 2700
Scenario.endTime = 3900
#Scenario.endTime = 5100

#-----Kecepatan dan type media transmisi node-----
WiFi80211.type = SimpleBroadcastInterface
```

```
WiFi80211.transmitSpeed = 1375k

#-----jarak transmisi tiap node-----

WiFi80211.transmitRange = 30

highspeedInterface.transmitSpeed = 10M
highspeedInterface.transmitRange = 1000

#---Banyaknya Jumlah Host Group yang diinisialisasi pada simulator -----

Scenario.nrofHostGroups = 4

#-----Mode pergerakan node pada protokol Routing-----

Group.movementModel = MapBasedMovement

#Group.router = EpidemicRouter
Group.router = SprayAndWaitRouter

Group.bufferSize = 150M
Group.waitTime = 10, 30
Group.speed = 1, 1
Group.nrofInterfaces = 1
Group.interface1 = WiFi80211

#---jumlah host node yang bergerak-----

Group.nrofHosts = 6

Group.msgTtl = 300
```

```
#----Skenario Rute-----  
  
#skenario rute 1  
#Group1.groupID= Blt_  
#Group1.routeFile = data/Maping/blitar.wkt  
#Group1.okMaps = 3  
#Group1.speed = 1, 1  
#Group1.bufferSize = 50M  
  
#skenario rute 2  
Group1.groupID= Tlg__  
Group1.routeFile = data/Maping/tulungAgung.wkt  
Group1.okMaps = 2  
#Group1.speed = 1, 1  
  
#skenario rute 3  
#Group1.groupID = Trglek_  
#Group1.routeFile = data/Maping/trenggalek.wkt  
#Group1.okMaps = 1  
#Group1.speed = 1, 1  
  
Group2.groupID = KotaTrenggalek_  
Group2.movementModel = StationaryMovement  
Group2.nrofHosts = 1  
Group2.nodeLocation = 203, 387
```

```
#Group2.bufferSize = 50M
```

```
Group3.groupID = KotaTulungAgung_
```

```
Group3.movementModel = StationaryMovement
```

```
Group3.nrofHosts = 1
```

```
Group3.nodeLocation = 350, 395
```

```
#Group3.bufferSize = 50M
```

```
Group4.groupID = KotaBlitar_
```

```
Group4.movementModel = StationaryMovement
```

```
Group4.nrofHosts = 1
```

```
Group4.nodeLocation = 549, 416
```

```
#Group4.bufferSize = 50M
```

```
MapBasedMovement.nrofMapFiles = 4
```

```
MapBasedMovement.mapFile1 = data/Maping/trenggalek.wkt
```

```
MapBasedMovement.mapFile2 = data/Maping/tulungAgung.wkt
```

```
MapBasedMovement.mapFile3 = data/Maping/blitar.wkt
```

```
MapBasedMovement.mapFile4 = data/Maping/connectionLines.wkt
```

```
# How many event generators
```

```
Events.nrof = 1
```

```
#Class of the first event generator
```

Events1.class = *MessageEventGenerator*

Events1.interval = 35, 45

#--Konversi Ukuran Pesan-----

#300MB = 300000 kB (uji 1)

#600MB = 600000 kB (uji 2)

#1,2GB = 1200000 KB (uji 3)

#Events1.size = 300000

Events1.size = 600000

#Events1.size = 1200000

Events1.hosts = 0, 3

Events1.prefix = A

MovementModel.rngSeed = 1

MovementModel.worldSize = 4500, 3400

MovementModel.warmup = 100

Report.nrofReports = 1

Report.warmup = 0

Report.granularity = 1

```
Report.reportDir = data/Mapping/report/pengujian/Epidemic
MessageLocationReport.granularity = 1
MessageLocationReport.messages = 1
Report.report1 = MessageStatsReport
Report.report2 = DeliveredMessagesReport
Report.report3 = CreatedMessagesReport
Report.report4 = DistanceDelayReport
Report.report5 = MessageGraphvizReport
Report.report6 = MessageDeliveryReport
Report.report7 = MessageDelayReport
Report.report8 = MessageReport
Report.report9 = EventLogReport

#-----
SprayAndWaitRouter.nrofCopies = 6
SprayAndWaitRouter.binaryMode = true

## Optimization settings -- these affect the speed of the simulation
## see World class for details.
Optimization.cellSizeMult = 5
Optimization.randomizeUpdateOrder = true

# GUI underlay image settings
```

```

GUI.UnderlayImage.fileName = data/maps.PNG
# Image offset in pixels (x, y)
GUI.UnderlayImage.offset = 602, 480
# Scaling factor for the image
GUI.UnderlayImage.scale = 0.00
# Image rotation (radians)
GUI.UnderlayImage.rotate = -0.015
# how many events to show in the log panel (default = 30)
GUI.EventLogPanel.nrofEvents = 100

```

Lampiran 2 *Source Code Protokol Routing Epidemic (EpidemicRouter.java)*

```

/*
 * Copyright 2010 Aalto University, ComNet
 * Released under GPLv3. See LICENSE.txt for details.
 */
package routing;

import core.Settings;

/**
 * Epidemic message router with drop-oldest buffer and only single transferring
 * connections at a time.
 */

```

```

public class EpidemicRouter extends ActiveRouter {

    /**
     * Constructor. Creates a new message router based on the settings in
     * the given Settings object.
     * @param s The settings object
     */
    public EpidemicRouter(Settings s) {
        super(s);
        //TODO: read&use Epidemic router specific settings (if any)
    }

    /**
     * Copy constructor.
     * @param r The router prototype where setting values are copied from
     */
    protected EpidemicRouter(EpidemicRouter r) {
        super(r);
        //TODO: copy Epidemic settings here (if any)
    }

    @Override
    public void update() {
        super.update();
        if (isTransferring() || !canStartTransfer()) {

```

```
        return; // transferring, don't try other connections yet
    }

    // Try first the messages that can be delivered to final recipient
    if (exchangeDeliverableMessages() != null) {
        return; // started a transfer, don't try others (yet)
    }

    // then try any/all message to any/all connection
    this.tryAllMessagesToAllConnections();
}

@Override
public EpidemicRouter replicate() {
    return new EpidemicRouter(this);
}
}
```

**Lampiran 3 Source Code Pada Routing Protokol Spray And Wait
(SprayAndWaitRouter.java)**

```
/*
 * Copyright 2010 Aalto University, ComNet
 * Released under GPLv3. See LICENSE.txt for details.
 */
package routing;

import java.util.ArrayList;
import java.util.List;

import core.Connection;
import core.DTNHost;
import core.Message;
import core.Settings;

/**
 * Implementation of Spray And Wait router as depicted in
 * <I>Spray and Wait: An Efficient Routing Scheme for Intermittently
 * Connected Mobile Networks</I> by Thrasyvoulos Spyropoulos et al.
 *
 */
public class SprayAndWaitRouter extends ActiveRouter {
    /** identifier for the initial number of copies setting ({@value})*/
    public static final String NROF_COPIES = "nrofCopies";
```

```

/** identifier for the binary-mode setting ({@value})*
public static final String BINARY_MODE = "binaryMode";

/** SprayAndWait router's settings name space ({@value})*
public static final String SPRAYANDWAIT_NS = "SprayAndWaitRouter";

/** Message property key */
public static final String MSG_COUNT_PROPERTY = SPRAYANDWAIT_NS + "." +
    "copies";

protected int initialNrofCopies;
protected boolean isBinary;

public SprayAndWaitRouter(Settings s) {
    super(s);
    Settings snwSettings = new Settings(SPRAYANDWAIT_NS);

    initialNrofCopies = snwSettings.getInt(NROF_COPIES);
    isBinary = snwSettings.getBoolean( BINARY_MODE);
}

/**
 * Copy constructor.
 * @param r The router prototype where setting values are copied from
 */
protected SprayAndWaitRouter(SprayAndWaitRouter r) {
    super(r);
}

```

```

        this.initialNrofCopies = r.initialNrofCopies;

        this.isBinary = r.isBinary;
    }

    @Override
    public int receiveMessage(Message m, DTNHost from) {
        return super.receiveMessage(m, from);
    }

    @Override
    public Message messageTransferred(String id, DTNHost from) {
        Message msg = super.messageTransferred(id, from);
        Integer nrofCopies = (Integer)msg.getProperty(MSG_COUNT_PROPERTY);

        assert nrofCopies != null : "Not a SnW message: " + msg;

        if (isBinary) {
            /* in binary S'n'W the receiving node gets ceil(n/2) copies */
            nrofCopies = (int)Math.ceil(nrofCopies/2.0);
        }
        else {
            /* in standard S'n'W the receiving node gets only single copy */
            nrofCopies = 1;
        }
    }

```

```

        msg.updateProperty(MSG_COUNT_PROPERTY, nrofCopies);
        return msg;
    }

    @Override
    public boolean createNewMessage(Message msg) {
        makeRoomForNewMessage(msg.getSize());

        msg.setTtl(this.msgTtl);

        msg.addProperty(MSG_COUNT_PROPERTY, Integer(initialNrofCopies));
        addToMessages(msg, true);
        return true;
    }

    @Override
    public void update() {
        super.update();
        if (!canStartTransfer() || isTransferring()) {
            return; // nothing to transfer or is currently transferring
        }

        /* try messages that could be delivered to final recipient */
        if (exchangeDeliverableMessages() != null) {
            return;
        }
    }

```

```

        /* create a list of SAWMessages that have copies left to distribute */
        @SuppressWarnings(value = "unchecked")

        List<Message>                copiesLeft                =
sortByQueueMode(getMessagesWithCopiesLeft());

        if (copiesLeft.size() > 0) {

            /* try to send those messages */

            this.tryMessagesToConnections(copiesLeft, getConnections());

        }
    }

/**
 * Creates and returns a list of messages this router is currently
 * carrying and still has copies left to distribute (nrof copies > 1).
 * @return A list of messages that have copies left
 */
protected List<Message> getMessagesWithCopiesLeft() {

    List<Message> list = new ArrayList<Message>();

    for (Message m : getMessageCollection()) {

        Integer                nrofCopies                =
(Integer)m.getProperty(MSG_COUNT_PROPERTY);

        assert nrofCopies != null : "SnW message " + m + " didn't have " +
            "nrof copies property!";

        if (nrofCopies > 1) {

```

```

        list.add(m);
    }
}

return list;
}

/**
 * Called just before a transfer is finalized (by
 * {@link ActiveRouter#update()}).
 * Reduces the number of copies we have left for a message.
 * In binary Spray and Wait, sending host is left with floor(n/2) copies,
 * but in standard mode, nrof copies left is reduced by one.
 */
@Override
protected void transferDone(Connection con) {
    Integer nrofCopies;
    String msgId = con.getMessage().getId();
    /* get this router's copy of the message */
    Message msg = getMessage(msgId);

    if (msg == null) { // message has been dropped from the buffer after..
        return; // ..start of transfer -> no need to reduce amount of
copies
    }
}

```

```
    /* reduce the amount of copies left */
    nrofCopies = (Integer)msg.getProperty(MSG_COUNT_PROPERTY);
    if (isBinary) {
        nrofCopies /= 2;
    }
    else {
        nrofCopies--;
    }
    msg.updateProperty(MSG_COUNT_PROPERTY, nrofCopies);
}

@Override
public SprayAndWaitRouter replicate() {
    return new SprayAndWaitRouter(this);
}
}
```