



**ANALISIS PERBANDINGAN UNJUK KERJA *TRANSPORT AGENT*
(TCP) *NEW RENO* DENGAN FULL TCP PADA JARINGAN MANET
PADA SAAT PERUBAHAN KONDISI WAKTU**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Teknik Elektro (S1)
dan mencapai gelar Sarjana Teknik

Oleh :

Anif Dharmawan

NIM 111910201042

PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS JEMBER

2017

PERSEMBAHAN

Puji syukur penulis panjatkan kepada Allah SWT karena rahmat dan hidayah-Nya, penulis mampu menyelesaikan naskah skripsi yang berjudul **“Analisis Perbandingan Unjuk Kerja *Transport Agent* (TCP) *NewReno* Dengan Full TCP Pada Jaringan MANET Pada Saat Perubahan Kondisi Waktu”**. Sholawat serta salam selalu tercurahkan kepada Nabi Muhammad SAW yang telah mengajarkan kita kebenaran serta kebaikan.

Dengan ini saya persembahkan skripsi kepada:

1. Allah SWT. Yang Maha Pemberi Petunjuk dan Maha Penyayang.
2. Kedua orang tua tercinta, Alm Drs. Syamsul Dharma dan Nurhanifah, Kedua kakak, Asri Juwita dan Irma Rahmasari, serta calon pendamping hidup yang selalu mendukung serta mendoakan agar selalu menjadi orang yang sukses.
3. Guru-guru mulai dari TK Tadika Puri Jakarta, SDN Kepatihan 16 Jember, SMPN 11 Jember, SMAN 5 Jember, serta Dosen-dosen Teknik Elektro Universitas Jember, terima kasih atas ilmu dan pelajaran yang telah diberikan untuk menjadi orang yang bermanfaat bagi masyarakat, nusa, dan bangsa.
4. Almamater tercinta Teknik Elektro Universitas Jember.

Semoga Allah SWT memberikan yang terbaik kepada saudar-saudara semua. Tentunya masih banyak kekurangan dalam penulisan skripsi ini, oleh karena itu penulis mengharapkan kritik dan saran yang membangun dalam memperbaiki skripsi ini sehingga menjadi lebih baik lagi. Diharapkan skripsi ini dapat bermanfaat.

Jember, 23 Februari 2017

Penulis,

Anif Dharmawan

MOTTO

“Berbicara dengan baik dan fasih adalah seni yang hebat, tapi mengetahui saat yang tepat untuk berhenti berbicara juga tindakan yang sama-sama hebat.”

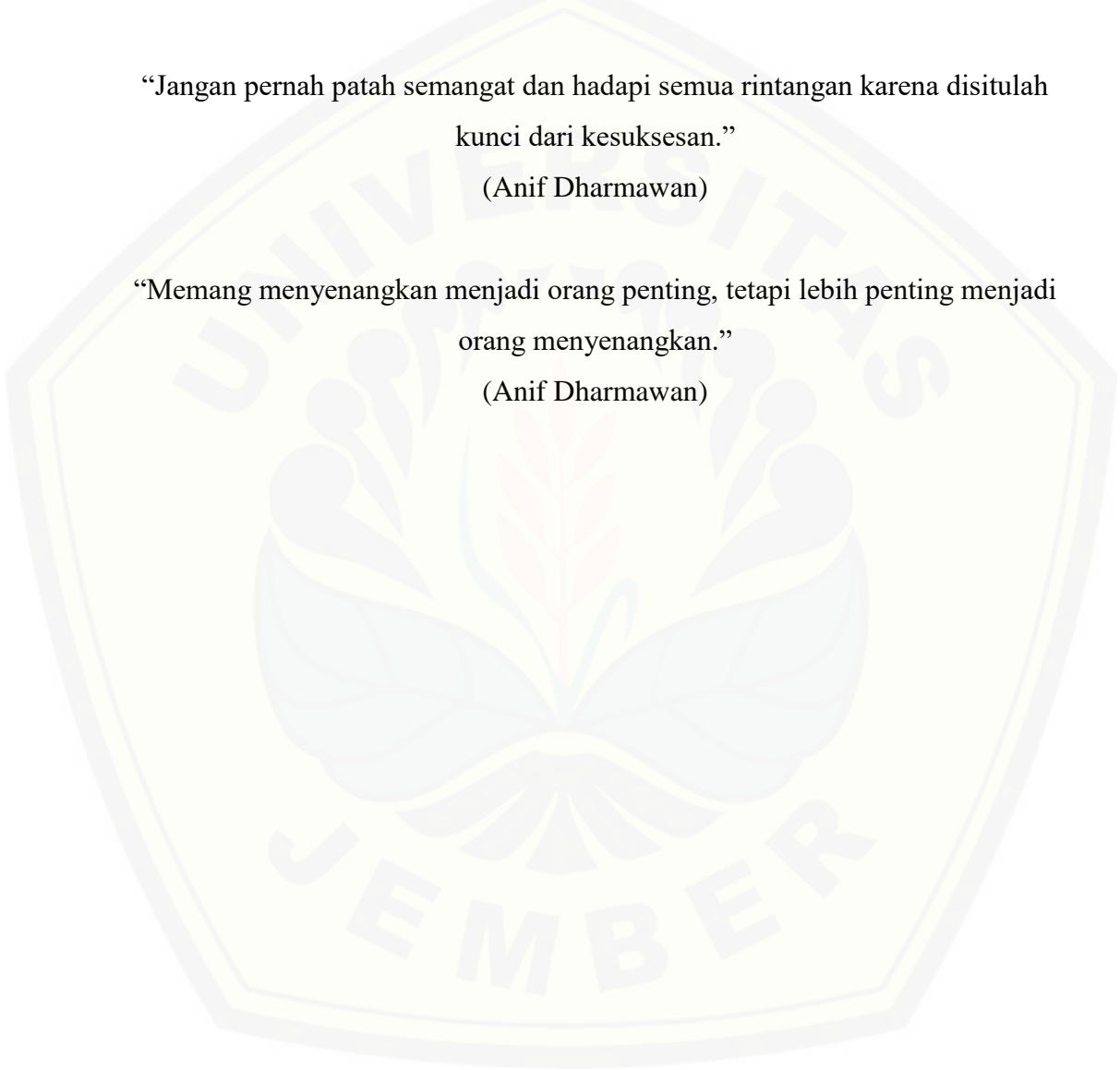
(Wolfgang Amadeus Mozart)

“Jangan pernah patah semangat dan hadapi semua rintangan karena disitulah kunci dari kesuksesan.”

(Anif Dharmawan)

“Memang menyenangkan menjadi orang penting, tetapi lebih penting menjadi orang menyenangkan.”

(Anif Dharmawan)



PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Anif Dharmawan

NIM : 111910201042

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “**Analisis Perbandingan Unjuk Kerja *Transport Agent* (TCP) *NewReno* Dengan Full TCP Pada Jaringan MANET Pada Saat Perubahan Kondisi Waktu**” adalah benar benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan itu tidak benar.

Jember, 23 Februari 2017

Penulis,

Anif Dharmawan

SKRIPSI

**ANALISIS PERBANDINGAN UNJUK KERJA *TRANSPORT AGENT*
(TCP) *NEW RENO* DENGAN FULL TCP PADA JARINGAN MANET
PADA SAAT PERUBAHAN KONDISI WAKTU**

SKRIPSI

Oleh :

Anif Dharmawan

NIM 111910201042

Pembimbing

Dosen Pembimbing Utama : Catur Suko Sarwono, S.T.

Dosen Pembimbing Anggota : Widya Cahyadi, S.T, M.T.

PENGESAHAN

Skripsi berjudul “**Analisis Perbandingan Unjuk Kerja *Transport Agent* (TCP) *NewReno* Dengan Full TCP Pada Jaringan MANET Pada Saat Perubahan Kondisi Waktu**” telah diuji dan disahkan oleh Program Studi S-1 Teknik Elektro, Fakultas Teknik, Universitas Jember dan dinyatakan lulus pada:

Hari : Kamis

Tanggal : 23Februari 2017

Tempat : Fakultas Teknik Elektro, Universitas Jember

Tim Penguji,

Pembimbing Utama,

Pembimbing Anggota,

Catur Suko Sarwono, S.T.
NIP. 19680119 199702 1 001

Widya Cahyadi, S.T., M.T.
NIP. 198511102014041001

Mengetahui,

Penguji I,

Penguji II,

Bambang Supeno, S.T., M.T
NIP 19690630 199512 1 001

Dodi Setia Budi., S.T., M.T.
NIP 19840531 208812 1 004

Mengesahkan,

Dekan Fakultas Teknik

Dr. Ir. Entin Hidayah, M.U.M.

NIP 19661215 199503 2 001

Analisis Perbandingan Unjuk Kerja *Transport Agent* (TCP) *NewReno* Dengan Full TCP Pada Jaringan MANET Pada Saat Perubahan Kondisi Waktu

Anif Dharmawan

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

ABSTRAK

Teknologi Jaringan nirkabel (*wireless*) *Mobile Ad Hoc Network* (MANET) merupakan teknologi informasi yang merupakan jenis jaringan *ad-hoc* yang terdiri dari banyak *mobile node* yang bersifat dinamis dan spontan, dapat diaplikasikan dimana saja dan kapan saja. Banyak ilmuwan yang meneliti teknologi ini namun masih banyak kondisi *node* yang digunakan diam atau tetap. Maka dari itu penelitian ini akan membandingkan performansi jaringan saat kondisi kinerja *transport agent* TCP *Newreno* dan Full TCP pada *mobile ad hoc network*. Skenario penelitiannya yaitu skenario perubahan jumlah *node*, dan perubahan waktu *node*. *Routing* yang digunakan yaitu AODV, dan DSR. Setelah dilakukan penelitian, hasilnya bahwa saat kondisi pergerakan *node* diatur secara *random* dan pada setiap skenario mengalami pergerakan yang sama, sedangkan skenario penelitian yang dilakukan hanya fokus pada perubahan waktu dan jumlah *node*. hasilnya bahwa saat kondisi perubahan jumlah *node* menghasilkan nilai *throughput* pada *routing* AODV *newreno* yang lebih besar daripada kondisi pada *routing* DSR *newreno* sebesar 920,53% dan 745,17% dengan jumlah 10 *node* dan 30 *node*; sedangkan nilai *throughput* pada *routing* AODV full TCP yang nilainya stabil dengan kondisi pada *routing* DSR full TCP dengan nilai 0,96%; nilai *delay* pada *routing* AODV *newreno* yang lebih besar daripada kondisi pada *routing* DSR *newreno* sebesar 2,455 s dan 2,3782 s dengan jumlah 10 *node* dan 30 *node*; sedangkan nilai *delay* pada *routing* DSR full TCP yang nilainya lebih besar dengan kondisi pada *routing* AODV full TCP dengan nilai 2,199019 s dan 2,691295 s dengan jumlah 20 *node* dan 30 *node*; nilai rasio paket terkirim (%) pada *routing* AODV *newreno* yang lebih besar daripada

kondisi pada *routing* DSR *newreno* sebesar 99,53% dan 99,02% dengan jumlah 10 *node* dan 30 *node*; sedangkan nilai rasio paket terkirim (%) pada *routing* AODV *full* TCP yang nilainya stabil dengan kondisi pada *routing* DSR *full* TCP dengan nilai 100%; Hal ini dikarenakan jarak pengiriman data semakin dekat dengan node tujuan sehingga menghasilkan data yang diterima semakin banyak. Pada perubahan waktu, *throughput* AODV *newreno* dengan DSR *newreno* mengalami penurunan hasil yaitu mulai dari 774,24% hingga 10,24% pada AODV *newreno* dan mulai dari 0,80% sampai dengan 0,96% mengalami kenaikan pada kondisi AODV *full* TCP. Nilai *delay* AODV *newreno* dengan DSR *newreno* mengalami kenaikan hasil yaitu mulai dari 3,0958% hingga 4,4390% pada AODV *newreno* dan mulai dari 0,097190% sampai dengan 0,679439% mengalami kenaikan pada kondisi DSR *full* TCP. Nilai rasio paket terkirim (%) AODV *newreno* dengan DSR *newreno* mengalami penurunan hasil yaitu mulai dari 99,02% hingga 0,583429% pada AODV *newreno* mulai dari 100% mengalami nilai yang stabil pada kondisi DSR *full* TCP.

Kata kunci: AODV, DSR, Full TCP, *Newreno*, *Throughput*, *Delay*, dan rasio paket terkirim

*Analysis Comparison Working on Transport Agent (TCP) Newreno with Full tcp
on the Network MANET at the changes in time*

Anif Dharmawan

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

ABSTRACT

Wireless network technology (wireless ad hoc) mobile network (manet) is information technology that is a kind of network ad-hoc consisting of many mobile nodes which is a dynamic and spontaneous , can be applied wherever and anytime.Many scientists who examines this technology but still so many cases nodes used silence or keep .Therefore this research il compare performansi network performance when the transport agent tcp newreno and full tcp on mobile ad hoc network .Research scenario that the amount of node scenario , and change time node .Routing used the aodv , and dsr .After investigation , the result that when the movement nodes dealt with in random and on each scenario had the same move the,While scenario research conducted only focused on change time and the number of node . The result is that at this change the number of nodes produce values on routing throughput aodv newreno larger than conditions on routing dsr newreno of 920,53 % and 745,17 % of 10 node and 30 nodes; while value throughput on routing aodv full tcp whose value stable with conditions on routing dsr full tcp worth 0,96 %; the delay in routing aodv newreno larger than conditions on routing dsr newreno of 2,455 s and 2,3782 s of 10 node and 30 nodes; while the delay in routing dsr full tcp whose value larger by conditions on routing aodv full tcp worth 2,199019 s and 2,691295 s of 20 node and 30 nodes; the value of the package sent (%) on routing aodv newreno larger thanConditions on routing dsr newreno of 99,53 % and 99,02 % of 10 node and 30 nodes; while the value of the package sent (%) on routing aodv full tcp whose value stable with conditions on routing dsr full tcp with the 100 %; this is because the data transmission closer to nodes objectives that produce numbers received more.To

changes in time, throughput aodv newreno with dsr newreno dropped results from of 774,24 % to 10,24 % on aodv newreno and the start of 0,80 % to 0,96 % increased on condition aodv full tcp. The delay aodv newreno with dsr newreno increased results from of 3,0958 % to 4,4390 % on aodv newreno and the start of 0,097190 % to 0,679439 % increased on condition dsr full tcp. The ratio of the package sent (%) aodv newreno with dsr newreno dropped results from of 99,02 % result of 0,583429 % on aodv newreno from 100 % Experienced the stable condition dsr full tcp.

Keywords: aodv, dsr, full tcp, newreno, throughput, delay, and the ratio package sent

RINGKASAN

Analisis Perbandingan Unjuk Kerja *Transport Agent* (TCP) *NewReno* Dengan Full TCP Pada Jaringan MANET Pada Saat Perubahan Kondisi Waktu: Anif Dharmawan, 111910201042; 2017: 71 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Perkembangan pada era modern saat ini banyak ilmuwan yang mengembangkan teknologi informasi salah satunya adalah teknologi jaringan nirkabel (*wireless*) dengan berbagai skenario untuk mendapatkan performansi yang lebih baik dan tuntutan masyarakat akan akses informasi dan data yang cepat. Jaringan nirkabel (*wireless*) *mobile ad-hoc network* (MANET) adalah teknologi yang terdiri dari banyak *mobile node* yang bersifat dinamis dan spontan, dan dapat diaplikasikan dimana saja dan kapan saja.

Standart teknologi MANET ini yaitu TIPHON yang memiliki karakteristik laju data rendah, konsumsi daya rendah, dan biaya yang rendah. Berbagai ilmuwan meneliti teknologi ini dengan berbagai skenario tetapi masih banyak skenario dengan kondisi *node* yang diam. Maka dari itu penelitian akan menggunakan skenario *node* yang bergerak dan akan dibandingkan dengan kondisi *node* yang diam untuk mendapatkan performansi yang lebih baik.

Routing protokol yang digunakan yaitu *routing* AODV, dan DSR yang akan dibandingkan performansinya dengan parameter jaringan yaitu *throughput*, *delay*, dan rasio paket terkirim (%). Skenario penelitian yang disusun yaitu skenario perubahan jumlah *node* (10 *node*, 20 *node*, dan 30 *node*), dan perubahan waktu *node* (90, 120, 150 sekon) sebagai tolak ukur performansi jaringan MANET. Setelah dilakukan penelitian dengan skenario yang telah disusun maka hasilnya menunjukkan bahwa *routing* yang memilikinya bahwa saat kondisi perubahan jumlah *node* menghasilkan nilai *throughput* pada *routing* AODV *newreno* yang lebih besar daripada kondisi pada *routing* DSR *newreno* sebesar 920,53% dan 745,17% dengan jumlah 10 *node* dan 30 *node*; sedangkan nilai *throughput* pada *routing* AODV full TCP yang nilainya stabil dengan

kondisi pada *routing* DSR full TCP dengan nilai 0,96%; nilai *delay* pada *routing* AODV *newreno* yang lebih besar daripada kondisi pada *routing* DSR *newreno* sebesar 2,455 s dan 2,3782 s dengan jumlah 10 *node* dan 30 *node*; sedangkan nilai *delay* pada *routing* DSR full TCP yang nilainya lebih besar dengan kondisi pada *routing* AODV full TCP dengan nilai 2,199019 s dan 2,691295 s dengan jumlah 20 *node* dan 30 *node*; nilai rasio paket terkirim (%) pada *routing* AODV *newreno* yang lebih besar daripada kondisi pada *routing* DSR *newreno* sebesar 99,53% dan 99,02% dengan jumlah 10 *node* dan 30 *node*; sedangkan nilai rasio paket terkirim (%) pada *routing* AODV full TCP yang nilainya stabil dengan kondisi pada *routing* DSR full TCP dengan nilai 100%; Hal ini dikarenakan jarak pengiriman data semakin dekat dengan *node* tujuan sehingga menghasilkan data yang diterima semakin banyak.

Pada skenario 2 perubahan waktu, *throughput* AODV *newreno* dengan DSR *newreno* mengalami penurunan hasil yaitu mulai dari 774,24% hingga 10,24% pada AODV *newreno* dan mulai dari 0,80% sampai dengan 0,96% mengalami kenaikan pada kondisi AODV full TCP. Nilai *delay* AODV *newreno* dengan DSR *newreno* mengalami kenaikan hasil yaitu mulai dari 3,0958% hingga 4,4390% pada AODV *newreno* dan mulai dari 0,097190% sampai dengan 0,679439% mengalami kenaikan pada kondisi DSR full TCP. Nilai rasio paket terkirim (%) AODV *newreno* dengan DSR *newreno* mengalami penurunan hasil yaitu mulai dari 99,02% hingga 0,583429% pada AODV *newreno* mulai dari 100% mengalami nilai yang stabil pada kondisi DSR full TCP.

PRAKATA

Puji syukur penulis panjatkan kepada Allah SWT karena rahmat dan hidayah-Nya, penulis mampu menyelesaikan naskah skripsi yang berjudul **“Analisis Perbandingan Unjuk Kerja *Transport Agent* (TCP) *NewReno* Dengan Full TCP Pada Jaringan MANET Pada Saat Perubahan Kondisi Waktu)**”. Sholawat serta salam selalu tercurahkan kepada Nabi Muhammad SAW yang telah mengajarkan kita kebenaran serta kebaikan.

Penulisan skripsi ini tak lepas dari bimbingan, dukungan, serta bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

1. Ibu Dr. Ir. Entin Hidayah, M.U.M., selaku dekan Fakultas Teknik Universitas Jember.
2. Bapak Catur Suko Sarwono, S.T. selaku pembimbing I yang selalu memberikan kritik dan saran, arahan, motivasi, bimbingan yang membangun.
3. Bapak Widya Cahyadi, S.T., M.T. selaku pembimbing II yang selalu memberikan arahan, masukan dan meluangkan waktunya bagi penulis untuk berkonsultasi sehingga naskah skripsi ini dapat terselesaikan.
4. Bapak Bambang Supeno, S.T., M.T., selaku dosen penguji I atas saran dan perbaikannya dalam penulisan naskah skripsi ini.
5. Bapak Dodi Setia Budi, S.T., M.T. selaku dosen penguji II yang banyak memberikan arahan dan motivasi.
6. Bapak Dr. Triwahju Hardianto S.T., M.T. selaku dosen pembimbing akademik yang telah membimbing dan memotivasi dalam perkuliahan selama penulis menjadi mahasiswa.
7. Seluruh dosen dan staf karyawan Teknik, khususnya program studi S-1 Teknik Elektro, Universitas Jember serta semua pihak yang senantiasa memberikan ilmunya dan membantu penulis selama proses penyusunan naskah skripsi ini.

8. Kedua orang tua tercinta, Alm. Drs.Syamsul Dharma dan Nurhanifah yang telah memberi semangat dan memberikan do'a tiada henti, dukungan, motivasi selama proses penulisan skripsi.
9. Kedua kakak, Asri Juwita, S.Km dan Irma Rahmasari, S.Sos yang selalu memberi semangat sehingga terselesaikan nya naskah skripsi ini.
10. Calon pendamping hidup yang tidak pernah lelah memberi semangat, memberi motivasi, dan menemani dalam menyelesaikan skripsi ini.
11. Teman seperjuangan Teknik Elektro 2011, Universitas Jember (GATEL) yang selalu memberi masukan dan bersama dalam suka maupun duka.
12. Teman ngopi, nongkrong dan touring (KOPI ASAP) (KURI-KURI) (DULUR JEMBER) yang selalu memberikan do'a, dukungan, bantuan, hiburan dan kenangan yang tidak terlupakan.
13. Serta berbagai pihak yang telah membantgu proses terselesaikannya skripsi ini yang tidak dapat disebutkan satu per satu.

Semoga Allah SWT memberikan yang terbaik kepada saudar-saudara semua. Tentunya masih banyak kekurangan dalam penulisan skripsi ini, oleh karena itu penulis mengharapkan kritik dan saran yang membangun dalam memperbaiki skripsi ini sehingga menjadi lebih baik lagi. Diharapkan skripsi ini dapat bermanfaat.

Jember, 23 Februari 2017
Penulis,

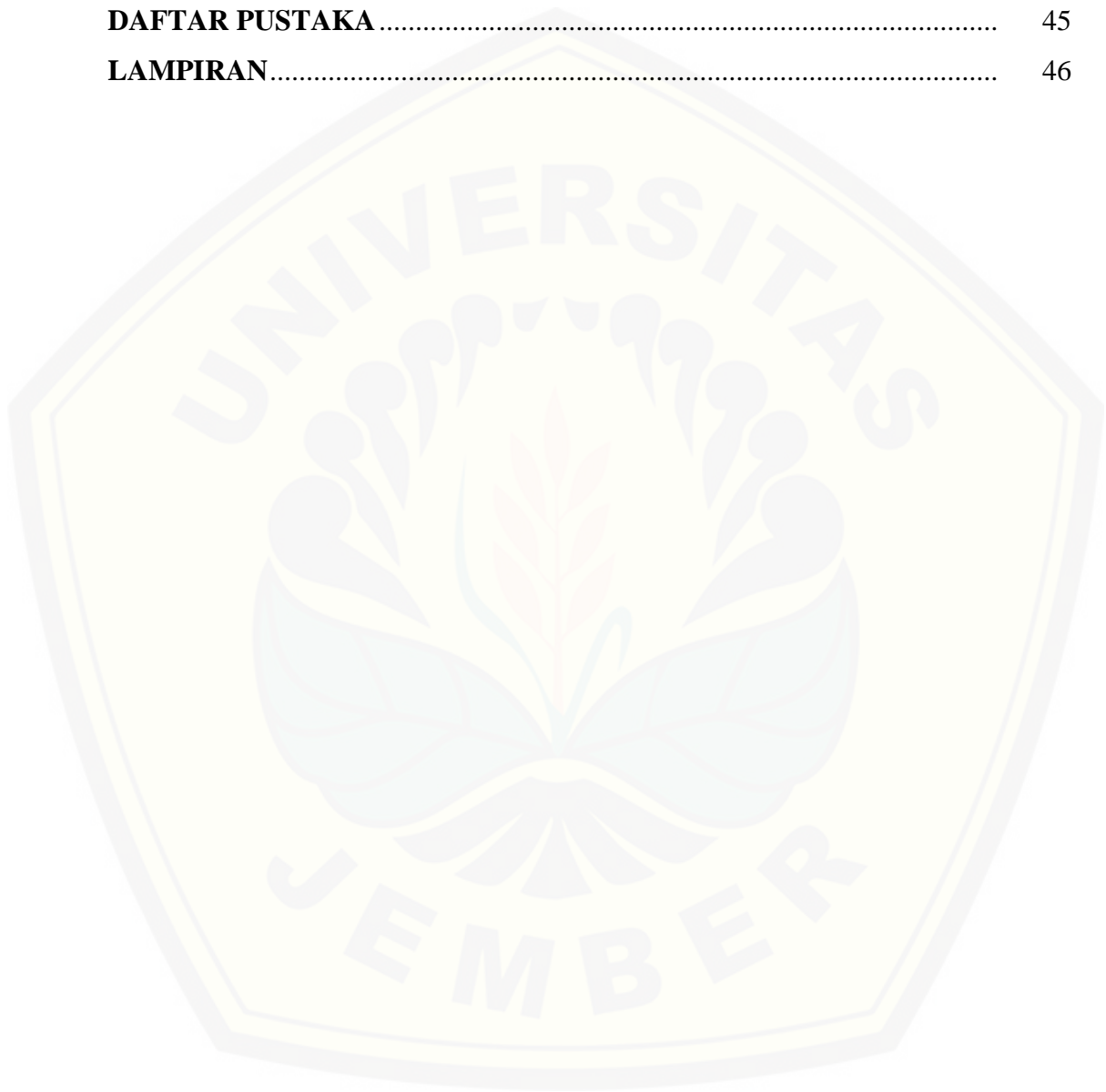
Anif Dharmawan

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN	iv
HALAMAN PEMBIMBINGAN	v
HALAMAN PENGESAHAN	vi
ABSTRAK	vii
ABSTRACT	ix
RINGKASAN	xi
PRAKATA	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xviii
DAFTAR TABEL	xx
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	2
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	5
1.5 Manfaat	6
BAB 2. TINJAUAN PUSTAKA	7
1.6 Mobile Ad Hoc Network (MANET)	7
2.1.1 Topologi jaringan Ad-hoc pada MANET	7
2.1.2 Karakteristik MANET	7
2.1.3 Protokol MANET	8
2.1.4 Protokol <i>Routing</i>	10
2.1.4.1 AODV (<i>Ad-Hoc On demand Distance Vector</i>).....	11
2.1.4.2 DSR (<i>Dynamic Source Routing</i>).....	12
2.2 Parameter Kualitas <i>Mobile Ad hoc Network</i> (MANET)	14
2.2.1 <i>Routing Overhead</i> (RO)	14

2.2.2 <i>Delay</i>	14
2.2.3 <i>Throughput</i>	15
2.3 Network Simulator 2 (NS-2)	16
2.3.1 Konsep Dasar NS-2	17
2.3.2 Output Simulasi NS-2.....	18
BAB 3. METODOLOGI PENELITIAN	19
3.1 Tempat dan Waktu Penelitian	20
3.2 Alat dan Bahan	20
3.3 Parameter Simulasi	21
3.4 Diagram Blok Sistem Penelitian	22
3.5 Pemodelan Jaringan MANET di NS-2	23
3.6 Skenario Penelitian	24
3.6.1 Skenario 1 Perubahan Jumlah <i>Node</i>	24
3.6.2 Skenario 2 Perubahan Waktu <i>Node</i>	24
BAB 4. HASIL DAN PEMBAHASAN	25
4.1 Pengujian Skenario Penelitian	26
4.2 Analisis Perbandingan Kinerja <i>Transport Agent TCP Newreno</i> dengan Full TCP Pada Skenario 1 Untuk Perubahan Jumlah <i>Node</i>	26
4.2.1 <i>Throughput AODV</i>	27
4.2.2 <i>Delay AODV</i>	28
4.2.3 Rasio Paket Terkirim AODV	29
4.2.4 <i>Throughput DSR</i>	30
4.2.5 <i>Delay DSR</i>	31
4.2.6 Rasio Paket Terkirim DSR.....	32
4.3 Analisis Perbandingan Kinerja <i>Transport Agent TCP Newreno</i> dengan Full TCP Pada Skenario 2 Untuk Perubahan Waktu <i>Node</i>.....	33
4.3.1 <i>Throughput AODV</i>	34
4.3.2 <i>Delay AODV</i>	35
4.3.3 Rasio Paket Terkirim AODV	37
4.3.4 <i>Throughput DSR</i>	39
4.3.5 <i>Delay DSR</i>	40

4.3.6 Rasio Paket Terkirim DSR.....	41
BAB 5. PENUTUP.....	43
5.1 Kesimpulan.....	43
5.2 Saran.....	44
DAFTAR PUSTAKA.....	45
LAMPIRAN.....	46



DAFTAR GAMBAR

Gambar 2.1 Topologi MANET	8
Gambar 2.2 Mekanisme penemuan rute.....	11
Gambar 2.3 Mekanisme data <i>route update</i> dan <i>route error</i>	11
Gambar 2.4 Route <i>Discovery</i> pada <i>Routing DSR</i>	12
Gambar 2.5 Pesan <i>Error</i>	13
Gambar 2.6 Komponen Pembangunan NS	17
Gambar 2.7 Arsitektur Dasar dari NS2	17
Gambar 2.8 Tampilan <i>File Trace</i> pada NS2	18
Gambar 2.9 Tampilan <i>File namtrace</i> pada NS2	18
Gambar 2.10 Tampilan Xgraph pada NS2	19
Gambar 3.1 Blok Diagram Sistem	20
Gambar 3.2 <i>Flowchart AWK delay</i> dan RO.....	22
Gambar 3.3 Rancangan Jaringan MANET di NS2	23
Gambar 4.1 Grafik <i>Throghput</i> skenario 1 saat perubahan jumlah node AODV	27
Gambar 4.2 Grafik <i>delay</i> skenario 1 saat perubahan jumlah node AODV...	28
Gambar 4.3 Grafik Rasio pengiriman data (%) skenario 1 saat perubahan jumlah node AODV	29
Gambar 4.4 Grafik <i>Throghputs</i> skenario 1 saat perubahan jumlah node DSR.....	31
Gambar 4.5 Grafik <i>delay</i> skenario 1 saat perubahan jumlah node DSR.....	32
Gambar 4.6 Grafik Rasio pengiriman data (%) skenario 1 saat perubahan jumlah node DSR	33
Gambar 4.7 Grafik <i>Throghputs</i> skenario 2 saat perubahan waktu node AODV	35
Gambar 4.8 Grafik <i>delay</i> skenario 2 saat perubahan waktu node AODV	36
Gambar 4.9 Grafik Paket rasio pengiriman data (%) skenario 2 saat perubahan waktu node AODV	37
Gambar 4.10 Grafik <i>Throghputs</i> skenario 2 saat perubahan waktu	

<i>node DSR</i>	39
Gambar 4.11 Grafik <i>delay</i> skenario 2 saat perubahan waktu <i>node DSR</i>	40
Gambar 4.12 GrafikRasio pengiriman data (%) skenario 2 saat perubahan waktu <i>node DSR</i>	41



DAFTAR TABEL

Tabel 2.1 Standar performansi jaringan TIPHON <i>delay</i>	15
Tabel 2.3 Standar performansi jaringan TIPHON <i>Throughput</i>	15
Tabel 3.1 Parameter Simulasi Jaringan	20
Tabel 3.2 Skenario 1 parameter berdasarkan perubahan jumlah <i>node</i>	24
Tabel 3.3 Skenario 2 parameter berdasarkan perubahan waktu simulasi	24
Tabel 4.1 Data hasil dari nilai parameter jaringan pada saat AODV	26
Tabel 4.2 Data hasil dari nilai parameter jaringan pada saat DSR.....	30
Tabel 4.3 Data hasil dari nilai parameter jaringan pada saat AODV	34
Tabel 4.4 Data hasil dari nilai parameter jaringan pada saat DSR.....	38

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Jaringan nirkabel(*wireless*) berkembang sangat pesat saat ini. Perkembangan ini merupakan tuntutan kebutuhan masyarakat akan akses informasi dan data yang cepat, bisa kapan saja dan dimana saja. Jaringan nirkabel dengan infrastuktur merupakan perluasan dari jaringan LAN. Sedangkan jaringan nirkabel tanpa infrastuktur bisa disebut jaringan *ad-hoc*. Jaringan *ad-hoc* adalah jaringan nirkabel *multihope* yang terdiri dari banyak *node* yang bersifat spontan dan dinamis. (Yonas sidartha, 2013).

Mobile Ad Hoc Network (MANET) merupakan jenis jaringan *ad-hoc* yang terdiri dari banyak *mobile node* yang bersifat dinamis dan spontan, dapat diaplikasikan dimana saja dan kapan saja. Pada jaringan MANET, *mobile node* bergerak kemanapun dengan kecepatan tertentu dan mengirimkan paket data secara *peer to peer*. *Mobile node* pada jaringan MANET tidak hanya berfungsi sebagai pengirim dan penerima data. *Mobile node* juga berperan sebagai *router* untuk menunjang perangkat lainnya dalam menentukan arah tujuan data atau informasi yang akan dikirim. Oleh karena itu, dibutuhkan *routing protocol* untuk menentukan dan mengatur rute pengiriman data tersebut agar data sampai tujuan dengan kecepatan yang optimal. *Routing protocol* yang cocok dengan jaringan MANET yaitu DSDV (*Destination Sequenced Distance Vector*), AODV (*Ad Hoc On demand Distance Vector*) dan DSR (*Dynamic Source Routing*) (Agustinus dimas Fitriyanto, 2015).

Aliran trafik pada MANET sangat dibutuhkan untuk proses pengiriman data antara *node* pengirim dan *node* penerima, dalam hal ini penentuan aliran trafik sangat dibutuhkan dan di dalam penentuan aliran trafik tersebut belum terdapat standar untuk pemilihan tipe aplikasi apa yang harus digunakan dan di implementasikan pada MANET. Masalah muncul ketika proses penentuan *transport agent* yang efisien dalam pengiriman paket data dari sumber ke tujuan. Dan permasalahan selanjutnya muncul ketika suatu kualitas pada kinerja *transport*

agent yang dipilih untuk proses pengiriman yang digunakan oleh *node – node* tidak sesuai yang diharapkan untuk implementasi yang diinginkan.

Beberapa penelitian sebelumnya, dilakukan penelitian terhadap analisis perbandingan unjuk kerja algoritma *congestion control* pada TCP Tahoe, Reno dan (*selective Acknowledgment*) SACK (Yuliana et al, 2013). Dalam penelitian ini menerangkan bahwa tidak ada algoritma TCP *congestion control* dengan unjuk kerja terbaik di semua parameter uji yang telah dilakukan. Oleh karena itu untuk karakteristik *network* yang berbeda-beda, maka nilai unjuk kerja algoritma – algoritma TCP memiliki *behaviour* yang berbeda-beda pula.

Oleh karena itu peneliti melakukan peninjauan kembali dari penelitian sebelumnya dan melakukan penelitian terhadap perbandingan kinerja *transport agent* TCP *Newreno* dan *FullTCP* pada MANET dengan menggunakan *network simulator 2* (NS-2). Performa dari dua *transport agent* tersebut akan diukur berdasarkan *delay* dan *Routing Overhead* (RO) dengan skenario perubahan waktu.

TCP *NewReno* itu sendiri dapat didefinisikan oleh RFC 6582 (yang *obseletes* definisi sebelumnya dalam RFC 3782 dan RFC 2582), untuk meningkatkan transmisi selama fase cepat untuk pemulihan dari TCP Reno. Selama pemulihan cepat, untuk setiap duplikat ACK yang dikembalikan ke TCP *NewReno*, paket terkirim baru diakhiri jendela kemacetan dikirim, untuk menjaga jendela mengirimkan penuh. Untuk setiap ACK yang membuat kemajuan parsial dalam ruang urut, pengirim mengansumsikan bahwa poin ACK ke lubang baru, dan paker berikutnya melampaui jumlah urutan *acked* dikirim.

Karena timer *timeout* ulang setiap kali ada kemajuan dalam *buffer* pancar, ini memungkinkan *NewReno* untuk mengisi lubang besar, atau beberapa lubang, di ruang urut - seperti TCP SACK. Karena *NewReno* dapat mengirim paket baru pada akhir jendela kemacetan selama pemulihan yang cepat, *throughput* yang tinggi dipertahankan selama proses lubang-mengisi, bahkan ketika ada beberapa lubang, beberapa paket masing-masing. Ketika TCP memasuki pemulihan yang cepat itu mencatat *outstanding* nomor urut paket diakui tertinggi. Ketika nomor urut ini diakui, TCP kembali ke status menghindari kemacetan.

Masalah terjadi dengan TCP *NewReno* ketika tidak ada kerugian paket tetapi sebaliknya, paket mengatur kembali oleh lebih dari 3 nomor urut paket. Ketika ini terjadi, TCP *NewReno* keliru memasuki pemulihan cepat, tetapi ketika paket mengatur kembali disampaikan, ACK urut-nomor kemajuan terjadi dan dari sana sampai akhir pemulihan yang cepat, setiap sedikit kemajuan urut-nomor menghasilkan duplikat dan transmisi perlu yang segera acked. TCP *NewReno* melakukan serta SACK di tingkat kesalahan paket yang rendah, dan *substansial* melebihi Reno pada tingkat kesalahan yang tinggi.

Full TCP itu sendiri dapat didefinisikan sebagai suatu protokol yang berada di lapisan *transport* (baik itu dalam tujuh lapis model referensi OSI atau model DARPA) yang berorientasi untuk menyambungan (*connection-oriented*) dan dapat diandalkan (*reliable*).

1.2 Rumusan Masalah

Dalam penelitian yang diusulkan dalam proposal ini adalah beberapa hal yang menjadi rumusan masalah antara lain :

1. Bagaimana Peformansi kinerja TCP *Newreno* dan *FullTCP* pada *mobile ad-hoc network* dengan *routing protocol* AODV dan DSR ?
2. Bagaimana hasil perbandingan dari hubungan perubahan waktu dan jumlah *node* terhadap kinerja TCP *Newreno* dan *FullTCP* pada *routing protocol* AODV dan DSR?

1.3 Batasan Masalah

Dari latar belakang penelitian diatas, agar pembahasan tidak terlalu luas maka diperlukan suatu pembatasan masalah sebagai berikut :

1. Simulasi menggunakan program NS-2 versi 2.31 (NS-2.31), dan untuk *routing protocol* AODV dan DSR, dan *traffic behaviour* yang digunakan FTP sedangkan hasil *output* dalam bentuk Grafik.
2. Banyaknya perubahan jumlah *node* dan perubahan waktu *node* yang digunakan 10, 20 dan 30 *node* serta 20 m/s.
3. Parameter kinerja yang digunakan adalah *delay*, *Routing Overhead* (RO).

4. Pergerakan *node* diatur secara *random* dan pada setiap skenario mengalami pergerakan yang sama, sedangkan skenario penelitian yang dilakukan hanya fokus pada perubahan waktu dan jumlah *node*.

1.4 Tujuan

Dalam penelitian yang diusulkan dalam proposal ini memiliki beberapa tujuan yang diantaranya :

1. Mengetahui Performansi kinerja TCP *Newreno* dan *FullTCP* pada *mobile ad hoc network* dengan *routing protocol* AODV dan DSR.
2. Menganalisis hasil perbandingan dari perubahan waktu dan jumlah *node* terhadap kinerja TCP *Newreno* dan *FullTCP* pada *routing protocol* AODV dan DSR.

1.5 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut :

1. Mengetahui Performansi kinerja *transport agent* TCP *Newreno* dan *FullTCP* pada *mobile ad hoc network* dengan *routing protocol* AODV dan DSR.
2. Mengetahui *transpot agent* yang lebih baik untuk *Mobile Ad hoc Network* terhadap perubahan waktu dan jumlah *node* dengan *routing protocol* AODV dan DSR.

1.6 SISTEMATIKA PENULISAN

Secara garis besar penyusunan proposal skripsi ini adalah sebagai berikut:

BAB 1. PENDAHULUAN

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan pembahasan, manfaat pembahasan dan sistematika pembahasan.

BAB 2. TINJAUAN PUSTAKA

Berisi tentang tinjauan pustaka yang menguraikan pendapat-pendapat atau hasil-hasil penelitian terdahulu yang berkaitan dengan penelitian yang dilakukan, landasan teori merupakan penjabaran dari tinjauan pustaka.

BAB 3. METODOLOGI PENELITIAN

Menjelaskan tentang metode kajian yang digunakan untuk menyelesaikan skripsi.

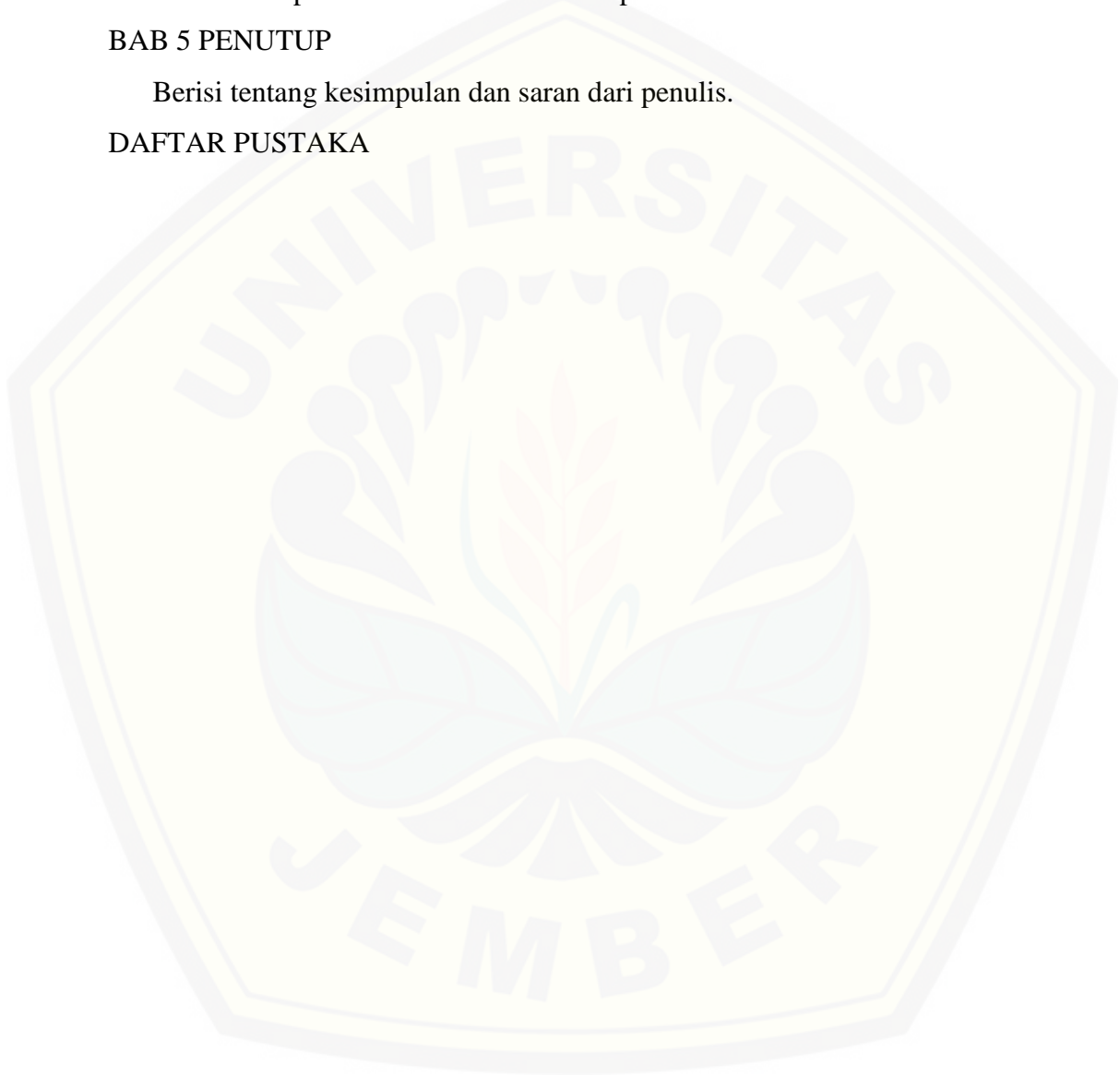
BAB 4 HASIL DAN PEMBAHASAN

Berisi hasil penelitian dan analisa hasil penelitian.

BAB 5 PENUTUP

Berisi tentang kesimpulan dan saran dari penulis.

DAFTAR PUSTAKA



BAB 2. TINJAUAN PUSTAKA

2.1 *Mobile AdHoc Network* (MANET)

Mobile Ad Hoc Network (MANET) menurut (Sun. 2010) adalah kumpulan dari beberapa *wireless node* yang dapat di *setup* secara dinamis dimana saja dan kapan saja tanpa menggunakan infrastruktur jaringan yang ada. MANET juga merupakan jaringan sementara yang dibentuk oleh beberapa *mobile node* tanpa adanya pusat administrasi dan infrastruktur kabel. Pada MANET, *mobile host* yang terhubung dengan *wireless* dapat bergerak bebas dan juga berperan sebagai *router*.

Menurut (Sun. 2010) terdapat beberapa perbedaan antara jaringan *ad hoc* dengan jaringan yang memiliki infrastruktur, antara lain :

- a. *PeertoPeer*, yaitu komunikasi antara dua *node* dalam satu *hop*.
- b. *RemotetoRemote*, yaitu komunikasi antar dua *node* diluar satu *hop*, namun masih tetap mengelola kestabilan rute di antara keduanya.
- c. *Dynamic Traffic*, terjadi ketika *node* bergerak, maka rute harus dikonstruksi ulang. Ini merupakan hasil dari tingkat konektifitas yang rendah.

2.1.1 Topologi Jaringan Ad-hoc Pada MANET

Pada tiap jaringan yang dibangun pasti menggunakan topologi jaringan tertentu, hal ini penting karena dengan adanya topologi jaringan maka alur-alur komunikasi antara *host* dan *router* pada suatu jaringan dapat diketahui. Jika pada jaringan kabel seperti *local area network* (LAN) memiliki beberapa model topologi yang dapat dipakai, berbeda dengan jaringan *ad hoc* pada MANET yang cenderung tidak memakai topologi yang spesifik. Hal ini dikarenakan pergerakan *node* pada jaringan MANET yang bebas sehingga menyebabkan topologi berubah-ubah sesuai letak tiap *node*.



Gambar 2.1 Topologi Manet
(Sumber : Imawan, 2009)

Tiap *node* pada MANET seperti yang ditunjukkan gambar 2.1 memiliki jangkauan sinyal tertentu sehingga komunikasi antara dua *node* yang saling berjauhan dapat dilakukan dengan melewati jalur *node* penghubung yang berada di tengah posisi kedua *node* tersebut. Kondisi komunikasi dua *node* inilah yang dikatakan sebagai komunikasi antara *router* dan *host* seperti pada jaringan kabel pada umumnya, karena tiap *node* pada MANET dapat bersifat sebagai *router* maupun sebagai *host*. Faktor-faktor inilah yang kemudian menjadikan topologi yang dimiliki oleh MANET bersifat dinamis atau berubah-ubah.

2.1.2 Karakteristik MANET

Menurut (Prasant *at al.* 2005:4), berdasarkan dokumen *Request for Comments* menjelaskan bahwa terdapat beberapa karakteristik dari *Mobile Ad Hoc Network* (MANET). Disana dijelaskan bahwa MANET terdiri dari *mobile platform* (seperti *router* dan perangkat *wireless*) dalam hal ini disebut dengan “*node*” yang bebas berpindah-pindah ke mana saja. *Node* tersebut bisa saja berada di pesawat, kapal, mobil dan dimana saja.

Setiap *node* dilengkapi dengan *transmitter* dan *receiver wireless* menggunakan antenna atau sejenisnya yang bersifat *omnidirectional (broadcast)*, *highly directional (point to point)*, memungkinkan untuk diarahkan, atau kombinasi dari beberapa hal tersebut. *Omnidirectional* maksudnya adalah

gelombang radio dipancarkan ke segala arah oleh perangkat *transmitter wireless*. Sedangkan *highly directional* adalah gelombang dipancarkan ke satu arah tertentu. Selain karakteristik di atas, *Ad hoc Network* juga memiliki beberapa karakteristik yang lebih menonjol, antara lain:

- a. Topologi yang dinamis : *Node* pada MANET memiliki sifat yang dinamis, yaitu dapat berpindah-pindah kemana saja. Maka topologi jaringan yang bentuknya adalah loncatan antara *hop* ke *hop* dapat berubah secara tidak terpola dan terjadi secara terus menerus tanpa ada ketetapan waktu untuk berpindah. Bisa saja didalam topologi tersebut terdiri dari *node* yang terhubung ke banyak *hop* lainnya, sehingga sangat berpengaruh secara signifikan terhadap susunan topologi jaringan.
- b. Otonomi : Setiap *node* pada MANET berperan sebagai *enduser* sekaligus sebagai *router* yang menghitung sendiri *route path* yang selanjutnya akan dipilih.
- c. Keterbatasan *bandwidth* : *Link* pada jaringan *wireless* cenderung memiliki kapasitas yang rendah jika dibandingkan dengan jaringan berkabel. Jadi, kapasitas yang keluar untuk komunikasi *wireless* juga cenderung lebih kecil dari kapasitas maksimum transmisi. Efek yang terjadi pada jaringan yang berkapasitas rendah adalah *congestion* (kemacetan).
- d. Keterbatasan energi : Semua *node* pada MANET bersifat *mobile*, sehingga sangat dipastikan *node* tersebut menggunakan tenaga baterai untuk beroperasi. Sehingga perlu perancangan untuk optimalisasi energi.
- e. Keterbatasan keamanan : Jaringan *wireless* cenderung lebih rentan terhadap keamanan daripada jaringan berkabel. Kegiatan pencurian (*eavesdropping*, *spoofing* dan *denial of service*) harus lebih diperhatikan.

2.1.3 Protokol MANET

Menurut (Cardei, 2004), terdapat berbagai jenis protokol *routing* untuk MANET yang secara keseluruhan dapat dibagi menjadi beberapa kelompok, antara lain adalah *Proactive Routing*, *Reactive Routing*, yang masing-masing kelompok dibagi menjadi beberapa bagian lagi.

2.1.4 Protokol Routing

Menurut (Imawan. 2009) *protocol routing* adalah standarisasi yang melakukan kontrol terhadap bagaimana sebuah *node* dapat meneruskan paket diantara perangkat komputasi dalam jaringan *mobile ad-hoc network* (MANET).

Dalam jaringan *ad hoc*, setiap *node* tidak mempunyai pengetahuan mengenai topologi jaringan sekitarnya, melainkannode-node tersebut harus dicari. Dasar pemikirannya adalah bahwa *node* baru yang masuk akan mendengarkan pesan *broadcast* dari tetangganya. Sebuah *node* akan mempelajari *node* baru didekatnya dan bagaimana cara menjangkau *node* baru tersebut. Pada suatu saat, setiap *node* akan mengetahui tentang *node-node* lain dan mengetahui bagaimana cara menjangkaunya.

Protocol routing layaknya sebuah *router* yang berkomunikasi dengan perangkat lain untuk menyebarkan informasi dan mengijinkan adanya pemilihan rute diantara dua *node* dalam jaringan. Pada jaringan *ad hoc* setiap *node* akan memiliki kemampuan layaknya *router* yang meneruskan pesan antar *node* di sekitarnya. Untuk itu dibutuhkan *protocol routing* untuk membantu tiap-tiap *node* melakukannya.

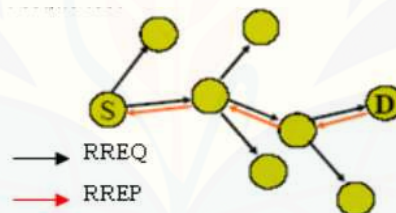
Protokol routing untuk jaringan *ad hoc* tentunya berbeda dengan protokol *routing* yang biasa diimplementasikan pada jaringan kabel. Hal ini disebabkan sifat jaringan *ad hoc* yang dinamis, sehingga memiliki topologi yang berubah-ubah, berbeda dengan jaringan kabel yang cenderung tetap. Dibawah ini beberapa protokol *routing* yang digunakan dalam penelitian ini adalah *Ad Hoc On demand Distance Vector* (AODV), *Dynamic Source Routing* (DSR), dan *Destination Sequenced Distance Vector* (DSDV).

2.1.4.1 AODV (*Ad Hoc On demand Distance Vector*)

Protokol *Ad-hoc On-demand Distance Vector* (AODV) adalah protokol yang mekanismenya merupakan perpaduan antara protokol DSR (*on demand*) dan protokol DSDV (*hop by hop*). Protokol ini melakukan mekanisme layaknya pada DSR, yaitu adanya rute pencarian (*Route Discovery*) dan rute pemeliharaan (*Route*

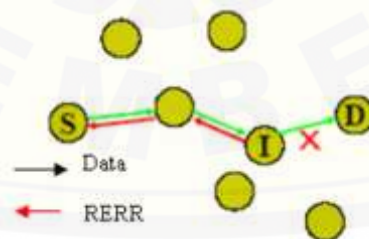
Maintenance), dan proses *routing* dilakukan secara *hop by hop* layaknya pada protokol DSDV.

AODV menggunakan tabel *routing* dengan satu *entry* untuk setiap tujuan. Tanpa menggunakan *routing* sumber, AODV mempercayakan pada tabel *routing* untuk menyebarkan *Route Reply* (RREP) kembali ke sumber dan secara sekuensial akan mengarahkan paket data menuju ketujuan. AODV juga menggunakan *sequence number* untuk menjaga setiap tujuan agar didapat informasi *routing* yang terbaru dan untuk menghindari *routing loops*. Semua paket yang diarahkan membawa *sequence number* ini. Penemuan jalur (*Path discovery*) atau *Route discovery* diinisiasi dengan menyebarkan *Route Reply* (RREP) (Prasant *at al.* 2005:73), seperti terlihat pada Gambar 2.2 Ketika RREP menjelajahi *node*, ia akan secara otomatis *setup path*. Jika sebuah *node* menerima RREP, maka *node* tersebut akan mengirimkan RREP lagi ke *node* atau *destination sequence number*. Pada proses ini, *node* pertama kali akan mengecek *destination sequence*.



Gambar 2.2. Mekanisme Penemuan Rute AODV

(Sumber: Arifin *at al.* 2010)



Gambar 2.3 Data Route Update dan Route Error AODV

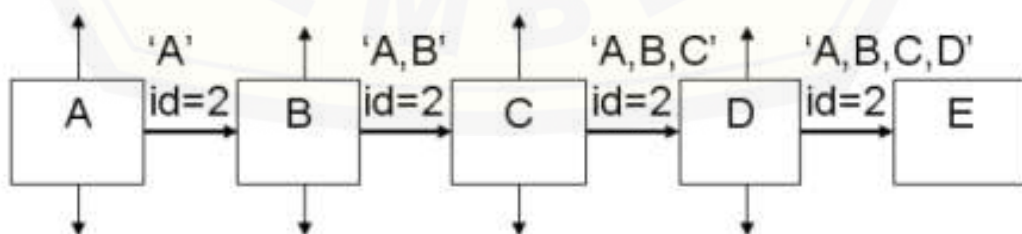
(Sumber: Arifin *at al.* 2010)

Number pada tabel *routing*, apakah lebih besar dari 1 pada *route request* (RREQ), jika benar, maka *node* akan mengirim RREP. Ketika RREP berjalan kembali ke *source* melalui *path* yang telah di *setup*, ia akan *setup* jalur kedepan dan memperbarui *time out*. Jika sebuah *link* ke *hop* berikutnya tidak dapat dideteksi dengan metode penemuan rute, maka *link* tersebut akan diasumsikan putus dan *Route Error* (RERR) akan disebarkan ke *nodeneighbour* seperti terlihat pada Gambar 2.3 Dengan demikian sebuah *node* bisa menghentikan pengiriman data melalui rute ini atau meminta rute baru dengan menyebarkan RREQ kembali. (Arifin *at al.* 2010).

2.1.4.2 DSR (Dynamic Source Routing)

Dynamic Source Routing (DSR) protokol adalah sebuah protokol *routing* yang sederhana dan efisien dirancang khusus untuk digunakan dalam jaringan *wireless multihop ad hoc mobile node*. DSR memungkinkan sebuah jaringan bisa mengatur dan mengkonfigurasi sendiri tanpa memerlukan infrastruktur jaringan yang ada atau admin. DSR telah dilakukan oleh berbagai kelompok, dan dicoba di beberapa *testbeds*. Jaringan yang menggunakan protokol DSR telah terhubung ke Internet. DSR dapat *interoperate* dengan *Mobile IP*, dan *node* menggunakan *Mobile IP* dan DSR telah bisa bermigrasi antara WLAN, layanan data seluler dan DSR jaringan *mobile ad hoc*.

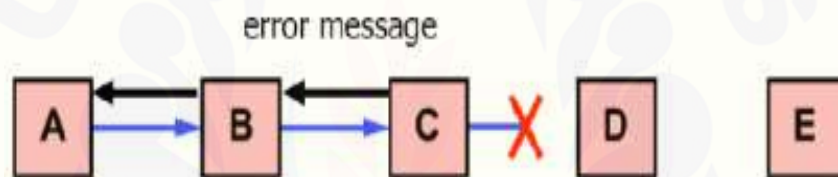
Protokol ini terdiri dari dua mekanisme utama "*Route Discovery*" dan "*Route Maintenance*", yang bekerjasama untuk memungkinkan *node* untuk menemukan dan mempertahankan rute ke tujuan dalam jaringan *ad hoc*.



Gambar 2.4 *Route Discovery* pada *Routing DSR*

Jika *node A* memiliki *Route Cache* rute ke *node E*, maka rute ini segera digunakan. Jika tidak, *protocol discovery route* dimulai:

- Node A* mengirim paket *routerrequest* dengan cara *flooding* pada jaringan.
- Jika *node B* mendapat paket *routerrequest* dari target yang sama (*node A*) atau jika alamat *node B* sudah terdaftar di *record route*, kemudian *node B* membuang *route request* tersebut.
- Jika *node B* adalah target dari *route discovery*, maka *node B* akan mengirim *routerreply* ke *node A*. *Routerreply* berisi daftar dari sumber *node* ke tujuan dengan jarak terpendek. Jika *A* menerima *route reply*, daftar tersebut akan disimpan dan akan dikirim ke *node* berikutnya untuk menuju target.
- Jika *node B* bukan tujuannya, maka *node B* akan mengirim *route request* ke *neighbore node* untuk melanjutkan *routing* nya.



Gambar 2.5 Pesan *Error*

Pada setiap *node* DSR bertanggung jawab untuk mengirimkan dan mengkonfirmasi bahwa *hop* berikutnya dalam sumber *Route* menerima paket. Juga masing-masing paket hanya diteruskan sekali oleh *node* (*hop by hop routing*). Jika sebuah paket tidak dapat diterima oleh sebuah *node*, paket ditransmisikan ulang beberapa kali sampai jumlah maksimum dan konfirmasi diterima dari *hop* berikutnya. Hanya jika hasil retransmisi mengalami kegagalan, pesan *RouteError* dikirim ke inisiator, yang dapat menghapus bahwa Sumber Rute dari *Route Cache*. Jadi inisiator dapat memeriksa *Route Cache* untuk rute lainnya untuk target. Jika tidak ada rute di *cache*, paket *RouteRequest* dikirimkan lagi (Prasant *at al.* 2005:72).

2.2 Parameter Kualitas *Mobile Ad Hoc Network* (MANET)

Menurut (Sidharta. 2013) dalam mengukur atau mengetahui kualitas pada jaringan *mobile ad-hoc network* ada beberapa parameter yang dapat mempengaruhi *performance* pada kualitas jaringan *ad-hoc*, parameter ini berfungsi untuk mengetahui kualitas pada jaringan *adhoc* tersebut, dan parameter – parameter yang mendukung adalah :

- a. *Routing Overhead* (RO)
- b. *Troughput*
- c. *Delay*

2.2.1 *Routing Overhead* (RO)

Routing Overhead adalah jumlah total *packet routing* yang dikirimkan oleh *routing protocol* dalam setiap pengiriman paket data. Jadi, nilai parameter ini dilihat dari sisi *node* sumber ataupun juga *nodeintermediate*. Parameter ini digunakan untuk menghitung efisiensi kinerja suatu *routing protocol*. Persamaan 2.1 menunjukkan perhitungan untuk mendapatkan nilai dari *Routing Overhead*.

$$RO = \frac{\sum \text{Paket Routing terkirim}}{\sum \text{Paket Data Terkirim}} \dots \dots \dots (2.1)$$

2.2.2 *Delay*

Waktu tunda adalah jumlah total waktu pengiriman paket dalam satu kali pengamatan. Dalam hal ini satu kali simulasi dibagi dengan jumlah usaha pengiriman yang berhasil dalam satu kali pengamatan tersebut. Secara umum delay rata-rata dapat dinyatakan dengan persamaan berikut :

$$T = \frac{2 \text{ waktu pengiriman dalam satu kali pengamatan}}{\sum \text{usaha pengiriman paket berhasil}} \dots \dots \dots (2.2)$$

Tabel 2.1 standar performansi jaringan TIPHON berdasarkan *delay*/latensi:

Kategori Latensi	Besar Delay
Sangat Bagus	< 150 ms
Bagus	150 s/d 300 ms
Sedang	300 s/d 450 ms
Jelek	> 450 ms

(Sumber : TIPHON)

2.2.3 Throughput

Yaitu kecepatan (*rate*) transfer data efektif, yang diukur dalam bps. *throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut dapat dinyatakan dengan persamaan berikut perhitungan *throughput* :

$$\text{Throughput} = \frac{\text{Paket data diterima}}{\text{Lama pengamatan}} \dots \dots \dots (2.3)$$

Tabel 2.2 standar performansi jaringan TIPHON berdasarkan *Troughput*:

Kategori Throughput	Throughput	Indeks
Sangat Bagus	100 %	4
Bagus	75 %	3
Sedang	50 %	2
Jelek	< 25 %	1

(Sumber : TIPHON)

2.3 Network Simulator 2 (NS-2)

Network Simulator 2 (NS-2) merupakan perangkat lunak simulator jaringan yang bersifat *even-driven* yang sering digunakan untuk menganalisa kondisi jaringan yang bersifat dinamis selama komunikasi jaringan berlangsung. Simulasi terhadap komunikasi jaringan kabel maupun jaringan *wireless* beserta protokol-protokol yang bekerja didalam jaringan dapat dilakukan pada simulator NS2. Secara umum, NS2 menyediakan fasilitas kepada pengguna untuk menentukan protokol jaringan yang akan digunakan didalam simulasi jaringan dan

mensimulasikan komunikasi didalam jaringan berdasarkan protokol yang telah ditentukan. Karena sifat yang fleksibel dan modular dari simulator NS2, NS2 mulai menjadi sangat populer digunakan sebagai simulator jaringan oleh peneliti di bidang jaringan komputer (Megawan. 2013).

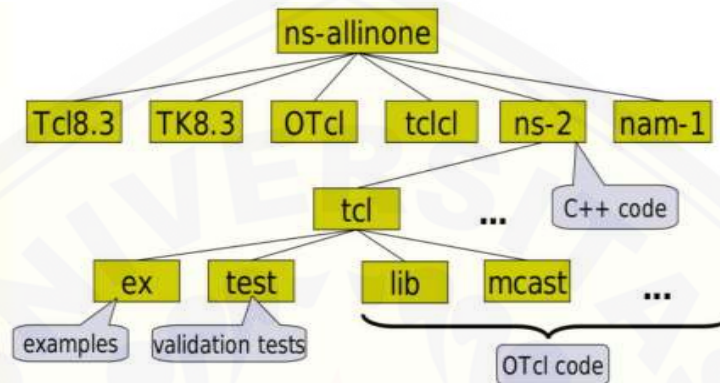
Selain itu juga dapat digunakan untuk mensimulasikan protokol jaringan (TCP/UDP/RTP), *Traffic behaviour* (FTP, Telnet, CBR), *Queue management* (RED, FIFO, CBQ), algoritma *routing unicast* (*Distance Vector*, *Link State*) dan *multicast* (PIM SM, PIM DM, DVRMP, *Shared Tree* dan *Bidirectional Shared Tree*), aplikasi multimedia yang berupa *layered video*, *Quality of Service video-audio* dan *transcoding*. NS juga mengimplementasikan beberapa MAC (IEEE 802.3, 802.11) di berbagai media, misalnya jaringan berkabel (seperti LAN, WAN, *point to point*), jaringan tanpa kabel (seperti *mobile IP*, *wireless LAN*), bahkan simulasi hubungan antar *node* jaringan yang menggunakan media satelit (Aji Sasongko *at al.* 2010).

Komponen utama lainnya pada NS di samping *object* jaringan adalah penjadwal kejadian. Sebuah kejadian dalam NS merupakan suatu paket ID yang unik untuk sebuah paket dengan waktu yang terjadwal dan penunjuk untuk suatu objek yang menangani masalah kejadian. Dalam NS, suatu penjadwal kejadian menjaga jalur dari waktu simulasi dan membawa seluruh kejadian dalam antrian kejadian terjadwal untuk saat ini dengan bantuan komponen jaringan yang sesuai, di mana biasanya hal tersebut merupakan salah satu hal yang disebutkan dalam kejadian, dan membiarkan hal tersebut menjalankan tindakan yang sesuai yang diwakilkan dengan paket yang ditunjuk oleh kejadian. Komponen jaring berkomunikasi dengan salah satu dari berbagai paket yang berlalu, bagaimanapun komponen ini tidak memakan waktu simulasi *actual*.

2.3.1 Konsep Dasar NS2

Pada *Network Simulator* terdapat tampilan atau *display* baik dengan *node* yang bergerak atau *node* yang tidak bergerak. Yang tentunya tidak sama dengan keadaan yang sebenarnya.

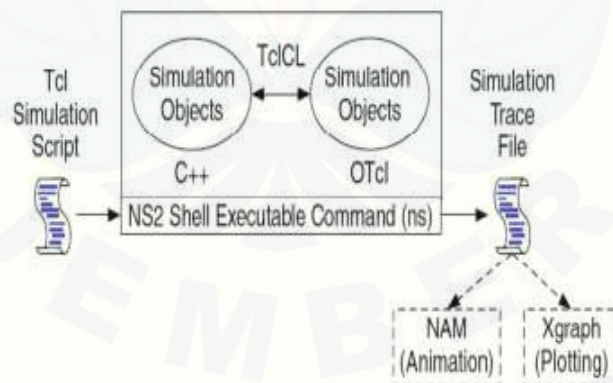
Network Simulator dibangun dengan menggunakan dua bahasa pemrograman, yaitu C++ dan Tcl/Otcl. C++ digunakan untuk *library* yang berisi *event scheduler*. Tcl/Otcl digunakan pada *script* simulasi yang ditulis oleh NS *user* dan pada *library* sebagai *simulator* objek. Otcl juga nantinya berperan sebagai interpreter. Pada gambar 2.6 adalah komponen pembangunan NS.



Gambar 2.6 Komponen Pembangunan NS

(Sumber : Aji Sasongko *at al.* 2010)

Untuk menampilkan hasil simulasi dalam bentuk grafik maupun animasi, biasanya dapat menggunakan *tool* seperti NAM (*Network Animator*) dan Xgraph.



Gambar 2.7 Arsitektur dasar dari NS2

(Sumber : Megawan. 2013)

Bahasa C++ digunakan pada *library* karena C++ mampu mendukung *runtime* simulasi yang cepat, meskipun simulasi melibatkan simulasi jumlah paket dan sumber data dalam jumlah besar.

Bahasa Tcl memberikan respon *runtime* yang lebih lambat daripada C++, namun jika terdapat kesalahan *syntax* dan perubahan *script* berlangsung dengan cepat dan interaktif. *User* dapat mengetahui letak kesalahannya yang dijelaskan pada *console*, sehingga *user* dapat memperbaiki dengan cepat. Karena alasan itulah bahasa ini dipilih untuk digunakan pada skrip simulasi.

2.3.2 Output Simulasi NS2

Pada saat suatu simulasi berakhir, NS membuat satu atau lebih *output* yang berisi detail simulasi jika dideklarasikan pada saat membangun simulasi. Ada tiga jenis *output* NS, yaitu :

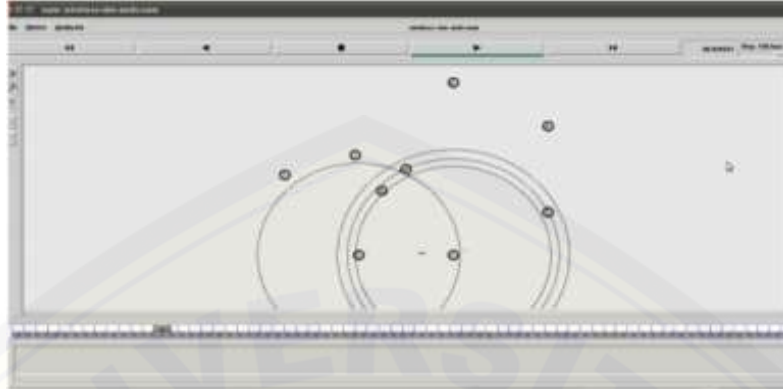
- File trace*, yang digunakan untuk analisa numerik, seperti yang ada pada gambar 2.8

```
+ 0.110419 1 2 tcp 1040 ----- 2 1.0 4.0 5 12
+ 0.110419 1 2 tcp 1040 ----- 2 1.0 4.0 6 13
- 0.110431 1 2 tcp 1040 ----- 2 1.0 4.0 5 12
- 0.110514 1 2 tcp 1040 ----- 2 1.0 4.0 6 13
r 0.11308 0 2 cbr 1000 ----- 1 0.0 3.0 2 8
+ 0.11308 2 3 cbr 1000 ----- 1 0.0 3.0 2 8
- 0.11308 2 3 cbr 1000 ----- 1 0.0 3.0 2 8
r 0.11316 0 2 cbr 1000 ----- 1 0.0 3.0 3 9
+ 0.11316 2 3 cbr 1000 ----- 1 0.0 3.0 3 9
- 0.113228 2 3 cbr 1000 ----- 1 0.0 3.0 3 9
r 0.115228 2 3 cbr 1000 ----- 1 0.0 3.0 0 6
r 0.115348 1 2 tcp 1040 ----- 2 1.0 4.0 3 10
+ 0.115348 2 4 tcp 1040 ----- 2 1.0 4.0 3 10
- 0.115348 2 4 tcp 1040 ----- 2 1.0 4.0 3 10
r 0.115376 2 3 cbr 1000 ----- 1 0.0 3.0 1 7
r 0.115431 1 2 tcp 1040 ----- 2 1.0 4.0 4 11
```

Gambar 2.8Tampilan *File Trace* pada NS2

(Sumber : Teerawat *at al*, 2009:31)

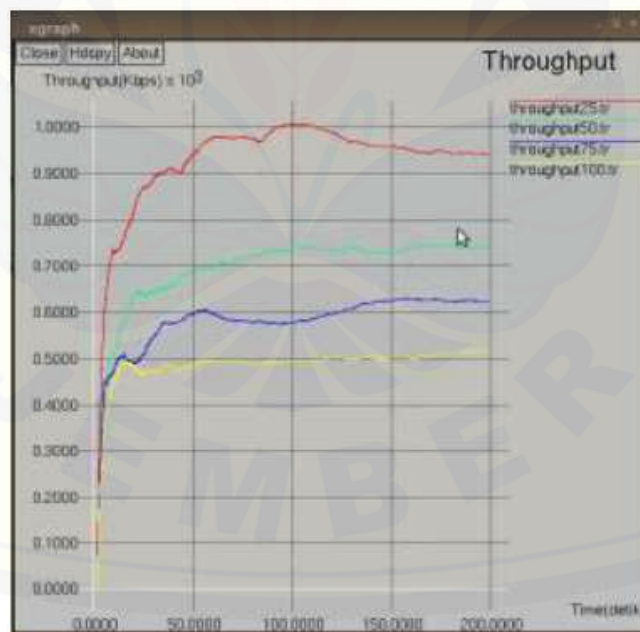
- b. *File namtrace*, yang digunakan sebagai *input* tampilan grafis simulasi, seperti pada gambar 2.9



Gambar 2.9 Tampilan *File namtrace* pada NS2

(Sumber : Kembuan at al 2012)

- c. Xgraph, yang digunakan untuk menampilkan analisa dalam bentuk grafik, seperti yang ada pada Gambar 2.10



Gambar 2.10 Tampilan Xgraph pada NS2

(Sumber : Nasrhol Haqqi. 2013)

BAB 3. METODOLOGI PENELITIAN

Dalam bab ini akan dijelaskan metode penelitian jaringan MANET menggunakan trafik TCP *Newreno* dan *Full TCP* agar penelitian dapat terarah dan mencapai tujuan yang diinginkan. Berikut beberapa metode penelitian yang dilakukan:

3.1 Tempat dan Waktu Penelitian

Penelitian ini akan dilaksanakan di Laboratorium Telekomunikasi dan Terapan Fakultas Teknik Universitas Jember Jl Slamet Riyadi No 62 Patrang, Jember. Dengan waktu mulai bulan Oktober 2016 hingga bulan Januari 2017.

3.2 Alat dan Bahan

Adapun alat dan bahan yang digunakan dalam penelitian ini yaitu :

1. Personal Computer (PC) atau Laptop.
2. Software Vmware 6.5.
3. Windows XP.
4. Software Cygwin.
5. Software Network Simulator 2 (NS2).

3.3 Parameter-Parameter Simulasi

Pada penelitian terdapat parameter-parameter jaringan simulasi MANET yang akan menjadi acuan jaringan yang digunakan untuk diteliti performansinya. Parameter-parameter jaringan tersebut seperti pada pada Tabel 3.3.

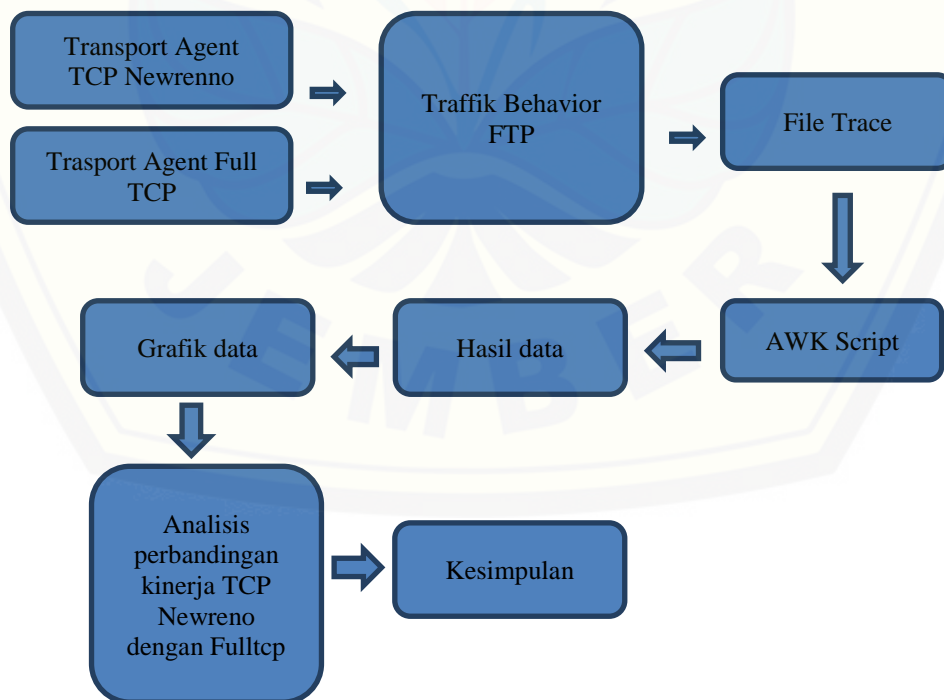
Tabel 3.1 Parameter-parameter simulasi

Parameter simulasi NS2	Value simulasi NS2
Simulator	Ns.2.31
Tipe MAC	Mac/802.11
Model Antena	Omni Antena
Model Propagasi	Two Ray Ground

Network Interface Type	Phy/WirelessPhy
Simulation Time	90, 120, dan 150 detik
Simulasi Area	X : 1000 dan Y : 1000
Routing Protocol	AODV/DSR
Trafik	FTP
Maximum Nodes	30 nodes
Ukuran Node	40
Maximum Kecepatan Node	20 m/s

3.4 Blok Diagram Sistem

Pada penelitian ini dilakukan pada teknologi MANET (*Mobile Ad hoc Network*) yang pada simulasi di *Network Simulator 2* (NS 2) memiliki suatu tahapan untuk membuat sebuah *script* yang harus dibuat agar dapat menjalankan simulasi.

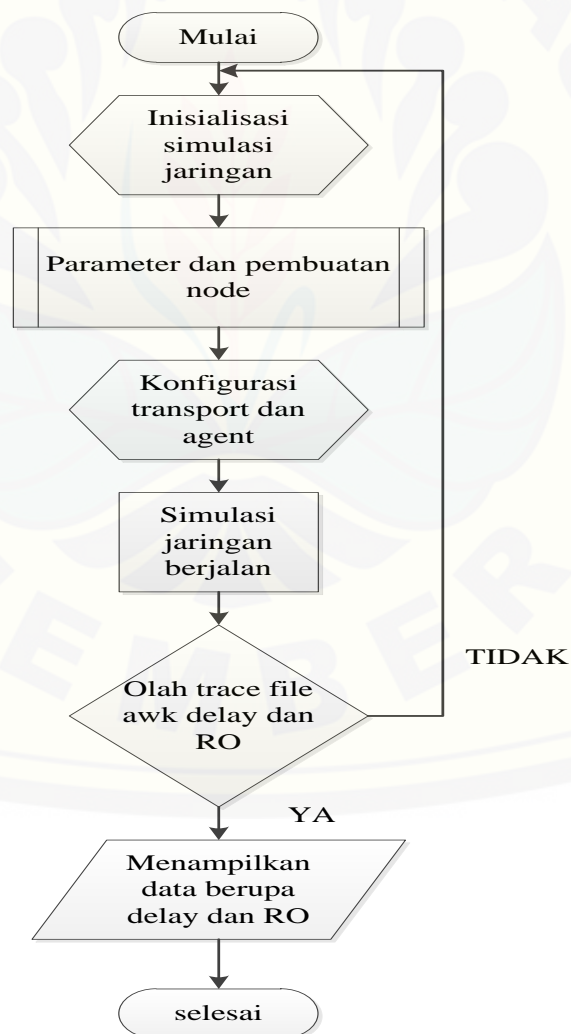


Gambar 3.2 Blok Diagram Sistem

Penentuan *transport agent* yang digunakan adalah TCP dan TCP*Sink* untuk *traffic behaviour* jenis FTP, serta UDP dan Null untuk *traffic behaviour* jenis FTP, untuk keluaran dari hasil simulasi adalah berupa *file trace* yang berupa catatan kejadian setiap waktu saat simulasi, setelah itu hasil dari *file trace* tersebut di saring dengan *file awk* yang berisi perhitungan untuk *delay* dan RO. Dari hasil *delay* dan RO tersebut menghasilkan *file trace* baru dan akan dirubah menjadi suatu grafik dari kumpulan hasil yang didapatkan.

3.5 Diagram Alir simulasi AWK *delay* dan RO

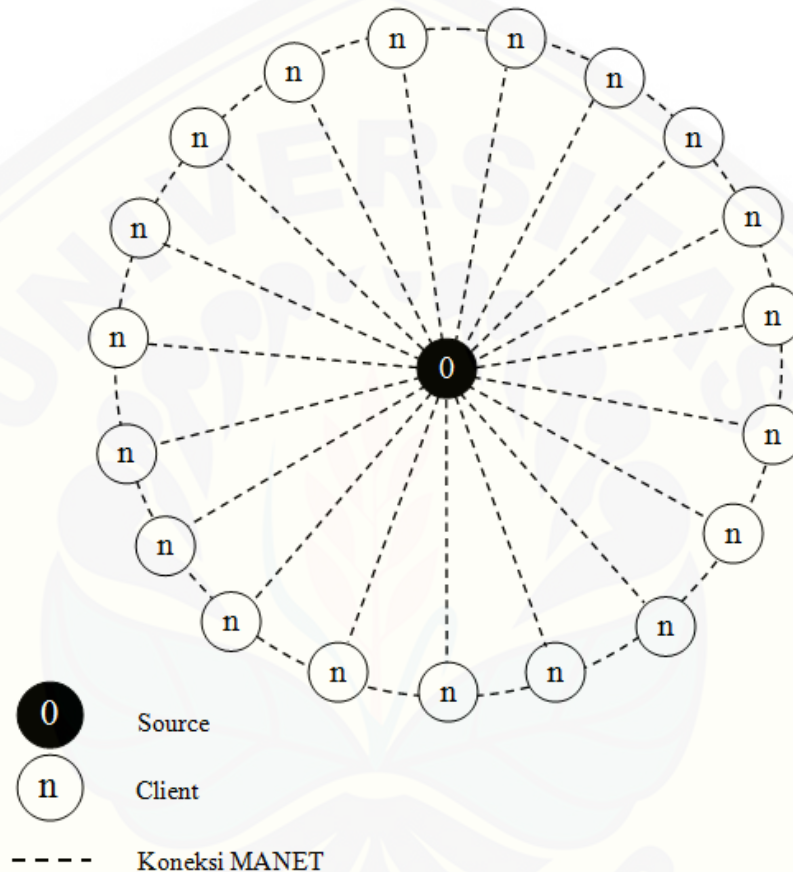
Adapun diagram alir simulasi NS-2 saat AWK *delay* yang nantinya berjalan pada penelitian sebagai berikut:



Gambar 3.3 Flowchart AWK *delay* dan RO

3.6 Pemodelan Jaringan MANET di NS2

Adapun rancangan jaringan MANET yang digunakan sebagai objek penelitian ini dalam upaya mengetahui kinerja *traffic behaviour* yang melalui beberapa algoritma *protokol*. Berikut gambaran umum dari jaringan yang dibentuk.



Gambar 3.4 Rancangan Jaringan MANET di NS2

Angka 0 dengan *background* hitam lingkaran digunakan sebagai *source node*, dan huruf n dengan lingkaran putih digunakan sebagai *client*. Didalam jaringan MANET, *coverage* tidak dibatasi oleh ketersediaan koneksi secara langsung ke *source node* maupun ke *client* lainnya. Disetiap *node user* di MANET tidak hanya beroperasi sebagai *host*, namun juga sebagai *router* untuk meneruskan (*forward*) paket untuk *node* lain yang membutuhkan paket data. Dan pada masing-masing *node* akan berpindah atau bergerak sementara ke tempat tujuan *node* tersebut.

3.7 Skenario Penelitian

Penelitian ini akan dilakukan dua skenario penelitian yang dapat dijadikan untuk melihat kualitas jaringan MANET yaitu skenario perubahan jumlah *node* dan perubahan waktu simulasi. Skenario ini akan diujikan pada jaringan MANET yang menggunakan trafik TCP *Newreno* dan trafik *Full TCP*. Penjelasan dari skenario penelitian akan dibahas dibawah ini.

3.8.1 Skenario 1 Perubahan Jumlah Node

Pada Skenario ini akan dilakukan percobaan dengan merubah jumlah *node* yang digunakan yaitu 10, 20, dan 30 *node*. jenis trafik yang digunakan berupa trafik TCP *Newreno* dan *Full TCP* dengan routing protokol AODV, dan DSR. Waktu simulasi yang digunakan yaitu 120 detik. Berikut tabel skenario perubahan jumlah *node*:

Tabel 3.2 Skenario 1 parameter berdasarkan perubahan jumlah *node*.

Skenario	Perubahan Jumlah Node	Kecepatan node
Skenario 1	10 node	20 m/s
	20 node	
	30 node	

3.8.2 Skenario 2 Perubahan waktu node

Pada skenario 2 ini, penelitian dilakukan dengan merubah waktu simulasi yang dijalankan agar dapat mengetahui tingkat performansi jaringan MANET pada saat waktu simulasi bertambah. Waktu simulasinya yaitu 90, 120, dan 150 detik dengan menggunakan trafik TCP *Newreno* dan *Full TCP*. Berikut tabel skenario 2 yaitu perubahan waktu simulasi:

Tabel 3.3 Skenario 2 parameter berdasarkan perubahan waktu simulasi

Skenario	Perubahan Waktu (second)	Jumlah node
Skenario 2	90 s	20 node
	120 s	
	150 s	

BAB 5. PENUTUP

5.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan dan analisis pada halaman sebelumnya tentang perbandingan kinerja *transport agent tcp newreno* dengan *full tcp* sesuai dengan skenario penelitian yang telah dibuat, dapat ditarik kesimpulan bahwa:

1. Pada *routing* protokol DSR memperoleh hasil *packet delivery ratio* yang lebih besar daripada *routing* protokol AODV yaitu sebesar 99,53%; 99,52%; 99,02% seperti pada tabel 4.1 saat kondisi *node* yang bergerak dengan jumlah *node* sebanyak 10, 20, dan 30*node*.
2. Pengaruh dari perubahan jumlah *node* dan perubahan waktu *node* yaitu semakin bertambah jumlah *node* maka semakin kecil nilai *throughput* dan rasio paket terkirim. Sedangkan pada perubahan waktu, semakin tinggi nilai waktu *node* maka semakin menurun nilai *throughput* dan rasio paket terkirim serta nilai *delay* yang semakin menurun.
3. Pada *routing* DSR memiliki performansi yang meningkat dari *routing* AODV dan saat kondisi *node* yang bergerak dengan kepadatan jumlah *node* sebanyak 20*node* serta dengan kecepatan yang rendah (20 m/s). Sedangkan *routing* AODV mengalami performansi yang meningkat saat kondisi *node* yang diam dengan jarak transmisi sebesar 20 meter serta dengan kepadatan *node* sebanyak 10*node*.

5.2 Saran

Beberapa saran untuk pengembangan dari penelitian ini tentang jaringan *Mobile Ad Hoc Network* (MANET) *transport agent tcp newreno* dengan *full tcp* pada penelitian selanjutnya yaitu::

1. Membandingkan *routing protokol* yang lainnya seperti *routing protocol DSDV* (*Destination Sequenced Distance Vector*) dan lain sebagainya agar dapat melihat performansi jaringan yang lebih jauh lagi.

2. Dapat melakukan penelitian jaringan *Mobile Ad Hoc Network* (MANET) dengan kondisi *node* yang memiliki posisi bergerak pada jaringan *mobile ad hoc*.
3. Dapat menggunakan parameter yang lainnya seperti *jitter*, *paket loss*, dan lain sebagainya agar dapat melihat kualitas jaringan yang lebih jauh lagi.



DAFTAR PUSTAKA

Aprillando, A., 2007, cara kerja dan kinerja *Dynamic Source Routing* (DSR) pada *Mobile Ad-Hoc Network* (MANET), Tugas Akhir, Jakarta: Fakultas Teknik Unika Atma Jaya.

Ali, S dan Ali A, “*Performance Analysis Of AODV, DSR dan OLSR in MANET*”, *Telcommunication Blekinge Institute of Teknologi, Sweden*, 2009.

C. Perkins, E. Belding-Royer, S. Das, Juli 2003, RFC 3561. *Ad hoc on demand distance vector* (AODV)

Imawan., 2009, “Analisis Kinerja Pola-Pola Trafik pada beberapa Protokol Routing dalam jaringan MANET”, Tugas Akhir. Surabaya: Institut Teknologi Sepuluh November.

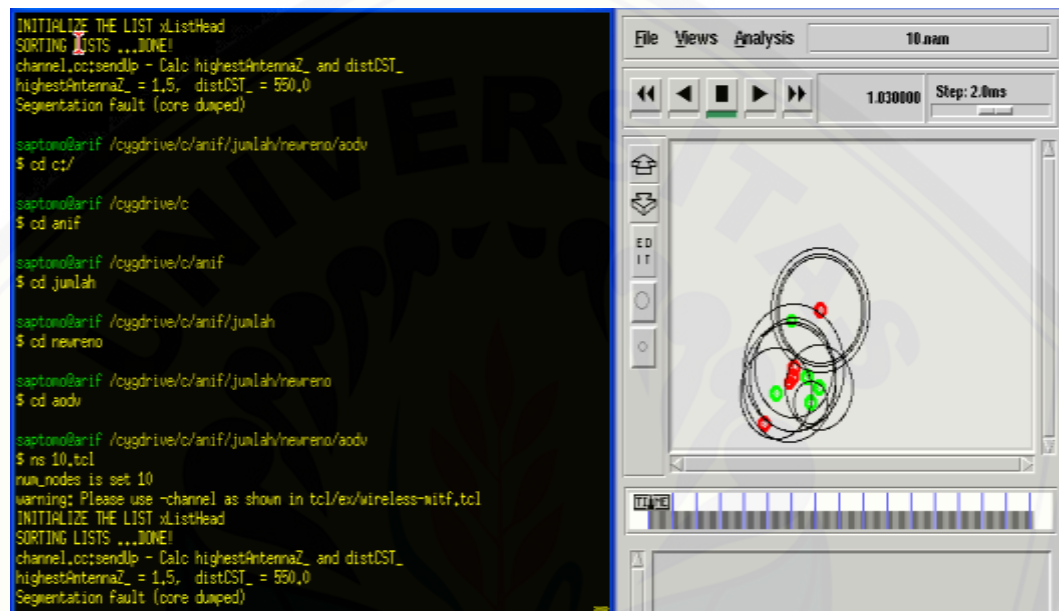
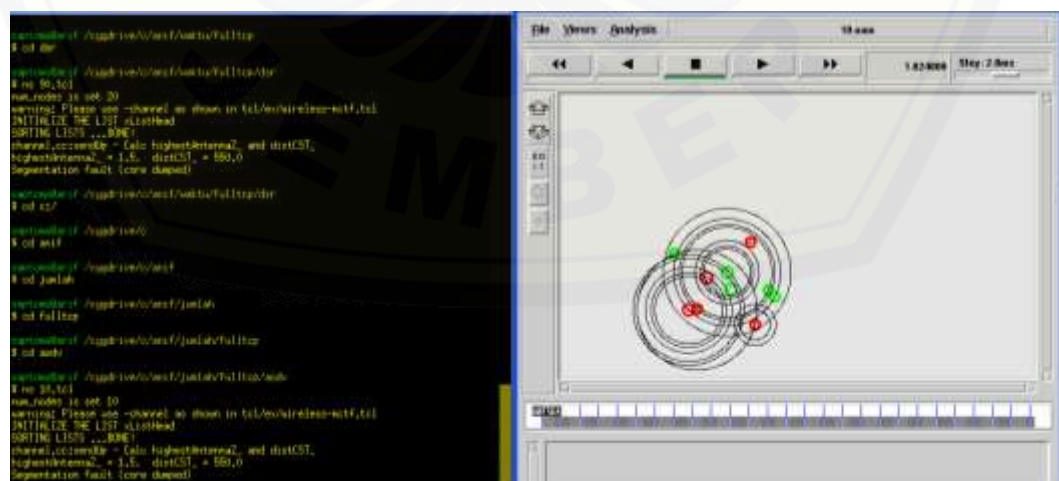
Rendy, 2007, Cara Kerja dan Kinerja Protokol DSR (*Dynamic Source Routing*) pada MANET (*Mobile Ad-Hoc Network*), Tugas Akhir. Jakarta: Fakultas Teknik Unika Atma Jaya.

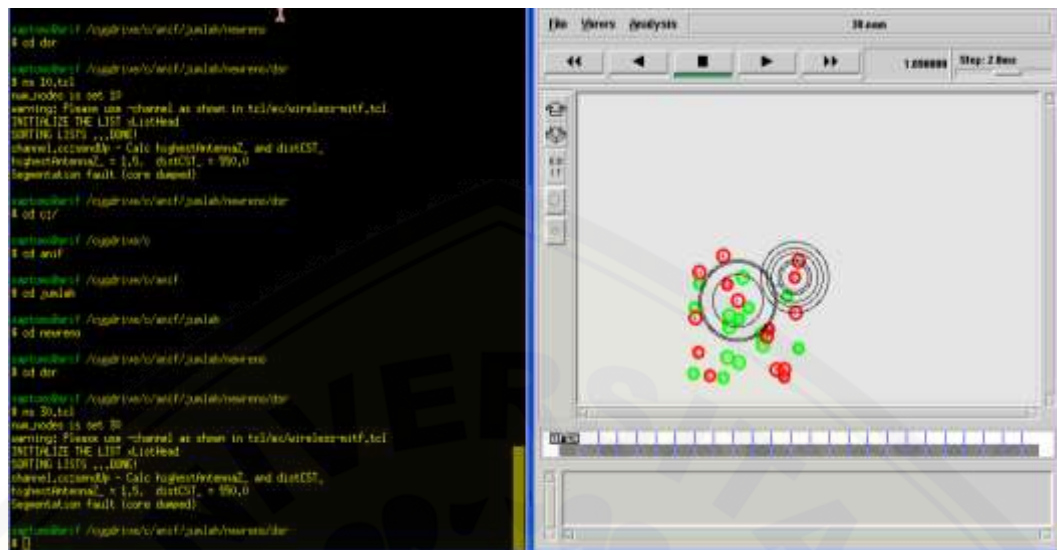
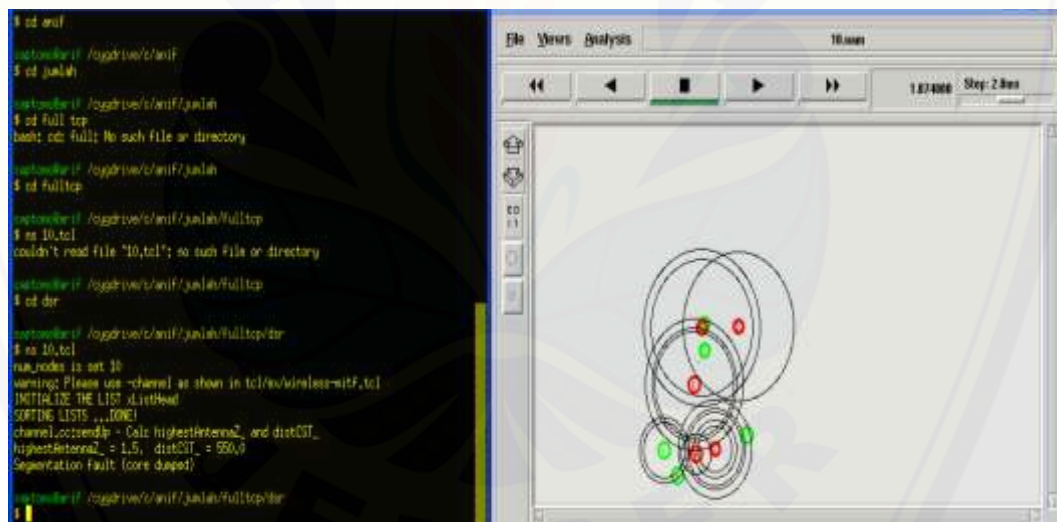
Sari, R.F., Abdusy Syarif., Bagio Budiardjo, Analisis Kinerja Protokol *Routing Ad Hoc On-Demand Distance Vector* (AODV) Pada Jaringan *Ad-Hoc Hybrid* : Perbandingan Hasil Simulasi dengan NS-2 dan Implementasi Pada *TESTBED* dengan PDA, Makara Teknologi, Volume 12, No.1 Hal 7-18, April 2008.

Uikey, Rajeshwari., Sanjeev Sharma, *Zigbee Cluster Tree Performance Improvement technique. International Journal of Applications* (0975-8887) volume 62-n0.19 january 2013

Jusak, Teknologi Komunikasi Data Modern, Andi Offset, Yogyakarta, 2013.

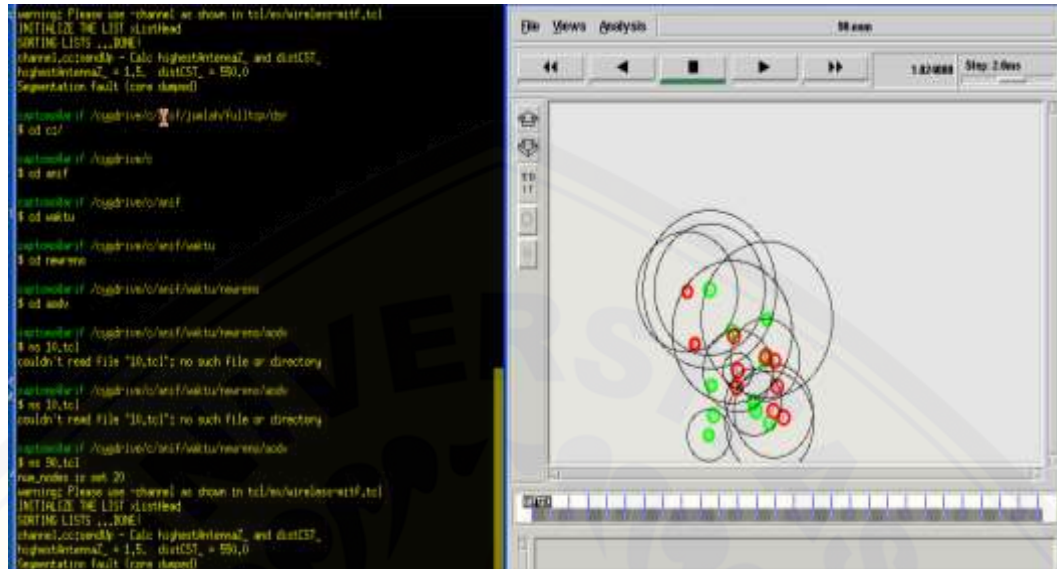
LAMPIRAN

Lampiran A. Gambar *node* Simulasi pada jaringan *Mobile ad-hoc* (MANET)1. Perubahan jumlah *nodes* simulasi pada *routing protocol*2. Gambar *node* simulasi *routing protocol* AODV NewrenoGambar 1. Skenariosaat perubahan jumlah *node* jaringan NewrenoAODV3. Gambar *node* simulasi *routing protocol* AODV FullTCPGambar 2. skenario 1 saat perubahan jumlah *node* jaringan full TCPnodeAODV

4. Gambar *node* simulasi *routing protocol* DSR AODVGambar 3. Skenario 1 saat perubahan jumlah *node* jaringan TCP Newreno DSR5. Gambar *node* simulasi *routing protocol* DSR FullTCPGambar 4. Skenario 1 saat perubahan jumlah *node* jaringan full TCP DSR

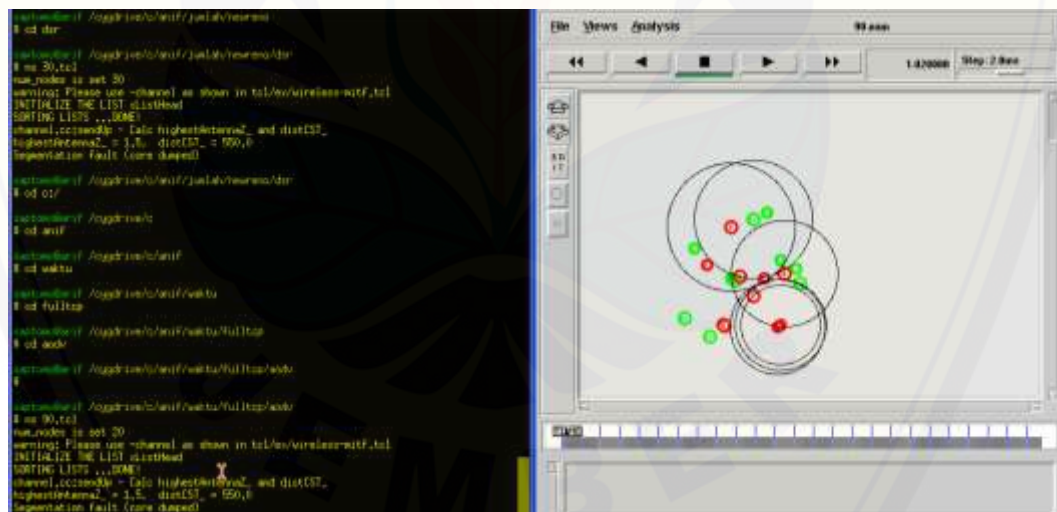
6. Perubahan waktu *node* simulasi pada *routing protocol*

7. Gambar *node* simulasi *routing protocol* AODV TCP Newreno



Gambar 5. Skenario 2saat perubahan waktu *node* jaringan TCP NewrenoAODV

8. Gambar *node* simulasi *routing protocol* AODV FullTCP



Gambar 6. Skenariosaat perubahan waktu *node* jaringan full TCP AODV

Lampiran B. Program Simulasi pada jaringan *Mobile ad-hoc* (MANET)

11. Perubahan jumlah *nodescripts* simulasi pada *routing protocol* AODV

Newreno

```

# Define options
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 10 ;# number of mobilenodes
set val(rp) AODV ;# routing protocol
set val(x) 1000 ;# X dimension of topography
set val(y) 1000 ;# Y dimension of topography
set val(stop) 120 ;# time of simulation end

#Creating simulation:
set ns [new Simulator]

#Creating nam and trace file:
set tracefd [open 10.tr w]
set namtrace [open 10.nam w]

$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)

# set up topography object
set topo [new Topography]

$topo load_flatgrid $val(x) $val(y)

set god_ [create-god $val(nn)]

# configure the nodes
$ns node-config -adhocRouting $val(rp) \
               -llType $val(ll) \
               -macType $val(mac) \
               -ifqType $val(ifq) \
               -ifqLen $val(ifqlen) \
               -antType $val(ant) \
               -propType $val(prop) \
               -phyType $val(netif) \

```

```

        -channelType $val(chan) \
        -topoInstance $topo \
        -agentTrace ON \
        -routerTrace ON \
        -macTrace ON \
        -movementTrace ON

## Creating node objects..
for {set i 0} {$i < 5 } { incr i } {
    set node_($i) [$ns node]
}
for {set i 0} {$i < 5 } {incr i } {
    $node_($i) color red
    $ns at 0.0 "$node_($i) color red"
}
for {set i 5} {$i < 10 } { incr i } {
    set node_($i) [$ns node]
}
for {set i 5} {$i < 10 } {incr i } {
    $node_($i) color white
    $ns at 1.0 "$node_($i) color green"
}

## Provide initial location of mobilenodes..
    for {set i 0} {$i < $val(nn) } { incr i } {
        set xx [expr rand()*500]
        set yy [expr rand()*500]
        $node_($i) set X_ $xx
        $node_($i) set Y_ $yy
    }

# Set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 1.0 "$ftp start"

# Set a TCP connection between node_(1) and node_(4)
set tcp [new Agent/TCP/Newreno]

```

```

$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(1) $tcp
$ns attach-agent $node_(4) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 3.0 "$ftp start"

# Set a TCP connection between node_(4) and node_(7)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(4) $tcp
$ns attach-agent $node_(7) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"

# Define node initial position in nam
for {set i 0} {$i < $val(nn)} { incr i } {
# 40 defines the node size for nam
$ns initial_node_pos $node_($i) 40
}

# Telling nodes when the simulation ends
for {set i 0} {$i < $val(nn)} { incr i } {
  $ns at $val(stop) "$node_($i) reset";
}

# dynamic destination setting procedure..
$ns at 0.0 "destination"
proc destination {} {
  global ns val node_
  set time 1.0
  set now [$ns now]
  for {set i 0} {$i < $val(nn)} { incr i } {
    set xx [expr rand()*500]
    set yy [expr rand()*500]
    $ns at $now "$node_($i) setdest $xx $yy 20.0"
  }
  $ns at [expr $now+$time] "destination"
}

```

```

#stop procedure..
$ns at $val(stop) "stop"
proc stop {} {
    global ns tracefd namtrace
    $ns flush-trace
    close $tracefd
    close $namtrace
exec nam 10.nam &
}

$ns run

```

12. Perubahan jumlah *nodescripts* simulasi pada *routing protocol* DSR

Newreno

```

# Define options
set val(chan)      Channel/WirelessChannel    ;# channel type
set val(prop)      Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)     Phy/WirelessPhy           ;# network interface type
set val(mac)       Mac/802_11                ;# MAC type
set val(ifq)       Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)        LL                         ;# link layer type
set val(ant)       Antenna/OmniAntenna       ;# antenna model
set val(ifqlen)    50                         ;# max packet in ifq
set val(nn)        10                         ;# number of mobilenodes
set val(rp)        DSR                       ;# routing protocol
set val(x)         1000                      ;# X dimension of topography
set val(y)         1000                      ;# Y dimension of topography
set val(stop)     120                       ;# time of simulation end

```

#Creating simulation:

```
set ns [new Simulator]
```

#Creating nam and trace file:

```
set tracefd [open 10.tr w]
```

```
set namtrace [open 10.nam w]
```

```
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)

# set up topography object
set topo [new Topography]

$topo load_flatgrid $val(x) $val(y)

set god_ [create-god $val(nn)]

# configure the nodes
$ns node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channelType $val(chan) \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace ON

## Creating node objects..
for {set i 0} {$i < 5} {incr i} {
    set node_($i) [$ns node]
}
```

```
for {set i 0} {$i < 5 } {incr i } {
    $node_($i) color red
    $ns at 0.0 "$node_($i) color red"
}
for {set i 5} {$i < 10 } { incr i } {
    set node_($i) [$ns node]
}
for {set i 5} {$i < 10 } {incr i } {
    $node_($i) color white
    $ns at 1.0 "$node_($i) color green"
}

## Provide initial location of mobilenodes..
for {set i 0} {$i < $val(nn) } { incr i } {
    set xx [expr rand()*500]
    set yy [expr rand()*500]
    $node_($i) set X_ $xx
    $node_($i) set Y_ $yy
}

# Set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
```



```
$ns at 1.0 "$ftp start"

# Set a TCP connection between node_(1) and node_(4)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(1) $tcp
$ns attach-agent $node_(4) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 3.0 "$ftp start"

# Set a TCP connection between node_(4) and node_(7)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(4) $tcp
$ns attach-agent $node_(7) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"

# Define node initial position in nam
for {set i 0} {$i < $val(nn)} { incr i } {
# 40 defines the node size for nam
$ns initial_node_pos $node_($i) 40
}
```

```
# Telling nodes when the simulation ends
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns at $val(stop) "$node_($i) reset";
}

# dynamic destination setting procedure..
$ns at 0.0 "destination"
proc destination {} {
    global ns val node_
    set time 1.0
    set now [$ns now]
    for {set i 0} {$i < $val(nn)} {incr i} {
        set xx [expr rand()*500]
        set yy [expr rand()*500]
        $ns at $now "$node_($i) setdest $xx $yy 20.0"
    }
    $ns at [expr $now+$time] "destination"
}

#stop procedure..
$ns at $val(stop) "stop"
proc stop {} {
    global ns tracefd namtrace
    $ns flush-trace
    close $tracefd
    close $namtrace
    exec nam 10.nam &
}

$ns run
```

13. Perubahan jumlah *nodescripts* simulasi pada *routing protocol AODV FullTCP*

```

# Define options
set val(chan)      Channel/WirelessChannel    ;# channel type
set val(prop)      Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)     Phy/WirelessPhy           ;# network interface type
set val(mac)       Mac/802_11                ;# MAC type
set val(ifq)       Queue/DropTail/PriQueue   ;# interface queue type
set val(ll)        LL                         ;# link layer type
set val(ant)       Antenna/OmniAntenna       ;# antenna model
set val(ifqlen)    50                         ;# max packet in ifq
set val(nn)        10                         ;# number of mobilenodes
set val(rp)        AODV                       ;# routing protocol
set val(x)         1000                       ;# X dimension of topography
set val(y)         1000                       ;# Y dimension of topography
set val(stop)      120                       ;# time of simulation end

#Creating simulation:
set ns              [new Simulator]

#Creating nam and trace file:
set tracefd        [open 10.tr w]
set namtrace       [open 10.nam w]

$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)

# set up topography object
set topo           [new Topography]

$topo load_flatgrid $val(x) $val(y)

```

```
set god_ [create-god $val(nn)]

# configure the nodes
  $ns node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channelType $val(chan) \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace ON

## Creating node objects..
for {set i 0} {$i < 5 } { incr i } {
  set node_($i) [$ns node]
}
for {set i 0} {$i < 5 } {incr i } {
  $node_($i) color red
  $ns at 0.0 "$node_($i) color red"
}
for {set i 5} {$i < 10 } { incr i } {
  set node_($i) [$ns node]
}
for {set i 5} {$i < 10 } {incr i } {
```

```
$node_($i) color white
$ns at 1.0 "$node_($i) color green"
}
```

```
## Provide initial location of mobilenodes..
```

```
for {set i 0} {$i < $val(nn)} {incr i} {
    set xx [expr rand()*500]
    set yy [expr rand()*500]
    $node_($i) set X_ $xx
    $node_($i) set Y_ $yy
}
```

```
# Set a TCP connection between node_(0) and node_(1)
```

```
set tcp [new Agent/TCP/FullTcp]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 1.0 "$ftp start"
```

```
# Set a TCP connection between node_(1) and node_(4)
```

```
set tcp [new Agent/TCP/FullTcp]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(1) $tcp
$ns attach-agent $node_(4) $sink
```

```
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 3.0 "$ftp start"

# Set a TCP connection between node_(4) and node_(7)
set tcp [new Agent/TCP/FullTcp]
$tcp set class_2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(4) $tcp
$ns attach-agent $node_(7) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"

# Define node initial position in nam
for {set i 0} {$i < $val(nn)} { incr i } {
# 40 defines the node size for nam
$ns initial_node_pos $node_($i) 40
}

# Telling nodes when the simulation ends
for {set i 0} {$i < $val(nn)} { incr i } {
    $ns at $val(stop) "$node_($i) reset";
}

# dynamic destination setting procedure..
$ns at 0.0 "destination"
proc destination {} {
```

```

global ns val node_
set time 1.0
set now [$ns now]
for {set i 0} {$i<$val(nn)} {incr i} {
    set xx [expr rand()*500]
    set yy [expr rand()*500]
    $ns at $now "$node_($i) setdest $xx $yy 20.0"
}
$ns at [expr $now+$time] "destination"
}

#stop procedure..
$ns at $val(stop) "stop"
proc stop {} {
    global ns tracefd namtrace
    $ns flush-trace
    close $tracefd
    close $namtrace
exec nam 10.nam &
}

$ns run

```

14. Perubahan jumlah *nodescripts* simulasi pada *routing protocol* DSR

FullTCP

```

# Define options
set val(chan)      Channel/WirelessChannel    ;# channel type
set val(prop)      Propagation/TwoRayGround    ;# radio-propagation model
set val(netif)     Phy/WirelessPhy            ;# network interface type
set val(mac)        Mac/802_11                ;# MAC type
set val(ifq)        Queue/DropTail/PriQueue    ;# interface queue type

```

```
set val(ll)      LL           ;# link layer type
set val(ant)     Antenna/OmniAntenna ;# antenna model
set val(ifqlen)  50           ;# max packet in ifq
set val(nn)      10           ;# number of mobilenodes
set val(rp)      DSR          ;# routing protocol
set val(x)       1000         ;# X dimension of topography
set val(y)       1000         ;# Y dimension of topography
set val(stop)    120          ;# time of simulation end
```

```
#Creating simulation:
```

```
set ns          [new Simulator]
```

```
#Creating nam and trace file:
```

```
set tracefd     [open 10.tr w]
```

```
set namtrace    [open 10.nam w]
```

```
$ns trace-all $tracefd
```

```
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
```

```
# set up topography object
```

```
set topo        [new Topography]
```

```
$topo load_flatgrid $val(x) $val(y)
```

```
set god_        [create-god $val(nn)]
```

```
# configure the nodes
```

```
    $ns node-config -adhocRouting $val(rp) \
```

```
        -llType $val(ll) \
```

```
        -macType $val(mac) \
```

```
        -ifqType $val(ifq) \
```



```
-ifqLen $val(ifqlen) \  
-antType $val(ant) \  
-propType $val(prop) \  
-phyType $val(netif) \  
-channelType $val(chan) \  
-topoInstance $topo \  
-agentTrace ON \  
-routerTrace ON \  
-macTrace ON \  
-movementTrace ON  
  
## Creating node objects..  
for {set i 0} {$i < 5 } { incr i } {  
    set node_($i) [$ns node]  
}  
for {set i 0} {$i < 5 } {incr i } {  
    $node_($i) color red  
    $ns at 0.0 "$node_($i) color red"  
}  
for {set i 5} {$i < 10 } { incr i } {  
    set node_($i) [$ns node]  
}  
for {set i 5} {$i < 10 } {incr i } {  
    $node_($i) color white  
    $ns at 1.0 "$node_($i) color green"  
}  
  
## Provide initial location of mobilenodes..  
for {set i 0} {$i < $val(nn) } { incr i } {  
    set xx [expr rand()*500]
```

```
        set yy [expr rand()*500]
        $node_($i) set X_ $xx
        $node_($i) set Y_ $yy
    }

# Set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/FullTcp]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 1.0 "$ftp start"

# Set a TCP connection between node_(1) and node_(4)
set tcp [new Agent/TCP/FullTcp]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(1) $tcp
$ns attach-agent $node_(4) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 3.0 "$ftp start"

# Set a TCP connection between node_(4) and node_(7)
set tcp [new Agent/TCP/FullTcp]
$tcp set class_ 2
```

```
set sink [new Agent/TCPSink]
$ns attach-agent $node_(4) $tcp
$ns attach-agent $node_(7) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"

# Define node initial position in nam
for {set i 0} {$i < $val(nn)} { incr i } {
# 40 defines the node size for nam
$ns initial_node_pos $node_($i) 40
}

# Telling nodes when the simulation ends
for {set i 0} {$i < $val(nn)} { incr i } {
    $ns at $val(stop) "$node_($i) reset";
}

# dynamic destination setting procedure..
$ns at 0.0 "destination"
proc destination {} {
    global ns val node_
    set time 1.0
    set now [$ns now]
    for {set i 0} {$i < $val(nn)} { incr i } {
        set xx [expr rand()*500]
        set yy [expr rand()*500]
        $ns at $now "$node_($i) setdest $xx $yy 20.0"
    }
}
```

```

    $ns at [expr $now+$time] "destination"
}

```

```

#stop procedure..

```

```

$ns at $val(stop) "stop"

```

```

proc stop {} {
    global ns tracefd namtrace
    $ns flush-trace
    close $tracefd
    close $namtrace
    exec nam 10.nam &
}

```

```

$ns run

```

15. Perubahan *waktunodescripts* simulasi pada *routing protocol* AODV *Newreno*

```

# Define options

```

```

set val(chan)      Channel/WirelessChannel    ;# channel type
set val(prop)      Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)     Phy/WirelessPhy           ;# network interface type
set val(mac)       Mac/802_11                ;# MAC type
set val(ifq)       Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)        LL                        ;# link layer type
set val(ant)       Antenna/OmniAntenna       ;# antenna model
set val(ifqlen)    50                        ;# max packet in ifq
set val(nn)        20                        ;# number of mobilenodes
set val(rp)        AODV                      ;# routing protocol
set val(x)         1000                      ;# X dimension of topography
set val(y)         1000                      ;# Y dimension of topography
set val(stop)      90                        ;# time of simulation end

```

#Creating simulation:

```
set ns      [new Simulator]
```

#Creating nam and trace file:

```
set tracefd  [open 90.tr w]
```

```
set namtrace  [open 90.nam w]
```

```
$ns trace-all $tracefd
```

```
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
```

set up topography object

```
set topo     [new Topography]
```

```
$topo load_flatgrid $val(x) $val(y)
```

```
set god_ [create-god $val(nn)]
```

configure the nodes

```
    $ns node-config -adhocRouting $val(rp) \
```

```
        -llType $val(ll) \
```

```
        -macType $val(mac) \
```

```
        -ifqType $val(ifq) \
```

```
        -ifqLen $val(ifqlen) \
```

```
        -antType $val(ant) \
```

```
        -propType $val(prop) \
```

```
        -phyType $val(netif) \
```

```
        -channelType $val(chan) \
```

```
        -topoInstance $topo \
```

```
        -agentTrace ON \
```

```
        -routerTrace ON \
```

```
-macTrace ON \  
-movementTrace ON  
  
## Creating node objects..  
for {set i 0} {$i < 10 } { incr i } {  
    set node_($i) [$ns node]  
}  
for {set i 0} {$i < 10 } {incr i } {  
    $node_($i) color red  
    $ns at 0.0 "$node_($i) color red"  
}  
for {set i 10} {$i < 20 } { incr i } {  
    set node_($i) [$ns node]  
}  
for {set i 10} {$i < 20 } {incr i } {  
    $node_($i) color white  
    $ns at 1.0 "$node_($i) color green"  
}  
  
## Provide initial location of mobilenodes..  
for {set i 0} {$i < $val(nn) } { incr i } {  
    set xx [expr rand()*500]  
    set yy [expr rand()*500]  
    $node_($i) set X_ $xx  
    $node_($i) set Y_ $yy  
  
}  
  
# Set a TCP connection between node_(0) and node_(1)  
set tcp [new Agent/TCP/Newreno]
```

```
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 1.0 "$ftp start"

# Set a TCP connection between node_(1) and node_(4)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(1) $tcp
$ns attach-agent $node_(4) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 3.0 "$ftp start"

# Set a TCP connection between node_(4) and node_(7)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(4) $tcp
$ns attach-agent $node_(7) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"
```

```
# Define node initial position in nam
for {set i 0} {$i < $val(nn)} { incr i } {
# 40 defines the node size for nam
$ns initial_node_pos $node_($i) 40
}

# Telling nodes when the simulation ends
for {set i 0} {$i < $val(nn)} { incr i } {
  $ns at $val(stop) "$node_($i) reset";
}

# dynamic destination setting procedure..
$ns at 0.0 "destination"
proc destination {} {
  global ns val node_
  set time 1.0
  set now [$ns now]
  for {set i 0} {$i < $val(nn)} { incr i } {
    set xx [expr rand()*500]
    set yy [expr rand()*500]
    $ns at $now "$node_($i) setdest $xx $yy 20.0"
  }
  $ns at [expr $now+$time] "destination"
}

#stop procedure..
$ns at $val(stop) "stop"
proc stop {} {
  global ns tracefd namtrace
  $ns flush-trace
```



```
close $tracefd
close $namtrace
exec nam 90.nam &
}
```

```
$ns run
```

16. Perubahan waktunodescript simulasi pada routing protocol DSR Newreno

```
# Define options
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 20 ;# number of mobilenodes
set val(rp) DSR ;# routing protocol
set val(x) 1000 ;# X dimension of topography
set val(y) 1000 ;# Y dimension of topography
set val(stop) 90 ;# time of simulation end

#Creating simulation:
set ns [new Simulator]

#Creating nam and trace file:
set tracefd [open 90.tr w]
set namtrace [open 90.nam w]

$ns trace-all $tracefd
```

```
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
```

```
# set up topography object
```

```
set topo [new Topography]
```

```
$topo load_flatgrid $val(x) $val(y)
```

```
set god_ [create-god $val(nn)]
```

```
# configure the nodes
```

```
  $ns node-config -adhocRouting $val(rp) \
```

```
    -llType $val(ll) \
```

```
    -macType $val(mac) \
```

```
    -ifqType $val(ifq) \
```

```
    -ifqLen $val(ifqlen) \
```

```
    -antType $val(ant) \
```

```
    -propType $val(prop) \
```

```
    -phyType $val(netif) \
```

```
    -channelType $val(chan) \
```

```
    -topoInstance $topo \
```

```
    -agentTrace ON \
```

```
    -routerTrace ON \
```

```
    -macTrace ON \
```

```
    -movementTrace ON
```

```
## Creating node objects..
```

```
for {set i 0} {$i < 10 } {incr i } {
```

```
  set node_($i) [$ns node]
```

```
}
```

```
for {set i 0} {$i < 10 } {incr i } {
```

```
  $node_($i) color red
```

```
    $ns at 0.0 "$node_($i) color red"
  }
for {set i 10} {$i < 20 } { incr i } {
  set node_($i) [$ns node]
}
for {set i 10} {$i < 20 } {incr i } {
  $node_($i) color white
  $ns at 1.0 "$node_($i) color green"
}

## Provide initial location of mobilenodes..
  for {set i 0} {$i < $val(nn) } { incr i } {
    set xx [expr rand()*500]
    set yy [expr rand()*500]
    $node_($i) set X_ $xx
    $node_($i) set Y_ $yy
  }

# Set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 1.0 "$ftp start"
```

```
# Set a TCP connection between node_(1) and node_(4)
```

```
set tcp [new Agent/TCP/Newreno]
```

```
$tcp set class_ 2
```

```
set sink [new Agent/TCPSink]
```

```
$ns attach-agent $node_(1) $tcp
```

```
$ns attach-agent $node_(4) $sink
```

```
$ns connect $tcp $sink
```

```
set ftp [new Application/FTP]
```

```
$ftp attach-agent $tcp
```

```
$ns at 3.0 "$ftp start"
```

```
# Set a TCP connection between node_(4) and node_(7)
```

```
set tcp [new Agent/TCP/Newreno]
```

```
$tcp set class_ 2
```

```
set sink [new Agent/TCPSink]
```

```
$ns attach-agent $node_(4) $tcp
```

```
$ns attach-agent $node_(7) $sink
```

```
$ns connect $tcp $sink
```

```
set ftp [new Application/FTP]
```

```
$ftp attach-agent $tcp
```

```
$ns at 5.0 "$ftp start"
```

```
# Define node initial position in nam
```

```
for {set i 0} {$i < $val(nn)} { incr i } {
```

```
# 40 defines the node size for nam
```

```
$ns initial_node_pos $node_($i) 40
```

```
}
```

```
# Telling nodes when the simulation ends
```

```
for {set i 0} {$i < $val(nn)} { incr i } {
```

```
$ns at $val(stop) "$node_($i) reset";
}

# dynamic destination setting procedure..
$ns at 0.0 "destination"
proc destination {} {
    global ns val node_
    set time 1.0
    set now [$ns now]
    for {set i 0} {$i<$val(nn)} {incr i} {
        set xx [expr rand()*500]
        set yy [expr rand()*500]
        $ns at $now "$node_($i) setdest $xx $yy 20.0"
    }
    $ns at [expr $now+$time] "destination"
}

#stop procedure..
$ns at $val(stop) "stop"
proc stop {} {
    global ns tracefd namtrace
    $ns flush-trace
    close $tracefd
    close $namtrace
    exec nam 90.nam &
}

$ns run
```

17. Perubahan *waktunodescripts* simulasi pada *routing protocol AODV FullTCP*

```

# Define options
set val(chan)      Channel/WirelessChannel    ;# channel type
set val(prop)      Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)     Phy/WirelessPhy           ;# network interface type
set val(mac)       Mac/802_11                ;# MAC type
set val(ifq)       Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)        LL                         ;# link layer type
set val(ant)       Antenna/OmniAntenna       ;# antenna model
set val(ifqlen)    50                         ;# max packet in ifq
set val(nn)        20                         ;# number of mobilenodes
set val(rp)        AODV                       ;# routing protocol
set val(x)         1000                       ;# X dimension of topography
set val(y)         1000                       ;# Y dimension of topography
set val(stop)     90                         ;# time of simulation end

#Creating simulation:
set ns              [new Simulator]

#Creating nam and trace file:
set tracefd        [open 90.tr w]
set namtrace       [open 90.nam w]

$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)

# set up topography object
set topo           [new Topography]

$topo load_flatgrid $val(x) $val(y)

```

```
set god_ [create-god $val(nn)]

# configure the nodes
  $ns node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channelType $val(chan) \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace ON

## Creating node objects..
for {set i 0} {$i < 10 } { incr i } {
  set node_($i) [$ns node]
}
for {set i 0} {$i < 10 } {incr i } {
  $node_($i) color red
  $ns at 0.0 "$node_($i) color red"
}
for {set i 10} {$i < 20 } { incr i } {
  set node_($i) [$ns node]
}
for {set i 10} {$i < 20 } {incr i } {
```

```
$node_($i) color white
$ns at 1.0 "$node_($i) color green"
}

## Provide initial location of mobilenodes..
for {set i 0} {$i < $val(nn)} {incr i} {
    set xx [expr rand()*500]
    set yy [expr rand()*500]
    $node_($i) set X_ $xx
    $node_($i) set Y_ $yy
}

# Set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/FullTcp]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 1.0 "$ftp start"

# Set a TCP connection between node_(1) and node_(4)
set tcp [new Agent/TCP/FullTcp]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(1) $tcp
$ns attach-agent $node_(4) $sink
```



```
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 3.0 "$ftp start"

# Set a TCP connection between node_(4) and node_(7)
set tcp [new Agent/TCP/FullTcp]
$tcp set class_2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(4) $tcp
$ns attach-agent $node_(7) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"

# Define node initial position in nam
for {set i 0} {$i < $val(nn)} { incr i } {
# 40 defines the node size for nam
$ns initial_node_pos $node_($i) 40
}

# Telling nodes when the simulation ends
for {set i 0} {$i < $val(nn)} { incr i } {
    $ns at $val(stop) "$node_($i) reset";
}

# dynamic destination setting procedure..
$ns at 0.0 "destination"
proc destination {} {
```

```

global ns val node_
set time 1.0
set now [$ns now]
for {set i 0} {$i<$val(nn)} {incr i} {
    set xx [expr rand()*500]
    set yy [expr rand()*500]
    $ns at $now "$node_($i) setdest $xx $yy 20.0"
}
$ns at [expr $now+$time] "destination"
}

#stop procedure..
$ns at $val(stop) "stop"
proc stop {} {
    global ns tracefd namtrace
    $ns flush-trace
    close $tracefd
    close $namtrace
exec nam 90.nam &
}

$ns run

```

18. Perubahan waktunodescriptsimulasi pada routing protocol DSR FullTCP

```

# Define options
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type

```

```
set val(ant)      Antenna/OmniAntenna      ;# antenna model
set val(ifqlen)   50                        ;# max packet in ifq
set val(nn)       20                        ;# number of mobilenodes
set val(rp)       DSR                       ;# routing protocol
set val(x)        1000                      ;# X dimension of topography
set val(y)        1000                      ;# Y dimension of topography
set val(stop)     90                        ;# time of simulation end

#Creating simulation:
set ns            [new Simulator]

#Creating nam and trace file:
set tracefd      [open 90.tr w]
set namtrace     [open 90.nam w]

$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)

# set up topography object
set topo        [new Topography]

$topo load_flatgrid $val(x) $val(y)

set god_ [create-god $val(nn)]

# configure the nodes
$ns node-config -adhocRouting $val(rp) \
               -llType $val(ll) \
               -macType $val(mac) \
               -ifqType $val(ifq) \
               -ifqLen $val(ifqlen) \
```

```
-antType $val(ant) \  
-propType $val(prop) \  
-phyType $val(netif) \  
-channelType $val(chan) \  
-topoInstance $topo \  
-agentTrace ON \  
-routerTrace ON \  
-macTrace ON \  
-movementTrace ON  
  
## Creating node objects..  
for {set i 0} {$i < 10 } { incr i } {  
    set node_($i) [$ns node]  
}  
for {set i 0} {$i < 10 } {incr i } {  
    $node_($i) color red  
    $ns at 0.0 "$node_($i) color red"  
}  
for {set i 10} {$i < 20 } { incr i } {  
    set node_($i) [$ns node]  
}  
for {set i 10} {$i < 20 } {incr i } {  
    $node_($i) color white  
    $ns at 1.0 "$node_($i) color green"  
}  
  
## Provide initial location of mobilenodes..  
for {set i 0} {$i < $val(nn) } { incr i } {  
    set xx [expr rand()*500]  
    set yy [expr rand()*500]
```

```
        $node_($i) set X_ $xx
        $node_($i) set Y_ $yy
    }

# Set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/FullTcp]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 1.0 "$ftp start"

# Set a TCP connection between node_(1) and node_(4)
set tcp [new Agent/TCP/FullTcp]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(1) $tcp
$ns attach-agent $node_(4) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 3.0 "$ftp start"

# Set a TCP connection between node_(4) and node_(7)
set tcp [new Agent/TCP/FullTcp]
$tcp set class_ 2
set sink [new Agent/TCPSink]
```

```

$ns attach-agent $node_(4) $tcp
$ns attach-agent $node_(7) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"

# Define node initial position in nam
for {set i 0} {$i < $val(nn)} { incr i } {
# 40 defines the node size for nam
$ns initial_node_pos $node_($i) 40
}

# Telling nodes when the simulation ends
for {set i 0} {$i < $val(nn)} { incr i } {
    $ns at $val(stop) "$node_($i) reset";
}

# dynamic destination setting procedure..
$ns at 0.0 "destination"
proc destination {} {
    global ns val node_
    set time 1.0
    set now [$ns now]
    for {set i 0} {$i < $val(nn)} { incr i } {
        set xx [expr rand()*500]
        set yy [expr rand()*500]
        $ns at $now "$node_($i) setdest $xx $yy 20.0"
    }
    $ns at [expr $now+$time] "destination"
}

```

```

}

#stop procedure..
$ns at $val(stop) "stop"
proc stop {} {
    global ns tracefd namtrace
    $ns flush-trace
    close $tracefd
    close $namtrace
    exec nam 90.nam &
}

$ns run

```

Lampiran C. Program AWK Simulasi pada jaringan *Mobile ad-hoc*(MANET)

1. *Throughput*.AWK

```

BEGIN {
    recvdSize = 0
    startTime = 0
    stopTime = 120
}

{
    event = $1
    time = $2
    node_id = $3
    pkt_size = $12
    level = $4

    # Store start time
    if (level == "AGT" && event == "s" && pkt_size >= 800) {
        if (time < startTime) {
            startTime = time
        }
    }
}

```

```

# Update total received packets' size and store packets arrival time
if (level == "AGT" && event == "r" && pkt_size >= 800) {
    if (time > stopTime) {
        stopTime = time
    }
    # Rip off the header
    hdr_size = pkt_size % 800
    pkt_size -= hdr_size
    # Store received packet's size
    recvdSize += pkt_size
}
}

END {
    printf("Average      Throughput[kbps]      =      %.2f\t\t\t
    StartTime=%.2f\t\tStopTime=%.2f\n", (recvdSize/(stopTime-
    startTime))*(8/1000), startTime, stopTime)
}

```

19. Delay.AWK

```

BEGIN{
sumdelay=0;
countdelay=0;
}
{
    sumdelay += $10;
    countdelay++;
}
END{
printf("Average Delay[s%] = %f \n", sumdelay/countdelay);
}

```

3. Routing Overhead(RO).AWK

```

BEGIN{
sumdelay=0;
countdelay=0;
}
{
    sumdelay += $10;
    countdelay++;
}
END{

```



```
printf("Average Delay[s%] = %f \n",sumdelay/countdelay);  
}
```

