

**PENCARIAN RUTE YANG HANDAL BERBASIS ENERGI MENGGUNAKAN
ALGORITMA EA-SHORT PADA PROTOKOL ROUTING ZRP DI
JARINGAN MANET**

Tugas Akhir

Untuk Memenuhi sebagian persyaratan
mencapai derajat Sarjana S-1 Program Studi Teknik Infomatika



Oleh:

MILA ROSIANA

F1D 015 048

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MATARAM

2020

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Demikian surat pernyataan ini dibuat tanpa tekanan dari pihak mana pun dan dengan kesadaran penuh terhadap tanggung jawab dan konsekuensi dari pernyataan tersebut.

Mataram, Juli 2020

Mila Rosiana

PRAKATA

Puji syukur penulis panjatkan kehadiran Allah Subhanahu Wa Ta'ala serta junjungan besar Nabi Muhammad Sallallahu 'Alaihi Wasallam, yang telah melimpahkan Rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan Tugas Akhir untuk memenuhi persyaratan akademis guna mencapai gelar kesarjanaan di Fakultas Teknik Universitas Mataram dengan judul "Pencarian Rute yang Handal Berbasis Energi Menggunakan Algoritma EA-SHORT Pada Protokol *Routing* ZRP di Jaringan MANET".

Dalam penulisan Tugas Akhir ini tentunya tidak luput dari kekurangan, baik aspek kualitas maupun aspek kuantitas dari materi penelitian yang disajikan. Semua ini didasarkan dari keterbatasan yang dimiliki penulis. Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kata sempurna sehingga penulis membutuhkan kritik dan saran yang membangun untuk kemajuan teknologi di masa yang akan datang. Akhir kata semoga tidaklah terlampau berlebihan, bila penulis berharap agar karya ini dapat bermanfaat bagi pembaca.

Mataram, Juli 2020

Mila Rosiana

UCAPAN TERIMA KASIH

Tugas Akhir ini dapat diselesaikan berkat bimbingan dan dukungan ilmiah maupun materiil dari berbagai pihak, oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih yang setulus-tulusnya kepada:

1. Kedua Orang Tua dan keluarga yang selama ini telah memberikan bimbingan, doa dan dukungannya dalam menjalani perkuliahan dan pengerjaan Tugas Akhir ini.
2. Bapak Andy Hidayat Jatmika, S.T., M.Kom., selaku dosen pembimbing utama yang telah memberikan bimbingan dan arahan kepada penulis selama penyusunan Tugas Akhir ini, sehingga dapat terselesaikan dengan baik.
3. Bapak Ariyan Zubaidi, S.Kom.,M.T., selaku dosen pembimbing pendamping yang telah memberikan bimbingan dan arahan kepada penulis selama penyusunan Tugas Akhir ini, sehingga dapat terselesaikan dengan baik.
4. Bapak Dr. Eng. I Gede Putu Wirarama Wedashwara W., S.T., M.T., Bapak Ahmad Zafrullah Mardiansyah, S.T., M.Eng. dan Bapak Fitri Bimantoro.,S.T.,M.Kom., selaku dosen penguji yang telah memberikan kritik dan saran yang bersifat membangun dalam penyelesaian tugas akhir ini.
5. Sahabat penulis, B. Wahyu Hairani, Marlia Zuhraeni, Relanti Saleha, Islam Hidayah, Emi Setiarini, Untuk Group Penguasa, dan Annisa Arindasari Yunia Rusman juga beserta keluarganya, serta teman-teman Teknik Informatika angkatan 2015 dan Mahasiswa Universitas Mataram lainnya atas segala bentuk dukungan dan semangat yang telah diberikan kepada penulis.
6. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan dukungan dan bimbingan kepada penulis dalam menyelesaikan Tugas Akhir ini.

Semoga Tuhan Yang Maha Pengasih lagi Maha Penyayang memberikan imbalan yang setimpal atas bantuan yang telah diberikan kepada penulis.

DAFTAR ISI

DAFTAR ISI.....	ivii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xi
ABSTRAK.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika Penulisan Penelitian.....	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Penelitian Terkait	5
2.3 Penelitian yang Diusulkan.....	8
2.4 Dasar Teori.....	8
2.4.1 Mobile Ad Hoc (MANET)	8
2.4.2 Protokol <i>Routing</i> pada MANET	9
2.4.3 Zone Routing Protocol (ZRP)	9
2.4.4 Mekanisme EA-SHORT.....	11
2.4.5 NS 2	12
BAB III METODE PENELITIAN	13
3.1 Diagram Alir Penelitian	13
3.1.1 Studi Literatur.....	14
3.1.2 Persiapan Perangkat <i>Hardware</i> dan <i>Software</i>	14
3.1.3 Pembuatan <i>Script</i> EA-SHORT	15
3.1.4 Perancangan Skenario Simulasi	15
3.1.5 Menjalankan Skenario simulasi.....	16
3.1.6 Analisis Hasil Simulasi.....	16
3.1.7 Membuat Grafik Hasil.....	16
3.1.8 Membuat Kesimpulan	16
3.1.9 Membuat Laporan Dokumentasi	17
3.2 Algoritma Pencarian Rute Menggunakan Protokol <i>Routing</i> ZRP	17
3.3 Algoritma EA-SHORT ZRP	19

3.4	Parameter Skenario Simulasi	22
BAB IV HASIL DAN PEMBAHASAN		24
4.1	Implementasi Metode	24
4.1.1	Pemodelan energi	24
4.1.2	Algoritma EA-SHORT	25
4.2	Merancang Skenario Simulasi.....	28
4.2.1	Membuat pola trafik	28
4.2.2	Membuat pola pergerakan <i>node</i>	29
4.2.3	Membuat skrip TCL	29
4.2.4	Melakukan <i>filtering</i> pada <i>file trace</i> (.tr)	31
4.3	Hasil Uji Coba.....	31
4.3.1	Hasil Uji Coba	31
4.4	Analisis kinerja hasil simulasi jaringan.....	38
BAB V KESIMPULAN DAN SARAN		47
5.1	Kesimpulan	47
5.2	Saran.....	47
DAFTAR PUSTAKA		49

DAFTAR GAMBAR

Gambar 2.1 Protokol <i>routing</i> ZRP.....	9
Gambar 2.2 Proses pencarian rute dengan algoritma EA-SHORT.....	11
Gambar 3.1 Diagram alir penelitian	13
Gambar 3.2 Proses pencarian rute pada protokol <i>routing</i> ZRP	17
Gambar 3.3 <i>Routing zone</i> pada <i>node</i> M.....	18
Gambar 3.4 <i>Routing zone</i> pada <i>node</i> P	18
Gambar 3.5 <i>Routing zone</i> pada <i>node</i> K	18
Gambar 3.6 <i>Flowchart</i> pencarian rute pada EA-SHORT.....	19
Gambar 3.7 <i>Flowchart</i> pencarian rute pada EA-SHORT (Lanjutan Gambar 3.6).....	20
Gambar 3.8 Contoh proses perutean ulang menggunakan algoritma EA-SHORT	22
Gambar 4. 1 Deklarasi variabel pada <i>zrp.h</i>	24
Gambar 4. 2 Inisialisasi variabel pada <i>zrp.cc</i>	25
Gambar 4. 3 Inisialisasi variabel untuk menampung <i>hop count</i> dan level energi	25
Gambar 4. 4 Skrip untuk melakukan penambahan <i>link</i> (rute)	26
Gambar 4. 5 Skrip untuk kondisi <i>sequence number</i> paket kurang dari <i>sequence number</i> paket yang sudah ada	26
Gambar 4. 6 Skrip untuk kondisi <i>sequence number</i> paket lebih besar daripada <i>sequence number</i> paket yang sudah ada	27
Gambar 4. 7 Skrip algoritma EA-SHORT.....	28
Gambar 4. 8 Membuat pola trafik.....	28
Gambar 4. 9 Membuat pola pergerakan <i>node</i>	29
Gambar 4. 10 Skrip TCL untuk ZRP	30
Gambar 4. 11 Skrip perintah untuk menjalankan <i>file trace</i>	31
Gambar 4. 12 Grafik perbandingan hasil uji coba 50 <i>node</i> terhadap nilai <i>throughput</i> pada (a) uji coba ke 1 (b) uji coba ke 2, (c) uji coba ke 3, (d) uji coba ke 4 (e) uji coba ke 5.....	39
Gambar 4. 13 Grafik perbandingan hasil uji coba 100 <i>node</i> terhadap nilai <i>throughput</i> pada (a) uji coba ke 1, (b) uji coba ke 2, (c) uji coba ke 3, (d) uji coba ke 4 (e) uji coba ke 5.....	39
Gambar 4. 14 Grafik perbandingan hasil uji coba 50 <i>node</i> terhadap nilai <i>average end to end delay</i> pada (a) uji coba ke 1, (b) uji coba ke 2, (c) uji coba ke 3, (d) uji coba ke 4, (e) uji coba ke 5	42

Gambar 4. 15 Grafik perbandingan hasil uji coba 100 *node* terhadap nilai *average end to end delay* pada (a) uji coba ke 1, (b) uji coba ke 2, (c) uji coba ke 3, (d) uji coba ke 4, (e) uji coba ke 5 42

Gambar 4. 16 Grafik perbandingan hasil uji coba 50 *node* terhadap nilai PDR pada (a) uji coba ke 1, (b) uji coba ke 2, (c) uji coba ke 3, (d) uji coba ke 4 (e) uji coba ke 5..... 44

Gambar 4. 17 Grafik perbandingan hasil uji coba 100 *node* terhadap nilai PDR pada (a) uji coba ke 1, (b) uji coba ke 2, (c) uji coba ke 3, (d) uji coba ke 4 (e) uji coba ke 5..... 45

DAFTAR TABEL

Tabel 3.1 Komponen <i>hardware</i> dan <i>software</i>	14
Tabel 3.2 Parameter skenario simulasi	23
Tabel 3.3 Rencana pelaksanaan penelitian	24
Tabel 4. 1 Hasil uji coba protokol ZRP standar dengan 50 <i>node</i> terhadap <i>throughput</i> ..	32
Tabel 4. 2 Hasil uji coba protokol ZRP standar dengan 100 <i>node</i> terhadap <i>throughput</i>	33
Tabel 4. 3 Hasil uji coba protokol ZRP standar dengan 50 <i>node</i> terhadap <i>average end to end delay</i>	34
Tabel 4. 4 Hasil uji coba protokol ZRP standar dengan 100 <i>node</i> terhadap <i>average end to end delay</i>	35
Tabel 4. 5 Hasil uji coba 50 <i>node</i> terhadap PDR.....	36
Tabel 4. 6 Hasil uji coba 100 <i>node</i> terhadap PDR.....	37

ABSTRAK

Mobile Ad-Hoc Network (MANET) adalah jaringan *wireless* dari kumpulan *node* yang tidak memiliki *router* tetap. Setiap *node* dalam jaringan bertindak sebagai *router* yang bertanggung jawab untuk menemukan dan menangani rute antar *node*. *Node* berperan sekaligus sebagai *router* sehingga konsumsi energi menjadi lebih tinggi. Sedangkan, energi untuk proses transmisi pada *node* bersifat terbatas, dan habisnya energi dapat menyebabkan rute terputus. Terputusnya rute dapat menurunkan performa jaringan. Dalam penelitian ini, konsep *energy aware* menggunakan algoritma EASHORT diterapkan pada kerangka kerja Zone Routing Protocol (ZRP). EA-SHORT mencoba mendistribusikan beban jaringan ke semua *node* yang ada dengan memanfaatkan variasi jumlah energi dengan memilih *node* yang memiliki cukup energi yang dapat berpartisipasi dalam rute dan menghindari *node* yang memiliki energi rendah. Kinerja ZRP akan dibandingkan dengan EA-SHORT ZRP yang telah dimodifikasi dengan EA-SHORT yang diukur dari nilai parameter yang ditentukan. Dari simulasi, hasilnya menunjukkan, pada *node* 50, *throughput* meningkat sebesar 12,374%. Untuk 100 *node*, peningkatan sebesar 44.597%. Pada rata-rata *average end to end delay*, dengan 50 *node*, nilai EA-SHORT ZRP menurun sebesar 20.063%, 100 *node* EA-SHORT ZRP menurun sebesar 8.375%. Hasil PDR pada EA-SHORT ZRP dengan 50 *node* meningkat 0,545%, dan untuk EA-SHORT ZRP 100 *node* meningkat sebesar 21,301%.

Kata Kunci: MANET, *routing*, *Energy-aware*, ZRP, EA-SHORT

ABSTRACT

Mobile Ad-Hoc Network (MANET) is a wireless network from a collection of nodes that do not have a fixed router. Each node in the network acts as a router that is responsible for finding and requesting routes between nodes. Use the node as a router that requires higher energy. Meanwhile, the energy for the transmission process at the node is limited, and depletion of energy can cause interrupted routes. route interruptions can reduce network performance. In this study, the concept of energy aware uses the EA-SHORT algorithm which is applied to the Zone Routing Protocol (ZRP). EA-SHORT tries to add networks to all existing nodes by using variations in the amount of energy by selecting nodes that have enough energy that can help in connecting and avoiding nodes that have low energy. ZRP performance will be compared with EA-SHORT ZRP that has been approved with EA-SHORT which is returned from the specified parameter values. From the simulation, the results are shown, at node 50, throughput increases by 12.374%. For 100 nodes, an increase of 44.597%. At an average end to end delay, with 50 nodes, the EA-SHORT ZRP value decreased by 20,063%, 100 EA-SHORT ZRP nodes decreased by 8,375%. PDR results on EA-SHORT ZRP with 50 vertices increased by 0.545%, and for EA-SHORT ZRP 100 vertices increased by 21.301%.

Keyword: MANET, routing, Energy-aware, ZRP, EA-SHORT

BAB I

PENDAHULUAN

1.1 Latar Belakang

Mobile Ad-Hoc Network (MANET) adalah sebuah jaringan *wireless* dari sekumpulan *node* yang tidak memiliki *router* tetap. Setiap *node* dalam jaringan ini memiliki fungsi sebagai *router* yang bertanggung jawab untuk mencari dan menangani rute antara *node* di dalam jaringan[1]. MANET memiliki komponen penting dalam proses penemuan rute yang disebut dengan proses *routing*[2]. Dalam proses transmisi data dari *node* sumber menuju *node* tujuan dibutuhkan komunikasi dengan *mobile node* terdekatnya. Hal tersebut bertujuan agar paket dapat diterima oleh *node* tujuan dengan efisien[3]. Protokol *routing* terdiri atas tiga macam yakni proaktif, reaktif dan *hybrid*. Protokol proaktif secara berkala melakukan pemeriksaan rute di jaringan, sehingga saat paket akan dikirim, rute sudah diketahui dan dapat langsung digunakan. Protokol reaktif sebaliknya melakukan pemeriksaan rute hanya ketika ada paket yang akan dikirim[4]. Protokol *routing hybrid* merupakan protokol yang menggabungkan konsep kerja reaktif dan proaktif[5].

Komponen penting lainnya pada MANET adalah energi. Dimana, energi yang digunakan untuk proses transmisi pada setiap *node* yang beroperasi jumlahnya terbatas. Hal ini dikarenakan pada umumnya sumber energi *node* berasal dari baterai. Dengan jumlah energi yang terbatas, *node* juga mengoperasikan protokol *routing* dan berperan sebagai *router*, sehingga kinerja *node* menjadi meningkat dan membutuhkan energi yang lebih besar[6]. Pada *routing*, ketika rute optimal sudah ditemukan, maka rute akan disimpan dalam *routing table*, jika *routing* tidak menerapkan konsep *energy aware* maka setiap *node* yang berada dalam rute tersebut akan digunakan terus menerus dalam proses pengiriman paket, sedangkan *node-node* disekitarnya dibiarkan bebas. Hal tersebut mengakibatkan jumlah energi pada setiap *node* bervariasi[7].

Selain jumlah energi pada *node* yang bervariasi, dalam MANET, *node* yang bergerak dengan bebas dan acak dapat menyebabkan topologi jaringan berubah dan tidak dapat diprediksi posisinya, dan hal ini dapat dimanfaatkan dalam proses perubahan rute, sehingga terputusnya rute (*link failure*) pada jaringan yang disebabkan oleh *node* yang kehabisan energi dapat dicegah. Terputusnya rute akan menyebabkan menurunnya performa jaringan. Konsep *energy aware* salah satunya adalah EA-SHORT (*Energy Aware-Self Healing and Optimizing Routing Techniques*) yang mencoba

mendistribusikan beban jaringan (*load balancing*) pada semua *node* yang ada dengan memanfaatkan variasi jumlah energi. Keandalan suatu jaringan pada algoritma EA-SHORT diukur dengan meningkatnya performa jaringan yang dilihat dari parameter uji yang digunakan yakni, meningkatnya nilai *throughput*, *packet delivery ratio* dan menurunnya nilai *average end to end delay*.

Pada penelitian ini konsep *routing* dengan *energy aware* menggunakan algoritma EA-SHORT untuk MANET akan diterapkan dalam kerangka protokol *routing Zone Routing Protocol (ZRP)* yang merupakan protokol *hybrid*. Protokol ini memiliki konsep membatasi area *routing* dari protokol proaktif kedalam bentuk zona di setiap *node*. Sedangkan konsep protokol reaktif digunakan jika pengiriman paket berada diluar zona[8]. Pada ZRP, rute yang digunakan bisa saja tidak *reliable*, artinya setiap *node* yang berpartisipasi dalam rute bisa saja kehabisan energi pada saat sedang dilakukan pengiriman paket.

Kinerja protokol *routing* yang diukur adalah jumlah *throughput*, *average end to end delay*, dan *packet delivery ratio* (PDR), dengan menerapkan skenario yang berbeda yakni memvariasikan beberapa parameter simulasi. Simulasi protokol *routing* pada penelitian ini akan dilakukan dengan menggunakan *tools network simulator 2 (NS2)* versi 2.35, sedangkan untuk analisis terhadap parameter uji menggunakan *AWK Script* yang mempunyai kegunaan untuk mengurai file.tr yang terdapat pada NS2. Hasil simulasi dari kedua protokol akan analisa dan dibandingkan untuk mengetahui pengaruh penerapan EA-SHORT pada protokol ZRP.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka dapat dirumuskan masalah dalam tugas akhir ini yaitu,

1. Bagaimana menemukan rute yang handal berbasis energi dengan menerapkan algoritma EA-SHORT pada kerangka protokol *routing* ZRP?
2. Bagaimana performansi protokol *routing* ZRP di jaringan MANET setelah diterapkan algoritma EA-SHORT?

1.3 Batasan Masalah

Untuk membatasi cakupan permasalahan yang akan dibahas serta tujuan dan sasaran penelitian dapat tercapai sesuai dengan rumusan masalah, maka diberikan beberapa batasan masalah yaitu:

1. Simulasi menggunakan NS-2 versi 2.35 (NS-2.35).
2. Menggunakan algoritma EA-SHORT dalam kerangka protokol *routing* ZRP.
3. Model mobilitas yang digunakan adalah *Random Waypoint*.
4. Parameter uji kinerja adalah *throughput*, *average end to end delay*, dan *packet delivery ratio*.
5. Tidak membahas keamanan jaringan.

1.4 Tujuan

Adapun tujuan dari penelitian ini adalah menerapkan algoritma EA-SHORT pada kerangka protokol *routing* ZRP untuk pencarian rute yang handal berbasis energi pada jaringan MANET.

1.5 Manfaat

Adapun manfaat yang dapat diperoleh dari tugas akhir ini adalah diharapkan dapat menemukan rute yang handal berbasis energi pada protokol *routing* ZRP menggunakan algoritma EA-SHORT.

1.6 Sistematika Penulisan Penelitian

Sistematika penulisan Tugas Akhir ini dibagi menjadi beberapa bagian, yaitu

1. BAB I. PENDAHULUAN

Bab ini berisi latar belakang penulisan tugas akhir, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

2. BAB II. TINJAUAN PUSTAKA dan DASAR TEORI

Bab ini menjelaskan mengenai tinjauan pustaka yang berkaitan dengan penelitian yang dilakukan sebelumnya dan menjelaskan penelitian yang diusulkan pada tugas akhir, serta berisi uraian mengenai dasar-dasar teori mengenai segala sesuatu yang dibutuhkan dalam penelitian, baik berupa bidang penelitian yang dilakukan, metode atau konsep yang digunakan, dan program yang diperlukan selama penelitian dilakukan.

3. BAB III. METODE PENELITIAN

Bab ini berisi perencanaan simulasi jaringan, membahas langkah-langkah yang dilakukan dalam penelitian, konsep penelitian yang digunakan, dan jadwal penelitian yang akan dilakukan.

4. BAB IV. HASIL DAN PEMBAHASAN

Bab ini berisi pelaksanaan simulasi dan analisis hasil simulasi dari skenario penelitian yang telah dilakukan.

5. **BAB V. KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan yang didapat dari analisis hasil pengujian skenario penelitian yang telah dilakukan dan saran yang diberikan sebagai masukan untuk digunakan pada penelitian selanjutnya

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka pada Tugas Akhir ini berisi penjelasan (gambaran) mengenai penelitian terkait yang telah dilakukan sebelumnya dan penelitian usulan yang akan dilakukan.

2.2 Penelitian Terkait

Penelitian yang telah dilakukan sebelumnya yang berkaitan dengan penelitian yang akan dilakukan antara lain yakni penelitian yang melakukan simulasi dan analisa pada protokol ZRP menggunakan simulator NS2 dengan melakukan perhitungan nilai Qos yakni *throughput*, *delay* dan PDR[5]. Jumlah *node* yang disimulasikan adalah, 10, 20, 50 dan 100. Dari hasil simulasi yang dilakukan, nilai *throughput* terendah terdapat pada 100 *node* dengan 34.5702 Kbps dan tertinggi ditemukan di jaringan dengan jumlah 10 *node*, sebesar 727.081 Kbps. Sementara rata-rata *delay* terendah dalam jaringan dengan 10 *node* sama dengan 0,606597 s dan yang tertinggi di jaringan dengan 100 *node*, dengan rata-rata *delay* 2,00939 s. Untuk nilai hasil PDR dari simulasi yang dilakukan, nilai PDR terendah terdapat pada jaringan dengan 100 *node*, yaitu sebesar 71,6509%, dan nilai PDR tertinggi terdapat pada jaringan dengan jumlah 10 *node*, yaitu sebesar 80,439%. Dari hasil pengujian dapat diketahui jika semakin banyak jumlah *node* maka lalu lintas jaringan akan semakin padat dan menurunkan kualitas pelayanan. Persamaan penelitian ini dengan penelitian yang akan dilakukan, yakni menggunakan protokol *routing* ZRP. Perbedaannya terletak penambahan metode EA-SHORT yang akan dilakukan sedangkan penelitian sebelumnya tidak menggunakan metode.

Penelitian yang melakukan analisis perbandingan kinerja antara tiga protokol yakni AOMDV, DSDV dan ZRP di jaringan MANET [3]. Ketiga protokol tersebut disimulasikan dengan NS2 di lingkungan simulasi yang memiliki beberapa variasi luas area dan jumlah *mobile node*. Kinerja ketiga protokol tersebut diuji dan dianalisis kinerjanya berdasarkan beberapa parameter, yaitu *throughput*, *average end to end delay*, PDR dan *normalized routing load*. Jumlah *node* yang digunakan yakni 20, 40, 80 dan 100. Untuk luas area simulasi yang digunakan yakni 500 x 500, 800 x 800 dan 1000 x 1000. Hasil pengujian untuk luas area 1000 x 1000, dimana ZRP memiliki nilai rata-rata PDR tertinggi sebesar 0.784%, 0.834%, 0.165%, dan 0.143%. ZRP mengalami penurunan

nilai PDR yang sangat drastis ketika *node* berjumlah 80 dan 100 walaupun sempat memiliki rasio tertinggi pada kepadatan *node* 40. ZRP memiliki nilai rata-rata *throughput* tertinggi, yaitu sebesar 79.985 bps, 83.420 bps, 16.573 kbps, dan 14.375 bps dan penurunan yang sangat drastis masih terjadi pada ZRP ketika *node* berjumlah 80 dan 100. *verage end to end delay* juga mengalami peningkatan cukup jauh dari 0,278 s dan 0,310 s pada jumlah *node* 20 dan 40 menjadi 6,553 dan 11,556 s pada jumlah *node* 80 dan 100. Rata-rata untuk *normalized routing load* ZRP yakni 8.609 bytes, 23.528 bytes, 328.435 bytes, 955.22 bytes. Penelitian ini menunjukkan bahwa performa ZRP mengalami penurunan seiring dengan bertambahnya jumlah *node* dan mengalami penurunan yang drastis antara jumlah *node* 24, 40 dengan 80, 100. Persamaan dengan penelitian ini dengan penelitian yang akan dilakukan yaitu protokol ZRP yang digunakan dan luas area 1000 m x 1000 m. Perbedaannya dengan penelitian yang akan dilakukan yakni pada penelitian ini tidak menerapkan metode apapun, sedangkan penelitian yang akan dilakukan akan diterapkan metode EA-SHORT untuk meningkatkan performa protokol ZRP.

Penelitian yang melakukan analisis Qos pada jaringan Ad Hoc dengan membandingkan protokol *Ad hoc on Demand Distance Vector* (AODV) dan ZRP dengan parameter uji yakni *end to end delay*, *jitter*, PDR dan *throughput*[9]. Hasil pengujian dengan menggunakan luas area simulasi 1000 m x 1000 m yakni protokol AODV dapat melakukan pengiriman paket lebih cepat dengan kecepatan rata-rata 0.3134 s dari *routing* protokol ZRP dengan nilai 0.3891 s, namun untuk jumlah paket yang terkirim *routing* protokol ZRP lebih baik dengan rata-rata *throughput* 963.34 bps dibandingkan *routing* protokol AODV. Persamaannya dengan penelitian yang akan dilakukan adalah sama sama menggunakan protokol ZRP dalam simulasi nya, dan cakupan luas area simulasi. Sedangkan perbedaannya adalah tidak ada penambahan metode *energy aware* pada penelitian sebelumnya.

Pada penelitian ini telah dilakukan analisis perbandingan kinerja antara protokol *routing* TORA dan TORA yang menerapkan algoritma SHORT[10]. Hasil simulasi menunjukkan bahwa *throughput* yang lebih tinggi meningkat hingga 3-50%, *end to end delay* yang lebih rendah hingga 50%, *delay* mengalami penurunan sebesar 50%, penurunan *packet loss* bervariasi dari 12-57%, perbaikan energi sekitar 43%, dan *life time* jaringan yang lebih tinggi. TORA yang dimodifikasi mengungguli protokol *routing* TORA dalam hal *throughput*, *packet loss*, *end-to-end delay*, dan energi. Persamaan dari penelitian terkait dengan penelitian yang dilakukan yaitu menggunakan algoritma EA-

SHORT. Sedangkan perbedaannya yaitu pada penelitian yang dilakukan menggunakan protokol *routing* ZRP namun penelitian sebelumnya menggunakan protokol *routing* TORA.

Penelitian yang melakukan analisis konsumsi energi pada protokol AODV, OLSR dan ZRP[11]. Penelitian ini melakukan analisis dengan parameter yakni *routing power*, *throughput*, konsumsi energi pada saat proses pengiriman, konsumsi energi pada saat penerimaan, dan *end to end delay*. Simulasi dilakukan selama 30 s dengan luas area 1000m x 1000m dengan variasi jumlah *node* yakni 10, 20, 30, 40 dan 50. Hasil simulasi menunjukkan *throughput* tertinggi pada AODV dan OLSR dan terendah pada protokol ZRP. Pada nilai *end to end delay*, protokol terendah yakni OLSR. PDR tertinggi adalah AODV diikuti OLSR lalu ZRP. Total konsumsi energi saat pengiriman sangat rendah pada OLSR dibandingkan dengan dua protokol lainnya. Konsumsi energi terendah pada saat penerimaan yakni protokol ZRP. *Routing power* pada AODV adalah yang tertinggi namun mengalami penurunan jika dilihat dari rata-rata *routing power* AODV dalam setiap simulasi. Persamaan penelitian ini dengan penelitian yang akan dilakukan yakni sama-sama menggunakan protokol ZRP, perbedaannya yakni pada penelitian sebelumnya hanya melakukan analisa konsumsi energi sedangkan pada penelitian yang akan dilakukan, protokol ZRP akan diterapkan metode Algoritma EA-SHORT dalam pencarian rute yang handal berbasis energi.

Penelitian tentang *energy aware* dengan menggunakan algoritma EA-SHORT yang bertujuan untuk menyeimbangkan konsumsi energi semua *node* pada jaringan[7]. Penelitian ini melakukan simulasi menggunakan protokol AODV dengan menggunakan 120 *node*, luas area simulasi yakni 1200m x 1200 m. Hasil uji coba dengan algoritma ini kemudian dibandingkan dengan AODV tanpa menggunakan EA-SHORT. Perbandingannya yakni jangka waktu atau lama hidup jaringan. Pada simulasi di s 200 kapasitas kedua protokol masih relatif sama. Namun pada 1210 s protokol AODV mulai mengalami interupsi karena partisi pada jaringan. Aliran data pada AODV terputus pada s 1825, namun bertahan sampai akhir simulasi (2000 s) menggunakan EA-SHORT. Persamaannya dengan penelitian yang akan dilakukan adalah menggunakan algoritma EA-SHORT, perbedaannya adalah penelitian selanjutnya akan menerapkan algoritma EA-SHORT pada kerangka protokol ZRP.

2.3 Penelitian yang Diusulkan

Pada penelitian terkait sebelumnya telah dijelaskan keunggulan protokol ZRP dan kelemahannya tanpa menggunakan metode, serta keunggulan EA-SHORT pada beberapa jenis *routing* di jaringan MANET, sehingga penelitian ini akan melakukan modifikasi pada kerangka protokol *routing* ZRP untuk pencarian rute yang mempertimbangkan jumlah energi pada *node* menggunakan algoritma EA-SHORT. Pencarian rute dengan EA-SHORT dilakukan dengan cara mengganti *node* dalam rute yang dilakukan oleh salah satu *node* di sekitarnya, ketika energi *node* dalam rute tersebut teridentifikasi lebih rendah maka dilakukan pergantian. Sehingga, dengan diterapkannya algoritma EA-SHORT ini akan meningkatkan performansi jaringan. Hasilnya akan dibandingkan dengan ZRP tanpa modifikasi. Parameter yang akan digunakan yakni *throughput*, *average end to end delay*, dan PDR.

2.4 Dasar Teori

2.4.1 Mobile Ad Hoc (MANET)

MANET merupakan jaringan nirkabel yang memiliki kemampuan dalam menciptakan konfigurasi sendiri (*self-configuring*). Jaringan MANET terdiri dari sekumpulan perangkat bergerak (*mobile node*) seperti, laptop, *smartphone*, PDA, dan tablet yang saling bertukar informasi secara nirkabel seperti yang ditunjukkan pada tiap *mobile node* dalam jaringan MANET juga dapat berperan sebagai *mobile router*, di mana tiap *mobile node* dapat meneruskan paket dari satu *mobile node* menuju *mobile node* lainnya secara nirkabel tanpa harus terhubung dengan perangkat *router* maupun *access point*[12].

Kelebihan jaringan MANET antara lain[13]:

1. Instalasi cepat dengan intervensi pengguna seminimal mungkin.
2. Perencanaan rinci dan pemasangan *Base Transceiver Station* (BTS) tidak diperlukan.
3. Jaringan *ad hoc* dapat dikaitkan ke Internet, dengan demikian menggabungkan banyak perangkat dan memungkinkan pengguna lain untuk menggunakan layanan yang tersedia.
4. Kapasitas, jangkauan, dan energi mendorong penggunaannya bersamaan dengan infrastruktur seluler yang ada dan dapat memperluas jangkauan dan interkonektivitasnya.

5. MANET juga disiapkan untuk menggunakan arsitektur 4G bertujuan untuk menyediakan komputer di mana-mana, lingkungan yang mendukung pengguna dalam menyelesaikan tugas, mengakses informasi dan berkomunikasi di mana saja, kapan saja dan dari perangkat apa pun.

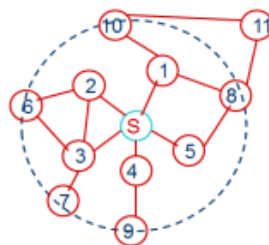
2.4.2 Protokol *Routing* pada MANET

MANET memiliki komponen penting di dalam proses penentuan rute yang optimal atau sering disebut dengan proses *routing*. Dalam proses pengiriman paket dari *source node* menuju *destination node* dibutuhkan beberapa komunikasi dengan *mobile node* terdekatnya. Hal tersebut bertujuan agar paket dapat diterima oleh *destination node* dengan efisien. Pada jaringan MANET protokol *routing* umumnya dibedakan menjadi tiga jenis [2], yaitu:

1. Protokol *routing* reaktif (*on demand*).
2. Protokol *routing* proaktif (*table driven*).
3. Protokol *routing hybrid*, menggabungkan kinerja protokol reaktif dan proaktif.

2.4.3 Zone Routing Protocol (ZRP)

Zone Routing Protocol (ZRP) merupakan *routing* protokol *hybrid* yang menggabungkan sifat antara *routing* protokol proaktif dan *routing* protokol reaktif untuk mengurangi *overhead* dari *routing* protokol proaktif dan mengurangi *latency* dari *routing* protokol reaktif ketika pencarian rute. Kekurangan protokol *routing* ZRP adalah rute yang ditemukan bisa saja tidak *reliable*, *node-node* yang berpartisipasi dalam rute tersebut bisa saja mati karena kehabisan energi ketika proses pengiriman paket berlangsung. Hal ini disebabkan karena protokol *routing* ZRP tidak melakukan proses penghitungan energi tiap *node* untuk rute tersebut[14].



Gambar 2.1 Protokol *routing* ZRP. [15]

Contoh zona *routing* untuk *node* S dengan radius zona 2 adalah ditunjukkan pada Gambar 2.1 [14]. *Node* dari 1 hingga 10 milik zona perutean S, kecuali *node* 11. Simpul 6 hingga 10 disebut *peripheral node* karena jarak *hop* dari S sama dengan nilai radius zona *routing*. Informasi tentang *node* tetangga diperlukan untuk membangun zona *routing* yang

diberikan *node*. Tetangga didefinisikan sebagai *node* yang dapat digunakan untuk mengarahkan komunikasi langsung.

ZRP terbagi dalam empat sub-protokol, yaitu:

1. *Intrazone Routing Protocol (IARP)*

IARP merupakan bagian dari protokol proaktif dengan jangkauan yang terbatas. Jangkauan IARP ditentukan oleh parameter *hop* yang membatasi zona *routing*. Untuk mendeteksi *node* tetangga dan *route* yang rusak, IARP dibantu oleh *Neighbour Discovery Protocol*. IARP akan menyediakan pembaruan *route* pada zona lokal ketika topologi dari zona *routing* berubah. Ketika ada *source node* yang ingin mengirimkan paket menuju sebuah *node* yang berada di dalam zona *routing source node*, *route* dari *source node* menuju *destination node* akan tersedia dengan cepat. Apabila ada jalur yang rusak, maka IARP akan memperbarui *route* pada zona lokal dan mencarikan jalur yang baru dengan mengirimkan *hello message* ke *node* tetangga yang sudah berbeda dari sebelumnya [16].

2. *Neighbour Discovery Protocol (NDP)*

NDP memiliki peranan yang sangat penting pada proses *routing* di ZRP. NDP merupakan sumber informasi dari keberadaan *node-node* yang berada pada zona IARP. NDP melakukan pendataan *node* tetangga dengan melakukan *broadcast beacons message* yang disebar ketetangga yang terkoneksi, *beacon message* yang dikirim berisikan MAC Address, IP Address, Destination, Source, apabila *broadcast beacons message* mendapatkan jawaban maka jawaban tersebut dijadikan indikator bahwa *node* tersebut merupakan tetangga dan akan disimpan pada tabel *neighbor* yang dimiliki NDP. *Node* dianggap tidak valid apabila *node* tersebut tidak mengirimkan balasan selama batasan waktu yang dialokasikan, dan tabel tersebutlah yang akan menjadi acuan bagi IARP untuk melakukan proses *routing* [15].

3. *Interzone Routing Protocol (IERP)*

IERP merupakan bagian dari *reactive routing*. Ketika *source node* ingin mengirimkan paket ke *node* yang berada di luar zona lokal, *source node* membutuhkan *route* yang belum tersedia, IERP akan membantu *source node* untuk mencari *route* baru. Dalam proses pencarian *route* baru, IERP dibantu oleh *Bordercast Resolution Protocol* yang bertugas melakukan *bordercast route request* yang dibentuk oleh *source node*. Ketika *border node* menerima RRP yang berasal dari *source node*, *border node* akan memeriksa *node* tujuan pada zona *routing* lokal. Jika *node* tujuan berada pada zona

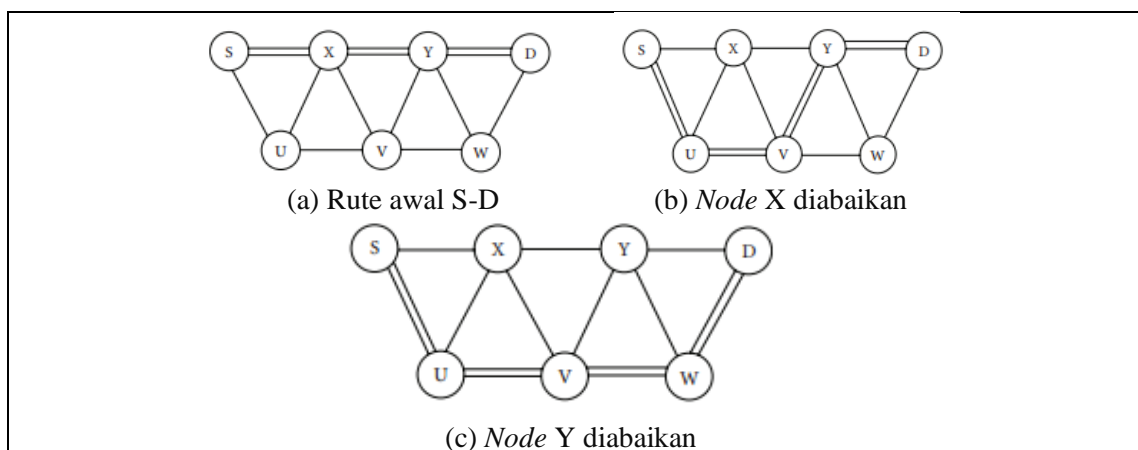
routing lokal, maka akan dikirim *route reply* ke *node* sumber, setelah menambahkan alamatnya pada paket tersebut, serta menambahkan *route* dari sumber menuju tujuan[16].

4. Bordercast Resolution Protocol (BRP)

BRP merupakan layanan pengiriman paket untuk mengarahkan *route reply* dari IERP menuju ke *border node*. Proses mengarahkan *route reply* menuju *border node* disebut *bordercast*. BRP bertanggung jawab untuk meneruskan *route reply* yang dibentuk IERP ke *border node*. Walaupun penerima dari *route reply* adalah *border node*, BRP tetap mengirim *route reply* pada setiap *hop*. BRP akan melacak *node* yang sudah tercakup oleh *route reply*. Ketika *node* menerima *route reply*, *node* tersebut menandai *node* tetangga dari *bordercast* sebagai *node* yang sudah tercakup oleh *route reply*. Jika *node* penerima adalah *border node* dari *node* yang melakukan *bordercast*, maka *border node* akan menjadi *node* yang akan melakukan *bordercast* yang baru dan *node* tetangganya akan tercakup oleh *route reply*. BRP menggunakan tabel *routing* IARP dari *node* yang melakukan *bordercast*[16].

2.4.4 Mekanisme EA-SHORT

EA-SHORT meningkatkan performa jaringan dengan proses pengiriman paket melalui *node* yang memiliki energi yang cukup dan mencegah menggunakan *node* yang memiliki energi yang lemah[10]. Protokol *routing* menggunakan mekanisme *self-healing* sehingga bisa mengatur dirinya pada situasi ketika terdapat *node* yang terlalu sering digunakan saat *node* yang lain bebas. EA-SHORT dengan merata akan mendistribusikan *traffic load* diantara semua *node* yang ada pada jaringan MANET. Jalur S-X-Y-D pada Gambar 2.3 adalah rute optimal dari *node* S ke D. *Node* X dan Y akan terus menerus digunakan dalam meneruskan paket, sedangkan *node* yang lain tidak digunakan. *Node* X dan Y akan kehabisan energi dan menyebabkan terjadinya *link failure*.



Gambar 2.2 Proses pencarian rute dengan algoritma EA-SHORT.[10]

Algoritma EA-SHORT akan mencoba membagi *traffic load* jaringan pada *node* yang lain. Ketika paket data berhasil diteruskan oleh dua *node* yakni X dan Y, *node* U juga mengetahuinya (*overhear*) paket serta perbedaan jumlah energi pada *node* X dan dirinya dengan bantuan tabel *overhear*. Jika perbedaannya cukup signifikan, *node* U akan melakukan perubahan rute. Hal yang sama juga berlaku pada perbedaan jumlah energi antara *node* Y dan V, jika perbedaannya cukup jauh, maka V akan melakukan perubahan rute. Dengan perubahan rute yang dilakukan, rute akan berganti pada *node alternative* rute *disjoint* yakni S-U-V-W-D seperti yang ditunjukkan pada Gambar 2.2, dilakukan pemeliharaan *Overhear table* pada setiap *node*.

2.4.5 NS 2

Network Simulator NS-2 adalah suatu interpreter yang *object-oriented*, dan *discrete event-driven* yang dikembangkan oleh University of California Berkeley dan USC ISI sebagai bagian dari proyek *Virtual Internet Testbed* (VINT). NS menjadi salah satu *tool* yang sangat berguna untuk menunjukkan simulasi jaringan melibatkan *Local Area Network* (LAN), *Wide Area Network* (WAN), tapi fungsi dari *tool* ini telah berkembang selama beberapa tahun belakangan ini untuk memasukkan didalamnya jaringan nirkabel (*wireless*) dan juga jaringan *ad hoc*. *Network Simulator* pertama kali dibangun sebagai varian dari *REAL Network Simulator* pada tahun 1989 di University of California Berkeley (UCB). Pada tahun 1995 pembangunan *Network Simulator* didukung oleh *Defense Advanced Research Project Agency* (DARPA) melalui *VINT Project*, yaitu sebuah tim riset gabungan yang beranggotakan tenaga ahli dari beberapa instansi ternama. Ada beberapa keuntungan menggunakan NS sebagai perangkat lunak simulasi pembantu analisis dalam riset, antara lain adalah NS dilengkapi dengan *tool* validasi yang digunakan untuk menguji kebenaran pemodelan yang ada pada NS. Secara *default*, semua pemodelan NS akan dapat melewati proses validasi ini. Pemodelan media, protokol dan komponen jaringan yang lengkap dengan perilaku trafiknya sudah disediakan pada *library* NS. NS juga bersifat *open source* dibawah *Gnu Public License* (GPL), sehingga NS dapat di-*download* dan digunakan secara gratis melalui *web site* NS yaitu <http://www.isi.edu/nsnam/>. Sifat *open source* juga mengakibatkan pengembangan NS menjadi lebih dinamis[17].

BAB III

METODE PENELITIAN

3.1 Diagram Alir Penelitian

Gambar 3.1 merupakan diagram alir yang menggambarkan langkah-langkah yang akan dilakukan pada penelitian.



Gambar 3.1 Diagram alir penelitian

3.1.1 Studi Literatur

Pada tahap studi literatur, dilakukan pengkajian berbagai penelitian yang telah dilakukan sebelumnya terkait dengan penelitian yang akan dilakukan. Pada tahap ini, dilakukan pengkajian terhadap berbagai jurnal maupun paper mengenai performa protokol *routing* ZRP, teknik untuk memodifikasi *routing protocol*, teknik yang berkaitan dengan konsep *energy aware* untuk meningkatkan kehandalan jaringan berbasis energi *mobile ad hoc network*, dan beberapa penelitian lainnya yang mendukung penelitian yang akan dilakukan. Data yang didapat dari studi literatur akan digunakan untuk menentukan *tools* simulasi yang akan digunakan dan skenario penelitian yang akan dijalankan

3.1.2 Persiapan Perangkat *Hardware* dan *Software*

Uji coba dilakukan dengan menggunakan PC dengan spesifikasi perangkat yang dapat dilihat pada Tabel 3.1

Tabel 3.1 Komponen *hardware* dan *software*

Komponen	Spesifikasi yang dibutuhkan
CPU	Intel(R) Core(TM) i3-2310M CPU @ 2.10 GHz atau spesifikasi yang lebih tinggi
Sistem Operasi	Linux Ubuntu
Memori	Minimal 2 GB
Harddisk	Minimal Alokasi memori 10 GB untuk penyimpanan data simulasi

Adapun perangkat lunak yang digunakan pada penelitian ini adalah sebagai berikut:

1. *Network Simulator 2 (NS2) 2.35* untuk melakukan simulasi jaringan MANET. Simulasi ini berkaitan dengan proses pengambilan data hasil simulasi untuk data dilakukan analisis terhadap performansi protokol yang digunakan. versi ns2 yang digunakan dapat menyesuaikan.
2. *Virtual Box* sebagai tempat instalasi linux. NS2 akan diinstal pada sistem operasi linux yang akan diinstal secara *virtual* pada *virtual box*, atau linux dapat di-*install* langsung pada perangkat PC.
3. *Software* untuk membuat grafik hasil simulasi.
4. *Linux Ubuntu 16.04 LTS 64 bit* sebagai sistem operasi atau versi lain yang dapat menyesuaikan.
5. *Software* sebagai pembuatan media membuat dokumentasi atau laporan.
6. *Software* yang dapat digunakan sebagai media pembuatan diagram.

3.1.3 Pembuatan *Script* EA-SHORT

Pembuatan skrip EA-SHORT ZRP dilakukan dengan cara memodifikasi kerangka protokol *routing* ZRP dengan menyisipkan Algoritma EA-SHORT. Penulisan kode dilakukan untuk pencarian rute yang handal berbasis energi dengan mekanisme perubahan rute. Jika rute baru ditemukan maka rute tersebut yang akan digunakan untuk melakukan pengiriman data.

3.1.4 Perancangan Skenario Simulasi

Pada bagian ini, peneliti melakukan perancangan skenario simulasi jaringan MANET untuk melakukan analisis pengaruh algoritma EA-SHORT pada protokol *routing* ZRP dengan kondisi yaitu protokol ZRP belum menggunakan EA-SHORT dan kondisi dimana ZRP sudah dimodifikasi menggunakan EA-SHORT. Adapun topologi jaringan yang digunakan dengan jenis protokol *routing* ZRP dengan jenis trafik menggunakan CBR (*Constant Bit Rate*) serta menggunakan mobilitas *node Random Waypoint*. Terdapat beberapa parameter uji yang digunakan untuk menganalisa hasil pengujian dari skenario simulasi yang dilakukan, diantaranya adalah:

1. *Throughput*

Throughput adalah kecepatan transfer data efektif yang merupakan jumlah total kedatangan paket data yang sukses diamati pada tujuan selama interval waktu tertentu dibagi dengan durasi interval waktu pengamatan.

Rumus:

$$\text{Throughput (bps)} = \frac{\text{ukuran data yang dikirim}}{\text{total waktu pengiriman data}} \quad (3-1)$$

2. *Average end to end delay*

Waktu jeda yang dibutuhkan dalam pengiriman paket dari pengirim ke penerima paket.

Rumus:

$$\text{Average delay (s)} = \frac{\text{total delay}}{\text{total paket yang diterima}} \quad (3-2)$$

3. *Packet delivery ratio* (PDR)

Perbandingan antara paket data yang terkirim dengan jumlah paket data yang diterima.

Rumus:

$$\text{Packet Delivery Ratio (\%)} = \frac{\text{paket data yang diterima}}{\text{paket data yang dikirim}} \times 100\% \quad (3-3)$$

3.1.5 Menjalankan Skenario simulasi

Pada tahap ini skenario simulasi yang telah dibuat pada file .tcl dijalankan dengan perintah `#ns namafilename.tcl`. Jika tidak ada kesalahan dalam penulisan, maka akan ada 3 jenis *output* yang bisa dilihat saat simulasi dijalankan, yaitu *output* pada editor CLI yang dikeluarkan perintah “puts”, *output* pada `projzrp10.tr` yang berisi data-data hasil simulasi dan *output* pada `zrp2.nam` yang merupakan animasi simulasi.

3.1.6 Analisis Hasil Simulasi

Pada tahap ini dilakukan proses analisis terhadap hasil yang diperoleh dari proses pengujian skenario penelitian yang dilakukan, dengan membandingkan hasil simulasi protokol *routing* ZRP dan protokol *routing* EA-SHORT ZRP, dengan percobaan pengiriman data dengan mobilitas *node* berdasarkan jumlah *node*, luas area, dan kecepatan *node*. Menggunakan *script* AWK, akan dilakukan analisis parameter berupa *throughput*, PDR, dan *end to end delay*. Hasil Analisa simulasi ditampilkan dalam bentuk grafik. Semakin besar nilai *throughput* dan PDR maka semakin banyak jumlah paket data yang dapat sampai ke tujuan dan semakin baik performa. Semakin rendah nilai *end to end delay* maka semakin cepat kemampuan protokol *routing* dalam menemukan jalur komunikasi data, sehingga semakin cepat dan semakin banyak data yang dapat sampai ke tujuan. Hasil pengujian yang dijalankan diharapkan dapat menunjukkan pengaruh penerapan algoritma EA-SHORT terhadap protokol *routing* ZRP pada lingkungan MANET.

3.1.7 Membuat Grafik Hasil

Pembuatan grafik Hasil dari simulasi yang telah dilakukan menggunakan Microsoft Excel. Grafik atau data-data tersebut kemudian dianalisis dan dapat digunakan untuk membandingkan performansi ZRP dengan EA-SHORT ZRP.

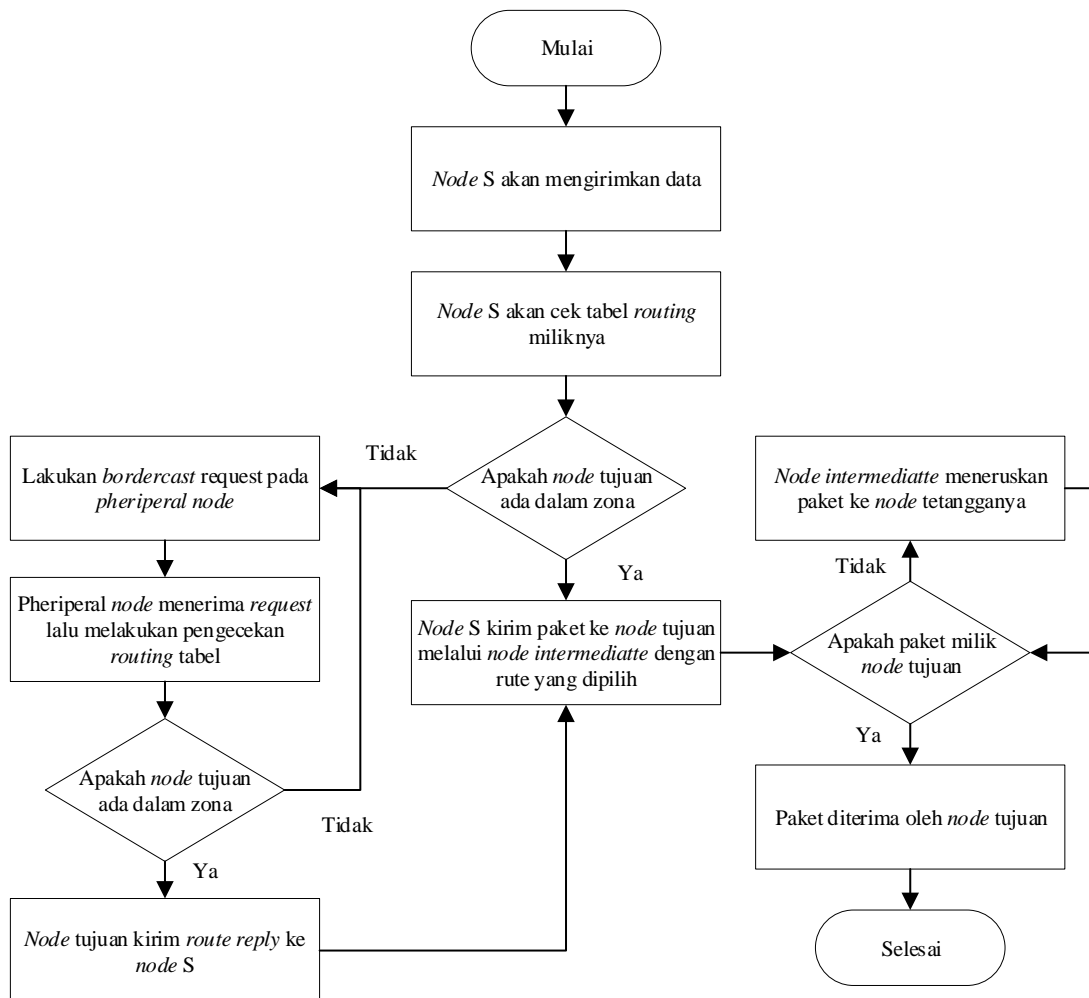
3.1.8 Membuat Kesimpulan

Pada tahap ini akan dilakukan *review* terhadap hasil modifikasi protokol *routing* ZRP dan pengujian skenario, untuk mengetahui performa terbaik dari skenario yang telah diuji. Sehingga, dapat ditarik kesimpulan dari penelitian yang dilakukan serta saran sebagai masukan untuk penelitian selanjutnya.

3.1.9 Membuat Laporan Dokumentasi

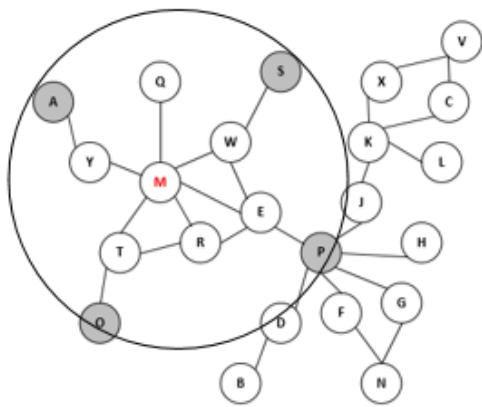
Pada tahap ini akan dilakukan pembuatan laporan dokumentasi terhadap penelitian yang telah dilakukan. Dokumentasi laporan yang dibuat dapat membantu untuk penelitian selanjutnya yang berkaitan dengan penelitian yang dilakukan.

3.2 Algoritma Pencarian Rute Menggunakan Protokol *Routing ZRP*

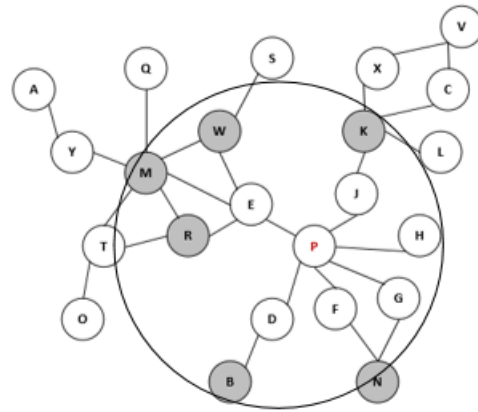


Gambar 3.2 Proses pencarian rute pada protokol *routing ZRP*

Dimisalkan jaringan pada Gambar 3.3, *node M* ingin mengirim paket menuju *node V*. dengan radius zona adalah kurang atau sama dengan 2. *Node* akan menggunakan *routing table* yang disediakan oleh IARP untuk mengecek apakah *node* tujuan berada dalam zona atau tidak. Karena tujuan tidak ditemukan, maka dilakukan *route request* menggunakan IERP. Pesan *request* diteruskan ke *peripheral node* yakni A,O,P,S. kemudian setiap *node* ini juga akan mengecek tujuan paket pada tabel *routing*-nya.

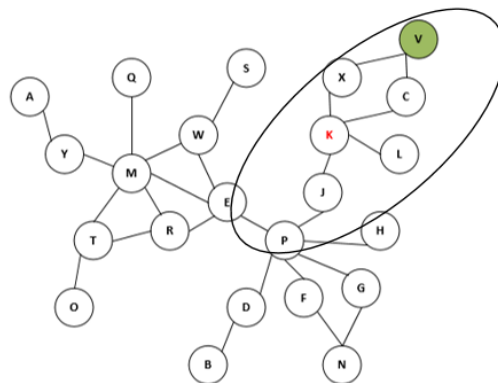


Gambar 3.3 Routing zone pada node M.



Gambar 3.4 Routing zone pada node P.

Node P tidak menemukan tujuan paket pada *routing table*-nya, lalu pesan akan diteruskan ke *pheriperal node* yakni *node* yang ditandai dengan warna abu-abu, karena adanya mekanisme *query control*, maka *request* tidak diteruskan ke *node* W,M,R.

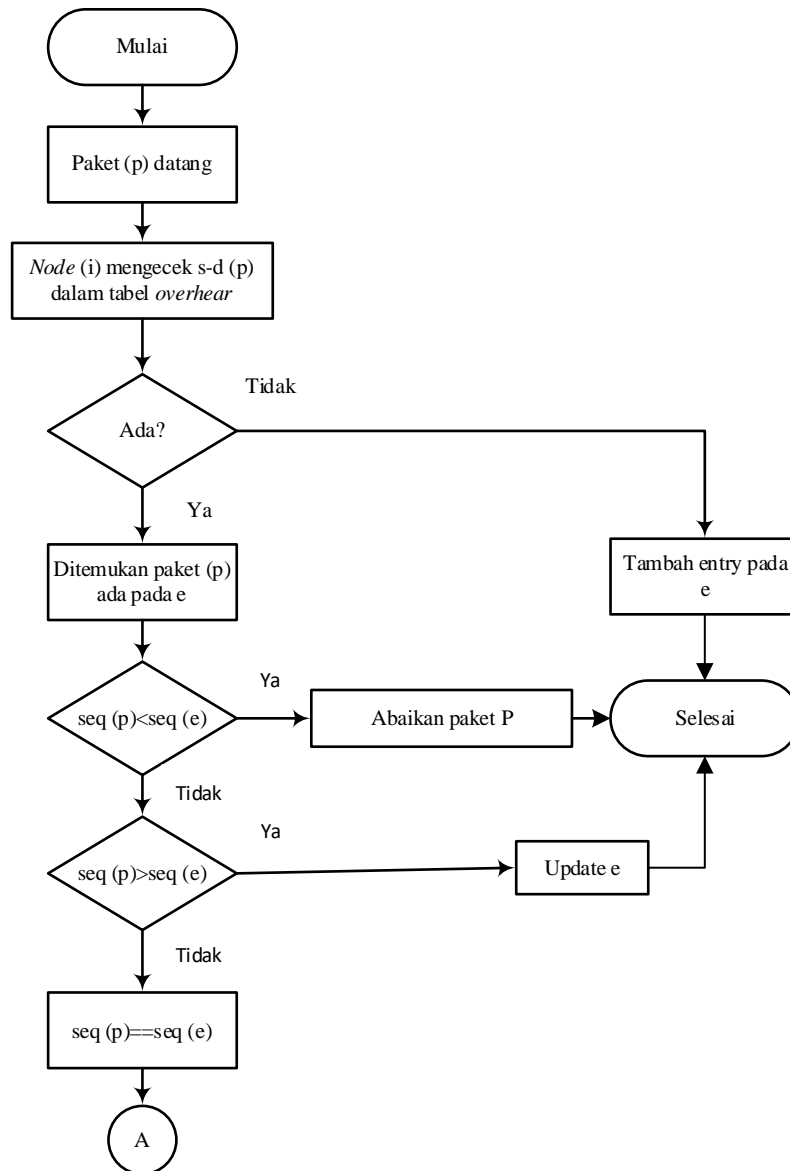


Gambar 3.5 Routing zone pada node K.

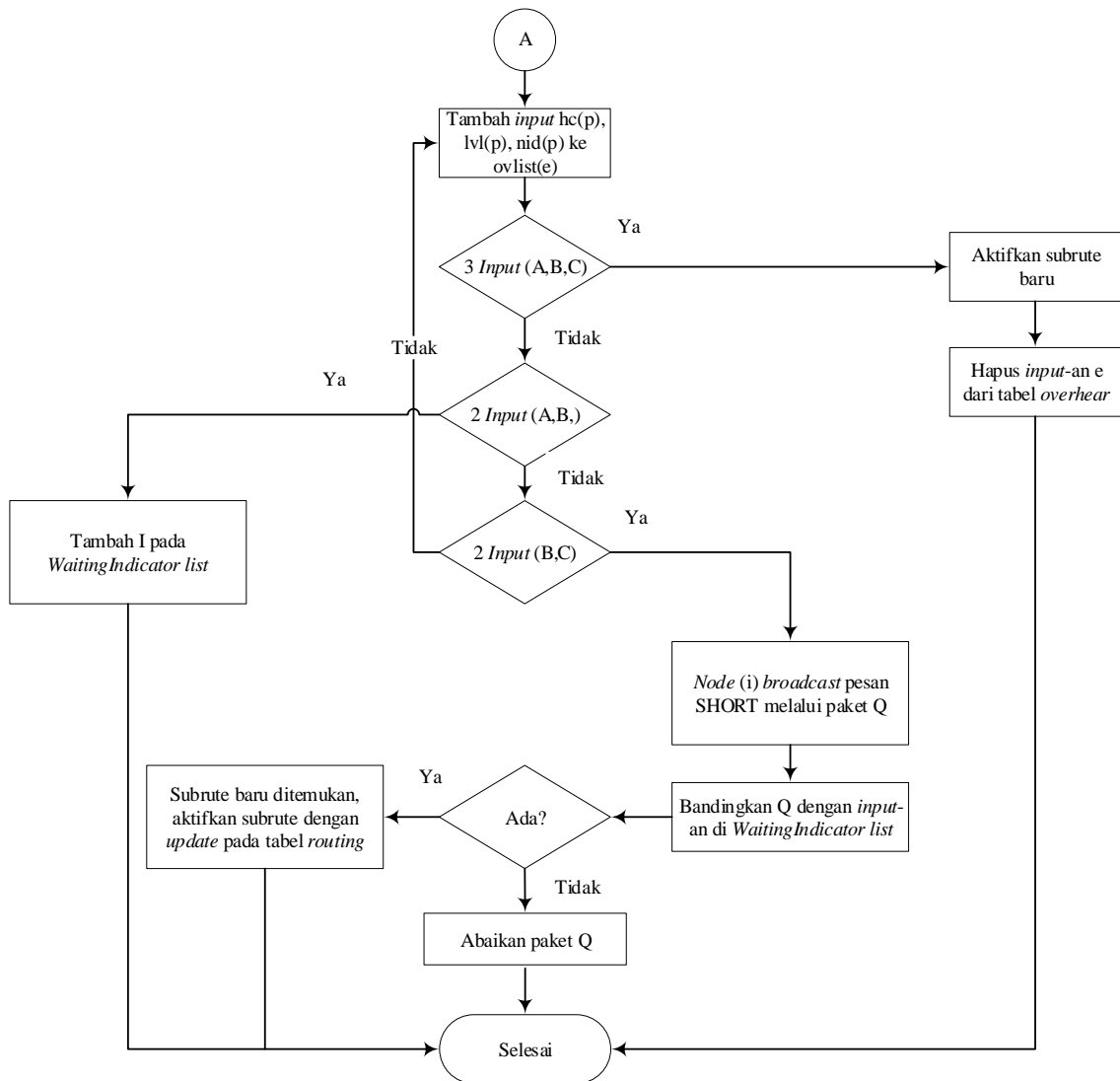
Ketika *route request* diterima oleh *pheriperal node* dari P, dan diterima oleh *node* K, dimana *node* K menemukan tujuan dari *route request* yang ditunjukkan dengan warna hijau pada Gambar 3.5 yakni *node* V yang berada dalam zona *node* K. *node* K akan menambahkan jalur menuju *node* pada jalur di *route request*. Pesan berupa *route reply* yang dengan jalur dari *node* tujuan (*node* V) menuju *node* pengirim (*node* M) berupa *reversed path* dan dikirim ke *node* M. jika *multipath* ke *node* tujuan tersedia, maka *node* sumber akan menerima beberapa *route reply*.

3.3 Algoritma EA-SHORT ZRP

Pencarian rute dengan menggunakan algoritma EA-SHORT dapat ditunjukkan pada Gambar 3.6 dan Gambar 3.7.



Gambar 3.6 Flowchart pencarian rute pada EA-SHORT.



Gambar 3.7 Flowchart pencarian rute pada EA-SHORT (Lanjutan Gambar 3.6).

Mekanisme EA-SHORT dalam Gambar 3.6 dan Gambar 3.7 dapat dijelaskan sebagai berikut:

Jika suatu *node* (i) mengetahui kedatangan paket P,

Maka;

1. *Node* akan melakukan pengecekan rute untuk paket P pada *tabel overhear*. *Tabel overhear* merupakan tabel yang digunakan oleh *node* untuk menyimpan rute suatu paket.
2. Jika tidak ada rute yang cocok. Tambahkan *input* variabel e', dimana rute (e') sama dengan rute (p), *sequence number* (e') sama dengan *sequence number* (p) lalu jadikan rute (e') sebagai list rute pertama pada *tabel overhear* dengan isi (e') berupa *hop count* (p), *level energi* (p), dan nomer id (p). Proses selesai.

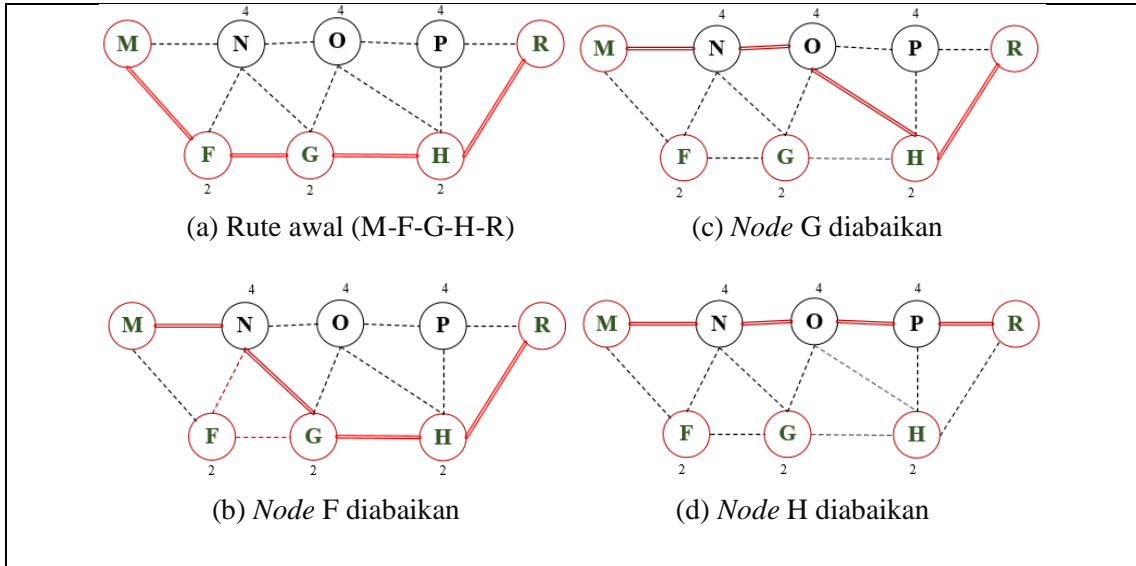
3. Misalkan rute (p) ditemukan pada *input*-an e, dengan kondisi *sequence number* (p) kurang dari *sequence number* (e), maka p akan diabaikan, dan proses selesai.
4. Jika *sequence number* (p) lebih besar dari *sequence number* pada (e), maka perbaharui e dengan ketentuan yakni *sequence number* (e) sama dengan *sequence number* (p), list rute pada tabel *overhear* (e) direset sehingga memiliki satu *input*-an rute dengan isi *hop count* (p), *level* energi (p). nomer id (p) dan proses selesai.
5. Jika *sequence number* (p) sama dengan *sequence number* (e), maka:
 - 5.1 Tambahkan *input*-an <hc (p), lvl (p), nid (p)> ke ov-list (e).
 - 5.2 Jika daftar *overhear* pada (e) memiliki tiga *input*-an yakni A,B,C yang memenuhi kondisi berikut, maka subpath ditemukan.
 - 1) *Hop count* pada (C) sama dengan *hop count* (B)+1 dan sama dengan *hop count* (A)+2
 - 2) *Level* energi pada (*node* i) lebih besar atau sama dengan jumlah maksimal energi pada ((A) dan *level* energi pada (C));
 - 3) *level* energi pada (*node* i) – *level* energi (B) ≥ 2 , maka akan diaktifkan sub-rute baru, dan hapus *input*-an (e) dari tabel *overhear*.
 - 5.3 Jika daftar *overhear* pada (e) memiliki dua *input*-an, yakni A dan B, dengan kondisi *hop count* (B) sama dengan *hop count* (A)+1 dan *level* energi pada pada *node* (i) \geq maksimal *level* energi (A) dan dan *level* energi (B)+2, tambahkan indikator I pada daftar tunggu indikator (*WaitingIndicator List*), dengan kandidat (I) sama dengan B. *sequence number* (I) disamakan dengan (e). rute (I) sama dengan rute (e), maka proses selesai.
 - 5.4 Jika daftar *overhear* (e) memiliki dua *input*-an B dan C, dengan kondisi *hop count* (C) sama dengan *hop count* (B)+1 dan *level* energi pada *node* (i) \leq dari maksimal *level* energi pada (B)+2 dan *level* energi (C), maka *node* I akan menyebarkan pesan SHORT untuk menginformasikan paket Q dengan ketentuan kandidat (Q) sama dengan B, *sequence number* pada (Q) sama dengan *sequence number* pada (e), dan rute (Q) sama dengan rute pada (e).

Ketika *node* i menerima paket informasi pesan SHORT Q, maka:

- 1) Bandingkan bidang Q dengan *input*-an yang valid dalam *WaitingIndicator list*.
- 2) Jika tidak ada yang cocok, maka abaikan paket Q;

Jika kondisi tidak terpenuhi, maka subrute yang lebih baik ditemukan. Aktifkan subrute ini dengan memperbaharui tabel *routing*. Proses selesai.

Contoh rute pada MANET yang dilakukan penerapan EA-SHORT



Gambar 3.8 Contoh proses perutean ulang dengan menggunakan algoritma EA-SHORT

Pada Gambar 3.8, menunjukkan contoh proses perutean ulang dengan menggunakan algoritma EA-SHORT, dimana rute awalnya yakni M-F-G-H-R (a). Terdapat *node* tetangga yakni N, O, P. *Node* N mengidentifikasi *node* F akan kehabisan energi dan kurang dari batas jumlah energi yang ditentukan dalam perutean, setelah dibandingkan dengan jumlah energi N, ternyata jumlah energi N lebih besar dari *node* F, sehingga *node* F diabaikan, rute berubah menjadi M-N-G-H-R (b). *Node* G diabaikan (c) karena *node* O mengidentifikasi jumlah energi *node* G yang hampir habis dan jumlah energi *node* O lebih besar, rute berubah menjadi M-N-O-H-R. Hal yang sama dilakukan pada *node* H, P mengidentifikasi jumlah energi *node* H, perutean ulang dilakukan karena energi P lebih besar dan jumlah energi H kurang dari batas jumlah energi yang ditetapkan (nilai *threshold*). Rute akhir setelah proses perutean ulang dengan algoritma EA-SHORT adalah M-N-O-P-R (d).

3.4 Parameter Skenario Simulasi

Parameter skenario simulasi digunakan sebagai suatu pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari suatu layanan. Dengan parameter performansi, suatu jaringan dapat memenuhi layanan yang berbeda menggunakan infrastruktur yang sama. Pada bagian ini merupakan skenario simulasi yang akan diimplementasikan menggunakan NS-2 versi 2.35. *Link layer* yang digunakan adalah berdasarkan pada standar IEEE 802.11. Fungsi

koordinasi terdistribusi IEEE 802.11 untuk nirkabel LAN digunakan sebagai lapisan MAC. Jumlah *node* yang ada pada area sebanyak 50 dan 100 *node* [3]. Simulasi dilakukan pada area persegi (*network area*) dengan ukuran 1000 m × 1000 m[3]. Waktu simulasi yang dilakukan selama 2000 s[7]. Digunakan variasi kecepatan *node* untuk melihat perbandingan kinerja kedua protokol, yakni sebesar 1 m/s, 5 m/s, dan 10 m/s serta menggunakan variasi *pause time* selama 45 s, 90 s, dan 180 s untuk mengetahui perbandingan kinerja kedua protokol. Jenis mobilitas yang digunakan pada simulasi adalah *Random Waypoint*. Setiap *node* secara independen memilih *destination* yang hendak dituju secara acak di dalam batasan jaringan. Parameter skenario dapat dilihat pada Tabel 3.2.

Tabel 3.2 Parameter skenario simulasi.

Parameter Skenario	
Type Parameter	Nilai Parameter
<i>MAC Layer</i>	IEEE 802.11
Jumlah <i>node</i>	50, 100
<i>Network area</i>	1000m x 1000m
Waktu simulasi	200 s
Kecepatan <i>node</i>	1m/s, 5m/s, 10m/s
<i>Pause time</i>	45s, 90s, 180s
Pergerakan <i>node</i>	<i>Random Waypoint</i>
<i>Propagation</i>	<i>TwoRayGround</i>
Antena	Omni <i>antenna</i>
Pola <i>traffic</i>	CBR
<i>Channel</i>	<i>Wireless</i>

BAB IV

HASIL DAN PEMBAHASAN

Bab ini membahas implementasi perancangan penelitian yang mengacu pada desain dan perancangan simulasi yang telah dibahas sebelumnya, serta membahas uji coba dan evaluasi hasil simulasi dari skenario NS-2 yang telah dilakukan. Pengujian akan dilakukan sesuai dengan perancangan metode penelitian pada Bab 3.

4.1 Implementasi Metode

Pada penelitian ini dilakukan penerapan metode *Energy Aware SHORT* (EA-SHORT) pada protokol ZRP, yang diharapkan dapat membantu protokol ZRP dalam menemukan rute pengiriman paket data yang *reliable*, dimana rute yang digunakan untuk mengirim paket dari *node* sumber ke *node* tujuan tidak mengalami *link failure* yang diakibatkan oleh *node* yang menjadi rute pengiriman data kehabisan energi. EA-SHORT diterapkan dengan mempertimbangkan energi yang dimiliki oleh tiap *node* untuk memilih apakah suatu *node* memiliki energi yang cukup untuk menerima dan meneruskan paket data ke *node* selanjutnya. Sehingga, dengan menerapkan metode tersebut diharapkan dapat meningkatkan performa protokol ZRP, dalam hal ini ditandai dengan adanya peningkatan nilai *throughput* dan *packet delivery ratio*, serta adanya penurunan *average end to end delay*. Implementasi metode EA-SHORT ke dalam protokol ZRP dilakukan dengan menambahkan skrip algoritma EA-SHORT pada kerangka protokol ZRP. Langkah-langkah penerapan algoritma EA-SHORT adalah sebagai berikut:

4.1.1 Pemodelan energi

Pada jaringan nirkabel, model energi adalah salah satu atribut opsional sebuah *node*. Model energi menyatakan *level* energi dari mobile *node*. Satuan energi yang digunakan adalah *Joule*. Langkah pertama yang dilakukan untuk mengimplementasikan EA-SHORT ke dalam kerangka ZRP yaitu mendeklarasikan variabel yang akan digunakan untuk melakukan perhitungan energi, adapun pendeklarasian variabel tersebut dilakukan pada file `zrp.h` berikut adalah skrip deklarasasi variabel yang digunakan:

```
double lvl, txPower, rxPower, txEnergy, rxEnergy, energiTransmisi;
```

Gambar 4. 1 Deklarasi variabel pada `zrp.h`

Berdasarkan skrip di atas dapat diketahui bahwa pada file `zrp.h` dilakukan deklarasi variabel sebagai berikut :

- `lvl` untuk menampung nilai sisa energi yang dimiliki oleh suatu *node*.

- `txPower` mendefinisikan besar daya yang dibutuhkan untuk mengirim paket.
- `rxPower` mendefinisikan besar daya yang dibutuhkan untuk menerima satu paket.
- `txEnergy` mendefinisikan energi yang dibutuhkan *node* untuk mengirim paket.
- `rxEnergy` mendefinisikan energi yang dibutuhkan *node* untuk menerima paket.
- `energiTransmisi` mendefinisikan energi yang dibutuhkan *node* dalam mengirim dan menerima paket data.

Setelah dilakukan pendeklarasian variabel yang digunakan, maka selanjutnya dilakukan inisialisasi nilai dari variabel yang telah dideklarasikan sebelumnya pada *file* `zrp.cc`, berikut adalah inisialisasi variabel yang akan digunakan :

```
lvl, rxPower, txPower, rxEnergy, txEnergy, energiTransmisi = 0.0;
```

Gambar 4. 2 Inisialisasi variabel pada `zrp.cc`

4.1.2 Algoritma EA-SHORT

Setelah didapatkan nilai energi yang dimiliki tiap *node* (`lvl`), selanjutnya akan dilakukan perbandingan antara energi pada *node* (*i*) dengan energi pada *node* tetangganya untuk menseleksi apakah *node* memiliki energi yang cukup untuk menerima dan mengirim paket data. Langkah awal adalah dengan menambahkan dua variabel yang akan digunakan untuk menampung informasi *hop count* dan *level* energi pada *header* paket, pada Gambar 4.5 adalah penambahan dilakukan pada *file* `zrp.h`.

```
int numHops_i, numHops_j;
double lvl_i, lvl_j;
```

Gambar 4. 3 Inisialisasi variabel untuk menampung *hop count* dan *level* energi

Variabel `numHops_i` dan `numHops_j` digunakan untuk menampung nilai *hop count* dari paket pada rute yang dilewatinya. Jumlah *hops* dari paket ketika melewati *node* tetangga dari *node* (*i*) akan dicatat dalam `numHops_i`, kemudian saat mencapai *node* selanjutnya, maka nilai *hops* dari *node* `numHops_i+1` akan disimpan pada `numHops_j`. Sehingga, nilai pada kedua variabel tersebut dapat dibandingkan untuk pengecekan kondisi pada saat melakukan proses pada *routing*. Selanjutnya adalah membuat skrip untuk algoritma EA-SHORT pada *file* `zrp.cc`. Berikut ini dilampirkan skrip algoritma EA-SHORT yang disisipkan ke dalam struktur protokol ZRP.

```
int
LinkStateList::findLink(nsaddr_t src, nsaddr_t dest, int seq, int
numHops_i, int numHops_j, double lvl_i, double lvl_j, int id, LinkState
**handle) {
    LinkState *cur;
    int foundFlag;
    cur = head ;
```

```

foundFlag = FALSE;
for(int i=0; i<numLinks_; i++) {
    if((src==cur->src_ && dest==cur->dest_) || (src==cur-
>dest_ && dest==cur->src_)) {
        foundFlag = TRUE;
        break;
    }
    cur=cur->next ;
}

```

Gambar 4. 4 Skrip untuk melakukan penambahan *link* (rute)

Skrip pada Gambar 4.4 yakni pada skrip `LinkStateList::findLink` adalah skrip untuk mendeteksi *link* atau rute, dimana informasi rute pada suatu paket digunakan untuk mendeteksi link, didapatkan dari rute pada paket yang telah dilewatinya dan *node* tujuannya. Fungsi tersebut akan mendeteksi paket yang datang dan mengambil informasi didalam paket berupa alamat *node* sumber `nsaddr_t src`, alamat *node* tujuan `nsaddr_t dest`, jumlah *sequence number* pada paket `int seq`, jumlah *hop count* pada paket `int numHops_i`, jumlah *hop count* pada paket `int numHops_j`, level energi pada *node* `i` `double lvl_i`, level energi pada *node* `j` `double lvl_j`, dan id dari paket `int id`, dan mengindikasikan *link* ditemukan pada potongan skrip `foundFlag = TRUE`, *link* kemudian akan ditambahkan.

```

.
*/kondisi sequence paket yang baru kurang dari sequence number paket
yang sudah ada */
void
LinkStateList::addLink(nsaddr_t src, nsaddr_t dest, int seq, int isup,
int numHops_i, int numHops_j, double lvl_i, double lvl_j, int id, Time
expiry) {
LinkState *newLS = new LinkState(src, dest, seq, numHops_i, numHops_j,
isup, lvl_i, lvl_j, id, expiry);
if(newLS == NULL) {
printf("### Memory Allocation Error in [LinkStateList::addLink] ###");
exit(0);
}
if (LinkStateList(pnew)->seq <LinkStateList(p)->seq)
{
(agent_->pktUtil_).pkt_drop(p);
}
}
.

```

Gambar 4. 5 Skrip untuk kondisi *sequence number* paket kurang dari *sequence number* paket yang sudah ada

Setelah paket terdeteksi, dan informasi untuk keperluan link diperoleh, maka skrip pada Gambar 4.5 adalah potongan skrip untuk menambahkan *link* dengan beberapa kondisi, yaitu kondisi pertama jika *sequence number* pada paket lebih kecil daripada *sequence number* pada informasi paket yang sudah ada. Pada `if`

(LinkStateList(pnew)->seq <LinkStateList(p)->seq), jika kondisi terpenuhi, maka paket akan di *drop*. (agent_->pktUtil_).pkt_drop(p).

```

.
*/kondisi sequence paket yang baru lebih besar dari sequence number
paket yang sudah ada */
if (LinkStateList(pnew)->seq > LinkStateList(p)->seq)
{
.
IARPUdate(nsaddr_t updateSrc, int seq, int numHops_i, numHops_j,
double lvl_i, lvl_j, Time expiry);
IARPUdate *newUpdate = new IARPUdate(updateSrc, seq_i,
seq_j,numHops_i, numHops_j, lvl_j, expiry);
if(newUpdate == NULL) {
printf("### Memory Allocation Error in
[IARPUdateDetectedList::addUpdate] ###");
exit(0); }
newUpdate->next_ = head_; // Make connections..
head_ = newUpdate;
numUpdates++; // Increment Updates..
}

```

Gambar 4. 6 Skrip untuk kondisi *sequence number* paket lebih besar daripada *sequence number* paket yang sudah ada

Skrip pada Gambar 4.6 adalah potongan skrip untuk kondisi kedua, dimana *sequence number* pada paket lebih besar dibandingkan dengan *sequence number* pada informasi paket yang sudah ada. Dimana pada kondisi `if (LinkStateList(pnew)->seq > LinkStateList(p)->seq)`, jika terpenuhi, maka akan dilakukan *update* pada *link* `IARPUdate *newUpdate = new IARPUdate(updateSrc, seq,numHops_i,numHops_j, lvl_j, expiry);`

```

.
if (LinkStateList (pnew)->seq==LinkStateList(p)) {
    newLS->next_ = head_;
    head_ = newLS;
    numLinks_++;
}
.
.
if (numLinks==3) {
.
If(LinkStateList->numHops==LinkStateList-
>numHops_i+1==LinkStateList->numHops_j+2) {
.
if ((MobileNode *node_)->lvl_i>=(Mobile *node)->->lvl_j) {
if ((MobileNode *node_)->lvl_i-(Mobile *node)->lvl_j)>=2) {
    (foundFlag == TRUE) {
        *handle = cur;
        removeLink(LinkState *prev, LinkState *toBeDeleted)
        return TRUE;
    }
.
.
}
}

```



```

.
.
else
if (numLinks==2) {
.
if (LinkStateList->numHops==LinkStateList->numHops_i+1)
{
if (Mobile *node)->lvl>=Mobile *node)->lvl_i, (Mobile *node)->lvl_j+2)
{ LinkStateList::addLink(nsaddr_t src, nsaddr_t dest, int seq, int
isup, int numHops_i, int numHops_j, double lvl_ double lvl_j, int id,
Time expiry) }
LinkState *newLS = new LinkState(src, dest, seq,numHops, isup,
lvl, id, expiry); // Create a new LinkState
if(newLS == NULL) { // Check for Allocation Error
printf("### Memory Allocation Error in
[LinkStateList::addLink] ###");
exit(0);
}
}
else
return FALSE;
}

```

Gambar 4. 7 Skrip algoritma EA-SHORT

Skrip pada Gambar 4.7 merupakan potongan skrip untuk kondisi ketiga, dimana *sequence number* paket yang datang sama dengan *sequence number* paket yang sudah ada. Jika kondisi `if (numLinks==3)` terpenuhi maka akan dibandingkan lvl energi pada *node*, jika level energi pada *node* (i) kurang dari nilai *threshold* yakni 2, maka akan dilakukan penghapusan rute dan kemudian rute baru akan digunakan. kondisi lainnya, jika `(numLinks==2)` terpenuhi, dan kondisi didalamnya juga terpenuhi, maka akan dilakukan penambahan *link* `LinkState *newLS = new LinkState(src, dest, seq,numHops_i,numHops_j, isup, lvl_i, lvl_j, id, expiry.`

4.2 Merancang Skenario Simulasi

4.2.1 Membuat pola trafik

Pada bagian skrip ini, menggunakan beberapa parameter yang akan ditentukan untuk simulasi. Parameter yang digunakan yaitu tipe koneksi yang digunakan, jumlah *node* dalam simulasi, jumlah *seed*, maksimum koneksi yang dikehendaki, dan paket *rate*. Cara menjalankan simulasi dapat dilihat pada skrip berikut.

```

//Masuk dalam direktori :
$ cd/home/ns-allinone-2.35/ns-2.35/indep-utils/cmu-scen-gen
//Untuk menjalankan perintah :
$ ns cbrgen.tcl -type [cbr] -nn [node] -seed [seed] -mc [connections]
-rate [rate] > [nama file untuk menyimpan data hasil generate]

```

Gambar 4. 8 Membuat pola trafik.

4.2.2 Membuat pola pergerakan *node*

Pada simulasi, jaringan MANET menggunakan pergerakan *node random waypoint* dan *node* bergerak dengan kecepatan antara 1 m/s , 5m/s dan 10 m/s. Beberapa parameter yang digunakan dalam skrip yaitu :

- a. `-n` : menyatakan jumlah *node* dalam jaringan
- b. `-p` : menunjukkan jeda waktu sebuah *node* dalam keadaan tidak bergerak
- c. `-M` : maksimum kecepatan *node* dalam bergerak
- d. `-t` : durasi waktu simulasi
- e. `-x` : ukuran (dalam sumbu x) topologi jaringan
- f. `-y` : ukuran (dalam sumbu y) topologi jaringan

Cara menjalankan simulasi dapat dilihat pada skrip berikut.

```
//Masuk kedalam direktori :  
$ cd/home/ns-allinone-2.35/ns-2.35/indep-utils/cmu-scen-gen/setdest  
//Untuk jalankan perintah :  
$ ./setdest -n [jumlah node] -p [pause time] -M [max speed] -t [waktu  
simulasi] -x [max X] -y [max Y] > [nama file untuk menyimpan data  
hasil generate]
```

Gambar 4. 9 Membuat pola pergerakan *node*.

4.2.3 Membuat skrip TCL

Langkah selanjutnya yaitu membuat *file* TCL untuk simulasi skenario uji coba. *File* yang berisi pola trafik dan pergerakan *node* akan dipanggil melalui *file* TCL. Terdapat dua jenis skrip *file* TCL yang digunakan, yaitu Gambar 4.8 adalah skrip untuk simulasi dengan ZRP standar dan pada Gambar 4.9 adalah skrip yang digunakan untuk simulasi EA-SHORT ZRP. Contoh skrip TCL dapat dilihat pada Gambar 4.8.

```
set val(chan) Channel/WirelessChannel ;  
set val(prop) Propagation/TwoRayGround ;  
set val(netif) Phy/WirelessPhy ;  
set val(mac) Mac/802_11 ;  
set val(ifq) Queue/DropTail/PriQueue ;  
set val(ll) LL ;  
set val(ant) Antenna/OmniAntenna ;  
set val(ifqlen) 50 ;  
set val(nn) 50 ;  
set val(rp) ZRP ;  
set val(cp) "50cbr20" ;  
set val(sc) "50a1" ;  
set val(x) 1000 ;  
set val(y) 1000 ;  
set val(stop) 200 ;  
Agent/ZRP set radius_ 2  
# program utama  
# Inisialisasi variabel umum
```

```

set ns_ [new Simulator] ;
set objektrace [open 50a1.tr w] ;
$ns_ trace-all $objektrace
set nam [open 50a2.nam w] ;
$ns_ namtrace-all-wireless $nam $val(x) $val(y)
# buat topologi jaringan dan implementasi area simulasi
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
# buat GOD untuk menyimpan informasi objek
create-god $val(nn)
#buat channel
set chan_1_ [new $val(chan)]
    # konfigurasi node
    $ns_ node-config -adhocRouting $val(rp) \
        -llType $val(ll) \
        -macType $val(mac) \
        -ifqType $val(ifq) \
        -ifqLen $val(ifqlen) \
        -agentTrace ON \
        -routerTrace ON \
        -macTrace OFF \
        -movementTrace ON \
        -channel $chan_1_ \
for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]

    $node_($i) random-motion 0;
    $ns_ initial_node_pos $node_($i) 20 ;
}
puts "Loading connection pattern..."
source $val(cp)puts "Loading skenario...."
source $val(sc)
# Mengatur waktu simulasi berakhir
    for {set i 0} {$i < $val(nn) } {incr i} {
$ns_ at 200.0 "$node_($i) reset";
    }
$ns_ at 200.0 "stop"
$ns_ at 200.0 "puts \"simulasi berakhir\" ; $ns_ halt"

#definisi prosedur stop untuk mengakhiri simulasi
proc stop {} {
    global ns_ objektrace
$ns_ flush-trace
    close $objektrace
}

    puts "mulai simulasi"
$ns_ run

```

Gambar 4. 10 Skrip TCL untuk ZRP

Pada Gambar 4.10, adalah contoh skrip TCL yang dijalankan untuk uji coba. Dalam skrip TCL terdapat paramater-parameter yang digunakan dalam simulasi. Seperti jenis antena set val(ant) Antenna/OmniAntenna ukuran luas area uji coba, yakni set val(x) 1000, set val(y) 1000, jenis protokol yang digunakan, waktu simulasi serta beberapa parameter lain yang dibutuhkan.

4.2.4 Melakukan *filtering* pada *file trace* (.tr)

Pada tahap ini didefinisikan *trace file* yang akan digunakan, seperti *throughput*, *end to end delay* dan PDR. *File trace* merupakan pencatatan seluruh *event* (kejadian) pada simulasi yang telah dibangun. Pembuatan *file trace* dilakukan dengan memanggil objek *trace* pada *library* yang dihasilkan setelah proses simulasi dengan file TCL. Untuk mendapatkan hasil pengujian terkait perbandingan kualitas jaringan MANET dengan menggunakan protokol ZRP dan EA-SHORT ZRP, maka dilakukan proses *filtering* dari *file trace* yang telah dibuat dengan menggunakan *file AWK*. Dari hasil *filtering* tersebut akan dianalisis untuk mendapatkan perbandingan kualitas kinerja protokol ZRP dan EA-SHORT ZRP dari segi *throughput*, *end-to-end delay* dan PDR. Pada Gambar 4.11 merupakan skrip perintah untuk menjalankan *file AWK* agar mendapatkan nilai kinerja dari *file trace*:

```
$ awk -f <nama_awk>.awk <nama_file_trace>.tr
```

Gambar 4. 11 Skrip perintah untuk menjalan *file trace*

4.3 Hasil Uji Coba

Hasil uji coba dari penelitian yang dilakukan yaitu meliputi analisis performansi protokol *routing* terhadap perubahan jumlah *node*, *pause time* dan kecepatan *node*. Jumlah *node* divariasikan untuk mengetahui perbedaan hasil uji coba terhadap kepadatan jaringan, *speed* digunakan untuk memvariasikan kecepatan mobilitas *node* dan *pause time* digunakan untuk memvariasikan lama waktu diam (stabilitas) dari *node* yang ada dalam jaringan. Hasil percobaan yang diperoleh merupakan hasil perulangan percobaan yang dilakukan sebanyak 5 kali pada setiap skenario. Hal tersebut dilakukan untuk mendapat akurasi data yang baik dalam membandingkan protokol *routing* ZRP standar dengan EA-SHORT ZRP. Parameter yang digunakan untuk menganalisis kinerja protokol *routing* adalah *throughput*, *end-to-end delay*, dan *packet delivery ratio* (PDR). Berikut adalah data yang diperoleh selama simulasi terhadap protokol *routing* ZRP standar dan EA-SHORT ZRP. Pada sub bab 4.3.1 dan sub bab 4.3.2, merupakan hasil simulasi yang dilakukan dengan 50 *node*, 100 *node* dengan luas area 1000 m x 1000 m terhadap *pause time* dan kecepatan.

4.3.1 Hasil Uji Coba

a. Hasil uji coba 50 *node* terhadap *throughput*

Skenario simulasi pertama, yakni mencari rata-rata jumlah paket data yang berhasil diterima disisi penerima setiap detiknya (*throughput*). Kinerja protokol ZRP berdasarkan

throughput didapat dari hasil simulasi skenario MANET dengan jumlah *node* yang terlibat sebanyak 50 *node* dengan variasi kecepatan *node* (*Speed*) dan *pause time* (PT) dalam *second* (s) dan dilakukan pengulangan pengujian sebanyak lima kali. Berikut merupakan hasil nilai kinerja *throughput* terhadap uji coba yang dilakukan dalam bentuk tabel seperti pada Tabel 4.1.

Tabel 4. 1 Hasil uji coba protokol ZRP standar dengan 50 *node* terhadap *throughput*.

Protokol routing	Speed (m/s)								
	1			5			10		
	PT (s)			PT (s)			PT (s)		
	45	90	180	45	90	180	45	90	180
ZRP Standar (kbps)	55,714	56,555	55,717	52,067	52,712	55,617	49,295	50,530	54,592
	55,380	56,042	55,549	51,417	51,819	55,250	48,708	49,870	54,175
	60,436	60,578	59,569	56,450	55,297	59,099	52,603	53,948	58,984
	51,060	51,370	51,619	48,893	48,675	51,426	45,073	47,364	51,182
	56,991	57,529	56,968	48,614	53,371	55,385	49,977	51,667	55,265
EA-SHORT ZRP (kbps)	65,028	65,433	65,518	60,902	60,696	64,084	56,862	58,999	64,084
	60,102	59,933	65,518	56,251	54,956	59,282	55,641	53,899	59,282
	68,436	68,346	68,577	63,270	63,800	66,968	58,928	60,943	66,968
	62,102	63,102	63,964	58,227	58,294	61,691	55,123	57,154	61,691
	62,996	62,533	63,968	56,368	58,992	61,303	55,396	53,795	61,385

Berdasarkan Tabel 4.1, hasil uji coba ke 1, nilai tertinggi yang didapatkan dari perbandingan ZRP Standar dan EA-SHORT ZRP dari *throughput* yakni 56,555 kbps dan 65,518 kbps. Pada uji coba ke 2, nilai tertinggi *throughput* yakni 56,042 kbps untuk ZRP standar dan 65,518 kbps untuk EA-SHORT ZRP. Pada uji coba ke 3, didapatkan nilai tertinggi 60,578 kbps untuk ZRP standar dan 68,577 kbps untuk EA-SHORT ZRP. Pada uji coba ke 4, nilai tertinggi yang diperoleh untuk ZRP standar yakni 51,619 kbps dan 63,964 kbps untuk EA-SHORT. Pada uji coba ke 5, nilai tertinggi *throughput* yang diperoleh yakni 57,529 kbps untuk ZRP standar dan 63,968 kbps untuk EA-SHORT ZRP.

Rata-rata nilai *throughput* yang didapatkan pada percobaan 1 sampai lima pada ZRP standar adalah 53,644 kbps, 53,134 kbps, 57,440 kbps, 49,629 kbps dan 53,974 kbps. Untuk EA-SHORT ZRP, nilai *throughput* yang dihasilkan pada percobaan 1 sampai 5 adalah 62,401 kbps, 58,318 kbps, 65,137 kbps, 60,150 kbps dan 59,637 kbps. Sedangkan rata-rata untuk *throughput* pada ZRP standar yakni 53,564 kbps dan EA-SHORT ZRP sebesar 61,129 kbps. Sehingga peningkatan nilai *throughput* dari ZRP standar terhadap EA-SHORT ZRP sebesar 7,564 kbps.

b. Hasil uji coba 100 *node* terhadap *throughput*

Skenario simulasi dilakukan dengan menggunakan protokol ZRP standar dan EA-SHORT ZRP dengan mencari rata-rata jumlah paket data yang berhasil diterima disisi penerima setiap detiknya (*throughput*). Kinerja protokol ZRP standar berdasarkan *throughput* didapat dari hasil simulasi skenario MANET dengan jumlah *node* yang terlibat sebanyak 100 *node* dengan variasi kecepatan *node* (*Speed*) dan *pause time* (PT) dalam *second* (s) dan dilakukan pengulangan pengujian sebanyak lima kali. Berikut merupakan hasil nilai kinerja *throughput* terhadap uji coba yang dilakukan dalam bentuk tabel seperti pada Tabel 4.2.

Tabel 4. 2 Hasil uji coba protokol ZRP standar dengan 100 *node* terhadap *throughput*

Protokol routing	Speed (m/s)								
	1			5			10		
	PT (s)			PT (s)			PT (s)		
	45	90	180	45	90	180	45	90	180
ZRP Standar (kbps)	11,143	10,222	11,371	9,679	8,911	11,929	7,895	9,942	8,560
	11,729	11,360	9,792	9,287	9,525	11,619	7,016	9,362	9,312
	11,585	11,646	11,660	9,127	9,135	11,716	7,293	8,541	9,673
	11,532	11,727	11,809	9,705	10,669	11,231	7,248	8,644	9,737
	11,510	11,738	11,854	9,863	10,439	11,822	7,097	9,144	9,832
EA-SHORT ZRP (kbps)	20,150	28,716	31,192	18,455	23,129	26,255	10,964	20,677	23,521
	20,238	28,697	31,397	18,747	20,359	20,936	10,544	13,660	20,257
	20,620	33,350	36,435	18,920	20,879	22,000	7,858	18,567	21,646
	18,526	33,797	36,787	18,212	18,103	22,258	10,834	18,227	22,197
	20,198	33,683	36,889	18,647	20,493	22,095	7,565	18,825	23,755

Berdasarkan Tabel 4.2, hasil uji coba ke 1, nilai tertinggi yang didapatkan dari perbandingan ZRP standar dan EA-SHORT ZRP terhadap *throughput* yakni 11,371 kbps dan 31,192 kbps. Pada uji coba ke 2, nilai tertinggi *throughput* yakni 11,729 kbps untuk ZRP standar dan 31,397 kbps untuk EA-SHORT ZRP. Pada uji coba ke 3, nilai tertinggi yakni 11,660 kbps untuk ZRP standar dan 36,435 kbps untuk EA-SHORT ZRP. Pada uji coba ke 4, nilai tertinggi yang diperoleh untuk ZRP standar yakni 11,809 kbps dan 36,787 kbps untuk EA-SHORT. Pada uji coba ke 5, nilai tertinggi *throughput* yang diperoleh yakni 11,854 kbps untuk ZRP standar dan 36,889 kbps untuk EA-SHORT ZRP.

Rata-rata nilai *throughput* yang didapatkan pada percobaan 1 sampai lima pada ZRP standar adalah 9,961 kbps, 9,889 kbps, 10,042 kbps, 10,256 kbps dan 10,367 kbps. Untuk EA-SHORT ZRP, nilai *throughput* yang dihasilkan pada percobaan 1 sampai 5 adalah 22,562 kbps, 20,537 kbps, 22,253 kbps, 22,105 kbps dan 22,461 kbps. Sedangkan rata-rata untuk *throughput* pada ZRP standar yakni 12,179 kbps dan EA-SHORT ZRP sebesar

21,984 kbps. Sehingga peningkatan nilai *throughput* dari ZRP standar terhadap EA-SHORT ZRP sebesar 9,804 kbps.

c. Hasil uji coba 50 *node* terhadap *average end to end delay*

Skenario uji coba kedua adalah mencari waktu yang dibutuhkan untuk mengirimkan paket data dari *node* sumber ke *node* tujuan (*end-to-end delay*) dengan variasi parameter kecepatan *node* dan *pause time* dalam *second* (s). Kinerja algoritma *routing* protokol ZRP standar berdasarkan *average end to end delay* didapat dari hasil simulasi skenario MANET dengan jumlah *node* yang terlibat sebanyak 50 *node* dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Hasil uji coba protokol ZRP standar dengan 50 *node* terhadap *average end to end delay*

Protokol routing	Speed (m/s)								
	1			5			10		
	PT (s)			PT (s)			PT (s)		
	45	90	180	45	90	180	45	90	180
ZRP standar (s)	0,079	0,068	0,057	0,125	0,116	0,069	0,127	0,113	0,058
	0,079	0,065	0,065	0,115	0,077	0,048	0,121	0,115	0,048
	0,074	0,071	0,067	0,140	0,114	0,059	0,125	0,107	0,049
	0,088	0,071	0,048	0,120	0,078	0,048	0,118	0,107	0,042
	0,079	0,068	0,054	0,108	0,068	0,058	0,101	0,104	0,047
EA-SHORT ZRP (s)	0,064	0,055	0,047	0,101	0,064	0,035	0,102	0,098	0,043
	0,064	0,057	0,057	0,107	0,063	0,035	0,114	0,098	0,043
	0,063	0,053	0,042	0,127	0,079	0,040	0,114	0,096	0,043
	0,071	0,057	0,042	0,095	0,052	0,033	0,109	0,087	0,047
	0,079	0,057	0,053	0,096	0,065	0,048	0,097	0,095	0,043

Berdasarkan Tabel 4.3, hasil uji coba ke 1, nilai *average end to end delay* terbaik yang didapatkan dari hasil uji ZRP standar yakni 0,057 s dan 0,035 s untuk EA-SHORT ZRP. Pada uji coba ke 2, nilai terbaik yakni 0,048 s untuk ZRP standar dan 0,035 s untuk EA-SHORT ZRP. Pada uji coba ke 3, didapatkan nilai terbaik 0,049 s untuk ZRP standar dan 0,040 s untuk EA-SHORT ZRP. Pada uji coba ke 4, nilai terbaik yang diperoleh untuk ZRP standar yakni 0,042 s dan 0,033 s untuk EA-SHORT. Pada uji coba ke 5, nilai *average end to end delay* terbaik yang diperoleh yakni 0,047 s untuk ZRP standar dan 0,043 s untuk EA-SHORT ZRP.

Rata-rata nilai *average end to end delay* yang didapatkan pada percobaan 1 sampai lima pada ZRP standar adalah 0,090 s, 0,081 s, 0,090 s, 0,080 s dan 0,076 s. Untuk EA-SHORT ZRP, nilai *average end to end delay* yang dihasilkan pada percobaan 1 sampai 5 adalah 0,068 s, 0,071 s, 0,073 s, 0,066 s dan 0,070 s. Sedangkan rata-rata untuk *average end to end delay* pada ZRP standar yakni 0,084 s dan EA-SHORT ZRP sebesar 0,070 s.

Sehingga penurunan nilai *average end to end delay* dari EA-SHORT ZRP terhadap ZRP standar sebesar 0,014 s.

d. Hasil uji coba 100 *node* terhadap *average end to end delay*

Skenario simulasi kedua adalah mencari rata-rata dari waktu yang dibutuhkan untuk mengirimkan paket data dari *node* sumber ke *node* tujuan (*end to end delay*) dengan variasi parameter kecepatan *node* dan *pause time* dalam *second* (s). Kinerja algoritma *routing* protokol ZRP standar berdasarkan rata-rata *end to end delay* didapat dari hasil simulasi skenario MANET dengan jumlah *node* yang terlibat sebanyak 100 *node* dapat dilihat pada Tabel 4.4

Tabel 4. 4 Hasil uji coba protokol ZRP standar dengan 100 *node* terhadap *average end to end delay*

Protokol routing	Speed (m/s)								
	1			5			10		
	PT (s)			PT (s)			PT (s)		
	45	90	180	45	90	180	45	90	180
ZRP Standar (s)	0,848	0,772	0,139	1,063	1,043	0,315	1,533	1,562	1,161
	0,877	0,749	0,135	1,070	1,093	0,377	1,529	1,342	1,238
	0,861	0,720	0,147	1,024	1,045	0,310	1,487	1,349	1,214
	0,885	0,746	0,130	1,287	1,060	0,346	1,412	1,217	1,169
	0,805	0,713	0,137	1,273	1,087	0,447	1,413	1,123	1,046
EA-SHORT ZRP (s)	0,807	0,162	0,037	0,856	0,861	0,227	1,261	1,245	0,696
	0,825	0,158	0,036	0,868	0,851	0,312	1,289	1,294	0,657
	0,866	0,161	0,023	0,818	0,895	0,298	1,248	1,212	0,683
	0,801	0,124	0,024	0,818	0,891	0,313	1,258	1,203	0,651
	0,802	0,155	0,038	0,839	0,821	0,336	1,226	1,111	0,647

Berdasarkan Tabel 4.4, hasil uji coba ke 1, nilai *end to end delay* terbaik yang didapatkan dari hasil uji ZRP standar yakni 0,139 s dan EA-SHORT ZRP yakni 0,037 s. Pada uji coba ke 2, nilai terbaik yakni 0,135 s untuk ZRP standar dan 0,036 s untuk EA-SHORT ZRP. Pada uji coba ke 3, didapatkan nilai terbaik 0,147 s untuk ZRP standar dan 0,023 s untuk EA-SHORT ZRP. Pada uji coba ke 4, nilai terbaik yang diperoleh untuk ZRP standar yakni 0,130 s dan 0,024 s untuk EA-SHORT. Pada uji coba ke 5, nilai *end to end delay* terbaik yang diperoleh yakni 0,137 s untuk ZRP standar dan 0,038 s untuk EA-SHORT ZRP.

Rata-rata nilai *average end to end delay* yang didapatkan pada percobaan 1 sampai lima pada ZRP standar adalah 0,937 s, 0,934 s, 0,906 s, 0,917 s dan 0,894 s. Untuk EA-SHORT ZRP, nilai *average end to end delay* yang dihasilkan pada percobaan 1 sampai 5 adalah 0,684 s, 0,699 s, 0,689 s, 0,676 s dan 0,664 s. Sedangkan rata-rata untuk *average end to end delay* pada ZRP standar yakni 0,918 s dan EA-SHORT ZRP sebesar 0,682 s.

Sehingga penurunan nilai *average end to end delay* dari EA-SHORT ZRP terhadap ZRP standar sebesar 0,235 s.

e. Hasil uji coba 50 *node* terhadap PDR

Skenario simulasi ketiga adalah mencari nilai perbandingan antara paket data yang terkirim (paket data yang berhasil diterima oleh *node* tujuan) dengan jumlah paket data yang dikirimkan oleh *node* sumber (paket data yang di-generate *node* sumber) dengan jumlah 50 *node* simulasi dan variasi parameter kecepatan *node* dan *pause time*.

Tabel 4. 5 Hasil uji coba 50 *node* terhadap PDR

Protokol routing	Speed (m/s)								
	1			5			10		
	PT (s)			PT (s)			PT (s)		
	45	90	180	45	90	180	45	90	180
ZRP standar (%)	98,168	99,086	98,122	90,860	91,512	98,028	86,219	89,820	95,748
	98,264	98,693	98,609	91,449	91,131	97,807	86,249	90,182	96,237
	98,220	98,417	98,166	90,421	90,828	97,327	85,677	88,854	95,405
	98,298	98,818	98,301	90,915	90,838	98,386	85,867	90,230	95,783
	98,463	98,292	98,681	90,946	91,211	97,272	86,504	89,731	95,581
EA-SHORT ZRP (%)	98,354	98,620	98,627	91,666	92,172	98,475	86,313	90,751	96,678
	98,648	99,038	98,627	92,392	91,494	98,429	86,294	89,490	96,274
	98,586	98,663	98,858	92,738	91,963	98,142	86,255	89,212	95,743
	98,331	99,209	98,471	91,109	92,549	98,882	86,245	91,466	96,388
	98,576	99,514	98,681	91,839	91,508	98,272	86,961	90,526	95,806

Berdasarkan Tabel 4.5, hasil uji coba ke 1, nilai tertinggi yang didapatkan dari perbandingan ZRP Standar dan EA-SHORT ZRP dari PDR yakni 98,086 % dan 98,627 %. Pada uji coba ke 2, nilai tertinggi PDR yakni 98,693 % untuk ZRP standar dan 99,627 % untuk EA-SHORT ZRP. Pada uji coba ke 3, didapatkan nilai tertinggi 98,417 % untuk ZRP standar dan 98,858 % untuk EA-SHORT ZRP. Pada uji coba ke 4, nilai tertinggi yang diperoleh untuk ZRP standar yakni 98,818 % dan 99,209 % untuk EA-SHORT ZRP. Pada uji coba ke 5, nilai tertinggi PDR yang diperoleh yakni 98,681 % untuk ZRP standar dan 99,514 % untuk EA-SHORT ZRP.

Rata-rata nilai PDR yang didapatkan pada percobaan 1 sampai lima pada ZRP standar adalah 94,174 %, 94,291 %, 93,702 %, 94,16 % dan 94,076 %. Untuk EA-SHORT ZRP, nilai PDR yang dihasilkan pada percobaan 1 sampai 5 adalah 94,628 %, 94,521 %, 094,462 %, 94,739 % dan 94,631 %. Sedangkan rata-rata untuk PDR pada ZRP standar yakni 94,080 % dan EA-SHORT ZRP sebesar 94,596 %. Sehingga peningkatan nilai PDR dari EA-SHORT ZRP terhadap ZRP standar sebesar 0,516 %.

f. Hasil uji coba 100 *node* terhadap PDR

Skenario simulasi ketiga adalah mencari nilai perbandingan antara paket data yang terkirim (paket data yang berhasil diterima oleh *node* tujuan) dengan jumlah paket data yang dikirimkan oleh *node* sumber (paket data yang di-generate *node* sumber) dengan jumlah 100 *node* simulasi dan variasi parameter kecepatan *node* dan *pause time*.

Tabel 4. 6 Hasil uji coba 100 *node* terhadap PDR

Protokol routing	Speed (m/s)								
	1			5			10		
	PT (s)			PT (s)			PT (s)		
	45	90	180	45	90	180	45	90	180
ZRP Standar (%)	15,895	17,462	21,702	12,816	15,589	15,576	6,413	13,039	14,971
	17,165	17,405	21,520	12,583	15,859	16,600	7,181	15,091	14,016
	13,992	14,802	19,000	12,029	14,333	16,697	6,224	15,601	14,293
	15,151	15,500	15,477	12,936	14,504	13,118	6,033	7,705	14,927
	15,741	23,363	23,075	12,787	16,186	13,097	7,665	14,264	15,381
EA-SHORT ZRP (%)	16,630	24,362	25,704	14,108	16,834	19,358	8,600	14,604	15,534
	20,078	21,520	28,710	15,859	16,298	19,780	12,751	16,246	16,041
	15,957	24,788	28,956	13,997	16,947	20,515	11,931	16,780	18,952
	16,615	28,178	30,578	14,235	15,386	20,121	12,769	16,409	18,931
	16,082	27,242	30,518	16,170	16,928	17,386	10,396	16,102	16,104

Berdasarkan Tabel 4.6, hasil uji coba ke 1, nilai tertinggi yang didapatkan dari perbandingan ZRP standar dan EA-SHORT ZRP terhadap PDR yakni 21,702 % dan 25,704 %. Pada uji coba ke 2, nilai tertinggi PDR yakni 21,520 % untuk ZRP standar dan 28,710 % untuk EA-SHORT ZRP. Pada uji coba ke 3, didapatkan nilai tertinggi 19,000 % untuk ZRP standar dan 28,956 % untuk EA-SHORT ZRP. Pada uji coba ke 4, nilai tertinggi yang diperoleh untuk ZRP standar yakni 15,500 % dan 30,578 % untuk EA-SHORT. Pada uji coba ke 5, nilai tertinggi PDR yang diperoleh yakni 23,363 % untuk ZRP standar dan 30,518 % untuk EA-SHORT ZRP.

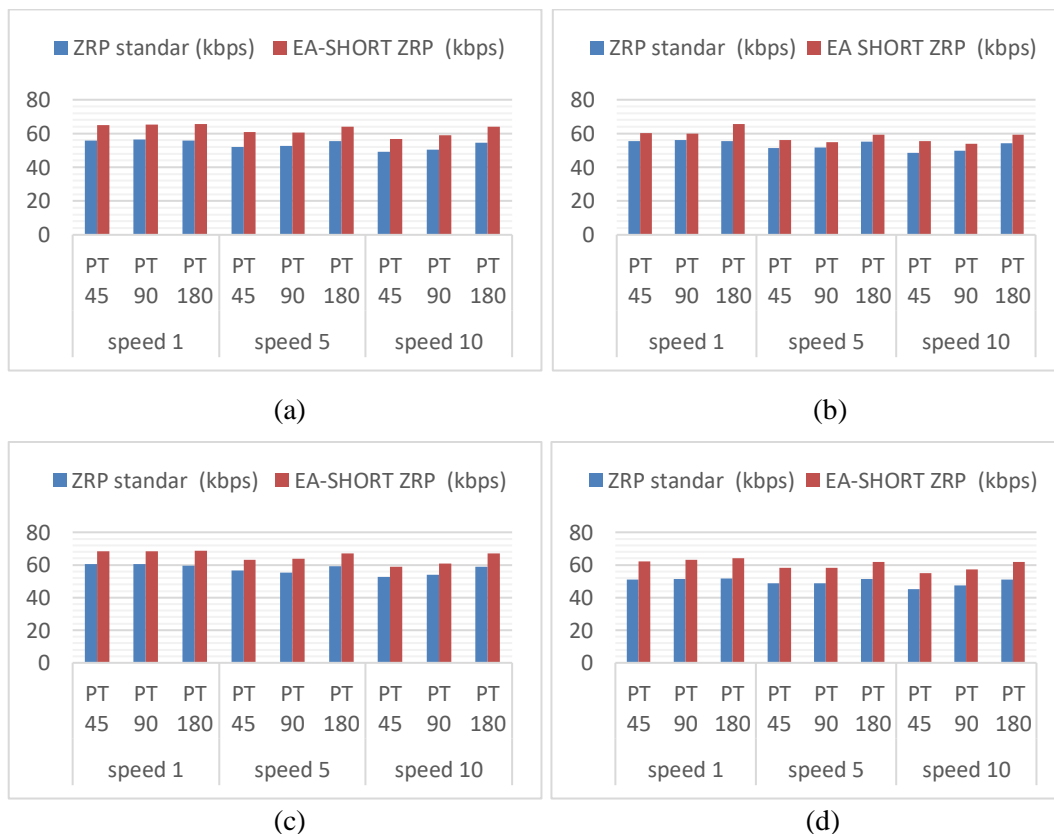
Rata-rata nilai PDR yang didapatkan pada percobaan 1 sampai lima pada ZRP standar adalah 14,829 %, 15,269 %, 14,108 %, 12,817 % dan 15,729 %. Untuk EA-SHORT ZRP, nilai PDR yang dihasilkan pada percobaan 1 sampai 5 adalah 17,304 %, 18,587 %, 18,758 %, 19,247 % dan 18,548 %. Sedangkan rata-rata untuk PDR pada ZRP standar yakni 14,550 % dan EA-SHORT ZRP sebesar 18,489 %. Sehingga peningkatan nilai PDR dari EA-SHORT ZRP terhadap ZRP standar sebesar 3,938 %.

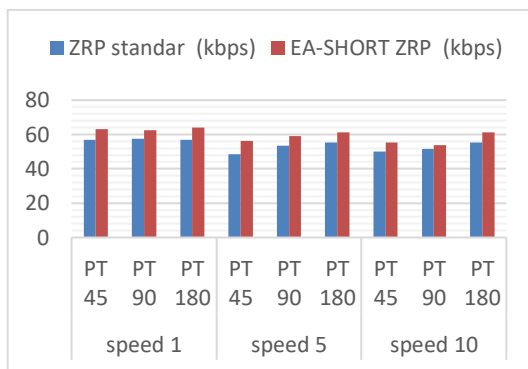
4.4 Analisis kinerja hasil simulasi jaringan

Pada sub bab ini, akan dilakukan analisis kinerja simulasi jaringan yang telah dilakukan pada sub bab 4.2 dan ditampilkan hasilnya dalam sub bab 4.3. Dalam hal ini akan dianalisis mengenai tingkat kualitas kinerja antara ZRP standar dan EA-SHORT ZRP dengan menggunakan rata-rata dari hasil simulasi jaringan.

a. Hasil pengujian terhadap *throughput*

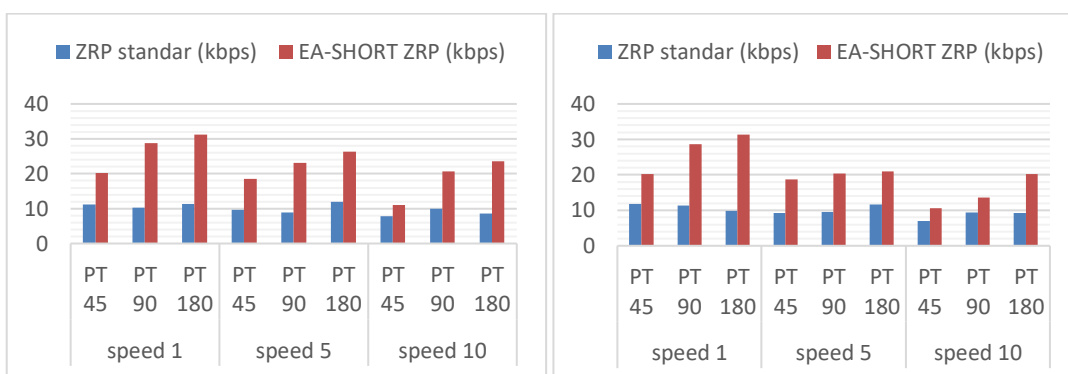
Throughput merupakan jumlah rata-rata pengiriman paket data yang diterima setiap detik. Hal ini dipengaruhi oleh beberapa faktor, seperti ketahanan dari *link* pengiriman paket data serta proses pencarian jalur pengiriman hingga jalur tersebut terbentuk. Pada situasi ZRP standar, energi pada setiap *node* akan berkurang selama simulasi berjalan dan *link* akan terputus ketika energi pada *node* habis. Dalam kondisi ini, akan terjadi *route error* dan paket akan dikirim kembali menuju *node* sumber, dan *node* sumber akan menghapus *link* tersebut dalam topologi tabel-nya kemudian *node* akan melakukan *update* tabel *routing* dan menentukan kembali rute terdekat untuk mencapai *node* tujuan. Setelah dilakukan penerapan algoritma pada kerangka protokol ZRP, dengan memperhatikan nilai energi pada setiap *node* agar tidak terjadi *link failure* dalam prosesnya, hasil *throughput* sebelum dan sesudah penerapan algoritma dapat dilihat perbandingan hasilnya dalam bentuk grafik pada Gambar 4. 12 dan Gambar 4.13.



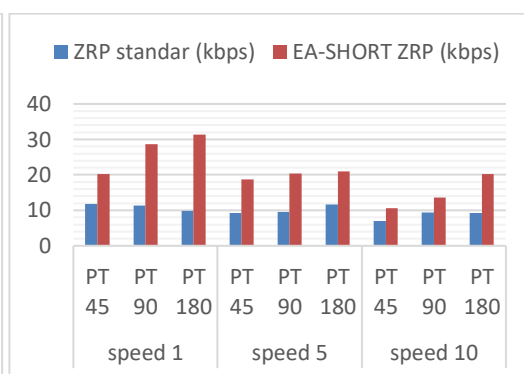


(e)

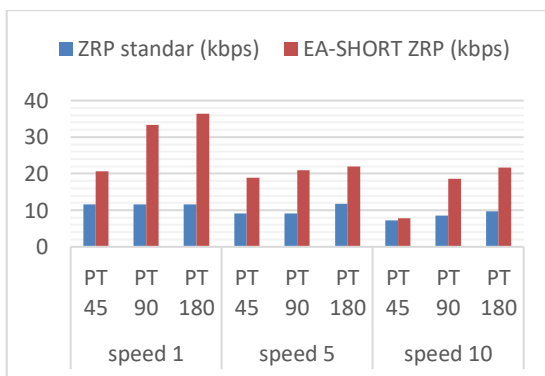
Gambar 4. 12 Grafik perbandingan hasil uji coba 50 *node* terhadap nilai *throughput* pada (a) uji coba ke 1 (b) uji coba ke 2, (c) uji coba ke 3, (d) uji coba ke 4 (e) uji coba ke 5



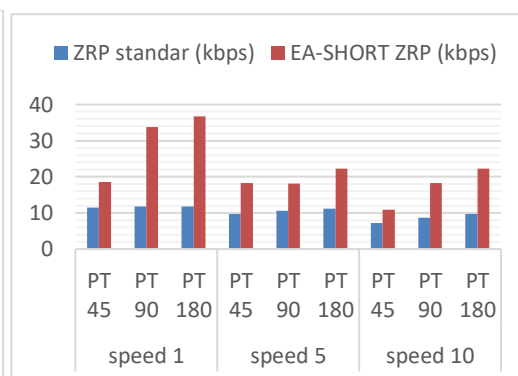
(a)



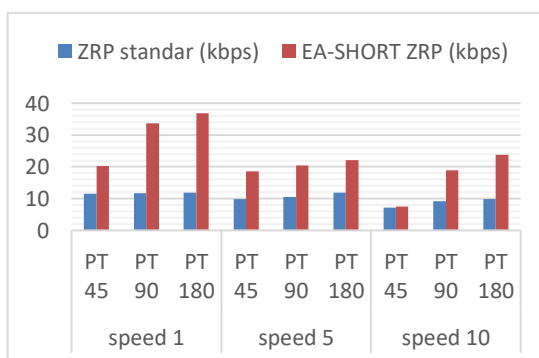
(b)



(c)



(d)



(e)

Gambar 4. 13 Grafik perbandingan hasil uji coba 100 *node* terhadap nilai *throughput* pada (a) uji coba ke 1, (b) uji coba ke 2, (c) uji coba ke 3, (d) uji coba ke 4 (e) uji coba ke 5

Berdasarkan grafik *throughput* pada Gambar 4.12 dan Gambar 4.13 menunjukkan bahwa kondisi protokol ZRP sesudah diterapkan algoritma EA-SHORT memiliki kualitas lebih baik daripada protokol ZRP standar. Berdasarkan kecepatan *node*, maka semakin besar kecepatan *node* maka nilai *throughput* semakin rendah. Kecepatan *node* menyebabkan posisi *node* dalam topologi jaringan bergerak sehingga jika parameter simulasi *speed* semakin tinggi, maka kemungkinan terjadi kerusakan rute juga akan semakin tinggi dan mengakibatkan menurunnya kualitas *throughput*.

Dari hasil *throughput*, jika dilihat berdasarkan *pause time*, nilai *throughput* mengalami peningkatan seiring dengan bertambahnya jumlah *pause time*, nilai *pause time* yang lebih tinggi artinya *node* berada dalam posisi diam yang lebih lama (stabil) sehingga menurunkan kemungkinan kerusakan rute akibat pergerakan *node*. Hal inilah yang menyebabkan nilai *throughput* lebih baik.

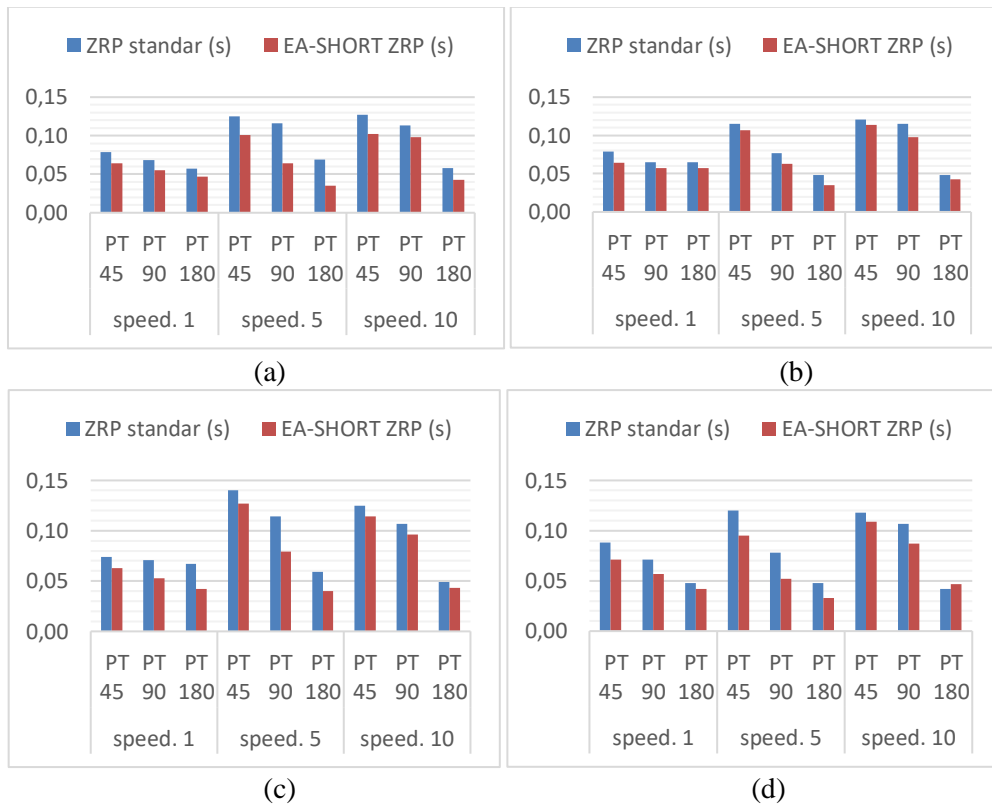
Nilai *throughput* pada 50 *node* lebih tinggi daripada 100 *node*. Jumlah *node* yang semakin bertambah membuat kecepatan *throughput* berkurang. Hal ini disebabkan karena kepadatan trafik yang lebih tinggi dan lebih banyak zona yang terbentuk oleh ZRP. Karena setiap *node* memiliki informasi *node* tetangga pada zona lokalnya, maka semakin banyak jumlah *node* yang tersedia menyebabkan semakin banyak jumlah trafik sehingga menyebabkan kongesti pada *node* perantara, kongesti adalah suatu kondisi dimana *node* perantara menanggung beban yang melebihi kapasitas jaringan. Hal ini menyebabkan beberapa paket akan di *drop*, sehingga mengakibatkan *throughput* koneksi utama menjadi turun.

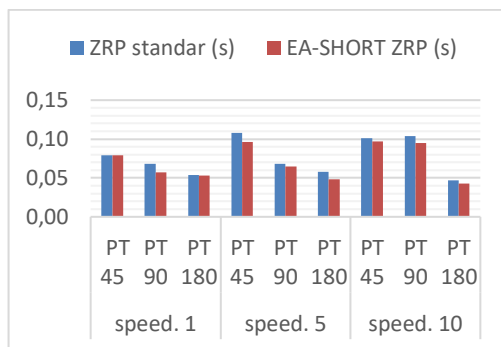
EA-SHORT ZRP membuktikan dapat menghasilkan nilai *throughput* yang lebih baik dibandingkan ZRP standar baik pada skenario dengan 50 *node* ataupun 100 *node*. Hal ini disebabkan algoritma EA-SHORT mengantisipasi terjadinya kerusakan rute akibat energi yang terdapat pada *node* habis dengan mengalihkannya pada *node* terdekatnya (*load balancing*). Antisipasi yang dilakukan sebelum rute terputus menyebabkan waktu yang dibutuhkan untuk mengirim paket lebih rendah. Nilai *throughput* meningkat pada EA-SHORT ZRP dikarenakan rata-rata *end to end delay* mengalami penurunan sehingga jumlah paket yang dapat sampai tujuan meningkat. EA-SHORT ZRP meningkatkan nilai *throughput* dengan memastikan paket tersebut melewati rute yang handal yakni jumlah energi yang cukup sehingga tidak terjadi *link failure*.

b. Hasil pengujian terhadap *average end to end delay*

Average end to end delay adalah rata-rata dari jumlah waktu yang digunakan oleh sebuah paket ketika dikirim oleh sebuah *node* dan diterima di *node* tujuan. Faktor yang sangat mempengaruhi *end to end delay* adalah waktu untuk *routing* protokol tersebut dalam menemukan rute, hal ini dikarenakan sebelum mengirim pesan, *node* sumber terlebih dahulu harus mengetahui rute yang menghubungkan ke *node* tujuan. Selain itu dapat dipengaruhi oleh proses *delay* pengiriman paket dari satu *node* ke *node* lainnya karena membutuhkan waktu untuk menganalisis harus dibawakan paket tersebut. Dalam proses pengiriman paket di luar zona, protokol ZRP akan melakukan *bordercast* paket RREQ pada *peripheral node* dan menyebabkan *delay* meningkat. Jika dalam tabel *routing peripheral node* tujuan tidak ditemukan, maka akan dilakukan *bordercast* lagi sampai *node* tujuan paket ditemukan.

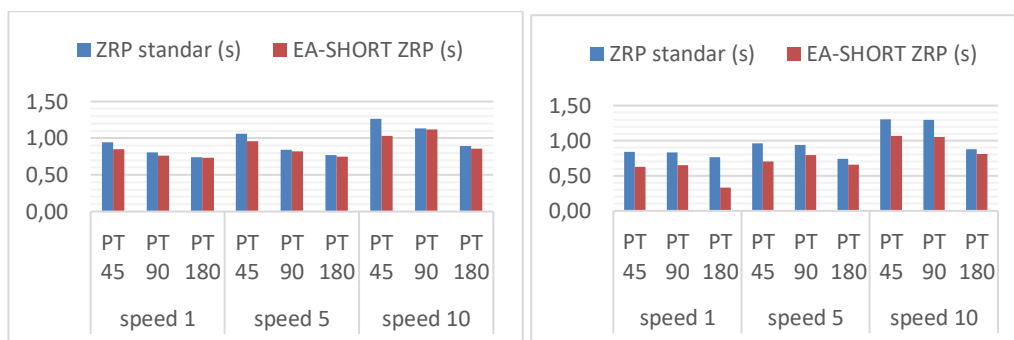
Berikut merupakan hasil dan grafik kualitas *average end-to-end delay* berdasarkan jumlah *node* yaitu 50 dan 100, variasi kecepatan *node* dan *pause time*, seperti pada Gambar 4.14 dan Gambar 4.15.



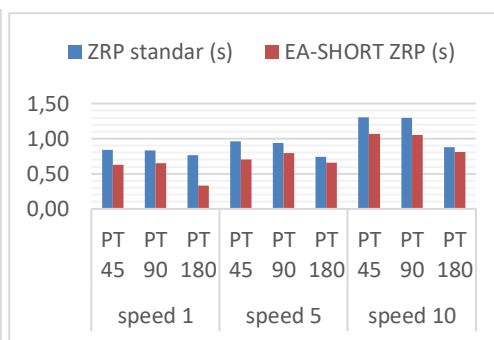


(e)

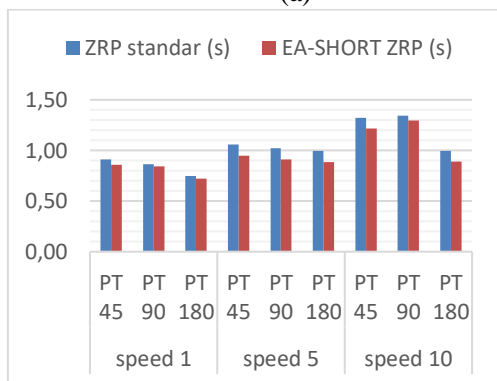
Gambar 4. 14 Grafik perbandingan hasil uji coba 50 *node* terhadap nilai *end to end delay* pada (a) uji coba ke 1, (b) uji coba ke 2, (c) uji coba ke 3, (d) uji coba ke 4, (e) uji coba ke 5



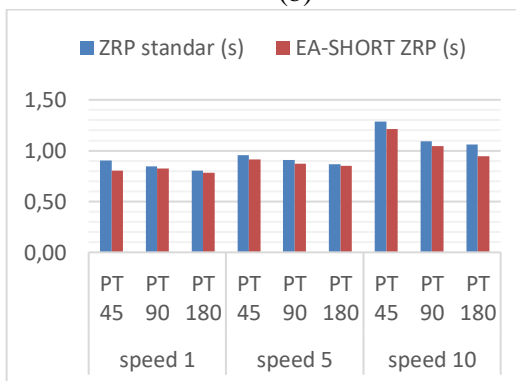
(a)



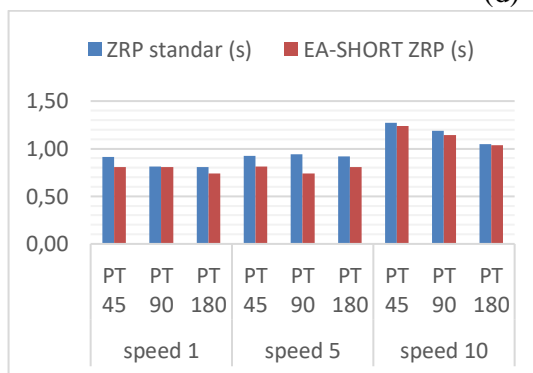
(b)



(c)



(d)



(e)

Gambar 4. 15 Grafik perbandingan hasil uji coba 100 *node* terhadap nilai *end to end delay* pada (a) uji coba ke 1, (b) uji coba ke 2, (c) uji coba ke 3, (d) uji coba ke 4, (e) uji coba ke 5

Berdasarkan grafik rata-rata *average end-to-end delay* pada Gambar 4.14 dan Gambar 4.15, bahwa *average end to end delay* setiap skenario pada 50 *node* dan 100 *node* mengalami perubahan pada setiap parameter simulasi. Perubahan kecepatan pada *node* menyebabkan perubahan hasil pada nilai *average end to end delay*. Pada *node* 100, nilai rata-rata *average end to end delay* lebih tinggi dibandingkan dengan *average end to end delay* pada *node* 50, dikarenakan jaringan yang lebih padat dan lebih banyak transmisi data yang dilakukan dan pada 100 *node*, zona yang terbentuk menjadi lebih banyak dibandingkan *node* 50, sehingga *node* sumber melalui IERP akan lebih sering melakukan *bordercast* terlebih dahulu saat akan mengirim paket.

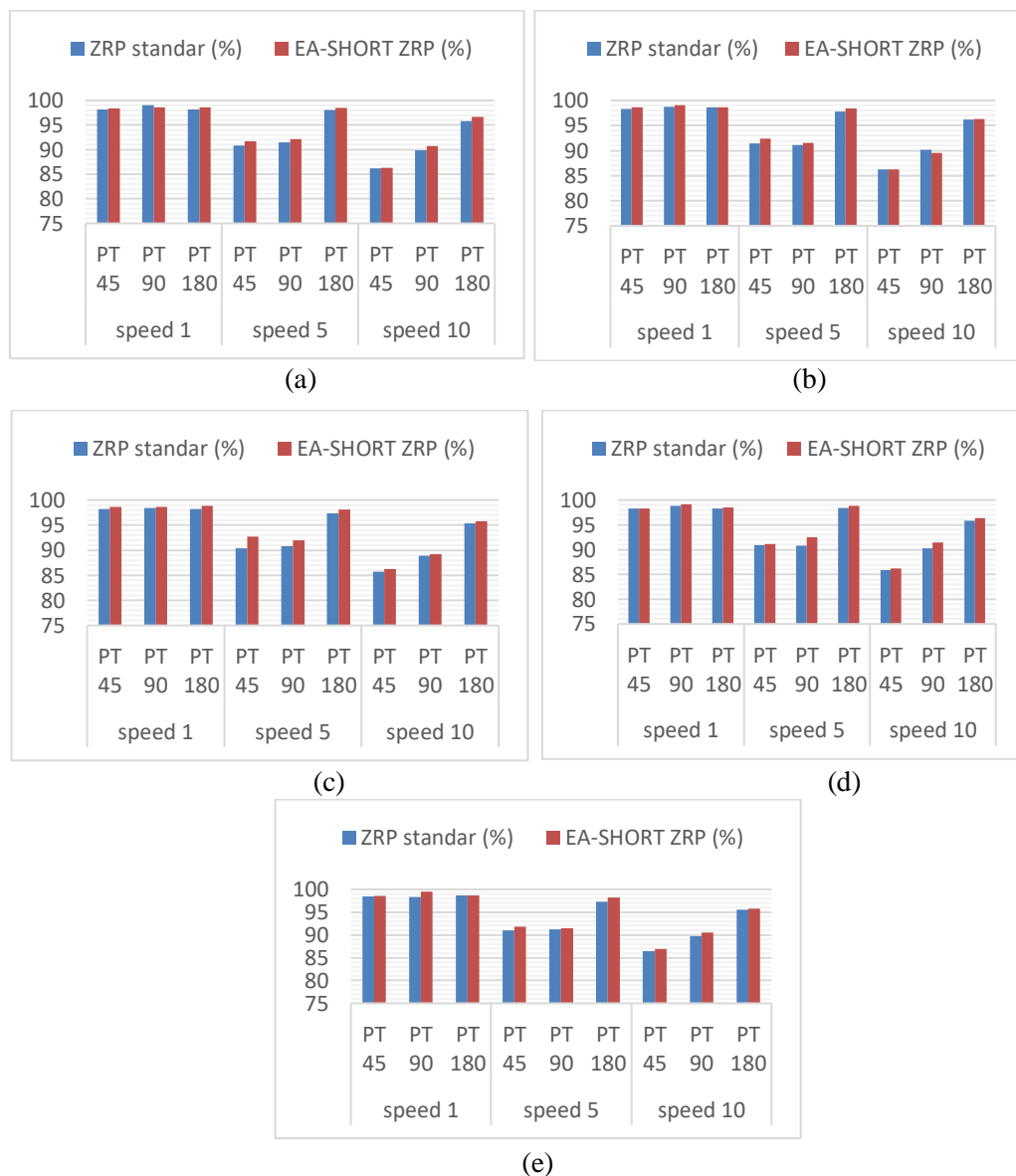
Kecepatan *node* juga berpengaruh pada hasil *average end to end delay*. Semakin tinggi kecepatan *node*, maka nilai *average end to end delay* semakin tinggi. Hal ini dikarenakan perubahan pada posisi *node* yang lebih cepat dan posisi *node* menjadi tidak stabil, dan mengakibatkan kerusakan pada rute sehingga IARP lebih sering melakukan pemeliharaan rute. Semakin sering pemeliharaan rute dilakukan, maka semakin tinggi nilai *average end to end delay*, hal ini menyebabkan nilai *buffer* pada paket data jaringan semakin tinggi. Perubahan *pause time* juga membuat perubahan pada nilai *average end to end delay*. Semakin tinggi nilai *pause time*, maka nilai *average end to end delay* semakin rendah, karena *pause time* yang lebih tinggi menyebabkan posisi *node* lebih stabil dan pengiriman paket dapat berjalan dengan baik. Selain kerusakan rute akibat pergerakan *node*, padatnya transmisi pada jaringan juga berakibat pada menurunnya jumlah energi yang lebih cepat, sehingga kemungkinan kerusakan rute akibat *node* kehabisan energi juga menjadi lebih tinggi.

Hasil penerapan algoritma EA-SHORT pada protokol ZRP, menunjukkan algoritma ini dapat menurunkan nilai *average end to end delay*. Dikarenakan algoritma EA-SHORT memastikan paket memiliki rute yang stabil dengan jumlah energi yang cukup selama proses transmisi. Sehingga jika rute mengalami perubahan akibat pergerakan *node*, algoritma EA-SHORT akan mengusahakan perubahan rute yang lebih baik kualitas energinya dibandingkan dengan ZRP standar. Dari rata-rata hasil pada setiap uji coba, didapatkan kualitas *average end to end delay* mengalami peningkatan seiring dengan bertambahnya jumlah *speed* yang digunakan. Hal ini dikarenakan topologi jaringan berubah lebih cepat dan lebih jauh pada area yang luas sehingga menyebabkan proses transmisi data menjadi lebih lama karena *hop* yang lebih panjang, dan kemungkinan untuk terjadi kerusakan rute lebih tinggi jika pergerakan *node* lebih cepat sehingga

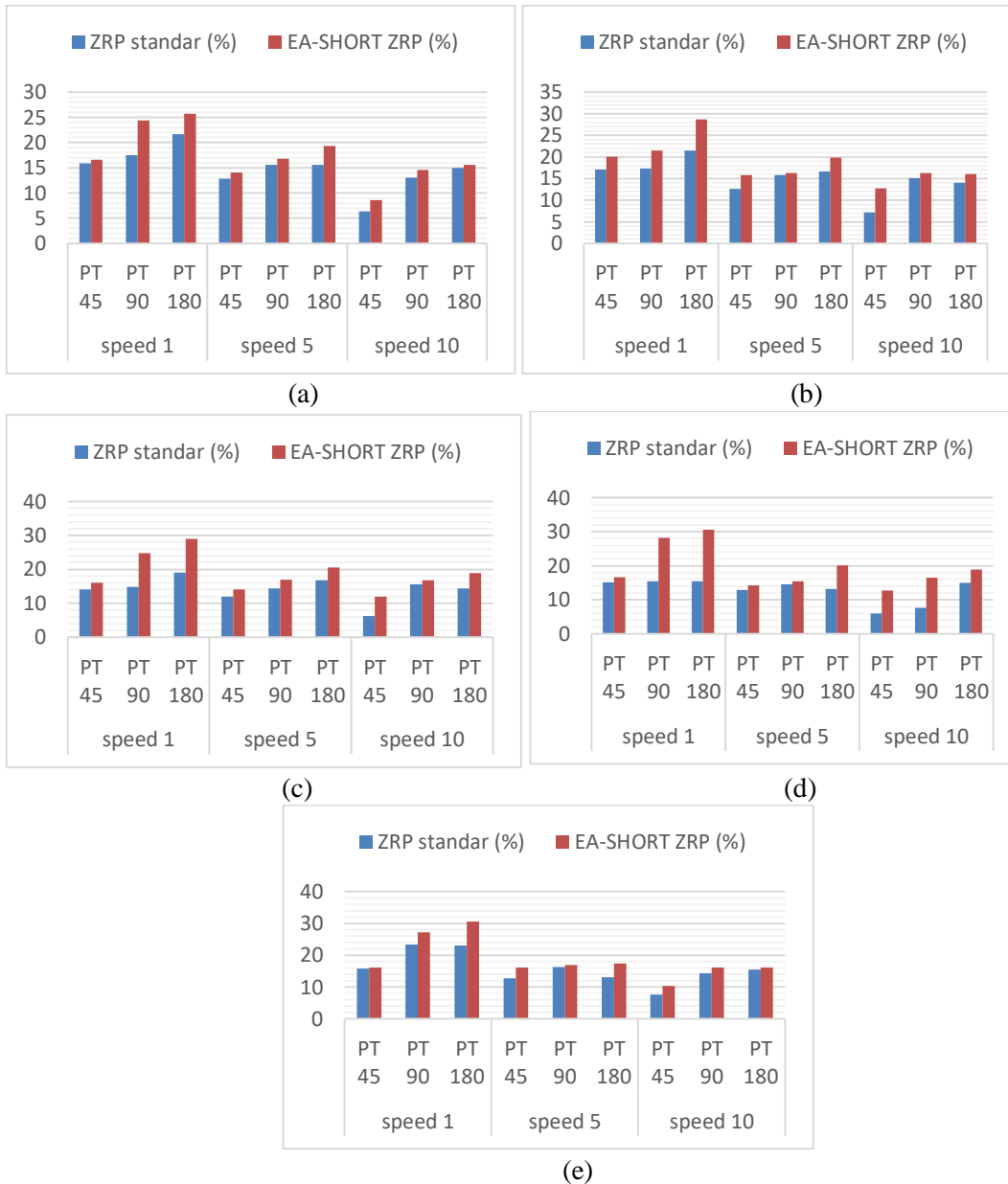
menemukan rute yang baru akan meningkatkan nilai *delay*. Pada setiap kondisi jumlah *node* memiliki kenaikan kualitas yang beragam dikarenakan kondisi simulasi jaringan yang bergerak secara *random* (kondisi *random waypoint*).

c. Hasil pengujian 50 *node* terhadap PDR

Skenario uji coba ketiga adalah mencari nilai perbandingan antara paket data yang terkirim (paket data yang berhasil diterima oleh *node* tujuan) dengan jumlah paket data yang dikirimkan oleh *node* sumber (paket data yang di-generate *node* sumber). Hasil dari peregrakan *node* terhadap PDR dapat dilihat pada grafik yang terdapat pada Gambar 4.16 dan Gambar 4.17.



Gambar 4. 16 Grafik perbandingan hasil uji coba 50 *node* terhadap nilai PDR pada (a) uji coba ke 1, (b) uji coba ke 2, (c) uji coba ke 3, (d) uji coba ke 4 (e) uji coba ke 5



Gambar 4. 17 Grafik perbandingan hasil uji coba 100 *node* terhadap nilai PDR pada (a) uji coba ke 1, (b) uji coba ke 2, (c) uji coba ke 3, (d) uji coba ke 4 (e) uji coba ke 5

Hasil nilai PDR yang dihasilkan pada kelima percobaan Gambar 4.16 dan Gambar 4.17 memperlihatkan nilai PDR pada EA-SHORT ZRP lebih tinggi daripada ZRP standar. Kecepatan pada *node* yang berubah menyebabkan nilai PDR juga berubah. Semakin tinggi kecepatan *node*, semakin rendah nilai PDR yang dihasilkan, perubahan rute karena kondisi *node* yang tidak stabil menyebabkan lebih banyak rute yang terputus dan terjadi paket *drop*.

Nilai PDR dengan 50 *node*, baik untuk ZRP standar ataupun EA-SHORT ZRP tidak jauh berbeda. *Node* 50 pada area 1000 m x 1000 m masih menghasilkan nilai rata-rata PDR yang cukup tinggi. Namun, EA-SHORT ZRP menghasilkan nilai rata-rata yang

lebih tinggi pada setiap percobaan dibandingkan dengan ZRP standar. Hal ini karena EA-SHORT ZRP berusaha memastikan paket sampai ketujuannya dengan mengalihkan rute (*load balancing*) menuju tetangga terdekatnya sebelum rute terputus akibat habisnya energi pada *node* selama transmisi. Hal ini dikarenakan semakin padat lingkungan simulasi, yang menyebabkan kemungkinan terjadinya putus *link* antar *node* semakin rendah, sehingga jumlah paket yang di-*drop* juga akan semakin berkurang.

Nilai PDR dengan 100 *node*, lebih rendah pada ZRP standar dibandingkan EA-SHORT ZRP dan besarnya berbeda cukup jauh dibandingkan nilai PDR pada 50 *node* yang menghasilkan nilai rata-rata PDR yang cukup tinggi. Pada 100 *node*, nilai PDR mengalami penurunan karena beberapa faktor seperti *node* yang lebih banyak dengan luas area yang besar sehingga menyebabkan jaringan menjadi lebih padat dan menurunkan kualitas pelayanan jaringan. Namun, EA-SHORT ZRP menghasilkan nilai rata-rata yang lebih tinggi pada setiap percobaan dibandingkan dengan ZRP standar. Hal ini karena pada area yang luas, EA-SHORT ZRP berusaha memastikan suatu paket agar paket dapat sampai ke tujuan dengan mengalihkan rute paket ke tetangga terdekatnya (*load balancing*).

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil simulasi dan penjelasan yang telah dipaparkan pada BAB IV, dapat ditarik beberapa kesimpulan, yaitu:

1. Penerapan algoritma EA-SHORT pada jaringan MANET menggunakan protokol ZRP menunjukkan bahwa protokol *routing* EA-SHORT ZRP lebih baik dari pada protokol *routing* ZRP standar, dikarenakan dari kekuatan transmisi paket pada rute lebih baik karena adanya mekanisme untuk memilih rute dengan *node* yang memiliki energi lebih tinggi sehingga meningkatkan nilai *throughput*, PDR serta menurunkan nilai *average end to end delay*.
2. Hasil simulasi yang dilakukan, performa protokol *routing* ZRP dengan menggunakan algoritma EA-SHORT meningkatkan kualitas pada *throughput*. Hasil 50 *node* terhadap *throughput* pada EA-SHORT ZRP mengalami rata-rata peningkatan, 12,374 % dari ZRP standar. Untuk 100 *node*, peningkatan EA-SHORT ZRP sebesar 44,597 % dibandingkan ZRP standar. Hasil uji coba terhadap nilai *average end to end delay*, pada 50 *node* nilai EA-SHORT ZRP turun sebesar 20,063 %, pada *node* 100 EA-SHORT ZRP mengalami penurunan sebesar 8,375 % dari ZRP standar. Hasil PDR pada EA-SHORT ZRP dengan 50 *node* mengalami peningkatan sebesar 0,545 % dari ZRP standar, dan untuk 100 *node* EA-SHORT ZRP mengalami peningkatan sebesar 21,301 %.

5.2 Saran

Dari penelitian yang telah dilakukan terhadap penerapan algoritma EA-SHORT pada protokol *routing* ZRP, dari kesimpulan yang telah dipaparkan, maka peneliti dapat memberikan saran sebagai berikut:

1. Melakukan penelitian menggunakan protokol *routing* yang sama dengan parameter simulasi yang berbeda dan menggunakan jumlah *node* lebih banyak, *pause time* yang lebih tinggi, waktu simulasi yang berbeda serta model pergerakan *node* yang berbeda untuk mengetahui kombinasi parameter yang dapat menghasilkan nilai terbaik.
2. EA-SHORT dalam penerapannya dapat menyebabkan tingkat komputasi pada *routing* menjadi lebih tinggi karena harus membandingkan level energi pada *node*,

maka perlu dilakukan penelitian terhadap jumlah peningkatan beban komputasi ketika penerapan Algoritma EA-SHORT.

DAFTAR PUSTAKA

- [1] K. Raheja and S. K. Maakar, "A Survey on Different Hybrid Routing Protocols of MANET," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 4, pp. 5512–5516, 2014.
- [2] A. H. Jatmika, S. Djanali, and M. Husni, "Optimasi Routing pada Jaringan MANET," *Pros. Semin. Nas. Manaj. Teknol. XIII*, pp. 1–7, 2011.
- [3] E. R. Widasari and A. Bhawiyuga, "Analisis Perbandingan Kinerja Protokol AOMDV, DSDV, Dan ZRP Sebagai Protokol Routing Pada Mobile Ad-Hoc Network (MANET)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 10, pp. 3671–3680, 2018.
- [4] E. S. Gsr, N. Gayathri, A. Rai, and J. K. R, "Energy Aware Zone Routing Protocol for MANET," *Asian J. Pharm. Clin. Res.*, vol. 10, no. 1, pp. 437–440, 2017.
- [5] S. A. Sasongko and A. A. Zahra, "Analisis Performansi dan Simulasi Protokol ZRP (Zone Routing Protocol) pada MANET (Mobile Ad Hoc Network) dengan Menggunakan NS-2," pp. 1–8, 2012.
- [6] I. N. R. Hendrawan, "Simulasi Penggunaan Energi pada Protokol Routing OLSR dengan Simulator ns-3," *SNAPTI*, pp. 3–6, 2016.
- [7] C. Gui and P. Mohapatra, "SHORT: Self-Healing and Optimizing Routing Techniques for Mobile Ad Hoc Networks," *Proc. 4th ACM Int. Symp. Mob. ad hoc Netw. Comput.*, pp. 279–290, 2003.
- [8] Z. J. Haas and M. R. Pearlman, "On the performance of a routing protocol for the Reconfigurable Wireless Networks," *IEEE Veh. Technol. Conf.*, vol. 1, no. C, pp. 102–106, 1998.
- [9] B. S. Kusuma, D. Risqiwati, and D. R. Akbi, "Analisis Perbandingan Performansi Protokol Ad Hoc On-Demand Distance Vector dan Zone Routing Protocol Pada Mobile Ad Hoc Network," *Kinetik*, vol. 2, no. 3, pp. 165–175, 2017.
- [10] A. H. Networks, R. Asokan, A. M. Natarajan, and C. Venkatesh, "Quality-of-Service Routing Using Path and Power Aware Techniques in Mobile Ad Hoc Networks," *J. Comput. Syst. Networks, Commun.*, no. 2, p. 7, 2014.
- [11] M. Otmani and A. Ezzati, "Performance Evaluation of Energy Consumption of Ad hoc Routing Protocols Abstract :," *Int. J. Comput. Eng. Research*, vol. 04, no. 4, pp. 30–37, 2014.
- [12] R. M. M. I. Denatama, D. N. Perdana, "Analisis Perbandingan Kinerja Protokol Routing DSDV dan OLSR Untuk Perubahan Kecepatan Mobilitas pada Standar IEEE 802.11ah," *J. Infotel*, vol. 8, no. 2, pp. 100–106, 2016.
- [13] N. Raza, M. Umar Aftab, M. Qasim Akbar, O. Ashraf, and M. Irfan, "Mobile Ad-Hoc Networks Applications and Its Challenges," *Commun. Netw.*, vol. 08, no. 03, pp. 131–136, 2016.
- [14] R. D. Mardianty, A. H. Jatmika, and A. Y. Husodo, "Analisis Routing pada Protokol ZRP Menggunakan Metode Energy Aware dan Static Intersection Node di Jaringan VANET Studi Kasus Peta Jalan Kota Mataram," *J-Cosine*, 2018.

- [15] D. Ravilla and C. S. R. Putta, “Energy management in Zone Routing Protocol (ZRP),” *Adv. Intell. Soft Comput.*, vol. 167 AISC, no. VOL. 2, pp. 355–366, 2012.
- [16] S. Kaur, Sandeep; Kaur, “Analysis of Zone Routing Protocol in Manet,” *Int. J. Res. Eng. Technol.*, vol. 02, no. 09, pp. 520–524, 2013.
- [17] R. F. Sari, A. Syarif, and B. 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,” *Technol. Ser.*, vol. 12, no. 1, pp. 7–18, 2008.

Lampiran 1

1. Skrip AWK untuk *Throughput*

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

{
    event = $1
    time = $2
    node_id = $3
    pkt_size = $8
    level = $4

    # Store start time
    if (level == "AGT" && event == "s" && pkt_size >= 512) {
        if (time < startTime) {
            startTime = time
        }
    }

    # Update total received packets' size and store packets arrival time
    if (level == "AGT" && event == "r" && pkt_size >= 512) {
        if (time > stopTime) {
            stopTime = time
        }
        recvdSize += pkt_size
    }
}

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

2. Skrip AWK untuk *average end to end delay*

```
BEGIN {
    seqno=-1;
    sent=0;
}

{
    if ($4=="AGT"&&$1=="s"&&seqno<$6)
    {
        seqno=$6;
    }

    #end_end delay
    if ($4=="AGT"&&$1=="s")
    {
        start_time[$6]=$2;
    }
    else if (($4=="AGT") && ($1=="r"))
    {
        end_time[$6]=$2;
    }
    else if ($1=="D"&&$7=="udp")
    {
    }
}
```



```

    {
        end_time[$6]=-1;
    }
}

END{
for(i=0;i<=seqno;i++)
{
    if(end_time[i]>0)
    {
        tdelay[i]=end_time[i]-start_time[i];
        sent++;
    }
    else
    {
        tdelay[i]=-1;
    }
}

for(i=0;i<=seqno;i++)
{
    if(tdelay[i]>0)
    {
        delay=delay+tdelay[i];
    }
}

delay=delay/sent;

print "Average End-to-End Delay    = " delay " s";
}

```

3. Skrip AWK untuk PDR

```

BEGIN {
seqno          = -1;
no_of_packets  = 0;
droppedPackets = 0;
receivedPackets = 0;

    total_delay    = 0;
}

{
    if($4 == "AGT"){
        if($1 == "s"){
            seqno = $6;
            no_of_packets++;
            start_time[$6] = $2;
        }

        else if($1 == "r"){
            receivedPackets++;
            end_time[$6] = $2;           #end-to-end delay
        }
    }
}

```

```
END {
    for(i=0; i<=seqno; i++) {
        delay[i] = end_time[i] - start_time[i];
    if(delay[i] > 0)
        total_delay = total_delay + delay[i];
    }

    print "Packet Delivery Ratio : "
receivedPackets/no_of_packets*100"%";
    print "Average Delay          : "
total_delay/receivedPackets;
}
```

Lampiran 2

- Potongan *file setdest*

```
# nodes: 50, pause: 45.00, max speed: 1.00, max x: 1000.00, max y:
1000.00
#
$node_(0) set X_ 426.335600129749
$node_(0) set Y_ 306.313286701488
$node_(0) set Z_ 0.000000000000
$node_(1) set X_ 481.936933063642
$node_(1) set Y_ 909.427620740827
$node_(1) set Z_ 0.000000000000
$node_(2) set X_ 759.340065750930
$node_(2) set Y_ 530.939559087956
$node_(2) set Z_ 0.000000000000
$node_(3) set X_ 892.388993086505
$node_(3) set Y_ 180.361422830069
$node_(3) set Z_ 0.000000000000
$node_(4) set X_ 388.702950405472
$node_(4) set Y_ 691.679795940732
$node_(4) set Z_ 0.000000000000
$node_(5) set X_ 85.154186168702
$node_(5) set Y_ 76.529052322275
$node_(5) set Z_ 0.000000000000
```