

**PERBANDINGAN KINERJA PROTOKOL ZRP (ZONE ROUTING
PROTOCOL) DAN HWMP (HYBRID WIRELESS MESH
PROTOCOL) PADA MANET (MOBILE AD HOC NETWORK)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Miftaqul Novandi
NIM: 155150200111146



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2020**

PENGESAHAN

PERBANDINGAN KINERJA PROTOKOL ZRP (ZONE ROUTING PROTOCOL) DAN
HWMP (HYBRID WIRELESS MESH PROTOCOL) PADA MANET (MOBILE AD HOC
NETWORK)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Miftaql Novandi

NIM: 155150200111146

Skripsi ini telah diuji dan dinyatakan lulus pada
03 Januari 2020

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Primantara Hari Trisnawan, M.Sc.

Reza Andria Siregar, S.T., M.Kom.

NIP: 19680912 199403 1 002

NIP: 19790621 200604 1 003

Mengetahui

Ketua Jurusan Teknik Informatika

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

NIP: 19710518 200312 1 001



PERNYATAAN ORISINALITAS

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

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

Malang, 03 Januari 2020

Miftaql Novandi

NIM: 155150200111146

UNIVERSITAS BRAWIJAYA



Penulis mengakui bahwa dalam penyusunan skripsi ini masih terdapat banyak kesalahan dan kekurangan, sehingga penulis membutuhkan kritik dan saran yang membangun dari teman-teman. Akhir kata penulis berharap skripsi ini dapat bermanfaat dan terimakasih.

Malang, 03 Januari 2020

Penulis

miftaqlnovandi@student.ub.ac.id



ABSTRAK

Miftaqul Novandi, Perbandingan Kinerja Protokol ZRP (Zone Routing Protocol) dan HWMP (Hybrid Wireless Mesh Protocol) Pada MANET (Mobile Ad hoc Network)

Pembimbing: Ir.Primantara Hari Trisnawan, M.Sc dan Reza Andria Siregar, S.T., M.Kom.

MANET merupakan jaringan *wireless* yang bersifat dinamis, dikarenakan setiap *node* dapat bergerak secara bebas sehingga topologi jaringan seringkali berubah tanpa bisa diprediksi. Kondisi pada topologi yang berubah-ubah tersebut menyebabkan pengaruh pada kinerja protokol *routing* yang digunakan pada MANET. *Mobile node* yang dimaksud dalam MANET adalah sebuah perangkat bergerak yang dapat berupa *tablet*, *laptop*, dan *smartphone* yang dapat menjalankan proses *routing* dan meneruskan komunikasi paket data dari satu *mobile node* menuju *mobile node* lainnya. MANET mempunyai beberapa jenis protokol *routing*, yaitu proaktif, reaktif, dan *hybrid*. Dalam penelitian sebelumnya telah dilakukan analisis mengenai kinerja dari protokol ZRP dan HWMP yang merupakan protokol *routing hybrid* (gabungan protokol *routing* reaktif dan proaktif), dengan ruang lingkup yang digunakan pada WMN (*Wireless Mesh Network*). WMN memiliki perangkat jangkauan yang tidak begitu luas dan topologi yang digunakan dalam keadaan statis. Penelitian ini menggunakan ruang lingkup MANET, memusatkan pada perangkat pengguna, mobilitas yang dibangun tanpa infrastruktur, dan kemampuan *ad-hoc*. Topologi dalam jaringan yang bergerak secara dinamis serta pengimplementasian pada ruang lingkup yang luas mampu mempengaruhi kinerja protokol ZRP dan HWMP sebagai *routing* protokol yang hendak disimulasikan dalam penelitian ini. Dengan protokol *routing* ini akan diamati perubahan yang terjadi berdasarkan parameter pengujian seperti *throughput*, *packet delivery ratio (PDR)*, *normalized routing load*, *end-to-end delay* dan *convergence time*. Skenario pengujian dijalankan dengan menggunakan *node* statis dan dinamis, untuk mengetahui protokol *routing* mana yang lebih mendominasi dengan parameter uji yang digunakan, disimulasikan pada Network Simulator 2.35 (NS-2.35). Hasil nilai simulasi sudah diperoleh dengan menyimpulkan bahwa protokol *routing* ZRP lebih mendominasi untuk kelima parameter uji yang dilakukan pada ruang lingkup topologi dinamis, sedangkan untuk topologi statis lebih didominasi HWMP.

Kata kunci: MANET, ZRP, HWMP, network simulator 2, protokol *routing*

ABSTRACT

Miftaql Novandi, Perbandingan Kinerja Protokol ZRP (Zone Routing Protocol) dan HWMP (Hybrid Wireless Mesh Protocol) Pada MANET (Mobile Ad hoc Network)

Supervisors: Ir.Primantara Hari Trisnawan, M.Sc dan Reza Andria Siregar, S.T., M.Kom.

MANET is a dynamic wireless network, because each node can move freely so the topology network often changes unpredictably. Conditions in the changing topology due to an effect on the performance of the routing protocol used on MANET. A mobile device in MANET that is implemented as a mobile node can be a smartphone, laptop, or tablet. Then the device acts as a node in the simulation can communicate data packets between one node with another node. MANET's routing protocols have several types, namely reactive, proactive and hybrid (a combination of reactive and proactive routing protocols). Previous research on the performance of the ZRP and HWMP protocols which is hybrid routing protocols, using the scope of the WMN (Wireless Mesh Network). WMN has a range of a devices that's not wide and topology that is used in a static state. This research using the scope of MANET, focusing on user devices, mobility built without infrastructure, and ad-hoc capabilities. The topology in a network that moves dynamically and the implementation in a broad scope is able to influence the performance of the ZRP and HWMP protocols as the routing protocol that we want to simulate in this study. With this routing protocol changes will be observed that occur based on testing parameters such as throughput, packet delivery ratio (PDR), normalized routing load, end-to-end delay and convergence time. The scenario test is done using static and dynamic nodes, and Network Simulator 2 (NS-2) as a test simulation. The simulation results that have been obtained are concluded that ZRP routing protocol is more dominant for the five test parameters carried out in the dynamic topology scope, while for the static topology is more dominated by HWMP.

Keywords: MANET, ZRP, HWMP, network simulator 2, routing protocol

DAFTAR ISI

PENGESAHAN	viii
PERNYATAAN ORISINALITAS	viii
PRAKATA	viii
ABSTRAK	vi
ABSTRACT	viii
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	3
1.3 Tujuan	4
1.4 Manfaat	4
1.5 Batasan masalah	4
1.6 Sistematika pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka	6
2.2 <i>Mobile AD-HOC NETWORK (MANET)</i>	8
2.3 <i>Protokol Routing MANET</i>	8
2.4 <i>Zone Routing Protocol (ZRP)</i>	9
2.5 <i>Hybrid Wireless Mesh Protocol (HWMP)</i>	13
2.6 Network Simulator (NS2)	18
2.6.1 Langkah Simulasi Pada NS2	18
2.7 <i>AWK Programming</i>	20
2.8 Parameter dalam kinerja protokol <i>routing</i>	20
2.8.1 <i>Throughput</i>	20
2.8.2 <i>Packet Delivery Ratio (PDR)</i>	20
2.8.3 <i>End-to-end Delay</i>	21
2.8.4 <i>Normalilized Routing Load</i>	21
2.8.5 <i>Convergence Time</i>	21



BAB 3 METODOLOGI	23
3.1 Studi Literatur	24
3.2 Analisis Kebutuhan	24
3.3 Perancangan dan Implementasi Lingkungan Pengujian	25
3.3.1 Perancangan Lingkungan Pengujian	25
3.3.2 Implementasi Lingkungan Pengujian	26
3.4 Pengujian dan Pengambilan Data	26
3.5 Hasil dan Analisis	26
3.5.1 Analisis Pengaruh Variasi Kepadatan <i>Node</i>	26
3.6 Kesimpulan dan Saran	26
BAB 4 PERANCANGAN DAN IMPLEMENTASI LINGKUNGAN PENGUJIAN	27
4.1 Perancangan Penelitian	27
4.1.1 Perancangan Protokol <i>Routing</i>	27
4.1.2 Perancangan Lingkungan Pengujian	27
4.1.3 Perancangan Konfigurasi Protokol <i>Routing</i>	28
4.1.3.1 Perancangan konfigurasi parameter pengujian.....	28
4.1.3.2 Perancangan skenario pengujian	28
4.1.3.3 Perancangan konfigurasi topologi jaringan	30
4.1.3.4 Perancangan konfigurasi pergerakan <i>node</i>	31
4.1.3.5 Perancangan konfigurasi pengiriman paket	32
4.1.3.6 Perancangan pemrosesan data <i>output</i>	32
4.2 Implementasi Lingkungan Pengujian	32
4.2.1 Implementasi Protokol <i>Routing</i>	32
4.2.1.1 Implementasi protokol <i>routing</i> ZRP	33
4.2.1.2 Implementasi protokol <i>routing</i> HWMP	33
4.2.2 Implementasi Pada Network Simulator 2	33
4.2.2.1 Implementasi parameter dan skenario pengujian.....	34
4.2.2.2 Implementasi topologi jaringan	35
4.2.2.3 Implementasi pergerakan <i>node</i>	36
4.2.2.4 Implementasi pengiriman paket	37
4.2.2.5 Implementasi pemrosesan data <i>output</i>	39
4.3 Pengujian	39

4.3.1 Skenario Pengujian	39
4.3.2 Eksekusi Simulasi Protokol <i>Routing</i> MANET	40
4.4 Pengambilan Data	41
4.4.1 Pengambilan Data <i>Packet Delivery Ratio</i>	42
4.4.2 Pengambilan Data <i>Throughput</i>	43
4.4.3 Pengambilan Data <i>End to end Delay</i>	43
4.4.4 Pengambilan Data <i>Normalized Routing Load</i>	43
4.4.5 Pengambilan Data <i>Convergence Time</i>	44
BAB 5 HASIL DAN PEMBAHASAN.....	45
5.1 Pengaruh Variasi Kepadatan <i>Node</i> Protokol <i>Routing</i> ZRP dan HWMP Pada Topologi Statis dan Dinamis.....	45
5.1.1 <i>Throughput</i>	45
5.1.2 <i>Packet Delivery Ratio</i>	47
5.1.3 <i>Normalized Routing Load</i>	49
5.1.4 <i>End-to-end Delay</i>	50
5.1.5 <i>Convergence Time</i>	52
BAB 6 KESIMPULAN	55
6.1 Kesimpulan.....	55
6.2 Saran	56
DAFTAR PUSTAKA.....	58
LAMPIRAN.....	60



DAFTAR TABEL

Tabel 2.1 Perbandingan Topik Penelitian Terdahulu	6
Tabel 4.1 Perancangan konfigurasi parameter pengujian	29
Tabel 4.2 Perancangan skenario pengujian	30
Tabel 4.3 Implementasi parameter pengujian ZRP	34
Tabel 4.4 Implementasi parameter pengujian HWMP	34
Tabel 4.5 Implementasi topologi jaringan <i>node</i> dinamis	35
Tabel 4.6 Implementasi topologi jaringan <i>node</i> statis	36
Tabel 4.7 Inisialisasi jarak oleh objek god	37
Tabel 4.8 Implementasi pengiriman paket	38
Tabel 4.9 Implementasi model pengiriman paket	38
Tabel 4.10 Implementasi pemrosesan data <i>output</i>	39
Tabel 5.1 Hasil pengujian <i>throughput</i> ZRP dan HWMP	45
Tabel 5.2 Hasil pengujian <i>packet delivery ratio</i> ZRP dan HWMP	47
Tabel 5.3 Hasil pengujian <i>noermalized routing load</i> ZRP dan HWMP	49
Tabel 5.4 Hasil pengujian <i>end-to-end delay</i> ZRP dan HWMP	51
Tabel 5.5 Hasil pengujian <i>convergence time</i> ZRP dan HWMP	52



DAFTAR GAMBAR

Gambar 2.1 Skema ilustrasi jaringan MANET.....8

Gambar 2.2 Jenis protokol routing pada MANET 9

Gambar 2.3 Zona *routing node* S dengan jumlah *hop* dua 10

Gambar 2.4 Zona *routing node* S dengan jumlah *hop* dua dalam IRP 10

Gambar 2.5 *Route request packet* 11

Gambar 2.6 Contoh kerja IERP 12

Gambar 2.7 *Path Request* HWMP 14

Gambar 2.8 *Path Reply* HWMP 15

Gambar 2.9 Proses penyebaran pesan PERR *routing* reaktif HWMP 16

Gambar 2.10 Mekanisme *Proactive* PREQ 17

Gambar 2.11 Mekanisme *Proactive* RANN 17

Gambar 2.12 Skema rancangan simulasi pada NS2 19

Gambar 3.1 Tahap Metodologi Penelitian 23

Gambar 4.1 Skema alur tahapan perancangan lingkungan pengujian pada NS2, 28

Gambar 4.2 Screenshot topologi jaringan dengan 30 *node* ZRP 31

Gambar 4.3 Screenshot topologi jaringan dengan 30 *node* HWMP 31

Gambar 4.4 Pembuatan mobilitas *node* 36

Gambar 4.5 Hasil *file* pembuatan mobilitas *node* 37

Gambar 4.6 Hasil eksekusi *file script* *zrp.tcl* 40

Gambar 4.7 Hasil *output network animation file* 40

Gambar 4.8 Hasil *output network trace file* 41

Gambar 4.9 Struktur *output network trace file* 41

Gambar 4.10 Hasil eksekusi program AWK *packet delivery ratio* 42

Gambar 4.11 Hasil eksekusi program AWK *throughput* 43

Gambar 4.12 Hasil eksekusi program AWK *end-to-end delay* 43

Gambar 4.13 Hasil eksekusi program AWK *normalized routing load* 43

Gambar 4.14 Hasil eksekusi program AWK *convergence time* 44

Gambar 5.11 Grafik *throughput* perbandingan kinerja ZRP dan HWMP 46

Gambar 5.12 Grafik *packet delivery ratio* perbandingan kinerja ZRP dan HWMP 48



Gambar 5.13 Grafik *normalized routing load* perbandingan kinerja ZRP dan HWMP

Gambar 5.14 Grafik *end-to-end delay* perbandingan kinerja ZRP dan HWMP

Gambar 5.15 Grafik *convergence time* perbandingan kinerja ZRP dan HWMP



BAB 1 PENDAHULUAN

1.1 Latar belakang

Mobile Ad hoc Network (MANET) merupakan kumpulan dari beberapa *wireless node* yang dapat di atur secara dinamis dimana saja dan kapan saja tanpa bergantung pada infrastruktur jaringan yang ada. MANET menurut Kukreja (2014) merupakan komunikasi antar sekumpulan *mobile node* yang tanpa kontrol terpusat dan infrastruktur yang tetap pada suatu jaringan nirkabel yang memiliki topologi dinamis. MANET merupakan jaringan *wireless* yang bersifat dinamis, hal ini dikarenakan setiap *node* dapat bergerak secara bebas sehingga topologi jaringan seringkali berubah tanpa dapat diprediksi. Kondisi pada topologi yang berubah-ubah tersebut akan mempengaruhi kinerja protokol *routing* pada MANET. *Mobile node* yang dimaksud dalam MANET ini adalah sebuah perangkat bergerak yang dapat berupa *tablet*, *laptop*, dan *smartphone*. Dalam hal ini, *mobile node* dalam MANET mampu menjalankan proses *routing* dan meneruskan komunikasi paket data dari satu *mobile node* menuju *mobile node* lainnya, istilah ini disebut *multi-hop*. Seperti penjelasan di atas, MANET memiliki keunggulan yaitu dapat berjalan pada topologi yang dinamis. Terlepas dari keunggulan yang dimiliki, MANET memiliki permasalahan efektifitas kinerja saat diterapkan di beberapa wilayah, terutama pada cakupan wilayah yang luas. Adapun beberapa faktor yang berpengaruh terhadap *routing* pada MANET adalah luas area yang digunakan, jumlah *mobile node* beserta kecepatan mobilitasnya. *Mobile node* yang berjalan pada MANET dapat bergerak secara bebas dan menyebabkan jaringan berubah sewaktu-waktu. Luas area pada MANET yang tidak sesuai (terlalu luas atau sempit) dan tidak diimbangi dengan jumlah *node* yang digunakan dalam topologi menyebabkan perubahan intensitas rute yang tinggi.

MANET mempunyai beberapa tipe atau jenis protokol *routing* yang dapat digunakan, yaitu protokol *routing* proaktif contohnya OLSR, WRP dan DSDV, protokol *routing* reaktif contohnya AODV, DYMO, DSR dan AOMDV dan protokol *routing hybrid* contohnya ZRP, HRPLS dan HWMP yang dalam implementasinya masing-masing protokol *routing* tersebut memiliki kinerja yang berbeda-beda satu dengan yang lain. Merujuk dari beberapa penelitian sebelumnya oleh Sandeep Gupta et.al (2015) yang berjudul "*Performance Comparison of Proactive Routing Protocols:OLSR, DSDV, WRP*" menjelaskan bahwa protokol *routing* proaktif adalah protokol yang selalu memperbarui informasi *routing* di dalam jaringan dan sangat efektif digunakan apabila letak *node* sumber dan *node* tujuan saling berdekatan atau tidak terlalu jauh. Kemudian untuk protokol *routing* reaktif merujuk pada penelitian oleh Shalini Kushwaha et al.(2015) yang berjudul "*Performance Analysis of AODV, CBRP, AOMDV, DSR MANET Routing Protocols using NS2 Simulation*" menjelaskan bahwa protokol *routing* reaktif merupakan protokol yang melakukan pencarian rute disaat rute tersebut dibutuhkan dan sangat efektif bila digunakan jika *node* sumber dan *node* tujuan memiliki jarak yang jauh atau tidak saling berdekatan. Berangkat dari perbedaan efektifitas masing-masing protokol *routing* tersebut maka dipilihlah protokol *routing hybrid* dalam penelitian ini yang

merupakan penggabungan antara protokol *routing* proaktif dan reaktif dan bisa efektif pada kondisi topologi apapun.

Dalam penelitian ini mengambil dua contoh protokol *routing hybrid*, yaitu *Zone Routing Protocol (ZRP)* dan *Hybrid Wireless Mesh Protocol (HWMP)*. Kedua protokol tersebut sama-sama bekerja pada MANET, memiliki komponen yang sama yaitu proaktif dan reaktif, tetapi memiliki alur atau proses *routing* yang berbeda. Dari penelitian yang dipelajari yakni oleh Bejar Nicklas (2015) berjudul "*Zone Routing Protocol (ZRP)*", dijelaskan bahwa pada *routing* protokol ZRP dilakukan dengan beberapa sub protokol di dalamnya yaitu *intrazone routing protocol (IARP)* yang menyediakan pembaruan rute pada zona lokal, dan *interzone routing protocol (IERP)* yang menyediakan pembaruan rute pada zona luar. Pada pembaruan rute zona luar ini memiliki keistimewaan, yaitu pada saat proses pencarian rutenya. Untuk permintaan satu rute dapat mengembalikan *route reply* lebih dari satu serta dapat digunakan untuk menentukan rute terbaik berdasarkan *link cost* yang paling rendah. Pada penelitian tersebut juga dijelaskan bahwa protokol *routing* ZRP menggunakan algoritma *dijkstra* sebagai penghitungan jalur terpendek. Untuk penelitian terhadap HWMP yang dilakukan oleh Suryani et.al (2012) berjudul *Analisis Performansi Routing Hybrid Wireless Mesh Protocol (HWMP) Pada Wireless Mesh Network (WMN) Berdasarkan Standar IEEE 802.11S* dijelaskan mengenai pencarian rute yang dilakukan menggunakan mekanisme nomor urut untuk menjaga konektifitas bebas *loop* setiap saat. Bagian reaktif pada HWMP mengikuti konsep umum dari protokol *routing Ad Hoc On Demand Distance Vector (AODV)* dengan menggunakan metode *distance vector, well-known route discovery* dengan *route request* dan *route reply*. Sedangkan untuk proaktif menggunakan *proactive tree building mode* yang akan membuat rute atau *path tree* dari *root* ke semua *node* yang ada dalam topologi jaringan. Dalam bagian ini terdapat dua sub-mekanisme dari pembuatan atau pembentukan informasi *routing*, yaitu mekanisme *proactive PREQ* dan mekanisme *proactive RANN (root announcement)*.

Penelitian ini lebih difokuskan kepada masalah yang sering ditemukan pada jaringan dalam MANET yang memiliki karakteristik kondisi jaringan yang dinamis pada suatu cakupan tertentu sehingga pengaruh perubahan jaringan tersebut menyebabkan trafik jaringan menjadi besar sehingga menyebabkan tabrakan paket data dan perlu dilakukan pencarian rute (*routing*) jalur yang optimal komunikasi antar *node*. Kondisi yang dinamis tersebut bisa merupakan akibat dari kecepatan atau jumlah variasi kepadatan *node* yang digunakan pada luas area yang besar. Penelitian yang dilakukan sebelumnya oleh Benny Pandopotan Sinaga (2013) berjudul *Analisis kinerja protokol ZRP dan HWMP pada Wireless Mesh Network* dijelaskan mengenai protokol *routing* ZRP dan HWMP yang difokuskan pada jaringan *wireless mesh network (WMN)* dan didapatkan hasil bahwa protokol *routing* ZRP mendominasi atas HWMP berdasarkan lima parameter uji yang dilakukan yaitu *packet delivery ratio (PDR)*, *packet loss ratio (PLR)*, *throughput*, *end to end delay*, dan *routing overhead*.

Dari penelitian yang dilakukan oleh peneliti sebelumnya tersebut, maka penulis mencoba mengaplikasikan protokol *routing* yang sama yaitu protokol *routing* ZRP dan HWMP tetapi pada ruang lingkup MANET (*mobile Ad-hoc Network*) yang bertujuan untuk mengetahui apakah protokol *routing* ZRP tetap mendominasi walaupun berbeda ruang lingkup yang digunakan. WMN (*Wireless Mesh Network*) dan jaringan ad-hoc mobile (Mobile ad-hoc network atau MANET) menggunakan konsep komunikasi yang sama antar nodenya, namun memiliki beberapa perbedaan. MANET memusatkan pada perangkat pengguna, mobilitas yang dibangun tanpa infrastruktur, dan kemampuan *ad-hoc*. WMN memusatkan pada perangkat statis (infrastruktur), kehandalan, dan kapasitas jaringan (Michael Bahr, 2006). WMN menggunakan topologi *node* statis, dimana posisi variasi kepadatan *node* dalam area simulasi adalah tetap atau tidak bergerak antara *node* satu dengan yang lainnya. MANET menggunakan topologi *node* dinamis, dimana posisi variasi kepadatan *node* dalam area simulasi dapat bergerak antara *node* satu dengan yang lainnya. Dalam hal ini, jelas terdapat perbedaan hasil terhadap parameter uji yang digunakan nantinya dalam menentukan protokol mana yang lebih mendominasi. Penelitian sebelumnya yang telah dilakukan oleh Benny Pandopotan Sinaga, belum dibahas mengenai pengaruh kondisi *node* yang digunakan, baik *node static* atau *node mobile* yang jelas akan memberikan pengaruh terhadap protokol mana yang lebih mendominasi terhadap parameter pengujian. Dari perbedaan konsep dan cara kerja yang digunakan oleh MANET dan WMN yang telah dijelaskan, memberikan beberapa pengaruh terhadap efektifitas kinerja masing-masing protokol *routing* ZRP dan HWMP. Untuk mengetahui efektifitas kinerja terbaik antara kedua protokol *routing* diantara keduanya dalam hal pengaruh penggunaan topologi *node* yang digunakan pada variasi kepadatan *node* dengan kondisi area yang luas pada MANET, maka dilakukan penelitian ini sebagai jawaban atas permasalahan tersebut. Penelitian dilakukan dengan analisis beberapa parameter yaitu *throughput*, *packet delivery ratio (PDR)*, *normalized routing load*, *end-to-end delay* dan *convergence time*. Analisis kinerja ini dilakukan menggunakan Network Simulator 2.35 (NS2.35) sebagai simulator penelitian.

1.2 Rumusan masalah

Berdasarkan latar belakang yang dibahas, maka rumusan masalah yang dapat dikaji adalah sebagai berikut :

1. Bagaimana pengaruh variasi kepadatan *node* terhadap kinerja protokol *routing* ZRP dan HWMP dengan area yang cukup luas dalam MANET ?
2. Bagaimana perbandingan kinerja protokol *routing* ZRP dan HWMP pada MANET menggunakan NS2 berdasarkan parameter *throughput*, *packet delivery ratio (PDR)*, *normalized routing load*, *end-to-end delay* dan *convergence time* ?
3. Bagaimana pengaruh penggunaan topologi statis dan dinamis jika diterapkan pada MANET ?

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Mengetahui perbandingan kinerja protokol *routing* ZRP dan HWMP berdasarkan parameter *throughput*, *packet delivery ratio* (PDR), *normalized routing load*, *end-to-end delay* dan *convergence time*.
2. Mengetahui pengaruh variasi kepadatan *node* terhadap kinerja protokol *routing* ZRP dan HWMP dengan luas area yang cukup besar pada MANET.
3. Mengetahui pengaruh penggunaan topologi statis dan dinamis dalam penerapannya pada MANET.

1.4 Manfaat

Penelitian ini menghasilkan kinerja perbandingan antara protokol *routing* ZRP dan HWMP. Dengan adanya hasil analisa ini, manfaat yang di dapat adalah memberikan gambaran mengenai kinerja dari dua jenis protokol *routing hybrid*, yaitu ZRP dan HWMP yang dijalankan pada ruang lingkup yang cukup luas dan jumlah *node* yang berbeda berdasarkan parameter *throughput*, *packet delivery ratio* (PDR), *normalized routing load*, *end-to-end delay* dan *convergence time*. Selain itu penelitian ini juga dapat digunakan sebagai dasar dari penelitian selanjutnya karena masih banyak aspek dalam penelitian ini yang dapat ditingkatkan. Manfaat berikutnya yaitu membantu penulis untuk lebih memahami protokol ZRP dan HWMP dan komponen di dalamnya.

1.5 Batasan masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka pada penelitian ini dibatasi dalam hal :

1. Protokol yang digunakan adalah ZRP dan HWMP yang sama-sama termasuk dalam protokol *routing hybrid*.
2. Tidak membahas mengenai masalah keamanan jaringan dan konsumsi energi.
3. Parameter kinerja yang digunakan adalah *throughput*, *packet delivery ratio* (PDR), *normalized routing load*, *end-to-end delay* dan *convergence time*.
4. *Transport agent* yang diterapkan pada simulasi adalah UDP dengan model *traffic CBR*.
5. Simulator yang digunakan adalah *Network Simulator (NS-2.35)* yang dilakukan di OS Ubuntu 18.04.

1.6 Sistematika pembahasan

Sistematika penulisan ditujukan untuk memberikan gambaran dan uraian dari penulisan skripsi secara garis besar yang meliputi beberapa bab, sebagai berikut :

BAB 1 PENDAHULUAN

Pada bab ini akan dijelaskan mengenai latar belakang pembuatan skripsi, rumusan masalah, tujuan dan manfaat, serta sistematika penulisan.

BAB 2 LANDASAN KEPUSATAKAAN

Landasan kepustakaan berisi uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan topik skripsi dan pertanyaan penelitian. Landasan kepustakaan ini digunakan sebagai perancangan, dan analisis hasil pada penelitian ini.

BAB 3 METODOLOGI PENELITIAN

Dalam bab metode penelitian dijelaskan langkah kerja dalam analisis perbandingan kinerja protokol ZRP dan HWMP pada MANET menggunakan NS2. Langkah-langkah yang dijelaskan disini meliputi studi literatur, perancangan sistem, pembahasan dan analisis serta pengambilan kesimpulan dan saran.

BAB 4 PERANCANGAN SISTEM

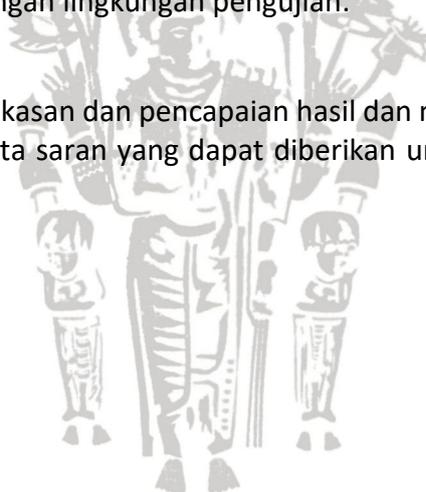
Pada bab ini menjelaskan tentang perancangan lingkungan pengujian berupa parameter dan skenario yang nanti digunakan untuk pengujian.

BAB 5 HASIL DAN PEMBAHASAN

Bab hasil dan pembahasan menjelaskan proses untuk menguji analisis yang telah dilakukan untuk memastikan apakah telah sesuai dengan kebutuhan yang disusun pada perancangan lingkungan pengujian.

BAB 6 PENUTUP

Pada bab ini berisi ringkasan dan pencapaian hasil dan menjawab pertanyaan dari rumusan masalah. Serta saran yang dapat diberikan untuk pengembangan lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

Pada penelitian ini landasan kepustakaan sebagai referensi berasal dari beberapa jurnal dan literatur terkait analisis protokol *hybrid* ZRP dan HWMP, MANET, dan metodologi pengujian. Jurnal dan literatur tersebut meliputi analisis protokol *routing* pada MANET, HWMP, dan ZRP. Landasan kepustakaan yang ada akan menjadi dasar pengembangan dan pengujian analisis perbandingan kinerja protokol *routing* ZRP dan HWMP pada MANET dengan simulator NS2.

2.1 Kajian Pustaka

Berikut adalah daftar penelitian terhadap protokol *routing hybrid* yang telah dilakukan sebelumnya dengan topik yang serupa dan perbandingannya terhadap penelitian yang dilakukan.

Tabel 2.1 Perbandingan Topik Penelitian Terdahulu

No	Penulis	Judul dan Tahun Penelitian	Persamaan	Perbedaan	
				Terdahulu	Saat ini
1	Ibikunle Frank et al.	"Performance Analysis of Hybrid Protocols in MANETs".(2013)	Menganalisis jenis protokol yang sama yaitu ZRP menggunakan NS2 pada MANET	Protokol <i>routing</i> yang digunakan ADV, ZRP, dan SHARP. Parameter pengujian yaitu <i>throughput</i> , <i>packet delivery ratio</i> (PDR) dan <i>end-to-end delay</i>	Protokol <i>routing</i> yang digunakan ZRP dan HWMP. Parameter pengujian yaitu <i>throughput</i> , <i>packet delivery ratio</i> (PDR), <i>normalized routing load</i> , <i>end-to-end delay</i> dan <i>convergence</i> .
2	Benny Sahat Pandapotan Sinaga	"Analisis kinerja protokol ZRP dan HWMP pada Wireless Mesh Network".(2013)	Menganalisa dua jenis protokol yaitu ZRP dan HWMP menggunakan NS2	Pengimplementasian pada <i>Wireless Mesh Network</i> (WMN) yang menggunakan topologi <i>node</i> statis parameter pengujian yang digunakan <i>packet delivery ratio</i> (PDR), <i>packet loss ratio</i> , <i>routing overhead</i> , <i>throughput</i> , dan <i>average end-to-end delay</i> .	Pengimplementasian pada <i>Mobile Adhoc Network</i> (MANET) yang menggunakan dua jenis topologi <i>node</i> , yaitu statis dan dinamis, parameter pengujian yaitu <i>throughput</i> , <i>packet delivery ratio</i> (PDR), <i>normalized routing load</i> , <i>end-to-end delay</i> dan <i>convergence time</i> .

Tabel 2.1 Perbandingan Topik Penelitian Terdahulu (lanjutan)

No	Penulis	Judul dan Tahun Penelitian	Persamaan	Perbedaan	
				Terdahulu	Saat ini
3	Fatkhurrozi,	"Analisis Perbandingan Kinerja Protokol AOMDV, DSDV, Dan ZRP Sebagai Protokol Routing Pada Mobile Ad-Hoc Network (MANET)"	Menganalisis jenis protokol yang sama yaitu ZRP menggunakan NS2 pada MANET.	Menggunakan tiga protokol <i>routing</i> , yaitu DSDV untuk proaktif, AOMDV untuk reaktif dan ZRP untuk <i>hybrid</i> . Parameter pengujian yaitu <i>throughput</i> , <i>packet delivery ratio</i> (PDR), <i>normalized routing load</i> , <i>end-to-end delay</i> .	Protokol <i>routing</i> yang digunakan ZRP dan HWMP yang sama-sama merupakan <i>routing hybrid</i> . Parameter pengujian yaitu <i>throughput</i> , <i>packet delivery ratio</i> (PDR), <i>normalized routing load</i> , <i>end-to-end delay</i> dan <i>convergence time</i> .

Dari daftar penelitian terhadap protokol *routing* hybrid yang dijadikan sebagai kajian pustaka pada Tabel 2.1 di atas dijelaskan bahwa pada penelitian yang dilakukan oleh Ibikunle Frank et al (2013) yang berjudul *Performance Analysis of Hybrid Protocols in MANETs* mengenai tiga kinerja protokol *hybrid* yang digunakan yaitu ADV, ZRP dan SHARP menggunakan simulator NS2 dengan parameter pengujian *packet delivery ratio* (PDR), *packet loss ratio*, *routing overhead*, *throughput*, dan *average end-to-end delay*. Pada penelitian tersebut digunakan luas area yang lebih kecil yaitu 400x400 meter, maksimum kecepatan 10 m/s, dengan variasi kepadatan *node* yang digunakan untuk masing-masing protokol adalah 50, 75 dan 100. Kesimpulan yang didapatkan dari penelitian tersebut adalah protokol ADV menunjukkan kinerja yang lebih baik diantara dua protokol yang digunakan. Kemudian untuk penelitian kedua yang dijadikan sebagai kajian pustaka yang dilakukan oleh Benny Sahat Pandapotan Sinaga (2013) berjudul *Analisis kinerja protokol ZRP dan HWMP pada Wireless Mesh Network*, dijelaskan mengenai kinerja protokol ZRP dan HWMP menggunakan simulator NS2, tetapi diimplementasikan pada *wireless mesh network* (WMN) dengan menggunakan topologi *node* statis atau tidak bergerak antara *node* satu dengan yang lainnya. Parameter pengujian yang digunakan adalah *packet delivery ratio* (PDR), *packet loss ratio*, *routing overhead*, *throughput*, dan *average end-to-end delay*. Dari penelitian yang dilakukan didapatkan secara keseluruhan ZRP mendominasi HWMP untuk kelima parameter uji yang digunakan dalam penelitian. Untuk penelitian selanjutnya oleh Fatkhurrozi yang berjudul "*Analisis Perbandingan Kinerja Protokol AOMDV, DSDV, Dan ZRP Sebagai Protokol Routing Pada Mobile Ad-Hoc Network (MANET)*" menjelaskan mengenai perbandingan kinerja pada protokol proaktif menggunakan DSDV, reaktif menggunakan AOMDV dan *hybrid* menggunakan ZRP, dengan parameter pengujian adalah *throughput*, *packet*



delivery ratio (PDR), *normalized routing load*, *end-to-end delay* dan didapatkan protokol AOMDV lebih mendominasi untuk keempat parameter kinerja dibandingkan dengan dua protokol lainnya.

2.2 Mobile AD-HOC NETWORK (MANET)

Mobile Ad-Hoc Network (MANET) adalah jaringan nirkabel yang berkemampuan dalam membangun konfigurasi sendiri. Jaringan pada MANET terdiri dari sekumpulan perangkat bergerak (*mobile node*) seperti *laptop*, *smartphone*, dan *tablet* yang dapat berkirir informasi secara nirkabel seperti yang ditunjukkan pada Gambar 2.1. Dalam jaringan MANET, setiap *mobile node* dapat berperan sebagai *mobile router*, dan dapat meneruskan paket data menuju *mobile node* lainnya secara nirkabel tanpa harus terhubung dengan perangkat *router* maupun *access point* (Raja, 2014). Di dalam penerapannya MANET memiliki beberapa keuntungan, yaitu (Kaur, 2013):

1. Jaringan dapat diperluas ditempat-tempat yang tidak dapat dijangkau kabel.
2. Jaringan nirkabel memberikan lebih banyak fleksibilitas dan kemudahan dalam beradaptasi dengan perubahan konfigurasi jaringan.
3. Pengaturan sistem jaringan nirkabel yang mudah dan cepat serta dapat mengurangi kebutuhan dalam penggunaan kabel.



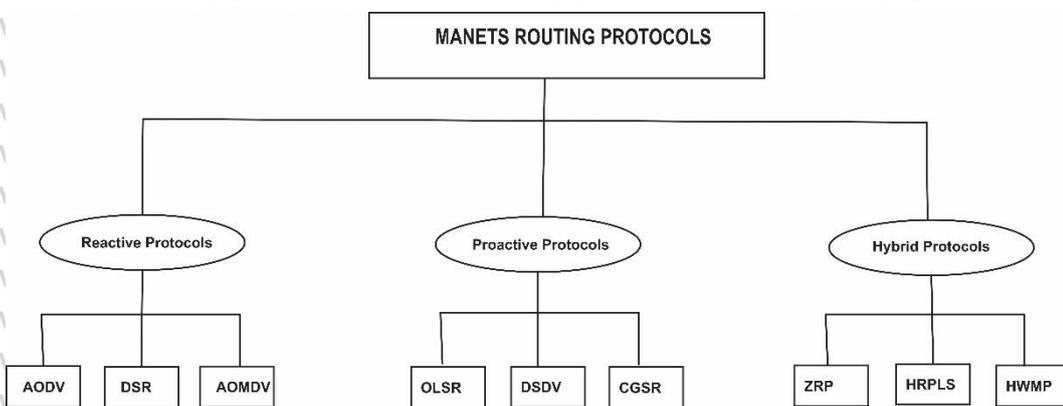
Gambar 2.1 Skema ilustrasi jaringan MANET

Sumber: Victor (2013)

2.3 Protokol Routing 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 seperti yang ditunjukkan pada Gambar 2.2, yaitu:

1. Protokol *routing* reaktif
2. Protokol *routing* proaktif
3. Protokol *routing hybrid*



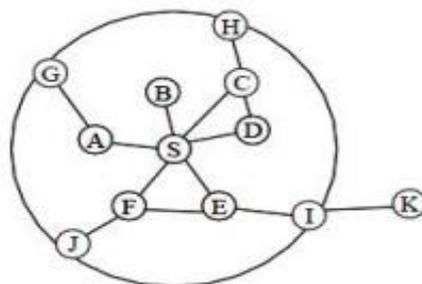
Gambar 2.2 Jenis protokol *routing* pada MANET

Sumber: Chhabra (2013)

2.4 Zone Routing Protocol (ZRP)

Zone Routing Protocol (ZRP) merupakan protokol *routing* hybrid dengan komponen proaktif dan reaktif. Proaktif *routing* melakukan *routing* dengan *node* tujuan yang berada di dalam zona lokal. Sedangkan reaktif *routing* melakukan *routing* menuju *node* tujuan yang berada di luar zona lokal. ZRP mengelompokkan kumpulan *node* yang berada dalam suatu zona lokal dengan zona sentral menjadi pusat. Zona lokal tersebut dibatasi oleh suatu parameter yang disebut *hop*. Semua *node* yang berjarak sebanyak jumlah *hop* dari *node* sentral, berarti masuk ke dalam zona *routing* dari *node* sentral tersebut. Pada Gambar 2.3, sebuah zona *routing* dengan *node* sentral S dan jumlah *hop* pada zona *node* S adalah dua. *Node* yang masuk ke dalam zona *routing* *node* S adalah *node* A-J, kecuali *node* K. *Node* A-F berjarak lebih kecil dari jumlah *hop*. Sedangkan *node* G-J berjarak sama dengan jumlah *hop*. *Node* H dan J dapat dicapai melalui dua jalur, yang satu berjarak dua *hop* dan yang satu lagi berjarak tiga *hop*. *Node* H dan J masih termasuk ke dalam *node* S, karena jalur terpendeknya lebih kecil atau sama dengan jumlah *hop*.





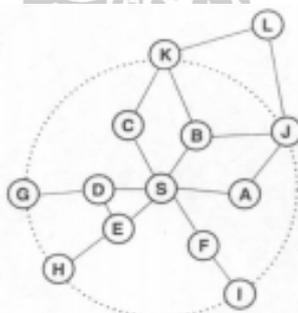
Gambar 2.3 Zona *routing node S* dengan jumlah *hop* dua

Sumber: Nicklas Beijar (2014)

ZRP terbagi dalam empat subprotokol, yaitu :

1. *Intrazone Routing Protocol*

Intrazone Routing Protocol (IARP) merupakan bagian dari protokol proaktif dengan jangkauan yang terbatas. Jangkauan IARP ditentukan dari suatu parameter *hop* yang membatasi zona *routing*. Untuk mendeteksi *node* tetangga dan rute yang rusak, IARP dibantu oleh *Neighbour Discovery Protocol* (NDP). IARP akan menyediakan pembaruan rute pada zona lokal ketika topologi dari zona *routing* berubah. Ketika ada *node* sumber yang ingin mengirimkan paket menuju sebuah *node* yang berada di dalam zona *routing node* sumber tersebut, rute dari *node* sumber menuju *node* tujuan akan tersedia dengan cepat. Apabila ada jalur yang rusak, maka IARP akan memperbarui rute pada zona lokal dan mencarikan jalur yang baru dengan mengirimkan *hello message* lagi ke *node* tetangga yang sudah berbeda dari sebelumnya.



Gambar 2.4 Zona *routing node S* dengan jumlah *hop* dua dalam IRP

Sumber: Nicklas Beijar (2014)

Sebagai contoh pada Gambar 2.4 *node S* sebagai *node* sumber ingin mengirimkan paket menuju *node J*, maka rute menuju *node J* akan tersedia dengan cepat. Paket bisa dengan segera dikirimkan menuju *node J*. Dengan adanya subprotokol ini, ZRP membatasi wilayah kerja protokol proaktif ini hanya pada *node* yang berada pada zona *routing* lokal saja. Oleh karena itu, jika ada *node* yang keluar dari zona *routing* lokal atau masuk ke zona *routing* lokal, hanya *node* dalam zona lokal saja yang perlu diinformasikan mengenai perubahan topologi tersebut.



2. Neighbour Discovery Protocol

Neighbour Discovery Protocol (NDP) merupakan subprotokol yang disediakan oleh layer MAC sehingga IARP dapat mengetahui node tetangga yang berada di sekitarnya. NDP mengirimkan *hello message* secara *broadcast* ke node tetangga yang berada dalam zona. Rute menuju node tetangga akan selalu diperbarui secara periodik.

3. Interzone Routing Protocol

Interzone Routing Protocol (IERP) merupakan bagian dari reaktif *routing*. Ketika node sumber ingin mengirimkan paket ke node yang berada di luar zona lokal, node sumber membutuhkan rute yang belum tersedia, IERP akan membantu node sumber untuk mencari rute baru. Dalam melakukan proses pencarian rute baru, IERP dibantu oleh *Bordercast Resolution Protocol* (BRP) yang bertugas melakukan *bordercast*. Pertama kali, node sumber memeriksa apakah node tujuan berada pada zona *routing* lokal atau tidak. Jika node tujuan tidak berada pada zona *routing* lokal, maka node sumber akan membentuk *route request packet* (RRP) dan melakukan *bordercast*. Format *route request packet* dapat dilihat pada Gambar 2.5.

0-31 (32 bit)		
Query Source Address		
Query Destination Address		
Query ID	Query Extension	Reserved
Previous Bordercast Address		
Encapsulated Packet		

Gambar 2.5 Route request packet

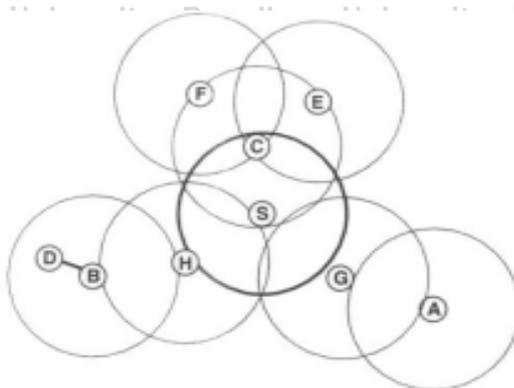
Sumber: Nicklas Beijar (2014)

Keterangan dari *route request packet* adalah sebagai berikut:

- Query Source Address* (32bit) merupakan alamat IP dari node sumber (node yang mengajukan permintaan rute).
- Query Destination Address* (32bit) merupakan alamat IP node tujuan.
- Query ID* (16bit) merupakan *sequence number* yang bersama-sama dengan *query source address* mengidentifikasi *query Bordercast Resolution Protocol* (BRP) lain di dalam jaringan.
- Query Extension* (8bit) untuk menentukan *query* harus diteruskan ke tujuan *query* yang mana.
- Reserved* (8bit) untuk alokasi sistem.
- Previous Bordercast Address* (32bit) merupakan alamat IP dari node yang sudah pernah menerima *bordercast*.
- Encapsulated Packet* merupakan paket yang sudah dienkapsulasi.



Ketika *border node* menerima *route request packet* yang berasal dari *node* sumber, *border node* akan memeriksa *node* tujuan pada zona *routing* lokal. Jika *node* tujuan berada pada zona *routing* lokal, maka sebuah *route reply* akan dikirim kembali ke *node* sumber, setelah menambahkan alamatnya pada paket tersebut, serta menambahkan rute dari sumber menuju tujuan.



Gambar 2.6 Contoh kerja IERP

Sumber: Nicklas Beijar (2014)

Sebagai contoh pencarian rute ditunjukkan pada Gambar 2.6. *Node* sumber S ingin mengirimkan paket menuju *node* tujuan D. *Node* S mencari *node* D di dalam zona *routing* milik S. Jika *node* D ada di dalam zona *routing* S, maka *node* S sudah mengetahui rute menuju *node* D. Jika *node* D belum ada di dalam zona *routing* S, S akan melakukan *bordercast* sebuah *route request packet* menuju semua *border node* yang berada dalam zona *routing* S (*node* C, G, dan H). Kemudian setiap *node* tersebut akan mencari dimana posisi dari *node* D dengan cara yang sama, yaitu melakukan *bordercast* dalam zonanya masing-masing *node* tersebut (*node* C, G, dan H). Pada Gambar 2.6 di atas, posisi *node* D yang menjadi tujuan dari *node* S berada pada zona B atau dicakup oleh zona B. Selanjutnya, jika sudah diketahui posisi *node* D tersebut maka sudah dapat dipastikan mengenai rute yang digunakan yaitu *node* H akan mengirim paket tersebut ke *node* B dengan zona *routing* mencakup *node* D. Lalu *node* D membalas paket tersebut, mengirimkan *route reply* setelah menambahkan alamat *node* D tersebut dan rute dari sumber menuju tujuan dengan rute S-H-B-D. Salah satu keistimewaan dari proses pencarian rute adalah sebuah permintaan satu rute dapat mengembalikan *route reply* lebih dari satu dan dapat digunakan untuk menentukan rute terbaik berdasarkan *link cost* yang paling rendah.

4. *Bordercast Resolution Protocol (BRP)*

Bordercast Resolution Protocol (BRP) merupakan layanan pengiriman paket yang digunakan pada ZRP untuk mengarahkan *route request packet* dari IERP menuju ke *border node*. Proses mengarahkan *route request packet* menuju *border node* disebut *bordercast*. Dengan adanya BRP ini, permintaan yang berlebihan dapat dihilangkan. BRP bertanggung jawab untuk meneruskan RRP yang dibentuk IERP ke *border node*. Walaupun penerima dari RRP adalah *border node*, BRP tetap

mengirim RRP pada setiap *hop*. BRP akan melacak *node* yang sudah tercakup oleh RRP. Ketika sebuah *node* menerima RRP, *node* tersebut menandai *node* tetangga dari *node* yang melakukan *bordercast* sebagai *node* yang sudah tercakup oleh RRP. Jika *node* penerima adalah *border node* dari *node* yang melakukan *bordercast*, maka *border node* tersebut akan menjadi *node* yang akan melakukan *bordercast* yang baru dan *node* tetangganya akan tercakup oleh RRP. BRP menggunakan tabel *routing* IARP dari *node* yang melakukan *bordercast*. Ketika BRP menerima RRP baru untuk di *bordercast*, maka RRP dari *node* pengirim akan mencakup *node* tetangga dari *node* pengirim dan RRP dikirim ke *border node*. BRP bisa memutuskan pengiriman, bila *node* menerima RRP yang sama dari *node* lain.

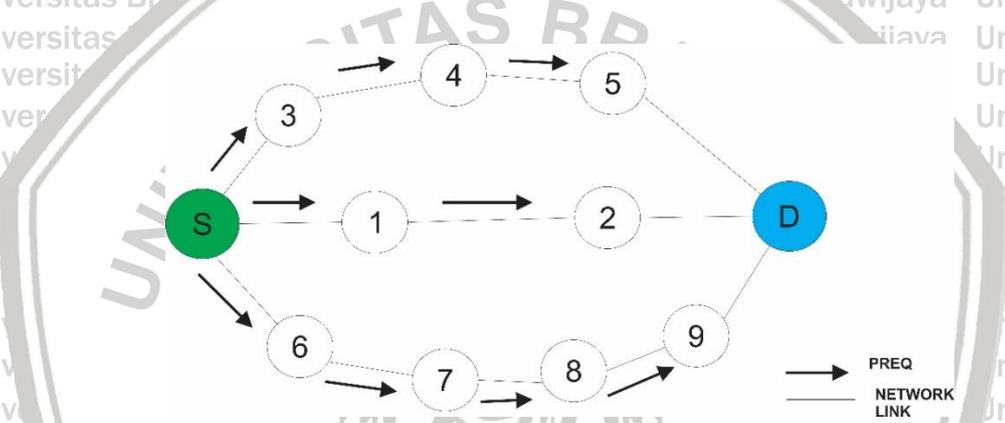
2.5 Hybrid Wireless Mesh Protocol (HWMP)

Hybrid Wireless Mesh Network (HWMP) merupakan default protokol *routing* dari IEEE 802.11s WLAN mesh networking. Sebagai salah satu jenis protokol *routing hybrid*, HWMP memiliki komponen reaktif dan komponen proaktif. Proses *routing* protokol HWMP mengikuti konsep umum dari protokol *Ad Hoc On-demand Distance Vector* (AODV). Kombinasi komponen reaktif dan proaktif dalam HWMP yang optimal dan efisien memungkinkan pemilihan jalur pada berbagai jaringan mesh dengan atau tanpa infrastruktur. Protokol *routing* HWMP menggunakan mekanisme nomor urut (*sequence number*) untuk menjaga konektivitas bebas loop setiap saat. Ada empat jenis pengiriman pesan yang digunakan dalam HWMP, yaitu *Path Request* (PREQ), *Path Reply* (PREP), *Path Error* (PERR), dan *Root Announcement* (RANN). Kecuali untuk PERR, seluruh kontrol pesan mengandung tiga bidang penting, yaitu *destination sequence number* (DSN), *time to live* (TTL) dan *metric*. DSN dan TTL dapat mencegah perhitungan masalah yang tak terbatas dan *metric* membantu untuk menemukan jalur *routing* yang lebih baik dibandingkan menggunakan *hop count*. Seluruh *routing* protokol tersebut terbentuk berdasarkan kontrol pesan yang ada.

1. Routing Reaktif Protokol HWMP

Proses *routing* reaktif untuk HWMP mengikuti konsep umum dari protokol AODV, yang disebut *Radio-Metric AODV* (RM-AODV), yang mana AODV bekerja pada lapisan ketiga yaitu lapisan *network* dengan menggunakan alamat IP dan menggunakan jumlah *hop* sebagai metrik peruteannya. Sedangkan RM-AODV, bekerja pada lapisan kedua yaitu *data link* dengan menggunakan alamat MAC dan menggunakan metrik radar radio (*radio-aware routing metric*) untuk pemilihan jalurnya. Pada *routing* reaktif ini, proses penemuan jalur akan dimulai hanya ketika *source node* melakukan permintaan. Seperti dijelaskan di atas bahwa *routing* HWMP menggunakan mekanisme penghitungan nomor urut (*sequence number*) untuk setiap jalurnya yang disebut dengan *Path Request* (PREQ) dan akan dikirimkan melalui *node-node* yang ada sampai menemukan dimana *destination node* berada. Di dalam PREQ terdapat tabel *routing* yang berisi *sequence number* dan informasi *hop* yang diperlukan untuk mencapai *destination node*. Dalam

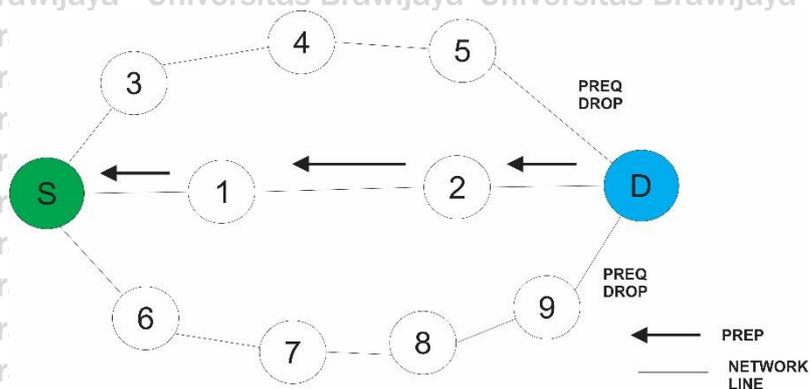
routing reaktif ini, *path request* (PREQ) dan *path reply* (PREP) bertindak sebagai rute penemuan (*route discovery*) sedangkan PERR bertindak sebagai rute deteksi (*route maintenance*). Ketika PREQ menjelajahi *node*, maka secara otomatis PREQ akan melakukan *set-up path*. Jika sebuah *node* menerima PREQ, tetapi *node* tersebut bukan merupakan *destination node*, maka *node* tersebut akan mengirimkan PREQ sebelumnya hingga sampai kepada *destination node*. Jika pesan PREQ sudah didapatkan *destination node*, maka *destination node* akan mengirimkan balasan pesan yaitu PREP menuju *source node*. Ketika PREP berjalan kembali menuju *source node* melalui *path* yang telah di *set-up* di awal tadi oleh PREQ, maka PREP akan melakukan *set-up* kembali mengarah kepada *source node* untuk menentukan jalur yang paling optimal dan melakukan *update timeout*. Proses PREQ dan PREP dapat di contohkan pada Gambar 2.7 dan Gambar 2.8 di bawah ini :



Gambar 2.7 Path Request HWMP

Sumber : Guesmia (2012)

Pada Gambar 2.7 yang bertindak sebagai *source node* adalah *node S*, dan *destination node* adalah *node D*. *Node S* akan membroadcast pesan PREQ ke semua tetangga, sampai *destination node* didapatkan. Saat *node D* menerima PREQ, *node D* akan mencetak jumlah *hop account* PREQ yang pertama. PREQ yang pertama dari *node 2* dengan jumlah *hop account* sebanyak 3, dan kemudian *node D* akan melakukan *reply* paket dari jalur *node 2*.



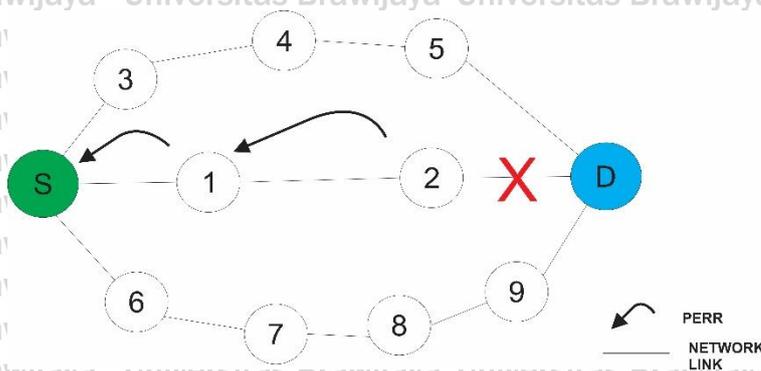
Gambar 2.8 Path Reply HWMP

Sumber : Guesmia (2012)

Pada Gambar 2.8, *node D* akan mengirimkan PREP ke *node 2*, kemudian *node 2* akan meneruskan paket PREP sampai *source node* atau *node S*. Sementara itu paket PREQ dari *node 5* datang, karena didapatkan jumlah *hop account* lebih besar, maka paket PREQ dari *node 5* akan di drop, begitu juga dengan paket PREQ dari *node 9* akan di drop juga. *Routing* menuju *node D* akan terbentuk yaitu melewati *node (1,2)*.

Kemudian pada tahap pemeliharaan jalur (*route maintenance*), adalah merespon adanya perubahan topologi yang terjadi setelah jalur terbentuk pada proses *route discovery* di atas. *Node* secara berkelanjutan akan mengamati *link* yang masih aktif (*link-layer feedback*) dan memperbarui batas valid *route* entri ketika mengirim dan menerima paket data. Jika sebuah *node* menerima paket data namun tidak memiliki valid *route* maka *node* tersebut akan merespon dengan pesan PERR. Ketika membuat pesan PERR, *node* akan membuat daftar yang berisi alamat dan *sequence number* dari *node* yang tidak terjangkau. Selain itu *node* akan menambahkan semua *route* entri yang bergantung pada *unreachable node*.

Tujuan dari langkah tersebut adalah untuk memberi tahu bahwa terdapat *route* yang tidak lagi tersedia. Proses penyebaran pesan PERR dijelaskan pada Gambar 2.9 di bawah ini :



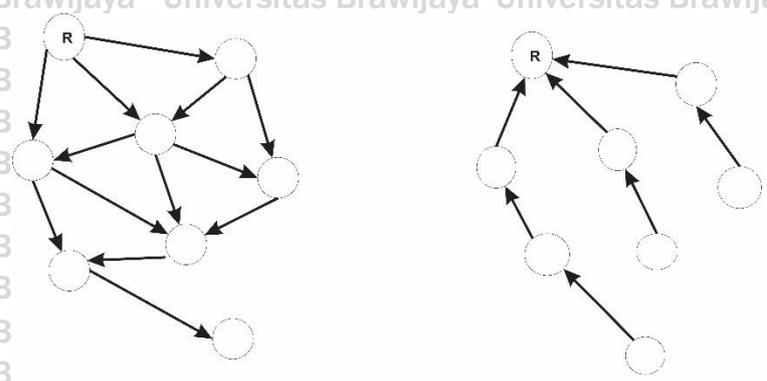
Gambar 2.9 Proses penyebaran pesan PERR routing reaktif HWMP

Sumber : Guesmia (2012)

Saat *node* 2 dan *node* D terputus, *node* 2 akan mengirimkan PERR ke tetangga jalur *routing* yaitu *node* 1. Kemudian *node* 1 akan meneruskan paket PERR sampai kepada *node* S. saat *node* S menerima PERR maka *node* S akan menghapus jalur *routing* tersebut dan memulai dari awal lagi.

2. Routing Proaktif Protokol HWMP

Dalam *routing* proaktif protokol HWMP terdapat dua mekanisme yang digunakan dalam pencarian jalur untuk mencapai *destination node*. Metode pertama menggunakan pesan *path request* (PREQ) proaktif untuk membuat jalur antara *root* dan semua *node* di jaringan secara proaktif. Metode yang kedua menggunakan pesan *root announcement* (RANN) untuk mendistribusikan informasi jalur untuk mencapai *root* tetapi jalur ter-update dapat dibangun sesuai permintaan. *Routing* proaktif HWMP adalah mode *routing* berdasarkan *tree based* yang terpusat di sebuah *node root gateway*. Di mode proaktif PREQ, sebuah *node root* secara periodik melakukan *broadcast* pesan PREQ dengan *sequence number* yang unik. *Node* lainnya menerima PREQ dan merekamnya di *routing metric* kemudian mengupdate PREQ, yang didalamnya terdapat *time to live* (TTL), *hop count*, *path metric* dan akan melakukan *broadcast* kembali membentuk *reverse path* menuju *node root*. *Forward path* dibentuk ketika proaktif PREP *flag* di dalam PREQ di *set value* menjadi satu oleh *node root*. Setelah itu ketika *node* tujuan telah ditemukan maka *node* akan menerima *path request* (PREQ) untuk membuat atau melakukan *update* jalur *path* pada *root*, melakukan *update path request* (PREQ) dengan informasi yang akan diperbarui, dan kemudian *path request* (PREQ) akan dikirim menuju *node* sumber. Mekanisme seperti pada Gambar 2.10 dibawah ini:

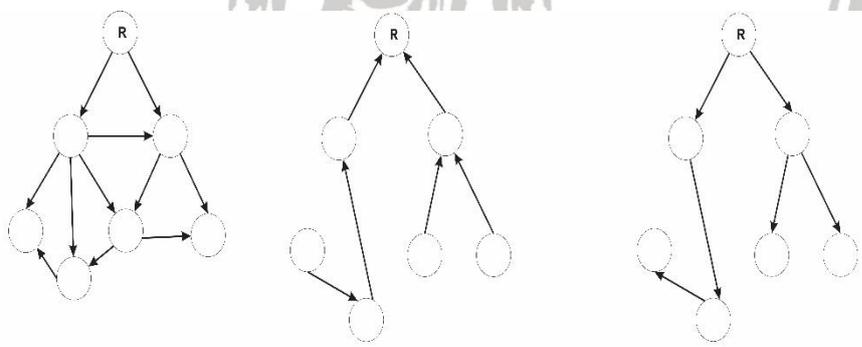


a. Root broadcast a PREQ b. Backward path update by PREP

Gambar 2.10 Mekanisme Proactive PREQ

Sumber : Guesmia (2012)

Dalam mekanisme *proactive* (RANN) *node root* akan secara periodik mengirimkan paket *root announcement* ke jaringan. Ketika *root announcement* sudah diterima maka akan dilakukan pembaruan jalur *routing* menuju *root*. Saat *root* menerima paket *unicast* PREQ maka akan dilakukan balasan dengan *path reply* (PREP). Dengan demikian *unicast* (PREQ) akan membalikkan rute dari *root* berasal, sementara itu *unicast* (PREP) akan membuat rute dari *node* asal ke *root*. Mekanisme seperti pada Gambar 2.11 dibawah ini:



a. Root broadcast a RANN b. Backward path update by the unicast PREQ c. Forward path update by the unicast PREP

Gambar 2.11 Mekanisme Proactive RANN

Sumber : Guesmia (2012)

Pada mekanisme yang ada pada HWMP, yaitu mekanisme RANN dan mekanisme PREQ memiliki pengaruh pada kinerja *proactive tree building mode* (konstruksi topologi rute yang dibuat). Jika jarak interval topologi antar rute pendek maka keandalan jalur menjadi tinggi, tetapi untuk biaya pemeliharaan akan semakin tinggi pula. Di sisi lain jika jarak interval topologi antar rute memiliki



jarak interval yang panjang, maka keandalan jalur semakin berkurang dan biaya pemeliharaan semakin rendah tetapi komunikasi antara titik-titik *mesh* akan meningkat karena pembaruan rute jalur *on-demand* akan dilakukan ketika kerusakan terjadi.

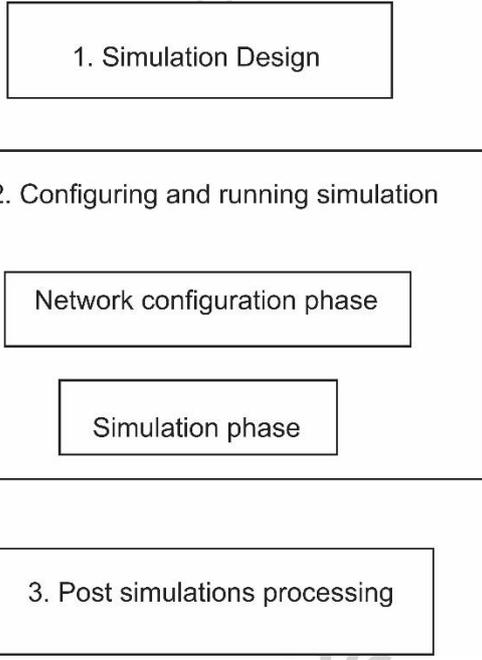
2.6 Network Simulator (NS2)

Network Simulator 2 (NS2) merupakan alat simulasi yang berguna untuk mempelajari komunikasi jaringan yang bersifat dinamis. NS2 dapat menjalankan simulasi kabel serta fungsi jaringan nirkabel dan protokol (algoritma *routing*, TCP, UDP). Secara umum konsep dari NS2 menyediakan bagi penggunanya bagaimana cara menentukan protokol jaringan yang diinginkan dan bagaimana simulasi yang sesuai terhadap protokol jaringan yang diinginkan tersebut. Dalam penggunaannya, NS2 dapat dijalankan dengan perintah eksekusi "ns" yang bisa mengambil satu argumen *input*, yaitu pada nama file *script.tcl* yang digunakan. Dalam NS2 terdapat jejak *file* yang digunakan sebagai plot grafik atau untuk membuat animasi. NS2 menggunakan dua Bahasa pemrograman, yaitu C++ berfungsi sebagai definisi mekanisme internal, yaitu *backend* dari simulasi. Yang kedua adalah OTcl (*Object oriented Tool Command Language*), yang digunakan untuk memulai simulasi dengan tahapan merakit dan mengkonfigurasi objek. Bahasa C++ dan OTcl tersebut dihubungkan bersama menggunakan *tcl*. Dalam domain OTcl terdapat variabel yang disebut sebagai *handles*.

Handles merupakan suatu *string* dalam domain yang terdapat pada OTcl. Jika terdapat fungsi, misal *receive packet* maka akan didefinisikan dalam objek C++. Dalam domain OTcl *handles* akan bertindak sebagai *frontend* dalam berinteraksi dengan objek-objek OTcl yang lain dan pengguna. Hal ini sebagai definisi variable dan prosedur sendiri untuk memfasilitasi suatu interaksi (Issariyakul, 2009).

2.6.1 Langkah Simulasi Pada NS2

Pada langkah simulasi pada NS2 ini menunjukkan langkah-langkah dalam simulasi umum NS2. Langkah simulasi umum ini dapat disesuaikan dengan kerangka NS2. Langkah-langkah simulasi yang dilakukan dijelaskan pada Gambar 2.12 sebagai berikut:



Gambar 2.12 Skema rancangan simulasi pada NS2

Sumber : Issariyakul (2009)

Dari Gambar 2.12 diatas diberikan keterangan mengenai skema rancangan simulasi pada NS2 sebagai berikut :

1. Desain Simulasi

Pada desain simulasi ini menjelaskan tentang langkah pertama yang dilakukan saat memulai perancangan simulasi. Dalam hal ini pengguna harus menentukan tujuan simulasi, konfigurasi jaringan, pengukuran kinerja dan hasil yang diharapkan.

2. Konfigurasi dan Menjalankan Simulasi

Pada proses konfigurasi dan menjalankan simulasi disini merupakan langkah kedua dalam proses simulasi. Langkah ini bertujuan untuk implementasi desain simulasi yang sudah dibuat pada langkah pertama. Pada langkah ini terdapat dua tahap yaitu:

- Tahap konfigurasi jaringan, dalam hal ini komponen jaringan yang digunakan seperti TCP, UDP dan *node* dibuat dan dikonfigurasi sesuai dengan desain simulasi pada langkah pertama dan juga penjadwalan *event* seperti transfer data yang dimulai pada waktu tertentu.
- Tahap simulasi, dalam hal ini simulasi dimulai. Simulasi ini merupakan hasil konfigurasi dari tahap konfigurasi jaringan, yang berfungsi untuk memelihara *clock* simulasi dan menjalankan *event* secara terurut. Dalam tahap ini umumnya dapat berjalan sampai dengan nilai *clock* simulasi yang mencapai batas yang ditentukan pada tahap konfigurasi jaringan.



3. Past Simulation Processing

Langkah yang ketiga merupakan tugas utama yaitu melakukan verifikasi integritas program dan melakukan evaluasi kinerja simulasi jaringan ketika tugas pertama sebagai *debugging*. Kemudian dilakukan pengumpulan dan penyusunan hasil simulasi dengan baik.

2.7 AWK Programming

Awk merupakan bahasa pemrograman yang dalam menangani tugas-tugasnya menggunakan program yang sangat singkat. Program awk menjelaskan apa yang harus dicari dalam *input* data serta apa yang harus dilakukan ketika hal tersebut ditemukan. Cara kerja program awk adalah mencari satu set *file* untuk baris yang cocok dengan salah satu pola, kemudian akan dilakukan tindakan yang sesuai ketika pola tersebut ditemukan. Pola tersebut menggunakan kombinasi dari ekspresi regular ketika memilih baris dan operasi bandingan pada angka, bidang, variabel sehingga dapat menimbulkan aksi yang dapat melakukan pengolahan dengan bebas pada baris yang akan di eksekusi. *File input* yang ada akan dipindai oleh program awk dan setiap baris input akan dibagi ke bidang secara otomatis. Oleh sebab itu banyak seperti *input*, manajemen penyimpanan, inialisasi yang semuanya dilakukan secara otomatis, maka program awk jauh lebih kecil dari bahasa yang lebih konvensional (Alvred V.Aho,1988).

2.8 Parameter dalam kinerja protokol routing

Dalam menganalisa kinerja sebuah *routing* protokol terdapat beberapa parameter yang dijadikan acuan untuk mengukur tingkat kualitas dari *routing* protokol tersebut. Parameter-parameter ini biasanya disebut sebagai parameter QOS (*Quality Of Service*). Beberapa parameter QOS ini antara lain adalah *throughput*, *packet delivery ratio* (PDR), *normalized routing load*, *end-to-end delay* dan *convergence time*.

2.8.1 Throughput

Throughput adalah kecepatan (rate) transfer data efektif di mana kalkulasi dilakukan dalam bit per second (bps) sebagai total banyaknya paket yang diterima dibagi dengan waktu pengiriman (Denatama, 2016). Rumus untuk menghitung *throughput* dituliskan pada Persamaan 2.1 berikut :

$$\text{Throughput} : \frac{Pr}{Tp} \times \text{Ukuran paket} \quad (2.1)$$

Pr = jumlah paket yang diterima

Tp = jumlah waktu pengamatan

2.8.2 Packet Delivery Ratio (PDR)

Packet delivery ratio adalah rata-rata waktu yang diambil dari paket data yang tiba pada tujuan. Dalam hal ini hanya paket data yang berhasil dikirim ke tujuan

lah yang dihitung (Rao, 2014). Rumus untuk menghitung *end-to-end delay* dituliskan pada Persamaan 2.2 berikut :

$$\text{Packet delivery ratio} : \frac{Pr}{Ps} \quad (2.2)$$

TPr = jumlah total waktu paket yang diterima

Pr = jumlah paket yang diterima

2.8.3 End-to-end Delay

End to end Delay adalah rasio antara jumlah total seluruh paket data yang sukses diterima, selanjutnya dibagi dengan jumlah total seluruh paket data yang dikirimkan (Denatama, 2016). Rumus untuk menghitung *packet delivery ratio* (PDR) dituliskan pada Persamaan 2.3 berikut :

$$\text{End - to - end delay} : \frac{TPr}{Pr} \quad (2.3)$$

Pr = jumlah paket yang diterima

Ps = jumlah paket yang dikirim

2.8.4 Normalized Routing Load

Normalized routing load adalah nilai perbandingan antara banyak paket *routing* yang dikirim *source node* dan diteruskan (*forwarding*) dengan jumlah paket data yang diterima pada *destination node*. Semakin tinggi nilai perbandingan paket *routing* terhadap paket yang diterima maka semakin kurang efisien kinerja suatu protokol *routing* (Parkins, 2001). Rumus untuk menghitung *normalized routing load* dituliskan pada Persamaan 2.4 berikut :

$$\text{Normalized Routing Load} : \frac{Prt}{Pr} \quad (2.4)$$

Prt = jumlah paket routing (sent & forwarded)

Pr = jumlah paket yang diterima

2.8.5 Convergence Time

Convergence adalah proses *update* ketika komunikasi terputus atau terjadi *link failure*, update akan dikirim ke seluruh jaringan yang menjelaskan perubahan dalam topologi jaringan. Setiap *node* kemudian menjalankan algoritma *routing* untuk menghitung ulang jalur dan membangun jalur baru berdasarkan informasi tersebut. Setelah semua *node* yang terputus kembali terhubung, proses konvergensi selesai dilakukan (Sheldon, 2001). Terdapat waktu saat proses penghitungan jalur baru sampai disimpan kembali, waktu inilah yang akan digunakan sebagai analisis penelitian. Rumus untuk menghitung *convergence time* dituliskan pada Persamaan 2.5 berikut :

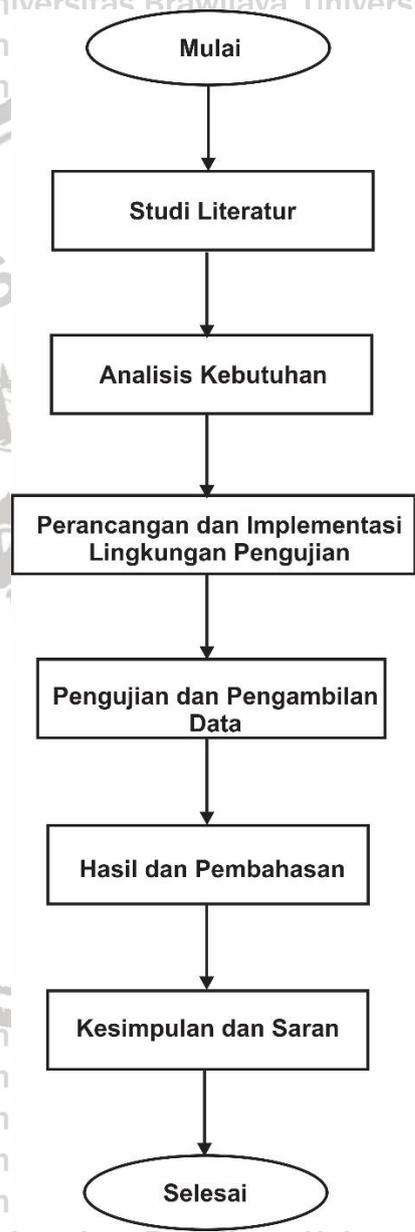
$$\text{Convergence time} : \sum_i^n \sum_j^n \frac{tij}{n} \quad (2.5)$$

Untuk $i \neq j$ dan t_{ij} adalah waktu informasi *routing* diterima oleh *node destination* — waktu informasi *routing* dikirim oleh *node source*. Sedangkan n adalah jumlah *node* yang menerima pesan waktu informasi *routing*.



BAB 3 METODOLOGI

Pada bab ini akan dijelaskan langkah-langkah yang dilakukan untuk menyelesaikan masalah pada penelitian ini. Langkah-langkah tersebut meliputi studi literatur, perancangan sistem, analisis dan pembahasan dan pengambilan kesimpulan. Metodologi penelitian yang akan digunakan pada penelitian ini digambarkan pada Gambar 3.1 berikut :



Gambar 3.1 Tahap Metodologi Penelitian



3.1 Studi Literatur

Studi literatur merupakan kegiatan yang dilakukan untuk menentukan objek penelitian yang sesuai dengan topik yang diambil. Studi literatur juga berfungsi sebagai referensi untuk penelitian yang dilakukan. Studi literatur digunakan untuk mendukung penelitian dalam penyelesaian masalah agar tercapainya tujuan penelitian. Teori pendukung dapat diperoleh dari jurnal, buku, dan penelitian sebelumnya. Literatur yang digunakan dalam penelitian ini meliputi pemahaman terhadap dasar teori dan kajian pustaka yang terkait, yaitu sebagai berikut :

1. *Mobile AdHoc Network (MANET)*

Studi literatur mengenai MANET mempelajari dengan mencari beberapa jurnal tentang karakteristik, fungsi dan penggunaan pada saat ini.

2. *Protokol Routing Hybrid*

Studi literatur terkait protokol *routing hybrid* pada MANET dilakukan dengan mempelajari dasar teori tentang mekanisme kinerja protokol dari beberapa jurnal terkait.

3. *Zone Routing Protocol (ZRP)*

Studi literatur terkait ZRP pada MANET dilakukan dengan mempelajari dasar teori dan kajian pustaka tentang mekanisme kinerja protokol dari beberapa jurnal terkait.

4. *Hybrid Wireless Mesh Protocol (HWMP)*

Studi literatur terkait HWMP pada MANET dilakukan dengan mempelajari dasar teori dan kajian pustaka tentang mekanisme kinerja protokol dari beberapa jurnal terkait.

5. *Network Simulator 2 (NS2)*

Studi literatur terkait NS2 dilakukan dengan mempelajari struktur dan lingkungan simulator beserta konfigurasi dalam menjalankan NS2 yang terdapat pada website NS2.

3.2 Analisis Kebutuhan

Pada tahap analisis kebutuhan penelitian ini akan menggunakan perangkat keras dan perangkat lunak yang mendukung dalam penelitian. Analisis kebutuhan dibagi menjadi dua, yaitu perangkat keras dan perangkat lunak.

A. Perangkat Keras

Kebutuhan perangkat keras yang digunakan dalam mengerjakan penelitian ini membutuhkan perangkat keras yang mampu mendorong berbagai aplikasi dan program yang digunakan dalam penelitian. Berikut ini adalah spesifikasi perangkat keras yang digunakan :

Laptop : Asus A455 L

Memori : 4GB

Prosesor : Intel Core i5-930M

CPU : 2,19 GHz

Sistem Operasi : Ubuntu 18.04

B. Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan dalam penelitian ini adalah Network Simulator 2 (NS-2) yang merupakan program simulasi jaringan yang bersifat open source. NS-2 dibangun berdasarkan 2 bahasa pemrograman C++ yang biasanya digunakan untuk event scheduler, protokol, network components, dan Tcl/OTcl digunakan untuk menulis script simulasi. Komponen – komponen NS-2 terdiri dari Tcl (Tool command language) merupakan scripting programming untuk konfigurasi network simulator. OTcl (Object tcl) merupakan Tcl Interpreter yang melakukan inisiasi event scheduler membangun topologi jaringan berbasis objek serta memberitahu sumber traffik saat memulai dan mengakhiri pengiriman paket. Tclcl merupakan bahasa pemrograman untuk menyediakan linkage antara C++ dan OTcl berupa class hierrarchy, object instantiation, variable binding dan command dispatching. NS disebut sebagai simulator, NAM adalah network animator yang berfungsi untuk memvisualisasikan output dari NS-2. Pre-processing berfungsi untuk membangkitkan trafik dan topologi jaringan Post-processing merupakan analisis tampilan output dari simulasi pada file .tr dimana sebageian dar hasil simulasi tersebut dapat di filter menggunakan perintah awk dan dapat dikonversikan dalam bentuk grafik.

3.3 Perancangan dan Implementasi Lingkungan Pengujian

Dalam tahap perancangan dan implementasi lingkungan pengujian ini, dijelaskan mengenai langkah-langkah untuk penentuan parameter pengujian dan dijabarkan mengenai tahapan-tahapan dalam perancangan skenario pengujian yang akan dilakukan. Dijabarkan juga mengenai implementasi lingkungan pengujian yang akan dijalankan pada NS2 sesuai skenario yang dirancang.

3.3.1 Perancangan Lingkungan Pengujian

Dalam perancangan lingkungan pengujian dijelaskan mengenai penentuan parameter lingkungan pengujian seperti luas area yang digunakan, jumlah *node* yang digunakan dalam implementasi, durasi pengujian dan kecepatan pergerakan *node*, model transport yang digunakan, ukuran paket yang akan dikirim, dan jumlah konektivitas yang dibangun. Dalam perancangan lingkungan pengujian ini juga dijelaskan mengenai perancangan skenario lingkungan pengujian beserta topologi yang akan diimplementasikan.

3.3.2 Implementasi Lingkungan Pengujian

Dalam implementasi lingkungan pengujian ini berisi bahasan mengenai implementasi terhadap proses perancangan lingkungan pengujian yang sudah dilakukan sebelumnya ke dalam NS2. Disini dibutuhkan beberapa konfigurasi pada NS2 seperti konfigurasi parameter lingkungan pengujian, konfigurasi topologi jaringan, pergerakan *node*, konfigurasi pengiriman paket dan pemrosesan data *output*.

3.4 Pengujian dan Pengambilan Data

Setelah dilakukan perancangan skenario lingkungan selesai dibuat dan implementasi lingkungan pengujian selesai di jalankan, maka langkah selanjutnya adalah melakukan pengujian yang bertujuan untuk menguji kinerja protokol *routing* ZRP dan HWMP sesuai skenario pengujian yang telah dirancang. Pengambilan data di dapatkan berdasar penghitungan parameter pengujian yang telah ditentukan sebelumnya, yaitu *throughput*, *packet delivery ratio* (PDR), *normalized routing load*, *end-to-end delay* dan *convergence time* menggunakan bahasa pemrograman AWK.

3.5 Hasil dan Analisis

Pada hasil dan analisis disini berisi penjabaran mengenai hasil kinerja masing-masing protokol yaitu ZRP dan HWMP yang didapat dari pengambilan data berdasar skenario lingkungan pengujian yang diimplementasikan pada NS2.

3.5.1 Analisis Pengaruh Variasi Kepadatan *Node*

Analisis pengaruh variasi kepadatan *node* dapat dilakukan dengan menjabarkan data hasil pengujian dan hasil yang didapat terhadap perbandingan kinerja *routing* protokol ZRP dan HWMP berdasarkan skenario luas yang digunakan sebesar 1000x1000 meter dengan variasi kepadatan *node* yang berbeda yaitu 30, 50, 70, 90 dan 110 pada masing-masing protokol. Kemudian dilakukan penghitungan parameter pengujian yang telah ditentukan yaitu *throughput*, *packet delivery ratio* (PDR), *normalized routing load*, *end-to-end delay* dan *convergence time*.

3.6 Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan dan hasil pembahasan selesai dilaksanakan dan didasarkan pada kesesuaian antara teori dengan penerapan. Kesimpulan dibuat untuk memberikan jawaban terhadap rumusan masalah tentang analisa perbandingan protokol *routing* ZRP dan HWMP pada MANET. Pembuatan saran merupakan tahapan akhir dalam penyusunan penelitian ini. Saran ditujukan untuk memperbaiki kesalahan yang terjadi selama proses penulisan dan implementasi serta merumuskan topik-topik yang dapat digunakan dalam penelitian berikutnya.

BAB 4 PERANCANGAN DAN IMPLEMENTASI LINGKUNGAN PENGUJIAN

Bab ini menjelaskan tentang tahapan perancangan dan implementasi lingkungan pengujian mengenai protokol *routing* ZRP dan HWMP. Perancangan yang dilakukan terdiri dari perancangan protokol *routing*, Implementasi lingkungan pengujian dalam penelitian ini diterapkan pada Network Simulator 2 (NS2).

4.1 Perancangan Penelitian

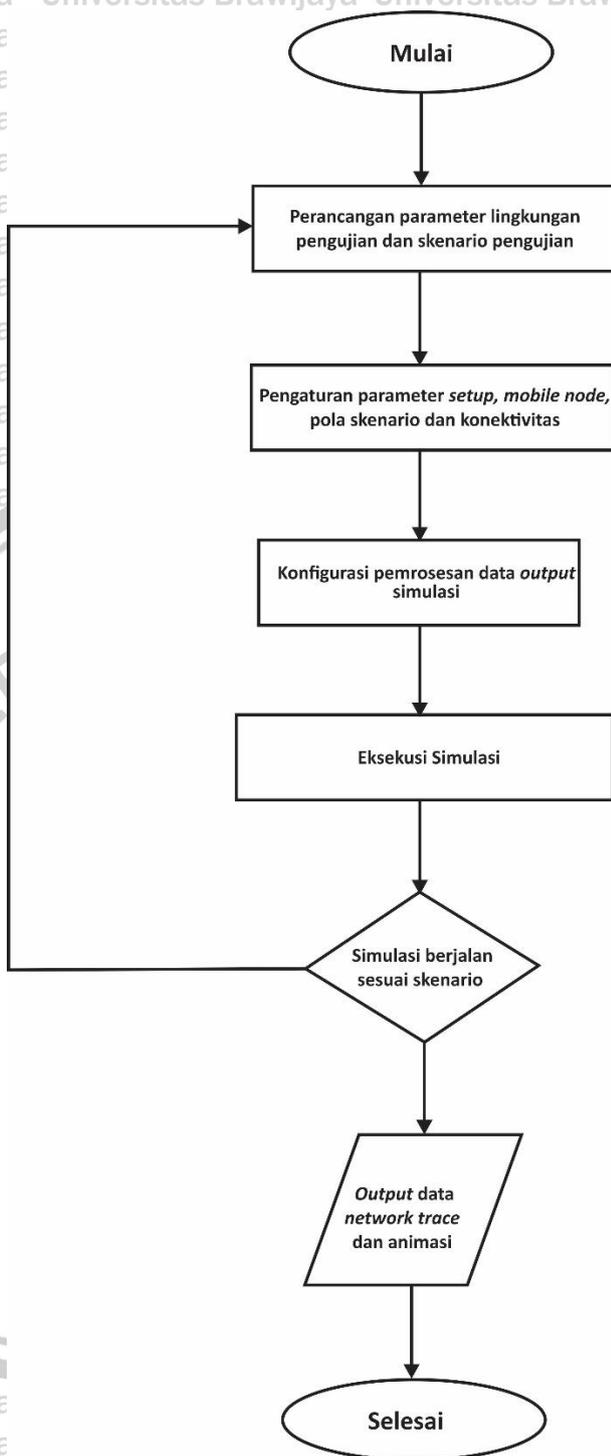
Pada sub bab ini akan dilakukan perancangan mengenai apa saja yang dibutuhkan dalam penelitian, yaitu perancangan protokol *routing*, perancangan lingkungan pengujian, perancangan parameter pengujian, perancangan pengujian dan perancangan topologi jaringan.

4.1.1 Perancangan Protokol *Routing*

Perancangan protokol *routing* menjelaskan alur perancangan protokol ZRP dan HWMP yang diimplementasikan pada Network Simulator (NS2). Langkah pertama yang dilakukan adalah proses *patching* ke dalam modul NS2. Setelah dilakukan *patching* kemudian install protokol pada NS2 menggunakan perintah *make install*. Jika proses instalasi berhasil, maka dilanjutkan ke tahap selanjutnya yaitu melakukan konfigurasi *file* *zrp.tcl* dan *hwmp.tcl*. Masing-masing *file* tersebut selanjutnya akan dilakukan modifikasi sesuai dengan perancangan parameter dan lingkungan pengujian yang digunakan. Jika protokol berhasil dijalankan akan dilakukan pengambilan data dari hasil pengujian yang didapatkan dari *file.awk* yang dijelaskan pada bab sebelumnya. Setelah melakukan pengujian dan analisis terhadap data yang diambil sesuai dengan parameter yang telah ditentukan, maka dilakukan penarikan kesimpulan.

4.1.2 Perancangan Lingkungan Pengujian

Dalam perancangan lingkungan pengujian menjelaskan mengenai tahapan yang disusun sebelum masuk pada tahap implementasi lingkungan pengujian yang akan dijalankan. Metode yang dipakai untuk perancangan menyesuaikan lingkungan dari simulator yang digunakan, yaitu NS2. Untuk metode yang akan dijalankan dapat dilihat pada Gambar 4.1 berikut :



Gambar 4.1 Skema Alur Tahapan Perancangan Lingkungan Pengujian pada NS2

4.1.3 Perancangan Konfigurasi Protokol Routing

Perancangan konfigurasi pada NS2 menjelaskan mengenai alur yang akan dilakukan untuk menerapkan implementasi parameter dan skenario pengujian yang telah dirancang untuk mendapatkan hasil yang sesuai harapan. Pada perancangan konfigurasi NS2 langkah yang dilakukan sebagai berikut :

4.1.3.1 Perancangan konfigurasi parameter pengujian

Dalam penentuan parameter-parameter yang akan digunakan selama proses simulasi protokol *routing* ZRP dan HWMP dalam jaringan MANET maka harus dilakukan perancangan parameter pengujian ini. Skema perancangan parameter pengujian akan dijabarkan pada Tabel 4.1 berikut:

Tabel 4.1 Perancangan konfigurasi parameter pengujian

Variasi <i>node</i>	30, 50, 70, 90, 110
Antenna	<i>Omni-directional</i> (RF)
Type MAC	802.11
Propagasi	Two Ray Ground
Ukuran paket	512 byte
Model transport	UDP (CBR)
Waktu simulasi	1000 sec
Simulator	Network Simulator 2 Versi 2.35
Interval paket	1
Luas area simulasi	1000x1000 meter
Mobilitas	Random Way Point
Kecepatan <i>node</i> dinamis	20 m/s

4.1.3.2 Perancangan skenario pengujian

Perancangan pengujian MANET dalam penelitian ini dilakukan terhadap dua jenis protokol *routing hybrid*, yaitu ZRP dan HWMP. Perancangan pengujian yang dibuat digunakan untuk menganalisa pengaruh kepadatan variasi jumlah *node* terhadap parameter pengujian. Pengujian dilakukan dengan satu koneksi menggunakan transport jaringan UDP dengan model *traffic constant bit-rate* (CBR). Skenario lingkungan pengujian yang dilakukan menggunakan luas area sebesar 1000x1000 meter. Selanjutnya untuk setiap skenario akan disimulasikan dengan variasi jumlah *node* yang berbeda-beda yakni 30, 50, 70, 90, dan 110 *node*

dengan durasi waktu selama 1000 detik. Besar paket yang dikirimkan ditentukan sebesar 512 byte. Dalam skenario pengujian paket mulai dikirimkan oleh *source node* menuju *destination node* sejak detik ke-0 sampai ke-1000. Parameter yang digunakan untuk mengukur kinerja dalam skenario pengujian ini adalah yaitu *throughput*, *packet delivery ratio* (PDR), *normalized routing load*, *end-to-end delay* dan *convergence time*. Perancangan skenario pengujian yang akan dilakukan ditunjukkan seperti pada Tabel 4.2 berikut :

Tabel 4.2 Perancangan skenario pengujian

Skenario	Luas area simulasi	Jumlah <i>node</i>	Kecepatan pergerakan <i>node</i>	Durasi simulasi
1	1000x1000 meter	30, 50, 70, 90, dan 110	Acak, kecepatan <i>node</i> dinamis minimal 10 m/s, maksimal 20 m/s	1000 detik

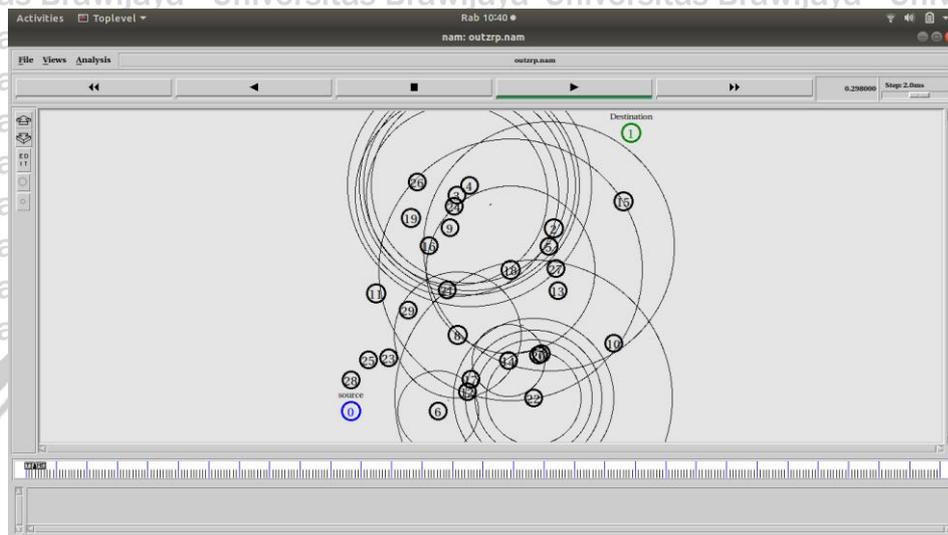
Dalam penelitian ini arah kecepatan dan pergerakan setiap *node* pada topologi dinamis diterapkan sebesar 20 m/s. Sedangkan untuk topologi statis, tidak perlu dilakukan pengaturan pergerakan setiap *node*, karena *node-node* akan berada pada posisi yang tetap selama proses *routing* berlangsung.

4.1.3.3 Perancangan konfigurasi topologi jaringan

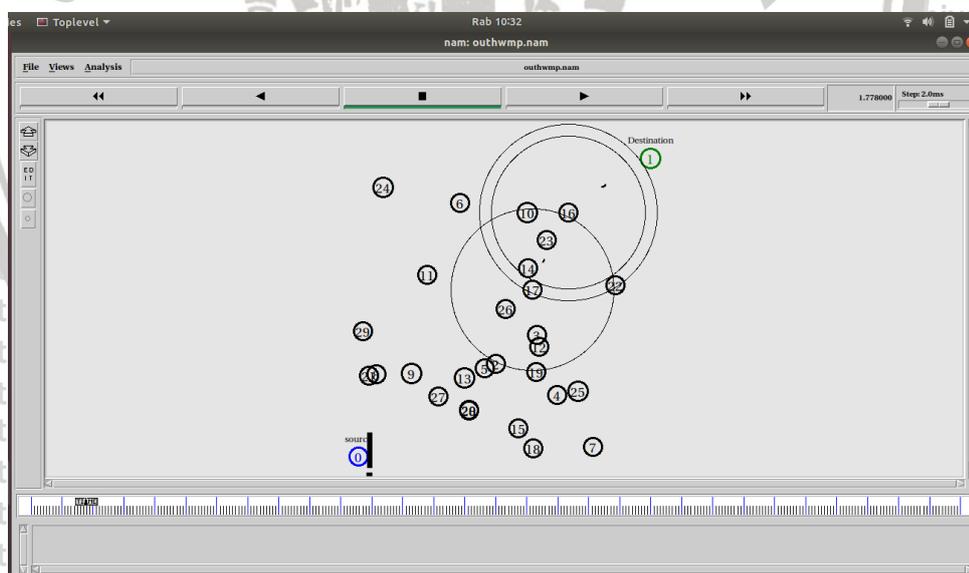
Pada perancangan konfigurasi topologi jaringan dijelaskan mengenai alur yang akan diimplementasikan pada topologi jaringan di NS2. Perancangan dilakukan sesuai dengan karakteristik MANET, yang memiliki *mobile node* yang bergerak bebas secara acak dengan arah dan posisi yang tidak menentu. Perancangan yang dilakukan pada topologi jaringan ini menyesuaikan dengan variasi kepadatan *node* yang digunakan, yaitu 30, 50, 70, 90 dan 110 pada masing-masing protokol *routing* ZRP dan HWMP. Dalam penelitian ini topologi dibuat dengan skema pergerakan tiap *node* dan posisi yang diatur secara acak, sehingga setiap skenario pengujian yang dijalankan akan berbeda pergerakan dan topologinya. *Source node* dan *destination node* yang digunakan masing-masing untuk variasi kepadatan *node* dilakukan pengujian sebanyak lima kali, misalkan pengujian pertama menggunakan *source node* (0) menuju *destination node* (1) pada pengujian variasi kepadatan *node* 30. Setelah pengujian tersebut selesai, dilakukan kembali pengujian dengan variasi kepadatan *node* yang sama tetapi dengan *source node* dan *destination node* yang berbeda, misalkan *source node* (2) menuju *destination node* (10). Pengujian *source node* dan *destination node* yang berbeda tersebut dilakukan sebanyak lima kali, kemudian untuk setiap pengujianya dilakukan rata-rata nilai dari masing-masing parameter pengujian. Tujuan pengujian ini agar diperoleh nilai yang valid daripada hanya menggunakan



satu pengujian; missal *source node* (0) menuju *destination node* (1) saja. Langkah pertama yang perlu dilakukan adalah proses inialisasi awal pembentukan *mobile node* sesuai jumlah variasi kepadatan *node* yang diinginkan. Selanjutnya dilakukan inialisasi warna yang digunakan untuk menandai *source node* dan *destination node*. Gambar 4.2 dan Gambar 4.3 di bawah ini merupakan sampel topologi jaringan dengan jumlah *node* 30 pada masing-masing protokol yang diatur secara acak sesuai skenario lingkungan pengujian.



Gambar 4.2 Screenshot topologi jaringan dengan 30 *node* ZRP



Gambar 4.3 Screenshot topologi jaringan dengan 30 *node* HWMP

4.1.3.4 Perancangan konfigurasi pergerakan *node*

Perancangan konfigurasi pergerakan *node* dilakukan dengan merancang skenario simulasi pergerakan setiap *node* pada *file.tcl*. Pembuatan pergerakan *node* dinamis yang dilakukan untuk proses konfigurasi dilakukan menggunakan fungsi *setdest* yang terdapat pada NS2. Pembuatan *file.tcl* yang berfungsi sebagai

pergerakan *node* dibuat di dalam folder *setdest* yang terdapat pada *ns-allinone*. Di dalam pembuatan pergerakan *node*, terdapat beberapa fungsi perintah untuk menetapkan jumlah *node*, pengaturan kecepatan pergerakan *node*, pengaturan waktu simulasi, dan luas area yang diinginkan. Setelah perintah tersebut dijalankan kemudian akan terbentuk *file.tcl* seperti yang disebutkan di atas. Pemberian nama *file.tcl* perlu dilakukan agar mempermudah dalam menentukan banyaknya variasi kepadatan *node* yang digunakan.

4.1.3.5 Perancangan konfigurasi pengiriman paket

Perancangan pengiriman paket perlu dilakukan untuk mengetahui bagaimana proses pengiriman paket data dari *source node* menuju *destination node*. Pengiriman paket yang digunakan menggunakan transport jaringan UDP. Dilakukan juga pemilihan *node* yang akan menjadi *source node* dan *destination node*. Pengaturan besar paket juga akan dimasukkan sesuai dengan parameter perancangan yang sudah dibuat sebelumnya yaitu sebesar 512 byte. Kemudian untuk transmisi data protokol yang digunakan adalah CBR (*Constant Bitrate*). Pengaturan interval yang digunakan untuk durasi pengiriman sebanyak 1 paket setiap 1 paket. Proses pengiriman berjalan pada durasi waktu mulai pada detik ke-0 sampai pada detik ke 1000.

4.1.3.6 Perancangan pemrosesan data output

Network Simulator 2 (NS2) memiliki dua file, yaitu *file(.tr)* dan *file(.nam)*. Pada *file(.tr)* berfungsi sebagai catatan rekam jejak alur pengiriman paket data dan proses *routing* yang dilakukan masing-masing protokol dari awal hingga berakhirnya simulasi. Kemudian untuk *file(.nam)* berfungsi untuk menampilkan secara visual simulasi yang telah dilakukan dengan bentuk animasi sesuai skema yang nantinya diimplementasikan pada *tcl script* NS2. Perancangan pemrosesan data *output* dilakukan agar simulator dapat merekam jejak paket data di tiap *link* dalam topologi yang nantinya akan sesuai dengan apa yang diimplementasikan dalam sistem dan akan ditampilkan pada NS2 sebagai visualisasi terhadap skenario pengujian yang sudah dijalankan nantinya.

4.2 Implementasi Lingkungan Pengujian

Dalam subbab ini menjelaskan tentang tahapan-tahapan yang dilakukan dalam implementasi lingkungan pengujian sesuai dengan perancangan yang telah dilakukan pada bab sebelumnya. Implementasi lingkungan pengujian akan diterapkan pada Network Simulator 2.35.

4.2.1 Implementasi Protokol Routing

Pada implementasi protokol *routing* ini dijelaskan langkah-langkah yang dilakukan proses *patching* masing-masing protokol *routing* ke dalam modul NS2. Setelah *patching* selesai dilakukan kemudian dijalankan perintah *make install* pada masing-masing protokol ZRP dan HWMP.

4.2.1.1 Implementasi protokol *routing* ZRP

Patching protokol *routing* ZRP dilakukan terlebih dahulu agar protokol ZRP dapat dijalankan pada NS2 dikarenakan protokol ini sebelumnya tidak ada di dalam *ns-allinone*. *Patching* protokol ZRP dapat dilakukan dengan menjalankan perintah sebagai berikut :

```
Cd ns-allinone-2.35
Patch -p0 < zrp-ns235.patch
./install
Cd ns-2.35
Cp ns ns235-zrp
Sudo cp ns235-zrp /usr/local/bin
```

4.2.1.2 Implementasi protokol *routing* HWMP

Seperti proses *patching* yang sebelumnya telah dilakukan pada protokol ZRP, protokol HWMP juga harus dilakukan agar dapat dijalankan pada NS2 dikarenakan juga protokol ini sebelumnya tidak ada di dalam *ns-allinone*. *Patching* protokol ZRP dapat dilakukan dengan menjalankan perintah sebagai berikut :

```
Cd ns-allinone-2.35
Patch -p0 < hwmp_ns235.patch
./install
Cd ns-2.35
Cp ns ns235-hwmp
Sudo cp ns235-hwmp /usr/local/bin
```

4.2.2 Implementasi Pada Network Simulator 2

Implementasi dilakukan dengan metode simulasi dengan menggunakan perangkat lunak Network Simulator 2 (NS2). Dalam mengimplementasikan skenario pengujian yang telah dirancang sebelumnya, dibutuhkan beberapa konfigurasi dalam menerapkan skenario tersebut pada NS2. Beberapa konfigurasi tersebut diatur dalam *file script.tcl*. Implementasi pada penelitian ini dilakukan dengan menerapkan dua protokol *routing* yakni ZRP dan HWMP dalam lingkungan pengujian yang memiliki luas area yaitu 1000x1000 meter dan berbeda untuk jumlah *node* yang digunakan yaitu 30, 50, 70, 90, dan 110. Berdasarkan skenario tersebut maka diperlukan dua *file script.tcl* yang mewakili masing-masing protokol, yaitu *zrp.tcl* dan *hwmp.tcl* yang berisi konfigurasi beberapa parameter pengujian yang dibahas pada bab selanjutnya.

4.2.2.1 Implementasi parameter dan skenario pengujian

Dalam konfigurasi parameter lingkungan pengujian ini sebagai implementasi dari rancangan parameter lingkungan pengujian yang telah dibuat dan akan diterapkan pada NS2. Konfigurasi parameter pengujian diatur dalam *file* yang telah dibuat, yaitu *zrp.tcl* dan *hwmp.tcl* dengan pengaturan seperti pada Tabel 4.3 untuk protokol ZRP dan Tabel 4.4 untuk protokol HWMP berikut :

Tabel 4.3 Implementasi parameter pengujian ZRP

1	Set val(chan) Channel/WirelessChannel ;
2	set val(prop) Propagation/TwoRayGround ;
3	set val(netif) Phy/WirelessPhy ;
4	set val(mac) Mac/802_11 ;
5	set val(ifq) Queue/DropTail/PriQueue ;
6	set val(ll) LL ;
7	set val(ant) Antenna/OmniAntenna ;
8	set val(ifqlen) 50 ;
9	set val(nn) 30 ;
10	set val(rp) ZRP ;
11	set val(x) 1000 ;
12	set val(y) 1000 ;
13	set val(stop) 1000 ;
14	set val(t1) 0.0 ;
15	set val(t2) 0.0 ;
16	Agent/ZRP set radius 2 ;

Tabel 4.4 Implementasi parameter pengujian HWMP

1	set val(chan) Channel/WirelessChannel ;
2	set val(prop) Propagation/TwoRayGround ;
3	set val(netif) Phy/WirelessPhy ;
4	set val(mac) Mac/802_11 ;
5	set val(ifq) Queue/DropTail/PriQueue ;
6	set val(ll) LL ;
7	set val(an) Antenna/OmniAntenna ;
8	set val(ifqlen) 50 ;
9	set val(nn) 30 ;
10	set val(rp) HWMP ;
11	set val(x) 1000 ;
12	set val(y) 1000 ;
13	set val(stop) 1000.0 ;
14	set val(t1) 0.0 ;
15	set val(t2) 0.0 ;

Simulasi dilakukan dengan menggunakan dua protokol *routing* yang berbeda yaitu ZRP dan HWMP, maka dalam implementasinya dilakukan dengan mengubah masing-masing set val (rp) pada tabel nomor 10. Tabel nomor 10 merupakan pengaturan protokol *routing* yang digunakan dalam simulasi. Sehingga *script* dapat diubah sesuai dengan protokol yang diamati, yaitu ZRP dan HWMP. Luas area yang digunakan dalam penelitian ini sebesar 1000x1000 meter.



Konfigurasinya dilakukan dengan mengubah set val (x) dan set val (y) dalam tabel nomor 11 dan 12 yang merupakan pengaturan luas area berdasarkan sumbu x dan y. Pengaturan luas area dilakukan dengan cara memberikan nilai val (x) 1000 dan nilai val (y) 1000. Kemudian pada beberapa jumlah *node* yang berbeda, konfigurasi jumlah *node* dilakukan dengan cara mengubah set val (nn) menjadi 30, 50, 70, 90, dan 110. Sedangkan dalam parameter lainnya seperti durasi waktu set val(stop) bernilai konstan yaitu 1000 detik. Dalam pengaturan IFQ *length*, yakni set val(ifqlen) bernilai tetap sebesar 50 dan dalam pengaturan model antenna, yakni set val(ant) dan set val(prop) jenis yang digunakan adalah *OmniAntenna* dan *TwoRayGround*.

4.2.2.2 Implementasi topologi jaringan

Dari implementasi rancangan topologi yang telah dibuat kemudian akan dilakukan konfigurasi topologi jaringan. Skenario simulasi menyesuaikan dengan jumlah *node* yang berbeda-beda yaitu 30, 50, 70, 90, dan 110. Perancangan dilakukan sesuai dengan karakteristik MANET, yaitu memiliki *mobile node* yang bergerak bebas secara acak dengan arah dan posisi yang tidak menentu. Namun dalam penelitian ini juga dilakukan implementasi simulasi pergerakan *node* dengan kondisi statis atau diam. Konfigurasi topologi jaringan diatur dalam *file* *zrp.tcl* dan *hwmp.tcl* dengan pengaturan seperti pada Tabel 4.5 berikut:

Tabel 4.5 Implementasi topologi jaringan *node* dinamis

```

1  for {set i 0} {$i < 30} {incr i} {
2      set node_($i) [$ns node]
3      $node_($i) random-motion 1
4      $ns initial_node_pos $node_($i) 30
5  }
6      source mob-30.tcl
7      $node_(0) color blue
8  $ns at 0.0 "$node_(0) color blue"
9      $ns at 0.0 "$node_(0) label source"
10     $node_(1) color green
11     $ns at 0.0 "$node_(1) color green"
12     $ns at 0.0 "$node_(1) label Destination"
    
```

Tabel 1-4 menjelaskan inisialisasi awal untuk membuat sebuah *mobile node*. Jumlah *node* ditentukan pada baris nomor 1, ditandai dengan perulangan sebanyak jumlah *node* yang diinginkan. Tabel baris nomor 4 menjelaskan inisialisasi besaran visual *node* pada tampilan simulasi. Tabel baris nomor 6 menjelaskan tentang mobilitas pergerakan *node* yang digunakan untuk simulasi. Selanjutnya tabel baris nomor 7 adalah inisialisasi warna yang digunakan dalam visualisasi pada NS2. Untuk *source node* menggunakan warna biru sebagai penanda, sedangkan untuk *destination node* menggunakan warna hijau sebagai penandanya. Sedangkan pada kondisi keadaan topologi *node* statis, diimplementasikan pada Tabel 4.6 berikut :



Tabel 4.6 Implementasi topologi jaringan *node* statis

```

1  for {set i 0} {$i < 30} {incr i} {
2      set node_($i) [$ns_ node]
3      set xx [expr rand()*1000]
4      set yy [expr rand()*1000]
5      $node_($i) set X_ $xx
6      $node_($i) set Y_ $yy
7      $node_($i) set Z_ 0.0
8      $node_($i) random-motion 1
9      $ns_ initial_node_pos $node_($i) 30
10 }
11 $node_(0) color blue
12 $ns_ at 0.0 "$node_(0) color blue"
13 $ns_ at 0.0 "$node_(0) label source"
14 $node_(1) color green
15 $ns_ at 0.0 "$node_(1) color green"
16 $ns_ at 0.0 "$node_(1) label Destination"

```

Jumlah *node* ditentukan pada baris nomor 1, ditandai dengan perulangan sebanyak jumlah *node* yang diinginkan. Kemudian untuk mengatur luas area ditentukan pada baris tabel nomor 3 dan 4 yaitu 1000x1000 untuk sumbu x dan sumbu y. Baris tabel nomor 5, 6, dan 7 menjelaskan mobilitas *node* statis yang digunakan pada masing-masing sumbunya. Untuk sintaks selanjutnya seperti baris tabel nomor 8 sampai dengan 16 dijelaskan mengenai peletakan *node* dan inisialisasi warna yang digunakan dalam visualisasi pada NS2.

4.2.2.3 Implementasi Pergerakan *Node*

Konfigurasi pergerakan *node* dilakukan sebagai implementasi dari perancangan konfigurasi pergerakan *node* yang telah dibuat dan diatur dalam *file.tcl* seperti yang dijelaskan pada konfigurasi topologi jaringan diatas yang telah dibuat, yaitu mobil-30.tcl dan seterusnya sampai bernilai sejumlah *node* yang digunakan dan selanjutnya akan diterapkan pada Network Simulator 2. Dalam pengaturan mobilitas untuk pergerakan masing-masing *node*, maka dibuat *file.tcl* (dalam penelitian ini diberi nama mobil-30.tcl) yang akan dipanggil fungsinya seperti pada tabel baris nomor 6 dan diletakkan pada folder ns-allinone-2.35/ns-2.35/indep-utils/cmu-scen-gen/setdest, kemudian agar dapat dijalankan pindah *file* tersebut ke satu folder dengan *file* konfigurasi skenario pengujian. Untuk pembuatan mobilitas pergerakan *node*, ditampilkan seperti pada Gambar 4.4 berikut:



Gambar 4.4 Pembuatan mobilitas *node*



```

Universi #
Universi # nodes: 30, speed type: 1, min speed: 10.00, max speed: 20.00
Universi # avg speed: 14.43, pause type: 1, pause: 0.00, max x: 1000.00, max y: 1000.00
Universi #
Universi $node_(0) set X_ 205.297535215944
Universi $node_(0) set Y_ 635.753497536464
Universi $node_(0) set Z_ 0.000000000000
Universi $node_(1) set X_ 324.086661546031
Universi $node_(1) set Y_ 568.180287066265
Universi $node_(1) set Z_ 0.000000000000

```

Gambar 4.5 Hasil file pembuatan mobilitas node

Pada Gambar 4.4 di atas ditunjukkan mengenai pembuatan *file.tcl* yang berfungsi sebagai mobilitas pergerakan *node*. Setelah masuk pada folder yang disebutkan di atas, maka dijalankan perintah `./setdest -v 2 -n 30 -m 10.0 -M 20.0 -t 1000.0 -x 1000.0 -y 1000.0 > mobil-30.tcl`. Setdest digunakan untuk menghasilkan posisi, kecepatan, dan arah bergerak masing-masing *node*. `-n 30` menunjukkan banyaknya jumlah *node* yang digunakan. `-m 10.0` merupakan kecepatan minimum yang digunakan pada waktu simulasi, yaitu sebesar 10 m/s. kemudian untuk kecepatan maksimum adalah 20 m/s yang ditunjukkan oleh `-M 20.0`, `-t 1000.0` merupakan waktu yang digunakan selama simulasi, yaitu sebesar 1000 detik. Kemudian untuk `-x 1000` dan `-y 1000` digunakan sebagai luas area yang digunakan yaitu 1000x1000 meter. Kemudian akan disimpan pada *file.tcl* seperti pada Gambar 4.5 diatas yang pada penelitian ini diberi nama *mobil-30.tcl*. Perintah di atas terus dijalankan sesuai dengan banyaknya variasi kepadatan *node* yang digunakan, misalkan pada node 50, maka perintah yang digunakan adalah `./setdest -v 2 -n 50 -m 10.0 -M 20.0 -t 1000.0 -x 1000.0 -y 1000.0 > mobil-50.tcl`. Konfigurasi pergerakan *node* pada *mobil-30.tcl* tersebut akan menghasilkan topologi dengan luas area 1000x1000 dengan 30 *node* (dalam hal ini diambil sample 30 *node*), yang di distribusikan secara acak oleh koordinat XY (Z). setelah dilakukan inialisasi posisi awal *node*, kemudian akan ditentukan *destination* dan kecepatan masing-masing *node*. Kemudian untuk inialisasi jarak (jumlah hop), akan di setting pada objek *god*, ditunjukkan pada Tabel 4.7 berikut ini :

Tabel 4.7 Inialisasi jarak oleh objek god

1	Set topo [new Topography]
2	\$topo load_flatgrid \$val(x) \$val(y)
3	set god_ [create-god \$val(nn)]

Objek *god* digunakan untuk menyimpan jumlah array hop terpendek yang diperlukan untuk mencapai satu *node* ke *node* yang lain. *Node* akan bergerak selama skenario simulasi, yaitu 1000 detik. Selama simulasi berjalan, informasi jarak hop akan berubah.

4.2.2.4 Implementasi pengiriman paket

Konfigurasi pengiriman paket dilakukan sebagai implementasi dari perancangan pengiriman paket data dari *source node* menuju *destination node*.



Dalam simulasi yang dilakukan pada NS2, konfigurasi pengiriman paket diatur dalam *file* *zrp.tcl* dan *hwmp.tcl* dengan pengaturan seperti pada Tabel 4.8 berikut:

Tabel 4.8 Implementasi pengiriman paket

1	set udp [new Agent/UDP]
2	\$ns attach-agent \$node_(0) \$udp
3	set null [new Agent/Null]
4	\$ns attach-agent \$node_(1) \$null
5	\$ns connect \$udp \$null
6	\$udp set packetSize 512

Tahapan pertama yang dilakukan untuk membangun sebuah konektivitas jaringan dalam NS2 adalah *setup agent*. Dalam penelitian ini protokol jaringan yang digunakan adalah UDP. Pengaturan dilakukan pada tabel baris nomor 1, yakni `set udp [new Agent/UDP]` dimana Agent/UDP ini berperan sebagai sumber pengiriman paket data yang dilakukan dengan transport jaringan UDP. Kemudian untuk tahapan pengaturan *node* yang dipilih menjadi Agent/UDP dilakukan dengan konfigurasi seperti pada tabel baris nomor 2, yaitu `$ns attach-agent $node_(0) $udp` dimana *node* 0 yang dipilih menjadi *source node*. Pengaturan besar paket yang dikirimkan *source node* terdapat pada tabel baris nomor 6, yakni 512 byte sesuai dengan perancangan sebelumnya.

Kemudian untuk pengaturan *destination node* dilakukan dengan membuat agen baru pada tabel baris nomor 4, yaitu `set null [new Agent/Null]`. Berbeda dengan TCP yang harus melakukan sinkronisasi dalam menerima setiap paket yang datang, dalam UDP Agent/Null yang berarti setiap agen bebas dalam menerima paket yang datang. Dalam mengatur *node* menjadi Agent/Null, yaitu dengan `$ns attach-agent $node_(1) $null` dimana *node* 1 yang dipilih menjadi *destination node*. Dalam membangun koneksi antar keduanya, yaitu *source node* yang merupakan Agent/UDP dan *destination node* yang merupakan Agent/Null dilakukan seperti tabel baris nomor 5 dengan *script* `$ns connect $udp $null`.

Tabel 4.9 Implementasi model pengiriman paket

1	set cbr [new Application/Traffic/CBR]
2	\$cbr attach-agent \$udp
3	\$cbr set packetSize 512
4	\$cbr set interval 1.0
5	\$cbr set random null
6	\$ns at 0.0 "\$cbr start"
7	\$ns at 1000.0 "\$cbr stop"

Pada Tabel 4.9 diatas dijelaskan proses transmisi data protokol transport UDP memerlukan konfigurasi tambahan untuk menentukan model transmisi datanya. Model transmisi yang digunakan yaitu CBR(*Constant Bitrate*). Pengaturan aplikasi CBR diatur seperti tabel baris nomor 1 dan 2. Inisialisasi awal dilakukan dengan `set cbr [new Application/Traffic/CBR]` yang disematkan pada agen UDP yang telah diatur sebelumnya. Kemudian dalam pengaturan ukuran paket sebesar 512 byte

diterapkan pada tabel baris nomor 3, yakni dengan *script* \$cbr set packetSize_512. Dalam pengaturan interval yang digunakan adalah sebesar 1.0 yang berarti setiap waktu 1 detik akan mengirimkan sebanyak 1 paket, diterapkan seperti tabel baris nomor 4, yakni \$cbr set interval_1.0. Durasi waktu pengiriman diatur pada tabel baris nomor 6 dan 7 yang dimulai pada detik ke-0 dan diakhiri pada detik ke -1000.

4.2.2.5 Implementasi pemrosesan data *output*

Konfigurasi pemrosesan data *output* dilakukan agar dapat memproses *output* sesuai skenario simulasi yang telah dirancang. Konfigurasi pemrosesan data *output* dilakukan dengan mengubah *file* *zrp.tcl* dan *hwmp.tcl*. Berikut merupakan Tabel 4.10 untuk konfigurasi pemrosesan data *output* dalam *file* *script.tcl*.

Tabel 4.10 Implementasi pemrosesan data *output*

1	set tracefile [open outzrp.tr w]
2	\$ns trace-all \$tracefile
3	set namfile [open outzrp.nam w]
4	\$ns namtrace-all \$namfile

NS2 memiliki dua *file output*, yaitu *file(.tr)* yang merupakan *network trace file* berisi catatan rekam jejak alur pengiriman paket data dan proses *routing* yang direkap dengan detail dari awal hingga berakhirnya simulasi. *Output* yang kedua yaitu *file(.nam)* adalah *file network animation* yang akan menampilkan visualisasi simulasi dalam bentuk animasi sesuai skema yang telah diimplementasikan pada *tcl script* NS2. Konfigurasi pertama adalah *set* nama *output* dari *file(.tr)* maupun *(.nam)* seperti dalam baris nomor 1 dan 3, setelah simulasi dieksekusi akan keluar sebuah *file* *outzrp.tr* sebagai *output network trace* dan *outzrp.nam* sebagai *output network animation*. Selanjutnya dalam mengatur simulator agar dapat merekam jejak paket data di tiap *link* dalam topologi, diterapkan dengan *script* \$ns trace-all \$tracefile dalam tabel baris nomor 2. Dalam tabel baris nomor 4 yaitu \$ns namtrace-all \$namfile berfungsi merekam seluruh detail simulasi yang nantinya dapat diputar dalam bentuk animasi.

4.3 Pengujian

Dalam subbab pengujian ini dibahas mengenai skenario pengujian serta teknik dalam pengambilan data berdasarkan simulasi protokol *routing* yang telah dilakukan berdasarkan rancangan skenario lingkungan pengujian yang dibuat. Dari hasil simulasi didapatkan beberapa data *output* yang kemudian diukur berdasarkan parameter-parameter yang telah ditentukan.

4.3.1 Skenario Pengujian

Pada pengukuran data hasil pengujian dilakukan dengan parameter yang telah ditentukan, yaitu *throughput*, *packet delivery ratio* (PDR), *normalized routing load*, *end-to-end delay* dan *convergence time*. Kedua protokol tersebut diuji dalam satu skenario lingkungan pengujian dengan luas area 1000x1000 meter. Skenario

lingkungan pengujian tersebut diuji berdasarkan pengaruh variasi kepadatan *node* yang berjumlah 30, 50, 70, 90, dan 110. Paket data yang digunakan sebesar 512 byte dengan interval pengiriman setiap satu detik mengirimkan satu paket data yang berjalan dalam durasi 1000 second. Pengujian skenario masing-masing dilakukan sebanyak lima kali untuk *source* dan *destination node* yang berbeda dan selanjutnya diambil nilai rata-ratanya untuk menambah keakuratan data analisis. Hal tersebut dilakukan karena topologi MANET yang dinamis sehingga menyebabkan data yang dihasilkan cenderung berbeda di tiap pengujian.

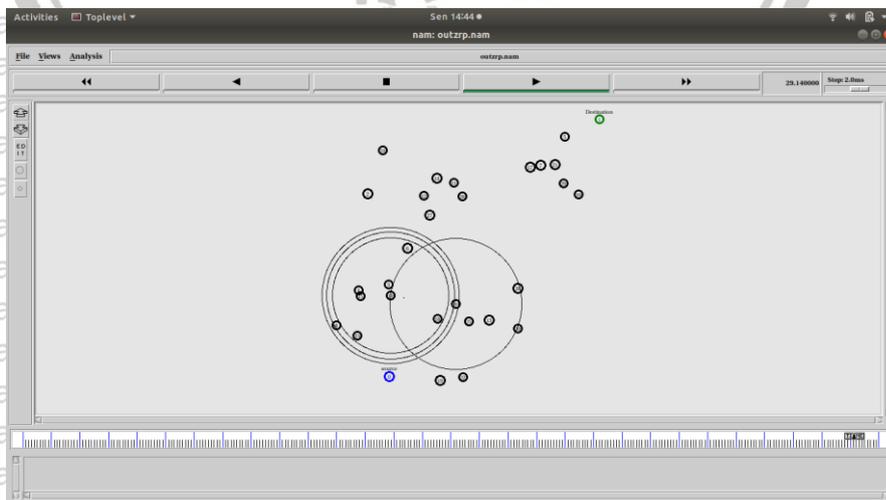
4.3.2 Eksekusi Simulasi Protokol Routing MANET

Sebagai implementasi dari beberapa scenario lingkungan pengujian pada protokol *routing* ZRP dan HWMP yang telah dirancang dan dikonfigurasi dalam NS-2, maka dilakukan simulasi protokol *routing* MANET. Simulasi ini dilakukan dengan cara mengeksekusi *file* *zrp.tcl* dan *hwmp.tcl*. Hasil eksekusi berjalan seperti pada Gambar 4.6 berikut:

```
novanmf@novanmf-X455LF:~/zrp$ ns zrp-30.tcl
num_nodes is set 30
INITIALIZE THE LIST xListHead
Starting Simulation...
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
```

Gambar 4.6 Hasil eksekusi file script *zrp.tcl*

Dalam menjalankan *file* *zrp.tcl* dan *hwmp.tcl* yang telah dikonfigurasi pada NS2, langkah yang diperlukan yakni dengan menjalankan perintah “ns” dilanjutkan dengan nama *file* *zrp.tcl* dan *hwmp.tcl* yang telah dibuat. Jika telah berhasil dieksekusi akan keluar beberapa notifikasi salah satunya berupa tulisan “*sorting list done*”. Setelah itu secara otomatis *output* berupa *network animation file* (.nam) akan tampil seperti pada Gambar 4.7 berikut:



Gambar 4.7 Hasil output *network animation file*

Network animation file (.nam) menampilkan jalannya simulasi secara visual sesuai dengan rancangan yang telah dikonfigurasi pada *tcl script* pada NS2. Sedangkan *output* lainnya, yaitu *network trace file* (.tr) akan menampilkan seluruh rekaman data lalu lintas jaringan selama jalannya simulasi secara detail seperti pada Gambar 4.8 berikut:

```

M 0.00000 26 (417.18, 508.03, 0.00), (218.69, 529.99), 5.92
M 0.00000 27 (376.87, 119.71, 0.00), (834.76, 834.77), 9.20
M 0.00000 28 (760.62, 782.10, 0.00), (238.90, 207.14), 4.76
M 0.00000 29 (703.58, 896.15, 0.00), (810.77, 595.69), 0.36
s 0.000000000_0_AGT --- 0 cbr 512 [0 0 0 0] ----- [0:0 1:0 32 0] [0] 0 0
r 0.000000000_0_RTR --- 0 cbr 512 [0 0 0 0] ----- [0:0 1:0 32 0] [0] 0 0
s 0.000339577_0_RTR --- 2 ZRP 30 [0 0 0 0] ----- [0:255 -1:255 1 0] ----- [0:255 -1:255 1 0]
s 0.000634577_0_MAC --- 2 ZRP 88 [0 ffffffff 8 800] ----- [0:255 -1:255 1 0] ----- [0:255 -1:255 1 0]
r 0.040960000_0_AGT --- 3 cbr 512 [0 0 0 0] ----- [0:0 1:0 32 0] [1] 0 0
r 0.040960000_0_RTR --- 3 cbr 512 [0 0 0 0] ----- [0:0 1:0 32 0] [1] 0 0
s 0.081920000_0_AGT --- 4 cbr 512 [0 0 0 0] ----- [0:0 1:0 32 0] [2] 0 0
r 0.081920000_0_RTR --- 4 cbr 512 [0 0 0 0] ----- [0:0 1:0 32 0] [2] 0 0
s 0.088122671_1_RTR --- 5 ZRP 30 [0 0 0 0] ----- [1:255 -1:255 1 0] ----- [1:255 -1:255 1 0]
r 0.088637671_1_MAC --- 5 ZRP 88 [0 ffffffff 1 800] ----- [1:255 -1:255 1 0] ----- [1:255 -1:255 1 0]
s 0.089342115_14_MAC --- 5 ZRP 30 [0 ffffffff 1 800] ----- [1:255 -1:255 1 0] ----- [1:255 -1:255 1 0]
r 0.089342227_15_MAC --- 5 ZRP 30 [0 ffffffff 1 800] ----- [1:255 -1:255 1 0] ----- [1:255 -1:255 1 0]
r 0.089367115_14_RTR --- 5 ZRP 30 [0 ffffffff 1 800] ----- [1:255 -1:255 1 0] ----- [1:255 -1:255 1 0]
s 0.089367115_14_RTR --- 0 ZRP 30 [0 0 0 0] ----- [14:255 1:255 1 1] ----- [14:255 1:255 1 1]
r 0.089367227_15_RTR --- 5 ZRP 30 [0 ffffffff 1 800] ----- [1:255 -1:255 1 0] ----- [1:255 -1:255 1 0]
s 0.089367227_15_RTR --- 7 ZRP 30 [0 0 0 0] ----- [15:255 1:255 1 1] ----- [15:255 1:255 1 1]
r 0.089542115_14_MAC --- 0 ARP 86 [0 ffffffff e 806] ----- [REQUEST 14/14 0/1]
r 0.090230559_15_MAC --- 0 ARP 28 [0 ffffffff e 806] ----- [REQUEST 14/14 0/1]
r 0.090230636_15_MAC --- 0 ARP 28 [0 ffffffff e 806] ----- [REQUEST 14/14 0/1]
r 0.090230693_29_MAC --- 0 ARP 28 [0 ffffffff e 806] ----- [REQUEST 14/14 0/1]
r 0.090230804_28_MAC --- 0 ARP 28 [0 ffffffff e 806] ----- [REQUEST 14/14 0/1]
s 0.090230836_15_MAC --- 0 ARP 86 [0 ffffffff f 806] ----- [REQUEST 15/15 0/1]
r 0.091009157_14_MAC --- 0 ARP 28 [0 ffffffff f 806] ----- [REQUEST 15/15 0/1]
r 0.091009192_14_MAC --- 0 ARP 28 [0 ffffffff f 806] ----- [REQUEST 15/15 0/1]
r 0.091009361_28_MAC --- 0 ARP 28 [0 ffffffff f 806] ----- [REQUEST 15/15 0/1]
s 0.091279192_14_MAC --- 0 RTS 44 [52e e 1 0]
r 0.091356916_22_RTR --- 8 ZRP 30 [0 0 0 0] ----- [22:255 -1:255 1 0] ----- [22:255 -1:255 1 0]
r 0.091631636_14_MAC --- 0 RTS 44 [52e e 1 0]
s 0.091641636_14_MAC --- 0 CTS 38 [3f4 1 0 0]
r 0.091791916_22_MAC --- 8 ZRP 88 [0 ffffffff 16 800] ----- [22:255 -1:255 1 0] ----- [22:255 -1:255 1 0]
r 0.091946079_14_MAC --- 0 CTS 38 [3f4 1 0 0]
r 0.091956079_14_MAC --- 0 ARP 86 [13a e 1 806] ----- [REPLY 1/1 14/14]
r 0.092496195_21_MAC --- 8 ZRP 30 [0 ffffffff 16 800] ----- [22:255 -1:255 1 0] ----- [22:255 -1:255 1 0]

```

Gambar 4.8 Hasil output network trace file

4.4 Pengambilan Data

Pengambilan data dilakukan dengan cara mengolah data *output network trace file* yang berisi rincian rekaman data lalu lintas jaringan selama proses simulasi dijalankan. Pengolahan data dijalankan dengan pemrograman AWK yang memproses data berdasarkan struktur kolom pada *output network trace file*. Berikut merupakan Gambar 4.9 yang menjelaskan struktur data *output network trace file* pada NS2 beserta penjelasannya.

```

s 0.081920000_0_AGT --- 4 cbr 512 [0 0 0 0] ----- [0:0 1:0 32 0] [2] 0 0
r 0.081920000_0_RTR --- 4 cbr 512 [0 0 0 0] ----- [0:0 1:0 32 0] [2] 0 0
s 0.088122671_1_RTR --- 5 ZRP 30 [0 0 0 0] ----- [1:255 -1:255 1 0] ----- [1:255 -1:255 1 0]
r 0.088637671_1_MAC --- 5 ZRP 88 [0 ffffffff 1 800] ----- [1:255 -1:255 1 0] ----- [1:255 -1:255 1 0]
s 0.089342115_14_MAC --- 5 ZRP 30 [0 ffffffff 1 800] ----- [1:255 -1:255 1 0] ----- [1:255 -1:255 1 0]
r 0.089342227_15_MAC --- 5 ZRP 30 [0 ffffffff 1 800] ----- [1:255 -1:255 1 0] ----- [1:255 -1:255 1 0]
r 0.089367115_14_RTR --- 5 ZRP 30 [0 ffffffff 1 800] ----- [1:255 -1:255 1 0] ----- [1:255 -1:255 1 0]

```

Gambar 4.9 Struktur output network trace file

1. Pada kolom pertama berisi keterangan mengenai status paket yang dikirim. Status paket dapat berupa symbol s (*send*) yaitu paket telah dikirim, r (*received*) paket telah diterima, d (*dropped*) paket di drop, f (*forwarded*) paket diteruskan dan c (*collison*) paket bertabrakan pada lapisan MAC.
2. Pada kolom kedua merupakan detail waktu aktivitas yang dilakukan dalam satuan waktu *second*.

3. Kolom ketiga menjelaskan identitas *node* tempat terjadinya aktivitas *routing*.
4. Kolom keempat menjelaskan dimana letak tingkatan lapisan jaringan tempat suatu aktivitas terjadi contohnya seperti AGT (*agent*) terdapat pada lapisan aplikasi, RTR (*routing*), MAC, IFQ (*interface queue*) merupakan antrian paket keluar yang terdapat diantara lapisan *link* dan MAC, LL (*link layer*) dan PHY (*physical*).
5. Pada kolom keenam merupakan identitas paket dalam bentuk *sequence number*.
6. Kolom ketujuh merupakan penjelasan tipe paket dimana dapat berbentuk data stream CBR (*constant bitrate*), paket *routing* (ZRP dan HWMP), ARP, RTS dan CTS.
7. Pada kolom kedelapan menjelaskan besaran paket (paket size) yang di proses.
8. Kolom kesembilan berisi alamat *source node* dan tujuan MAC yang masing-masing adalah "0" dan "ffffff". Protokol MAC mengasumsikan bahwa penundaan diatas saluran nirkabel yang mendasari dari nilai 0. Nilai 800 merupakan IP paket yang berjalan diatas jaringan *Ethernet*.
9. Pada kolom kesepuluh berisi alamat IP *source* dan *destination node* beserta port masing-masing, juga berisi *time to live* dan alamat dari *hop* berikutnya.
10. Pada kolom keduabelas berisi alamat IP *source* dan *destination node* beserta port masing-masing, juga berisi *time to live* dan alamat dari *hop* tujuan.

Berdasarkan penjelasan struktur data *output network trace file* diatas, maka pemrograman AWK sangat berperan dalam mengolah dan untuk akumulasi data berdasarkan kolom-kolom informasi yang tertera berdasarkan rumusan matematis dari parameter yang ditentukan seperti *throughput*, *packet delivery ratio* (PDR), *normalized routing load*, *end-to-end delay* dan *convergence time*.

4.4.1 Pengambilan Data *Packet Delivery Ratio*

Proses pengambilan data *packet delivery ratio* dalam *output network trace file* NS2 dilakukan dengan perancangan pemrograman AWK berdasarkan rumusan matematis dari *packet delivery ratio*. Kode program disimpan dengan format ekstensi *file* AWK yang dalam penelitian ini yaitu *pdr.awk*. Berikutnya untuk proses eksekusi program AWK *packet delivery ratio* dilakukan dengan menjalankan perintah `awk-f pdr.awk (nama file AWK) outzrp.tr (nama output network trace file)`. Maka hasil pengolahan data *packet delivery ratio* akan tertera seperti Gambar 4.10 berikut:

```
novanmf@novanmf-X455LF:~/zrp$ awk -f pdr.awk outzrp.tr
total paket dikirim:733.000 total paket diterima:203.000, PDR:0.277
```

Gambar 4.10 Hasil eksekusi program AWK *packet delivery ratio*

4.4.2 Pengambilan Data *Throughput*

Proses pengambilan data *throughput* dalam *output network trace file* NS2 dilakukan dengan perancangan pemrograman AWK berdasarkan rumusan matematis dari *throughput*. Selanjutnya kode program disimpan dengan format eksekusi *file* AWK yang dalam penelitian ini, yaitu *throughput.awk*. Dalam proses eksekusi program AWK *throughput* dilakukan dengan menjalankan perintah `awk -f throughput.awk (nama file AWK) outzrp.tr (nama output network trace file)` maka hasil pengolahan data *throughput* akan tertera seperti pada Gambar 4.11 berikut:

```
novanmf@novanmf-X455LF:~/zrp$ awk -f throughput.awk outzrp.tr
Average Throughput[kbps] = 54.329      StartTime=0.000      StopTime=0.000      totalpaket=203000.000
novanmf@novanmf-X455LF:~/zrp$
```

Gambar 4.11 Hasil eksekusi program AWK *throughput*

4.4.3 Pengambilan Data *End to end Delay*

Pengambilan data *end to-end delay* dalam *output network trace file* NS2 dilakukan dengan perancangan pemrograman AWK berdasarkan rumusan matematis dari *end to-end delay*. Selanjutnya kode program disimpan dengan format ekstensi *file* AWK yang dalam penelitian ini, yaitu *end2end.awk*. Berikutnya dalam proses eksekusi program AWK *endto-end delay* dilakukan dengan menjalankan perintah `awk -f end2end.awk (nama file AWK) outzrp.tr (nama output network trace file)`. Hasil pengolahan data *end to-end delay* akan tertera seperti pada Gambar 4.12 berikut:

```
novanmf@novanmf-X455LF: ~/zrp
File Edit View Search Terminal Help
novanmf@novanmf-X455LF:~/zrp$ awk -f end2end.awk outzrp.tr
average delay = 5.078novanmf@novanmf-X455LF:~/zrp$
```

Gambar 4.12 Hasil eksekusi program AWK *end to-end delay*

4.4.4 Pengambilan Data *Normalized Routing Load*

Pengambilan data *normalized routing load* dalam *output network trace file* NS2 dilakukan dengan perancangan pemrograman AWK berdasarkan rumusan matematis dari *normalized routing load*. Selanjutnya kode program disimpan dengan format ekstensi *file* AWK yang dalam penelitian ini, yaitu *nrl.awk*. Dalam proses eksekusi program AWK *normalized routing load* dilakukan dengan menjalankan perintah `awk -f nrl.awk (nama file AWK) outzrp.tr (nama output network trace file)`. Hasil pengolahan data *normalized routing load* akan tertera seperti Gambar 4.13 berikut :

```
novanmf@novanmf-X455LF:~/zrp$ awk -f nrl.awk outzrp.tr
Total paket routing = 4192.000 Total paket diterima = 203.000
Normalized Routing Load = 20.650
```

Gambar 4.13 Hasil eksekusi program AWK *normalized routing load*

4.4.5 Pengambilan Data *Convergence Time*

Pengambilan data *convergence* dalam *output network trace file* NS2 dilakukan dengan perancangan pemrograman AWK berdasarkan rumusan matematis dari *convergence time*. Selanjutnya kode program disimpan dengan format ekstensi *file AWK* yang dalam penelitian ini, yaitu *konv.awk*. Dalam proses eksekusi program AWK: *convergence* dilakukan dengan menjalankan perintah `awk -f convergence.awk (nama file AWK) outzrp.tr (nama output network trace file)`. Hasil pengolahan data *normalized routing load* akan tertera seperti Gambar 4.14 berikut :

```
novanmf2@novanmf-X455LF:~/hwmp$ awk -f convergence.awk outhwmp.tr
Total Waktu Konvergensi : 763.115
```

Gambar 4.14 Hasil eksekusi program AWK *convergence time*



BAB 5 HASIL DAN PEMBAHASAN

Dalam bab ini berisi mengenai penjabaran data hasil pengujian beserta analisis dan pembahasan perbandingan kinerja protokol *routing* ZRP dan HWMP pada jaringan MANET berdasarkan skenario lingkungan pengujian yang telah diimplementasikan pada bab sebelumnya. Analisis hasil bertujuan untuk mengetahui perbandingan kinerja masing-masing protokol berdasarkan parameter pengujian yang digunakan yaitu *throughput*, *packet delivery ratio* (PDR), *normalized routing load*, *end-to-end delay* dan *convergence time*.

5.1 Pengaruh Variasi Kepadatan Node Protokol *Routing* ZRP dan HWMP Pada Topologi Statis dan Dinamis

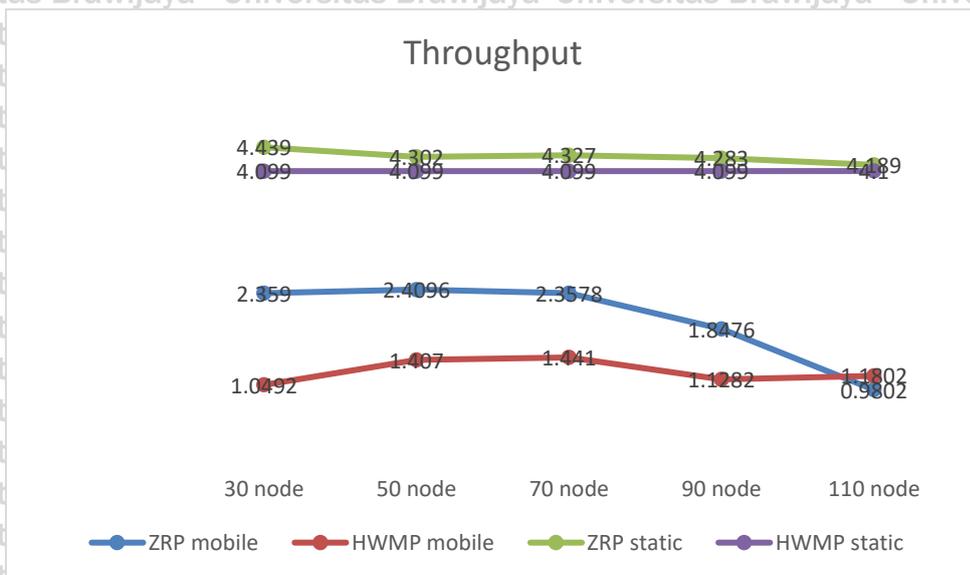
Pada subbab ini berisi tabel grafik perbandingan yang merupakan hasil pengujian dari kinerja protokol *routing* ZRP dan HWMP. Hasil pengujian didapat berdasarkan implementasi dari skenario lingkungan pengujian dengan luas area 1000x1000 meter yang dijalankan pada variasi jumlah *node* yang berbeda-beda, yaitu 30, 50, 70, 90 dan 110. Pengukuran yang dilakukan berdasar parameter yang telah disebutkan sebelumnya, yaitu *throughput*, *packet delivery ratio* (PDR), *normalized routing load*, *end-to-end delay* dan *convergence time*.

5.1.1 *Throughput*

Berdasar pengujian yang telah dilakukan dalam skenario luas area 1000x1000 meter dengan *node* yang berjumlah 30, 50, 70, 90, dan 110 diperoleh rata-rata kecepatan *throughput* pada masing-masing protokol *routing* yang dijabarkan seperti pada Tabel 5.1 dan Gambar 5.11 berikut ini :

Tabel 5.1 Hasil pengujian *throughput* ZRP dan HWMP

Jumlah <i>node</i>	30 <i>node</i>	50 <i>node</i>	70 <i>node</i>	90 <i>node</i>	110 <i>node</i>
ZRP <i>mobile</i>	2,359	2,409	2,357	1,847	0,98
HWMP <i>mobile</i>	1,049	1,407	1,441	1,128	1,18
ZRP <i>static</i>	4,439	4,302	4,327	4,283	4,189
HWMP <i>static</i>	4,099	4,099	4,099	4,099	4,1



Gambar 5.11 Grafik *throughput* perbandingan kinerja ZRP dan HWMP

Pada Gambar 5.11 terlihat dalam skenario lingkungan pengujian dengan luas area 1000x1000 meter diperoleh bahwa nilai rata-rata *throughput* yang dimiliki protokol HWMP berada di bawah nilai rata-rata *throughput* yang dimiliki protokol ZRP. Nilai *throughput* protokol HWMP cenderung lebih stabil walaupun terjadi penambahan *node*, Sedangkan nilai *throughput* yang dimiliki protokol ZRP cenderung tidak stabil, jika *node* yang digunakan bertambah banyak maka nilai *throughput* akan semakin menurun. Nilai rata-rata kecepatan *throughput* yang dimiliki HWMP memiliki nilai masing-masing *node* yang digunakan yaitu posisi *node* 30 sebesar 1,0492, *node* 50 sebesar 1,407, *node* 70 sebesar 1,441, *node* 90 sebesar 1,1282 dan *node* 110 sebesar 1,1802. Sedangkan pada ZRP nilai *throughput* pada posisi *node* 30 sebesar 2,359, *node* 50 sebesar 2,4096, *node* 70 sebesar 2,3578, *node* 90 sebesar 1,8476 dan *node* 110 sebesar 0,9802. Penurunan nilai *throughput* terjadi pada saat posisi *node* 90 dan penurunan drastis terjadi pada posisi *node* 110. Sedangkan pada keadaan *node* statis diperoleh hasil masing-masing *node* untuk ZRP adalah 4,439 pada *node* 30, 4,302 *node* 50, 4,327 *node* 70, 4,283 *node* 90, dan 4,189 *node* 110. Untuk nilai rata-rata *throughput* pada *node* statis protokol ZRP lebih tinggi dibandingkan dengan HWMP yang memiliki nilai *throughput* lebih rendah pada masing-masing *node* yang digunakan yaitu 4,099 pada *node* 30, 4,099 *node* 50, 4,099 *node* 70, 4,099 *node* 90, 4,1 *node* 110.

Berdasarkan hasil pengujian tersebut dapat disimpulkan variasi kepadatan pada topologi dinamis sangat mempengaruhi *throughput* dari protokol ZRP yang mengalami penurunan di setiap pertambahan jumlah *node* yang digunakan. Sedangkan pada HWMP pengaruh variasi kepadatan *node* menyebabkan perubahan nilai *throughput* naik turun walaupun selisih nilai yang didapat tidak berjarak jauh dengan masing-masing variasi *node* yang digunakan. Penurunan nilai secara drastis terjadi pada saat kepadatan *node* sebesar 90 dan 110. Protokol HWMP memiliki nilai rata-rata *throughput* yang cenderung tidak menurun jika



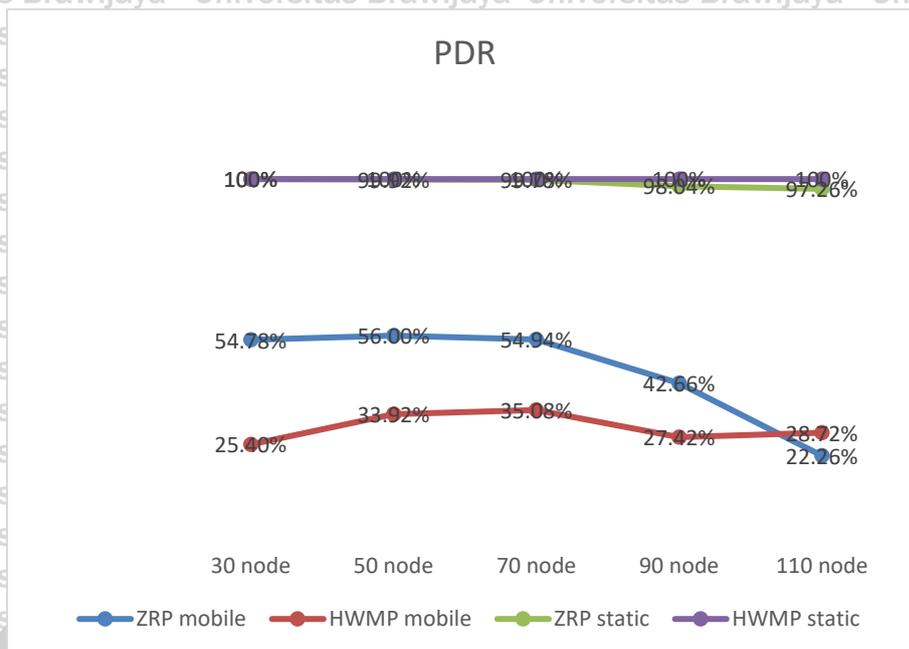
variasi kepadatan *node* di tambah daripada ZRP karena proses akumulasi jalur yang berjalan lebih efisien, sehingga dapat memaksimalkan jumlah paket yang terkirim pada durasi waktu yang ditentukan. Keunggulan HWMP dalam menentukan rute pengiriman paket adalah jalur yang ditentukan dapat dipastikan bebas *looping* serta dapat memelihara beberapa jalur alternatif menuju *destination node*. Protokol ZRP memiliki kecepatan *throughput* yang cukup tinggi namun hanya berjalan pada topologi yang sederhana hal tersebut disebabkan oleh proses akumulasi jalur ZRP akan menjadi kompleks jika jumlah *node* bertambah banyak dan otomatis akan mengalami penurunan nilai *throughput* di setiap penambahan variasi jumlah *node* yang digunakan tersebut. Sedangkan pada topologi statis didapatkan hasil lebih tinggi daripada penggunaan *node mobile* untuk kedua protokol, dikarenakan pada *node* statis jarak antar *node* yang digunakan pada posisi tetap, tidak bergerak menjauh maupun mendekat seperti *node mobile*, sehingga proses *routing* menuju *destination node* dapat tercapai lebih maksimal daripada *node* dinamis. Nilai *throughput* ZRP pada kondisi topologi statis lebih tinggi daripada HWMP dikarenakan proses *routing* ZRP yang berbasis *zone* akan melakukan perutean jalur dengan cara membuat *hop* dengan luas yang ditentukan sehingga akan mendapatkan hasil yang lebih besar daripada HWMP yang dalam perutean jalur *routing* menggunakan konsep broadcast dan bebas *looping* seperti yang telah dijelaskan sebelumnya.

5.1.2 Packet Delivery Ratio

Berdasarkan pengujian yang telah dilakukan dalam skenario luas area 1000x1000 meter dengan *node* yang berjumlah 30, 50, 70, 90, dan 110 diperoleh rata-rata nilai hasil *packet delivery ratio* (PDR) pada masing-masing protokol *routing* yang dijabarkan pada Tabel 5.2 dan Gambar 5.12 berikut ini:

Tabel 5.2 Hasil pengujian *packet delivery ratio* ZRP dan HWMP

Jumlah <i>node</i>	30 <i>node</i>	50 <i>node</i>	70 <i>node</i>	90 <i>node</i>	110 <i>node</i>
ZRP <i>mobile</i>	54,78%	56,00%	54,94%	42,66%	22,26%
HWMP <i>mobile</i>	25,40%	33,92%	35,08%	27,42%	28,72%
ZRP <i>static</i>	100%	99,92%	99,78%	98,04%	97,26%
HWMP <i>static</i>	100%	100%	100%	100%	100%



Gambar 5.12 Grafik *packet delivery ratio* perbandingan kinerja ZRP dan HWMP

Gambar 5.12 di atas mengenai skenario lingkungan pengujian dengan luas area 1000x1000 meter pada topologi dinamis protokol *routing* HWMP memiliki nilai rata-rata *packet delivery ratio* di bawah ZRP pada topologi dinamis, yakni sebesar 25,40%, 33,92%, 35,08%, 27,42% dan 28,72% di kepadatan *node* yang berjumlah 30, 50, 70, 90, dan 110. Sedangkan pada ZRP memiliki nilai *packet delivery ratio* yang lebih tinggi daripada HWMP, yaitu pada posisi *node* 30 sebesar 54,78%, *node* 50 sebesar 56,00%, *node* 70 sebesar 54,94%, kemudian terjadi penurunan pada *node* 90 menjadi 42,66% dan terakhir pada *node* 110 bernilai paling rendah menjadi sebesar 22,26%. Sedangkan pada topologi statis didapatkan hasil *routing* PDR sebesar 100% pada *node* 30, 99,92% pada *node* 50, 99,78% pada *node* 70, 98,04% pada *node* 90, dan 97,26% pada *node* 110 untuk *routing* ZRP. Sedangkan pada HWMP didapatkan nilai PDR sebesar 100% untuk masing-masing variasi kepadatan *node*.

Dari hasil pengukuran tersebut terlihat pada protokol *routing* ZRP variasi kepadatan *node* yang bergerak secara *mobile* pada topologi dinamis serta semakin padat membuat nilai *packet delivery ratio* mengalami penurunan. Hal tersebut disebabkan karena jumlah *node* yang tersebar sangat berpengaruh dalam proses *routing* ZRP yang berjalan secara proaktif memelihara dan memperbaiki topologi dalam zona radius lokalnya. Sehingga menyebabkan proses pengiriman paket menjadi terhambat ketika *node* dalam jumlah yang besar tersebar pada ruang lingkup yang cukup luas. Sedangkan pada HWMP variasi kepadatan jumlah *node* tidak menyebabkan penurunan nilai seperti ZRP tetapi menyebabkan nilai *packet delivery ratio* menjadi naik turun walaupun nilai yang diperoleh masing-masing kepadatan *node* yang digunakan tidak terlalu jauh. Protokol HWMP memiliki nilai yang lebih rendah dibanding ZRP, cenderung lebih stabil jika terjadi penambahan *node* yang digunakan, karena HWMP memelihara jalur alternatif yang menuju



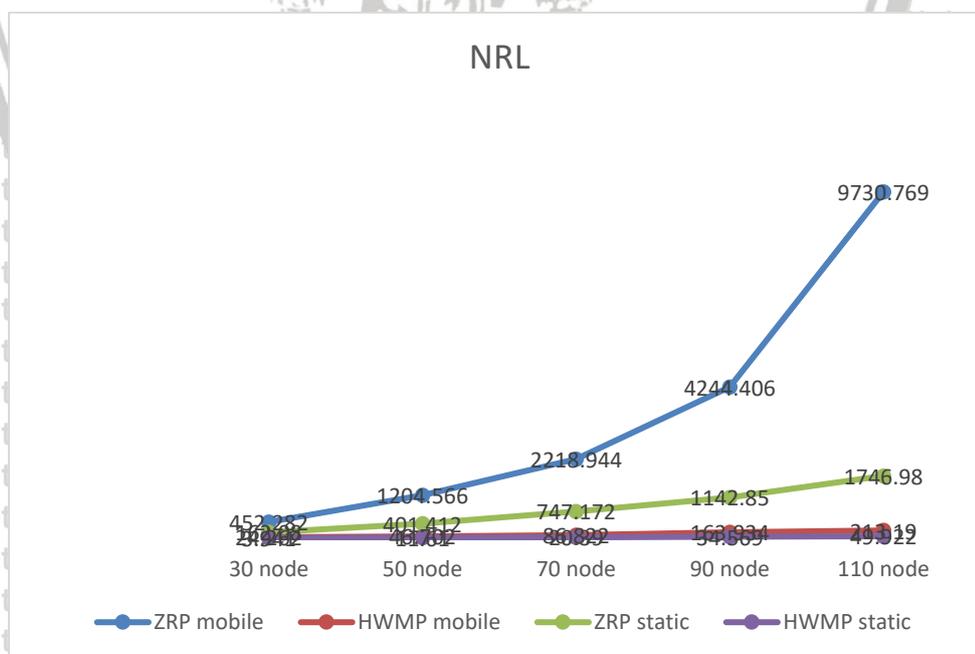
destination node, sehingga tidak terjadi *looping* yang berulang ketika proses penemuan rute. Sedangkan pada saat topologi dalam keadaan statis protokol HWMP lebih mendominasi daripada ZRP dikarenakan proses *routing* yang dilakukan oleh HWMP seperti yang dijelaskan di atas dapat berjalan maksimal dikarenakan *node* untuk jalur komunikasi tidak bergerak menjauh maupun mendekati.

5.1.3 Normalized Routing Load

Berdasarkan pengujian yang telah dilakukan dalam skenario luas area 1000x1000 meter dengan *node* yang berjumlah 30, 50, 70, 90, dan 110 diperoleh nilai rata-rata hasil *normalized routing load* pada masing-masing protokol *routing* yang dijabarkan pada Tabel 5.3 dan Gambar 5.13 berikut ini:

Tabel 5.3 Hasil pengujian *normalized routing load* ZRP dan HWMP

Jumlah <i>node</i>	30 <i>node</i>	50 <i>node</i>	70 <i>node</i>	90 <i>node</i>	110 <i>node</i>
ZRP mobile	452,282	1204,566	2218,944	4244,406	9730,769
HWMP mobile	24,262	46,702	86,822	163,934	213,19
ZRP static	162,08	401,412	747,172	1142,85	1746,98
HWMP static	3,941	11,61	20,39	34,569	49,922



Gambar 5.13 Grafik *normalized routing load* perbandingan kinerja ZRP dan HWMP

Pada Gambar 5.13 terlihat dalam skenario lingkungan pengujian dengan luas area 1000x1000 meter pada topologi dinamis diperoleh nilai rasio *normalized routing load* yang dihasilkan HWMP, yakni *node* 30 sebesar 24,262, *node* 50 sebesar 46,702, *node* 70 sebesar 86,822, *node* 90 sebesar 163,934 dan *node* 110 sebesar 213,19. Lalu lintas jaringan yang semakin padat membuat pencarian jalur rute semakin rumit sehingga membuat paket *routing* yang dikirim semakin meningkat. Hal tersebut terlihat jelas pada protokol *routing* ZRP yang memiliki nilai rata-rata *normalized routing load* yang lebih tinggi, yakni 452,282 pada *node* 30, 1204,566 pada *node* 50, 2218,944 pada *node* 70, 4244,406 pada *node* 90 dan 9730,769 pada *node* 110. Nilai rasio ZRP menunjukkan nilai yang sangat tinggi saat jumlah *node* semakin bertambah, sementara pada HWMP kenaikan nilai rasio *normalized routing load* berjalan lebih rendah dan relatif stabil. Sedangkan pada topologi statis didapatkan hasil *normalized routing load* pada protokol *routing* ZRP sebesar 162,08 pada *node* 30, 401,412 pada *node* 50, 747,172 pada *node* 70, 1142,85 pada *node* 90, dan 1746,98 pada *node* 110. Sedangkan pada protokol HWMP didapatkan nilai NRL sebesar 3,941 pada *node* 30, 11,61 pada *node* 50, 20,39 pada *node* 70, 34,569 pada *node* 90 dan 49,992 pada *node* 110. Nilai *normalized routing load* protokol ZRP pada keadaan statis lebih tinggi daripada HWMP, sama dengan pada kondisi topologi dinamis.

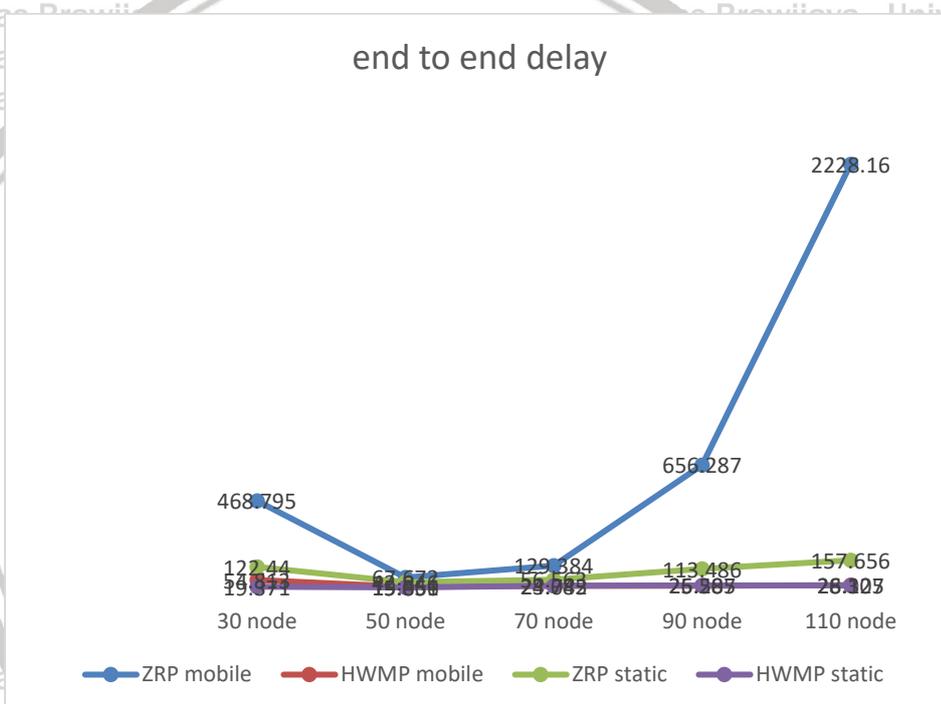
Berdasar hasil pengujian yang telah didapat, terlihat bahwa variasi kepadatan jumlah *node* sangat berpengaruh terhadap naiknya nilai rasio *normalized routing load* yang dihasilkan masing-masing protokol. Pada Gambar 5.13 terlihat kinerja terbaik terdapat pada protokol HWMP. Hal tersebut disebabkan karena HWMP bekerja secara reaktif dalam menangani rute yang terputus. Protokol HWMP memelihara seluruh jalur alternatif yang dapat terhubung menuju *destination node* sebagai jalur pengganti ketika salah satu jalur rusak atau terputus. Hal tersebut akan berjalan efisien ketika *node* yang tersebar berjumlah sangat banyak dan memungkinkan adanya beberapa jalur yang bercabang menuju *destination node*. Berbeda dengan ZRP yang membutuhkan paket *request routing* yang cukup banyak. Proses *routing* ZRP berjalan secara proaktif dalam memelihara informasi topologi zona radius lokal yang dijalankan di tiap zona masing-masing *node* yang tersebar. Ketika *destination node* berada jauh diluar zona radius *source node*, maka pencarian rute akan dialihkan kepada *peripheral node*. Namun ketika rute menuju *destination node* masih belum ditemukan, maka mekanisme yang sama akan diterapkan. *Peripheral node* milik *source node* akan meminta bantuan kepada *peripheral node* miliknya untuk mencari keberadaan *destination node* dengan mekanisme *routing* reaktif. Hal tersebut menyebabkan nilai *normalized routing load* ZRP memiliki nilai rata-rata yang sangat tinggi dibandingkan HWMP.

5.1.4 End-to-end Delay

Berdasarkan pengujian yang telah dilakukan dalam skenario luas area 1000x1000 meter dengan *node* yang berjumlah 30, 50, 70, 90, dan 110 diperoleh nilai rata-rata hasil *end-to-end delay* pada masing-masing protokol *routing* yang dijabarkan pada Tabel 5.4 dan Gambar 5.14 berikut ini:

Tabel 5.4 Hasil pengujian *end-to-end delay* ZRP dan HWMP

Jumlah node	30 node	50 node	70 node	90 node	110 node
ZRP mobile	468,795	67,672	129,384	656,287	2228,16
HWMP mobile	54,813	19,056	23,745	25,207	28,125
ZRP static	122,44	42,846	56,563	113,486	157,656
HWMP static	19,371	15,861	24,082	26,585	26,307



Gambar 5.14 Grafik *end-to-end delay* perbandingan kinerja ZRP dan HWMP

Pada Gambar 5.14 terlihat dalam skenario lingkungan pengujian topologi dinamis dengan luas area 1000x1000 meter diperoleh nilai rata-rata waktu *end-to-end delay* HWMP yakni sebesar 54,813, 19,056, 23,745, 25,207 dan 28,125 sec pada kepadatan *node* yang berjumlah 30, 50, 70, 90, dan 110. Bertambahnya jumlah *node* yang tersebar sangat berpengaruh pada naiknya nilai waktu *end-to-end delay* dalam pengiriman paket protokol ZRP. Hal tersebut terlihat jelas pada protokol *routing* ZRP yang memiliki rata-rata nilai *end-to-end delay* sebesar 468,795, 67,672, 129,384, 656,287 dan 2228,16 sec. Nilai waktu *end-to-end delay* ZRP menunjukkan nilai yang tinggi ketika jumlah *node* ditambah menjadi 90 dan 110. Sementara pada HWMP kenaikan nilai waktu *end-to-end delay* berjalan lebih stabil karena selisih nilai delay antar *node* yang digunakan tidak terlalu besar.

Sedangkan pada topologi statis didapatkan hasil *end to end delay* pada protokol ZRP sebesar 122,44 pada *node* 30, 42,846 pada *node* 50, 56,563 pada *node* 70, 113,486 pada *node* 90 dan 157,656 pada *node* 110. Sedangkan pada HWMP didapatkan nilai *end to end delay* sebesar 19,371 pada *node* 30, 15,861 pada *node* 50, 24,082 pada *node* 70, 26,585 pada *node* 90 dan 26,307 pada *node* 110. Pada topologi statis juga didapatkan nilai ZRP lebih tinggi daripada nilai HWMP seperti pada topologi dinamis.

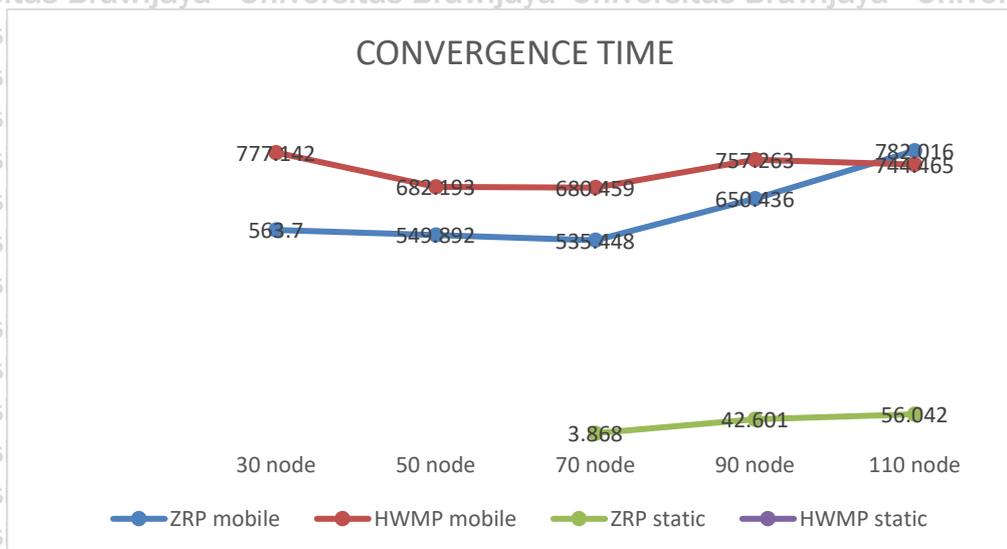
Berdasarkan grafik hasil pengujian pada Gambar 5.14 variasi kepadatan *node* cukup mempengaruhi naiknya nilai waktu *end-to-end delay* pada protokol ZRP dengan kenaikan nilai *end-to-end delay* yang cukup signifikan. Pada protokol HWMP nilai *end-to-end delay* masih cukup stabil, dikarenakan pemeliharaan jumlah jalur alternatif menuju *destination node* dapat mengurangi waktu *end-to-end delay* dalam proses pencarian rute yang sewaktu-waktu dapat berubah. Namun, pembaruan rute yang dilakukan secara reaktif menyebabkan proses rekonstruksi ulang pencarian rute harus dijalankan ketika terdapat permintaan baru dalam pengiriman paket. Pada protokol ZRP pemeliharaan topologi dalam zona radius akan berjalan rumit ketika *node* berjumlah 90 dan 110. Hal tersebut disebabkan oleh proses *routing* diluar zona radius lokal akan diserahkan pada *peripheral node*. Selanjutnya *peripheral node* akan mencari informasi *destination node* secara reaktif pada zonanya. Sehingga hal ini menyebabkan nilai *end-to-end delay* yang sangat tinggi pada topologi dinamis maupun statis.

5.1.5 Convergence Time

Berdasarkan pengujian yang telah dilakukan dalam skenario luas area 1000x1000 meter dengan *node* yang berjumlah 30, 50, 70, 90, dan 110 diperoleh nilai rata-rata hasil *convergence* pada masing-masing protokol *routing* yang dijabarkan pada Tabel 5.5 dan Gambar 5.15 berikut ini:

Tabel 5.5 Hasil pengujian *convergence time* ZRP dan HWMP

Jumlah <i>node</i>	30 <i>node</i>	50 <i>node</i>	70 <i>node</i>	90 <i>node</i>	110 <i>node</i>
ZRP mobile	563,7	549,892	535,448	650,436	782,016
HWMP mobile	777,142	682,193	680,459	757,263	744,465
ZRP static			3,868	42,601	56,042
HWMP static					



Gambar 5.15 Grafik *convergence* perbandingan kinerja ZRP dan HWMP

Pada Gambar 5.15 terlihat dalam skenario lingkungan pengujian topologi dinamis dengan luas area 1000x1000 meter diperoleh nilai rata-rata waktu *convergence time* HWMP berada di atas nilai ZRP yakni sebesar 777,142, 682,193, 680,459, 757,263 dan 744,465 sec pada kepadatan *node* yang berjumlah 30, 50, 70, 90, dan 110. Bertambahnya kepadatan jumlah *node* yang tersebar tidak terlalu menimbulkan loncatan nilai rata-rata yang tinggi terhadap hasil yang diperoleh. Sedangkan pada *routing* ZRP memiliki rata-rata nilai *convergence time* sebesar 563,7, 549,892, 535,448, 650,436 dan 782,016 sec pada kepadatan *node* yang berjumlah 30, 50, 70, 90, dan 110. Sedangkan pada topologi statis didapatkan hasil *convergence time* pada protokol ZRP hanya terjadi pada kepadatan *node* 70 sebesar 3,868, *node* 42,601 pada *node* 90 dan pada *node* 110 sebesar 56,042. Sedangkan pada HWMP tidak terjadi *convergence* dikarenakan semua paket menuju *destination node* dapat sampai ke tujuan tanpa terputus. Terbukti dengan nilai *packet delivery ratio* bernilai 100% di semua variasi kepadatan *node* yang digunakan.

Bertambahnya variasi kepadatan *node* yang tersebar tidak menimbulkan perubahan loncatan nilai rata-rata yang drastis dalam *routing* HWMP. Nilai waktu *convergence time* ZRP menunjukkan nilai rata-rata yang rendah di bawah nilai *convergence time* protokol HWMP ketika kondisi topologi dinamis. Pada topologi dinamis, nilai *convergence time* protokol ZRP dan HWMP cenderung naik turun walaupun selisih nilai yang didapat dari variasi kepadatan *node* tidak terlalu besar. Proses *routing* ZRP menggunakan mekanisme *bordercast* dalam penentuan jalur menuju *destination node*. Penggunaan mekanisme *bordercast* dalam cakupan luas area yang besar di tambah dengan kepadatan jumlah *node* yang banyak serta mobilitas *node* yang bergerak bebas membuat *recovery* atau pembaruan jalur jika terdapat *node* yang tidak tersedia atau rusak akan mempunyai nilai *convergence* yang rendah dikarenakan *source node* akan melakukan *bordercast* dari awal sampai menemukan *update* atau *recovery* jalur menuju *destination node*. Hal

inilah yang menyebabkan nilai *convergence* protokol ZRP cenderung di bawah dari nilai *convergence* HWMP. Sedangkan mekanisme *routing* yang digunakan oleh HWMP yaitu berdasarkan nomor urut untuk setiap jalurnya sehingga dipastikan konektifitas bebas *looping* setiap saat. Protokol HWMP memelihara seluruh jalur alternatif yang dapat terhubung menuju *destination node* sebagai jalur pengganti ketika salah satu jalur rusak atau terputus. Hal tersebut akan berjalan efisien ketika *node* yang tersebar berjumlah banyak dan memungkinkan adanya beberapa jalur yang bercabang menuju *destination node*. Jika terdapat *node* yang tidak tersedia atau rusak, maka dalam proses *recovery* jalurnya untuk *node* yang rusak tersebut akan melakukan pengiriman sinyal PERR kepada tetangga *node* jalur yang rusak tadi. Kemudian akan diteruskan pada *source node* dan akan memulai pencarian atau *update* jalur kembali, sehingga walaupun variasi kepadatan *node* di tambah nilai *convergence* yang di dapatkan tetap stabil. Sedangkan pada topologi statis, dikarenakan *node* yang saling berkomunikasi tidak bergerak menjauh ataupun mendekat, maka HWMP lebih mendominasi karena proses *routing* HWMP untuk penemuan jalur seperti yang dijelaskan di atas dapat berjalan maksimal.



BAB 6 KESIMPULAN

6.1 Kesimpulan

Berdasarkan hasil pengujian dan pembahasan yang telah dilakukan maka dapat disimpulkan sebagai berikut :

1. Pengaruh variasi kepadatan *node* yang dijalankan dalam skenario pengujian dengan area yang luas memberikan pengaruh yang signifikan pada kinerja protokol *routing* ZRP. Penurunan kualitas kinerja ZRP dapat dilihat berdasarkan nilai rata-rata *throughput*, *packet delivery ratio*, *normalized routing load*, dan *end-to-end delay* yang berubah ketika jumlah *node* dan luas area ditambahkan. Sedangkan pada HWMP penurunan kualitas kinerja juga terjadi ketika variasi kepadatan *node* di tambah, namun tidak terjadi selisih nilai yang drastis antara *node* satu dengan *node* lainnya, yang tidak seperti halnya ZRP. Walaupun terjadi penurunan efektifitas kinerja pada protokol ZRP, tetapi jika dibandingkan dengan protokol *routing* HWMP untuk efektifitas terbaik masih dimiliki oleh protokol *routing* ZRP untuk topologi dinamis, sedangkan untuk topologi statis efektifitas terbaik pada HWMP.
2. Berdasarkan skenario pengujian yang telah dilakukan terhadap kinerja protokol *routing* ZRP dan HWMP dengan parameter pengukuran *throughput*, *packet delivery ratio*, *normalized routing load*, *end-to-end delay*, dan *convergence time* didapatkan hasil analisis perbandingan kinerja sebagai berikut :
 - a) Berdasarkan nilai *throughput*, nilai rata-rata terbaik di semua skenario lingkungan pengujian dimiliki oleh protokol *routing* ZRP di semua topologi yang digunakan baik statis maupun dinamis, dibandingkan dengan nilai rata-rata protokol *routing* HWMP jika berjalan pada topologi yang sederhana. Jika variasi kepadatan *node* ditambah, maka nilai *throughput* ZRP menjadi menurun. Nilai *throughput* HWMP berada di bawah nilai *throughput* ZRP, tetapi jika variasi kepadatan *node* di tambah, tidak terjadi penurunan yang signifikan, tidak seperti halnya ZRP.
 - b) Berdasarkan nilai *packet delivery ratio* protokol ZRP memiliki nilai rata-rata di atas protokol HWMP jika berjalan pada topologi yang sederhana pada kondisi dinamis. Sama seperti halnya pada *throughput*, jika terjadi penambahan variasi *node* pada ZRP maka nilai PDR yang di dapatkan akan menurun. Sedangkan nilai rata-rata *packet delivery ratio* pada protokol HWMP yang dihasilkan berada di atas dari *packet delivery ratio* ZRP, jika menggunakan topologi statis. Tidak seperti ZRP, jika variasi kepadatan *node* ditambah, maka nilai *packet delivery ratio* yang dihasilkan cenderung naik turun walaupun selisih nilai yang di dapatkan antar *node* tidak berbeda drastis.

c) Berdasarkan nilai *normalized routing load*, protokol *routing* HWMP memiliki nilai rata-rata terbaik di seluruh skenario lingkungan pengujian jika dibandingkan dengan dengan nilai rata-rata ZRP, baik pada topologi statis maupun dinamis. Nilai rata-rata *normalized routing load* protokol HWMP yang dihasilkan cukup rendah dan tidak terjadi kenaikan nilai yang signifikan. Sedangkan nilai rata-rata *normalized routing load* pada protokol *routing* ZRP sangat tinggi dan mengalami kenaikan yang drastis ketika jumlah *node* bertambah.

d) Berdasarkan nilai *end-to-end delay*, protokol *routing* HWMP memiliki nilai rata-rata terbaik di semua skenario lingkungan pengujian baik topologi statis maupun dinamis. Nilai rata-rata *end-to-end delay* pada protokol HWMP cukup stabil dan tidak mengalami kenaikan yang signifikan. Sedangkan nilai rata-rata *end-to-end delay* pada protokol *routing* ZRP yang dihasilkan sangat tinggi dan cenderung mengalami kenaikan yang drastis ketika jumlah *node* bertambah.

e) Berdasarkan nilai waktu *convergence time*, protokol *routing* ZRP memiliki nilai rata-rata terbaik di semua skenario lingkungan pengujian pada topologi dinamis. Nilai rata-rata *convergence time* pada protokol ZRP berada di bawah nilai *convergence time* dari HWMP, dan tidak mengalami kenaikan yang signifikan. Sedangkan ketika dalam kondisi topologi statis, nilai rata-rata *convergence time* pada protokol *routing* HWMP berada di bawah nilai dari ZRP, dengan nilai yang didapatkan tidak berjarak terlalu jauh dengan variasi kepadatan *node* yang berbeda.

3. Penggunaan topologi statis dan dinamis pada MANET sangat mempengaruhi kondisi *routing* pada masing-masing protokol baik ZRP maupun HWMP. Kondisi topologi statis memiliki nilai lebih baik daripada saat kondisi topologi dinamis untuk masing-masing parameter pengujian yang dilakukan, dikarenakan untuk topologi statis *node* yang menjadi jalur perutean tidak bergerak sehingga proses *routing* menuju *destination node* dapat tercapai dengan mudah. Berbeda dengan topologi dinamis, *node-node* yang menjadi jalur perutean paket selalu bergerak menjauh maupun mendekati, sehingga akan mempengaruhi hasil terhadap parameter pengujian yang dilakukan.

6.2 Saran

Berdasarkan penelitian yang telah dilakukan didapatkan saran sebagai upaya dalam pengembangan sistem protokol *routing* pada MANET selanjutnya yaitu :

1. Dalam pengujian selanjutnya dapat menggunakan protokol jaringan TCP dimana pengiriman paket datanya terdapat mekanisme sinkronisasi sehingga pengiriman data lebih reliable.
2. Menambah parameter pengujian tentang bagaimana efisiensi penggunaan energi setiap protokol *routing*.

- 3. Menggunakan skenario variasi kecepatan mobilitas *node* sehingga menambah variasi dalam uji kinerja dalam rekonstruksi jalur *routing*.
- 4. Melakukan pengujian dengan skenario serangan keamanan sebagai analisis kerentanan sebuah protokol *routing* dalam hal keamanan jaringan.



DAFTAR PUSTAKA

- Bahr, M., 2006. *Proposed Routing for IEEE 802.11s WLAN Mesh Networks*. Siemens corporate technology, information & communication. Munchen. Germany.
- Bejar Nicklas., 2015. *Zone Routing Protocol (ZRP)*. Networking Laboratory, Helsinki University of Technology. P.O. Box 3000. FIN-02015 HUT. Finland
- Budiawan, A., 2008. *Analisis Ujuk Kerja*. Universitas Indonesia.
- Chhabra, M., 2013. *A Novel Solution to Handle DDOS Attack in MANET*. Journal of Information Security, 4(3).
- Denatama, M. I., 2016. *Analisis Perbandingan Kinerja Protokol Routing DSDV dan OLSR Untuk Perubahan Kecepatan Mobilitas pada Standar IEEE 802.11ah*. Infotel , 8(2), pp. 2085-3688
- Fatkhurrozi., Widasari, E.R., and Bhawiyuga, A., 2018. *Analisis Perbandingan Kinerja Protokol AOMDV, DSDV, Dan ZRP Sebagai Protokol Routing Pada Mobile Ad-Hoc Network (MANET)*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer .Vol. 2, No. 10. hlm. 3671-3680 .e-ISSN: 2548-964X.
- Frank, Ibikunle., Seun, Ebiesuwa.,2013. *Performance Analysis of Hybrid Protocols in MANETs*. International Journal of Computing Academic Research (IJCAR). ISSN 2305-9184. Volume 2, Number 5, pp. 224-233.
- Guesmia, M., Gezouri, M., Mbarek, N., 2012. *Performance evaluating of the HWMP proactive tree mode for IEEE 802.11s based Wireless Mesh Networks*. Signal Image Laboratory, Department of Electronics, Faculty of Electrical Engineering, University of Science and Technology, P.O. Box 1505, El-M'Naouer, Oran, ALGERIA.
- Gupta, Sandeep., Malhotra, Rahul., Dhaliwal,BS.,2015. *Performance Comparison Of Proactive Routing Protocols: OLSR, DSDV, WRP*. International Journal of Advanced Computer Research, 6(8).
- Issariyakul, T., Hossain, E., 2009. *Introduction to Network Simulator NS2*. Springer Science+Business Media, LCC. DOI: 10.1007/978-0-387-71760-9_1.
- Kaur, S., 2013. *An Overview of Mobile Ad hoc Network: Application, Challenges and Comparison of Routing Protocols*. IOSR Journal of Computer Engineering (IOSR-JCE), 11(5), pp. 07-11.
- Kiran Rampurkar., Nikhil Lavande., Santosh Shilgire., and Mane, Sayali N., 2017. *Study of Routing Overhead and Its Protocols*. International Journal of Advanced Engineering and Management. Vol. 2, No.2, pp.52-55.
- Kukreja, B., 2014. *Performance Comparison of Routing Protocols in MANET*. International Journal of Computer Science and Network Security, 14(8), pp. 108-114.

Kushwaha, Shalini., Chaudhary Deepak.,2015. *Performance Analysis of AODV, CBRP, AOMDV, DSR MANET Routing Protocol using NS2 Simulation*. International Journal of Computer Applications (0975-8887). Volume 116, No.5.

Manullang, F.B., Cahyani, N.D.W., and Suryani, V., 2012. *Analisis Performansi Routing Hybrid Wireless Mesh Protocol (HWMP) Pada Wireless Mesh Network (WMN) Berdasarkan Standar IEEE 802.11S*. Institut Teknologi Telkom. Vol 13, No.1. ISSN. 1412-0100.

Perkins, C. E., 2001. *Performance comparison of two on-demand routing protocols for ad hoc networks*. IEEE Personal communications, 8(1), pp. 16-28

Raja, M. L., 2014. *An Overview of MANET: Applications, Attacks and Challenges*. International Journal of Computer Science and Mobile Computing, 3(1), pp. 408-417.

Rao, A. R., 2014. *Performance Evaluation of DSR, AOMDV and ZRP Routing Protocols in MANETS by using NS2*. (IJCSIT) International Journal of Computer Science and Information Technologies, 5(1), pp. 711-714.

Sahat, Benny Pandopotan Sinaga.,2013. *Analisis Performansi Routing ZRP (Zone Routing Protocol) dan HWMP (Hybrid Wireless Mesh Protocol) Pada Jaringan Hybrid Wireless Mesh Network*. Bandung: Universitas Telkom.

Sheldon, T., 2001. *Convergence of Routing Tables*. [online] Tersedia di: <https://www.linktionary.com/c/converge_routing.html> [Diakses 3 Desember 2019].

S. Basagni., M. Conti., S.Giordano., and I. Stojmenovic., 2004. *Mobile Ad Hoc Networking*.

Victor, K. R., 2013. *Routing and Reducing Perturbation in Mobile ad Hoc Networks (Manets) for Efficient Communication*. International Journal of Advanced Computer Research, 4(4).

Wirawan, B., Andi., and Indarto, E., 2004. *Mudah Membangun Simulasi dengan Network Simulator-2 (NS2)*. Yogyakarta.

LAMPIRAN

A.1 Source Code AWK *Throughput*

```

1 BEGIN {
2     recvdSize = 0
3     start_time = 120
4     end_time = 0
5 }
6     event = $1
7     time = $2
8     pkt_size = $8
9     Level = $4
10    if (level == "AGT" && event == "s") {
11        if (time < start_time) {
12            start_time = time }
13        if (level == "AGT" && event == "r") {
14            if (time > end_time) {
15                end_time = time }
16        recvdSize += pkt_size }
17    }END{
18        printf ("Average Throughput[kbps] =
%.3f\t\t StartTime=%.3f\t StopTime=%.3f\t
totalpaket=%.3f\n", (recvdSize/(end_time
start_time))*(8/1000), start_time,
stop_time, recvdSize); }

```

A.2 Source Code AWK *End to-end delay*

```

1 BEGIN {
2     seqno=-1;
3     cnt=0;
4 }
5     if($4 == "AGT" && $1 == "s" && seqno < $6) {
6         seqno=$6; }
7     if($4 == "AGT" && $1 == "s") {
8         start_time[$6] = $2; }
9     else if (($4 == "AGT") && ($1 == "r")) {
10        end_time[$6] = $2; }
11    else if ($1 == "D" && $7 = "cbr") {
12        end_time[$6] = -1; }
13    }END{
14        for (i=0;i<=seqno;i++) {
15            if (end_time[i]>0) {
16                delay[i]=end_time[i]-start_time[i];
17                cnt++;
18            }else{
19                delay[i]=-1; }
20        for(i=0;i<=seqno;i++) {
21            if (delay[i]>0){
22                ssdelay=ssdelay+delay[i];}
23        ssdelay=ssdelay/ (cnt);

```

```
24 printf("average delay = %.3f", ssdelay); }
```

A.3 Source Code AWK Normalized routing load

```
1 BEGIN{
2   recv = 0
3   rtr_pkt = 0
4 }{
5   if (($1 == "r") && ($4=="AGT" )) recv++;
6   if (($1 == "s" || $1 == "f") && $4 == "RTR" && ($7
7   == "ZRP" || $7 == "message" || $7 == "HWMP")) rtr_pkt++;
8 }END{
9   printf("Total paket routing = %.3f Total paket
10  diterima = %.3f\n", rtr_pkt, recv);
11   printf("Normalized Routing Load = %.3f\n",
12  rtr_pkt/recv); }
```

A.4 Source Code AWK Packet delivery ratio

```
1 BEGIN {
2   send=0;
3   recv=0;
4 }{
5   if (($1 == "s") && ($4=="AGT" )) send++;
6   if (($1 == "r") && ($4=="AGT" )) recv++;
7 }END{
8   printf "total paket dikirim: %.3f total paket
9   diterima: %.3f, PDR: %.3f\n", send, recv, (recv/send); }
```

A.5 Source Code AWK Convergence time

```
1 BEGIN{dps=0; dpa=0;routing_packets=0.0;oldVal=0;}
2 #data packet yg dikirim -> data packet sent(dps):
3 ($1=="s" && $4=="AGT") {sender[$6] = $2; send++;}
4 #data packet yg diterima -> data packet arrived(dpa):
5 ($1=="r" && $4=="AGT") {recv[$6] = $2; receive++;}
6 #printf("nilai receive id = %d\n", $6);
7 #if( oldVal > $6 )
8 #printf("OldVal %d > NewVal %d\n",oldVal,$6);
9 #oldVal = $6;
10 END{
11 #Mencocokkan Sender ID dengan Receiver ID
12 #PROCINFO["sorted_in"] = "@ind_num_asc";
13 PROCINFO["sorted_in"]="@ind_num_asc"
14 no = 1;
15 prev = 0;
16 not_match = 0;
17 for(id in sender){
18   #printf ("Nomor %d id %d\n",no,id);
19   if(id in recv){
20     if(not_match == 1){
21       konvergensi += (recv[id] - prev);
```

```
22     #print konvergensi;
23     not_match = 0;
24     prev = recv[id];
25     }else{
26     prev = recv[id];
27     }
28     }else{
29     # print "packet id yg hilang: " id;
30     not_match = 1;
31     }
32     #no = no + 1;
33     }
34     print "Total Waktu Konvergensi : "konvergensi;
35 }
```



UNIVERSITAS BRAWIJAYA