



TESIS - IF 185401

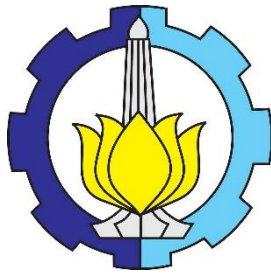
**PENGEMBANGAN *ZONE ROUTING PROTOCOL*
(ZRP) MENGGUNAKAN KONSEP ZONASI DINAMIS
DI *VANETS***

MUHSIN BAYU AJI FADHILLAH
05111850010024

Dosen Pembimbing
Dr. Eng. Radityo Anggoro, S.Kom., M.Sc.

Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
2020

(Halaman ini sengaja dikosongkan)



TESIS - IF 185401

**PENGEMBANGAN *ZONE ROUTING PROTOCOL*
(ZRP) MENGGUNAKAN KONSEP ZONASI DINAMIS
DI *VANETS***

MUHSIN BAYU AJI FADHILLAH

05111850010024

Dosen Pembimbing

Dr. Eng. Radityo Anggoro, S.Kom., M.Sc.

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

2020

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M. Kom)

di

Institut Teknologi Sepuluh Nopember

Oleh:

MUHSIN BAYU AJI FADHILLAH

NRP: 05111850010024

Tanggal Ujian: 22 Juli 2020

Periode Wisuda: September 2020

Disetujui oleh:

Pembimbing:

1. Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.

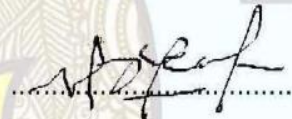
NIP: 1984101620081210002



Penguji:

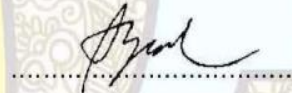
1. Waskitho Wibisono, S.Kom., M.Eng., Ph.D.

NIP: 197410222000031001



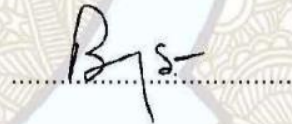
2. Prof. Ir. Supeno Djanali, M.Sc., Ph.D.

NPP: 194806191973011001

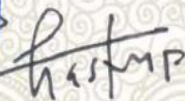


3. Bagus Jati Santoso, S. Kom., Ph. D.

NIP: 198611252018031001



Kepala Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas



Dr. Eng. Chastine Fatichah, S.Kom., M.Kom

NIP: 19751220 200112 2 002

(Halaman ini sengaja dikosongkan)

Pengembangan *Zone Routing Protocol* (ZRP) Menggunakan Konsep Zonasi Dinamis Di *VANETs*

Nama Mahasiswa : Muhsin Bayu Aji Fadhillah
NRP : 05111850010024
Pembimbing : Dr. Eng. Radityo Anggoro, S.Kom., M.Sc.

ABSTRAK

Vehicular Ad Hoc Network (VANETs) merupakan bentuk pengembangan teknologi yang memungkinkan pembentukan jaringan nirkabel pada perangkat bergerak seperti kendaraan bermotor. Teknologi VANETs dapat digunakan untuk meningkatkan keamanan dan efisiensi trafik lalu lintas.

Routing protocol pada VANETs dapat dikategorikan dalam 3 golongan, yaitu proaktif, reaktif, dan *hybrid*. Protokol proaktif akan mempertahankan *routing table* dengan melakukan pembaruan informasi dalam periodik waktu, sementara protokol reaktif akan mencari jalur hanya apabila ada permintaan untuk rute pengiriman paket. *Routing protocol* hybrid seperti *Zone Routing Protocol* (ZRP) mengkombinasikan keuntungan dari sifat protokol proaktif dan reaktif sehingga bisa menangani *delay* yang terlalu lama, *packet flooding*, dan penggunaan *bandwidth* berlebih untuk menjaga *routing table*.

ZRP bekerja secara proaktif untuk semua *node* yang berada di dalam zona, dan bekerja secara reaktif untuk meneruskan pencarian rute ke *node* yang tidak berada dalam zona. Besarnya zona dalam ZRP bergantung pada nilai radius, sehingga penting untuk mendapatkan nilai radius optimal agar besaran zona yang terbentuk bisa efektif. Zona yang terlalu kecil memungkinkan peningkatan *end-to-end delay*. Sedangkan zona dengan radius yang terlalu besar dapat mengakibatkan *routing overhead* yang tinggi.

Sebagai usaha untuk mencari zonasi optimal, ZRP dikembangkan agar dapat melakukan zonasi secara dinamis berdasarkan nilai radius. Pencarian nilai radius dilakukan dengan penggunaan nilai *threshold* stabilitas *link*. Nilai *threshold* dan radius akan diperbarui dalam periodik waktu untuk menyesuaikan dengan keadaan jaringan.

Kata Kunci: *VANETs*, ZRP, Zonasi Dinamis, Radius, Stabilitas Link.

(Halaman ini sengaja dikosongkan)

Improvement of Zone Routing Protocol (ZRP) Using the Concept of Dynamic Zoning in VANETs

By : Muhsin Bayu Aji Fadhillah
Student Identity Number : 05111850010024
Supervisor : Dr. Eng. Radityo Anggoro, S.Kom., M.Sc.

ABSTRACT

Vehicular Ad Hoc Networks (VANETs) are a form of technological development that enables the formation of wireless networks on mobile devices such as motorized vehicles. VANETs technology can be used to improve the safety and efficiency of traffic.

Routing protocols on VANETs can be categorized into 3 groups, namely proactive, reactive, and hybrid. The proactive protocol will maintain the routing table by updating information in a periodic time, while the reactive protocol will search for paths only if there is a request for a route of sending packets. Hybrid routing protocols such as Zone Routing Protocol (ZRP) combine the benefits of proactive and reactive protocol properties so that they can handle too long delays, packet flooding, and use of excess bandwidth to maintain the routing table.

ZRP works proactively for all nodes in the zone, and works reactively to continue route search to nodes that are not in the zone. The size of zone in the ZRP depends on the value of the radius, so it is important to get the optimal value of the radius so that the size of zone formed can be effective. Zones that are too small allow increased end-to-end delay. While zones with a radius that is too large can result in high routing overhead.

In an effort to find optimal zoning, ZRP was developed to be able to dynamically zonate based on radius values. The search for the radius value is done by using the link stability threshold value. Threshold and radius values will be updated in a periodic time to adjust to the state of the network.

Keywords: *VANETs, ZRP, Dynamic Zoning, Radius, Link Stability*

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur kepada Allah Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul “Pengembangan *Zone Routing Protocol (ZRP)* Menggunakan Konsep Zonasi Dinamis Di VANETs”.

Pengerjaan tesis ini menjadi sebuah sarana untuk penulis memperdalam ilmu yang telah didapatkan di Institut Teknologi Sepuluh Nopember Surabaya, khususnya dalam disiplin ilmu Informatika. Harapan penulis semoga apa yang tertulis di dalam buku tesis ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini. terselesaikannya buku tesis ini tidak terlepas dari bantuan dan dukungan semua pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT atas semua rahmat yang diberikan sehingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Bapak Widodo dan Ibu Sri Karyaningsih selaku kedua orang tua penulis atas segala dukungan, motivasi serta doa sehingga penulis dapat mengerjakan Tugas Akhir ini.
3. Shinta Khoiri Fadhillah selaku adik penulis atas segala dukungan dan doa yang telah diberikan sehingga penulis tetap semangat dalam mengerjakan Tugas Akhir ini.
4. Bapak Dr. Eng. Radityo Anggoro, S.Kom., M.Sc., selaku dosen pembimbing, atas bimbingan, arahan dan bantuannya dalam pengerjaan Tugas Akhir ini.
5. Rizky Fenaldo Maulana, Kharisma Monika, Buthoro Kunto Raharjo dan seluruh teman-teman S2 Informatika ITS angkatan 2018 dan S1 Informatika ITS Angkatan 2014 yang selama ini sudah membantu penulis menyelesaikan Tugas Akhir.
6. Agil, Juna, Dheva, Satria dan seluruh rekan muda-mudi Semampir yang selama ini sudah menyemangati dan mendoakan penulis selama menjalankan kuliah di ITS.

7. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa masih terdapat kekurangan, kesalahan, maupun kelalaian yang telah penulis lakukan dalam penelitian ini. Oleh karena itu, saran dan kritik yang membangun sangat dibutuhkan untuk penyempurnaan tesis ini.

Surabaya, 10 Agustus 2020

Muhsin Bayu Aji Fadhillah

DAFTAR ISI

HALAMAN PENGESAHAN.....	i
ABSTRAK	i
ABSTRACT.....	i
KATA PENGANTAR	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR	vii
DAFTAR TABEL.....	ix
DAFTAR LAMPIRAN.....	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Kontribusi Masalah.....	4
1.6 Batasan Masalah	4
BAB 2 KAJIAN PUSTAKA.....	5
2.1 Dasar Teori.....	5
2.1.1 <i>Vehicular Ad hoc Networks (VANETs)</i>	5
2.1.2 <i>Zone Routing Protocol (ZRP)</i>	7
2.2 Studi Literatur	10
2.2.1 Radius Optimal	10
2.2.2 Zonasi Dinamis	11
2.2.3 Faktor Pembentuk Nilai Radius	12
2.2.4 <i>Predicting Node Trend (PNT)</i>	13
BAB 3 METODE PENELITIAN.....	15
3.1 Tahapan Penelitian.....	15
3.2 Hasil Kajian Tentang ZRP dan Zonasi Dinamis	15
3.3 Perancangan Algoritma.....	16

3.3.1	Perancangan Nilai <i>Threshold</i> Stabilitas <i>Link</i>	16
3.3.2	Perancangan Zonasi Dinamis dengan <i>Threshold</i> Stabilitas <i>Link</i>	19
3.4	Perancangan Skenario Mobilitas	20
3.4.1	Skenario Mobilitas <i>Grid</i>	20
3.4.2	Skenario Mobilitas <i>Real</i>	22
3.5	Implementasi	22
3.6	Pengujian.....	23
3.7	Analisa Hasil	24
3.7.1	<i>Packet Delivery Ratio</i>	24
3.7.2	<i>Routing Overhead</i>	24
3.7.3	<i>Average end-to-end delay</i>	24
BAB 4 HASIL DAN PEMBAHASAN		27
4.1	Implementasi Sistem	27
4.2	Analisa Hasil	27
4.2.1	Hasil uji coba pembobotan faktor stabilitas <i>link</i>	28
4.2.2	Hasil uji coba skenario <i>grid</i>	33
4.2.3	Hasil uji coba skenario <i>real</i>	42
BAB 5 KESIMPULAN DAN SARAN		53
5.1	Kesimpulan	53
5.2	Saran.....	53
LAMPIRAN		55
DAFTAR PUSTAKA.....		59

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Jaringan VANETs	6
Gambar 2.2 Bordercasting oleh BRP	8
Gambar 2.3 Topologi ZRP dengan Radius 2 Hop	9
Gambar 2.4 Node Peripheral Mencari Node Destinasi di Dalam Zonanya	9
Gambar 2.5 Rute yang Terbentuk	10
Gambar 3.1 Diagram Alur Tahapan Penelitian.....	15
Gambar 3.2 Diagram Alur Penghitungan Radius	20
Gambar 3.3 Alur perancangan skenario mobilitas <i>grid</i>	21
Gambar 4.1 Rancangan peta <i>grid</i>	33
Gambar 4.2 Grafik PDR skenario <i>grid</i> dengan variasi radius	34
Gambar 4.3 Grafik PDR skenario <i>grid</i> dengan variasi waktu pembaruan.....	35
Gambar 4.4 Grafik rata-rata <i>end to end delay</i> skenario <i>grid</i> dengan variasi radius	37
Gambar 4.5 Hasil Rata-rata End to End Delay (ms) dengan Variasi Waktu Pembaruan.....	39
Gambar 4.6 Grafik rata-rata RO skenario <i>grid</i>	40
Gambar 4.7 RO ZRP Modifikasi dengan Variasi Waktu Pembaruan Radius.....	41
Gambar 4.8 Rancangan peta <i>real</i>	43
Gambar 4.9 Grafik PDR skenario <i>real</i> dengan variasi batasan radius.....	44
Gambar 4.10 Grafik PDR Skenario <i>Real</i> dengan Variasi Waktu Pembaruan Radius	45
Gambar 4.11 Grafik rata-rata <i>end to end delay</i> skenario <i>real</i> (ms)	47
Gambar 4.12 End to End Delay (ms) pada Skenario Real dengan Variasi Waktu Pembaruan Radius.....	48
Gambar 4.13 Grafik Rata-rata RO Skenario <i>Real</i>	50
Gambar 4.14 RO ZRP Modifikasi dengan Variasi Waktu Pembaruan Radius.....	51

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 3.1 Hello Message.....	18
Tabel 3.2 Parameter Lingkungan Simulasi	23
Tabel 4.1 Lingkungan Simulasi	27
Tabel 4.2 Pembobotan Faktor Stabilitas Link.....	28
Tabel 4.3 Hasil PDR Uji Pencarian Komposisi pada Skenario Real	29
Tabel 4.4 Hasil PDR Uji Pencarian Komposisi pada Skenario Grid	29
Tabel 4.5 Hasil End-to-end Delay Uji Pencarian Komposisi pada Skenario Grid	29
Tabel 4.6 Hasil End-to-end Delay Uji Pencarian Komposisi pada Skenario Real	30
Tabel 4.7 Hasil Routing Overhead Uji Pencarian Komposisi pada Skenario Grid	30
Tabel 4.8 Hasil Routing Overhead Uji Pencarian Komposisi pada Skenario Real	30
Tabel 4.9 Perbandingan Hasil PDR Pencarian Komposisi Faktor pada Skenario Grid	31
Tabel 4.10 Perbandingan Hasil PDR Pencarian Komposisi Faktor pada Skenario Real	31
Tabel 4.11 Perbandingan Hasil End-to-end Delay Pencarian Komposisi Faktor pada Skenario Grid	31
Tabel 4.12 Perbandingan Hasil End-to-end Delay Pencarian Komposisi Faktor pada Skenario Real	32
Tabel 4.13 Perbandingan Hasil RO Pencarian Komposisi Faktor pada Skenario Grid	32
Tabel 4.14 Perbandingan Hasil RO Pencarian Komposisi Faktor pada Skenario Real	32
Tabel 4.15 Hasil perhitungan rata-rata PDR (%) skenario grid	34
Tabel 4.16 Hasil PDR ZRP Modifikasi dengan Variasi Waktu Pembaruan Radius	36

Tabel 4.17 Hasil perhitungan rata-rata <i>end to end delay</i> (milisekon) skenario <i>grid</i> dengan variasi radius	37
Tabel 4.18 Rata-rata End to End Delay (ms) dengan Variasi Waktu Pembaruan Radius Skenario <i>Grid</i>	39
Tabel 4.19 Hasil Perhitungan Rata-rata RO pada Skenario <i>Grid</i>	41
Tabel 4.20 RO ZRP Modifikasi dengan Variasi Waktu Pembaruan Radius.....	42
Tabel 4.21 Hasil perhitungan rata-rata PDR (%) skenario <i>real</i>	44
Tabel 4.22 PDR ZRP Modifikasi dengan Variasi Waktu Pembaruan Radius pada Skenario <i>Real</i>	45
Tabel 4.23 Perbandingan PDR ZRP Modifikasi	46
Tabel 4.24 Hasil perhitungan rata-rata <i>end to end delay</i> (ms) skenario <i>real</i>	47
Tabel 4.25 End to End Delay (ms) dengan Variasi Waktu Pembaruan Radius Skenario <i>Real</i>	49
Tabel 4.26 Hasil perhitungan rata-rata RO (paket) skenario <i>real</i>	50
Tabel 4.27 Hasil RO ZRP Modifikasi dengan Variasi Waktu Pembaruan Radius	51

DAFTAR LAMPIRAN

Lampiran 1. Modifikasi <i>file</i> zrp.h	55
Lampiran 2. Modifikasi ZRPAgent : public Agent pada <i>file</i> zrp.h.....	56
Lampiran 3. Modifikasi fungsi NeighborList::addNeighbor pada <i>file</i> zrp.cc	56
Lampiran 4. Modifikasi fungsi NDPBeaconTransmitTimer::handle pada <i>file</i> zrp.cc	56
Lampiran 5. Modifikasi fungsi NDPAgent::recv_NDP_BEACON pada <i>file</i> zrp.cc	56
Lampiran 6. Modifikasi fungsi NDPAgent::recv_NDP_BEACON_ACK pada <i>file</i> zrp.cc	57
Lampiran 7. Modifikasi fungsi IERPAgent::recv_IERP_ROUTE_REQUEST_UNI <i>file</i> zrp.cc	58

(Halaman ini sengaja dikosongkan)

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi sudah sampai pada tahap dimana informasi dapat dikirimkan melalui perangkat-perangkat bergerak yang saling terkoneksi tanpa memerlukan bantuan infrastruktur fisik yang tetap. Hal ini diadopsi pada beberapa bidang seperti militer, medis, hingga keamanan karena pengiriman informasi bisa terjadi lebih cepat dan menyesuaikan dengan kondisi lingkungan yang ada. Salah satu teknologi yang menunjang hal itu adalah MANETs.

Mobile Ad Hoc Network (MANET) adalah jaringan nirkabel pada perangkat bergerak yang memungkinkan setiap perangkat dapat berkomunikasi satu sama lain tanpa menggunakan infrastruktur jaringan tetap (Nasipuri, 2004). Ketiadaan infrastruktur tetap memungkinkan setiap *node* yang ada pada jaringan MANETs bergerak bebas dan independen ke segala arah. Jaringan MANETs dapat terbentuk dari kumpulan otonom perangkat seluler yang berkomunikasi satu sama lain melalui tautan nirkabel dan bekerja sama secara terdistribusi untuk sebuah tujuan (Raja dan Baboo, 2014).

Vehicular Ad Hoc Network (VANETs) merupakan subkategori hasil pengembangan dari MANETs. Jaringan VANETs dapat berjalan di atas kendaraan. Karena berjalan di atas kendaraan, mobilitas pada VANETs pun lebih tinggi dan dinamis. Teknologi VANETs dapat digunakan untuk meningkatkan keamanan dan efisiensi trafik lalu lintas. Arsitektur jaringan VANETs dapat dibuat dengan memanfaatkan kendaraan maupun teknologi seluler yang sudah ada.

Algoritma *routing protocol* digunakan untuk mengatur jalannya pertukaran informasi tiap *node* pada VANETs. Ada beberapa *routing protocol* yang ada pada VANETs, dimana mereka dapat diklasifikasikan dalam 3 klasifikasi, yaitu protokol proaktif, protokol reaktif, dan protokol *hybrid* (Jayanti, 2014). Protokol proaktif melakukan pembaruan periodik pada *routing table* sehingga *end-to-end*

delay rendah. Protokol reaktif hanya akan membuat rute ketika ada permintaan untuk pengiriman paket sehingga mempunyai *routing overhead* yang rendah. Sementara protokol *hybrid* menggunakan kombinasi metode yang ada pada protokol proaktif dan reaktif.

Protokol proaktif dan reaktif sama-sama memiliki kelebihan dan kekurangan. Protokol proaktif bisa saja menggunakan *bandwidth* berlebih untuk menjaga *routing table*, sedangkan protokol reaktif mempunyai masalah *delay* akibat paket *route request* yang terlalu panjang dan juga kemungkinan terjadi *flooding* dari paket *route request* yang membanjiri jaringan dalam pencarian rute (Oigawa dan Sato, 2018). ZRP sebagai protokol *hybrid* memanfaatkan kelebihan protokol proaktif yang memperbarui *routing table* secara periodik sehingga mengurangi *delay* serta kelebihan protokol reaktif yang hanya membuat rute ketika pengiriman paket sehingga mengurangi *routing overhead*. ZRP menggunakan *IntraZone Routing Protocol* (IARP) sebagai protokol proaktif dan *InterZone Routing Protocol* (IERP) sebagai protokol reaktif.

IARP pada ZRP bekerja secara proaktif dengan memperbarui informasi *routing table* setiap *node* yang berada di dalam sebuah zona. Besarnya zona yang dibentuk bergantung pada nilai radius yang dihitung dalam satuan *hop*, sehingga penting untuk mendapatkan nilai radius optimal agar besarnya zona yang terbentuk bisa efektif. Zona yang terlalu kecil memungkinkan cakupan node di dalam zona terlalu sedikit sehingga saat pengiriman paket ZRP harus mencari rute terlebih dahulu karena informasi tentang node tidak terdapat di dalam *routing table*. ZRP akan mencari rute secara reaktif dengan IERP, paket *route request* akan terus dikirim hingga rute ditemukan, dan memungkinkan *end-to-end delay* mengalami peningkatan. Sementara itu, nilai radius yang membentuk zona yang terlalu besar justru bisa mengakibatkan *routing overhead* yang tinggi karena IARP harus mengumpulkan informasi *routing table* dari *node* dalam jumlah yang terlalu banyak.

Pada penelitian ini, metode dinamisasi zona diusulkan guna membentuk zona dengan radius optimal. Nilai *threshold* stabilitas *link* digunakan sebagai acuan

dalam mencari nilai radius. Perumusan nilai *threshold* dihitung dari faktor stabilitas *link* seperti tingkat kepadatan *node* tetangga, kecepatan, percepatan, jarak, dan kualitas *link* dari *node*. Keadaptifan zonasi dari ZRP dibentuk dengan nilai *threshold* dan nilai radius yang akan diperbarui setiap periodik waktu untuk menyesuaikan dengan kondisi lalu lintas jaringan.

1.2 Perumusan Masalah

Rumusan Masalah yang diangkat dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merumuskan nilai stabilitas *link* yang diambil dari faktor kecepatan, percepatan, jarak, kualitas *link* dan tingkat kepadatan *node* tetangga?
2. Bagaimana menggunakan nilai stabilitas *link* sebagai acuan untuk melakukan dinamisasi zona pada ZRP?

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah pengembangan protokol ZRP yang mampu melakukan zonasi dinamis dengan berpedoman pada nilai stabilitas *link*.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah meningkatkan performa ZRP dengan mengoptimalkan kelebihan sifat proaktif dan reaktif pada ZRP menggunakan metode zonasi dinamis

1.5 Kontribusi Masalah

Kontribusi dari penelitian ini adalah pengembangan kemampuan ZRP untuk beradaptasi terhadap kondisi lingkungan jaringan berdasarkan stabilitas *link* yang ada dengan melakukan dinamisasi zona.

1.6 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Jaringan yang digunakan adalah VANETs.
2. Implementasi dan uji coba menggunakan simulator *Simulation of Urban Mobility* (SUMO) dan NS-2.35.
3. Radius minimal adalah 1 *hop* dan radius maksimal 7 *hop*.
4. Posisi *node* pengirim dan *node* penerima bersifat statis.
5. Simulasi dilakukan menggunakan peta berbentuk *grid* dengan ukuran 1500 meter x 1500 meter dan peta real, yaitu peta wilayah Semolowaru, Surabaya dengan ukuran 1500 m x 1500m

BAB 2

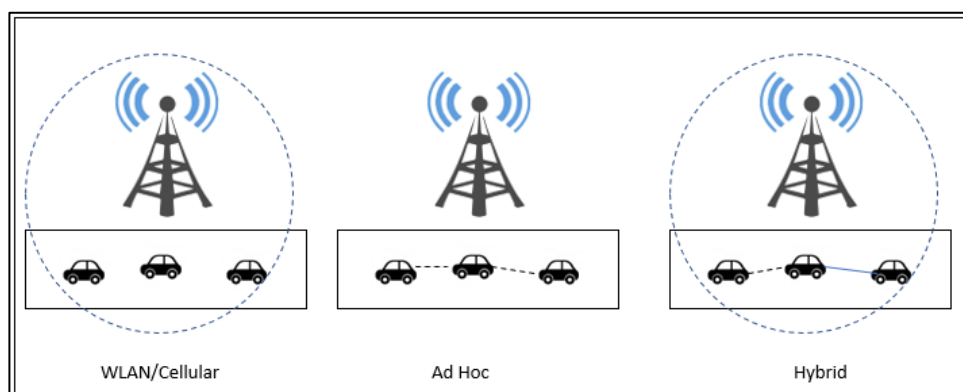
KAJIAN PUSTAKA

2.1 Dasar Teori

2.1.1 Vehicular Ad hoc Networks (VANETs)

Mobile Ad Hoc Network (MANETs) merupakan jaringan nirkabel tanpa infrastruktur tetap yang membuat setiap perangkatnya dapat dikonfigurasi secara mandiri untuk dapat saling berkomunikasi dengan bertautan secara nirkabel. Pada MANETs, setiap *node* bisa bergerak bebas ke segala arah secara independen. Kedinamisan yang dihadirkan oleh MANETs membuatnya menjadi pilihan untuk melakukan komunikasi dan pertukaran informasi di lingkungan tanpa infrastruktur tetap seperti daerah perang/konflik, zona militer, daerah rawan bencana, hingga daerah yang sulit terjangkau infrastruktur komunikasi (Hoebeke *et al.*, 2010).

Vehicular Ad Hoc Network (VANETs) merupakan pengembangan dari MANETs dan masih mengikuti prinsip yang sama namun diterapkan pada lingkungan transportasi yang sangat dinamis. Arsitektur jaringan VANETs dapat dibangun dengan memanfaatkan kendaraan, teknologi seluler, atau menggabungkan keduanya sekaligus seperti diperlihatkan dalam Gambar 2.1. Teknologi VANETs mengintegrasikan penggunaan WLAN dan teknologi seluler yang dipadukan dengan komunikasi antar kendaraan cerdas sehingga bisa digunakan untuk meningkatkan keamanan dan efisiensi trafik lalu lintas (Li dan Wang, 2007). VANETs dapat menyediakan *ubiquitous connectivity* pada pengguna kendaraan di jalan sehingga memungkinkan *Intelligent Transportation System* (ITS) dengan komunikasi antar kendaraan yang efisien. VANETs juga memiliki beberapa konstrain berbeda karena berjalan di lingkungan transportasi yang membuatnya berbeda dari MANETs seperti kebiasaan sopir, keterbatasan mobilitas dan kecepatan mobilitas yang tinggi (Yousefi, Mousavi dan Fathy, 2007).



Gambar 2.1 Arsitektur Jaringan VANETs

Routing protocol pada jaringan *ad hoc* dapat diklasifikasikan ke dalam tiga kategori, yaitu routing protokol proaktif, reaktif, dan *hybrid* (Jayanti, 2014). Protokol proaktif bekerja dengan membuat *node* melakukan pembaruan periodik yang kontinyu untuk menjaga informasi *routing table*. Akibat terus melakukan pembaruan informasi pada *routing table*, protokol proaktif cenderung memiliki *end-to-end delay* yang rendah, karena setiap *node* mempunyai informasi rute pada *routing table*, namun menghasilkan *routing overhead* tinggi karena banyaknya *control packet* yang dikirimkan. Contoh protokol proaktif adalah *Destination Sequenced Distance-Vector Routing (DSDV)* dan *Optimized Link State Routing (OLSR)*. Kebalikan dari protokol proaktif, sebuah protokol reaktif hanya akan membuat rute ketika ada permintaan untuk pengiriman paket. Tanpa permintaan paket, *node* tidak akan membuat rute dan hanya mempertahankan informasi nya dengan mengirimkan *hello messages/beacon messages* secara berkala. Pada protokol reaktif, *routing overhead* yang dihasilkan cenderung rendah karena tidak banyak paket yang dikirimkan saat *node* tidak sedang mencari rute, hanya saja protokol reaktif memiliki kecenderungan *end-to-end delay* yang tinggi karena harus menunggu terbentuknya rute sebelum proses pengiriman paket dilakukan. Skema kerja protokol reaktif juga memungkinkan terjadinya *flooding* paket *route request* hasil *broadcast* saat proses pembentukan rute dimulai. Contoh protokol reaktif adalah *Dynamic Source Routing (DSR)* dan *Ad Hoc On Demand Distance Vector Routing (AODV)*. Protokol *hybrid* adalah kombinasi metode yang ada pada protokol proaktif dan reaktif. Protokol *hybrid* mencoba memanfaatkan kelebihan pada protokol proaktif yang dapat menghasilkan nilai *end-to-end delay* yang rendah

sekaligus protokol reaktif yang mempunyai kecenderungan menghasilkan *routing overhead* yang rendah karena hanya membuat rute ketika ada permintaan pengiriman paket. Contoh dari protokol *hybrid* adalah *Zone Routing Protocol* (ZRP).

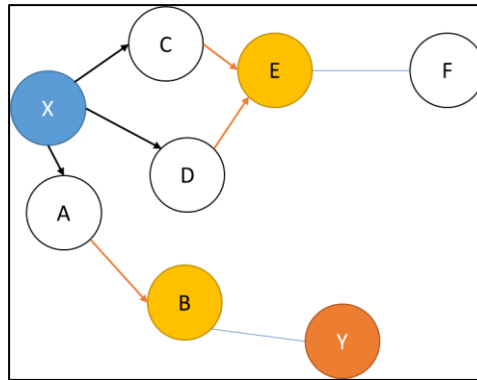
2.1.2 *Zone Routing Protocol* (ZRP)

Zone Routing Protocol (ZRP) merupakan protokol *hybrid* yang menggabungkan keunggulan protokol proaktif dan reaktif dengan mempertahankan peta topologi zona yang berpusat pada setiap *node* (Bejar, 2002). ZRP memiliki keunggulan untuk mengurangi kemungkinan *routing overhead* yang tinggi dengan bekerja secara reaktif dan juga *delay* yang tinggi saat proses pembentukan rute dengan bekerja secara proaktif untuk memperbarui informasi *routing table*. Fitur utama protokol ZRP adalah perilaku dinamisnya, dimana operasi skemanya diatur oleh sebuah parameter tunggal, radius zona, yang menyesuaikan perilaku skema dari protokol yang murni reaktif menjadi protokol proaktif (Haas dan Pearlman, 1998).

ZRP memang dirancang untuk lingkungan MANETs/VANETs, dimana setiap *node* akan membentuk zonanya masing-masing sesuai dengan nilai radius yang ditetapkan. Zona yang dibentuk setiap *node* memiliki kemungkinan untuk saling tumpang tindih karena besarnya ditentukan oleh radius yang dihitung dalam satuan *hop*. Transfer data di dalam zona akan diatur oleh *IntraZone Routing Protocol* (IARP) yang menjadi bagian proaktif dan perutean di luar zona akan diatur oleh *InterZone Routing Protocol* (IERP) yang menjadi bagian reaktif (Jain dan Chaba, 2014). ZRP juga mempunyai *Bordercast Routing Protocol* (BRP) yang berguna sebagai mekanisme *query-control*.

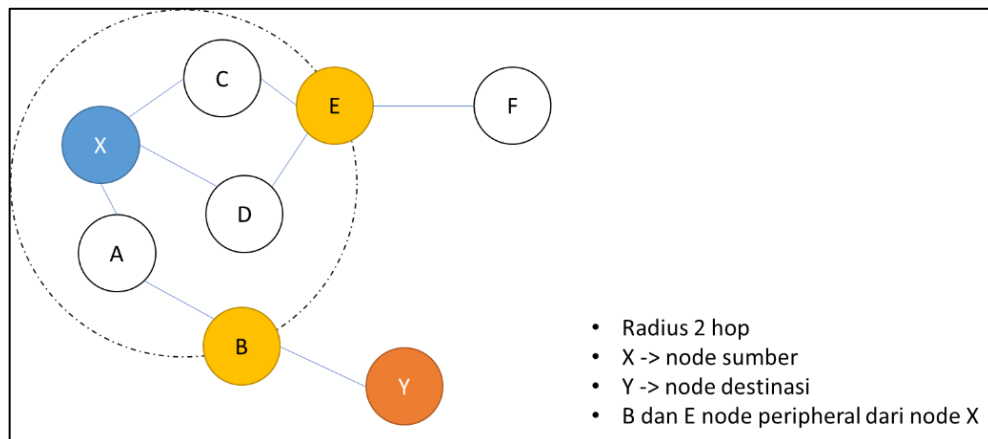
Secara garis besar, ZRP bekerja secara proaktif dengan menggunakan IARP, sedangkan IERP dibantu dengan BRP akan digunakan saat ZRP bekerja secara reaktif. IARP digunakan untuk mengatur topologi dari dalam zona yang dibatasi oleh radius. *Node* di dalam zona yang memiliki jarak *hop* kurang dari radius dapat disebut *interior node*, sedangkan *node* dalam zona yang memiliki jarak *hop*

sama dengan radius disebut *peripheral node*. Setiap *node* di dalam zona akan saling bertukar informasi *routing*.



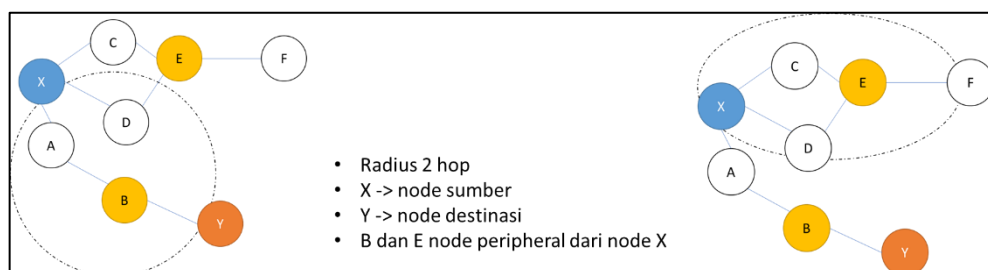
Gambar 2.2 Bordercasting oleh BRP

Sebagai protokol proaktif, IARP pada ZRP akan mempertahankan dan memperbarui informasi *routing table* dari setiap *node* yang berada di dalam zona yang terbentuk berdasarkan nilai radius. IERP digunakan untuk mengirim data ketika *node* tujuan berada di luar zona, yang akan melakukan pencarian rute secara reaktif alih-alih dengan *flooding* namun justru memakai *bordercasting* dengan mekanisme *query-control* yang ditangani oleh BRP. Hal inilah yang ditangani BRP yang berada dalam IERP dengan menggunakan *multicast* untuk meneruskan paket secara spesifik ke *peripheral node* dengan menggunakan informasi *node* dari IARP. Proses kerja BRP seperti dicontohkan dalam Gambar 2.2, saat pengiriman *route request* dari *node X* akan mengirim ke *node D*, *A*, dan *C*. Dari *node C*, paket *route request* akan diteruskan ke *node* tetangganya lagi, tetapi karena proses *bordercast* yang dilakukan maka paket *route request* hanya dikirim ke *node E*, sementara *node A* hanya melakukan *bordercast* ke *node B* dan *node D bordercast* ke *node E*. Sampailah paket *route request* ke *peripheral node* dari *node X*, yaitu *node B* dan *node E*. Penggunaan BRP di IERP dapat menghindari permintaan yang berlebihan dalam proses perutean zona (Huang, Cheng dan Tang, 2012).



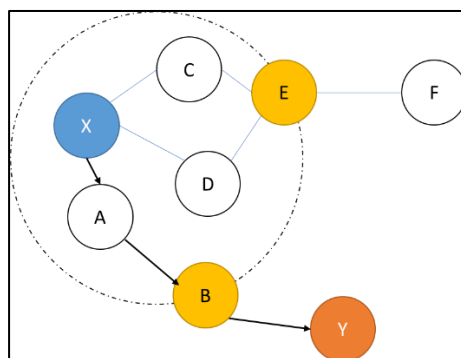
Gambar 2.3 Topologi ZRP dengan Radius 2 Hop

Topologi ZRP dan cara kerjanya seperti ditunjukkan pada Gambar 2.3. Seperti dicontohkan, *node X* sebagai *node sumber* ingin mengirim paket menuju *node Y* sebagai *node destinasi*. Pertama kali, *node X* akan bekerja secara proaktif dan mengecek zonanya menggunakan IARP untuk mencari informasi mengenai *node destinasi* yang dituju, dimana di dalam zona *node X* terdapat *node A*, *C*, dan *D* sebagai *interior node* dan *node E* dan *B* sebagai *peripheral node*. Karena *node Y* tidak berada di dalam zona *node X* maka paket akan diteruskan melalui *peripheral node* dengan IERP dan BRP. Dalam contoh ini, BRP akan melakukan *bordercast* menuju setiap *peripheral node*, yaitu *node E* dan *B*. Setelah paket *route request* sampai pada *node E* dan *B* selaku *peripheral node* maka mereka akan melakukan pengecekan secara proaktif dengan IARP seperti yang dilakukan oleh *node X* seperti dalam Gambar 2.4.



Gambar 2.4 Node Peripheral Mencari Node Destinasi di Dalam Zonanya

Node E ternyata tidak memiliki informasi mengenai *node* destinasi di dalam zonanya. Sementara itu, *Node B* yang juga mendapat terusan paket *route request* ternyata memiliki jalur menuju *node Y* yang berada di dalam zonanya, maka *node B* akan mengirimkan paket balasan konfirmasi untuk terbentuknya jalur. Sehingga rute yang terbentuk melalui *node X*, menuju *node A*, menuju *node B*, menuju *node Y* seperti dalam Gambar 2.5.



Gambar 2.5 Rute yang Terbentuk

2.2 Studi Literatur

2.2.1 Radius Optimal

Pada protokol ZRP, nilai radius menjadi faktor penting sebab besarnya zona yang terbentuk ditentukan lewat nilai radius. Nilai radius dihitung dalam satuan hop. Zona yang terbentuk dengan nilai radius yang terlalu besar dapat mengakibatkan jumlah *node* yang terlalu banyak untuk diperbarui informasi *routing table*-nya. Pada kondisi zona yang terlalu besar dan jumlah *node* di dalam zona yang terlalu banyak, *routing overhead* akan meningkat karena jumlah *control packet* yang diperlukan saat pembaruan *routing table* dengan jumlah *node* yang banyak. Sebaliknya, apabila nilai radius terlalu kecil maka zona memiliki kemungkinan tidak dapat mengakomodasi cukup banyak *node*. Akibatnya, peningkatan *delay* dapat terjadi karena ZRP harus mencari rute terlebih dahulu karena informasi tidak terdapat pada *routing table*, lalu terdapat kemungkinan

terjadinya *packet flooding* karena *broadcast* dari paket *route request* guna mencari rute dari *node* sumber menuju *node* destinasi.

Beberapa penelitian dilakukan untuk mencari radius optimal yang bisa diterapkan pada ZRP. (Pearlman dan Haas, 1999) menunjukkan ZRP dapat dikonfigurasi dengan penyesuaian pada parameter radiusnya untuk jaringan tertentu. Penelitian yang dilakukan mengajukan teknik *min searching* dan *traffic adaptive* untuk membuat setiap *node* mampu mengidentifikasi dan bereaksi dengan tepat sesuai kondisi trafik jaringan.

(Surateno, 2007) juga mencoba mengimplementasikan teknik *min searching* untuk melakukan optimasi pada penentuan radius zona yang digunakan dalam algoritma HOPNET yang berbasis pada ZRP. (Yélémou *et al.*, 2012) melakukan penelitian dengan serangkaian pengujian untuk menemukan nilai radius optimal dengan mengukur performa ZRP dalam metrik analisis berupa *routing overhead*, *packet delivery ratio*, dan *delay*. Hasilnya, radius mencapai nilai ideal sebesar 3 hop untuk mendapatkan performa ZRP terbaik bila *packet delivery ratio* dan *routing overhead* menjadi acuan, sementara untuk *delay* didapatkan nilai radius 1 dan 4 sebagai nilai radius optimal.

2.2.2 Zonasi Dinamis

ZRP tradisional memiliki pengaturan radius yang menghasilkan nilai radius yang sama untuk semua *node*. Nilai radius berpengaruh terhadap besarnya zona, untuk itu kedinamisan pembentukan zona dengan nilai radius yang dinamis menjadi dasar pengembangan ZRP yang mampu membuat zonasi secara adaptif.

(Chenn-Jung Huang *et al.*, 2006) menggunakan *evolving fuzzy neural networks* untuk membuat radius zona rute pada setiap *routing table* dapat menyesuaikan diri secara dinamis. Teknik seleksi juga dilakukan pada penyiaran paket secara *unicast* atau *broadcast* untuk *node-node* yang berada diluar zona.

(Samar, Pearlman dan Haas, 2004) mengembangkan *Independent Zone Routing* (IZR) dengan konfigurasi adaptif dan terdistribusi untuk ukuran optimal zona

perutean setiap *node*. Konfigurasi yang dilakukan berbasis pada kondisi setiap *node* secara independen.

2.2.3 Faktor Pembentuk Nilai Radius

Pada dasarnya, zonasi pada ZRP mengacu pada nilai radius *default* yang ditentukan. Nilai radius ini berlaku untuk semua *node* di dalam jaringan pada ZRP tradisional. Beberapa parameter dapat dijadikan dasar pembentukan/penentuan nilai radius dalam pengembangan ZRP. Faktor-faktor seperti tingkat kepadatan *node* tetangga (*density*), kecepatan serta arah pergerakan *node* dapat digunakan sebagai bahan untuk mencari/menentukan nilai radius yang nantinya akan berpengaruh pada zonasi yang dilakukan.

Selain nilai radius yang memengaruhi besarnya zona, *density* dari node adalah faktor penting lain yang memengaruhi performa ZRP (Lakhtaria, 2014). Semakin banyak zona yang bertindihan pada jaringan dengan tingkat kepadatan *node* yang tinggi maka nilai *end-to-end delay* semakin berkurang. Disisi lain, jumlah *node* yang terlalu banyak dapat mengakibatkan penurunan performa akibat *control packet* yang terlalu banyak (Sinha dan Sen, 2012).

(Zheng *et al.*, 2017) membuat protokol ADRP yang dikembangkan dari AODV dengan memanfaatkan tingkat kepadatan *node* tetangga yang dihitung dengan *rebroadcast probability* untuk menentukan pengiriman *broadcast* paket RREQ. Nilai *rebroadcast probability* dipengaruhi oleh tingkat kepadatan node tetangga dan akan dibandingkan dengan nilai *threshold* random. ADRP sendiri menggunakan Hello Messages untuk perantara pengiriman informasi jumlah *node* tetangga. Tiap *node* sumber dan *intermediate* mengirim informasi jumlah *neighbor nodes* nya terhadap tetangga 1 *hop* nya yang dikirim paket RREQ.

(Natsheh dan Buragga, 2010) mengembangkan protokol AODV dengan modifikasi pada *sending* dan *forwarding* RREQ yang menggunakan penghitungan nilai dari tingkat kepadatan *node* tetangga untuk menyesuaikan dengan lingkungan yang memiliki tingkat kepadatan *node* yang berbeda-beda. (Wang, Chao dan

Member, 2017) menggunakan pertimbangan dari nilai *density* untuk menentukan arah pengiriman *broadcast* paket.

Selain tingkat kepadatan *node* tetangga, kecepatan dan arah pergerakan *node* juga kerap dijadikan acuan untuk menentukan nilai radius. (Zhou, Cheng dan Lu, 2007) serta (Jaiswal dan Singh, 2010) menggunakan parameter kecepatan dan mobilitas dari *node* untuk menentukan nilai radius setiap *node*. (Yang *et al.*, 2018) membuat klasterisasi zona pada ZRP. Setiap klaster kemudian akan menunjuk satu *cluster head* (CH) dan membagi area dengan radius 2 hop. Pemilihan *cluster head* didasarkan pada mobilitas *node* sehingga sebuah *node* dengan kecepatan dan mobilitas rendah memiliki kecenderungan untuk menjadi *cluster head*.

2.2.4 Predicting Node Trend (PNT)

Karakteristik perilaku dari node-node pada jaringan VANETs antara lain adalah pergerakannya yang dinamis dan daya yang terbatas. *Predicting Node Trend* (PNT) adalah metode yang digunakan oleh (Shen *et al.*, 2014) untuk mencari rute terbaik dengan mempertimbangkan aspek perilaku atau karakteristik dari setiap *node* pada *routing protocol* AODV. Metode PNT melakukan penghitungan *Total Weight of Route* (TWR) sebagai representasi karakteristik perilaku *node* yang diukur dari faktor kecepatan, percepatan, arah pergerakan *node*, serta kualitas *link* diantara *node*. Penghitungan ini menghasilkan nilai TWR yang kemudian akan dibandingkan dengan perkiraan nilai TWR di masa depan. Selisih nilai TWR sekarang dan di masa depan inilah yang dipakai sebagai acuan untuk menentukan *node* yang digunakan dalam pembentukan rute.

Supaya bisa menggunakan nilai TWR, setiap *node* akan melakukan *broadcast* paket Hello Message yang disisipi informasi metrik perutean. Dari nilai-nilai tersebut, setiap *node* dapat melakukan penghitungan nilai TWR sekarang dan masa depan sebelum menentukan pembentukan rute yang sesuai.

(Arifiyan, Djanali dan Anggoro, 2017) menambahkan faktor jumlah *node* tetangga dalam penghitungan nilai TWR. Penelitian yang dilakukan menunjukkan *density* atau tingkat kepadatan *node* dapat ditambahkan sebagai metrik perutean

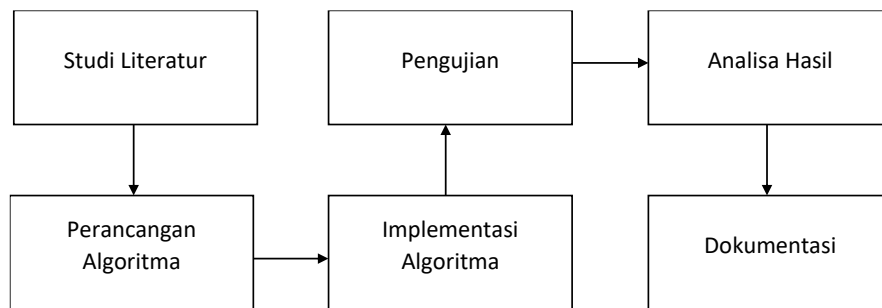
untuk mencari rute optimal. Hasil penelitian menunjukkan penambahan faktor jumlah *node* tetangga dapat meningkatkan performa *routing protocol*.

BAB 3

METODE PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian dibuat untuk membantu mencapai tujuan yang diharapkan dari penelitian. Langkah pertama dari penelitian ini adalah studi literatur. Studi literatur digunakan untuk mempelajari masalah yang ditemukan serta mencari referensi dari penelitian yang sudah dilakukan mengenai masalah tersebut. Tahapan penelitian yang dilakukan seperti ditunjukkan oleh Gambar 3.1.



Gambar 3.1 Diagram Alur Tahapan Penelitian

Langkah selanjutnya adalah perancangan dan penerapan algoritma. Algoritma yang dirancang pada penelitian ini adalah dinamisasi zona pada ZRP. Tahap pengujian dilakukan dengan simulasi di lingkungan VANETs menggunakan perangkat lunak NS-2 versi 2.35 untuk menguji performa ZRP. Hasil dari tahap pengujian kemudian akan dianalisis untuk mendapatkan kesimpulan dari algoritma yang telah dirancang dan akan ditulis dalam dokumentasi berupa buku tesis

3.2 Hasil Kajian Tentang ZRP dan Zonasi Dinamis

Penelitian diawali dengan melakukan kajian yang berkaitan dengan topik penelitian yang diajukan. Referensi yang digunakan dalam penelitian ini berasal dari jurnal, konferensi, dan buku yang berkaitan dengan zonasi dinamis pada protokol ZRP di lingkungan VANETs. Berdasarkan studi literatur tentang

zonasi dinamis pada protokol ZRP yang telah dilakukan, didapatkan informasi sebagai berikut:

1. ZRP memiliki nilai radius optimal. Nilai radius yang terlalu tinggi akan membuat zona yang terlalu besar dan peningkatan *routing overhead*, sementara radius yang terlalu kecil membuat zona yang terlalu kecil dan meningkatkan *delay*.
2. Zonasi pada ZRP dapat dilakukan secara dinamis dengan mengatur parameter radius untuk menyesuaikan dengan kondisi lingkungan jaringan.
3. Penentuan nilai radius dengan mempertimbangkan beberapa faktor seperti tingkat kepadatan *node* tetangga (*density*), kecepatan, percepatan, kualitas *link* dan jarak antar *node*.

3.3 Perancangan Algoritma

Alur perancangan dan penerapan algoritma zonasi adaptif beracuan *threshold* dinamis dibagi menjadi beberapa tahap, yaitu: (1) Perancangan Nilai *Threshold* Stabilitas *Link* (2) Perancangan Zonasi Dinamis dengan *Threshold* Stabilitas *Link*. Secara lebih rinci tahapan-tahapan tersebut dijelaskan pada sub-sub bab dibawah ini:

3.3.1 Perancangan Nilai *Threshold* Stabilitas *Link*

Setiap *node* akan menghitung nilai dari faktor-faktor stabilitas *link* seperti kecepatan, percepatan, arah pergerakan, kualitas *link*, serta tingkat kepadatan *node* tetangga seperti yang ditunjukkan dalam persamaan (3.1). Rumus TWR yang digunakan mengacu dari penelitian (Arifiyan, Djanali dan Anggoro, 2017) yang digunakan di AODV untuk mencari rute dengan nilai TWR paling kecil yang dianggap paling stabil.

$$TWR = f_s \times |S_n - S_d| + f_a \times |A_n - A_d| + f_d \times |\theta_n - \theta_d| + f_q \times Q + f_n \times N \quad (3.1)$$

dengan:

f_s = faktor koefisien kecepatan

f_a = faktor koefisien percepatan

f_d = faktor koefisien arah pergerakan

f_q = faktor koefisien kualitas *link*

f_n = faktor koefisien jumlah *node* tetangga

S_n, A_n, θ_n = kecepatan, percepatan, dan arah pergerakan *node* tetangga

S_d, A_d, θ_d = kecepatan, percepatan, dan arah pergerakan *node*

Q = kualitas *link* antara *node* dengan *node* tetangganya

N = jumlah *node* tetangga di dalam zona

Kualitas link bisa dihitung dari indeks stabilitas s_{ij} dari *link* (i, j) berdasarkan informasi dari pergerakan kendaraan dengan persamaan (3.2).

$$s_{ij} = 1 - \frac{\min(\sqrt{(i_x - j_x)^2 + (i_y - j_y)^2}; r)}{r} \quad (3.2)$$

dengan:

r = jarak transmisi maksimum

i_x, j_x = koordinat dari *node* x

i_y, j_y = koordinat dari *node* y

Nilai dari kualitas link kemudian dapat dihitung seperti pada persamaan (3.3).

$$Q = \frac{1}{s_{ij}} \quad (3.3)$$

dengan:

S_{ij} = indeks stabilitas dari dua *node*

Nilai *threshold* didapatkan dari rata-rata nilai penghitungan faktor stabilitas *link* setiap *node* tetangga dari sebuah *node* seperti pada persamaan (3.4). Nilai inilah yang akan menjadi acuan untuk menentukan nilai radius dari *node*.

$$Threshold = \frac{\Sigma TWRn}{n} \quad (3.4)$$

dengan:

ΣTWR = total nilai faktor metrik perutean setiap *node* tetangga

n = jumlah *node* tetangga

Sebelum melakukan penghitungan nilai stabilitas *link*, setiap *node* terlebih dahulu mengirimkan informasi yang berisi faktor-faktor stabilitas *link* yang disisipkan dalam paket *Hello Message*. Hal ini dilakukan agar setiap *node* bisa melakukan penghitungan nilai *threshold*. Supaya bisa disisipi faktor-faktor stabilitas *link* untuk penghitungan nilai *threshold*, maka isi dari paket *Hello Message* pun dimodifikasi seperti pada Tabel 3.1.

Tabel 3.1 Hello Message

Tipe	ID Sumber	Kecepatan	Percepatan	Jumlah <i>Node</i> Tetangga	Destinasi
------	--------------	-----------	------------	-----------------------------------	-----------

Nilai *threshold* akan digunakan sebagai pedoman untuk menentukan nilai radius dalam proses zonasi. Nilai *threshold* akan diperbarui setiap waktu pengiriman paket *hello message* yang berisikan informasi terbaru dari setiap *node* yang ada di dalam jaringan.

Simulasi dilakukan dengan *node* sumber dan *node* tujuan berada dalam posisi diam (*stationery mode*) sehingga harus ada penyesuaian terhadap formula TWR. Karena *node* tujuan bersifat *stationery mode*, maka nilai faktor arah menjadi tidak relevan. Karena ketika dalam posisi diam, maka *node* tersebut tidak memiliki arah. Maka dari itu perlu adanya perubahan faktor arah agar tidak mengalami pengurangan bobot pada TWR. Faktor arah dapat diubah menjadi nilai jarak antara *node* dengan *node* tetangganya. Koordinat posisi dari setiap *node* dapat diketahui karena meskipun *node* tersebut dalam kondisi diam. Berikut adalah perubahan formula perhitungan TWR:

$$TWR = f_s \times |S_n - S_d| + f_a \times |A_n - A_d| + f_a \times |\theta_n - \theta_d| + f_q \times Q + f_n \times N \quad (3.5)$$

dengan:

θ_n = jarak *node* tetangga

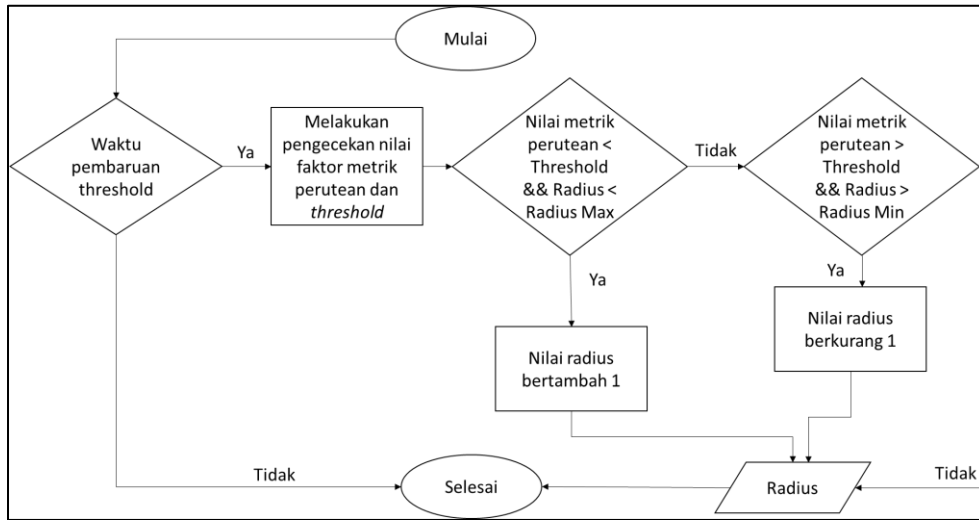
θ_d = jarak *node*

Selain formula TWR, formula indeks stabilitas juga perlu diubah karena jika hasil fungsi minimum antara jarak dan radius transmisi maksimum menghasilkan nilai dari radius transmisi maksimum, akan terjadi pembagian dengan nol pada saat perhitungan kualitas hubungan. Berikut adalah perubahan formula indeks stabilitas:

$$s_{ij} = 1 - \frac{\min(\sqrt{(i_x - j_x)^2 + (i_y - j_y)^2}; r)}{r+1} \quad (3.6)$$

3.3.2 Perancangan Zonasi Dinamis dengan *Threshold* Stabilitas *Link*

Zonasi dinamis diimplementasikan dengan mengubah nilai radius yang dapat menyesuaikan dengan kondisi jaringan. Bentuk penyesuaian nilai radius dilakukan dengan mengubah nilai radius apabila kondisi terpenuhi seperti yang dijelaskan dalam Gambar 3.2.



Gambar 3.2 Diagram Alur Penghitungan Radius

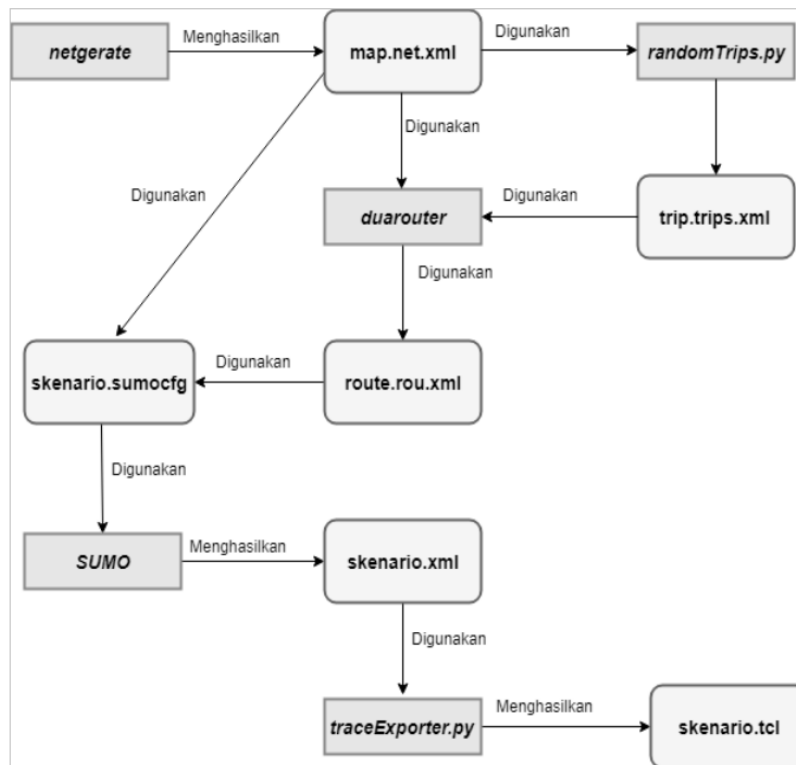
Jika nilai hasil penghitungan stabilitas *link* yang dihitung dengan persamaan (3.1) dari sebuah *node* ternyata kurang dari nilai *threshold* dan nilai radius dari *node* lebih kecil dari nilai radius maksimal yang ditentukan, maka nilai radius akan bertambah satu. Sebaliknya, jika nilai hasil penghitungan faktor metrik perutean dari sebuah *node* melebihi nilai *threshold* dan nilai radius lebih besar dari nilai radius minimal yang ditentukan maka nilai radius berkurang satu. Pengecekan dan pembaruan nilai radius akan terus dilakukan setiap periodik waktu.

3.4 Perancangan Skenario Mobilitas

Pada penelitian ini skenario mobilitas akan menggunakan 2 bentuk peta, yaitu peta berbentuk *grid* dan peta *real*. Skenario mobilitas dengan varian peta *grid* dan *real* dianggap merepresentasikan lingkungan VANETs. Penjelasan kedua skenario mobilitas adalah sebagai berikut:

3.4.1 Skenario Mobilitas *Grid*

Perancangan skenario mobilitas *grid* diawali dengan menentukan luas area simulasi yang dibutuhkan. Dari luas area tersebut ditentukan panjang dan lebar area. Selanjutnya dari panjang dan lebar area, ditentukan banyaknya persimpangan atau perpotongan jalan yang dibutuhkan, sehingga dapat diketahui jumlah petak. Setelah jumlah petak diketahui, dapat menentukan panjang dan lebar setiap petak.



Gambar 3.3 Alur perancangan skenario mobilitas *grid*

Peta *grid* yang telah ditentukan luas areanya tersebut kemudian akan mengikuti alur perancangan skenario mobilitas *grid* yang digambarkan pada Gambar 3.3. Langkah pertama adalah membuat peta dengan area yang ditentukan sebelumnya menggunakan *tools Simulation of Urban Mobility (SUMO)* yaitu *netgenerate*. Selain titik persimpangan dan panjang tiap petak *grid*, dibutuhkan juga pengaturan jumlah *node* dan kecepatan maksimal *node* dalam pembuatan peta *grid*. Peta *grid* yang dihasilkan oleh *netgenerate* akan memiliki ekstensi *.net.xml*. Peta *grid* ini kemudian digunakan untuk membuat pergerakan *node* dengan *tools SUMO* yaitu *RandomTrips* dan *duarouter*.

Skenario mobilitas *grid* dihasilkan dengan menggabungkan file peta *grid* dan file pergerakan *node* yang telah dibuat sebelumnya, akan menghasilkan file dengan ekstensi *.xml*. Selanjutnya, untuk dapat menerapkannya pada NS-2.35 file skenario mobilitas *grid* yang memiliki ekstensi *.xml* dikonversi ke dalam bentuk *file .tcl*. Konversi ini dilakukan menggunakan *tools traceExporter*. Hasilnya berupa file

yang berisi mobilitas dari setiap *node* (*mobility.tcl*) dan informasi *lifetime* dari setiap *node* (*activity.tcl*).

3.4.2 Skenario Mobilitas *Real*

Perancangan skenario mobilitas *real* diawali dengan memilih area yang akan dijadikan simulasi. Pada penelitian ini, digunakan peta dari *OpenStreetMap* untuk mengambil area yang dijadikan model simulasi. Area yang diambil adalah daerah Ngindensemolo Kota Surabaya. Setelah memilih area, unduh dengan menggunakan fitur *export* dari *OpenStreetMap*. Peta hasil *export* dari *OpenStreetMap* ini memiliki ekstensi *.osm*.

Setelah mendapatkan peta area yang dijadikan simulasi, peta tersebut dikonversi ke dalam bentuk *file* dengan ekstensi *.net.xml* menggunakan tools *SUMO* yaitu *netconvert*. Tahap berikutnya memiliki tahapan yang sama seperti tahapan ketika merancang skenario mobilitas grid yang ditunjukkan pada Gambar 3.3, yaitu membuat pergerakan *node* menggunakan *RandomTrips* dan *duarouter*. Kemudian gabungkan *file* peta *real* yang sudah dikonversi ke dalam *file* dengan ekstensi *.net.xml* dan file pergerakan *node* yang sudah di buat sebelumnya. Hasil dari penggabungan tersebut merupakan *file* skenario yang memiliki ekstensi *.xml*. *File* yang dihasilkan tersebut dikonversi ke dalam bentuk *file* *.tcl* agar dapat diterapkan pada NS-2.35. Alur pembuatan skenario *real* hampir sama dengan alur pembuatan skenario *grid*. Namun peta *map.net.xml* dihasilkan dari peta OSM yang telah dikonversi. Proses selanjutnya sama dengan pembuatan skenario *grid*.

3.5 Implementasi

Simulasi *VANETs* pada NS-2.35 dilakukan dengan menggabungkan file skenario yang telah dibuat menggunakan *Simulation of Urban Mobility (SUMO)* dan skrip *file Tool Command Language (TCL)* yang berisikan konfigurasi lingkungan simulasi. Sebelumnya, harus dilakukan instalasi protokol ZRP terlebih dahulu pada perangkat lunak NS-2 versi 2.35. Parameter simulasi seperti terlihat pada Tabel 3.2.

Tabel 3.2 Parameter Lingkungan Simulasi

No.	Parameter	Spesifikasi
1	<i>Network Simulator</i>	NS-2 Versi 2.35
2	<i>Routing Protocol</i>	ZRP tradisional, ZRP modifikasi
3	Radius	Minimal 1 maksimal 7 hop, Minimal 1 maksimal 3 hop, Minimal 3 maksimal 5 hop, Minimal 5 maksimal 7 hop
4	Waktu Simulasi	200 detik
5	Waktu Pengiriman Paket Data	2,5 – 200 detik
6	Area Simulasi	1500 m x 1500 m
7	Jumlah Kendaraan	25, 50, 100, 200
8	Kecepatan Maksimal	55 km/h
9	Radius Transmisi	250 m
10	Waktu Pembaruan Radius	5, 10, 15, 20, 25 detik
11	<i>Source/Destination</i>	Posisi statis
12	Radius ZRP Tradisional	2

3.6 Pengujian

Pengujian yang dilakukan membutuhkan sebuah *file Tool Command Language* (TCL) yang berisi deskripsi dari lingkungan simulasi VANETs pada perangkat lunak NS-2.35 yang akan diujikan. *File* TCL berisi konfigurasi dari setiap node dan beberapa *event* yang perlu diatur. Pada setiap skenario lingkungan simulasi, akan di generate skenario mobilitas sebanyak 10 kali. Pengujian dilakukan dengan menggunakan protokol ZRP tradisional dan ZRP hasil pengembangan yang diusulkan

3.7 Analisa Hasil

Analisa hasil dilakukan dengan mengolah hasil *trace file* simulasi menggunakan *script awk*. Dimana metrik yang digunakan adalah sebagai berikut:

3.7.1 Packet Delivery Ratio

Packet delivery ratio (PDR) adalah metrik yang digunakan untuk mengukur rasio jumlah paket yang dikirimkan berbanding dengan paket yang berhasil diterima (Astya, Nand dan Sharma, 2011). PDR dihitung dengan persamaan (3.3) dan dinyatakan dalam bentuk persentase.

$$PDR = \frac{Received}{Sent} \times 100 \% \quad (3.3)$$

dengan:

Sent = jumlah paket yang telah dikirimkan

Received = jumlah paket yang berhasil diterima

3.7.2 Routing Overhead

Routing overhead (RO) adalah jumlah paket *routing* yang diperlukan untuk komunikasi jaringan. *Routing overhead* dapat dihitung menggunakan *awk script* yang memproses *trace file* dan menghasilkan nilai *routing overhead*. *Routing overhead* dihitung dengan persamaan (3.4).

$$RO = sentRTR + forwardedRTR \quad (3.4)$$

dengan:

sentRTR = Jumlah paket *routing* yang dikirim

forwardedRTR = Jumlah paket *routing* yang diteruskan

3.7.3 Average end-to-end delay

Rata-rata *end-to-end delay (E2E)* merupakan rata-rata dari *delay* atau waktu yang dibutuhkan tiap paket untuk sampai ke *node* tujuan dalam satuan detik. *Delay* tiap paket didapatkan dari rentang waktu antara *node* asal mengirimkan paket dan *node* tujuan menerima paket. Dari *delay* tiap paket tersebut semua dijumlahkan dan dibagi dengan jumlah paket yang berhasil diterima, maka akan didapatkan rata-rata

end-to-end delay, Penundaan karena rute waktu penemuan, antrian, propagasi, dan transfer termasuk dalam metrik *End-to-End Delay* (Astya, Nand dan Sharma, 2011). Penghitungan *End-to-End Delay* seperti pada persamaan (3.5).

$$E2E = \frac{\sum_{m=1}^{recvnum} CBRRecvTime - CBRSentTime}{recvnum} \quad (3.5)$$

dengan:

CBRRecvTime = Waktu *node* asal mengirimkan paket

CBRSentTime = Waktu *node* tujuan menerima paket

recvnum = Jumlah paket yang berhasil diterima

Besar *end to end delay* dapat menunjukkan tingkat konektivitas dan efisiensi pada sebuah rute yang digunakan. Semakin besar *end to end delay* maka rute yang dipilih merupakan rute yang tidak efisien dan memiliki kemungkinan memiliki konektivitas yang buruk atau terlalu padat.

(Halaman ini sengaja dikosongkan)

BAB 4

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Spesifikasi perangkat keras yang digunakan untuk penelitian ini dapat dilihat pada Tabel 4.1 Lingkungan Simulasi.

Tabel 4.1 Lingkungan Simulasi

Komponen	Spesifikasi
Sistem Operasi	Ubuntu 18.04.4 LTS (Bionic Beaver)
CPU	Intel Core i3-6006U @ 2,0 GHz x 2
Memori	8 GB DDR4
Penyimpanan	120 GB

Sedangkan perangkat lunak yang digunakan dalam penelitian ini adalah:

1. *Simulation of Urban Mobility* (SUMO) versi 1.4.0 untuk pembuatan skenario mobilitas baik peta grid maupun peta real pada VANETs
2. *Java OpenStreetMap Editor* (JOSM) versi 15553 untuk penyuntingan peta kota yang didapatkan dari OpenStreetMap.
3. Network Simulator (NS) versi 2.35 dengan gcc versi 5.4 untuk melakukan simulasi routing protokol pada peta yang telah dibuat oleh SUMO.
4. Visual Studio Code versi 1.40 sebagai editor kode dalam proses perancangan algoritma.

4.2 Analisa Hasil

Analisa hasil penelitian ini dibagi menjadi tiga, yaitu hasil pencarian komposisi faktor *stabilitas link* serta simulasi menggunakan skenario mobilitas *grid* dan simulasi menggunakan skenario mobilitas *real* untuk mengetahui hasil penelitian. Setiap hasil simulasi mencakup pembahasan tentang analisa implementasi zonasi dinamis pada ZRP serta *routing* metrik yang dibandingkan

antara lain PDR (*Packet Delivery Ratio*), RO (*Routing Overhead*) dan E2E (*end to end delay*).

4.2.1 Hasil uji coba pembobotan faktor stabilitas *link*

Sebelum melakukan uji coba pada protokol ZRP modifikasi dan ZRP tradisional, terlebih dahulu dilakukan rangkaian uji coba guna mencari pembobotan faktor stabilitas *link* yang akan digunakan pada ZRP modifikasi. Pengujian dilakukan menggunakan skenario simulasi peta *grid* dan *real*. Setiap varian jumlah *node*, dilakukan sebanyak 10 kali pengujian kemudian dihitung rata-rata dari hasil metrik analisisnya, dimana metrik analisis yang digunakan adalah *Packet Delivery Ratio*, *End-to-end Delay*, dan *Routing Overhead*. Pembobotan faktor yang digunakan seperti ditunjukkan oleh Tabel 4.2 dimana digunakan 3 varian pembobotan baru dan 1 varian pembobotan yang mengacu pada penelitian (Shen *et al.*, 2014).

Tabel 4.2 Pembobotan Faktor Stabilitas Link

Faktor Stabilitas <i>Link</i>	Pembobotan Faktor <i>Stabilitas Link</i>			
	Komposisi 1	Komposisi 2	Komposisi 3	PNT
Kecepatan	20	50	10	15
Akselerasi	10	10	10	10
Jarak	50	5	10	20
Kualitas <i>Link</i>	5	5	30	50
Jumlah <i>Node</i> Tetangga	15	30	40	5

Dari pengujian yang dilakukan, pada skenario *real* seperti yang ditunjukkan oleh Tabel 4.3 komposisi 2 menghasilkan nilai PDR yang paling baik. Begitu juga pada skenario *grid* seperti yang ditunjukkan oleh Tabel 4.4, komposisi 2 juga menghasilkan nilai PDR yang paling baik dibanding komposisi lainnya.

Tabel 4.3 Hasil PDR Uji Pencarian Komposisi pada Skenario Real

Jumlah Node	PDR ZRP (%)		
	Komposisi 1	Komposisi 2	Komposisi 3
25	79,71	81,57	83,17
50	87,19	90,43	87,86
100	64,54	77,21	69,43

Tabel 4.4 Hasil PDR Uji Pencarian Komposisi pada Skenario Grid

Jumlah Node	PDR ZRP (%)		
	Komposisi 1	Komposisi 2	Komposisi 3
25	47,54	49,11	49,86
50	85,30	85,07	83,59
100	81,33	85,27	83,03

Dari pengujian yang dilakukan, komposisi 3 memiliki hasil *end-to-end delay* yang paling baik pada skenario *grid* seperti ditunjukkan oleh Tabel 4.5, dimana hanya ketika jumlah *node* 100 pada simulasi komposisi 1 menghasilkan nilai yang lebih baik. Pada pengujian yang dilakukan di skenario *real* seperti ditunjukkan oleh Tabel 4.6, komposisi 1 menghasilkan nilai yang paling baik, kecuali saat jumlah *node* mencapai 100 dimana komposisi 2 lah yang menghasilkan nilai paling baik.

Tabel 4.5 Hasil End-to-end Delay Uji Pencarian Komposisi pada Skenario Grid

Jumlah Node	End to End Delay (ms) ZRP		
	Komposisi 1	Komposisi 2	Komposisi 3
25	1.922,26	2.422,02	1.870,53
50	550,60	552,69	401,66
100	84,90	108,21	100,73

Tabel 4.6 Hasil End-to-end Delay Uji Pencarian Komposisi pada Skenario Real

Jumlah Node	End to End Delay (ms) ZRP		
	Komposisi 1	Komposisi 2	Komposisi 3
25	968,80	1.177,14	1.322,88
50	139,84	361,22	209,61
100	132,06	66,23	81,17

Hasil RO dari pengujian menunjukkan komposisi 2 menghasilkan nilai yang paling baik. Seperti yang ditunjukkan oleh Tabel 4.7 komposisi memiliki nilai RO terendah pada skenario *grid*, begitu juga pada skenario *real* yang ditunjukkan oleh Tabel 4.8.

Tabel 4.7 Hasil Routing Overhead Uji Pencarian Komposisi pada Skenario Grid

Jumlah Node	RO ZRP		
	Komposisi 1	Komposisi 2	Komposisi 3
25	8.150	7.939	8.387
50	31.962	29.214	32.019
100	115.234	94.770	108.217

Tabel 4.8 Hasil Routing Overhead Uji Pencarian Komposisi pada Skenario Real

Jumlah Node	RO ZRP		
	Komposisi 1	Komposisi 2	Komposisi 3
25	9.264	8.579	9.357
50	34.919	29.426	34.188
100	119.892	87.753	108.671

Dari nilai PDR, *end-to-end delay*, maupun RO, komposisi 2 menghasilkan nilai yang lebih baik dibanding komposisi 1 dan komposisi 3. Selanjutnya, hasil pengujian komposisi 2 dibandingkan dengan komposisi faktor yang mengacu pada penelitian sebelumnya (Shen *et al.*, 2014).

Dari hasil pengujian, baik pada skenario *grid* seperti Tabel 4.9 dan skenario *real* seperti ditunjukkan oleh Tabel 4.10, Komposisi 2 menghasilkan nilai PDR yang lebih baik dari komposisi sebelumnya.

Tabel 4.9 Perbandingan Hasil PDR Pencarian Komposisi Faktor pada Skenario Grid

Jumlah Node	PDR ZRP (%)	
	Komposisi 2	PNT
25	49,11	46,52
50	85,07	81,22
100	85,27	80,74

Tabel 4.10 Perbandingan Hasil PDR Pencarian Komposisi Faktor pada Skenario Real

Jumlah Node	PDR ZRP (%)	
	Komposisi 2	PNT
25	81,57	81,28
50	90,43	90,07
100	77,21	64,90

Hasil *end-to-end delay* pada skenario *grid* yang ditunjukkan oleh Tabel 4.11 dan skenario *real* yang ditunjukkan oleh Tabel 4.12 menggambarkan bahwa komposisi 2 menghasilkan performa dengan hasil *delay* yang lebih baik.

Tabel 4.11 Perbandingan Hasil End-to-end Delay Pencarian Komposisi Faktor pada Skenario Grid

Jumlah Node	End to End Delay (ms) ZRP	
	Komposisi 2	PNT
25	2.422,02	1.740,79
50	552,69	249,94
100	108,21	84,44

Tabel 4.12 Perbandingan Hasil End-to-end Delay Pencarian Komposisi Faktor pada Skenario Real

Jumlah Node	End to End Delay (ms) ZRP	
	Komposisi 2	PNT
25	1.177,14	1.262,25
50	361,22	445,13
100	66,23	114,77

Dari hasil pengujian, komposisi 2 juga menghasilkan nilai RO yang lebih baik pada skenario *grid* seperti yang ditunjukkan oleh Tabel 4.13 dan skenario *real* yang ditunjukkan oleh Tabel 4.14.

Tabel 4.13 Perbandingan Hasil RO Pencarian Komposisi Faktor pada Skenario Grid

Jumlah Node	RO ZRP	
	Komposisi 2	PNT
25	7.939	8.356
50	29.214	33.002
100	94.770	117.742

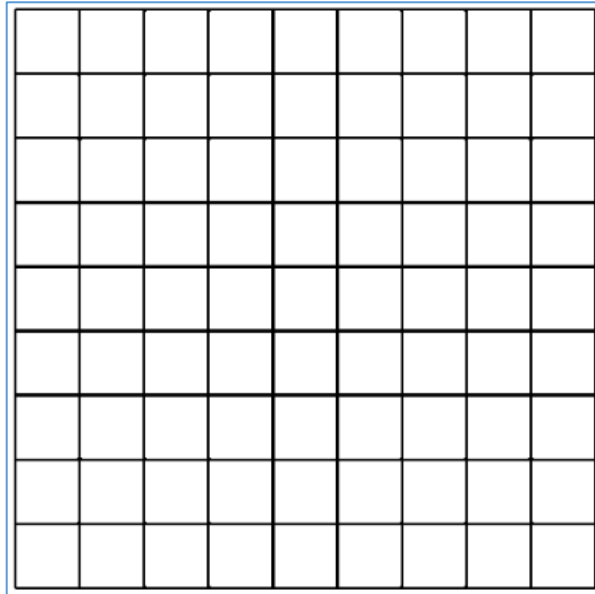
Tabel 4.14 Perbandingan Hasil RO Pencarian Komposisi Faktor pada Skenario Real

Jumlah Node	PDR RO ZRP	
	Komposisi 2	PNT
25	8.579	9.498
50	29.426	35.531
100	87.753	120.576

Mempertimbangkan hasil uji coba dengan melihat metrik analisis yang digunakan, komposisi 2 dipilih sebagai pembobotan faktor stabilitas *link* ZRP modifikasi yang dipakai pada uji coba tahap selanjutnya.

4.2.2 Hasil uji coba skenario *grid*

Uji coba skenario *grid* dilakukan sebanyak 10 kali dengan skenario mobilitas *random*. Peta *grid* yang digunakan memiliki luas area 1500 meter x 1500. Jumlah *node* yang digunakan pada uji coba sebanyak 25, dan 50 *node* sebagai representasi area dengan jumlah kepadatan yang jarang, 100 *node* sebagai representasi area dengan jumlah kepadatan yang sedang dan 200 *node* sebagai representasi area dengan jumlah kepadatan yang tinggi. *Node* sumber dan *node* pengirim diletakan secara statis dan tidak bergerak. Setiap *node* memiliki kecepatan maksimum 55 Km/jam. Rancangan simulasi peta *grid* dapat dilihat pada Gambar 4.1 Rancangan peta *grid*.

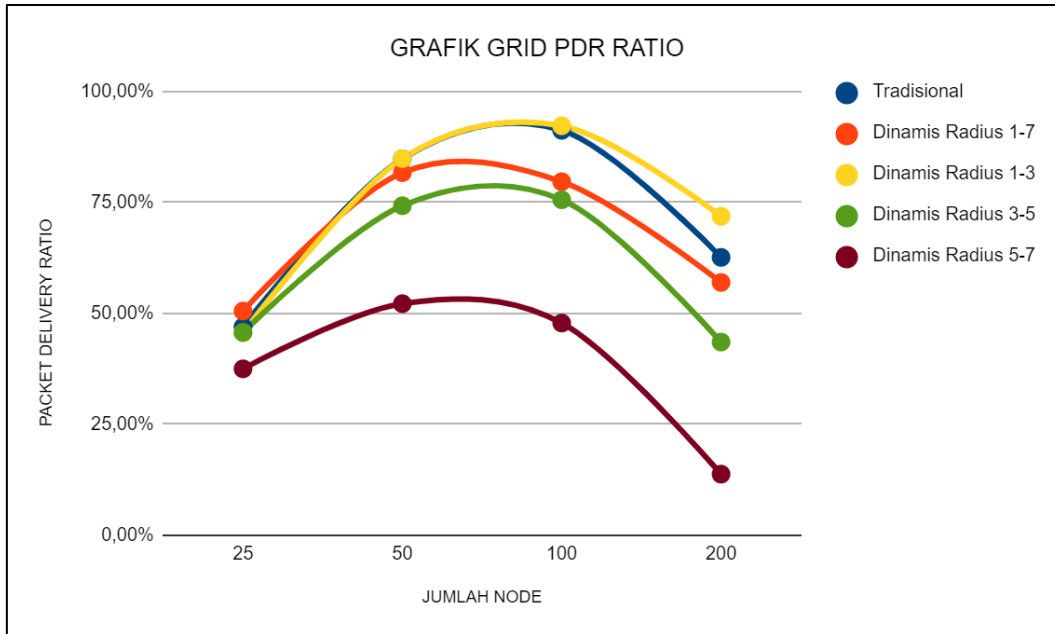


Gambar 4.1 Rancangan peta *grid*

4.2.2.1 Analisa implementasi zonasi dinamis

Protokol ZRP modifikasi melakukan dinamisasi zona dengan melakukan pembaruan nilai radius secara periodik. Perubahan nilai radius didapatkan dengan membandingkan nilai TWR dari sebuah *node* dengan rata-rata TWR dari *node* tetangganya. Variasi nilai radius dicoba untuk mendapatkan batasan nilai radius terbaik bagi protokol ZRP modifikasi. Variasi waktu pembaruan nilai radius juga diberikan untuk mengetahui waktu periodik terbaik untuk mendapatkan performa terbaik pada protokol ZRP modifikasi.

4.2.2.2 Packet Delivery Ratio (PDR)



Gambar 4.2 Grafik PDR skenario *grid* dengan variasi radius

Gambar 4.2 merupakan grafik perbandingan PDR antara protokol ZRP tradisional dengan protokol ZRP modifikasi dengan variasi batasan radius minimal 1 maksimal 7, minimal 1 maksimal 3, minimal 3 maksimal 5, dan minimal 5 maksimal 7. Hasil perhitungan rata-rata PDR dari skenario *grid* dengan penambahan jumlah node dapat dilihat pada Tabel 4.15.

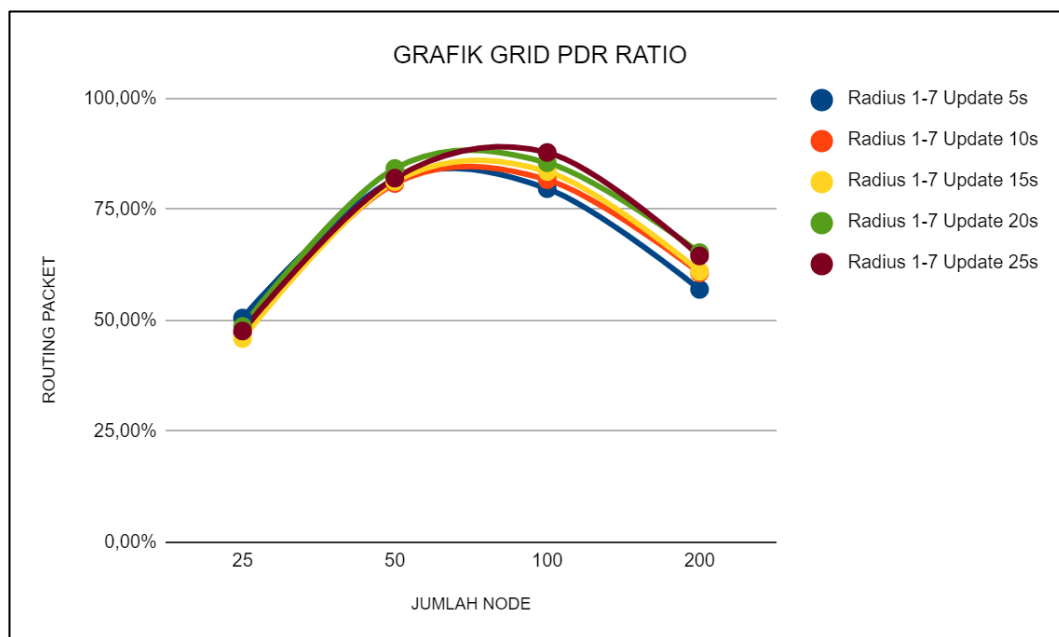
Tabel 4.15 Hasil perhitungan rata-rata PDR (%) skenario grid

Jumlah Node	PDR ZRP (%)				
	Tradisional	Radius 1 - 7	Radius 1 - 3	Radius 3 - 5	Radius 5 - 7
25	55,41	50,52	45,62	45,69	37,51
50	84,80	81,67	84,81	74,22	52,18
100	91,19	79,63	92,22	75,57	47,83
200	62,60	56,96	71,84	43,54	13,81

Berdasarkan Gambar 4.2 terlihat bahwa PDR dari protokol ZRP modifikasi dengan batasan radius minimal 1 maksimal 3 memiliki performa terbaik

dibandingkan dengan ZRP tradisional dan ZRP modifikasi dengan variasi batasan radius yang lain. Ketika pengujian dengan jumlah *node* 25, ZRP tradisional menghasilkan nilai PDR terbaik sebesar 55,41. Sementara ketika pengujian dengan jumlah *node* yang lainnya, ZRP modifikasi dengan batasan radius minimal 1 maksimal 3 selalu menghasilkan nilai PDR terbaik dibanding yang lainnya. Hasil pengujian menunjukkan batasan radius minimal 1 dan maksimal 3 *hop* adalah yang paling optimal, dimana performa ZRP modifikasi menjadi lebih baik dibanding ZRP tradisional. Hal ini dapat disebabkan karena zona yang terbentuk tidak terlalu besar maupun terlalu kecil sehingga sifat proaktif ZRP dapat bekerja dengan optimal dari zona yang terbentuk, baru kemudian bekerja secara reaktif untuk mencari *node* apabila berada di luar zona.

Hasil pengujian juga menunjukkan ZRP modifikasi dengan batasan radius minimal 5 maksimal 7 menghasilkan nilai PDR terendah di setiap variasi jumlah *node*, hal ini disebabkan karena mobilitas dari *node* yang tinggi yang menyebabkan *node* tersebut sudah berada di luar jangkauan sedangkan *node* tersebut masih tercatat berada di dalam zona oleh IARP pada ZRP.



Gambar 4.3 Grafik PDR skenario grid dengan variasi waktu pembaruan

Pada Gambar 4.2 protokol ZRP modifikasi memakai waktu pembaruan radius 5 detik. Pengujian selanjutnya dilakukan untuk mengetahui waktu pembaruan yang tepat bagi protokol ZRP modifikasi sehingga dapat mengeluarkan performa terbaik. Variasi waktu pembaruan yang dipakai adalah 5, 10, 15, 20 dan 25 detik. Sedangkan batasan radius yang dipakai adalah minimal 1 maksimal 7 hop. Hasil penghitungan PDR dari pengujian waktu pembaruan radius dapat dilihat pada Tabel 4.16.

Tabel 4.16 Hasil PDR ZRP Modifikasi dengan Variasi Waktu Pembaruan Radius

Jumlah Node	PDR ZRP dengan Variasi Waktu Pembaruan (%)				
	5 detik	10 detik	15 detik	20 detik	25 detik
25	50,52	47,05	45,89	48,65	47,61
50	81,67	80,79	81,25	84,20	82,01
100	79,63	81,65	83,49	85,47	87,79
200	56,96	60,66	61,00	65,25	64,49

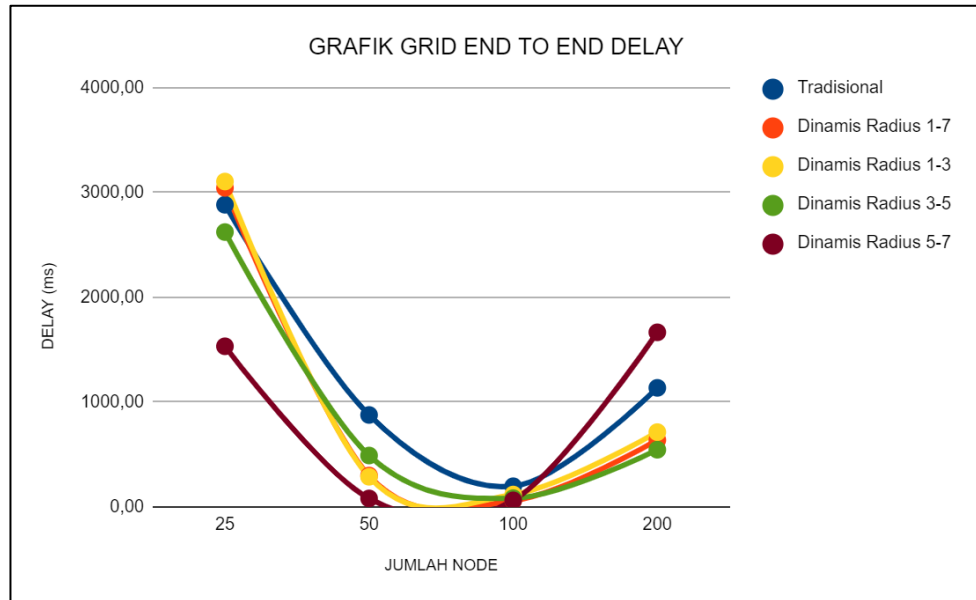
Seperti ditunjukkan oleh Gambar 4.3, ketika jumlah *node* pengujian 25, ZRP modifikasi menghasilkan PDR paling baik dengan waktu pembaruan radius 5 detik. Namun, ketika jumlah *node* adalah 50 dan 200, waktu pembaruan radius 20 detik menghasilkan nilai PDR tertinggi. Sementara, ketika jumlah *node* mencapai 100 *node*, waktu pembaruan radius 25 detik yang menghasilkan nilai PDR tertinggi.

Dari analisa tersebut, maka dapat disimpulkan bahwa pada skenario *grid*, ZRP modifikasi dengan radius minimal 1 maksimal 3 menghasilkan nilai PDR yang paling baik. Sedangkan waktu pembaruan radius 20 detik menghasilkan nilai PDR yang paling baik.

Hasil pengujian memperlihatkan PDR dari ZRP mengalami peningkatan dari pengujian dengan 25, 50, dan 100 *node* lalu kemudian mengalami penurunan ketika pengujian dengan 200 *node*. Dari Gambar 4.2 dan Gambar 4.3 menunjukkan titik puncak dari PDR ZRP diantara 50 dan 100 *node* sebelum mengalami grafik penurunan. Hal ini mungkin saja dipengaruhi oleh mobilitas dari *node* dan juga

tingkat kepadatan dari *node*, dimana dengan mobilitas yang tinggi dan lingkungan yang terlalu padat membuat *node* sudah berada di luar jangkauan zona.

4.2.2.3 Average End to End Delay



Gambar 4.4 Grafik rata-rata *end to end delay* skenario *grid* dengan variasi radius

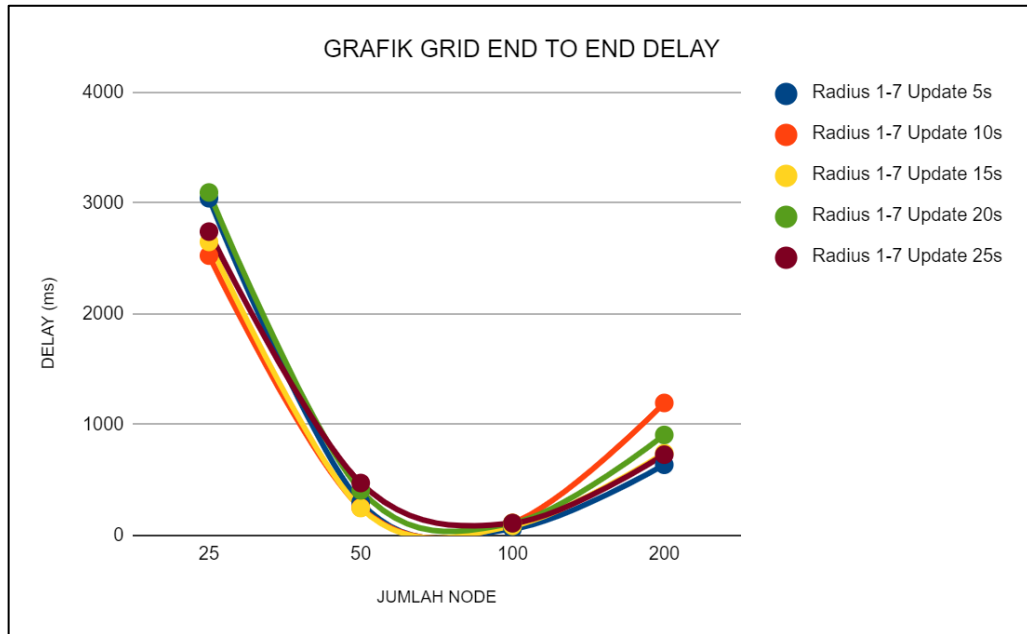
Gambar 4.4 menunjukkan grafik perbandingan rata-rata *end to end delay* pada protokol ZRP tradisional dan ZRP modifikasi dari skenario *grid*. Hasil perhitungan rata-rata dan selisih nilai *end to end delay* dari skenario *grid* dengan penambahan jumlah node dapat dilihat pada Tabel 4.17.

Tabel 4.17 Hasil perhitungan rata-rata *end to end delay* (milisekon) skenario *grid* dengan variasi radius

Jumlah Node	End to End Delay (ms) ZRP				
	Tradisional	Radius 1-7	Radius 1-3	Radius 3-5	Radius 5-7
25	2881,62	3042,52	3103,17	2621,68	1532,87
50	878,22	300,66	288,90	491,73	82,68
100	198,91	61,59	120,19	85,39	64,25
200	1136,72	639,13	714,61	547,03	1665,96

Seperti ditunjukkan Tabel 4.17 ZRP modifikasi cenderung memiliki nilai *end to end delay* yang lebih rendah dibanding ZRP tradisional. Dimana hanya ketika jumlah *node* 25, ZRP modifikasi dengan batasan radius minimal 1 dan maksimal 7 serta ZRP modifikasi dengan radius minimal 1 maksimal 3 yang memiliki *end-to-end delay* yang tidak lebih baik dari ZRP tradisional. Ketika jumlah *node* mencapai 200 *node* hanya ZRP modifikasi dengan batasan radius minimal 5 dan maksimal 7 yang tidak lebih baik dari ZRP tradisional. Seperti ditunjukkan pada Gambar 4.4 dan Tabel 4.17, ZRP modifikasi dengan batasan radius minimal 1 dan maksimal 7 memiliki nilai *end to end delay* yang paling rendah, berbanding tipis dengan ZRP modifikasi dengan batasan radius minimal 1 maksimal 3.

Hasil pengujian juga memperlihatkan *end-to-end delay* saat pengujian dengan 25 dan 50 *node* dari ZRP modifikasi dengan batasan radius yang besar memiliki *end-to-end delay* yang lebih baik, sementara saat pengujian dengan 100 dan 200 *node* menunjukkan ZRP modifikasi dengan batasan radius yang lebih kecil menghasilkan *end-to-end delay* yang lebih baik. Hal ini dapat terjadi karena batasan radius yang besar membuat zona yang terbentuk dapat memuat lebih banyak *node* sehingga dapat mengurangi *delay* di lingkungan dengan kepadatan *node* yang jarang. Sebaliknya, batasan radius yang lebih kecil membuat zona yang terbentuk lebih optimal saat pengujian dengan tingkat kepadatan *node* yang lebih padat. Mobilitas yang tinggi dari *node* bisa membuat *node* sudah tidak berada di dalam zona padahal IARP masih mencatat *node* tersebut di dalam zona, akhirnya ZRP dipaksa untuk bekerja secara reaktif sehingga meningkatkan *end-to-end delay*.



Gambar 4.5 Hasil Rata-rata End to End Delay (ms) dengan Variasi Waktu Pembaruan

Hasil penghitungan rata-rata *end to end delay* dengan variasi waktu pembaruan dari Gambar 4.5 dapat dilihat pada Tabel 4.18, dimana Pengujian dilakukan dengan menggunakan ZRP modifikasi dengan batasan radius minimal 1 maksimal 7.

Tabel 4.18 Rata-rata End to End Delay (ms) dengan Variasi Waktu Pembaruan Radius Skenario *Grid*

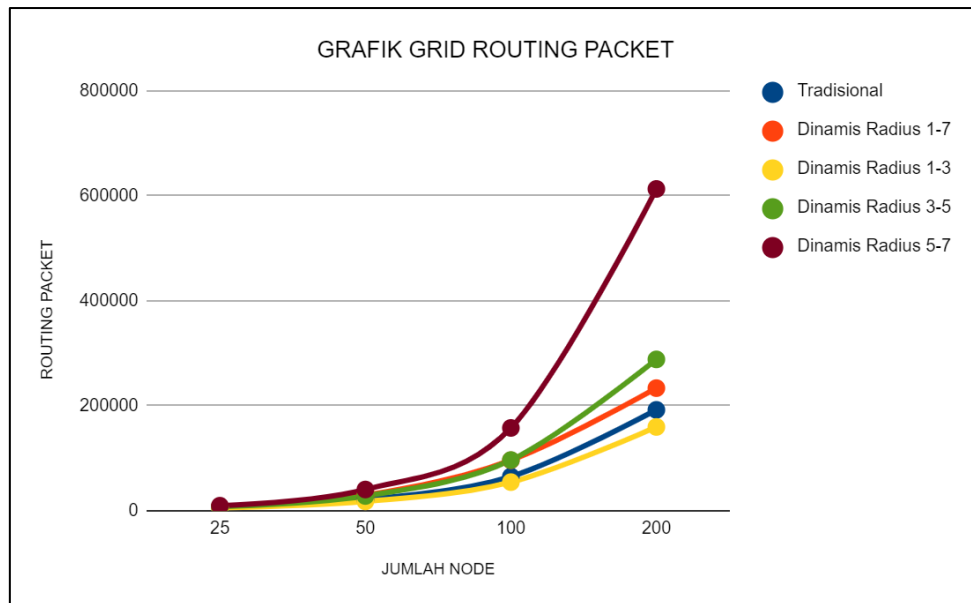
Jumlah Node	End to End Delay ZRP dengan Variasi Waktu Pembaruan (ms)				
	5 detik	10 detik	15 detik	20 detik	25 detik
25	3042,52	2525,40	2648,76	3094,70	2741,68
50	300,66	253,07	249,26	408,22	474,67
100	61,59	115,75	87,54	112,76	112,21
200	639,13	1196,64	744,15	908,03	730,20

Seperti ditunjukkan pada Gambar 4.5 *end to end delay* yang dihasilkan antar variasi waktu tidak memiliki selisih yang terlalu jauh. Namun dapat dilihat pada Tabel 4.18, ketika lingkungan pengujian adalah tingkat kepadatan yang jarang, jumlah node mencapai 25 dan 50, waktu pembaruan radius 15 detik menghasilkan

end to end delay paling baik, sedangkan untuk lingkungan kepadatan sedang (100 *node*) dan padat (200 *node*) yang menghasilkan *end to end delay* paling baik adalah waktu pembaruan radius 5 detik.

Dari analisa tersebut, dapat disimpulkan bahwa ZRP modifikasi memiliki nilai *end to end delay* yang lebih rendah daripada ZRP tradisional, dimana ZRP modifikasi dengan radius minimal 1 maksimal 7 memiliki *end to end delay* paling baik. Pada pengujian variasi waktu pembaruan radius, ketika lingkungan kepadatan rendah (20 dan 50 *node*) waktu pembaruan radius 15 detik menghasilkan *end to end delay* terbaik. Sedangkan ketika lingkungan kepadatan sedang (100 *node*) dan lingkungan kepadatan tinggi (200 *node*) waktu pembaruan radius 5 detik menghasilkan *end to end delay* paling baik.

4.2.2.4 Routing Overhead (RO)



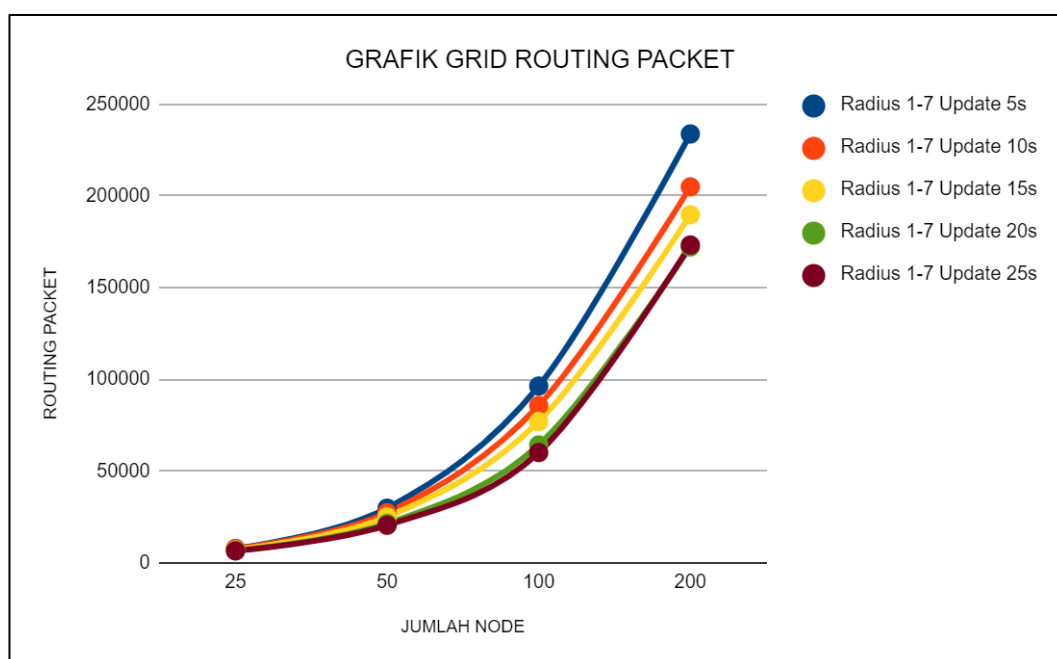
Gambar 4.6 Grafik rata-rata RO skenario *grid*

Gambar 4.6 menunjukkan grafik perhitungan rata-rata *routing overhead* pada skenario *grid* dimana nilai RO linear naik mengikuti naiknya jumlah *node* pada simulasi. Jumlah RO dari ZRP seperti ditunjukkan pada Tabel 4.19.

Tabel 4.19 Hasil Perhitungan Rata-rata RO pada Skenario *Grid*

Jumlah Node	RO ZRP				
	Tradisional	Radius 1- 7	Radius 1 – 3	Radius 3- 5	Radius 5 – 7
25	7538	7968	5992	8159	9975
50	21676	29924	18199	28479	40833
100	65565	96443	54861	96860	157984
200	192356	233726	159850	288442	612809

Seperti dapat dilihat pada Tabel 4.19 jumlah RO dari ZRP tradisional lebih kecil dibanding ZRP modifikasi, kecuali pada ZRP modifikasi dengan batasan radius minimal 1 maksimal 3. Semakin besar nilai radius memengaruhi jumlah RO karena dengan nilai radius yang besar, IARP pada ZRP akan bekerja lebih ekstra untuk menjaga keterbaruan informasi dari *node-node* yang berada dalam *intrazone*.



Gambar 4.7 RO ZRP Modifikasi dengan Variasi Waktu Pembaruan Radius

Pengujian dengan variasi waktu pembaruan radius juga dilakukan pada ZRP modifikasi. Sama seperti ketika pengujian PDR dan *end to end delay*, ZRP modifikasi menggunakan batasan radius minimal 1 dan maksimal 7. Hasil

penghitungan RO ZRP modifikasi dengan variasi waktu pembaruan radius seperti pada Tabel 4.20.

Tabel 4.20 RO ZRP Modifikasi dengan Variasi Waktu Pembaruan Radius

Jumlah Node	RO ZRP Modifikasi dengan Variasi Waktu Pembaruan Radius				
	5 detik	10 detik	15 detik	20 detik	25 detik
25	7968	7837	7227	6784	6678
50	29924	27471	25228	21651	20639
100	96443	85829	77134	64357	60215
200	233726	204905	189715	172227	173236

Seperti terlihat pada Gambar 4.7 dan Tabel 4.20, nilai RO dari ZRP modifikasi dengan waktu pembaruan radius 25 detik adalah yang paling rendah. Dapat dilihat pula bahwa semakin jarang waktu pembaruan radiusnya maka semakin rendah pula jumlah RO-nya. Hal ini dapat terjadi karena semakin jarang pembaruan dilakukan, maka semakin sedikit pula *routing packet* yang harus dikirimkan oleh setiap *node*.

Dari analisa tersebut, maka dapat disimpulkan bahwa ZRP modifikasi dengan batasan radius minimal 1 maksimal 3 memiliki nilai RO yang paling rendah. Sementara waktu pembaruan radius sebesar 25 detik memiliki menghasilkan RO terendah pada ZRP modifikasi. Hal ini tidak terlepas dari semakin jarang pembaruan yang dilakukan, semakin sedikit pula *routing packet* yang dikirimkan. Dan semakin tinggi nilai radius, semakin banyak *routing packet* dari IARP untuk menjaga informasi dari *node* yang berada dalam *intrazone*.

4.2.3 Hasil uji coba skenario *real*

Uji coba skenario *real* dilakukan sebanyak 10 kali dengan skenario mobilitas *random*. Peta *real* yang digunakan adalah peta daerah Semolowaru, Surabaya dengan luas area 1500 meter x 1500. Jumlah *node* yang digunakan pada uji coba sebanyak 25 dan 50 *node* sebagai representasi area dengan jumlah kepadatan yang jarang, 100 *node* sebagai representasi area dengan jumlah kepadatan yang sedang dan 200 *node* sebagai representasi area dengan jumlah

kepadatan yang tinggi. *Node* sumber dan *node* pengirim diletakan secara statis dan tidak bergerak. Setiap *node* memiliki kecepatan maksimum 50 km/jam. Rancangan simulasi peta *real* dapat dilihat pada Gambar 4.8 Rancangan peta *real*.

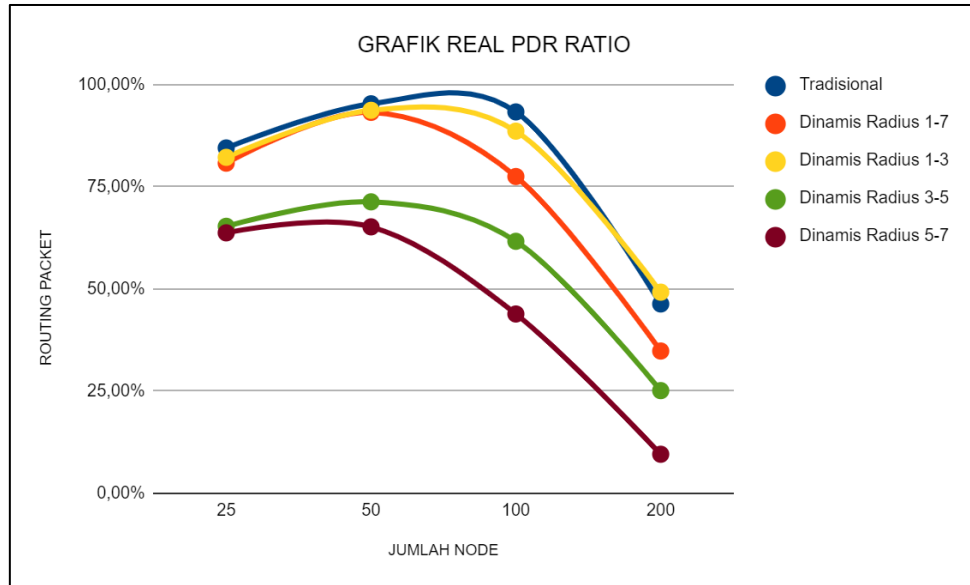


Gambar 4.8 Rancangan peta *real*

4.2.3.1 Analisa implementasi zonasi dinamis

Sama seperti pengujian pada skenario *grid*, pengujian pada ZRP pada skenario *real* juga menggunakan variasi batasan nilai radius dan variasi waktu pembaruan radius. Pada pengujian dengan variasi batasan radius, waktu pembaruan yang dipakai adalah 5 detik. Pada pengujian dengan variasi waktu pembaruan, nilai batasan radius yang dipakai adalah minimal 1 dan maksimal 7 *hop*.

4.2.3.2 Packet Delivery Ratio (PDR)



Gambar 4.9 Grafik PDR skenario *real* dengan variasi batasan radius

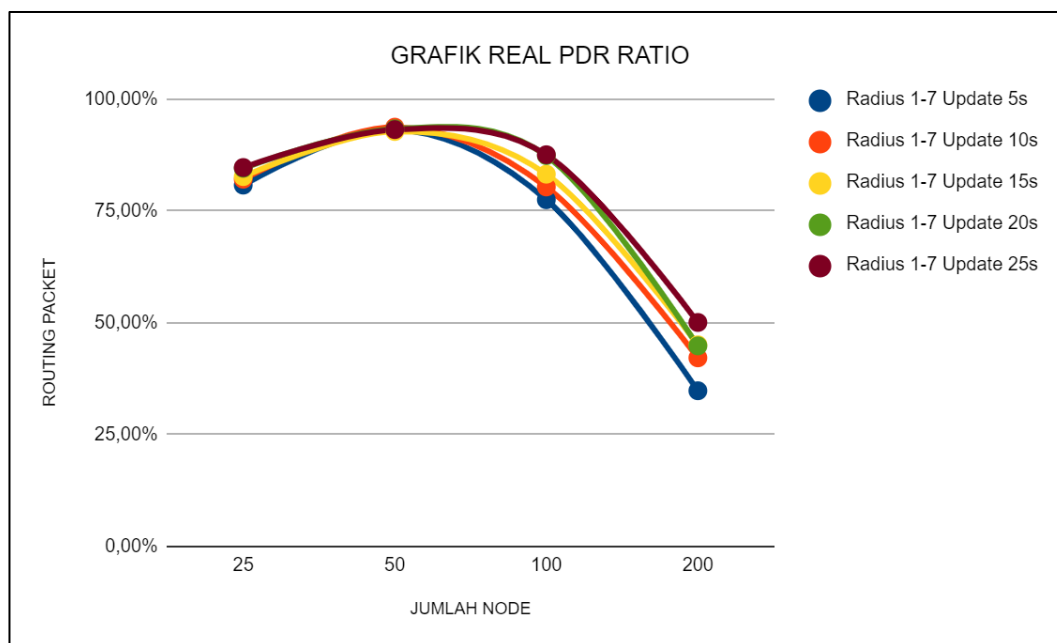
Gambar 4.9 merupakan grafik perbandingan PDR antara protokol ZRP tradisional dengan protokol ZRP modifikasi dengan variasi batasan nilai radius. Hasil perhitungan rata-rata nilai PDR dari skenario *real* dengan penambahan jumlah node dapat dilihat pada Tabel 4.21.

Tabel 4.21 Hasil perhitungan rata-rata PDR (%) skenario *real*

Jumlah Node	PDR ZRP (%)				
	Tradisional	Radius 1 - 7	Radius 1 - 3	Radius 3 - 5	Radius 5 - 7
25	84,47	80,75	82,17	65,31	63,72
50	95,26	93,17	93,66	71,29	65,18
100	93,27	77,48	88,58	61,65	43,88
200	46,34	34,83	49,22	25,15	9,60

Berdasarkan Gambar 4.9 dan Tabel 4.21, dapat dilihat bahwa PDR protokol ZRP modifikasi dengan batasan radius minimal 1 maksimal 3 memiliki PDR paling baik saat pengujian dengan jumlah *node* 200, sementara ZRP tradisional memiliki nilai PDR paling baik saat pengujian dengan jumlah *node* 25,

50 dan 100. Seperti pada pengujian di skenario *grid*, ZRP modifikasi dengan radius minimal 1 maksimal 3 memiliki PDR terbaik dibanding ZRP modifikasi dengan batasan radius lain. Namun, PDR ZRP tradisional masih lebih baik dari ZRP modifikasi di skenario *real*.



Gambar 4.10 Grafik PDR Skenario *Real* dengan Variasi Waktu Pembaruan Radius

Hasil penghitungan PDR dari ZRP modifikasi dengan batasan radius minimal 1 maksimal 7 seperti ditunjukkan pada Tabel 4.22.

Tabel 4.22 PDR ZRP Modifikasi dengan Variasi Waktu Pembaruan Radius pada Skenario *Real*

Jumlah Node	PDR ZRP dengan Variasi Waktu Pembaruan (%)				
	5 detik	10 detik	15 detik	20 detik	25 detik
25	80,75	82,07	82,64	84,50	84,68
50	93,17	93,69	92,72	93,32	93,14
100	77,48	80,33	83,19	87,34	87,54
200	34,83	42,18	45,08	44,85	50,07

Dapat dilihat dari Tabel 4.22 hasil pengujian menunjukkan waktu pembaruan sebesar 25 detik menghasilkan nilai PDR paling tinggi. Selisih PDR

ketika jumlah *node* mencapai 25 dan 50 *node* (kepadatan jarang) maksimal 3,93%, diambil dari selisih hasil PDR terendah dengan hasil PDR tertinggi. Ketika jumlah *node* naik menjadi 100 dan 200 *node* (kepadatan sedang dan tinggi) selisih hasil PDR mulai nampak signifikan. Selisih diantara hasil PDR terendah dengan PDR tertinggi mencapai 10,1% ketika jumlah *node* 100 dan 15,24% ketika jumlah *node* mencapai 200.

Sama seperti di skenario grid, grafik hasil pengujian juga menunjukkan peningkatan PDR dari saat pengujian dengan 25, 50, dan 100 *node* sebelum kemudian mengalami penurunan saat pengujian dengan 200 *node*. Hal ini berkaitan jumlah *node* yang masuk zona *intrazone* dan mobilitas dari *node*. Lingkungan dengan kepadatan tinggi menyebabkan IARP bekerja menjaga informasi *node* dalam jumlah yang cukup banyak, sedangkan faktor mobilitas *node* dapat membuat *node* sudah berpindah posisi sehingga tidak terjangkau lagi oleh IARP.

Tabel 4.23 Perbandingan PDR ZRP Modifikasi

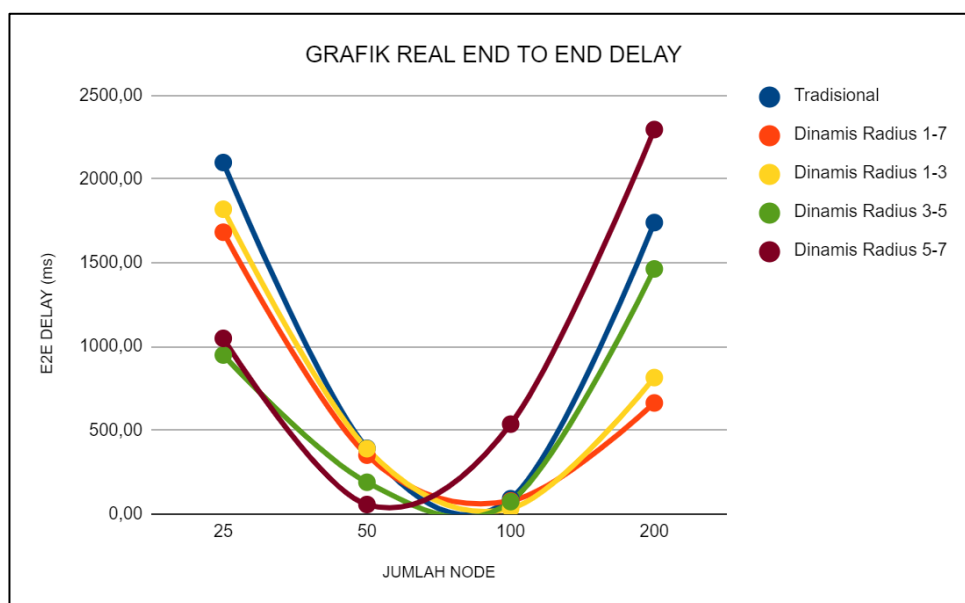
Jumlah Node	PDR ZRP Modifikasi (%)		
	ZRP tradisional	ZRP modifikasi radius minimal 1 maksimal 3 waktu pembaruan radius 5 detik	ZRP modifikasi radius minimal 1 maksimal 7 waktu pembaruan radius 25 detik
25	84,47	82,17	84,68
50	95,26	93,66	93,14
100	93,27	88,58	87,54
200	46,34	49,22	50,07

Dari hasil pengujian skenario *real*, dapat dilihat bahwa dengan waktu pembaruan radius 25 detik, ZRP modifikasi dengan batasan radius minimal 1 maksimal 7 dapat mengalami peningkatan nilai PDR yang signifikan dibanding saat waktu pembaruan radius dikonfigurasi 5 detik. Hasil PDR ZRP modifikasi dengan radius minimal 1 maksimal 7 dengan waktu pembaruan radius 25 detik bahkan mampu bersaing dengan ZRP tradisional dan ZRP modifikasi dengan batasan

radius minimal 1 maksimal 3 dengan waktu pembaruan radius 5 detik yang memiliki nilai PDR terbaik. Hasil perbandingan dapat dilihat pada Tabel 4.23.

Dari analisa tersebut, maka dapat disimpulkan bahwa ZRP tradisional menghasilkan PDR tertinggi pada skenario *real*. Sedangkan waktu pembaruan radius 25 detik menghasilkan PDR terbaik pada ZRP modifikasi.

4.2.3.3 Average End to End Delay



Gambar 4.11 Grafik rata-rata *end to end delay* skenario *real* (ms)

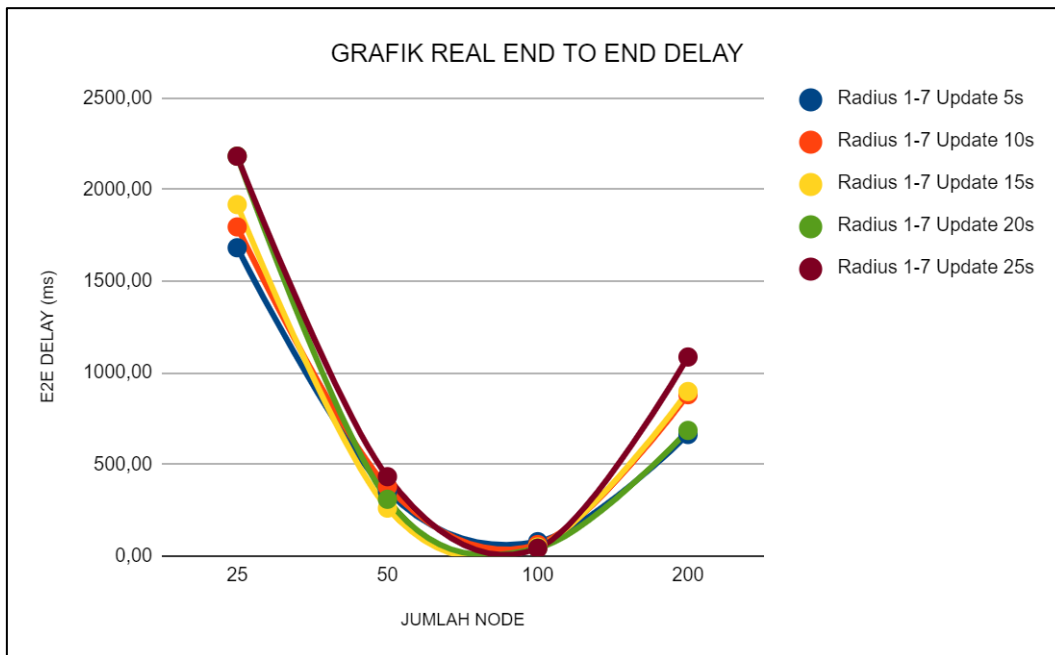
Tabel 4.24 Hasil perhitungan rata-rata *end to end delay* (ms) skenario *real*

Jumlah Node	End to End Delay (ms) ZRP				
	Tradisional	Radius 1 – 7	Radius 1 - 3	Radius 3- 5	Radius 5 – 7
25	2098,88	1683,08	1820,24	950,15	1049,70
50	393,77	352,62	391,25	191,05	58,12
100	92,72	79,69	34,44	75,44	537,81
200	1740,66	664,25	815,55	1464,02	2295,68

Gambar 4.11 menunjukkan grafik perbandingan rata-rata *end to end delay* pada protokol ZRP tradisional dan ZRP modifikasi pada skenario *real*, dimana

protokol ZRP modifikasi dengan batasan radius minimal 1 maksimal 3 memiliki nilai *delay* paling baik. Hasil perhitungan rata-rata nilai *end to end delay* dari skenario *real* dengan penambahan jumlah node dapat dilihat pada Tabel 4.24.

Tabel 4.24 menunjukkan bahwa ketika lingkungan kepadatan jarang, ZRP modifikasi dengan batasan radius minimal 3 maksimal 5 dan ZRP modifikasi radius minimal 5 maksimal 7 menghasilkan *end to end delay* paling baik. Namun, ketika lingkungan sedang dan padat, ZRP modifikasi radius minimal 1 maksimal 7 dan ZRP modifikasi dengan radius minimal 1 maksimal 3 memiliki *end to end delay* paling baik. Hal ini dapat terjadi karena dengan cakupan zona dengan radius kecil dan lingkungan yang jarang membuat ZRP harus secara reaktif mengirim paket melalui *intermediate* node yang berada diluar *intrazone*. Sedangkan ZRP modifikasi dengan radius besar dapat mencakup lebih banyak *node* di dalam *intrazone* pada lingkungan dengan kepadatan jarang. Pada lingkungan dengan tingkat kepadatan node tinggi, zona dengan radius yang terlalu besar justru memungkinkan menambah *delay* karena faktor mobilitas *node* dapat membuat *node* masih dicatat berada di dalam zona padahal *node* tersebut sudah tidak berada di dalam zona.



Gambar 4.12 End to End Delay (ms) pada Skenario Real dengan Variasi Waktu Pembaruan Radius

Hasil penghitungan rata-rata *end to end delay* pada ZRP modifikasi dengan radius minimal 1 maksimal 7 memakai variasi waktu pembaruan seperti ditunjukkan pada Tabel 4.25.

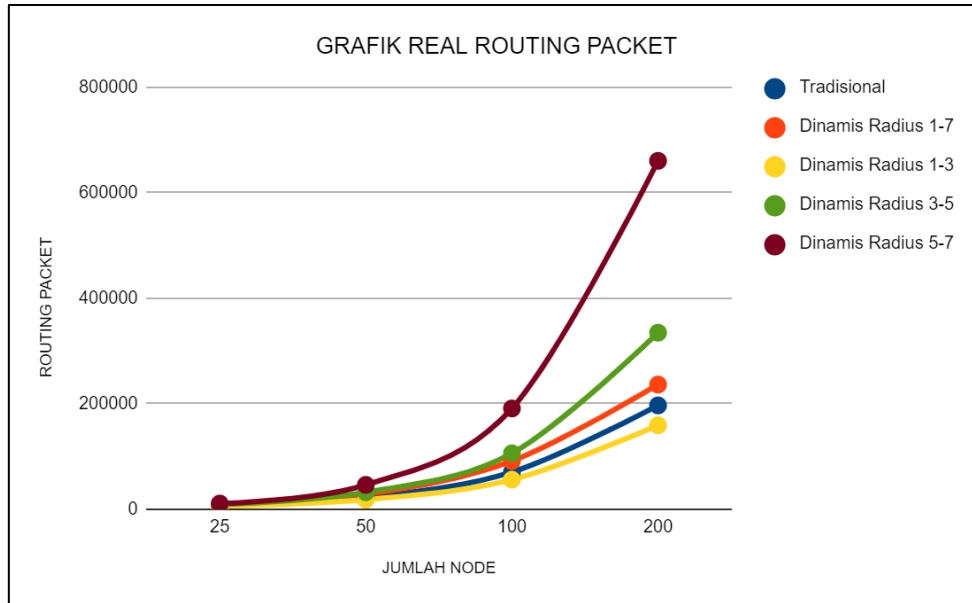
Tabel 4.25 End to End Delay (ms) dengan Variasi Waktu Pembaruan Radius Skenario Real

Jumlah Node	End to End Delay ZRP dengan Variasi Waktu Pembaruan (ms)				
	5 detik	10 detik	15 detik	20 detik	25 detik
25	1683,08	1796,50	1918,22	2182,56	2182,05
50	352,62	379,79	262,76	310,86	434,20
100	79,69	64,00	50,59	44,49	43,49
200	664,25	881,36	898,51	686,38	1086,94

Dilihat dari Gambar 4.12 dan Tabel 4.25, secara keseluruhan waktu pembaruan radius 5 detik memiliki *end to end delay* paling rendah. Dimana hanya ketika jumlah node 50 dan 100, waktu pembaruan radius 5 detik tidak menghasilkan nilai *end to end delay* terbaik.

Dari analisa tersebut, pada skenario *real* dapat disimpulkan bahwa pada lingkungan yang jarang, ZRP modifikasi dengan batasan radius besar memiliki nilai *end to end delay* yang baik, sementara untuk lingkungan kepadatan sedang dan tinggi ZRP modifikasi dengan batasan radius kecil menghasilkan *end to end delay* yang lebih baik. Waktu pembaruan radius sebesar 5 detik menghasilkan nilai *end to end delay* paling baik ketika pengujian dengan variasi waktu pembaruan radius.

4.2.3.4 Routing Overhead (RO)



Gambar 4.13 Grafik Rata-rata RO Skenario *Real*

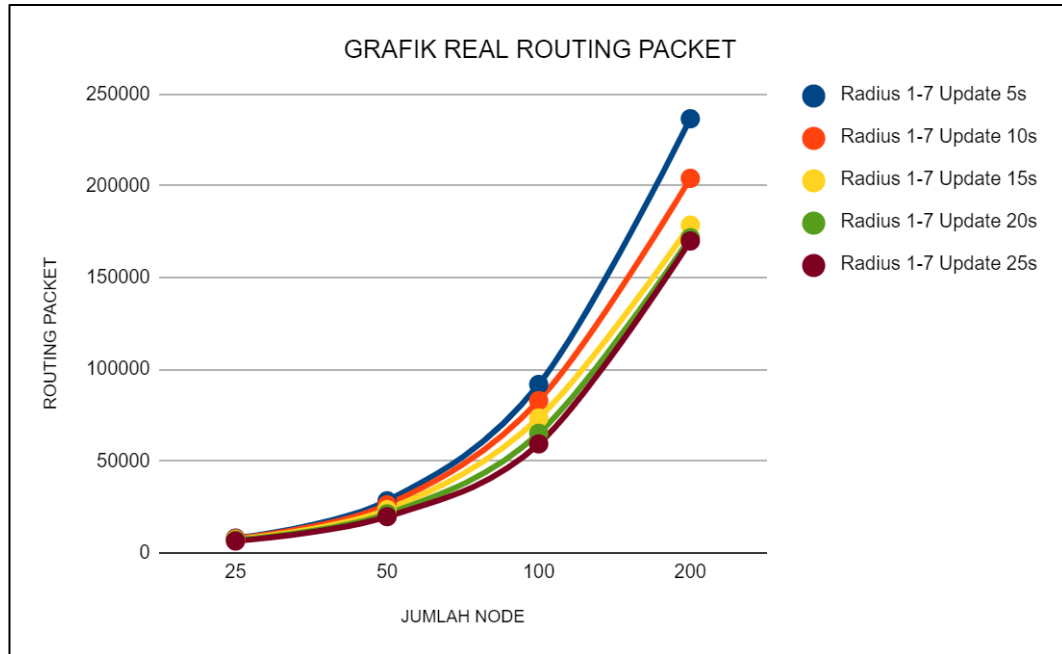
Gambar 4.13 menunjukkan grafik perhitungan rata-rata *routing overhead* pada skenario *real* dimana nilai RO linear naik mengikuti naiknya jumlah *node*. Hasil perhitungan RO skenario *real* dengan variasi batasan radius seperti pada Tabel 4.26.

Tabel 4.26 Hasil perhitungan rata-rata RO (paket) skenario *real*

Jumlah Node	RO ZRP				
	Tradisional	Radius 1 - 7	Radius 1 - 3	Radius 3- 5	Radius 5 – 7
25	8023	8175	6432	9242	11210
50	23058	28494	18649	32508	46581
100	70918	91863	56523	106353	191291
200	197029	236561	159477	334741	660132

Seperti ditunjukkan oleh Gambar 4.13 dan Tabel 4.26 ZRP modifikasi dengan batasan radius minimal 1 maksimal 3 memiliki jumlah RO paling sedikit. Sementara ZRP modifikasi dengan radius minimal 5 maksimal 7 memiliki RO

paling tinggi. Hal ini dapat terjadi karena nilai radius yang besar, membuat zona yang terbentuk dapat mencakup cukup banyak *node*. Hal ini mengakibatkan IARP mengirimkan cukup banyak paket untuk menjaga informasi dari *node* yang berada di dalam *intrazone*.



Gambar 4.14 RO ZRP Modifikasi dengan Variasi Waktu Pembaruan Radius

Gambar 4.14 menunjukkan nilai RO dari ZRP modifikasi dengan batasan radius minimal 1 maksimal 7 dengan variasi batasan waktu pembaruan radius. Hasil penghitungan RO seperti ditunjukkan oleh Tabel 4.27.

Tabel 4.27 Hasil RO ZRP Modifikasi dengan Variasi Waktu Pembaruan Radius

Jumlah Node	RO ZRP Modifikasi dengan Variasi Waktu Pembaruan Radius				
	5 detik	10 detik	15 detik	20 detik	25 detik
25	8175	7884	7377	7004	6605
50	28494	26314	23708	21381	19861
100	91863	83053	73589	65253	59486
200	236561	204118	178628	171746	170002

Seperti terlihat pada Gambar 4.14 dan Tabel 4.27, ZRP modifikasi dengan waktu pembaruan 25 detik menghasilkan nilai RO terendah. Hal ini dapat terjadi karena waktu pembaruan yang tidak terlalu rapat membuat ZRP tidak mengirimkan *routing packet* terlalu banyak untuk memperbarui informasi nilai radius.

Dari analisa tersebut, dapat disimpulkan bahwa pada skenario *real*, ZRP modifikasi dengan batasan radius minimal 1 maksimal 3 memiliki nilai RO yang paling rendah. Sementara waktu pembaruan radius sebesar 25 detik menghasilkan RO terendah pada ZRP modifikasi. Hal ini tidak terlepas dari semakin jarang pembaruan yang dilakukan, semakin sedikit pula *routing packet* yang dikirimkan. Semakin tinggi nilai radius, semakin banyak *routing packet* dari IARP untuk menjaga informasi dari setiap *node* yang berada dalam *intrazone*.

Berdasar hasil pengujian baik di skenario *real* dan *grid*, dengan mempertimbangkan metrik analisis PDR, *end-to-end delay*, dan RO, modifikasi ZRP dengan batasan radius minimal 1 maksimal 3 dan waktu pembaruan radius 25 detik menghasilkan performa yang paling optimal. Disimpulkan bahwa modifikasi ZRP dengan zonasi dinamis dapat meningkatkan performa dari ZRP di lingkungan VANETs.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pengujian dan analisis yang telah dilakukan menghasilkan beberapa kesimpulan penelitian, yaitu:

1. Nilai *stabilitas link* dirumuskan berdasarkan komposisi pembobotan faktor yang menghasilkan performa terbaik mengacu pada hasil pengujian, yaitu bobot kecepatan 50, bobot percepatan 10, bobot kualitas *link* 5, bobot jarak 5, dan bobot tingkat kepadatan *node* tetangga 30.
2. Nilai *stabilitas link* digunakan sebagai nilai acuan untuk menentukan besar nilai radius pada dinamisasi zona di ZRP. Berdasar hasil pengujian, dinamisasi zona pada ZRP yang mengacu pada nilai *stabilitas link* mampu membuat ZRP bekerja lebih optimal dengan syarat faktor batasan radius dan waktu pembaruan radius yang tepat.

5.2 Saran

Saran yang dapat diberikan dari hasil uji coba dan evaluasi adalah sebagai berikut:

1. Studi lebih lanjut tentang *stabilitas link* dan komposisinya sebagai formula acuan untuk mencari nilai radius.
2. Studi lain mengenai batasan radius dan waktu pembaruan radius dengan faktor acuan lain dalam melakukan dinamisasi zona.

(Halaman ini sengaja dikosongkan)

LAMPIRAN

Lampiran 1. Modifikasi *file* zrp.h

```
...
#include <math.h>
#include <algorithm>
#include <vector>
#include <zrp/mobilenode.h>
...
class Neighbor
{
public:
    ...
    int pnt; // Menyimpan nilai pnt node tetangga
    ...
};
...
class NeighborList
{
public:
    ...
    int thresholdTWR; // Menyimpan rata2 pnt node tetangga
    ...
};
...
class NDPAgent
{
public:
    ...
    float posX;
    float posY;
    double lastUpdateTime_ = 0.0;
    double lastAccel_ = 0.0;
    double lastSpeed_ = 0.0;
    double iSpeed;
    double now;
    float nb_x_;
    float nb_y_;
    double nb_accel_;
    double nb_speed_;
    int num_neighbor;
    double TWR;
    int fSpeed = 50;
    int fAccel = 10;
    int fDistance = 5;
    int fQuality = 5;
    int fNeighbor = 30;
    int timeModifier = 3;
    int maxTxRange = 250;
    int thresholdW = 100;
    double xDst = 750.00;
    double yDst = 750.00;
    ...
};
```

Lampiran 2. Modifikasi ZRPAgent : public Agent pada *file* zrp.h

```
class ZRPAgent : public Agent {
public:
    ...
    int intchecker = 0;
    ...
};
```

Lampiran 3. Modifikasi fungsi NeighborList::addNeighbor pada *file* zrp.cc

```
void NeighborList::addNeighbor(nsaddr_t addr, Time lastack,
int linkStatus, int pnt)
{
    Neighbor *newNb = new Neighbor(addr, lastack,
linkStatus, pnt); // Create a new Neighbor
    ...
    int threshold = thresholdTWR;
    threshold = ((threshold*(numNeighbors_-
1))+pnt)/numNeighbors_;
    thresholdTWR = threshold;
}
```

Lampiran 4. Modifikasi fungsi NDPBeaconTransmitTimer::handle pada *file* zrp.cc

```
void NDPBeaconTransmitTimer::handle(Event* e) {
    ...
    if (agent_->ndpAgt_.TWR < (agent_-
>ndpAgt_).neighborLst_.thresholdTWR && agent_->radius_ < 7)
{agent_->radius_ += 1;}
    else if (agent_->ndpAgt_.TWR > (agent_-
>ndpAgt_).neighborLst_.thresholdTWR && agent_->radius_ > 1)
{agent_->radius_ -= 1;}
    ...
}
```

Lampiran 5. Modifikasi fungsi NDPAgent::recv_NDP_BEACON pada *file* zrp.cc

```
void NDPAgent::recv_NDP_BEACON(Packet* p) {
    ...
    MobileNode *iNode;
    iNode = (MobileNode *)
(Node::get_node_by_address(agent_->myaddr_));
    posX = iNode->X();
    posY = iNode->Y();
    ...
}
```

Lampiran 6. Modifikasi fungsi NDPAgent::recv_NDP_BEACON_ACK pada file zrp.cc

```

void NDPAgent::recv_NDP_BEACON_ACK(Packet* p) {
    ...
    num_neighbor = (agent_
>ndpAgt_).neighborLst_.numNeighbors_;
    MobileNode *iNode;
    iNode = (MobileNode *)
(Node::get_node_by_address(agent_>myaddr_));
    iSpeed = ((MobileNode *) iNode)->speed();
    now = ((MobileNode *) iNode)->getUpdateTime();
    nb_x_ = iNode->X(); // x coordinate
    nb_y_ = iNode->Y(); // y coordinate
    if (now - lastUpdateTime_ <= 0.000000) { // if not
updated, just in case
        double nb_accel_ = lastAccel_;
    } else {
        // acceleration = delta speed / delta time
        nb_accel_ = (iSpeed - lastSpeed_) / (now -
lastUpdateTime_);
        lastAccel_ = nb_accel_;
        lastSpeed_ = iSpeed;
    }
    nb_speed_ = iSpeed; // speed
    lastUpdateTime_ = now; // Update its latest update time
    // TWR Calculation
    double nb_distance_;
    nb_distance_ = sqrt(((nb_x_ - xDst)*(nb_x_ - xDst)) +
((nb_y_ - yDst)*(nb_y_ - yDst)));
    double radius_x = posX - nb_x_;
    double radius_y = posY - nb_y_;
    double new_radius_x = radius_x*radius_x;

    double new_radius_y = radius_y*radius_y;
    double radius = std::min(sqrt((new_radius_x +
new_radius_y)), (double) maxTxRange);
    double quality = 1.0 / (1.0 - (radius / ((double)
maxTxRange + 1.0)));
    double modQuality = fQuality * quality;
    double modSpeed = fSpeed * nb_speed_;
    double modAccel = fAccel * nb_accel_;
    double modDistance = fDistance * nb_distance_;
    double modNeighbor = fNeighbor * num_neighbor;
    if(modAccel < 0){modAccel = modAccel*-1;}
    TWR = modSpeed + modAccel + modDistance + modNeighbor +
modQuality; //TWR
    ...
    if(foundNeighbor == TRUE) {
        ...
        handleToFoundNb->pnt = TWR;
        ...
    }
    else {
        ...
        neighborLst_.addNeighbor(hdrz->src_, now,
LINKUP, TWR);
        ... }
    ...
}

```

Lampiran 7. Modifikasi fungsi IERPAgent::recv_IERP_ROUTE_REQUEST_UNI file zrp.cc

```
void IERPAgent::recv_IERP_ROUTE_REQUEST_UNI(Packet* p) {
    ...
    // [Task-3]: Bordercast The Query...
    // [SubTask-1]: Bordercast it(Find all Next-Hop
neighbors to un-covered-perpheral-nodes)...
    ...
    for(int i=0; i<(agent_-
>iarpAgt_).pnLst_.numPerNodes_; i++) { // For all Peripheral
Nodes
        foundRouteFlag = (agent_-
>iarpAgt_).irLst_.findRoute(curPN->addr_,
&handleToFoundRoute);
        if(foundRouteFlag == TRUE) { // Add these
Neighbors into List
            ...
            if(foundNbFlag == FALSE &&
handleToFoundRoute->nextHop_ != hdrz->lastbc_ &&
handleToFoundRoute->nextHop_ != alreadySentNb && (agent_-
>pktUtil_).pkt_AmIOnTheRoute(p, handleToFoundRoute-
>nextHop_) == FALSE)
                {
                    NbLst.addNeighbor(handleToFoundRoute-
>nextHop_, 0.0, LINKDOWN, 0); //MODIFIKASI
                }
            }
            curPN = curPN->next_;
        }
    }
    ...
}
```

DAFTAR PUSTAKA

- Arifiyan, G., Djanali, S. dan Anggoro, R. (2017) “AODV-PNT Performance Study with Added Factor Number of Neighbor Nodes on VANET,” in *The 3rd International Seminar on Science and Technology*, hal. 69–71.
- Astya, R., Nand, P. dan Sharma, S. C. (2011) “Parametric analysis of zone routing protocol,” *Communications in Computer and Information Science*, 169 CCIS(91), hal. 247–251. doi: 10.1007/978-3-642-22577-2_33.
- Beijar, N. (2002) “Zone Routing Protocol (ZRP),” hal. 1–12.
- Chenn-Jung Huang *et al.* (2006) “A Zone Routing Protocol for Bluetooth MANET with Online Adaptive Zone Radius,” in *International Conference on Information Communications & Signal Processing*, hal. 579–583. doi: 10.1109/icics.2005.1689113.
- Haas, Z. J. dan Pearlman, M. R. (1998) “On the Performance of a Routing Protocol for the Reconfigurable Wireless Networks,” *Journal of Applied Statistics*, 11(2), hal. 1–4.
- Hoebeke, J. *et al.* (2010) “An overview of mobile ad hoc networks: Applications and challenges,” *Communications Network*, 2(1), hal. 87–110. doi: 10.5121/jgraphhoc.2010.2107.
- Huang, T. C., Cheng, W. C. dan Tang, L. (2012) “An effective scheme for detecting articulation points in zone routing protocol,” *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, 98 LNICST, hal. 521–533. doi: 10.1007/978-3-642-30493-4_50.
- Jain, N. dan Chaba, Y. (2014) “Simulation based Performance Analysis of Zone Routing Protocol in Manet,” *International Journal of Computer Applications*, 88(4), hal. 47–52. doi: 10.5120/15344-3684.
- Jaiswal, A. K. dan Singh, P. (2010) “Optimizing velocity based adaptive zone routing protocol,” in *2010 International Conference on Computer and*

Communication Technology, ICCCT-2010, hal. 149–152. doi: 10.1109/ICCCT.2010.5640413.

Jayanti, V. N. (2014) “Routing Protocols in MANET,” *International Journal of Computer Science and Mobile Computing*, 3(7), hal. 119–125.

Lakhtaria, K. (2014) “Analy zing Zone Routing Protocol in MANET Applying Authentic Parameter Analyzing Zone Routing Protocol in MANET Applying Authentic Parameter,” (May).

Li, F. dan Wang, Y. (2007) “Routing in Vehicular Ad Hoc Networks: A Survey,” *IEEE VEHICULAR TECHNOLOGY MAGAZINE*, hal. 12–22. doi: 10.1109/MVT.2007.912927.

Nasipuri, A. (2004) “Mobile Ad Hoc Networks,” in, hal. 1–26.

Natsheh, E. dan Buragga, K. (2010) “Density Based Routing Algorithm for Spare / Dense Topologies in Wireless Mobile Ad-Hoc Networks” 3(2), hal. 312–319.

Oigawa, Y. dan Sato, F. (2018) “Improvement in IntErzone Routing Protocol of ZRP Based on Bloom Filter,” *Journal of Information Processing*, 26(0), hal. 124–131. doi: 10.2197/ipsjjip.26.124.

Pearlman, M. R. dan Haas, Z. J. (1999) “Determining the optimal configuration for the zone routing protocol,” *IEEE Journal on Selected Areas in Communications*, 17(8), hal. 1395–1414. doi: 10.1109/49.779922.

Raja, L. dan Baboo, S. (2014) “An Overview of MANET: Applications, attacks and challenges,” *International Journal of Computer Science and Mobile Computing*, 3131(1), hal. 408–417. Tersedia pada: <http://ijcsmc.com/docs/papers/January2014/V3I1201472.pdf>.

Samar, P., Pearlman, M. R. dan Haas, Z. J. (2004) “Independent Zone Routing: An Adaptive Hybrid Routing Framework for Ad Hoc Wireless Networks,” *IEEE/ACM TRANSACTIONS ON NETWORKING*, 12(4), hal. 529–560. doi: 10.1007/978-1-4613-0223-0_14.

Shen, X. *et al.* (2014) “AODV-PNT : An Improved Version of AODV Routing

Protocol with Predicting Node Trend in VANET,” in *The 7th IEEE/International Conference on Advanced Infocomm Technology*, hal. 91–97.

Sinha, S. dan Sen, B. (2012) “Effect of Varying Node Density and Routing Zone Radius in ZRP: A Simulation Based Approach,” *International Journal on Computer Science and Engineering (IJCSE)*, 4(06), hal. 1069–1077.

Surateno (2007) *Optimasi Penentuan Zona Pada Protokol Routing Hopnet Dengan Teknik Min-Searching*.

Wang, A., Chao, H. dan Member, S. (2017) “Direction Density-Based Secure Routing Protocol for Healthcare Data in Incompletely Predictable Networks,” *IEEE Access*. IEEE, 4, hal. 9163–9173. doi: 10.1109/ACCESS.2016.2637887.

Yang, X. *et al.* (2018) “Improved ZRP Routing Protocol Based on Clustering,” *Procedia Computer Science*. Elsevier B.V., 131, hal. 992–1000. doi: 10.1016/j.procs.2018.04.242.

Yélémou, T. *et al.* (2012) “Impact of ZRP Zone Radius Value on Wireless Network Performance Tiguiane,” 92, hal. 164–173. doi: 10.1007/978-3-642-29093-0.

Yousefi, S., Mousavi, M. S. dan Fathy, M. (2007) “Vehicular Ad hoc Networks (VANETs): Challenges and perspectives,” in *ITST 2006 - 2006 6th International Conference on ITS Telecommunications, Proceedings*, hal. 761–766. doi: 10.1109/ITST.2006.289012.

Zheng, X. *et al.* (2017) “An Adaptive Density-based Routing Protocol for Flying Ad Hoc Networks,” 040113(October). doi: 10.1063/1.5005315.

Zhou, J., Cheng, Y. dan Lu, J. (2007) “Velocity Based Adaptive Zone Routing Protocol,” in *International Symposium on Intelligent Signal Processing and Communication Systems*, hal. 337–340.

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS



Muhsin Bayu Aji Fadhillah, lahir di Sukoharjo, 16 November 1996. Penulis adalah sulung dari dua bersaudara. Penulis menempuh pendidikan sekolah dasar di SDN Gayam 1 Sukoharjo, pendidikan sekolah menengah pertama di SMPN 1 Sukoharjo, dan menempuh pendidikan menengah atas di SMA Negeri 1 Sukoharjo. Selanjutnya penulis melanjutkan pendidikan sarjana dan pascasarjana di Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember Surabaya.

Penulis pernah aktif di berbagai organisasi, seperti staf Minat Bakat Himpunan Mahasiswa Teknik-Computer (HMTC) ITS, staf UKM Sepakbola ITS, dan wakil ketua Remaja Masjid Luhur Al-Ikhlas Medokan Semampir Surabaya. Penulis pernah melakukan kerja praktik di PT Perkebunan Nusantara IX Juni - Juli 2019 dan pernah bekerja di PT. Info Medik Solusi Juni – Oktober 2019. Penulis dapat dihubungi melalui nomor *handphone*: 087852387001 atau *email*: muhsin21baf@gmail.com

(Halaman ini sengaja dikosongkan)