



TESIS - KI142502

PENINGKATAN KINERJA *CLUSTERED SHORTEST GEOPATH ROUTING PROTOCOL* MENGGUNAKAN *ADAPTIVE PATH* DAN AGREGASI DATA PADA LINGKUNGAN *WIRELESS SENSOR NETWORK*

Rozita
5116201037

DOSEN PEMBIMBING

Waskitho Wibisono, S.Kom, M.Eng., Ph.D.
NIP. 19741022 200003 1 001

Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.
NIP. 19841016 200812 1 002

PROGRAM MAGISTER

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2018

[Halaman ini sengaja dikosongkan]



TESIS - KI142502

The Improvement of Performance Clustered Shortest Geopath Routing Protocol Using Adaptive Path and Data Aggregation in Wireless Sensor Network Environments

Rozita
5116201037

SUPERVISOR:

Waskitho Wibisono, S.Kom, M.Eng., Ph.D.
NIP. 19741022 200003 1 001

Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.
NIP. 19841016 200812 1 002

MASTER PROGRAM

DEPARTEMENT OF INFORMATICS

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2018

[Halaman ini sengaja dikosongkan]

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember Surabaya

oleh:
Rozita
NRP. 5116201037

Dengan judul :
Peningkatan Kinerja *Clustered Shortest Geopath Routing Protocol*
Menggunakan *Adaptive Path* dan Agregasi Data pada Lingkungan
Wireless Sensor Network

Tanggal Ujian : 20 Juli 2018
Periode Wisuda : September 2018

Disetujui oleh:

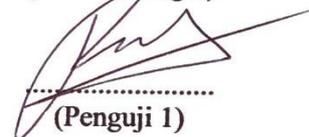
Waskitho Wibisono, S.Kom., M.Eng., Ph.D.
NIP. 197410222000031001


.....
(Pembimbing 1)

Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.
NIP. 1984101620081210002


.....
(Pembimbing 2)

Royyana Muslim I, S.Kom., M.Kom., Ph.D.
NIP. 197708242006041001

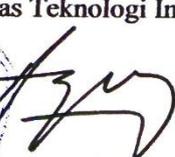

.....
(Penguji 1)

Tohari Ahmad, S.Kom., MIT., Ph.D.
NIP. 197505252003121002


.....
(Penguji 2)

Daniel Oranova Siahaan, S.Kom., M.Sc., P.D.Eng.
NIP. 197411232006041001


.....
(Penguji 3)


Dekan Fakultas Teknologi Informasi,

Dr. H. Agus Zainal Arifin, S.Kom., M.Kom
NIP. 197208091995121001

Peningkatan Kinerja *Clustered Shortest Geopath Routing Protocol* Menggunakan *Adaptive Path* dan Agregasi Data pada Lingkungan *Wireless Sensor Network*

Nama Mahasiswa : Rozita
NRP : 5116201037
Pembimbing I : Waskitho Wibisono, S.Kom, M.Eng., Ph.D.
Pembimbing II : Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.

ABSTRAK

Perkembangan teknologi informasi telah banyak mendukung kegiatan manusia, salah satunya adalah *wireless sensor network* (WSN) yang diimplementasikan pada bidang perawatan kesehatan, militer, pemantauan lingkungan dan lain sebagainya. WSN merupakan salah satu jaringan nirkabel yang terdiri dari banyak *node sensor* yang ditempatkan di wilayah tertentu untuk fungsionalitas tertentu. *Node sensor* pada WSN umumnya berukuran kecil dan memiliki kemampuan komputasi, komunikasi serta penginderaan. Tugas utama *node sensor* adalah mengumpulkan informasi dengan melakukan sensing target seperti panas, cahaya dan suhu yang berkomunikasi melalui sinyal radio jarak dekat. Kemudian informasi ini dikirimkan ke *sink* dalam bentuk respon *query* menggunakan *routing protocol* tertentu. *Sink* bertugas untuk mengumpulkan data penginderaan dari *node sensor*, kemudian meneruskannya ke perangkat atau sistem lain, seperti ke *database server* untuk penyimpanan. Namun *node sensor* memiliki keterbatasan memori, sumber daya pengolahan, energi dan masa pakai. Keterbatasan dari *node sensor* bisa mempengaruhi kinerja jaringan, salah satunya pada sisi *network lifetime*. salah satu *protocol* yang menangani masalah *network lifetime* adalah *Clustered Shortest geopath routing protocol* (CSGP).

Pada penelitian ini dikembangkan sebuah protocol dari metode CSGP dengan melakukan seleksi *cluster head* yang optimal yaitu berdasarkan jumlah tetangga terbanyak dan proses pergantian *cluster head* ketika sisa energi *cluster head* terpilih $\leq 50\%$. Proses seleksi ini dilakukan agar konsumsi energi lebih merata, kemudian untuk mengurangi jumlah transmisi di jaringan dilakukan agregasi data di *cluster head* berdasarkan prioritas data yang diterima oleh *cluster head*, jika data yang diterima merupakan prioritas tinggi maka akan langsung dikirim ke *sink* menggunakan jalur terpendek, akan tetapi jika data dengan prioritas sedang dan rendah maka akan ditampung sementara sehingga batasan tertentu. kemudian data tersebut akan di agregasi dengan cara melakukan rata-rata dari total data yang ditampung untuk tiap-tiap prioritas.

Hasil penelitian menunjukkan, bahwa metode CSGP-APA dapat meningkatkan *network lifetime* yaitu 602 menit lebih lama, *event detection* meningkat 4,6% lebih tinggi, *packet delivery ratio* meningkat 2,4% dan *latency* 15 ms lebih rendah dibandingkan dengan CSGP-WA.

Kata kunci : CSGP, Adaptive path, Agregasi data

[Halaman ini sengaja dikosongkan]

The Improvement of Performance Clustered Shortest Geopath Routing Protocol Using Adaptive Path and Data Aggregation in Wireless Sensor Network Environments

Name of Student : Rozita
NRP : 5116201037
Supervisor I : Waskitho Wibisono, S.Kom, M.Eng., Ph.D.
Supervisor II : Dr.Eng. Radityo Anggoro, S.Kom.,M.Sc.

ABSTRACT

The development of information technology had been supporting humans activity, one of them was a wireless sensor network (WSN) which implemented in the field of healthcare, military, environmental monitoring and so forth. WSN as one of the wireless networks that consisted of many sensor nodes that placed in certain areas for certain functionality. Sensor nodes on WSN were generally small and had computational, communication and sensing capabilities. The main task of sensor nodes collected information by sensing targets such as heat, light, and temperature communicating through short distance radio signals. Then the information sent to the sink in the form of a query response by using a specific routing protocol. the sink was responsible to collect sensing data from sensor nodes, then forwarding them to other devices or systems, such as to a storage server database. However, sensor nodes had limited memory, processing resources, energy and lifetime. Limitations of sensor nodes could affect network performance, such as on the network lifetime side. One protocol that handled network lifetime issues was Clustered Shortest Geopath routing protocol (CSGP).

In this research, developed the CSGP method by selecting the optimal cluster head based on the highest number of neighbors and processing of replacement cluster head when the remaining cluster head energy selected by $\leq 50\%$. The selection process was done in order to the energy consumption was spread evenly, then to decrease the number of network transmission was done by data aggregation in the cluster head based on the priority data received from the cluster head. If the data was received in high priority, then it would be directly sent to the sink by using the shortest path. Another side, if the data in medium and low priority, then it would be temporarily accommodated to a certain extent. Then, the data would be aggregated by calculating the average of the total data stored for each priority.

The results showed that CSGP-APA method can improve the network lifetime up to 602 minutes longer, packet delivery ratio improved by 2,4%, event detection (P1) improved by 4,6% and latency improved by 15 ms lower than to original CSGP.

Keywords: CSGP, Adaptive Path, Data Aggregation

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Segala puji syukur penulis panjatkan kepada ALLAH.SWT atas limpahan rahmat dan petunjuk-Nya, sehingga penulis mampu menyelesaikan tesis yang berjudul “Peningkatan Kinerja Clustered Shortest Geopath Routing Protocol Menggunakan Adaptive Path dan Agregasi Data pada Lingkungan Wireless Sensor Network”.

Dalam proses pembuatan tesis ini sangat banyak bantuan – bantuan yang penulis terima dari berbagai pihak, untuk itu tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada :

1. Kedua mak penulis, ibunda Saniah dan ibunda Nurhayati, yang senantiasa mendoakan serta memberikan support yang sangat berarti dalam segala aspek kehidupan penulis, kedua alm.abah, bapak Abdul Rahman dan bapak Jamil yang menjadi salah satu motivasi terbesar untuk penulis bisa menyelesaikan pendidikan ini.
2. Suami, Kakanda Izan dan anak-anak penulis, ananda Shofia Humaira dan ananda Omar Fakhri. Menjadi motivasi utama penulis untuk menyelesaikan pendidikan ini, yang senantiasa memberikan support, doa dan sanggup berkorban banyak hal.
3. Pembimbing I, Bapak Waskitho Wibisono, S.Kom., M.Eng., Ph.D. dan Pembimbing II, Bapak Dr.Eng. Radityo Anggoro, S.Kom, M.Sc. yang telah sangat banyak membimbing, memberikan kepercayaan, nasehat, solusi dan pembelajaran serta dukungan sehingga penulis bisa menyelesaikan tesis ini. Semoga allah membalas semua jasa baik bapak.
4. Bapak Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D., Bapak Tohari Ahmad S.Kom., MIT., Ph.D. dan Bapak Daniel Oranova Siahaan, S.Kom., M.Sc., P.D.Eng., selaku dosen penguji Tesis ini dan juga telah banyak memberikan saran untuk penyempurnaan Tesis ini.
5. Bapak Ibu Dosen program studi S2 Departemen Informatika yang telah memberi bimbingan perkuliahan sehingga menambah wawasan dan pengetahuan penulis.
6. Kementrian Riset, Teknologi dan Pendidikan Tinggi Republik Indonesia dan Lembaga Pengelola Dana Pendidikan yang telah memberikan beasiswa kepada penulis sehingga penulis dapat mengikuti perkuliahan Magister di Institut Teknologi Sepuluh Nopember.
7. Yaya, Ozzy, Vynska, eva, amel, ulum dan alifia yang telah banyak memberikan dukungan dan motivasi untuk bisa menyelesaikan pendidikan bersama, terima kasih untuk dua tahun kebersamaan dan belajar bersamanya.
8. Seluruh teman-teman angkatan 2016 yang tidak bisa penulis sebutkan satu persatu. Terima kasih untuk kekeluargaan, dukungan dan kerjasamanya.
9. Seluruh pihak yang tidak dapat disebutkan satu persatu, yang juga telah membantu penulis untuk menyelesaikan perkuliahan.

Tesis ini masih jauh dari kesempurnaan. Besar harapan penulis untuk dapat menerima kritik dan saran, sehingga dapat menjadi bahan perbaikan di penulisan berikutnya.

Surabaya, 20 Juli 2018

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN TESIS	v
ABSTRAK.....	vi
ABSTRACT	viii
KATA PENGANTAR.....	x
DAFTAR ISI	xii
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL	xvi
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	3
1.6. Kontribusi	3
BAB 2 KAJIAN PUSTAKA	5
2.1. <i>Wireless Sensor Network (WSN)</i>	5
2.2. <i>Low Energy Adaptive Clustering Hierarchy (LEACH)</i>	7
2.3. <i>Shortest Geopath Routing Protocol (SGP)</i>	8
2.4. <i>Clustered Shortest Geopath Routing Protocol (CSGP)</i>	9
2.5. Agregasi data	11
2.6. <i>Simulator and Integrated Development Platform for Sensor Networks Applications (SIDnet Swans)</i>	12
2.7. <i>Heartbeat protocol</i>	15
BAB 3 METODE PENELITIAN	17
3.1. Tahapan Penelitian.....	17
3.2. Studi Literatur	18
3.3. Perancangan Algoritma.....	18

3.3.1.	Menjalankan <i>Heartbeat Protocol</i>	20
3.3.2.	Pemilihan dan Pergantian <i>Cluster Head</i>	21
3.3.3.	Prioritas dan Agregasi Data	23
3.4.	Pengujian Algoritma.....	25
3.4.1.	Lingkungan pengujian	26
3.4.2.	Parameter Uji.....	26
3.5.	Analisa Pengujian.....	29
3.6.	Evaluasi Kinerja	30
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....		33
4.1.	Tahapan Implementasi Metode	33
4.1.1.	Modifikasi <i>Application layer</i>	35
4.1.2.	Modifikasi <i>Network Layer</i>	36
4.2.	Tahapan Uji Coba.....	40
4.2.1.	Skenario Pengujian	40
4.2.2.	Parameter Pengujian dan Eksekusi Simulator	42
4.3.	Hasil dan Analisis.....	45
4.3.1.	Analisis Skenario 1	45
4.3.2.	Analisis Skenario 2	49
4.3.3.	Analisis Skenario 3	53
4.3.4.	Analisis Skenario 4	57
4.3.5.	Analisis Skenario 5	60
4.3.6.	Analisis skenario 6.....	64
BAB V KESIMPULAN DAN SARAN		69
5.1.	Kesimpulan.....	69
5.2.	Saran.....	70
DAFTAR PUSTAKA		71

DAFTAR GAMBAR

Gambar 2.1. Alur pengiriman data (Brunelli & Rossi, 2013)	5
Gambar 2.2. Arsitektur Komunikasi <i>Wireless Sensor Network</i>	6
Gambar 2.3. WSN <i>Routing protocols</i> (Singh & Sharma, 2015)	6
Gambar 2 4. Bentuk <i>Cluster</i> (Astuti & Wibisono, 2017).....	10
Gambar 2 5. Agregasi data berbasis <i>cluster</i> (Desai et al., 2017).....	12
Gambar 2 6. Tampilan Graphical User Interface (GUI) SIDnet SWANS	13
Gambar 2 7. Arsitektur Operational SIDnet SWANS (Ghica, 2010).....	14
Gambar 2 8. Stack Jaringan SIDnet SWANS (Ghica, 2010)	15
Gambar 3.1. Diagram Alur Penelitian	17
Gambar 3. 2. Diagram alur <i>Heartbeat protocol</i>	21
Gambar 3. 3. Diagram Alur Pemilihan dan Pergantian <i>Cluster Head</i>	23
Gambar 3. 4. Diagram Alur Prioritas dan Agregasi Data.....	25
Gambar 4. 1. Modul <i>Layer</i> pada Simulator SIDnet SWANS(Ghica, 2010).....	35
Gambar 4. 2. Modifikasi pada <i>Application Layer</i>	36
Gambar 4. 3. Modifikasi pada <i>Network Layer</i>	39
Gambar 4. 4. <i>Input</i> Parameter Pengujian pada <i>Run Arguments</i>	43
Gambar 4.5. <i>Network Lifetime</i> Skenario 1.....	46
Gambar 4.6. <i>Latency</i> Skenario 1.....	46
Gambar 4.7. <i>Event Detection</i> Skenario 1.....	48
Gambar 4.8. <i>Packet Delivery Ratio</i> Skenario 1	48
Gambar 4.9. <i>Network Lifetime</i> Skenario 2.....	50
Gambar 4. 10. <i>Latency</i> Skenario 2.....	50
Gambar 4.11. <i>Event Detection</i> Skenario 2.....	52
Gambar 4.12. <i>Packet Delivery Ratio</i> Skenario 2.....	52
Gambar 4.13. <i>Network Lifetime</i> Skenario 3.....	54
Gambar 4.14. <i>Latency</i> Skenario 3.....	54
Gambar 4.15. <i>Event Detection</i> Skenario 3.....	56
Gambar 4.16. <i>Packet Delivery Ratio</i> Skenario 3	56
Gambar 4.17. <i>Network Lifetime</i> Skenario 4.....	57

Gambar 4.18. <i>Latency</i> Skenario 4.....	58
Gambar 4. 19. <i>Event Detection</i> Skenario 4.....	59
Gambar 4. 20. <i>Packet Delivery Ratio</i> Skenario 4	60
Gambar 4.21. <i>Network Lifetime</i> Skenario 5	61
Gambar 4.22. <i>Latency</i> skenario 5.....	62
Gambar 4.23. <i>Event Detection</i> Skenario 5	63
Gambar 4.24. <i>Packet Delivery Ratio</i> Skenario 5	64
Gambar 4.25. <i>Network Lifetime</i> Skenario 6.....	65
Gambar 4.26. <i>Latency</i> Skenario 6.....	65
Gambar 4.27. <i>Event Detection</i> Skenario 6.....	66
Gambar 4.28. <i>Packet Delivery Ratio</i> Skenario 6	67

DAFTAR TABEL

Tabel 3. 1. Parameter Uji.....	26
Tabel 4. 1. Rangkuman Parameter Pengujian Pada Simulator	44
Tabel 4.2. <i>Network Lifetime</i> dan <i>Latency</i> Hasil pengujian skenario 1	45
Tabel 4.3. <i>Event Detection</i> dan <i>Packet Delivery Ratio</i> Hasil pengujian skenario 1....	47
Tabel 4.4. <i>Network Lifetime</i> dan <i>Latency</i> Hasil pengujian skenario 2	49
Tabel 4.5. Event Detection dan packet delivery ratio Hasil pengujian skenario 2.....	51
Tabel 4.6. <i>Network Lifetime</i> dan <i>Latency</i> Hasil pengujian skenario 3	53
Tabel 4.7. <i>Event Detection</i> dan <i>Packet Delivery Ratio</i> Hasil pengujian skenario 3....	55
Tabel 4.8. <i>Network Lifetime</i> dan <i>Latency</i> Hasil pengujian skenario 4	57
Tabel 4. 9. <i>Event Detection</i> dan <i>Packet Delivery Ratio</i> Hasil pengujian skenario 4... 59	
Tabel 4. 10. <i>Network Lifetime</i> dan <i>Latency</i> Hasil pengujian skenario 2	61
Tabel 4.11. <i>Event Detection</i> dan <i>Packet Delivery Ratio</i> Hasil pengujian skenario 5..	62
Tabel 4.12. <i>Network Lifetime</i> dan <i>Latency</i> Hasil pengujian skenario 6	64
Tabel 4.13. <i>Event Detection</i> dan <i>Packet Delivery Ratio</i> Hasil pengujian skenario 6..	66

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Pada saat ini perkembangan teknologi terjadi sangat pesat, banyak kegiatan manusia sangat terbantu dengan adanya teknologi. Salah satu yang sudah banyak digunakan dan dimanfaatkan adalah *wireless sensor network* (WSN), yang bisa diimplementasikan pada banyak bidang seperti di bidang militer, pertanian, pemantauan lingkungan, kesehatan dan lain sebagainya. WSN merupakan salah satu jaringan nirkabel terdistribusi yang terdiri dari *sink* dan kumpulan dari *node sensor* yang tersebar disuatu area tertentu, yang digunakan sebagai media untuk melakukan proses *sensor*, mengumpulkan data sensing yang kemudian menghasilkan informasi sesuai dengan karakteristik tertentu sehingga dapat digunakan untuk memonitoring suatu kondisi tertentu seperti temperatur, radiasi, tekanan, kelembapan, getaran dan lain-lain. Setiap *node sensor* dilengkapi dengan peralatan sistem komunikasi sehingga *node* dapat saling terhubung dengan *node sensor* lainnya (Pratama & Suakanto, 2015)

Sistem komunikasi *node* pada WSN dilakukan secara nirkabel dan data sensing yang di peroleh oleh *node sensor* dikirim ke *sink* menggunakan *routing protocol* tertentu. *Node sensor* memiliki sumber daya yang terbatas seperti keterbatasan memori, energi, masa pakai serta sumber daya pengolahan. Keterbatasan ini dapat mempengaruhi kinerja jaringan, salah satu yang banyak dilakukan penelitian adalah keterbatasan energi. Kinerja jaringan yang bisa dipengaruhi oleh energi diantaranya adalah *network lifetime*, *event detected* dan *Latency* .

Low Energy Adaptive Clustering Hierarchy (LEACH) merupakan salah satu algoritma yang dikembangkan untuk meningkatkan kinerja jaringan pada sisi *network lifetime*. Algoritma ini pertama kali diperkenalkan oleh Heinzelman dengan memanfaatkan *routing protocol* berbasis hierarki atau *cluster* (Heinzelman, Chandrakasan, & Balakrishnan, 2000). *Cluster* dapat dibentuk dengan melakukan

proses *clustering*. Proses ini mengelompokkan data ke dalam beberapa *cluster*, sehingga data dalam satu *cluster* memiliki tingkat kemiripan yang tinggi (Tan, 2006). Jika diterapkan pada WSN kelebihan ini dapat menghemat konsumsi energi. Salah satu metode pengembangan dari LEACH adalah *Clustered Shortest Geopath Routing Protocol (CSGP)*.

CSGP merupakan salah satu *routing protocol* berbasis *cluster* dan berbasis lokasi. Pembentukan *cluster* dikembangkan berbentuk *hexagonal* dan *cluster head* dipilih berdasarkan posisi *node* yang paling dekat dengan titik tengah sebuah *cluster* yang dihitung berdasarkan jarak *euclidean*. *Node sensor* yang tidak menjadi *cluster head* secara otomatis akan menjadi *cluster member* dari *cluster* tersebut. Semua data *sensing* yang diperoleh *cluster member* di agregasi berdasarkan prioritas data sebelum dikirim ke *cluster head*, kemudian *cluster head* akan meneruskan data ke *sink* menggunakan *Shortest Geopath Routing Protocol* (Astuti & Wibisono, 2017). Agregasi merupakan Kumpulan dari berbagai data *sensor* selanjutnya diproses dijadikan sebuah data baru yang mewakili semua data yang sebelumnya telah dikumpulkan (Rajagopalan & K.Pramod, 2006).

Pengiriman data dari *cluster head* yang bersifat terpusat menuju *sink* menggunakan *Shortest Geopath Routing Protocol* dapat menyebabkan tidak meratanya konsumsi energi pada *node* yang berada pada jalur tersebut dan padatnya transmisi pada jalur tersebut dapat mempengaruhi kinerja jaringan.

Berdasarkan pemaparan diatas maka diperlukan skema pemilihan *cluster head* dan pergantian *cluster head* yang optimal untuk menentukan *adaptive path* sehingga jalur yang akan dilalui menuju *sink* akan berubah mengikuti *cluster head* terpilih dan konsumsi energi masing-masing *node* akan lebih merata. Untuk mengurangi kepadatan transmisi jaringan dilakukan agregasi data di *cluster head* berdasarkan prioritas data. Tahapan tersebut diatas dilakukan untuk dapat meningkatkan kinerja jaringan seperti *network lifetime*, *packet delivery ratio*, *event detection* dan *latency* . Implementasi dari tahapan tersebut akan dilakukan pada lingkungan simulator yaitu SIDNet SWANS yang merupakan simulator yang berjalan diatas simulator JiST-SWANS.

1.2. Rumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut:

1. Bagaimana menyeleksi *cluster head* yang optimal untuk pemilihan jalur secara *adaptive* sehingga dapat meningkatkan *network lifetime*, *event detection*, *packet delivery ratio* dan menurunkan *latency*.
2. Bagaimana proses dan penerapan agregasi data di *cluster head* untuk mengurangi jumlah transmisi di jaringan guna meningkatkan *network lifetime*.

1.3. Batasan Masalah

Batasan masalah pada penelitian ini adalah :

1. Simulasi yang digunakan adalah SIDNet SWANS versi 1.5.6.
2. Masing-masing *node* pada simulator memiliki 1 *sensor* suhu dan 1 *sensor* GPS untuk mengetahui lokasi *node*.
3. Jumlah *node* yang akan digunakan 100-300 *node*.
4. Luas area berkisar antara 300-500 m
5. Jumlah *sink node* pada simulator adalah 1 *node*.
6. Sumber energi dari *node sensor* bersifat terbatas pada simulator.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini untuk meningkatkan kinerja CSGP menggunakan seleksi *cluster head* yang optimal serta agregasi data.

1.5. Manfaat Penelitian

Manfaat penelitian ini adalah mengembangkan metode CSGP dalam hal seleksi *cluster head* yang optimal dan melakukan agregasi data di *cluster head* agar kinerja *wireless sensor network* menjadi lebih baik.

1.6. Kontribusi

Mengusulkan seleksi *cluster head* yang optimal sehingga konsumsi energi *node* lebih merata, mengembangkan metode agregasi data di *cluster head* berdasarkan prioritas data guna mengurangi jumlah transmisi di jaringan. Hal ini

dilakukan untuk meningkatkan kinerja CSGP pada lingkungan *wireless sensor network*.

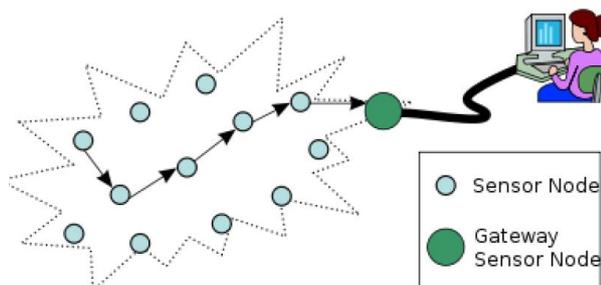
BAB 2

KAJIAN PUSTAKA

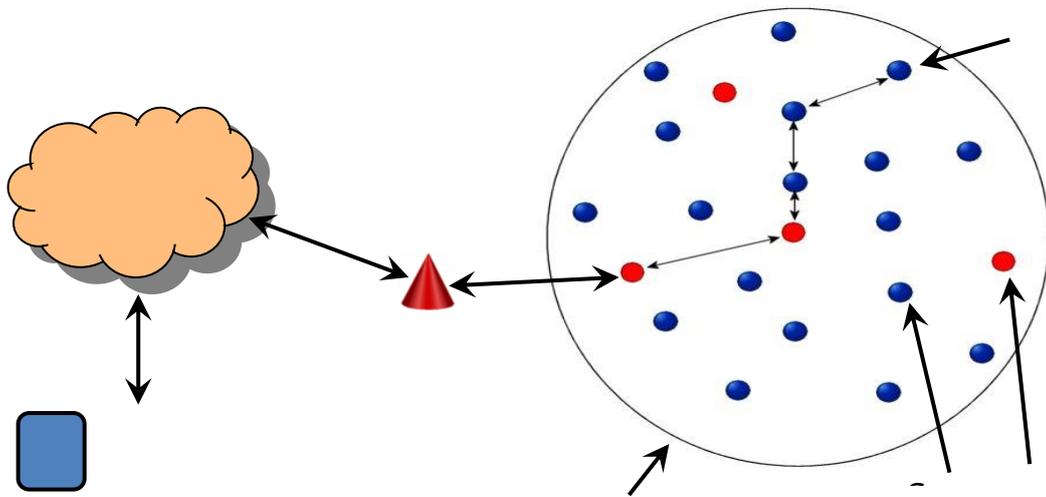
2.1. *Wireless Sensor Network* (WSN)

Wireless Sensor Network merupakan salah satu jaringan nirkabel terdistribusi yang terdiri dari *sink* dan kumpulan dari *node sensor* yang tersebar disuatu area tertentu, yang digunakan sebagai media untuk melakukan proses *sensing*, mengumpulkan data *sensing* yang kemudian menghasilkan informasi sesuai dengan karakteristik tertentu sehingga dapat digunakan untuk memonitoring suatu kondisi seperti temperatur, radiasi, tekanan, kelembapan, getaran dan lain – lain. Setiap jenis *sensor* memiliki perangkat lunak (aplikasi dan sistem operasi) dan perangkat keras masing – masing kemudian digabungkan dan dijalankan kedalam sistem *wireless sensor network*. Setiap *node sensor* dilengkapi dengan peralatan sistem komunikasi sehingga *node* dapat saling terhubung dengan *node sensor* lainnya. (Pratama & Suakanto, 2015)

Data *sensing* yang di peroleh oleh *node sensor* akan dikirim ke *sink* atau *gateway* pusat melalui beberapa *node sensor* lainnya, hal ini dilakukan karena keterbatasan hardware dan untuk menghemat energi masing – masing *node*. WSN menggunakan komunikasi rendah daya yang berarti jarak transmisi masing – masing *node* terbatas, sehingga setiap *node sensor* harus mengirimkan data ke *node sensor* lainnya hingga paket data sampai ke *sink*. Alur pengiriman data terlihat seperti pada gambar 2.1. (Brunelli & Rosi, 2013) dan arsitektur komunikasi *wireless sensor network* terlihat pada gambar 2.2. (Mihoubi, Rahmoun, Lorenz, & Lasla, 2017)

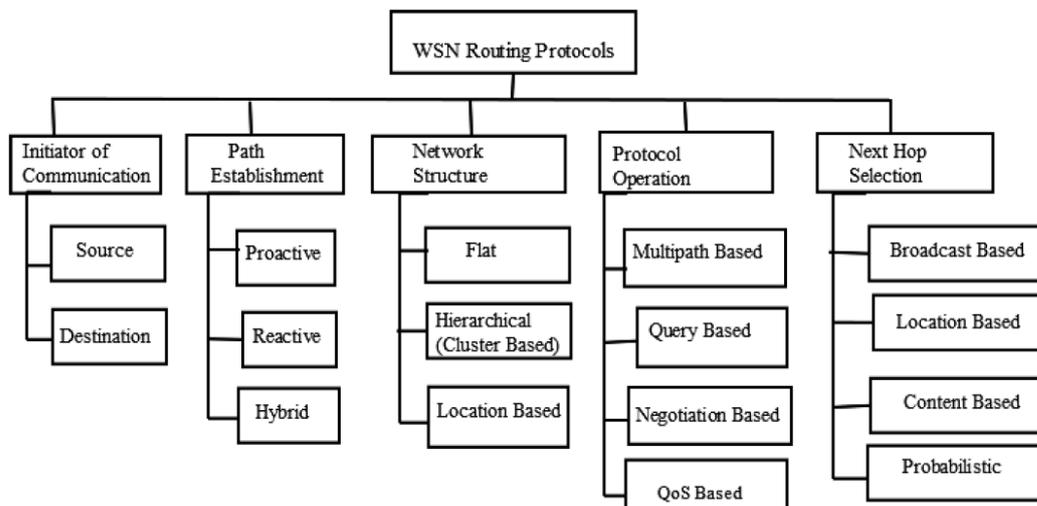


Gambar 2.1. Alur pengiriman data (Brunelli & Rossi, 2013)



Gambar 2.2. Arsitektur Komunikasi *Wireless Sensor Network* (Mihoubi et al., 2017)

Pengiriman data dari *node sensor* tidak bisa lepas dari pengaturan *Routing protocol* seperti terlihat pada gambar 2.3. *WSN Routing protocols* dengan struktur jaringan pada WSN terbagi menjadi 3 bagian yaitu *flat-based*, *hierarchical based* dan *location-based*, (Heinzelman et al., 2000). Pada penelitian ini menggunakan *routing protocol location based* yaitu *Shortest Geopath Routing Protocol (SGP)*. *Routing protocol* ini memiliki kelebihan dalam menghemat energi, karena *Node sensor* tidak perlu melakukan pencarian dan perhitungan yang rumit untuk bisa menemukan *node* tetangganya dan *sink*.



Gambar 2.3. *WSN Routing protocols* (Singh & Sharma, 2015)

2.2. *Low Energy Adaptive Clustering Hierarchy (LEACH)*

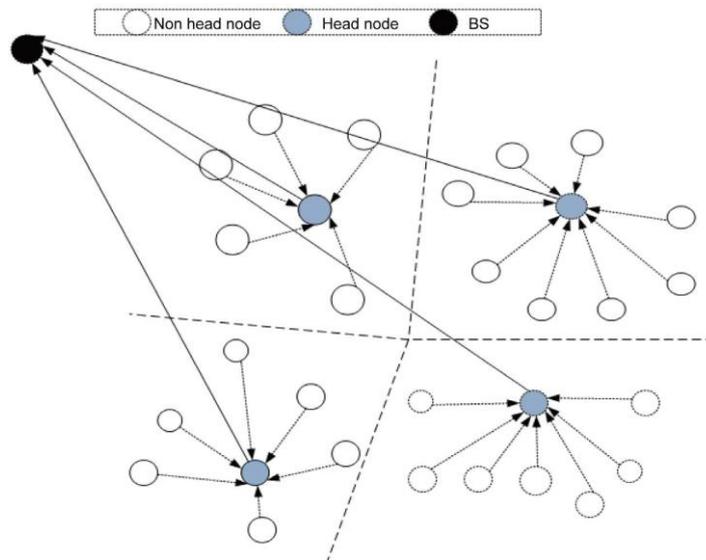
Metode *clustering* merupakan salah satu metode yang dikenal dengan keunggulan dalam skalabilitas dan komunikasi yang efisien. Hal ini digunakan untuk menghemat energi di WSN karena dalam metode *cluster*, *node sensor* yang memiliki energi tertinggi akan digunakan untuk memproses dan mengirimkan informasi ke *sink*, sedangkan *node sensor* yang memiliki energi lebih rendah dapat digunakan untuk melakukan *sensing* di area target.

Salah satu metode *clustering* yang selalu digunakan dan dikembangkan adalah *Low Energy Adaptive Clustering Hierarchy (LEACH)*. LEACH merupakan *routing protocol* yang membentuk *cluster* dari kumpulan *node* berdasarkan kekuatan sinyal yang diterima dan menjadikan *cluster head* sebagai pengirim paket data ke *sink* (Heinzelman et al., 2000). Algoritma ini melalui dua fase yaitu fase *setup* dan fase *steady state*. Pada fase *setup* dimulai dengan pemilihan suatu *node* sebagai *cluster head (CH)* dan pembentukan *cluster* atau yang disebut dengan algoritma *clustering*.

Pada awalnya CH akan ditentukan secara acak berdasarkan level energi yang tersisa pada interval waktu tertentu. Pada interval berikutnya, setiap *node sensor* n mengambil sebuah bilangan acak x sedemikian rupa sehingga $0 < x < 1$, dan membandingkannya dengan *threshold* tertentu $T(n)$. Jika $x < T(n)$, maka *node* tersebut menjadi *cluster head* pada interval tersebut. Jika tidak, maka menjadi *cluster member*. Selanjutnya CH akan mengirimkan pesan *broadcast* ke *node sensor* dan jika *node sensor* menerima pesan tersebut akan membalas dengan pesan bahwa *node sensor* tersebut akan bergabung sebagai anggota *cluster*. Sedangkan Pada fase *steady state* terjadi proses pengiriman data antar *node* yang memakan waktu lebih lama dibandingkan dengan fase *setup* karena pengiriman data terjadi melalui transmisi radio secara intensif sedangkan fase *setup* hanya menentukan CH dan pembentukan *cluster*.

Pada awalnya *node-node* tersebar dalam jumlah besar pada suatu area dan proses pengiriman data masih terpusat pada *sink*. Namun dengan adanya algoritma LEACH, *node-node* tersebut dikelompokkan dalam beberapa *cluster* pada satu jaringan WSN. Masing-masing *cluster* memiliki sebuah *cluster head* yang bertugas

untuk mengkoordinasi pengiriman data dari *node sensor* ke *sink*. Mekanisme ini dapat menghemat energi karena hanya CH yang melakukan pengiriman data ke *sink* sedangkan *node sensor* hanya mengirim data ke CH, sehingga konsumsi energi *node sensor* berkurang dan menyebabkan *network lifetime* jadi lebih maksimal. Arsitektur LEACH secara sederhana dapat direpresentasikan oleh gambar 2.3. sebagai berikut :



Gambar 2.3. Arsitektur LEACH (Saeed, Murad, Nawaz, & Irum, 2017)

Akan tetapi algoritma LEACH memiliki masalah jika sebaran *Node* tidak merata, maka CH yang terpilih bisa berada pada pinggir *cluster* sehingga tidak bisa menjangkau semua *node sensor* secara langsung (Handy, Hasse, & Timmermann, 2002). Hal ini diperbaiki oleh penelitian selanjutnya seperti algoritma perilaku fauna semut (Permatasari & Wibisono, 2015). Algoritma *cluster* berbasis lokasi dengan memanfaatkan teknologi GPS, membagi area WSN menjadi beberapa grid dan pengiriman data dilakukan dengan teknik *geographical routing* (Jannu & Jana, 2014)

2.3. Shortest Geopath Routing Protocol (SGP)

Protokol *routing* SGP merupakan salah satu protokol *routing* dalam kategori *location-based routing*. Untuk bisa menjalankan protokol ini, masing-masing *sensor node* harus memiliki penyedia layanan lokasi, misalnya GPS. *Source node*

menggunakan informasi geografis dari *node* tujuan untuk menemukan rute pengiriman. (Imielinzi & Navas, 2016).

Dalam SGP, *node* yang mengirim pesan akan meneruskan pesan ke tetangganya yang memiliki jarak paling dekat dengan tujuan. Jarak antara suatu *node* dengan tujuan dapat dihitung dengan rumus *Euclidean* jika koordinat kedua *node* diketahui pada bidang dua dimensi (x,y)(Ardakani, 2014) (Sarkar, Yin, Gao, Luo, & Gu, 2009).

Syarat digunakannya protokol SGP adalah:

- a. Masing-masing *node* mengetahui posisi geografisnya
- b. Masing-masing *node* mengetahui posisi geografis tetangga satu langkahnya
- c. Tujuan *routing* didefinisikan sebagai sebuah *node* pada koordinat posisi dua dimensi, atau sebuah area geografis yang direpresentasikan dengan polygon tertutup
- d. Setiap packet dapat membawa sejumlah kecil informasi *routing* Protokol SGP dengan teknik greedy dapat menemukan kegagalan jika terdapat hole, atau area dimana posisi pesan terakhir berada pada *node* yang paling dekat dengan *sink*, namun bukan merupakan *sink*, dan tidak ada tetangga yang lebih dekat dengan *sink* dibandingkan *node* terakhir pembawa pesan.

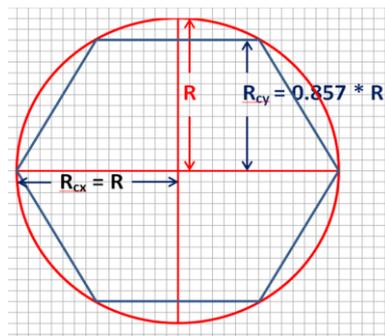
2.4. *Clustered Shortest Geopath Routing Protocol (CSGP)*

Clustered Shortest Geopath Routing Protocol merupakan *protocol* yang dikembangkan dari *LEACH* yang berbasis lokasi yaitu menggunakan jalur terdekat menuju *sink*(SGP), dengan menggunakan SGP pengiriman paket data akan dilakukan dengan memilih rute terpendek untuk bisa sampai ke *sink*. Pada metode ini dibagi menjadi beberapa tahapan yaitu *Neighbor Recovery*, *Clustering*, *Route Discovery* dan *Cluster maintenance* (Astuti & Wibisono, 2017)

Tahapan *neighbor recovery*, setelah *node sensor* di sebarakan selanjutnya *node sensor* akan menjalankan *Heartbeat protocol* (seggelman R, Tuexen M, 2012). *Heartbeat protocol* digunakan untuk menemukan *node* tetangga dan mengumpulkan informasi koordinat masing – masing *node* tetangga, informasi

yang dikumpulkan berupa jarak *euclidean* antara *node sensor* dengan tetangganya dan semua informasi tersebut disimpan dalam *array neighbor list*.

Selanjutnya tahapan *clustering*, *sink* akan membagi wilayah menjadi sejumlah *k cluster*, Bentuk *cluster* yang akan digunakan berbentuk *hexagonal* karena lebih menyerupai jangkauan transmisi suatu *node sensor* seperti terlihat pada gambar 2.4. Bentuk *Cluster*, dimana *R* adalah jangkauan Transmisi *node*, *R_{cx}* adalah jarak dari titik tengah ke koordinat X tertinggi dan terendah dan *R_{cy}* merupakan jarak dari titik tengah ke koordinat Y tertinggi dan terendah. *Node sensor* yang berada paling dekat dengan titik tengah hexagonal maka *node* tersebut akan menjadi *Cluster head (CH)* karena merupakan posisi paling ideal untuk dicapai oleh semua *node sensor* hanya dengan satu hop.



Gambar 2.4. Bentuk *Cluster* (Astuti & Wibisono, 2017)

Kemudian untuk mengetahui tata letak dan jumlah *cluster* perlu menentukan jumlah baris (*r*) dan kolom (*c*) didapat dari membagi nilai panjang (*p*) dan lebar (*l*) area WSN dengan nilai *R_{cx}* dan nilai *R_{cy}* seperti pada persamaan (2.1) dan (2.2).

$$r = \frac{l}{2 \times R_{cy}} + 1 \quad (2.1)$$

$$c = \frac{l}{2 \times R_{cx}} + 1 \quad (2.2)$$

Setelah mengetahui jumlah dan baris maka akan diperoleh jumlah *Cluster* (*nκ*) dengan (2.3), selanjutnya akan dapat menghitung koordinat titik tengah masing – masing *cluster*

$$n\kappa = r * c \quad (2.3)$$

Tahapan *route discovery* melakukan proses pengiriman paket data yang telah dikumpulkan dan diagregasi di *node sensor* berdasarkan prioritas tertentu.

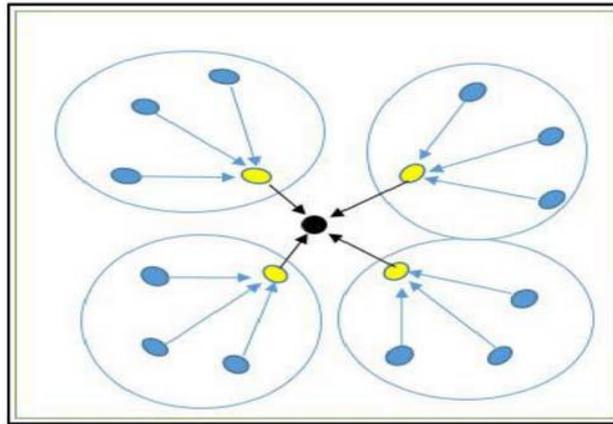
data *sensing* dari *node sensor* akan dikirim ke CH melalui satu langkah (one hop) sedangkan data dari CH akan dikirim ke *sink* menggunakan *Shortest Geopath Routing Protocol* yaitu memilih jalur dengan jarak terpendek menuju *sink*. Setelah melakukan pengiriman data ke *sink* maka energi CH akan lebih cepat berkurang dibandingkan *node sensor* lainnya, oleh karena itu diperlukan tahapan *Cluster maintenance* dimana pada tahapan ini akan dilakukan pemilihan ulang CH, jika sisa energi CH pertama kurang dari 60% maka akan dilakukan pemilihan CH kedua berdasarkan sisa energi tertinggi dan jarak kedua terdekat dengan titik tengah *cluster*, jika sisa energi CH kurang dari 30% maka akan dilakukan pemilihan CH yang ketiga. Setiap pergantian CH maka secara otomatis jalur SGP nya juga berubah secara dinamis.

2.5. Agregasi data

Agregasi data adalah proses mengumpulkan dan menggabungkan informasi yang berguna di wilayah tertentu (Randhawa & Jain, 2017). Agregasi juga bisa diartikan Kumpulan dari berbagai data *sensor* tersebut selanjutnya diproses dijadikan sebuah data baru yang mewakili semua data yang sebelumnya telah dikumpulkan (Rajagopalan & K.Pramod, 2006). Pengumpulan data dianggap sebagai prosedur pemrosesan mendasar yang bisa mengurangi konsumsi energi dan menghemat sumber daya yang terbatas. Teknik pengumpulan data yang efektif dapat meningkatkan efisiensi energi dan umur jaringan (Qayyum, Saeed, & Roberts, 2014). Agregasi data juga bisa berguna bila beberapa *node* merasakan fenomena yang sama karena kepadatan *node* yang tinggi, juga dikenal sebagai data overlapping problem (Cayirci, 2003).

Ada beberapa teknik agregasi data yang bisa digunakan yaitu : pendekatan terpusat, pendekatan dalam jaringan, pendekatan berbasis pohon dan pendekatan berbasis *cluster*. Untuk penelitian ini menggunakan pendekatan berbasis *cluster* yang dapat dilihat pada gambar 2.5. Pendekatan berbasis *cluster* adalah pendekatan yang paling efisien dimana jaringan dibagi kedalam kelompok yang berisi dengan *node –node*. Pada setiap *cluster* ada satu *node* yang dipilih sebagai *cluster head* dengan menggunakan algoritma tertentu dan setelah satu proses *cluster head* berubah sehingga optimalisasi dapat dicapai. *Cluster head* melakukan agregasi

dengan mengurangi ukuran paket serta redundansi dan kemudian mengirimkan paket ke lokasi yang diperlukan (Desai, Vyas, R.Nair, & Jain, 2017).

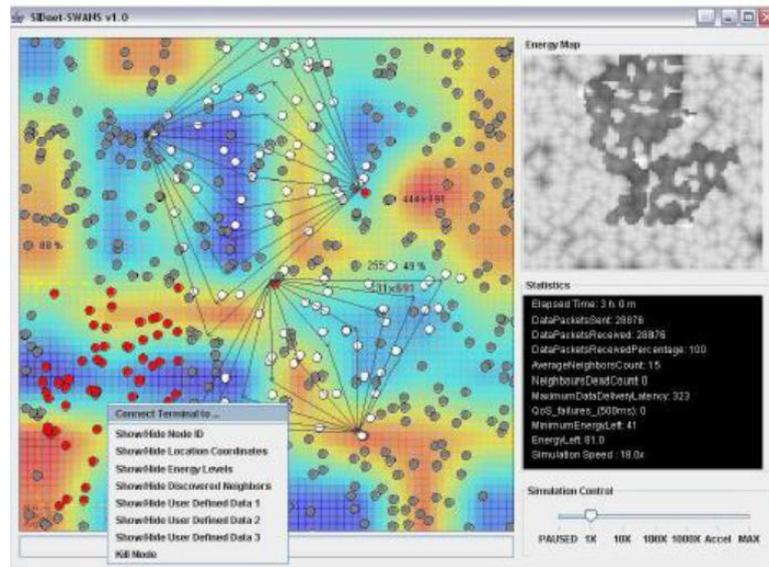


Gambar 2 5. Agregasi data berbasis *cluster* (Desai et al., 2017)

Pada jaringan agregasi data dibagi menjadi dua jenis yaitu *with size reduction* yang merupakan metode yang melakukan kombinasi dan kompresi pada data yang dikumpulkan dengan tujuan untuk mengurangi informasi yang dikirim melalui jaringan dan *without size reduction* yang merupakan metode yang dilakukan dengan cara menyatukan paket menjadi satu kesatuan tanpa adanya pengurang dari besar total paket yang disatukan.

2.6. Simulator and Integrated Development Platform for Sensor Networks Applications (SIDnet Swans)

SIDnet SWANS merupakan simulator berbasis java dan paket distribusi SIDnet SWANS termasuk didalamnya komponen JIST (*Java in Simulation Time*) dan SWANS (*Scalable Wireless Ad-hoc Network Simulator*) yang dirancang khusus untuk aplikasi *sensor network*. Simulator ini dipatenkan oleh Northwestern University pada tahun 2008 dan yang menciptakan adalah Oliviu C. Ghica dkk. (Ghica, 2010). Untuk dapat menampilkan *Graphical User Interface* (GUI) berbasis *java swing* seperti terlihat pada gambar 2.6. Tampilan GUI SIDnet SWANS.



Gambar 2 6. Tampilan Graphical User Interface (GUI) SIDnet SWANS (Ghica, 2010)

Dengan tampilan GUI SIDNet SWANS memungkinkan pengguna untuk berintraksi dengan sistem, dibutuhkan tiga file utama yaitu (Ghica, 2010) :

a. *Application layer implementation* yang menerapkan beberapa langkah agar skenario yang diinginkan bisa bekerja : menunggu intraksi pengguna, mengirim permintaan *query* dan memproses *query* yang masuk dengan mengambil contoh fenomena yang mendasarinya, kemudian mengirimnya kembali ke pengguna. *Node* pengguna yang menghubungkan dan mengirim *query* disebut sebagai *sink*, sedangkan *node* yang mengambil data menjadi *node* sumber.

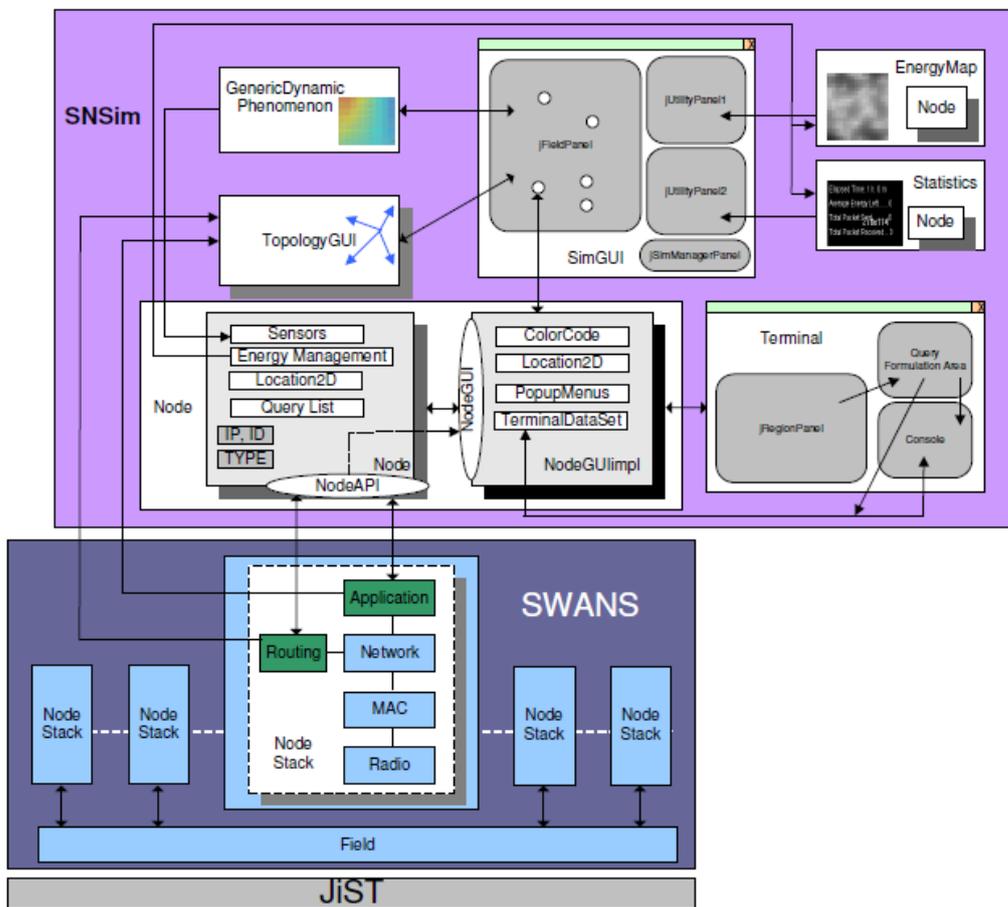
b. *Routing Algorithm*

Algoritma *routing* mengimplementasikan konstruksi jalur antara *node sink* dan *node* sumber.

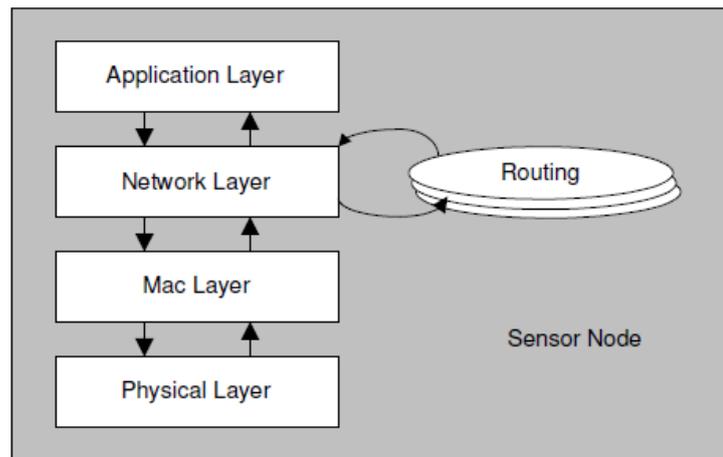
c. *Driver*

Driver mengikat semua informasi secara bersamaan untuk bisa mengakses aplikasi SIDnet. Dalam sebuah *driver* bisa ditentukan *stack* jaringan yang diinginkan, algoritma pada setiap lapisan, fenomena yang akan diukur, penempatan *node* diarea yang diinginkan dll.

Arsitektur SIDnet terdiri dari tiga komponen yaitu : Struktur kode, *Sensor Field GUI* dan *Internal Operational Architecture*. *Sensor Field GUI* mewakili wadah dimana *node –node* akan ditempatkan. Setiap *node* merupakan abstraksi inti dari simulator SIDnet, *node* mewakili antarmuka antara *stack* jaringan dan semua komponen simulator lainnya termasuk GUI, area pemindaian, layanan lokasi, manajemen energi dan lain sebagainya. Hubungan antara *stack* jaringan, GUI dan komponen lainnya terlihat pada gambar 2.7. Arsitektur Operasional SIDnet SWANS (Ghica, 2010) yang tergabung dala arsitektur operasional SIDnet. Kemudian untuk *stack* jaringan pada SIDnet terlihat pada gambar 2.8. Stack Jaringan SIDnet SWANS.



Gambar 2 7. Arsitektur Operasional SIDnet SWANS (Ghica, 2010)



Gambar 2 8. Stack Jaringan SIDnet SWANS (Ghica, 2010)

2.7. *Heartbeat protocol*

Heartbeat adalah protokol yang menyediakan mekanisme pertukaran pesan *beat* antar masing-masing komponen jaringan secara periodik. Selama proses p masih menerima pesan beat dari proses q, maka proses p mengenal proses q dan media komunikasi antara kedua proses masih berjalan dengan baik. Jika p tidak lagi menerima pesan beat dari q, maka p akan mengidentifikasi bahwa q telah terputus atau telah diterminasi, dan komunikasi di antara keduanya juga terputus (Gouda & McGuire, 1998). Konsep *heartbeat* diimplementasikan secara luas, dan dikembangkan menjadi berbagai protokol untuk memenuhi kebutuhan masing-masing jenis komunikasi.

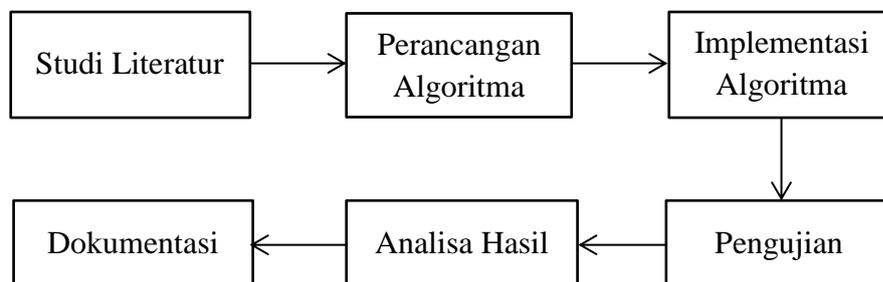
[Halaman ini sengaja dikosongkan]

BAB 3

METODE PENELITIAN

3.1. Tahapan Penelitian

Secara umum penelitian ini membutuhkan beberapa tahapan agar penelitian ini lebih terarah dan tercapainya tujuan yang diharapkan, tahapannya terlihat pada gambar 3.1 Diagram Alur Penelitian. Tahapan awal yang akan dilakukan adalah studi literatur yang digunakan untuk mempelajari masalah dan menemukan referensi yang berkaitan dengan penelitian ini, selanjutnya perancangan algoritma yang merupakan lanjutan dari studi literatur, pada tahapan ini akan merancang algoritma untuk menyelesaikan permasalahan yang ditemukan, dalam hal ini untuk meningkatkan kinerja *Clustered Shortest Geopath Routing Protocol* dengan *adaptive path* dan agregasi data. Algoritma tersebut akan diimplementasikan menggunakan Simulator SIDNet SWANS dengan melakukan pemilihan dan pergantian *cluster head* selanjutnya melakukan agregasi data di *cluster head* berdasarkan prioritas data. Pengujian dilakukan dengan variasi parameter pemilihan *cluster head*, pergantian *cluster head*, proses agregasi data, perbandingan dengan metode lain, variasi jumlah *node* dan luas area yang berbeda – beda. Kemudian hasil pengujian akan dianalisa dengan melakukan perbandingan dengan beberapa metode lain yang relevan dengan penelitian ini untuk selanjutnya didokumentasikan dalam bentuk tulisan yang berupa buku tesis.



Gambar 3.1. Diagram Alur Penelitian

3.2. Studi Literatur

Studi literatur merupakan tahapan awal untuk melakukan kajian yang berkaitan dengan topik penelitian dengan mempelajari masalah dan menemukan referensi yang relevan. Referensi yang digunakan dalam penelitian ini berasal dari jurnal, konferensi dan buku yang berkaitan dengan algoritma *Clustered Shortest Geopath Routing Protocol* pada lingkungan *Wireless Sensor Network*. Berdasarkan studi literatur yang telah dilakukan, diperoleh informasi sebagai berikut :

1. Karakteristik dan Keterbatasan WSN.
2. Perkembangan metode yang berbasis *cluster* pada WSN.
3. CSGP merupakan salah satu pengembangan metode *routing protocol* berbasis *cluster* dan lokasi.
4. Pada metode *cluster* data sensing dikirim ke *cluster head* baru kemudian dikirim ke *sink*.
5. Untuk mengurangi traffic jaringan bisa dilakukan dengan agregasi data dan prioritas data
6. Pemilihan jalur pada WSN bisa dilakukan dengan *routing protocol* tertentu.

3.3. Perancangan Algoritma

Perancangan algoritma dengan melakukan pengembangan terhadap algoritma *Clustered Shortest Geopath Routing Protocol* dengan pemilihan *adaptive path* dan agregasi data. Setiap *node* akan di sebarakan secara acak pada area tertentu, *sink* akan membagi wilayah menjadi sejumlah *k cluster* sesuai dengan luas area sensing dan rata- rata jumlah jangkauan *node sensor* dan akan menyebarkan pesan *query* yang berisi informasi interval *sensing* dan durasi, selanjutnya *Node sensor* akan menjalankan *heartbeat protocol*. Setiap *node sensor* akan melakukan sensing sesuai dengan pesan *query* yang telah diterima dan mengirim hasil sensing ke *cluster head*.

Pada setiap *cluster* akan dipilih *cluster head* berdasarkan *node* yang memiliki jumlah tetangga terbanyak dalam satu *cluster* dan akan dilakukan *maintenance* atau pergantian *cluster head* jika sisa energi *cluster head* terpilih telah mencapai 50%. *Cluster head* adalah *node* yang bertugas menampung sementara data hasil sensing dan kemudian meneruskan pesan tersebut ke *sink*.

Selanjutnya data *sensing* yang diperoleh akan dikirim ke *cluster head* yang terpilih, ketika data sensing sampai di *cluster head* akan dilakukan pengecekan status dari pesan tersebut apakah sudah pernah melewati *cluster head* atau belum, jika belum melewati *cluster head* maka akan dilakukan pengecekan tingkat prioritas data, jika prioritas data tinggi atau *fire* maka data akan langsung diteruskan ke *sink* menggunakan jalur terpendek ke *sink*, sedangkan data dengan prioritas sedang dan rendah maka akan dilakukan agregasi dengan cara data yang sampai di *cluster head* akan ditampung sementara sehingga batasan tertentu yang kemudian akan di rata-rata dan hasil rata-rata tersebut akan di tandai sebagai pesan yang telah melewati *cluster head* dan selanjutnya akan dikirim ke *sink*. Namun Jika status telah melewati *cluster head* maka pesan akan diteruskan *node* selanjutnya yang memiliki jarak terdekat ke *sink* untuk selanjutnya diteruskan ke *sink*.

Pada penelitian ini digunakan beberapa istilah, yaitu:

1. *Message* : data *sensor* yang dihasilkan oleh *source node* yang merupakan hasil pemindaian terhadap lingkungan.
2. *Message Space* : merupakan variabel yang menampung *message* sebelum dikirim ke *cluster head*.
3. *Received Message* : merupakan variabel yang menampung *message* yang dikirim dari *Node source* sebelum dikirim ke *sink*.
4. *Cluster head* : merupakan *node* yang bertugas menampung sementara dan meneruskan data dari *cluster member* ke *sink node*.
5. *Cluster Member* : merupakan *node* yang yang tidak terpilih menjadi *cluster head* pada suatu *cluster*.
6. *Cluster* : merupakan suatu area tertentu yang terdiri dari kumpulan beberapa member *node* yang memiliki *cluster head* yang sama.
7. *Sensing Query* : permintaan yang dibuat oleh *sink* yang ditujukan ke semua *node* untuk melakukan pemindaian dengan interval waktu tertentu dan hasil pemindaian tersebut akan dikirim ke *sink node*.
8. *Prioritas Data*: merupakan informasi yang terdapat pada *message* mengenai kondisi lingkungan tersebut.

3.3.1. Menjalankan *Heartbeat Protocol*

Node sensor akan menjalankan *heartbeat protocol* yang telah dimodifikasi dengan menambahkan sejumlah informasi *cluster* seperti *cluster id*, jarak *node* ke *cluster center*, sisa energi dan jumlah tetangga. *Node* akan bertukar pesan *beat* secara periodik untuk saling menemukan tetangga dan koordinatnya. Informasi yang dikumpulkan berupa jarak *euclidean* antar satu *node* dengan *node* yang lainnya, *cluster id*, jarak *node* ke *cluster center*, sisa energi dan jumlah tetangga. Jarak *euclidean* merupakan panjang garis yang menghubungkan dua titik pada suatu bidang koordinat. Jika koordinat yang digunakan merupakan bidang dua dimensi, maka jarak *euclidean* antara dua *node* $p(p_1, p_2)$ dan $q(q_1, q_2)$ dapat dihitung menggunakan persamaan :

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} \quad (3.1)$$

Dimana :

$d(p, q)$ = jarak antara *node* p dan *node* q

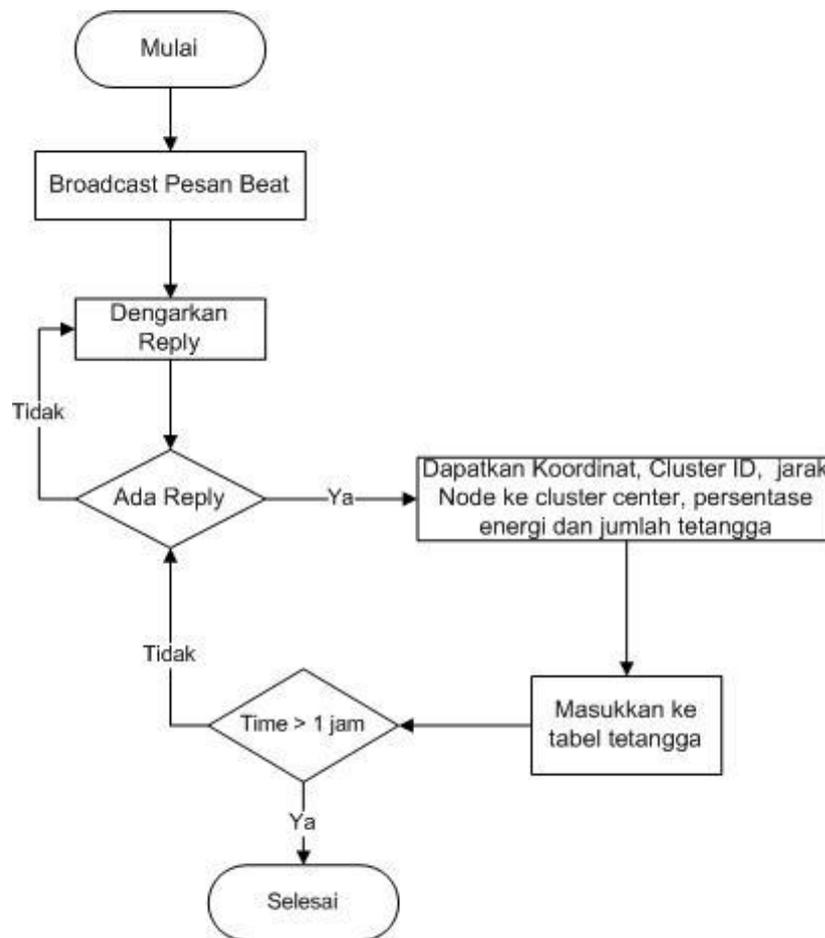
p_1 = posisi p pada sumbu X

p_2 = posisi p pada sumbu Y

q_1 = posisi q pada sumbu X

q_2 = posisi q pada sumbu Y

Jika *node* merupakan tetangga dari sebuah *node*, maka informasi mengenai *node* tersebut akan dimasukkan ke dalam sebuah tabel tetangga (*neighbour list/node entry*). Setelah semua tetangga dan lokasinya ditemukan maka *heartbeat protocol* hanya akan dipanggil pada saat dibutuhkan. Diagram alir *heartbeat protocol* divisualisasikan di Gambar 3.2.



Gambar 3. 2. Diagram alur *Heartbeat protocol*

3.3.2. Pemilihan dan Pergantian *Cluster Head*

Cluster head merupakan *node* sensing terpilih yang bertugas meneruskan *message* dari *cluster member* menuju *sink*. *Cluster head* diperlukan agar bisa mengurangi jumlah transmisi jaringan, karena dengan adanya *cluster head* maka pengiriman *message* akan lebih terpusat dimana *cluster member* atau *node sensor* tidak perlu meneruskan *message* secara langsung ke *sink* yang bisa menyebabkan padat nya transmisi di jaringan. Hasil sensing dari *node sensor* akan dikirim ke *cluster head* terpilih dan ditampung di sementara *received message*, kemudian data di *received message* akan di agregasi sehingga menghasilkan data baru dan data hasil agregasi tersebut yang akan diteruskan ke *sink* melalui jalur terpendek menuju *sink*.

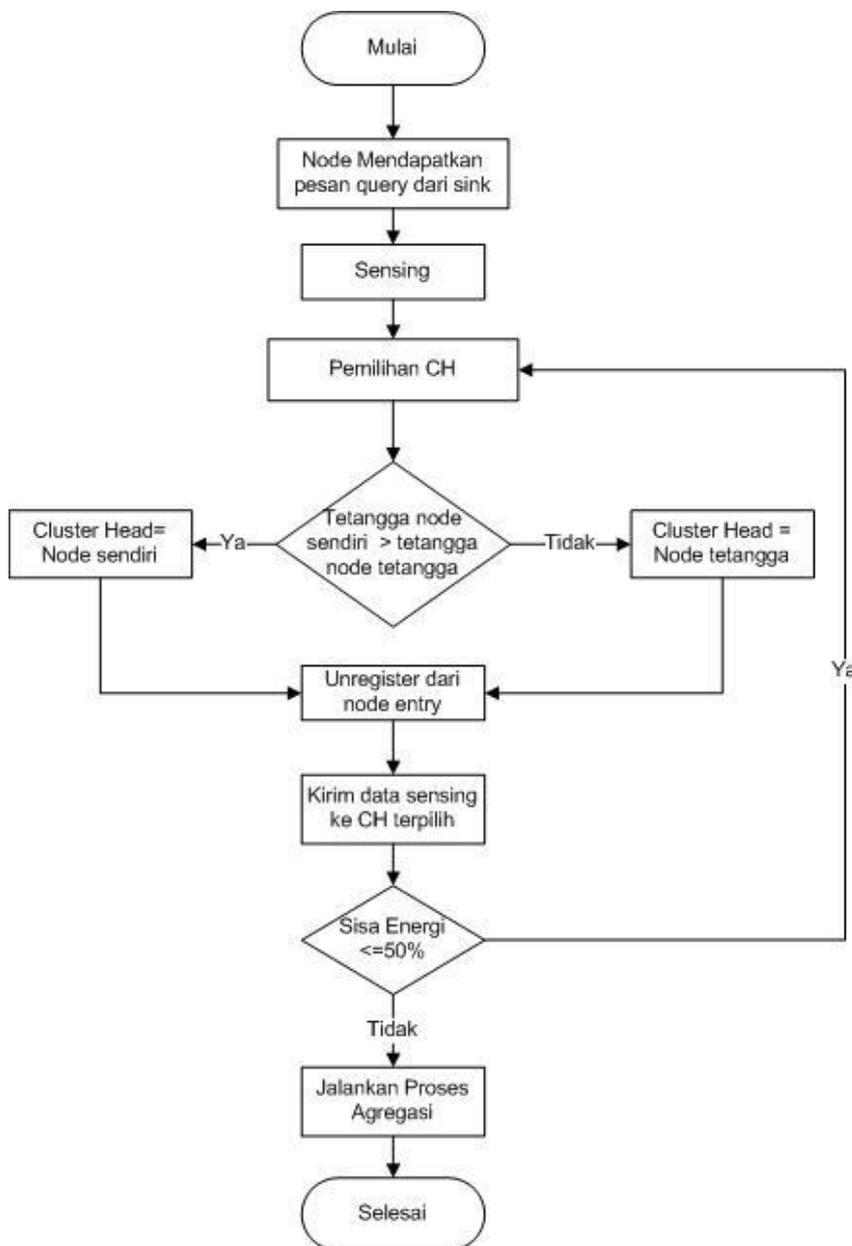
Untuk pemilihan cluster head terdiri dari beberapa parameter yaitu berdasarkan sisa energi tertinggi, jarak terdekat ke cluster center, jumlah tetangga terbanyak dan kombinasi antara sisa energi tertinggi dan jarak terdekat ke cluster center. Kombinasi yang digunakan adalah 0.7 sisa energi tertinggi dan 0.3 cluster center, 0.5 sisa energi tertinggi dan 0.5 cluster center, 0.3 sisa energi tertinggi dan 0.7 cluster center, nilai ini diperoleh setelah dilakukan pra pengujian terhadap parameter pemilihan cluster head kombinasi dengan interval 0.0 sampai 1.0 dan diperoleh hasil yang optimal di nilai 0.7, 0.5 dan 0.3. untuk lebih jelas data hasil uji coba terlihat pada lampiran 1.

Setiap *node* akan melakukan sensing sesuai dengan pesan *query* yang diterima dari *sink* dan hasil sensing akan dikirim ke *cluster head*. Pemilihan *cluster head* dilakukan pertama kali setelah pesan *heartbeat* selesai dijalankan, parameter pemilihan *cluster head* berdasarkan *node* yang memiliki jumlah tetangga terbanyak. Setiap *node* akan mengetahui jumlah tetangga nya sendiri dan jumlah tetangga dari masing – masing tetangganya yang berada dalam *cluster* yang sama dengan *node* tersebut ketika pesan *heartbeat* di sebar, jika jumlah tetangga *node* tersebut lebih banyak dari jumlah tetangga masing-masing tetangganya maka *node* tersebut yang akan menjadi *cluster head*, akan tetapi jika jumlah tetangga salah satu *node* tetangganya lebih banyak dari pada *node* tersebut maka *node* tetangganya itulah yang akan menjadi *cluster head*.

Node yang terpilih menjadi *cluster head* akan di unregister dari *node entry* atau list node kandidat CH dengan artian *node* tersebut akan diabaikan pada pemilihan *cluster head* selanjutnya. *Node* akan mengirim paket data ke cluster head terpilih, kemudian setiap *node* akan memeriksa apakah sisa energi node $\leq 50\%$, jika ya maka akan dipilih cluster head baru berdasarkan jumlah tetangga terbanyak kedua dan jika tidak maka data akan dilakukan proses agregasi data.

Kedua tahapan diatas dilakukan untuk menghemat energi *node* yang menjadi *cluster head* karena suatu *node* menjadi *cluster head* maka konsumsi energinya akan lebih besar dari *node sensor* biasa sehingga dengan dilakukan pergantian atau pemilihan ulang *cluster head* konsumsi energi tiap *node* akan lebih merata. Disamping itu dengan pergantian *cluster head* maka jalur pengiriman akan berubah secara adaptative sesuai dengan lokasi *cluster head* yang terpilih sehingga

dapat menghemat konsumsi energi *node* yang berada jalur yang terpilih. Untuk lebih memperjelas proses seleksi *cluster head* selanjutnya akan divisualisasikan pada gambar diagram alur 3.3. berikut ini.



Gambar 3. 3. Diagram Alur Pemilihan dan Pergantian *Cluster Head*

3.3.3. Prioritas dan Agregasi Data

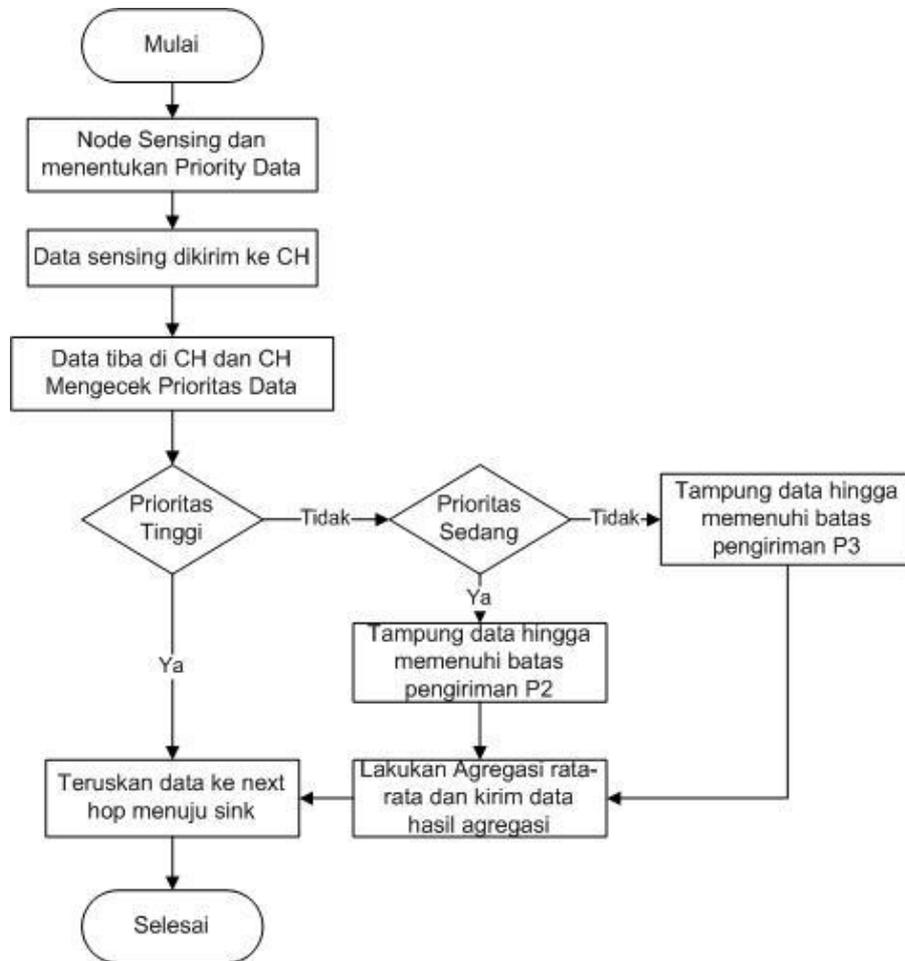
Untuk mengurangi jumlah transmisi di jaringan dapat dilakukan dengan melakukan agregasi di *cluster head* sehingga dengan berkurangnya jumlah transmisi akan meningkatkan usia jaringan tersebut. Untuk skenario jaringan yang

memindai data-data seperti temperatur, kelembaban atau data-data lain yang ada kalanya tidak perlu dikirim setiap saat, maka dapat digunakan teknik agregasi data untuk kemudian dikirim secara adaptif sesuai dengan prioritas dari data yang terpindai. Pada penelitian ini ujicoba akan dilakukan pada skenario kebakaran dengan data *sensing* berbentuk temperatur atau suhu. Setelah informasi tetangga dan *sink* tersimpan di tabel tetangga, selanjutnya *node sensor* akan melakukan *sensing* secara periodik sesuai dengan interval *sampling* yang telah ditentukan, hasil *sensing* akan di kelompokkan kedalam beberapa prioritas sesuai dengan range nilai tertentu yaitu prioritas tinggi, prioritas sedang dan prioritas rendah. Prioritas yang dimaksud adalah mengenali kondisi suatu area dimana jika kondisi kebakaran atau *fire* maka dianggap sebagai prioritas tinggi dan diperlukan penanganan yang lebih cepat dibandingkan prioritas yang lainnya. Prioritas sedang dianggap sebagai kondisi abnormal yaitu waspada dan prioritas rendah dianggap kondisi normal. Adapun prioritas datanya sesuai dengan suhu atau temperatur seperti dibawah ini:

- a. Prioritas Tinggi (P1): temperatur hasil pemindaian yang nilainya berkisar antara 44 sampai dengan 110 derajat *celcius*.
- b. Prioritas Sedang(P2): temperatur hasil pemindaian yang nilainya berkisar antara 33 sampai dengan 43 derajat *celcius*.
- c. Prioritas Rendah(P3): temperatur hasil pemindaian yang nilainya berkisar antara 18 sampai dengan 32 derajat *celcius*.

Setelah prioritas data dari suatu data *sensing* diketahui maka data tersebut akan di agregasi oleh *node sensing* dan memerikan label prioritas yang selanjutnya data tersebut akan dikirim ke *cluster head*. Data yang masuk di *cluster head* selanjutnya akan dikirim ke *sink* berdasarkan prioritas datanya menggunakan protokol SGP. Jika data yang masuk merupakan data dengan prioritas tinggi maka data tersebut akan langsung dikirim ke *sink* menggunakan jalur terpendek menuju *sink*. Sedangkan jika data yang masuk dengan prioritas sedang atau rendah akan ditampung sementara di *cluster head* hingga mencapai batasan tertentu, data prioritas sedang akan ditampung hingga *received message* sama dengan 10 data dan data dengan prioritas rendah akan ditampung hingga *received message* sama dengan 20 data. Setelah data prioritas sedang atau rendah mencapai batasan masing

- masing maka akan dilakukan agregasi data dengan melakukan rata-rata terhadap semua data yang telah ditampung dan kemudian hanya data hasil agregasi tersebut yang akan dikirim ke *sink* menggunakan jalur terpendek menuju *sink*.



Gambar 3. 4. Diagram Alur Prioritas dan Agregasi Data

3.4. Pengujian Algoritma

Pengujian algoritma akan dilakukan menggunakan Simulator SIDnet-SWANS dengan dengan menggunakan beberapa skenario ujicoba. Hasil evaluasi nantinya akan dibandingkan dengan beberapa metode yang lain untuk mengukur kinerja performa dari protokol *Clustered-Shortest Geopath Routing Protocol with adaptive path and data aggregation* dengan parameter uji coba yang sama. Selain itu evaluasi juga dilakukan dengan membandingkan dengan beberapa parameter uji coba yang berbeda.

3.4.1. Lingkungan pengujian

Dalam proses pengujian, algoritma yang dikembangkan diuji pada sebuah komputer. Adapun spesifikasi dari komputer yang digunakan adalah sebagai berikut:

- a. Processor Intel Core i3-3110M, 2.4 GHz
- b. RAM 4 GB
- c. Harddisk 500 GB
- d. Sistem Operasi Windows 7 Ultimate x86
- e. Java Development Kit 5 dan 8

3.4.2. Parameter Uji

Parameter Uji yang akan digunakan pada *Clustered shortest geopath routing protocol with adaptive path and aggregation* untuk mengetahui kinerja protokol ini bersifat tetap untuk semua skenario. Parameter uji ini diimplementasikan langsung pada *source code* simulator. Adapun parameter uji digunakan terlihat pada tabel 3.1.

Tabel 3. 1. Parameter Uji

No	Keterangan		Detail
1	Radio	Bandwidth (bps)	40000
		Transmit (dBm)	-12
2	Energy Consumption Parameters	ProcessCurrentDrawn_ActiveMode[mA]	8
		ProcessCurrentDrawn_SleepMode[mA]	0.015
		RadioCurrentDrawn_TransmitMode[mA]	27
		RadioCurrentDrawn_ReceiveMode[mA]	10
		RadioCurrentDrawn_ListenMode[mA]	3
		RadioCurrentDrawn_SleepMode[mA]	0.5
		SensorCurrentDrawn_ActiveMode[mA]	10
		SensorCurrentDrawn_PassiveMode[mA]	0.01
3	Node Source	Placement	Random
		Mobility	Static
		Sensor	GPS dan Temperature
		Battery Capacity (mAh)	40
		Battery Voltage (volt)	3
4	Node Sink	Placement	Grid (1,1)
		Battery Capacity (mAh)	80
		Battery Voltage (volt)	3
5	Sensing Query	Duration(hour)	60
		Sampling Interval(second)	3

6	Prioritas Data	Range temperatur Prioritas Tinggi (Kebakaran)	44 ^{°C} - 100 ^{°C}
		Range temperatur Prioritas Sedang (Abnormal)	33 ^{°C} - 43 ^{°C}
		Range temperatur Prioritas Rendah (Normal)	18 ^{°C} - 32 ^{°C}
7	Agregasi Data	Received Message Prioritas sedang (P2)	10
		Received Message Prioritas rendah (P3)	20
9	Event Timeline	Menit ke 90-120	<i>Cluster 1, 10 & 30 = suhu abnormal</i>
		Menit ke 120-200	<i>Cluster 1, 10 & 30 = kebakaran</i>
		Menit ke 200-250	<i>Cluster 1, 10 & 30 = suhu abnormal</i>
		Menit ke 900-950	<i>Cluster 5, 18 & 25 = suhu abnormal</i>
		Menit ke 950-1100	<i>Cluster 5, 18 & 25 = kebakaran</i>
		Menit ke 1100-1150	<i>Cluster 5, 18 & 25 = suhu abnormal</i>
		Menit ke 1800-1850	<i>Cluster 8 = suhu abnormal</i>
		Menit ke 1850-2000	<i>Cluster 8 = kebakaran</i>
		Menit ke 2000-2100	<i>Cluster 8 = suhu abnormal</i>

Keterangan :

Radio merupakan parameter pengujian pada simulator yang merupakan keterangan dari perangkat nirkabel yang digunakan pada masing-masing *node*. Parameternya meliputi kecepatan transfer dari perangkat, dan sensitivitas jarak pengiriman data.

Energy consumption Parameters merupakan nilai konsumsi untuk masing – masing *node* ketika *node* berada pada mode tertentu. mode tersebut tergantung dari

kondisi yang sedang dilakukan yaitu proses, radio dan *sensor*. untuk masing – masing kondisi memiliki nilai konsumsi energi masing- masing. Kondisi yang konsumsi energi paling besar terjadi pada kondisi radio mode transmit atau kondisi *node* tersebut melakukan pengiriman data yaitu 27 [mA] dan kondisi konsumsi energi paling kecil terjadi pada kondisi proses mode sleep yaitu 0.015[mA].

Node Source merupakan *node* yang melakukan proses sensing pada suatu area berdasarkan pesan query yang diterima di *sink*. Setiap *node source* memiliki kemampuan untuk melakukan sensing berdasarkan *sensor* temperatur dan mengetahui lokasi *node* itu sendiri dan tetangga berdasarkan *sensor* GPS. Setiap *node* di set memiliki battery dengan kapasitas 40 mAh dengan voltase 3 volt. *Node* – *node* ini akan ditempatkan secara *random* pada wilayah yang sudah ditentukan sebelumnya dan mobilitynya bersifat static atau tetap.

Node Sink merupakan *node* yang menyebarkan informasi atau pesan query untuk kebutuhan sensing dan juga yang menerima serta menampung pesan hasil sensing yang telah dilakukan oleh *node source*. *Sink* di set memiliki *battery* dengan kapasitas dua kali lebih besar dibandingkan *node source* yaitu 80 mAh dan voltase 3 volt. Hal ini dilakukan karena *sink* harus memiliki waktu hidup yang lebih lama dibandingkan *node source* agar tetap bisa menerima pesan hasil sensing yang dikirim oleh *node source*. *Node sink* di tempatkan secara static pada lokasi 1,1 dari wilayah dan *node* ini merupakan total keseuruhan *node sensor-1*.

Sensing Query merupakan perintah untuk pemantauan lingkungan. *Sampling* interval adalah interval waktu dimana *node* melakukan pemindaian data secara periodik. *Sampling* interval ditentukan oleh admin jaringan melalui GUI interaktif pada simulator. Pengambilan nilai dari *sensor* suhu dilakukan berdasarkan interval waktu tertentu yaitu per 3 detik dengan durasi pemantauan yaitu 60 jam.

Prioritas data merupakan kategori untuk data hasil sensing yang dilakukan oleh *node sensor*, dipengujian ini terdapat 3 jenis prioritas data yaitu prioritas tinggi dengan rentang suhu 44°C - 100°C yang menandakan bahwa data ini merupakan data kebakaran dan perlu segera ditangani, selanjutnya prioritas sedang dengan rentang suhu 33°C - 43°C yang merupakan data abnormal atau kondisi waspada, dan yang terakhir prioritas rendah dengan rentang suhu 18°C - 32°C yang merupakan data normal.

Agregasi data merupakan suatu proses mengumpulkan data sensing yang yang dikirim dari *node source* dan dikumpulkan di *cluster head*. Proses ini terjadi di *cluster head* berdasarkan prioritas data yang masuk dan hanya berlaku untuk data prioritas sedang dan rendah, untuk data prioritas sedang akan ditampung di *received message* hingga size nya sama dengan 10 dan data dengan prioritas rendah akan ditampung di *received message* hingga size nya sama dengan 20.

Received Message adalah struktur data yang berupa list untuk menampung *message* dari *source node* yang akan diteruskan ke *sink*. batas maksimal *received message* bersifat *adaptive* tergantung dari prioritas data *message*. Semakin tinggi prioritas data, maka akan semakin kecil *received message* sehingga *message* akan lebih cepat sampai di *sink* dan *event* yang terjadi akan lebih cepat di deteksi oleh *sink*.

3.5. Analisa Pengujian

Pengujian sistem yang dilakukan untuk mengetahui kinerja dari pengembangan protokol *Clustered-Shortest Geopath Routing Protocol with Adaptive Path and aggregation* yaitu meningkatkan kinerja dari *wireless sensor network*. Kinerja yang diuji dan dianalisa berdasarkan parameter tertentu yaitu *network lifetime*, *Event detection*, *packet delivery ratio* dan *Latency*. Berikut adalah penjelasan dari masing – masing parameter analisis pengujian:

a. *Network lifetime*

Network lifetime adalah rentang waktu mulai *node* disebarkan hingga *node* pertama mati.

b. *Event Detection*

Perbandingan antara jumlah *event* yang terdeteksi oleh *node sensor* dengan jumlah *event* yang diterima oleh *sink*. *Event* yang dimaksud adalah kondisi kebakaran atau data prioritas tinggi (P1) dengan sensed value 44°C - 100°C . Semakin banyak *event* yang terdeteksi maka semakin baik.

c. *Packet Delivery Ratio*

Packet Delivery Ratio (PDR) adalah Perbandingan antara total jumlah paket yang diterima oleh sink dengan total jumlah paket yang dikirim oleh *node source* yaitu *cluster head*. Semakin besar nilai PDR, maka semakin baik nilainya. Untuk bisa mengetahui nilai dari *packet delivery ratio* dihitung berdasarkan persamaan 3.2 berikut:

$$PDR = \frac{\sum \text{Paket yang diterima}}{\sum \text{Paket yang dikirim}} \% \quad (3.2)$$

d. *Latency*

Latency adalah rentang waktu yang diperlukan mulai paket dikirim hingga paket diterima oleh sink. Semakin kecil nilai dari parameter analisis pengujian ini menunjukkan bahwa kinerja performa protokol yang baik.

3.6. Evaluasi Kinerja

Evaluasi kinerja sistem ini dilakukan dengan cara mengamati dan menganalisa data hasil penelitian yang diperoleh dari menjalankan sistem dengan simulator SIDnet-SWANS. Evaluasi yang dilakukan dengan merancang skenario sistem pada kondisi jaringan yang berbeda – beda. Berikut rancangan skenario yang akan dievaluasi.

- a. Skenario 1 dengan melakukan pemilihan *cluster head* yang optimal berdasarkan beberapa parameter yaitu:
- Berdasarkan jarak terdekat ke *cluster center*.
 - Berdasarkan sisa energi tertinggi.
 - Berdasarkan jumlah tetangga terbanyak.
 - Berdasarkan 0.5 sisa energi tertinggi dan 0.5 jarak terdekat ke *cluster center*.
 - Berdasarkan 0.7 sisa energi tertinggi dan 0.3 jarak terdekat ke *cluster center*.
 - Berdasarkan 0.3 sisa energi tertinggi dan 0.7 jarak terdekat ke *cluster center*.

b. Skenario 2 variasi pergantian *cluster head*

Pada skenario ini akan dilakukan evaluasi terhadap kondisi jaringan tanpa pergantian *cluster head*, satu kali pergantian CH ketika energi $CH \leq 50\%$, dua kali pergantian CH ketika energi $CH < 75\%$ dan 50% dan tiga kali pergantian CH ketika energi $CH < 80\%$, $< 60\%$ dan $< 40\%$.

c. Skenario 3 variasi jumlah data yang ditampung untuk di agregasi

Pada skenario ini akan dilakukan evaluasi terhadap jumlah data yang akan di agregasi tergantung dari tingkat prioritas datanya, yaitu prioritas sedang (P2) dan prioritas rendah (P3) :

- P2=2 data, P3=5 data
- P2=5 data, P3=10 data
- P2=10 data, P3=20 data
- P2=30 data, P3=50 data
- P2=50 data, P3=100 data
- P2=100 data, P3=200 data

d. Skenario 4 membandingkan metode usulan dengan metode lainnya dengan luas area 400x400m dan jumlah *node* 200 *node*, adapun metode pembandingnya adalah: *Clustered-Shortest Geopath Routing Protocol without data aggregation(CSGP-WA)*, *Clustered-Shortest Geopath Routing Protocol with random cluster head (CSGP-RCH)* dan *Shortest Geopath Routing Protocol without adaptive path and aggregation (SGP-WAPA)*.

e. Skenario 5 dengan melakukan variasi luas area ujicoba dengan jumlah *node* yang sama yaitu 100: 100x100m, 200x200m, 300x300m, 400x400m dan 500x500m.

f. Skenario 6 dengan melakukan variasi jumlah *node* pada suatu luas area tertentu yaitu 300, variasi *node* nya yaitu: 100 *node*, 200 *node* dan 300 *node*.

[halaman ini sengaja dikosongkan]

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Tahapan Implementasi Metode

Pada tahapan ini akan dilakukan implementasi metode dengan menggunakan simulator yaitu SIDnet-SWANS. Implementasi yang dilakukan dengan memperbaharui modul-modul yang ada pada simulator SIDnet SWANS, modifikasi dilakukan pada modul app, modul *routing*, modul driver dan modul *heartbeat*. Tahapan implementasi metode penelitian yang digunakan adalah sebagai berikut:

1. Tahapan pertama adalah melakukan perancangan terhadap protokol pengiriman untuk *wireless sensor network* dengan pengembangan metode *Clustered shortest geopath routing protocol* dengan menggunakan *adaptive path* dan agregasi data guna meningkatkan kinerja *wireless sensor network*. Pada tahapan ini akan dihasilkan protokol pengiriman yang disebut dengan *clustered shortest geopath routing protocol with adaptive path and aggregation (CSGP-APA)* dan akan diimplementasikan dengan menggunakan simulator berbasis java yaitu SIDnet SWANS.
2. Tahapan kedua adalah perancangan skenario pengujian yang akan dilakukan untuk menguji kinerja *clustered shortest geopath routing protocol with adaptive path and aggregation* yang telah dihasilkan pada tahap pertama. Skenario yang dirancang meliputi variasi parameter pemilihan *cluster head*, variasi parameter pergantian *cluster head*, variasi jumlah data untuk proses agregasi data, variasi luas area dan variasi jumlah *node* pada wilayah yang amati. Pada tahapan kedua ini akan dihasilkan parameter yang akan digunakan pada tahapan ketiga yaitu pengujian protokol yang telah dirancang.
3. Tahapan ketiga adalah tahapan pengujian terhadap *clustered shortest geopath routing protocol with adaptive path and aggregation* berdasarkan parameter yang telah dirancang sebelumnya. Pada tahapan ini akan dihasilkan data hasil pengujian protokol yang telah di uji dalam bentuk file teks.

4. Tahapan keempat adalah tahapan terakhir yaitu analisa hasil yang diperoleh pada tahapan sebelumnya, data hasil pengujian yang telah dilakukan akan disimpan dalam bentuk file teks yang kemudian akan dianalisa sesuai dengan variasi skenario yang telah dirancang sebelumnya. Tahapan ini dilakukan untuk mengetahui kinerja dari protokol yang telah diuji dan memastikan berjalan sesuai dengan yang dirancang tanpa adanya kesalahan pada proses ujicoba.

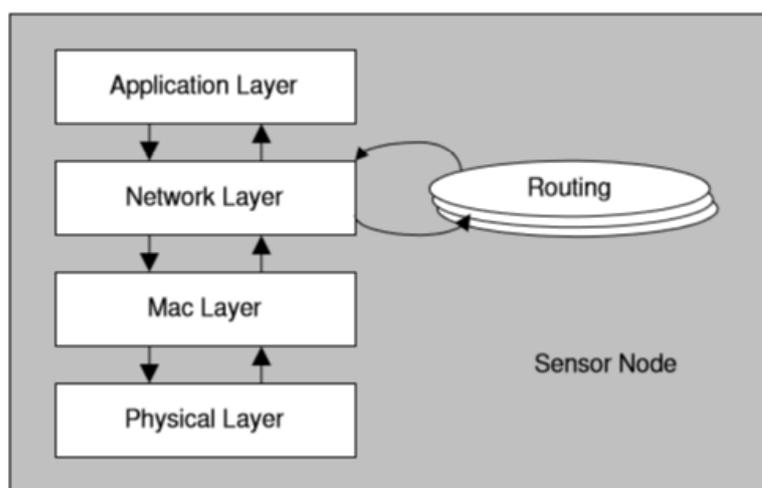
Semua tahapan yang telah dijelaskan sebelumnya akan diimplementasikan menggunakan simulator SIDnet SWANS dengan melakukan modifikasi pada beberapa modul-modul yang terdapat pada simulator. Modifikasi yang dilakukan pada modul app, modul driver, modul *routing* dan *heartbeat*. Modul-modul yang dimodifikasi berada pada *application layer* dan *network layer*, dimana kedua *layer* ini saling terhubung untuk bisa menyelesaikan suatu tugas. Hubungan antara kedua *layer* ini terlihat pada gambar 4.1. Modul *Layer* pada Simulator SIDnet SWANS (Ghica, 2010).

Application layer merepresentasikan setiap *node sensor* akan melakukan sensing sesuai dengan kriteria yang telah ditentukan oleh *sink* melalui pesan *heartbeat* dan mendapatkan data hasil *sensing* yang kemudian akan dikirim ke *sink*, untuk dapat mengirimkan data tersebut setiap *node* harus mengirimkan data menggunakan protokol tertentu. Proses pengumpulan informasi tetangga dan mendapatkan hasil sensing terjadi pada *application layer* dan kemudian data dan informasi tersebut akan dikirim ke *network layer* menggunakan *routing* tertentu. *Network Layer* merepresentasikan sebuah mekanisme *switchboard* antara paket yang datang dari *Application layer*, *Mac Layer*, dan paradigma *routing*. *Network Layer* meneruskan pesan berdasarkan alamat destinasi dari pesan tersebut.

Pada *network layer* akan memeriksa setiap pesan yang diterima apakah alamat dari pesan tersebut merepresentasikan tujuan akhir dari pesan tersebut. Jika tidak maka *network layer* akan meminta *routing protocol* untuk menentukan hop selanjutnya dan meneruskan pesan tersebut, akan tetapi jika alamat dari pesan tersebut merepresentasikan *node* tujuan akhir dari pesan tersebut maka *network layer* akan menangani pesan tersebut menggunakan method *receive* dan selanjutnya akan meneruskan pesan ke *application layer*.

Untuk bisa mengimplementasikan *clustered shortest geopath routing protocol with adaptive path and aggregation* perlu dilakukan modifikasi pada level *application*

layer dan *network layer*, modifikasi yang dilakukan dengan cara menggandakan dan memodifikasi modul yang sudah ada ke dalam package `sidnet.stack.users.csqp_adaptivepath.driver`, `sidnet.stack.users.csqp_adaptivepath.routing`, `sidnet.stack.users.csqp_adaptivepath.app` dan `sidnet.stack.std.routing.heartbeat`.



Gambar 4. 1. Modul *Layer* pada Simulator SIDnet SWANS(Ghica, 2010)

4.1.1. Modifikasi *Application layer*

Pada *application layer* modifikasi dilakukan pada *class* `app`, yaitu dengan menggandakan modul *application layer* yang telah ada secara *default* dari package `users.java.sidnet.stack.users.sample_p2p.app` dengan nama *class* `AppSampleP2P.java`, `MessageDataValue.java`, dan `MessageQuery.java` ke dalam `sidnet.stack.users.csqp_adaptivepath.app` dengan nama *class* `CSGPAP_App.java`, `MessageDataValue.java`, dan `MessageQuery.java`.

Pada kelas ini modifikasi dilakukan pada proses pengumpulan informasi tetangga dan pemindaian yaitu pada line 166 hingga 185. Proses pengumpulan informasi tetangga (`MessageHeartbeat`) ditambahkan dengan informasi sisa energi *node* dan jumlah tetangga dari *node* tetangga. Proses ini berfungsi untuk menangani proses pertukaran informasi untuk setiap tetangga *node* pada heartbeat protocol.

Gambar 4.2. merupakan hasil Modifikasi pada *application layer*

```

1  if (JistAPI.getTime() > 0 && !heartbeatInitiated){
...
4      MessageHeartbeat msg = new MessageHeartbeat();
5      msg.setNCS_Location(myNode.getNCS_Location2D());
6      msg.setClusterID(myNode.clusterId);
7      msg.setDistToCluster(myNode.distToClusterCenter);
8      msg.setBattery(myNode.getEnergyManagement().getBattery().getPer
centageEnergyLevel());
9      msg.setJumlahTetangga(myNode.neighboursList.size());
10     netEntity.send(msg,    NetAddress.ANY,    Constants    .NET    _
PROTOCOL_HEARTBEAT,    Constants.NET_PRIORITY_NORMAL,
11     heartbeatInitiated = true;
12 }

```

Gambar 4. 2. Modifikasi pada *Application Layer*

4.1.2. Modifikasi *Network Layer*

Pada *network layer* modifikasi dilakukan pada kelas *routing* yaitu dengan membuat *package* baru yaitu *sidnet.stack.users.csgp_adaptivepath.routing* untuk menampung kelas *routing* yang digandakan dari *sidnet.stack.std.routing.shortestgeopath*. Pada *package* ini terdapat 2 *class* yang dimodifikasi yaitu *class* *SGPWrapperMessage.java* menjadi *CSGPWrapperMessage.java* yang berfungsi sebagai pesan wrapper untuk semua tipe pesan, dan *class* *ShortestGeopathRouting.java* menjadi *class* *csgpaproutingmodif.java*, *class* ini akan menangani mekanisme routing pada metode *clustered shortest geopath routing protocol with adaptive path and aggregation*.

Pada *class* *csgpaproutingmodif.java* dilakukan modifikasi pada method *handleMessageDataValue*, yaitu hop selanjutnya dan proses agregasi data yang terjadi di *cluster head* terpilih. Pemilihan *cluster head* dilakukan berdasarkan beberapa parameter dan dilanjutkan dengan proses pergantian *cluster head*, adapun parameter pemilihan *cluster head* adalah berdasarkan jarak terdekat ke *cluster*

center, sisa energi tertinggi, kombinasi sisa energi tertinggi dan jarak terdekat ke *cluster* center, dan berdasarkan jumlah tetangga terbanyak dalam satu *cluster*.

Sebelum proses agregasi terjadi, setiap data value yang diterima oleh *node* akan dilakukan pengecekan apakah pesan tersebut berstatus *true* atau *false*, jika berstatus *true* maka akan diteruskan ke hop selanjutnya yaitu *node* yang berada paling dekat dengan *sink*. Jika berstatus *false* maka dilakukan pengecekan apakah ip *cluster head* sama dengan ip *node* tersebut, jika tidak sama maka kirim ke *cluster head*. Jika ya maka akan diperiksa prioritas data yang diterima.

Pemilihan cluser head dilakukan dengan membuat beberapa method baru di class *csgproutingmodif.java* sesuai dengan parameter yang telah ditentukan sebelumnya yaitu method *getClusterHead()*, *getCHDistance()*, *getCHTetangga* dan *getCHKombinasi()*. Kemudian untuk pergantian cluster head dilakukan dengan membuat beberapa method baru yaitu method *getSecondCH()*, *getThirdCH()* dan *getForthCH()*.

Sedangkan modifikasi untuk proses agregasi dilakukan di method *handleMessageDataValue* setelah proses seleksi cluster head selesai. Proses agregasi di *cluster head* terjadi dengan kondisi sebagai berikut :

- a. Jika data yang diterima oleh *cluster head* merupakan data dengan prioritas tinggi (P1) maka data akan diteruskan ke *sink* menggunakan jalur SGP dan setiap data yang dikirim dari *cluster head* pada *network layer* dicatat sebagai pesan terkirim.
- b. Jika data yang diterima oleh *cluster head* merupakan data dengan prioritas sedang (P2) maka data akan ditampung sementara di *cluster head* hingga mencapai batasan yang telah ditentukan yaitu 10 data, jika data yang diterima telah mencapai 10 data maka selanjutnya akan dilakukan proses agregasi dengan cara data-data tersebut akan dijumlahkan dan dirata-rata sehingga akan menghasilkan data baru hasil agregasi, maka data yang akan diteruskan ke *sink* adalah data baru hasil agregasi tersebut dalam bentuk pesan yang dikirim menggunakan jalur SGP.
- c. Jika data yang diterima oleh *cluster head* merupakan data dengan prioritas rendah(P3) maka data akan ditampung sementara di *cluster head* hingga mencapai batasan yang telah ditentukan yaitu 20 data, jika data yang

diterima telah mencapai 20 data maka selanjutnya akan dilakukan proses agregasi dengan cara data-data tersebut akan dijumlahkan dan dirata-rata sehingga akan menghasilkan data baru hasil agregasi, maka data yang akan diteruskan ke *sink* adalah data baru hasil agregasi tersebut dalam bentuk pesan yang dikirim menggunakan jalur SGP.

```

1  Public class csgpaproutingmodif implements RouteInterface{
...      ....
2  //method yang berfungsi menangani data dari node source ke cluster head
    dan sink
3  Private void handleMessageDataValue(...) {
4      //pemilihan cluster head berdasarkan parameter yang telah
        ditentukan
5      nextHop=getCHTetangga();

6      //kondisi pergantian cluster head
7      If (this.myNode.getEnergyManagement().getBattery()
            .getPercentageEnergyLevel()<=50) { ... }

8      //kondisi pesan belum melewati CH
9      if(msgDV.getStatus()==false){

10     //kondisi node tersebut adalah cluster head
11     if(nextHop==this.myNode.getIP()){

12         //kondisi pesan merupakan data prioritas tinggi(P1)
13         if(msgDataValue.isFire()==true){
14             wrapp message as new wrapper message;
15             set status=true;
16             set messageID;
17             send message;
18             stat as packet first priority sent;
19         }
20         //kondisi pesan merupakan data prioritas sedang(P2)
21         else if(msgDataValue.isAbnormal()==true){
22             //Proses Agregasi
23             receivedMessage2();
24             If(receivedMessage2=10){
25                 Calculate average value;
26                 Wrapp message as new wrapper message;
27                 set status=true;
28                 set messageID;
29                 send message;
30                 stat as packet second priority sent;
31             }
32         }
33     }
34 }

```

```

30 //kondisi pesan merupakan data prioritas rendah(P2)
31 else if(msgDataValue.isNormal()==true){
32 //Proses Agregasi
33 receivedMessage3();
34 If(receivedMessage3=20){
35 Calculate average value;
36 Wrapp message as new wrapper message;
37 set status=true;
38 set messageID;
39 send message;
40 stat as packet third priority sent;
    }

41 //kondisi node bukan cluster head
42 else if(nextHop!=myNode.getIP()){ ... }

43 //kondisi status pesan sudah melewati CH
44 else if(msgDV.getStatus() == true) {...}

45 //method yang menangani pemilihan CH berdasarkan sisa energi tertinggi
46 Private NetAddress getClusterHead() { ... }

47 // method yang menangani pemilihan Cluster head berdasarkan jumlah
tetangga //terbanyak
48 Private NetAddress getCHTetangga(){ ... }

49 //method yang menangani pemilihan cluster head berdasarkan kombinasi
sisa //energi tertinggi dan jarak terdekat ke CC
50 Private NetAddress getCHKombinasi() { ... }

51 //method yang menangani pemilihan cluster head berdasarkan jarak
terdekat ke CC
52 Private NetAddress getCHDistance() { ... }

53 //method yang menganani pemilihan CH kedua
54 private NetAddress getSecondCH(){ ... }

55 //method yang menganani pemilihan CH ketiga
56 private NetAddress getThirdCH(){ ... }

57 //method yang menganani pemilihan CH keempat
58 private NetAddress getForthCH(){ ... }
    }

```

Gambar 4. 3. Modifikasi pada Network Layer

4.2. Tahapan Uji Coba

Pengujian dilakukan untuk mengetahui kinerja protokol usulan yaitu *clustered shortest geopath routing protocol with adaptive path and aggregation*. Ujicoba yang dilakukan untuk mengetahui parameter pemilihan *cluster head* yang optimal pada metode pengiriman data yang bersifat terpusat. Kemudian untuk mengetahui parameter pergantian *cluster head* dan ujicoba juga dilakukan untuk proses agregasi data. Pengujian yang dilakukan bertujuan untuk mengetahui kinerja *clustered shortest geopath routing protocol with adaptive path and aggregation* dengan metode lain sehingga dapat mengetahui perbandingan kinerja masing-masing protokol sesuai dengan skenario yang telah dipaparkan sebelumnya.

Langkah –langkah pengujian pada penelitian ini adalah dengan membuat skenario pengujian, menentukan parameter pengujian dan menganalisa hasil pengujian yang telah dilakukan.

4.2.1. Skenario Pengujian

Untuk mengetahui kinerja *clustered shortest geopath routing protocol with adaptive path and aggregation*, dilakukan pengujian sesuai dengan yang telah dipaparkan pada bab 3 sebelumnya. Adapun skenario pengujian yang dilakukan adalah variasi parameter pemilihan *cluster head*, variasi parameter pergantian *cluster head*, variasi parameter agregasi data, perbandingan dengan protokol lain, variasi luas area dan variasi jumlah *node*. Pengujian dilakukan untuk membuktikan hipotesis bahwa teknik *clustering* yang diusulkan dapat meningkatkan kinerja *clustered shortest geopath routing protocol* pada lingkungan *wireless sensor network*. *Clustered shortest geopath routing protocol with adaptive path and aggregation* dikembangkan menggunakan simulator SIDnet SWANS, karena tidak memungkinkan untuk melakukan ujicoba pada jaringan nyata yang skala ujicoba cukup besar.

Pengujian ini melibatkan semua *node* yang berada dalam *field* menjadi *source node* yang bertugas mendapatkan dan mengirim data hasil pengamatan kepada *sink node*. Dalam pengujian dari kinerja protokol ini terdapat 6 skenario pengujian yang akan dijabarkan sebagai berikut:

a. Skenario ujicoba 1

Pengujian yang dilakukan untuk mengetahui parameter pemilihan *cluster head* yang optimal untuk protokol berbasis *clustering*, adapun parameter yang digunakan untuk pengujian adalah jarak terdekat ke *cluster head*, sisa energi tertinggi, kombinasi (0.5 sisa energi tertinggi dan 0.5 jarak terdekat ke *cluster center*), kombinasi (0.7 sisa energi tertinggi dan 0.3 jarak terdekat ke *cluster center*), kombinasi (0.3 sisa energi tertinggi dan 0.7 jarak terdekat ke *cluster center*) dan jumlah tetangga terbanyak. Informasi dari parameter – parameter ini diperoleh *node sensor* dari proses *heartbeat*.

b. Skenario ujicoba 2

Setelah mengetahui parameter pemilihan *cluster head* yang optimal, selanjutnya adalah melakukan pengujian untuk mengetahui parameter pergantian *cluster head*, skenario ini berguna untuk tahapan skenario selanjutnya. Adapun parameter pergantian *cluster head* adalah tanpa pergantian *cluster head*, sekali pergantian *cluster head* yaitu ketika sisa energi *node* $\leq 50\%$, dua kali pergantian *cluster head* yaitu ketika sisa energi *node* 75% dan $< 50\%$, tiga kali pergantian *cluster head* yaitu ketika sisa energi *node* 80%, 60% dan $< 40\%$.

c. Skenario ujicoba 3

Skenario selanjutnya pengujian parameter untuk proses agregasi data, yang mana pada skenario ini parameter pemilihan dan pergantian *cluster head* yang digunakan adalah parameter yang dihasilkan pada pengujian skenario sebelumnya yaitu skenario pemilihan *cluster head* dan skenario pergantian *cluster head*. Agregasi dilakukan untuk data dengan prioritas sedang (P2) dan prioritas rendah (P3) yaitu dengan menampung data yang diterima di *cluster head* sampai batasan tertentu. Adapun parameter batasan data yang diterima di *cluster head* sebelum di agregasi adalah sebagai berikut : P2=2 dan P3=5, P2=5 dan P3=10, P2=10 dan P3=20, P2=30 dan P3=50, P2=50 dan P3=100, P2=100 dan P3=200.

d. Skenario ujicoba 4

Pada skenario ini akan dilakukan perbandingan dengan beberapa metode lain, hal ini bertujuan untuk mengetahui kemampuan *Clustered shortest geopath routing*

protocol with adaptive path and aggregation dibandingkan dengan protokol yang lainnya. Parameter jumlah *node* yang digunakan dalam skenario ini adalah 200 *node* dengan parameter luas area 400x400, dan protokol pembanding yang digunakan adalah sebagai berikut: *Clustered shortest geopath routing protocol without aggregation*, *Clustered shortest geopath with random cluster head* dan *shortest geopath routing protocol without adaptive path and aggregation*.

e. Skenario ujicoba 5

Ujicoba dilakukan dengan melakukan variasi luas area luas area 100x100, 200x200m, 300x300m, 500x500m dengan jumlah *node* 200

f. Skenario ujicoba 6

Ujicoba dilakukan dengan memvariasikan jumlah *node* yaitu 100 *node*, 200 *node*, 300 *node* dengan luas area 300x300 m.

4.2.2. Parameter Pengujian dan Eksekusi Simulator

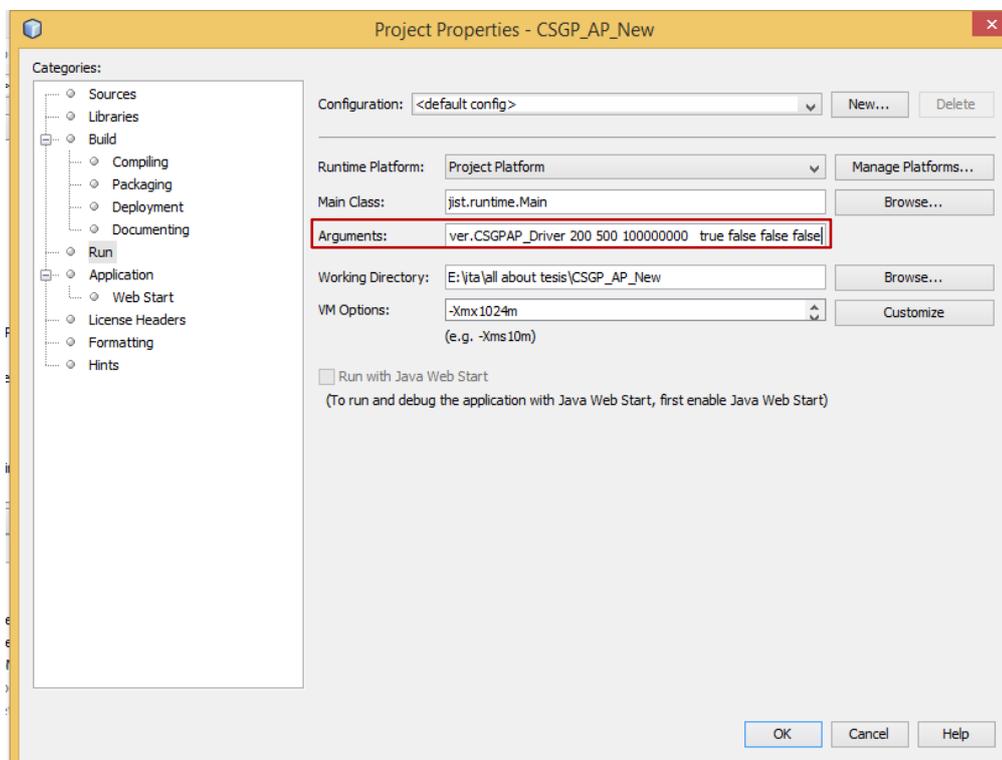
Untuk bisa mengimplementasikan skenario yang telah dirancang, diperlukan parameter pengujian yang diterapkan pada simulator. Melalui parameter pengujian simulator dapat menjalankan skenario yang telah ditentukan. Parameter yang dimaksud disini adalah kelas driver yang digunakan, jumlah *Node*, luas area simulasi, maksimum waktu simulasi dalam milisecond, dan mode protokol yang digunakan yaitu *Clustered shortest geopath routing protocol with adaptive path and aggregation*, *Clustered shortest geopath routing protocol without aggregation*, *shortest geopath routing protocol without adaptive path and aggregation*, *Clustered shortest geopath routing protocol with random cluster head*.

Parameter alamat kelas driver yang digunakan adalah alamat yang akan mengarahkan eksekusi program ke kelas driver yang telah dibuat yang berada pada `sidnet.stack.users.csgp_adaptivepath.driver`. kelas ini berfungsi sebagai modul yang menggabungkan informasi pada masing-masing *layer* dan menjadi entry point pada aplikasi SIDnet. Di dalam driver ditentukan kriteria *node*, topologi jaringan, algoritma yang dipakai pada masing-masing *layer*, statistik simulasi, dan lain-lain. Modul driver pada simulator SIDnet SWANS memiliki *class* utama, yaitu

CSGPDriver.java yang menyatukan semua informasi terkait simulasi dan menjadi entry point ke dalam aplikasi SIDNet.

Parameter selanjutnya adalah jumlah *node* yang akan digunakan sesuai dengan skenario yang di eksekusi. Parameter ketiga merupakan luas area simulasi yang memiliki area persegi dan yang diinput di simulator hanya panjang dari salah satu sisi tergantung dari skenario yang dijalankan. Selanjutnya parameter protokol yang digunakan, parameter yang menerima masukan berupa *true* dan *false*. Jika pada skenario pengujian menggunakan salah satu protokol yang telah ditentukan maka parameter yang diberikan adalah *true*, sedangkan jika jika protokol yang telah ditentukan tidak digunakan pada saat skenario pengujian maka input parameter yang diberikan adalah *false*.

Parameter pengujian yang dijelaskan diatas akan dieksekusi oleh simulator dengan memasukkan parameter pengujian melalui *run arguments* yang dapat diakses pada fitur *project properties* di NetBeans sebagaimana divisualisasikan pada Gambar 4.6, parameter ini akan dimasukkan ke *run arguments* sebelum program simulator berjalan.



Gambar 4. 4. *Input* Parameter Pengujian pada *Run Arguments*

Argumen yang diperlukan adalah jumlah node, luas area, waktu maksimal simulasi dan protokol routing yang digunakan. Untuk lebih memperjelas parameter pengujian masing-masing skenario akan ditampilkan pada tabel 4.1. Rangkuman parameter pengujian simulator.

Tabel 4. 1. Rangkuman Parameter Pengujian Pada Simulator

Skenario	Protokol	Luas area	Jumlah node	Skrip pengujian
Skenario 1	<i>Clustered shortest geopath routing protocol with adaptive path and aggregation (CSGP-APA)</i>	400x400	200	jist.swans.Main sidnet.stack.users.csgp_adaptivepath.driver.CS GPAP_Driver 200 400 100000000 true false false false
Skenario 2				
Skenario 3				
Skenario 4	<i>Clustered shortest geopath routing protocol with adaptive path and aggregation (CSGP-APA)</i>	400x400	200	jist.swans.Main sidnet.stack.users.csgp_adaptivepath.driver.CS GPAP_Driver 200 400 100000000 true false false false
	<i>Clustered shortest geopath routing protocol without aggregation (CSGP-WA)</i>			jist.swans.Main sidnet.stack.users.csgp_adaptivepath.driver.CS GPAP_Driver 200 400 100000000 false true false false
	<i>shortest geopath routing protocol without adaptive path and aggregation (CSGP-WAPA)</i>			jist.swans.Main sidnet.stack.users.csgp_adaptivepath.driver.CS GPAP_Driver 200 400 100000000 false false true false
	<i>Clustered shortest geopath routing protocol with random cluster head (CSGP-RCH)</i>			jist.swans.Main sidnet.stack.users.csgp_adaptivepath.driver.CS GPAP_Driver 200 400 100000000 false false false true
Skenario 5	CSGP-APA	100x100	200	jist.swans.Main sidnet.stack.users.csgp_adaptivepath.driver.CS GPAP_Driver 200 400 100000000 true false false false
		200x200		
		300x300		
		500x500		

Skenario 6	CSGP-APA	300x300	100	jist.swans.Main sidnet.stack.users.csgp_ <i>adaptivepath.driver.CS</i> GPAP_Driver 200 400 100000000 <i>true false</i> <i>false false</i>
			300	
			400	

4.3. Hasil dan Analisis

Hasil pengujian pada sub bab ini diperoleh dengan melakukan analisis terhadap file teks yang dihasilkan saat pengujian *Clustered shortest geopath routing protocol with adaptive path and aggregation*.

4.3.1. Analisis Skenario 1

Pada skenario ini dilakukan ujicoba terhadap pemilihan *cluster head* yang optimal berdasarkan beberapa parameter yang telah ditentukan, dimana setiap parameter diuji sebanyak 3 kali dan hasil yang digunakan untuk analisa merupakan rata-rata dari 3 pengujian yang telah dilakukan. Pengujian dilakukan pada jumlah *node* dan luas area yang sama yaitu 200 *node* dengan luas area 400x400 meter.

Hasil pengujian selanjutnya akan dianalisa berdasarkan aspek yang telah ditentukan yaitu *network lifetime*, *event detection*, *packet delivery ratio* dan *Latency*. Dari pengujian yang telah dilakukan diperoleh rangkuman hasil pengujian yang terlihat pada tabel 4.2. *Network Lifetime* dan *Latency* Hasil pengujian skenario 1 dan Tabel 4.3. *Event Detection* dan *Packet Delivery Ratio* Hasil pengujian skenario 1.

Tabel 4.2. *Network Lifetime* dan *Latency* Hasil pengujian skenario 1

Parameter Pemilihan CH	Network Lifetime (Menit)	Latency (ms)		
		P1	P2	P3
Energi tertinggi	2215	92	105	106
Jarak terdekat ke cc	2211.3	96,7	105	106
Tetangga terbanyak	2249	91,3	106	107
0.7 SE + 0.3 JCC	2155	91,3	103	106,3
0.5 SE + 0.5 JCC	2189.7	91,3	106	107
0.3 SE + 0.7 JCC	1941.3	95,3	105,3	107,3

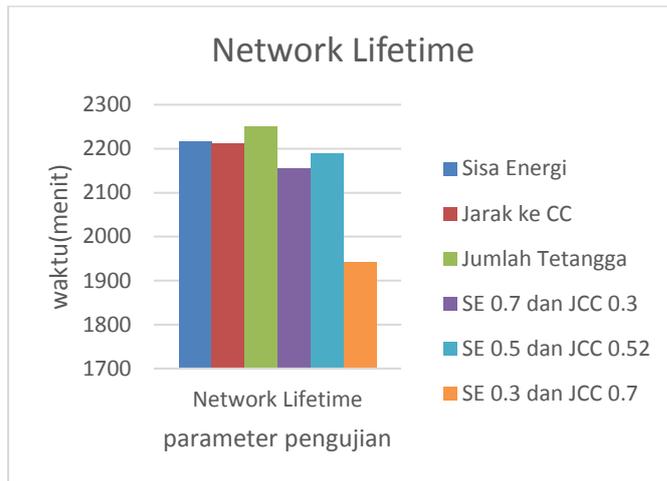
Keterangan :

SE = Sisa Energi

JCC = Jarak ke cluster center

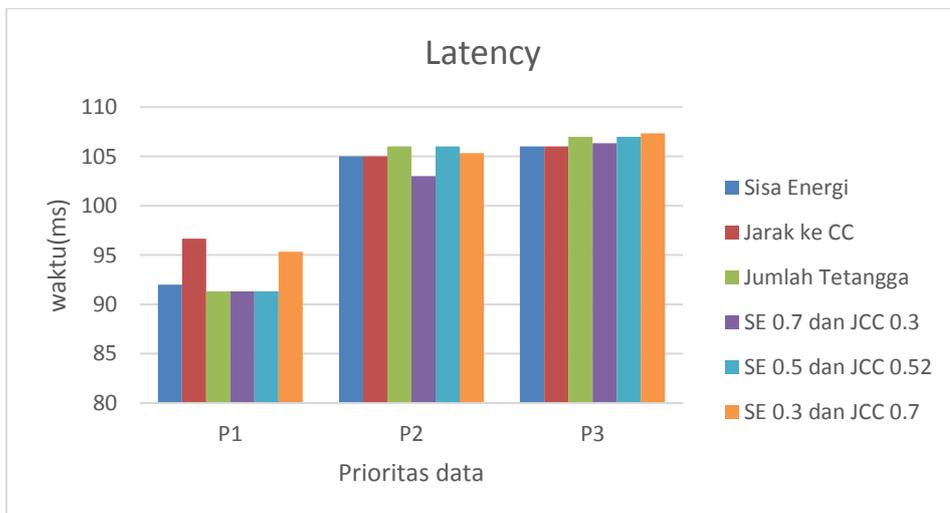
CC = Cluster center

Dari tabel 4.2 terlihat *network lifetime* tertinggi pada parameter jumlah tetangga terbanyak yaitu 2249 menit. Untuk lebih jelas maka rangkuman hasil pengujian divisualisasi pada gambar 4.5. *Network lifetime* skenario 1 dan gambar 4.6. *Latency* skenario 1.



Gambar 4.5. *Network Lifetime* Skenario 1

Dari gambar 4.5 terlihat pemilihan *cluster head* berdasarkan jumlah tetangga paling banyak memiliki *network lifetime* tertinggi dibandingkan dengan parameter yang lainnya, yaitu 34 menit lebih lama dibandingkan dengan parameter sisa energi tertinggi, 37,7 menit dibandingkan dengan parameter jarak ke *cluster center*.



Gambar 4.6. *Latency* Skenario 1

Berdasarkan gambar 4.6 terlihat bahwa Parameter jumlah tetangga terbanyak, parameter kombinasi 0,7 sisa energi dan 0,3 jarak terdekat ke cluster center dan parameter kombinasi 0,5 sisa energi dan 0,5 jarak terdekat ke cluster center memiliki *latency* terendah untuk data dengan prioritas tinggi (P1) yaitu 5,4ms lebih rendah dibandingkan dengan parameter jarak terdekat ke *cluster center*, sedangkan untuk data dengan prioritas sedang(P2) parameter kombinasi 0,7 sisa energi dan 0,3 jarak terdekat ke cluster center memiliki *latency* 3ms lebih rendah dibandingkan dengan parameter jumlah tetangga terbanyak dan parameter kombinasi 0,5 sisa energi dan 0,5 jarak terdekat ke *cluster center*. Untuk data dengan prioritas rendah (P3) parameter sisa energi tertinggi dan jarak terdekat ke *cluster center* memiliki *latency* 1,3ms lebih rendah dibandingkan parameter kombinasi 0,3 sisa energi dan 0,7 jarak terdekat ke *cluster center*.

Tabel 4.3. *Event Detection* dan *Packet Delivery Ratio* Hasil pengujian skenario 1

Parameter Pemilihan CH	<i>Packet Delivery Ratio</i> (P1,P2,P3)			<i>Event Detection (P1)</i>		
	Paket dikirim	Paket diterima	Persentase (%)	Event terjadi	Event terdeteksi	Persentase (%)
Energi tertinggi	59214	56930	96,1	10306	9940	96,4
Jarak terdekat ke cc	59182	56301	95,1	10306	9500	92,2
Tetangga terbanyak	59614	59005	99,0	10308	10218	99,1
0.7 SE + 0.3 JCC	58415	57664	98,7	10308	10210	99,0
0.5 SE + 0.5 JCC	58878	56324	95,7	10306	10199	99
0.3 SE + 0.7 JCC	53223	51320	96,4	10306	10109	98,1

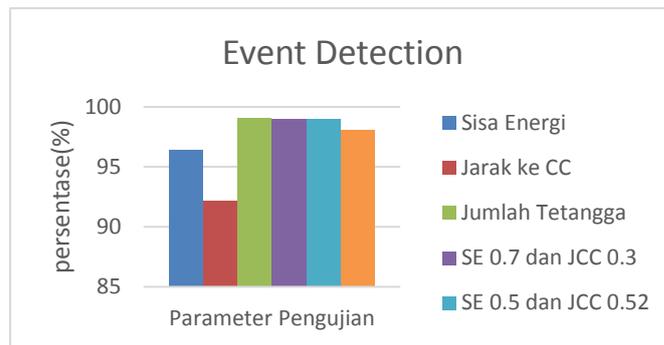
Keterangan :

SE = Sisa Energi

JCC = Jarak ke cluster center

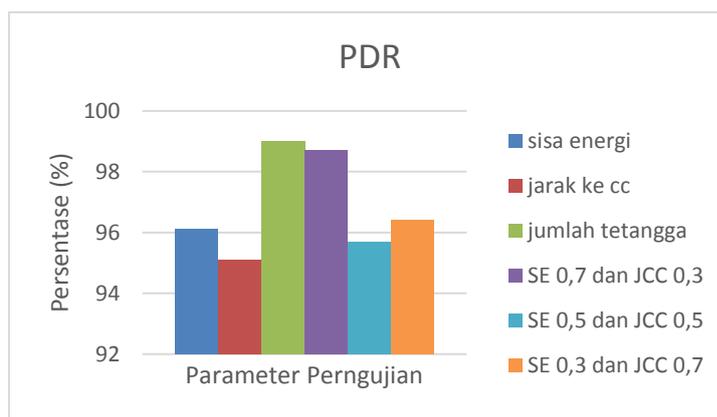
CC = Cluster center

Dari tabel 4.3 terlihat *event detection* tertinggi terdapat pada parameter jumlah tetangga terbanyak yaitu 99,1% dengan total *event* yang terdeteksi adalah 10218 *event* dari 10308 *event* yang terjadi. Untuk *packet delivery ratio* parameter jumlah tetangga terbanyak memiliki nilai tertinggi yaitu 99,0% dengan total paket yang diterima adalah 59005 dari total 59614 paket yang dikirim. Untuk lebih jelas maka rangkuman hasil pengujian divisualisasi pada gambar 4.6. *Event Detection* skenario 1 dan gambar 4.7 *packet delivery ratio* skenario 1



Gambar 4.7. *Event Detection* Skenario 1

Berdasarkan gambar 4.7, pemilihan *cluster head* berdasarkan parameter jumlah tetangga terbanyak memiliki nilai *event detection* tertinggi dibandingkan dengan parameter yang lain nya, yaitu lebih tinggi 6,9% dibandingkan dengan paramater jarak terdekat ke *cluster center*.



Gambar 4.8. *Packet Delivery Ratio* Skenario 1

Berdasarkan gambar 4.8 pemilihan *cluster head* berdasarkan jumlah tetangga terbanyak memiliki nilai *packet delivery ratio* tertinggi dibandingkan

dengan parameter yang lainnya, yaitu lebih tinggi 3,9% dibandingkan dengan parameter jarak terdekat ke *cluster center*.

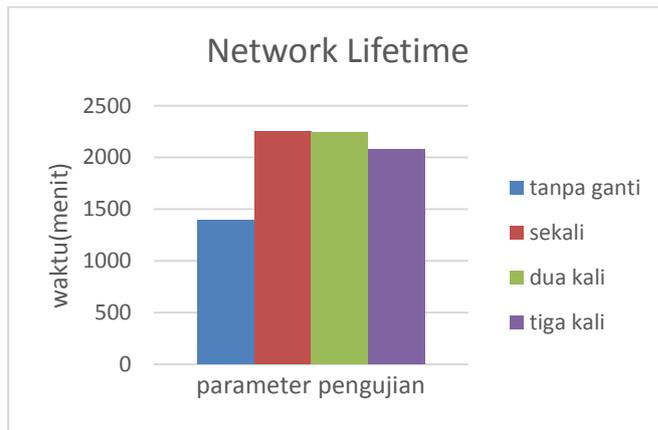
4.3.2. Analisis Skenario 2

Setelah pemilihan *cluster head* yang optimal diperoleh yaitu berdasarkan jumlah tetangga terbanyak, maka selanjutnya dilakukan skenario pergantian dari *cluster head* yang terpilih. Pengujian pergantian *cluster head* dilakukan berdasarkan beberapa parameter yaitu tanpa pergantian *cluster head*, sekali pergantian *cluster head* yaitu pada saat sisa energi *node* mencapai $\leq 50\%$, dua kali pergantian *cluster head* yaitu ketika sisa energi *node* 75% dan 50%, tiga kali pergantian *cluster head* yaitu ketika sisa energi *node* 80%, 60% dan $< 40\%$. Pada pengujian ini juga dilakukan sebanyak 3 kali untuk masing-masing parameter, selanjutnya hasil dari pengujian tersebut akan di kumpulkan dan dirata-rata dan hasil akhir rata-rata tersebut yang akan digunakan untuk perbandingan untuk masing-masing parameter. Adapun rangkuman hasil akhir rata-rata pengujian terlihat pada tabel 4.4. *Network Lifetime* dan *Latency* Hasil pengujian skenario 2 dan Tabel 4.5. *Event Detection* dan *packet delivery ratio* Hasil pengujian skenario 2.

Tabel 4.4. *Network Lifetime* dan *Latency* Hasil pengujian skenario 2

Parameter Pergantian CH	Network Lifetime (Menit)	Latency (ms)		
		P1	P2	P3
Tanpa Ganti CH	1491,3	89	88,3	105,7
Sekali ganti CH (Sisa energi $\leq 50\%$)	2246,3	81	91,7	93
Dua kali ganti CH (Sisa energi $< 75\%$ dan $< 50\%$)	2238,7	79,3	89,3	90,7
Tiga kali ganti CH (Sisa energi $< 80\%$, 60% dan 40%)	2079	78	85	91,3

Hasil pengujian skenario 2 terlihat pada tabel 4.4, dari tabel diatas terlihat bahwa parameter sekali pergantian *cluster head* memiliki hasil *network lifetime* tertinggi jika dibandingkan dengan parameter yang lain nya yaitu dengan 2246,4 menit. Untuk lebih memperjelas maka rangkuman hasil pengujian divisualisasikan pada gambar 4.9. *Network lifetime* skenario 2 dan gambar 4.10. *Latency* skenario 2.



Gambar 4.9. *Network Lifetime* Skenario 2

Dari gambar 4.9 terlihat perbedaan *network lifetime* untuk masing-masing parameter pergantian *cluster head*. Dimana parameter sekali pergantian *cluster head* memiliki *network lifetime* tertinggi yaitu 50,6% atau 755 menit lebih lama dibandingkan dengan parameter tanpa pergantian *cluster head*. Hal ini terjadi karena tanpa pergantian *cluster head* maka beban *cluster head* terpilih menjadi lebih tinggi dibandingkan dengan *node* yang lain nya sehingga *node* yang terpilih menjadi *cluster head* lebih cepat kehabisan energi yang menyebabkan *network lifetime*nya semakin pendek, mengingat *network lifetime* yang dimaksud disini adalah rentang waktu mulai *node* disebarkan hingga *node* pertama mati.



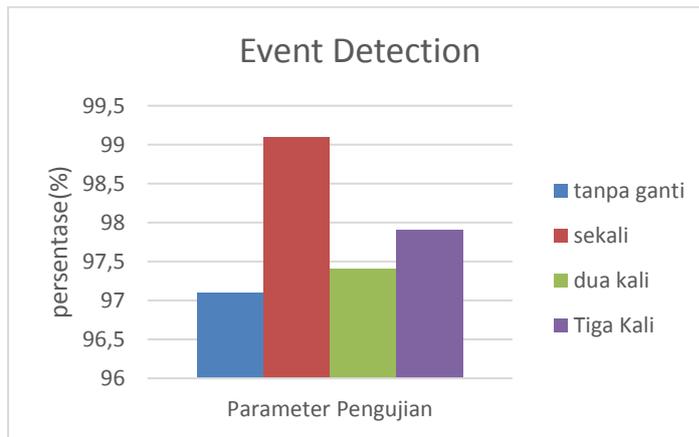
Gambar 4. 10. *Latency* Skenario 2

Berdasarkan gambar 4.10 terlihat bahwa parameter tiga kali pergantian *cluster head* memiliki *Latency* terendah untuk data dengan prioritas tinggi (P1) yaitu 11 ms lebih rendah dibandingkan dengan parameter tanpa pergantian *cluster head* dan 6,7 ms lebih rendah dibandingkan dengan parameter sekali ganti *cluster head* untuk data dengan prioritas sedang (P2). Sedangkan untuk data dengan prioritas rendah, parameter dua kali pergantian *cluster head* memiliki *Latency* lebih rendah 15ms dibandingkan dengan parameter tanpa pergantian *cluster head*. Hal ini terjadi karena ketika pergantian *cluster head* maka posisi *cluster head* terpilih juga berganti dan membuat rata-rata jarak kirim menjadi lebih beragam jika dibandingkan tanpa pergantian dan sekali pergantian *cluster head*.

Tabel 4.5. Event Detection dan packet delivery ratio Hasil pengujian skenario 2.

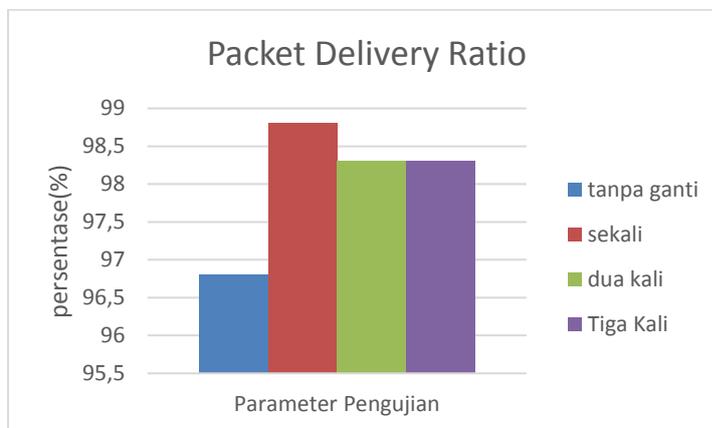
Parameter Pergantian CH	Packet Delivery Ratio (P1,P2,P3)			Event Detection (P1)		
	Paket dikirim	Paket diterima	Persentase (%)	Event terjadi	Event terdeteksi	Persentase (%)
Tanpa Ganti CH	50207	48576	96,8	10305	10003	97,1
Sekali ganti CH (Sisa energi $\leq 50\%$)	59607	58877	98,8	10305	10210	99,1
Dua kali ganti CH (Sisa energi $< 75\%$ dan $< 50\%$)	59567	58581	98,3	10305	10036	97,4
Tiga kali ganti CH (Sisa energi $< 80\%$, 60% dan 40%)	55132	54128	98,3	10306	10085	97,9

Hasil pengujian skenario 2 terlihat pada tabel 4.5, dari tabel diatas terlihat bahwa parameter sekali pergantian *cluster head* memiliki hasil *event detection* tertinggi yaitu 99,1% dengan total *event* yang terdeteksi adalah 10210 *event* dari 10305 *event* yang terjadi, sedangkan untuk *packet delivery ratio* yaitu 98,8% dengan total paket yang diterima adalah 58877 paket dari total 59607 paket yang dikirim. Untuk lebih memperjelas maka rangkuman hasil pengujian divisualisasikan pada gambar 4.10. *Event Detection* skenario 2, gambar 4.11. *packet delivery ratio* skenario 2.



Gambar 4.11. *Event Detection* Skenario 2

Selanjutnya berdasarkan gambar 4.11 yang memperjelas perbandingan *event detection* untuk masing-masing parameter pengujian. dari gambar diatas terlihat bahwa parameter sekali pergantian *cluster head* memiliki hasil *Event Detection* tertinggi dibandingkan dengan parameter tanpa pergantian *cluster head* yaitu 2,0%. Dari 59607 total paket yang dikirim, Total *event* yang terdeteksi adalah 10210 *event* dari 10305 *event* yang terjadi. Perbedaan persentase hasil pengujian tidak terlalu signifikan karena pada parameter tanpa pergantian *cluster head*, data yang terkirim belum sebanyak data yang terkirim pada parameter sekali pergantian *cluster head*, hal ini terjadi karena *network lifetime* parameter tanpa pergantian *cluster* lebih pendek jika dibandingkan dengan parameter sekali pergantian *cluster head*.



Gambar 4.12. *Packet Delivery Ratio* Skenario 2

Selanjutnya berdasarkan gambar 4.12 yang memperjelas perbandingan *packet delivery ratio* untuk masing-masing parameter pengujian. dari gambar diatas terlihat bahwa parameter sekali pergantian *cluster head* memiliki hasil *packet delivery ratio* tertinggi yaitu 2% lebih tinggi dibandingkan dengan parameter tanpa pergantian *cluster head* dengan total paket yang diterima adalah 58877 paket dari total 59607 paket yang dikirim.

4.3.3. Analisis Skenario 3

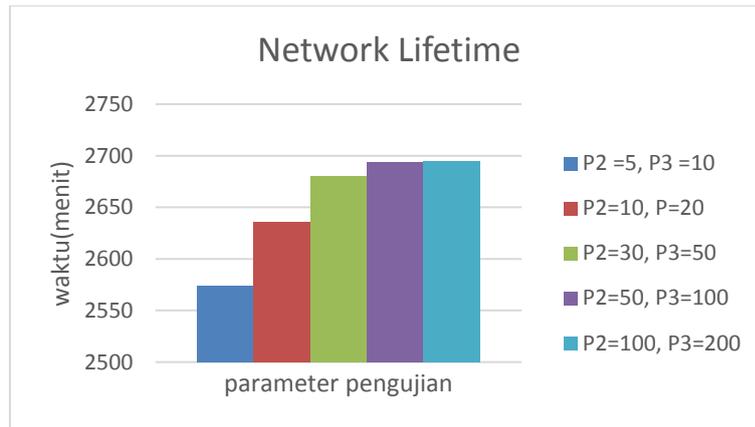
Analisis skenario 3 dilakukan setelah pengujian pemilihan dan pergantian *cluster head* selesai dilakukan dan memperoleh parameter pergantian *cluster head* yang optimal yaitu dengan sekali pergantian *cluster head*. Pengujian dilakukan terhadap parameter untuk proses agregasi berdasarkan prioritas data yang diterima oleh *cluster head*, dimana yang akan dijadikan parameter nya adalah jumlah data yang ditampung oleh *cluster head* pada batasan tertentu untuk dikirim ke *cluster head*. Adapun parameter yang digunakan adalah data untuk prioritas sedang (P2)=5, data prioritas rendah (P3)=10, P2=10 dan P3=20, P2=30 dan P3=50, P2=50 dan P3=100, P2=100 dan P3=200.

Pada pengujian ini juga dilakukan sebanyak 3 kali untuk masing-masing parameter, selanjutnya hasil dari pengujian tersebut akan di kumpulkan dan dirata-rata dan hasil akhir rata-rata tersebut yang akan digunakan untuk perbandingan antar masing-masing parameter. Adapun hasil akhir rata pengujian terlihat pada tabel 4.6. *Network Lifetime* dan *Latency* Hasil pengujian skenario 3 dan Tabel 4.7. *Event Detection* dan *packet delivery ratio* Hasil pengujian skenario 3.

Tabel 4.6. *Network Lifetime* dan *Latency* Hasil pengujian skenario 3

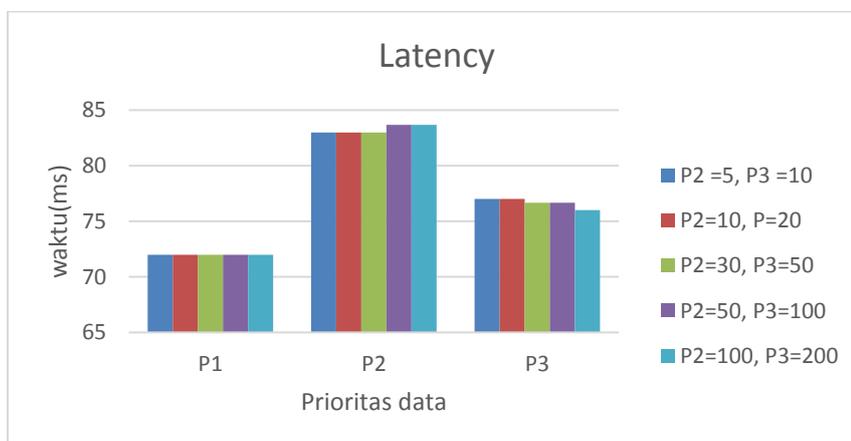
Parameter Agregasi Data	Network Lifetime (Menit)	Latency (ms)		
		P1	P2	P3
P2=5 dan P3=10	2574,3	72	83	77
P2=10 dan P3=20	2636	72	83	77
P2=30 dan P3=50	2680	72	83	76,7
P2=50 dan P3=100	2694	72	83,7	76,7
P2=100 dan P3=200	2694,3	72	83,7	76

Dari tabel 4.6 terlihat hasil pengujian untuk skenario 3 *network lifetime* terendah terdapat pada parameter P2=5 dan P3=10 dan *network lifetime* tertinggi terdapat pada Parameter P2=100 dan P3=200 yaitu 2694,3 menit. Sedangkan untuk *latency* relatif konstan untuk masing-masing parameter. Untuk lebih memperjelas maka divisualisasi pada gambar 4.13. *Network lifetime* skenario 3 dan gambar 4.14. *Latency* skenario 3.



Gambar 4.13. *Network Lifetime* Skenario 3

Gambar 4.13 memaparkan *network lifetime* untuk skenario 3, jika dilihat dari tabel ini parameter P2=100 dan P3=200 memiliki *network lifetime* tertinggi yaitu 120 menit lebih lama dibandingkan dengan parameter P2=5 dan P3=10. Hal ini terjadi karena transmisi data pada parameter P2=100 dan P3=200 lebih sedikit dibandingkan dengan parameter yang lainnya, karena paket yang dikirim lebih sedikit dan transmisi jaringan yang relatif rendah sehingga bisa lebih menghemat energi masing-masing node.



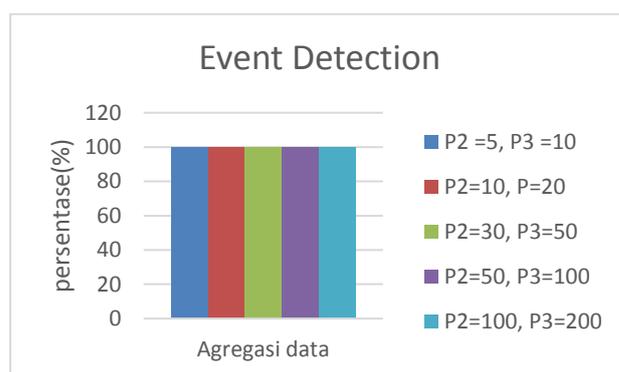
Gambar 4.14. *Latency* Skenario 3

Pada gambar 4.14 dapat terlihat bahwa *latency* untuk masing -masing parameter relatif konstan, hal ini terjadi karena luas area dan jumlah node yang sama dan setiap parameter telah melakukan agregasi untuk data yang akan dikirim sehingga transmisi data yang terjadi di jaringan tidak sesibuk jaringan tanpa adanya agregasi data.

Tabel 4.7. *Event Detection* dan *Packet Delivery Ratio* Hasil pengujian skenario 3

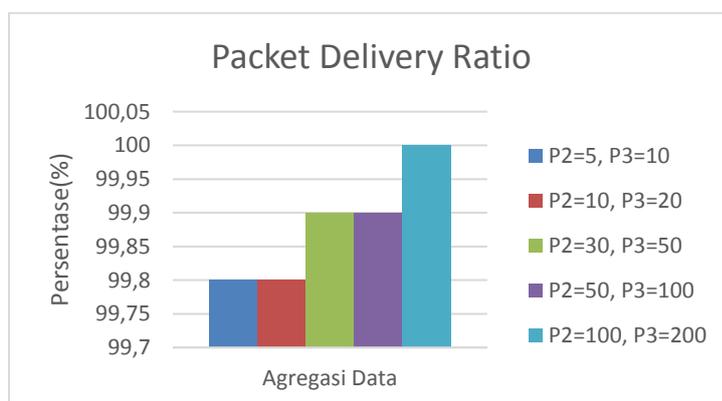
Parameter Agregasi Data	<i>Packet Delivery Ratio (P1,P2,P3)</i>			<i>Event Detection (P1)</i>		
	Paket dikirim	Paket diterima	Persentase (%)	Event terjadi	Event terdeteksi	Persentase (%)
P2=5 dan P3=10	24066	24024	99,8	10304	10304	100
P2=10 dan P3=20	17264	17227	99,8	10307	10307	100
P2=30 dan P3=50	12924	12912	99,9	10307	10307	100
P2=50 dan P3=100	11642	11635	99,9	10304	10304	100
P2=100 dan P3=200	10931	10927	100	10306	10306	100

Dari tabel 4.7 terlihat hasil pengujian untuk skenario 3 pada parameter P2=10 dan P3=20, karena untuk *network lifetime*, *event detection*, *packet delivery ratio* dan *latency* yang relatif konstan. Parameter ini memiliki *event detection* 100% dengan total *event* yang terdeteksi adalah 10307 *event* dari 10307 *event* yang terjadi, sedangkan untuk *packet delivery ratio* yaitu 99,8% dengan total paket yang diterima adalah 17264 paket dari total 17227 paket yang dikirim. Untuk lebih memperjelas maka divisualisasi pada gambar 4.15. *Event Detection* skenario 3 dan gambar 4.16. *packet delivery ratio* skenario 3.



Gambar 4.15. *Event Detection* Skenario 3

Selanjutnya gambar 4.15 memaparkan *event detection* untuk masing - masing parameter. Pada skenario 3 masing – masing parameter memiliki *event detection* yang relatif konstan yaitu 100%, hal ini terjadi karena paket dengan prioritas tinggi(P1) akan langsung dikirim ke sink tanpa dilakukan agregasi sebelumnya.



Gambar 4.16. *Packet Delivery Ratio* Skenario 3

Berdasarkan gambar 4.16, *packet delivery ratio* dipengaruhi oleh jumlah transmisi data, semakin sedikit transmisi data yang terjadi maka semakin tinggi *packet delivery rasionya*. Pada skenario ini parameter P2=100 dan P3=200 memiliki *packet delivery ratio* tertinggi yaitu 10927 data yang diterima dari 10931 data yang dikirim. Hal ini karena dengan agregasi data dengan nilai tersebut maka paket data yang dikirim semakin sedikit dan lebih banyak paket yang diterima oleh sink.

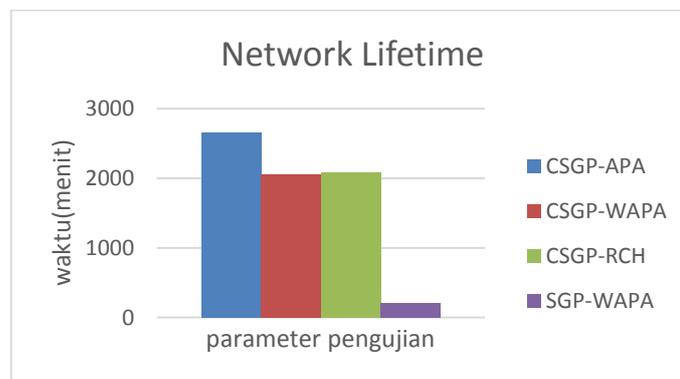
4.3.4. Analisis Skenario 4

Pada skenario ini, *Clustered Shortest Geopath Routing Protocol with adaptive path and aggregation* dibandingkan dengan protokol lain yaitu *Clustered Shortest Geopath Routing Protocol without aggregation* (CSGP-WA), *Clustered Shortest Geopath Routing Protocol with random cluster head* (CSGP-RCH) dan *Shortest Geopath Routing Protocol without adaptive path and aggregation* (SGP-WAPA). Masing-masing protokol disimulasikan sebanyak 3 kali dengan jumlah *node* 200 dan luas area simulasi 500x500 meter. Hasil 3 kali pengujian kemudian akan di rata-rata dan hasil rata-rata tersebut yang kemudian dikomparasi kemudian dianalisa. Dari simulasi yang dilakukan diperoleh hasil yang ditampilkan pada tabel 4.8. *Network Lifetime* dan *Latency* Hasil pengujian skenario 4 dan Tabel 4.9. *Event Detection* dan *Packet Delivery Ratio* Hasil pengujian skenario 4.

Tabel 4.8. *Network Lifetime* dan *Latency* Hasil pengujian skenario 4

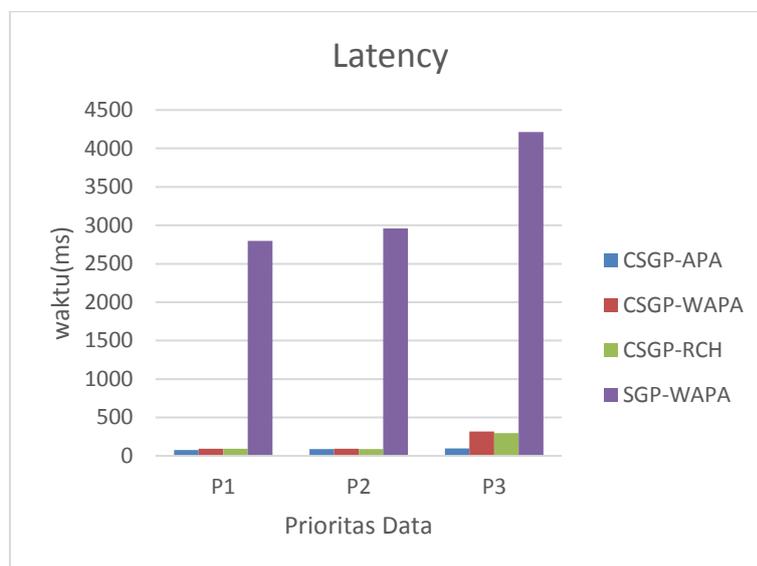
Metode	Network Lifetime (Menit)	Latency (ms)		
		P1	P2	P3
CSGP-APA	2658.7	77,3	87	96,3
CSGP-WA	2056.7	92,3	93	329,3
CSGP-RCH	2083	92	88,7	294,7
SGP-WAPA	206.7	2796,7	2961	4215

Dari tabel 4.8 terlihat metode *Clustered shortest geopath routing protocol with adaptive path and aggregation* (CSGP-APA) memiliki *network lifetime* tertinggi yaitu 2658.7 menit dan *latency* terendah. Untuk lebih memperjelas maka rangkuman hasil pengujian divisualisasi pada gambar 4.17. *Network lifetime* skenario 4 dan gambar 4.18. *Latency* skenario 4.



Gambar 4.17. *Network Lifetime* Skenario 4

Pada gambar 4.17 terlihat perbandingan *network lifetime* untuk masing-masing metode, dimana metode CSGP-APA memiliki *network lifetime* tertinggi yaitu 2452 menit lebih lama dibandingkan dengan metode SGP-WAPA, hal ini terjadi karena pada metode SGP-WAPA tidak menggunakan kontrol pada pengiriman data sehingga data yang terkirim tidak terkontrol yang menyebabkan trafik jaringan menjadi sangat padat dan konsumsi energi menjadi lebih besar. Selanjutnya jika dibandingkan dengan CSGP-WA, metode CSGP-APA memiliki *network lifetime* yang lebih lama yaitu 602 menit.



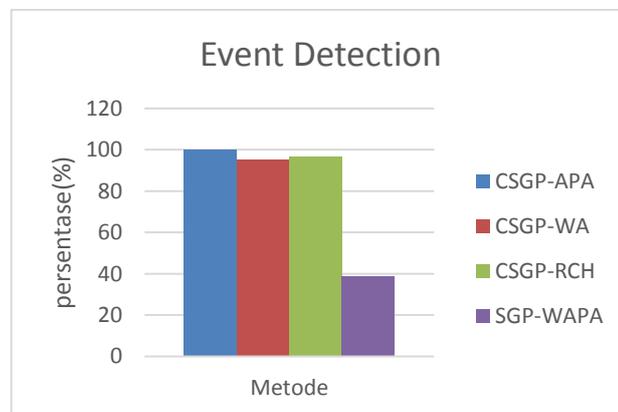
Gambar 4.18. *Latency* Skenario 4

Berdasarkan gambar 4.18, metode CSGP-APA memiliki *latency* terendah untuk masing-masing prioritas data, yaitu menurun atau lebih cepat 15 ms untuk data dengan prioritas tinggi, 6ms untuk data dengan prioritas sedang dan 223ms untuk data dengan prioritas rendah dibandingkan dengan metode CSGP-WA. Sedangkan jika dibandingkan dengan metode SGP-WAPA, metode CSGP-APA mampu menurun 2719,4ms untuk data dengan prioritas tinggi, 2874ms lebih cepat untuk data dengan prioritas sedang, dan 4118ms lebih cepat untuk data dengan prioritas rendah.

Tabel 4. 9. *Event Detection* dan *Packet Delivery Ratio* Hasil pengujian skenario 4

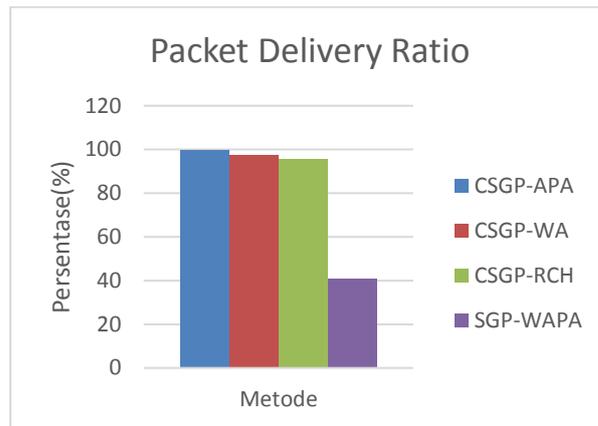
Metode	<i>Packet Delivery Ratio (P1,P2,P3)</i>			<i>Event Detection (P1)</i>		
	Paket dikirim	Paket diterima	Persentase (%)	Event terjadi	Event terdeteksi	Persentase (%)
CSGP-APA	14271	14260	99,9	7837	7837	100
CSGP-WA	39719	38721	97,5	6176	5893	95,4
CSGP-RCH	40138	38246	95,3	6175	5993	97,0
SGP-WAPA	584518	238486	40,8	6487	2515	38,8

Dari tabel 4.9 terlihat metode *Clustered shortest geopath routing protocol with adaptive path and aggregation (CSGP-APA)* memiliki *event detection* dan *packet delivery ratio* tertinggi dibandingkan dengan metode lainnya. CSGP-APA memiliki *event detection* 100% dengan 7837 *event* yang terdeteksi oleh *sink* dari 7837 *event* yang dideteksi oleh *node sensor*, *packet delivery ratio* 99,9% dengan total 14260 paket yang diterima oleh *sink* dari 14271 paket yang dikirim dari *cluster head*. yang Untuk lebih memperjelas maka rangkuman hasil pengujian divisualisasi pada gambar 4.19. *Event Detection* skenario 4 dan gambar 4.20. *Packet Delivery Ratio* skenario 4.



Gambar 4. 19. *Event Detection* Skenario 4

Selanjutnya pada gambar 4.19 terlihat metode CSGP-APA memiliki *event detection* tertinggi yaitu 100% lebih tinggi dibandingkan dengan metode SGP-WAPA, dan 3% lebih tinggi dibandingkan dengan metode CSGP-RCH. Pada metode CSGP-APA berhasil mendeteksi semua *event* yang terjadi yaitu 7837 *event*.



Gambar 4. 20. *Packet Delivery Ratio* Skenario 4

CSGP-APA memiliki *packet delivery ratio* tertinggi yaitu 99,9% dengan total paket yang diterima 14260 paket dari total 14271 paket data yang dikirim. *Packet delivery ratio* CSGP-APA lebih tinggi 59,1% dibandingkan dengan metode SGP-WAPA dan 2,4% lebih tinggi dibandingkan dengan metode CSGP-WA.

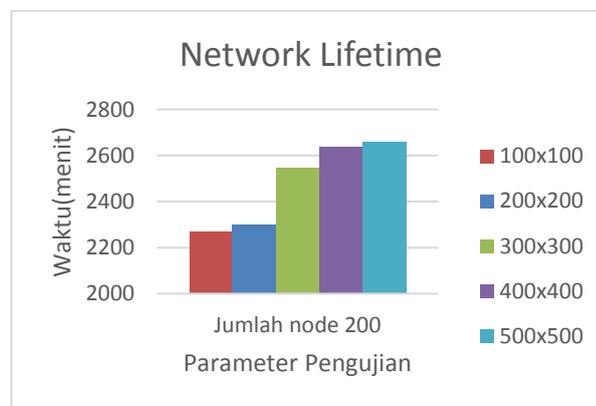
4.3.5. Analisis Skenario 5

Berdasarkan pengujian sebelumnya dan mendapatkan hasil pemilihan cluster head berdasarkan jumlah tetangga terbanyak, dengan sekali pergantian cluster head dan agregasi data berdasarkan jumlah data $P2=10$ dan $P3=20$ data, hasil pengujian yang disebutkan diatas kemudian dinamakan dengan *clustered shortest geopath routing protocol with adaptive path and aggregation* (CSGP-APA). Pengujian skenario ini dilakukan dengan melakukan ujicoba terhadap CSGP-APA dengan jumlah *node* yang sama yaitu 200 dan variasi pada luas area simulasi yaitu 100×100 , 200×200 , 300×300 , 400×400 dan 500×500 . Masing-masing luas area disimulasikan sebanyak 3 kali. Hasil 3 kali pengujian kemudian akan di rata-rata dan hasil rata-rata tersebut yang kemudian dikomparasi kemudian dianalisa. Dari simulasi yang dilakukan diperoleh hasil yang ditampilkan pada tabel 4.10. *Network Lifetime* dan *Latency* Hasil pengujian skenario 5 dan Tabel 4.11. *Event Detection* dan *Packet Delivery Ratio* Hasil pengujian skenario 5.

Tabel 4. 10. *Network Lifetime* dan *Latency* Hasil pengujian skenario 2

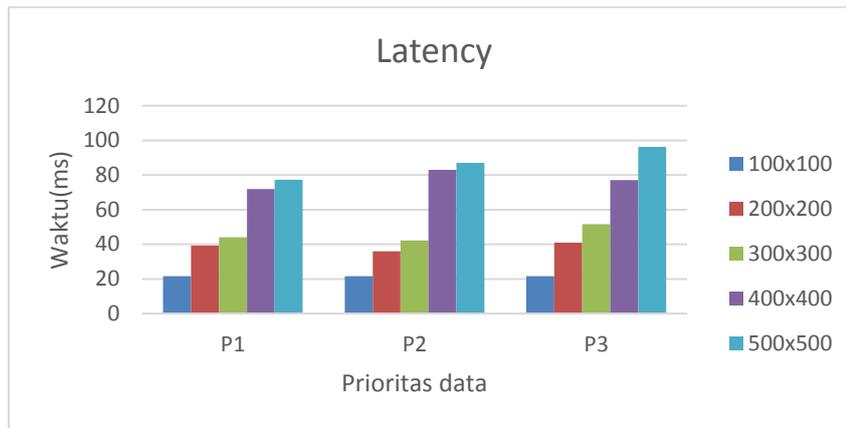
Parameter Luas Area	Network Lifetime (Menit)	Latency (ms)		
		P1	P2	P3
100x100	2269,3	21,6	21,6	21,6
200x200	2299,3	39,3	36	41
300x300	2548	44	42,3	51,6
400x400	2640,3	72	83	76,7
500x500	2658,7	77,3	87	96,3

Untuk hasil pengujian skenario 5 seperti yang terlihat pada tabel 4.10 bahwa luas area 500x500 meter memiliki *network lifetime* tertinggi dibandingkan luas area yang lain nya hal ini dikarenakan semakin luas area simulasi dengan jumlah node yang sama maka jumlah transmisi di jaringan juga akan berkurang, dan *latency* yang terlama hal ini dikarenakan semakin luas area simulasi maka akan semakin banyak waktu yang dibutuhkan untuk meneruskan paket sampai kesink. Untuk lebih memperjelas maka divisualisasi pada gambar 4.21. *Network lifetime* skenario 5 dan gambar 4.22. *Latency* skenario 5.



Gambar 4.21. *Network Lifetime* Skenario 5

Berdasarkan gambar 4.21 terdapat kecenderungan kenaikan *network lifetime* berbanding lurus dengan peningkatan luas area. Pada skenario ini luas area 500x500 memiliki *network lifetime* tertinggi yaitu 389,4 menit lebih lama dibandingkan dengan luas area 100x100. Hal ini disebabkan oleh kepadatan jaringan yang semakin menurun sehingga kesibukan jaringan juga semakin menurun.



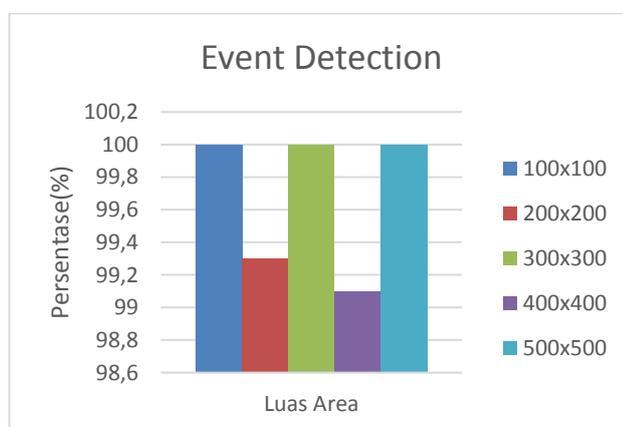
Gambar 4.22. Latency skenario 5

Berdasarkan gambar 4.22 terlihat parameter luas area 100x100 meter memiliki *latency* terendah dibandingkan dengan parameter yang lain nya. Kenaikan *latency* berbanding lurus dengan kenaikan luas area, hal ini terjadi karena semakin kecil luas area suatu area sensing maka semakin dekat jarak antara node ke sink sehingga paket data lebih cepat sampai dibandingkan dengan luas area 500x500 meter karena semakin luas area simulasi maka semakin jauh perjalanan data yang akan dikirim. Pada kepadatan jaringan yang rendah, data sering menemui hole dan harus melalui rute yang cenderung tidak lurus ke arah sink. Sehingga waktu yang dibutuhkan untuk menuju ke sink lebih lama. Luas area 100x100 memiliki *latency* terendah yaitu 55,7 ms lebih rendah dibandingkan dengan luas area 500x500 untuk data dengan prioritas tinggi(P1), sedangkan untuk data dengan prioritas sedang(P2) lebih rendah 65,4 ms dan data dengan prioritas rendah 74,7 ms lebih rendah dibandingkan dengan luas area 500x500.

Tabel 4.11. *Event Detection* dan *Packet Delivery Ratio* Hasil pengujian skenario 5

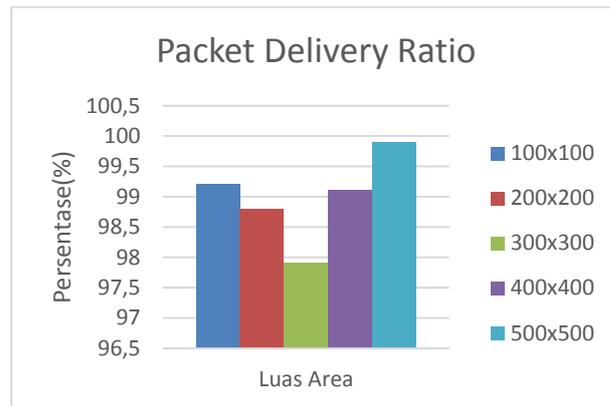
Parameter Luas Area	<i>Packet Delivery Ratio (P1,P2,P3)</i>			<i>Event Detection (P1)</i>		
	Paket dikirim	Paket diterima	Persentase (%)	Event terjadi	Event terdeteksi	Persentase (%)
100x100	11915	11825	99,2	3170	3170	100
200x200	16135	15949	98,8	6582	6539	99,3
300x300	10351	10134	97,9	3549	3549	100
400x400	17272	17123	99,1	10305	10215	99,1
500x500	14271	14260	99,9	7837	7837	100

Untuk hasil pengujian skenario 5 seperti yang terlihat pada tabel 4.11 bahwa luas area 500x500 meter memiliki *event detection* dan *packet delivery ratio* tertinggi dibandingkan luas area yang lain nya. Luas area 500x500 memiliki *event detection* 100% dengan 7837 *event* yang terdeteksi oleh *sink* dari 7837 *event* yang dideteksi oleh *node sensor*, *packet delivery ratio* 99,9% dengan total 14260 paket yang diterima oleh *sink* dari 14271 paket yang dikirim dari *cluster head*. Untuk lebih memperjelas maka divisualisasi pada gambar 4.23. *Event Detection* skenario 5 dan gambar 4.24. *Packet Delivery Ratio* skenario 5.



Gambar 4.23. *Event Detection* Skenario 5

Berdasarkan gambar 4.23 terlihat bahwa *event detection* pada luas tertentu tidak selalu mengalami peningkatan. Hal tersebut terjadi karena pada luas area tertentu seperti luas area 100, 300 dan 500 terdapat beberapa *cluster* yang memiliki sebaran *node* yang rendah dan skenario perubahan temperatur atau prioritas data sehingga transmisi jaringan lebih rendah dibandingkan dengan luas area 200 dan 400. Semakin rendahnya sebaran *node* pada *cluster* tertentu menyebabkan semakin sedikit *event* yang berhasil di deteksi tergantung dari banyak *node* yang melakukan pemindaian.



Gambar 4.24. *Packet Delivery Ratio* Skenario 5

Berdasarkan gambar 4.24, parameter luas area 500x500 memiliki *packet delivery ratio* tertinggi dibandingkan parameter yang lainnya yaitu 2% lebih tinggi jika dibandingkan dengan parameter luas area 300x300.

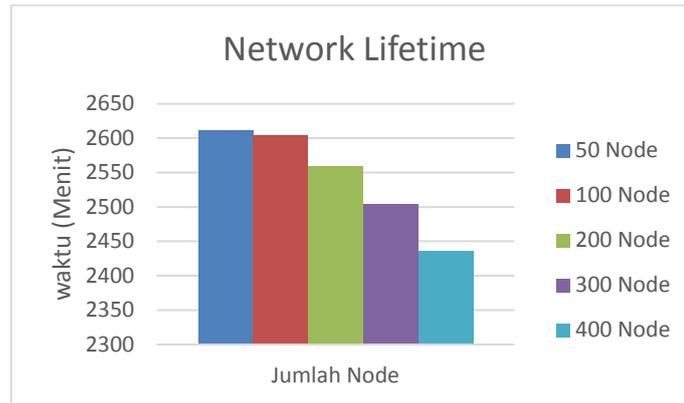
4.3.6. Analisis skenario 6

Pengujian skenario ini dilakukan dengan melakukan ujicoba terhadap *clustered shortest geopath routing protocol* dengan luas area simulasi yang sama yaitu 300x300 dengan melakukan variasi pada jumlah *node* yaitu 100 *node*, 200 *node*, dan 300 *node*. Masing-masing jumlah *node* di simulasikan sebanyak 3 kali dengan luas area yang sama yaitu 300x300. Hasil 3 kali pengujian kemudian akan di rata-rata dan hasil rata-rata tersebut yang kemudian dikomparasi untuk dianalisa. Dari simulasi yang dilakukan diperoleh hasil yang ditampilkan pada tabel 4.12. *Network Lifetime* dan *Latency* Hasil pengujian skenario 6 dan Tabel 4.13. *Event Detection* dan *Packet delivery ratio* Hasil pengujian skenario 6.

Tabel 4.12. *Network Lifetime* dan *Latency* Hasil pengujian skenario 6

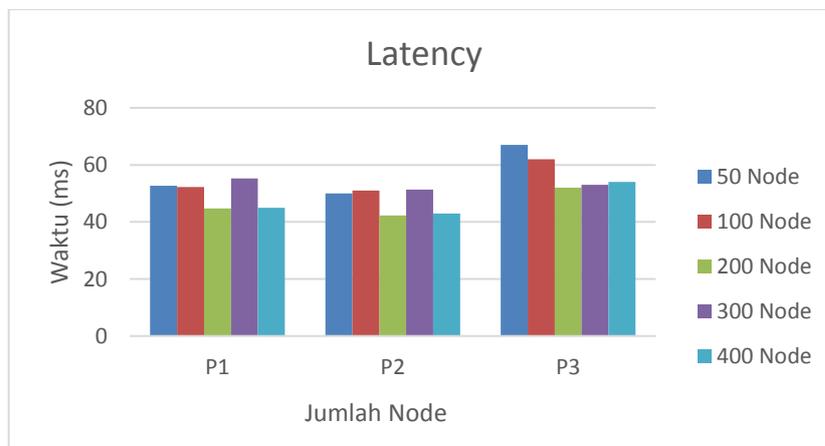
Parameter Jumlah Node	Network Lifetime (Menit)	Latency (ms)		
		P1	P2	P3
50 node	2611,3	52,7	50	67
100 node	2604	52,3	51	62
200 node	2559	44,7	42,3	52
300 node	2504,3	55,3	51,3	53
400 node	2435,7	45	43	54

Berdasarkan tabel 4.12 dengan luas area yang sama, parameter 50 node memiliki network lifetime tertinggi dibandingkan dengan parameter yang lainnya yaitu 2559 menit sedangkan untuk *latency* cenderung menyesuaikan dengan sebaran node pada *cluster* tertentu. Untuk lebih memperjelas maka divisualisasi pada gambar 4.25. *Network lifetime* skenario 6 dan Gambar 4.26. *Latency* skenario 6.



Gambar 4.25. *Network Lifetime* Skenario 6

Pada Gambar 4.25 dengan luas area yang sama, parameter 50 node memiliki *network lifetime* tertinggi yaitu 175,6 lebih lama dibandingkan dengan 400 node. Terlihat bahwa terjadi penurunan *network lifetime* seiring dengan bertambahnya jumlah node. Hal ini dikarenakan kepadatan jaringan yang meningkat sehingga jumlah data yang ditransmisikan ikut meningkat, sehingga konsumsi energi semakin besar.



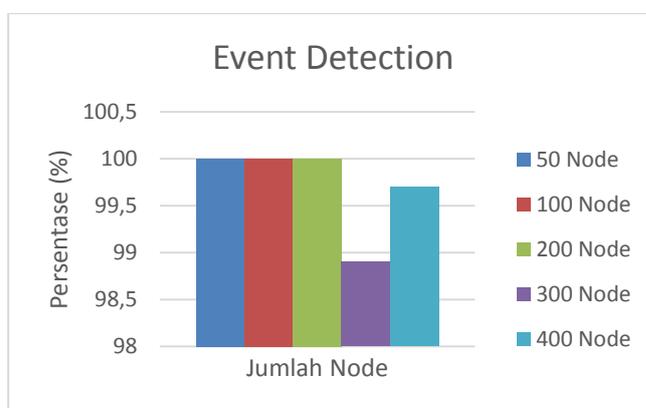
Gambar 4.26. *Latency* Skenario 6

Berdasarkan gambar 4.26 terlihat 200 node memiliki *latency* terendah yaitu 10,6 ms lebih rendah untuk data dengan prioritas tinggi, 9 ms lebih rendah untuk data prioritas sedang dibandingkan dengan 300 node, sedangkan untuk data prioritas rendah 15 ms lebih rendah dibandingkan dengan 50 node. Hal ini terjadi karena dengan luas area 300x300 meter dan jumlah node 200 kepadatan jaringan lebih stabil dibandingkan dengan parameter jumlah node lainnya.

Tabel 4.13. *Event Detection* dan *Packet Delivery Ratio* Hasil pengujian skenario 6

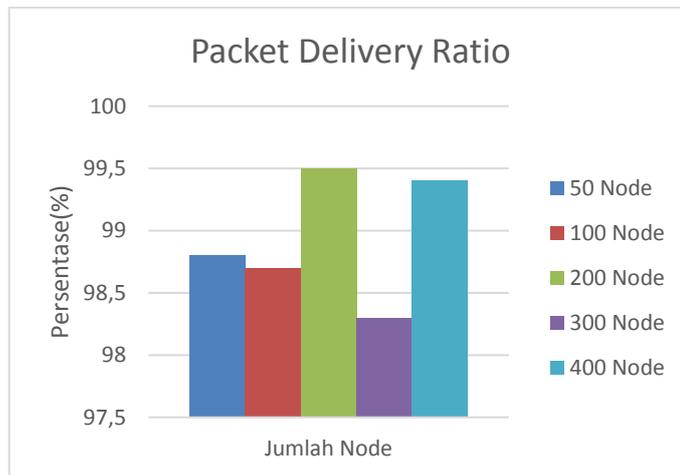
Parameter Jumlah Node	<i>Packet Delivery Ratio</i> (P1,P2,P3)			<i>Event Detection (P1)</i>		
	Paket dikirim	Paket diterima	Persentase (%)	Event terjadi	Event terdeteksi	Persentase (%)
50 node	2195	2169	98,8	900	900	100
100 node	4540	4480	98,7	1618	1618	100
200 node	8631	8585	99,5	3518	3518	100
300 node	13043	12822	98,3	5547	5488	98,9
400 node	17378	17271	99,4	7617	7591	99,7

Berdasarkan tabel 4.13 dengan luas area yang sama, parameter 200 *node* memiliki *event detection* dan *packet delivery ratio* relatif lebih stabil. 200 *node* memiliki *event detection* 100% dengan 3518 *event* yang terdeteksi oleh *sink* dari 3518 *event* yang dideteksi oleh *node* sensor, *packet delivery ratio* 99,5% dengan total 8585 paket yang diterima oleh *sink* dari 8631 paket yang dikirim dari *cluster head*. Untuk lebih memperjelas maka divisualisasi pada Gambar 4.27. *Event Detection* skenario 6 dan Gambar 4.28. *Packet Delivery Ratio* skenario 6.



Gambar 4.27. *Event Detection* Skenario 6

Berdasarkan gambar 4.27 terlihat bahwa *event detection* dipengaruhi oleh jumlah node dan jumlah data yang dikirim. Semakin banyak node dan jumlah data yang terkirim maka *event detection* akan menurun. Hal ini dikarenakan semakin banyak node yang berada pada luas area tertentu menyebabkan kepadatan jaringan semakin meningkat sehingga transmisi di jaringan menjadi lebih tinggi dibandingkan dengan kepadatan jaringan yang rendah. Dari tabel 4.12 dan tabel 4.13 terlihat bahwa untuk luas area 300x300, jumlah node yang stabil adalah 200 *node* yang dapat mendeteksi semua *event* yang terjadi dengan jumlah data yang lebih banyak yaitu 3518 *event* dibandingkan dengan parameter 50 yang mendeteksi 900 *event* dan 100 node mendeteksi 1618 *event*.



Gambar 4.28. *Packet Delivery Ratio* Skenario 6

Parameter 200 node memiliki *packet delivery ratio tertinggi* dibandingkan dengan parameter yang lainnya, yaitu dapat mendeteksi 99,5% dari total 8631 *event* yang terjadi, hal ini terjadi karena 200 node memiliki kepadatan jaringan yang cenderung stabil dibandingkan dengan parameter yang lainnya.

[Halaman ini Sengaja dikosongkan]

BAB V

KESIMPULAN DAN SARAN

Bab ini memaparkan kesimpulan yang dapat diambil berdasarkan pada penelitian yang telah dilakukan. Dalam bab 5 ini diuraikan tentang hal-hal yang perlu dipertimbangkan untuk pengembangan penelitian lebih lanjut. Penjelasan yang lebih terperinci tentang hal-hal tersebut diuraikan pada sub-bab berikut

5.1. Kesimpulan

Berdasarkan hasil ujicoba yang telah dilakukan dapat diambil kesimpulan sebagai berikut:

1. Modifikasi proses seleksi cluster head pada *clustered shortest geopath routing protocol with adaptive path and aggregation* (CSGP-APA) berdasarkan jumlah tetangga terbanyak mampu meningkatkan kinerja CSGP dengan beberapa peningkatan sebagai berikut:
 - a. *network lifetime* meningkat 307,7 menit dibandingkan dengan dengan parameter kombinasi 0.3 sisa energi dan 0.7 jarak ke *cluster center* dan 34 menit dibandingkan dengan paramter sisa energi tertinggi.
 - b. *Event Detection* meningkat 6,9% dibandingkan dengan paramater jarak terdekat ke *cluster center*.
 - c. *Packet Delivery Ratio* meningkat 3,9% dibandingkan dengan paramater jarak terdekat ke *cluster center*.
 - d. *Latency* menurun 5,4 ms lebih rendah dibandingkan dengan parameter jarak terdekat ke cluster center.

2. Modifikasi pada pengiriman data yaitu dengan agregasi data pada *clustered shortest geopath routing protocol with adaptive path and aggregation* (CSGP-APA) mampu meningkatkan kinerja CSGP dengan beberapa peningkatan sebagai berikut:
 - a. *Network lifetime* metode CSGP-APA meningkat 2452 menit dibandingkan dengan metode SGP-WAPA dan 602 menit dibandingkan dengan metode CSGP-WA. Hal ini terjadi karena pada

SGP-WAPA tidak ada kontrol pengiriman data menggunakan cluster head dan agregasi data sehingga trafik jaringan menjadi lebih padat dan konsumsi energi lebih besar.

- b. CSGP-APA memiliki *Event Detection* tertinggi yaitu 61,2% lebih tinggi dibandingkan dengan metode SGP-WAPA, dan 3% lebih tinggi dibandingkan dengan metode CSGP-RCH.
- c. CSGP-APA memiliki *packet delivery ratio* tertinggi yaitu 99,9% dengan total 14260 paket yang diterima dari total 14271 paket data yang dikirim. 59,1% lebih tinggi dibandingkan dengan metode SGP-WAPA dan 2,4% lebih tinggi dibandingkan dengan metode CSGP-WA.
- d. *Latency* metode CSGP-APA menurun pada semua prioritas data yaitu menurun atau lebih cepat 2719,4ms untuk data dengan prioritas tinggi, 2874ms lebih cepat untuk data dengan prioritas sedang, dan 4118ms lebih cepat untuk data dengan prioritas rendah dibandingkan dengan metode SGP-WAPA. Sedangkan jika dibandingkan dengan CSGP-WA, metode ini menurun 15 ms lebih rendah untuk data dengan prioritas tinggi, 6ms untuk data dengan prioritas sedang dan 223ms untuk data dengan prioritas rendah.

5.2. Saran

Dari penelitian yang sudah dilakukan, beberapa hal yang mungkin masih bisa dikembangkan adalah sebagai berikut:

- Pengembangan metode pemilihan adaptive path tanpa melakukan pergantian cluster head.
- Mekanisme penanganan hole yang sering terjadi akibat tingkat kepadatan jaringan yang relatif rendah.

DAFTAR PUSTAKA

- Ardakani, S. P. (2014). *Wireless Sensor Network Routing Protocols for Data Aggregation* Saeid Pourroostaei Ardakani University of Bath.
- Astuti, L. D., & Wibisono, W. (2017). Peningkatan *Network lifetime* pada *Wireless Sensor Network* Menggunakan *Clustered Shortest Geopath Routing (C-SGP) Protocol*. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 4(3), 148–153.
- Brunelli, D., & Rossi, M. (2013). *Wireless Sensor Network*.
- Cayirci, E. (2003). Data *aggregation* and dilution by modulus addressing in *wireless sensor networks*. *IEEE Communications Letters*, 7(8), 355–357. <https://doi.org/10.1109/LCOMM.2003.815663>
- Desai, S., Vyas, T., R.Nair, A., & Jain, H. (2017). Survey on *Cluster Based Data Aggregation in Wireless Sensor Network*. *International Journal of Advanced Research in Computer Science*, 8(5), 311–314.
- Ghica, O. C. (2010). *SIDnet-SWANS Manual*. In *SIDnet-SWANS Manual* (pp. 1–50).
- Gouda, M. G., & McGuire, T. M. (1998). Accelerated *heartbeat protocols*. *Proceedings. 18th International Conference on Distributed Computing Systems*, 202–209. <https://doi.org/10.1109/ICDCS.1998.679503>
- Handy, M. J., Hasse, M., & Timmermann, D. (2002). Low Energy *Adaptive Clustering Hierarchy with Deterministic Cluster head selection*. *Proceeding of IEEE 4th International Workshop on Mobile and Wireless Communications Network*, (September), 368–372.
- Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-Efficient Communication *Protocol for Wireless Microsensor Networks*. In *Published in the Proceedings of the Hawaii International Conference on System Sciences*.
- Jannu, S., & Jana, P. K. (2014). Energy Efficient Grid Based *Clustering and Routing Algorithms for Wireless Sensor Networks*. In *Fourth International Conference on Communication Systems and Network Technologies*. <https://doi.org/10.1109/CSNT.2014.245>
- Mihoubi, M., Rahmoun, A., Lorenz, P., & Lasla, N. (2017). An effective Bat algorithm for *Node localization in distributed wireless sensor network*. *Security and Privacy*, (December), e7. <https://doi.org/10.1002/spy2.7>
- Permatasari, D. I., & Wibisono, W. (2015). Optimasi pemilihan *cluster head* dengan bird-behavior pada lingkungan manet 1,2. *Melek IT Information Technology Journal*, 1(1), 45–50.
- Pratama, i putu agus eka, & Suakanto, S. (2015). *Wireless Sensor Network Jaringan Sensor nirkabel diimplementasikan dalam berbagai bidang seperti :militer, pertanian, kesehatan, bencana alam, bangunan/rumah, transportasi, pendidikan dan berbagai bidang lainnya*.
- Qayyum, B., Saeed, M., & Roberts, J. (2014). Data *Aggregation in Wireless Sensor Networks with Minimum Delay and Minimum Use of Energy : A comparative*

- Study. *2nd BCS International IT Conference on Mobility, Intelligent Networks & Smart Societies*, 1–5.
- Rajagopalan, R., & K.Pramod. (2006). Data Aggregation Techniques in *Sensor Network* : a Survey. *IEEE Communications Surveys & Tutorials*, 8(4), 48–63.
- Randhawa, S., & Jain, S. (2017). *Data Aggregation in Wireless Sensor Networks: Previous Research, Current Status and Future Directions*. *Wireless Personal Communications*. Springer US. <https://doi.org/10.1007/s11277-017-4674-5>
- Saeed, N., Murad, M., Nawaz, M., & Irum, M. (2017). Survey on Single *Path* and *Multipath* Energy Efficient *Routing Protocols* for *Wireless Sensor Networks*. *Journal of Computer and Communications*, 5(5), 1–11. <https://doi.org/10.4236/jcc.2017.55001>
- Sarkar, R., Yin, X. Y. X., Gao, J. G. J., Luo, F. L. F., & Gu, X. D. (2009). Greedy *routing with* guaranteed delivery using Ricci flows. *2009 International Conference on Information Processing in Sensor Networks*, 121–132.
- seggelman R, Tuexen M, W. M. (2012). Transport *Layer Security* (TLS) and *Datagram Transport Layer Security* (DTLS) *Heartbeat* Extension Abstract, 1–9.
- Singh, S. P., & Sharma, S. C. (2015). A survey on *cluster based routing protocols* in *wireless sensor networks*. *Procedia Computer Science*, 45(C), 687–695. <https://doi.org/10.1016/j.procs.2015.03.133>
- Tan, P.-N. (2006). *Introduction to Data Mining*. *Pearson Addison-Wesley*. Pearson Addison-Wesley. [https://doi.org/10.1016/0022-4405\(81\)90007-8](https://doi.org/10.1016/0022-4405(81)90007-8)

BIOGRAFI PENULIS



Rozita, atau yang akrab disapa ita, adalah anak ke empat dari tujuh bersaudara yang lahir di Basilam Baru, Dumai – Riau pada 26 Juni 1987. Penulis menempuh pendidikan dasar dikampung kelahirannya dan pendidikan menengah pertama dan menengah atas di kota Pekanbaru, selanjutnya melanjutkan pendidikan jenjang sarjana di Universitas Islam Indonesia (UII) Yogyakarta di Jurusan Teknik Informatika. Setelah menyelesaikan pendidikan sarjana penulis bekerja di Sekolah Tinggi Ilmu Komputer Dumai pada tahun 2011 selaku dosen luar biasa dan menikah pada tahun 2012. Disamping itu penulis pernah bekerja di Politeknik Negeri Bengkalis dan Akademi Komunitas Negeri Bengkalis hingga akhirnya melanjutkan Pendidikan Magister (S2) di Institut Teknologi Sepuluh Nopember jurusan Teknik Informatika pada tahun 2016. Penulis memiliki minat di bidang Komputasi Berbasis Jaringan (KBJ). Penulis dapat dihubungi melalui thazhy@gmail.com

[Halaman ini Sengaja dikosongkan]

LAMPIRAN 1

Parameter Pemilihan Cluster Head	Network Lifetime (Menit)	Packet Delivery Ratio (P1,P2,P3)			Event Detectin			Latency (ms)		
		Paket dikirim	Paket diterima	Persentase(%)	Event terjadi	Event terdeteksi	Persentase(%)	P1	P2	P3
Tetangga terbanyak	2249	59614	59005	99,0	10308	10218	99,1	91,3	106	107
Energi tertinggi	2215	59214	56930	96,1	10306	9940	96,4	92	105	106
0.9 sisa energi + 0.1 jarak terdekat ke cc	2180,7	58681	56089	95,6	10307	9556	92,7	93	95,3	112,7
0.8 sisa energi + 0.2 jarak terdekat ke cc	2143,7	58288	56046	96,3	10308	9618	93,3	97	107	109
0.7 sisa energi + 0.3 jarak terdekat ke cc	2155	58415	57664	98,7	10308	10210	99,0	91,3	103	106,3
0.6 sisa energi + 0.4 jarak terdekat ke cc	2149	58307	53533	91,8	10308	10207	99,0	91,3	90,7	110
0.5 sisa energi + 0.5 jarak terdekat ke cc	2189,7	58878	56324	95,7	10306	10199	99	91,3	106	107
0.4 sisa energi + 0.6 jarak terdekat ke cc	2187	58813	57277	97,4	10305	10031	97,3	90,7	100,3	107,3
0.3 sisa energi + 0.7 jarak terdekat ke cc	2190,3	53223	51320	96,4	10306	10109	98,1	95,3	105,3	107,3
0.2 sisa energi + 0.8 jarak terdekat ke cc	2181,3	58748	57042	97,1	10307	10100	97,9	91	101,6	107,3
0.1 sisa energi + 0.9 jarak terdekat ke cc	2156,3	58283	55721	95,6	10309	9696	94,1	92,3	102	108
Jarak terdekat ke cc	2211,3	59182	56301	95,1	10306	9500	92,2	96,7	105	106

[halaman ini sengaja dikosongkan]