



TESIS - KI092361

Peningkatan Network Lifetime pada Wireless Sensor Network dengan Menggunakan Clustered Shortest Geopath Routing (C-SGP)

LISTYANTI DEWI ASTUTI
5112201003

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

PROGRAM MAGISTER
KEAHLIAN KOMPUTASI BERBASIS JARINGAN
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

[Halaman ini sengaja dikosongkan]



TESIS - KI092361

**EXTENDING WIRELESS SENSOR NETWORK
LIFETIME USING CLUSTERED SHORTEST GEOPATH
ROUTING PROTOCOL (C-SGP)**

LISTYANTI DEWI ASTUTI
5112201003

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

MASTER PROGRAM
INFORMATICS ENGINEERING DEPARTEMENT
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN TESIS

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

oleh:

LISTYANTI DEWI ASTUTI

NRP: 5112201003

Dengan Judul:

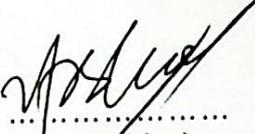
Peningkatan Network Lifetime Pada Wireless Sensor Network Dengan Menggunakan *Clustered Shortest Geopath Routing (C-SGP)*

Tanggal Ujian : Jumat, 12 Mei 2017

Periode Wisuda : September 2017

Disetujui oleh:

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


.....
(Pembimbing)

Royyana Muslim I, S.Kom., M.Kom., Ph.D.
NIP. 19770824 200604 1 001


.....
(Penguji 1)

Tohari Ahmad, S.Kom., MIT., Ph.D.
NIP. 19750525 200312 1 002


.....
(Penguji 2)

Henning Titi Ciptaningtyas, S. Kom, M. Kom.
NIP. 19840708 201012 2 004


.....
(Penguji 3)

Dekan Fakultas Teknologi Informasi


Dr. H. Agus Zainal Arifin, S.Kom., M.Kom.
NIP. 19720809 199512 1 001

[Halaman ini sengaja dikosongkan]

Peningkatan Network Lifetime Pada Wireless Sensor Network Dengan Menggunakan *Clustered Shortest Geopath Routing* (C-SGP)

Nama Mahasiswa : Listyanti Dewi Astuti
NRP : 5112201003
Nama Pembimbing : Waskitho Wibisono, S.Kom., M.Eng., Ph.D.

ABSTRAK

Jaringan sensor nirkabel atau *wireless sensor network* adalah sebuah jaringan yang terdiri dari banyak *sensor node* yang berfungsi untuk memindai fenomena tertentu di sekitarnya. Masing-masing *sensor node* pada umumnya memiliki sumber daya energi berupa baterai, yang memiliki kapasitas terbatas, sehingga diperlukan sebuah protokol untuk meningkatkan *network lifetime* pada *wireless sensor network*.

Pada penelitian ini dikembangkan sebuah protokol *routing* berbasis *cluster*, dengan membagi jaringan menjadi beberapa *cluster* yang masing-masing memiliki *cluster head*. Pembentukan *cluster* dilakukan dengan menggunakan informasi geografis area jaringan untuk membentuk *cluster* berupa segi enam (*hexagon*) dengan panjang dan lebar *cluster* sesuai dengan jangkauan transmisi *cluster head* jika berada tepat di titik tengah area segi enam. Penentuan *cluster head* dilakukan dengan mencari *node* yang memiliki posisi geografis paling dekat dengan titik tengah area *cluster*. Dengan demikian, *cluster head* diharapkan mampu menjangkau seluruh anggotanya dalam satu *hop*.

Pengiriman data dilakukan dengan sistem adaptif, dimana data dengan prioritas rendah dikirim setelah mencapai jumlah data tertentu, dan dikirim setelah melalui proses rata-rata sehingga hanya menghasilkan satu data untuk dikirim. Data dengan prioritas tinggi, dikirim langsung menuju *sink*.

Simulasi dilakukan dengan menggunakan simulator SIDnet SWANS. Hasil simulasi menunjukkan peningkatan *network lifetime* hingga 562.7% dibandingkan dengan protokol *routing Shortest Geopath* asli yang tidak melalui proses *clustering* dan tidak menggunakan mekanisme transmisi adaptif, dan meningkat 5.06% jika dibandingkan dengan protokol *routing Shortest Geopath* yang menggunakan transmisi adaptif saja, tanpa proses *clustering* pada jaringan dengan kepadatan 2222/km².

Kata kunci: *Network Lifetime, clustering, Shortest Geopath Routing, Wireless Sensor Network, SIDnet SWANS*

[Halaman ini sengaja dikosongkan]

Extending Wireless Sensor Network Lifetime using Clustered Shortest Geopath Routing Protocol (C-SGP)

Student's Name : Listyanti Dewi Astuti
Student's ID : 5112201003
Supervisor : Waskitho Wibisono, S.Kom., M.Eng., Ph.D.

ABSTRACT

Wireless sensor network is a type of network consisting hundreds or thousands of sensor nodes. The nodes sense the environment for a specified phenomenon. Each sensor node has limited battery capacity which most of the time cannot be recharged. Thus, researches on extending network lifetime of wireless sensor networks are indispensable.

In this research, proposed a cluster-based routing protocol, which divides sensor network into clusters. Each cluster has a cluster head. Cluster formation is done by forming imaginary hexagons using geographical informations about the network. The size and the number of the hexagons are determined by the radio transmission range of the sensor node, and the wide of the network area. Node which is nearest to the center point of the hexagon is appointed to be cluster head, so the cluster members are hopefully located one hop away from the cluster head.

Data transmissions are controlled by an adaptive system. Low priority datas are only sent after the responsible node already obtained a specified number of datas. The datas, then, are being averaged and wrapped into a single data to be sent to the sink. Very high priority datas are, however, directly sent to the sink without passing the averaging mechanism. This reduces the number of datas being sent, which eventually help extending the network lifetime together with the clustering mechanism.

Simulation was done by using SIDnet SWANS Simulator, and the results showed that the proposed protocol increases network lifetime up to 562.7% compared to the original Shortest Geopath Routing without clustering and adaptive transmission mechanism, and increases up to 25.68% compared to the original Shortest Geopath Routing without clustering but using only adaptive transmission mechanism.

Keywords: *Network Lifetime, clustering, Shortest Geopath Routing, Wireless Sensor Network, SIDnet SWANS*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Segala puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas limpahan petunjuk-Nya, sehingga penulis mampu menyelesaikan tesis yang berjudul “Peningkatan Network Lifetime pada Wireless Sensor Network dengan Menggunakan *Clustered Shortest Geopath Routing (C-SGP)*”. Terimakasih penulis sampaikan pula kepada:

1. Orang tua penulis, Haryanto dan Sri Gendro Winarti, yang senantiasa membimbing dan berdoa, serta mendukung dalam segala aspek kehidupan penulis
2. Pembimbing, Bapak Waskitho Wibisono, S.Kom., M.Eng., Ph.D., yang juga merupakan dosen wali penulis, yang telah membimbing dan membantu penulis, serta memberi kepercayaan, nasehat, solusi, dan banyak sekali dukungan, sehingga penulis dapat menyelesaikan perkuliahan di program studi pascasarjana Teknik Informatika.
3. Bapak Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D., Bapak Tohari Ahmad S.Kom., MIT., Ph.D., dan Ibu Henning Titi Ciptaningtyas, S. Kom, M. Kom., selaku dosen penguji Tesis ini dan juga telah banyak memberikan saran untuk penyempurnaan Tesis ini.
4. Bapak Ibu Dosen program studi S2 Teknik Informatika yang telah memberi bimbingan perkuliahan sehingga menambah wawasan dan pengetahuan penulis
5. Seluruh teman-teman angkatan 112 yang telah memberi bantuan dan dukungan moral dan material
6. 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, 10 Mei 2017

Penulis

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

| | |
|--|-----------|
| Halaman Judul | i |
| Lembar Pengesahan Tesis | v |
| Abstrak | vii |
| Kata Pengantar | xi |
| Daftar Isi | xiii |
| Daftar Gambar | xv |
| Daftar Tabel | xvii |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Perumusan Masalah | 3 |
| 1.3 Batasan Masalah | 3 |
| 1.4 Tujuan dan Manfaat Penelitian | 3 |
| 1.5 Kontribusi Penelitian | 4 |
| BAB II KAJIAN PUSTAKA | 5 |
| 2.1 <i>Wireless Sensor Network</i> | 5 |
| 2.2 <i>Shortest Geopath Routing Protocol (SGP)</i> | 7 |
| 2.3 <i>Clustering pada Wireless Sensor Network</i> | 9 |
| 2.4 <i>Network Lifetime</i> | 12 |
| 2.5 <i>SIDnet SWANS (Simulator and Integrated Development Platform for Sensor Networks Applications)</i> | 13 |
| 2.6 <i>Heartbeat Protocol</i> | 16 |
| BAB III METODE PENELITIAN | 19 |
| 3.1 Rancangan Penelitian | 19 |
| 3.1.1 Studi Literatur | 19 |
| 3.1.2 Desain Sistem | 19 |
| 3.1.2.1 Gambaran Umum Desain Sistem | 20 |
| 3.1.2.2 <i>Heartbeat Protocol</i> | 21 |
| 3.1.2.3 Pembentukan <i>Cluster</i> | 22 |
| 3.1.2.4 Algoritma <i>Clustered Shortest Geopath Routing (C-SGP)</i> | 29 |
| 3.1.2.5 Penentuan Prioritas Pesan | 31 |
| 3.1.3 Pengujian sistem | 32 |
| 3.1.3.1 Lingkungan Uji Coba | 32 |
| 3.1.3.2 Analisis Pengujian | 32 |
| 3.1.3.3 Parameter Uji | 33 |
| 3.1.4 Evaluasi Kinerja | 35 |
| 3.2 Rancangan dan Jadwal Kegiatan Penelitian | 35 |

| | |
|--|----|
| BAB IV HASIL DAN PEMBAHASAN | 37 |
| 4.1 Tahap Implementasi | 37 |
| 4.1.1 Modifikasi Pada <i>Network Layer</i> | 38 |
| 4.1.2 Modifikasi Pada <i>Application Layer</i> | 41 |
| 4.1.3 Modifikasi Pada <i>Driver Simulator</i> | 42 |
| 4.2 Tahap Uji Coba | 44 |
| 4.2.1 Skenario Pengujian | 44 |
| 4.2.3 Input Parameter Pengujian dan Eksekusi Simulator | 45 |
| 4.3 Hasil dan Analisis | 49 |
| 4.3.1 Hasil dan Analisis Skenario 1 | 49 |
| 4.3.2 Hasil dan Analisis Skenario 2 | 61 |
| 4.3.3 Hasil dan Analisis Skenario 3 | 63 |
| BAB V KESIMPULAN DAN SARAN | 67 |
| 5.1 Kesimpulan | 67 |
| 5.2 Saran | 68 |
| DAFTAR PUSTAKA | 69 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 Topologi Wireless Sensor Network dan Arsitektur Node ... | 5 |
| Gambar 2.2 Komponen Sensor Node (Jamal & Kamal, 2004) | 6 |
| Gambar 2.3 (Jamal & Kamal, 2004) | 7 |
| Gambar 2.2 SGP Dengan Teknik <i>Greedy Routing</i> | 7 |
| Gambar 2.3 Permasalahan yang Dapat Terjadi pada SGP | 8 |
| Gambar 2.4 <i>FACE routing</i> | 9 |
| Gambar 2.5 Jaringan yang Dibagi Menjadi Beberapa <i>Cluster</i> dan Memiliki <i>Cluster Head</i> | 9 |
| Gambar 2.6 <i>Cluster</i> pada Sebuah WSN | 10 |
| Gambar 2.7 Contoh Skenario LEACH yang Baik dan Buruk | 11 |
| Gambar 2.11. Energy Efficient Grid Based Clustering (EEGC) | 11 |
| Gambar 2.8. GUI SIDnet SWANS: Pengguna dapat berinteraksi dengan <i>node</i> dan mengirimkan <i>query</i> tertentu | 13 |
| Gambar 2.9 Komponen SIDnet SWANS | 14 |
| Gambar 2.10 Arsitektur SIDnet SWANS yang Berada di atas JiST/SWANS | 15 |
| Gambar 3.11 Skema Rancangan Penelitian | 19 |
| Gambar 3.2 Skema <i>Clustered Shortest Geopath Routing</i> | 20 |
| Gambar 3.3 Diagram Alir Protokol <i>Heartbeat</i> | 22 |
| Gambar 3.4 Bentuk <i>Cluster</i> | 23 |
| Gambar 3.5 Tata Letak <i>Cluster</i> | 24 |
| Gambar 3.6 Penentuan Koordinat Titik Tengah Cluster | 25 |
| Gambar 3.7 Diagram Alir Proses Pembentukan Cluster..... | 27 |
| Gambar 3.8 Proses Penentuan <i>Cluster Head</i> | 28 |
| Gambar 3.9 Protokol Routing C-SGP | 31 |
| Gambar 4.1. Modul <i>layer</i> pada Simulator SIDnet SWANS | 36 |
| Gambar 4.2 Ilustrasi Alur Pesan Pada <i>Network Layer</i> | 36 |
| Gambar 4.3. Modifikasi <i>method handleMessageDataValue</i> | 38 |
| Gambar 4.4. Penentuan Method Untuk Menangani Pesan | 39 |
| Gambar 4.5 Detail Penambahan <i>Method</i> untuk Pencarian <i>Cluster Head</i> ... | 39 |
| Gambar 4.6. <i>Pseudocode</i> Untuk Membuat Skenario Perubahan Suhu..... | 40 |
| Gambar 4.7 Modifikasi <i>Method sensing</i> Untuk Mengatur Pengiriman Data | 41 |
| Gambar 4.8 Pembentukan <i>Node</i> dan <i>Properties</i> -nya | 42 |
| Gambar 4.9 Pencarian Titik Tengah <i>Cluster</i> | 43 |

| | |
|--|----|
| Gambar 4.10 Skenario Perubahan Temperatur..... | 44 |
| Gambar 4.11 Penentuan <i>Run Argument</i> di <i>Project Properties</i> | 45 |
| Gambar 4.12 Tampilan Simulator..... | 47 |
| Gambar 4.13 Tampilan Terminal Simulator..... | 48 |
| Gambar 4.14 GUI Aplikasi SIDnet..... | 48 |
| Gambar 4.15 Grafik Perbandingan <i>Network Lifetime 100 Nodes, 300x300 m</i> (1111/km ²)..... | 51 |
| Gambar 4.16. Penurunan Rata-Rata Energi (C-SGP dan SGP-adaptive) pada Kepadatan 1.111 nodes/km ² | 51 |
| Gambar 4.17 <i>Packet Delivery Ratio 100 Nodes, 300x300 m</i> (1111/km ²)... .. | 52 |
| Gambar 4.18 <i>Network Latency 100 Nodes, 300x300 m</i> (1111/km ²)..... | 53 |
| Gambar 4.19. Perbandingan <i>Network Lifetime</i> dengan Kepadatan Jaringan 2222/km ² | 54 |
| Gambar 4.20 <i>Packet Delivery Ratio</i> Dengan Kepadatan 2222/km ² (200 nodes, 300x300 m) | 55 |
| Gambar 4.21 <i>Latency</i> Dengan Kepadatan 2222/km ² (200 nodes, 300x300 m)..... | 55 |
| Gambar 4.22 Perbandingan <i>Network Lifetime</i> pada Kepadatan 3333/km ² (300 nodes, 300x300 m) | 56 |
| Gambar 4.23 Perbandingan <i>Packet Delivery Ratio</i> pada Kepadatan 3333/km ² (300 nodes, 300x300 m) | 58 |
| Gambar 4.24 Perbandingan <i>Latency</i> pada Kepadatan 3333/km ² (300 nodes, 300x300 m) | 59 |
| Gambar 4.25 Perbandingan <i>Average Energy Left C-SGP non Adaptive</i> dan <i>SGP non Adaptive</i> (Kepadatan 2222/km ²) | 60 |
| Gambar 4.26 Perbandingan <i>Network Lifetime C-SGP</i> dengan Variasi Luas Area | 61 |
| Gambar 4.27 Perbandingan <i>Packet Delivery Ratio C-SGP</i> dengan Variasi Luas Area | 62 |
| Gambar 4.28 Perbandingan <i>Latency C-SGP Adaptive</i> dengan Variasi Luas Area..... | 62 |
| Gambar 4.29 <i>Network Lifetime</i> Skenario 3 | 63 |
| Gambar 4.30 PDR Skenario 3 | 64 |
| Gambar 4.31 <i>Latency</i> Skenario 3 | 64 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 3.1 Parameter Uji | 33 |
| Tabel 3.2 Jadwal Kegiatan Penelitian..... | 35 |
| Tabel 4.1 Rangkuman Skenario Pengujian..... | 44 |
| Tabel 4.2 Contoh <i>Run Arguments</i> | 46 |
| Tabel 4.3. Hasil Simulasi Skenario I untuk Kepadatan 1111/km ² | 50 |
| Tabel 4.4 Hasil Simulasi Skenario I untuk Kepadatan 2222/km ² | 53 |
| Tabel 4.5 Hasil Simulasi Skenario I pada Kepadatan Jaringan 3333/km ² | 57 |
| Tabel 4.6 Data Hasil Uji Coba Variasi Luas Area (Skenario 2) | 60 |
| Tabel 4.7 Rangkuman Hasil Uji Coba Skenario 3 | 63 |

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Jaringan sensor nirkabel (*wireless sensor network/WSN*) adalah kumpulan dari piranti-piranti sensor nirkabel (*sensor nodes*) yang pada umumnya digunakan untuk merekam data-data mengenai fenomena tertentu dalam suatu wilayah. *Sensor nodes* memiliki kemampuan komputasi dan komunikasi yang terbatas. Pola komunikasi umum pada WSN adalah pengiriman data hasil pemindaian (*sensing*) fenomena kepada *base station* atau sejumlah node yang sudah ditentukan sebagai penghimpun data. Penghimpun data atau *base station* melakukan *request* kepada *sensor nodes*, yang berisi *query* tentang fenomena yang akan dikoleksi, interval *sampling*, dan waktu total *sampling* (Frey & Stojmenovic, 2005).

Sensor nodes memiliki sumber daya energi yang terbatas. Pengisian ulang baterai pada umumnya tidak bisa dilakukan karena posisi *sensor nodes* yang dapat berada pada kondisi wilayah yang ekstrim, seperti lokasi bencana alam. Oleh karena itu, masa hidup WSN (*wireless sensor network life time*) perlu ditingkatkan agar manfaat WSN dapat dirasakan lebih lama.

Geographic routing (GR) atau penentuan rute berdasarkan posisi geografis diperkenalkan untuk mengatasi keterbatasan protokol *routing* berbasis topologi. Protokol *geographic routing* mengandalkan informasi lokasi fisik *node* dalam WSN yang diperoleh dari *location service* (misalnya: GPS). Dengan memanfaatkan informasi posisi geografis, GR tidak perlu melakukan *maintenance routing table*, dan bahkan bisa berjalan tanpa *routing table* sama sekali (Maghsoudlou, St-Hilaire, & Kunz, 2011).

Beberapa protokol GR telah diperkenalkan, misalnya *Greedy Routing* yang dikembangkan oleh Takagi dan Kleinrock (1980) yang menghitung semua jalur berdasarkan jarak Euclidean dari *sensor node* menuju *sink node*, lalu memilih jalur yang paling sesuai. Protokol-protokol terbaru pada umumnya merupakan pengembangan dari protokol *Greedy Routing*, dengan cara menambahkan modifikasi tertentu sesuai pertimbangan *delay*, *throughput*, *energy*, dan lain-lain.

Dari beberapa protokol yang telah dikembangkan, *Shortest Geopath Routing* (SGP) merupakan salah satu protokol yang banyak diteliti pada WSN, karena menjanjikan *packet delivery ratio* yang baik dengan waktu *delay* yang rendah. Selain itu, protokol SGP tidak memerlukan komputasi yang rumit, sehingga cocok untuk diterapkan pada WSN yang memiliki kemampuan komputasi terbatas. (Ghica, Trajcevski, Scheuermann, Bischof, & Valtchanov, 2007)

Beberapa protokol *energy aware routing* untuk WSN sudah banyak dikembangkan, dan salah satu yang memiliki tingkat keberhasilan tinggi adalah algoritma LEACH (*Low-Energy Adaptive Clustering Hierarchy*). (Heinzelman, Chandrakasan, & Balakrishnan, 2000). Algoritma LEACH membagi WSN menjadi beberapa *cluster* yang masing-masing memiliki *cluster head*. *Sensor nodes* mengirimkan data hasil pemindaian ke masing-masing *cluster head*, dan *cluster head* meneruskan data tersebut ke *base station* (Permana, 2011).

Algoritma LEACH tidak menggunakan informasi geografis sebagai dasar penentuan rute, pembentukan *cluster*, maupun pencarian *neighbor nodes*. Namun protokol *geographic routing*, terutama *shortest geopath routing* dapat memanfaatkan kelebihan LEACH, dalam hal ini teknik *clustering*, untuk meningkatkan *network lifetime*. Namun perlu dilakukan beberapa modifikasi teknik *clustering* tersebut agar dapat diimplementasikan pada protokol SGP.

Modifikasi yang dilakukan peneliti diimplementasikan dalam simulator khusus untuk WSN, yaitu SIDnet SWANS. SIDnet SWANS dikembangkan oleh *Electrical Engineering and Computer Science Departement (EECS) Northwestern University, USA*. SIDnet SWANS berjalan di atas arsitektur JiST (*Java in Simulation Time*) dan SWANS (*Scalable Wireless Ad-hoc Network Simulator*) serta bersifat *open source* (Ghica, Trajcevski, Scheuermann, Bischof, & Valtchanov, 2007)

1.4 Perumusan Masalah

Berdasarkan latar belakang yang telah disebutkan, maka rumusan masalah yang ada dalam tesis ini adalah sebagai berikut:

1. Bagaimana menerapkan teknik *clustering* pada protokol *Shortest Geopath Routing* untuk meningkatkan *network lifetime* pada *wireless sensor network* (WSN)?
2. Bagaimana menentukan mekanisme pengiriman data yang optimal untuk meningkatkan reliabilitas pengiriman data pada WSN dengan menggunakan protokol routing CSGP?
3. Bagaimana kinerja dari protokol routing CSGP bila dibandingkan dengan protokol routing SGP?

1.5 Batasan Masalah

Batasan masalah yang ada dalam tesis ini antara lain:

1. Sensor nodes bersifat *stationary* (berada pada posisi yang tetap).
2. Simulator yang digunakan adalah SIDnet SWANS yang berjalan di atas arsitektur JiST/SWANS.
3. Fenomena yang dipindai adalah temperature
4. Sensor nodes memiliki kapasitas baterai yang tidak dapat diisi ulang.
5. Sink nodes memiliki kapasitas baterai dua kali lipat dari sensor nodes, dan tidak dapat diisi ulang
6. Setiap sensor nodes memiliki GPS (*Global Positioning System*).
7. Protokol *neighbor discovery* menggunakan *heartbeat protocol* (RFC 6520) (Seggelmann, Tuexen, & Williams, 2013).
8. Sebaran nodes pada wilayah yang ditentukan bersifat random.

1.6 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah:

1. Meningkatkan *network lifetime* pada WSN dengan memodifikasi protokol *routing SGP* menggunakan teknik *clustering*, menjadi protokol baru yang diberi nama Clustered Shortest Geopath Routing Protokol (CSGP).

2. Menentukan mekanisme pengiriman data yang optimal sehingga bisa meningkatkan reliabilitas pengiriman data protokol CSGP, meliputi *Packet Delivery Ratio* (PDR) dan *Packet Delivery Latency* (PDR).
3. Mengevaluasi kinerja protokol CSGP dibandingkan dengan protokol SGP.

1.7 Kontribusi Penelitian

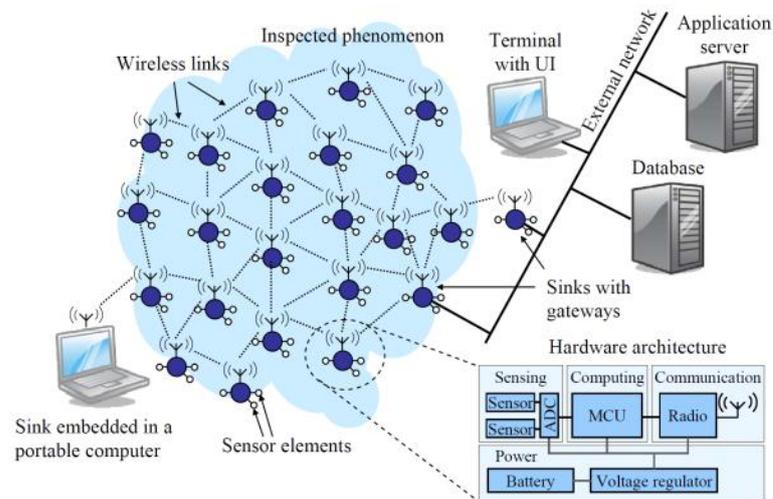
Kontribusi dalam penelitian ini adalah pengembangan protokol *routing* CSGP sehingga mampu meningkatkan *network lifetime* WSN yang pada akhirnya meningkatkan kapasitas pengiriman data dan mengoptimalkan performa WSN dalam menjalankan fungsinya.

BAB 2

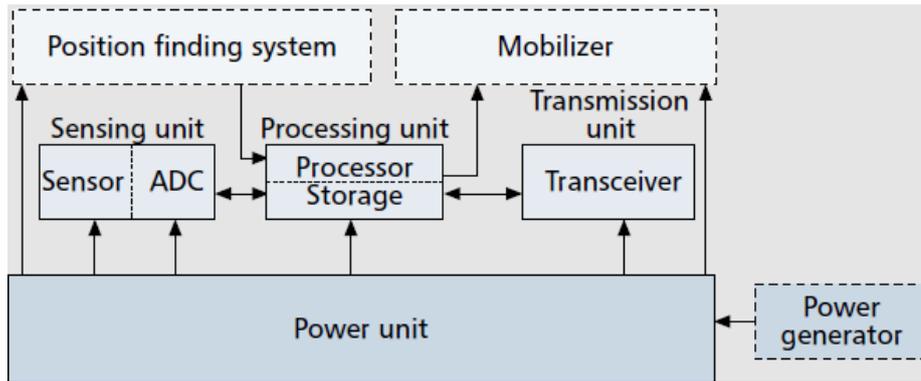
KAJIAN PUSTAKA

2.7 Wireless Sensor Network

Jaringan sensor nirkabel (*wireless sensor network*) terdiri dari banyak sensor yang didistribusikan secara spasial. Sensor-sensor tersebut digunakan untuk memonitor beragam kondisi lingkungan, seperti temperatur, kelembaban udara, cuaca, dan sebagainya. Data-data hasil pengamatan ditransformasikan menjadi sinyal elektronik. Sebuah sensor dilengkapi dengan *radio transceiver*, *microcontroller*, dan sumber energi. Pada umumnya sensor yang digunakan berukuran kecil dan tidak mahal. Performa sensor kecil biasanya tidak sebaik sensor besar yang mahal, namun dengan ukuran kecil dan harga rendah, luas area yang diamati menjadi semakin luas karena sensor yang dipasang bisa menjadi sangat banyak. Sensor-sensor pada jaringan WSN mengirim data ke *base station*, atau dapat juga saling berkomunikasi satu sama lain. *Node* dalam WSN memiliki energi yang terbatas dan seringkali tidak memungkinkan untuk mengisi ulang baterai (Abad & Jamali, 2011).



Gambar 2.12 Topologi Wireless Sensor Network dan Arsitektur Node (Laukkarinen, 2017)



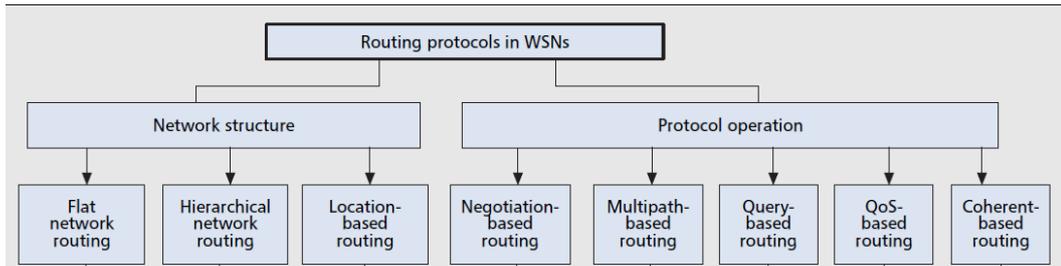
Gambar 13.2 Komponen *Sensor Node* (Jamal & Kamal, 2004)

Sebagaimana jaringan komputer pada umumnya, WSN juga dapat diimplementasikan dalam beberapa topologi, yaitu:

- *Point to Point Topology*
- *Star Topology*
- *Mesh Topology*
- *Hybrid Technology*
- *Tree Topology*

Dalam penelitian ini, peneliti menggunakan topologi *tree*. Pada topologi *tree* terdapat sebuah perangkat yang menjadi root atau menduduki level teratas, dan menjadi pusat utama komunikasi bagi semua node yang terhubung dengannya. Kemudian pada level di bawahnya, terdapat perangkat yang menjadi pusat koneksi pada perangkat lain yang berada di bawahnya yang membentuk topologi *star*. Dalam penelitian ini, level terbawah adalah *cluster member*, level tengah adalah *cluster head*, dan level tertinggi adalah *sink*.

Beberapa protokol *routing* juga dikembangkan untuk diterapkan pada WSN. Protokol-protokol tersebut dapat dibagi menjadi delapan jenis, dilihat dari struktur jaringan dari operasi protokolnya.



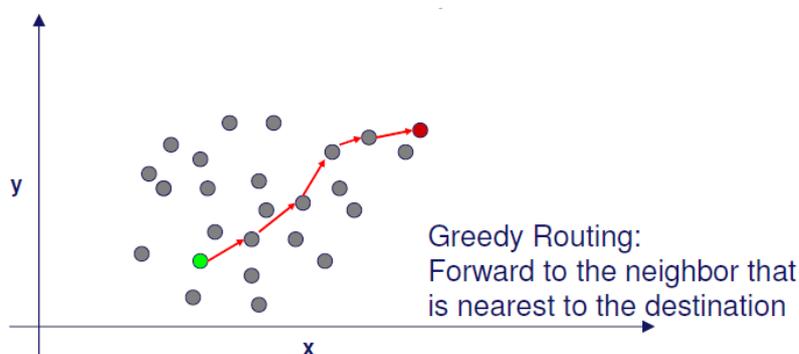
Gambar 2.14 (Jamal & Kamal, 2004)

Dalam penelitian ini, digunakan *Shortest Geopath Routing Protocol (SGP)* yang termasuk dalam kategori *Location-based routing*. Jenis *routing* ini memiliki kelebihan dalam menghemat baterai, karena *node* tidak perlu melakukan pencarian dan perhitungan yang rumit, bahkan beberapa protokol tidak perlu menggunakan *routing table* (Jamal & Kamal, 2004).

2.8 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. (Imielinzki & 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) (Sarkar, Yin, Jie, Feng, & Xianfeng, 2009).

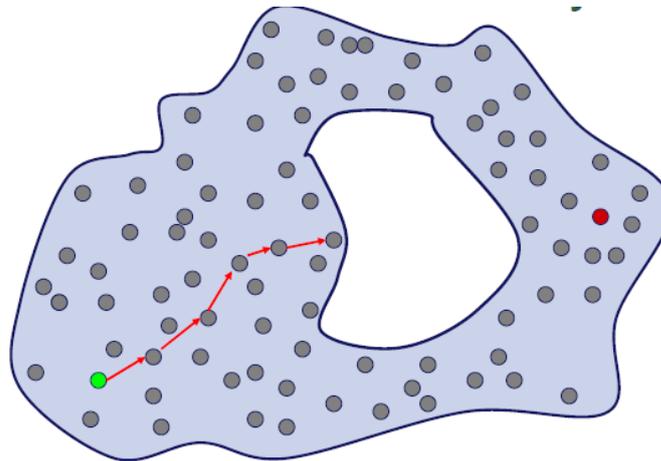


Gambar 2.15 SGP Dengan Teknik *Greedy Routing*. (Sarkar, Yin, Jie, Feng, & Xianfeng, 2009)

Syarat digunakannya protokol SGP adalah:

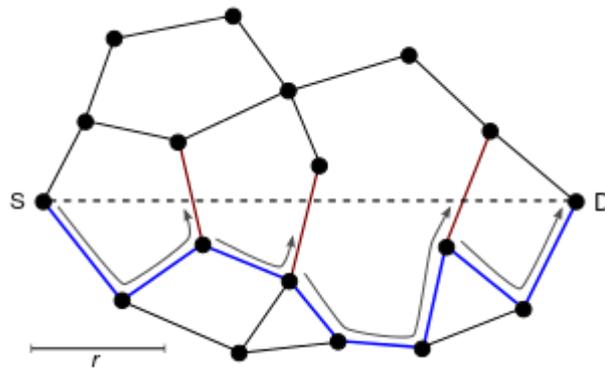
- Masing-masing *node* mengetahui posisi geografisnya
- Masing-masing *node* mengetahui posisi geografis tetangga satu langkahnya
- Tujuan routing didefinisikan sebagai sebuah node pada koordinat posisi dua dimensi, atau sebuah area geografis yang direpresentasikan dengan *polygon* tertutup
- Setiap *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.



Gambar 2.16 Permasalahan yang Dapat Terjadi pada SGP. (Sarkar, Yin, Jie, Feng, & Xianfeng, 2009)

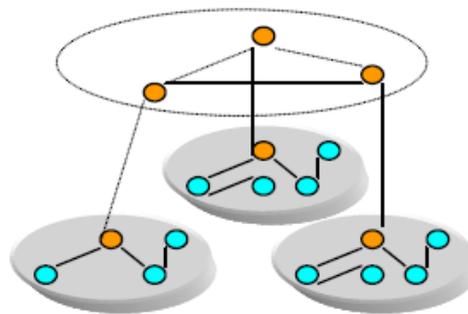
Pada Gambar 2.10, node terakhir yang membawa pesan tidak dapat meneruskan pesan karena dirinya adalah *node* yang terdekat dengan sink dari seluruh tetangganya. Permasalahan ini pada umumnya diatasi dengan teknik *FACE routing*, yaitu ketika node menemui *hole*, maka pesan akan kembali dua atau tiga langkah, dan beralih ke jalur yang lain (tetangga terdekat kedua, ketiga, dan seterusnya atau tetangga terjauh) pada saat menemui garis S-D seperti pada Gambar 2.7.



Gambar 2.17 FACE routing. (wikipedia)

2.9 Clustering pada Wireless Sensor Network

Teknik *clustering* pada WSN umumnya digunakan dengan tujuan penghematan sumber daya energy. Salah satu protokol *clustering* yang populer adalah LEACH. LEACH (*Low-Energy Adaptive Clustering Hierarchy*) adalah sebuah protokol *routing* yang bertujuan untuk meningkatkan konservasi energi. Protokol ini dikembangkan oleh Heinzelman, Chandrakasan, dan Balakrishnan pada tahun 2000. (Abad & Jamali, 2011).

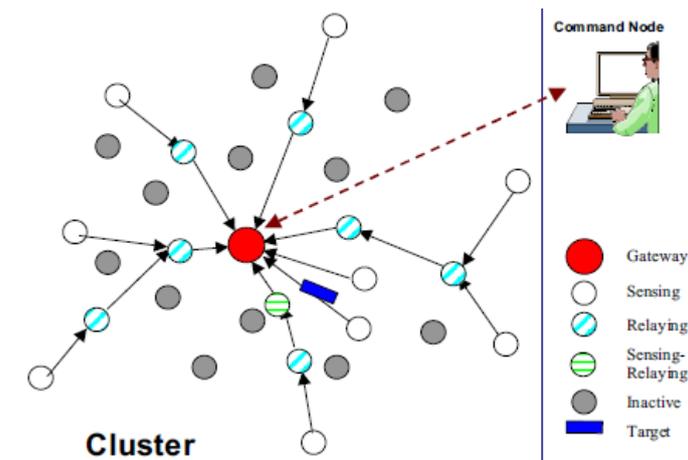


Gambar 2.18 Jaringan yang Dibagi Menjadi Beberapa *Cluster* dan Memiliki *Cluster Head* (CH) (Abad & Jamali, 2011)

Ide utama LEACH adalah untuk membagi jaringan menjadi beberapa *cluster* berdasarkan kekuatan sinyal (*signal strength*) dan menggunakan *cluster head* sebagai *router* untuk mengirim pesan ke *cluster lain* atau ke *base station*. Pemrosesan data juga dilakukan di dalam *cluster head*.

Karena sifatnya dinamis, LEACH membagi waktu menjadi beberapa interval yang sama rata dan memilih *cluster head* setiap interval. Pada awal

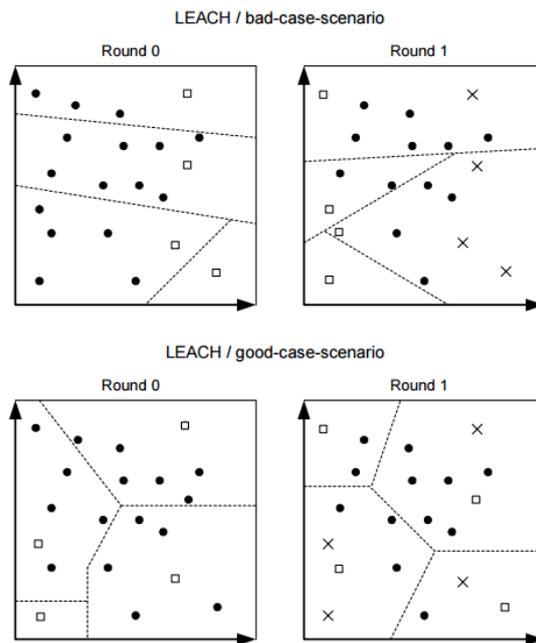
interval, *cluster head* ditentukan secara acak dari beberapa *node* yang memiliki energi di atas level tertentu. Pada interval berikutnya, setiap sensor n mengambil sebuah bilangan acak x sedemikian rupa sehingga $0 < x < 1$, dan membandingkannya dengan threshold tertentu $T(n)$. Jika $x < T(n)$, maka node tersebut menjadi *cluster head* pada interval tersebut. Jika tidak, maka menjadi *member node*.



Gambar 2.19 Cluster pada Sebuah WSN (Youseff, Younis, & Arisha, 2015)

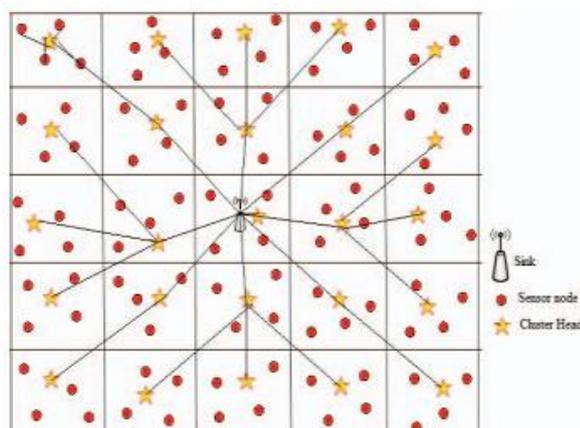
Setelah terpilih menjadi *cluster head*, maka *node* tersebut melakukan *broadcast* kepada node tetangga, dan node tetangga akan menisbatkan diri menjadi *member* kepada *cluster head* yang sinyalnya paling kuat. Setelah *cluster* terbentuk, *cluster head* akan menyusun jadwal *sampling*, dan mengirimkannya ke seluruh *member node*. *Member node* akan melakukan pemindaian sesuai jadwal, dan mengirimkan hasil pemindaian kepada *cluster head*. Menjelang akhir interval, *cluster head* akan mengirim data yang telah dikumpulkan menuju *base station*.

Meskipun mampu menghemat energi jaringan, LEACH juga memiliki kekurangan jika sebaran sensor *node* tidak merata. *Cluster head* yang terpilih bisa berada pada tepi *cluster*, sehingga tidak bisa menjangkau semua *member node* secara langsung. (Handy, Haase, & Timmermann, 2002). Hal ini diperbaiki pada penelitian-penelitian berikutnya, seperti pemilihan *cluster head* menggunakan *fuzzy logic* (Natarajan & Selvaraj, 2014), menggunakan algoritma perilaku fauna semut (Permatasari & Wibisono, 2014), dan lain-lain.



Gambar 2.20 Contoh Skenario LEACH yang Baik dan Buruk. (Sarkar, Yin, Jie, Feng, & Xianfeng, 2009)

Perkembangan penggunaan teknologi GPS pada WSN, memicu perkembangan teknik *clustering* berbasis lokasi dengan membagi area WSN menjadi beberapa *grid* yang relatif dengan luas area WSN. Pemilihan *cluster head* dilakukan dengan mencari total jarak minimal dari *node* dengan tetangga dalam satu *cluster* yang memiliki batas energi tertentu. (Jannu & Jana, 2014) Pengiriman data dilakukan dengan teknik *geographical routing*. Visualisasi pembentukan cluster dengan metode *grid* dapat dilihat pada Gambar 2.11.



Gambar 2.11. *Energy Efficient Grid Based Clustering (EEGC)*
(Jannu & Jana, 2014)

Teknik *grid based clustering* mampu meningkatkan *network lifetime* pada WSN, namun pemilihan *cluster head* dan pembagian area *cluster* masih memungkinkan untuk dioptimasi.

Dari sejumlah penelitian terkait, posisi yang paling tepat dari *cluster head* adalah berada di tengah-tengah *cluster*. Oleh karena itu, teknik *clustering* dengan memanfaatkan posisi geografis seperti yang dikembangkan oleh peneliti, diharapkan mampu menentukan *cluster head* yang memiliki posisi paling tepat.

2.10 *Network Lifetime*

Network lifetime atau masa hidup sebuah jaringan sensor memiliki berbagai definisi. Masing-masing definisi memiliki batasan dan belum ada yang bisa diterapkan pada semua kriteria jaringan. Penelitian awal yang merumuskan *network lifetime* sebagai rentang waktu sejak dimulainya transmisi data yang pertama hingga *node* terakhir mati dikemukakan oleh Tian dan Georganas (2002). Namun pada kenyataannya, sebuah jaringan sensor sudah tidak bisa mengirim data pemindaian, meskipun belum semua *node* mati.

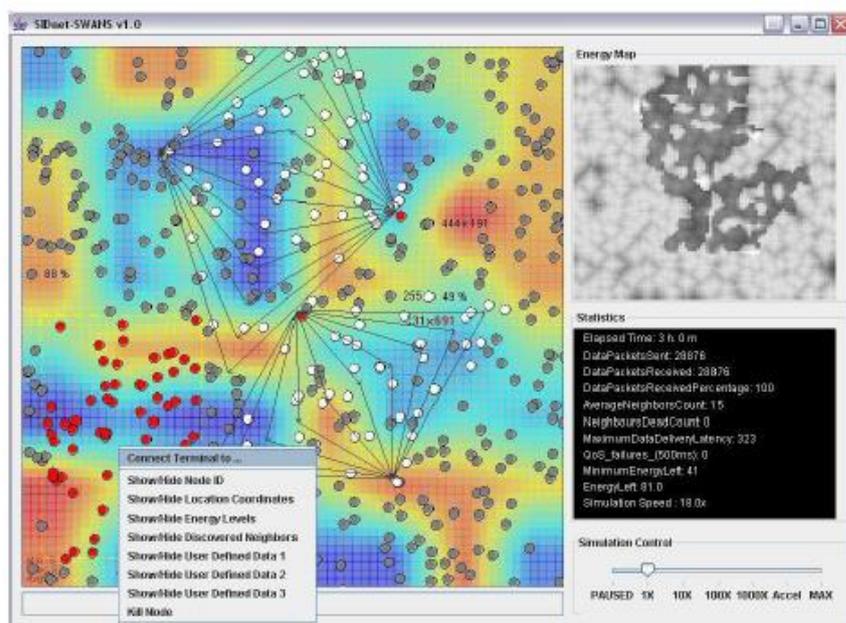
Definisi berikutnya yang lebih realistis dikemukakan seiring dengan berkembangnya metode routing berbasis *cluster*. Soro dan Heinzelman (2005) menyatakan bahwa *network lifetime* adalah waktu hingga *cluster head* yang pertama mati. Definisi ini menjadi tidak relevan ketika protokol *clustering* mulai mampu menangani perubahan topologi dengan mengganti *cluster head*.

Pengertian lain dari *network lifetime* didefinisikan sebagai waktu jaringan mulai mengirimkan data hingga α persen dari total *sensor node* mati (Rajagopalan & Varshney, 2006). Dalam beberapa kasus pada jaringan sensor nirkabel yang sangat mengutamakan lama waktu kemampuan operasi pada salah satu *node* sensor, *lifetime* didefinisikan sebagai waktu *sensor node* yang pertama mengalami kehabisan energi untuk beroperasi. Definisi ini digunakan oleh peneliti dalam mengevaluasi kinerja protokol CSBP, karena skenario yang dijalankan sangat mengutamakan kemampuan operasi pada setiap *sensor node*. Jika salah satu *sensor node* mati, maka data yang diterima oleh *sink* menjadi tidak lengkap karena ada wilayah yang tidak terdeteksi suhunya.

Network lifetime menjadi karakteristik utama pada saat melakukan evaluasi kinerja sebuah WSN. Bahkan ukuran *quality of service* dapat menurun dengan pertimbangan *network lifetime*. *Network lifetime* yang baik berpengaruh secara langsung dalam ketersediaan data hasil pemindaian dan penghematan biaya yang diperlukan untuk pemasangan baterai pada *sensor node* (Dietrich & Dressler, 2009).

2.11 SIDnet SWANS (*Simulator and Integrated Development Platform for Sensor Networks Applications*)

SIDnet SWANS adalah sebuah simulator berbasis Java yang dirancang khusus untuk aplikasi *sensor network*. SIDnet SWANS dikembangkan oleh *Northwestern University* pada tahun 2010. Simulator ini memungkinkan interaksi pengguna dengan sistem melalui *graphical user interface (GUI)* berbasis *Java Swing* (Ghica, Trajcevski, Scheuermann, Bischof, & Valtchanov, 2007)

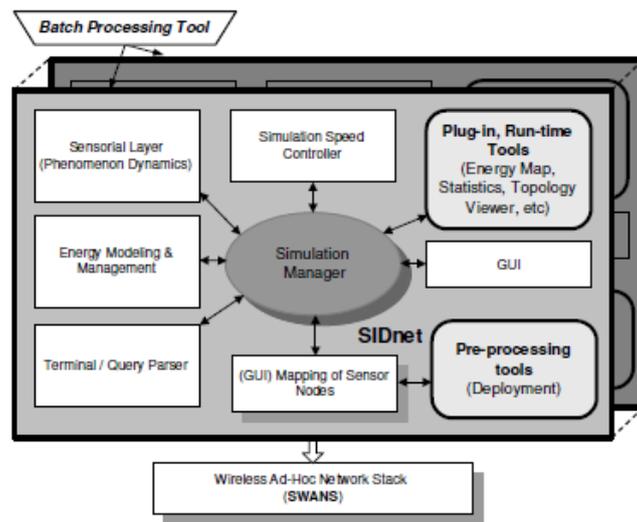


Gambar 2.21. GUI SIDnet SWANS: Pengguna dapat berinteraksi dengan *node* dan mengirimkan *query* tertentu (Ghica, Trajcevski, Scheuermann, Bischof, & Valtchanov, 2007)

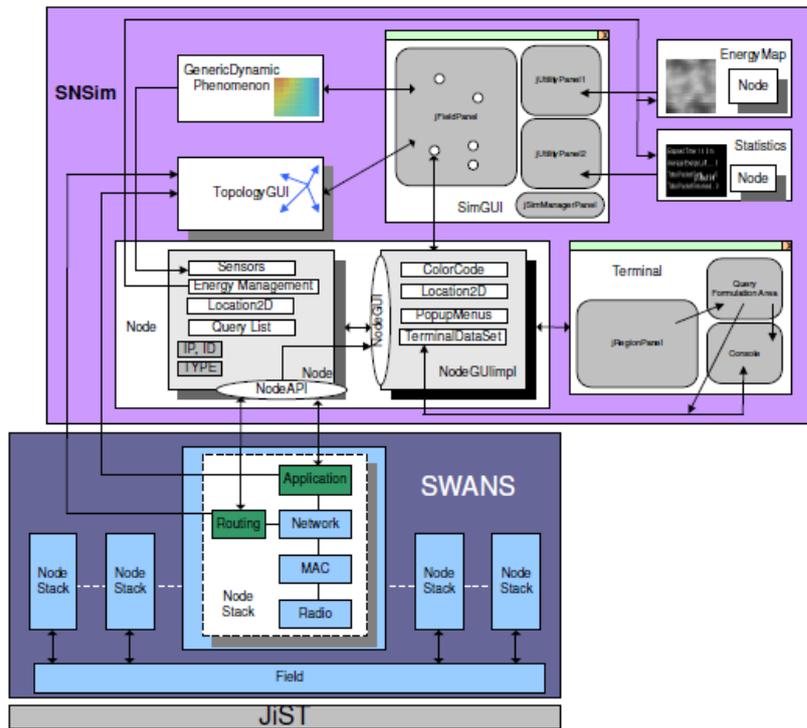
SIDnet SWANS berjalan di atas arsitektur JiST (*Java in Simulation Time*) dan SWANS (*Scalable Wireless Ad-hoc Network Simulator*), sehingga bahasa

pemrograman yang digunakan adalah Java. (Ghica, SIDnet SWANS Manual, 2010). SIDnet SWANS memiliki fitur sebagai berikut:

1. Interaksi pengguna secara grafis terhadap *sensor node*
2. *Visual feedback* dari kondisi jaringan secara *real time*
3. Kontrol terhadap kecepatan simulasi
4. Model konsumsi energi yang terintegrasi
5. Desain yang modular dengan API yang tertata dengan baik



Gambar 2.22 Komponen SIDnet SWANS (Ghica, SIDnet SWANS Manual, 2010)



Gambar 2.23 Arsitektur SIDnet SWANS yang Berada di atas JiST/SWANS (Ghica, SIDnet SWANS Manual, 2010)

Tahap-tahap simulasi pada SIDnet SWANS dapat dijabarkan sebagai berikut:

a. *Deployment Selection*

Pada tahap ini sensor node di-*deploy* dengan salah satu dari empat pilihan *deployment*, yaitu:

- *Single-node deployment*
- *Group-level*
- *uniformly distributed deployment*
- *Flight based deployment*

b. Mengirimkan *request* kepada jaringan

Pada tahap ini pengguna melakukan *submit query* tertentu kepada *node* atau jaringan dengan menggunakan GUI atau dapat menggunakan bahasa *tinySQL*.

c. Struktur *routing*

Pada tahap ini pengguna dapat memilih protokol *routing* yang disediakan (SGP, ZRP, AODV, DSR), atau menyusun sendiri protokol *routing* yang diinginkan.

d. Observasi pada level *node*

Pada tahap ini pengguna dapat melakukan observasi pada *node* untuk mengetahui *node ID*, *energy level*, posisi *node*, dan melakukan penjejakan terhadap paket yang dikirim atau diterima oleh *node*

e. Observasi pada level jaringan

Pada tahap ini pengguna dapat melakukan observasi pada jaringan, meliputi fenomena yang sedang terjadi, pola *routing*, dan berbagai macam statistik (PDR, *latency*, *average energy left*, jumlah *node* hidup dan mati, dan lain-lain)

f. Kontrol kecepatan dan mengubah *run-time condition*

Pada tahap ini pengguna dapat merubah kecepatan simulasi, dan memberikan kondisi-kondisi tertentu pada saat simulasi sedang berjalan.

2.12 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, 2012).

Konsep *heartbeat* diimplementasikan secara luas, dan dikembangkan menjadi berbagai protokol untuk memenuhi kebutuhan masing-masing jenis komunikasi. Dalam penelitian ini, peneliti menggunakan protokol *heartbeat* dari SGP *routing* dalam SIDnet SWANS yang dimodifikasi untuk menyertakan sejumlah informasi mengenai *cluster*. Pada awal simulasi, *node* akan bertukar

pesan *beat* secara periodik untuk saling menemukan tetangga dan koordinatnya. Setelah semua tetangga dan lokasinya ditemukan, maka protokol heartbeat hanya akan dipanggil jika dibutuhkan. Misalnya jika ada *node* yang hampir kehabisan baterai, maka dia akan memberi sinyal kepada tetangganya untuk menemukan jalur yang baru.

[Halaman ini sengaja dikosongkan]

BAB 3

METODE PENELITIAN

3.1 Rancangan Penelitian

Secara umum, penelitian ini diawali dengan studi literatur, pengumpulan data, desain sistem, pengujian sistem, analisis hasil, dan penyusunan laporan.



Gambar 3.24 Skema Rancangan Penelitian

3.1.5 Studi Literatur

Pada tahap ini, peneliti mengumpulkan literatur yang mendukung terlaksananya penelitian. Literatur yang dihimpun dapat berupa buku, artikel dalam jurnal, laporan tugas akhir, laporan thesis, *proceeding*, *conference publication*, dan lain-lain. Sedangkan informasi yang dihimpun berkaitan dengan topic-topik berikut:

1. Karakteristik, fungsi, dan batasan-batasan *wireless sensor network*.
2. Protokol *routing* yang dapat diimplementasikan pada *wireless sensor network*, terutama yang berbasis geografis.
3. Teknik *clustering* pada *wireless sensor network*, terutama yang bertujuan untuk meningkatkan *network lifetime*.
4. Penelitian-penelitian terkait tentang *energy aware routing protocol* pada WSN
5. Dokumentasi dan *manual book* SIDnet SWANS.
6. *Request For Comments* (RFC) mengenai protokol-protokol terkait

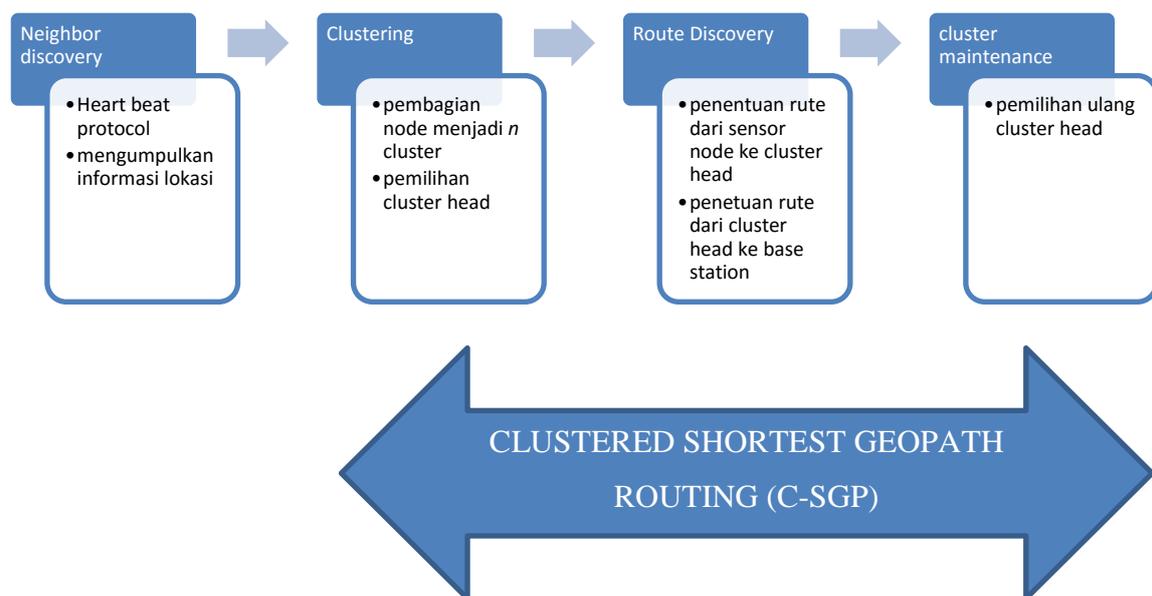
3.1.6 Desain Sistem

Pada tahap ini, peneliti melakukan perancangan sistem meliputi perumusan secara matematis dan perancangan diagram alir.

3.1.6.1 Gambaran Umum Desain Sistem

Secara umum, sistem yang dikembangkan ada modifikasi protokol *routing Shortest Geo-path* (SGP) dengan menambahkan mekanisme *clustering*. Setelah *sensor nodes* di-deploy, maka masing-masing *sensor node* akan menjalankan protokol *heartbeat* (Seggelmann, Tuexen, & Williams, 2013) untuk menemukan *neighbor node* dan mengumpulkan informasi koordinatnya. Lalu *base station* akan membagi wilayah menjadi sejumlah k cluster, sesuai dengan rata-rata luas jangkauan *sensor nodes*. Satu buah *sensor node* akan dipilih menjadi *cluster head* pada masing-masing *cluster*. *Node* yang dipilih harus memiliki level energi di atas *threshold* dan memiliki posisi paling dekat dengan titik tengah *cluster*. Setelah *cluster head* ditentukan, *member sensor nodes* akan menjalankan protokol *Shortest Geopath Routing (SGP)* untuk menentukan rute menuju *cluster head*, dan *cluster head* akan menentukan rute menuju *sink node* atau *base station*.

Setelah rute tersimpan dalam *routing table*, maka *sensor nodes* akan memulai melakukan pemindaian secara periodik sesuai dengan interval *sampling* yang telah ditentukan. Hasil pemindaian akan dikirim ke masing-masing *cluster head*. *Cluster head* mengirimkan data yang diperoleh dari *source node*, menuju *sink node*. Pemilihan ulang *cluster head* dapat dilakukan jika *cluster head* yang aktif mencapai level energi tertentu, dan beralih tugas menjadi *member nodes* biasa.



Gambar 3.2 Skema *Clustered Shortest Geopath Routing*

Pada penelitian terdapat beberapa istilah yang digunakan, yaitu:

1. *Message*: data sensor yang dihasilkan pada source node yang merupakan hasil pemantauan lingkungan dari node tersebut.
2. *Message Space*: merupakan variabel yang menampung *message* sebelum dikirimkan menuju sink node.
3. *Cluster Head*: node yang berfungsi meneruskan data dari *cluster member* ke *sink node*
4. *Cluster*: merupakan kumpulan beberapa member *node* yang memiliki *cluster head* yang sama.
5. *Sensing Query*: permintaan yang dibuat oleh *sink* yang ditujukan ke semua node untuk melakukan pemindaian dengan interval tertentu, dan hasil pemindaian dikirim ke *sink node*.
6. *Abnormal temperature*: temperatur hasil pemindaian yang hasilnya berkisar antara 40 hingga 70 derajat celcius.
7. *Fire temperature*: temperatur hasil pemindaian yang hasilnya lebih dari 70 derajat celcius, yang menandakan terjadi kebakaran.
8. *Adaptive sending*: mekanisme pengiriman data yang diatur kecepatannya sesuai dengan urgensi data.

3.1.6.2 Heartbeat Protocol

Tahap pertama yang dilalui oleh *sensor node* adalah eksekusi protokol *heartbeat*. Protokol ini berfungsi untuk mengumpulkan informasi mengenai *one-hop neighbor*. Informasi yang dikumpulkan berupa jarak *Euclidean* antara *node* dengan *neighbor*. 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 $p(p1, p2)$ dan $q(q1, q2)$ dapat dihitung dengan menggunakan persamaan:

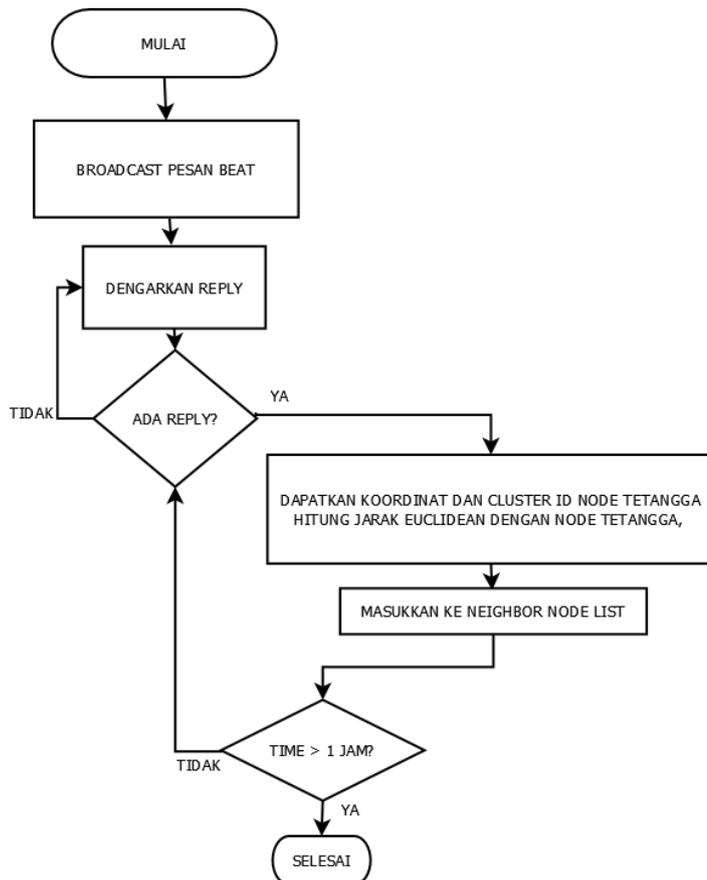
$$d(p, q) = \sqrt{(q1 - p1)^2 + (q2 - p2)^2} \dots\dots\dots(1)$$

Dimana :

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

- $p1$ = posisi p pada sumbu X
- $p2$ = posisi p pada sumbu Y
- $q1$ = posisi q pada sumbu X
- $q2$ = posisi q pada sumbu Y

Jika *node* merupakan *neighbor*, maka informasi mengenai *node* tersebut akan dimasukkan ke dalam sebuah *array neighbor list*. Diagram alir protokol *heartbeat* divisualisasikan di Gambar 3.3.



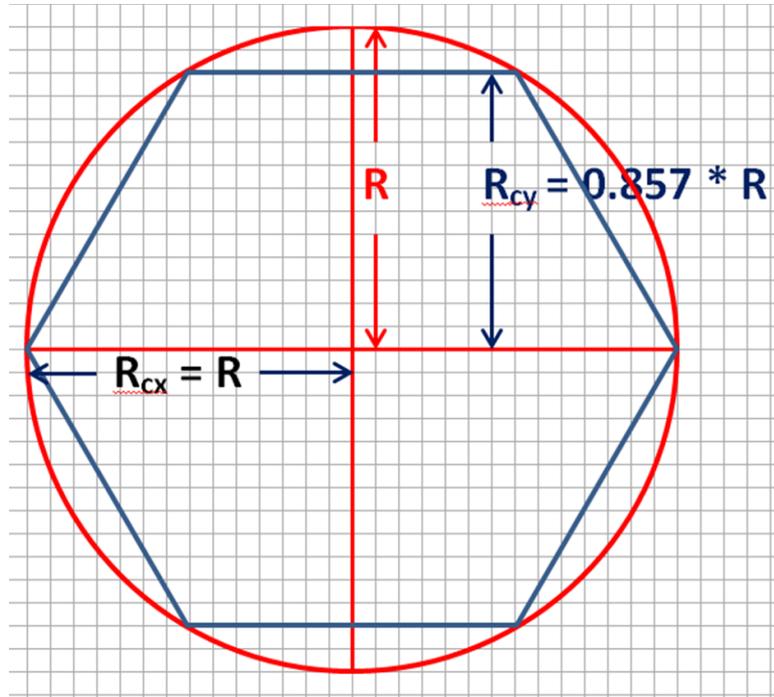
Gambar 3.3 Diagram Alir Protokol *Heartbeat*

3.1.6.3 Pembentukan *Cluster*

a). Bentuk *Cluster*

Masing-masing *node* memiliki area jangkauan transmisi berbentuk lingkaran dengan jari-jari R , di mana R sama dengan jangkauan sinyal terkecil dari *node*, dikurangi konstanta c . Dengan mempertimbangkan hal tersebut, maka *cluster* pada C-SGP diimplementasikan secara geografis dengan membagi wilayah

menjadi sejumlah k cluster, dan masing-masing cluster berbentuk *hexagon* seperti divisualisasikan pada Gambar 3.4.



Gambar 3.4 Bentuk Cluster

Lingkaran berwarna merah adalah area jangkauan transmisi *node* yang sebenarnya, dengan asumsi bahwa *node* berada tepat di tengah lingkaran. Jika *node* tersebut menjadi *cluster head*, maka seluruh tetangganya akan berada satu langkah dari *cluster head*. Sehingga, titik tengah lingkaran merupakan posisi yang paling ideal untuk *cluster head*.

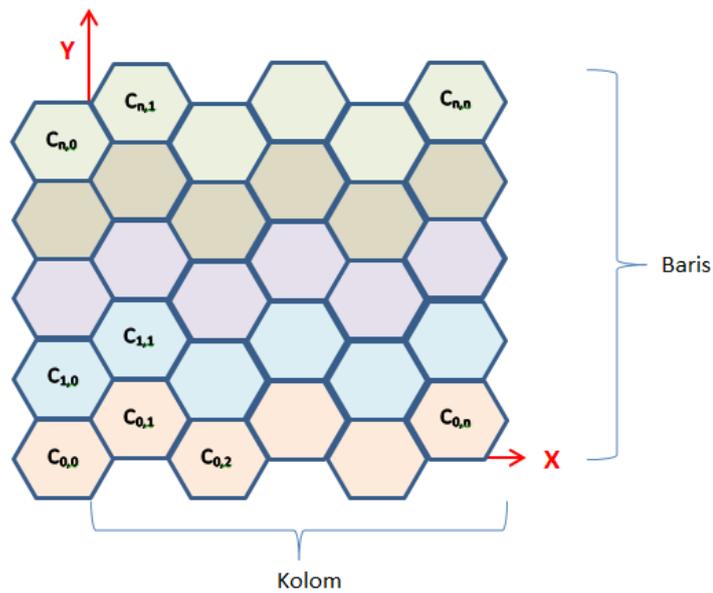
Bangun *hexagon* berwarna biru didesain dengan luas yang lebih kecil dari luas jangkauan transmisi minimal *node* yang sebenarnya. Namun seluruh sudut *cluster* menyentuh sisi lingkaran. Hal ini dilakukan untuk menghindari adanya *node* yang posisinya di luar jangkauan, namun ditentukan untuk menjadi anggota *cluster* tersebut. Karena luas *cluster* lebih kecil dari luas jangkauan transmisi, maka *cluster* diberi nilai R_{cy} (jarak dari titik tengah ke koordinat Y tertinggi dan terendah) dan R_{cx} (jarak dari titik tengah ke koordinat X tertinggi dan terendah).

Nilai R_{cx} sama dengan nilai R , karena sudut samping *cluster* bersentuhan dengan sisi samping lingkaran, sehingga jarak dari titik tengah cluster ke

koordinat X tertinggi dan terendah adalah sama dengan panjang jari-jari lingkaran. Sedangkan nilai R_{cy} lebih rendah dari R , sehingga dihitung ukuran R_{cy} secara proporsional, yaitu 85.7% dari ukuran R . Dengan demikian, maka panjang dan lebar *cluster* adalah dua kali nilai R_{cx} dan R_{cy} .

b). Tata Letak dan Jumlah Cluster

Tata letak *cluster* pada area WSN dapat dilihat pada Gambar 3.5, di mana *cluster* diformasikan menjadi baris dan kolom.



Gambar 3.5 Tata Letak Cluster

Penentuan 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) dan (3).

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

$$c = \frac{p}{2 \times R_{cx}} + 1 \dots\dots\dots(3)$$

Keterangan:

- r = jumlah baris
- c = jumlah kolom
- l = lebar area WSN
- p = panjang area WSN
- R_{cx} = separuh panjang *cluster*
- R_{cy} = separuh tinggi *cluster*

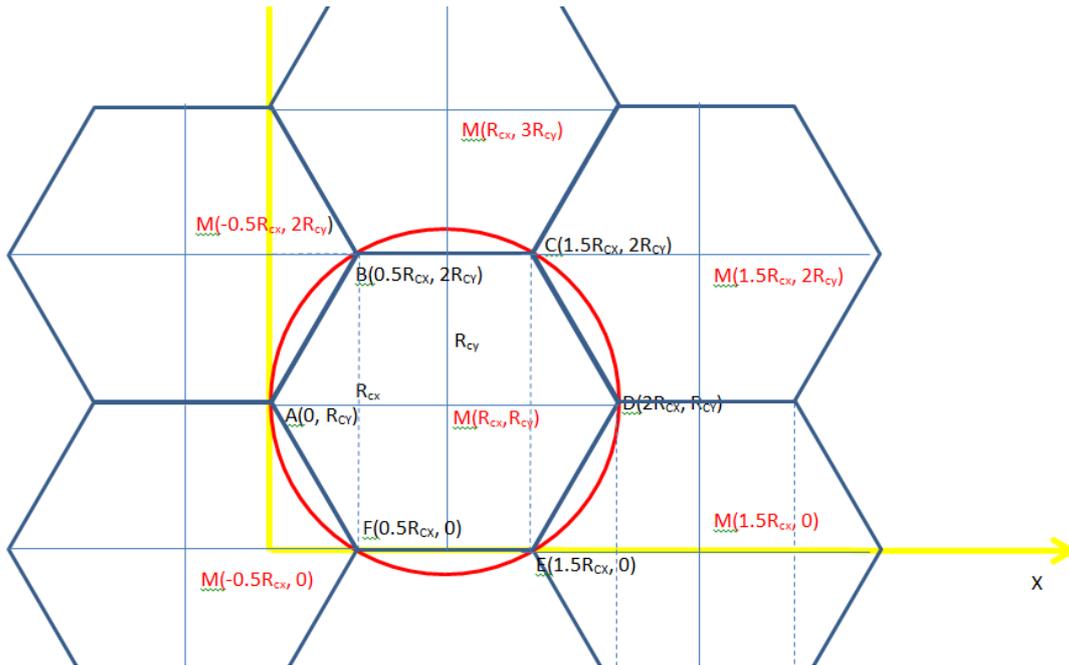
Dari jumlah baris dan kolom, didapat jumlah *cluster* (nk), yaitu:

$$nk = r * c \dots\dots\dots(4)$$

Cluster yang berada paling tepi, dapat memiliki koordinat sudut pada sumbu X dan Y negatif. Namun hal ini tidak berpengaruh, karena *node* akan selalu berada di area koordinat sumbu X dan Y positif.

c). Penentuan Koordinat Titik Tengah *Cluster*

Titik tengah *cluster* merupakan koordinat yang paling strategis untuk penempatan *cluster head*. Namun jika tidak ada *node* yang berada persis di tengah *cluster*, maka *node* yang posisinya paling dekat dengan titik tengah akan dijadikan sebagai *cluster head*. Penentuan koordinat titik tengah (M) untuk beberapa *cluster* pertama dapat divisualisasikan seperti pada Gambar 3.6.



Gambar 3.6 Penentuan Koordinat Titik Tengah Cluster

Dari Gambar terlihat bahwa kenaikan nilai koordinat X untuk setiap titik M pada baris yang sama ada sebesar $1.5 \cdot R_{cx}$ dan bersifat tetap pada baris-baris berikutnya. Sehingga nilai koordinat X untuk setiap titik M dapat dihitung dengan persamaan (5).

$$M(\text{current } x) = M(\text{previous } x + 1.5 \cdot R_{cx}). \dots\dots\dots(5)$$

Sedangkan kenaikan koordinat Y untuk setiap titik M adalah:

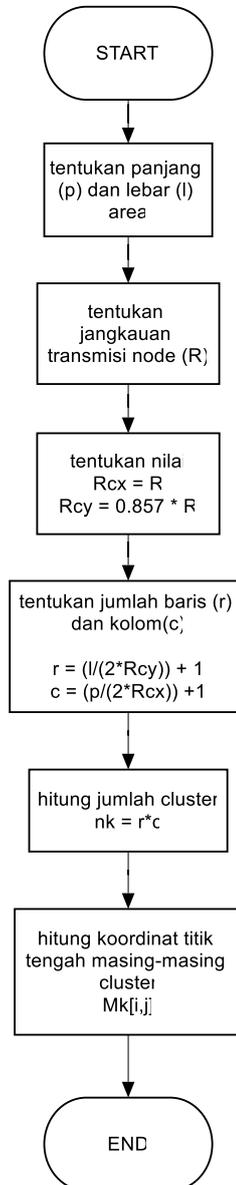
- Pada posisi kolom 0 atau genap, nilai $M(y)$ bernilai sama pada baris yang sama, dan naik sebesar $2 \cdot R_{cy}$ pada kolom yang sama.

$$M(\text{current } y)_{\text{genap}} = M(\text{previous } y + 2 \cdot R_{cy})_{\text{genap}} \dots\dots\dots(6)$$

- Pada posisi kolom ganjil, nilai $M(y)$ bernilai sama pada baris yang sama, dan naik sebesar $2 \cdot R_{cy}$ pada kolom yang sama, namun lebih tinggi sebesar R_{cy} jika dibandingkan dengan nilai $M(y)$ pada posisi kolom genap.

$$M(\text{current } y)_{\text{ganjil}} = M(\text{previous } y + 2 \cdot R_{cy})_{\text{ganjil}} \dots\dots\dots(7)$$

Secara keseluruhan, proses pembentukan *cluster* dapat digambarkan melalui diagram alir seperti Gambar 3.7.

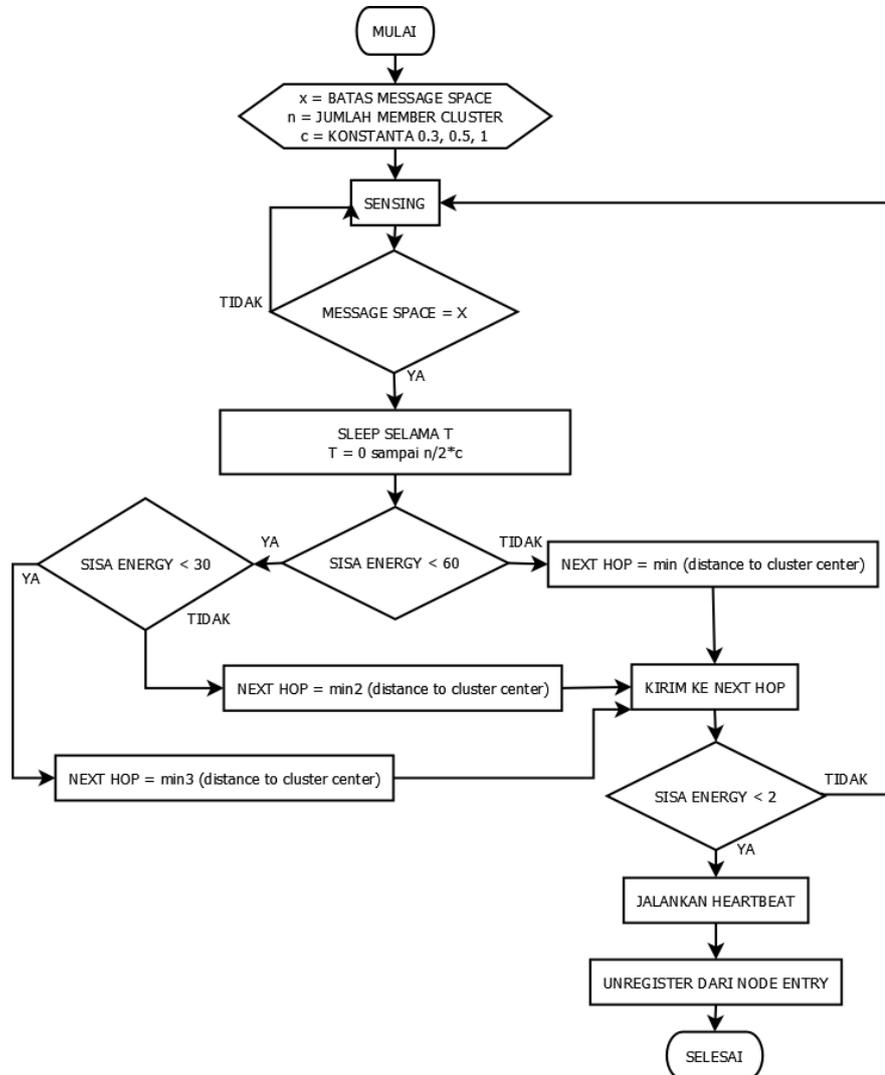


Gambar 3.7 Diagram Alir Proses Pembentukan Cluster

d). Penentuan *Cluster Head*

Setelah didapatkan koordinat titik tengah dari masing-masing *cluster*, maka masing-masing sensor node mulai melakukan sensing dan sending. Sebelum mengirim data, setiap sensor node mencari *node* yang paling dekat posisinya dengan titik M pada masing-masing *cluster* dan memiliki sisa baterai

minimal sebesar *Threshold T*. Semakin dekat posisi *node* terhadap titik tengah *cluster*, maka semakin banyak pula *node* yang dapat dijangkau pada *cluster* tersebut. Proses penentuan *cluster* divisualisasikan pada Gambar 3.8.



Gambar 3.8 Proses Penentuan *Cluster Head*

Node yang terpilih sebagai *cluster head* adalah *node* pada *cluster* yang sama dan memiliki jarak paling dekat dengan titik tengah *cluster*. Pada umumnya *cluster head* akan mengalami beban kerja yang jauh lebih berat jika dibandingkan dengan *member node*. Jika *cluster head* mengalami penurunan energi yang drastis, maka perlu dilakukan pergantian *cluster head*. Pada penelitian ini *cluster head* diganti sebanyak dua kali, yaitu pada saat sisa energi (T) sebesar 60%, dan sisa energi (T) sebesar 30%. Setiap terjadi pergantian *cluster head*, *node* yang terpilih

adalah *node* yang terdekat kedua dan ketiga dengan titik tengah *cluster*. Dengan demikian, diharapkan *cluster head* dapat menjangkau seluruh *member node* yang terdapat pada *cluster*-nya.

3.1.6.4 Algoritma Clustered Shortest Geopath Routing (C-SGP)

Pada algoritma C-SGP, pengiriman data hasil pemindaian dilaksanakan melalui dua tahap. Tahap pertama adalah pengiriman data hasil pemindaian dari *member node* ke *cluster head*. Karena *cluster head* pada dasarnya memiliki jarak satu langkah dari *member node*, maka data dari *member node* bisa langsung dikirim ke *cluster head*. Tahap kedua adalah pengiriman data dari *cluster head* ke *sink (base station)*. Pengiriman data ke *base station* dilakukan dengan menggunakan protokol *routing* SGP, yaitu melalui *node* yang lokasi koordinatnya paling dekat dengan koordinat *base station*.

Pada mekanisme *routing* berbasis *cluster*, sering terjadi tabrakan data karena *member cluster* mengirim data bersama-sama ke *cluster head*. Tabrakan data sangat mungkin terjadi apabila *member cluster* berjumlah banyak. Oleh karena itu, peneliti menjadwalkan pengiriman data dari *member cluster* ke *cluster head*. Penjadwalan ini diadaptasi dari teknik *Smart Sleep Scheduling for Cluster Based M2M Network*, yang dikembangkan oleh Al-Kaseem dan Al-Raweshidy (2016), dimana *node* yang sedang melakukan pemindaian data diberi waktu *sleep* bervariasi sesuai dengan prioritas data dan status *node*.

Penjadwalan dilakukan dengan memberikan instruksi *sleep* kepada *member cluster* sebelum mengirim data. Lama waktu *sleep* dibuat bervariasi untuk menghindari *member cluster* melakukan *transmit* data bersama-sama. Kelemahan dari teknik ini adalah naiknya *delay*. Sehingga dalam penentuan lama waktu *sleep*, harus mempertimbangkan aspek urgensi data yang dikirim. Jika data yang dikirim adalah data berprioritas tinggi, maka lama waktu *sleep* dibuat sekecil mungkin. Sebaliknya, jika data yang dikirim memiliki prioritas rendah, maka lama waktu *sleep* data lebih lama. Perhitungan waktu *sleep* yang digunakan peneliti adalah sesuai persamaan (8) untuk data dengan prioritas tertinggi, sesuai persamaan (9) untuk data dengan prioritas menengah, dan sesuai persamaan (10) untuk data dengan prioritas rendah.

$$T = x/2 * 0.1 \text{ s} \dots\dots\dots (8)$$

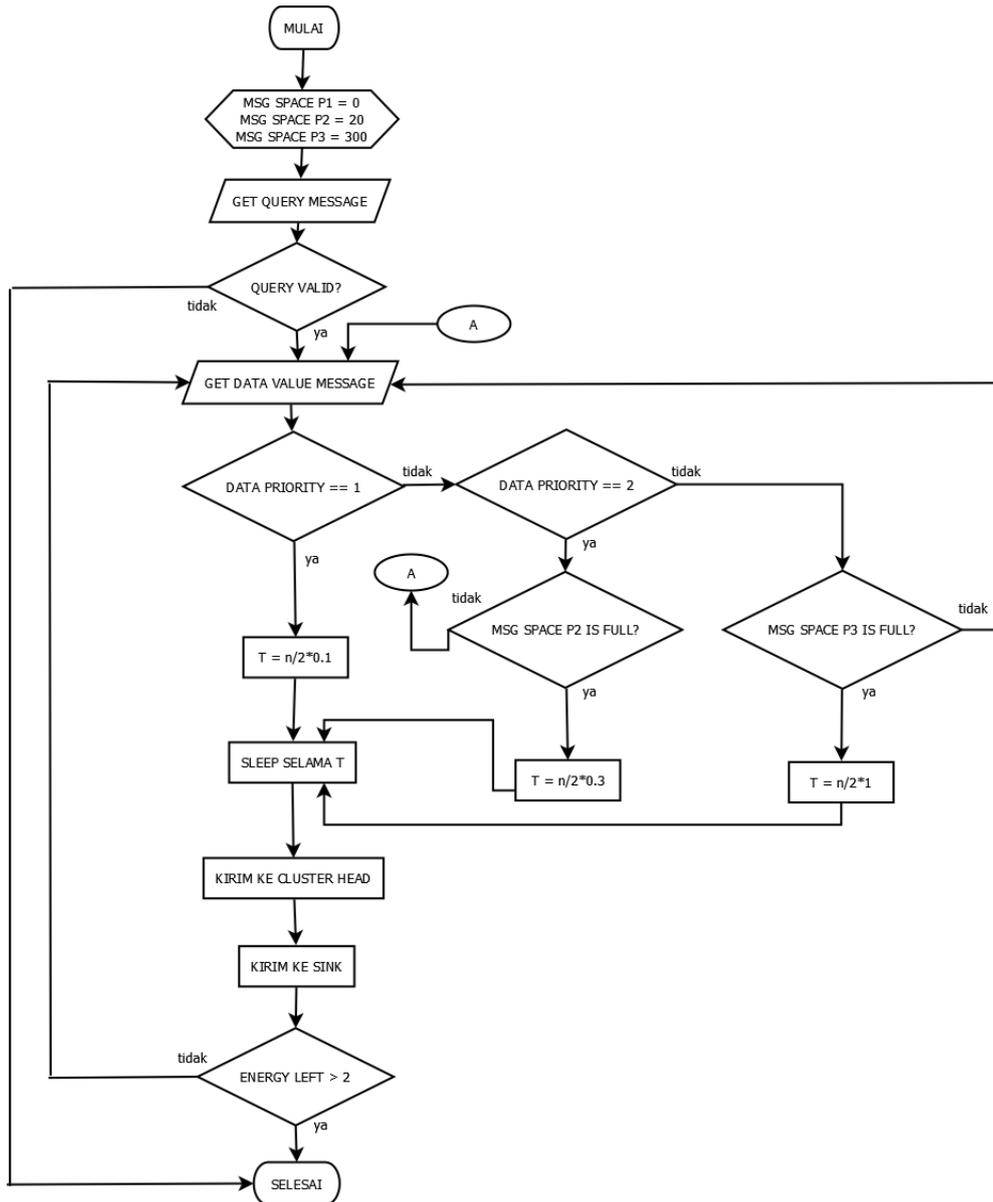
$$T = x/2 * 0.3 \text{ s} \dots\dots\dots (9)$$

$$T = x/2 * 1 \text{ s} \dots\dots\dots (10)$$

Keterangan:

T = lama waktu *sleep* sebelum mengirim data

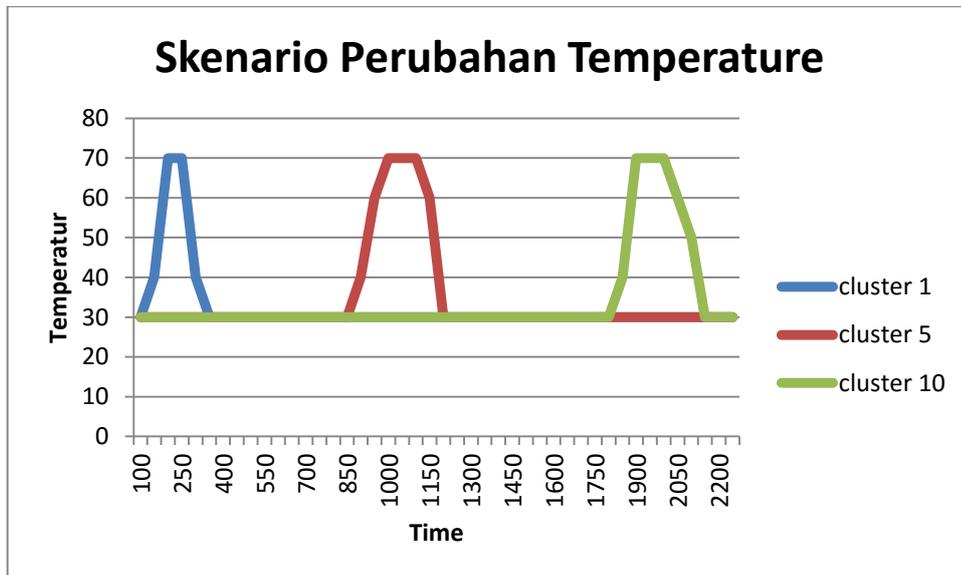
x = bilangan random dari 0 hingga n , dimana n adalah jumlah anggota *cluster*



Gambar 3.9 Protokol Routing C-SGP

3.1.6.5 Penentuan Prioritas Pesan

Kekurangan protokol routing SGP salah satunya adalah terjadinya *bottleneck* yang mengakibatkan penuhnya antrian jaringan, sehingga menurunkan *packet delivery ratio*. Salah satu teknik mengurangi kasus penuhnya antrian jaringan adalah dengan cara menurunkan jumlah data yang dikirim. 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. Untuk data dengan prioritas tinggi, maka langsung dikirim ke *sink*. Untuk data dengan prioritas sedang, *member node* akan mengumpulkan dua puluh data untuk dirata-rata dan mengirimkan data baru hasil rata-rata tersebut ke *cluster head*, untuk diteruskan ke *sink*. Sedangkan data dengan prioritas normal, maka akan dikirim setiap lima belas menit, atau ketika *member node* sudah mengumpulkan tiga ratus data, dan dirata-rata.



Gambar 3.10. Skenario Perubahan Temperature

3.1.7 Pengujian sistem

3.1.7.1 Lingkungan Uji Coba

Uji coba sistem akan dilakukan menggunakan simulator SIDnet-SWANS dengan lingkungan sebagai berikut:

a). Hardware

- CPU Intel Core i5 - RAM 6GB
- Graphics NVIDIA GeForce 3094 MB
- Harddisk Seagate 500 GB

b). Software

- SIDnet SWANS v1.5.6
- JDK 1.5 dan Direct X 10

3.1.7.2 Analisis Pengujian

Berikut adalah parameter-parameter yang akan dianalisa setelah simulasi dilakukan.

a). Packet Delivery Ratio (PDR)

PDR adalah perbandingan antara jumlah paket yang diterima oleh node tujuan dan jumlah paket yang dikirim oleh *node source*. Semakin besar nilai PDR, maka semakin baik nilainya. Rumus untuk menghitung PDR adalah sebagai berikut:

$$PDR = \frac{\Sigma \text{Paket yang diterima}}{\Sigma \text{Paket yang dikirimkan}} (\%)$$

b). *Network Lifetime*

Network lifetime adalah rentang waktu mulai *node* di-deploy hingga *node* pertama mati.

c). *Latency*

Latency adalah rentang waktu yang diperlukan mulai paket dikirim hingga paket diterima oleh *sink*.

3.1.7.3 Parameter Uji

Parameter uji yang digunakan pada penelitian ini bersifat tetap untuk semua skenario, dan langsung diimplementasikan di *source code* simulator. Nilai masing-masing parameter dinyatakan pada Tabel 3.1.

MsgSpace adalah struktur data berupa *list* untuk menampung data hasil sensing. Batas maksimal *msgSpace* bersifat adaptif, yaitu menyesuaikan dengan prioritas data *sensing*. Semakin tinggi prioritas data, maka *msgSpace* semakin kecil, sehingga data *sensing* lebih cepat sampai ke *sink* dan *event* yang terjadi lebih cepat terdeteksi.

Sleep time adalah lama waktu sleep sebelum node member mengirim data ke *cluster head*. *Sleep time* bervariasi sesuai dengan prioritas data dan jumlah *member cluster*. Mekanisme ini diterapkan untuk menghindari pengiriman data yang bersama-sama sehingga mengakibatkan antrian pesan menjadi penuh atau terjadi tabrakan data.

Sampling interval adalah interval waktu dimana node melakukan pemindaian data secara periodik. *Sampling interval* ditentukan oleh admin jaringan melalui GUI interaktif pada simulator.

Tabel 3.3 Parameter Uji

| No | Keterangan | Detail |
|----|--------------------------|---------------------------------|
| 1. | MsgSpace (Adaptive) | |
| | MsgSpace priority 1 | 1 |
| | MsgSpace priority 2 | 20 |
| | MsgSpace priroty 3 | 300 |
| 2. | Radio Bandwidth | 40000 |
| 3 | Sleep time (C-SGP): | |
| | Priority 1 | $0..n/2*0.1 \text{ seconds}$ |
| | Priority 2 | $0..n/2*0.3 \text{ seconds}$ |
| | Priority 3 | $0..n/2*1 \text{ seconds}$ |
| | | n=jumlah anggota <i>cluster</i> |
| 4 | <i>Sampling interval</i> | 3 seconds |

| | | |
|----|--------------------------------------|---|
| 5 | Penempatan <i>node</i> | <i>Random</i> |
| 6 | Konsumsi baterai | |
| | - <i>Active mode</i> | 10 mA |
| | - <i>Sleep mode</i> | 0.5 mA |
| | - <i>Transmit mode</i> | 27 mA |
| | - <i>Receive mode</i> | 10 mA |
| | - <i>Listen mode</i> | 3 mA |
| | - <i>Passive mode</i> | 0.01 |
| 7 | Kapasitas baterai <i>source node</i> | 40 mAh |
| 8 | Kapasitas baterai <i>sink node</i> | 100 mAh |
| 9 | Panjang garis tengah <i>cluster</i> | 60 |
| 10 | Tinggi <i>cluster</i> | $0.987 * 60$ |
| 11 | Data temperatur: | |
| | - Prioritas 1 (suhu kebakaran) | $>70^0$ |
| | - Prioritas 2 (suhu abnormal) | $40^0 - 70^0$ |
| | - Prioritas 3 (suhu normal) | $18^0 - 39^0$ |
| 12 | <i>Event timeline:</i> | |
| | - Menit ke 100-150 | <i>Cluster 1 & 20 = suhu abnormal</i> |
| | - Menit ke 150-200 | <i>Cluster 1 & 20 = kebakaran</i> |
| | - Menit ke 200-250 | <i>Cluster 1 & 20 = suhu abnormal</i> |
| | - Menit ke 900-950 | <i>Cluster 5 & 23 = suhu abnormal</i> |
| | - Menit ke 950-1100 | <i>Cluster 5 & 23 = kebakaran</i> |
| | - Menit ke 1100-1150 | <i>Cluster 5 & 23 = suhu abnormal</i> |
| | - Menit ke 1800-1850 | <i>Cluster 10 = suhu abnormal</i> |
| | - Menit ke 1850-2000 | <i>Cluster 10 = kebakaran</i> |
| | - Menit ke 2000-2100 | <i>Cluster 10 = suhu abnormal</i> |

3.1.8 Evaluasi Kinerja

Pada tahap ini akan dievaluasi PDR dan *network lifetime* pada kondisi jaringan yang berbeda-beda sesuai dengan skenario-skenario berikut:

a. Skenario perbandingan dengan protokol lain

Pada skenario ini, protokol yang sedang dikembangkan dibandingkan hasil uji cobanya dengan protokol lain, yaitu:

- *Shortest Geopath Routing (SGP)*
- *Shortest Geopath Routing using Adaptive Sending Mechanism (SGP-Adaptive)*
- *Clustered Shortest Geopath Routing without Adaptive Sending Mechanism (C-SGP-NonAdaptive)*
- *Clustered Shortest Geopath Routing with Random Clusterhead (C-SGP-RandomCH)*

b. Skenario variasi luas area

Pada skenario ini, jumlah node ditentukan sebanyak 200 node. Namun luas area bervariasi antara 200 m², 300 m², 400 m², dan 500 m².

c. Skenario variasi jumlah node

Pada skenario ini, luas area uji coba tetap, yaitu 300 m². Namun jumlah node bervariasi antara 100, 150, 200, dan 250 node.

3.2 Rancangan dan Jadwal Kegiatan Penelitian

Pada bagian ini diperlihatkan jadwal kegiatan selama berlangsungnya penelitian ini yang dapat dilihat pada Tabel 3.2. Penelitian dibagi menjadi enam tahap, mulai dari studi literatur, pengumpulan data, desain sistem, pengujian system, dan penulisan laporan. Semua tahap dijadwalkan untuk dapat diselesaikan dalam waktu tiga minggu.

Tabel 3.4 Jadwal Kegiatan Penelitian

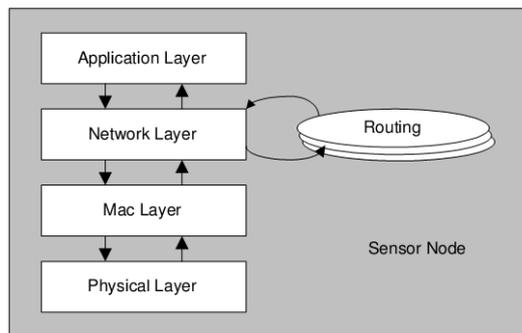
| Kegiatan | Bulan | | | | | | | | | | | |
|-------------------|-------|---|---|---|-------|---|---|---|-------|---|---|---|
| | 1 | | | | 2 | | | | 3 | | | |
| | Pekan | | | | Pekan | | | | Pekan | | | |
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Studi Literatur | ■ | | | | | | | | | | | |
| Pengumpulan Data | | | | | ■ | | | | | | | |
| Desain Sistem | | | | | ■ | | | | | | | |
| Pengujian Sistem | | | | | ■ | | | | | | | |
| Analisa Hasil | | | | | | | | | ■ | | | |
| Penulisan Laporan | | | | | | | | | ■ | | | |

BAB IV

HASIL DAN PEMBAHASAN

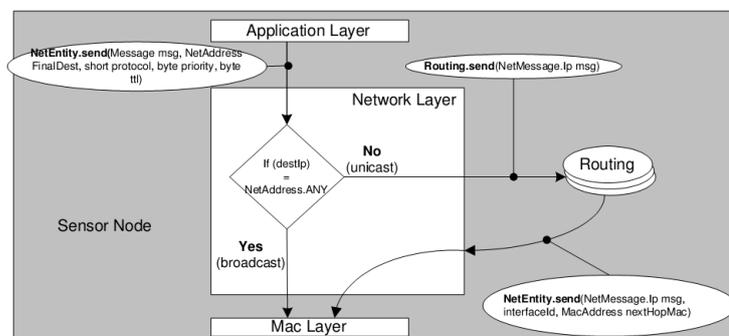
4.2 Tahap Implementasi

Implementasi dilakukan dengan memodifikasi dan menambah modul-modul yang tersedia di simulator SIDnet SWANS. Modifikasi dan penambahan dilakukan pada *application layer* dan *network layer*. Modul *layer* pada simulator SIDnet SWANS merepresentasikan kondisi nyata sensor node seperti yang direspresentasikan pada Gambar 4.1.



Gambar 4.1. Modul *Layer* pada Simulator SIDnet SWANS (Ghica, SIDnet SWANS Manual, 2010)

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. Mekanisme alur pesan antara *Application Layer*, *Mac Layer*, *Network Layer* dan *routing* diilustrasikan pada Gambar 4.2.



Gambar 4.2 Ilustrasi Alur Pesan Pada *Network Layer* (Ghica, SIDnet SWANS Manual, 2010)

Ketika *Network Layer* menerima pesan dari *Mac Layer*, maka akan diperiksa apakah *node* yang bersangkutan merepresentasikan tujuan akhir dari pesan tersebut. Jika tidak, maka *Network Layer* akan meminta protokol *routing* untuk menentukan *next hop* dan mengirim pesan tersebut. Jika *node* yang bersangkutan merepresentasikan tujuan akhir dari pesan tersebut, maka *Network Layer* akan menangani pesan tersebut menggunakan *method Receive*, dan pada umumnya akan mengirim pesan ke *Application Layer*.

Application Layer mengimplementasikan mekanisme agar skenario simulasi dapat bekerja dengan cara:

1. Menangani interaksi pengguna
2. Mengirim permintaan pemindaian dengan kriteria yang ditentukan dan memproses permintaan yang diterima dengan cara melakukan *sampling* terhadap fenomena yang diminta dan mengirimkan hasilnya kembali ke pengguna

Algoritma *routing* menangani pembentukan jalur dari *source node* ke *sink node* yang akan bekerja sama dengan *Network Layer* untuk menangani mekanisme pengiriman data. Sedangkan *driver* simulator berfungsi sebagai *entry point* ke dalam aplikasi SIDNet. Di dalam *driver* dispesifikasikan pembuatan dan karakteristik *sensor node*, algoritma yang diimplementasikan pada masing-masing *layer*, area pemindaian, penempatan *node*, dan lain-lain. Pada intinya, *driver* SIDNet berjalan di atas SWANS (*Scalable Wireless Ad Hoc Network Simulator*, yang berjalan di atas JIST (*Java in Simulation Time*), dimana JIST bekerja di atas JVM (*Java Virtual Machine*).

4.1.1 Modifikasi Pada *Network Layer*

Modifikasi pada *Network Layer* dilakukan dengan menggandakan terlebih dahulu modul *Network Layer* yang secara *default* sudah terdapat dalam SIDNet. Di dalamnya terdapat *class* *ShortestGeoPathRouting.java* yang dimodifikasi menjadi *class* *Csgp.java*. *Class* ini menangani mekanisme *routing* yang diusulkan.

Modifikasi pertama dilakukan pada *method* *handleMessageDataValue*, dimana penentuan *next hop* bagi *member node* adalah menuju *cluster head*. Pada *method* ini penentuan *next hop* dilakukan berdasarkan kriteria sebagai berikut:

1. Jika sisa energi lebih dari 60%, maka *next hop* ditentukan melalui *method* `getFirstCH`.
2. Jika sisa energi lebih dari 30% dan kurang dari 60%, maka *next hop* ditentukan melalui *method* `getSecondCH`.
3. Jika sisa energi kurang dari 30%, maka *next hop* ditentukan melalui *method* `getThirdCH`.
4. Jika pesan sudah melewati *cluster head*, maka status pesan di-*set* menjadi *true*, sehingga mekanisme pengiriman dilakukan dengan *method* `getThroughShortestPath`.

Modifikasi pada *method* `handleMessageDataValue` divisualisasikan pada Gambar 4.3.

| | |
|---|---|
| 1 | <code>Initialize nextHop = getFirstCH();</code> |
| | <code>...</code> |
| 2 | <code>If sisaBaterai < 60% then</code> |
| 3 | <code> nextHop = getSecondCH()</code> |
| 4 | <code>else if sisaBaterai < 30% then</code> |
| 5 | <code> nextHop = getThirdCH()</code> |
| | <code>...</code> |
| | <code>//kirim ke nextHop</code> |
| 6 | <code>sendToLinkLayer(copyOfMsg, nextHop);</code> |

Gambar 4.3. Modifikasi *method* `handleMessageDataValue`

Modifikasi berikutnya adalah baris program untuk menangani pesan yang masuk ke *network layer*. Jika pesan merupakan *query message*, maka akan ditangani oleh *method* `handleQueryMessage`. Jika pesan merupakan data hasil pemindaian, maka akan ditangani oleh *method* `handleMessageDataValue`. Jika pesan tidak dikenal, maka tampilkan notifikasi “*unknown message*”. Detail modifikasi dijabarkan pada Gambar 4.4.

Penentuan *cluster head* dilakukan dengan mencari *neighbor node* dari *node* yang akan mengirimkan data, yang paling dekat posisi geografisnya dengan titik tengah *cluster*. Jika pesan sudah melewati *cluster head*, maka pencarian *next hop* ditentukan oleh *method* `getTroughShortestPath`.

| | |
|---|--|
| | // jika baterai tidak cukup, maka tidak dikirim |
| 1 | If sisaBaterai < 2 % then |
| 2 | Unregister from node entry |
| | |
| | //dapatkan target location |
| 3 | targetLocation = call function msgSGP.getTargetLocation() |
| | |
| | //jika pesan adalah query message |
| 4 | if msgSGP.getPayload() instanceof MessageQuery then |
| 5 | call function handleQueryMessage(msg) |
| | //jika pesan adalah data value |
| 6 | else if msgSGP.getPayload() instanceof MessageDataValue then |
| 7 | call function handleMessageDataValue(msg, targetLocation) |
| 8 | else |
| 9 | write error: "unknown Message" |

Gambar 4.4. Penentuan Method Untuk Menangani Pesan

| | |
|----|--|
| 1 | Function getFirstCH |
| 2 | Initialize jarak |
| 3 | While node entry is member of cluster neighbor linked list |
| 4 | If node entry cluster id = active node cluster id then |
| 5 | If node entry distance to cluster center < jarak then |
| 6 | jarak = node entry distance to cluster center |
| 7 | next hop IP address = node entry ip address |
| | set node entry status = true |
| | |
| 8 | function getSecondCH |
| 9 | Initialize jarak |
| 10 | While node entry is member of cluster neighbor linked list |
| 11 | and node entry status is false |
| 12 | If node entry cluster id = active node cluster id then |
| 13 | If node entry distance to cluster center < jarak then |
| 14 | jarak = node entry distance to cluster center |
| 15 | next hop IP address = node entry ip address |
| 16 | set node entry status = true |

Gambar 4.5 Detail Penambahan *Method* untuk Pencarian *Cluster Head*

4.1.2 Modifikasi Pada *Application Layer*

Modifikasi pada *application layer* meliputi modifikasi pada proses pemindaian serta pengiriman dan penerimaan pesan pada *application layer*. Pada *method* `sensing()` di *class* `ChApp.java` ditambahkan beberapa operasi. Yang pertama adalah skenario *sensing*, dimana pada waktu-waktu tertentu beberapa *cluster* mengalami perubahan kecepatan pengiriman data akibat perubahan temperatur yang menyebabkan prioritas pesan turut berubah.

```
1  If cluster id = 1 or cluster id = 20 then
2      If simulation time > 100 minutes AND
          simulation time <= 150 minutes
3          Set temperature > 40 AND temperature < 50
4          Set message priority = P2
5      Else if simulation time > 150 minutes AND
          simulation time <= 200 minutes
6          Set temperature >= 50 AND temperature < 70
7          Set message priority = P3
...

```

Gambar 4.6. *Pseudocode* Untuk Membuat Skenario Perubahan Suhu

Modifikasi berikutnya sebagaimana tertulis pada Gambar 4.7 adalah mekanisme penanganan pengiriman pesan yang adaptif sesuai dengan prioritas pesan. Jika suhu yang terdeteksi $>70^{\circ}$ C, maka pesan akan langsung dikirim menuju *sink*. Kecepatan pengiriman pesan akan diturunkan jika suhu yang terpindai juga menurun, yaitu hingga 20 data jika suhu $>40^{\circ}$ C, dan hingga 200 data jika suhu $<40^{\circ}$ C. Sebelum dikirim, data hasil *sensing* dirata-rata terlebih dahulu, sehingga yang terkirim adalah satu pesan saja yang mewakili seluruh pesan yang tertampung dalam `msgSpace`.

```

1  if message priority = P1 then
2      if message space > 0 then
3          send message
4          clear message space
5  else if message priority = P2 then `
6      if message space > 20 then
7          calculate average value
8          wrap as msgSGP
9          send msgSGP
10         clear message space
11 else if message priority = P3 then
12     if message space > 200 then
13         calculate average value
14         wrap as msgSGP
15         send msgSGP
16         clear message space

```

Gambar 4.7 Modifikasi *Method* sensing Untuk Mengatur Pengiriman Data

4.1.3 Modifikasi Pada *Driver Simulator*

Driver simulator adalah 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. Pada CSGPDriver ditentukan jumlah *node*, luas area, algoritma yang digunakan, penentuan keanggotaan *cluster*, sensor, fenomena yang dipindai, dan lain-lain. Peneliti memodifikasi *class* DriverSampleP2P yang terdapat pada SIDNet SWANS.

Dalam hal pembuatan *node*, modifikasi dilakukan pada method createNode dan menambahkan *method* baru, yaitu createSink. Modifikasi dilakukan dengan mengubah protokol *routing* menjadi C-SGP dan menyertakan atribut *cluster* pada masing-masing *node*. Penentuan atribut *cluster* diproses dalam sebuah method tambahan, yaitu estimateClusterArea. Pada *method* tersebut, ditentukan luas area cluster dan koordinat titik tengah *cluster*. Atribut *cluster*

tersebut dikirim pada saat protokol *heartbeat* dijalankan, sehingga pada tahap pencarian *neighbor*, node bisa saling bertukar informasi *cluster* dan menentukan keanggotaannya. Atribut *cluster* juga langsung disertakan pada tabel *neighbor node*, dengan cara memodifikasi proses instantiasi *NodeEntry* di class *Heartbeat*.

Peneliti memodifikasi method *createNode()* pada *driver default* simulator untuk membuat *node* yang sesuai dengan algoritma *routing* yang diusulkan. Sebaran *node* ditentukan secara acak, sedangkan posisi *sink node* ditentukan pada koordinat (1,1) atau di pojok kiri atas. Pembuatan *node* beserta *properties*-nya dapat dilihat pada *pseudocode* di Gambar 4.8.

| | |
|---|---|
| 1 | set node placement to random |
| 2 | set node mobility to static |
| 3 | set sink node placement to coordinate 1,1 |
| 4 | set sink node mobility to static |
| 5 | add sensor |
| 6 | set sensor parameter to phenomenaLayer(temperature) |
| 7 | set node battery to 40 MJ |
| 8 | set sink node battery to 70 MJ |
| | //panggil fungsi untuk membentuk cluster |
| 9 | call function estimateClusterArea |

Gambar 4.8 Pembentukan *Node* dan *Properties*-nya

Setelah *node* terbentuk, maka *cluster* dibentuk dengan cara mencari kumpulan titik tengah *cluster* dengan rumus titik tengah segi enam. Pada saat protokol *heartbeat* dijalankan, masing-masing *node* akan mencari titik tengah *cluster* yang paling dekat dengan dirinya, dan *node* akan bergabung ke dalam *cluster* tersebut. Lalu setiap *node* akan diberi routing protokol CS GP. Informasi jarak *node* ke titik tengah *cluster* yang diperlukan dalam tahap pemilihan *cluster head*, dipertukarkan antar *node* melalui fase *hearbeat*. Sehingga class *Hearbeat.java* dimodifikasi untuk menambah informasi jarak ke titik tengah *cluster* pada *node entry*.

| | |
|---|--|
| 1 | function estimateClusterArea |
| | //tentukan lebar cluster (Ry), tinggi cluster (Rx) |
| 2 | set R to 60 |
| 3 | set Rx = R |
| 4 | set Ry = 0.875 * R |
| | //menghitung jumlah cluster (c) |
| 5 | set c = (fieldlength/2*Rx+1) * (fieldlength/2*Ry+1) |
| | //menghitung titik tengah cluster M(x,y) |
| 6 | M(x,y) = M(previous x + 1.5*Rcx, previous y + 2*Rcy) |
| | //melengkapi properties cluster dengan cluster id |
| 7 | for i=0 to i<c |
| 8 | cluster[i] id = i |

Gambar 4.9 Pencarian Titik Tengah *Cluster*

4.2 Tahap Uji Coba

4.2.1 Skenario Pengujian

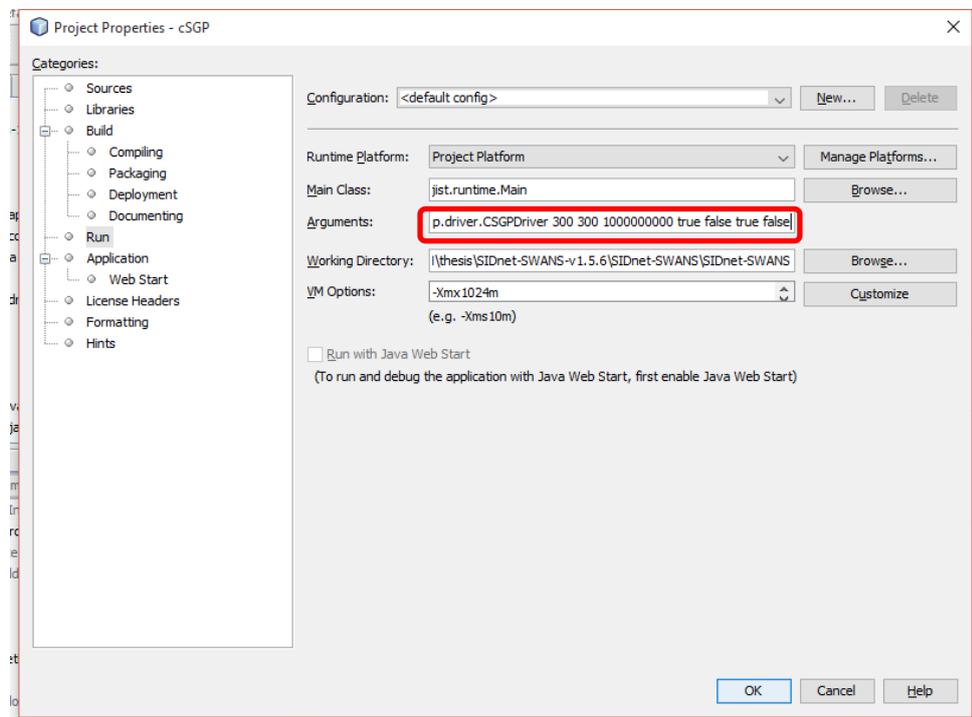
Pengujian dilakukan untuk membuktikan hipotesis bahwa teknik *clustering* yang diusulkan dapat meningkatkan *network lifetime*. Pengujian dengan menggunakan jaringan nyata tidak memungkinkan dilakukan karena harus dengan skala cukup besar (mencapai ratusan *node* dengan luas area mencapai ratusan meter persegi), sehingga simulasi dilakukan dengan simulator SIDNet SWANS. Untuk melihat reliabilitas protokol yang diusulkan, pengujian pada penelitian ini dibagi menjadi tiga skenario yang telah dijabarkan dalam Bab III, meliputi variasi luas area, variasi jumlah *node*, dan perbandingan dengan metode lain. Selama pengujian, terdapat perubahan suhu yang mengakibatkan berubahnya prioritas pesan. Perubahan suhu terjadi pada *cluster* satu, *cluster* lima, dan *cluster* sepuluh, serta bersifat tetap. Area lokasi *cluster* bersifat tetap untuk luas area yang sama, dan lokasi sink bersifat tetap untuk jumlah *node* yang sama. Perubahan temperatur dapat dilihat pada Gambar 4.10.

Tabel 4.1 Rangkuman Skenario Pengujian

| | Skenario I | Skenario II | Skenario III |
|-------------|--|---|--------------------|
| Jumlah Node | 100, 200, 300 | 200 | 100, 150, 200, 250 |
| Luas Area | 300x300 M | 200x200 M, 300x300 M, 400x400 M, 500x500 M | 300X300 m |
| Protokol | <i>C-SGP (diusulkan), SGP, SGP Non-Adaptive, C-SGP Non-Adaptive, C-SGP Random Cluster Head</i> | C-SGP (diusulkan) | C-SGP (diusulkan) |

4.2.3 Input Parameter Pengujian dan Eksekusi Simulator

Masukkan parameter untuk pengujian dilakukan melalui *run arguments* yang dapat diakses pada fitur project properties di NetBeans sebagaimana divisualisasikan pada Gambar 4.11.



Gambar 4.11 Penentuan *Run Argument* di *Project Properties*

Argumen yang diperlukan adalah jumlah *node*, luas area, waktu maksimal simulasi, protokol *routing* yang digunakan, mekanisme pengiriman data yang digunakan, dan metode pemilihan *cluster head*. *Syntax* yang digunakan adalah sebagai berikut:

```
jist.swans.Main sidnet.stack.users.csgp.driver.CSGPDriver
<jumlah node> <luas area> <waktu maksimal simulasi> <c-SGP??>
<SGP??> <Adaptive sending??> <random cluster head??>
```

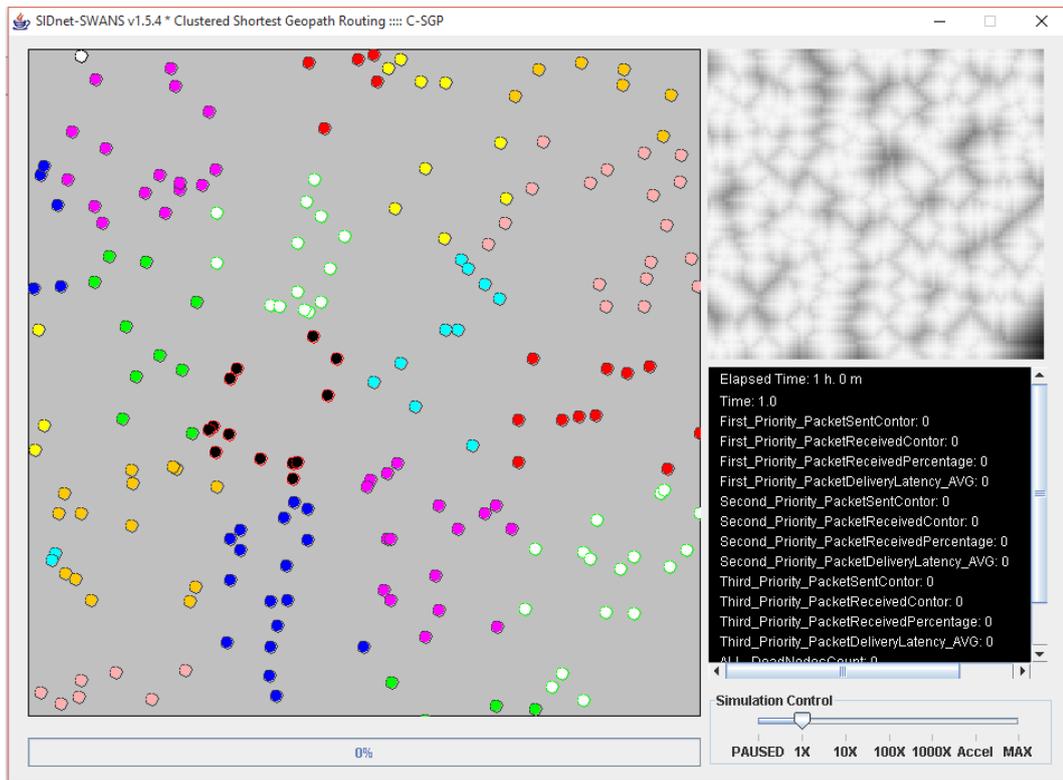
Contoh *run arguments* pengujian dapat dilihat di Tabel 4.2.

Eksekusi program dilakukan dengan memilih menu *run* pada IDE NetBeans. Simulator akan menampilkan tampilan GUI yang dapat digunakan untuk melakukan interaksi dengan *node*. Selama satu jam pertama, simulator akan menjalankan protokol heartbeat dan waktu simulasi dipercepat hingga seribu kali. Pada menit ke 59, waktu simulasi dilambatkan menjadi *real-time* untuk memberi kesempatan berinteraksi dengan simulator.

Tabel 4.2 Contoh *Run Arguments*

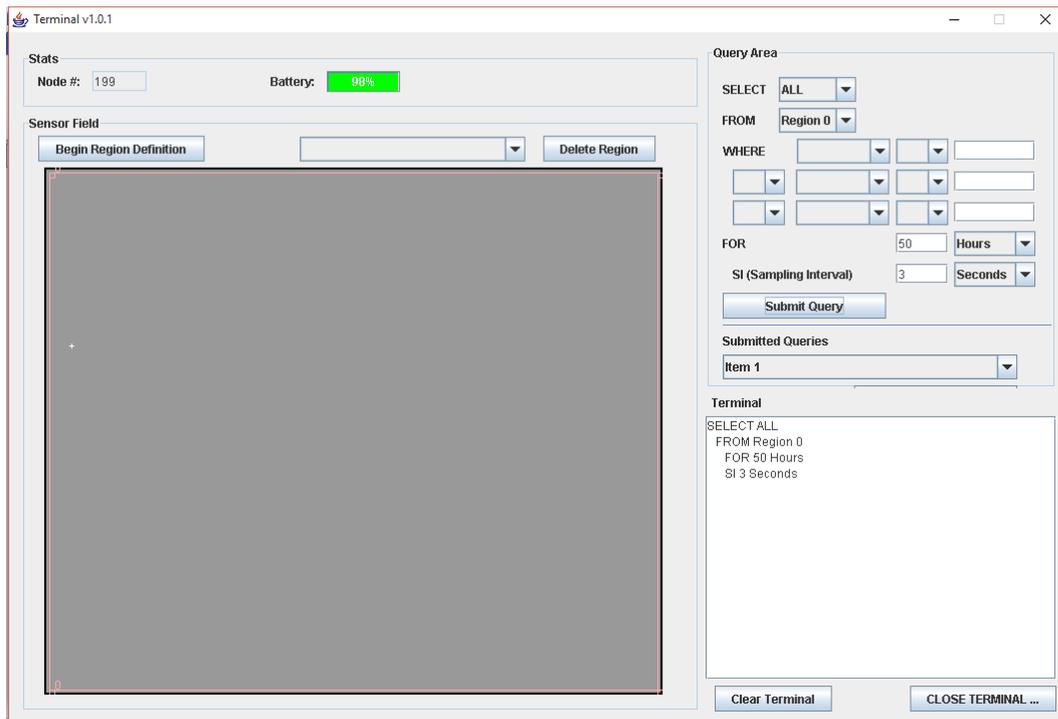
| | | |
|---------------|--|---|
| Skenario 1 | Jumlah Node: 100 Luas Area: 300 m ² Routing protocol: C-SGP Adaptive Sending: true Random clusterhead: false | jist.swans.Main sidnet.stack.users.csgp.driver.CSGPDriver 100 300 1000000000 true false true false |
| | Jumlah Node: 100 Luas Area: 300 m ² Routing protocol: SGP Adaptive Sending: false Random clusterhead: false | jist.swans.Main sidnet.stack.users.csgp.driver.CSGPDriver 100 300 1000000000 false true false false |
| | Jumlah Node: 100 Luas Area: 300 m ² Routing protocol: C-SGP Adaptive Sending: false Random clusterhead: false | jist.swans.Main sidnet.stack.users.csgp.driver.CSGPDriver 100 300 1000000000 true false false false |

| | |
|--|--|
| <p>Jumlah Node: 100 Luas Area: 300 m² Routing protocol: SGP Adaptive Sending: true Random clusterhead: false</p> | <pre>jist.swans.Main sidnet.stack.users.csgp.driver.CSGPDriver 100 300 10000000000 false true true false</pre> |
| <p>Jumlah Node: 100 Luas Area: 300 m² Routing protocol: C-SGP Adaptive Sending: true Random clusterhead: true</p> | <pre>jist.swans.Main sidnet.stack.users.csgp.driver.CSGPDriver 100 300 10000000000 true false true true</pre> |

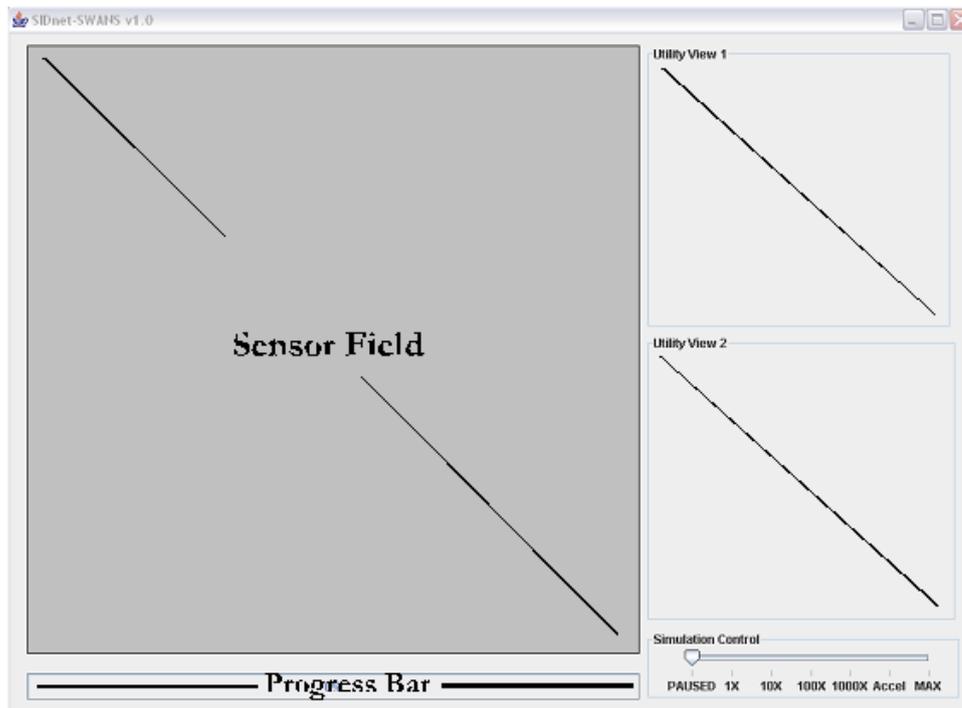


Gambar 4.12 Tampilan Simulator

Node yang dipilih dapat dijadikan *sink* dengan cara membuka aplikasi terminal simulator melalui *node* tersebut. Parameter pengujian meliputi batas waktu simulasi, interval pemindaian, area pemindaian, dan batas waktu pengamatan dapat dispesifikasikan melalui aplikasi terminal.



Gambar 4.13 Tampilan Terminal Simulator



Gambar 4.14 GUI Aplikasi SIDnet

4.3 Hasil dan Analisis

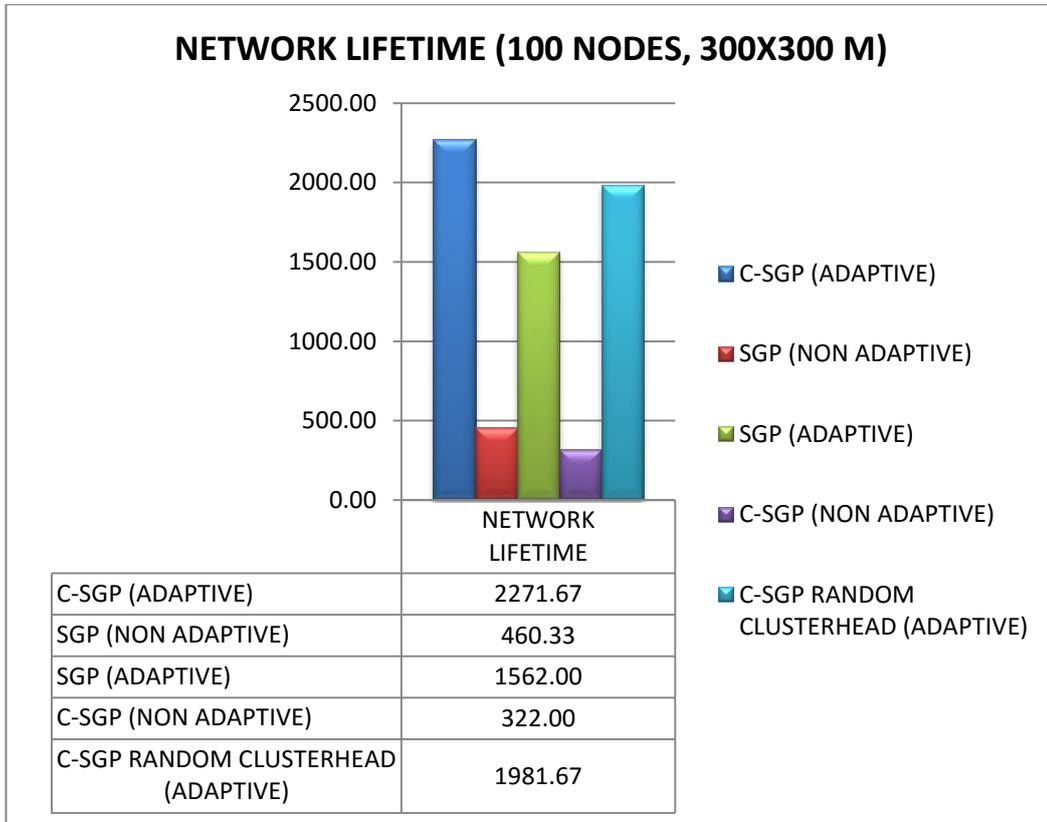
4.3.1 Hasil dan Analisis Skenario 1

Pada skenario I, protokol yang dikembangkan dibandingkan dengan protokol lain, yaitu *Shortest Geopath Routing*, *Shortest Geopath Routing with Adaptive Sending Mechanism*, *Clustered Shortest Geopath Routing without Adaptive Sending Mechanism*, dan *Clustered Shortest Geopath Routing using Random Cluster Head*. Masing-masing protokol disimulasikan sebanyak tiga kali, dan divariasikan kepadatan jaringannya dengan cara menambah jumlah *node* tanpa mengubah luas area simulasi. Hasil yang diperoleh ditampilkan di Tabel 4.3, Tabel 4.4, dan Tabel 4.5. Sedangkan untuk mempermudah pembacaan hasil, data direpresentasikan sebagai grafik yang divisualisasikan pada Gambar 4.15, Gambar 4.16, dan Gambar 4.17.

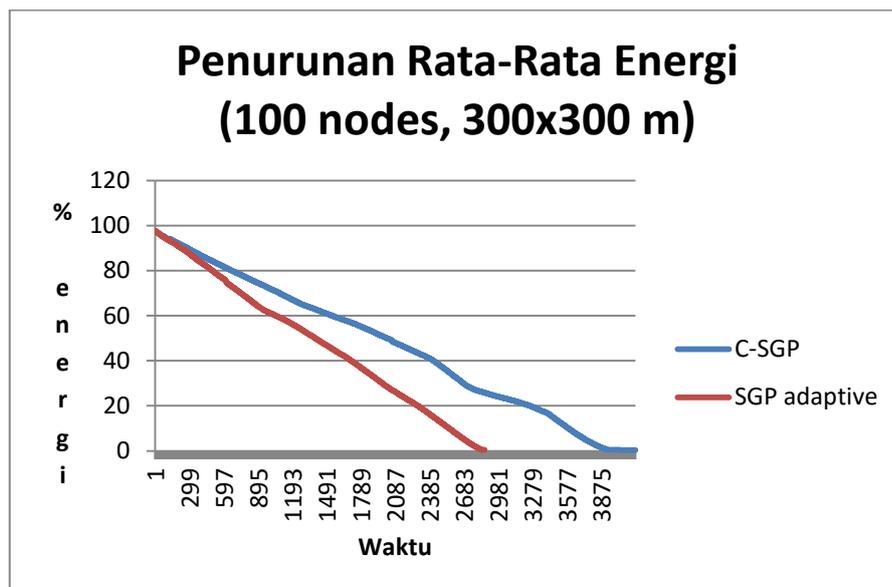
Pada Gambar 4.15 terlihat bahwa protokol yang dikembangkan (C-SGP) memiliki *network lifetime* yang tertinggi dibandingkan dengan protokol lain (31,2% lebih lama jika dibandingkan dengan *SGP-adaptive*, dan 393.5% lebih lama dibandingkan dengan *SGP-Non Adaptive*). Namun protokol C-SGP yang *non-adaptive* justru memiliki *network lifetime* yang paling rendah. Hal ini disebabkan karena *cluster head* memiliki beban yang jauh lebih besar daripada *node* lain, sehingga lebih cepat kehabisan energi dibandingkan dengan SGP, dan menjadi *node* yang pertama mati dalam waktu yang relatif singkat. Selain itu, terjadinya bottleneck pada protokol C-SGP *Non Adaptive* membuat beberapa *node* berkali-kali melakukan *retry* karena mendapat pesan kegagalan “NET QUEUE FULL”. Hal ini menambah beban energi pada beberapa *node*. Dengan definisi *network lifetime* yang merupakan rentang waktu dari beroperasinya jaringan, hingga *node* pertama mati, maka protokol C-SGP *Non Adaptive* justru memiliki *network lifetime* yang singkat. Dapat disimpulkan bahwa C-SGP *Non Adaptive* tidak cocok untuk digunakan untuk jaringan sensor yang memindai dan mengirim data dengan interval yang sangat singkat. Penerapan mekanisme *adaptive sending* terbukti meningkatkan *network lifetime* pada protokol C-SGP.

Tabel 4.3. Hasil Simulasi Skenario I untuk Kepadatan 1111/km²

| 100 NODES, 300X300 METER (1111/KM ²) | | C-SGP (ADAPTIVE) | | | | SGP (NON ADAPTIVE) | | | | SGP (ADAPTIVE) | | | | C-SGP (NON ADAPTIVE) | | | | C-SGP RANDOM CLUSTERHEAD (ADAPTIVE) | | | |
|---|----|------------------|------|------|----------------|--------------------|-----|-----|---------------|----------------|-----|------|----------------|----------------------|--------|--------|-----------|--|------|------|----------------|
| | | 1 | 2 | 3 | RATA-RATA | 1 | 2 | 3 | RATA-RATA | 1 | 2 | 3 | RATA-RATA | 1 | 2 | 3 | RATA-RATA | 1 | 2 | 3 | RATA-RATA |
| PDR (%) | P1 | 80 | 69 | 80 | 76.33 | 54 | 92 | 84 | 76.67 | 75 | 75 | 80 | 76.67 | 99.8 | 99.7 | 95.0 | 98.17 | 61 | 90 | 70 | 73.67 |
| | P2 | 57 | 51 | 57 | 55.00 | 60 | 90 | 63 | 71.00 | 74 | 74 | 90 | 79.33 | 99.0 | 80.0 | 98.0 | 92.33 | 78 | 94 | 84 | 85.33 |
| | P3 | 93 | 97 | 98 | 96.00 | 85 | 44 | 60 | 63.00 | 87 | 88 | 81 | 85.33 | 62.0 | 30.0 | 40.0 | 44.00 | 83 | 95 | 87 | 88.33 |
| LATENCY (ms) | P1 | 87 | 85 | 86 | 86.00 | 962 | 107 | 112 | 393.67 | 149 | 143 | 150 | 147.33 | 50.0 | 47.0 | 56.0 | 51.00 | 62 | 72 | 72 | 68.67 |
| | P2 | 83 | 83 | 83 | 83.00 | 847 | 98 | 102 | 349.00 | 131 | 179 | 165 | 158.33 | 50.0 | 52.0 | 60.0 | 54.00 | 57 | 65 | 64 | 62.00 |
| | P3 | 76 | 87 | 79 | 80.67 | 801 | 445 | 99 | 448.33 | 741 | 757 | 801 | 766.33 | 316.0 | 120.0 | 214.0 | 216.67 | 73 | 84 | 78 | 78.33 |
| NETWORK LIFETIME (menit) | | 2280 | 2270 | 2265 | 2271.67 | 476 | 440 | 465 | 460.33 | 1975 | 855 | 1856 | 1562.00 | 245.00 | 290.00 | 431.00 | 322.00 | 1245 | 2345 | 2355 | 1981.67 |



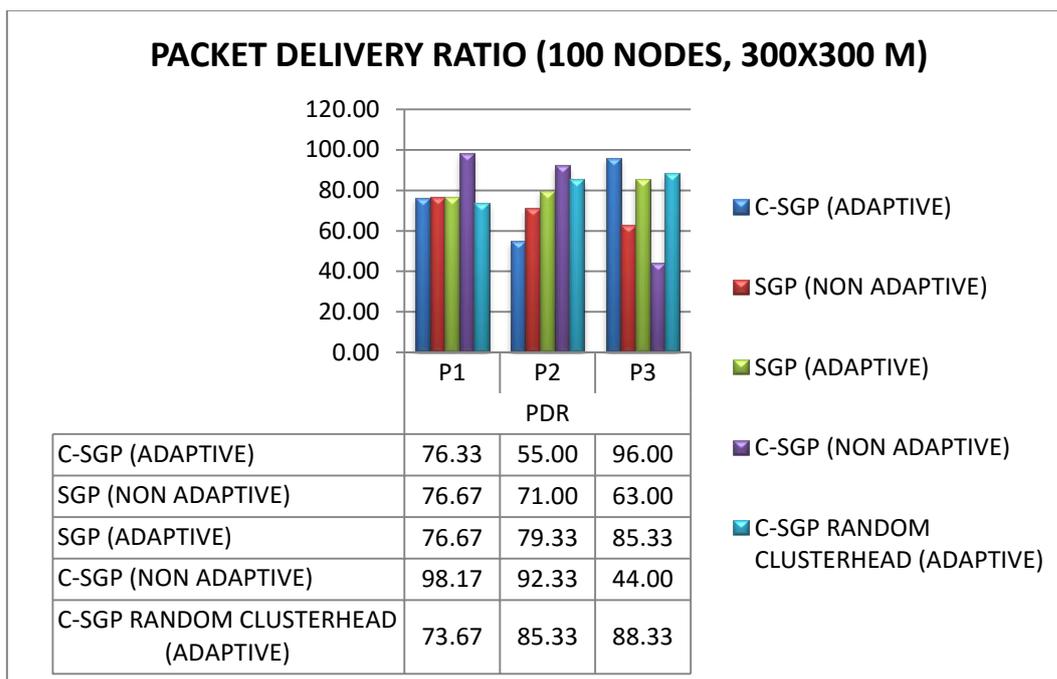
Gambar 4.15 Grafik Perbandingan *Network Lifetime* 100 Nodes, 300x300 m (1111/km²)



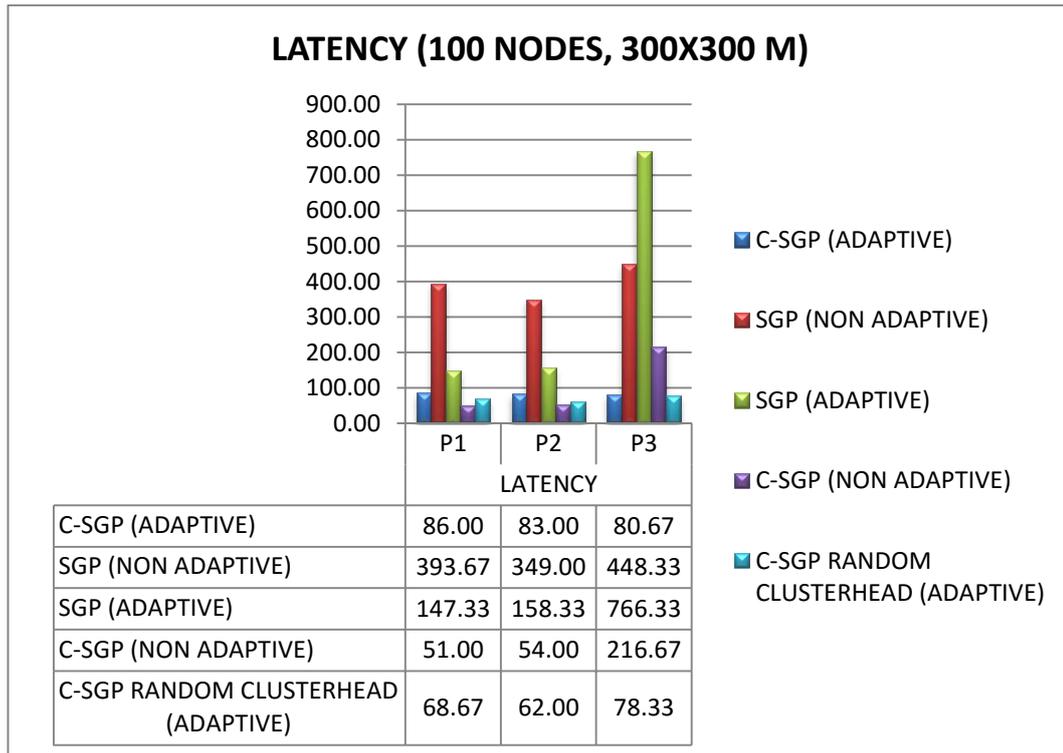
Gambar 4.16. Penurunan Rata-Rata Energi (C-SGP dan SGP-adaptive) pada Kepadatan 1.111 nodes/km².

Pada Gambar 4.17, PDR (*packet delivery ratio*) pada protokol C-SGP untuk pesan prioritas tertinggi (P1) justru menduduki peringkat terakhir. Hal ini disebabkan karena kepadatan node yang rendah sehingga banyak node yang menemui hole. Begitu pula untuk prioritas kedua (P2). Pada protokol C-SGP non adaptive, PDR justru tinggi pada P1 dan P2. Hal ini sebenarnya karena data P1 dan P2 yang terkumpul hanya yang terjadi pada menit 100-200 saja (awal simulasi). Sedangkan data P1 dan P2 yang terjadi pada menit ke 800 dan sesudahnya, tidak terdeteksi karena jaringan sudah mati.

Pada gambar 4.18, divisualisasikan data *latency* jaringan. Nilai *latency* yang kecil menunjukkan performa jaringan yang baik, dan sebaliknya. C-SGP memiliki *latency* yang rendah dan stabil untuk semua prioritas. Namun C-SGP non adaptive dan random cluster head memiliki *latency* yang lebih rendah pada data P1 dan P2.



Gambar 4.17 *Packet Delivery Ratio* 100 Nodes, 300x300 m (1111/km²)

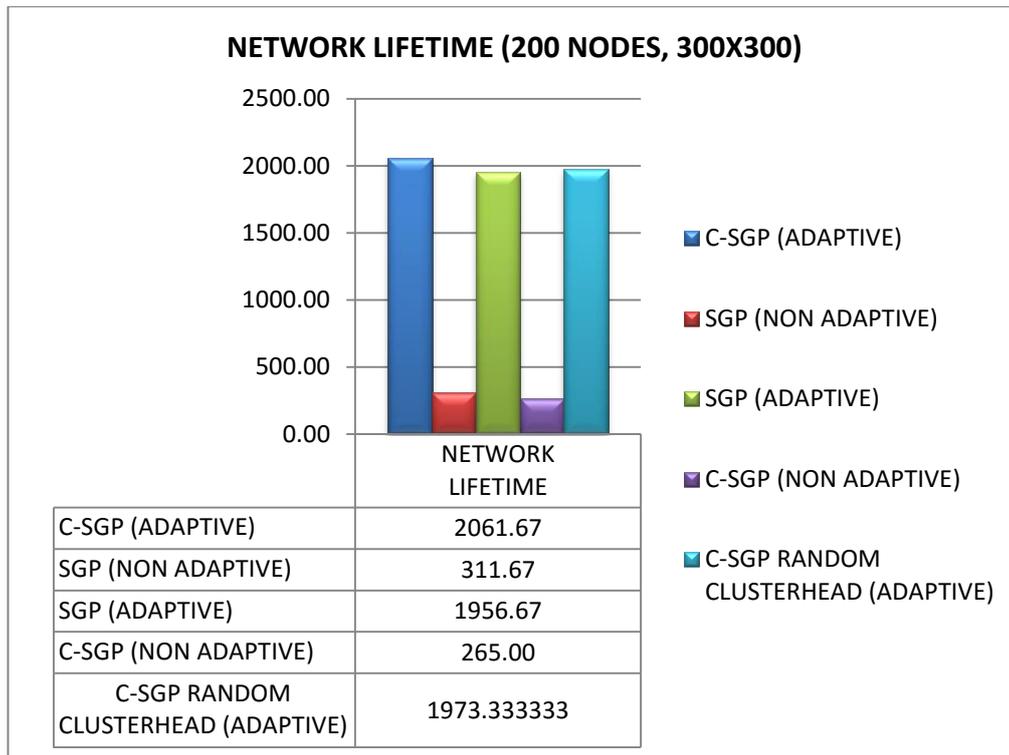


Gambar 4.18 *Network Latency 100 Nodes, 300x300 m (1111/km²)*

Performa protokol C-SGP meningkat setelah diujicoba dengan kepadatan lebih tinggi, yaitu 2222/km² (200 nodes, 300x300 m). Packet delivery ratio meningkat 20% dan *network lifetime* masih menempati peringkat tertinggi (lebih tinggi 5.06% jika dibandingkan dengan SGP-*adaptive*), seperti yang dipaparkan dalam Tabel 4.4. Untuk mempermudah analisa, maka data hasil simulasi divisualisasikan dalam bentuk grafik batang, seperti pada Gambar 4.19, Gambar 4.20, dan Gambar 4.21.

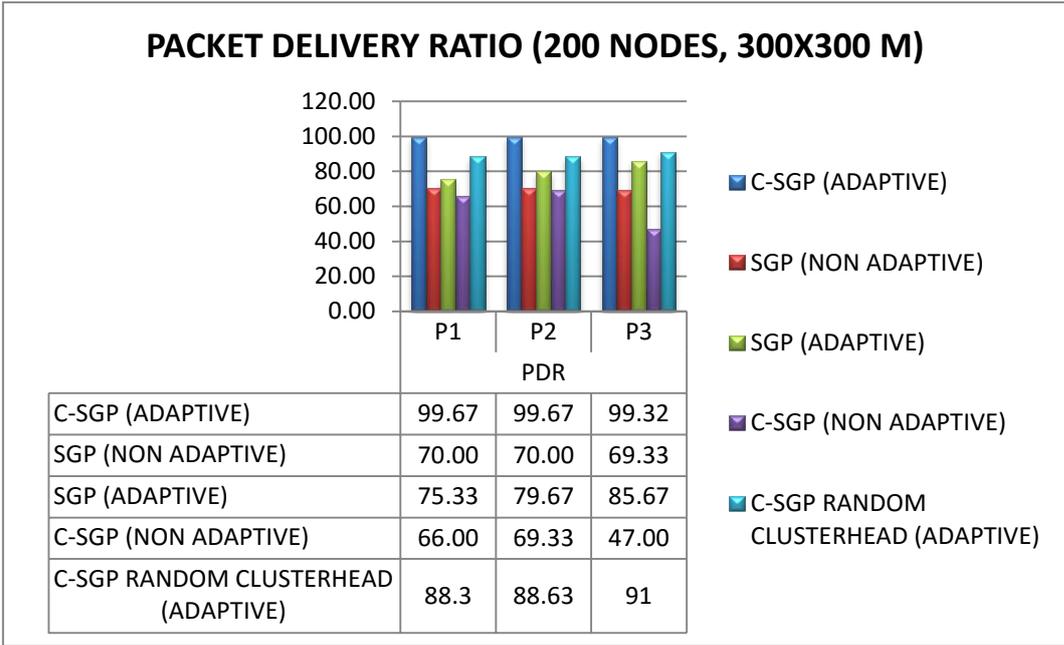
Tabel 4.4 Hasil Simulasi Skenario I untuk Kepadatan 2222/km²

| 200 NODES, 300X300 METER (2222/KM ²) | | C-SGP (ADAPTIVE) | | | | SGP (NON ADAPTIVE) | | | | SGP (ADAPTIVE) | | | | C-SGP (NON ADAPTIVE) | | | | C-SGP RANDOM CLUSTERHEAD (ADAPTIVE) | | | |
|---|----|------------------|------|------|----------------|--------------------|-----|-----|---------------|----------------|------|------|----------------|----------------------|---------|---------|----------------|--|------|------|---------------|
| | | 1 | 2 | 3 | RATA-RATA | 1 | 2 | 3 | RATA-RATA | 1 | 2 | 3 | RATA-RATA | 1 | 2 | 3 | RATA-RATA | 1 | 2 | 3 | RATA-RATA |
| PDR (%) | P1 | 100.00 | 100 | 99 | 99.67 | 50 | 80 | 80 | 70.00 | 77 | 67 | 82 | 75.33 | 63.00 | 70.00 | 65.00 | 66.00 | 89 | 99.9 | 76 | 88.3 |
| | P2 | 100.00 | 99 | 100 | 99.67 | 50 | 90 | 70 | 70.00 | 87 | 71 | 81 | 79.67 | 65.00 | 68.00 | 75.00 | 69.33 | 93 | 99.9 | 73 | 88.63 |
| | P3 | 99.96 | 99 | 99 | 99.32 | 53 | 83 | 72 | 69.33 | 85 | 81 | 91 | 85.67 | 42.00 | 49.00 | 50.00 | 47.00 | 94 | 99 | 80 | 91 |
| LATENCY (ms) | P1 | 68.00 | 67 | 68 | 67.67 | 110 | 172 | 167 | 149.67 | 224 | 208 | 300 | 244.00 | 49.00 | 48.00 | 55.00 | 50.67 | 70 | 72 | 89 | 77 |
| | P2 | 65.00 | 61 | 64 | 63.33 | 109 | 121 | 123 | 117.67 | 189 | 186 | 210 | 195.00 | 51.00 | 58.00 | 52.00 | 53.67 | 62 | 65 | 82 | 69.67 |
| | P3 | 84.00 | 84 | 85 | 84.33 | 916 | 812 | 921 | 883.00 | 1598 | 1362 | 1421 | 1460.33 | 1693.00 | 1013.00 | 1432.00 | 1379.33 | 82 | 83 | 90 | 85 |
| NETWORK LIFETIME (menit) | | 2060 | 2065 | 2060 | 2061.67 | 345 | 290 | 300 | 311.67 | 2145 | 1710 | 2015 | 1956.67 | 240.00 | 235.00 | 320.00 | 265.00 | 1920 | 2055 | 1945 | 1973.3 |

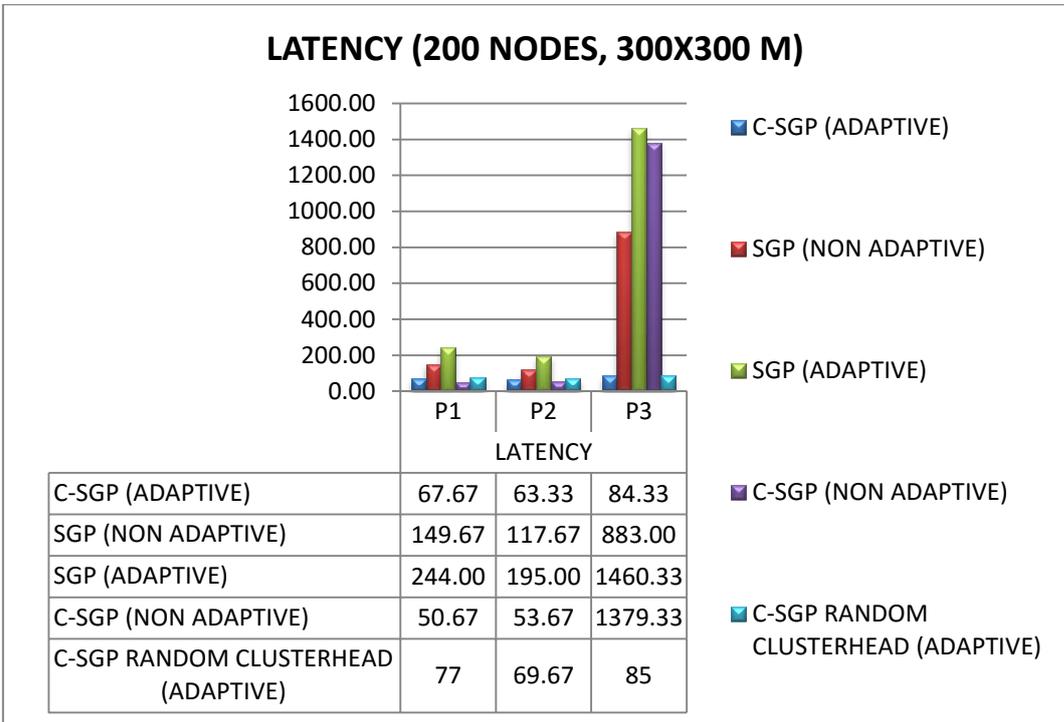


Gambar 4.19. Perbandingan Network Lifetime dengan Kepadatan Jaringan $2222/\text{km}^2$

Pada Gambar 4.19, protokol C-SGP (*adaptive*) memiliki *network lifetime* tertinggi di antara empat protokol lainnya, namun selisihnya tidak signifikan dengan SGP *adaptive*, dan C-SGP *random clusterhead*. Sedangkan packet delivery ratio C-SGP *adaptive* menempati urutan tertinggi dibandingkan protokol pembandingan. Hal ini cenderung disebabkan karena kepadatan jaringan lebih ideal, tidak terlalu renggang, dan tidak terlalu padat. Begitu pula *latency*, juga menurun jika dibandingkan dengan kepadatan $1111/\text{km}^2$. Dapat disimpulkan bahwa protokol C-SGP baik *adaptive* maupun non *adaptive*, kurang cocok jika diimplementasikan untuk jaringan dengan kepadatan rendah.

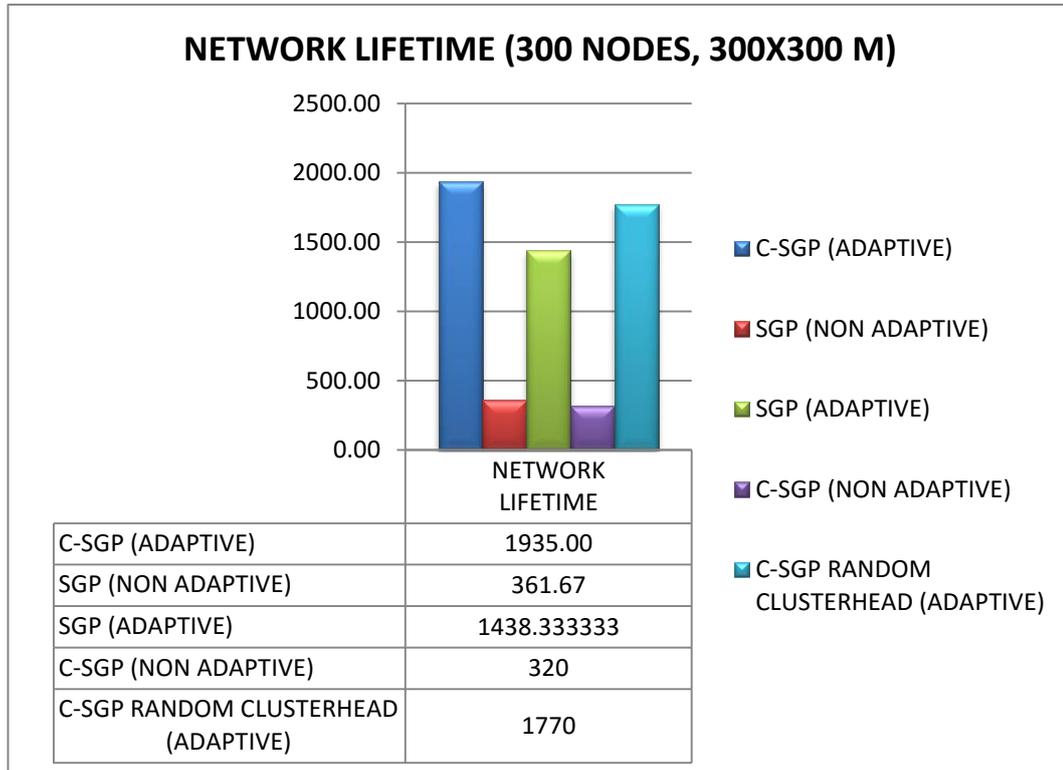


Gambar 4.20 *Packet Delivery Ratio* Dengan Kepadatan 2222/km² (200 nodes, 300x300 m)



Gambar 4.21 *Latency* Dengan Kepadatan 2222/km² (200 nodes, 300x300 m)

Untuk mengetahui perbandingan kinerja protokol yang dikembangkan pada jaringan padat, maka pengujian juga dilakukan di kepadatan jaringan tinggi, yaitu 3333/km². Hasil pengujian dirangkum di Tabel 4.5.

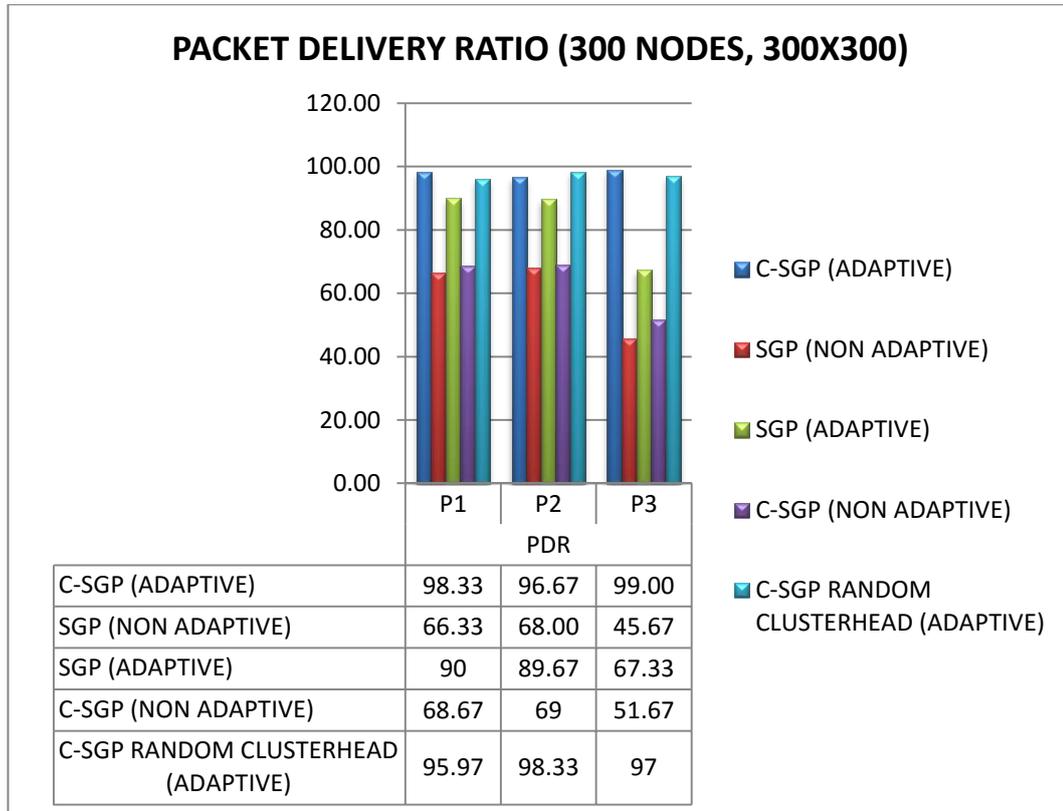


Gambar 4.22 Perbandingan *Network Lifetime* pada Kepadatan 3333/km² (300 nodes, 300x300 m)

Pada kepadatan jaringan tinggi, C-SGP adaptive masih memiliki *network lifetime* tertinggi (lebih tinggi 25,68% jika dibandingkan dengan SGP-*adaptive*, sedangkan C-SGP *non adaptive* masih menempati *level* terbawah meskipun tidak terlalu jauh selisihnya dengan protokol SGP *non adaptive*. Sedangkan *packet delivery ratio* dan *latency* memiliki nilai yang hampir sama dengan protokol C-SGP random clusterhead. Hal ini karena kepadatan jaringan yang tinggi, sehingga meskipun pemilihan *cluster head* dilakukan secara random, maka jalur *routing* kemungkinan besar tidak menemui hole.

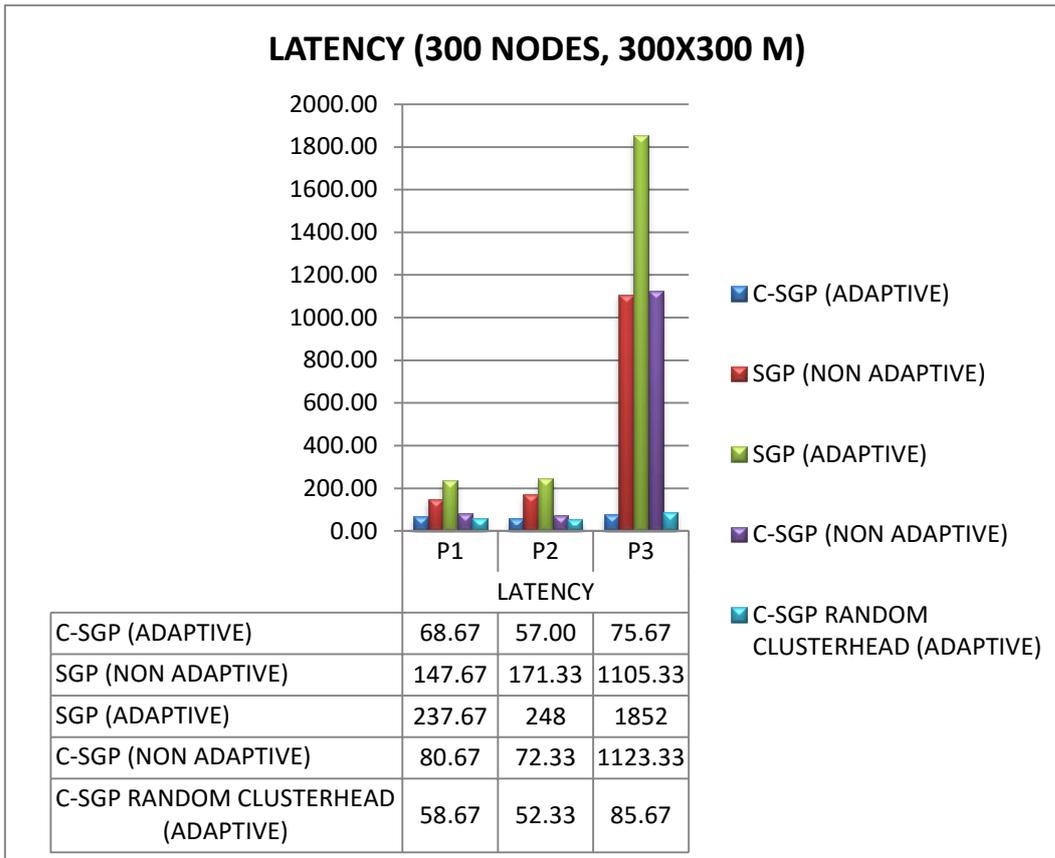
Tabel 4.5 Hasil Simulasi Skenario I pada Kepadatan Jaringan 3333/km²

| 300 NODES, 300X300 METER (3333/KM ²) | | C-SGP (ADAPTIVE) | | | | SGP (NON ADAPTIVE) | | | | SGP (ADAPTIVE) | | | | C-SGP (NON ADAPTIVE) | | | | C-SGP RANDOM CLUSTERHEAD (ADAPTIVE) | | | |
|---|----|------------------|------|------|----------------|--------------------|------|-----|----------------|----------------|------|------|-----------|----------------------|------|------|-----------|--|------|------|-----------|
| | | 1 | 2 | 3 | RATA-RATA | 1 | 2 | 3 | RATA-RATA | 1 | 2 | 3 | RATA-RATA | 1 | 2 | 3 | RATA-RATA | 1 | 2 | 3 | RATA-RATA |
| PDR (%) | P1 | 100 | 96 | 99 | 98.33 | 46 | 61 | 92 | 66.33 | 73 | 99 | 98 | 90 | 50 | 76 | 80 | 68.67 | 99.9 | 98 | 90 | 95.97 |
| | P2 | 100 | 92 | 98 | 96.67 | 50 | 73 | 81 | 68.00 | 81 | 96 | 92 | 89.67 | 53 | 80 | 74 | 69 | 99 | 98 | 98 | 98.33 |
| | P3 | 99 | 99 | 99 | 99.00 | 21 | 73 | 43 | 45.67 | 60 | 71 | 71 | 67.33 | 34 | 45 | 76 | 51.67 | 95 | 99 | 97 | 97 |
| LATENCY (ms) | P1 | 67 | 69 | 70 | 68.67 | 159 | 182 | 102 | 147.67 | 303 | 198 | 212 | 237.67 | 71 | 80 | 91 | 80.67 | 55 | 67 | 54 | 58.67 |
| | P2 | 50 | 60 | 61 | 57.00 | 174 | 189 | 151 | 171.33 | 254 | 197 | 293 | 248 | 78 | 76 | 63 | 72.33 | 55 | 59 | 43 | 52.33 |
| | P3 | 56 | 90 | 81 | 75.67 | 1232 | 1102 | 982 | 1105.33 | 1696 | 2037 | 1823 | 1852 | 1138 | 1211 | 1021 | 1123.33 | 98 | 82 | 77 | 85.67 |
| NETWORK LIFETIME (menit) | | 1905 | 1940 | 1960 | 1935.00 | 355 | 300 | 430 | 361.67 | 1125 | 1575 | 1615 | 1438.33 | 240 | 355 | 365 | 320 | 1950 | 1930 | 1430 | 1770 |



Gambar 4.23 Perbandingan *Packet Delivery Ratio* pada Kepadatan $3333/\text{km}^2$ (300 nodes, 300x300 m)

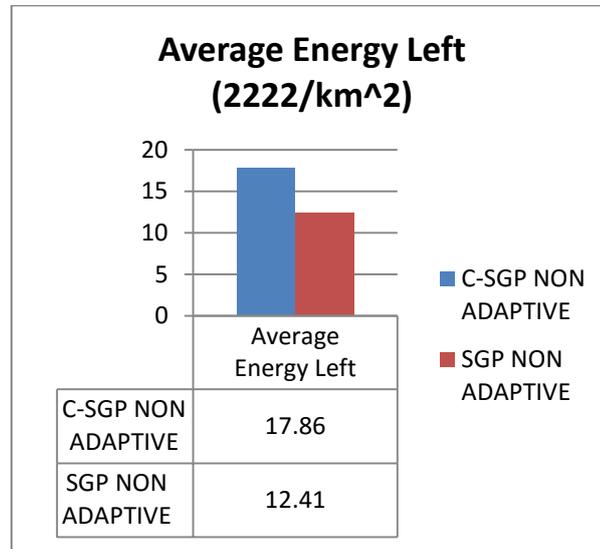
Dari keseluruhan uji coba pada skenario I, dapat disimpulkan bahwa protokol C-SGP *adaptive* cukup stabil meningkatkan *network lifetime*, dan tidak terlalu terpengaruh dengan kepadatan jaringan. Namun dari sisi PDR, protokol C-SGP kurang cocok jika diterapkan pada jaringan dengan kepadatan yang rendah karena dengan jumlah node yang jarang, pembagian area *cluster* secara geografis akan merugikan area-area tertentu yang memiliki *node* sangat sedikit atau memiliki *hole*. Sehingga pada kepadatan rendah, protokol SGP *adaptive* lebih menguntungkan karena jalur yang terbentuk lebih fleksibel.



Gambar 4.24 Perbandingan *Latency* pada Kepadatan $3333/\text{km}^2$ (300 nodes, 300x300 m)

Peneliti menelaah lebih lanjut mengenai rendahnya *network lifetime* pada protokol C-SGP *non adaptive*. Telaah lanjutan dilakukan dengan melihat statistik lain, yaitu *average energy left*. Setelah dilakukan pengamatan selama 35 jam, protokol C-SGP *non adaptive* memiliki *average energy left* yang lebih tinggi jika dibandingkan dengan protokol SGP *non adaptive*. Rendahnya *network lifetime* yang dibahas pada analisa sebelumnya disebabkan oleh batasan *network lifetime*, dimana *network lifetime* merupakan lama waktu mulai dari jaringan beroperasi hingga node pertama mati. Pada protokol C-SGP *non adaptive*, *cluster head* dan jalur yang dilewatinya memiliki beban kerja yang lebih tinggi, sehingga lebih cepat kehabisan energi. Sedangkan member *node* cenderung memiliki beban kerja yang ringan. Namun jika dilihat secara keseluruhan, rata-rata konsumsi energi pada C-SGP *non adaptive*, tetap lebih rendah dibandingkan SGP *non adaptive*. Perbandingan *average energy left* antara C-SGP *non adaptive* dan SGP *non*

adaptive divisualisasikan pada Gambar 4.25, yang menunjukkan penghematan energi sebesar 29.4%.



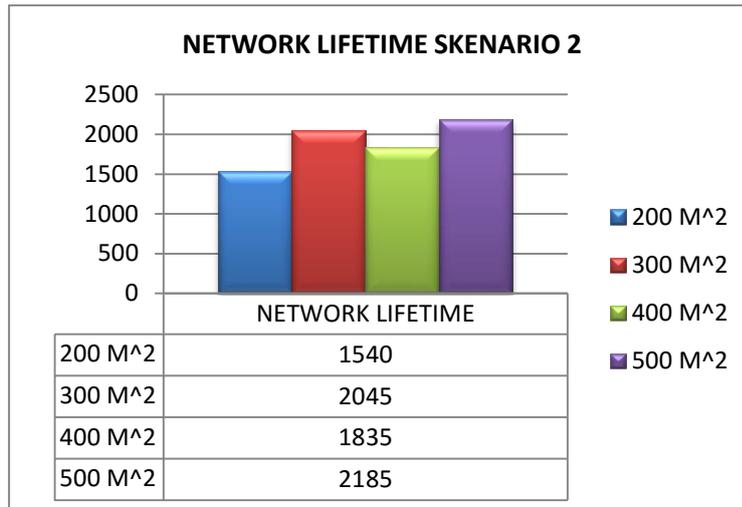
Gambar 4.25 Perbandingan *Average Energy Left* C-SGP *non Adaptive* dan SGP *non Adaptive* (Kepadatan 2222/km²)

4.3.2 Hasil dan Analisis Skenario 2

Pada skenario 2, protokol yang dikembangkan diuji coba dengan luas area yang bervariasi. Uji coba ini untuk mengetahui pengaruh perubahan luas area terhadap protokol C-SGP. Hasil ujicoba dirangkum pada Tabel 4.6. Uji coba dilaksanakan sebanyak tiga kali untuk luas area yang sama. Data yang dipaparkan merupakan data hasil rata-rata.

Tabel 4.6 Data Hasil Uji Coba Variasi Luas Area (Skenario 2)

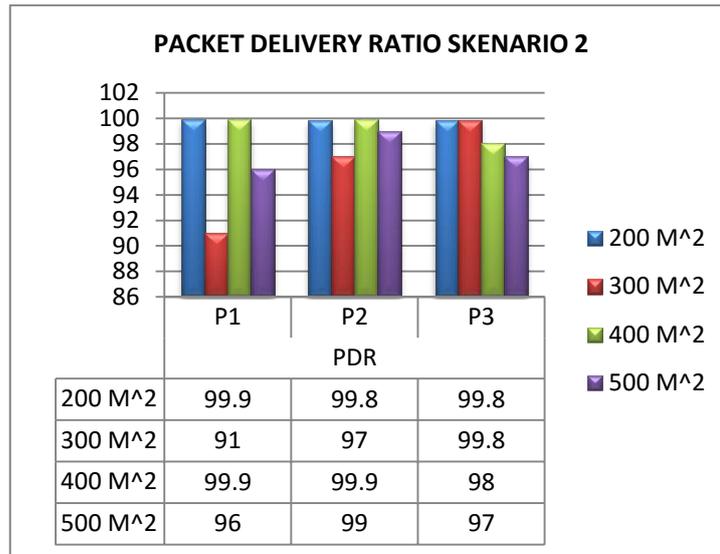
| LUAS AREA: | | 200 M ² | 300 M ² | 400 M ² | 500 M ² |
|------------------|----|--------------------|--------------------|--------------------|--------------------|
| PDR | P1 | 99.9 | 91 | 99.9 | 96 |
| | P2 | 99.8 | 97 | 99.9 | 99 |
| | P3 | 99.8 | 99.8 | 98 | 97 |
| LATENCY | P1 | 56 | 67 | 20 | 74 |
| | P2 | 54 | 60 | 23 | 71 |
| | P3 | 53 | 85 | 134 | 263 |
| NETWORK LIFETIME | | 1540 | 2045 | 1835 | 2185 |



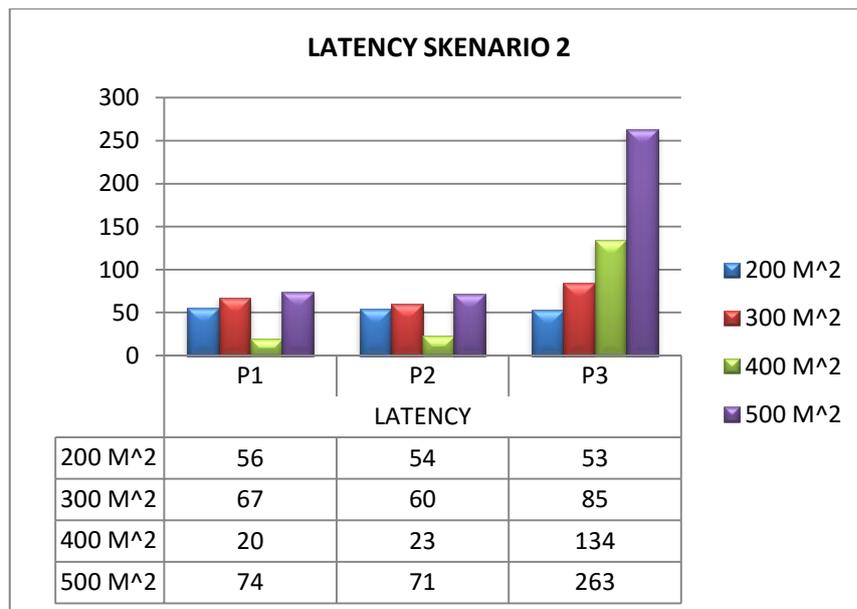
Gambar 4.26 Perbandingan Network Lifetime C-SGP Adaptive dengan Variasi Luas Area

Pada Gambar 4.26 terdapat kecenderungan kenaikan *network lifetime* berbanding lurus dengan peningkatan luas area. Hal ini disebabkan oleh kepadatan jaringan yang semakin menurun sehingga kesibukan jaringan juga semakin menurun. Sedangkan *packet delivery ratio* tidak menunjukkan pola yang teratur pada prioritas 1 dan 2, namun menunjukkan pola menurun pada prioritas 3. Hal ini disebabkan karena data P1 dan P2, hanya dikirim pada saat kejadian dengan interval kirim yang cepat, dan terjadi di *cluster* yang berada di tengah-tengah area. Oleh karena itu, PDR-nya sering terpengaruh oleh banyak faktor lain. Sedangkan penurunan PDR pada data P3, disebabkan oleh kesibukan jaringan yang ikut menurun karena kepadatan yang menurun.

Pada grafik perbandingan *latency* (Gambar 4.28), terdapat *trend* kenaikan *latency* berbanding lurus dengan kenaikan luas area. Hal ini dikarenakan 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.



Gambar 4.27 Perbandingan *Packet Delivery Ratio* C-SGP dengan Variasi Luas Area



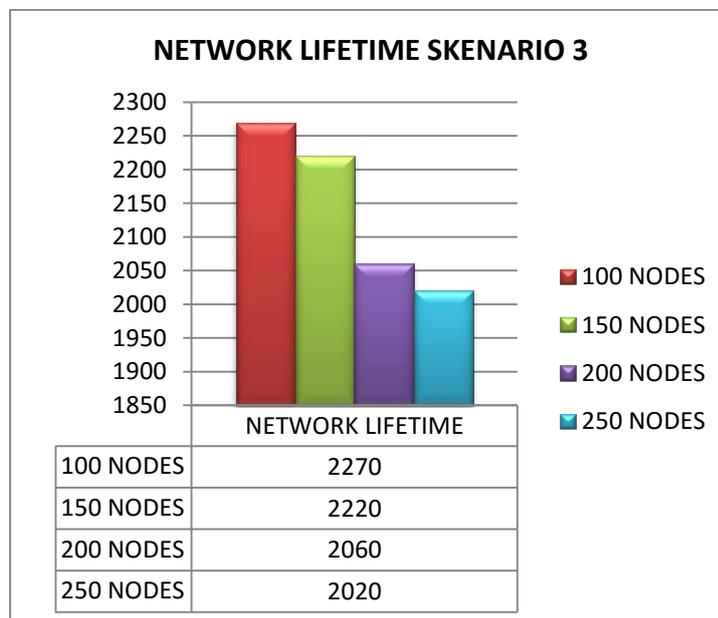
Gambar 4.28 Perbandingan *Latency* C-SGP dengan Variasi Luas Area

4.3.3 Hasil dan Analisis Skenario 3

Pada skenario 3 protokol yang dikembangkan diuji coba dengan memvariasikan jumlah node, sedangkan luas area tetap. Skenario ini bertujuan untuk melihat pengaruh perubahan jumlah node pada protokol C-SGP. Hasil uji coba dirangkum di Tabel 4.7.

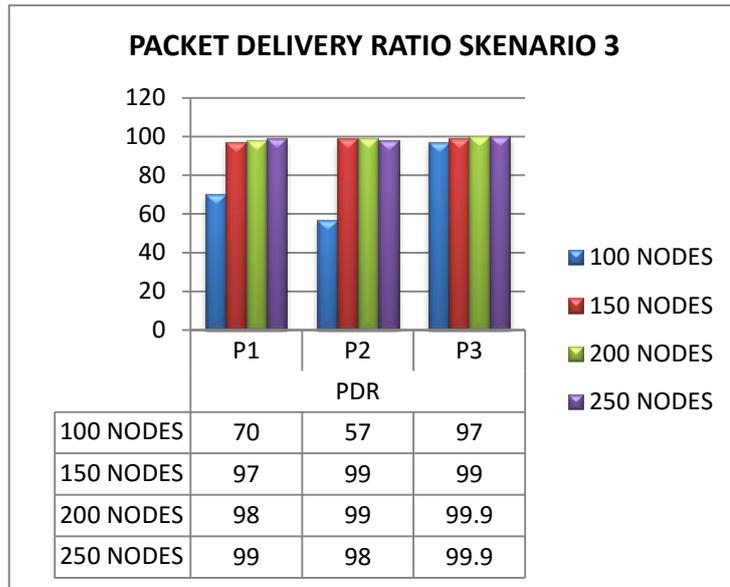
Tabel 4.7 Rangkuman Hasil Uji Coba Skenario 3

| Jumlah Node: | | 100 NODES | 150 NODES | 200 NODES | 250 NODES |
|---------------------|----|--------------|--------------|--------------|--------------|
| PDR | P1 | 70 | 97 | 98 | 99 |
| | P2 | 57 | 99 | 99 | 98 |
| | P3 | 97 | 99 | 99.9 | 99.9 |
| LATENCY | P1 | 85 | 66 | 67 | 62 |
| | P2 | 81 | 63 | 61 | 70 |
| | P3 | 79 | 75 | 83 | 76 |
| NETWORK LIFETIME | | 2270 | 2220 | 2060 | 2020 |

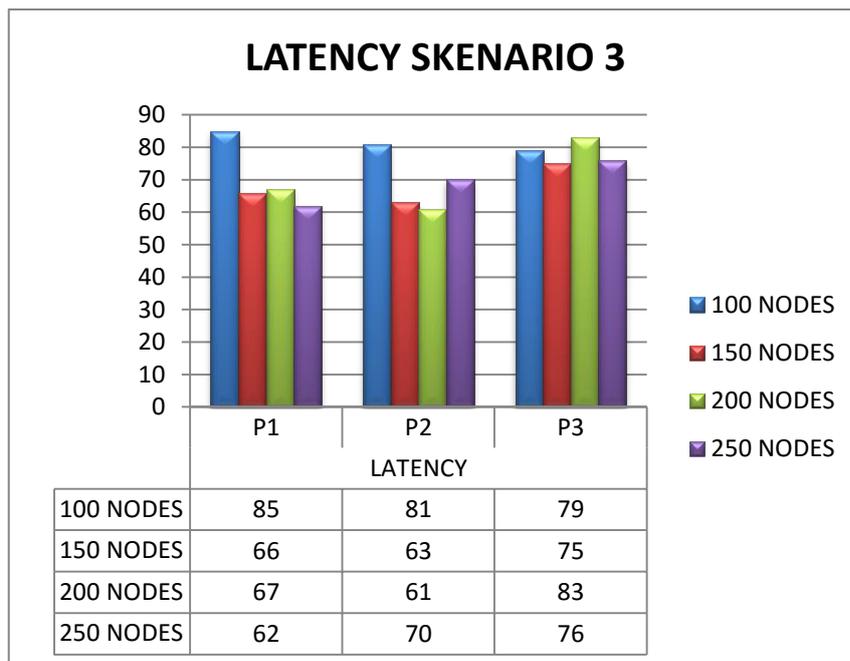


Gambar 4.29 Network Lifetime Skenario 3

Pada Gambar 4.29, terjadi penurunan *network lifetime* seiring dengan bertambahnya jumlah *node*. Hal ini dikarenakan kepadatan jaringan yang meningkat sehingga jumlah data yang ditransmisikan ikut meningkat. Akibatnya, energi yang diperlukan juga semakin besar. Sedangkan perbandingan PDR tidak terlalu terpengaruh oleh perubahan jumlah *node*, kecuali jumlah node 100, dimana jaringan memiliki kepadatan yang sangat rendah sehingga pengiriman pesan dapat terhambat oleh *hole*. Kepadatan yang sangat rendah juga berpengaruh terhadap latency, dimana nilai *latency* pada uji coba dengan 100 *node* terhitung memiliki nilai paling tinggi pada P1 dan P2, dan tertinggi kedua pada P3.



Gambar 4.30 PDR Skenario 3



Gambar 4.31 Latency Skenario 3

[Halaman ini sengaja dikosongkan]

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari serangkaian kegiatan penelitian yang dilakukan dapat diambil kesimpulan sebagai berikut:

1. Pengembangan protokol *routing C-SGP* dilakukan dengan cara memodifikasi protokol *routing SGP*. Modifikasi dilakukan dengan menambahkan tahapan *clustering*, dimana jaringan dibagi menjadi beberapa *cluster* berbentuk segi enam berdasarkan informasi geografis jaringan. Pemilihan *cluster head* dilakukan dengan mencari *node* yang memiliki posisi paling tengah atau paling dekat dengan titik tengah geografis *cluster*. Pergantian *cluster head* dilakukan dua kali, pada sisa energi 60% dan 30%, untuk mengurangi beban kerja berlebih pada *cluster head*.
2. Mekanisme pengiriman data pada protokol C-SGP, dimodifikasi sehingga data dengan prioritas tinggi dikirim dengan kecepatan tinggi, sedangkan data dengan prioritas rendah ditampung sementara di member node hingga memenuhi kapasitas tertentu, dan dilakukan rata-rata nilai data. Setelah dirata-rata, data dikirim ke *cluster head* untuk diteruskan ke *sink*.
3. Hasil uji coba skenario pertama, di mana C-SGP dibandingkan dengan protokol SGP, SGP *adaptive*, C-SGP *non-adaptive*, dan C-SGP dengan *random clusterhead*, menunjukkan bahwa protokol C-SGP mampu meningkatkan *network lifetime* pada kepadatan jaringan 1.111 *nodes/km²*, 2.222 *nodes/km²*, dan 3.333 *nodes/km²*.
4. Hasil uji coba skenario kedua, di mana C-SGP diuji dengan memvariasikan luas area jaringan, menunjukkan bahwa semakin luas area jaringan, *network lifetime* yang terukur semakin tinggi. Hal ini dikarenakan dengan bertambahnya luas jaringan, maka kepadatan jaringan semakin berkurang yang turut menyebabkan berkurangnya kesibukan jaringan.

5. Hasil uji coba skenario ketiga, di mana C-SGP diuji dengan memvariasikan jumlah *node*, menunjukkan bahwa semakin banyak jumlah *node*, *network lifetime* yang terukur semakin rendah. Hal ini disebabkan oleh bertambahnya kepadatan jaringan yang menyebabkan kesibukan jaringan juga bertambah.

5.2 Saran

Dari penelitian yang dilakukan, perlu dikembangkan lebih lanjut mekanisme pergantian cluster head untuk lebih mengoptimalkan konsumsi sumber daya. Selain itu perlu penelitian lebih lanjut pada protokol C-SGP non adaptive agar tidak mudah terjadi pesan kegagalan NET QUEUE FULL yang mengakibatkan konsumsi energi berlebih karena banyak node harus melakukan pengulangan (retry) dalam mengirim pesan.

DAFTAR PUSTAKA

- Abad, F. M., & Jamali, M. A. (2011). Modify LEACH Algorithm for Wireless Sensor Network. *International Journal of Computer Science Issues, Vol. 8, Issue 5, No 1*.
- Al-Kaseem, B. R., & Al-Raweshidy, H. S. (2016). Energy Efficient MAC Protocol with Smart Sleep Scheduling for Cluster-Based M2M Networks. *International Conference on Information Communication and Management (ICICM)* (pp. 80-87). Hatfield, UK: IEEE.
- Dietrich, I., & Dressler, F. (2009). On the Lifetime of Wireless Sensor Network. *ACM Transactions on Sensor Networks, Vol. 5, No. 1*, 1-38.
- Frey, H., & Stojmenovic, I. (2005). Geographic and Energy-Aware Routing in Sensor Network. In I. Stojmenovic, *Handbook of Sensor Networks: Algorithm and Architecture* (pp. 381-415). New Jersey: John Wiley & Sons. Inc.
- Ghica, O. (2010). *SIDnet SWANS Manual*. Northwestern University.
- Ghica, O., Trajcevski, G., Scheuermann, P., Bischof, Z., & Valtchanov, N. (2007). *Demo Abstract: SIDnet-SWANS: A Simulator and Integrated Development Platform for Sensor Networks Application*. Evanston, IL, USA: Departement of EECS, Northwestern University.
- Gouda, M., & McGuire, T. (2012). Accelerated Hearbeat Protocol. *Proceedings. 18th International Conference on Distributed Computing Systems (Cat. No.98CB36183)* (pp. 202-209). Amsterdam: IEEE.
- Handy, M., Haase, M., & Timmermann, D. (2002). Low-Energy Adaptive Clustering Hierarchy with Deterministic Cluster Head Selection. Stockholm: World Scientific Publishing Co. Pte. Ltd.
- Heinzelman, W., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-Efficient Communication Protocols for Wireless Microsensor Networks. *Proceedings of the 33rd Hawaiian International Conference on Systems Science (HICSS)*.
- Imielinski, T., & Navas, J. (2016). GPS-Based Geographic Addressing, Routing, and Resource Discovery. *Communication of the ACM*.

- Jamal, A.-K., & Kamal, A. (2004). Routing Tehcniques in Wireless Sensor Network: A Survey. *IEEE Wireless Communications*.
- Jannu, S., & Jana, P. K. (2014). Energy Efficient Grid Based Clustering and Routing Algorithms for Wireless Sensor Networks. *Fourth International Conference on Communication Systems and NetworkTechnologies*, 63-68.
- Laukkarinen, T. (2017). *Computer Engineering teaching*. Retrieved from TIE-52106 Wireless Sensor Networks and Applications: www.tkt.cs.tut.fi
- Maghsoudlou, A., St-Hilaire, M., & Kunz, T. (2011). *A Survey on Geographic Routing Protocols for Mobile Ad Hoc Networks*. Technical Report SCE-11-03, Carleton Unversity.
- Natarajan, H., & Selvaraj, S. (2014). A Fuzzy Based Predictive Cluster Head Selection Scheme for Wireless Sensor Networks. *Proceedings of the 8th International Conference on Sensing Technology*. Liverpool.
- Permana, A. M. (2011). Analisa Algoritma LEACH Pada Jaringan Sensor Nirkabel. *Proceeding Seminar Tugas Akhir Jurusan Teknik Elektro FTI-ITS*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Permatasari, D., & Wibisono, W. (2014). Optimasi Pemilihan Cluster Head dengan Ant-Behavior di Lingkungan MANET. Institut Teknologi Sepuluh Nopember.
- Rajagopalan, R., & Varshney, P. K. (2006). Data Agregatin Techniques in Wireless Sensor Network: A Survey. *IEEE Communications Surveys and Tutorials*, 48-63.
- Sarkar, R., Yin, X., Jie, G., Feng, L., & Xianfeng, D. G. (2009). Greedy Routing with Guaranteed Delivery Using Ricci Flows. *Information Processing in Sensor Networks*, 121-132.
- Seggelmann, R., Tuexen, M., & Williams, M. (2013). Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension. *RFC 6520, RFC - Proposed Standard*.
- Sharma, S., Kumar, D., & Keshav, K. (2013). Wireless Sensor Networks- A Review on Topologies and Node Architecture. *International Journal of Computer Sciences and Engineering*, 19-25.

Tian, D., & Georganas, N. D. (2003). Energy efficient routing with guaranteed delivery in wireless sensor networks. *IEEE Wireless Communications and Networking*.

Youseff, M., Younis, M., & Arisha, K. (2015). A Constraint Shortest Path Energy Aware Routing Algorithm for Wireless Sensor Network. *Wireless Communications and Networking Conference*. Orlando.

[Halaman ini sengaja dikosongkan]

BIOGRAFI PENULIS



Listyanti Dewi Astuti, atau akrab disapa Dewi, adalah anak sulung dari dua bersaudara yang lahir di Pacitan, 22 Desember 1988. Penulis menempuh pendidikan dasar hingga menengah atas di kota kelahirannya, dan melanjutkan jenjang sarjana di Universitas Negeri Malang (UM) jurusan Pendidikan Teknik Informatika yang dilanjutkan dengan pendidikan magister (S2) di Institut Teknologi Sepuluh Nopember Surabaya jurusan Teknik Informatika. Selain menempuh kuliah, penulis juga merupakan Pegawai Negeri Sipil (PNS) di lingkungan Dinas Pendidikan Provinsi Jawa Timur. Penulis memiliki minat di bidang Arsitektur dan Jaringan Komputer (AJK), dan juga menekuni pembuatan aplikasi komersil sebagai hobi. Penulis dapat dihubungi melalui listyantidewi@gmail.com