



**TESIS**

**PENINGKATAN KINERJA PROTOKOL PEGASIS  
(POWER-EFFICIENT GATHERING IN SENSOR  
INFORMATION SYSTEM ) DENGAN PEMBAGIAN  
AREA DAN PEMBENTUKAN CHAIN SECARA  
ADAPTIF UNTUK PENGHEMATAN ENERGI**

**MUHAMMAD RIKO ANSHORI PRASETYA  
NRP. 05111850010040**

**Dosen Pembimbing  
Waskitho Wibisono, S.Kom.,M.Eng.,Ph.D**

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

*[Halaman ini sengaja dikosongkan]*

# LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

**Magister Komputer (M. Kom)**

di

**Institut Teknologi Sepuluh Nopember**

Oleh:

**MUHAMMAD RIKO ANSHORI PRASETYA**

**NRP: 05111850010040**

Tanggal Ujian: 16 Juli 2020

Periode Wisuda: September 2020

Disetujui oleh:

**Pembimbing:**

1. Waskhitho Wibisono, S.Kom., M.Kom., Ph.D.  
NIP: 19720809 199512 1 001

.....  .....

**Penguji:**

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

.....  .....

2. Tohari Ahmad, S.Kom., M.IT., Ph.D.  
NIP: 19750525 200312 1 002

.....  .....

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

.....  .....

Kepala Departemen Teknik Informatika  
Fakultas Teknologi Elektro dan Informatika Cerdas



.....  .....

**Dr. Eng. Hastine Fatichah, S.Kom., M.Kom**

**NIP: 19751220 200112 2 002**

*[Halaman ini sengaja dikosongkan]*

# **Peningkatan Kinerja Protokol PEGASIS (Power-Efficient Gathering In Sensor Information System) Dengan Pembagian Area dan Pembentukan Chain Secara Adaptif Untuk Penghematan Energi**

Nama Mahasiswa : Muhammad Riko Anshori Prasetya  
NRP : 05111850010040  
Pembimbing : Waskitho Wibisono. S.Kom.,M.Eng.,Ph.D

## **ABSTRAK**

*Wireless sensor networks* (WSN) merupakan sebuah perangkat jaringan nirkabel yang terdiri dari sebuah *Base Station* (BS) dan gabungan dari beberapa sensor *node* yang berguna untuk mengumpulkan informasi. WSN sendiri banyak digunakan dalam berbagai bidang seperti bidang industri, kesehatan, militer, dan pengelolaan bencana. Penggunaan WSN ini sendiri ditunjukkan untuk mendapatkan informasi terkait suhu, intensitas cahaya, getaran, panas, dan lain-lain. Dalam mengumpulkan informasi tersebut, sensor *node* tersebut membutuhkan energi dalam penggunaannya sedangkan sensor *node* sendiri memiliki energi yang terbatas. Hal ini menjadi sebuah tantangan tersendiri bagi peneliti karena penempatan sensor *node* sendiri biasanya disebar pada kondisi yang sulit dan ekstrim sehingga sulit untuk dilakukan pengisian ulang maupun pergantian sensor *node*. Salah satu langkah dalam penyelesaian masalah WSN tersebut adalah menggunakan algoritma PEGASIS yaitu sebuah algoritma *routing* protokol dengan sistem *chain-based* yang tujuannya mengirim data dari sensor yang paling jauh dari BS dengan membentuk sistem transmisi *chain* untuk mentransmisikan datanya - hingga ke BS. Namun dalam penggunaan berskala luas, protokol ini membutuhkan energi besar dalam melakukan transmisi data. Oleh karena itu, penelitian ini mengusulkan sebuah pengembangan PEGASIS, yaitu VCLUS-PEGASIS. VCLUS-PEGASIS memodifikasi area PEGASIS dengan membaginya menjadi beberapa grid secara vertikal. Selain itu, *chain* dalam satu *grid* akan diubah apabila terjadi transmisi yang panjang antara satu node dengan node tetangganya yang masih dalam satu area menggunakan algoritma K-Means. Hal tersebut bertujuan untuk meminimalisasi pengiriman energi setiap transmisi. Hasil pengujian menunjukkan jika protokol VCLUS-PEGASIS mampu untuk mempertahankan masa hidup node dan menurunkan transmisi energi daripada PEGASIS.

**Kata Kunci:** *Pegasis, cluster, pemotongan node, efisiensi energi, network lifetime, wireless sensor networks.*

# **The Enhancement of Pegasus Protocol Performance by Adaptive Area Distribution and Chain Information**

Nama Mahasiswa : Muhammad Riko Anshori Prasetya  
NRP : 05111850010040  
Pembimbing : Waskitho Wibisono. S.Kom.,M.Eng.,Ph.D

## **ABSTRACT**

Wireless Sensor Network (WSN) is a wireless network devices-based made of a Base Station (BS) and a combination of several sensor node that are construct for gathering information. WSN itself is widely used in various field such as industry, healthcare, military, and disaster management. The purpose of WSN itself is shown to obtain information related to temperature, vibration, heat, and et cetera. However, for gathering this information, the sensor node requires energy in its use while the sensor node itself has limited energy. This problem becomes a challenge for researchers because the placement of sensor nodes themselves are usually deployed in difficult and extreme conditions that making it difficult to refill or change sensor nodes. PEGASIS is one of the routing protocols with a chain-based system that is useful for solving these problems whose purpose is to send data from the sensor farthest away from the BS by forming a chain transmission system to transmit the data to the BS. But in wide-scale use, this protocol requires a lot of energy when transmitting data. Therefore, this research proposes a development of PEGASIS, namely VCLUS-PEGASIS. VCLUS-PEGASIS modifies the PEGASIS area by dividing it into vertical grids. In addition, the chain in one grid will be changed if there is a long transmission between one node and its neighbor nodes that are still in one area. It aims to minimize the energy delivery of each transmission using the K-Means algorithm. The test result shows that the VCLUS-PEGASIS protocol is able to maintain the life span of nodes and decrease energy transmission than those of PEGASIS.

**Keywords:** PEGASIS, cluster, node cutting, energy efficiency, network lifetime, wireless sensor networks.

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ  
Bismillahirrohmanirrohim.

*Alhamdulillah* *robbil'alamin*, Puji syukur penulis panjatkan kehadiran Allah SWT. dan *shalawat* serta salam semoga senantiasa tercurahkan kepada *Rasulullah* Muhammad SAW. sehingga penulis dapat menyelesaikan penelitian tesis dengan judul **“Peningkatan Kinerja Protokol PEGASIS (Power-Efficient Gathering in Sensor Information System) Dengan Pembagian Area dan Pembentukan Chain Secara Adaptif Untuk Penghematan Energi”** dengan baik. Pada kesempatan ini, dengan segala kerendahan hati penulis ingin mengucapkan ribuan terima kasih dan penghargaan setinggi-tingginya karena tanpa dukungan, dorongan, bimbingan dan bantuannya sehingga penulis dapat menyelesaikan penelitian tesis ini dengan baik, antara lain kepada :

1. Orang tua, I Gusti Ketut Prasetya, Meirina Yani yang selalu memberikan dorongan, semangat, do'a, serta motivasi kepada penulis agar dapat menyelesaikan tesis ini.
2. Orang tua sambung, (alm) Mahriyan, Dewy Hastuty yang juga memberikan dorongan, doa, serta motivasi kepada penulis agar dapat menyelesaikan tesis ini.
3. Bapak Waskitho Wibisono, S.Kom., M. Eng., Ph.D, selaku dosen pembimbing yang telah banyak memberikan arahan serta mendidik dan membimbing penulis dengan sabar dalam menyelesaikan penelitian tesis ini.
4. Sahabat-sahabat W5 terutama Arif Mudi Priyatno, Awwib Hasana, Dino yang selama ini sangat membantu dalam memotivasi dan menyemangati penulis dalam menyelesaikan tesis ini.
5. Kepada Adam Safri, Cahyaningtyas Sekar Wahyuni, Yohanes Setiawan yang selama ini membantu dalam hal penulisan buku tesis ini.

6. Kepada adik-adik Ratu Kumalasari, Muhammad Rizky Akmal Prasetya, Aliyya Namira Prasetya yang selalu memberikan dukungan dan semangat dalam menempuh pendidikan magister ini.
7. Dekan Fakultas Teknologi Elektro dan Informatika Cerdas, Ketua Departemen Informatika, Ketua dan Sekretaris Program Pascasarjana Informatika yang telah memberikan izin menggunakan fasilitas kampus kepada penulis sehingga dapat menyelesaikan pendidikan magister ini.
8. Seluruh dosen, staf laboratorium, staf tata usaha, dan karyawan departemen informatika ITS.
9. Teman seperjuangan mahasiswa s2 TC ITS angkatan 2018.
10. Seluruh pihak yang tidak dapat disebutkan satu-persatu atas dukungan baik moral dan materil.

Semoga Allah SWT senantiasa memberikan limpahan karunia-Nya dan memberikan balasan pahala yang berlipat ganda kepada semuanya atas semua kebaikan yang telah diberikan kepada penulis. Penulis menyadari bahwa laporan penelitian ini masih jauh dari kesempurnaan. Oleh sebab itu, saran dan masukan yang membangun sangat diharapkan untuk perbaikan dimasa yang akan datang. Semoga penelitian ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan dan dapat memberi kontribusi bagi penelitian selanjutnya.

Surabaya, 28 Januari 2020

Muhammad Riko Anshori Prasetya

## DAFTAR ISI

|   |      |
|---|------|
| LEMBAR PENGESAHAN TESIS .....   | IV   |
| ABSTRAK .....   | VI   |
| ABSTRACT .....  | VII  |
| KATA PENGANTAR .....  | VIII |
| DAFTAR ISI .....  | X    |
| DAFTAR GAMBAR .....   | XII  |
| DAFTAR TABEL .....  | XIII |
| BAB 1 PENDAHULUAN .....   | 1    |
| 1.1 LATAR BELAKANG .....  | 1    |
| 1.2 PERUMUSAN MASALAH .....   | 4    |
| 1.3 TUJUAN PENELITIAN .....   | 4    |
| 1.4 MANFAAT PENELITIAN .....  | 4    |
| 1.5 KONTRIBUSI PENELITIAN .....   | 5    |
| 1.6 BATASAN MASALAH .....   | 5    |
| BAB 2 .....   | 7    |
| 2.1 <i>WIRELESS SENSOR NETWORK</i> (WSN) .....                              | 7    |
| 2.2 <i>DATA GATHERING</i> PADA WSN ( <i>WIRELESS SENSOR NETWORK</i> ) ..... | 8    |
| 2.3 LEACH .....   | 9    |
| 2.2.1 <i>Set-up Fase</i> .....  | 10   |
| 2.2.2 <i>Steady-State Fase</i> .....  | 11   |
| 2.4 PEGASIS .....   | 12   |
| 2.3.1 <i>Fase Pembentukan Chain</i> .....                                   | 13   |
| 2.3.2 <i>Fase Pengumpulan Data</i> .....                                    | 14   |
| 2.5 <i>CLUSTERING</i> .....   | 15   |
| 2.2.3 <i>Algoritma K-Means</i> .....  | 16   |
| BAB 3 METODE PENELITIAN .....   | 19   |
| 3.1 METODOLOGI PENELITIAN .....   | 19   |
| 3.2 TAHAPAN VCLUS-PEGASIS .....   | 20   |
| 3.2.1 <i>Fase Pembagian Area Vertikal</i> .....                             | 21   |

|   |   |           |
|---|---|-----------|
| 3.2.2                                   | <i>Fase Pembentukan Chain</i> .....       | 21        |
| 3.2.3                                   | <i>Fase Pembentukan Leader</i> .....      | 24        |
| 3.3                                     | MODEL JARINGAN.....                       | 26        |
| 3.4                                     | RADIO DAN MODEL ENERGI.....               | 27        |
| 3.5                                     | UJI COBA DAN EVALUASI.....                | 28        |
| 3.6                                     | DOKUMENTASI DAN JADWAL PENELITIAN.....    | 30        |
| <b>BAB 4 HASIL DAN PEMBAHASAN</b> ..... |   | <b>33</b> |
| 4.1.                                    | HASIL.....                                | 33        |
| 4.1.1                                   | <i>Tahap Penelitian</i> .....             | 33        |
| 4.1.2                                   | <i>Lingkungan Uji Coba</i> .....          | 34        |
| 4.1.3                                   | <i>Tahapan Uji Coba</i> .....             | 34        |
| 4.2.                                    | MODIFIKASI PADA PROTOKOL PEGASIS.....     | 35        |
| 4.2.1                                   | <i>Fase Pembagian Area Vertikal</i> ..... | 35        |
| 4.2.2                                   | <i>Fase Pembentukan Chain</i> .....       | 36        |
| 4.2.3                                   | <i>Fase Pembentukan Leader</i> .....      | 40        |
| 4.3.                                    | ANALISIS HASIL.....                       | 41        |
| 4.3.1                                   | <i>Penentuan Nilai Threshold</i> .....    | 41        |
| 4.3.2                                   | <i>Analisa Skenario 1</i> .....           | 42        |
| 4.3.3                                   | <i>Analisa Skenario 2</i> .....           | 48        |
| 4.3.4                                   | <i>Analisa Skenario 3</i> .....           | 54        |
| <b>BAB 5 KESIMPULAN DAN SARAN</b> ..... |   | <b>57</b> |
| 5.1                                     | KESIMPULAN.....                           | 57        |
| 5.2                                     | SARAN.....                                | 58        |
| <b>DAFTAR PUSTAKA</b> .....             |   | <b>60</b> |

## DAFTAR GAMBAR

|             |   |    |
|-------------|---|----|
| Gambar 2. 1 | Arsitektur Standar WSN (Afif 2017) .....  | 7  |
| Gambar 2. 2 | <i>Data Gathering</i> Pada WSN (Chong et al. 2009).....   | 9  |
| Gambar 2. 3 | Ilustrasi LEACH Protokol (Sharma, Swati 2015).....  | 10 |
| Gambar 2. 4 | Fase Pada LEACH (Sharma, Swati 2015) .....  | 10 |
| Gambar 2. 5 | Ilustrasi PEGASIS Protokol (Lindsey and Raghavendra 2002).....  | 13 |
| Gambar 2. 6 | Pembentukan <i>Chain</i> dengan <i>Greedy Alghorithm</i> .....  | 14 |
| Gambar 2. 7 | Alur Pengiriman Data PEGASIS Protokol.....  | 15 |
| Gambar 2. 8 | <i>Clustering</i> (Ridwan, Ferdian, and Rahmadi Kurnia 2021) .....  | 16 |
| Gambar 2. 9 | Algoritma K-Means. (a)Dataset Original. (b) <i>Random cluster centroid</i> . (c)Iterasi K-means dengan <i>centroid</i> terbaik..... | 18 |
| Gambar 3. 1 | Alur Tahapan Penelitian.....  | 20 |
| Gambar 3. 2 | Alur Perancangan VCLUS-PEGASIS .....  | 20 |
| Gambar 3. 3 | Area Vertikal .....   | 21 |
| Gambar 3. 4 | Alur Fase Pembentukan <i>Chain</i> pada VCLUS-PEGASIS.....  | 22 |
| Gambar 3. 5 | Skenario Pemotongan <i>Node Area Vertikal</i> .....   | 23 |
| Gambar 3. 6 | Skenario Pembentukan <i>Chain Baru Area Vertikal</i> .....  | 24 |
| Gambar 3. 7 | Fase Pembentukan <i>Leader Round 1</i> .....  | 25 |
| Gambar 3. 8 | Fase Pembentukan <i>Leader Round 2</i> .....  | 26 |
| Gambar 3. 9 | Radio Model (Anjum and Ahmad 2017).....   | 27 |

## DAFTAR TABEL

|  |    |
|--|----|
| <i>Tabel 3. 1 Parameter Uji Coba</i> .....   | 29 |
| <i>Tabel 3. 2 Parameter Evaluasi</i> .....   | 30 |
| <i>Tabel 3. 3 Jadwal Kegiatan Penelitian</i> .....   | 31 |
| <i>Tabel 4. 1 Lingkungan Uji Coba</i> .....  | 34 |
| <i>Tabel 4. 2 Parameter Pengujian Skenario</i> .....   | 35 |
| <i>Tabel 4. 3 Pseudocode Fase Area Vertikal</i> .....  | 35 |
| <i>Tabel 4. 4 Pseudocode 1 Hitung Jarak dengan Tetangga, Nilai Rata-Rata, dan Total Jarak Antar Node</i> ..... | 36 |
| <i>Tabel 4. 5 Pseudocode 2 Cek Threshold dan Pembentukan Chain dengan Algoritma Kmeans</i> .....               | 38 |
| <i>Tabel 4. 6 Pseudocode Pembentukan Leader</i> .....  | 40 |
| <i>Tabel 4. 7 Perbandingan Uji Coba Threshold</i> .....  | 42 |
| <i>Tabel 4. 8 FND pada Skenario 1</i> .....  | 44 |
| <i>Tabel 4. 9 HND pada Skenario 1</i> .....  | 45 |
| <i>Tabel 4. 10 Persentase Hidup Node pada Skenario 1</i> .....   | 46 |
| <i>Tabel 4. 11 Average Energy Consumed by A Node 100 node Skenario 1</i> .....                                 | 48 |
| <i>Tabel 4. 12 Average Energy Consumed by A Node 200 node Skenario 1</i> .....                                 | 48 |
| <i>Tabel 4. 13 FND pada Skenario 1</i> .....   | 49 |
| <i>Tabel 4. 14 HND pada Skenario 1</i> .....   | 51 |
| <i>Tabel 4. 15 Persentase Hidup Node pada Skenario 2</i> .....   | 51 |
| <i>Tabel 4. 16 Average Energy Consumed by A Node 100 node Skenario 2</i> .....                                 | 54 |
| <i>Tabel 4. 17 Average Energy Consumed by A Node 200 node Skenario 2</i> .....                                 | 54 |

*[Halaman ini sengaja dikosongkan]*

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Kemajuan teknologi di bidang *Micro Electro Mechanical System* (MEMS) mendorong peningkatan penggunaan dari *Wireless Sensor Network* (WSN). WSN merupakan sebuah perangkat jaringan nirkabel yang terdiri dari sebuah *Base Station* (BS) dan gabungan dari banyak sensor *node* yang berguna untuk mengumpulkan informasi lalu mengirimkannya ke BS. Sensor *node* sendiri akan melakukan proses *data gathering* (pengumpulan data) dari audio, gelombang seismik, dan macam-macam data lainnya untuk mendapatkan informasi tersebut lalu mengirimkan data tersebut ke BS (Warrier and Kumar 2016). Saat ini, WSN sendiri mendapat perhatian besar dengan digunakannya aplikasi WSN didalam berbagai bidang seperti bidang industri, kesehatan, militer, dan pengelolaan bencana. Namun dari sekian banyak keunggulan yang dimiliki WSN, WSN sendiri memiliki banyak keterbatasan dalam penggunaannya. Sensor *node* dalam WSN biasanya disebarkan dalam lingkungan yang sulit dan ekstrim sehingga sulit untuk dilakukan pengisian ulang maupun pergantian sensor *node* ketika energinya habis sementara sensor *node* sendiri memiliki kapasitas dan daya yang sangat terbatas. Oleh karena itu, cara untuk memperpanjang masa hidup dan juga untuk mengurangi konsumsi energi dari sensor *node* ini masalah yang sangat penting dalam bidang WSN (Guo, Zhang, and Lu 2010).

Berbagai langkah dan juga strategi telah banyak dilakukan dalam menyelesaikan permasalahan dalam *data gathering* pada WSN tersebut. Hal tersebut dapat dilihat dengan banyaknya protokol *routing* yang telah diusulkan. Berdasarkan dari topologi jaringan, protokol *routing* ini dapat diklasifikasikan menjadi *flat-based* dan *hierarchical-based*. Dalam pengklasifikasian ini *hierarchical-based* protokol memberikan energi yang lebih hemat dalam penggunaannya dibandingkan dari *flat-based* protokol dikarenakan *flat-based* protokol membutuhkan lebih banyak data *aggregation* yang disebabkan dari

pengiriman data yang *overlapping* antara sesama *node* tetangga (Jung, Han, and Chung 2007). Ide dari *hierarchical-based* protokol adalah membagi *node* tersebut menjadi beberapa *cluster*. Dari beberapa *cluster* ini, salah satu sensor *node* tersebut dipilih menjadi *leader* untuk menggabungkan berbagai informasi dari *node* yang lain lalu *leader node* ini mengirimkannya ke BS. Sensor *node* sendiri mengambil informasi hanya dalam wilayah *cluster* tersebut lalu mengirimkannya ke *leader node* (Sen, Bing, and Liangrui 2011). Salah satu *hierarchical-based* protokol yang paling banyak dibahas adalah *Low-Energy Adaptive Clustering Hierarchy* (LEACH) (Heinzelman, Chandrakasan, and Balakrishnan 2000).

LEACH merupakan sebuah protokol *routing* pada WSN yang paling awal dalam menerapkan *cluster-based protocol* yang bertujuan untuk mengurangi konsumsi energi sewaktu melakukan pengiriman antara sensor *node* dan BS. Tujuan utama dari protokol ini adalah untuk membentuk *cluster* secara dinamis lalu dalam *cluster* tersebut dipilih salah satu sensor *node* sebagai *Cluster Head* (CH) yang berfungsi untuk menggabungkan data dari anggota *node* dan CH tadi mengirimkannya ke BS (Rana, Vhatkar, and Mohommad Atique 2015). Namun dalam kedinamisan *clustering* dari LEACH dibutuhkan tambahan energi dikarenakan dilakukan pengelompokkan *clustering* tersebut dilakukan pada setiap iterasi. Selain itu CH yang lebih jauh dari BS cenderung menghabiskan energi dengan lebih cepat (Ramluckun and Bassoo 2018). Dalam menyelesaikan permasalahan tersebut, PEGASIS (*Power-Efficient Gathering in Sensor Information Systems*) dibuat sebagai pengembangan dari LEACH protokol (Lindsey and Raghavendra 2002).

PEGASIS adalah sebuah protokol dengan sistem *chain-based* yang berbeda daripada LEACH yang menggunakan *cluster*. Didalam PEGASIS, formasi *chain* dimulai dari *node* yang paling jauh dari BS. Setiap *node* tadi mengirimkan data ke *node* tetangganya yang paling dekat hingga sampai ke BS dengan sistem *chain*. Salah satu dari *node* di dalam *chain* tersebut akan menjadi *leader* untuk mengirimkan data ke BS (Y. Chen and Lin 2012). *Leader* ini sendiri dipilih secara merata agar energi yang dikeluarkan juga merata antara satu *node* dengan *node* lainnya. Proses ini diulang pada setiap iterasinya. Tujuan dari PEGASIS sendiri adalah meningkatkan *network lifetime* dan juga mengurangi jarak transmisi antar

node. Performa dari PEGASIS sendiri menghemat daya dua kali lebih banyak dari pada LEACH (Jung, Han, and Chung 2007). Namun, *routing* berbasis *chain-base* sendiri memiliki kekurangan seperti delay yang besar dan juga komunikasi yang sering *overhead* (Ramluckun and Bassoo 2018).

Beberapa penelitian tentang PEGASIS sendiri menggabungkan antara teknik *routing* dengan menggunakan *chain-based* dan *clustering-based* untuk mengurangi kelemahan dari kedua metode ini. Penelitian (K.-H. Chen, Huang, and Chieh-Chuan Hsiao 2009) melakukan pembagian pada *sensing area* menjadi beberapa area kecil dengan menggunakan konsep BeamStar dan menamakan penelitiannya sebagai CHIRON. Setiap *clustering* tadi membentuk PEGASIS *chain* yang baru. Penelitian ini mempertimbangkan semakin panjang *chain* yang terbentuk maka akan semakin besar pula energi untuk melakukan transmisi datanya. Metode CHIRON ini memberikan performa yang lebih baik daripada PEGASIS.

Penggunaan *clustering* pada PEGASIS juga diteliti oleh (Ramluckun and Bassoo 2018) yang idenya adalah membangun *clustering* berdasarkan titik koordinat x dari *nodes* yang dinamakan *PEG-ACO Vertical Clustering* dan membangun *clustering* berdasarkan koordinat y dari *nodes* yang dinamakan *PEG-ACO Horizontal Clustering*. *Chain* sendiri dibuat pada setiap *clustering* dan mengadopsi transmisi yang multihop untuk menuju ke BS. CH pada setiap *chain* sendiri ditentukan berdasarkan perbandingan antara energi dan juga jarak. Performa dari metode ini sendiri meningkatkan *network lifetime*, sisa energi, dan mengurangi delay pada setiap transmisinya.

Dari beberapa penelitian yang dibahas berkaitan dengan pengembangan PEGASIS ini didapatkan suatu permasalahan dimana akan ada suatu transmisi yang panjang diantara *node* dengan *node* tetangga ketika terbentuk suatu *chain* yang menyebabkan energi yang dikeluarkan dalam *node* tersebut dalam melakukan transmisi akan menjadi lebih besar sehingga diperlukannya tambahan *chain* secara otomatis dalam satu area apabila terjadi kasus dimana transmisi yang panjang antara satu *node* dengan *node* tetangganya terjadi.

Dalam penelitian ini diusulkan juga pengembangan untuk meningkatkan kinerja dari protokol PEGASIS yaitu VCLUS-PEGASIS. VCLUS-PEGASIS memodifikasi *chain* dalam satu area apabila terjadi transmisi yang panjang antara

satu *node* dengan *node* tetangganya dalam satu area. Selain itu akan digunakan pembagian *node* dalam beberapa *grid* dengan bentuk vertikal agar mengurangi energi dalam transmisinya.

## 1.2 Perumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini berdasarkan dari latar belakang yang ada adalah sebagai berikut :

1. Bagaimana merancang modifikasi PEGASIS dengan pembagian *node* menjadi beberapa area dan melakukan pembentukan *chain* secara adaptif serta menambahkan *leader* apabila terjadi transmisi yang panjang antar *node* tetangga untuk meminimalkan dan memperpanjang masa hidup dari *node* dalam lingkungan PEGASIS?
2. Bagaimana pengaruh penerapan ide diatas untuk memodifikasi protokol PEGASIS dengan mempertimbangkan parameter-parameter seperti *alive node in network* dan *average consumption energy per transmission*?

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah membangun protokol baru untuk meningkatkan kinerja dari PEGASIS dengan cara:

1. Meneliti pengaruh dari pembagian *node* menjadi beberapa area dalam lingkungan PEGASIS.
2. Melakukan pembentukan *chain* secara adaptif dan menambahkan *leader* dalam satu area apabila terjadi transmisi yang panjang antar *node* tetangga.
3. Membangun modifikasi PEGASIS menggunakan simulasi MATLAB dengan mempertimbangkan parameter *alive node in* dan *average consumption energy per transmission*.

## 1.4 Manfaat Penelitian

Manfaat dari hasil penelitian ini diharapkan dapat dikembangkannya protokol PEGASIS dalam penggunaannya untuk mengurangi konsumsi energi yang dikeluarkan ketika melakukan transmisi dan menambah masa hidup *node*.

## 1.5 Kontribusi Penelitian

Kontribusi dari penelitian ini adalah menambah serta melihat pengaruh dengan melakukan pembagian *node* menjadi beberapa area, membentuk *chain* secara adaptif pada satu *node* dengan *node* tetangganya apabila terjadi transmisi panjang didalam satu area dan menambahkan *leader* pada setiap *chain* dalam satu area untuk meminimalkan energi pada saat transmisi pada protokol PEGASIS.

## 1.6 Batasan Masalah

Batasan masalah pada penelitian ini adalah :

1. Pembagian area grid dilakukan secara manual dengan teknik vertikal.
2. Pembentukan *chain* dalam penelitian ini masih menggunakan algoritma *greedy*.
3. Iterasi pada saat pembentukan *chain* pada satu *cluster* hanya dilakukan sekali.
4. Simulasi dibangun dalam lingkungan MATLAB developer.

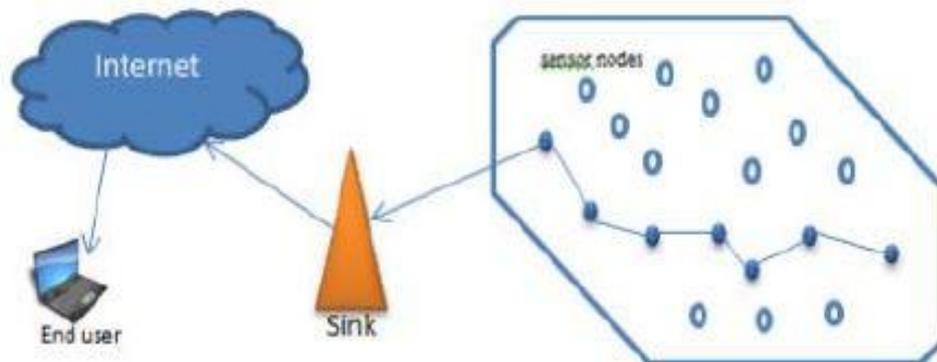
*[halaman ini sengaja dikosongkan]*

## BAB 2

### KAJIAN PUSTAKA DAN DASAR TEORI

#### 2.1 *Wireless Sensor Network (WSN)*

*Wireless Sensor Network (WSN)* merupakan sebuah model jaringan yang terdiri atas beberapa *device* yang saling berkomunikasi melalui sebuah jaringan nirkabel. WSN sendiri memiliki ratusan sensor *node* dan juga sebuah BS atau *sink*, dimana sensor *node* tadi memiliki kemampuan untuk melakukan *sensing*, *processing*, komunikasi, dan juga transmisi. (Afif 2017). Seperti terlihat pada Gambar 2.1, sensor *node* ini melakukan *sensing* untuk mendapatkan data (*data gathering*) dari beberapa faktor seperti suhu, kelembaban, tekanan, gerakan, dan beberapa variabel lainnya pada area yang ditempatkan sensor *node* tersebut lalu mentransmisikannya ke sensor yang lain dan terakhir mentransmisikannya ke BS. BS didalam WSN sendiri memiliki fungsi sebagai penerima informasi (Li 2016). Saat ini, WSN sendiri mendapat perhatian besar dengan digunakannya aplikasi WSN didalam berbagai bidang seperti bidang industri, militer, monitoring gedung, perawatan kesehatan, pertanian, pemantauan lingkungan, pengelolaan bencana, dan lain-lain (Guo, Zhang, and Lu 2010).



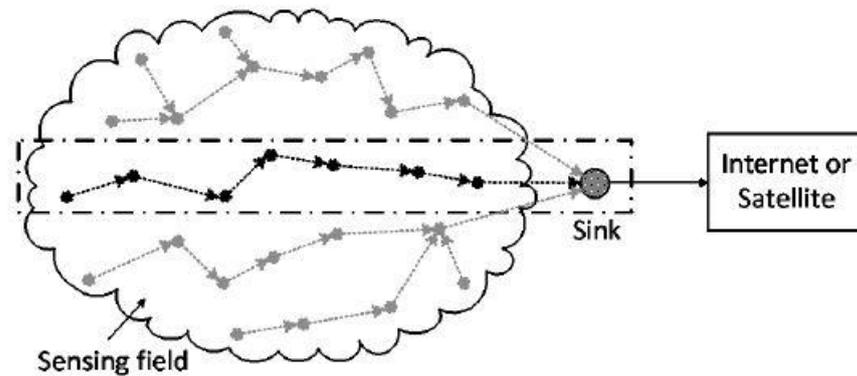
Gambar 2. 1 Arsitektur Standar WSN (Afif 2017)

Meskipun memiliki kelebihan yang banyak, WSN sendiri memiliki keterbatasan terutama dalam permasalahan energi yang dipakai dalam sensor *node*. Sementara itu, dalam beberapa aplikasi seperti *monitoring* dan perawatan

kesehatan, sensor *node* cenderung melakukan transmit data yang redundansi atau informasi yang sering sama ke BS yang menyebabkan energi yang dipakai cenderung lebih banyak, membuang-buang *bandwidth*, dan mengurangi kapasitas jaringan (Jung, Han, and Chung 2007). Oleh karena itu, untuk meningkatkan kapasitas dari energi dan menambah kapasitas jaringan, penelitian di bidang WSN sangat dibutuhkan didalam bidang pengumpulan data (*data gathering*) ini.

## **2.2 Data Gathering pada WSN (Wireless Sensor Network)**

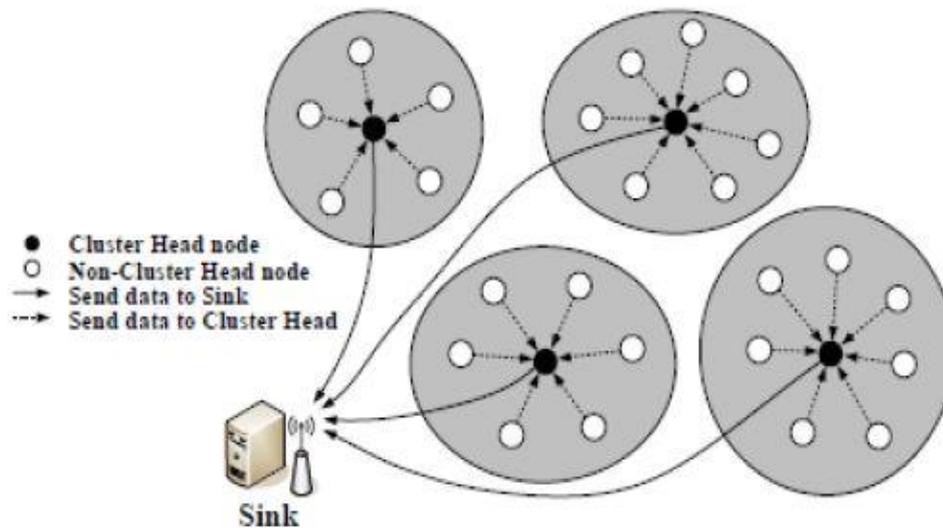
Dalam WSN, mengirimkan data dari *node* langsung ke BS akan menyebabkan beberapa permasalahan terutama berkaitan dengan penghematan energi terutama dalam skala yang besar. Beberapa masalah tersebut yaitu kualitas data yang kurang akibat beberapa paket yang dikirim hilang pada saat proses transmisi data, data yang terkumpul di BS akan terdapat banyak redundansi yang terkadang dapat menyebabkan *overhead*, dan menyebabkan baterai cepat habis karena membutuhkan lebih banyak energi pada saat melakukan pengiriman data sehingga dibutuhkan suatu cara agar meminimalkan penggunaan energi dan untuk meningkatkan masa hidup dari sensor *node* tersebut (A.Rajeswari and R.Manavalan 2014). Berbagai penelitian dalam bidang WSN tentang langkah dalam *data gathering* banyak menghasilkan algoritma. Tujuan *data gathering* sendiri adalah melakukan *sensing* data dari setiap sensor *node* dan mengumpulkan data tersebut menjadi satu lalu mengirimkannya ke BS. Adapun algoritma untuk *data gathering* sendiri bertujuan agar memperpanjang masa hidup dari sensor *node* tersebut dengan cara meminimumkan penggunaan energi dan transmisi yang dilakukan sehingga meminimalkan *delay* yang terjadi (Agrawal and Shivendra 2013). Beberapa algoritma yang populer dalam bidang ini antara lain LEACH dan juga PEGASIS. Penelitian ini sendiri akan berfokus pada modifikasi dari LEACH protokol yaitu PEGASIS protokol.



Gambar 2. 2 *Data Gathering* Pada WSN (Chong et al. 2009)

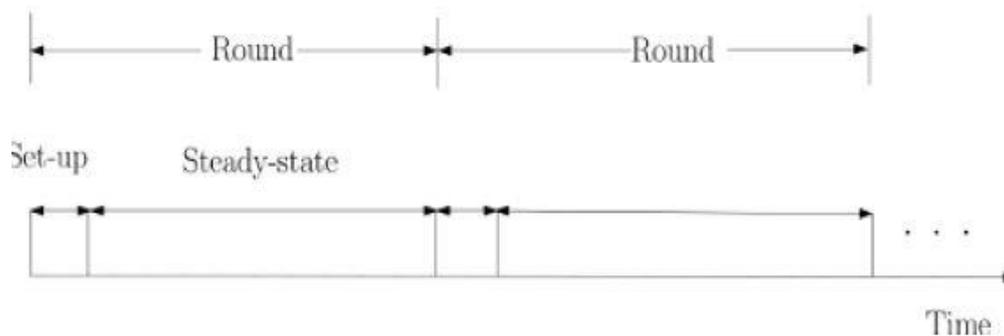
### 2.3 LEACH

LEACH (*Low Energy Adaptive Clustering Hierarchy*) merupakan sebuah protokol pertama berbasis *clustering* dan merupakan algoritma *hierarchical* yang paling populer dengan tujuan sebagai peningkatan efisiensi energi yang digunakan dalam WSN (Akkari, Bouhdid, and Belghith 2015). LEACH mengasumsikan jika posisi BS didalam tetap dan berlokasi jauh dari sensor. Semua sensor *node* sendiri diasumsikan dengan sensor *node* yang homogen dan *node* tadi memiliki sumber daya energi yang terbatas, lalu sensor *node* dapat melakukan *sensing* pada area yang sama, dapat berkomunikasi dengan sensor *node* lainnya, dan sensor dapat berkomunikasi langsung dengan BS (Sunil Kumar Singh, Kumar, and Jyoti Prakash Singh 2017). Konsep utama dari *LEACH* adalah mengorganisir seluruh *node* menjadi sebuah *cluster* agar energi yang terpakai menjadi sama rata di lingkungan *cluster* tersebut lalu setiap *cluster* tadi memilih satu *node* sebagai CH (*Cluster Head*). CH pada dipilih dengan menggunakan teknik rotasi sehingga energi yang dikeluarkan pada setiap sensor *node* di WSN berkurang. CH pada LEACH juga melakukan penggabungan data agar data tersebut menjadi lebih kecil sebelum data ini dikirim CH ke BS agar mengurangi pemborosan pada saat pemakaian energi dan meningkatkan masa hidup dari sensor *node* (Hani and Ijeh 2013). Adapun ilustrasi dari LEACH dapat dilihat pada Gambar 2.2 berikut:



Gambar 2. 3 Ilustrasi LEACH Protokol (Sharma, Swati 2015)

Di setiap *round* dalam LEACH terdapat dua buah fase seperti terlihat pada Gambar 2.3. *Cluster* di dalam LEACH akan terbentuk pada saat *set-up* fase dan data transfer akan terjadi pada saat *steady-state* fase.



Gambar 2. 4 Fase Pada LEACH (Sharma, Swati 2015)

### 2.2.1 Set-up Fase

Pada saat permulaan *set-up* fase, seluruh *node* selain CH akan memilih nomor antara 0 dan 1 lalu akan dibandingkan dengan menggunakan formula *threshold*. Jika nomor yang dipilih oleh sensor *node* tersebut kurang dari *threshold* maka *node* tersebut akan menjadi CH. Persamaan dari *threshold* sendiri dapat dilihat pada persamaan (1)

$$T(n) = \begin{cases} \text{if } n \in G & \frac{P}{1 - P * (r \bmod \frac{1}{p})} \\ \text{Jika tidak} & 0 \end{cases} \quad (1)$$

Dalam persamaan 1, P diartikan sebagai persentase yang diinginkan oleh LEACH terhadap sensor *node* untuk menjadi CH dari seluruh sensor *node* (LEACH protokol menspesifikasikan sensor *node* memiliki 5% probabilitas menjadi CH), r adalah *round* fase saat ini, dan G adalah anggota dari sensor *node* yang belum menjadi CH selama  $1/p$  *round*. Sensor *Node* yang menjadi CH pada round 0 tidak diperbolehkan lagi menjadi CH pada  $1/p$  *round* yang selanjutnya. Setelah  $1/p - 1$  *round*, nilai dari *threshold* akan menjadi  $T(n) = 1$  dan semua sensor *node* tadi berhak menjadi CH kembali. Dengan langkah ini, setiap *node* akan memiliki kesempatan untuk menjadi CH dan energi yang terpakai akan terbagi secara rata kepada semua *node*. Setelah pembentukan *cluster* selesai dilakukan, setiap CH membuat sebuah jadwal TDMA (*Time Division Multiple Access*) dan akan mengirimkan jadwal ini ke seluruh member dari *cluster* itu. Penjadwalan TDMA sendiri akan menghindari tabrakan data yang dikirim dari anggota *node* dan mengizinkan beberapa sensor *node* yang sudah melakukan *sensing* memasuki *sleep mode*. Setelah semua *node* mengetahui penjadwalan TDMA tersebut, maka *set-up* fase akan selesai dan memasuki fase yang berikutnya yaitu *steady-state* fase (Sunil Kumar Singh, Kumar, and Jyoti Prakash Singh 2017).

### 2.2.2 Steady-State Fase

Pada *steady-state* fase, pengiriman data *sensing* dari setiap *node* dari anggota *cluster* ke CH dan CH ke BS akan dilakukan menggunakan penjadwalan TDMA. Anggota dari *cluster* akan mengirim data ke CH hanya pada waktu yang sudah dialokasikan pada penjadwalan TDMA. Ketika salah satu dari anggota *node* mengirimkan data ke CH, maka anggota *node* lain akan memasuki *sleep mode*. Akibat dari *node* yang memasuki *sleep mode* ini, maka tabrakan pada saat pengiriman data di dalam *cluster* ini akan berkurang dan akan meningkatkan masa hidup dari baterai pada sensor *node*. Setelah itu, CH akan menggabungkan data dari semua anggota *node* yang ada di dalam *cluster* tersebut lalu CH tadi akan mengirimkan data itu ke BS. Pengiriman data CH ke BS juga masih menggunakan

penjadwalan TDMA. Pada tahap ini, ketika BS sedang sibuk untuk menerima data dari salah satu CH, maka CH lainnya akan menunggu antrian tersebut (Sunil Kumar Singh, Kumar, and Jyoti Prakash Singh 2017).

Hasil dari kedua fase tersebut didapatkan jika penggunaan LEACH pada WSN termasuk efektif dalam menambah daya baterai dari sensor *node* tersebut, namun meskipun begitu, LEACH sendiri memiliki beberapa kekurangan yaitu (Rana, Vhatkar, and Mohommad Atique 2015):

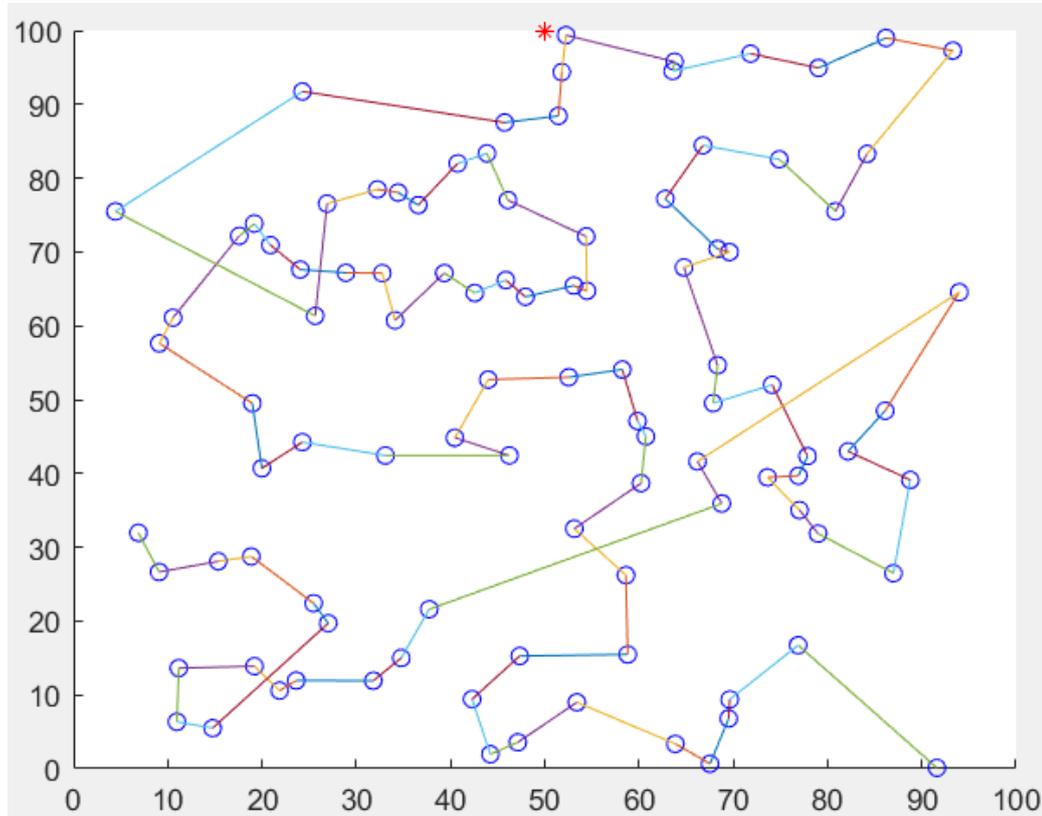
- LEACH menggunakan *single-hop* pada dari CH menuju BS, sehingga tidak cocok untuk lingkungan yang luas.
- Karena pemilihan CH sendiri berdasarkan probabilitas, menyebabkan CH tidak terlalu terbagi secara merata pada *cluster*.
- Akibat *clustering* yang dinamis akan menyebabkan tambahan *overhead*.

Untuk meningkatkan efisiensi dari kekurangan dari LEACH protokol tersebut, PEGASIS (*Power-Efficient Gathering in Sensor Information Systems*) dibuat sebagai pengembangan dari LEACH protokol (Lindsey and Raghavendra 2002).

## 2.4 PEGASIS

PEGASIS (*Power-Efficient Gathering in Sensor Information Systems*) merupakan sebuah protokol berbasis *chain*. PEGASIS sendiri diklasifikasikan sebagai *hierarchical* protokol yang merupakan peningkatan dari LEACH protokol (Linping 2010). Pada PEGASIS, setiap sensor *node* ditaruh secara acak dan setiap *node* memiliki kemampuan untuk mendeteksi data, menggabungkan data, dan mengetahui posisi dari *node* lainnya (Rana, Vhatkar, and Mohommad Atique 2015). Konsep utama dari PEGASIS adalah untuk menciptakan *chain* dari sensor *node* yang ada sehingga setiap sensor *node* dapat menerima dan mengirimkan data ke sensor *node* tetangga terdekatnya. Selain itu, pada salah satu dari sensor *node* didalam *chain* akan menjadi sebuah *leader* dengan tugas utama untuk mengirimkan data langsung ke BS. *Leader* didalam PEGASIS sendiri terpilih berdasarkan urutan agar konsumsi energinya menjadi sama rata. Performa protokol PEGASIS

diketahui lebih baik dari pada protokol LEACH (Prajapat and Barwar 2018). Adapun ilustrasi pada PEGASIS dapat dilihat pada Gambar 2.5.



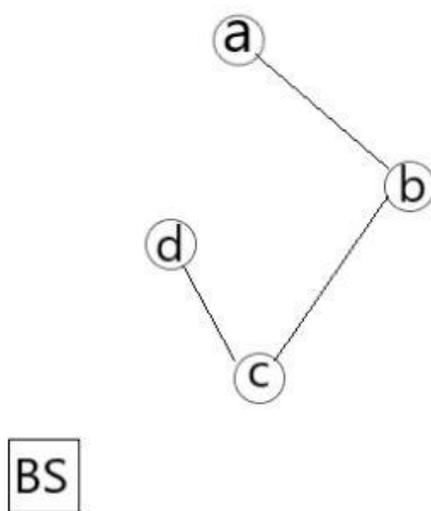
Gambar 2. 5 Ilustrasi PEGASIS Protokol (Lindsey and Raghavendra 2002)

Untuk membentuk *chain* dan melakukan pengiriman data Protokol PEGASIS juga dibagi dalam beberapa fase yaitu:

### 2.3.1 Fase Pembentukan *Chain*

Dalam fase pembentukan *chain*, PEGASIS memiliki dua buah langkah. Langkah pertama dalam pembentukan *chain* adalah sensor *node* dan juga BS akan menata *chain* dengan menggunakan algoritma *greedy* seperti yang terlihat pada Gambar 2. Berikutnya, BS akan mulai melakukan *broadcast* informasi ke semua *node* setelah *node* tadi menata *chain* tersebut. Langkah terakhir adalah *chain* akan terbentuk dengan mempertimbangkan sensor *node* yang paling jauh dari BS sebagai titik awalnya (Du, Wu, and Zhou 2003). Pada Gambar 2.6. Terlihat jika *node* a terletak paling jauh dari BS, sehingga bisa dikatakan jika *node* a merupakan titik

awal, *node* a sendiri akan mengirimkan data kepada *node* tetangganya yang paling dekat yaitu *node* b. Langkah-langkah ini akan terulang hingga semua *node* tadi membentuk sebuah *chain*. Pada tahap ini setiap *node* menggunakan kekuatan sinyal untuk mengukur jarak dengan tetangganya dan menyesuaikan kekuatan sinyalnya dengan tetangga tersebut sehingga hanya tetangga tersebut yang dapat menerima pesan tersebut (Jung, Han, and Chung 2007).



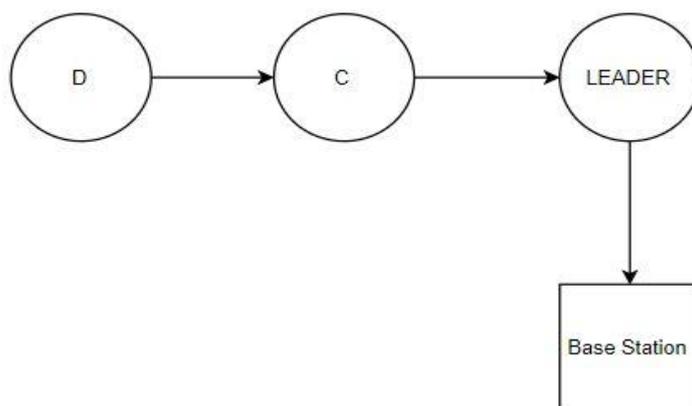
Gambar 2. 6 Pembentukan *Chain* dengan *Greedy Algorithm*

### 2.3.2 Fase Pengumpulan Data

Setelah PEGASIS berhasil dalam proses pembentukan *chain*, setiap *node* akan mengirimkan data *sensing* mereka ke tetangga terdekatnya. Setelah menerima data tersebut, *node* tetangga tadi akan melakukan penggabungan dengan data mereka lalu akan mengirimkannya kepada *node* tetangga yang berikutnya. Setiap *node* akan melakukan tugas tersebut secara bergantian dari *node* yang menjadi titik awal hingga data tersebut terkirim ke BS (Lindsey and Raghavendra 2002).

Seperti terlihat pada Gambar 2.7, alur pengiriman data pada Protokol PEGASIS menggunakan token yang berisi sebuah pesan. *Node* yang menjadi *leader* pada *chain* akan mengirimkan token tersebut ke *node* lainnya secara menyeluruh. Setelah token tadi sampai ke *node* yang paling jauh dari BS, *node* yang paling ujung tadi akan kembali mengirimkan data *sensing* ke *node* tetangganya. Seperti yang terlihat pada Gambar 2.7, *node* d akan mengirimkan data *sensing* ke *node* c lalu *node* c akan menggabungkan data tersebut dengan dirinya lalu mengirimkan data

ke *node leader*. Ringkasnya, PEGASIS protokol membuat *chain* dan setiap *node* akan mengumpulkan dan mengirimkan datanya ke *node* tetangganya. Dikarenakan pada PEGASIS, *leader* yang dipilih berurutan, protokol ini akan memiliki kemampuan untuk penyimpanan energi yang lebih baik daripada LEACH (Jung, Han, and Chung 2007).

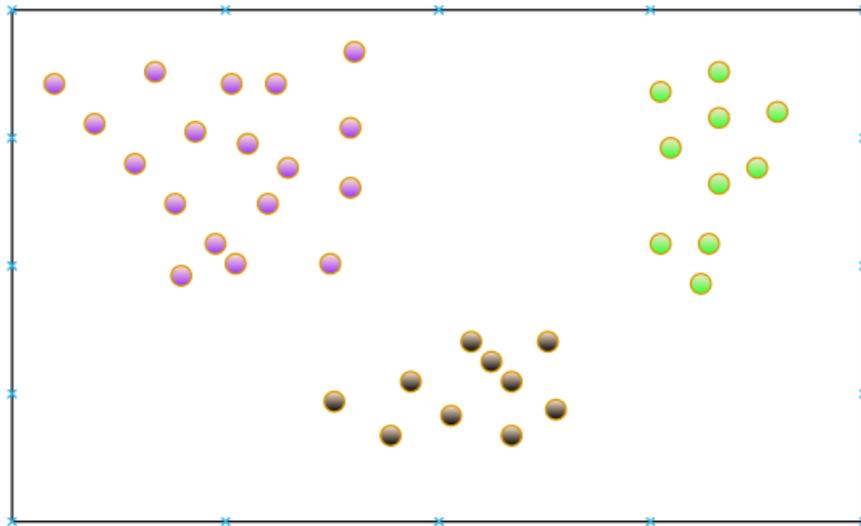


Gambar 2. 7 Alur Pengiriman Data PEGASIS Protokol

## 2.5 Clustering

*Clustering* merupakan sebuah proses untuk mengidentifikasi suatu kelompok dalam suatu data berdasarkan beberapa parameter tertentu untuk membaginya menjadi beberapa kelas atau berdasarkan kesamaan dari objek lainnya (Berkhin 2006). *Clustering* juga dapat dilihat sebagai sebuah teknik permodelan data untuk menyediakan ringkasan suatu data. *Clustering* juga merupakan suatu proses yang sangat penting untuk *pattern recognition* dan *machine learning* (Melton et al. 1999). Aplikasi dari *clustering* juga biasanya dipakai untuk suatu *dataset* dengan atribut yang banyak. Aplikasi dari *clustering* sendiri dipakai dalam banyak bidang seperti segmentasi gambar, *data mining*, kompresi data, *machine learning*, dan lain-lain (Omran, Engelbrecht, and Salman 2007). Ada dua jenis teknik *clustering* yang sering digunakan saat ini yaitu *hierachical* dan *partitional clustering*. *Hierarchical clustering* adalah sebuah metode yang menghasilkan sebuah *cluster tree (dendrogram)* dengan menggunakan teknik pemisahan ataupun penggabungan. Beberapa keunggulan dari teknik ini yaitu berlaku untuk semua

atribut, lebih mudah dalam menangani *similarity*, fleksibel terhadap *granularity*, serta tidak tergantung pada kondisi awal (Berkhin 2006), sedangkan pada metode *partitional clustering* setiap *cluster* akan mempunyai suatu titik pusat atau nilai *centroid* serta secara umum metode ini berfokus pada meminimumkan jarak dari pusat atau *centroid* ke seluruh data (Han, Kamber, and Pei 2014). Adapun contoh *clustering* dapat dilihat pada Gambar 2.8



Gambar 2. 8 *Clustering* (Ridwan, Ferdian, and Rahmadi Kurnia 2021)

### 2.2.3 Algoritma K-Means

Algoritma K-Means merupakan algoritma *clustering* yang menggunakan metode *partitional*. Algoritma K-Means *clustering* membagi kumpulan data menjadi sekelompok data dan mengklasifikasikan data tersebut menjadi beberapa *cluster* (Arora and Varshney 2016). Algoritma K-Means terdiri dari 2 fase yang berbeda, pada fase pertama, algoritma ini menghitung nilai dari  $k$  *centroid* (pusat *cluster*) dan pada fase kedua yaitu mengambil data dari setiap titik ke *cluster* yang paling dekat ke titik *centroid* nya. Dalam mencari titik *centroid*, ada beberapa cara untuk menghitung jarak terdekat ke titik tersebut dan salah satu cara yang paling banyak dipakai adalah dengan menggunakan *euclidean distance*. Setelah pengelompokan data tadi selesai dilakukan, algoritma ini akan menghitung ulang nilai dari *centroid* baru dari setiap *cluster* yang sudah ada dan berdasarkan dari nilai *centroid* tersebut, jarak baru antara nilai *centroid* dan nilai dari setiap data tadi akan dihitung dengan *euclidean distance* lalu membentuk sebuah *cluster* baru yang

memiliki jarak minimum dengan *centroid* yang baru dibentuk. Setiap bagian *cluster* sendiri ditentukan oleh objek anggotanya dan nilai dari *centroid*. Nilai *centroid* sendiri ditentukan berdasarkan jumlah jarak minimal dari seluruh objek yang berada pada *cluster* tersebut. Jadi algoritma K-Means sendiri merupakan sebuah algoritma iteratif dimana algoritma ini meminimalkan jarak dari setiap objek data ke *centroid cluster* (Wang et al. 2018)

Contoh dari algoritma ini misalkan ada sebuah gambar dengan ukuran resolusi  $x*y$  dan gambar tersebut harus dibagi berdasarkan *cluster*. Misal  $p(x*y)$  menjadi masukkan pixel untuk menjadi *cluster* dan  $c_k$  menjadi pusat *cluster*. Seperti terlihat pada Gambar 2.9a, 2.9b, dan 2.9c, bagaimana algoritma K-means membentuk dataset. Detail langkah dalam melakukan *cluster* K-means adalah sebagai berikut (Felipe 2016):

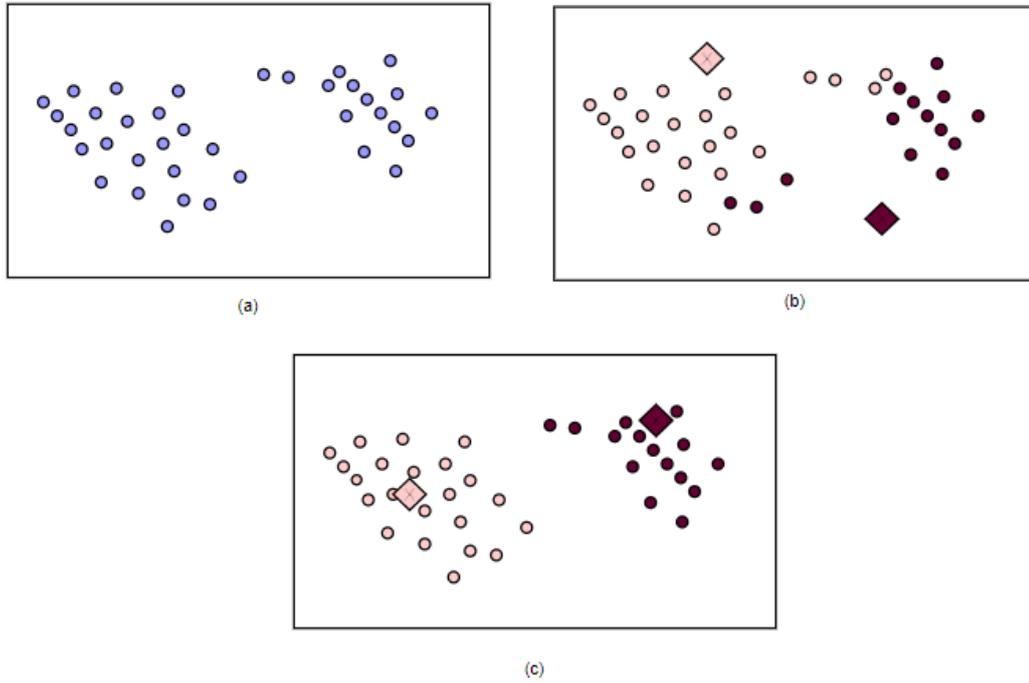
1. Inisialisasi jumlah dari *cluster*  $k$  dan pusatnya.
2. Untuk setiap pixel gambar, hitung jarak *euclidean distance*  $d$ , antara *center* dan setiap pixel gambar dihitung dengan menggunakan persamaan 2 berikut:

$$d = \|p(x, y) - c_k\| \quad (2)$$

3. Tetapkan semua piksel ke pusat terdekat berdasarkan jarak  $d$
4. Setelah semua pixel ditetapkan *cluster*, hitung posisi baru dari *center* menggunakan persamaan 2 berikut:

$$c_k = \frac{1}{k} \sum_{y \in c_k} \sum_{x \in c_k} p(x, y) \quad (1)$$

5. Ulangi proses ini hingga memenuhi batas yang diinginkan dari *error value*.
6. Bentuk kembali piksel *cluster* tersebut menjadi gambar



Gambar 2. 9 Algoritma K-Means. (a)Dataset Original. (b)*Random cluster centroid*. (c)Iterasi K-means dengan *centroid* terbaik

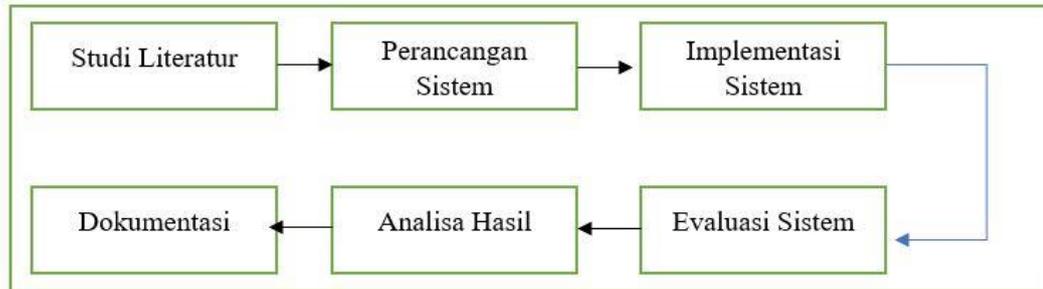
## **BAB 3**

### **METODE PENELITIAN**

#### **3.1 Metodologi Penelitian**

Metodologi penelitian ini dirancang untuk mengetahui langkah-langkah dalam menyelesaikan penelitian ini sehingga didapatkan sebuah tujuan akhir yang diinginkan. Tahap awal dalam penyelesaian penelitian ini sendiri adalah dengan mengumpulkan informasi dari berbagai literatur yang ada. Studi literatur ini dilakukan untuk mengetahui perkembangan dari penelitian sebelumnya tentang PEGASIS. Sumber literatur sendiri adalah buku, artikel ilmiah, jurnal, dan prosiding konferensi yang berkaitan dengan algoritma PEGASIS pada lingkungan WSN. Dari berbagai studi literatur yang telah dirangkum tersebut didapat informasi tentang beberapa masalah yang menjadi motivasi untuk melakukan penelitian tentang penyelesaian masalah tersebut. Tahap selanjutnya berupa perancangan sistem yang akan digunakan pada penelitian ini. Perancangan ini dikerjakan untuk penyelesaian masalah yang ada dengan kontribusi penelitian yang belum pernah dikerjakan sebelumnya oleh penelitian lainnya. Perancangan sistem ini sendiri berisi rumusan dan juga perancangan dari *flowchart* sistem. Tahapan selanjutnya dari penelitian ini adalah implementasi dari sistem yang sudah dirancang. Implementasi sendiri dilakukan dengan cara memasukkan langkah yang sudah dirancang dalam tahapan sebelumnya. Pada tahap ini, pengimplementasian rancangan tadi akan dilakukan pada *software* MATLAB. Setelah implementasi diselesaikan, maka akan dilakukan sebuah evaluasi dari sistem yang telah dibuat. Hasil simulasi akan dianalisa dengan beberapa parameter seperti *network lifetime*, *consumption energy*, dan *avarage energy* yang dikeluarkan oleh *node* dari sistem. Tahap berikutnya setelah hasil evaluasi didapat adalah dilakukan analisis terhadap hasil yang didapatkan. Analisis ini dilakukan agar diketahui kekurangan serta kelebihan dari sistem yang telah dirancang lalu dihasilkan sebuah kesimpulan terhadap hasil program yang dirancang. Tahapan terakhir dari penelitian ini adalah proses dokumentasi dari hasil penelitian yang telah dilakukan. Dokumentasi ini

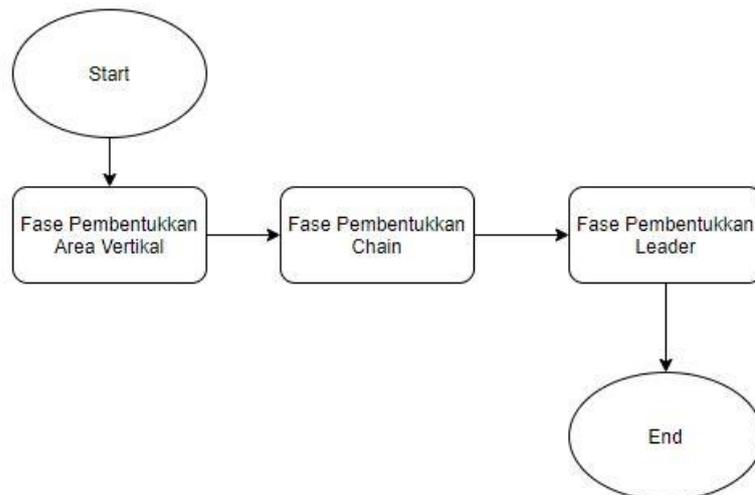
berisi tentang semua hasil dari studi literatur, perancangan, implementasi, evaluasi, dan analisa yang akan dibuat dalam bentuk buku tesis. Adapun keseluruhan alur tahapan penelitian dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Alur Tahapan Penelitian

### 3.2 Tahapan VCLUS-PEGASIS

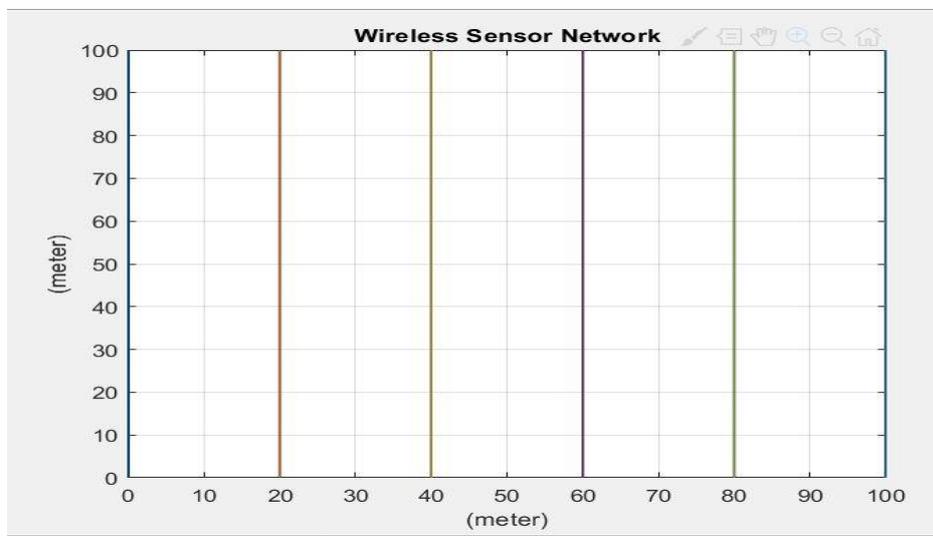
Alur tahapan perancangan strategi untuk melakukan modifikasi pada PEGASIS dapat dilihat pada Gambar 3.2. Secara keseluruhan, tahapan dalam pembentukan VCLUS-PEGASIS sendiri dibagi menjadi 3 fase dimana modifikasi sendiri terjadi pada tahap pertama dan tahap kedua. Adapun fase VCLUS-PEGASIS yaitu: (1) fase pembagian area dalam bentuk vertikal, (2) fase pembentukan *chain*, (3) fase pembentukan *leader*. Dalam setiap fase yang dilakukan akan dibahas dengan detail pada sub-bab sebagai berikut:



Gambar 3. 2 Alur Perancangan VCLUS-PEGASIS

### 3.2.1 Fase Pembagian Area Vertikal

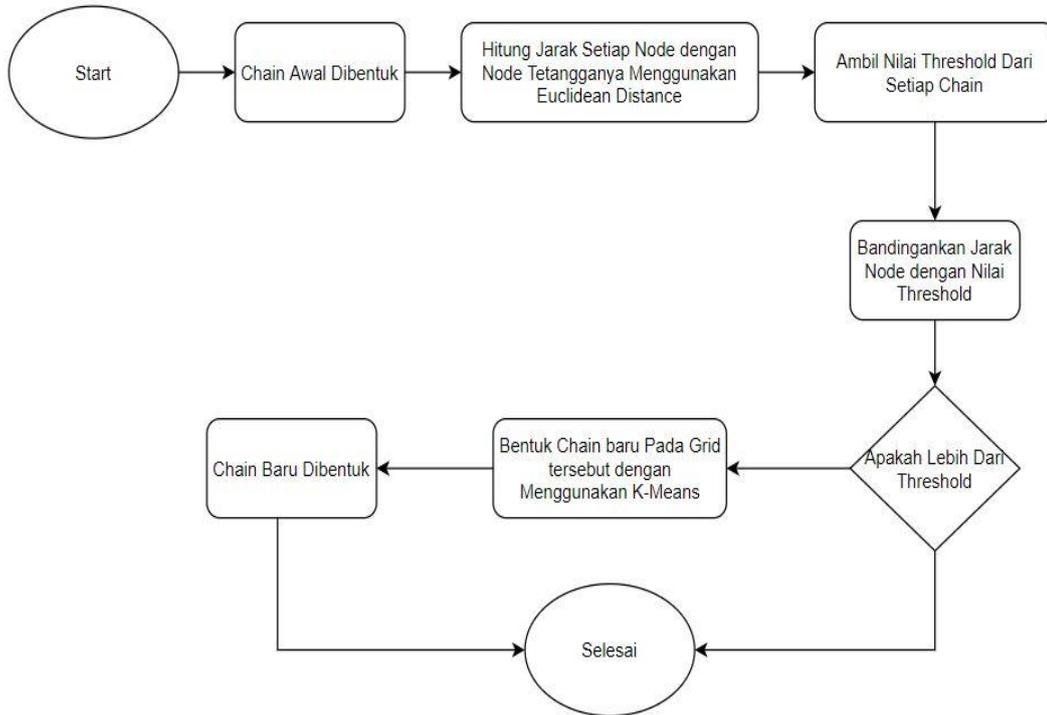
Tujuan utama pada fase ini adalahh membagi area *sensing* menjadi beberapa area yang lebih kecil dengan tujuan untuk memperkecil energi pada saat melakukan transmisi dengan cara membagi area itu menjadi sama rata berdasarkan koordinat dari lebar *node* nya (y). Untuk area  $(A \cdot A) \text{ m}^2$ , ukuran area dalam satu *grid* akan  $(A \cdot A/k) \text{ m}^2$  dimana k sendiri adalah jumlah dari *grid* yang diinginkan. Dalam setiap masing-masing area sendiri akan dibentuk *chain*. Adapun pembagian area ini dapat dilihat pada Gambar 3.3 berikut:



Gambar 3. 3 Area Vertikal

### 3.2.2 Fase Pembentukan *Chain*

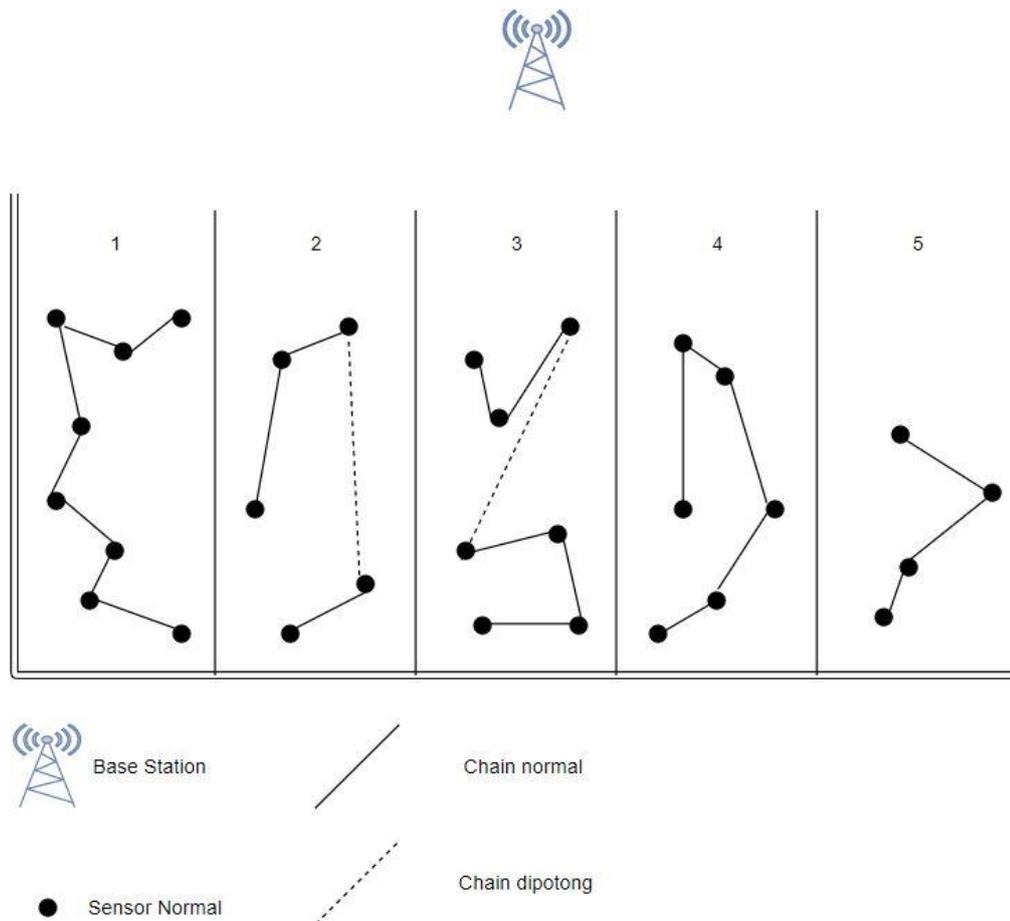
Dalam fase ini, *node* dalam setiap area akan saling menghubungkan untuk membentuk sebuah *chain*. Pembentukan *chain* ini memiliki konsep yang sama dengan protokol PEGASIS (Lindsey and Raghavendra 2002). Modifikasi pada tahap ini dapat dilihat pada *flowchart* di Gambar 3.4 berikut:



Gambar 3. 4 Alur Fase Pemebeentukan *Chain* pada VCLUS-PEGASIS

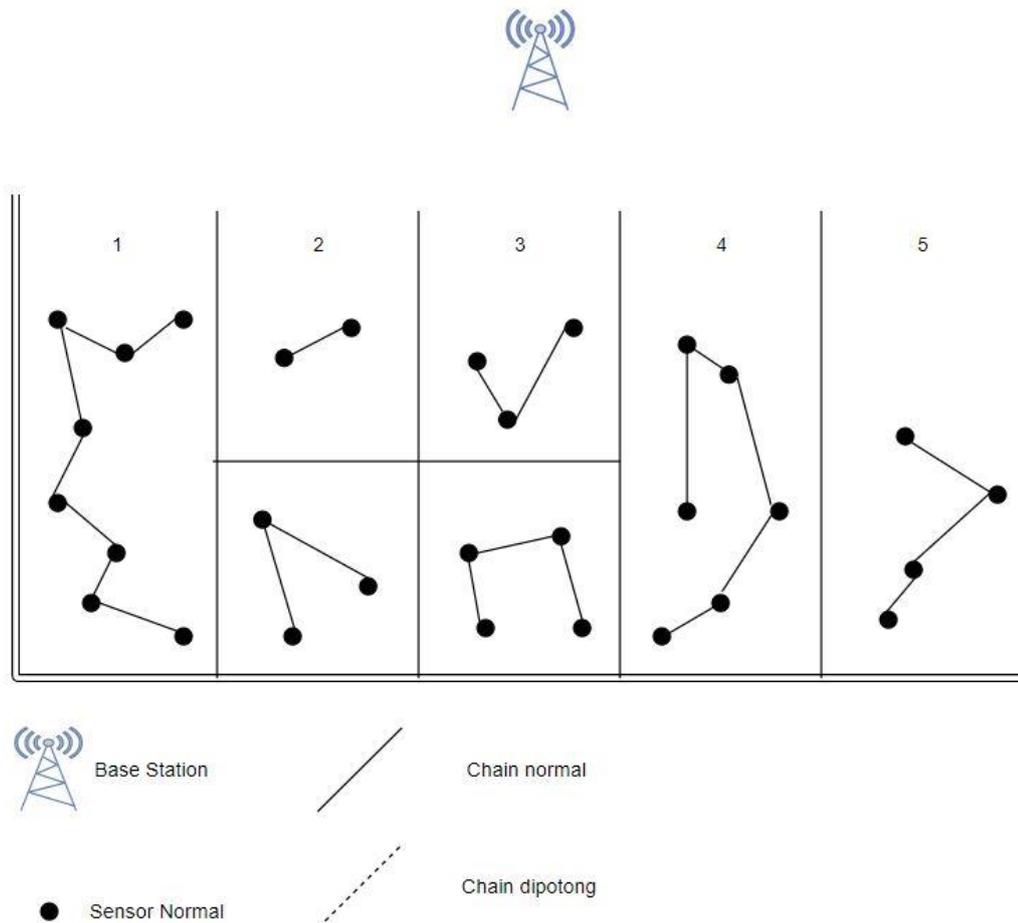
Pada penelitian ini, setelah *chain* awal terbentuk, seluruh jarak antara satu *node* dengan *node* tetangganya pada satu *grid* tadi akan dihitung. Untuk menghitung jarak antara setiap *node* dengan *node* tetangganya digunakan rumus *Euclidean Distance* dengan menggunakan persamaan (3). Lalu setelah didapat jarak antar *node* dengan *node* tetangganya tadi, maka jarak tadi akan ditotalkan lalu dihitung nilai rata-ratanya. Nilai rata-rata ini nantinya akan dikalikan tiga. Nilai yang didapat tersebut akan digunakan sebagai nilai batas (*threshold*) sebagai jarak maksimal suatu *node*. Apabila *node* pada suatu *grid* tadi memiliki jarak yang melebihi nilai *threshold*, maka akan dilakukan pembuatan *chain* baru pada *grid* tersebut. Adapun pembuatan *chain* baru pada setiap *node* dapat dilihat pada Gambar 3.5.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3)$$



Gambar 3. 5 Skenario Pemotongan *Node Area Vertikal*

Pada Gambar 3.5 terlihat pada *grid 2* dan *grid 3* jika ada sebuah *node* yang memiliki jarak yang melebihi nilai *threshold* maka pada masing-masing *grid* tersebut akan dilakukan pembuatan *chain* baru dengan menggunakan algoritma K-Means. Cara pembentukan *chain* yang dilakukakn dengan menggunakan algoritma K-mean, pembentukan ini dimulai dengan kelompok pertama yang dibentuk berdasarkan *centroid* yang dipilih secara acak yang digunakan sebagai titik awal setiap *cluster*, lalu dilakukan penghitungan ulang hingga menemukan nilai optimal posisi *centroid*. Setelah *centroid* telah stabil, *chain* baru akan terbentuk dalam satu *grid*. Adapun contoh skenario pembentukkan *chain* baru tersebut terlihat pada Gambar 3.6 berikut:

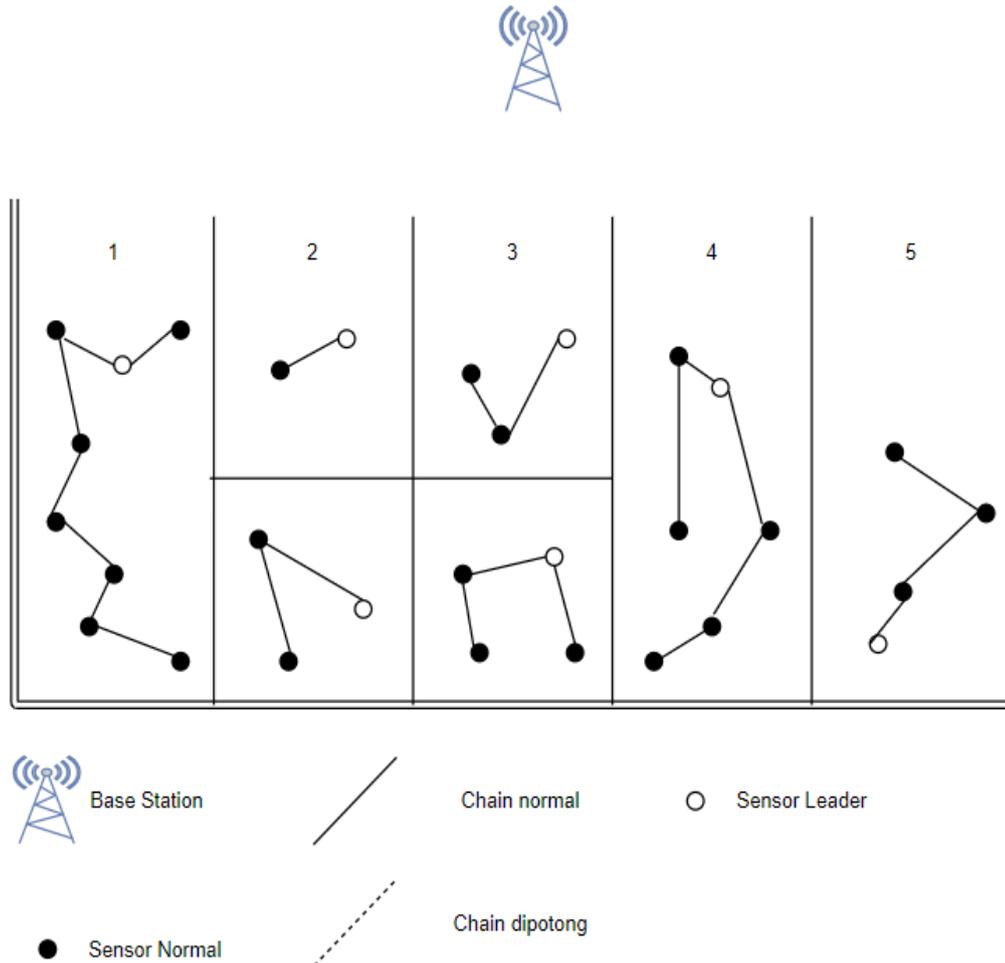


Gambar 3. 6 Skenario Pembentukan *Chain* Baru Area Vertikal

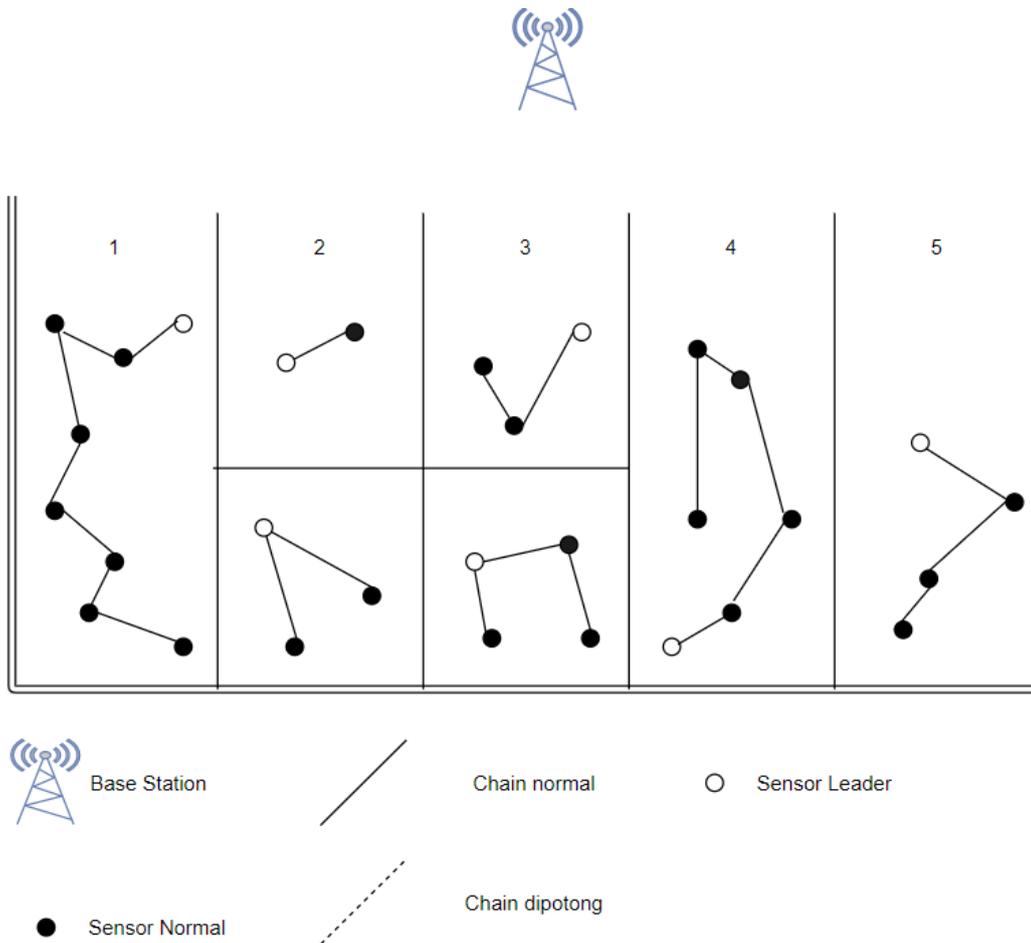
### 3.2.3 Fase Pembentukan *Leader*

Pembentukan *leader* di dalam penelitian ini mirip dengan pemilihan *leader* pada PEGASIS protokol (Lindsey and Raghavendra 2002). *Leader* sendiri akan dibentuk pada setiap area yang ada. *Leader* pada setiap area sendiri akan dipilih secara acak dan akan berganti pada setiap putaran. *Leader* sendiri memiliki fungsi sebagai *node* yang mengumpulkan semua data lalu mengirimkannya ke BS. Ketika *leader node* dipilih dan memberitahukan ke BS jika *node* tersebut adalah *leader*, setiap *node* pada *chain* akan melakukan transmisi data dan menerima gabungan data

hingga ke *leader*. Adapun skema dari fase ini dapat dilihat pada Gambar 3.7 dan Gambar 3.8



Gambar 3.7 Fase Pembentukan *Leader Round 1*



Gambar 3. 8 Fase Pembentukan *Leader Round 2*

Pada Gambar 3.7 dan Gambar 3.8 terlihat jika sensor *leader* pada *round 1* dan *round 2* berbeda. *Leader* pada setiap *grid* sendiri akan dipilih secara acak dan akan berganti pada setiap putaran. Pengacakan pada *leader* sendiri berfungsi agar energi untuk melakukan transmisi data menjadi rata pada setiap *node* dikarenakan *leader* pada PEGASIS membutuhkan lebih banyak energi pada saat melakukan transmisi data (Marhoon, Mahmuddin, and Nor 2016). Apabila dalam satu area terjadi pembentukan *chain* baru maka *leader* juga akan bertambah secara otomatis pada *chain* yang dibentuk.

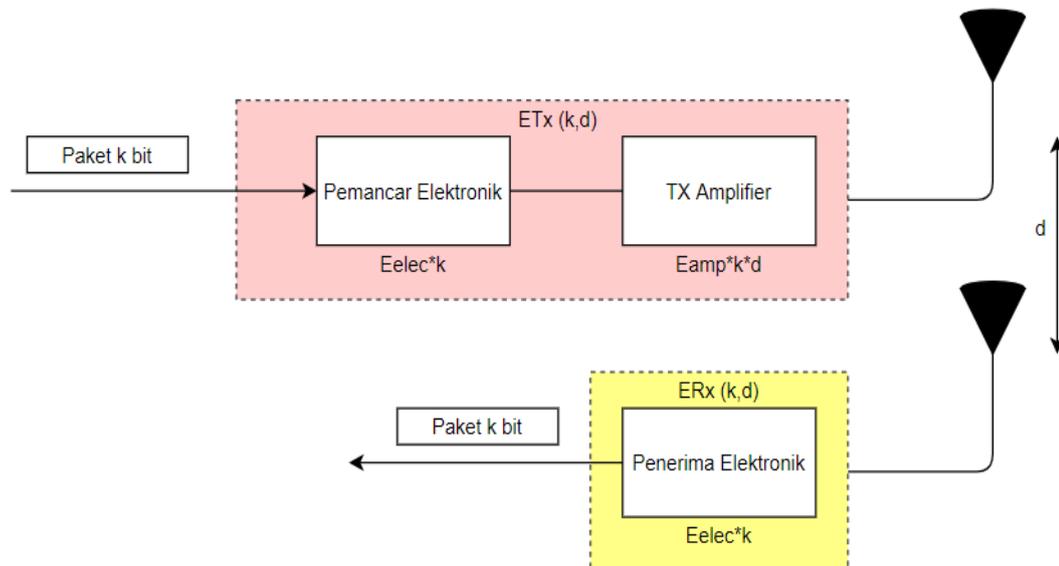
### 3.3 Model Jaringan

Dalam penelitian ini digunakan beberapa asumsi dalam model jaringan WSN. Adapun asumsi model jaringan WSN yang digunakan adalah sebagai berikut:

- N sensor *node* disebar secara acak dalam bidang persegi panjang didalam beberapa area dalam kondisi semua sensor *node* statis dengan jumlah energi yang sama pada saat pertama disebar dan spesifikasi perangkat yang sama. Baterai sensor *node* sendiri tidak dapat diisi ulang hingga semua sensor *node* mati.
- Setiap sensor *node* mampu mengetahui koordinat posisi sensor *node* tetangganya namun pada saat pencarian tetangga terdekatnya akan menghabiskan energi lebih banyak.
- Sink bersifat statis.

### 3.4 Radio dan Model Energi

Radio dan model energi yang digunakan pada penelitian ini pada setiap transmisinya akan dihitung menggunakan persamaan yang sama dengan Heinzelman dkk (Heinzelman, Chandrakasan, and Balakrishnan 2000). Model radio ini digunakan dalam banyak penelitian (Marhoon, Mahmuddin, and Nor 2016; Ramluckun and Bassoo 2018). Dapat terlihat pada Gambar 3.7 dapat dilihat model radio yang digunakan dalam perhitungan energi yang digunakan.



Gambar 3. 9 Radio Model (Anjum and Ahmad 2017)

Dalam penggunaan energi, energi yang paling digunakan terletak pada saat transmisi dan penerimaan data. Pada saat melakukan transmisi, ada dua buah cara

dalam perhitungan disipasi energi didalam jaringan yaitu dengan model *free space* dan model *multipath*. Langkah dalam menghitung energi yang dibutuhkan suatu Paket k-bit *packet data unit* untuk transmisi digambarkan dalam persamaan (4) berikut:

$$E_{TX} = E_{TX-elec}^{(k)} + E_{TX-amp}^{(k,d)} \quad (4)$$

$$= \begin{cases} k * E_{elec} + k * \varepsilon_{fs} * d^2 & \text{if } d < d_0 \\ k * E_{elec} + k * \varepsilon_{mp} * d^4 & \text{if } d \geq d_0 \end{cases} \quad \text{where } d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}$$

Dimana  $E_{TX}$  adalah energi yang diperlukan ketika proses transmisi dilakukan,  $E_{elec}$  merupakan energi yang dibutuhkan untuk melakukan proses sebelum *packet data unit* ditransmisikan,  $d$  adalah jarak antara perangkat transmisi dan penerima,  $d_0$  merupakan nilai *threshold* untuk menentukan apakah akan memakai model *free space* atau *multipath*,  $\varepsilon_{fs}$  merupakan disipasi energi per bit yang digunakan *amplifier* pada mode *free space* dan  $\varepsilon_{mp}$  yang digunakan *amplifier* pada model *multipath*.

Pada perangkat penerima data sendiri, energi yang digunakan saat menerima data sebesar k-bit *packet data unit*, maka energi yang dibutuhkan terlihat pada persamaan (5) berikut:

$$E_{rx}^{(k)} = E_{rx-elec}^{(k)} = k * E_{elec} \quad (5)$$

Dimana  $E_{rx}^{(k)}$  energi yang dibutuhkan oleh penerima untuk menerima sejumlah bit *packet data unit* setiap data k-bit. Terakhir pada saat perangkat menerima, akan ada penggabungan data pada data di setiap *node* pada *chain*. Adapun energi yang dikeluarkan pada saat penggabungan data ini dapat dilihat pada persamaan (6) berikut:

$$E_{DA-tot} = s * k * E_{da} \quad (6)$$

Dimana  $E_{da}$  adalah energi per bit yang dikeluarkan pada saat melakukan penggabungan data dan  $s$  adalah jumlah dari sinyal yang digabungkan.

### 3.5 Uji Coba dan Evaluasi

Uji coba dilakukan untuk memastikan sistem yang telah dirancang dan dibangun berfungsi sesuai dengan tujuan. Uji coba sendiri dilakukan dengan

menggunakan bahasa pemrograman MATLAB seperti yang dilakukan pada beberapa penelitian seperti pada penelitian Elshrkawey dkk dan Kamimura dkk (Elshrkawey, Elsherif, and Wahed 2017; Kamimura, Wakamiya, and Murata 2006; Su and Zhao 2017). Untuk mengetahui kinerja dari protokol yang diajukan dalam beberapa parameter ini, beberapa parameter yang bersifat tetap diperlukan untuk melakukan uji coba sesuai dengan beberapa penelitian yang terkait dengan perbaikan PEGASIS seperti penelitian Ramluckun dan Basso, Young-Long Chen dan Jia-Sheng Lin, Heinzeman dkk (Y. Chen and Lin 2012; Heinzelman, Chandrakasan, and Balakrishnan 2000; Ramluckun and Basso 2018). Parameter yang digunakan dalam lingkungan *sensor network* nya menggunakan 100 *node* yang ditempatkan secara acak dalam area 100\*100. Parameter pada uji coba sendiri menggunakan parameter uji coba yang sama dengan Lindsey dan Raghavendra (Lindsey and Raghavendra 2002) dan dapat dilihat pada Tabel 3.1.

Evaluasi diperlukan demikian untuk mengetahui seberapa bagus metode dan strategi yang diusulkan melalui hasil uji coba. Parameter evaluasi yang digunakan didalam ujicoba adalah menggunakan *alive node in network* yang akan dibagi menjadi 3 bagian yaitu FND (*First Node Die*), HND (*Half Node Die*), dan LND (*Last Node Die*) serta *average consumption energy per transmission*. Detail dari parameter sendiri dapat dilihat pada Tabel 3.2. Sebagai pembandingan hasil dari uji coba dalam penelitian ini, akan dilakukan perbandingan dengan penelitian sebelumnya yaitu penelitian tentang PEGASIS asli oleh Lindsey dan Raghavendra (Lindsey and Raghavendra 2002).

*Tabel 3. 1 Parameter Uji Coba*

| <b>Parameter</b>                         | <b>Kapasitas</b>             |
|--|------------------------------|
| Jumlah node                              | 100 dan 200                  |
| Area Simulasi                            | 100*100                      |
| Node Initial Energy ( $E_{init}$ )       | 0,5 J                        |
| Electronic circuit energy ( $E_{elec}$ ) | 50 nJ/bit                    |
| Energy of amplifier ( $E_{amp}$ )        | 0,0013 pJ/bit/m <sup>4</sup> |
| Sink                                     | 50*100                       |

Tabel 3. 2 Parameter Evaluasi

| No | Parameter                                   | Penjelasan  |
|----|---|---|
| 1  | Alive Node in Network                       | Jumlah total <i>round</i> sensor <i>node</i> pertama kali disebarkan hingga sensor <i>node</i> mati                                   |
| 2  | First Node Die                              | Rentang waktu sensor <i>node</i> pertama kali disebarkan sampai <i>node</i> pertama kali mati   |
| 3  | Half Node Die                               | Rentang waktu sensor <i>node</i> pertama kali disebarkan sampai <i>node</i> mati berjumlah 50%  |
| 4  | Last Node Die                               | Rentang waktu sensor <i>node</i> pertama kali disebarkan sampai <i>node</i> terakhir mati   |
| 5  | Average consumption Energy per Transmission | Rata-rata energi transmisi setiap <i>round</i> dari sensor <i>node</i> pertama kali disebarkan hingga sensor <i>node</i> pertama mati |

### 3.6 Dokumentasi dan Jadwal Penelitian

Proses dokumentasi digunakan dalam penulisan laporan hasil penelitian yang dilakukan. Setiap tahapan dalam proses penelitian juga disertakan dalam laporan yang ditulis. Kegiatan dokumentasi ini akan dicantumkan pada jadwal penelitian yang direncanakan mulai bulan Februari 2020 hingga Juni 2020 yang secara rinci terlihat pada Tabel 3.3.

*Tabel 3. 3 Jadwal Kegiatan Penelitian*

| No. | Kegiatan                | Bulan    |   |   |   |       |   |   |   |       |   |   |   |     |   |   |   |      |   |   |  |
|-----|-------------------------|----------|---|---|---|-------|---|---|---|-------|---|---|---|-----|---|---|---|------|---|---|--|
|     |                         | Feburari |   |   |   | Maret |   |   |   | April |   |   |   | Mei |   |   |   | Juni |   |   |  |
| 1.  | Studi Literatur         | ■        | ■ | ■ | ■ |       |   |   |   |       |   |   |   |     |   |   |   |      |   |   |  |
| 2.  | Analisa dan Perancangan | ■        | ■ | ■ | ■ | ■     | ■ | ■ |   |       |   |   |   |     |   |   |   |      |   |   |  |
| 3.  | Ujian Proposal          |          |   |   |   |       |   | ■ |   |       |   |   |   |     |   |   |   |      |   |   |  |
| 4.  | Implementasi            |          |   |   |   |       |   | ■ | ■ | ■     | ■ | ■ | ■ | ■   | ■ |   |   |      |   |   |  |
| 5.  | Pengujian dan Evaluasi  |          |   |   |   |       |   |   |   | ■     | ■ | ■ | ■ | ■   | ■ | ■ |   |      |   |   |  |
| 6.  | Penyusunan Buku Tesis   |          |   |   |   |       |   |   |   |       |   |   |   | ■   | ■ | ■ | ■ | ■    | ■ | ■ |  |
| 7.  | Ujian Tesis             |          |   |   |   |       |   |   |   |       |   |   |   |     |   |   |   |      |   | ■ |  |

*[halaman ini sengaja dikosongkan]*

## BAB 4

### HASIL DAN PEMBAHASAN

#### 4.1. Hasil

Pada sub-bab ini akan dipaparkan tentang ujicoba yang akan dibahas dengan metode yang dijabarkan agar didapatkan hasil evaluasi sesuai dengan yang diharapkan. Selain itu juga dijabarkan tentang penggunaan perangkat keras dan perangkat lunak yang digunakan sebagai bahan uji coba untuk melakukan penelitian yang dilakukan.

##### 4.1.1 Tahap Penelitian

Pada sub-bab ini akan dijelaskan tentang bagaimana tahapan yang dilakukan untuk mendapatkan hasil dari metode yang dilakukan. Adapun tahapan penitilian yang dilakukan adalah sebagai berikut:

- a. Pada tahap pertama yaitu melakukan perancangan pada protokol VCLUS-PEGASIS dalam lingkungan WSN dimulai dengan membagi area menjadi beberapa *grid* dengan bentuk vertikal yang dibagi dalam area *grid* yang sama. Kemudian setelah *grid* dibentuk, *node* akan disebar secara acak dan setiap area *grid* akan terbentuk *chain*. Setelah *chain* awal terbentuk, jarak pada setiap *node* dengan *node* tetangganya akan dilakukan penghitungan menggunakan *euclidean distance*. Setelah jarak dari setiap *node* dengan *node* tetangganya didapatkan kemudian dicari nilai rata-rata dari perhitungan jarak antar *node* dengan *node* tetangganya tersebut. Setelah rata-rata didapatkan, maka nilai rata-rata tersebut akan dikalikan tiga dan kemudian nilai ini dijadikan sebagai nilai ambang batas (*threshold*). Apabila jarak dari *node* dalam suatu *chain* di dalam *grid* tersebut melebihi nilai *threshold*, maka *chain* didalam *grid* tersebut akan dibuat ulang. Adapun pembuatan *chain* tersebut akan dibagi menjadi dua bagian dengan menggunakan algoritma K-Means. Setelah area dibagi kemudian akan *chain* akan dibentuk ulang. Pembagian ini berguna agar masa hidup dari *node* dapat lebih panjang.

- b. Pada tahap kedua akan dilakukan perancangan skenario untuk mengetahui kinerja dari protokol usulan dari VCLUS-PEGASIS yang dijelaskan pada tahap pertama. Adapun skenario uji coba ini dilakukan dengan mengubah variasi parameter jumlah *node*, luas dari area uji coba, serta perbandingan dengan metode lainnya.
- c. Pada tahap ketiga ini akan dilakukan analisa akhir yang diperoleh dari tahapannya yang telah dilakukan sebelumnya. Tahapan ini berguna untuk melihat hasil dari VCLUS-PEGASIS yang sudah dilakukan dengan skenario yang telah dirancang

#### 4.1.2 Lingkungan Uji Coba

Spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk mengimplementasikan penelitian ini dapat dilihat pada Tabel 4.1. Lingkungan uji coba perangkat keras digunakan sebagai suatu fasilitas agar dapat mengimplementasikan strategi yang telah dibahas pada metode penelitian sebelumnya. Rancangan strategi yang sudah dilakukan kemudian akan diubah dalam bentuk baris kode pada lingkungan perangkat lunak.

*Tabel 4. 1 Lingkungan Uji Coba*

| Komponen       | Spesifikasi         |
|----------------|---------------------|
| CPU            | Intel Core i7-8500U |
| Memori         | 8 GB PC3-12800 DDR3 |
| VGA            | 6 GB DDR4           |
| Penyimpanan    | 1 TB                |
| Sistem Operasi | Windows 10          |
| <i>Tools</i>   | Matlab Versi R2018  |

#### 4.1.3 Tahapan Uji Coba

Uji coba ini berguna untuk mengetahui kinerja hasil VCLUS-PEGASIS dengan teknik pembagian area dan pembentukan *chain* secara adaptif dengan metode lain sesuai dengan skenario yang telah dijelaskan pada bab 3 sebelumnya. Adapun skenario pengujian dilakukan dengan melakukan perubahan pada jumlah *node* dan juga luas area tiap skenario dijalankan. Dalam melakukan skenario uji

coba sendiri digunakanlah Matlab sebagai *tools* karena tidak memungkinkannya untuk melakukan uji coba real karena terbentur nya area untuk melakukan uji coba. Adapun parameter pengujian setiap skenario dapat dilihat pada Tabel 4.2

*Tabel 4. 2 Parameter Pengujian Skenario*

| Parameter                                | Kapasitas                        |
|--|----------------------------------|
| Area Simulasi                            | 100*100                          |
| Jumlah Grid                              | 5 dan 7                          |
| Jumlah node                              | 100 node, 200 node, dan 300 node |
| Node Initial Energy ( $E_{init}$ )       | 2 J                              |
| Electronic circuit energy ( $E_{elec}$ ) | 50 nJ/bit                        |
| Energy of amplifier ( $E_{amp}$ )        | 0,0012 pJ/bit/m <sup>4</sup>     |
| Base Station (BS)                        | 50*50                            |

#### 4.2. Modifikasi Pada Protokol PEGASIS

Dalam melakukan modifikasi pada protokol PEGASIS dibagi menjadi 3 tahap yaitu fase pembagian area vertikal, fase pembentukan *chain*, dan fase pembentukan *leader*. Dalam setiap fase yang dimodifikasi akan dibahas dengan detail pada sub-bab sebagai berikut:

##### 4.2.1 Fase Pembagian Area Vertikal

Modifikasi pada fase ini terjadi pada saat melakukan pembagian area *sensing* seperti yang dijelaskan pada sub bab 3.2.1. Dalam penelitian ini area sendiri dibagi menjadi 2 skenario yaitu 5 *grid* dan 7 *grid*. Adapun fase pembagian area vertikal ini dijelaskan pada *pseudocode* dalam tabel 4.3 berikut:

*Tabel 4. 3 Pseudocode Fase Area Vertikal*

|   |  |
|---|--|
| 1 | Area = Deklarasi jumlah grid yang diinginkan |
| 2 | A = Luas area sensing (vertikal) / Area      |
| 3 | For i = 1: Area                              |
| 4 | $U(i) = 0 + ((i-1) * a)$                     |

|   |                                    |
|---|------------------------------------|
| 5 | End for                            |
| 6 | x = u                              |
| 7 | y = Luas Area Sensing (horizontal) |
| 8 | [x,y] = create meshgrid (x,y);     |

#### 4.2.2 Fase Pembentukan *Chain*

Modifikasi pada fase ini terjadi setelah *chain* pertama kali dibentuk pada setiap area. Setelah *node* disebar secara *random*, maka *chain* awal sendiri dibentuk menggunakan algoritma *greedy* (Lindsey and Raghavendra 2002). Setelah *chain* awal terbentuk, seluruh jarak antara satu *node* dengan *node* tetangganya pada satu *grid* tadi akan dihitung. Untuk menghitung jarak antara setiap *node* dengan *node* tetangganya digunakan rumus *Euclidean Distance*. Lalu setelah didapat jarak antar *node* dengan *node* tetangganya tadi, maka jarak tadi akan ditotalkan lalu dihitung nilai rata-ratanya. Nilai rata-rata ini nantinya akan dikalikan tiga. Nilai yang didapat tersebut akan digunakan sebagai nilai batas (*threshold*) sebagai jarak maksimal suatu *node*. Langkah awal dalam modifikasi ini dapat dilihat pada Tabel 4.4 berikut:

Tabel 4. 4 Pseudocode 1 Hitung Jarak dengan Tetangga, Nilai Rata-Rata, dan Total Jarak Antar Node

|   |   |
|---|---|
| 1 | totaljarak = []   |
| 2 | Rata-rata = []  |
| 3 | %Hitung jarak dengan tetangga<br>For z = 1:size (area,2)  |
| 4 | Total_global_jarak = 0  |
| 5 | For i = 1 : size (area[1,z],2)  |
| 6 | If Variabel (area [1,z](i)).Energy > 0 dan Variabel (area[1,z](i).sudah_beroperasi == 0) dan Variabel (area[1,z](i)).node hidup == 1) |
| 7 | Set = area[1,z](i);<br>Variabel(set).sudah_beroperasi == 1<br>Variabel(set)   |
| 8 | Break;  |

|    |   |
|----|---|
| 9  | End if  |
| 10 | Selected = 0;   |
| 11 | While selected < jumlah_node  |
| 12 | Min_distance = 9999;  |
| 13 | For i = i:jumlah_node   |
| 14 | If Variabel(i).sudah_beroperasi == 0  |
| 15 | If Variabel(i).label == z   |
| 16 | Jarak_antar_tetangga = sqrt((Variabel(i).x)^2 + (Variabel(i).y)^2)                                    |
| 17 | If jarak_antar_tetangga < min_distance  |
| 18 | Min_distance = d;   |
| 19 | Next = i;   |
| 20 | end if  |
| 21 | End if  |
| 22 | End if  |
| 23 | End for   |
| 24 | Selected = selected + 1;  |
| 25 | Variabel(set).closest = next;   |
| 26 | Variabel(set).dis = min_distance;   |
| 27 | Total_global = total_global + min_distance;   |
| 28 | Sort_temp = sort (order[:,z])   |
| 29 | Temp = 1;   |
| 30 | End while;  |
| 30 | For i= 1:temp;  |
| 31 | If Variabel(sort_temp(i)).closest == set)   |
| 32 | Variabel(set).dis2 = sqrt ((Variabel(i).x – Variabel(set).x)^2 + (Variabel(i).y – Variabel(set).y)^2) |
| 33 | End if  |
| 34 | End for   |
| 35 | totaljarak[z] = total_global;   |
| 36 | Rata-rata[z] = total_global/temp;   |
| 37 | Treshold[z] = Rata-rata *3;   |
| 38 | End for   |

Dalam penelitian ini, *threshold* didapatkan berdasarkan hasil eksperimen terbaik yang akan dibahas lebih lanjut pada subbab 4.3.1. Setelah mendapatkan nilai *threshold*, apabila *node* pada suatu *grid* tadi memiliki jarak yang melebihi nilai *threshold*, maka pada masing-masing *grid* yang memiliki jarak yang melebihi nilai *threshold* tersebut akan dilakukan pembuatan *chain* baru dengan menggunakan algoritma K-Means. Adapun *pseudocode* dalam modifikasi ini dapat dilihat pada Tabel 4.5 berikut:

*Tabel 4. 5 Pseudocode 2 Cek Threshold dan Pembentukan Chain dengan Algoritma Kmeans*

|     |   |
|-----|---|
| 1   | %cek potong atau tidak?<br>Apakah_potong = zeros(1, size (area, 2));  |
| 2   | For z = 1 : size (area,2)   |
| 3   | For i = 1:size (area[1,z],2)  |
| 3   | If Variabel(area(1,z)(i)).distance > threshold (z)  |
| 4   | Apakah_potong(z) == 1;  |
| 5   | Break   |
| 6   | End   |
| 7   | End   |
| 8   | End   |
| 9   | %cluster  |
| 10  | AreaSetelahPotong = [];   |
| 11  | Area_a = 1;   |
| 12. | For z = 1 : size (area,2)   |
| 13  | If apakah_potong(1,z) == 1  |
| 14  | dataKmean = [];   |
| 15  | For idataKmeans = 1:size (area[1,z],2)  |
| 16  | dataKmean(idataKmeans, :) = [Variabel(area[1,z](idataKmeans)).x ,<br>Variabel(area[1,z](idataKmeans)).y ] ; |
| 17  | End for   |
| 18  | [iData, centroid] = kmeans(dataKmean,2);  |
| 19  | Potonggrid = [];  |

|    |  |
|----|--|
| 20 | Pc1 = 1;                                     |
| 21 | Potonggrid2 = [];                            |
| 22 | Pc2 = 1;                                     |
| 23 | For idataKmeans = 1:size (area[1,z].2)       |
| 24 | If iData (idataKmeans) == 1                  |
| 25 | Potonggrid(Pc1) = area[1,z](idataKmeans);    |
| 26 | Pc1 = Pc1 + 1;                               |
| 27 | Else   |
| 28 | Potonggrid(Pc2) = area[1,z](idataKmeans);    |
| 29 | Pc2 = Pc2 + 1;                               |
| 30 | End if                                       |
| 31 | End for                                      |
| 32 | Area[1,Area_a] = potonggrid;                 |
| 33 | Area_a = Area_a + 1                          |
| 34 | Area[1,Area_a] = potonggrid2;                |
| 35 | Area_a = Area_a + 1                          |
| 36 | Else   |
| 37 | Area[1,Area_a] = area[1,z];                  |
| 38 | Area_a = Area_a + 1                          |
| 39 | End if                                       |
| 30 | %Grid<br>Potonggrid = Area_a - 1;            |
| 31 | For z = 1 : size (area,2)                    |
| 32 | For i = 1:size (area[1,z],2)                 |
| 33 | Variabel(area[1,z](i)).label = z;            |
| 34 | Variabel(area[1,z](i)).sudah_beroperasi = 0; |
| 35 | Execute plot;                                |
| 36 | End for                                      |
| 37 | End for                                      |

### 4.2.3 Fase Pembentukan *Leader*

Modifikasi pada fase ini terjadi pada saat semua *chain* dibentuk dan tidak ada lagi penambahan *chain* baru. *Leader* sendiri akan ada pada setiap area *chain*. Adapun fase pembentukan *leader* dijelaskan pada Tabel 4.6 berikut:

Tabel 4. 6 Pseudocode Pembentukan *Leader*

|    |  |
|----|--|
| 1  | For z = 1 : size (area,2)                                  |
| 2  | UrutD = size [1,z];  |
| 3  | UrutD = sort [UrutD];                                      |
| 4  | urutId [1,z] = UrutD;                                      |
| 5  | End for  |
| 6  | leadermati = [];   |
| 7  | Jumlahmati =1 ;  |
| 8  | %Leader Election<br>For z = 1 : size (area,2)              |
| 9  | A = size (area[1,z] , 2)                                   |
| 10 | Urutan = mod(turn, A)+1;                                   |
| 11 | %ambil leader per grid<br>Leader = urutId [1,z](urutan);   |
| 12 | %kondisi =1 node hidup<br>If Variabel(leader).kondisi == 0 |
| 13 | While Variabel(leader).kondisi == 0                        |
| 14 | If ismember(leader, leadermati) == 0                       |
| 15 | Leadermati(jumlahmati) = leader;                           |
| 16 | Jumlahmati = jumlahmati +1;                                |
| 17 | If jumlahmati == (n+1)                                     |
| 18 | Break  |
| 19 | End if   |
| 20 | End if   |
| 21 | Turn = turn + 1;   |
| 22 | Leader = mod(turn,A) + 1;                                  |
| 23 | End while  |

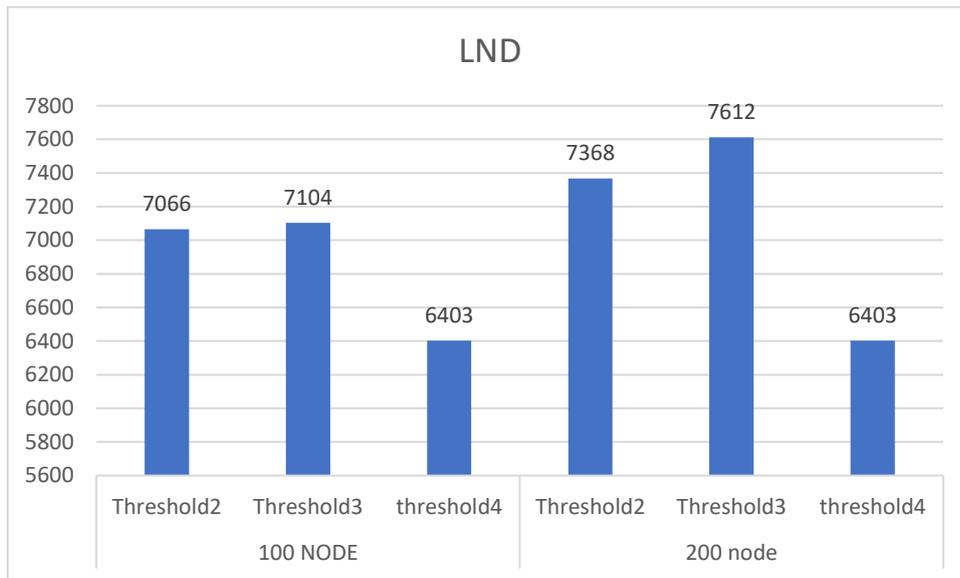
|    |   |
|----|---|
| 24 | End if  |
| 24 | If Variabel(leader).kondisi == 1  |
| 25 | %role =1 leader dan role = 0 bukan leader<br>Variabel(leader).role = 1; |
| 26 | Execute plot;   |
| 27 | End if  |

### 4.3. Analisis Hasil

Dalam sub-bab analisis hasil penelitian ini akan dipaparkan tentang hasil yang diperoleh dari beberapa uji coba skenario yang telah dilakukan berdasarkan pengujian yang dikerjakan. Pertama akan dicari nilai terbaik untuk menentukan nilai *threshold*. Kedua pengujian sendiri akan dilakukan dengan membandingkan teknik pembagian area dan pembentukan *chain* secara adaptif ini dengan menggunakan 5 *grid* dengan protokol PEGASIS aslinya.

#### 4.3.1 Penentuan Nilai Threshold

Pada penelitian ini dalam menentukan nilai *threshold* pada saat fase pembentukan *chain*, akan dilakukan penelitian berdasarkan eksperimen terbaik dari 3 parameter yaitu pada saat *threshold* bernilai 2,3, dan 4 kali nilai rata-rata jarak *node* dengan tetangganya. Dalam mencari nilai terbaik, parameter yang akan diukur adalah berapa lama *node* itu mati dalam satu kali uji coba atau dihitung berdasarkan nilai LND nya. *Node* yang disebar dalam penentuan nilai *threshold* adalah 100 dan 200 *node*. Adapun hasil uji coba tersebut dapat dilihat pada Gambar 4.1 berikut:



Gambar 4. 1 Uji coba *Threshold*

Terlihat pada Gambar 4.1 jika *threshold* menggunakan nilai rata-rata yang dikalikan 3 lebih baik dalam pengujian baik pada *node* dengan kerapatan 100 maupun 200 *node*. Selisih yang dihasilkan seperti terlihat pada Tabel 4.7 didapatkan jika *threshold* yang dikalikan 3x nilai rata-rata menghasilkan angka 0,54% lebih baik dari *threshold* yang dikalikan 2x nilai rata-rata pada kerapatan 100 *node* dan lebih baik 3,31% pada *node* dengan kerapatan 200 *node*. Berdasarkan hasil uji coba yang dilakukan ditarik kesimpulan jika nilai terbaik untuk *threshold* pada penelitian ini adalah menggunakan 3x nilai rata-rata dari seluruh jarak antara satu *node* dengan *node* tetangganya.

Tabel 4. 7 Perbandingan Uji Coba *Threshold*

| Alive Node | 100 NODE     |              |              |          | 200 node    |             |             |          |
|------------|--------------|--------------|--------------|----------|-------------|-------------|-------------|----------|
|            | Threshol d*2 | Threshol d*3 | Threshol d*4 | Selis ih | Thresh old2 | Thresh old3 | thresh old4 | Selis ih |
| LND        | 7066         | 7104         | 6403         | 38       | 7368        | 7612        | 6403        | 244      |

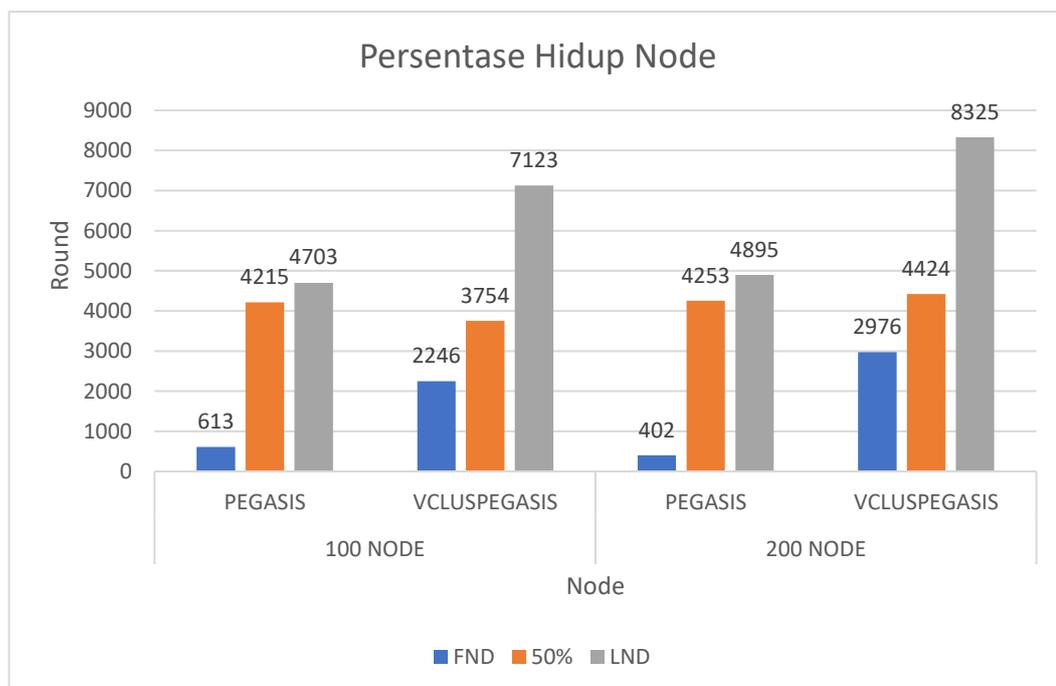
#### 4.3.2 Analisa Skenario 1

Dalam membandingkan protokol ini digunakan beberapa parameter seperti rentang waktu antara *node* pertama disembarkan hingga *node* pertama mati (*Network Lifetime*), persentase hidup *node* (*Alive Node in Network*), serta rata-rata energi yang dikonsumsi setiap kali melakukan transmisi (*Average Energy Consumed by A Node per Transmission*).

Pengujian pada skenario 1 dilakukan dengan cara membandingkan teknik pembagian area dan pembentukan *chain* secara adaptif dengan parameter area 100\*100, jumlah total *node* yang disebar adalah 100 dan 200, serta jumlah *grid* dalam satu area terdiri dari 5 *grid* dengan protokol PEGASIS asli yang di hitung dengan parameter area 100\*100 dan *node* yang disebar sebanyak 100 dan 200. Dari pengujian skenario yang dilakukan diperoleh hasil pengujian sebagai berikut:

#### 4.3.2.1. *Persentase Hidup Node (Alive Node in Network)*

Pengukuran utama dalam permasalahan WSN adalah persentase hidup *node* (*Alive Node in Network*) karena keterbatasannya sumber daya pada setiap *node* yang ada. Persentase hidup *node* dianggap sebagai suatu permasalahan yang sangat penting bagi para peneliti ketika merancang protokol routing WSN (Marhoon, Mahmuddin, and Nor 2016). Persentase hidup *node* sendiri merupakan rentang waktu antara *node* pertama disebar hingga *node* terakhir mati. Dalam analisa tentang persentase hidup *node* diukur berdasarkan beberapa kriteria utama yaitu berapa *round* yang dibutuhkan dalam area simulasi tersebut ketika *node* pertama mati (*First Node Die*), lalu pada *round* berapa 50 % *node* mati (*Half Node Die*), dan terakhir pada *round* berapa seluruh *node* mati (Marhoon, Mahmuddin, and Nor 2016).



Gambar 4. 2 *Alive Node in Network* pada Skenario I

Terlihat pada Gambar 4.2, protokol VCLUS-PEGASIS memiliki jumlah persentase hidup *node* yang lebih baik daripada PEGASIS. Dalam hal persentase hidup *node* nya baik itu dalam area yang memiliki 100 *node* maupun 200 *node*, VCLUS-PEGASIS memiliki jumlah *node* hidup yang lebih lama dari pada PEGASIS.

Pada kriteria pertama yaitu FND (*First Node Die*) pada 100 *node*, VCLUS-PEGASIS mati pada transmisi (*round*) ke 2246 sedangkan pada PEGASIS asli, transmisi pertama mati pada saat transmisi ke 613. Begitu pula pada saat diuji menggunakan 200 *node*, VCLUS-PEGASIS bahkan *node* pertama yang mati di transmisi yang lebih tinggi saat menggunakan 200 *node* ini yaitu mati pada saat transmisi ke 2976 sedangkan pada PEGASIS asli dengan menggunakan 200 *node* ini, transmisi pertama mati lebih cepat dari percobaan menggunakan 100 *node*. Adapun PEGASIS asli mati pada *node* ke 402. Perbedaan jauh matinya *node* pada saat awal ini (*First Node Die*) ini disebabkan pada saat pengiriman transmisi ke BS, VCLUS-PEGASIS melakukan pengiriman dengan mempertimbangkan jarak antara satu *node* dengan *node* tetangganya dalam satu *grid*, sehingga antar satu *node* dengan *node* tetangganya tidak perlu untuk mengirim data terlalu jauh dan menyebabkan energi yang diperlukan dalam pengiriman awal tidak terlalu banyak. Berdasarkan hasil pengujian pada skenario pertama tentang kriteria FND dapat terlihat pada Tabel 4.8 berikut:

Tabel 4. 8 FND pada Skenario 1

| Alive Node | 100 Node |              | 200 Node |              |
|------------|----------|--------------|----------|--------------|
|            | PEGASIS  | VCLUSPEGASIS | PEGASIS  | VCLUSPEGASIS |
| FND        | 613      | 2246         | 402      | 2976         |

Pada kriteria kedua yaitu pada saat *node* mati berjumlah 50% (*Half Node Die*) terlihat pada saat 100 *node* jika PEGASIS asli lebih baik dari pada VCLUS-PEGASIS. Pada PEGASIS, *node* tersisa 50% ketika transmisi berada di *round* 4215, sedangkan VCLUS-PEGASIS menyisakan 50% *node* pada saat transmisi di *round* 3754. Namun hal berbeda terlihat pada saat dilakukan uji coba menggunakan 200 *node*, VCLUS-PEGASIS mencapai hasil yang lebih baik daripada PEGASIS asli dimana 50% *node* mati pada saat transmisi berada di *round* 4424 sedangkan

PEGASIS asli mencapai 50% *node* mati pada saat melakukan transmisi di *round* 4253. Hasil yang tidak terlalu baik pada HND disebabkan karena pada VCLUS-PEGASIS ada *grid* yang jauh dari BS, sehingga pada *grid* tersebut *leader* melakukan transmisi lebih jauh dari pada *leader* yang lebih dekat dengan BS. Berdasarkan hasil pengujian pada skenario pertama tentang kriteria HND dapat terlihat pada Tabel 4.9 berikut:

Tabel 4. 9 HND pada Skenario 1

| Alive Node | 100 Node |              | 200 Node |              |
|------------|----------|--------------|----------|--------------|
|            | PEGASIS  | VCLUSPEGASIS | PEGASIS  | VCLUSPEGASIS |
| HND        | 4215     | 3754         | 4253     | 4424         |

Pada kriteria terakhir pada persentase hidup *node* (*Alive Node in Network*) adalah pada saat semua *node* mati (*Last Node Die*). Terlihat pada Gambar 4.1 jika VCLUS-PEGASIS memiliki LND yang jauh lebih dari pada PEGASIS asli baik pada saat di uji coba menggunakan 100 *node*. Pada VCLUS-PEGASIS yang memiliki 100 *node* terlihat jika *node* terakhir mati pada saat melakukan transmisi di *round* 7123 sedangkan pada PEGASIS asli mati pada saat transmisi berada di *round* ke 4703. Pengujian yang dilakukan dengan jumlah 200 *node* juga mendapatkan hasil dimana VCLUS-PEGASIS memiliki transmisi LND yang lebih tinggi daripada PEGASIS asli. Terlihat jika pada PEGASIS asli, *node* terakhir mati pada saat transmisi berada pada *round* ke 4895 sedangkan pada VCLUS-PEGASIS mati pada saat transmisi berada pada *round* ke 8325. Perbedaan jauh matinya seluruh *node* antara PEGASIS asli dan VCLUS-PEGASIS ini disebabkan karena VCLUS-PEGASIS mengeluarkan energi yang lebih merata pada setiap *node* nya pada satu *grid* walaupun pada *grid* terjauh dari BS akan mengirimkan energi yang lebih besar. Berdasarkan hasil pengujian pada skenario pertama tentang kriteria LND tersebut, dapat dibuat tabel dari hasil pengujian persentase hidup *node* yang dapat dilihat pada Tabel 4.10 berikut:

Tabel 4. 10 Persentase Hidup Node pada Skenario 1

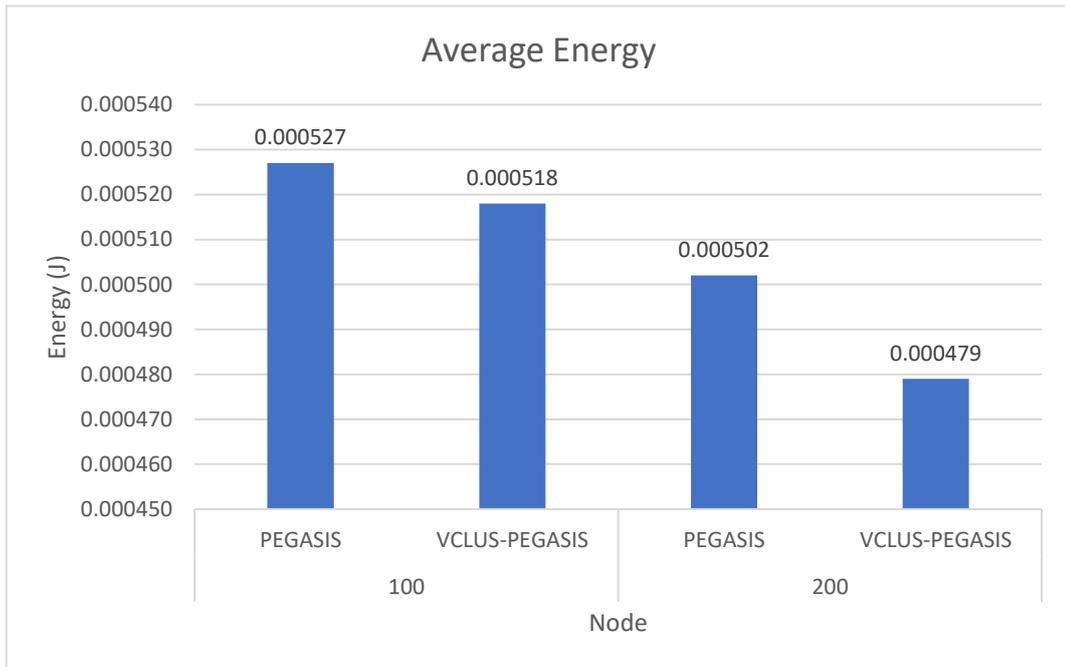
| Alive Node | 100 Node |               | Selisih | 200 Node |               | Selisih |
|------------|----------|---------------|---------|----------|---------------|---------|
|            | PEGASIS  | VCLUS-PEGASIS |         | PEGASIS  | VCLUS-PEGASIS |         |
| FND        | 613      | 2246          | 1633    | 402      | 2976          | 2574    |
| HND        | 4215     | 3754          | 461     | 4253     | 4424          | 171     |
| LND        | 4703     | 7123          | 2420    | 4895     | 8325          | 3430    |

Dari analisa yang telah dilakukan pada percobaan pertama berdasarkan dari Tabel 4.10 dapat disimpulkan jika dalam percobaan pertama untuk persentase hidup *node* dengan jumlah 100 *node* dan 200 *node* dengan 5 *grid* bahwa VCLUS-PEGASIS memiliki nilai persentase hidup yang lebih baik dari pada PEGASIS asli baik pada saat *node* berada pada lingkungan yang lebih rapat maupun yang tidak. Pada lingkungan 100 *node*, FND pada VCLUS-PEGASIS mendapatkan selisih transmisi sebanyak 1633 *round* atau sekitar 72,7% lebih dari PEGASIS, namun pada saat tersisa 50 % *node* atau HND, PEGASIS mencatatkan hasil yang lebih baik dari VCLUS-PEGASIS dengan selisih transmisi 461 *round* atau 10,94% lebih banyak dari VCLUS-PEGASIS tetapi pada saat *node* habis atau LND, VCLUS-PEGASIS mendapatkan kembali hasil yang lebih baik dimana VCLUS-PEGASIS mampu untuk bertahan sekitar 33,97% *round* lebih baik dari PEGASIS

Pada percobaan ke 2, FND pada VCLUS-PEGASIS mendapatkan selisih transmisi sebanyak 2574 *round* atau 86,49% lebih baik dari pada PEGASIS asli. Berbeda dengan percobaan pertama, HND pada VCLUS-PEGASIS juga mendapatkan hasil selisih lebih baik dari pada PEGASIS asli dimana terlihat *node* mampu bertahan 3,87% *round* lebih lama dari PEGASIS asli dan terakhir pada LND terlihat jika VCLUS-PEGASIS mampu untuk bertahan 3430 *round* atau 41,2% lebih baik dari pada PEGASIS asli.

#### 4.3.2.2. Average Energy Consumed by A Node

Dalam WSN, selain dari pentingnya persentase hidup *node* sebagai evaluasi pertama dalam pengukuran kinerja dari WSN, diperlukan juga pengukuran rata-rata energi yang diperlukan *node* dalam melakukan transmisi ke *base station* hingga *node* pertama mati (*Average Energy Consumed by A Node*).



Gambar 4. 3 *Average Energy Consumed by A Node* pada Skenario 1

Dapat dilihat pada Gambar 4.3 perbandingan rata-rata energi yang diperlukan *node* pada PEGASIS dan VCLUS-PEGASIS. Pada percobaan pertama yaitu pada 100 *node* tercatat jika VCLUS-PEGASIS memiliki energi rata-rata yang dikeluarkan untuk melakukan transmisi lebih rendah daripada PEGASIS asli. Pada VCLUS-PEGASIS rata-rata energi yang dikonsumsi pada setiap transmisinya adalah 0,000518 *joule* sedangkan pada PEGASIS asli memiliki rata-rata energi yang dikonsumsi sebanyak 0,000527 *joule*. Terlihat pada Tabel 4.11 jika selisih yang dihasilkan pada percobaan pertama adalah 0,000009 *joule* yang berarti pada percobaan pertama ini VCLUS-PEGASIS mencatatkan energi yang lebih rendah 1.71% dari PEGASIS asli.

Pada percobaan ke 2 yaitu dilakukan dengan 200 *node* tercatat apabila VCLUS-PEGASIS juga mencatatkan transmisi energi yang lebih rendah daripada

PEGASIS. Rata-rata energi yang dikeluarkan pada setiap transmisi dari VCLUS-PEGASIS adalah 0,000479 *joule* sedangkan pada PEGASIS asli rata-rata energi yang dikeluarkan mencapai 0,000502 *joule*. Terlihat pada Tabel 4.12 jika selisih pada kedua teknik ini adalah 0,0000023 *joule* yang berarti VCLUS-PEGASIS mentransmisikan 4,58 % energi yang lebih rendah dari pada PEGASIS asli.

Tabel 4. 11 Average Energy Consumed by A Node 100 node Skenario 1

| Node   | 100 Node |              | Selisih  |
|--------|----------|--------------|----------|
| Metode | PEGASIS  | VCLUSPEGASIS |          |
| Energi | 0.000527 | 0.000518     | 0.000009 |

Tabel 4. 12 Average Energy Consumed by A Node 200 node Skenario 1

| Node   | 200 Node |              | Selisih  |
|--------|----------|--------------|----------|
| Metode | PEGASIS  | VCLUSPEGASIS |          |
| Energi | 0.000502 | 0.000479     | 0.000023 |

Dapat disimpulkan pada 2 percobaan diatas antara VCLUS-PEGASIS yang memiliki 5 *grid* dan PEGASIS asli dengan 100 *node* dan 200 *node* didapat jika VCLUS-PEGASIS memiliki transmisi dengan rata-rata energi yang lebih rendah dari pada PEGASIS asli. Terlihat juga jika semakin rapat *node* pada satu area juga berpengaruh terhadap transmisi rata-rata energi yang dikeluarkan dimana semakin rendah energi transmisi dari suatu *node*. Walaupun dapat disimpulkan jika VCLUS-PEGASIS memiliki nilai rata-rata energi yang lebih rendah dari pada PEGASIS namun selisihnya tidak terlalu jauh. Nilai yang tidak terlalu jauh berbeda ini disebabkan juga karena pengaruh *grid* dimana semakin jauh *grid* dari *BS* menyebabkan energi yang dikeluarkan akan semakin besar.

### 4.3.3 Analisa Skenario 2

Pengujian pada skenario 1 dilakukan dengan cara membandingkan teknik pembagian area dan pembentukan *chain* secara adaptif dengan parameter area 100\*100, jumlah total *node* yang disebar adalah 100 dan 200, serta jumlah *grid* dalam satu area terdiri dari 7 *grid* dengan protokol PEGASIS asli yang di hitung dengan parameter area 100\*100 dan *node* yang disebar sebanyak 100 dan 200. Dari pengujian skenario yang dilakukan diperoleh hasil pengujian sebagai berikut:

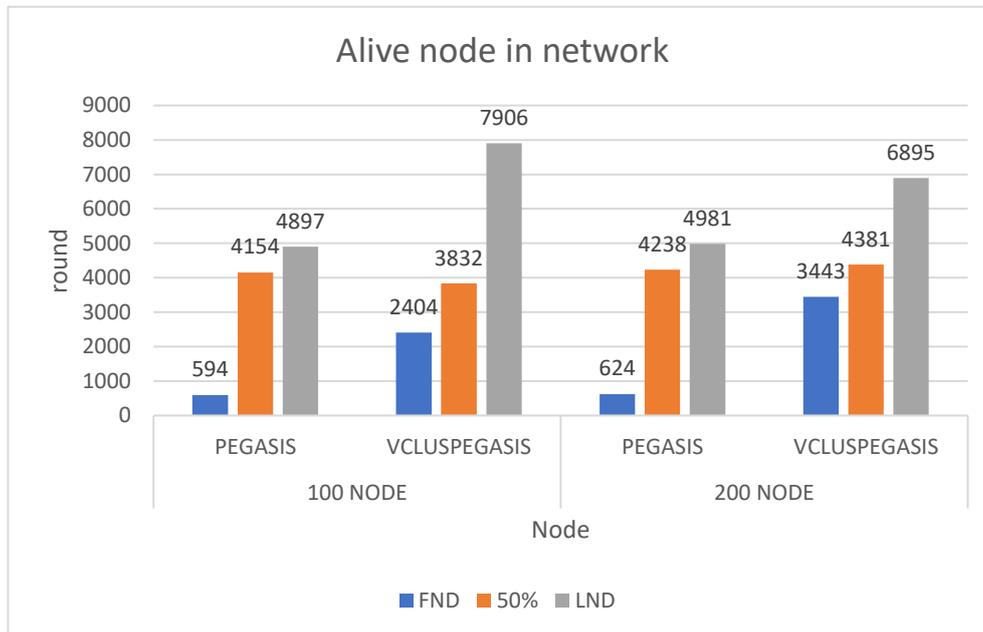
#### 4.3.3.1. *Persentase Hidup Node (Alive Node in Network) Skenario 2*

Seperti pada sub 4.3.2.1, dalam analisa tentang persentase hidup *node* diukur berdasarkan beberapa kriteria utama yaitu berapa *round* yang dibutuhkan dalam area simulasi tersebut ketika *node* pertama mati (*First Node Die*), lalu pada *round* berapa 50 % *node* mati (*Half Node Die*), dan terakhir pada *round* berapa seluruh *node* mati. Terlihat pada Gambar 4.2, protokol VCLUS-PEGASIS memiliki jumlah persentase hidup *node* yang lebih baik daripada PEGASIS. Dalam hal persentase hidup *node* nya baik itu dalam area yang memiliki 100 *node* maupun 200 *node*, VCLUS-PEGASIS memiliki jumlah *node* hidup yang lebih lama dari pada PEGASIS.

Pada kriteria pertama yaitu FND (*First Node Die*) pada 100 *node*, VCLUS-PEGASIS mati pada transmisi (*round*) ke 2404 sedangkan pada PEGASIS asli, transmisi pertama mati pada saat transmisi ke 594. Begitu pula pada saat diuji menggunakan 200 *node*, VCLUS-PEGASIS bahkan *node* pertama yang mati di transmisi yang lebih tinggi saat menggunakan 200 *node* ini yaitu mati pada saat transmisi ke 3443 sama seperti pada PEGASIS asli dengan menggunakan 200 *node* ini, transmisi pertama mati pada *node* ke 624. Perbedaan jauh matinya *node* pada saat awal ini (*First Node Die*) ini disebabkan pada saat pengiriman transmisi ke BS, VCLUS-PEGASIS melakukan pengiriman dengan mempertimbangkan jarak antara satu *node* dengan *node* tetangganya dalam satu *grid*, sehingga antar satu *node* dengan *node* tetangganya tidak perlu untuk mengirim data terlalu jauh dan menyebabkan energi yang diperlukan dalam pengiriman awal tidak terlalu banyak. Berdasarkan hasil pengujian pada skenario kedua tentang kriteria FND dapat terlihat pada Tabel 4.13 berikut:

Tabel 4. 13 FND pada Skenario 1

| Alive Node | 100 Node |              | 200 Node |              |
|------------|----------|--------------|----------|--------------|
|            | PEGASIS  | VCLUSPEGASIS | PEGASIS  | VCLUSPEGASIS |
| FND        | 594      | 2404         | 624      | 3443         |



Gambar 4. 4 *Alive Node in Network* pada Skenario II

Pada kriteria kedua yaitu pada saat *node* mati berjumlah 50% (*Half Node Die*) terlihat pada saat 100 *node* jika PEGASIS asli lebih baik dari pada VCLUS-PEGASIS. Pada PEGASIS, *node* tersisa 50% ketika transmisi berada di *round* 4154, sedangkan VCLUS-PEGASIS menyisakan 50% *node* pada saat transmisi di *round* 3832. Namun hal berbeda terlihat pada saat dilakukan uji coba menggunakan 200 *node*, VCLUS-PEGASIS mencapai hasil yang lebih baik daripada PEGASIS asli dimana 50% *node* mati pada saat transmisi berada di *round* 4238 sedangkan PEGASIS asli mencapai 50% *node* mati pada saat melakukan transmisi di *round* 4381. Hasil yang tidak terlalu baik pada HND sama seperti pada skenario 1 yang menggunakan 5 *grid* yaitu disebabkan karena pada VCLUS-PEGASIS ada *grid* yang jauh dari BS, sehingga pada *grid* tersebut *leader* melakukan transmisi lebih jauh dari pada *leader* yang lebih dekat dengan BS. Berdasarkan hasil pengujian pada skenario kedua tentang kriteria HND dapat terlihat pada Tabel 4.14 berikut:

Tabel 4. 14 HND pada Skenario 1

| Alive Node | 100 Node |              | 200 Node |              |
|------------|----------|--------------|----------|--------------|
|            | PEGASIS  | VCLUSPEGASIS | PEGASIS  | VCLUSPEGASIS |
| HND        | 4154     | 3832         | 4238     | 4381         |

Pada kriteria terakhir pada persentase hidup *node* (*Alive Node in Network*) adalah pada saat semua *node* mati (*Last Node Die*). Terlihat pada Gambar 4.15 jika VCLUS-PEGASIS memiliki LND yang jauh lebih dari pada PEGASIS asli baik pada saat di uji coba menggunakan 100 *node*. Pada VCLUS-PEGASIS yang memiliki 100 *node* terlihat jika *node* terakhir mati pada pada saat melakukan transmisi di *round* 7906 sedangkan pada PEGASIS asli mati pada saat transmisi berada di *round* ke 4897. Pengujian yang dilakukan dengan jumlah 200 *node* juga mendapatkan hasil dimana VCLUS-PEGASIS memiliki transmisi LND yang lebih tinggi daripada PEGASIS asli. Terlihat jika pada PEGASIS asli, *node* terakhir mati pada saat transmisi berada pada *round* ke 4981 sedangkan pada VCLUS-PEGASIS mati pada saat transmisi berada pada *round* ke 6895. Perbedaan jauh matinya seluruh *node* antara PEGASIS asli dan VCLUS-PEGASIS ini disebabkan karena VCLUS-PEGASIS mengeluarkan energi yang lebih merata pada setiap *node* nya pada satu *grid* walaupun pada *grid* terjauh dari BS akan mengirimkan energi yang lebih besar. Berdasarkan hasil pengujian pada skenario kedua tentang kriteria LND tersebut, dapat dibuat tabel dari hasil pengujian persentase hidup *node* yang dapat dilihat pada Tabel 4.15 berikut:

Tabel 4. 15 Persentase Hidup Node pada Skenario 2

| Alive Node | 100 Node |               | Selisih | 200 Node |               | Selisih |
|------------|----------|---------------|---------|----------|---------------|---------|
|            | PEGASIS  | VCLUS-PEGASIS |         | PEGASIS  | VCLUS-PEGASIS |         |
| FND        | 594      | 2404          | 1810    | 624      | 3443          | 2819    |
| HND        | 4154     | 3832          | 322     | 4238     | 4381          | 143     |
| LND        | 4897     | 7906          | 3009    | 4981     | 6895          | 1914    |

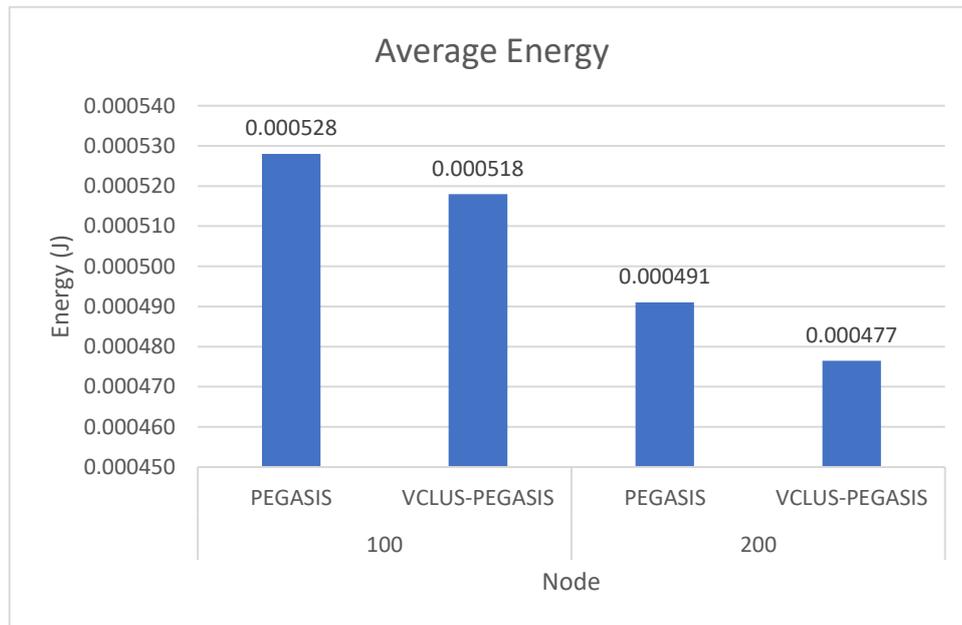
Dari analisa yang telah dilakukan pada percobaan pertama berdasarkan dari Tabel 4.15 dapat disimpulkan jika dalam percobaan kedua untuk persentase hidup *node* dengan jumlah 100 *node* dan 200 *node* dengan 7 *grid* bahwa VCLUS-PEGASIS memiliki nilai persentase hidup yang lebih baik dari pada PEGASIS asli baik pada saat *node* berada pada lingkungan yang lebih rapat maupun yang tidak.

Pada lingkungan 100 *node*, FND pada VCLUS-PEGASIS mendapatkan selisih transmisi sebanyak 1810 *round* atau sekitar 75,3% lebih dari PEGASIS, namun pada saat tersisa 50 % *node* atau HND, PEGASIS mencatatkan hasil yang lebih baik dari VCLUS-PEGASIS dengan selisih transmisi 322 *round* atau 7,75% lebih banyak dari VCLUS-PEGASIS tetapi pada saat *node* habis atau LND, VCLUS-PEGASIS mendapatkan kembali hasil yang lebih baik dimana VCLUS-PEGASIS mampu untuk bertahan sekitar 38,1% *round* lebih baik dari PEGASIS

Pada percobaan ke 2, FND pada VCLUS-PEGASIS mendapatkan selisih transmisi sebanyak 2819 *round* atau 81,9% lebih baik dari pada PEGASIS asli. Berbeda dengan percobaan pertama, HND pada VCLUS-PEGASIS juga mendapatkan hasil selisih lebih baik dari pada PEGASIS asli dimana terlihat *node* mampu bertahan 3,26% *round* lebih lama dari PEGASIS asli dan terakhir pada LND terlihat jika VCLUS-PEGASIS mampu untuk bertahan 1914 *round* atau 27,76% lebih baik dari pada PEGASIS asli.

#### ***4.3.3.2. Average Energy Consumed by A Node***

Dapat dilihat pada Gambar 4.5 perbandingan rata-rata energi yang diperlukan *node* pada PEGASIS dan VCLUS-PEGASIS. Pada percobaan pertama yaitu pada 100 *node* tercatat jika VCLUS-PEGASIS memiliki energi rata-rata yang dikeluarkan untuk melakukan transmisi lebih rendah daripada PEGASIS asli. Pada VCLUS-PEGASIS rata-rata energi yang dikonsumsi pada setiap transmisinya adalah 0,000518 *joule* sedangkan pada pada PEGASIS asli memiliki rata-rata energi yang dikonsumsi sebanyak 0,000528 *joule*. Terlihat pada Tabel 4.16 jika selisih yang dihasilkan pada percobaan pertama adalah 0,000010 *joule* yang berarti pada percobaan pertama ini VCLUS-PEGASIS mencatatkan energi yang lebih rendah 1,93% dari PEGASIS asli.



Gambar 4. 5 *Average Energy Consumed by A Node* pada Skenario 2

Pada percobaan ke 2 yaitu dilakukan dengan 200 *node* tercatat apabila VCLUS-PEGASIS juga mencatatkan transmisi energi yang lebih rendah daripada PEGASIS. Rata-rata energi yang dikeluarkan pada setiap transmisi dari VCLUS-PEGASIS adalah 0,000477 *joule* sedangkan pada PEGASIS asli rata-rata energi yang dikeluarkan mencapai 0,000491 *joule*. Terlihat pada Tabel 4.12 jika selisih pada kedua teknik ini adalah 0,000014 *joule* yang berarti VCLUS-PEGASIS mentransmisikan 2,85 % energi yang lebih rendah dari pada PEGASIS asli.

Dapat disimpulkan pada 2 percobaan diatas antara VCLUS-PEGASIS yang memiliki 7 *grid* dan PEGASIS asli dengan 100 *node* dan 200 *node* didapat jika VCLUS-PEGASIS memiliki transmisi dengan rata-rata energi yang lebih rendah dari pada PEGASIS asli. Sama seperti pada percobaan pertama jika jika semakin rapat *node* pada satu area juga berpengaruh terhadap transmisi rata-rata energi yang dikeluarkan dimana semakin rendah energi transmisi dari suatu *node*. Walaupun dapat disimpulkan jika VCLUS-PEGASIS memiliki nilai rata-rata energi yang lebih rendah dari pada PEGASIS namun selisihnya tidak terlalu jauh. Nilai yang tidak terlalu jauh berbeda ini disebabkan juga karena pengaruh *grid* yang jauh dari BS dimana semakin jauh *grid* dari BS menyebabkan energi yang dikeluarkan akan semakin besar.

Tabel 4. 16 Average Energy Consumed by A Node 100 node Skenario 2

| Node   | 100 Node |              | Selisih  |
|--------|----------|--------------|----------|
| Metode | PEGASIS  | VCLUSPEGASIS |          |
| Energi | 0,000528 | 0,000518     | 0,000010 |

Tabel 4. 17 Average Energy Consumed by A Node 200 node Skenario 2

| Node   | 200 Node |              | Selisih  |
|--------|----------|--------------|----------|
| Metode | PEGASIS  | VCLUSPEGASIS |          |
| Energi | 0,000491 | 0,000477     | 0,000014 |

#### 4.3.4. Analisa Skenario 3

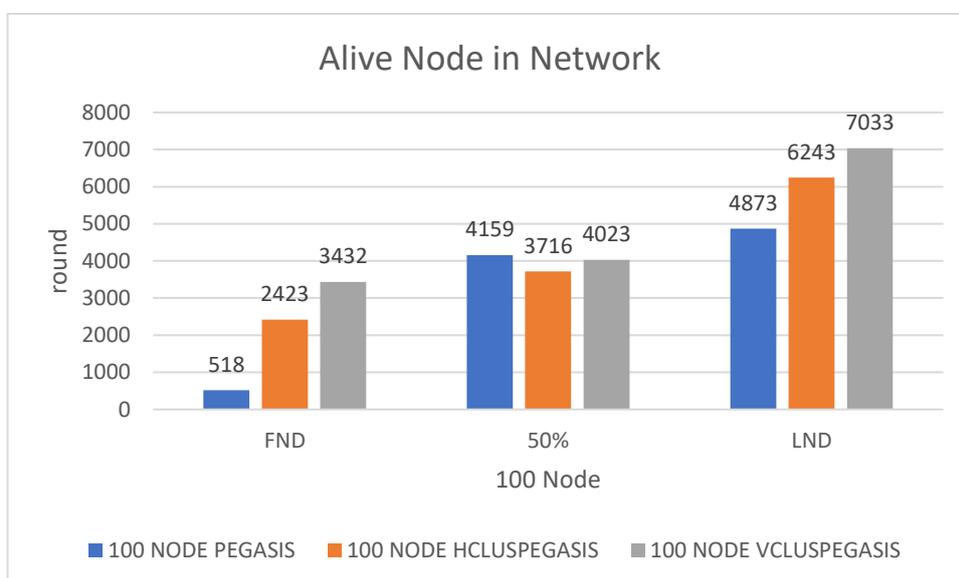
Pada pengujian pada skenario 3 ini akan dilakukan dengan cara membandingkan teknik pembagian area dan pembentukan *chain* secara adaptif dengan menambahkan area horizontal (HCLUS-PEGASIS) sebagai pembanding dengan area vertikal dan PEGASIS asli dengan parameter area 100\*100, jumlah total *node* yang disebar adalah 100, serta jumlah *grid* dalam satu area terdiri dari 5 *grid*. Dari pengujian skenario yang dilakukan diperoleh hasil pengujian sebagai berikut:

##### 4.3.4.1. Persentase Hidup Node (Alive Node in Network) Skenario 3

Seperti pada sub 4.3.2.1, dalam analisa tentang persentase hidup *node* diukur berdasarkan beberapa kriteria utama yaitu berapa *round* yang dibutuhkan dalam area simulasi tersebut ketika *node* pertama mati (*First Node Die*), lalu pada *round* berapa 50 % *node* mati (*Half Node Die*), dan terakhir pada *round* berapa seluruh *node* mati. Terlihat pada Gambar 4.6, protokol HCLUS-PEGASIS memiliki jumlah persentase hidup *node* yang lebih baik daripada PEGASIS namun tidak lebih baik dari VCLUS-PEGASIS. VCLUS-PEGASIS memiliki jumlah *node* hidup yang lebih lama dari pada PEGASIS maupun HCLUS-PEGASIS.

Dari analisa yang telah dilakukan pada percobaan ini berdasarkan dari Tabel 4.18 dapat disimpulkan jika dalam percobaan kedua untuk persentase hidup *node* dengan jumlah 100 *node* dengan 5 *grid* bahwa VCLUS-PEGASIS memiliki nilai persentase hidup yang lebih baik dari pada PEGASIS dan HCLUS-PEGASIS. Pada lingkungan 100 *node* ini, FND pada HCLUS-PEGASIS mendapatkan selisih transmisi sebanyak 1829 *round* atau sekitar 75,48% lebih dari PEGASIS, namun FND pada VCLUS-PEGASIS lebih baik dengan selisih 29,4% lebih baik dari

HCLUS-PEGASIS, pada saat tersisa 50 % *node* atau HND, PEGASIS mencatatkan hasil yang lebih baik dari HCLUS-PEGASIS dan VCLUS-PEGASIS. Selisih HND antara VCLUS-PEGASIS dan HCLUS-PEGASIS adalah 307 *round* atau 7,63% lebih baik VCLUS-PEGASIS, namun jika dibandingkan dengan PEGASIS asli, HND pada VCLUS-PEGASIS sama seperti skenario 1 dan 2 dimana PEGASIS asli lebih baik sekitar 131 *round* atau 3,15% dari VCLUS-PEGASIS. Pada LND baik HCLUS-PEGASIS maupun VCLUS-PEGASIS mampu menampilkan hasil yang lebih baik dari PEGASIS asli dimana HCLUS-PEGASIS mampu mempertahankan sekitar 1346 *round* lebih baik dari pada PEGASIS asli dan VCLUS-PEGASIS mampu mempertahankan 790 *round* lebih baik dari HCLUS-PEGASIS.



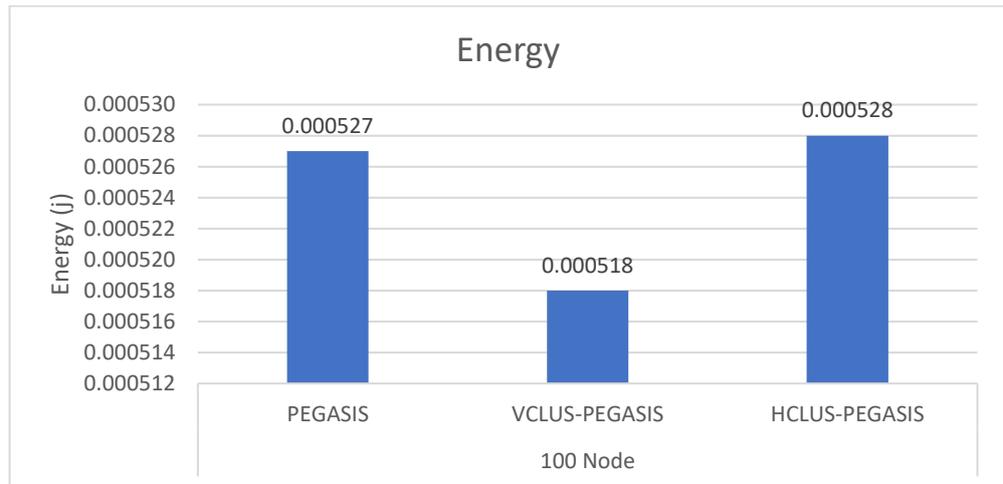
Gambar 4. 6 *Alive Node in Network* pada Skenario III

Tabel 4. 18 *Persentase Hidup Node* pada Skenario 3

| Alive Node | 100 Node |          |               |          |               |
|------------|----------|----------|---------------|----------|---------------|
|            | PEGASIS  | Selisi h | HCLUS-PEGASIS | Selisi h | VCLUS-PEGASIS |
| FND        | 594      | 1829     | 2423          | 1009     | 3432          |
| HND        | 4154     | 438      | 3716          | 307      | 4023          |
| LND        | 4897     | 1346     | 6243          | 790      | 7033          |

#### 4.3.3.3. Average Energy Consumed by A Node

Dapat dilihat pada Gambar 4.7 perbandingan rata-rata energi yang diperlukan *node* pada PEGASIS, HCLUS-PEGASIS, dan VCLUS-PEGASIS. Terlihat pada Tabel 4.19 jika selisih yang dihasilkan pada skenario ketiga ini terlihat jika VCLUS-PEGASIS mampu untuk menghasilkan energi yang paling rendah dan HCLUS-PEGASIS membutuhkan energi yang paling besar diantara ketiganya. Pada skenario ketiga ini VCLUS-PEGASIS mampu untuk menghemat 0,000009 joule atau 1,74% lebih baik dari PEGASIS asli sedangkan PEGASIS asli sendiri lebih baik 0,19% dari HCLUS-PEGASIS. Tidak hematnya HCLUS-PEGASIS ini disebabkan karena *node* pada *grid* yang paling jauh dari BS, sehingga pada *grid* tersebut *leader* melakukan transmisi lebih jauh dari pada *leader* yang lebih dekat dengan BS yang menyebabkan energi pada *grid* yang paling jauh tersebut akan memuat beban yang besar pada saat melakukan transmisi energi.



Gambar 4. 7 Average Energy Consumed by A Node pada Skenario 3

Tabel 4. 19 Average Energy Consumed by A Node Skenario 3

| Node   | 100 Node |          |               |          |              |
|--------|----------|----------|---------------|----------|--------------|
| Metode | PEGASIS  | Selisih  | HCLUS-PEGASIS | Selisih  | VCLUSPEGASIS |
| Energi | 0,000527 | 0,000001 | 0,00058       | 0,000009 | 0,000518     |

## BAB 5

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Pengujian dan analisis yang telah dilakukan berhasil mendapatkan beberapa kesimpulan, yaitu:

1. Protokol VCLUS-PEGASIS merupakan modifikasi dari PEGASIS dengan cara melakukan pembagian area WSN menjadi beberapa *grid* dalam bentuk vertikal dengan luas area *sensing* yang sama dan melakukan pembentukan *chain* baru pada setiap *grid* dengan menggunakan algoritma K-Means jika ada *node* di dalam *chain* yang mempunyai jarak yang melebihi nilai *threshold*.
2. Berdasarkan pada hasil uji coba yang telah dilakukan pada protokol VCLUS-PEGASIS dengan menggunakan MATLAB didapatkan jika:
  - a. Nilai *threshold* terbaik berdasarkan uji coba yang dilakukan dalam penelitian ini didapatkan adalah nilai rata-rata jarak antara satu *node* dengan *node* tetangganya dikalikan dengan 3 kali lipat. Dari uji coba menggunakan kerapatan 100 *node* dan 200 *node* hasilnya menunjukkan jika nilai dikalikan 3 kali lipat nilai rata-rata lebih baik dari nilai yang dikalikan 2 kali lipat nilai rata-rata dan 4x lipat nilai rata-rata.
  - b. Persentase hidup *node* pada metode VCLUS-PEGASIS di skenario 1 meningkat sekitar 33,97% dari PEGASIS asli pada 100 *node* sedangkan pada kerapatan *node* yang lebih banyak yaitu 200 *node*, VCLUS-PEGASIS juga mampu bertahan sekitar 41,2% lebih lama dari pada PEGASIS asli. Pada skenario 2 meningkat sekitar 38,1 % dari PEGASIS asli pada kerapatan 100 *node* dan 27,76% pada kerapatan 200 *node*.
  - c. *Average energy consumed by a node* pada VCLUS-PEGASIS di skenario 1 dapat membuat transmisi energi menjadi lebih rendah hingga 4,58% daripada PEGASIS asli. Pada skenario 7 *grid* dengan kepadatan jumlah *node* yaitu 100 dan 200, VCLUS-PEGASIS mampu menghemat hingga 2,85% dari PEGASIS asli.

## 5.2 Saran

Saran yang dapat diberikan dari hasil uji coba dan analisis yang telah dilakukan yaitu:

1. Studi lebih lanjut mengenai pembagian area WSN dengan menerapkan *grid* dengan bentuk horisontal atau gabungan dari keduanya.
2. Pengembangan metode VCLUS-PEGASIS dengan menambahkan *multihop routing* secara dinamis pada setiap *leader* yang jaraknya lebih jauh dari BS

*[halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- A.Rajeswari, and R.Manavalan. 2014. "Data Collection Methods in Wireless Sensor Network : A Study." *INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)* 2(Ix): 259–72.
- Afif. 2017. "Jurnal Instek Volume 2 Nomor 2 April 2017." 2(April): 81–91.
- Agrawal, Chetan, and Dubey Shivendra. 2013. "A Survey of Data Collection Techniques in Wireless Sensor Network." *International Journal of Advances in Engineering & Technology* 6(4): 1664–73.
- Akkari, Wafa, Badia Bouhdid, and Abdelfettah Belghith. 2015. "LEATCH : Low Energy Adaptive Tier Clustering Hierarchy." *Procedia - Procedia Computer Science* 52(Ant): 365–72. <http://dx.doi.org/10.1016/j.procs.2015.05.110>.
- Anjum, Naziya, and Masood Ahmad. 2017. "Gateway Based Energy Efficient Routing : GEER." *International Journal of Advances Research, Ideas, And Innovations In Technology* 3(4): 483–88.
- Arora, Preeti, and Shipra Varshney. 2016. "Analysis of K-Means and K-Medoids Algorithm For Big Data." *Procedia - Procedia Computer Science* 78(December 2015): 507–12. <http://dx.doi.org/10.1016/j.procs.2016.02.095>.
- Berkhin, P. 2006. "A Survey of Clustering Data Mining." *Springer, Berlin, Heidelberg* (c): 25–71.
- Chen, Kuong-Ho, Jyh-Ming Huang, and Chieh-Chuan Hsiao. 2009. "CHIRON : An Energy-Efficient Chain-Based Hierarchical Routing Protocol in Wireless Sensor Networks." In *2009 Wireless Telecommunications Symposium*, Prague: IEEE, 1–5.
- Chen, Young-long, and Jia-sheng Lin. 2012. "Energy Efficiency Analysis of a Chain-Based Scheme via Intra-Grid for Wireless Sensor Networks." *Computer Communications* 35(4): 507–16. <http://dx.doi.org/10.1016/j.comcom.2011.12.002>.
- Chong, Lou, Feng Wu, Jun Sun, and Chang Wen Chen. 2009. "Compressive Data Gathering for Large-Scale Wireless." In *Proceedings of the 15th Annual*

- International Conference on Mobile Computing and Networking, MOBICOM 2009*, , 145–56.
- Du, Kemei, Jie Wu, and Dan Zhou. 2003. “Chain-Based Protocols for Data Broadcasting and Gathering in the Sensor Networks £.” 00(C).
- Elshrkawey, Mohamed, Samiha M Elsherif, and M Elsayed Wahed. 2017. “An Enhancement Approach for Reducing the Energy Consumption in Wireless Sensor Networks.” *Journal of King Saud University - Computer and Information Sciences*: 0–8. <http://dx.doi.org/10.1016/j.jksuci.2017.04.002>.
- Felipe, B. 2016. “Combining K-Means and K-Harmonic with Fish School Search Algorithm for Data Clustering Task on Graphics Processing Units ADRIANE.” *Applied Soft Computing Journal (2016)*,: 3–32.
- Guo, Wenjing, Wei Zhang, and Gang Lu. 2010. “PEGASIS Protocol in Wireless Sensor Network Based on an Improved Ant Colony Algorithm.”
- Han, Jiawei, Micheline Kamber, and Jian Pei. 2014. Proceedings - 2013 International Conference on Machine Intelligence Research and Advancement, ICMIRA 2013 *Data Mining: Data Mining Concepts and Techniques*.
- Hani, Raed M Bani, and Abdalraheem A Ijeh. 2013. “A Survey on LEACH-Based Energy Aware Protocols for Wireless Sensor Networks.” 8(3).
- Heinzelman, Wendi Rabiner, Anantha Chandrakasan, and Hari Balakrishnan. 2000. “Energy-Efficient Communication Protocol for Wireless Microsensor Networks.” 00(c): 1–10.
- Jung, Sung-min, Young-ju Han, and Tai-myung Chung. 2007. “The Concentric Clustering Scheme.” *ICACT*: 260–65.
- Kamimura, Junpei, Naoki Wakamiya, and Masayuki Murata. 2006. “A Distributed Clustering Method for Energy-Efficient Data Gathering in Sensor Networks Junpei Kamimura , Naoki Wakamiya \* and Masayuki Murata.” *International Journal of Wireless and Mobile Computing* 1(2): 113–20.
- Li, Luo. 2016. “Data Aggregation in Wireless Sensor Networks.” *International Journal of Online and Biomedical Engineering* 12(11): 28–33. <https://doi.org/10.3991/ijoe.v12i11.6233>.
- Lindsey, Stephanie, and Cauligi S Raghavendra. 2002. “PEGASIS: Power-

- Efficient Gathering in Sensor Information Systems.” In *IEEE Aerospace Conference Proceedings*, Big Sky, 1125–30.
- Linping, Wang. 2010. “Improved Algorithm of PEGASIS Protocol Introducing Double Cluster Heads in Wireless Sensor Network.” : 148–51.
- Marhoon, Haydar Abdulameer, M Mahmuddin, and Shahrudin Awang Nor. 2016. “DCBRP : A Deterministic Chain - Based Routing Protocol for Wireless Sensor Networks.” *SpringerPlus*: 1–21.
- Melton, Jim et al. 1999. *Data Mining : Concepts and Techniques*.
- Omran, Mahamed, Andries Engelbrecht, and Ayed A Salman. 2007. “An Overview of Clustering Methods.” (November).
- Prajapat, Mukesh, and N.C. Barwar. 2018. “Performance Analysis of Energy Dissipation in WSNs Using Multi-Chain Performance Analysis of Energy Dissipation in WSNs Using Multi-Chain PEGASIS.” *International Journal of Computer Science and Information Technologies* 5(4): 8033–36.
- Ramluckun, Natasha, and Vandana Bassoo. 2018. “Energy-Efficient Chain-Cluster Based Intelligent Routing Technique for Wireless Sensor Networks Applied Computing and Informatics Energy-Efficient Chain-Cluster Based Intelligent Routing Technique for Wireless Sensor Networks.” *Applied Computing and Informatics* (April).  
<https://doi.org/10.1016/j.aci.2018.02.004>.
- Rana, Jigish, Sangeeta Vhatkar, and Mohommad Atique. 2015. “Comparative Study of PEGASIS and PDCH Protocols in Wireless Sensor Network.” In *International Conference and Workshop on Emerging Trends in Technology (ICWET)*, , 13–18.
- Ridwan, Ahmad, Rian Ferdian, and Rahmadi Kurnia. 2021. “Optimasi Protokol LEACH Untuk Meningkatkan Stabilitas Pada Wireless Sensor Network.” *Jurnal Rekayasa Sistem dan Teknologi Informasi* 1(10): 193–200.  
<http://jurnal.iaii.or.id>.
- Sen, Feng, Qi Bing, and Tang Liangrui. 2011. “An Improved Energy-Efficient PEGASIS-Based Protocol in Wireless Sensor Networks.” In *2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, Shanghai: IEEE, 2230–33.

- Sharma, Swati, Mahesh Kumar. 2015. "LEACH PROTOCOL : A Survey." *International Journal of Computer Science & Communication Networks* 5(4): 228–32.
- Su, Shengchao, and Shuguang Zhao. 2017. "An Optimal Clustering Mechanism Based on Fuzzy-C Means for Wireless Sensor Networks." *Sustainable Computing: Informatics and Systems*.  
<http://dx.doi.org/10.1016/j.suscom.2017.08.001>.
- Sunil Kumar Singh, Prabhat Kumar, and Jyoti Prakash Singh. 2017. "A Survey on Successors of LEACH Protocol." *IEEE Access* 5: 4298–4328.
- Wang, Zhibin et al. 2018. "Image Segmentation of Overlapping Leaves Based on Chan–Vese Model and Sobel Operator." *Information Processing in Agriculture* 5(1): 1–10.
- Warrier, Maya M., and Ajay Kumar. 2016. "An Energy Efficient Approach for Routing in Wireless Sensor Networks." In *Global Colloquium in Recent Advancement and Effectual Researches in Engineering, Science and Technology (RAEREST)*, The Author(s), 520–27.  
<http://dx.doi.org/10.1016/j.protcy.2016.08.140>.
- Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. 2000. "Energy-Efficient Communication Protocol for Wireless Microsensor Networks." In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Maui: IEEE, 1–10.