

# Perancangan Algoritma Pemilihan Cluster Head pada Jaringan Sensor Nirkabel secara Dinamis dengan Indikator Energi

I Gusti Ngurah Ady Kusuma

STMIK STIKOM Bali

Jl. Raya Puputan No 86 Renon, Denpasar-Bali, Telp: 0361-244445, Fax: 0361-264773  
e-mail: ady\_kusuma@stikom-bali.ac.id

## Abstrak

Jaringan sensor nirkabel merupakan salah satu bagian dari jaringan nirkabel yang merupakan sekumpulan dari beberapa *devices* berukuran kecil, dengan berat yang ringan, dan termasuk dalam *low-cost networks*. Untuk mengurangi jumlah paket data yang dikirim pada sebuah jaringan terdapat sebuah penelitian yang mengusulkan metode untuk mengumpulkan semua data sensor disebut *data aggregation*. Penggabungan data biasanya terjadi pada sebuah *cluster head* yang merupakan *node coordinator* dari sebuah *cluster*. Tingginya beban kerja *cluster head* cenderung akan menyebabkan kondisi *drop-state* yang lebih cepat dibandingkan *node* yang lain. Apabila *cluster head* mengalami *drop-state* maka wilayah *cluster* tersebut akan terisolir. Untuk menghindari kondisi *drop-state* yang terlalu dini, penelitian ini merancang algoritma pemilihan *cluster head* secara dinamis yang mampu mengurangi resiko terisolirnya sebuah *cluster* akibat *cluster head* yang tidak aktif. Penelitian ini memanfaatkan pertukaran informasi indikator antar *node* dan memberikan penilaian pada masing-masing *node* berdasarkan 3 indikator yaitu *energi-level*, *sink-distance*, dan *neighbors-count*. Berdasarkan informasi tersebut, *node* melakukan penilaian untuk menilai seluruh *node* dan memilih *node* dengan nilai tertinggi sebagai *cluster head*. Penilaian ini dilakukan secara *periodic* sehingga masing-masing *node* akan memiliki nilai yang sesuai dengan kondisi terbarunya.

**Kata kunci:** *sensor, nirkabel, cluster, dynamic, energy*

## 1. Pendahuluan

*Wireless communication* atau komunikasi nirkabel merupakan komunikasi yang terjadi antara dua *devices* yang disebut dengan *transmitter* dan *receiver* tanpa menggunakan perantara kabel [1]. *Transmitter* berfungsi sebagai pengirim data, sedangkan *receiver* berfungsi sebagai penerima data. Pada komunikasi nirkabel terdapat dua jenis *devices* yaitu *field device* dan *infrastructure device*. *Field device* berfungsi sebagai pengukur nilai dari suatu lingkungan sedangkan *infrastructure device* berfungsi sebagai tempat berkumpulnya semua informasi yang dikirim oleh *field device*.

*Wireless sensor network* atau WSN atau jaringan sensor nirkabel, merupakan salah satu bagian dari jaringan nirkabel yang merupakan sekumpulan dari beberapa *devices* berukuran kecil, dengan berat yang ringan, dan termasuk dalam *low-cost networks* [2]. WSN atau jaringan sensor nirkabel beroperasi dengan menggunakan sumber daya terbatas dan memiliki kemampuan *sensing*, *computation*, dan komunikasi nirkabel [3]. *Node* dari sebuah jaringan sensor nirkabel memiliki tujuan yang sama seperti mendeteksi lingkungan atau suatu kejadian pada sebuah lingkungan. Contohnya pada skenario industri, *emergency response*, *traffic monitoring*, bidang kesehatan dan medis, dan dapat digunakan untuk mendeteksi ledakan [4].

*Network lifetime* didefinisikan sebagai jumlah dari *data aggregation* hingga  $\alpha$  persen dari total sensor *node* mati [5]. Dalam beberapa kasus pada jaringan sensor nirkabel yang sangat mengutamakan lama waktu kemampuan operasi pada salah satu *node* sensor, *lifetime* didefinisikan sebagai waktu *node* sensor yang pertama mengalami kehabisan energi untuk beroperasi. *Network lifetime* sangat menjadi aspek yang diperhitungkan karena energi yang tersedia pada *node* di jaringan sensor nirkabel sangat terbatas dan diperlukannya efisiensi energi. Sehingga efisiensi energi dan *network lifetime* memiliki sinonim untuk mengoptimalkan penggunaan efisiensi energi yang dapat memperpanjang durasi dari *network lifetime*.

Untuk mengurangi jumlah paket data yang dikirim pada sebuah jaringan terdapat sebuah penelitian yang mengusulkan metode untuk mengumpulkan semua data sensor. Kumpulan dari berbagai data sensor tersebut selanjutnya diproses dijadikan sebuah data baru yang mewakili semua data yang sebelumnya telah dikumpulkan. Metode ini disebut dengan *Data-Aggregation* [6]. Metode ini bertujuan untuk melakukan efisiensi energi dengan *latency* data yang seminimal mungkin. *Latency* data merupakan hal yang penting dalam pemantauan lingkungan karena menentukan kecepatan respon dari sistem tersebut.

*Data-Aggregation* pada sebuah jaringan disebut dengan istilah *in-network data aggregation* yang merupakan sebuah metode agregasi yang dilakukan secara menyeluruh pada pengumpulan data *routing* melalui jaringan *multi-hop*, mengolah data pada *intermediate node* dengan tujuan meningkatkan *network lifetime* [7]. *In-network data aggregation* ini dibagi

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

Ketika menggunakan metode Data-Aggregation, penggabungan data biasanya terjadi pada sebuah *cluster head* yang merupakan *node* koordinator yang bertugas untuk menentukan tingkat agregasi dan sebagai wadah dimana agregasi dilakukan. Teknik agregasi yang dilakukan juga telah dikembangkan dengan mempertimbangkan faktor kondisi dari lingkungan yang diawasi dengan mengatur jumlah data yang diaggregasi sehingga secara tidak langsung dapat mengatur kepadatan aliran data pada jaringan [8]. Teknik dan metode agregasi juga telah dikembangkan lebih jauh dengan menerapkannya tidak hanya pada *cluster head* namun juga pada level individual *node* yang dilakukan secara adaptive terinspirasi dari pertahanan tubuh ketika terjadi kerusakan *cell* [9].

Namun dari semua penelitian yang telah disampaikan, agregasi masi tetap dibebankan pada *cluster head* yang mengakibatkan tingginya beban kerja yang harus dijalankan oleh *cluster head*. Tingginya beban kerja berbanding lurus dengan jumlah konsumsi energi yang diperlukan sehingga *cluster head* cenderung akan mengalami *drop-state* yang lebih cepat dibandingkan *node* yang lain. Selain hal tersebut, *cluster head* cenderung dipilih diawal ketika jaringan baru dibentuk dan ditetapkan secara permanen (tidak tergantikan oleh *node* yang lain). Apabila *cluster head* mengalami *drop-state* maka wilayah *cluster* tersebut akan terisolir.

Untuk menghindari kondisi *drop-state* yang terlalu dini, diperlukan algoritma pemilihan *cluster head* secara dinamis. Kondisi *cluster head* dapat dilihat dari jumlah energi yang tersisa pada *cluster head* tersebut. Jika berada dalam kondisi rendah maka *node member* pada *cluster* tersebut akan melakukan evaluasi kepada *node* tetangga atau *node*-nya sendiri yang dapat dipromosikan sebagai *cluster head*. Dari hasil penelitian ini, diharapkan dapat memberikan sebuah pengembangan dari metode yang sudah ada pada bidang *wireless sensor network* khususnya untuk menangani masalah *clustering* dan data-aggregation.

## 2. Pembahasan

### 2.1. Tahapan Penelitian

Penelitian ini memiliki beberapa tahapan-tahapan yang perlu diperhitungkan untuk membentuk sebuah algoritma yang bisa diterapkan dalam penentuan *cluster head* secara dinamis pada jaringan sensor nirkabel. Adapun hal yang perlu dipersiapkan dalam perancangan adalah sebagai berikut:

- a. Penentuan indikator penentuan *cluster head*

Indikator penilaian kelayakan sebuah *node* untuk dipilih menjadi *cluster head* ini diperlukan untuk memastikan bahwa sebuah *node* yang dipilih menjadi *cluster head* layak dipilih dan merupakan *node* dengan kondisi terbaik dari *node* yang tersedia.

- b. Penyebaran informasi indikator

Nilai dari indikator yang telah ditentukan didapatkan dari informasi yang diberikan oleh *node* tetangga.

- c. Penentuan proses pemilihan *cluster head*

Setelah informasi indikator diterima oleh sebuah *node* maka selanjutnya informasi tersebut diproses untuk mendapatkan *node* yang layak dipilih menjadi *cluster head*.

### 2.2. Perancangan Indikator Penentuan *Cluster Head*

Penentuan ini memerlukan informasi mengenai kondisi masing-masing dari *node* tetangga sehingga *node* dapat menghitung rata-rata energi yang tersisa pada *node* tetangga. Penghitungan rata-rata energi yang tersisa dihitung menggunakan Persamaan (1) di bawah ini:

$$AE = \frac{1}{n} \times \sum_{i=1}^n e_i \quad \dots(1)$$

*AE* merupakan *Average Energy* yang merupakan hasil dari rata-rata informasi energi yang tersisa, *n* merupakan jumlah *node* yang mengirimkan informasi energi yang tersisa dan *e* merupakan energi yang tersisa dari masing-masing *node* yang dihitung menggunakan skala 100 dengan nilai minimum 0 dan maksimum 100.

Setelah menentukan rata-rata energi yang tersisa pada sebuah *cluster* maka selanjutnya *node* akan memberikan penilaian kepada masing-masing *node* untuk menentukan *node* yang layak dipilih menjadi *cluster head*. Penentuan ini menggunakan faktor dukungan tambahan selain energi yang tersisa yaitu kedekatan antara kordinat *sink/central node* dan jumlah *node* tetangga yang dimiliki. Tabel 1 merupakan bobot penilaian yang digunakan dalam penentuan *cluster head*.

**Tabel 1. Bobot Penentuan *Cluster Head***

Indikator	Bobot (%)
Energy Level (EL)	50
Sink Distance (SD)	30
Neighbors Count (NC)	20

Faktor *Energy Level* merupakan faktor penentu utama pada algoritma ini. Faktor ini menggambarkan sisa energi yang tersisa pada sebuah *node* (skala 0 hingga 100) yang kemudian dihitung selisih dengan nilai *AE* dimana semakin tinggi nilai dari *energy level* maka semakin besar pilihan *cluster head* akan ditentukan pada *node* tersebut.

Faktor pendukung *sink distance* merupakan perhitungan antara jarak dari *node* menuju *sink node*. Ini diperlukan untuk menghindari aliran data yang justru menjauh dari *sink node* yang mengakibatkan tingginya jumlah lompatan data yang diperlukan yang secara

langsung akan menurunkan performa jaringan dan meningkatkan konsumsi energi yang diperlukan karena setiap lompatan atau *forwarding* data akan memerlukan energi untuk melakukan transmisi data. *Sink distance* dihitung dengan Persamaan (2).

Faktor pendukung *neighbors count* diperlukan untuk memastikan bahwa *node cluster head* yang dipilih memiliki pilihan jalur *forwarding* yang banyak. Ini diperlukan untuk menghindari *forwarding* data yang dilakukan menuju *node* buntu atau tidak memiliki tentangga. *Node* buntu ini akan menyebabkan terjadinya kondisi *ping-pong* dimana data akan terjebak pada kedua *node* tersebut.

### 2.3. Perancangan Penyebaran Informasi Indikator

Penyebaran informasi indikator ini dilakukan secara periodik oleh masing-masing *node* ke *node* yang bertetangga atau *neighbors node*. Jumlah lompatan *node* yang dilakukan adalah satu *hop*. Data dikirimkan dalam bentuk frame yang disesuaikan. Tabel 2 merupakan rancangan data yang akan dikirimkan dalam penyebaran informasi indikator.

**Tabel 2. Data Informasi Indikator**

Data	Type Data
Heartbeat ID	Integer
Node ID	Integer
Energy Level (e)	Floating
Location Coordinate (a,b)	Array Floating
Neighbors Count (n)	Integer

*Heartbeat ID* merupakan nomor data informasi yang dikirimkan oleh *node*. Setiap informasi yang dikirimkan akan memiliki nomor yang bersifat *increment* sesuai dengan jumlah data informasi indikator yang telah dikirimkan sebelumnya. Tipe data yang digunakan adalah tipe data bilangan bulat (*integer*).

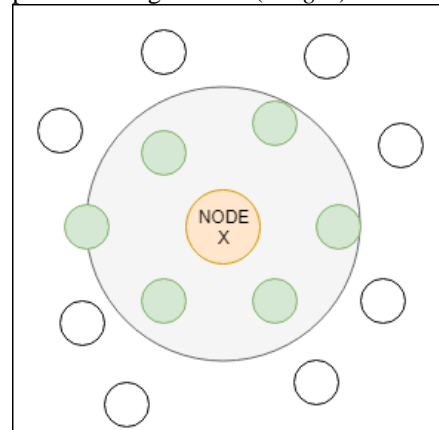
*Node ID* merupakan nomor *node* yang mengirimkan informasi tersebut. Ini berguna untuk mengetahui asal dari sebuah informasi indikator yang diterima. Tipe data yang digunakan adalah tipe data bilangan bulat (*integer*).

*Energy level* merupakan nilai persentase dari sisa energi yang tersisa dari sebuah *node*. *Energy level* didapatkan dari nilai selisih total penggunaan energi pada sebuah *node* dengan kapasitas penyimpanan baterai. Nilai kemudian di konversi dalam bentuk persentase. Tipe data yang digunakan adalah tipe data bilangan decimal pecahan (*floating*).

*Location coordinate* merupakan nilai kordinat lokasi dari sebuah *node* yang didapatkan dari sensor GPS yang terdapat pada setiap *node*. Terdapat dua nilai yang dikirimkan yaitu nilai kordinat lintang dan kordinat bujur. Tipe data yang digunakan adalah tipe data bilangan decimal pecahan (*floating*) yang disimpan dalam *array* dua dimensi.

*Neighbors count* merupakan jumlah tetangga yang berjarak satu lompatan/*hop* dari sebuah *node*. Nilai ini didapatkan ketika *node* melakukan inisialisasi

pertama kali yang selanjutnya akan diperbarui seiring pesan indikator yang diterima. Tipe data yang digunakan adalah tipe data bilangan bulat (*integer*).



**Gambar 1.** Cakupan wilayah penyebaran informasi indikator

Informasi indikator ini dikirimkan secara periodik setiap 3600 detik untuk memberikan informasi indikator terbaru kepada *node* tetangga. Ketika sebuah *node* menerima data informasi indikator, *node* akan menyimpan informasi tersebut beserta dengan *heartbeat ID* yang dikirimkan. Sebelum menyimpan sebuah informasi indikator, *node* terlebih dahulu melakukan komparasi antara *heartbeat ID* dari masing-masing *node* yang telah disimpan sebelumnya dengan *heartbeat ID* dengan data informasi indikator yang diterima. Jika *heartbeat ID* yang diterima lebih besar dari *heartbeat ID* yang telah disimpan, maka informasi indikator akan diperbarui. Gambar 1 merupakan cakupan wilayah dari penyebaran informasi indikator dari sebuah *node*.

**Tabel 3. Penyimpanan Data Informasi Indikator**

Node ID	Data	Type Data
1	Node ID	Integer
	Energy Level	Floating
	Location Coordinate	Array Floating
	Neighbors Count	Integer
	..	.....
n	Node ID	Integer
	Energy Level	Floating
	Location Coordinate	Array Floating
	Neighbors Count	Integer

Pada Gambar 1, lingkaran merupakan semua *node* yang tersedia, yang berwarna oranye merupakan *node x* yang mengirimkan informasi indikator. Cakupan pengiriman *node x* digambarkan dengan radius abu-abu. *Node* yang menerima informasi indikator adalah *node* yang berwarna hijau. Berdasarkan Gambar 1, kita juga bisa melihat bahwa terdapat enam *node* yang merupakan tetangga dari *node x* yang akan menjadi nilai dari

*neighbors count* dan mengirim informasi indikator keseluruhan *node* tetangga.

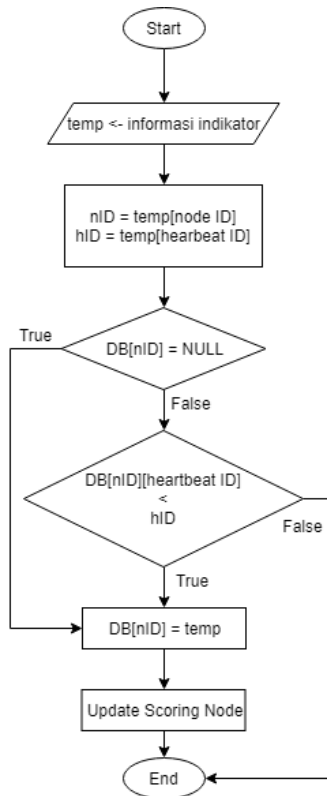
Data yang disimpan pada *node* disimpan dalam bentuk *array* dua dimensi. Penyimpanan informasi indikator disimpan dalam struktur pada Tabel 3. Pada tabel tersebut *n* merupakan *ID* dari masing-masing *node* yang mengirimkan dengan nilai tertinggi.

**2.4. Perancangan Proses Pemilihan Cluster Head**

Pada proses pemilihan *cluster head*, *node* melakukan perhitungan secara mandiri untuk menentukan *cluster head* yang akan dipilih. Terdapat dua proses pada pemilihan *cluster head*. Yang pertama ada proses untuk mengumpulkan informasi indikator, menghitung bobot penilaian masing-masing *node* berdasarkan indikator yang telah diberikan, dan yang terakhir adalah proses penentuan *cluster head*.

**2.4.1. Proses pengumpulan informasi indikator**

Proses pengumpulan informasi indikator dilakukan secara pasif, menunggu sebuah informasi indikator dikirimkan oleh *node* tetangga dan ketika diterima maka baru dilakukan proses penyimpanan. Gambar 2 merupakan diagram alir dari proses pengumpulan informasi indikator.



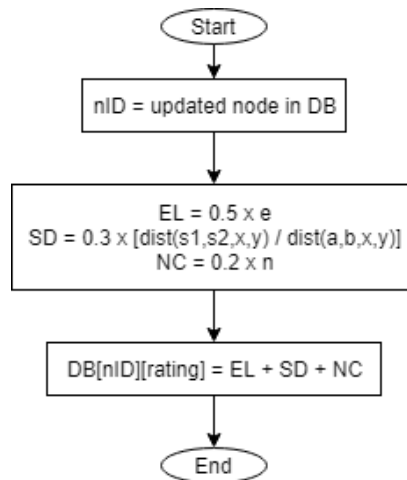
**Gambar 2.** Diagram alir proses penyimpanan informasi indikator

Proses pada Gambar 2 diawali dengan diterimanya sebuah informasi indikator pada sebuah *node*. Ketika sebuah informasi tersebut diterima, maka selanjutnya *node* akan menyimpan sementara informasi tersebut pada variabel *temp*. Selanjutnya dilakukan

ekstraksi nilai dari informasi indikator yang telah diterima dari variabel *temp* yaitu informasi *node ID* dan *heartbeat ID*. Kemudian *node* akan melakukan pemeriksaan apakah *node ID* tersebut pernah disimpan sebelumnya pada variabel *DB*. Jika belum pernah (*DB* dengan *nID* tersebut bernilai *null*) maka *DB* dengan *nID* tersebut akan langsung diperbarui dengan menyimpan nilai *temp* pada variabel *DB* dengan *nID* tersebut. Jika ternyata sudah pernah disimpan (tidak bernilai *null*) maka akan dilakukan pemeriksaan apakah *heartbeat ID* yang disimpan pada *DB* memiliki nilai yang lebih rendah dari nilai *heartbeat ID* yang diterima. Jika lebih rendah, maka itu menandakan bahwa informasi indikator yang diterima lebih baru dibandingkan dengan informasi yang telah disimpan sebelumnya. Sehingga berdasarkan hal tersebut *node* akan menyimpan nilai informasi indikator yang terbaru.

**2.4.2. Proses penghitungan penilaian indikator**

Masing-masing bobot indikator dari masing-masing *node* akan dihitung selanjutnya setelah terdapat pembaruan data pada database informasi indikator. Proses yang terjadi ini merupakan penjelasan lebih rinci mengenai proses *update scoring node* pada Gambar 2. Gambar 3 merupakan proses yang terjadi pada *update scoring node*.

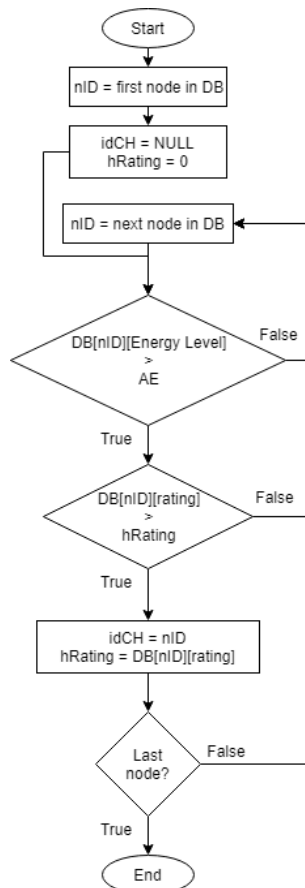


**Gambar 3.** Diagram alir proses perhitungan penilaian rating node

Proses pada Gambar 3 merupakan tahapan detail dari proses *update scoring node* pada Gambar 2. Pada tahap ini nilai *energy level* (*EL*), *sink distance* (*SD*), dan *neighbors count* (*NC*) dihitung dari informasi indikator yang didapatkan dan penilaiannya telah disesuaikan dengan bobot yang telah ditentukan pada Tabel 1. Penghitungan jarak antar dua kordinat menggunakan fungsi *Distance* (*dist*) dengan paramater *x1*, *y1*, *x2*, *y2*. Jarak yang dibandingkan adalah dengan ratio antara jarak *node* bersangkutan dengan *node* tetangga yang diujikan. Setelah semua indikator dihitung sesuai bobot kemudian dijumlahkan dan disimpan dalam *database* (variabel *DB*) sesuai dengan *node* yang diproses saat itu

### 2.4.3. Proses penentuan *cluster head*

Proses penentuan *cluster head* dilakukan secara periodik oleh setiap *node*. Proses penentuan ini ditentukan secara konstan setiap 1800 detik. Pada proses penentuan, *node* akan mencari nilai *rating* tertinggi dengan energi yang nilainya diatas rata-rata. Gambar 4 merupakan proses yang terjadi ketika memilih *node* sebagai *cluster head*.



**Gambar 4.** Diagram alir proses pemilihan *cluster head* berdasarkan *rating*

Proses pada Gambar 4 merupakan skema pemilihan *cluster head*. Terdapat variabel *idCH* yang merupakan *node id* dari *cluster head* dan *hRating* yang merupakan nilai *rating* yang dimiliki oleh *cluster head* yang terpilih. Selanjutnya dilakukan perbandingan antara nilai *energy level* yang dimiliki *node* tersebut dengan nilai rata-rata dari seluruh *node* pada *cluster* tersebut. Jika *energy level* pada *node* tersebut diatas rata-rata maka baru kemudian diperiksa apakah memiliki *rating* yang lebih tinggi dari *cluster head* yang sebelumnya. Ketika nilai *rating*nya lebih tinggi maka *cluster head* akan diubah menjadi *cluster head* yang baru. Hal ini terus terjadi berulang-ulang dengan melakukan perbandingan ke seluruh *node* yang terdapat dalam *database* hingga semua *node* selesai dibandingkan. Hal ini memastikan bahwa *cluster head* yang dipilih merupakan pilihan terbaik. Proses ini juga dilakukan

secara berkala setiap 1800 detik sehingga memastikan bahwa *cluster head* yang dipilih dimungkinkan untuk bersifat dinamis sesuai dengan kondisi jaringan maupun kondisi rata-rata nilai *energy (AE)* pada *cluster* tersebut.

### 3. Kesimpulan

Perancangan algoritma pemilihan *cluster head* ini memberikan fleksibilitas dalam pemilihan *cluster head* pada sebuah jaringan sensor nirkabel yang dibagi/di-*cluster* dalam beberapa bagian. Masing-masing *cluster* bekerja dengan asumsi telah diterapkan dengan metode *in-network data aggregation*. *Data aggregation* pada jaringan sensor nirkabel memerlukan sebuah metode penentuan *cluster head* (yang bertugas sebagai *aggregator*) yang bersifat dinamis sesuai dengan kondisi *cluster* tersebut. Rancangan metode yang telah disampaikan ini diharapkan mampu mengurangi kemungkinan terisolirnya sebuah *cluster* dikarenakan *cluster head* yang merupakan gerbang data menuju *sink node* mengalami *drop-state* dan kehabisan energi. Dengan kemampuannya untuk memilih *cluster head* secara dinamis, maka ketika *cluster head* mengalami *drop-state*, anggota *cluster* mampu memilih *cluster head* yang baru.

### Daftar Pustaka

- [1] B. Mehta and Y. J. Reddy, "Wireless communication," in *Industrial Process Automation Systems*, Oxford, Elsevier's Science & Technology, 2015, pp. 417-457.
- [2] R. S. Carbajo, E. S. Carbajo, B. Basu and C. Mc Goldrick, "Routing in wireless sensor networks for wind turbine monitoring," *Pervasive and Mobile Computing*, vol. 39, pp. 1-35, 2017.
- [3] M. Elshrkawey, S. M. Elsherif and M. Elsayed Wahed, "An Enhancement Approach for Reducing the Energy Consumption in Wireless Sensor Networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 2, pp. 259-267, 2018.
- [4] A. J. AL-Mousawi and H. K. AL-Hassani, "A survey in wireless sensor network for explosives detection," *Computers & Electrical Engineering*, vol. 39, pp. 1-35, 2017.
- [5] M. R. and R. Datta, "A novel source location privacy preservation technique to achieve enhanced privacy and network lifetime in WSNs," *Pervasive and Mobile Computing*, vol. 44, pp. 58-73, 2018.
- [6] P. Padmaja and G. Marutheswar, "Energy efficient data aggregation in wireless sensor networks," Telangana, India, 2018.
- [7] C. Li, J. Bai, J. Gu, X. Yan and Y. Luo, "Clustering routing based on mixed integer programming for heterogeneous wireless sensor networks," *Ad Hoc Networks*, vol. 72, pp. 81-90, 2018.
- [8] I. G. N. A. Kusuma and W. Wibisono, "Adaptive Data Aggregation for Shortest Geopath Routing Protocol in Wireless Sensor Network," *IPTEK, The Journal for Technology and Science*, vol. 28, no. 2, pp. 51-54, 2017.
- [9] W. Wibisono, I. G. N. Ady Kusuma, Y. Ishida and I. Winarmo, "Towards an Immunity-Based Approach for Preserving Energy of Data-Gathering Processes in Wireless Sensor Network Environments," in *The 2016 International Conference on Advanced Informatics: Concepts, Theory and Application (ICAICTA2016)*, Penang, 2016.