

OPTIMISASI PENEMPATAN NODESENSOR PADA WIRELESS SENSOR NETWORK

Dzuriyat Muhammad Mukhtar¹), Fachrudin Hunaini²), Faqih Rofii³)

¹ Teknik Elektro, Universitas Widyagama, Malang
Email: dzuriyatmuhammad@gmail.com

² Teknik Elektro, Universitas Widyagama, Malang
Email : fachrudin_h@widyagama.ac.id

³ Teknik Elektro, Universitas Widyagama, Malang
Email : faqihromi@gmail.com

Abstrak

Wireless Sensor Network (WSN) merupakan salah satu teknologi komunikasi yang menggunakan prosesor sederhana, konsumsi daya rendah dan fleksibel. Namun padapenerapannya WSN diperlukan penempatan *nodesensor* yang optimal pada suatu area agar didapatkan *coverage area* yang terluas dan konsumsi daya yang terendah karena penyebaran *nodesensor* secara acak dapat menyebabkan adanya *blank spot* dan konsumsi daya yang tidak dijamin efisien untuk mencapai cakupan yang diperlukan. Oleh karena itu, dilakukan simulasi penempatan *nodesensor* dengan menggunakan *Firefly Algorithm* pada program aplikasi MATLAB dari kondisi penempatan lokasi yang acak hingga penempatan yang optimal. Hasil simulasi optimisasi penempatan *nodesensor* menggunakan *Firefly Algorithm* pada luas area 50x50m² dengan 30 *nodesensor* terjadi peningkatan *coverage area* dari 33.92% menjadi 59.92% dan penurunan konsumsi daya sebesar 48.24%.

Kata kunci: WSN, *Nodesensor*, *Firefly Algorithm*, *Coverage area*, Daya

Abstract

Wireless Sensor Network (WSN) is one communication technology that uses a simple processor, low power consumption and flexible. However, the WSN application requires optimal sensor node placement in an area to obtain the widest coverage area and lowest power consumption because random sensor node deployment can cause blank spots and power consumption that are not guaranteed to be efficient to achieve the required coverage. Therefore, simulating the placement of sensor nodes is done using the Firefly Algorithm in the MATLAB application program from the condition of random location placement to optimal placement. The results of optimization of sensor node placement using Firefly Algorithm in an area of 50x50m² with 30 sensor nodes increased coverage area from 33.92% to 59.92% and decreased power consumption by 48.24%.

Keywords: WSN, *Nodesensor*, *Firefly Algorithm*, *Coverage area*, Power.

PENDAHULUAN

Teknologi komunikasi yang telah banyak mendukung untuk pembuatan sebuah jaringan *sensor* nirkabel atau *Wireless Sensor Network (WSN)* adalah modul Xbee yaitu suatu modul komunikasi nirkabel yang menggunakan frekuensi radio. Hal ini adalah salah satu keunggulan modul Xbee untuk

memperbaiki kekurangan yang ada pada jaringan *sensor* dengan kabel pada sistem yang luas (Firnandes, dkk., 2013), dimana *node* yang digunakan adalah '*smart sensor*', yaitu sebuah prosesor kecil yang dilengkapi dengan fungsi penginderaan (termal, tekanan, dan sebagainya) dan *wireless transceiver*. Dalam tipe jaringan ini, pertukaran informasi *sensor* pada lingkungan ditujukan dalam rangka membangun cakupan wilayah yang dapat dipantau secara global, yang dapat diakses oleh pengguna melalui satu atau lebih *node gateway*. Peletak titik-titik *nodesensor* tidak perlu direkayasa sedemikian rupa atau ditetapkan sebelumnya (fixed). Hal ini untuk menampilkan kondisi titik-titik yang tersebar secara acak sampai pada suatu daerah yang tidak dapat diakses (Rijal, dkk., 2011).

Isu utama yang menjadi pokok perancangan WSN adalah keterbatasan daya dan masalah *Coverage Area* (Area Cakupan), yang mencerminkan kualitas layanan (Qos) yang disediakan oleh penyedia tertentu pada masing-masing *nodesensor* (Buettner, dkk., 2006). Salah satu usaha yang sangat mungkin dikembangkan adalah membangun metode optimisasi. Dinamika algoritma optimisasi yang berkembang salah satunya adalah *Firefly Algorithm* yang dikembangkan pertama kali oleh Yang mulai akhir tahun 2007 hingga tahun 2008 di Universitas Cambridge (Setiawan, dkk., 2015). Algoritma kunang-kunang terinspirasi oleh perilaku dan pola berkedip kunang-kunang. Kunang-kunang yang lebih tidak terang akan bergerak menuju kunang-kunang yang lebih terang. Jika tidak ada yang lebih terang, kunang-kunang akan bergerak secara acak. Tingkat keterangan kunang-kunang ditentukan oleh kondisi dari fungsi objektif (Lukasik, dkk., 2009).

Oleh karena itu, pada paper ini dilakukan perencanaan dan simulasi penempatan *nodesensor* menggunakan *Firefly Algorithm* dengan software MATLAB sehingga penempatan *nodesensor* akan mendapatkan penggunaan daya yang minimum yang berkorelasi kuat terhadap jarak rata-rata antar *nodesensor* dan mendapatkan nilai *Coverage Area* yang terluas.

METODE PENELITIAN

Metode penelitian pada paper ini dilakukan terdiri dari beberapa tahapan meliputi merancang antarmuka aplikasi, penentuan jumlah *nodesensor*, penentuan luas area, radius masing-masing *nodesensor*, perancangan total *coverage*, total daya yang digunakan oleh *nodesensor*, proses optimisasi *Coverage Area* dan konsumsi daya yang digunakan oleh *nodesensor* menggunakan *Firefly Algorithm* (Algoritma

Firefly). Hasil optimisasi ditampilkan dalam bentuk Visualiasi dan data tabel yang ada dalam Aplikasi.

1. Merancang Antarmuka Aplikasi

Perancangan aplikasi dilakukan menggunakan Software MATLAB, yang berbasis GUI (*Graphical User Interface*). Antarmuka aplikasi ini terdiri dari beberapa panel yang digunakan untuk memperlihatkan lokasi *nodesensor*, juga terdapat panel input dan tombol pengoperasian untuk input parameter dan menjalankan aplikasi, serta tabel untuk memperlihatkan nilai yang didapatkan setelah optimisasi.

2. Menentukan Parameter optimisasi

2.1. Luas Area *NodeSensor*

Luas area dalam program ini sebesar 50 m x 50 m berbentuk kubus terbentuk didalam sebuah panel yang berisi *plot figure* dalam Software MATLAB dalam satuan *grid* dengan inisial (L) pada program.

2.2. Lokasi *NodeSensor*

Dalam software MATLAB banyaknya *NodeSensor* memiliki Variable (n) pada program yang dimasukkan melalui *panel input* oleh *user*, penentuan lokasi *nodesensor* menggunakan persamaan (1) dengan hasil 2 kolom yang berisi angka / koordinat sebagai lokasi awal *nodesensor* (Erwin, dkk., 2010).

$$po = (L - 0) * rand(2, n) - 0 \quad (1)$$

2.3. Radius *NodeSensor*

Radius *nodesensor* dalam program ini memiliki nilai yang sama yaitu 5m dengan variable (Rs) dengan bentuk *visual* lingkaran sempurna berwarna biru (Rahayu B.S, 2011).

2.4. Menentukan Parameter *Firefly Algorithm*

Penentuan parameter *Firefly Algorithm* berupa nilai *Alpha*, *Gamma*, *Delta*, dan *Beta*.digunakan untuk menjalankan optimisasi guna mendapatkan hasil yang terbaik (Yang, X.S, 2009).

3. Menghitung Nilai *Objective Function*

Nilai *objective function* dalam paper ini digunakan untuk proses perpindahan *nodesensor* mencari lokasi yang terbaik, dimana nilai yang dicari adalah nilai yang

terkecil. Proses penentuan nilai *objective function* didapatkan dari *fitness coverage* dan *fitness daya* (Lukasik, dkk., 2009).

3.1. Menghitung Total Coverage

Untuk mengetahui berapa luas *Coverage Area* pertama harus mengetahui luas total area yang digunakan, kemudian menentukan jarak antar *node* yang digunakan untuk menentukan *Coverage Area* dengan rumus *Cartesian distance* pada persamaan (2) (Rahayu B.S, 2011):

$$d(i,j) = \sqrt{(x_{i,k} - x_{j,k})^2 + (y_{i,k} - y_{j,k})^2} \quad (2)$$

d adalah jarak antara masing-masing *nodesensor*, setelah mendapatkan nilai (d) dapat menentukan nilai area yang *coverage* dengan menggunakan persamaan 3.

$$cov(i,j) = \begin{cases} 1, & d(i,j) \leq R_s \\ 0, & else \end{cases} \quad (3)$$

Cov (i,j) merupakan luas area yang tercakup oleh *nodesensor* bernilai biner [0,1] dalam bentuk matriks.

3.2. Menghitung Besar Daya NodeSensor

Setelah menentukan lokasi awal secara *random*, lokasi *nodesensor* harus berpindah agar dapat mencapai *Coverage Area* yang luas dan konsumsi daya minimum. Konsumsi daya *nodesensor* dinyatakan sebagai rata-rata jarak antar *nodesensor* dalam jaringan. Dengan daya yang dihitung merupakan besar jarak antar *nodesensor*, maka besar daya yang dibutuhkan dapat ditentukan dengan persamaan 4 (Anastasi, dkk., 2009).

$$d_{total} = \sqrt{\frac{\sum_{i=1}^n d_i^2}{n}} \quad (4)$$

Dimana d_i^2 merupakan jarak antar *nodesensor* dengan yang lainnya dan d_{total} adalah besar daya yang merupakan rata-rata jarak *nodesensor* dimana semakin jauh jaraknya maka semakin besar daya yang digunakan.

3.3. Menghitung Objective Function Sebagai Nilai Fitness

Objective function didapatkan dari *fitness function* yang digunakan untuk meminimalkan *coverage point* dan meminimalkan penggunaan daya (Anastasi, dkk., 2009):

$$F_{cov} = 1 - \frac{F}{L \times L} \quad (5)$$

Sedangkan untuk minimumsi daya menggunakan persamaan 6:

$$F_{\text{daya}} = \frac{d_{\text{total}}}{2R_s}$$

(6)

Dimana (Rs) merupakan jari-jari dari *coveragenodesensor*. Setelah semua *fitness function* ditentukan maka dapat ditentukan *objective function*.

$$Fitness_{\text{firefly}} = F_{\text{cov}} + 1 * F_{\text{daya}}$$

(7)

4. Mencari Lokasi *NodeSensor* Terbaik Menggunakan *Firefly Algorithm*

Tahap optimisasi dilakukan dengan menggunakan *Firefly Algorithm* untuk mendapatkan nilai *coverage* yang luas dan daya yang minimum, nilai yang optimal didapatkan jika *nodesensor* dapat berpindah mencari lokasi terbaik dengan nilai *coverage* yang luas dan daya yang minimal dengan melalui beberapa langkah:

4.1. Menentukan Nilai Intensitas Cahaya

Pada *Firefly Algorithm* proses optimisasi dipengaruhi oleh tingkat intensitas cahaya kunang-kunang, besar tidaknya nilai intensitas cahaya dipengaruhi oleh *objective function* yang di dapatkan dari F_{cov} (*Fitness Coverage*) dan F_{daya} (*Fitness Daya*).

4.2. Mencari Lokasi Berdasarkan Intensitas Cahaya

Dalam *Firefly Algorithm* (Algoritma Kunang-kunang) cara kunang-kunang satu dengan lainnya mencari lokasi terbaik ditentukan oleh terang tidaknya intensitas cahaya masing-masing kunang-kunang.

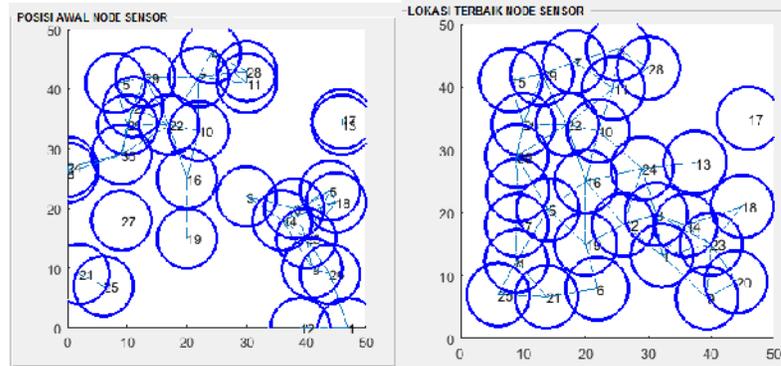
Oleh karena itu, untuk sepasang kunang-kunang yang sedang berkedip, kunang-kunang yang lebih redup bergerak menuju kunang-kunang yang lebih terang, Jika tidak ada kunang-kunang yang lebih terang, kunang-kunang akan bergerak secara acak. Tingkat keterangan kunang-kunang ditentukan oleh kondisi dari *Objective Function*.

HASIL DAN PEMBAHASAN

Hasil simulasi dikatakan terbaik jika nilai total *coverage area* dapat diperbesar sehingga dapat memperbesar *Coverage Area* dan meminimalkan konsumsi daya *nodesensor* dalam melakukan komunikasi yang dalam hal ini dinyatakan dalam rata-rata jarak antar *node*. Hasil optimisasi *coverage* dan daya didalam aplikasi berupa *visual* yang ada pada panel **Lokasi Terbaik *NodeSensor*** dan berupa data tabel yang ada didalam aplikasi .

1. Lokasi Terbaik

Hasil terbaik didapatkan jika *nodesensor* dapat berpindah dan menemukan lokasi yang tepat dengan nilai *coverage* yang luas dan konsumsi daya yang minimum, hasil terbaik dapat dilihat pada panel **Lokasi Terbaik NodeSensor**.



Gambar 1. Lokasi Awal NodeSensor Gambar 2. Lokasi Terbaik NodeSensor

Tabel 1. Hasil Percobaan *Nodesensor*

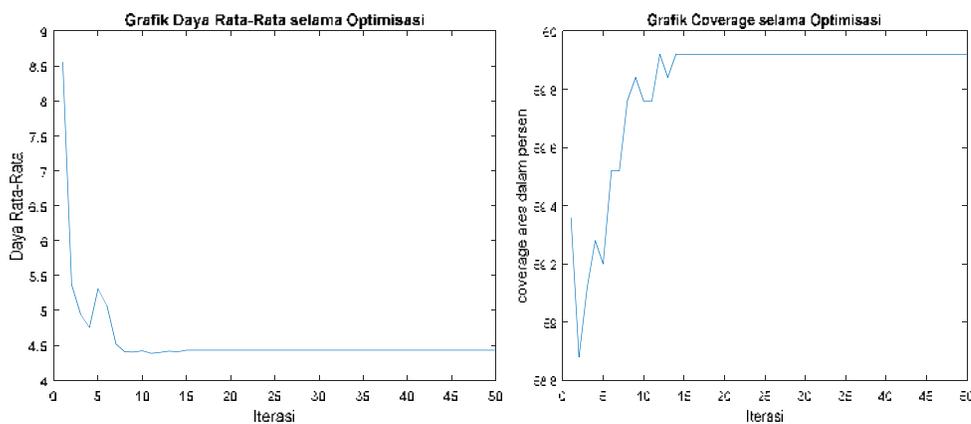
Jumlah <i>NodeSensor</i> (unit)	30
Alpha	0,2
Gamma	1
Delta	0,98
Jumlah Percobaan (iterasi)	50
Luas Area (m ²)	2500
<i>Coverage</i> Sebelum optimisasi (%)	33,92 %
<i>Coverage</i> Setelah optimisasi (%)	59,92 %
Daya Rata-rata Sebelum optimisasi berkorelasi terhadap jarak rata-rata antar <i>nodesensor</i> (m) sebelum optimisasi	8,5578
Daya Rata-rata Setelah optimisasi berkorelasi terhadap jarak rata-rata antar <i>nodesensor</i> (m) setelah optimisasi	4,4295
Selisih Daya Rata-rata berkorelasi terhadap selisih jarak rata-rata antar <i>nodesensor</i> (m)	4,1283
Penurunan Daya Rata-rata (%)	48,24%

Pada gambar 2 menunjukkan adanya perubahan lokasi *nodesensor* dengan *Coverage Area* awal sebesar 33,92 % menjadi 59,92%, sedangkan daya rata-rata yang semula (dinyatakan dalam jarak rata-rata antar *nodesensor*) sebesar 8,5578 m menjadi 4,4295 m, Luas *Coverage Area* meningkat sebesar 26 % dan konsumsi daya rata-ratanya menurun setara dengan 4,1283 matau terjadi penurunan daya rata-rata sebesar 48,24% pada iterasi ke-11, namun dengan *coverage* yang lebih kecil dari iterasi sebelumnya sehingga nilai yang terbaik diambil ketika proses optimisasi pada *coverage* mencapai nilai yang konvergen pada iterasi ke-14 dengan

nilaicoverage sebesar 59,92% dan jarak rata-ratanya sebesar 4,4295. Jarak rata-rata merupakan fungsi dari daya rata-rata antar *nodesensor*.

2. Data Percobaan Dalam Grafik

Konsumsi daya *nodesensor* pada percobaan 30 *nodedengan* 50 iterasi, *sensor* mengalami perubahan atau penurunan setelah optimisasi. Daya rata-rata setelah optimisasi mengalami penurunan dan menjadi konvergen pada percobaan ke-11, dari awalnya sebesar 8,5578 m menjadi 4,4295 m yang didapatkan dari jarak rata-rata *nodesensor*.



Gambar 3. Grafik Daya node sensor Gambar 4. Grafik Coverage area

Grafik pada gambar 4 menunjukkan adanya peningkatan *coverage* dimana yang awalnya memiliki 33,92 % *coverage* menjadi 59,92% *coverage* dan menjadi konvergen pada percobaan ke 16.

KESIMPULAN

Kesimpulan yang dapat diambil dari simulasi optimisasi penempatan *nodesensors* pada jaringan *sensor nirkabel* adalah:

1. Simulasi optimisasi *Coverage Area* dan konsumsi daya dengan *Firefly Algorithm* menggunakan *software* MATLAB dilakukan dengan luas area sebesar 50x50 m, penentuan *coverage area* yang luas dan daya yang minimum diperoleh dari nilai *fitness coverage area* dan daya yang dipengaruhi oleh jarak *nodesensor*. Nilai *fitness* digunakan sebagai *objective function* untuk menggerakkan *nodesensor* menuju lokasi yang terbaik. Penentuan parameter optimisasi dilakukan dengan memasukkan nilai *alpha* sebesar 0,2, nilai *delta* sebesar 0,98, nilai *gamma* sebesar 1, dengan jumlah percobaan sebanyak 30 iterasi.

2. Optimisasi dilakukan memindahkan *nodesensor* dengan *Firefly Algorithm* sebanyak 30 iterasi dengan hasil *Coverage* awal dari 30 *nodesensor* adalah 33,92% konvergen pada percobaan ke-16 meningkat sebesar 26% menjadi *coverage area* sebesar 59,92%.
3. Besar konsumsi daya rata-rata ditentukan oleh jarak rata-rata antar *nodesensor*, pada percobaan 30 *nodesensor* jarak rata-rata awal sebesar 8,5578 setelah 30 iterasi menjadi konvergen pada percobaan ke-11 dengan jarak rata-rata sebesar 4,4295 menurun sebesar 4,1283 sehingga terjadi penurunan sebesar 48,24%.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada LPPM Universitas Widyagama Malang yang telah membantu penyelesaian paper ini. Kepada Direktorat Riset dan Pengabdian Masyarakat Kemenristek Dikti yang telah membiayai penelitian ini.

REFERENSI

- A. B. K. Rijal, P. Kristalina, dan T. B. Santoso, (2011). Simulasi Komunikasi Multihop pada jaringan *sensor* nirkabel menggunakan algoritma H-LEACH, *EEPIS Final Proj.*, 2011.
- G. Anastasi, M. Conti, M. Di Francesco, dan A. Passarella, (2009). *Energy conservation in wireless sensor networks: A survey*, *Ad Hoc Netw.*, vol. 7, no. 3, hal. 537–568.
- H. Setiawan, L. H. Hanafi, dan K. R. Prilianti, (2015). *Implementasi Algoritma Kunang-Kunang Untuk Penjadwalan Mata Kuliah di Universitas Ma Chung*, *J. Buana Inform.*, vol. 6, no. 4.
- I. M. Erwin, B. Sugiarto, dan I. Sakti, (2010). *Rancang bangun sistem monitoring kualitas udara menggunakan teknologi wireless sensor network (WSN)*, *INKOM J.*, vol. 3, no. 1–2, hal. 90–96.
- M. Buettner, G. V. Yee, E. Anderson, dan R. Han, (2006). X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks, *in Proceedings of the 4th international conference on Embedded networked sensor systems*, 2006, hal. 307–320.
- S. Lukasik dan S. Zak, (2009). Firefly Algorithm for Continuous Constrained optimization Tasks., *in ICCCI*, 2009, hal. 97–106.
- S. Bina Rahayu, (2011). Analisa Kinerja dan Simulasi *Coverage* Wireless Sensor Network dengan Sum of Weighted Cost Function Genetic Algorithm, *EEPIS Final Proj.*, 2011.
- T. Firnandes, S. K. Risandriya, dan K. Kamarudin, (2013). *Aplikasi Wireless Sensor Network (WSN) Berbasis Radio Frequency (RF) dan SMS Alert GSM*, *J. Integrasi*, vol. 5, no. 1, hal. 53–60.
- X.-S. Yang, (2009). Firefly algorithms for multimodal optimization,” in *International symposium on stochastic algorithms*, 2009, hal. 169–178.