

# Deployment Jaringan Sensor Nirkabel Berdasarkan Cakupan Area Sensor Node Menggunakan Algoritma Particle Swarm Optimization

Hani Rubiani  
Teknik Elektro  
Universitas Muhammadiyah Tasikmalaya  
Tasikmalaya, Indonesia  
hani.rubiani@umtas.ac.id

Sri Mulyatun  
Sistem Informasi  
STMIK AMIKOM  
Yogyakarta, Indonesia  
sri.m@amikom.ac.id

**Abstrak**—Deployment merupakan topik penting dalam bidang Wireless Sensor Network (WSN) karena mencerminkan biaya dan kemampuan deteksi WSN. Perangkat deployment adalah masalah mendasar dari pengadaan WSN. Jenis, jumlah dan lokasi perangkat menentukan beberapa sifat intrinsik WSN seperti cakupan area, konektivitas, dan konsumsi energi. Sebuah skema Deployment yang tepat dapat mengurangi kompleksitas masalah dalam WSN seperti perutean, fusi data, komunikasi, dll. Selain itu, dapat memperpanjang umur WSN dengan meminimalkan konsumsi energi. Pada penelitian ini, diajukan sebuah aplikasi sensor node deployment secara otomatis berdasarkan Algoritma Particle Swarm Optimization (PSO) dengan mempertimbangkan konsumsi energi sensor node. Secara khusus tujuan penelitian yang ingin dicapai adalah rancang bangun aplikasi untuk menempatkan sensor node pada lingkungan secara nyata sehingga kondisi optimal dari penempatan sensor node di suatu gedung tercapai. Luaran yang ditargetkan yaitu terciptanya suatu aplikasi sensor node deployment yang memperhatikan penyebaran node sensor dengan meminimalkan konsumsi energi sehingga dapat memperpanjang masa jaringan sensor nirkabel. Penelitian ini diharapkan dapat memudahkan dalam perancangan dan optimalisasi penempatan sensor node dalam lingkungan khususnya pada suatu gedung, serta dapat mendukung penelitian di bidang pervasive dan mobile computing.

**Kata kunci**— *Wireless Sensor Network (WSN); Deployment Sensor; Particle Swarm Optimization (PSO)*

## I. PENDAHULUAN

Dengan perkembangan komunikasi nirkabel dan mikro-elektronik, Wireless Sensor Network (WSN) menjadi teknologi yang menjanjikan dan mendapat perhatian penelitian yang signifikan dalam beberapa tahun terakhir [1]. Aplikasi jaringan sensor yang berbeda telah dibangun oleh berbagai perguruan tinggi, seperti pemantauan habitat oleh University of California di Berkeley dan College of Atlantic [2], Proyek Zebrantet untuk memantau kebiasaan hewan di Princeton University [3], dan jaringan sensor nirkabel untuk kegiatan pemantauan gunung berapi di Ekuador oleh Harvard University, University of New Hampshire, dan University of North Carolina [4].

Analisis kinerja seperti cakupan, konektivitas, konsumsi energi, waktu hidup dan biaya untuk jaringan sensor telah dipelajari oleh banyak peneliti. Daerah penelitian yang berbeda

terkait dengan WSN, seperti keamanan jaringan sensor, cakupan, komunikasi dan lain-lain. Secara khusus, penelitian ini berfokus pada masalah konsumsi energi yang menjadi perhatian utama bagi WSN.

Dalam sebagian besar aplikasi, masa sistem sangat penting untuk efektivitas. Konsumsi energi merupakan kendala utama untuk node jaringan sensor nirkabel karena sebagian besar adalah self-powered. Di antara bagian yang berbeda dari node jaringan individu, baterai adalah satu-satunya sumber daya untuk seluruh kehidupan. Beberapa aplikasi terbatas untuk komponen yang paling penting dan tidak memiliki pilihan untuk pengisian baterai. Semua tindakan node seperti penginderaan, berkomunikasi, komputasi mengkonsumsi energi. Jadi, sekali baterai atau sensor kehabisan daya, node sensor tidak bisa digunakan lagi. Hal ini dapat menurunkan kualitas layanan untuk seluruh jaringan sensor. Dalam beberapa penelitian mengenai deployment terdapat beberapa algoritma yang digunakan, diantaranya Genetic Algorithm (GA) dan algoritma Particle Swarm Optimization (PSO), penelitian ini menggunakan algoritma PSO untuk meminimalkan konsumsi energi sensor sehingga dapat memperpanjang masa hidup jaringan sensor dan menggunakan diagram Voronoi untuk memastikan sensor dengan kisaran adjustable dapat menutupi seluruh bidang penginderaan. Dengan demikian, masalahnya adalah bagaimana menggunakan energi dengan cara memperpanjang masa jaringan sensor dengan mempertimbangkan radius penginderaan yang dioptimalkan untuk menghemat energi.

## II. MODEL PENGINDERAAN

### A. Model Penginderaan Biner

Jika ada sensor simpul  $S$  di lokasi  $X_s, Y_s$ , asumsikan bahwa rentang penginderaan sensor  $S$  adalah area melingkar dengan radius  $RS$  dan berpusat di  $X_s, Y_s$ ,  $RS$  disebut radius penginderaan node  $S$ . Dalam model biner, sensor  $S$  mampu mendeteksi target dalam jangkauan penginderaan  $RS$  dengan probabilitas 1 dan tidak mampu mendeteksi setiap sasaran yang berada di luar jangkauan penginderaannya. Dengan demikian dalam model biner sensor yang dapat mendeteksi target dengan probabilitas 1 jika jarak antara area target dan sensor kurang dari jari-jari penginderaan  $RS$ . Namun, jika jarak antara target dan

sensor lebih jauh dari RS, sensor akan memiliki probabilitas nol dalam arti target tidak terdeteksi oleh sensor.

Asumsikan target T terletak pada koordinat  $X_T$ ,  $Y_T$ , probabilitas bahwa target T akan terdeteksi oleh sensor S dalam model biner dapat dinyatakan dengan persamaan berikut:

$$P_{sb} = \begin{cases} 1 & D_{TS} \leq R_S \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

Dimana  $P_{sb}$  merupakan probabilitas penginderaan dalam model biner dan adalah jarak antara target T dan sensor S yang dapat dihitung dengan persamaan berikut:

$$D_{TS} = \sqrt{(X_S - X_T)^2 + (Y_S - Y_T)^2} \quad (2)$$

Tidak ada masa transisi dalam model biner. Sedikit perbedaan dalam lokasi dapat menghasilkan output deteksi yang sama sekali berbeda. Model biner adalah model untuk penginderaan yang paling sederhana. Pada penelitian ini menggunakan model biner sebagai model penginderaan.

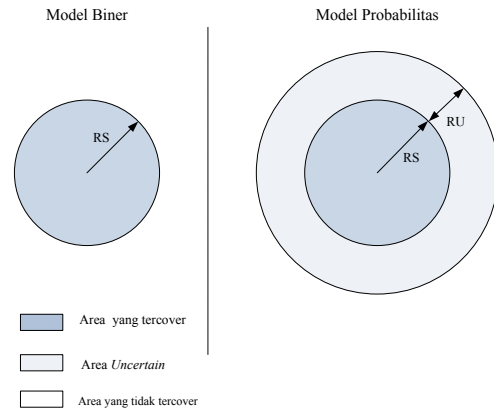
### B. Model Penginderaan Probabilitas

Dalam model probabilitas ada periode transisi ketika sensor benar-benar bisa dan tidak bisa mendeteksi target seperti dalam model biner. Selain itu juga ada daerah kisaran pasti untuk masing-masing sensor, dimana sensor tidak bisa mengatakan jika target dapat dideteksi. Karena pada daerah tersebut target akan memiliki probabilitas untuk dideteksi antara 0 dan 1.

Dalam model probabilitas, ada dua jarak kritis untuk sensor. Yang pertama adalah RS, yang sama seperti yang ada dalam model biner. Jika jarak antara target dan sensor kurang dari RS, target dapat dideteksi oleh sensor dengan probabilitas 1. Jarak kritis kedua adalah RU, yang merupakan singkatan dari radius uncertain atau kisaran pasti. Jika jarak antara target dan sensor berada dalam kisaran antara RS dan RS + RU, probabilitas bahwa target akan terdeteksi oleh sensor tersebut terkait dengan jarak antara mereka. Jika jarak antara target dan sensor yang lebih jauh dari RS + RU, target tidak akan terdeteksi oleh sensor. Ekspresi matematika model probabilitas adalah sebagai berikut:

$$P_{sp} = \begin{cases} 1 & D_{TS} \leq R_S \\ e^{-\alpha \left( \frac{D_{TS} - R_S}{R_S + R_U - R_S} \right)} & R_S < D_{TS} < R_S + R_U \\ 0 & R_S + R_U < D_{TS} \end{cases} \quad (3)$$

dimana  $\alpha = D_{TS} - R_S$ , dan  $\lambda$  dan  $\beta$  adalah konstanta yang berkaitan dengan sifat perangkat keras sensor.



Gambar 1. Perbedaan model biner dan model probabilitas

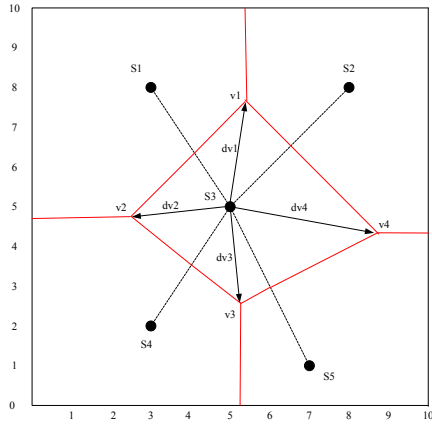
### C. Diagram Voronoi

Sebuah diagram Voronoi adalah metode dekomposisi suatu daerah. Asumsikan ada satu set node N dikerahkan di suatu daerah tanpa hambatan, diagram Voronoi akan membagi seluruh area ke N subarea, dan masing-masing subarea memiliki satu node di dalamnya. Karakteristik diagram Voronoi adalah bahwa setiap subarea terdiri dari daerah yang paling dekat dengan node di dalamnya, yang bertentangan dengan node lain. Generasi dari diagram Voronoi membutuhkan informasi lokasi dari semua node. Diagram Voronoi sangat berguna dalam masalah cakupan jaringan sensor nirkabel [5]. Jika setiap sensor dapat menutupi Voronoi subarea sendiri, seluruh bidang penginderaan dapat ditutup. Oleh karena itu, dalam rangka untuk menutupi seluruh bidang penginderaan, radius penginderaan setiap sensor harus ditetapkan sama dengan jarak antara sensor dan vertex yaitu titik perpotongan garis subarea terjauh diagram Voronoi:

$$R_S = \text{Max} (d_{vj}) \quad (4)$$

dimana  $d_{vj}$  adalah jarak antara node sensor dan vertex subarea.

Sebuah contoh dari diagram Voronoi ditunjukkan pada Gambar 2 Lima sensor S1, S2, ..., S5 dikerahkan di bidang penginderaan. Oleh karena itu, bidang penginderaan dibagi menjadi lima subareas oleh diagram Voronoi. Garis-garis lurus menunjukkan tepi subareas. Dengan mempertimbangkan subarea sensor S3 yang memiliki empat vertex. Di antara keempat vertex tersebut, v4 memiliki jarak terpanjang dari S3 node pusat. Oleh karena itu, dalam kasus ini radius penginderaan S3 harus  $d_{v4}$ .



Gambar 2. Diagram Voronoi dan kalkulasi radius penginderaan

#### D. Model Energi

Sebuah *node* sensor terdiri dari unit penginderaan, unit pemrosesan, unit pengiriman, dan unit daya. Masing-masing unit mengkonsumsi energi yang berbeda. Dalam penelitian ini hanya menganalisa untuk unit penginderaan saja. Beberapa model energi sudah digunakan untuk menganalisa hubungan antara radius penginderaan sensor dan energi yang dikonsumsi, dan biasanya tergantung pada karakteristik perangkat penginderaan [6]. Beberapa jenis model tersebut diantaranya model linear dan model quadratic. Pada penelitian ini hanya menggunakan model linear.

Ketika  $R_S$  lebih kecil dibandingkan  $R_{Smax}$ , konsumsi energi mempunyai hubungan dengan radius penginderaan  $R_S$ . Dalam model linear konsumsi energi suatu perangkat sensor mempunyai hubungan secara linear dengan  $R_S$ :

$$P_l = k_l \cdot R_S \quad (R_S < R_{SMax}) \quad (5)$$

dimana  $P_l$  adalah daya yang digunakan dalam penginderaan dengan menggunakan model linear dan  $k_l$  adalah konstanta perangkat. Untuk memaksimalkan masa pakai sensor, energi yang dikonsumsi harus diminimalkan. Hubungan antara *Lifetime* sensor ( $L$ ) dan energi yang dikonsumsi oleh sensor dapat digambarkan sebagai berikut :

$$\begin{aligned} E_{Total} &= E_S + E_{other} \\ E_{Total} &= L \cdot P_S + L \cdot P_{other} \end{aligned} \quad (6)$$

Dimana  $E_{Total}$  adalah total energi baterai,  $E_S$  adalah energi yang dikonsumsi dalam penginderaan dan  $E_{other}$  adalah energi yang digunakan untuk hal-hal lain, seperti komputasi dan komunikasi [7]. Dari persamaan (6) dapat diketahui berapa besar *Lifetime* sensor sebagai berikut :

$$L = \frac{E_{Total}}{P_S + P_{Other}} \quad (7)$$

#### E. Particle Swarm Optimization (PSO)

*Particle Swarm Optimization* (PSO) pertama kali diperkenalkan [8] pada tahun 1995. Gagasan tentang PSO berasal dari perilaku alami burung mencari makanan. Ketika sekelompok burung mencari makanan bersama-sama, masing-masing burung akan melihat-lihat di daerah yang dekat dengan dirinya. Setiap burung akan berkomunikasi dengan burung-burung lain dimana ia menemukan jumlah makanan yang paling dekat dengan wilayahnya. Dengan demikian, semua burung dapat mengetahui daerah mana yang memiliki jumlah besar makanan di seluruh area makan kolektif mereka. Burung akan terus mencari makanan di tempat-tempat terdekat, terutama jumlah sebagian besar makanan yang ditemukan di seluruh wilayah. Algoritma PSO menyederhanakan konsep yang terorganisir. Mirip dengan *Genetic Algorithm* (GA), sekelompok partikel akan dihasilkan dalam PSO dari seluruh ruang. Setiap partikel adalah satu set vektor yang berisi variabel yang berhubungan dengan masalah tersebut. Kelompok partikel akan berkembang dengan kombinasi *personal best fitness* (pBest) dan *group global best fitness* (gBest). Hal ini seperti proses burung mencari makanan. Dibandingkan dengan GA, PSO memiliki keuntungan lebih mudah untuk membuat program dan implementasi [1].

Tujuan dari PSO adalah untuk bertanggung jawab atas partikel terbaik dari suatu permasalahan. Masalah penyesuaian mencakup fungsi fitness yang menggambarkan suatu masalah. Pengejaran partikel ditujukan dengan mendirikan posisi terbaik dalam pencarian ruang yang dianggap sebagai posisi terbaik yang ditetapkan oleh partikel [9]. Pada bagian ini, terlepas dari encoding partikel dan fungsi fitness, beberapa akses dalam formasi, berat inersia, *velocity*, faktor konstiksi dan koefisien metode percepatan dalam memecahkan *deployment* WSN dengan PSO akan dibahas. *Particle Swarm Optimization* (PSO) disajikan untuk dimensi dari posisi dan kecepatan partikel ke- $i$  sebagai berikut :

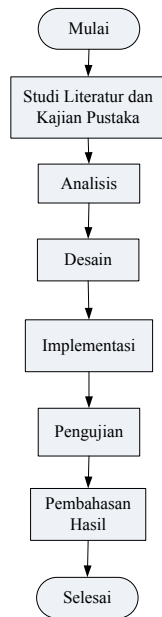
$$V_i^{k+1} = V_i^k + n1.r1(pB_i^k - X_i^k) + n2.r2(gB_i^k - X_i^k) \quad (8)$$

dengan

$$\begin{aligned} V_i^{k+1} &= \text{velocity partikel } i \text{ pada iterasi } k+1 \\ &= \text{bobot inersia} \\ X_i^{k+1} &= X_i^k + V_i^{k+1} \end{aligned} \quad (9)$$

### III. METODE PENELITIAN

Penelitian ini merupakan penelitian untuk pengembangan perangkat lunak *sensor node deployment* dan menggunakan bahasa pemrograman JAVA untuk membuat aplikasi *sensor node deployment*. Sensor yang digunakan dalam penelitian yaitu sensor IQRF dengan spesifikasi berdasarkan [10]. Metodologi penelitian yang akan dilakukan meliputi langkah-langkah sebagai berikut:



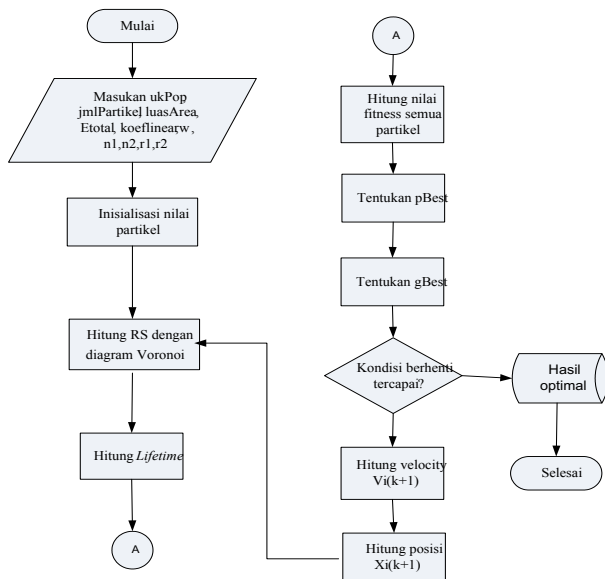
Gambar 3. Diagram Alir Penelitian

Rancangan sistem secara garis besar dapat dilihat pada Gambar 4



Gambar 4. Rancangan konseptual

#### A. Rancangan Diagram Alir Sistem



Gambar 5. Rancangan diagram alir sistem

#### B. Rancangan Optimasi

Rancangan optimasi dengan algoritma PSO adalah sebagai berikut:

- Ukuran Swarm

Pada penelitian ini jumlah partikel dibatasi dengan pertimbangan efisiensi waktu dan pencapaian solusi yang mendekati optimum global.

- Representasi Partikel.

Partikel ( $X_i$ ) merupakan suatu solusi yang dibangkitkan secara acak dan selanjutnya dioptimasi untuk menghasilkan solusi yang baik. Pada penelitian ini fokus optimasi adalah mengatur posisi node sensor ketika disebar sehingga bisa diperoleh radius penginderaan dengan menggunakan diagram Voronoi. Oleh karena itu partikel yang digunakan merepresentasikan posisi dari sejumlah *node sensor* dalam koordinat 2D dengan *area deployment* berbentuk bujur sangkar. Jumlah *node sensor* yang akan disebar bervariasi disesuaikan dengan kebutuhan pengguna aplikasi. Representasi dari partikel dapat dilihat pada Gambar 6. Untuk menjaga *deployment node sensor* berada dalam ruang masalah maka nilai  $X_{i,j}$  dibatasi pada rentang tertentu yaitu batas bawah ( $X_a = 0,0$ ) dan batas atas ( $X_b = \text{luas area sebaran}$ ).

|           |           |       |           |
|-----------|-----------|-------|-----------|
| $X_{1,1}$ | $X_{1,1}$ | ..... | $X_{1,j}$ |
|-----------|-----------|-------|-----------|

|           |           |       |           |
|-----------|-----------|-------|-----------|
| $X_{2,1}$ | $X_{2,2}$ | ..... | $X_{2,j}$ |
|-----------|-----------|-------|-----------|

⋮

|           |           |       |           |
|-----------|-----------|-------|-----------|
| $X_{i,1}$ | $X_{i,2}$ | ..... | $X_{i,j}$ |
|-----------|-----------|-------|-----------|

Gambar 6. Representasi partikel

$X_{i,j}$  = posisi partikel  $i$  node sensor  $j$  pada koordinat  $(x,y)$

$i$  = ukuran swarm

$j$  = jumlah node sensor

- Fungsi *fitness*

Selain representasi partikel, maka komponen penting lainnya dari PSO adalah fungsi *fitness*. Pada perancangan *deployment node sensor* ini digunakan fungsi *fitness* yang ditentukan oleh hal-hal berikut:

a) Radius penginderaan yang menentukan ratio cakupan area diperoleh dengan menggunakan diagram Voronoi.

Radius penginderaan mempunyai hubungan linear dengan penggunaan konsumsi energi, semakin besar radiusnya maka *ratio coverage*-nya akan besar tetapi energi yang dipakai akan semakin besar juga, untuk itu fungsi *fitness* di desain untuk menghasilkan nilai optimal dengan meminimalkan konsumsi energi sehingga akan memperpanjang waktu hidup jaringan tersebut.

b) Ketika mengimplementasi PSO, tidak ada himpunan nilai parameter yang tetap untuk penyelesaian semua jenis permasalahan.

Sehingga untuk menentukan himpunan parameter yang tepat terhadap suatu kasus dilakukan dengan cara mencoba-coba. Berdasarkan skenario tersebut maka ditetapkan fungsi *fitness* seperti pada persamaan (10) yaitu:

$$F(X_i) = \max \left( \frac{1}{L} + k_f \cdot R_{s_i} \right) \quad (10)$$

- Laju pembelajaran

Pada umumnya nilai-nilai untuk laju pembelajaran  $n1$  dan  $n2 = 2.0$  dan berada pada rentang antara 0 sampai 4. Pada penelitian ini, demi menyeimbangkan antara *cognitive part* dan *social part* pada algoritma PSO digunakan nilai  $n1$  dan  $n2 = 2.0$ .

- Bobot inersia

Perubahan *velocity* pada algoritma PSO terdiri atas tiga bagian yaitu *sosial part*, *cognitive part* dan *momentum part*. Ketiga bagian tersebut menentukan keseimbangan antara kemampuan penelusuran *global* dan *local*, karena dapat memberikan performansi yang baik pada PSO. *Inertia weight* diperkenalkan untuk keseimbangan antara kemampuan penelusuran *global* dan *local* seperti pada persamaan (8). Pada penelitian ini parameter *inertia weight* ( $w$ ) diperoleh dengan menggunakan persamaan (11) sebagai berikut.

$$= \max \frac{\max \min}{Iter_{\max}} Iter_i \quad (11)$$

dengan

$\max$  = bobot maksimal

$\min$  = bobot minimal

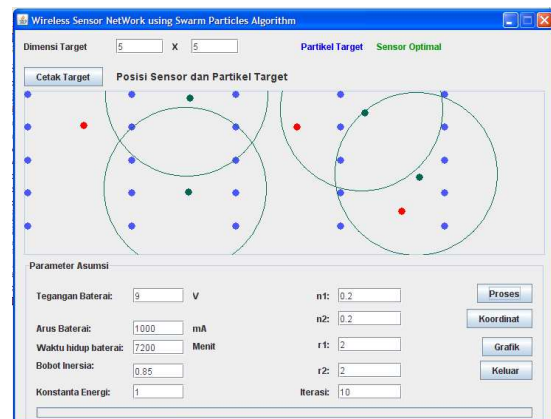
$Iter_{\max}$  = jumlah iterasi maksimal

$Iter_i$  = iterasi ke  $i$

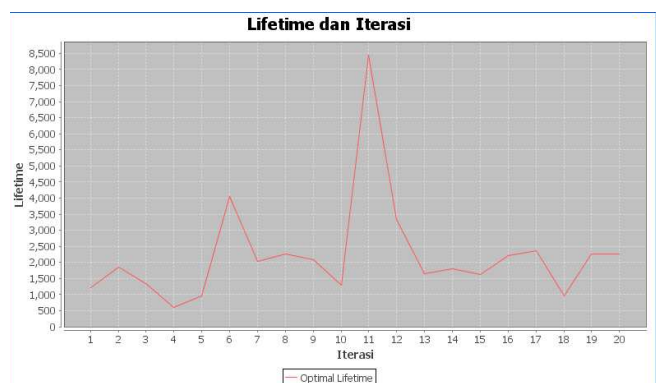
#### IV. ANALISIS HASIL IMPLEMENTASI

##### A. Pengujian Pertama

Pada pengujian pertama ini dilakukan dengan menggunakan ruang bidang penginderaan 5 meter x 5 meter dan jumlah sensor node sebanyak 4 buah. Kondisi *deployment node* sensor optimal ditunjukkan pada Gambar 7.



Gambar 7. Hasil layout *deployment* sensor dengan bidang 4x4 dan jumlah sensor

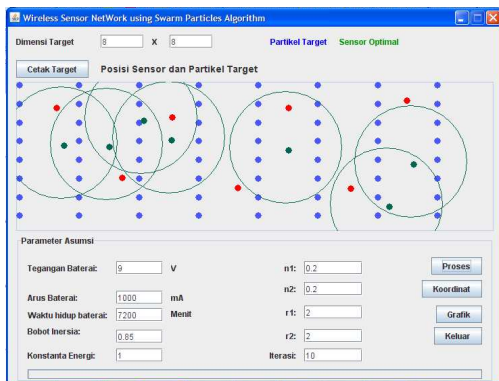


Gambar 8. Grafik perbandingan waktu hidup terhadap iterasi pada bidang 4x4.

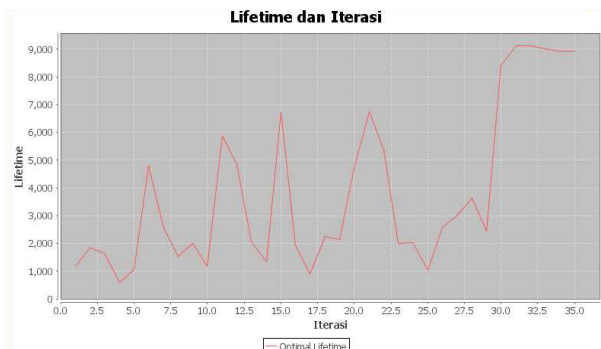
Dari grafik yang ditunjukkan oleh Gambar 8 menunjukkan sumbu x untuk iterasi dan sumbu y untuk nilai waktu hidup *node* sensor, terlihat bahwa sebelum iterasi yang ke 20 nilai waktu hidup masih terlihat mempunyai hasil yang fluktuatif yang artinya posisi node masih terus dilakukan perbaikan sehingga akhirnya mencapai optimal yang ditunjukkan dengan grafik yang sudah konvergen artinya setiap partikel swarm sudah mempunyai nilai posisi yang sama atau berdekatan. Nilai konvergen pada saat iterasi ke 20 diperoleh nilai sebesar 2500 menit, hal ini menunjukkan waktu hidup *node* sensor yang masih relatif sebentar dibandingkan dengan waktu hidup dari spesifikasi sensor.

Setelah dilakukan pengujian dengan ruang penginderaan 4x4, maka untuk melihat hasil aplikasi yang sudah di buat dilakukan dengan mencoba- coba memasukkan parameter-parameter yang berbeda-beda. Untuk pengujian kali ini dengan ruang penginderaan menggunakan 8 meter x 8 meter dan jumlah node sebanyak 8 buah node sensor.





Gambar 9. Hasil layout deployment sensor dengan bidang 8x8 dan 8 jumlah sensor



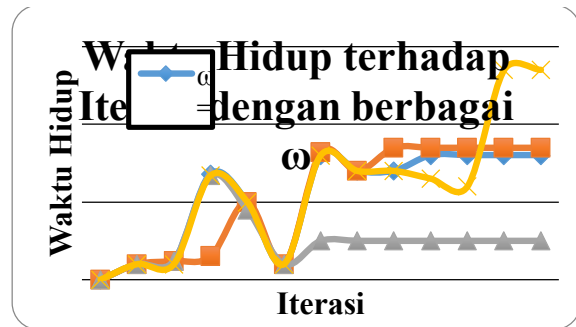
Gambar 10. Grafik waktu hidup dengan iterasi pada bidang 8x8

Gambar 10 yang menunjukkan grafik waktu hidup node sensor dengan iterasi pada saat pengujian aplikasi dilakukan dengan menggunakan ruang bidang penginderaan 8 meter x 8 meter dan jumlah *node* sensor sebanyak 8 buah. Dengan parameter-parameter yang sama pada saat pengujian dengan ruang bidang dan jumlah node sensor yang sebelumnya sudah dilakukan. Dari grafik tersebut menunjukkan pada saat iterasi ke 32 hasil sudah menunjukkan nilai yang konvergen yang artinya *deployment node* sensor sudah optimal sesuai dengan algoritma yang digunakan yaitu *Particle Swarm Optimization* (PSO). Hasil yang diperoleh menunjukkan nilai waktu hidup node sensor selama 9000 menit. Dengan hasil tersebut yang melebihi nilai spesifikasi sensor maka node sensor dapat digunakan sesuai spesifikasi yang ada, sehingga masa jaringan sensor nirkabel akan lebih lama. Namun dengan waktu hidup yang lama pada kasus ini mempunyai nilai cakupan area yang lebih sedikit dibanding pada saat pengujian yang dilakukan sebelumnya.

Dari kedua pengujian yang dilakukan dengan membedakan parameter ruang bidang penginderaan dan jumlah *node* sensor yang digunakan dapat diperoleh kesimpulan bahwa apabila nilai cakupan area besar maka waktu hidup akan lebih sebentar dibandingkan dengan nilai cakupan area yang kecil tetapi mempunyai waktu hidup yang lama. Cakupan area yang besar ini berarti radius penginderaan juga besar dan apabila radius penginderaan lebih besar maka energi yang dikonsumsi juga akan semakin besar. Untuk itu nilai fungsi *fitness* dirancang sedemikian rupa sehingga menghasilkan nilai trade-off diantara memaksimalkan cakupan area dan meminimalkan konsumsi energi sehingga masa jaringan sensor nirkabel akan lebih lama.

## B. Pengujian Kedua

Pengujian kedua dilakukan dengan membedakan nilai bobot inersia. Nilai bobot inersia yang diujikan yaitu 0.1, 0.3, 0.5 dan 0.7. Ruang bidang penginderaan yang digunakan sebesar 10 meter x 10 meter dengan jumlah *node* sensor sebanyak 10 buah dan parameter-parameter yang lainnya sama dengan pengujian sebelumnya. Hasil perbandingan berbagai bobot inersia dengan melihat hubungan antara waktu hidup *node* sensor dengan iterasi ditunjukkan dengan grafik yang dapat dilihat pada Gambar 11 sebagai berikut:



Gambar 11. Grafik waktu hidup terhadap iterasi dengan berbagai  $\omega$

Gambar 11 menunjukkan hasil dari algoritma untuk model energi linear. Sumbu x dan y masing-masing menunjukkan iterasi dan waktu hidup node sensor yang memiliki konsumsi energi optimal. Ketika  $\omega = 0,7$ , node sensor memiliki waktu hidup yang lama, yaitu sekitar 13000 meter dibandingkan dengan nilai  $\omega$  lainnya.

Dalam rangka untuk melihat manfaat dari algoritma optimasi, kita menghitung konsumsi energi sebelum menjalankan algoritma dan konsumsi energi dari hasil optimal setelah algoritma untuk setiap kasus. Karena nilai fungsi *fitness* adalah representasi dari konsumsi energi penginderaan, kita membandingkan penghematan energi untuk nilai-nilai bobot yang berbeda. Hasilnya ditunjukkan pada Gambar 4.5, di mana  $\omega$  berkisar 0,1-0,7 dengan langkah 0.2. Penghematan tersebut dihitung dengan menggunakan perbedaan nilai fungsi *fitness* dibagi dengan nilai *fitness* dari penyebaran awal. Sebagai contoh, asumsikan nilai fungsi *fitness* adalah 1,5 ketika sensor pada awalnya ditempatkan, dan nilai fungsi *fitness* terbaik adalah 0,9 setelah algoritma, maka dapat dihitung sebagai berikut  $(1,5-0,9) / 1,5 = 0,4 = 40\%$  yang artinya sudah memperoleh penghematan energi sebesar 40% setelah iterasi berikutnya dilanjutkan sampai mencapai nilai konvergen atau iterasi berakhir karena posisi setiap partikel swarm sudah mempunyai nilai yang sama atau hampir sama.

## V. KESIMPULAN

Dari hasil pengujian yang dilakukan terhadap aplikasi yang dibuat sesuai dengan rancangan konsep dan tujuan penelitian yang ingin didapat, maka didapatkan kesimpulan Radius penginderaan berpengaruh konsumsi energi node sensor. Semakin besar radius penginderaan maka konsumsi energi yang digunakan juga akan semakin besar sehingga masa jaringan atau waktu hidup node sensor akan lebih kecil atau pendek. Sebaliknya apabila radius penginderaan kecil maka konsumsi

energi yang digunakan sedikit sehingga masa jaringan akan lebih lama.

Dengan menggunakan bobot inersia pada algoritma PSO penghematan dapat dihitung dengan menggunakan perbedaan nilai fungsi fitness dibagi dengan nilai fitness dari penyebaran awal.

#### DAFTAR PUSTAKA

- [1] Y. Qu, "Wireless Sensor Network Deployment," Ph.D. dissertation, University Graduate School Florida International University, 2013.
- [2] K. Low, H. Nguyen dan H. Guo, "Optimization of sensor node locations in a wireless sensor network," in *Fourth International Conference on Natural Computation (ICNC)*, 2008, vol. 5, pp. 286–290.
- [3] H. Guo, H. Low dan H. Nguyen, "Optimizing the localization of a wireless sensor network in real time based on a low-cost microcontroller," *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 741–749, 2011.
- [4] T. Wimalajeewa dan S. Jayaweera, "Optimal Power Scheduling for Correlated Data Fusion in Wireless Sensor Networks via Constrained PSO," *IEEE Trans. Wireless Commun.*, vol. 7, no. 9, pp. 3608–3618, 2008.
- [5] H. Jang dan H. Lee, "A Voronoi dEtection Range Adjustment (VERA) approach for energy saving of wireless sensor networks," in *International Conference on Parallel and Distributed Systems*, 2007, vol. 2, pp. 1–7.
- [6] X. Wang, J. Ma, S. Wang dan D. Bi, "Distributed Energy Optimization for Target Tracking in Wireless Sensor Networks," *IEEE Trans. Mobile Comput.*, vol. 9, pp. 73–86, 2010.
- [7] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh dan D. Rubenstein, "Energy efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," in *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, San Jose, CA, Oct. 2002.
- [8] J. Kennedy, R. Eberhart dan Y. Shi, *Swarm Intelligence* / book auth. Kafman Morgan. - Los Altos, CA, 2001.
- [9] R. Eberhart, Y. Shi, "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization". In *Proceedings of IEEE International IEEE International*, La Jolla, CA , USA ,2001, - pp. 84–88.
- [10] Datasheet IQRF TR52-B. [terhubung berkala]. <http://www.iqrf.org/weben/downloads.php?id=91>. [19 apr 2011].
- [11] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, dan E. Cayirci, "Wireless Sensor Networks: a Survey". *Computer Networks* vol 38: 393– 422, 2002.
- [12] X. Bai, S. Li, C. Jiang dan Z. Gao, "Coverage Optimization in Wireless Mobile Sensor Networks," in *5th International Conference on Wireless Communications, Networking and Mobile Computing*, 2009, pp. 1–4, Beijing, China, September 24–26
- [13] R. Burne, A. Buczak dan Y. Jin, "A self-organizing, cooperative sensor network for remote surveillance:current result," in *13th Annual International Symposium on AeroSense Conference–Unattended Ground Sensor Technologies and Applications*, pp. 238–248, Orlando, FL, 1999..
- [14] R. Kulkarni dan G. Venayagamoorthy, "Particle Swarm Optimization in Wireless Sensor Networks: A Brief Survey," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 41, no. 2, pp. 262–267, 2001