

Simulasi *Deployment* Jaringan Sensor Nirkabel Berdasarkan Algoritma *Particle Swarm Optimization*

Zawiyah Saharuna¹, Widyawan², Sujoko Sumaryono³

Abstract— Deployment is one of several important issues in Wireless Sensor Network (WSN). During WSN deployment, the connectivity between each sensor nodes must be considered carefully to create reliable communication. In this research, we propose a WSN deployment tool based on Particle Swarm Optimization (PSO) algorithm with connectivity of the wireless to be our concern. Implementation of the PSO algorithm is focused to optimize received power of each sensor node based on its position in the 2D space. Therefore, every sensor node in the network will be able to reach its best position and improves the network connectivity. The deployment simulation results with various transmit power (e.g. -25 dB, -28 dB, -31 dB, -34 dB, -37 dB, -40 dB, -43 dB, and -46 dB) are successfully form a network with well maintained connectivity. After 100 times of simulation, the average of convergence is reached at iteration 29.

Intisari— *Deployment* merupakan topik penting dalam bidang *Wireless Sensor Network* (WSN). Proses *deployment* harus mempertimbangkan konektivitas, sehingga komunikasi antar *node sensor* dalam jaringan bisa berlangsung dengan baik. Pada penelitian ini, dilakukan simulasi *deployment* dari aplikasi *sensor node deployment* berdasarkan Algoritma *Particle Swarm Optimization* (PSO) dengan mempertimbangkan konektivitas jaringan nirkabel. Implementasi PSO difokuskan pada optimasi posisi *node sensor* pada koordinat 2D sehingga *node sensor* mendapatkan posisi terbaiknya dan bisa meningkatkan konektivitasnya dalam jaringan sensor nirkabel. Hasil *deployment* dengan daya pancar sebesar -25dB, -28dB-31 dB, -34 dB, -37 dB, -40 dB, -43 dB hingga -46 dB membentuk jaringan dengan konektivitas yang tetap terjaga dan kondisi konvergen dari 100 pengujian rata-rata dicapai pada iterasi ke 29.

Kata Kunci— WSN, *Deployment*, Konektivitas, PSO

I. PENDAHULUAN

Wireless Sensor Network (WSN) atau jaringan sensor nirkabel merupakan salah satu teknologi inti dalam bidang *Ubiquitous Computing*. WSN terdiri atas satu atau lebih sensor (*node sensor*) digunakan untuk menangkap informasi sesuai dengan karakteristiknya[1]. Teknologi WSN dapat diterapkan di berbagai area karena didukung oleh bentuk perangkat keras yang kecil serta media komunikasi yang menggunakan

jaringan nirkabel sehingga bisa berinteraksi dengan baik dalam lingkungan.

Kondisi fisik lingkungan diterapkannya WSN akan mempengaruhi tingkat konektivitas dari *node sensor* dalam jaringannya. Padahal konektivitas merupakan penentu keberhasilan dalam komunikasi antar *node sensor*. Oleh karena itu perlu memperhatikan posisi *node sensor* ketika proses *deployment* dalam penerapan WSN. Beberapa peneliti telah mengajukan berbagai algoritma untuk mengoptimalkan proses *deployment* di lingkungan sehingga kinerja WSN bisa meningkat.

Lee menggunakan *basic deployment algorithm* untuk proses *deployment node* secara otomatis berdasarkan algoritma *Swarm Intelligence* kemudian mengoptimasi beberapa parameter seperti lebar area cakupan dan jumlah *overlapping* [2]. Dong Li, dkk menghadirkan sebuah aplikasi optimasi proses *deployment* menggunakan DT-ACO (*Deployment Technology based on Ant Colony Optimization Algorithm*) dengan mempertimbangkan konektivitas jaringan nirkabel, area cakupan sensor, dan biaya. Mereka juga menawarkan PACS (*Power-Aware Cross layer Scheme*) untuk menambah waktu hidup dan keakuratan pengamatan bagi *node sensor* [3]. Selain itu, *Memetic Simulated annealing* juga digunakan sebagai solusi atas permasalahan dalam proses *deployment* untuk optimasi area cakupan dan waktu hidup dari *node sensor* [4].

Beberapa penelitian juga mencoba menawarkan algoritma baru seperti algoritma *Robot-deployment* untuk mengatasi penghalang yang tidak diperkirakan dan mengoptimalkan area sebaran bagi *node sensor* dalam jumlah minimal [5]. Serta algoritma *obstacle-resistant robot deployment* (ORRD) yang melibatkan kebijakan desain penempatan *node*, kebijakan *serpentinemovement*, keberadaan penghalang, dan batasan area. Algoritma ini mampu dengan cepat menyebarkan *node sensor* dalam jumlah minimal untuk mencakup *sensing area* secara penuh, meskipun terdapat penghalang yang tidak diperkirakan dengan bentuk teratur atau tidak teratur [6].

Sebagian besar penelitian di bidang *deployment* WSN fokus pada permasalahan *sensing coverage* bukan pada konektivitas jaringan [2], [4], [5], dan [6]. Sedangkan pada penelitian ini dilakukan simulasi *deployment node sensor* berdasarkan Algoritma *Particle Swarm Optimization* (PSO) dengan mempertimbangkan konektivitas jaringan nirkabel. Algoritma PSO dipilih berdasarkan keunggulannya, yaitu mudah diimplementasikan karena persamaan matematisnya sederhana dan hanya memiliki sedikit fungsi operasi dan parameter yang harus ditentukan [7]. Hal inilah yang menjadikan algoritma PSO memiliki karakteristik cepat dalam eksekusi.

¹ Jurusan Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada, Jln. Grafika 2 Yogyakarta 55281 INDONESIA (e-mail: wia_saharuna_s210@mail.te.ugm.ac.id)

^{2,3} Jurusan Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada, Jln. Grafika 2 Yogyakarta 55281 INDONESIA (e-mail: widyawan@gmail.com, sujoko@mti.ugm.ac.id)

II. MODEL PROPAGASI GELOMBANG DI RUANG BEBAS

Propagasi ruang bebas terjadi bila ada jalur *line of sight* (LOS) yaitu tidak ada penghalang yang menghalangi propagasi gelombang radio antara pemancar dan penerima. Pada propagasi ruang bebas hanya ada rugi-rugi sinyal sebagai fungsi jarak antara pemancar dan penerima. Model yang paling sederhana adalah apabila kondisi saling melihat antara pemancar dan penerima terpenuhi dan hanya ada satu sinyal langsung yang diterima, sehingga perhitungan redaman dilakukan dengan menggunakan rumus redaman ruang bebas (free space loss). (1) adalah rumus redaman ruang bebas dalam *decibel* [8].

$$L = 32,44 + 20 \log f + 20 \log d \quad (1)$$

L : redaman ruang bebas.

f : frekuensi dalam MHz.

d : jarak antar pengirim-penerima dalam Km.

Besarnya kekuatan sinyal yang diterima pada penerima dapat dihitung berdasarkan rumus berikut:

$$Pr = Pt + Gt + Gr - L \quad (2)$$

$$Pr = Pt + Gt + Gr - (32,44 + 20 \log f + 20 \log d) \quad (3)$$

Pt : daya yang dipancarkan oleh pengirim.

Gt : penguatan antena pemancar/pengirim.

Gr : penguatan antena penerima.

III. PARTICLE SWARM OPTIMIZATION (PSO) DAN IMPLEMENTASI

Algoritma *Particle Swarm Intelligence* (PSO) merupakan salah satu algoritma optimasi berbasis populasi yang terinspirasi oleh perilaku sosial binatang seperti kawanan burung atau ikan. Algoritma ini diperkenalkan oleh Eberhart dan Kennedy pada tahun 1995. Berbeda dengan teknik optimasi lainnya, setiap partikel dalam PSO berhubungan dengan suatu *velocity*. Partikel tersebut bergerak melalui penelusuran ruang dengan *velocity* yang dinamis disesuaikan menurut perilaku historisnya. Oleh karena itu partikel-partikel mempunyai kecenderungan untuk bergerak ke area penelusuran yang lebih baik setelah melewati proses penelusuran[9].

PSO dimulai dengan sekumpulan partikel (solusi) yang dibangkitkan secara acak. Kemudian setiap partikel dievaluasi kualitasnya menggunakan fungsi *fitness*. Selanjutnya, partikel-partikel akan terbang mengikuti partikel yang optimum. Pada setiap generasi (iterasi), setiap partikel diperbaharui berdasarkan dua nilai terbaik yaitu *fitness* terbaik yang dicapai oleh satu partikel saat ini yang dikenal dengan istilah personal best (pBest) dan *fitness* terbaik yang dicapai dari semua partikel yang ada pada suatu kawanan yang disebut Global best (gBest). Setelah menemukan dua nilai terbaik tersebut maka setiap partikel *i* pada posisi *Xi* memperbaharui vektor *velocity* dan posisinya berdasarkan persamaan berikut:

$$V_i^{k+1} = \begin{cases} K[V_i^k + n1 \cdot r(pBi^k - X_i^k) + n2 \cdot r(gBi^k - X_i^k)] & \text{jika } X_{\min} < X_i^k < X_{\max} \\ 0 & \text{lainnya} \end{cases} \quad (4)$$

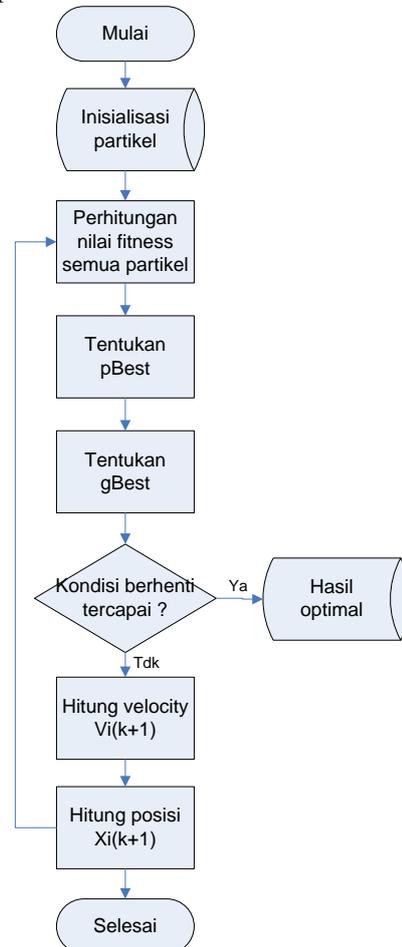
$$K = \frac{2}{2 - n - \sqrt{n^2 - 4n}} \quad (5)$$

$$X_i^{k+1} = \begin{cases} X_i^k + V_i^{k+1} & \text{jika } X_{\min} < (X_i^k + V_i^{k+1}) < X_{\max} \\ X_{\max} & \text{jika } (X_i^k + V_i^{k+1}) > X_{\max} \\ X_{\min} & \text{jika } (X_i^k + V_i^{k+1}) < X_{\min} \end{cases} \quad (6)$$

X_{\max} : Batas atas permasalahan

X_{\min} : Batas bawah permasalahan

Prosedur yang digunakan untuk menerapkan algoritma PSO dapat dilihat pada Gbr. 1.



Gbr. 1 Diagram alir PSO

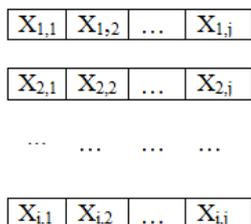
Implementasi PSO pada proses *deployment* adalah sebagai berikut:

A. Ukuran Swarm

Carlisle menyatakan bahwa ukuran *swarm* akan mempengaruhi kecepatan proses. Ukuran *swarm* yang terlalu kecil bisa terjebak pada optimum lokal meskipun waktu prosesnya sangat cepat. Sebaliknya, ukuran *swarm* yang besar jarang terjebak pada optimum lokal tetapi waktu prosesnya lebih lama[10]. Pada penelitian ini digunakan 30 partikel dengan pertimbangan efisiensi waktu dan pencapaian solusi yang mendekati optimum global.

B. 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 posisi *node sensor* ketika disebar, sehingga 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 Gbr. 2.



Gbr. 2 Representasi partikel

- $X_{i,j}$: posisi partikel i *node sensor* j di koordinat (x,y)
- i : ukuran swarm
- j : jumlah *node sensor*

Untuk menjaga pencarian solusi tetap berada dalam ruang masalah maka nilai $X_{i,j}$ dibatasi pada range tertentu yaitu batas bawah ($X_a = 0,0$) dan batas atas ($X_b =$ luas area sebaran).

C. Fungsi fitness

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

1) Konektivitas diukur berdasarkan daya yang diterima oleh sebuah *node*, dalam hal ini batasan maksimal daya yang diterima suatu *node* dianggap terkoneksi dengan *node* yang lain adalah -80 dBm. Nilai batasan yang ditetapkan lebih besar dari batas minimal -110 dBm karena pertimbangan jangkauan frekuensi radio TR-52B bisa mencapai 700 m (1.2 kb/s) dan 500 m (19.2 kb/s), padahal luas area sebaran yang ditetapkan dalam pengujian tidak seluas area tersebut [11].

2) Skenario proses *deployment* dirancang agar semua *node sensor* dalam jaringan saling terkoneksi. Kondisi terbaik ketika jarak antar *node sensor* maksimal tetapi daya yang diterima oleh *node sensor* tersebut ≥ -110 dB dan jumlah koneksi mendekati $N(N-1)$, dengan N adalah jumlah *node sensor* yang disebar.

Berdasarkan skenario tersebut maka ditetapkan fungsi *fitness* sebagai berikut:

$$F(X_i) = \sum_0^{j-1} Pr_b(X_{ij}) - \sum_0^{j-1} A(X_{ij}) \tag{7}$$

$F(X_i)$: fungsi *fitness* partikel i

$Pr_b(X_{ij})$: sinyal terbaik yang diterima partikel i *node sensor* j .

$A(X_{ij})$: jumlah *node tetangga* yang terdeteksi pada partikel i *node sensor* j .

$$Pr_b(X_{ij}) = \min[Pr(X_{ij})] > (-110db) \tag{8}$$

$Pr_b(X_{ij})$ merupakan sinyal terbaik yang diterima oleh partikel i *node sensor* j dengan asumsi bahwa kekuatan sinyal terbaik adalah yang mendekati nilai -110dB. Perhitungan kekuatan sinyal yang diterima menggunakan (3) dengan mengabaikan nilai penguatan pada antenna karena kecilnya penguatan yang terjadi pada antenna *node sensor* tipe TR52B[11]. Sedangkan untuk menghitung jarak dengan rumus *Euclidean distance*, seperti menghitung jarak antar koordinat (x_i,y_i) dengan (x_j,y_j) pada (9).

$$D(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{9}$$

D. Laju Pembelajaran

Pada penelitian ini digunakan nilai $n1=2,8$ dan $n2 = 1,3$ dengan pertimbangan untuk menyeimbangkan antara *cognitive part* dan *social part* pada PSO.

E. Bobot Inersia

Perubahan *velocity* pada algoritma PSO terdiri atas tiga bagian yaitu *social 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.

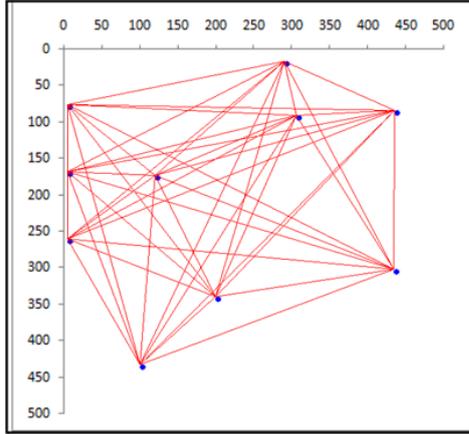
IV. HASIL PENGUJIAN APLIKASI

Untuk mengetahui kelayakan implementasi algoritma PSO pada *deployment* WSN dilakukan simulasi pengujian dengan menggunakan bahasa pemrograman Java. Pengujian aplikasi ini menggunakan 10 *node sensor* yang akan disebar pada area dengan ukuran 500x500m². Komunikasi dalam jaringan menggunakan frekuensi 868 MHz, daya pancar sebesar -25dB, -28dB-31 dB, -34 dB, -37 dB, -40 dB, -43 dB hingga -46 dB yang disesuaikan dengan spesifikasi WSN jenis IQRF tipe TR-52B. Batasan daya yang diterima oleh *node sensor* sehingga dianggap terhubung dengan *node sensor* tetangganya adalah -110dB. Parameter komponen PSO yang digunakan diatur dengan jumlah partikel sebanyak 30 (0-29), dan iterasi maksimal sebanyak 45 iterasi.

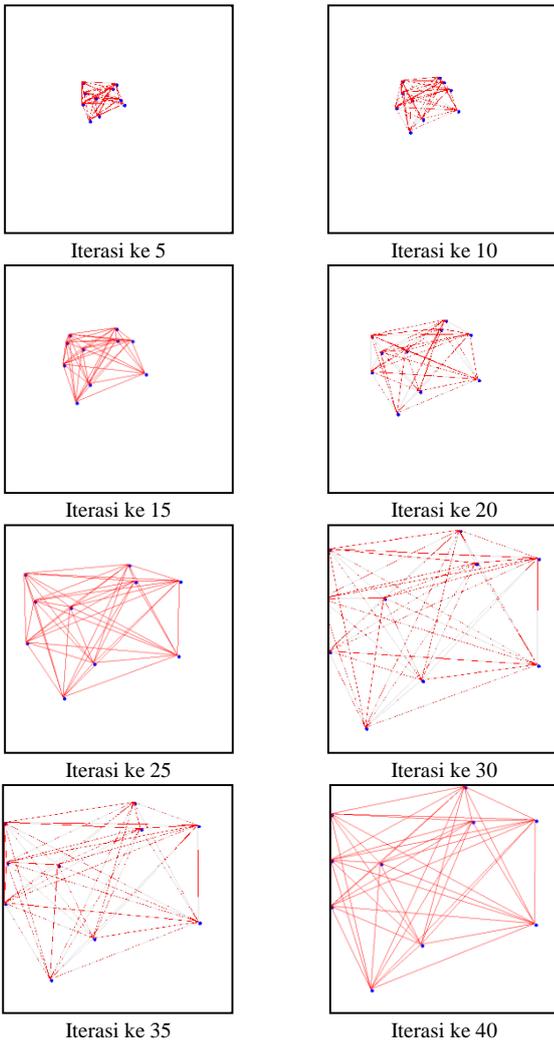
Pada pengujian proses *deployment* dengan delapan variasi daya pancar yaitu -25dB, -28dB, -31 dB, -34 dB, -37 dB, -40 dB, -43 dB hingga -46 dB diperoleh hasil *deployment* terbaik pada penggunaan daya pancar -25dB. Hasil *deployment* seperti pada Gbr. 3 membentuk jaringan *node sensor* yang saling terhubung satu sama lain sehingga konektivitasnya terjaga seperti yang diinginkan pada skenario desain. Kondisi dua *node sensor* tersebut dinyatakan terhubung adalah ketika kedua *node sensor* tersebut dihubungkan oleh garis berwarna merah. Garis merah merupakan representasi daya ≥ -110 dB. Sedangkan kondisi tidak terhubung dilambangkan oleh garis abu-abu yang merepresentasikan daya < -110 dB.

Desain hasil *deployment* pada Gbr. 3 merupakan representasi solusi terbaik dari partikel pada *swarm* yaitu solusi dari partikel ke 16 yang memiliki nilai *fitness* terbaik sebesar -

9493,9437. Pada Gbr. 4 menjelaskan hasil *deployment* partikel terbaik dengan daya pancar -25dB setiap lima iterasi.



Gbr. 3 Hasil *deployment* partikel ke 16 pada iterasi maksimal

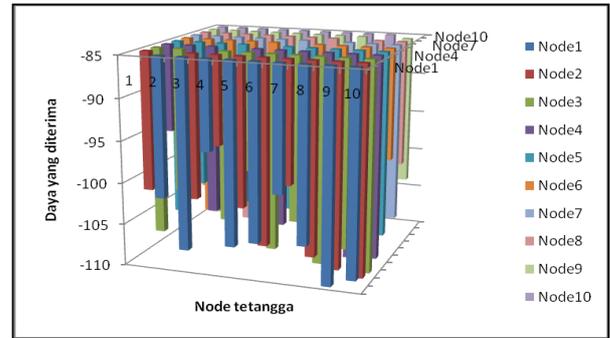


Gbr. 4 Hasil *deployment* -25dB 868 MHz per 5 iterasi

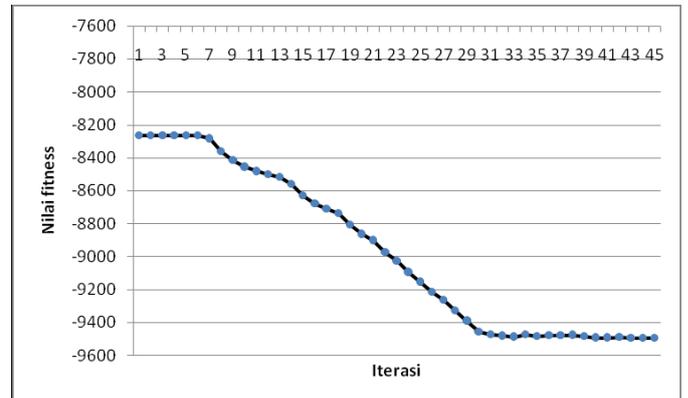
Skenario penelitian ini bisa mengatasi penyebaran *node sensor* di luar area *deployment*. Hal tersebut terjadi karena pada saat perbaikan *velocity* dan posisi *node sensor* terdapat kontrol

pada *Vmax* agar *node sensor* senantiasa berada dalam area *deployment*.

Adapun besarnya daya yang diterima oleh masing-masing *node sensor* pada partikel 16 ketika kondisi konvergen berada dalam jangkauan -92,67 dB sampai dengan -110 dB. Beberapa nilai tidak memenuhi batas yang disyaratkan untuk konektivitas. Untuk lebih jelasnya rentang nilai daya yang diterima dapat dilihat pada Gbr. 5. Sumbu x menjelaskan *node sensor* tetangga yang terdeteksi, sedangkan sumbu y adalah nilai daya yang diterima dalam besaran dB.



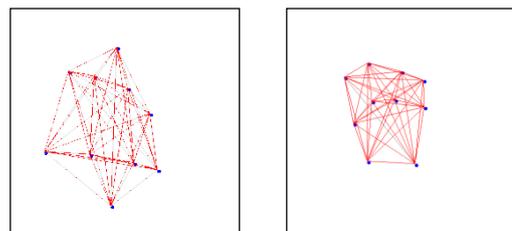
Gbr. 5 Grafik penerimaan sinyal setiap node pada partikel ke 16

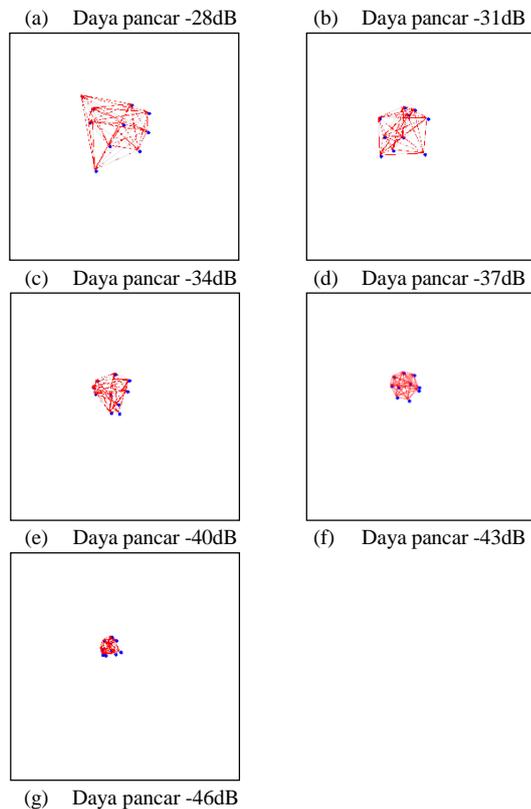


Gbr. 6 Laju konvergensi

Pada Gbr. 6 menjelaskan tentang laju konvergensi yang dihasilkan ketika proses *deployment* 10 *node sensor*. Hasilnya diperoleh dari nilai *fitness* terbaik pada setiap iterasi hingga iterasi maksimal. Kondisi konvergen mulai terjadi pada iterasi ke 30 dengan nilai *fitness* terbaik -9456.0557 pada indeks partikel ke 21 hingga iterasi maksimal dengan nilai *fitness* terbaik -9493.9437 pada indeks partikel ke 16.

Setelah melakukan simulasi *deployment* sebanyak 100 kali untuk semua daya pancar, maka aplikasi mencapai kondisi konvergen pada rata-rata iterasi ke 29.





Gbr. 7 Hasil *deployment node* dengan variasi daya pancar

Seperti Gbr. 3 hasil *deployment* dengan daya pancar yang berbeda-beda juga ditunjukkan oleh Gbr. 7. Hasilnya juga memenuhi ketentuan pada skenario desain yaitu memiliki tingkat konektivitas yang tetap terjaga. Gbr. 7 (a) sampai dengan 7 (g) merupakan hasil *deployment node sensor* dengan daya pancar secara berurutan sebesar -28 dB, -31 dB, -34 dB, -37 dB, -40 dB, -43 dB dan -46 dB. Ketujuh hasil *deployment* tersebut membentuk jaringan dengan kualitas yang berbeda. Semakin besar daya yang dipancarkan maka semakin besar jarak antar *node sensor* sehingga hasil *deployment* bisa mencakup area yang luas. Sebaliknya semakin kecil daya yang dipancarkan maka jarak antar *node sensor* semakin dekat dan hanya bisa mencakup area *deployment* yang lebih sempit. Untuk semua hasil *deployment* diperoleh kualitas konektivitas jaringan yang tetap terjaga. Hasil tersebut memenuhi kondisi yang disyaratkan pada penetapan fungsi *fitness* yang digunakan untuk mengevaluasi nilai *fitness* partikel yaitu membentuk jaringan yang tetap mempertahankan konektivitas. Hasil *deployment* terbaik diperoleh pada proses *deployment* menggunakan daya pancar -25 dB dengan pencapaian jarak antara dua *node sensor* maksimal tetapi daya yang diterima oleh *node sensor* tersebut ≥ -110 dB dan jumlah koneksi mendekati 90 buah sehingga nilai fungsi *fitness*nya mendekati nilai *fitness* maksimal -9990.

Pada Tabel 1 menjelaskan batasan nilai daya yang diterima oleh masing-masing *node sensor* berdasarkan perbedaan daya pancar *node sensor* tetangganya. Ketika daya yang dipancarkan sebesar -28dB, -31dB, -34dB, -37dB, -40dB, -43dB dan -46dB daya minimum yang diterima oleh *node* adalah -110dB dan

membentuk jaringan *full-mesh*, sehingga *node sensor* tetap terhubung dengan tetangganya dan konektivitasnya tetap terjaga. Kondisi tersebut seperti yang ditetapkan pada skenario *deployment*. Sedangkan ketika semua daya yang diterima lebih kecil dari -110dB, akan menyebabkan *node sensor* tidak terhubung dengan *node sensor* tetangganya atau dengan kata lain konektivitas jaringan buruk.

TABEL I
DAYA YANG DITERIMA PARTIKEL TERBAIK BERDASARKAN VARIASI DAYA PANCAR

Daya Pancar (dB)	Partikel terbaik	Daya yang Diterima (dB)	
		Maksimal	Minimal
-28	11	-93,90	-109,98
-31	18	-96,31	-109,99
-34	10	-94,89	-109,99
-37	10	-88,46	-109,99
-40	14	-94,96	-109,99
-43	28	-91,02	-109,99
-46	20	-93,16	-109,99

Penelitian ini bisa mengatasi penyebaran *node sensor* luar area *deployment* karena terdapat kontrol pada V_{max} saat perbaikan nilai *velocity* serta posisi *node sensor* dan Algoritma PSO berhasil digunakan pada *deployment* jaringan sensor nirkabel. Meski demikian, penelitian ini juga memiliki beberapa keterbatasan seperti hasil simulasi cenderung membentuk jaringan *fullmesh* padahal ketentuan konektivitas tidak harus *fullmesh*. Selain itu jarak antara *node sensor* dengan *node sensor* tetangganya beberapa masih terlalu dekat sehingga belum bisa memperhitungkan berapa jumlah *node sensor* yang dibutuhkan untuk area sebaran dengan luasan tertentu.

V. KESIMPULAN

Algoritma PSO berhasil digunakan pada *deployment* jaringan sensor nirkabel seperti algoritma jenis *swarm intelligence* lainnya. Pengujian *deployment* WSN jenis IQRf tipe TR-52B dengan delapan daya pancar yang berbeda mendapatkan hasil bahwa semakin besar daya yang dipancarkan maka semakin besar jarak antar *node sensor* dan kualitas konektivitas jaringannya tetap terjaga. Sebaliknya semakin kecil daya yang dipancarkan maka jarak antar *node sensor* semakin dekat dan untuk kondisi tertentu kualitas konektivitas jaringannya akan berkurang. Meski demikian, Algoritma ini memiliki kekurangan seperti hasil simulasi cenderung membentuk jaringan *fullmesh* padahal ketentuan konektivitas tidak harus *fullmesh*. Selain itu jarak antara *node sensor* dengan *node sensor* tetangganya beberapa masih terlalu dekat sehingga belum bisa memperhitungkan berapa jumlah *node sensor* yang dibutuhkan untuk area sebaran dengan luasan tertentu. Hal ini bisa dijadikan pertimbangan untuk penelitian selanjutnya.

REFERENSI

- [1] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.
- [2] J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

- [3] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.
- [4] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in *Proc. ECOC'00*, 2000, paper 11.3.4, p. 109.
- [5] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.
- [6] (2002) The IEEE website. [Online]. Available: <http://www.ieee.org/>
- [7] M. Shell. (2002) IEEETran homepage on CTAN. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEETran/>
- [8] *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.
- [9] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
- [10] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
- [11] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.
- [12] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.