

# Implementasi Algoritma Konsensus untuk Melokalisir Keberadaan Api Kebakaran pada Jaringan Sensor Nirkabel

Rudi Prasongko<sup>1</sup>, Prima Kristalina<sup>2</sup>, Achmad Subhan K.H<sup>2</sup>

<sup>1</sup>Mahasiswa Jurusan Teknik Telekomunikasi

<sup>2</sup>Dosen Politeknik Elektronika Negeri Surabaya

Institut Teknologi Sepuluh Nopember, Kampus ITS, Surabaya 60111

Email : [rud\\_pras@yahoo.co.id](mailto:rud_pras@yahoo.co.id)

## Abstrak

Indonesia merupakan salah satu dari 6 negara di dunia yang memiliki luas hutan hampir 60% luas hutan di dunia. FAO mencatat laju kerusakan hutan di Indonesia adalah yang tercepat di dunia. Dampaknya akan menjadi besar apabila pendeteksian dan lokalisir kebakaran lambat sehingga api akan cepat menyebar secara luas dan kerusakan yang ditimbulkan semakin besar.

Solusi yang ditawarkan dari permasalahan tersebut adalah adanya suatu *jaringan sensor nirkabel* yang dipasang di hutan. Sistem ini dapat memonitoring kondisi hutan, mendeteksi dan melokalisir api saat terjadi kebakaran hutan. Untuk membangun sistem tersebut memerlukan dana yang cukup besar maka perlu adanya suatu aplikasi untuk mensimulasikan pendeteksian dan melokalisir api dengan mengimplementasikan suatu algoritma untuk mengatasi suatu permasalahan yang timbul saat terjadi kebakaran hutan.

Hasil simulasi menunjukkan bahwa daya optimasi yang dihitung dengan Algoritma ANC (*Average Network Consensus*) dengan input data Castalia lebih tinggi nilainya daripada dengan simulasi ideal (dengan Matlab) dan jika jumlah sensor semakin banyak maka nilai MSE semakin turun. Sensor yang menghasilkan daya besar adalah sensor yang dekat dengan sumber api, sehingga dengan membaca output masing-masing sensor diketahui lokasi sumber api yang sebenarnya.

**Kata kunci :** Lokalisir, Jaringan Sensor Nirkabel, Algoritma Konsensus

## 1. PENDAHULUAN

Perkembangan teknologi komunikasi *wireless* saat ini semakin pesat sehingga telah ditemukan suatu alat komunikasi *wireless* yang dinamakan *Wireless sensor network (WSN)*. *Wireless sensor network* merupakan suatu infrastruktur yang memiliki suatu kemampuan sensing (deteksi), penghitungan dan elemen-elemen komunikasi yang memberikan kemampuan kepada administrator untuk mengukur, mengobservasi, dan memberikan reaksi kepada event (kejadian) dan fenomena pada lingkungan tertentu seperti temperatur, suara, cahaya, getaran, tekanan dan lain-lain.

Aplikasi WSN pada umumnya digunakan untuk kebutuhan monitoring, tracking, dan kontroling. Adapun aplikasi yang lebih spesifik adalah memonitor habitat hewan, pelacakan suatu objek, kontrol reaktor nuklir, pendeteksian api, memonitor lalu lintas dan lain-lain. Dalam beberapa aplikasi dilapangan, WSN seringkali digunakan untuk mengamati suatu daerah yang cukup luas tetapi pada kenyataannya daya jangkauan WSN tersebut sangatlah terbatas.

Pada proyek akhir ini diterapkan algoritma konsensus untuk melakukan lokalisir titik api pada wilayah yang telah disebari sensor. Tujuannya agar sensor-sensor yang berada tepat di dekat titik api segera mengirimkan informasi *sensing*-nya,

sehingga dapat dilakukan pencegahan terhadap meluasnya titik-titik api kebakaran.

Sebelumnya telah dilakukan penelitian tentang deteksi dan lokalisir kebakaran hutan berdasarkan algoritma konsensus menggunakan jaringan sensor nirkabel oleh Alireza Khadivi dan Martin Hasler [6] yang hasilnya dapat mendeteksi dengan cepat adanya api kebakaran dan dapat melakukan monitoring secara *real time*.

Susunan penulisan makalah ini adalah sebagai berikut: bagian 1 yaitu pendahuluan, bagian 2 teori pendukung, bagian 3 metodologi, bagian 4 analisa dan pembahasan, bagian 5 kesimpulan dan bagian terakhir yaitu daftar pustaka. Untuk lebih jelasnya sebagai berikut:

## 2. TEORI PENUNJANG

### 2.1 Jaringan Sensor Nirkabel

*Wireless sensor network (WSN)* atau jaringan sensor nirkabel merupakan suatu jaringan nirkabel yang terdiri dari beberapa sensor (sensor node) yang diletakkan ditempat - tempat yang berbeda untuk memonitoring kondisi suatu plan. Sebuah WSN dapat menghemat waktu dan biaya untuk penyebaran sensor, menambah efisiensi dan mudah perawatannya.[1]

## 2.2 Algoritma Konsensus Terdistribusi pada Jaringan Sensor Nirkabel

Secara sederhana algoritma konsensus terdistribusi adalah suatu algoritma dimana data yang diperoleh oleh setiap sensor akan diperbaharui setiap waktu oleh sensor tetangganya. Yang dimaksud sensor tetangga adalah semua sensor yang terkoneksi dengan sensor tersebut. Tujuan dari algoritma ini adalah untuk mendapatkan nilai kesepakatan global antar node. Ada algoritma ini, sejumlah sensor saling bekerja sama untuk melakukan pendeteksian kejadian. Hasil endeteksian tersebut akan diberikan kepada sensor tetangganya kemudian dilakukan penggabungan sehingga diperoleh hasil deteksi yang optimal dari informasi yang ada. Hasil keputusan sensor yang telah dioptimasi tersebut diharapkan dapat memberikan nilai probabilitas error deteksi yang rendah apabila dibandingkan dengan hasil dari satu buah sensor saja. Pada algoritma ini konsensus yang digunakan adalah bahwa data yang diperoleh oleh setiap sensor akan diperbaharui oleh sensor

Terdapat dua macam algoritma yang termasuk algoritma konsensus terdistribusi, algoritma tersebut adalah *Average Network Distribution* (A-ND) dan *Average Network Consensus* (A-NC). Algoritma ANC ini yang akan digunakan dalam tugas akhir ini.

## 2.3 Algoritma ANC(Average Network Consensus)

Pada algoritma ini digunakan rumus dasar algoritma konsensus terdistribusi untuk mendapatkan rata-rata dari data pemilihan sensor ( $x^p(0) = \epsilon R^{(N+1)}$ ) dan langkah selanjutnya sama mengikuti rumus-rumus sebelumnya [2]. Dari rumus yang digunakan maka akan mendapatkan rumus untuk mencari keputusan akhir tiap sensor, sebagai berikut:

$$x^p(i+1) = x^p(i) - \alpha(L_x P(i) + n^p(i)), 0 \leq i \leq p; 0 \leq p \leq \rho; x^p(0) = x(0) \dots \dots \dots (1)$$

Dimana  $X^p(i)$  adalah hasil keputusan sensor pada iterasi ke- $i$  dan telah dijalankan selama  $p$  kali dan  $X^p(i+1)$  adalah hasil keputusan sensor pada iterasi ke  $i+1$  yang telah dijalankan selama  $p$  kali.  $L$  adalah jumlah iterasi yang harus dilakukan dan  $p$  adalah jumlah run yang harus dilakukan, rumus tersebut dapat disederhanakan sebagai berikut:

$$x^p(i+1) = Wx^p(i) + x^p(i), 0 \leq i \leq p; 0 \leq p \leq \rho; x^p(0) = x(0) \dots \dots \dots (2)$$

Dimana:

$$W = I - \alpha L, x^p(i) = -\alpha n^p(i), 0 \leq i \leq p; 0 \leq p \leq \rho;$$

Dalam Algoritma ANC ini menggunakan beberapa asumsi sebagai berikut

- a Menggunakan jaringan yang statik  
Nilai Laplacian  $L$  yang dihasilkan dari penyebaran node pada daerah tertentu harus sama dan memenuhi persyaratan bahwa
- b Menggunakan noise Gaussian
- c Menggunakan bobot yang konstan  
Besarnya nilai beban pada algoritma ini adalah konstan

## 2.4 OMNeT++ dan Castalia Simulator

Pada proyek akhir ini akan digunakan sebuah simulator yang diperuntukkan khusus untuk jaringan sensor nirkabel yaitu Castalia. Castalia merupakan simulator berbasis OMNeT++.

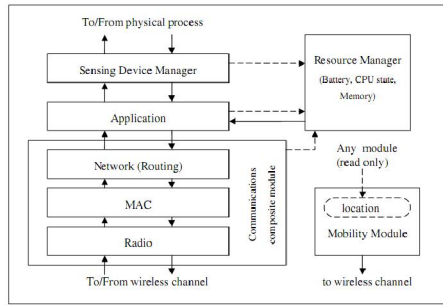
### 2.5.1 OMNeT++.

OMNeT++ adalah sebuah *framework* simulasi event-diskrit. OMNeT++ telah dikembangkan oleh Andrés Varga. Area aplikasi utamanya adalah simulasi jaringan komunikasi, tetapi karena arsitekturnya yang generik dan fleksibel, sehingga dapat digunakan di bidang lain seperti simulasi kompleks sistem TI, antrian jaringan atau perangkat keras arsitektur juga.

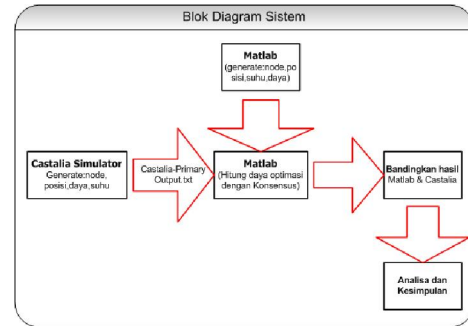
OMNeT++ menyediakan arsitektur komponen untuk model. Komponen (*modul*) yang diprogram dalam C++, kemudian dirakit menjadi komponen yang lebih besar dan model menggunakan bahasa tingkat tinggi (*NED*). OMNeT++ memiliki dukungan luas untuk GUI(Graphic User Interface), dan karena arsitektur yang modular, simulasi kernel dan model yang dapat dimasukkan dengan mudah ke aplikasi

### 2.5.2 Castalia Simulator

Castalia adalah simulator khusus untuk *Wireless Sensor Networks* (WSN), *Body Area Network* (BAN) dan umumnya digunakan oleh jaringan berdaya rendah pada perangkat embedded [3] Hal ini didasarkan pada platform OMNeT++ dan dapat digunakan oleh para peneliti dan pengembang yang ingin menguji algoritma terdistribusi atau protokol di saluran nirkabel secara realistis dan model radio, dengan perilaku node yang realistis terutama berkaitan dengan akses radio.



Gambar 1. Struktur Castalia



Gambar 2 Blok Diagram Jalannya Simulasi

### 3 METODOLOGI

#### 3.1 Perencanaan Sistem

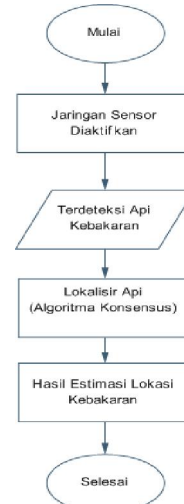
Perancangan sistem didasarkan pada perencanaan area simulasi, jumlah node sensor, dan sistem komunikasi antar node dengan *sink*. Area simulasi ditetapkan sebagai luasan yang didefinisikan sebagai sumbu x dan sumbu y. Jumlah node yang telah ditentukan kemudian diletakkan atau diposisikan dengan aturan yang telah ditentukan, dimana dalam perancangan ini sensor diposisikan dengan atran grid.

Perancangan simulasi jaringan sensor nirkabel ini dengan skenario sebagai berikut:

- Luas area 100 m X 100m
- Jumlah node 100 node
- Penyebaran sensor secara uniform
- Sumber energi menggunakan 2 baterai ukuran AA.
- Sensor yang digunakan berupa sensor temperatur
- Kanal wireless yang digunakan adalah WChannel\_Realistic.
- Radio / sensor node yang digunakan adalah TelosB CC2420
- Layer MAC yang digunakan adalah tunable MAC
- Aplikasi yang digunakan adalah valuePropagation

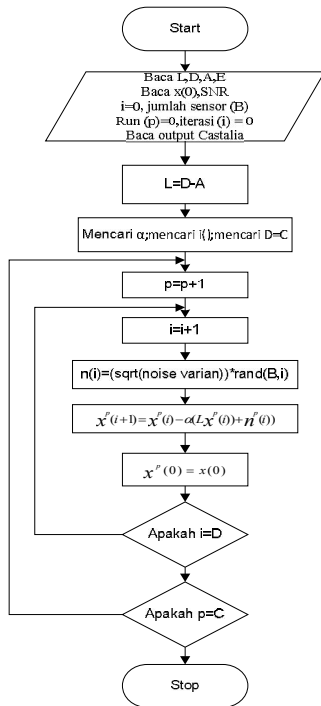
Parameter-parameter simulasi seperti diatas dimodelkan pada program. Inialisasi node, proses pengiriman data dari node ke node lain dilakukan pada *application module* yang digunakan. Output simulator berupa file text yang kemudian dilakukan parsing data untuk mendapatkan nilai suhu dan energi. Kemudian hasilnya ditampilkan dengan Matlab.

#### Flowchart Sistem



Gambar 3. Flowchart Sistem

Berdasarkan flowchart diatas, mula-mula jaringan sensor diaktifkan ,kemudian sensor melakukan proses menyensing kondisi disekitarnya, apakah ada perubahan suhu yang menandakan adanya api. Setelah itu algoritma konsensus mulai bekerja, dengan melakukan kesepakatan sensor mana yang menjadi wakil dalam jaringan tersebut untuk mengirim data lokasi sensor ke bagian *sink*. Data tentang lokasi sensor tersebut kemudian ditampilkan/divisualisasikan dan proses selesai. Berikut ini adalah flowchart algoritma *Average Network Consensus* :



Gambar 4. Flowchart Algoritma ANC

Flowchart diatas menggambarkan tentang proses yang terjadi pada Algoritma Konsensus terdistribusi menggunakan Algoritma ANC. Proses yang terjadi pada algoritma ini yaitu saat sistem dijalankan, maka akan dicari sensor yang aktif, kemudian diambil data dari sensor aktif tersebut. Selanjutnya terjadi proses pencarian konektivitas antar sensor yang aktif, lalu dilanjutkan dengan proses pembaharuan data dengan algoritma ANC sehingga didapat keputusan bersama antar node.

### 3.2 Pembuatan Simulasi dengan Castalia Simulator

Pada tugas ini digunakan tool Castalia sebagai WSN simulator. Untuk melakukan simulasi dengan Castalia, minimal dibutuhkan 3 file yaitu *omnetpp.ini*, *running application* dan *node\_locations.ini* pada folder simulasi. Selain itu juga harus membuat *application module* sesuai dengan simulasi yang dibuat. Application module ini menggunakan bahasa C++. Format output dari Castalia ditentukan disini. File lainnya yang diperlukan adalah file untuk mendeskripsikan jaringan yaitu file dengan ekstensi *.ned*. Saat dilakukan running simulasi, file yang dieksekusi memanggil program *CastaliaBin* lalu memanggil file parameter yang perlu diikuti kemudian menampilkan outputnya pada file text. Hasil simulasi ada 2 macam yaitu hasil output

Castalia dan hasil proses dengan Matlab. Berikut ini contoh dari hasil simulasi:

#### • Hasil Simulasi dengan Castalia

```

OMNeT++/OMNEST Discrete Event Simulation (C)
1992-2005 Andras Varga Release: 3.3p1, edition:
Academic Public License.
See the license for distribution terms and warranty
disclaimer
Setting up Cmdenv...
Processing listfile: /home/rudi/Castalia-
2.3b/Simulations/valuePropagation/././nedfiles.lst
a
Loading NED file: ./src/Node/Node.ned
Loading NED file:
./src/Node/Application/throughputTest/throughputTest
_ApplicationModule.ned
Loading NED file:
./src/Node/Application/ApplicationGenericModule.ned
.....
.....
Preparing for Run #1...
Setting up network 'SN'...
Initializing...
Running simulation...
** Event #0 T=0.0000000 (0.00s) Elapsed: 0.000s
(0m 00s) ev/sec=0
<!-- Simulation time limit reached -- simulation
stopped. b
Calling finish() at end of Run #1...
Temperatur Node[0] = 40.163
Daya Output Node[0] = 0.915913
Koordinat X Node[0] = 77.1321
Koordinat Y Node[0] = 29.8761
.....
.....
Simulation's parameter settings were
[General]
preload-ned-files = *.ned @././nedfiles.lst
network = SN
c
debug-on-errors = true
num-rngs = 11
# 11 random number streams (or generators as
OMNeT calls them)
SN.node[*].mobilityModuleName =
"NoMobilityModule"
.....

```

Gambar 5 Contoh output simulasi dengan Castalia

Keterangan:

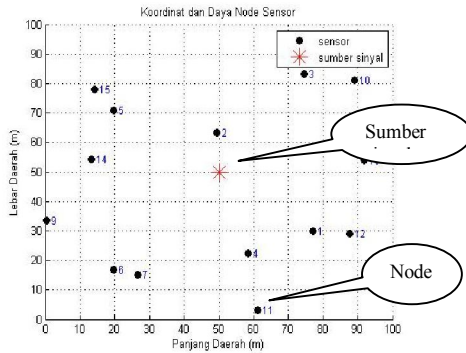
- a Modul-modul yang diperlukan
- b Tampilan hasil output simulasi
- c Tampilan setting parameter yang digunakan

Pada tampilan hasil simulasi tersebut tampak ada 3 bagian informasi yaitu informasi tentang modul-modul yang di-load/ diikuti dalam simulasi. Module yang diikuti berupa \*.ned yang merupakan program untuk mendefinisikan suatu jaringan pada OMNeT++. Nama-nama file \*.ned yang digunakan disimpan pada file *nedfiles.lst*. File ini dipanggil, kemudian semua module \*.ned diikuti

Pada bagian b berupa informasi output dari simulasi yang dijalankan. Informasi yang ditampilkan bisa diatur sesuai dengan data apa yang ingin ditampilkan. Pada simulasi ini yang ditampilkan adalah informasi daya, suhu dan koordinat sensor.

Pada bagian c berupa informasi tentang parameter setting. Parameter ini berada pada file sendiri. Nilai variable pada file tersebut dapat di-set atau di-tuning sesuai model aplikasi yang dijalankan pada simulasi.

### 3.3 Pembuatan Simulasi dengan Matlab



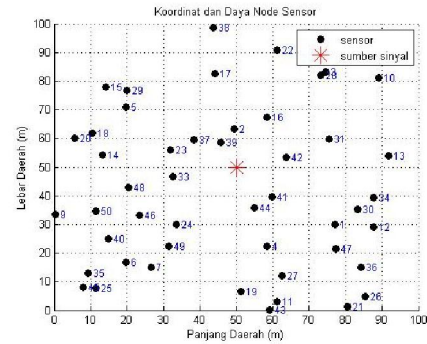
Gambar 4.1 Visualisasi Node Dengan Matlab

Pada gambar diatas terlihat ada node-node sensor dalam suatu area jaringan, jumlah node yang tampil pada visualisasi sesuai dengan jumlah node pada program saat simulasi dilakukan. Di bagian tengah area tersebut terdapat tanda \* yaitu merupakan sumber sinyal. Nilai yang terdapat di dekat node merupakan nilai daya node tersebut.

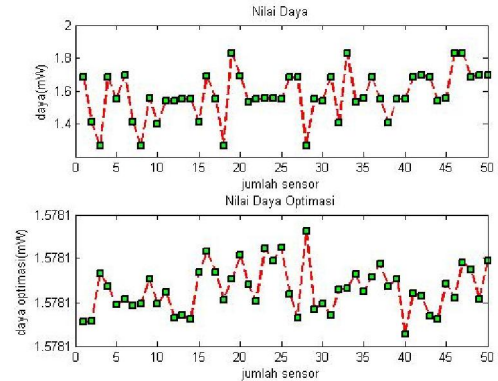
## 4. ANALISA DAN PEMBAHASAN

### 4.1 Pengujian dan Analisa Daya Optimasi Sensor

Pada bagian ini akan membahas tentang pengujian daya optimasi sensor pada jaringan sensor nirkabel. Optimasi disini dimaksudkan untuk memperoleh nilai yang konvergen/ sama atau mendekati sama antar node yang melakukan *sensing*. Nilai optimasi merupakan nilai hasil proses menggunakan Algoritma Konsensus. Nilai daya sebelumnya berasal dari simulator Castalia, diproses kemudian outputnya berupa daya optimasi. Berikut ini adalah gambar *deploy* 50 sensor dan grafik perbandingan daya sebelum dan sesudah optimasi:



Gambar 6 Visualisasi WSN dengan 50 Sensor ter-deploy



Gambar 7 Perbandingan daya sebelum dan sesudah optimasi untuk 50 sensor ter-deploy

Dari grafik diatas terlihat pada grafik daya sebelum optimasi, nilai daya berkisar antara 0-1.8 W, dengan nilai kenaikan dan penurunan yang cukup tajam. Kemudian pada grafik setelah optimasi terlihat nilainya disekitar 1.5 W. Dapat dikatakan saat daya dioptimasi, nilai daya node sensor satu dengan yang lain disinkronkan/disamakan untuk mendapatkan kesepakatan nilai yang akan dikirim ke *sink*.

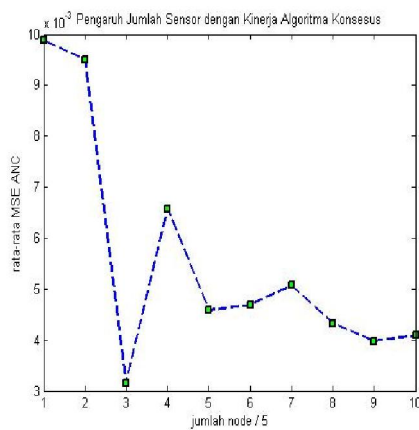
### 4.2 Pengujian dan Analisa Pengaruh Perubahan Jumlah sensor dengan Kinerja Algoritma Konsensus

Berikut ini adalah tabel pengaruh perubahan jumlah sensor dengan rata-rata MSE daya dari 10 kali running :

Tabel 4.1 Tabel Data Rata-rata MSE ANC

No	Jumlah Sensor	Rata-rata MSE ANC (W)
1	5	0.00988
2	10	0.0095
3	15	0.00316
4	20	0.00658
5	25	0.00458
6	30	0.00469
7	35	0.00507
8	40	0.00433
9	45	0.00398
10	50	0.00409

MSE (Mean Square Error ) merupakan salah satu dari banyak cara untuk mengukur perbedaan antara perkiraan dan nilai sebenarnya kuantitas yang diperkirakan. MSE disini merupakan rata-rata kesalahan kuadrat dari seluruh node pada proses pentransmisian data. Pada tugas akhir ini digunakan nilai SNR 30dB [8]. Data rata-rata MSE ANC merupakan nilai rata-rata daya yang diperoleh node pada jaringan sensor. Data tersebut didapatkan dari percobaan dengan mengubah-ubah jumlah sensor lalu disajikan dalam bentuk grafik. Berikut ini adalah gambar grafik pengaruh perubahan jumlah sensor dengan kinerja Algoritma Konsensus :



**Gambar 8** Pengaruh Perubahan Jumlah Node terhadap Kinerja Algoritma Konsensus

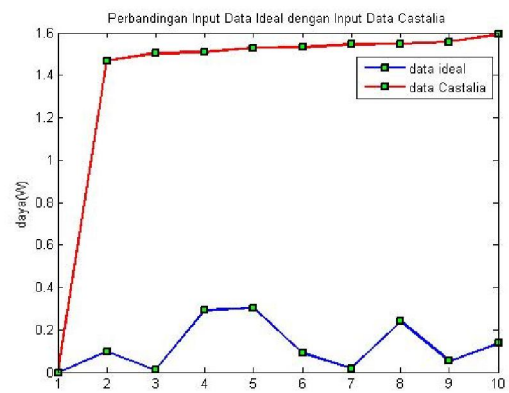
Berdasarkan grafik pengaruh perubahan jumlah sensor terhadap kinerja Algoritma Konsensus diatas, terlihat adanya penurunan nilai MSE saat jumlah sensor semakin banyak. MSE berhubungan dengan nilai SNR yang digunakan saat simulasi. Nilai SNR yang digunakan saat simulasi yaitu 30dB. Pada algoritma ANC semakin tinggi nilai SNR maka nilai nilai MSE semakin menurun.

#### 4.3 Perbandingan Rata-rata MSE dari Input Ideal dengan Input Data dari Castalia

Berikut ini adalah tabel perbandingan rata-rata MSE yang diperoleh dengan input daya dari simulator Castalia dengan daya optimasi yang diperoleh dengan pembuatan simulasi dengan Matlab. Perbandingan ini dilakukan untuk mengetahui kinerja dari simulasi dengan Castalia dengan simulasi yang dilakukan sepenuhnya dengan Matlab.

**Tabel 4.2** Perbandingan Daya antara Input Data dari Castalia dengan Data Ideal

No	Jumlah Node	Perbandingan	
		Data ideal (W)	Data Castalia (W)
1	10	0.0982	1.4689
2	15	0.0097	1.5033
3	20	0.2903	1.5072
4	25	0.3040	1.5282
5	30	0.0900	1.5300
6	35	0.0182	1.5441
7	40	0.2398	1.5473
8	45	0.0532	1.5568
9	50	0.1376	1.5924



**Gambar 9** Grafik Perbandingan Daya dari Input Ideal dengan Input Castalia

Pada grafik diatas terlihat adanya perbedaan antara daya dari data ideal dengan Castalia. Pada simulasi dengan Castalia nilai dayanya lebih tinggi dibanding simulasi ideal. Hal ini disebabkan karena perbedaan setting parameter simulasi yang berbeda. Simulasi dengan Castalia lebih realistis karena parameter setting yang digunakan lebih banyak dan berasal dari pemodelan parameter jaringan sesungguhnya.

## 5 KESIMPULAN

Setelah melakukan serangkaian proses pengujian terhadap system yang telah dibuat, maka dapat diambil kesimpulan bahwa:

- Algoritma ANC pada jaringan sensor nirkabel membantu proses optimasi daya yang diterima sensor agar proses konvergen cepat tercapai.
- Pada grafik perbandingan input data dari Castalia dengan data ideal, nilai daya dari Castalia lebih besar, yaitu 1,533W. Sedangkan pada data ideal bernilai 0,137889W. Perbedaan ini disebabkan oleh setting parameter jaringan yang berbeda.

- Sensor yang menghasilkan daya besar adalah sensor yang dekat dengan sumber api, sehingga dengan membaca output masing-masing sensor diketahui lokasi sumber api yang sebenarnya.

## 6 DAFTAR PUSTAKA

- [1] Sohraby, Kazem. Wireless Sensor Networks : Technology, Protocols, and Applications. New York: Wiley-Interscience. ,2007
- [2] Olfatisaber, "Consensus and Cooperation in Networked Multi-Agent Systems", Proceedings of the IEEE, 95(1):215233, Jan. 2007
- [3] Castalia, <http://castalia.npc.nicta.com.au> (diakses Juli 2010)
- [4] Boulis ,Athanasios , How to Install Castalia, NICTA-Australia 2009
- [5] Boulis ,Athanasios , Castalia A Simulator for Wireless Networks and Body Area Networks User's Manual Ver.2.3, NICTA-Australia 2009
- [6] Khadivi,Alireza ,Hasler,Martin, Fire Detection and Localization using Wireless Sensor Networks, Ecolé Polytechnique Fédérale de Lausanne, Switzerland
- [7] Fahmi,Iqbal Achmad, Algoritma Konsensus Terdistribusi dengan Menggunakan Komunikasi yang Tidak Sempurna pada Jaringan Sensor Nirkabel, Jurusan Teknik Elektro-FTI, Institut Teknologi Sepuluh Nopember, Surabaya 2010