

# OPTIMASI POSISI RELAY NODE PADA WIRELESS SENSOR NETWORK MENGGUNAKAN ALGORITMA GENETIKA

SKRIPSI

Untuk memenuhi Sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Eki Renaldi

NIM: 15515020111289



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2020



## PENGESAHAN

OPTIMASI POSISI RELAY NODE PADA WIRELESS SENSOR NETWORK MENGGUNAKAN  
ALGORITMA GENETIKA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer


Disusun Oleh :  
Eki Renaldi  
155150201111289

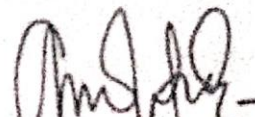
Skripsi ini telah diuji dan dinyatakan lulus pada  
23 Juli 2020

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing 2

  
Ir. Primantara Hari Trisnawan, M.Sc.  
NIP: 19680912 199409 1 003

  
Kasyful Annon, S.T., M.Sc.  
NIP: 19750803 200312 1 003

Mengetahui

Ketua Jurusan Teknik Informatika



Achmad Basuki, S.T., M.MG., Ph.D.  
NIP: 19741118 200312 1 002

# PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 1 Juli 2020

Eki Renaldi

NIM: 15515020111289

## ABSTRAK

### Eki Renaldi, Optimasi Posisi Relay Node Pada Wireless Sensor Network Menggunakan Algoritma Genetika

Pembimbing: Ir. Primantara Hari Trisnawan, M.Sc. dan Kasyful Amron, S.T, M.Sc.

Relay adalah salah satu perangkat yang berperan sangat penting untuk meneruskan data indera menuju *sink*. Oleh karena itu konektivitas dari relay harus dijaga supaya jaringan *Wireless Sensor Network* bekerja secara optimal. Permasalahan umum pada WSN adalah peletakkan relay secara deterministik kurang mendapatkan hasil yang optimal. Perhitungan heuristik dilakukan dengan harapan didapatkan posisi relay yang optimal. Pada penelitian ini akan dilakukan optimasi relay menggunakan metode heuristik. Metode heuristik digunakan karena umumnya heuristik bisa mendapatkan hasil lebih cepat dengan hasil mendekati optimum. Pada penelitian ini diberikan 3 skenario dengan lokasi internet *gateway* dan relay yang statis. Skenario 1 memiliki area 90x77 dengan relay sebanyak 45. Skenario 2 memiliki area 60x77 dengan relay sebanyak 31. Skenario 3 memiliki area 120x77 dengan relay sebanyak 59. Pada penelitian ini akan digunakan algoritma genetika untuk menyelesaikan permasalahan optimasi posisi relay. Algoritma genetika merupakan algoritma yang dapat menemukan solusi mendekati optimum dari kemungkinan solusi yang luas. Pada penelitian ini pindah silang yang digunakan adalah *extended intermediate crossover*, mutasi yang digunakan adalah *random mutation*, dan seleksi yang digunakan adalah *elitism*. Hasil dari penelitian ini adalah konvergensi dini terjadi pada generasi ke-752 untuk skenario 1, generasi ke-846 untuk skenario 2, dan generasi ke-424 untuk skenario 3. Ukuran populasi didapatkan pada ukuran populasi 150 untuk skenario 1, 100 untuk skenario 2 dan 175 untuk skenario 3. Rasio pindah silang didapatkan pada 0,6 untuk skenario 1 dan 0,7 untuk skenario 2 dan 3. Hasil dari penelitian ini adalah optimasi posisi relay menggunakan algoritma genetika tidak sebaik posisi relay secara deterministik karena nilai *fitness* deterministik lebih besar dan waktu yang dibutuhkan algoritma genetika lebih lama.

**Kata kunci:** *Wireless Sensor network*, relay node, algoritme genetika, pindah silang *extended intermediate crossover*, mutasi *random mutation*



# ABSTRACT

**Eki Renaldi, Relay Node Optimization on Wireless Sensor Network Using Genetic Algorithm**

**Supervisors: Ir. Primantara Hari Trisnawan, M.Sc. and Kasyful Amron, S.T, M.Sc**

Relay is one of the most important devices to forward sensory data to sink. Therefore, relay's connectivity must be kept well so Wireless Sensor Network works optimally. Common problem in Wireless Sensor Network is relay node placement in a deterministic way is not getting optimal results. Heuristic way is done with hope that optimal position of relay can be achieved. In this research 3 scenarios with static relay and internet gateway locations are carried out. Scenario 1's area is 90x77 and has 45 relays. Scenario 2's area is 60x77 and has 31 relays. Scenario 3's area is 120x77 and has 59 relays. In this research genetic algorithm is carried out to solve relay's position optimization. Genetic algorithm is an algorithm that can be used to find the nearly optimum solution from wide possibility solution area. In this research, researcher used extended intermediate crossover, random mutation, and elitism selection. The results from this research are early convergence happened at 752<sup>nd</sup> generation for scenario 1, 846<sup>th</sup> generation for scenario 2, and 424<sup>th</sup> generation for scenario 3. Population size for scenario 1 is 150 scenario 2's is 100 and scenario 3's is . Ratio for scenario 1 is 0,6 and for scenario 2 and scenario 3 are 0,7. The result from this research is that relays position optimization using genetic algorithm is not as good as using deterministic because fitness from genetic algorithm is not greater as fitness from deterministic and genetic algorithm needs longer time.

**Keywords:** *Wireless Sensor Network, relay, genetic algorithm, extended intermeiate crossover, random mutation*

# DAFTAR ISI

OPTIMASI POSISI RELAY NODE PADA WIRELESS SENSOR NETWORK MENGUNAKAN ALGORITMA GENETIKA.....	i
PENGESAHAN.....	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS.....	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN.....	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	3
1.3 Rumusan Masalah.....	4
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Batasan Masalah.....	4
1.7 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN.....	6
2.1 Tinjauan Kepustakaan.....	6
2.2 <i>Wireless Sensor Network</i> (WSN).....	14
2.2.1 Node Sensor.....	15
2.2.2 Node Relay.....	15
2.2.3 <i>Internet Gateway</i> .....	15
2.2.4 <i>Client Coverage</i> .....	16
2.2.5 <i>Euclidean Distance</i> .....	16
2.3 Algoritma Genetika.....	17
2.3.1 Inisialisasi Awal.....	17
2.3.2 Pindah Silang.....	18
2.3.3 Mutasi.....	18
2.3.4 Evaluasi.....	18
2.3.5 Seleksi.....	19
BAB 3 METODOLOGI.....	20
3.1 Metodologi Penelitian.....	20
3.2 Studi Literatur.....	21



3.3 Tipe Penelitian .....	21
3.4 Strategi Penelitian .....	22
3.5 Teknik Pengumpulan Data .....	22
3.6 Subjek Penelitian .....	22
3.7 Pemodelan dan Perancangan Metode .....	23
3.7.1 Formulasi Permasalahan .....	23
3.7.2 Alir Perancangan Algoritma .....	24
3.7.3 Perancangan Inisialisasi Awal .....	28
3.7.4 Manualisasi .....	30
3.7.5 Perancangan Pengujian .....	56
3.8 Hasil dan Pengujian .....	58
3.9 Kesimpulan dan Saran .....	58
BAB 4 implementasi .....	60
4.1 Inisialisasi Populasi Awal .....	60
4.2 Perhitungan <i>Fitness</i> .....	62
4.3 Pindah Silang .....	64
4.4 Mutasi .....	64
4.5 Evolusi dan Seleksi .....	65
BAB 5 pengujian dan analisis .....	67
5.1 Pengujian Konvergensi .....	67
5.2 Pengujian Ukuran Populasi .....	72
5.3 Pengujian Rasio .....	76
5.4 Pengujian Akhir & Analisis .....	78
BAB 6 penutup .....	84
6.1 Kesimpulan .....	84
6.2 Saran .....	85
Daftar referensi .....	86
LAMPIRAN A SOURCE CODE .....	91
LAMPIRAN B HASIL PENGUJIAN KONVERGENSI SKENARIO 1 .....	98
LAMPIRAN C HASIL PENGUJIAN KONVERGENSI SKENARIO 2 .....	99
LAMPIRAN D HASIL PENGUJIAN KONVERGENSI SKENARIO 3 .....	100
LAMPIRAN E HASIL PENGUJIAN UKURAN POPULASI SKENARIO 1 .....	101
LAMPIRAN F HASIL PENGUJIAN UKURAN POPULASI SKENARIO 2 .....	102
LAMPIRAN G HASIL PENGUJIAN UKURAN POPULASI SKENARIO 3 .....	103
LAMPIRAN H HASIL PENGUJIAN RASIO SKENARIO 1 .....	104
LAMPIRAN I HASIL PENGUJIAN RASIO SKENARIO 2 .....	105
LAMPIRAN J HASIL PENGUJIAN RASIO SKENARIO 3 .....	106
LAMPIRAN K HASIL PENGUJIAN FINAL .....	107





# DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka .....	12
Tabel 2.2 Tabel Pseudocode Algoritma Genetika .....	17
Tabel 2.3 Tabel representasi Kromosom .....	18
Tabel 3.1 Tabel representasi Kromosom Skenario 1 .....	25
Tabel 3.2 Tabel representasi Kromosom Skenario 2 .....	25
Tabel 3.3 Tabel representasi Kromosom Skenario 3 .....	25
Tabel 3.4 Tabel Keterangan Nilai <i>Fitness</i> .....	26
Tabel 3.5 Tabel Representasi kromosom Manualisasi <i>Parent</i> 1 .....	30
Tabel 3.6 Tabel Representasi kromosom Manualisasi <i>Parent</i> 2 .....	31
Tabel 3.7 Tabel Representasi kromosom Manualisasi <i>Parent</i> 3 .....	31
Tabel 3.8 Tabel Nilai <i>Fitness</i> Relay 1 <i>Parent</i> 1 .....	33
Tabel 3.9 Tabel Nilai <i>Fitness</i> Relay 2 <i>Parent</i> 1 .....	33
Tabel 3.10 Tabel Nilai <i>Fitness</i> Relay 3 <i>Parent</i> 1 .....	33
Tabel 3.11 Tabel Nilai <i>Fitness</i> Relay 4 <i>Parent</i> 1 .....	34
Tabel 3.12 Tabel Nilai <i>Fitness</i> Relay 5 <i>Parent</i> 1 .....	34
Tabel 3.13 Tabel Nilai <i>Fitness</i> Relay 6 <i>Parent</i> 1 .....	34
Tabel 3.14 Tabel Nilai <i>Fitness</i> Relay 7 <i>Parent</i> 1 .....	34
Tabel 3.15 Tabel Nilai <i>Fitness</i> Relay 8 <i>Parent</i> 1 .....	34
Tabel 3.16 Tabel Nilai <i>Fitness</i> Relay 9 <i>Parent</i> 1 .....	34
Tabel 3.17 Tabel Nilai <i>Fitness</i> Relay 10 <i>Parent</i> 1 .....	35
Tabel 3.18 Tabel Nilai <i>Fitness</i> Relay 1 <i>Parent</i> 2 .....	35
Tabel 3.19 Tabel Nilai <i>Fitness</i> Relay 2 <i>Parent</i> 2 .....	35
Tabel 3.20 Tabel Nilai <i>Fitness</i> Relay 3 <i>Parent</i> 2 .....	35
Tabel 3.21 Tabel Nilai <i>Fitness</i> Relay 4 <i>Parent</i> 2 .....	35
Tabel 3.22 Tabel Nilai <i>Fitness</i> Relay 5 <i>Parent</i> 2 .....	35
Tabel 3.23 Tabel Nilai <i>Fitness</i> Relay 6 <i>Parent</i> 2 .....	36
Tabel 3.24 Tabel Nilai <i>Fitness</i> Relay 7 <i>Parent</i> 2 .....	36
Tabel 3.25 Tabel Nilai <i>Fitness</i> Relay 8 <i>Parent</i> 2 .....	36
Tabel 3.26 Tabel Nilai <i>Fitness</i> Relay 9 <i>Parent</i> 2 .....	36
Tabel 3.27 Tabel Nilai <i>Fitness</i> Relay 10 <i>Parent</i> 2 .....	36
Tabel 3.28 Tabel Nilai <i>Fitness</i> Relay 1 <i>Parent</i> 3 .....	36
Tabel 3.29 Tabel Nilai <i>Fitness</i> Relay 2 <i>Parent</i> 3 .....	37
Tabel 3.30 Tabel Nilai <i>Fitness</i> Relay 3 <i>Parent</i> 3 .....	37
Tabel 3.31 Tabel Nilai <i>Fitness</i> Relay 4 <i>Parent</i> 3 .....	37
Tabel 3.32 Tabel Nilai <i>Fitness</i> Relay 5 <i>Parent</i> 3 .....	37
Tabel 3.33 Tabel Nilai <i>Fitness</i> Relay 6 <i>Parent</i> 3 .....	37



Tabel 3.34	Tabel Nilai <i>Fitness</i> Relay 7 <i>Parent</i> 3.....	37
Tabel 3.35	Tabel Nilai <i>Fitness</i> Relay 8 <i>Parent</i> 3.....	38
Tabel 3.36	Tabel Nilai <i>Fitness</i> Relay 9 <i>Parent</i> 3.....	38
Tabel 3.37	Tabel Nilai <i>Fitness</i> Relay 10 <i>Parent</i> 3.....	38
Tabel 3.38	Tabel Kromosom <i>Child</i> 1.....	39
Tabel 3.39	Tabel Kromosom <i>Child</i> 2.....	39
Tabel 3.40	Tabel Nilai <i>Fitness</i> Relay 1 <i>Child</i> 1.....	40
Tabel 3.41	Tabel Nilai <i>Fitness</i> Relay 2 <i>Child</i> 1.....	40
Tabel 3.42	Tabel Nilai <i>Fitness</i> Relay 3 <i>Child</i> 1.....	41
Tabel 3.43	Tabel Nilai <i>Fitness</i> Relay 4 <i>Child</i> 1.....	41
Tabel 3.44	Tabel Nilai <i>Fitness</i> Relay 5 <i>Child</i> 1.....	41
Tabel 3.45	Tabel Nilai <i>Fitness</i> Relay 6 <i>Child</i> 1.....	41
Tabel 3.46	Tabel Nilai <i>Fitness</i> Relay 7 <i>Child</i> 1.....	41
Tabel 3.47	Tabel Nilai <i>Fitness</i> Relay 8 <i>Child</i> 1.....	41
Tabel 3.48	Tabel Nilai <i>Fitness</i> Relay 9 <i>Child</i> 1.....	42
Tabel 3.49	Tabel Nilai <i>Fitness</i> Relay 10 <i>Child</i> 1.....	42
Tabel 3.50	Tabel Nilai <i>Fitness</i> Relay 1 <i>Child</i> 2.....	42
Tabel 3.51	Tabel Nilai <i>Fitness</i> Relay 2 <i>Child</i> 2.....	42
Tabel 3.52	Tabel Nilai <i>Fitness</i> Relay 3 <i>Child</i> 2.....	42
Tabel 3.53	Tabel Nilai <i>Fitness</i> Relay 4 <i>Child</i> 2.....	42
Tabel 3.54	Tabel Nilai <i>Fitness</i> Relay 5 <i>Child</i> 2.....	43
Tabel 3.55	Tabel Nilai <i>Fitness</i> Relay 6 <i>Child</i> 2.....	43
Tabel 3.56	Tabel Nilai <i>Fitness</i> Relay 7 <i>Child</i> 2.....	43
Tabel 3.57	Tabel Nilai <i>Fitness</i> Relay 8 <i>Child</i> 2.....	43
Tabel 3.58	Tabel Nilai <i>Fitness</i> Relay 9 <i>Child</i> 2.....	43
Tabel 3.59	Tabel Nilai <i>Fitness</i> Relay 10 <i>Child</i> 2.....	43
Tabel 3.60	Tabel Kromosom <i>Child</i> 3.....	44
Tabel 3.61	Tabel Nilai <i>Fitness</i> Relay 1 <i>Child</i> 3.....	45
Tabel 3.62	Tabel Nilai <i>Fitness</i> Relay 2 <i>Child</i> 3.....	45
Tabel 3.63	Tabel Nilai <i>Fitness</i> Relay 3 <i>Child</i> 3.....	45
Tabel 3.64	Tabel Nilai <i>Fitness</i> Relay 4 <i>Child</i> 3.....	46
Tabel 3.65	Tabel Nilai <i>Fitness</i> Relay 5 <i>Child</i> 3.....	46
Tabel 3.66	Tabel Nilai <i>Fitness</i> Relay 6 <i>Child</i> 3.....	46
Tabel 3.67	Tabel Nilai <i>Fitness</i> Relay 7 <i>Child</i> 3.....	46
Tabel 3.68	Tabel Nilai <i>Fitness</i> Relay 8 <i>Child</i> 3.....	46
Tabel 3.69	Tabel Nilai <i>Fitness</i> Relay 9 <i>Child</i> 3.....	46
Tabel 3.70	Tabel Nilai <i>Fitness</i> Relay 10 <i>Child</i> 3.....	47
Tabel 3.71	Tabel Seleksi Kromosom .....	47



Tabel 3.72 Tabel Kromosom <i>Parent</i> 1.....	48
Tabel 3.73 Tabel Kromosom <i>Parent</i> 2.....	48
Tabel 3.74 Tabel Kromosom <i>Parent</i> 3.....	48
Tabel 3.75 Tabel Kromosom <i>Child</i> 1.....	48
Tabel 3.76 Tabel Kromosom <i>Child</i> 2.....	49
Tabel 3.77 Tabel Kromosom <i>Child</i> 3.....	49
Tabel 3.78 Tabel Nilai <i>Fitness</i> Relay 1 <i>Child</i> 1.....	51
Tabel 3.79 Tabel Nilai <i>Fitness</i> Relay 2 <i>Child</i> 1.....	51
Tabel 3.80 Tabel Nilai <i>Fitness</i> Relay 3 <i>Child</i> 1.....	51
Tabel 3.81 Tabel Nilai <i>Fitness</i> Relay 4 <i>Child</i> 1.....	52
Tabel 3.82 Tabel Nilai <i>Fitness</i> Relay 5 <i>Child</i> 1.....	52
Tabel 3.83 Tabel Nilai <i>Fitness</i> Relay 6 <i>Child</i> 1.....	52
Tabel 3.84 Tabel Nilai <i>Fitness</i> Relay 7 <i>Child</i> 1.....	52
Tabel 3.85 Tabel Nilai <i>Fitness</i> Relay 8 <i>Child</i> 1.....	52
Tabel 3.86 Tabel Nilai <i>Fitness</i> Relay 9 <i>Child</i> 1.....	52
Tabel 3.87 Tabel Nilai <i>Fitness</i> Relay 10 <i>Child</i> 1.....	53
Tabel 3.88 Tabel Nilai <i>Fitness</i> Relay 1 <i>Child</i> 2.....	53
Tabel 3.89 Tabel Nilai <i>Fitness</i> Relay 2 <i>Child</i> 2.....	53
Tabel 3.90 Tabel Nilai <i>Fitness</i> Relay 3 <i>Child</i> 2.....	53
Tabel 3.91 Tabel Nilai <i>Fitness</i> Relay 4 <i>Child</i> 2.....	53
Tabel 3.92 Tabel Nilai <i>Fitness</i> Relay 5 <i>Child</i> 2.....	53
Tabel 3.93 Tabel Nilai <i>Fitness</i> Relay 6 <i>Child</i> 2.....	54
Tabel 3.94 Tabel Nilai <i>Fitness</i> Relay 7 <i>Child</i> 2.....	54
Tabel 3.95 Tabel Nilai <i>Fitness</i> Relay 8 <i>Child</i> 2.....	54
Tabel 3.96 Tabel Nilai <i>Fitness</i> Relay 9 <i>Child</i> 2.....	54
Tabel 3.97 Tabel Nilai <i>Fitness</i> Relay 10 <i>Child</i> 2.....	54
Tabel 3.98 Tabel Nilai <i>Fitness</i> Relay 1 <i>Child</i> 3.....	54
Tabel 3.99 Tabel Nilai <i>Fitness</i> Relay 2 <i>Child</i> 3.....	55
Tabel 3.100 Tabel Nilai <i>Fitness</i> Relay 3 <i>Child</i> 3.....	55
Tabel 3.101 Tabel Nilai <i>Fitness</i> Relay 4 <i>Child</i> 3.....	55
Tabel 3.102 Tabel Nilai <i>Fitness</i> Relay 5 <i>Child</i> 3.....	55
Tabel 3.103 Tabel Nilai <i>Fitness</i> Relay 6 <i>Child</i> 3.....	55
Tabel 3.104 Tabel Nilai <i>Fitness</i> Relay 7 <i>Child</i> 3.....	55
Tabel 3.105 Tabel Nilai <i>Fitness</i> Relay 8 <i>Child</i> 3.....	56
Tabel 3.106 Tabel Nilai <i>Fitness</i> Relay 9 <i>Child</i> 3.....	56
Tabel 3.107 Tabel Nilai <i>Fitness</i> Relay 10 <i>Child</i> 3.....	56
Tabel 3.108 Tabel Seleksi kromosom.....	56
Tabel 3.109 Tabel Perancangan Pengujian Konvergensi.....	57



Repository Universitas Brawijaya Repository Universitas Brawijaya Repository Universitas Brawijaya Repository Universitas Brawijaya Repository Universitas Brawijaya

Tabel 3.110 Tabel Perancangan Pengujian Ukuran Populasi.....	57
Tabel 3.111 Tabel Perancangan Pengujian Rasio.....	58
Tabel 5.1 Pengujian Konvergensi Skenario 1.....	67
Tabel 5.2 Pengujian Konvergensi Skenario 2.....	68
Tabel 5.3 Pengujian Konvergensi Skenario 3.....	69

# DAFTAR GAMBAR

Gambar 3.1 Gambar Alir Metode Penelitian .....	20
Gambar 3.2 Diagram Alir Algoritma Genetika .....	24
Gambar 3.3 Gambar Diagram Alir Inisialisasi Populasi .....	27
Gambar 3.4 Gambar Rancangan Awal Skenario 1 .....	29
Gambar 3.5 Gambar Rancangan Awal Skenario 2 .....	29
Gambar 3.6 Gambar Rancangan Awal Skenario 3 .....	30
Gambar 3.7 Gambar Pemetaan <i>Parent</i> 1 .....	32
Gambar 3.8 Gambar Pemetaan <i>Parent</i> 2 .....	32
Gambar 3.9 Gambar Pemetaan <i>Parent</i> 3 .....	33
Gambar 3.10 Gambar Pemetaan <i>Child</i> 1 .....	39
Gambar 3.11 Gambar Pemetaan <i>Child</i> 2 .....	40
Gambar 3.12 Gambar Pemetaan <i>Child</i> 3 .....	45
Gambar 3.13 Gambar Pemetaan <i>Child</i> 1 .....	50
Gambar 3.14 Gambar Pemetaan <i>Child</i> 2 .....	50
Gambar 3.15 Gambar Pemetaan <i>Child</i> 3 .....	51
Gambar 3.16 Gambar Alir Pengujian .....	58
Gambar 5.11 Grafik Pengujian Rasio Skenario 2 .....	77
<b>Gambar 5.12 Grafik Pengujian Akhir Skenario 1 .....</b>	<b>79</b>
Gambar 5.13 Grafik Pengujian Akhir Skenario 2 .....	80
Gambar 5.14 Gambar Plotting Relay Skenario 1 .....	81
Gambar 5.15 Gambar Plotting Relay Skenario 2 .....	81
Gambar 5.16 Gambar Plotting relay Skenario 3 .....	82

# DAFTAR LAMPIRAN

<a href="#">LAMPIRAN A SOURCE CODE</a> .....	91
<a href="#">LAMPIRAN B HASIL PENGUJIAN KONVERGENSI SKENARIO 1</a> .....	96
<a href="#">LAMPIRAN C HASIL PENGUJIAN KONVERGENSI SKENARIO 2</a> .....	98
<a href="#">LAMPIRAN D HASIL PENGUJIAN KONVERGENSI SKENARIO 3</a> .....	102
<a href="#">LAMPIRAN E HASIL PENGUJIAN UKURAN POPULASI SKENARIO 1</a> .....	Error! Bookmark not defined.103
<a href="#">LAMPIRAN F HASIL PENGUJIAN UKURAN POPULASI SKENARIO 2</a> .....	Error! Bookmark not defined.104
<a href="#">LAMPIRAN G HASIL PENGUJIAN UKURAN POPULASI SKENARIO 3</a> .....	Error! Bookmark not defined.105
<a href="#">LAMPIRAN H HASIL PENGUJIAN RASIO SKENARIO 1</a> .....	Error! Bookmark not defined.106
<a href="#">LAMPIRAN I HASIL PENGUJIAN RASIO SKENARIO 2</a> .....	Error! Bookmark not defined.107
<a href="#">LAMPIRAN J HASIL PENGUJIAN RASIO SKENARIO 3</a> .....	Error! Bookmark not defined.108
<a href="#">LAMPIRAN K HASIL PENGUJIAN FINAL</a> .....	Error! Bookmark not defined.109



## BAB 1 PENDAHULUAN

Pada BAB 1 Pendahuluan akan dijabarkan beberapa subbab mengenai penelitian yang akan dilakukan. Pada subbab 1.1 akan dijelaskan latar belakang penelitian yang menjelaskan tentang penjelasan umum tentang WSN, permasalahan yang akan diselesaikan, penelitian terdahulu yang menyelesaikan masalah tersebut, algoritma yang bisa digunakan untuk menyelesaikan permasalahan tersebut, dan algoritma yang dipilih untuk menyelesaikan permasalahan penelitian serta alasan kenapa peneliti menggunakan algoritma yang digunakan. Berikutnya pada subbab 1.2 identifikasi masalah akan dipaparkan permasalahan yang akan diselesaikan serta paparan secara ringkas tentang proses dan hasil akhir yang diharapkan akan didapat pada penelitian. Pada subbab berikutnya yaitu subbab 1.2 rumusan masalah akan menjelaskan tentang permasalahan apa yang didapatkan dari latar belakang yang dijawab di kesimpulan. Subbab berikutnya 1.3 adalah tujuan. Subbab 1.3 akan menjelaskan tujuan dari penelitian yang akan dilakukan sementara manfaat dari penelitian akan dijelaskan pada subbab 1.4. Subbab berikutnya adalah Batasan masalah. Pada subbab 1.5 akan menjelaskan lingkup masalah agar titik focus penelitian sesuai dengan masalah penelitian. Terakhir subbab 1.6 sistematika pembahasan. Subbab ini akan menjabarkan gambaran keseluruhan mengenai permasalahan yang akan dibahas pada penelitian ini.

### 1.1 Latar Belakang

Jaringan *Wireless Sensor Network* (WSN) merupakan jaringan nirkabel yang tersusun dari perangkat node relay, node sensor, dan node sink yang terhubung secara ad-hoc komunikasi multi-hop. (Alzaid, 2009) WSN adalah jaringan komputer yang menyebarkan sejumlah node sensor, dikembangkan dan diprogram untuk melakukan peenginderaan terhadap lingkungan sekitar, memanfaatkan parameter pengukuran berupa suhu, tekanan, gerakan, ataurangsangan lainnya. Untuk membangun komunikasi jaringan WSN dibutuhkan 3 perangkat yaitu router, node sensor dan node sink. Router yang berguna untuk komunikasi multi-hop, router tersebut dijadikan node yang bertugas untuk meneruskan paket data dari beberapa node menuju node yang lain. Sensor node adalah node yang dilengkapi dengan sensor untuk mengambil rangsangan fisik dan kondisi dari lingkungan sekitar seperti suara, getaran, suhu, tekanan, gerakan, dan lain-lain. Node sink atau internet *gateway* adalah node yang mengumpulkan dan menyimpan data hasil baca dari node sensor. Node sensor digunakan untuk melakukan mengambil rangsangan fisik dan kondisi dari lingkungan sekitar seperti suara, getaran, suhu, tekanan, gerakan, dan lain-lain.

Pemanfaatan teknologi jaringan WSN memang sangat berguna dibidang pengawasan. Namun tetap saja ada hal yang harus diperhatikan dalam merancang jaringan untuk perangkat *internet of thing* seperti arsitektur dan infrastruktur jaringan agar tidak ada masalah yang membuat kurang efektifnya jalanya teknologi jaringan WSN. Salah satu topik permasalahan pada WSN adalah tentang



*node placement* yang dirasa cukup sulit untuk diselesaikan (Wang, et al., 2013) dan (Yu, et al., 2012). Dikarena sumber daya energi perangkat yang terbatas menyebabkan hal tersebut menjadi sebuah tantangan topik i yang dioptimasi dengan bermacam pendekatan dan perspektif oleh banyak peneliti (Sarwesh, et al., 2015). Masih banyak penerapan WSN yang menggunakan arsitektur *single-tiered*. Arsitektur *single-tiered* memanfaatkan sensor sebagai pengindera lingkungan sekitarnya dan juga sebagai relay yang berperan sebagai penerus data menuju internet *gateway* (Mampentzidou, et al., 2012). Peran dari sensor yang berfungsi sebagai pengindera dan penerus data ini menyebabkan usia baterai lebih cepat habis, sementara aktivitas penginderaan dan pengiriman data indera ke internet *gateway* dibutuhkan dalam jangka waktu yang lama.

Jaringan WSN secara *two-tiered* dapat memberikan konektivitas antar perangkat agar dapat mengirimkan data menuju internet *gateway*. Hal tersebut bisa dilakukan karena adanya pembagian peran oleh relay yang bertugas untuk meneruskan data menuju internet *gateway* (Wang, et al., 2013). Maka dari itu peran relay menjadi sangat penting untuk menjaga konektivitas jaringan, sehingga lokasi penempatannya perlu untuk dipikirkan secara matang. Penempatan relay node menggunakan pendekatan deterministik lebih mudah dilakukan dan lebih nyata (Asheer & Kumar, 2015). Akan tetapi nilai konektivitas yang didapat masih dibawah 30 persen. Maka dari itu strategi penempatan relay secara heuristik bisa dipertimbangkan untuk meningkatkan hasil konektivitas jaringan.

Seiring tenarnya penggunaan teknologi jaringan WSN, seorang *network administrator* harus bisa menrancang arsitektur jaringan WSN dengan benar sehingga teknologi jaringan WSN dapat difungsikan dengan baik. Banyak penelitian mengenai optimasi peletakkan node relay. Penelitian Lin yang meneliti tentang optimasi penempatan node relay yang berdasar kepada letak internet *gateway* dan *quality of service* jaringan WSN (Lin, et al., 2015). Penelitian selanjutnya adalah penelitian yang dilakukan Hamdi tentang optimasi peletakkan node relay yang bisa bergerak dinamis (Hamdi & Mhiri, 2015). Kedua penelitian tersebut menggunakan pendekatan heuristik yaitu menggunakan algoritma *Particle Swarm optimization* (PSO). Akan tetapi kedua penelitian tersebut belum bisa menjawab kebutuhan jumlah relay yang dibutuhkan untuk bisa memastikan seluruh relay terhubung untuk bisa mengirimkan data pindai hingga sampai ke internet *gateway*.

Berdasarkan studi kasus dari konektivitas jaringan yang dirasa merupakan hal yang sangat penting dalam pembangunan arsitektur jaringan WSN, maka pada penelitian ini akan diselesaikan permasalahan *relay node placement* pada WSN dengan arsitektur jaringan *two-tiered*. Tujuan dari penelitian ini adalah merancang strategi heuristik untuk menentukan posisi relay node yang ideal supaya semua relay yang ada pada area implementasi WSN bisa terhubung sampai ke internet *gateway*. Umumnya peletakkan relay dilakukan secara deterministik menggunakan aturan tertentu, namun pada penelitian ini peletakkan secara deterministik akan dioptimasi menggunakan heuristik dengan kalkulasi dari algoritma heuristik. Strategi heuristik dibuat dengan tujuan untuk melakukan





optimasi konektivitas sebuah *planned network* berdasar kepada hasil peletakkan relay menggunakan pendekatan deterministik. Penelitian ini akan menggunakan algoritma genetika sebagai metode heuristik optimasi peletakkan relay, metode heuristik dipilih karena bisa mendapatkan hasil lebih cepat dan mendekati optimum. Algoritma genetika dipilih karena metode meta-heuristik bisa jadi solusi penyelesaian permasalahan diskrit (Sharma, et al., 2015). Selain itu, penelitian sebelumnya tentang *relay node placement* banyak yang menggunakan metode meta-heuristik dan didapatkan hasil yang signifikan (Lin, et al., 2016).

## 1.2 Identifikasi Masalah

Masalah pada penelitian yang akan diselesaikan pada penelitian ini adalah pada seperangkat jaringan terencana WSN. Jaringan WSN berjalan dengan komponen sensor node, relay node, dan internet *gateway*. Jaringan WSN bekerja secara optimal apabila data hasil pindai dari node sensor harus dipastikan sampai menuju internet *gateway* melalui beberapa relay node. Maka dari itu relay node berperan sangat penting dalam pengantaran data dari node sensor menuju internet *gateway*. Pada penelitian ini akan dilakukan optimasi penempatan relay node untuk mendapatkan konektivitas dari relay node seoptimal mungkin. Pada penelitian ini akan ditawarkan solusi permasalahan dengan menggunakan metode meta heuristik yaitu menggunakan algoritma genetika.

Pada penelitian ini akan dibuat sejumlah  $n$  populasi yang berisi gen, gen pada penelitian ini adalah koordinat  $x$  dan  $y$  dari setiap relay. gen atau kromosom memiliki jumlah sebanyak ukuran populasi. Untuk melihat kelayakkan setiap individu tersebut akan dihitung nilai *fitness* setiap individu dengan kriteria konektivitas yang baik. Setelah dilakukan perhitungan *fitness* dari masing-masing individu akan dilakukan tahapan reproduksi yang terdiri dari tahap pindah silang dan mutasi. Pada tahap pindah silang akan dipilih 2 individu sebagai *parent* untuk dilakukan pindah silang yang akan menghasilkan 2 *child* sebagai individu yang baru lalu dihitung nilai *fitness* individu tersebut. Pada tahap mutasi akan dipilih 1 individu sebagai *parent* untuk dilakukan mutasi yang akan menghasilkan 1 *child* sebagai individu baru lalu dihitung nilai *fitness* individu tersebut. Setelah tahapan reproduksi selesai akan dilakukan evaluasi untuk mengurutkan individu dari *fitness* tertinggi hingga terendah. Setelah tahap evaluasi akan dilakukan seleksi untuk memilih individu-individu dengan *fitness* tertinggi untuk dijadikan *parent* pada proses reproduksi pada iterasi berikutnya.

Hasil akhir dari penelitian ini akan mendapatkan sejumlah relay node beserta posisinya yang optimal dengan parameter ukur konektivitas dari relay. Pada generasi yang diinginkan dari perhitungan algoritma genetika akan didapatkan sejumlah individu dengan nilai *fitness* yang dianggap paling layak. Individu inilah yang paling optimal untuk diimplementasikan menjadi WSN. Selanjutnya akan dipetakan gen dari individu terbaik ini dan bisa disimpulkan apakah peletakkan relay pada WSN secara heuristik lebih baik atau tidak dibandingkan dengan secara deterministik.



### 1.3 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya, maka dapat diambil beberapa rumusan masalah yang dapat dijadikan sebagai skripsi yaitu:

1. Bagaimana melakukan implementasi algoritma genetika untuk mengoptimasi posisi relay pada jaringan *Wireless Sensor Network*?
2. Bagaimana kinerja algoritma genetika untuk menyelesaikan permasalahan optimasi posisi relay dengan constraint konektivitas?

### 1.4 Tujuan

Tujuan didakannya penelitian skripsi ini yaitu:

1. Untuk mengembangkan metode heuristik agar didapatkan estimasi koordinat relay untuk menghubungkan seperangkat *planned network* dengan konektivitas maksimum.
2. Mengeksplorasi kemampuan algoritma genetika untuk menyelesaikan permasalahan *relay node placement*.

### 1.5 Manfaat

Manfaat yang diharapkan pada penelitian yang dibuat adalah sebagai berikut:

1. Meningkatkan kinerja WSN setelah didupatkannya posisi relay yang optimal.

### 1.6 Batasan Masalah

Agar penekanan tujuan dalam penelitian ini dapat secara tepat mencapai sasaran, maka diberikan batasan masalah yang akan dibahas yaitu:

1. Algoritma yang digunakan adalah algoritma genetika.
2. Arsitektur jaringan WSN yang digunakan adalah *two-tiered*.
3. Penelitian berfokus pada pemodelan strategi heuristik optimasi peletakkan node relay.
4. Jaringan yang digunakan terencana dengan posisi relay dan internet *gateway* yang terdefinisi di awal dan statis.

### 1.7 Sistematika Pembahasan

Untuk menggambarkan keseluruhan mengenai permasalahan yang akan dibahas pada skripsi ini, sistematika penulisan dibagi menjadi 7 Bab dengan rincian sebagai berikut:

#### BAB I Pendahuluan

Bab ini akan menjelaskan mengenai latar belakang penelitian optimasi relay, kilasan dari penelitian terdahulu dengan topik *node placement*, identifikasi masalah rumusan masalah, tujuan penelitian, manfaat, batasan masalah, dan sistematika pembahasan.



## BAB II Landasan Kepustakaan

Bab ini merangkum referensi dari beberapa jurnal penelitian terdahulu yang bertopik tentang *node placement*, dasar teori dari WSN, dan dasar teori algoritma genetika.

## BAB III Metodologi

Bab ini menguraikan metode penyelesaian yang digunakan untuk menyelesaikan masalah yang telah dirangkum, perancangan tahap optimasi, dan metode pengujian. Perancangan tahapan optimasi dijabarkan dengan merujuk studi literatur dan penelitian terdahulu yang sudah dipaparkan di Bab 2.

## BAB IV Implementasi

Bab ini menjelaskan Langkah dan proses teknis implementasi dari hasil perancangan Bab 3 yang diwujudkan pada sebuah sistem optimasi peletakkan relay. Teknis implementasi akan dijelaskan dalam bentuk potongan pseudocode program.

## BAB V Pengujian

Bab ini menguraikan hasil pengujian dari implementasi optimasi peletakkan relay. Pengujian dilakukan sesuai dengan rancangan skenario yang sudah ditetapkan pada Bab 3 Metodologi. Pembahasan pengujian akan menjawab berbagai pertanyaan yang sudah didefinisikan pada Bab 1 rumusan masalah.

## BAB VI Penutup

Bab ini menguraikan kesimpulan yang diperoleh dari penelitian optimasi peletakkan relay yang berdasarkan hasil dari pengujian pada Bab 5. Bab 6 juga memaparkan saran-saran supaya hasil penelitian ini bisa digunakan untuk pengembangan lebih lanjut.



## BAB 2 LANDASAN KEPUSTAKAAN

Berdasarkan permasalahan penelitian yang sudah dijabarkan, sudah dijelaskan bahwa pada penelitian ini akan meneliti bahasan topik *relay node placement* pada WSN menggunakan algoritma genetika sebagai metode untuk mengoptimasi peletakkan relay. Dasar teori yang digunakan untuk penelitian ini akan dijelaskan pada bab ini dengan masing-masing penjelasannya akan dibagi menjadi 3 subbab. Tinjauan literatur tentang penelitian terdahulu akan dijelaskan pada subbab 2.1. Subbab 2.2 akan menjelaskan tentang dasar teori dari WSN. Dasar teori dari algoritma genetika dan rumus yang akan digunakan pada penelitian ini akan dijelaskan pada subbab 2.3

Tinjauan literatur yang dijelaskan pada subbab 2.1 berisi penelitian terdahulu yang membahas topik *relay node placement* yang menjadi dasar literatur penelitian. Penelitian terdahulu dijadikan sebagai acuan dan alasan melakukan penelitian ini. Ada 4 penelitian terdahulu tentang topik *relay node placement* yang menjadi dasar penelitian ini. Penelitian yang pertama adalah penelitian yang dilakukan oleh Abidin dan Din tahun 2012. Penelitian kedua adalah penelitian yang dilakukan Abidin dan Din tahun 2013. Penelitian ketiga adalah penelitian yang dilakukan Nitesh dan Jana pada tahun 2014. Penelitian keempat adalah penelitian yang dilakukan oleh Widada dan rekan pada tahun 2019. Di akhir setiap penelitian akan dipaparkan relasi dari masing-masing penelitian terdahulu dengan penelitian yang sedang dikerjakan.

Dasar teori dari WSN akan dijelaskan pada subbab 2.2. subbab 2.2 akan menjelaskan pengertian dari WSN, mekanisme kerja WSN, dan juga konsep serta beberapa teori yang mendukung dan digunakan pada penelitian. Penjelasan tentang WSN bisa disimpulkan beberapa landasan yang akan dirumuskan menjadi persamaan. Persamaan yang didapat akan digunakan sebagai dasar perhitungan heuristic untuk mengoptimasi relay.

Subbab 2.3 berisi penjelasan teoritis mengenai algoritma genetika yang digunakan untuk menyelesaikan strategi heuristic. Tujuan penulisan subbab 2.3 adalah untuk memenuhi kebutuhan informasi pengetahuan teori algoritma genetika yang digunakan pada penelitian ini. Penjelasan asal mula algoritma genetika, inisialisasi awal, pindah silang, mutasi, evaluasi, dan seleksi akan dijelaskan pada subbab ini.

### 2.1 Tinjauan Kepustakaan

Banyak penelitian tentang permasalahan optimasi peletakkan relay yang bertujuan untuk mendapatkan konektivitas maksimal. beberapa contoh penelitian tersebut adalah penelitian yang dilakukan Wang yang memaksimalkan konektivitas dengan cara memperpanjang network *lifetime* (Wang, et al., 2013) (Wang, et al., 2008), mengimplementasikan *fault-tolerance* pada jaringan *two-tiered* (Azharudin & Jana, 2015), hingga pembuatan metode perkiraan *connectivity-aware* yang mengkalkulasi kemungkinan sensor mengirim data



menuju internet *gateway* (Ma, et al., 2016). Dapat diambil kesimpulan bahwa faktor utama yang mempengaruhi konektivitas pada *two-tiered* WSN adalah sumber daya perangkat dan jumlah relay yang terhubung dengan sensor dan internet *gateway*. Supaya konektivitas relay pada sebuah perangkat dapat maksimal, perlu dipelajari tentang cakupan dari sensor dan internet *gateway*. Pada topik *relay node placement* posisi fungsi sensor dan internet *gateway* menentukan kinerja jaringan WSN *two-tiered* (Niculescu, 2004). Berikut adalah 4 penelitian yang dijadikan tinjauan kepustakaan pada penelitian optimasi peletakkan relay ini

Penelitian pertama adalah penelitian oleh (Abidin & Din, 2012). Pada penelitian Abidin akan ditentukan posisi peletakkan sejumlah node sensor yang optimal pada jaringan WSN terencana dengan menggunakan pendekatan heuristik. Algoritma yang digunakan pada penelitian Abidin adalah algoritma minimax. Abidin menganggap bahwa permasalahan WSN adalah tentang jumlah sensor yang tetap untuk pemantauan. Dikarenakan jumlah node sensor yang diletakkan terbatas, diperlukannya lokasi peletakkan node sensor yang optimal karena lokasi node sensor dapat mempengaruhi konsumsi energi dan cakupan untuk penginderaan. Sebagai contoh, jarak yang besar diantara setiap node melemahkan komunikasi, menurunkan *throughput*, dan membutuhkan konsumsi energi yang lebih banyak. Sebagai solusinya pada penelitian ini akan menggunakan algoritma minimax untuk menemukan posisi node sensor yang optimal. Skema minimax yang ditujukan adalah menempatkan beberapa node sensor secara acak di area yang dipantau yang terdapat beberapa lokasi pengawasan. Lokasi pengawasan akan diobservasi oleh node sensor terdekat. Tujuan dari skema ini adalah untuk memastikan bahwa beberapa lokasi pengawasan berhasil dicakup oleh sejumlah node sensor yang jumlahnya optimum.

Penelitian ini berdasar pada permasalahan dari (Romoozi, et al., 2010) yang menjelaskan bahwa ada pertukaran antara konsumsi energi dari node dengan cakupan jaringan. Jika jarak antara dua node sensor lebih dekat tapi cakupan menjadi lebih kecil akan menyebabkan energi yang dibutuhkan untuk komunikasi lebih sedikit. Pada penelitian (Ingle & Bawane, 2011) menganggap bahwa penyebaran node yang berdasar *grid* mudah terjadi kesalahan seperti ketidakselarasan dan penempatan yang salah secara acak, jadi dibutuhkan peletakkan sensor yang tepat dan akurat pada titik titik *grid*. Penelitian ini menggunakan diagram Voronoi yang dinamakan Node Network Voronoi (NNV) untuk solusi cakupan WSN yang dikontrol menggunakan parameter tunggal dan menggunakan Edge Network Voronoi (ENV) untuk optimalisasi energi. Penelitian ini diajukan dengan sebuah skema penempatan node sensor berdasar dari model yang dibuat oleh (Ingle & Bawane, 2011) dengan permasalahan lokasi minimax atau permasalahan pusat vertex. (Ogryczak, 1997) menjelaskan minimax sebagai meminimumkan jarak maksimum atau waktu perjalanan dari pelanggan menuju fasilitas terdekat. Model yang dibuat (Ingle & Bawane, 2011) adalah menempatkan fasilitas umum agar jarak dari lokasi manapun menuju fasilitas umum menjadi minimum. Area yang dipantau direpresentasikan sebagai jaringan dimana setiap lokasi merepresentasikan lingkungan.



Percobaan disimulasikan dengan model jaringan dengan luas area pantau berdimensi 100m x 100m yang terdiri dari 16 area pengawasan yang sama besarnya dimana disetiap area pengawasan dilengkapi dengan *access point* yang terletak ditengah dan jumlahnya tidak lebih dari satu node sensor. Setiap lokasi dan node sensor terletak saling berdekatan. Jumlah node sensor yang diimplementasikan bervariasi yang ditentukan menggunakan rumus oleh (Aziz, et al., 2007) dan (Kershner, 1939)

$$\text{Jumlah node sensor} = \frac{A}{3\sqrt{3R_s/2}}$$

Dimana A adalah area pengawasan dan  $R_s$  area penginderaan dari node sensor.

Berdasarkan perhitungan dari rumus tersebut didapatkan jumlah node sensor minimum yang dibutuhkan adalah sekitar 10 buah. Dikarenakan setiap lokasi pengawasan hanya boleh terdapat satu node sensor menjadikan jumlah node sensor yang dipertimbangkan dalam penelitian ini bervariasi antara 10 dan 16. Setiap lokasi pengawasan akan dipantau oleh node sensor yang memiliki jarak Euclidean terdekat dengan *access point* nya dan didalam area penginderaan sensor. Dapat disimpulkan bahwa WSN yang diimplementasikan dengan banyak jumlah node sensor memiliki rasio cakupan yang tinggi dan nilai keseragaman yang rendah. Relasi dari penelitian yang akan dilakukan dengan penelitian Abidin adalah penelitian abidin tidak menggunakan algoritma genetika tapi menggunakan algoritma minimax dan obyek yang diteliti bukan node relay tapi node sensor.

Penelitian kedua adalah penelitian yang dilakukan oleh (Abidin & Din, 2013). Pada penelitian Abidin akan ditentukan posisi peletakkan sejumlah node sensor yang optimal pada arsitektur jaringan WSN terencana dengan menggunakan pendekatan heuristik. Algoritma yang digunakan pada penelitian Abidin adalah algoritma *Territorial Scent Mark Algorithm* TPSMA. Abidin menganggap bahwa salah satu permasalahan yang penting dalam pengimplementasian WSN adalah posisi dari node sensor didalam area pengawasan. Lokasi dari node sensor sangat berpengaruh kepada konsumsi energi, masa operasional, dan cakupan penginderaan. Menurut (Romoozi, et al., 2010) ada yang dikorbankan dari konsumsi energi dan cakupan penginderaan dalam WSN. Cakupan akan semakin besar didapatkan apabila jarak antara dua node sensor dijauhkan. Dengan menjauhkan jarak antara sensor tersebut menyebabkan tingginya konsumsi energi pada saat transmisi data. Peneliti menganggap bahwa cakupan dari node sensor harus dimaksimalkan untuk memastikan bahwa kejadian dan objek yang dimaksudkan dalam area pengawasan bisa terdeteksi. Mengurangi konsumsi energi merupakan hal yang penting karena merepotkannya untuk mengganti baterai perangkat WSN apalagi jika WSN ditempatkan di area terpencil. Dengan alasan itu skema penempatan node sensor dalam WSN harus bisa dipastikan bisa memberi cakupan yang maksimum dan konsumsi energi yang minimum. Penempatan node sensor ini menggunakan teknik optimisasi biologis yang dikenal dengan nama *Territorial Predator Scent Algorithm* (TPSMA).



Penelitian sebelumnya adalah penelitian dari (Abassy, et al., 2010) yang membandingkan kinerja penempatan node sensor dengan syarat konsumsi energi menggunakan penyebaran *random based* dan *plan based*. Hasilnya adalah *random based* membutuhkan banyak energi daripada *plan based*. Pada penelitian (Deyab, et al., 2011) dan (Chakrabarty, et al., 2002) menggunakan *Integer Linear Programming* (ILP) untuk penempatan node sensor guna memenuhi cakupan dari titik pemantauan. Tujuan dari penelitian ini adalah untuk mengurangi total *cost* dari node sensor tanpa mengganggu cakupan dan konektivitas. Penelitian menggubakan TPSMA ini akan dibandingkan dengan penelitian dari (Deyab, et al., 2011) dan (Chakrabarty, et al., 2002) dalam hal cakupan, titik pemantauan, konektivitas, dan konsumsi energi.

*Territorial Predator Scent Algorithm* (TPSMA) memiliki dua fase yang pertama adalah fase marking yaitu disaat pemangsa memeriksa semua sumber makanan di lokasi  $L$  dan menandai lokasi tersebut sebagai lokasi  $L_{\text{marked}}$  yang memiliki jumlah sumber makanan terbesardengan rumus sebagai berikut.

$$\forall x = \{1, 2, \dots, L\} : (R_x / R_x \in \mathbb{Z}^+)$$

$$R_x = F(x)$$

$$L_{\text{marked}} = \max (R_x)$$

Dimana  $R_x$  adalah sumber makanan pada lokasi  $x$ ,  $F(x)$  nilai fungsi objektif,  $L_{\text{marked}}$  adalah lokasi yang ditandai, dan  $\mathbb{Z}^+$  adalah integer positif. Fase yang kedua adalah fase *matching* yaitu adalah fase dimana pemangsa mencari dan menandai lokasi  $L_{\text{marked}}$ . Anggap bahwa pemangsa menemukan  $L_{\text{marked}}$  berdasar dari *Linear Search* yang dapat dilihat dari algoritma berikut.

( $L_{\text{marked}}$  : integer,  $R_1, R_2, \dots, R_L$  : distinct integers)

$i := 1$

**while** ( $i \leq L \cup L_{\text{marked}} \neq R_i$ )

$i := i + 1$

**if**  $i \leq L$

**then**  $location := i$

**else**  $location := 0$

**return**  $location$

Penelitian ini disimulasikan dengan area sebesar 60m x 60m yang dibagi menjadi 144 area pengawasan yang sama besarnya. Rasio cakupan dari WSN akan meningkat dengan meningkatnya jumlah node sensor. Rasio cakupan dengan menggunakan TPSMA lebih tinggi daripada menggunakan ILP. TPSMA bisa memberikan cakupan sebesar 95% dengan hanya menggunakan 10 node sensor dan hanya membutuhkan 14 node sensor untuk mencapai cakupan 100%. TPSMA bisa mendapatkan konektivitas 100% sementara menggunakan ILP dibutuhkan 24 node sensor untuk mencapai 100% konektivitas. TPSMA memiliki penggunaan energi yang rendah tepatnya 53% lebih rendah dibanding ILP. Relasi dari penelitian yang akan dilakukan dengan penelitian oleh Abidin adalah penelitian Abidin tidak



menggunakan algoritma genetika tapi menggunakan algoritma TPSMA yang hasilnya akan dibandingkan dengan menggunakan algoritma ILP dan obyek yang diteliti bukan node relay tapi node sensor.

Penelitian ketiga adalah penelitian yang dilakukan oleh (Nitesh & Jana, 2014). Pada penelitian ini akan ditentukan posisi peletakkan sejumlah relay node yang optimal pada jaringan WSN dengan menggunakan pendekatan heuristik. Algoritma yang digunakan pada penelitian ini adalah algoritma greedy. Pada penelitian ini Nitesh dan Jana menganggap bahwa WSN terdiri dari ratusan node sensor yang tersebar di suatu area untuk memindai fenomena. Node sensor mengambil data pindaian dan mengirimkan data tersebut ke *remote base station* yang disebut sebagai node sink. Namun permasalahannya adalah WSN diimplementasikan dengan node sensor yang beroperasi menggunakan baterai yang kecil dengan daya yang terbatas. Dengan alasan tersebut perlu dilakukannya penelitian untuk penghematan energi dari node sensor. Penggunaan energi dari sensor dan relay berbanding lurus dengan jarak antara pengirim dan penerima yang dianggap sebagai biaya komunikasi yang perlu diminimalkan untuk memperpanjang masa penggunaan jaringan. Pada WSN berbasis *cluster* dan *two-tiered* node sensor dari tingkat arsitektur terbawah membentuk cluster yang terdiri dari pemimpin yang disebut *cluster head* (CH) pada tingkat arsitektur tertinggi node sensor dari cluster ini akan meneruskan data hasil penginderaan ke *cluster head*, lalu akan di kumpulkan dan dipilih jalur untuk data tersebut menuju sink. Hal ini menyebabkan daya pada *cluster head* habis dan mengurangi kinerja jaringan. Solusinya adalah dibutuhkannya node yang bisa menggantikan fungsi *cluster head* yaitu node relay. Node relay bertanggung jawab untuk pengumpulan data dan pengaturan rute data.

Pada penelitian ini membahas permasalahan penempatan node relay dan mengusulkan algoritma greedy untuk penyelesaiannya. Diberikan lokasi dari node sensor, lalu diletakkannya node relay dengan *cost* komunikasi yang minimum agar semua node sensor dalam WSN bisa tercakup semua oleh node relay dan terhubung untuk komunikasi dengan *base station*. Permasalahan pada penelitian ini dinamakan sebagai RNCCP (*Minimum Relay Node Connected Cover Problem*) algoritma yang diusulkan berdasar dari formasi spiral dari seperangkat titik pada penelitian (Chazelle, et al., 1985) dengan kompleksitas waktu adalah  $O(n \log n)$ , dimana  $n$  adalah jumlah titik yang dalam kasus ini adalah node sensor. Penelitian ini hampir sama dengan penelitian (Tang, et al., 2006) hanya saja pada penelitian tersebut permasalahan ini disebut sebagai CRNSC (*Connected Relay Node Single Cover*).

Peneliti beranggapan bahwa jaringan sensor terbentuk dari seperangkat node sensor yang terdistribusi secara seragam dalam daerah persegi. Inti dari algoritma ini adalah untuk membagi seluruh daerah persegi menjadi kotak kecil yang disebut sel dengan Panjang sisi  $l/2r$  dimana  $r$  adalah jarak dari setiap sensor dan  $l$  adalah integer. Jumlah dari node relay yang dibutuhkan untuk mencakup area sel akan dihitung dan akan dijumlahkan untuk mendapatkan jumlah node relay yang bisa mencakup seluruh area pantauan. Algoritma ini hanya bisa digunakan untuk





penyebaran node sensor yang terdistribusi seragam dan tidak bisa digunakan untuk penyebaran node sensor secara acak. Maka dari itu penelitian yang dilakukan ini menggunakan skenario penyebaran node sensor secara acak. Dalam skenario simulasi digunakan total node relay sebagai matrik kinerja. Hasil akan dibandingkan dengan perhitungan solusi optimal menggunakan CPLEX dan CRNSC pada penelitian (Tang, et al., 2006).

Skenario simulasi dianggap mirip dengan penelitian (Tang, et al., 2006) dimana sensor disebar di area seluas 480m x 480m persegi dan jarak komunikasi bervariasi dari 20 sampai 40 dan menjaga jarak komunikasi dari node relay sebesar 2 kali node sensor. Simulasi ini dilakukan dalam 3 kasus pertama 500 node sensor dengan jarak komunikasi 24m, kedua 400 node sensor dengan jarak komunikasi sebesar 30m, dan ketiga 400 node sensor dengan jarak komunikasi sebesar 40m. Rasio kinerja dihitung sebagai rasio hasil keluaran dari algoritma sampai solusi optimal. Hasil yang didapatkan adalah hasil dari algoritma sangat dekat dengan solusi optimal dan lebih baik daripada CRNSC dalam berbagai skenario rentang sensor dan kepadatan. Relasi dari penelitian yang akan dilakukan dengan penelitian Nitesh adalah penelitian Nitesh tidak menggunakan algoritma genetika tapi menggunakan algoritma greedy yang akan dibandingkan dengan algoritma CRNSC namun obyek yang diteliti sama yaitu node relay.

Penelitian keempat adalah penelitian yang dilakukan (Widada, et al., 2019) mengenai RNP (*Relay Node Placement*) pada *two-tiered WSN* akan dilakukan optimasi penempatan relay node menggunakan pendekatan heuristik. Algoritma yang digunakan pada penelitian ini adalah algoritma *Particle Swarm Optimization* (PSO). Peneliti menganggap topik *node placement* merupakan salah satu hal yang sulit diselesaikan. Tujuan dari penelitian ini adalah merancang strategi heuristic untuk menentukan jumlah node relay yang bisa mencakup sekumpulan perangkat berupa node sensor dan internet *gateway*. Penelitian ini menggunakan algoritma PSO (*Particle Swarm Optimization*). Hasilnya adalah penentuan lokasi node relay menggunakan algoritma PSO dinilai mampu untuk mengoptimasi penempatan relay. Untuk mengoptimasi lokasi penempatan relay dibutuhkan sejumlah relay.

Penelitian ini dilakukan mengkaji dari penelitian Deif yang menyelesaikan permasalahan *relay node placement* dengan konteks *planned deployment network* (Deif & Gadallah, 2013). Penelitian Deif menggunakan 4 algoritma yaitu algoritma yaitu *computational geometry artificial potential fields* algoritma genetika, dan *Particle Swarm Optimization* dan didapatkan hasil yang optimal dengan menggunakan algoritma *Particle Swarm Optimization*. Penelitian yang kedua adalah penelitian Lin yang mengoptimasi *relay node placement* dengan tujuan memaksimalkan konektivitas jaringan. Pada penelitian ini menggunakan algoritma *Particle Swarm Optimization* dengan skenario luas area 32 x 32 terdapat 4 internet *gateway*, 16 node relay, dan 48 node sensor, lalu dijalankan sebanyak 20 kali dengan lokasi sensor yang berbeda-beda. Hasil dari penelitian ini adalah metode heuristic menggunakan algoritma *Particle Swarm Optimization* menghasilkan nilai *fitness* yang maksimal dan juga menghasilkan astrigensi global dan kecepatan konvergensi yang lebih cepat.



Penelitian ini dilakukan dengan 3 proses utama, yaitu proses estimasi rentang relay, proses optimasi menggunakan algoritma Particle Swarm optimization dan proses perhitungan konektivitas. Proses estimasi jumlah relay membutuhkan variable jumlah sensor jumlah internet *gateway*, dan jumlah maksimal relay yang mencakup perangkat sekitarnya. Proses estimasi jumlah relay dilakukan untuk mendapatkan nilai minimal relay dan nilai maksimal relay. Selanjutnya dilakukan proses optimasi menggunakan algoritma particle swarm optimization yang berjalan sesuai banyaknya rentan jumlah minimal dan maksimal relay. Selanjutnya dilakukan perhitungan nilai konektivitas jaringan yang berlandaskan penelitian Deif tentang *reliability assessment* yaitu *coverage failure* dan *connectivity failure* (Deif & Gadallah, 2016). Hal yang dipertimbangkan dalam menghitung nilai konektivitas adalah semua perangkat sensor dan internet *gateway* terhubung dengan minimal 2 relay disekitarnya.

Terdapat 4 skenario untuk pengujiannya. Skenario pertama dilakukan di area seluas 500 m<sup>2</sup> dan terdiri dari 30 sensor dan 1 internet *gateway*. Skenario kedua dilakukan di area seluas 300 m<sup>2</sup> dan terdiri dari 50 buah sensor dan 1 internet *gateway*. Skenario ketiga dilakukan di area seluas 500m<sup>2</sup> dan terdiri dari 30 buah sensor dan 1 internet *gateway*. Skenario keempat dilakukan di area seluas 500 m<sup>2</sup> dan terdiri dari 50 buah sensor dan 1 internet *gateway*. Internet *gateway* diletakkan diujung area keempat skenario. Sensor diletakkan secara acak dan grid. Nilai konektivitas dinilai dari set posisi dan radius cakupan komunikasi dari perangkat dan akan dibandingkan hasilnya dari heuristik dan deterministik.

Kesimpulannya pada penelitian *relay node placement* menggunakan metode heuristik dengan agoritma particle swarm optimization dinilai bisa untuk mengoptimasi peletakkan relay pada seperangkat jaringan terencana. Untuk mengoptimasi peletakkan relay dibutuhkan beberapa relay yang diiterasi dan dioptimasi lokasi dan dan radius cakupannya. Dari setiap jumlah relay yang dioptimasi, jumlah relay yang dipertimbangkan berengaruh terhadap kompleksitas penyelesaian optimasinya. Kompleksitas penyelesaian akan meningkat 19-35 persen lebih tinggi sesuai banyaknya relay yang dipertimbangkan. Relasi dari penelitian yang akan dilakukan dengan penelitian Widada adalah Penelitian ini tidak menggunakan algoritma genetika tapi menggunakan algoritma Particle Swarm Optimization namun objek yang diteliti sama yaitu node relay. rangkuman dari tinjauan pustaka akan dijabarkan pada Tabel 2.1.

**Tabel 2.1 Tinjauan Pustaka**

No.	Penelitian	Obyek	Hasil	Relasi
1.	(Abidin & Din, 2012)	Penentuan posisi node sensor menggunakan algoritma minimax	Penempatan node sensor berdasar skema algoritma minimax menghasilkan pengimplementasian WSN dengan banyak node sensor memiliki rasio cakupan yang	Penelitian ini tidak menggunakan algoritma genetika tapi menggunakan algoritma



		dengan mengevaluasi cakupan dan masa hidup dari perangkat WSN	lebih tinggi dan nilai keseragaman yang rendah.	minimax, objek yang diteliti bukan node relay tapi node sensor, namun fokus penelitian ini pada cakupan dan masa hidup perangkat WSN
2.	(Abidin & Din, 2013)	Penentuan posisi node sensor WSN menggunakan algoritma TPSM dengan mengevaluasi rasio cakupan, konektivitas, dan masa hidup perangkat WSN lalu akan dibandingkan dengan penelitian lain yang menggunakan ILP	Pengimplementasian WSN dengan skema penempatan node sensor menggunakan TPSMA memiliki rasio cakupan yang lebih tinggi, konektifitas yang penuh, dan energi yang digunakan lebih rendah daripada penelitian Deyab et al yang menggunakan <i>Integer Linear Programming</i> (ILP)	Penelitian ini tidak menggunakan algoritma genetika tapi menggunakan algoritma TPSMA yang hasilnya akan dibandingkan dengan menggunakan algoritma ILP dan objek yang diteliti bukan node relay tapi node sensor focus penelitian sama sama akan mengevaluasi konektivitas
3.	(Nitesh & Jana, 2014)	Penentuan posisi node relay WSN menggunakan algoritma greedy dengan Batasan meminimalka n biaya komunikasi lalu akan dibandingkan dengan penelitian lain yang	Hasil yang didapatkan menggunakan algoritma greedy lebih baik daripada CRNSC karena hasil yang didapat sangat mendekati solusi optimal daripada CRNSC dengan perbedaan scenario jarak sensor dan kepadatan.	Penelitian ini tidak menggunakan algoritma genetika tapi menggunakan algoritma greedy yang akan dibandingkan dengan algoritma CRNSC namun obyek yang diteliti sama yaitu relay node, namun fokus penelitian ini pada biaya



		menggunakan CRNSC		komunikasi yang minimal
4.	(Widada, et al., 2019)	Penentuan posisi dan jumlah node relay menggunakan algoritma Particle Swarm Optimization (PSO)	Optimasi penempatan relay menggunakan algoritma <i>Particle Swarm Optimization</i> (PSO). Dari setiap jumlah relay yang dioptimasi, banyaknya relay yang dipertimbangkan mempengaruhi kompleksitas penyelesaian optimasinya. Kompleksitas optimasi meningkat 19-35 persen lebih tinggi seiring dengan banyaknya relay yang dipertimbangkan	Penelitian ini tidak menggunakan algoritma genetika tapi menggunakan algoritma <i>Particle Swarm Optimization</i> dan objek yang diteliti sama yaitu node relay

## 2.2 Wireless Sensor Network (WSN)

Sesuai dari latar belakang penelitian yang sudah dijelaskan, permasalahan *relay node placement* masih dalam ranah konsep *Wireless Sensor Network (WSN)*. Untuk mempermudah penelitian maka dibutuhkan penjelasan teoritis tentang WSN, komponen yang berpengaruh terhadap kinerja WSN, serta bagaimana WSN diimplementasikan di kehidupan nyata.

Jaringan WSN merupakan teknologi dalam bidang *Ubiquitous Computing* yang berkonsep mengacu kepada sekumpulan alat sensor yang dibentuk dan dikembangkan secara khusus untuk memantau dan merekam keadaan sebuah benda fisik, dalam kondisi statis ataupun dinamis dan mengelola data yang akan dihimpun pada *edge of networks* (M. Viera, et al., 2003). WSNs digunakan untuk mengukur kondisi lingkungan seperti tingkat kelembapan, kesuburan tanah, suara, dan sebagainya (Saharuna, et al., 2012). Jaringan WSN merupakan jaringan nirkabel yang menghubungkan beberapa perangkat seperti router, node sensor, dan sink node yang terhubung secara ad-hoc dengan komunikasi multi-hop. (Alzaid, 2009). Mekanisme kerja WSN hampir sama dengan jaringan ad hoc nirkabel. Hal tersebut dikarenakan perangkat node sensor pada WSN memiliki ketergantungan kepada konektivitas yang bersifat nirkabel, sehingga hasil penginderaan yang direkam sensor dapat terhubung secara nirkabel.

Perangkat dalam WSN mengirimkan data mereka secara kooperatif menuju *edge of network* atau pusat manajemen jaringan yang berupa *storage server, database* dan juga internet *gateway* melalui jaringan nirkabel. seiring berkembangnya penggunaan WSN yang bersifat dua arah yang memungkinkan kontrol aktivitas dan kondisi sensor dilakukan. Pada awalnya perkembangan WSN terbentuk dikarenakan kebutuhan militer untuk mengawasi medan perang. Seiring tingginya kebutuhan manusia akan penggunaan WSN dikembangkanlah



untuk digunakan dibanyak sektor seperti kontrol dan pengawasan proses industri manufaktur, pemantauan kondisi lahan, dan lain sebagainya (M. Viera, et al., 2003).

### 2.2.1 Node Sensor

Node sensor adalah sebuah perangkat yang berfungsi memindai rangsangan dan merekam respon dari suatu kondisi terhadap perubahan kondisi fisik contohnya seperti suhu dan tekanan, lalu data hasil rekaman tersebut akan diteruskan ke server (De La Concepcion, et al., 2014). Node sensor tersusun dari unit penginderaan, unit pemrosesan, unit daya, dan unit komunikasi. Unit penginderaan bertugas untuk mengambil rangsangan dari lingkungan. Unit pemrosesan bertugas untuk mengumpulkan hasil penginderaan dan memproses data hasil penginderaan tersebut dan mengirimkannya ke jaringan. Unit daya bertugas untuk memberi daya untuk kelangsungan kegiatan proses penginderaan dan pengiriman data menuju jaringan. Unit komunikasi bertugas untuk menyiapkan saluran komunikasi jaringan komputer dan komunikasi untuk pemeliharaan dan membangun jaringan WSN yang pada umumnya berbentuk jaringan ad-hoc.

### 2.2.2 Node Relay

Node relay adalah sebuah perangkat yang berperan untuk menentukan rute pengiriman dan mengirimkan data hasil penginderaan sensor menuju internet *gateway* atau node sink melalui jalur multi-hop secara nirkabel. Relay berfungsi untuk mentolerir kesalahan pada jaringan WSN (Bin, et al., 2014) Node relay memiliki tiga bagian tugas operasi komunikasi, yaitu sebagai *radio communication service* sebagai unit yang melayani komunikasi radio untuk mentransmisikan data dan menerima data. Node relay juga bertugas sebagai *information recording service*, unit ini bertugas untuk melayani perekaman informasi dan menyimpan data yang sudah dipindai node sensor. Terakhir tugas dari node relay adalah sebagai *information courier service* unit ini melayani pengantaran data informasi juga sebagai penentu tujuan informasi.

### 2.2.3 Internet Gateway

*Internet gateway* atau *sink node* adalah sebuah perangkat yang ada didalam topologi jaringan komputer yang berperan sebagai penghubung antar komputer didalam topologi jaringan tersebut agar komputer bisa saling bertukar data dan informasi satu dan yang lainnya. Dasarnya internet *gateway* menjembatani koneksi dari bermacam perangkat yang berbeda yang bisa disebut sebagai *converting protocol* dimana dua arsitektur perangkat yang berbeda tapi bisa saling bertukar informasi. Sink adalah sebuah perangkat yang berperan sebagai tujuan akhir dari pendistribusian data (Lin, et al., 2016). Node sink juga bisa diibaratkan sebagai gerbang untuk masuk dan keluar paket data dari komputer sumber menuju komputer tujuan begitu juga sebaliknya.



#### 2.2.4 Client Coverage

Dari pembahasan tentang sensor, relay, dan internet *gateway* pada subbab 2.2.1, 2.2.2, dan 2.2.3, sudah dijelaskan bahwa perangkat WSN tersebut memiliki kemampuan untuk mengirim informasi dan atau menerima informasi. Informasi tersebut terkirim bila ada komunikasi antar perangkat (Hao, et al., 2004). Sensor pada arsitektur *single-tier* memiliki fungsi ganda yaitu sebagai pengindera lingkungan dan juga sebagai relay yang bertugas meneruskan informasi ke internet *gateway*, sehingga alur komunikasi antara sensor dan internet *gateway* terjadi secara langsung. Pada arsitektur WSN *two-tiered* dikarenakan peran mengirimkan informasi menuju internet *gateway* tidak ditugaskan kepada sensor, maka alur komunikasi data dari pangkal hingga ujung secara urut adalah sensor mengambil data penginderaan dikirimkan ke relay, setelah diterima relay akan diteruskan ke sesama relay hingga data tersebut sampai ke internet *gateway*.

Dilihat dari cara kerja arsitektur *two-tiered*, setiap sensor, relay, dan internet *gateway* bisa saling berkomunikasi dikarenakan semua perangkat WSN tersebut berada dalam radius cakupan komunikasi masing masing perangkat. Hal itu disebabkan karena perangkat WSN tersebut saling berdekatan, serta relay yang sanggup melayani komunikasi sensor-sensor tersebut karena sensor-sensor tersebut jumlahnya masih dalam batas kapasitas perangkat yang bisa terhubung dengan relay. Berdasarkan penjelasan pada subbab 2.2.2 tentang node relay, dijelaskan bahwa relay bertugas sebagai *radio communication service*, *information recording service*, dan *information courier service*. Untuk mengetahui kemampuan dari komunikasi relay, perlu dipastikan bahwa seluruh perangkat pada jaringan terencana WSN terhubung dan berkomunikasi dengan satu relay ataupun multi relay. Subbab 2.2.4 akan menjelaskan fungsi *client coverage* ( $\phi_{ij}$ ). *Client coverage* atau keterjangkauan antar 2 perangkat diasumsikan jika sebuah perangkat 1 memasuki radius cakupan komunikasi perangkat  $j$ . Nilai keterjangkauan antar 2 perangkat didapat jika nilai jarak antar 2 perangkat  $i$  dan  $j$  ( $d_{ij}$ ) dan jumlah dari radius komunikasi perangkat  $i$  dan  $j$  (dinotasikan sebagai  $rad_i + rad_j$ ) (Lin, et al., 2015).

$$\phi_{ij} = \begin{cases} 1, & \text{jika } d_{ij} \leq rad_i + rad_j, \\ 0, & \text{sebaliknya} \end{cases}$$

#### 2.2.5 Euclidean Distance

Berdasarkan buku Geometry I tahun 1987, disebutkan bahwa penjumlahan jarak 2 titik pada sebuah *Euclidean space* merupakan pengertian dari *Euclidean distance*. *Euclidean space* ditemukan oleh seorang matematikawan asal Yunani pada tahun 300 SM yang bernama Euclid. Euclid mempelajari hubungan antar sudut dan jarak. *Euclidean* pada topik ini berhubungan dengan teori *phytagoraas* dan dapat digunakan untuk menghitung jarak pada 1 dimensi, 2 dimensi, 3 dimensi bahkan pada konteks lain pada dimensi yang lebih tinggi. Persamaan *Euclidean distance* dirasa masih mampu, relevan dan banyak digunakan pada penelitian



geometri saat ini (Berger, 1987). Persamaan Euclidean digunakan untuk menghitung jarak antar 2 perangkat  $i$  dan  $j$  ( $d_{ij}$ ) dengan  $x$  dan  $y$  adalah titik pada kartesian.

Rumus Euclidean distance yang digunakan adalah

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Sumber: (Anton, 1994)

### 2.3 Algoritma Genetika

Algoritma Genetika adalah algoritma pencarian yang bekerja berdasarkan teknik seleksi alam dan genetika alam. Algoritma Genetika merupakan jenis algoritma evolusi yang paling populer karena kemampuan algoritma ini untuk menyelesaikan berbagai macam permasalahan optimasi yang cukup kompleks. Algoritma Genetika merupakan algoritma *meta-heuristic* yang mampu untuk menemukan solusi dari area probabilitas yang luas agar solusi penyelesaian yang hampir optimum bisa didapatkan. Algoritma Genetika merupakan simulasi dari tahapan evolusi Darwin dan operasi genetika atas kromosom (Tobing, 2010). Untuk menjelaskan bagaimana struktur umum algoritma genetika dengan  $P(t)$  dan  $C(t)$  sebagai populasi (*parents*) dan *offsprings* pada generasi ke- $t$  akan dijelaskan pada Tabel 2.2. (Mahmudy, 2016)

**Tabel 2.2 Tabel Pseudocode Algoritma Genetika**

No	Source code
1.	procedure AlgoritmaGenetika
2.	begin
3.	$t = 0$
4.	inisialisasi $P(t)$
5.	while (bukan kondisi berhenti) do
6.	reproduksi $C(t)$ dari $P(t)$
7.	evaluasi $P(t)$ dan $C(t)$
8.	seleksi $P(t+1)$ dari $P(t)$ dan $C(t)$
9.	$t = t + 1$
10.	end while
11.	end

Sumber: (Mahmudy, 2016)

#### 2.3.1 Inisialisasi Awal

Inisialisasi adalah tahapan yang bertujuan untuk membangkitkan himpunan solusi baru secara acak, solusi ini terdiri dari beberapa untai kromosom dan ditempatkan di populasi. Beberapa hal harus ditentukan dalam tahap ini seperti ukuran populasi (*popSize*) yang merupakan nilai yang menyatakan banyaknya individu/kromosom yang berada dalam populasi. Panjang setiap untai kromosom (*stringLen*) yang dihitung berdasar kepada presisi variable solusi yang dicari (Mahmudy, 2016).



Representasi kromosom yang digunakan pada penelitian ini adalah representasi *real* dengan menyatakan koordinat dari node relay representasi ditampilkan dalam Tabel 2.3.

**Tabel 2.3 Tabel representasi Kromosom**

Kromosom							
R1		R2		...		R(n)	
x	y	x	y	x	y	x	y
1	2	3	4	....	....	5	6

Dari representasi kromosom diatas bisa didapatkan bahwa posisi r1 ada di koordinat (1,2), r2 di koorinat (3,4) sampai relay ke n.

### 2.3.2 Pindah Silang

Pindah silang atau perkawinan silang merupakan tahapan yang bertujuan menambah keberagaman kromosom pada generasi berikutnya berdasar kepada kromosom-kromosom generasi terkini. Pindah silang dalam prakteknya adalah dengan menukar gen dari dua induk degan cara acak. Hasil dari penukaran gen tersebut adalah kromosom baru yang mewarisi sebagian dari sifat kromosom induknya (USU, 2010). Pindah silang yang digunakan dalam penelitian ini adalah *extended intermediate crossover*.

### 2.3.3 Mutasi

Mutasi merupakan contoh dari metode reproduksi. Mutasi adalah sebuah tahap untuk mengubah nilai dari satu atau banyak gen dalam kromosom secara random. Pada tahapan mutasi dilakukan penukaran pasangan gen dalam suatu kromosom yang dipilih secara acak. Proses mutasi menghasilkan kromosom baru dengan melakukan modifikasi terhadap satu atau lebih sifat pada kromosom yang sama. Individu yang sudah melalui tahap seleksi mengalami penyilangan akan menghasilkan individu baru (*offspring*) dan akan dimutasi agar mempercepat terjadinya perbedaan individu didalam populasi. Pada tahap seleksi terjadi peristiwa dimana gen hilang, mutasi bisa menggantikan gen yang hilang tersebut dengan memungkinkan munculnya gen yang tidak ada pada saat inialisasi populasi dan akan meningkatkan variasi populasi (Handayani, 2010). Mutasi yang digunakan dalam penelitian ini adalah mutasi *random mutation*.

### 2.3.4 Evaluasi

*Fitness* adalah nilai ukuran kinerja suatu individu dalam menyelesaikan permasalahan untuk bisa tetap hidup dalam lingkungannya. Evaluasi adalah proses mengulas nilai *fitness* yang dihasilkan oleh setiap kromosom. Evaluasi menggunakan *fitness* guna mendapatkan kromosom yang diinginkan. Untuk mengetahui kromosom yang baik dapat dilakukan perbandingan dari kualitas kromosom, hal ini adalah fungsi dari *fitness* (Ramuna & Mahmudy, 2015). Pada penelitian ini *fitness* berperan penting sebagai penentu dari individu terbaik yang





akan diteruskan untuk proses reproduksi untuk mendapatkan hasil keturunan yang lebih baik lagi yang kromosomnya akan dipetakan sebagai posisi relay yang sudah optimal.

### 2.3.5 Seleksi

Seleksi merupakan tahapan untuk pemilihan individu dari himpunan populasi dan *offspring* yang akan diteruskan ke generasi berikutnya. Individu yang dipilih adalah individu yang kromosomnya memiliki nilai *fitness* yang besar agar generasi berikutnya lebih baik daripada generasi sekarang. Langkah awal pada tahapan seleksi adalah pencarian nilai *fitness*. Setiap individu dalam wadah seleksi akan mendapatkan kemungkinan reproduksi yang dependen dengan nilai objektif individu itu sendiri terhadap nilai objektif dari semua individu yang ada dalam wadah tersebut. *Fitness* yang didapatkan ini yang akan digunakan pada tahap seleksi berikutnya (Lim, 2015).

Salah satu jenis seleksi adalah elitisme. Elitisme merupakan prosedur membuat penggandaan atau salinan dari individu dengan nilai *fitness* yang tinggi. Elitisme menjaga agar individu dengan nilai *fitness* yang tertinggi tidak hilang selama tahapan evolusi. Individu yang hilang disebabkan tahapan seleksi yang dilakukan secara acak sehingga individu dengan nilai *fitness* tinggi tidak selalu terpilih, jika terpilih dan akan dilakukan pindah silang akan menyebabkan nilai *fitness* individu tersebut turun (Sitepu, 2018). Seleksi yang digunakan pada penelitian ini adalah elitism.

## BAB 3 METODOLOGI

Pada Pendahuluan di bab 1 dipaparkan tentang latar belakang penelitian, rumusan masalah dan Batasan penelitian. Kemudian, pada landasan kepusakaan di bab 2 menjelaskan secara detail tentang literatur penelitian yang sudah pernah dilakukan yang memiliki relasi dengan penelitian yang akan dilakukan, dan juga dasar teori keilmuan yang dipakai pada penelitian ini. Selanjutnya diperlukan strategi penelitian dan rumusan metodologi. Pada Bab 3 metodologi penelitian akan dipaparkan strategi serta alur penelitian yang digunakan pada optimasi peletakkan relay.

### 3.1 Metodologi Penelitian

Metodologi penelitian merupakan penjelasan atas tahapan sistematis yang dilakukan peneliti pada penelitian ini. Metodologi penelitian pada penelitian ini terdiri dari studi dan analisis literatur, analisis dan penetapan kebutuhan penelitian, perancangan metode, pengujian perancangan metode hasil pengujian, dan penarikan kesimpulan metodologi penelitian akan dipaparkan strategi serta alur penelitian yang digunakan pada optimasi peletakkan relay. Strategi akan dijabarkan pada Gambar 3.1 berikut.



Gambar 3.1 Gambar Alir Metode Penelitian



### 3.2 Studi Literatur

Studi literatur dilakukan sebelum penelitian dilakukan untuk mencari solusi penyelesaian dan juga metode yang tepat untuk menyelesaikan permasalahan peletakkan relay. Literatur yang dirujuk adalah penelitian dengan topik permasalahan *relay node placement* dari bermacam-macam metode dan kondisi (*unconstrained* dan *constrained*). Hasil analisa penelitian tersebut referensi yang tepat dengan kebutuhan penelitian optimasi peletakkan relay ini dirangkum dan diklasifikasikan. Rangkuman penelitian berguna untuk memperkuat dasar penelitian yang akan dilakukan, dan juga sebagai sasaran dalam menentukan *state of the art* dalam penelitian.

Studi literatur penelitian yang pertama adalah menelaah bermacam-macam penelitian dan literatur tentang WSN. Kajian pada WSN penting dilakukan supaya penelitian ini berjalan sesuai dengan literatur WSN dan penelitian terdahulu. Kajian WSN juga berfungsi sebagai penelaah dasar teori untuk merumuskan metode heuristik peletakkan relay. Hal tersebut merujuk pada tujuan yang didapatkan pada hasil akhir penelitian yaitu adalah menentukan posisi relay yang bisa memberikan konektivitas maksimum.

Selanjutnya adalah referensi terkait penelitian yang menggunakan metode meta-heuristik untuk menyelesaikan permasalahan peletakkan relay. Referensi yang dipelajari dan ditelaah adalah penelitian optimasi peletakan relay dengan algoritma minimax, algoritma *Territorial Predator Scent Algorithm*, algoritma *Particle Swarm Optimization*, dan algoritma greedy. Tujuan mengkaji penelitian yang menggunakan metode meta-heuristik adalah untuk membandingkan dan menentukan metode manakah yang ideal untuk digunakan menyelesaikan permasalahan peletakkan relay.

### 3.3 Tipe Penelitian

Pada penelitian yang akan dilakukan tentang *relay node placement*. Tipe penelitian ini menggunakan tipe penelitian non-implementatif deskriptif, karena algoritma meta-heuristik algoritma genetika ditawarkan sebagai strategi penyelesaian permasalahan peletakkan relay. Penelitian non-implementatif deskriptif adalah penelitian yang menitikberatkan pada penelitian terhadap kondisi tertentu, penelitian yang berdasarkan fenomena kejadian yang dikaji secara keilmuan. Hasil dari penelitian ini adalah investigasi dari fenomena tersebut yang dapat didukung dengan sebuah pemodelan.

Fenomena yang diinvestigasi oleh penelitian adalah permasalahan konektivitas relay pada area WSN. Investigasi yang sudah dilakukan berawal dari berbagai hasil penelitian tentang WSN dengan solusi yang ditawarkan adalah rancangan strategi heuristik peletakkan relay pada jaringan terencana. Pandangan keberhasilan penelitian ini dilihat dari sudut pandang konektivitas relay. Maka dari itu diharapkan dengan dilakukannya optimasi peletakkan relay bisa menghubungkan semua relay pada area jaringan terencana tersebut. Optimasi peletakkan relay



iniilah yang diharapkan bisa menyelesaikan permasalahan konektivitas relay pada area WSN.

### 3.4 Strategi Penelitian

Penelitian akan mengimplementasikan algoritma genetika untuk mendapatkan letak node relay yang paling optimal. Algoritma genetika dipilih karena permasalahan *relay node placement* pada WSN adalah permasalahan diskrit dan dibutuhkan strategi heuristik dan algoritma genetika dipilih untuk menyelesaikan permasalahan letak node relay dalam WSN. Data yang digunakan dalam penelitian ini adalah luas area WSN, Jumlah node relay dan seberapa besar cakupan dari node relay, dan jumlah serta jarak dari sejumlah node relay. Dikarenakan ada sejumlah aturan dari pengimplementasian WSN maka terlebih dahulu dilakukan analisis terhadap data yang ada untuk dianalisis. Dari hasil analisis akan ditentukan pengkodean yang terbaik dengan algoritma genetika.

### 3.5 Teknik Pengumpulan Data

Pada subbab 1.1 disebutkan titik berat pada penelitian ini adalah optimasi peletakkan relay pada jaringan WSN terencana. Agar solusi penyelesaian didapatkan, maka dirumuskan menjadi representasi masalah sesuai alur algoritma genetika pada Tabel 2.2. representasi permasalahan WSN pada penelitian ini berupa 2 perangkat jaringan terencana *two-tiered* yaitu relay dan internet *gateway*. Maka data yang dibutuhkan dalam untuk masalah optimasi ini adalah menenai sifat dasar dan spesifikasi dari relay dan internet *gateway*.

Pada umumnya relay dan internet *gateway* diletakkan pada suatu posisi tertentu. Maka masing-masing dari relay dan internet *gateway* memiliki koordinat posisi 2 dimensi relay ( $x_r$  dan  $y_r$ ) serta radius cakupan komunikasinya ( $rad_r$ ) dan koordinat posisi 2 dimensi internet *gateway* ( $x_{igw}$  dan  $y_{igw}$ ) serta radius cakupan komunikasinya ( $rad_{igw}$ ). Dikarenakan penelitian ini yang hanya sebatas pemodelan strategi, maka nilai dari koordinat posisi dan radius cakupan dari relay dan internet *gateway* disesuaikan dengan kondisi skenario permasalahan dikarenakan jaringan terencana. Koordinat posisi dari relay dan internet *gateway* telah terdefinisi diawal optimasi. Calon koordinat posisi relay ( $x_r$  dan  $y_r$ ) dibangkitkan secara acak dengan rentang batas luas area yang bisa diletakkan perangkat relay (panjang dan lebar). Radius cakupan dari setiap relay bernilai statis.

### 3.6 Subjek Penelitian

Dilihat dari rumusan masalah, subjek atau inti dari pembahasan penelitian ini adalah relay. Merujuk dari subbab 2.4 tentang metode meta-heuristik yaitu algoritma genetika yang digunakan untuk menyelesaikan permasalahan optimasi peletakkan relay, dijelaskan bahwa subjek penelitian relay perlu dimasukkan kedalam representasi masalah kromosom. dikarenakan hasil akhir dari optimasi sebuah jaringan terencana mencakup posisi relay yang optimal. Maka dari itu penelitian ini akan mempertimbangkan posisi optimal relay dengan menjaga konektivitas masing masing relay pada seluruh area pada sebuah jaringan



terencana. Posisi awal relay yang dibutuhkan didapat berdasar dari gambaran menggunakan grid secara deterministic. Hasil perhitungan perkiraan dibutuhkan untuk menemukan lokasi relay yang dibutuhkan algoritma genetika.

### 3.7 Pemodelan dan Perancangan Metode

Pemodelan dilakukan untuk menjelaskan hasil penelitian dari studi literatur menjadi sebuah alur tahapan penyelesaian. Merujuk dari subbab literatur algoritma genetika, ada beberapa poin yang digunakan pada penelitian ini yaitu, rumus mutasi, rumus pindah silang, dan rumus *fitness*. Rumus tersebut akan dipraktikkan pada proses manualisasi. Manualisasi bertujuan untuk simulasi dasar guna mengetahui kesesuaian hipotesis hasil investigasi dengan arah tujuan penelitian. Manualisasi juga berguna untuk mengetahui jika hasil akhir sudah seperti yang diharapkan, yaitu posisi relay terletak seoptimum mungkin dan bisa memberikan konektivitas yang maksimum.

Setelah dilakukan manualisasi tahap penelitian selanjutnya adalah merancang hasil rancangan berdasarkan manualisasi yang telah dilakukan. Pada tahap ini akan membuat kode program dari rumus dan konfigurasi variable hasil manualisasi yang sudah disesuaikan dengan kebutuhan penelitian. Kode program tersebut dikalkulasikan dengan jumlah iterasi yang lebih banyak, sehingga didapatkan hasil optimasi yang sangat mendekati optimal. Hasil dari kalkulasi kode program tersebut akan menghasilkan generasi terbaik dengan komposisi posisi optimal dari relay ( $x_r$  dan  $y_r$ ).

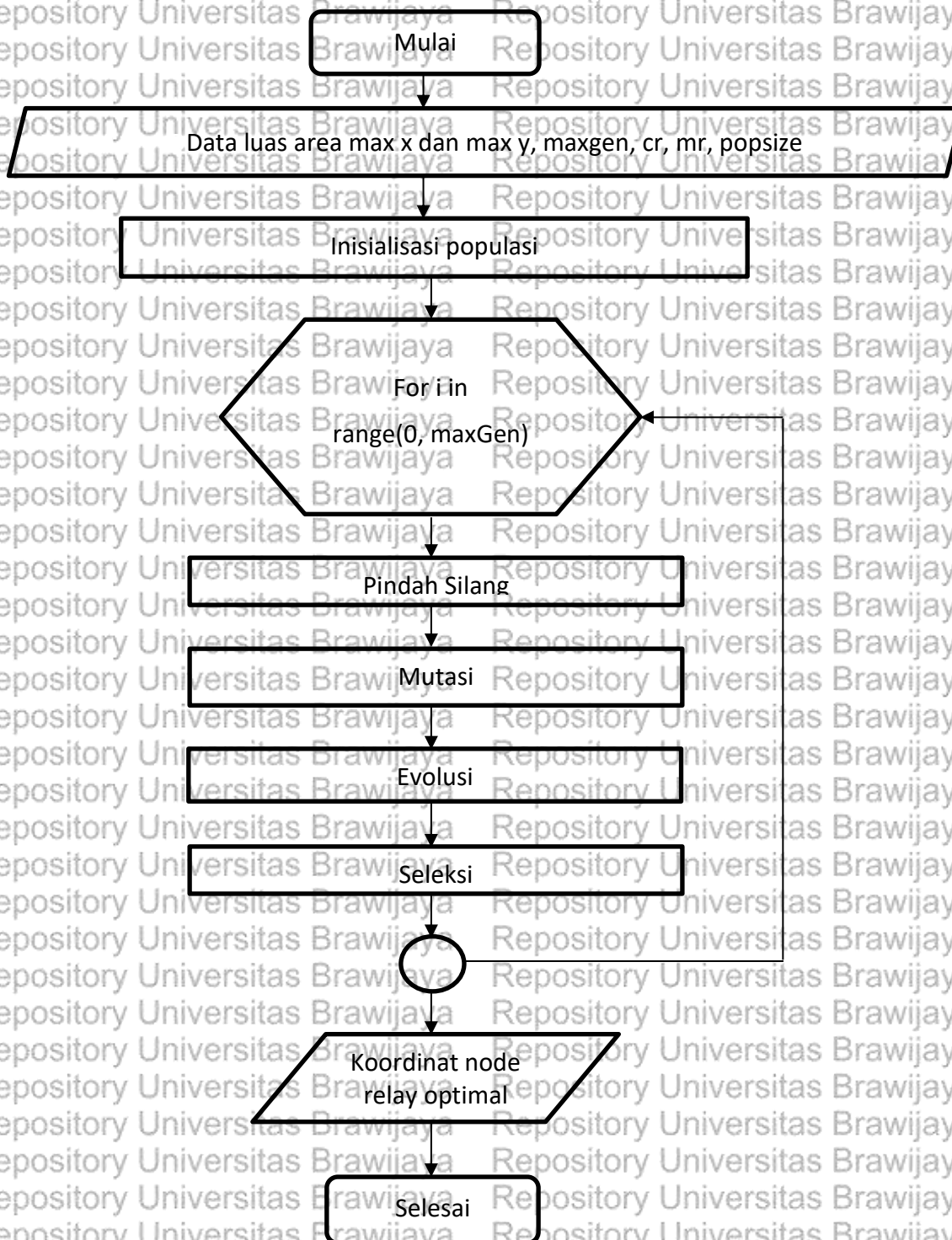
#### 3.7.1 Formulasi Permasalahan

Dilihat dari subbab 1.1 latar belakang penelitian, penelitian ini bermula dari permasalahan konektivitas relay. Permasalahan yang dimaksud adalah dibutuhkannya cakupan yang semaksimal mungkin supaya konektivitas dari WSN tetap optimal. Agar konektivitas jaringan terencana baik maka diperlukan mekanisme terstruktur untuk menemukan posisi relay yang optimal. Menurut teori tentang WSN pada subbab 2.3 komponen pada arsitektur *two-tiered* terdiri dari 3 perangkat dengan fungsinya masing-masing yaitu sensor, relay, dan internet *gateway*. Perangkat WSN tersebut secara nirkabel terhubung sehingga bisa disebar dimana saja selama di area implementasi dan mempunyai radius komunikasi masing-masing. Pada latar belakang dan subjek penelitian sudah disebutkan bahwa inti penelitian ini adalah optimasi posisi peletakkan relay untuk membangun koneksi satu sama lain.

Pengimplementasian jaringan terencana pada penelitian ini adalah, internet *gateway* yang berjumlah satu berikut posisinya ( $x_{igw}$  dan  $y_{igw}$ ) serta radius cakupan komunikasinya ( $rad_{igw}$ ) telah terdefinisi. Dilihat dari Batasan masalah penelitian bahwa posisi internet *gateway* statis dengan konsep jaringan terencana tersebut dan subjek penelitian, maka akan dirancang metode heuristik untuk menemukan posisi peletakkan relay ( $x_r$ ) dan menjamin konektivitas dari antar masing-masing relay.



### 3.7.2 Alir Perancangan Algoritma



**Gambar 3.2 Diagram Alir Algoritma Genetika**

Algoritma Genetika digunakan oleh peneliti untuk mencari posisi peletakkan relay yang optimum pada WSN. Proses yang akan dilakukan untuk menyelesaikan permasalahan optimasi peletakkan node relay pada WSN ditunjukkan dengan



diagram alir diatas. Diagram alir diatas terdiri dari proses analisis data, data *preprocessing*, representasi kromosom, inialisasi populasi, proses mutasi, proses pindah silang, proses seleksi, dan menghitung *fitness*. Diagram alir dapat dilihat pada Gambar 3.2.

### 3.7.2.1 Representasi Kromosom

Pada penelitian ini kromosom akan direpresentasikan menggunakan representasi permutasi. Kromosom akan terdiri dari beberapa gen yang berisi koordinat x dan y dari node relay yang tersebar di area WSN. Kromosom hanya berisi koordinat x dan y dari node relay karena posisi internet *gateway* statis. Jumlah gen pada penelitian ini adalah 90 untuk skenario 1, 62 untuk skenario 2, dan 118 untuk skenario 3. Berikut representasi kromosom dapat dilihat pada Tabel 3.1, 3.2, dan 3.3.

**Tabel 3.1 Tabel representasi Kromosom Skenario 1**

Relay 1		Relay 2		Relay 3		...	Relay 90	
15	20	30	10	45	20	...	...	165 160

**Tabel 3.2 Tabel representasi Kromosom Skenario 2**

Relay 1		Relay 2		Relay 3		...	Relay 62	
15	20	30	10	45	20	...	...	165 160

**Tabel 3.3 Tabel representasi Kromosom Skenario 3**

Relay 1		Relay 2		Relay 3		...	Relay 118	
15	20	30	10	45	20	...	...	165 160

### 3.7.2.2 Menghitung *Fitness*

*Fitness* digunakan untuk menilai kecocokan kromosom sebagai solusi optimasi peletakan relay. Cara untuk memeriksa kecocokan ini dilakukan dengan 2 aturan *fitness*, yang pertama jika *fitness* yang dihasilkan akan bernilai mendekati 0 jika jumlah relay yang saling terhubung dan yang terhubung secara langsung dengan internet *gateway* sedikit. Dan *fitness* mendekati nilai konstanta jika jumlah relay yang saling terhubung dan yang terhubung secara langsung dengan internet *gateway* besar. Rumus ini berlaku dengan asumsi radius komunikasi setiap node relay tetap dan sama. Pada perhitungan *fitness* relay yang terdekat dengan internet gateway akan dihitung jaraknya mengikuti aturan rumus *fitness*. Relay yang terhubung sesuai aturan *fitness* menjadi *tier* pertama. Relay lain akan dihitung jaraknya dengan *tier* pertama yang akan menjadi *tier* kedua dan seterusnya. Rumus *fitness* yang kedua adalah jika jarak antar relay lebih dari 0,8 radius komunikasi akan bernilai 1 jika kurang dari 0,8 radius komunikasi maka akan



bernilai 0. Aturan *fitness* kedua ini digunakan agar relay tidak tersebar secara menumpuk. Hasil dari kedua aturan *fitness* tersebut akan dinormalisasi dengan nilai maksimal adalah 2. Rumus fungsi *fitness* yang digunakan pada penelitian ini akan ditampilkan pada Tabel 3.4

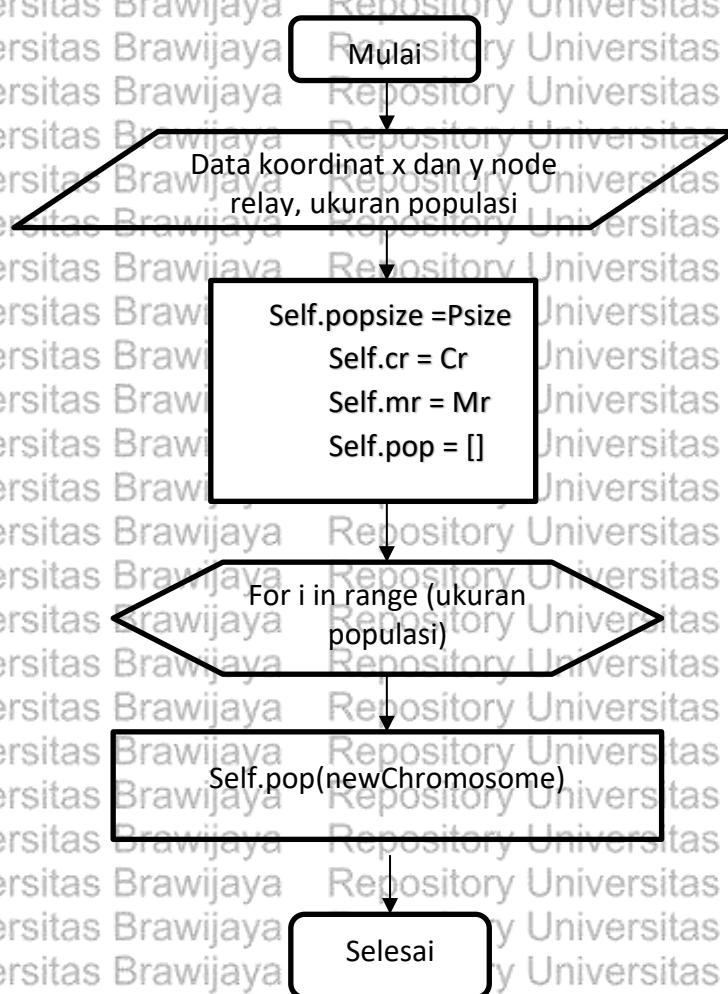
**Tabel 3.4 Tabel Keterangan Nilai *Fitness***

No	keterangan	Nilai
1.	Jika suatu relay terhubung langsung/berada dalam cakupan internet <i>gateway</i>	1
2.	Jika jarak antar relay dan internet <i>gateway</i> < radius relay	$\frac{\text{jarak antar relay dan igw}}{\text{panjang radius cakupan relay}}$
3.	Jika jarak antar relay dan internet <i>gateway</i> > radius relay	0
4.	Jika jarak antara dua relay = panjang radius cakupan relay	1
5.	Jika jarak antara dua relay < panjang radius cakupan relay	$\frac{\text{jarak antar relay}}{\text{panjang radius cakupan relay}}$
6.	Jika jarak antara dua relay > panjang radius cakupan relay	0
7.	Jika jarak antara dua perangkat < 0,8 radius komunikasi	0
8.	Jika jarak antara dua perangkat > 0,8 radius komunikasi	1

### 3.7.2.3 Inisialisasi Populasi

Inisialisasi populasi merupakan tahapan untuk membuat populasi. Populasi merupakan kumpulan kromosom. ukuran populasi memiliki jumlah yang sama dengan jumlah kromosom pada populasi awal. Inisialisasi populasi dapat dilihat pada diagram alir pada Gambar 3.3.





**Gambar 3.3 Gambar Diagram Alir Inisialisasi Populasi**

#### 3.7.2.4 Proses Pindah Silang

Pindah silang yang digunakan pada penelitian ini adalah extended intermediate crossover. Pindah silang dengan metode intermediate extended crossover dilakukan dengan cara memilih dua induk secara acak yang kemudian akan membangkitkan nilai  $a$  sebanyak panjang kromosom tersebut dengan rumus persamaan dibawah (Mahmudy, 2016).

$$C1 = P1 + a \times (P2 - P1)$$

$$C2 = P2 + a \times (P1 - P2)$$

$$C1 = \text{child1}$$

$$C2 = \text{child2}$$

$$P1 = \text{parent1}$$

$$P2 = \text{parent2}$$

$$a = \text{bilangan random}$$



### 3.7.2.5 Proses Mutasi

Teknik mutasi yang digunakan pada penelitian ini adalah *random mutation*. Mutasi secara *random mutation* dilakukan dengan cara menambah atau mengurangi nilai gen yang terpilih dengan angka acak. Rumus mutasi dapat dilihat dari persamaan dibawah (Mahmudy, 2016).

$$C = P - (a(x_{\max} \text{ atau } y_{\max} - x_{\min} \text{ atau } y_{\min}))$$

$$C = \text{child}$$

$$P = \text{parent}$$

a = angka random

$x_{\max}$  atau  $y_{\max}$  = titik x dan/atau y maksimal kartesian

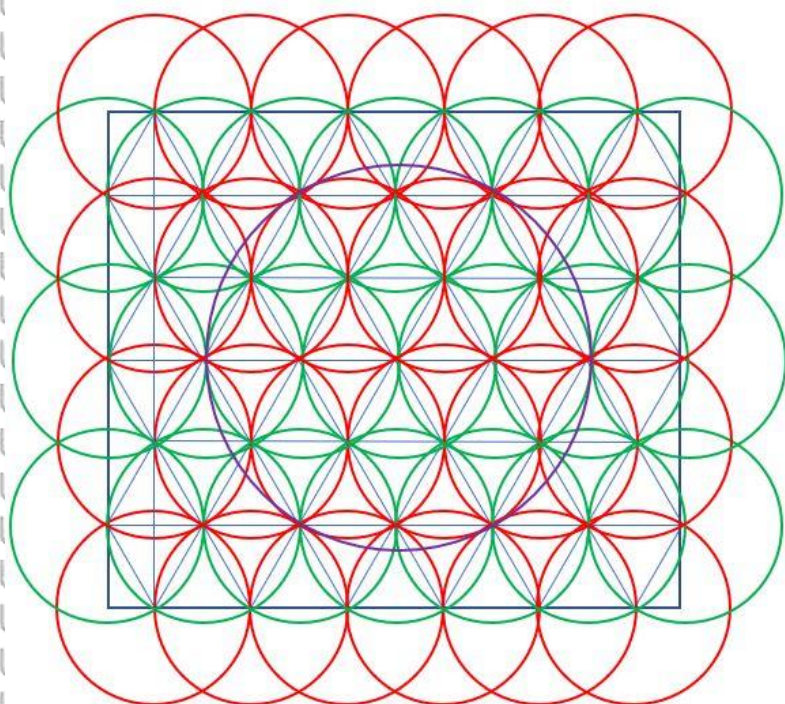
$x_{\min}$  atau  $y_{\min}$  = titik x dan/atau y minimal kartesian

### 3.7.2.6 Proses Seleksi

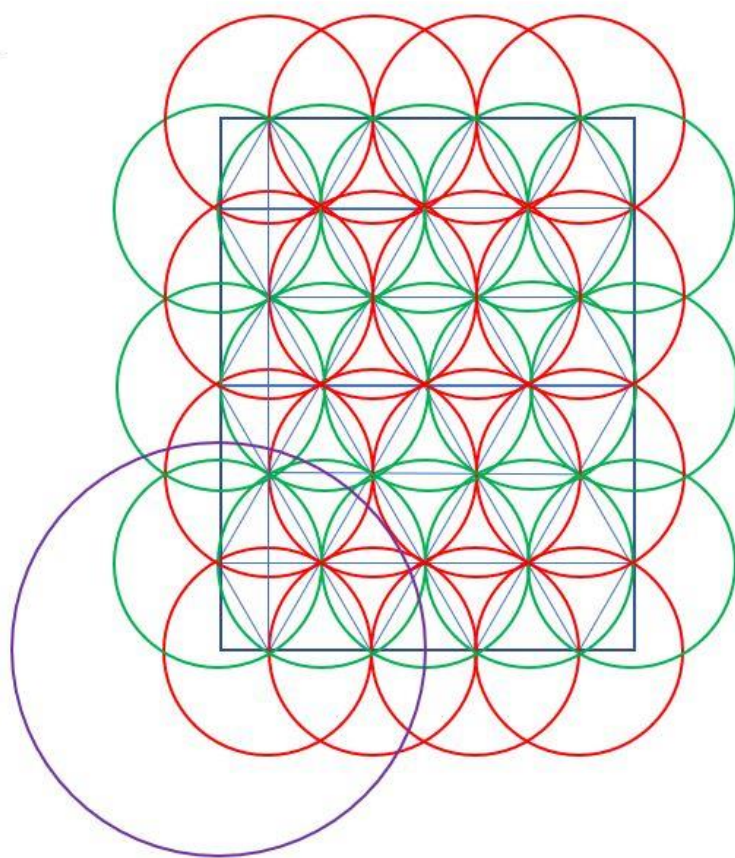
Merujuk dari subbab 2.4.5 tahap seleksi adalah tahapan untuk pemilihan kromosom dari sekumpulan populasi dan *offspring* yang akan diteruskan ke generasi selanjutnya. Seleksi dilakukan setelah evaluasi dengan mempertimbangkan *fitness*. Individu dengan nilai *fitness* yang besar akan dipilih supaya generasi selanjutnya akan lebih baik daripada generasi sekarang. *Elitism* adalah Teknik seleksi yang digunakan pada penelitian ini (Mahmudy, 2016).

### 3.7.3 Perancangan Inisialisasi Awal

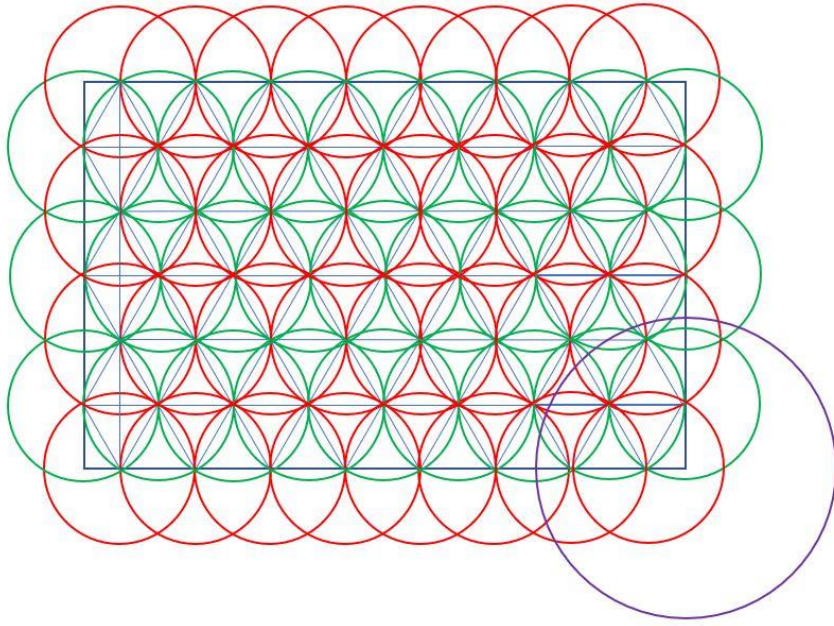
Optimasi penempatan node relay pada WSN adalah objek yang akan diteliti pada penelitian ini. Perancangan inisialisasi awal optimasi peletakkan relay pada WSN yang pertama adalah area WSN. Area WSN pada penelitian ini terdiri dari 3 skenario. Skenario 1 berukuran 90x77, skenario 2 berukuran 60x77, dan skenario 3 berukuran 120x77. Pada area tersebut akan dibagi menjadi beberapa grid segitiga sama sisi dengan panjang sisi sebesar 15 dan. Node relay yang digunakan pada penelitian ini memiliki radius cakupan yang berbentuk lingkaran dengan panjang jari-jari sebesar panjang grid segitiga sama sisi yaitu 15 dan luas area cakupan sebesar 706,5m<sup>2</sup> dan diletakkan di setiap sudut grid segitiga sama sisi. Skenario 1 berisi 45 relay dengan internet *gateway* dengan jari-jari 30 yang berada pada koordinat (45, 45). Skenario 2 berisi 31 relay dengan internet *gateway* dengan jari-jari 30 yang berada pada koordinat (0, 0). Skenario 3 berisi 59 relay dengan internet *gateway* dengan jari-jari 30 yang berada pada koordinat (120, 0). Gambaran area implementasi WSN dengan warna merah dan hijau adalah relay dan warna ungu adalah internet gateway dapat dilihat pada Gambar 3.4, 3.5, dan 3.6.



**Gambar 3.4 Gambar Rancangan Awal Skenario 1**



**Gambar 3.5 Gambar Rancangan Awal Skenario 2**



**Gambar 3.6 Gambar Rancangan Awal Skenario 3**

### 3.7.4 Manualisasi

Manualisasi diperlukan agar mempermudah dalam memahami tahapan kerja algoritma dan juga bisa menjadi acuan untuk kode program. Sebelum dilakukan manualisasi akan didefinisikan terlebih dahulu sebuah jaringan terencana dengan luas 10m x 10m, disebar sebanyak 10 relay ( $r_1, r_2, r_3, \dots, r_{10}$ ) dan 1 internet gateway (igw). Radius cakupan dari setiap relay adalah 1m dan radius cakupan dari internet gateway adalah 2m. internet gateway diletakkan dikoordinat (5,5). Tahapan-tahapan pada manualisasi ini adalah inisialisasi populasi, reproduksi, evaluasi, dan seleksi. Reproduksi terdiri dari pindah silang dan mutasi. Inisialisasi akan diberikan representasi kromosom pada manualisasi ini.

#### 3.7.4.1 Representasi Kromosom

Merujuk dari subbab 2.3.1 kromosom merupakan representasi awal dalam algoritma genetika. Kromosom tersusun dari beberapa gen. gen tersebut berdasarkan dari posisi koordinat  $x$  dan  $y$  dari relay ( $x_r$  dan  $y_r$ ). Dan dibuat 3 kromosom ( $parent_1, parent_2, parent_3$ ). Kromosom  $parent_1$  berisi kondisi awal peletakkan relay atau kondisi deterministik. Sementara kromosom  $parent_2$  dan  $parent_3$  adalah posisi acak yang akan digunakan untuk tahapan. Representasi kromosom dapat dilihat pada Tabel 3.5, 3.4 dan 3.5.

**Tabel 3.5 Tabel Representasi kromosom Manualisasi Parent 1**

Kromosom ( $parent_1$ )																			
R1		R2		R3		R4		R5		R6		R7		R8		R9		R10	
x	y	x	y	x	y	x	y	x	y	x	y	x	y	X	y	x	y	X	Y
4	4	5	4	6	4	3	8	6	2	6	3	7	9	6	8	5	7	6	4



**Tabel 3.6 Tabel Representasi kromosom Manualisasi Parent 2**

Kromosom ( <i>parent2</i> )																			
R1		R2		R3		R4		R5		R6		R7		R8		R9		R10	
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
5	5	4	6	6	6	4	4	4	5	6	4	5	5	5	3	5	7	6	5

**Tabel 3.7 Tabel Representasi kromosom Manualisasi Parent 3**

Kromosom ( <i>parent3</i> )																			
R1		R2		R3		R4		R5		R6		R7		R8		R9		R10	
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
5	6	4	5	5	5	8	5	5	1	5	3	3	5	2	5	7	5	6	5

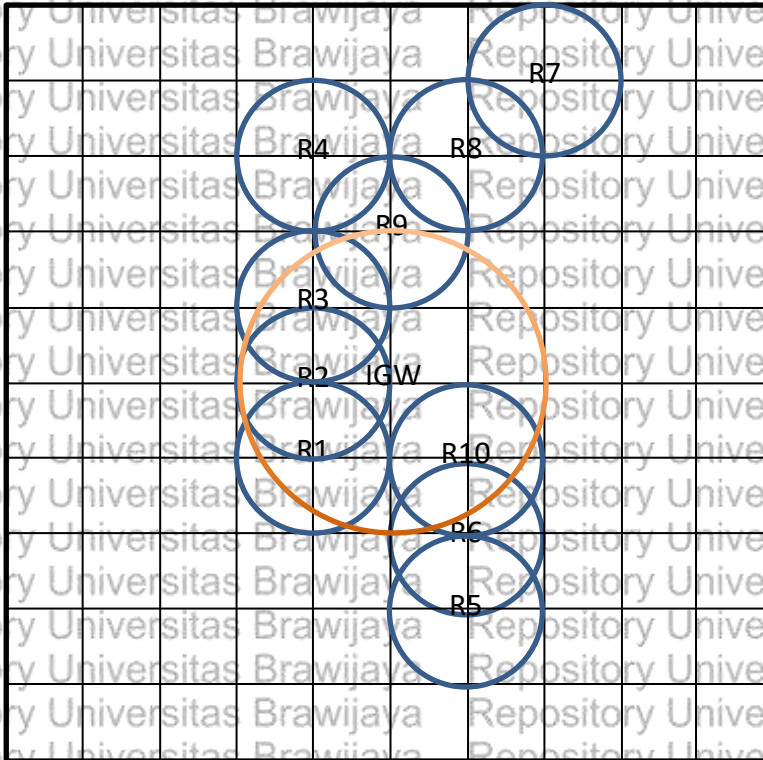
Parameter Algoritma genetika yang digunakan pada penelitian ini yaitu:

1. ukuran populasi (*popsiz*) = 3
2. rasio pindah silang (*cr*) = 0,5
3. rasio mutasi (*mr*) = 0,5

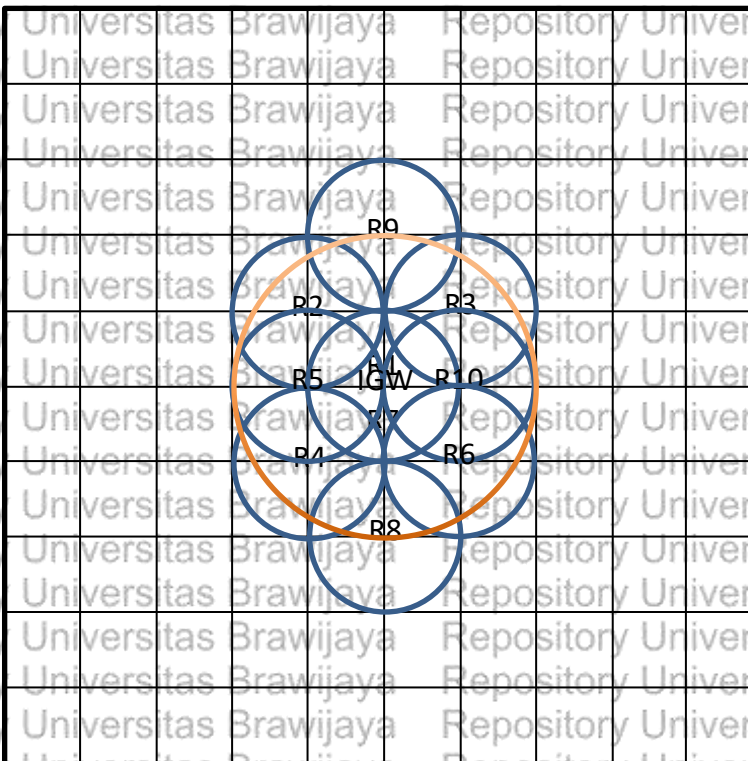
kromosom dibuat dengan parameter diatas. Setelah kromosom dibuat tahap selanjutnya adalah menghitung *fitness* pada tahap selanjutnya.

#### 3.7.4.2 Menghitung *Fitness*

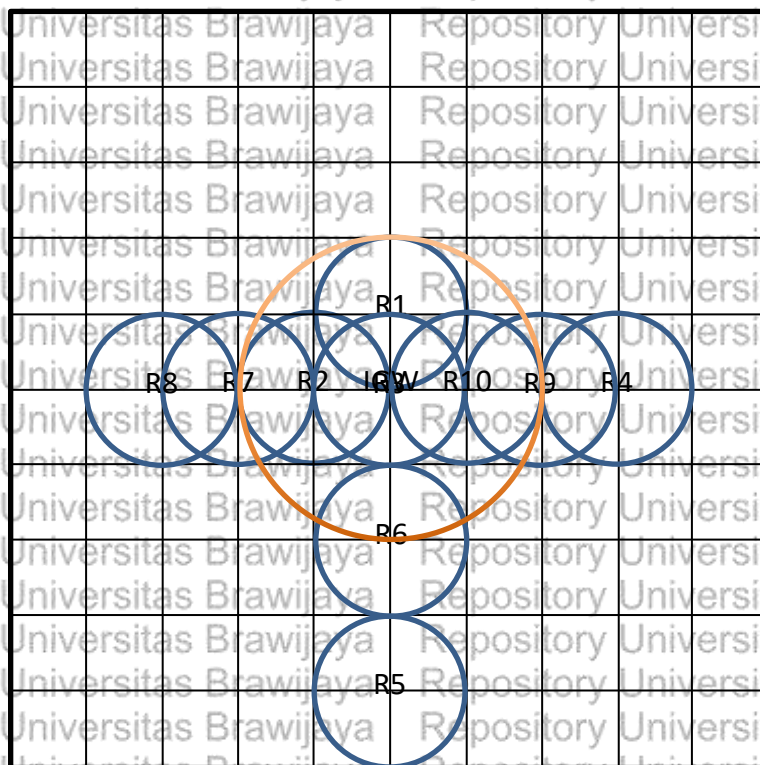
Pada subbab 3.7.4.2 sudah diberikan aturan persamaan *fitness*, untuk menghitung *fitness* dari setiap *parent* dibutuhkan jarak antar relay dengan relay lain dan jarak antar relay dengan internet *gateway*. Untuk mengetahui luas cakupan relay dan jumlah relay yang terhubung dengan internet *gateway* akan ditampilkan gambar implementasi *parent1*, *parent2*, dan *parent3* pada Gambar 3.7, 3.8, dan 3.9



Gambar 3.7 Gambar Pemetaan Parent 1



Gambar 3.8 Gambar Pemetaan Parent 2



Gambar 3.9 Gambar Pemetaan Parent 3

Dilihat dari persamaan *fitness* pada subbab 2.3.4, maka perhitungan *fitness* dari masing-masing *parent* akan ditampilkan pada Tabel 3.8 sampai 3.37

Tabel 3.8 Tabel Nilai *Fitness* Relay 1 Parent 1

R1 terhadap									
IGW	R2	R3	R4	R5	R6	R7	R8	R9	R10
0	1	0	0	0	0	0	0	0	0
Total = 1									

Tabel 3.9 Tabel Nilai *Fitness* Relay 2 Parent 1

R2 terhadap									
IGW	R1	R3	R4	R5	R6	R7	R8	R9	R10
1	1	1	0	0	0	0	0	0	0
Total = 3									

Tabel 3.10 Tabel Nilai *Fitness* Relay 3 Parent 1

R3 terhadap									
IGW	R1	R2	R4	R5	R6	R7	R8	R9	R10
0	0	1	0	0	0	0	0	0	0
Total = 1									



**Tabel 3.11 Tabel Nilai *Fitness* Relay 4 Parent 1**

R4 terhadap									
IGW	R1	R2	R3	R5	R6	R7	R8	R9	R10
0	0	0	0	0	0	0	0	0	0
Total = 0									

**Tabel 3.12 Tabel Nilai *Fitness* Relay 5 Parent 1**

R5 terhadap									
IGW	R1	R2	R3	R4	R6	R7	R8	R9	R10
0	0	0	0	0	1	0	0	0	0
Total = 1									

**Tabel 3.13 Tabel Nilai *Fitness* Relay 6 Parent 1**

R6 terhadap									
IGW	R1	R2	R3	R4	R5	R7	R8	R9	R10
0	0	0	0	0	1	0	0	0	1
Total = 2									

**Tabel 3.14 Tabel Nilai *Fitness* Relay 7 Parent 1**

R7 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R8	R9	R10
0	0	0	0	0	0	0	0	0	0
Total = 0									

**Tabel 3.15 Tabel Nilai *Fitness* Relay 8 Parent 1**

R8 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R9	R10
0	0	0	0	0	0	0	0	0	0
Total = 0									

**Tabel 3.16 Tabel Nilai *Fitness* Relay 9 Parent 1**

R9 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R10
0	0	0	0	0	0	0	0	0	0
Total = 0									



**Tabel 3.17 Tabel Nilai *Fitness* Relay 10 Parent 1**

R10 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R9
0	0	0	0	0	0	1	0	0	0
Total = 1									

**Tabel 3.18 Tabel Nilai *Fitness* Relay 1 Parent 2**

R1 terhadap									
IGW	R2	R3	R4	R5	R6	R7	R8	R9	R10
1	0	0	0	1	0	1	0	0	1
Total = 4									

**Tabel 3.19 Tabel Nilai *Fitness* Relay 2 Parent 2**

R2 terhadap									
IGW	R1	R3	R4	R5	R6	R7	R8	R9	R10
0	0	0	0	1	0	0	0	0	0
Total = 1									

**Tabel 3.20 Tabel Nilai *Fitness* Relay 3 Parent 2**

R3 terhadap									
IGW	R1	R2	R4	R5	R6	R7	R8	R9	R10
0	0	0	0	0	0	0	0	0	1
Total = 1									

**Tabel 3.21 Tabel Nilai *Fitness* Relay 4 Parent 2**

R4 terhadap									
IGW	R1	R2	R3	R5	R6	R7	R8	R9	R10
0	0	0	0	1	0	0	0	0	0
Total = 1									

**Tabel 3.22 Tabel Nilai *Fitness* Relay 5 Parent 2**

R5 terhadap									
IGW	R1	R2	R3	R4	R6	R7	R8	R9	R10
1	1	1	0	1	0	1	0	0	0
Total = 5									

**Tabel 3.23 Tabel Nilai Fitness Relay 6 Parent 2**

R6 terhadap									
IGW	R1	R2	R3	R4	R5	R7	R8	R9	R10
0	0	0	0	0	0	0	0	0	1
Total = 1									

**Tabel 3.24 Tabel Nilai Fitness Relay 7 Parent 2**

R7 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R8	R9	R10
1	1	0	0	0	1	0	0	0	1
Total = 4									

**Tabel 3.25 Tabel Nilai Fitness Relay 8 Parent 2**

R8 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R9	R10
0	0	0	0	0	0	0	0	0	0
Total = 0									

**Tabel 3.26 Tabel Nilai Fitness Relay 9 Parent 2**

R9 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R10
0	0	0	0	0	0	0	0	0	0
Total = 0									

**Tabel 3.27 Tabel Nilai Fitness Relay 10 Parent 2**

R10 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R9
1	1	0	1	0	0	1	1	0	0
Total = 5									

**Tabel 3.28 Tabel Nilai Fitness Relay 1 Parent 3**

R1 terhadap									
IGW	R2	R3	R4	R5	R6	R7	R8	R9	R10
1	0	1	0	0	0	0	0	0	0
Total = 2									

**Tabel 3.29 Tabel Nilai Fitness Relay 2 Parent 3**

R2 terhadap									
IGW	R1	R3	R4	R5	R6	R7	R8	R9	R10
1	0	1	0	0	0	1	0	0	0
Total = 3									

**Tabel 3.30 Tabel Nilai Fitness Relay 3 Parent 3**

R3 terhadap									
IGW	R1	R2	R4	R5	R6	R7	R8	R9	R10
1	1	1	0	0	0	0	0	0	1
Total = 4									

**Tabel 3.31 Tabel Nilai Fitness Relay 4 Parent 3**

R4 terhadap									
IGW	R1	R2	R3	R5	R6	R7	R8	R9	R10
0	0	0	0	0	0	0	0	1	0
Total = 1									

**Tabel 3.32 Tabel Nilai Fitness Relay 5 Parent 3**

R5 terhadap									
IGW	R1	R2	R3	R4	R6	R7	R8	R9	R10
0	0	0	0	0	0	0	0	0	0
Total = 0									

**Tabel 3.33 Tabel Nilai Fitness Relay 6 Parent 3**

R6 terhadap									
IGW	R1	R2	R3	R4	R5	R7	R8	R9	R10
0	0	0	0	0	0	0	0	0	0
Total = 0									

**Tabel 3.34 Tabel Nilai Fitness Relay 7 Parent 3**

R7 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R8	R9	R10
0	0	1	0	0	0	0	1	0	0
Total = 2									

Tabel 3.35 Tabel Nilai *Fitness* Relay 8 Parent 3

R8 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R9	R10
0	0	0	0	0	0	0	1	0	0
Total = 0									

Tabel 3.36 Tabel Nilai *Fitness* Relay 9 Parent 3

R9 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R10
0	0	0	0	1	0	0	0	0	1
Total = 2									

Tabel 3.37 Tabel Nilai *Fitness* Relay 10 Parent 3

R10 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R9
1	1	0	1	0	0	0	0	0	1
Total = 4									

Dari hasil perhitungan *fitness* didapatkan nilai *fitness* untuk *parent1* adalah 9, nilai *fitness* untuk *parent2* adalah 22, dan *parent3* adalah 18.

### 3.7.4.3 Pindah Silang

Merujuk pada subbab 2.3.2, pindah silang adalah salah satu dari tahapan reproduksi dalam algoritma genetika. Cara kerja pindah silang adalah menukar dua gen secara acak untuk mendapatkan keberagaman untuk generasi berikutnya yang sesuai kromosom-kromosom pada generasi sekarang. Pindah silang pada manualisasi ini menggunakan pindah silang *extended intermediate crossover*. Pindah silang dilakukan kepada dua kromosom yang terpilih secara acak. Pada manualisasi ini *parent* yang terpilih adalah *parent1* dan *parent2* untuk dilakukan pindah silang. Nilai jumlah pindah silang adalah 0,3 dan dibulatkan menjadi 1. Dari hasil pindah silang antara *parent1* dan *parent 2* akan dihasilkan 2 jumlah keturunan yaitu *child1* dan *child2*. Rumus pindah silang yang digunakan adalah seperti persamaan berikut

$$C1 = P1 + a \times (P2 - P1)$$

$$C2 = P2 + a \times (P1 - P2)$$

$$C1 = \text{child1}$$

$$C2 = \text{child2}$$

$$P1 = \text{parent1}$$

$$P2 = \text{parent2}$$

$$a = \text{bilangan random}$$



setelah dilakukan perhitungan pindah silang dengan rumus diatas, didapatkan hasil keturunan yaitu *child1* dan *child2* dengan kromosom seperti tabel 3.38. dan 3.39.

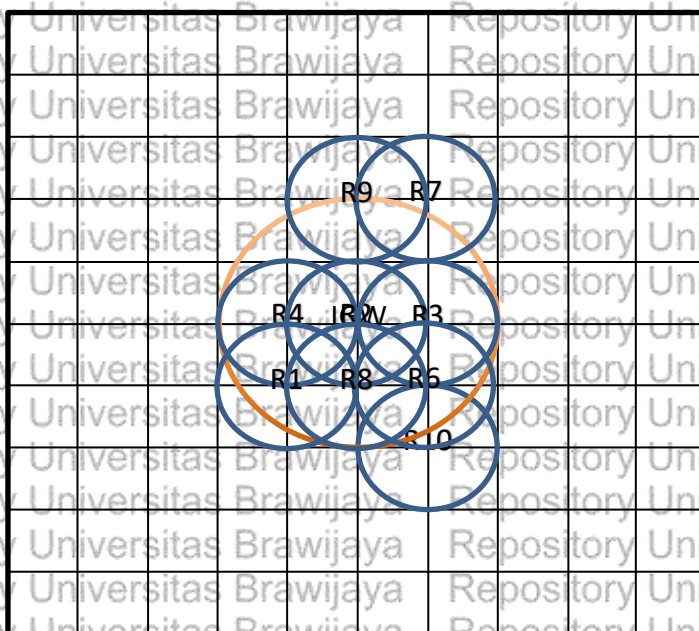
Tabel 3.38 Tabel Kromosom *Child 1*

Kromosom <i>child1</i>																			
R1		R2		R3		R4		R5		R6		R7		R8		R9		R10	
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
4,1	4,2	4,7	4,8	6,2	5,7	3,7	4,8	4,2	5,7	6,3	3,25	6,3	7,2	5,45	3,7	5,7	7,6	4,01	

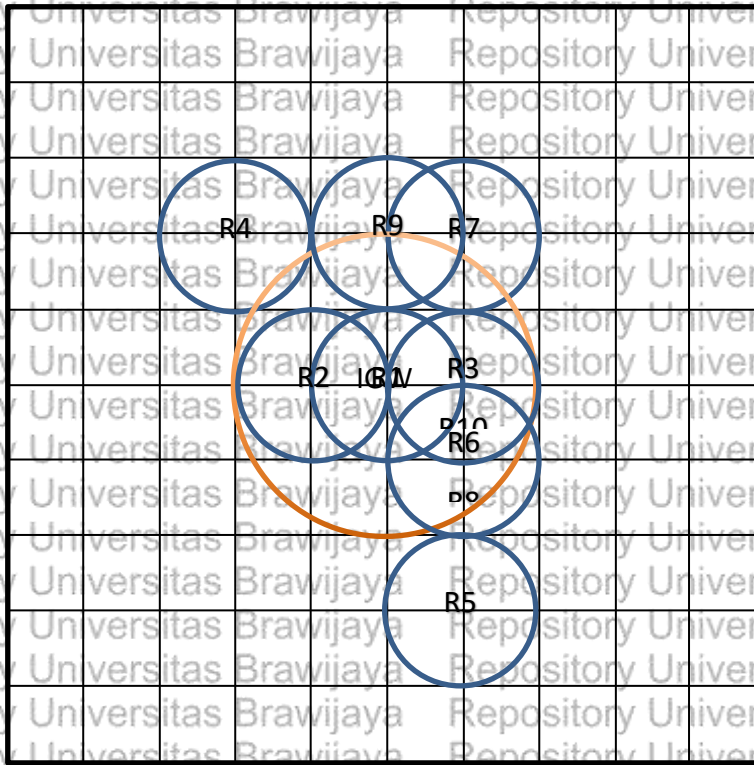
Tabel 3.39 Tabel Kromosom *Child 2*

Kromosom <i>child2</i>																			
R1		R2		R3		R4		R5		R6		R7		R8		R9		R10	
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
4,9	4,8	4,3	5,2	6,8	4,3	3,7	7,2	5,8	2,8	6,75	3,7	5,8	6,55	4,3	5,7	7,6	4,99		

Setelah didapatkan hasil pindah silang antara *parent1* dan *parent2* yang berupa *child1* dan *child2* dan akan digambarkan area implementasi beserta posisi relay dan internet *gateway* untuk memudahkan mencari luas area cakupan relay. Gambar implementasi *child1* dan *child2* akan ditampilkan pada Gambar 3.10. dan 3.11.



Gambar 3.10 Gambar Pemetaan *Child 1*



Gambar 3.11 Gambar Pemetaan Child 2

Dari hasil penggambaran implementasi *child1* dan *child2* akan didapatkan luas total cakupan relay. Setelah itu akan dihitung nilai *fitness* dari *child1* dan *child2* dengan rumus yang sama seperti rumus *fitness parent1*, *parent2*, dan *parent3*. Perhitungan *fitness child1* dan *child2* akan ditampilkan pada Tabel 3.40 sampai 3.59.

Tabel 3.40 Tabel Nilai *Fitness* Relay 1 *Child 1*

R1 terhadap									
IGW	R2	R3	R4	R5	R6	R7	R8	R9	R10
0	0	0	1	1	0	0	1	0	0
Total = 3									

Tabel 3.41 Tabel Nilai *Fitness* Relay 2 *Child 1*

R2 terhadap									
IGW	R1	R3	R4	R5	R6	R7	R8	R9	R10
1	0	1	1	1	0	0	1	0	0
Total = 5									

**Tabel 3.42 Tabel Nilai *Fitness Relay 3 Child 1***

R3 terhadap									
IGW	R1	R2	R4	R5	R6	R7	R8	R9	R10
1	0	1	0	0	1	0	0	0	0
Total = 3									

**Tabel 3.43 Tabel Nilai *Fitness Relay 4 Child 1***

R4 terhadap									
IGW	R1	R2	R3	R5	R6	R7	R8	R9	R10
1	1	1	0	1	0	0	0	0	0
Total = 4									

**Tabel 3.44 Tabel Nilai *Fitness Relay 5 Child 1***

R5 terhadap									
IGW	R1	R2	R3	R4	R6	R7	R8	R9	R10
1	1	1	0	1	0	0	0	0	0
Total = 4									

**Tabel 3.45 Tabel Nilai *Fitness Relay 6 Child 1***

R6 terhadap									
IGW	R1	R2	R3	R4	R5	R7	R8	R9	R10
0	0	0	1	0	0	0	1	0	1
Total = 3									

**Tabel 3.46 Tabel Nilai *Fitness Relay 7 Child 1***

R7 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R8	R9	R10
0	0	0	0	0	0	0	0	1	0
Total = 1									

**Tabel 3.47 Tabel Nilai *Fitness Relay 8 Child 1***

R8 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R9	R10
1	1	1	0	0	0	1	0	0	0
Total = 4									

**Tabel 3.48 Tabel Nilai *Fitness* Relay 9 Child 1**

R9 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R10
0	0	0	0	0	0	0	1	0	0
Total = 1									

**Tabel 3.49 Tabel Nilai *Fitness* Relay 10 Child 1**

R10 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R9
0	0	0	0	0	0	1	0	0	0
Total = 1									

**Tabel 3.50 Tabel Nilai *Fitness* Relay 1 Child 2**

R1 terhadap									
IGW	R2	R3	R4	R5	R6	R7	R8	R9	R10
1	1	1	0	0	0	0	0	0	1
Total = 4									

**Tabel 3.51 Tabel Nilai *Fitness* Relay 2 Child 2**

R2 terhadap									
IGW	R1	R3	R4	R5	R6	R7	R8	R9	R10
1	1	0	0	0	0	0	0	0	0
Total = 2									

**Tabel 3.52 Tabel Nilai *Fitness* Relay 3 Child 2**

R3 terhadap									
IGW	R1	R2	R4	R5	R6	R7	R8	R9	R10
1	1	0	0	0	1	0	1	0	1
Total = 5									

**Tabel 3.53 Tabel Nilai *Fitness* Relay 4 Child 2**

R4 terhadap									
IGW	R1	R2	R3	R5	R6	R7	R8	R9	R10
0	0	0	0	0	0	0	0	0	0
Total = 0									



**Tabel 3.54 Tabel Nilai Fitness Relay 5 Child 2**

R5 terhadap									
IGW	R1	R2	R3	R4	R6	R7	R8	R9	R10
0	0	0	0	0	0	0	0	0	0
Total = 0									

**Tabel 3.55 Tabel Nilai Fitness Relay 6 Child 2**

R6 terhadap									
IGW	R1	R2	R3	R4	R5	R7	R8	R9	R10
0	0	0	1	0	0	0	1	0	1
Total = 3									

**Tabel 3.56 Tabel Nilai Fitness Relay 7 Child 2**

R7 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R8	R9	R10
0	0	0	0	0	0	0	0	1	0
Total = 1									

**Tabel 3.57 Tabel Nilai Fitness Relay 8 Child 2**

R8 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R9	R10
0	0	0	1	0	0	1	0	0	1
Total = 5									

**Tabel 3.58 Tabel Nilai Fitness Relay 9 Child 2**

R9 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R10
0	0	0	0	0	0	0	1	0	0
Total = 1									

**Tabel 3.59 Tabel Nilai Fitness Relay 10 Child 2**

R10 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R9
1	1	0	1	0	0	1	0	1	0
Total = 5									



Dari hasil perhitungan nilai *fitness* terhadap *child1* dan *child2*, didapatkan nilai *fitness* dari *child1* adalah 29 dan *fitness* untuk *child2* adalah 31.

### 3.7.4.4 Mutasi

Merujuk pada subbab 2.3.3 mutasi adalah salah satu tahapan reproduksi dalam algoritma genetika. Cara kerja mutasi adalah menukar pasangan gen dalam sebuah kromosom yang terpilih secara acak. Proses mutasi menghasilkan kromosom anak dengan memodifikasi satu atau lebih sifat pada kromosom yang sama. Mutasi yang digunakan pada manualisasi ini adalah random mutation. Cara kerja random mutation adalah dengan cara menambah atau mengurangi nilai gen yang dipilih dengan bilangan acak. *Parent* yang terpilih untuk dimutasi terpilih secara acak. Pada manualisasi ini *parent* yang dimutasi adalah *parent3*. Nilai jumlah mutasi adalah 0,3 yang dibulatkan menjadi 1. Dari hasil mutasi akan didapatkan 1 jumlah keturunan dalam manualisasi ini dijadikan *child3*. Rumus pindah silang yang digunakan paa manualisasi ini adalah sebagai berikut

$$C3 = P3 - (a(x_{max} \text{ atau } y_{max} - x_{min} \text{ atau } y_{min}))$$

$$C3 = \textit{child3}$$

$$P3 = \textit{parent3}$$

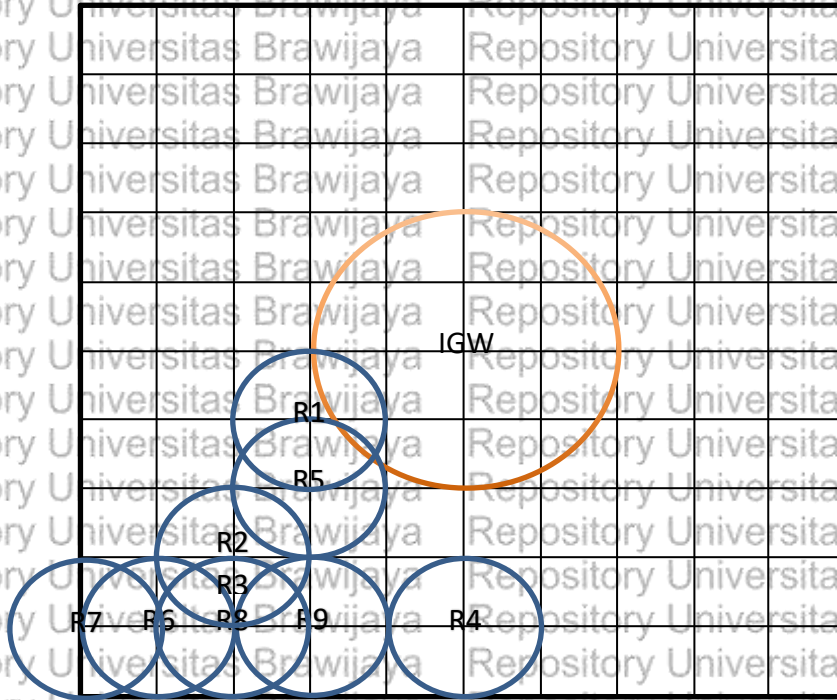
$$a = \text{bilangan random}$$

setelah dilakukan mutasi pada *parent3* dengan rumus diatas, didapatkan hasil keturunan *child3* dengan kromosom yang dapat dilihat pada tabel 3.60.

Tabel 3.60 Tabel Kromosom *Child 3*

Kromosom <i>child3</i>																			
R1		R2		R3		R4		R5		R6		R7		R8		R9		R10	
x	y	x	Y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	Y
3	3,	1,	1,	1,	2,	4,	0,	2,	2,	1,	0,	0,	0,	2,	0,	2,	0,	2,	2,
9	5	6	6	8	4	9	9	6	5	2	8	3	6	1	7	8	5	3	1

Setelah didapatkan hasil mutasi dari *parent3* yang berupa *child3* dan akan digambarkan area implementasi beserta posisi relay dan internet *gateway* untuk memudahkan mencari luas area cakupan relay. Gambar implementasi *child3* akan ditampilkan pada Gambar 3.12.



Gambar 3.12 Gambar Pemetaan Child 3

Dari hasil penggambaran implementasi *child3* akan didapatkan luas total cakupan relay. Setelah itu akan dihitung nilai *fitness* dari *child3* dengan rumus yang sama seperti rumus *fitness parent1*, *parent2*, dan *parent3*. Perhitungan *fitness child3* akan ditampilkan pada Tabel 3.61 sampai 3.70.

Tabel 3.61 Tabel Nilai *Fitness* Relay 1 *Child 3*

R1 terhadap									
IGW	R2	R3	R4	R5	R6	R7	R8	R9	R10
0	0	0	0	1	0	0	0	0	0
Total = 1									

Tabel 3.62 Tabel Nilai *Fitness* Relay 2 *Child 3*

R2 terhadap									
IGW	R1	R3	R4	R5	R6	R7	R8	R9	R10
0	0	1	0	0	0	0	1	0	0
Total = 2									

Tabel 3.63 Tabel Nilai *Fitness* Relay 3 *Child 3*

R3 terhadap									
IGW	R1	R2	R4	R5	R6	R7	R8	R9	R10
0	0	1	0	0	0	0	1	0	0
Total = 2									

**Tabel 3.64 Tabel Nilai Fitness Relay 4 Child 3**

R4 terhadap									
IGW	R1	R2	R3	R5	R6	R7	R8	R9	R10
0	0	0	0	0	0	0	0	0	0
Total = 0									

**Tabel 3.65 Tabel Nilai Fitness Relay 5 Child 3**

R5 terhadap									
IGW	R1	R2	R3	R4	R6	R7	R8	R9	R10
0	1	0	0	0	0	0	0	0	0
Total = 1									

**Tabel 3.66 Tabel Nilai Fitness Relay 6 Child 3**

R6 terhadap									
IGW	R1	R2	R3	R4	R5	R7	R8	R9	R10
0	0	0	0	0	0	1	0	0	0
Total = 2									

**Tabel 3.67 Tabel Nilai Fitness Relay 7 Child 3**

R7 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R8	R9	R10
0	0	0	0	0	0	1	0	0	0
Total = 1									

**Tabel 3.68 Tabel Nilai Fitness Relay 8 Child 3**

R8 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R9	R10
0	0	1	1	0	0	1	0	1	1
Total = 5									

**Tabel 3.69 Tabel Nilai Fitness Relay 9 Child 3**

R9 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R10
0	0	0	0	0	0	0	0	1	0
Total = 1									

Tabel 3.70 Tabel Nilai *Fitness* Relay 10 *Child 3*

R10 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R9
0	0	1	1	0	0	0	0	1	0
Total = 3									

Dari hasil perhitungan nilai *fitness* terhadap *child3*, didapatkan filai *fitness* dari *child3* adalah 18.

#### 3.7.4.5 Seleksi

Setelah dilakukan tahapan reproduksi yaitu pindah silang dan mutasi yang menghasilkan *child*. Tahapan selanjutnya adalah tahapan seleksi. Merujuk dari subbab 4.2.6 teknik seleksi yang digunakan adalah elitism. Berikut diberikan tabel 3.71 berisi urutan *fitness* dari yang tertinggi hingga terendah pada iterasi pertama.

Tabel 3.71 Tabel Seleksi Kromosom

Peringkat	Kromosom	<i>Fitness</i>
1.	<i>Child2</i>	31
2.	<i>Child1</i>	29
3.	<i>Child3</i>	22
4.	<i>Parent2</i>	18
5.	<i>Parent3</i>	18
6.	<i>Parent1</i>	9

Dari Tabel 3.71 bisa dilihat bahwa 3 kromosom dengan *fitness* tertinggi adalah *Parent2*, *Child1*, dan *Child3*, ketiga kromosom tersebut dipilih untuk menjadi *parent* di iterasi berikutnya.

#### 3.7.4.6 Iterasi Kedua

Setelah didapatkan *child1*, *child2*, dan *child3* dari tahapan reproduksi yaitu pindah silang dan mutasi, hasil dari reproduksi akan direproduksi lagi di iterasi kedua. Tahapan pada iterasi keua terdiri dari representasi kromosom, menghitung *fitness*, pindah silang, mutasi, dan seleksi. Dari hasil perhitungan *fitness*, yang dipilih untuk iterasi kedua adalah kromosom dengan urutan *fitness* tertinggi yaitu *Parent2*, *Child1*, dan *Child3*. Ketiga kromosom dengan *fitness* terbaik tersebut dijadikan *parent* pada iterasi kedua yaitu menjadi *Parent1*, *Parent2*, dan *Parent3* yang selanjutnya akan dilakukan pindah silang dan mutasi dengan rumus *fitness* yang sama dengan iterasi pertama. *Parent* pada iterasi kedua bisa dilihat pada tabel 3.72, 3.73, dan 3.74.



**Tabel 3.72 Tabel Kromosom Parent 1**

Kromosom Parent1																					
R1		R2		R3		R4		R5		R6		R7		R8		R9		R10			
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y		
5	5	4	6	6	6	4	4	4	4	5	6	4	5	5	5	3	5	7	6	5	5

**Tabel 3.73 Tabel Kromosom Parent 2**

Kromosom Parent2																					
R1		R2		R3		R4		R5		R6		R7		R8		R9		R10			
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y		
4,	4,	4,	4,	6	5,	3,	4,	4,	4,	5	6	3,	6,	7,	5,	3,	5,	7	6	4,	4,
1	2	7	8	2	7	8	2	2	2	5	25	3	3	2	45	7	7	5	6	01	01

**Tabel 3.74 Tabel Kromosom Parent 3**

Kromosom Parent3																			
R1		R2		R3		R4		R5		R6		R7		R8		R9		R10	
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
3	3,	1,	1,	1,	2,	4,	0,	2,	2,	1,	0,	0,	0,	2,	0,	2,	0,	2,	2,
9	5	6	8	4	9	9	6	5	5	2	8	3	6	1	7	8	5	3	1

Pada iterasi kedua akan dilakukan pindah silang antara *parent1* dan *parent2* yang akan menghasilkan keturunan *child1* dan *child2*. Dengan menggunakan rumus yang sama pada iterasi pertama berikut didapatkan hasil pindah silang dari *Parent1* dan *Parent2* yang menghasilkan *Child1* dan *Child2* hasil dari pindah silang akan ditampilkan pada Tabel 3.75 dan 3.76.

**Tabel 3.75 Tabel Kromosom Child 1**

Kromosom Child1																					
R1		R2		R3		R4		R5		R6		R7		R8		R9		R10			
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y		
4,	4,	4,	5,	6	5,	3,	4,	4,	4,	5	6	3,	5,	5,	5,	3,	5	7	6	4,	4,
9	8	2	5	5	7	6	1	1	8	4	9	2	4	9	2	4	9	9	9	9	9
1	4	1	2	2	9	4	8	1	1	5	9	4	5	9	4	5	9	9	9	9	9



**Tabel 3.76 Tabel Kromosom Child 2**

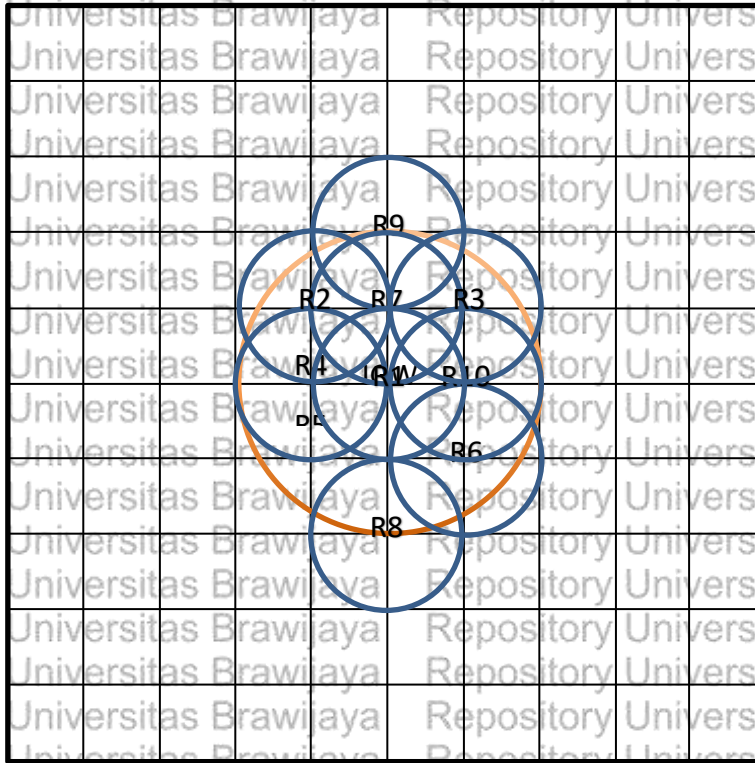
Kromosom Child2																				
R1		R2		R3		R4		R5		R6		R7		R8		R9		R10		
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	
4,	4,	4,	5,	6	5,	3,	4,	4,	5	6	3,	5,	6,	5,	3,	5	7	6	4,	01
19	96	49	28	68	91	16	02	43	84	21	20	24								

Setelah dilakukan pindah silang dari *Parent1* dan *Parent2* yang menghasilkan *child1* dan *Child2*, selanjtunya kan dilakukan mutasi. Mutasi dilakukan terhadap *Parent3* yang akan menghasilkan *Child3*. Dengan menggunakan rumus yang sama pada iterasi pertama berikut didapatkan hasil mutasi dari *Parent3* yang menghasilkan *Child3*. Hasil dari mutasi akan ditampilkan pada Tabel 3.77

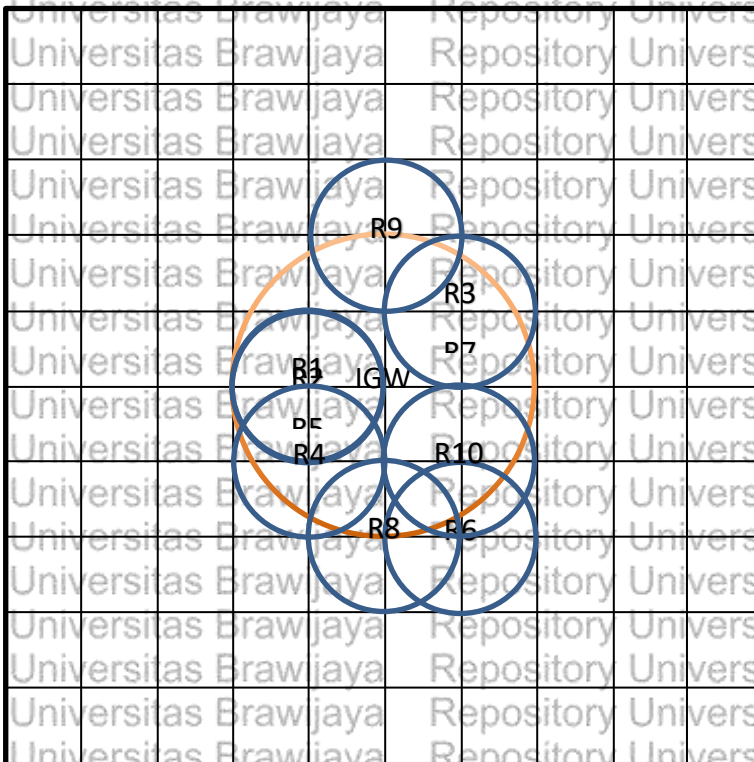
**Tabel 3.77 Tabel Kromosom Child 3**

Kromosom child3																			
R1		R2		R3		R4		R5		R6		R7		R8		R9		R10	
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
1	1,	1	1,	1,	0,	1,	3,	0,	1	2,	1,	3	3,	2	3,	1,	4	1,	0,
8	8	8	4	2	8	2	2	2	6	4	8	8	6	4	4	4	4	8	8

Selanjutnya adalah menghitung *fitness* dari masing masing individu. Untuk memudahkan menghitung *fitness* akan diberikan gambar penyebaran relay dari *Child1*, *Child2*, dan *Child3* pada Gambar3.75, 3.76, dan 3.77. perhitungan *fitness* akan ditampilkan pada Tabel 3.78 sampai 3.107 *Fitness* dari semua kromosom pada reproduksi iterasi kedua akan diurutkan dari yang tertinggi hingga terendah pada tabel 3.108. Hasil dari iterasi kedua yaitu 3 kromosom dengan *fitness* tertinggi akan dilanjutkan ke iterasi ketiga dan seterusnya.

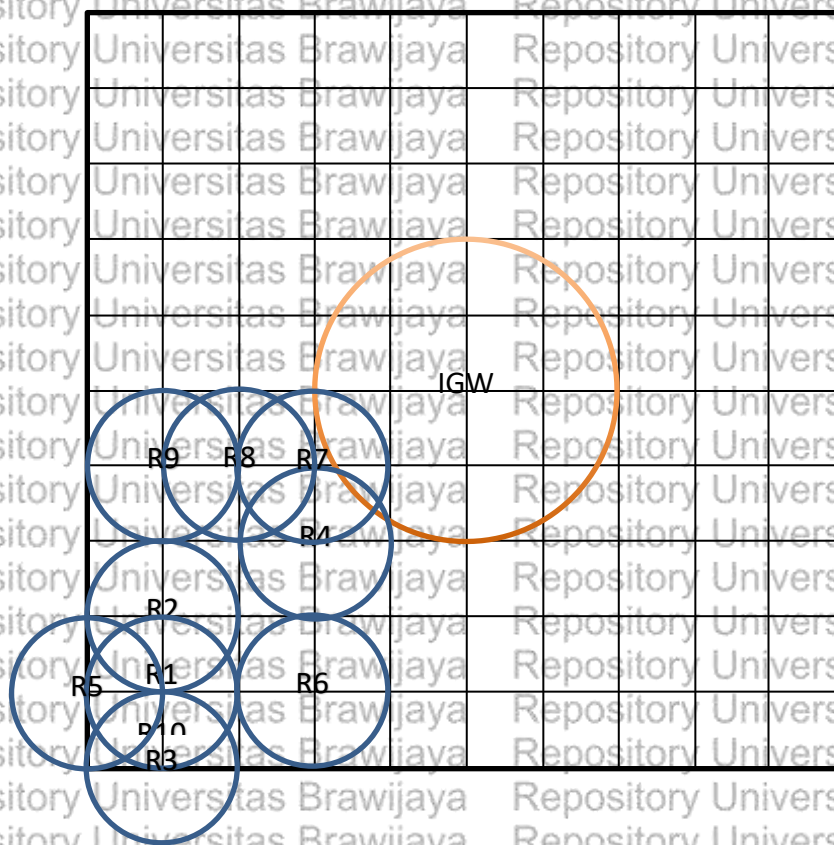


Gambar 3.13 Gambar Pemetaan Child 1



Gambar 3.14 Gambar Pemetaan Child 2





Gambar 3.15 Gambar Pemetaan Child 3

Tabel 3.78 Tabel Nilai *Fitness* Relay 1 Child 1

R1 terhadap									
IGW	R2	R3	R4	R5	R6	R7	R8	R9	R10
1	1	1	1	1	1	1	0	1	1
Total = 9									

Tabel 3.79 Tabel Nilai *Fitness* Relay 2 Child 1

R2 terhadap									
IGW	R1	R3	R4	R5	R6	R7	R8	R9	R10
1	1	1	0	1	0	0	0	1	0
Total = 5									

Tabel 3.80 Tabel Nilai *Fitness* Relay 3 Child 1

R3 terhadap									
IGW	R1	R2	R4	R5	R6	R7	R8	R9	R10
1	1	0	0	0	0	1	0	1	1
Total = 5									

**Tabel 3.81 Tabel Nilai *Fitness* Relay 4 Child 1**

R4 terhadap									
IGW	R1	R2	R3	R5	R6	R7	R8	R9	R10
1	1	1	0	1	0	1	0	0	0
Total = 5									

**Tabel 3.82 Tabel Nilai *Fitness* Relay 5 Child 1**

R5 terhadap									
IGW	R1	R2	R3	R4	R6	R7	R8	R9	R10
1	1	1	0	1	0	1	0	0	0
Total = 5									

**Tabel 3.83 Tabel Nilai *Fitness* Relay 6 Child 1**

R6 terhadap									
IGW	R1	R2	R3	R4	R5	R7	R8	R9	R10
1	1	0	0	0	0	0	1	0	1
Total = 4									

**Tabel 3.84 Tabel Nilai *Fitness* Relay 7 Child 1**

R7 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R8	R9	R10
0	1	1	1	1	1	0	0	1	1
Total = 8									

**Tabel 3.85 Tabel Nilai *Fitness* Relay 8 Child 1**

R8 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R9	R10
0	0	0	0	0	0	1	0	0	0
Total = 1									

**Tabel 3.86 Tabel Nilai *Fitness* Relay 9 Child 1**

R9 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R10
1	1	1	1	0	0	0	1	0	0
Total = 5									



**Tabel 3.87 Tabel Nilai Fitness Relay 10 Child 1**

R10 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R9
1	1	0	1	0	0	1	1	0	0
Total = 5									

**Tabel 3.88 Tabel Nilai Fitness Relay 1 Child 2**

R1 terhadap									
IGW	R2	R3	R4	R5	R6	R7	R8	R9	R10
1	1	0	1	1	0	0	0	0	0
Total = 4									

**Tabel 3.89 Tabel Nilai Fitness Relay 2 Child 2**

R2 terhadap									
IGW	R1	R3	R4	R5	R6	R7	R8	R9	R10
1	1	0	1	1	0	0	0	0	0
Total = 4									

**Tabel 3.90 Tabel Nilai Fitness Relay 3 Child 2**

R3 terhadap									
IGW	R1	R2	R4	R5	R6	R7	R8	R9	R10
1	0	0	0	0	0	1	0	1	0
Total = 3									

**Tabel 3.91 Tabel Nilai Fitness Relay 4 Child 2**

R4 terhadap									
IGW	R1	R2	R3	R5	R6	R7	R8	R9	R10
1	1	1	0	1	0	0	1	0	0
Total = 5									

**Tabel 3.92 Tabel Nilai Fitness Relay 5 Child 2**

R5 terhadap									
IGW	R1	R2	R3	R4	R6	R7	R8	R9	R10
1	1	1	0	1	0	0	0	0	0
Total = 4									

**Tabel 3.93 Tabel Nilai *Fitness* Relay 6 Child 2**

R6 terhadap									
IGW	R1	R2	R3	R4	R5	R7	R8	R9	R10
0	0	0	0	0	0	0	1	0	1
Total = 2									

**Tabel 3.94 Tabel Nilai *Fitness* Relay 7 Child 2**

R7 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R8	R9	R10
1	0	0	1	0	0	0	0	1	0
Total = 3									

**Tabel 3.95 Tabel Nilai *Fitness* Relay 8 Child 2**

R8 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R9	R10
0	0	0	0	1	0	1	0	0	1
Total = 3									

**Tabel 3.96 Tabel Nilai *Fitness* Relay 9 Child 2**

R9 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R10
0	0	0	1	0	0	0	1	0	0
Total = 2									

**Tabel 3.97 Tabel Nilai *Fitness* Relay 10 Child 2**

R10 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R9
1	0	0	0	0	0	1	0	1	0
Total = 3									

**Tabel 3.98 Tabel Nilai *Fitness* Relay 1 Child 3**

R1 terhadap									
IGW	R2	R3	R4	R5	R6	R7	R8	R9	R10
0	1	1	0	0	0	0	0	0	1
Total = 3									

**Tabel 3.99 Tabel Nilai Fitness Relay 2 Child 3**

R2 terhadap									
IGW	R1	R3	R4	R5	R6	R7	R8	R9	R10
0	1	0	0	0	0	0	0	0	1
Total = 2									

**Tabel 3.100 Tabel Nilai Fitness Relay 3 Child 3**

R3 terhadap									
IGW	R1	R2	R4	R5	R6	R7	R8	R9	R10
0	1	0	0	0	0	0	0	0	1
Total = 2									

**Tabel 3.101 Tabel Nilai Fitness Relay 4 Child 3**

R4 terhadap									
IGW	R1	R2	R3	R5	R6	R7	R8	R9	R10
0	0	0	0	0	0	1	0	0	0
Total = 1									

**Tabel 3.102 Tabel Nilai Fitness Relay 5 Child 3**

R5 terhadap									
IGW	R1	R2	R3	R4	R6	R7	R8	R9	R10
0	1	0	0	0	0	0	0	0	0
Total = 1									

**Tabel 3.103 Tabel Nilai Fitness Relay 6 Child 3**

R6 terhadap									
IGW	R1	R2	R3	R4	R5	R7	R8	R9	R10
0	0	0	0	0	0	0	0	0	0
Total = 0									

**Tabel 3.104 Tabel Nilai Fitness Relay 7 Child 3**

R7 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R8	R9	R10
0	0	0	0	1	0	0	1	0	0
Total = 2									

**Tabel 3.105 Tabel Nilai Fitness Relay 8 Child 3**

R8 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R9	R10
0	0	0	0	0	0	0	1	1	0
Total = 2									

**Tabel 3.106 Tabel Nilai Fitness Relay 9 Child 3**

R9 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R10
0	0	0	0	0	0	0	0	1	0
Total = 1									

**Tabel 3.107 Tabel Nilai Fitness Relay 10 Child 3**

R10 terhadap									
IGW	R1	R2	R3	R4	R5	R6	R7	R8	R9
0	1	1	1	0	0	0	0	0	0
Total = 3									

Dari hasil perhitungan *fitness Child1*, *Child2*, dan *Child3* didapatkan hasil *fitness* berturut-turut sebesar 52, 28, dan 17. Hasil *fitness* dari iterasi kedua akan ditampilkan secara urutan dari tertinggi hingga terendah pada tabel 3.108

**Tabel 3.108 Tabel Seleksi kromosom**

Peringkat	Kromosom	<i>Fitness</i>
1	<i>Child1</i>	52
2	<i>Parent1</i>	50
3	<i>Parent2</i>	43
4	<i>Parent3</i>	38
5	<i>Child2</i>	28
6	<i>Child3</i>	17

### 3.7.5 Perancangan Pengujian

Tahapan pengujian akan dirancang pada subbab ini. Pengujian dilakukan dengan tujuan untuk mendapatkan parameter terbaik pada algoritma. Pengujian terdiri dari pengujian konvergensi, pengujian ukuran populasi, dan pengujian rasio pindah silang.



### 3.7.5.1 Pengujian Konvergensi

Salah satu metode pengujian yang dilakukan pada penelitian ini adalah pengujian konvergensi. Pengujian konvergensi dilakukan dengan tujuan mendapatkan jumlah generasi terbaik. Cara kerja pengujian konvergensi adalah dengan memilih jumlah generasi ketika terjadinya konvergensi global. Pada penelitian ini pengujian akan dilakukan pada generasi ke 1000, nilai ukuran populasi 100, rasio pindah silang dan rasio mutasi 0,5.

**Tabel 3.109 Tabel Perancangan Pengujian Konvergensi:**

Uji ke-	Fitness generasi ke				
	1	2	...	999	1000
1					
2					
...					
9					
10					

### 3.7.5.2 Pengujian Ukuran Populasi

Salah satu metode pengujian yang dilakukan pada penelitian ini adalah pengujian ukuran populasi. Pengujian ukuran populasi dilakukan dengan tujuan mendapatkan ukuran populasi teroptimal. Parameter pengujian pada pengujian ukuran populasi ini adalah generasi ke 233 dengan rasio pindah silang dan mutasi 0,5.

**Tabel 3.110 Tabel Perancangan Pengujian Ukuran Populasi**

Uji ke-	Fitness generasi ke				
	1	2	...	232	233
1					
2					
...					
9					
10					

### 3.7.5.3 Pengujian Rasio Pindah silang dan Rasio Mutasi

Salah satu metode pengujian yang dilakukan pada penelitian ini adalah pengujian rasio pindah silang dan mutasi. Pengujian rasio pindah silang dan mutasi

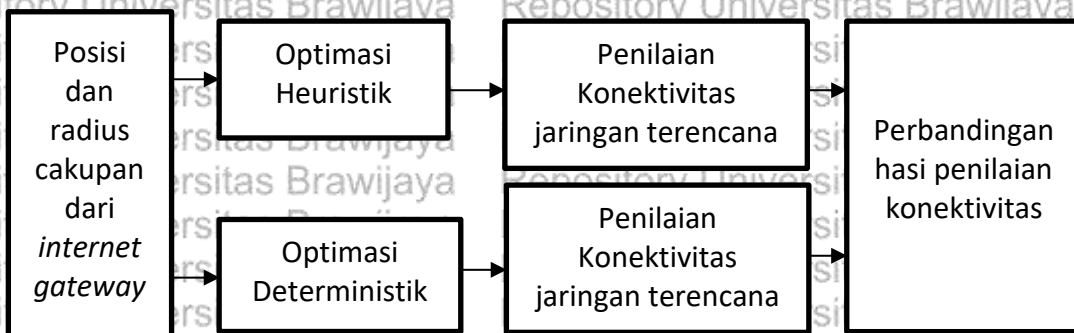
dilakukan dengan tujuan mendapatkan rasio pindah silang dan rasio mutasi teroptimal. Parameter pengujian pada pengujian rasio pindah silang dan mutasi ini adalah generasi ke 233 dan ukuran populasi 200.

**Tabel 3.111 Tabel Perancangan Pengujian Rasio**

Uji Ke-	{0,1;0,9}	{0,2;0,8}	...	{0,8;0,2}	{0,9;0,1}
1					
2					
...					
10					

### 3.8 Hasil dan Pengujian

Pengujian yang dilakukan pada penelitian optimasi peletakkan relay ini adalah membandingkan hasil penyelesaian optimasi menggunakan metode heuristik dengan deterministik pada jaringan terencana. Hasil penelitian didapatkan dari data hasil akhir optimasi peletakkan relay dengan metode algoritma genetika yang sudah diuji. Pengujian optimasi peletakkan relay dengan cara heuristic menggunakan algoritma genetika akan diperbandingkan dengan metode deterministik. Ditinjau berdasarkan penelitian yang menggunakan metode deterministik, peletakkan relay secara metode deterministik dengan parameter dan ukuran nyata dilapangan menghasilkan cakupan dan konektivitas jaringan yang cukup optimal (Asheer & Kumar, 2015). Dikarenakan tujuan penelitian ini supaya metode heuristik dengan algoritma genetika lebih nyata diaplikasikan dilapangan, maka metode deterministik digunakan untuk diandingkan dengan hasil optimasi menggunakan metode heuristik. Alur pengujian deterministik dijelaskan pada Gambar 3. Berikut



**Gambar 3.16 Gambar Alir Pengujian**

### 3.9 Kesimpulan dan Saran

Kesimpulan dan saran dibuat berdasarkan pengujian dan analisis. Berdasarkan hasil pengujian jaringan dengan relay yang sudah teroptimasi lallu akan dibahas





hasil pengujiannya. Hasil pengujian akan disimpulkan dan diharapkan bisa menjawab rumusan masalah pada BAB 1. Kesimpulan juga bisa menjadi dasar dilakukannya penelitian dimasa mendatang. Saran diharapkan bisa menambah wawasan dan menjadi referensi untuk penelitian-penelitian selanjutnya. Kesimpulan dan saran akan dijabarkan pada BAB 7.



## BAB 4 IMPLEMENTASI

Pada subbab 3.7 Perancangan dilakukan sebuah perancangan bagaimana penelitian ini akan dilakukan mulai dari formulasi permasalahan, alir perancangan algoritma, perancangan inialisasi awal, manualisasi, dan perancangan pengujian. Setelah strategi penelitian dirancang pada subbab 3.7 selanjutnya adalah dilakukan implementasi. Implementasi adalah mewujudkan rancangan tersebut kedalam kode program. Pada bab implementasi ini akan diimplementasikan algoritma genetika untuk mengoptimasi relay dari awal hingga akhir. Hasil yang didapatkan dari implementasi akan diuji dan dijabarkan hasil pengujiannya pada bab pengujian dan analisis.

### 4.1 Inialisasi Populasi Awal

Pada implementasi algoritma genetika dimulai dengan inialisasi populasi awal dengan menyimpan parameter-parameter yang akan digunakan. Selanjutnya adalah membuat gen yang nilainya acak lalu disimpan kedalam kromosom sebanyak jumlah ukuran populasi. Kromosom-kromosom tersebut akan dihitung nilai *fitness* kromosom tersebut. Kromosom selanjutnya akan diurutkan dari nilai *fitness* yang tertinggi sampai ke terendah implementasi dari inialisasi populasi awal, pengurutan kromosom, membuat inialisasi gen secara acak, dan inialisasi kromosom akan ditampilkan pada Tabel 4.1, 4.2, 4.3, dan 4.4

**Table 4.1 Pseudocode Inialisasi Populasi Awal**

```

Populasi Awal
START
  INPUT: self, Psize, Cr, Mr, xIGw, yIGw, RadIGw, numRelay, xMax,
  yMax, Rad
  self.popsiz ← Psize
  self.cr ← Cr
  self.mr ← Mr
  self.pop ← []
  FOR setiap i pada 0 sampai self.popsiz
    Tambah nilai pada self.pop
    (chro.Chromosome.generateRandom(xIGw,yIGw,RadIGw,numRelay,xMax,y
    Max,Rad))
  self.pop ← self.sorting(copy.deepcopy(self.pop))
END
  
```

**Table 4.2 Pseudocode Pengurutan Kromosom**

```

Sorting
START
  INPUT: self, objek data
  FOR setiap i pada data
    FOR setiap j pada i+1 sampai data terakhir
      IF fitness pada data index ke-i lebih kecil dari fitness
        pada data index ke-j THEN
          tukar posisi antara kedua nilai
        ENDIF
      ENDFOR
    ENDFOR
  OUTPUT: data
END

```

**Table 4.3 Psudocode Inisialisasi Gen Secara Acak**

```

Random
START
  INPUT: x, y, rad, numRelay, xMax, yMax, Rad
  f <- angka acak dari 0 sampai xMax
  g <- angka acak dari 0 sampai ymax
  gene <-
  OUTPUT: CALL hasil chromosome dengan paramaeter (xIGw, yIGw,
  gene, xMax, yMax, Rad)
END

```

**Table 4.4 Pseudocode Inisialisasi Kromosom**

```

Inisialisasi kromosom
START
  INPUT: self, xIGw, yIGw, RadIGw, gen, xMax, yMax, Rad
  self.xIGw <- xIGw
  self.yIGw <- yIGw
  self.radIGw <- RadIGw
  self.gene <- gen
  self.xmax <- xMax
  self.ymax <- yMax
  self.rad <- Rad

```



```
self.fitness <- self.calcfit()
END
```

## 4.2 Perhitungan *Fitness*

Setelah kromosom sudah dibentuk selanjutnya adalah perhitungan *fitness*. *Fitness* digunakan untuk mengetahui seberapa bagus dan optimalnya nilai dari suatu individu dengan syarat yang diinginkan. Syarat pada *fitness* sudah dijabarkan pada Tabel 3.4. disertai nilai *fitness* yang digunakan. Setiap relay yang melanggar syarat ataupun mengikuti syarat akan mendapatkan nilai sesuai dengan Tabel 3.4. Implementasi *fitness* akan menghitung jarak relay terdekat dengan internet gateway sebagai *tier* pertama lalu relay *tier* pertama akan dihitung jaraknya dengan relay sekitarnya menjadi *tier* kedua dan seterusnya. Implementasi perhitungan *fitness* diuraikan pada Tabel 4.6.

**Table 4.5 Pseudocode Perhitungan *Fitness***

```
fitness
BEGIN
INPUT: objek self
fitness <- 0
variabel before <- list sepanjang gene dibagi 2
variabel after <- list sepanjang gene dibagi 2
variabel notTaken <- list sepanjang gene dibagi 2
variabel tier <- list sepanjang gene dibagi 2
FOR i <- 0 sampai panjang (self.gene/2)
  IF (i == 0):
    FOR j <- 0 sampai panjang (self.gene/2):
      S <- nilai dari self.euclid([self.xIGw, self.yIGw],
self.gene[j*2:(j*2)+2])
      IF (S <= self.radIGw):
        before[j] <- -1
        tambah nilai j pada tier di urutan ke-i
        hapus item notTaken yang bernilai j
        fitness += S/self.radIGw
      ENDIF
    ENDFOR
  cetak "belum diambil" dan nilai dari notTaken
  cetak "tier yang terisi" dan nilai dari tier
ELSE:
  FOR posj,j <- 0 sampai enumerator(notTaken):
```



```

terdekat <- -1
indterdekat <- -1
Cetak "tier sekarang adalah" dan nilai dari i
Jarak <- []
for k in range(1, len(tier[i])):
    jarak.append(self.euclid(self.gene[j*2:(j*2)+2], self.gene[k*2:(k*2)+2]))
for k in tier[i-1]:
    cetak nilai jarak
FOR k <- 0 sampai panjang jarak:
    IF (terdekat < jarak[k] AND jarak[k] <= self.rad):
        Terdekat <- jarak[k]
        Indterdekat <- k
    ENDIF
ENDIFOR
IF (terdekat != -1):
    before[j] <- tier[i-1][indterdekat]
    tambah nilai notTaken[pos] pada tier di urutan ke-i
    hapus nilai notTaken[pos] pada notTaken
    fitness += terdekat/self.rad
ENDIF
ENDIFOR
IF not tier[i]:
    Break
ENDIF
ENDIF
for i in range(len(tier)):
    for j in range(len(tier[i])):
        if (j < (len(tier[i]) - 1)):
            for k in range(j+1, len(tier[i])):
                fitness_1_count += 1
                if (self.euclid(self.gene[tier[i][j]*2:(tier[i][j]*2)+2], self.gene[tier[i][k]*2:(tier[i][k]*2)+2]) > (0.8 * self.rad)):
                    fitness_1 += 1
            print(tier[i], len(tier[i]))
            print("tier ke", i, fitness_1, fitness_1_count)
            if fitness_1_count == 0:
                fitness_1_count = 1
            print("fitness kedua", fitness_1, fitness_1_count)
            self.fitness1 = fitness / (len(self.gene) / 2)
            self.fitness2 = (fitness_1 / fitness_1_count)

```



```

return
(fitness/(len(self.gene)/2))+(fitness_1/fitness_1_count)END

```

### 4.3 Pindah Silang

Setelah diimplementasikannya *fitness* dan diitung *fitness* dari *parent* selanjutnya adalah mengimplementasikan tahapan reproduksi pindah silang. Pindah silang yang akan diimplementasikan adalah pindah silang metode intermediate extended crossover. Aturan pindah silang pada penelitian ini sudah dijabarkan pada subbab 4.2.4. Implementasi pindah silang diuraikan pada Tabel 4.6.

**Table 4.6 Pseudocode Pindah Silang**

```

Crossover
START
INPUT: self, objek partner
childgene1 <- self.gene
childgene2 <- partner.gene
FOR setiap i pada childgene1
childgene1 index ke-i <- hasil pembulatan (childgene1 index
ke-i + (angka random * (childgene2 index ke-i - childgene1 index
ke-i)))
childgene2 index ke-i <- hasil pembulatan (childgene2 index
ke-i + (angka random * (childgene1 index ke-i - childgene2 index
ke-i)))
ENDFOR
OUTPUT: CALL hasil Chromosome dengan parameter (self.xIGw,
self.yIGw, self.radIGw, childgene1, self.xmax, self.ymax,
self.Rad), CALL hasil Chromosome dengan parameter (self.xIGw,
self.yIGw, self.radIGw, childgene2, self.xmax, self.ymax,
self.Rad)
END

```

### 4.4 Mutasi

Tahap selanjutnya setelah pindah silang adalah mutasi. Mutasi yang akan diimplementasikan adalah mutasi dengan metode random mutation. Aturan mutasi pada penelitian ini sudah dijabarkan pada subbab 4.2.5. implementasi mutasi diuraikan pada Tabel 4.7.

**Table 4.7 Pseudocode Mutasi**

```

Mutasi
START
INPUT: objek self
childgene <- self.gene

```



```

FOR setiap nilai i pada childgene
  IF nilai i genap THEN
    childgene index ke-i <- childgene index ke-i + (angka acak
    * self.max)
  IF childgene index ke-i lebih besar dari self.xmax THEN
    childgene index ke-i <- childgene index ke-i - self.xmax
  ENDIF
  ELSE IF nilai i ganjil THEN
    childgene index ke-i <- childgene index ke-i + (angka acak
    * self.ymax)
  IF childgene index ke-i lebih besar dari self.ymax THEN
    childgene index ke-i <- childgene index ke-i - self.ymax
  ENDIF
  pembulatan2 angka dibelakang koma dari nilai childgene index
  ke-i
  ENDIF
ENDFOR
OUTPUT: CALL hasil Chromosome dengan parameter (self.xIGw,
self.yIGw, self.radIGw, childgene, self.xMax, self.yMax,
self.Rad)
END

```

#### 4.5 Evolusi dan Seleksi

Tahapan akhir dari implementasi algoritma genetika pada penelitian ini adalah evolusi dan seleksi. Evolusi adalah proses membentuk hasil dari proses pindah silang dan mutasi yang menghasilkan child yang akan digunakan untuk iterasi berikutnya. Seleksi adalah proses memilih kromosom yang bagus dan menghilangkan kromosom yang tidak terpilih sehingga jumlah kromosom sama seperti ukuran populasi. Implementasi evolusi dan seleksi diuraikan pada Tabel 4.8.

**Table 4.8 Pseudocode Evolusi dan Seleksi**

```

evolusi dan seleksi
BEGIN
INPUT: objek self
descendent <- array kosong
cross <- pembulatan ((self.cr * self.popsiz) / 2)
muta <- pembulatan (self.mr * self.popsiz)
I <- 0
WHILE (I < cross) DO
  par1 <- nilai acak dari 0 sampai self.popsiz - 1

```



```
par2 <- par1
WHILE (par1 == par2)
  par2 <- nilai acak dari 0 sampai (self.popsize - 1)
ENDDO
kid <- penyilangan nilai self[par1] dengan self[par2]
tambah kid[0] pada descendant
tambah kid[1] pada descendant
increment i
ENDDO
I <- 0
WHILE (I < muta) DO:
  tambah nilai acak self pada descendant
  increment i
ENDDO
tambah nilai self pada descendant
urutkan self secara descending
END
```





## BAB 5 PENGUJIAN DAN ANALISIS

Pada bab 5 sudah dipaparkan bagaimana optimasi peletakkan relay diimplementasikan yang terdiri dari pseudocode inisialisasi populasi awal, perhitungan *fitness*, pindah silang, mutasi, dan evolusi dan seleksi. Setelah implementasi tersebut dijalankan akan dilakukan pengujian yang menghasilkan data hasil implementasi yang akan di analisa pada bab 6. Pengujian yang akan dilakukan adalah penguujian konvergensi, pengujian ukuran populasi, pengujian rasio, dan pengujian akhir dan analisis. Hasil pengujian dan analisis akan diambil kesimpulan dan dijabarkan pada bab penutup.

### 5.1 Pengujian Konvergensi

Untuk mendapatkan hasil perhitungan terbaik menggunakan agoritma genetika perlu didapatkan parameter terbaik untuk digunakan pada proses optimasi peletakkan relay. Pengujian konvergensi dilakukan untuk mendapatkan parameter generasi terbaik. Pada kasus optimasi node relay pada jaringan WSN generasi saat konvergensi global adalah generasi yang optimal untuk perhitungan algoritma genetika. Karena algoritma genetika memiliki probabilitas yang bisa menyebabkan banyak variasi hasil setiap kali dijalankan, maka pada pengujian konvergensi ini akan dilakukan sepuluh kali percobaan. Hasil pengujian konvergensi ini akan digunakan untuk pengujian selanjutnya. Pengujian konvergensi pada penelitian ini akan berlangsung sampai 1000 generasi dengan menggunakan 1000 ukuran populasi, 0,5 rasio pindah silang, dan 0,5 rasio mutasi yang akan dilakukan pada 3 skenario. Hasil pengujian konvergensi ditampilkan pada Tabel 5.1, 5.2, dan 5.3 dan Gambar 5.1, 5.2, dan 5.3

**Tabel 5.1 Pengujian Konvergensi Skenario 1**

Generasi ke-	Pengujian ke-									
	1	2	3	4	5	6	7	8	9	10
0	1,59 57	1,59 08	1,58 68	1,65 61	1,62 62	1,61 14	1,64 92	1,57 34	1,59 34	1,61 85
1	1,60 73	1,61 68	1,61 97	1,65 61	1,62 62	1,61 46	1,64 92	1,62 00	1,59 35	1,62 69
2	1,60 73	1,61 68	1,61 97	1,65 61	1,62 62	1,61 46	1,65 59	1,63 89	1,61 11	1,65 53
...										
64				1,69 95						
...										
136									1,68 73	
...										



254							1,74			
...							86			
378						1,74				
...						17				
445							1,73			
...							20			
592				1,72						
...				23						
596									1,71	
...									19	
671						1,71				
...						83				
750	1,69									
...	60									
752		1,70								
...		35								
1000	1,69	1,70	1,69	1,72	1,71	1,74	1,73	1,78	1,68	1,71
	60	35	95	23	83	17	20	49	73	19

Tabel 5.2 Pengujian Konvergensi Skenario 2

Generasi ke-	Pengujian ke-									
	1	2	3	4	5	6	7	8	9	10
0	1,5007	1,5490	1,5443	1,5155	1,5627	1,5969	1,5323	1,5535	1,5946	1,5236
1	1,5030	1,5799	1,5443	1,5572	1,6343	1,5979	1,5478	1,5738	1,5946	1,5945
2	1,5831	1,5799	1,5868	1,5572	1,6343	1,6068	1,5618	1,5738	1,5946	1,5979



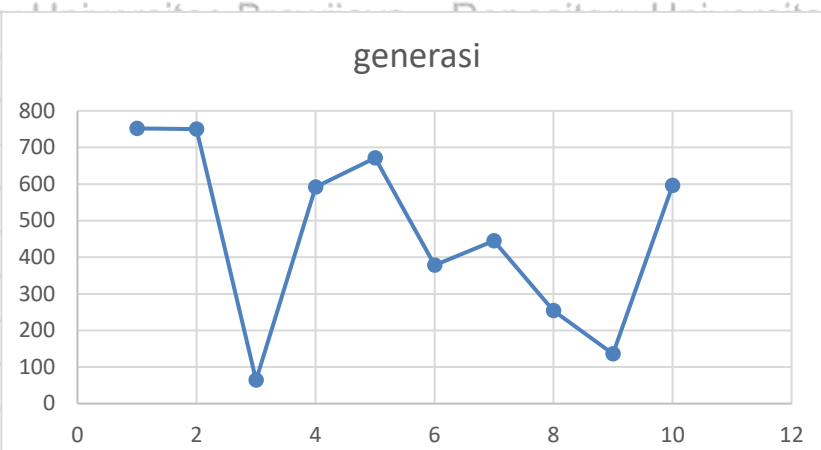
...										
146							1,74			
...							15			
367					1,75					
...					56					
370							1,69			
...							89			
419									1,72	
...									56	
433								1,79		
...								55		
459	1,70									
...	47									
465			1,70							
...			46							
682		1,75								
...		65								
712										1,68
...										06
846			1,68							
...			36							
1000	1,70	1,75	1,68	1,70	1,75	1,69	1,74	1,79	1,72	1,68
...	47	65	36	46	56	89	15	55	56	06

**Tabel 5.3 Pengujian Konvergensi Skenario 3**

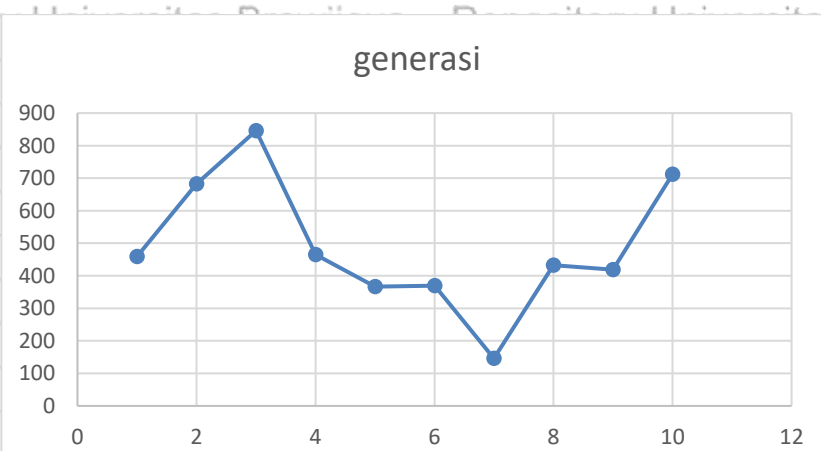
Generasi ke-	Pengujian ke-									
	1	2	3	4	5	6	7	8	9	10
0	1,50	1,54	1,54	1,51	1,56	1,59	1,53	1,55	1,59	1,52
	07	90	43	55	27	69	23	35	46	36



1	1,50 30	1,57 99	1,54 43	1,55 72	1,63 43	1,59 79	1,54 78	1,57 38	1,59 46	1,59 45
2	1,58 31	1,57 99	1,58 68	1,55 72	1,63 43	1,60 68	1,56 18	1,57 38	1,59 46	1,59 79
...										
234	1,69 62									
...										
259						1,73 13				
...										
261									1,66 19	
...										
263		1,72 07								
...										
283				1,68 74						
...										
309					1,67 65					
...										
315								1,70 89		
...										
325							1,72 07			
...										
396									1,70 04	
...										
424			1,71 00							
...										
1000	1,69 62	1,72 07	1,71 00	1,68 74	1,67 65	1,73 13	1,72 07	1,70 89	1,66 19	1,70 04



**Gambar 5.1 Grafik Pengujian Konvergensi Skenario 1**



**Gambar 5.2 Grafik Pengujian Konvergensi Skenario 2**



**Gambar 5.3 Grafik Pengujian Konvergensi Skenario 3**

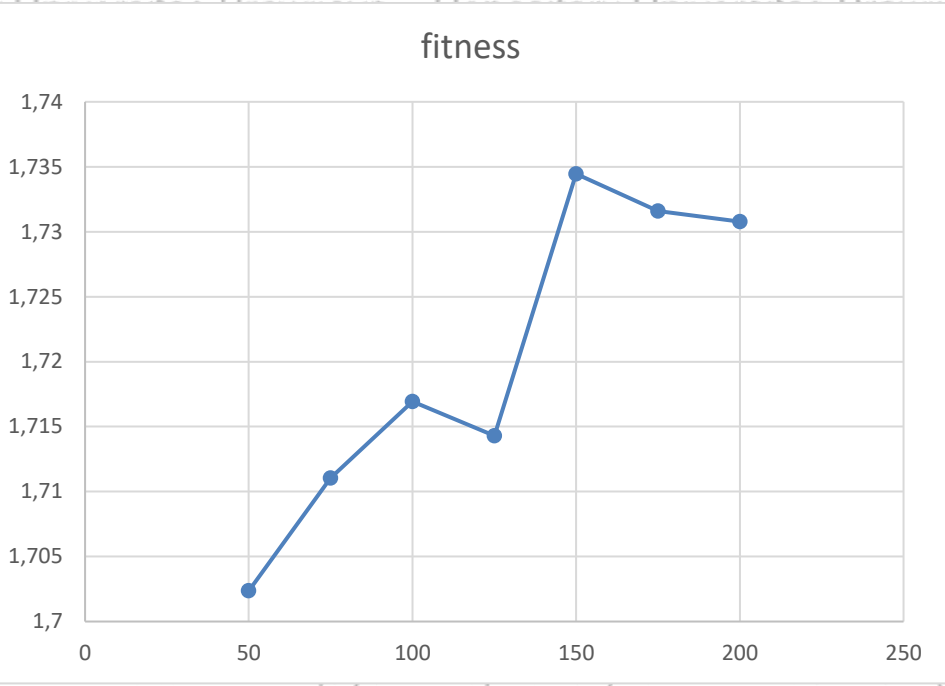
Hasil pengujian konvergensi dapat dilihat pada Tabel 5.1 dan Gambar 5.1 konvergensi global pada skenario 1 terjadi setelah generasi ke-752. Generasi ke-752 dianggap sebagai generasi paling optimal pada algoritma genetika untuk kasus optimasi node relay WSN. Generasi tersebut dianggap paling optimal karena generasi dengan angka terbesar mendekati parameter pengujian dan tidak ada



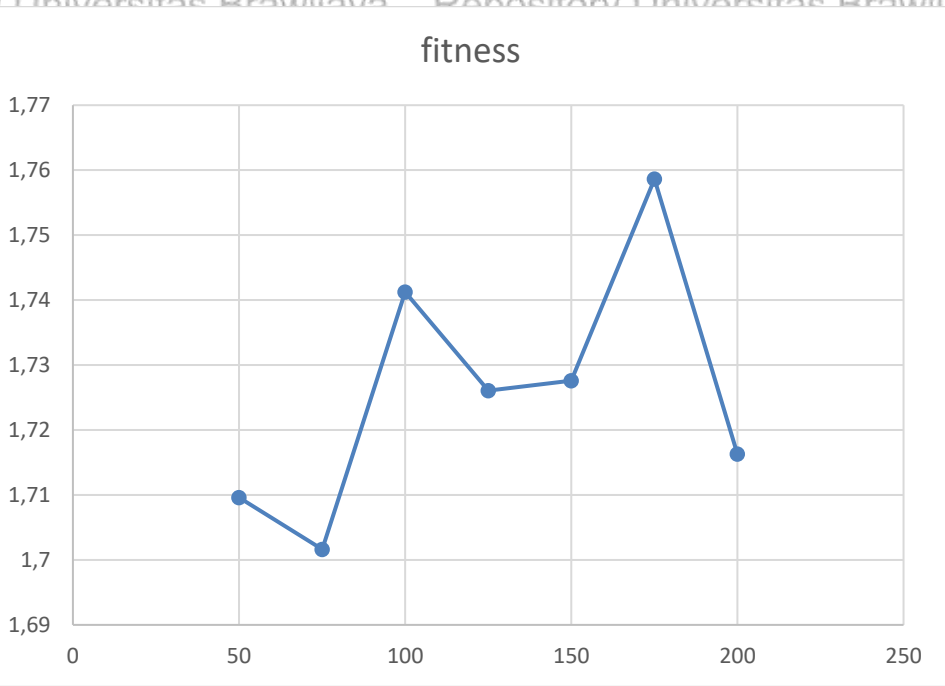
nilai kenaikan *fitness* pada generasi setelah itu. Untuk memastikan keoptimalannya maka nilai generasi akan ditambah 3 menjadi 755. Pada skenario 2 hasil pengujian konvergensi ditampilkan pada Tabel 5.2 dan Gambar 5.2. Konvergensi global terjadi setelah generasi ke-846. Generasi ke-846 dianggap sebagai generasi paling optimal pada algoritma genetika karena nilainya paling besar dan tidak ada kenaikan *fitness* setelah generasi tersebut. Untuk memastikan keoptimalannya maka nilai generasi akan ditambah 3 menjadi 849. Hasil pengujian konvergensi untuk skenario 3 dapat dilihat pada tabel dan Gambar. Konvergensi global terjadi pada generasi ke-424. Generasi ke-424 dianggap sebagai generasi teroptimasi pada algoritma genetika untuk kasus optimasi peletakkan node relay WSN. Generasi ke-427 paling optimal karena nilai generasi yang terbesar dan tidak ada nilai kenaikan *fitness* setelah generasi tersebut.

## 5.2 Pengujian Ukuran Populasi

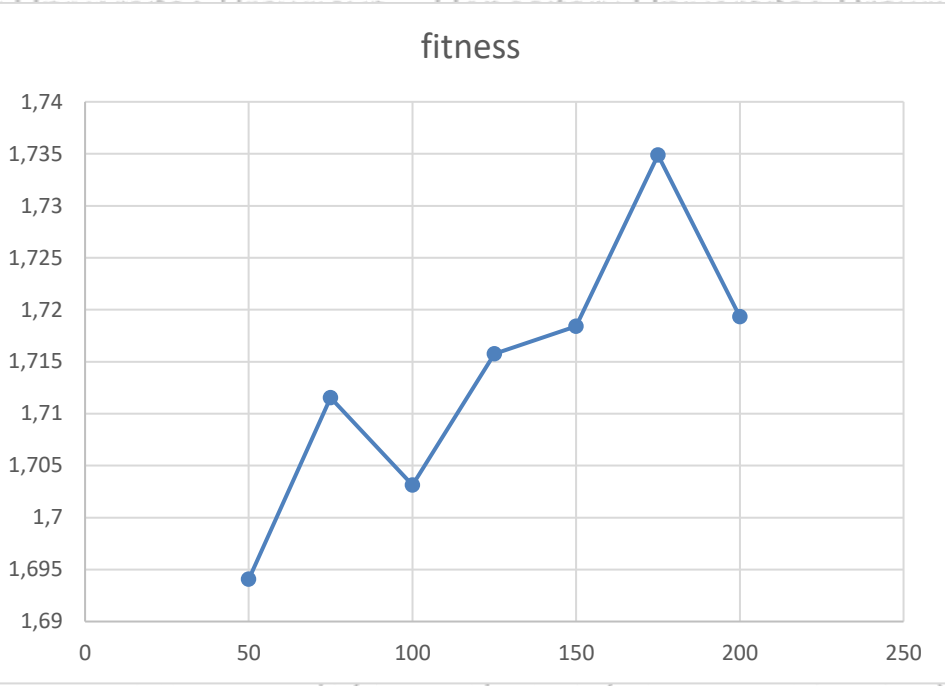
Setelah pengujian konvergensi dilakukan, selanjutnya adalah melakukan pengujian ukuran populasi. Pengujian ukuran populasi akan menghasilkan nilai parameter ukuran populasi yang paling optimal algoritma genetika untuk kasus optimasi peletakkan node relay pada jaringan WSN. Hasil dari pengujian ukuran populasi didapatkan dengan melihat rata-rata dari setiap pengujian yang dilakukan lalu dipilih dengan pertimbangan *fitness* terbaik dan waktu kalkulasi tersingkat. Pengujian ukuran populasi akan menggunakan jumlah generasi dari pengujian konvergensi yaitu 755 untuk skenario 1, 849 untuk skenario 2 dan 427 untuk skenario 3. Parameter rasio pindah silang dan rasio mutasi yang digunakan masing-masing adalah 0,5. Pengujian ukuran populasi akan mencoba setiap ukuran populasi 50, 75, 100, 125, 150, 175, dan 200 yang dilakukan terhadap masing-masing skenario. Hasil pengujian ukuran populasi ditampilkan pada Gambar 5.4 sampai 5.9



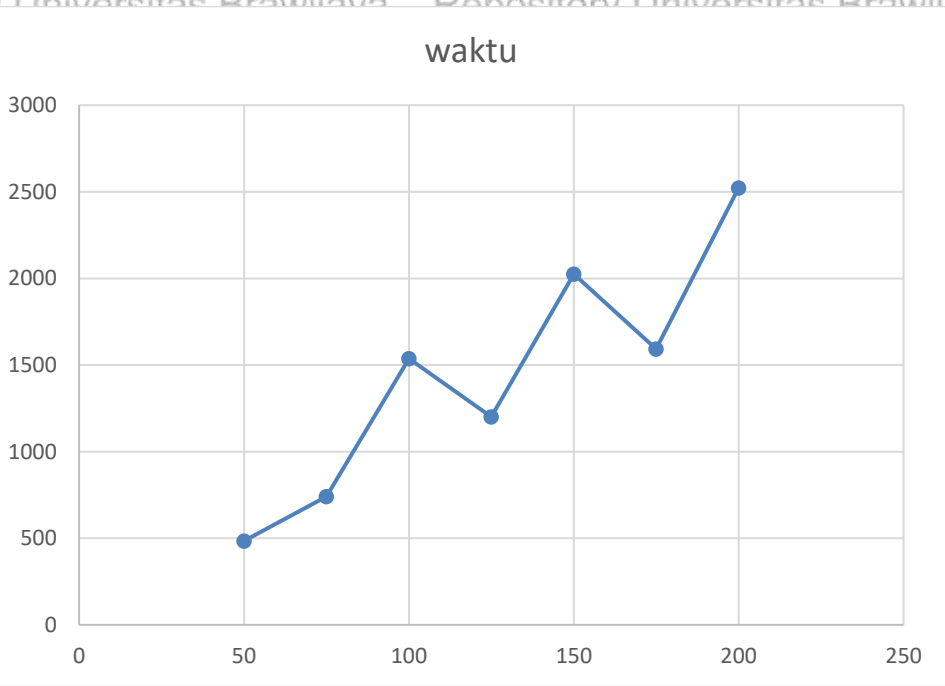
Gambar 5.4 Grafik *Fitness* Pengujian Ukuran Populasi Skenario 1



Gambar 5.5 Grafik *Fitness* Pengujian Ukuran Populasi Skenario 2

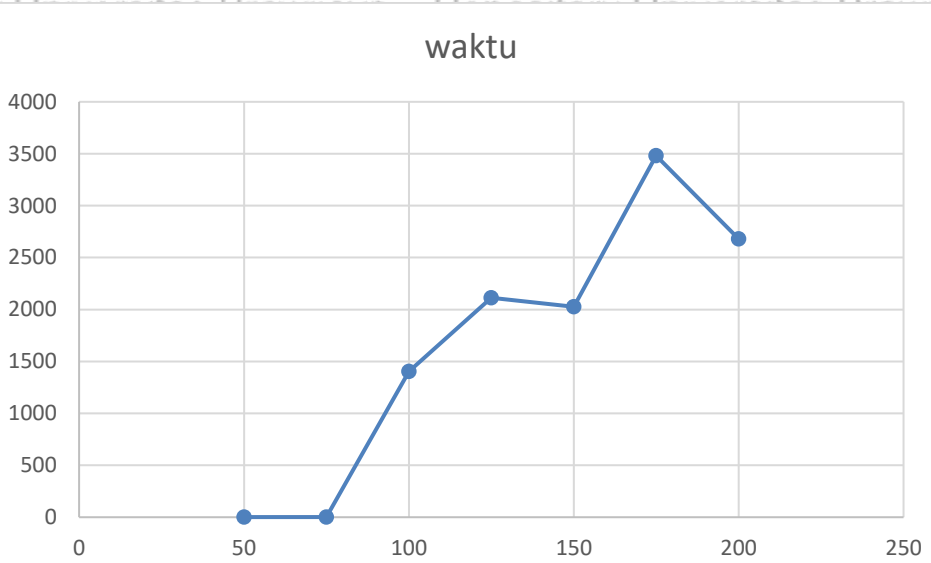


Gambar 5.6 Grafik *Fitness* Pengujian Ukuran Populasi Skenario 3

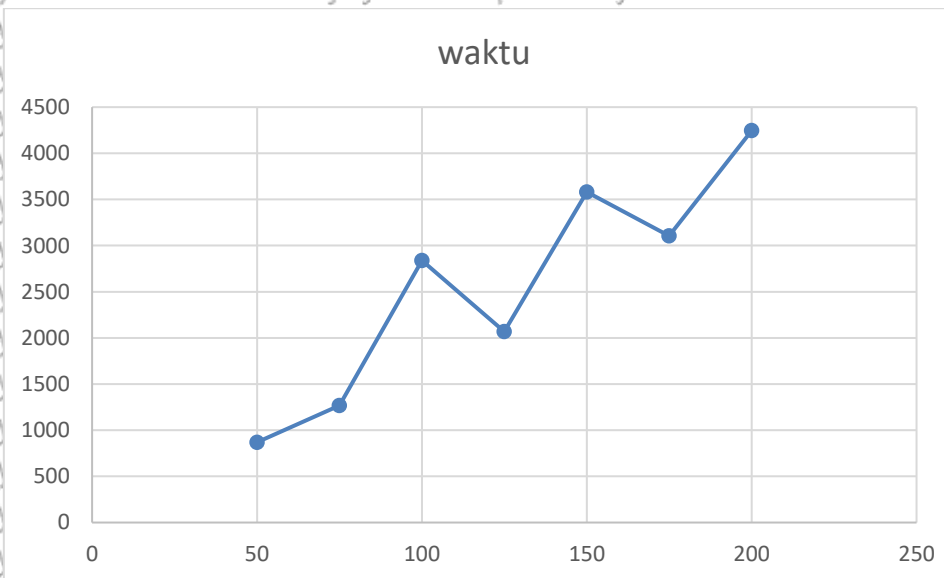


Gambar 5.7 Grafik Waktu Pengujian Konvergensi Skenario 1





Gambar 5.8 Grafik Waktu Pengujian Konvergensi Skenario 2



Gambar 5.9 Grafik Waktu Pengujian Konvergensi Skenario 3

Hasil pengujian ukuran populasi untuk skenario 1 ditampilkan pada Gambar dan Gambar 5.4 dan 5.7. Pada ambar *fitness* skenario 1 *fitness* tertinggi didapatkan pada ukuran populasi 150 dan terendah pada ukuran populasi 50, sementara untuk grafik waktu tertinggi ada di ukuran populasi 200 dan terendah ada di ukuran populasi 50. Dari kedua grafik tersebut ukuran populasi yang optimal algoritma genetika untuk kasus optimasi peletakkan node relay pada jaringan WSN adalah 150. Ukuran populasi 150 dipilih karena dirasa paling ideal dengan *fitness* 1,73445812 dan waktu 2025 detik karena ukuran populasi 150 memiliki waktu kalkulasi yang tidak terlalu lama dengan *fitness* tertinggi. Dengan pertimbangan waktu untuk mendapatkan proses kalkulasi yang efisien maka ukuran populasi yang dipilih untuk skenario 1 adalah 150.

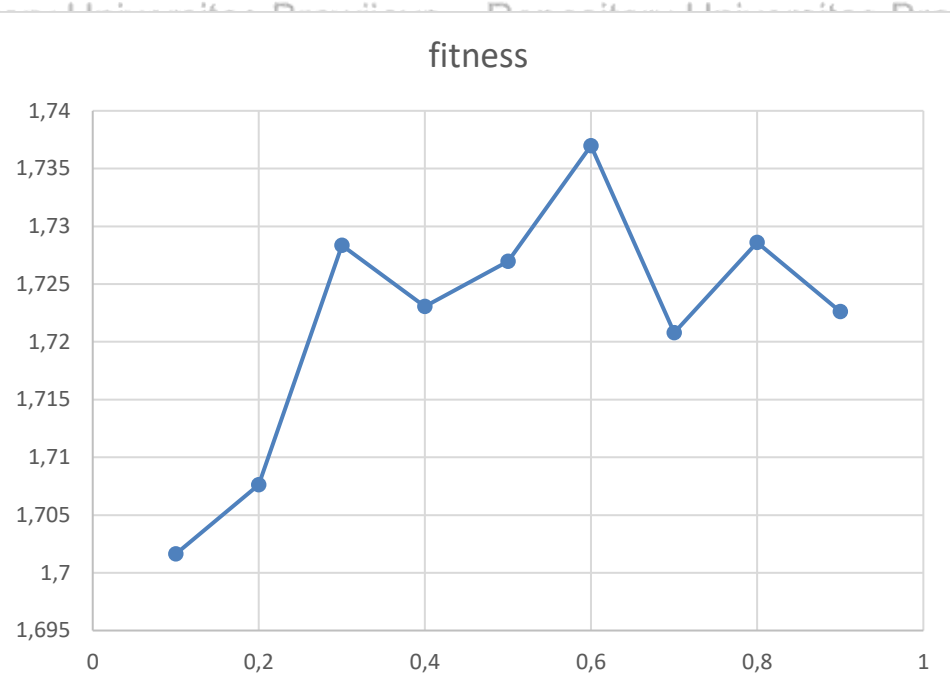


Hasil pengujian ukuran populasi untuk skenario 2 ditampilkan pada Gambar 5.5 dan Gambar 5.8. Pada grafik *fitness* skenario 2, *fitness* tertinggi didapatkan pada ukuran populasi 175 dan terendah pada ukuran populasi 75, sementara untuk grafik waktu tertinggi didapatkan pada ukuran populasi 175 dan terendah pada ukuran populasi 50. Dari kedua grafik tersebut ukuran populasi yang optimal algoritma genetika untuk kasus optimasi peletakkan node relay pada jaringan WSN adalah 100. Ukuran populasi 100 dipilih karena dirasa paling ideal dengan *fitness* sebesar 1,741253472 dan waktu kalkulasi 1405 detik walaupun ukuran populasi 175 memiliki *fitness* yang lebih tinggi daripada 175 akan tetapi ukuran populasi tersebut memiliki waktu kalkulasi yang lebih banyak. Dengan pertimbangan waktu untuk mendapatkan proses kalkulasi yang lebih efisien maka ukuran populasi yang dipilih untuk skenario 2 adalah 100.

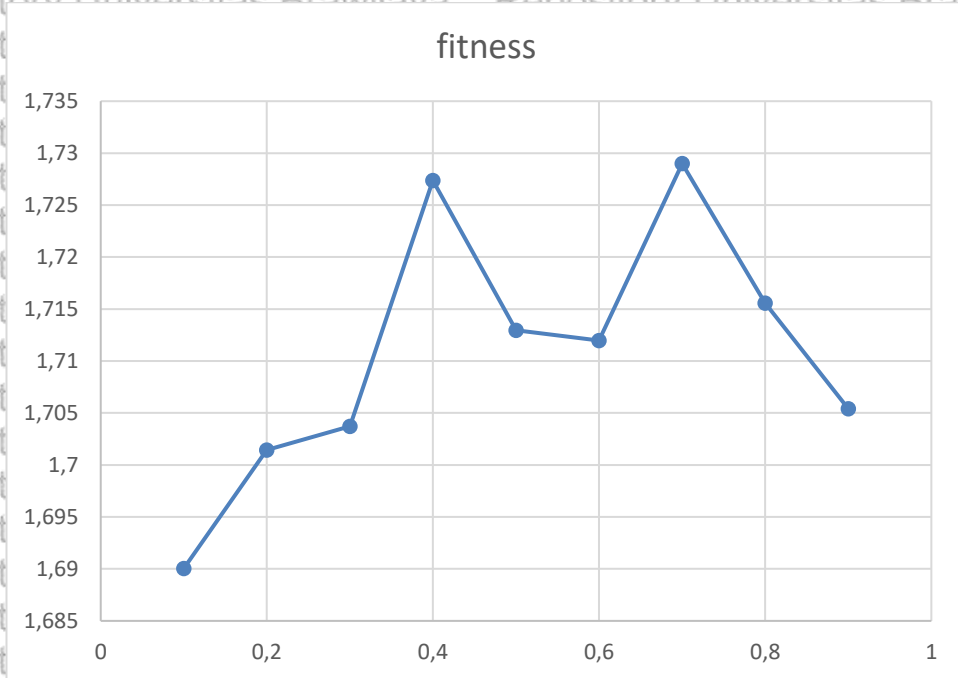
Hasil pengujian ukuran populasi untuk skenario 3 ditampilkan pada Gambar 5.6 dan Gambar 5.9. Pada grafik *fitness* skenario 3 *fitness* tertinggi didapatkan pada ukuran populasi 175 dan terendah pada ukuran populasi 50. Sementara pada grafik waktu skenario 3 waktu tertinggi didapatkan pada ukuran populasi 200 dan terendah pada ukuran populasi 50. Dari kedua grafik tersebut ukuran populasi yang optimal algoritma genetika untuk kasus optimasi peletakkan node relay pada jaringan WSN adalah 175. Ukuran populasi 175 dipilih karena dirasa paling ideal dengan *fitness* sebesar 1,734873769 dan waktu 3105 detik karena *fitness* tertinggi kedua dan ketiga adalah 200 dan 150 namun 200 dan 150 memiliki waktu kalkulasi yang lebih sangat lama. Dengan pertimbangan waktu untuk mendapatkan proses kalkulasi yang lebih efisien maka ukuran populasi yang dipilih untuk skenario 3 adalah 175.

### 5.3 Pengujian Rasio

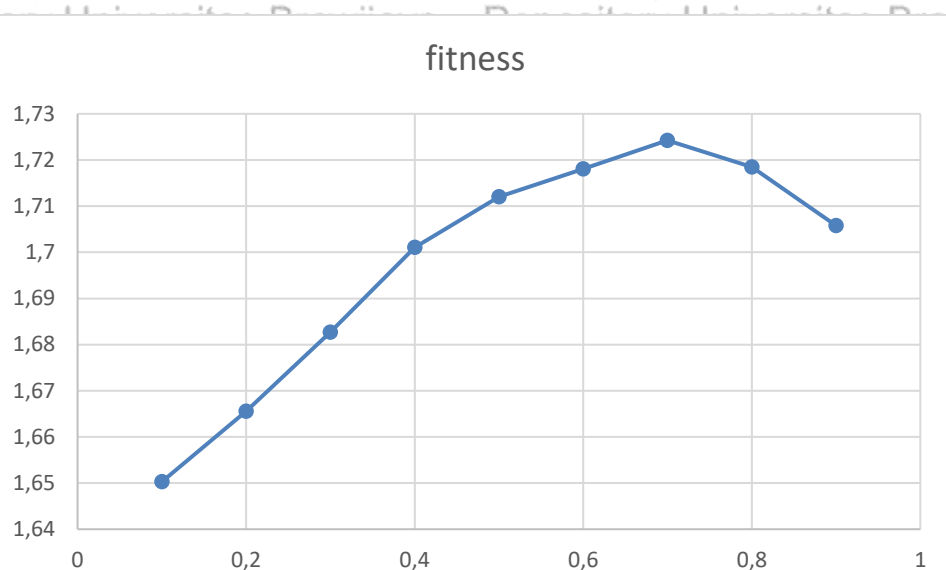
Setelah pengujian konvergensi dan ukuran populasi dilakukan, pengujian selanjutnya adalah pengujian rasio. Pengujian rasio dilakukan untuk mendapatkan parameter rasio pindah silang dan rasio mutasi yang paling optimal algoritma genetika untuk kasus optimasi peletakkan node relay pada jaringan WSN. Hasil dari pengujian rasio didapatkan dengan melihat rata-rata dari setiap pengujian dengan rasio pindah silang 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, dan 0,9. Dari rata-rata masing masing rasio pindah silang tersebut dipilih rasio pindah silang dengan nilai *fitness* tertinggi. Hasil pengujian rasio ditampilkan pada Gambar 5.1., 5.11, dan 5.12.



Gambar 5.10 Grafik Pengujian Rasio Skenario 1



Gambar 5.11 Grafik Pengujian Rasio Skenario 2



**Gambar 5.12 Grafik Pengujian Rasio Skenario 3**

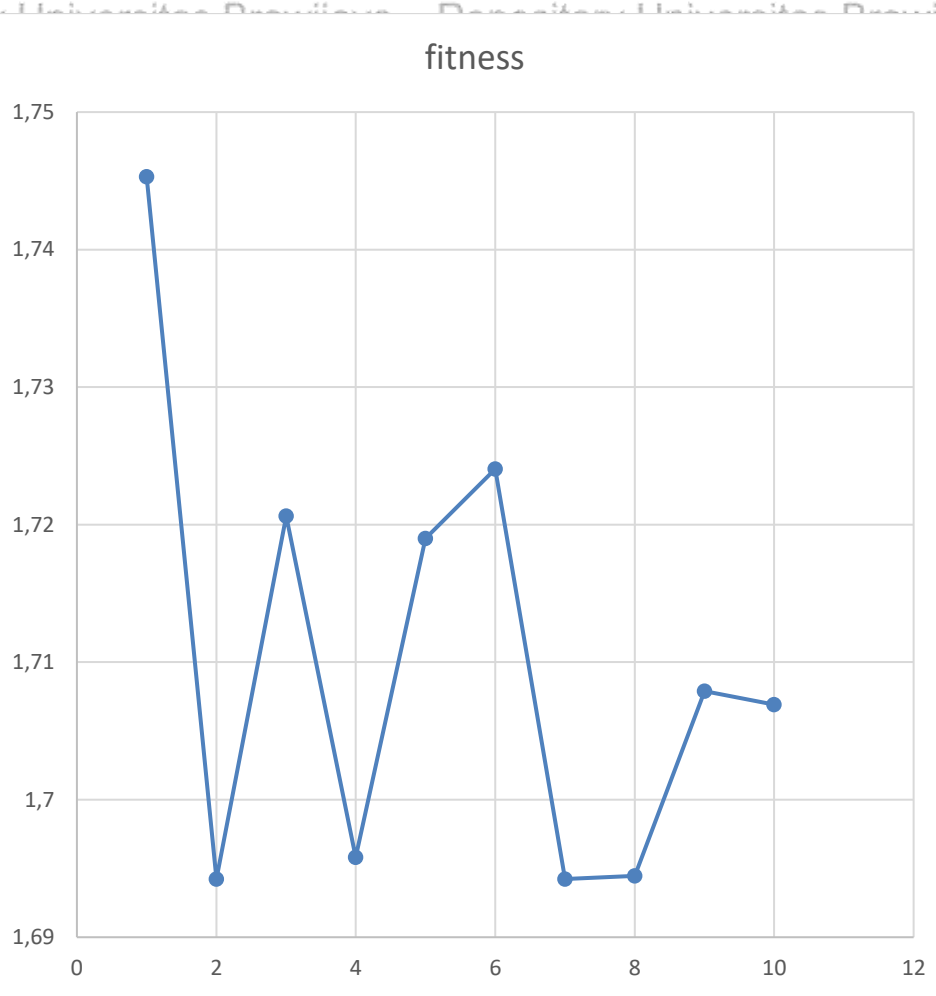
Hasil pengujian rasio untuk skenario 1 dapat dilihat pada Gambar 5.10. Pada grafik *fitness* skenario 1 *fitness* tertinggi didapatkan pada rasio pindah silang 0,6 dan rasio pindah silang terendah pada rasio pindah silang 0,1. Dari grafik ini bisa disimpulkan bahwa rasio pindah silang yang optimal algoritma genetika untuk kasus optimasi peletakkan node relay pada jaringan WSN adalah 0,6. Rasio pindah silang 0,6 dipilih karena dari hasil pengujian didapatkan nilai *fitness* tertinggi pada rasio pindah silang tersebut dengan nilai 1,736984187. Pada skenario 2 hasil pengujian rasio ditampilkan pada Gambar 5.11. Pada grafik *fitness* skenario 2 *fitness* tertinggi didapatkan pada rasio pindah silang 0,7 dan rasio pindah silang terendah pada rasio pindah silang 0,1. Dari grafik ini bisa disimpulkan bahwa rasio pindah silang yang optimal algoritma genetika untuk kasus optimasi peletakkan node relay pada jaringan Wireless Sensor Network adalah 0,7. Rasio pindah silang 0,7 dipilih karena dari hasil pengujian didapatkan nilai *fitness* tertinggi pada rasio pindah silang tersebut dengan nilai 1,728994125. Pada grafik *fitness* skenario 3 *fitness* tertinggi didapatkan pada rasio pindah silang 0,7 dan rasio pindah silang terendah pada rasio pindah silang 0,1. Dari grafik ini bisa disimpulkan bahwa rasio pindah silang yang optimal algoritma genetika untuk kasus optimasi peletakkan node relay pada jaringan Wireless Sensor Network adalah 0,7. Rasio pindah 0,7 dipilih karena dari hasil pengujian didapatkan nilai *fitness* tertinggi pada rasio pindah silang tersebut dengan nilai 1,724246588.

#### 5.4 Pengujian Akhir & Analisis

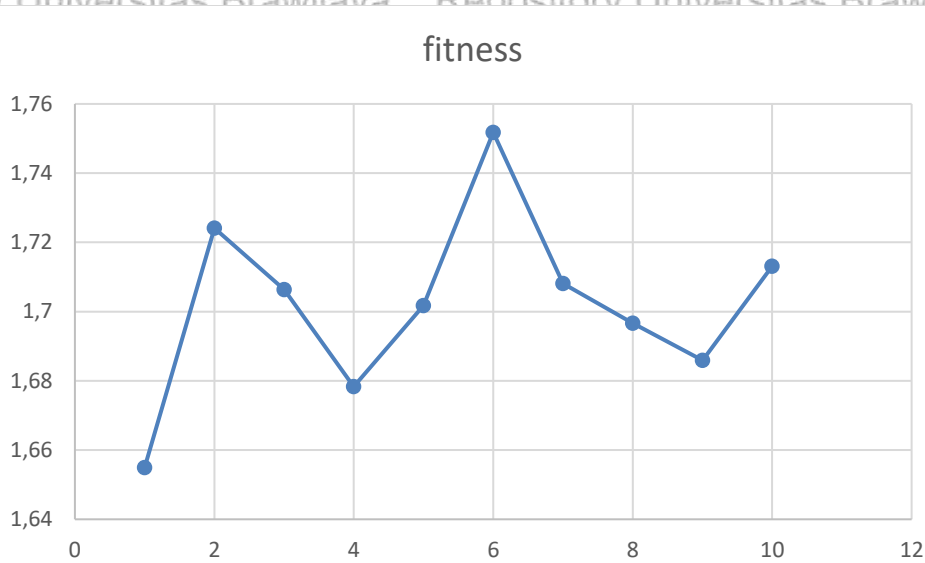
Setelah dilakukan pengujian konvergensi, ukuran populasi, dan rasio yang menghasilkan parameter generasi, ukuran popuasi, dan rasio pindah silang dan mutasi, selanjutnya adalah melakukan implementasi algoritma genetika menggunakan parameter tersebut. Pengujian akhir ini dilakukan untuk mengetahui hasil dari optimasi peletakkan node relay pada jaringan WSN menggunakan algoritma genetika. Pengujian ini akan dilakuka sebanyak sepuluh



kali pada masing-masing skenario dengan parameter-parameternya. Pengujian dengan hasil *fitness* tertinggi akan dipilih sebagai yang paling optimal. Hasil pengujian akhir ditampilkan pada Gambar 5.12, 5.13, dan 5.14.



Gambar 5.12 Grafik Pengujian Akhir Skenario 1

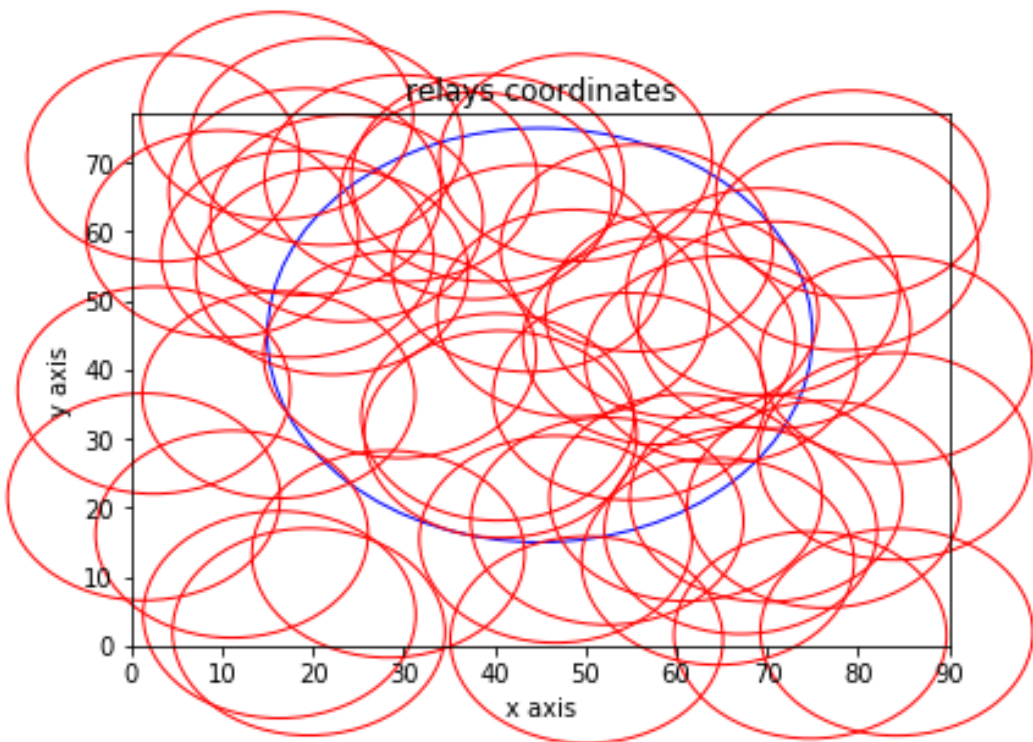


Gambar 5.13 Grafik Pengujian Akhir Skenario 2

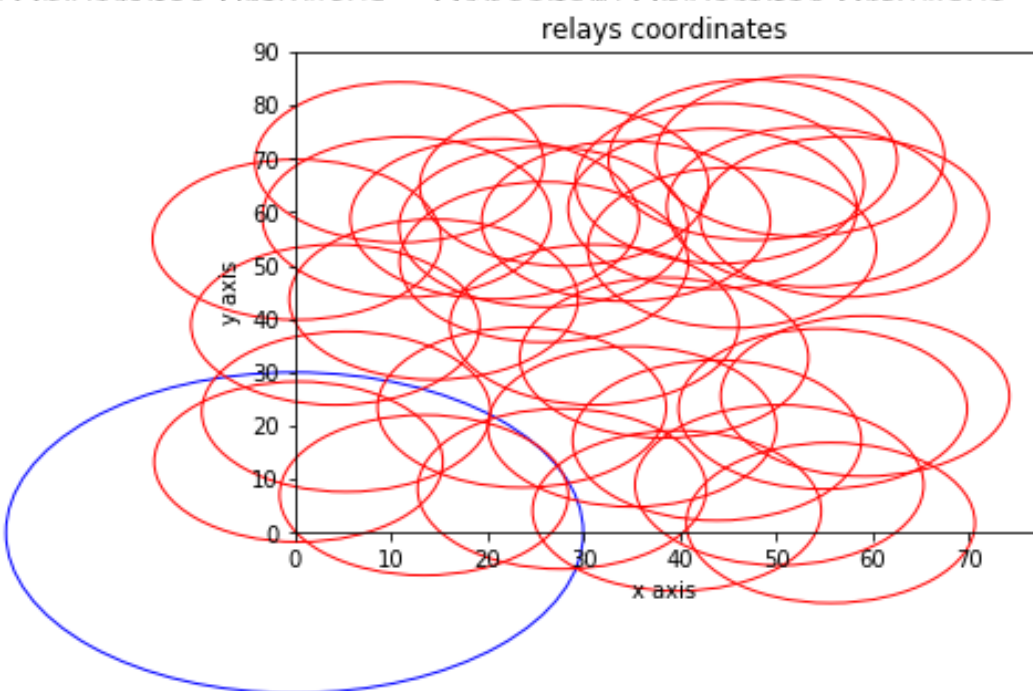


Gambar 5.15 Grafik Pengujian Rasio Skenario 3

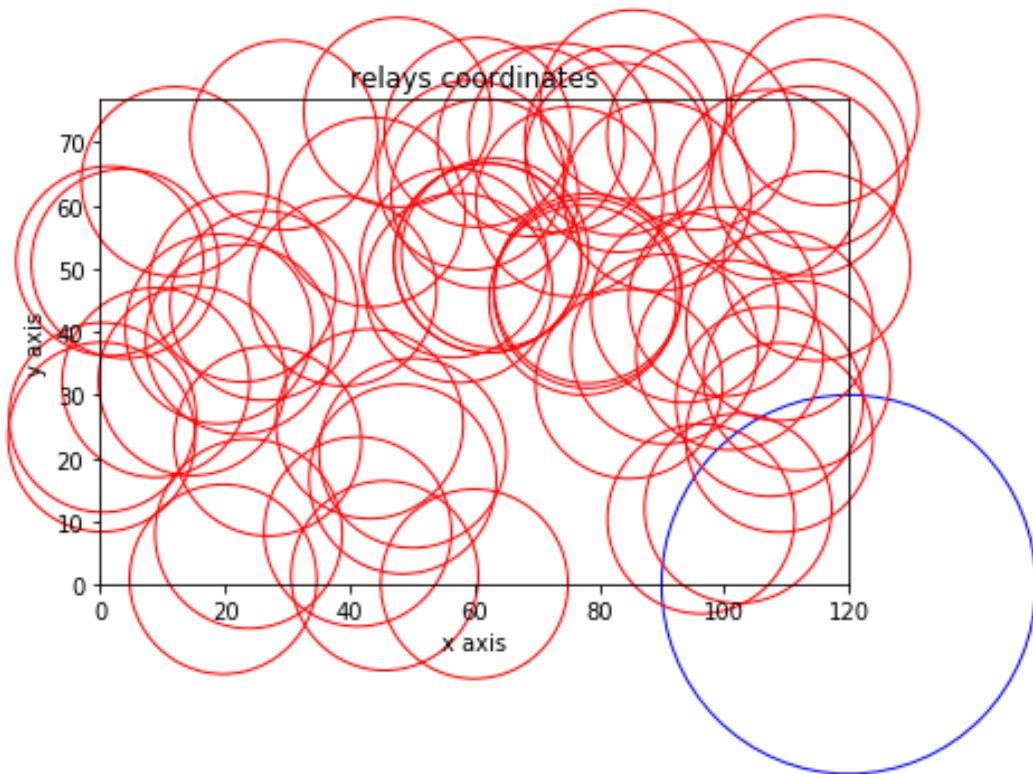
Hasil pengujian dari ketiga skenario dapat dilihat pada Gambar 5.13, 5.14, dan 5.15. Grafik pengujian skenario 1 didapatkan *fitness* tertinggi pada pengujian pertama dengan nilai *fitness* 1,745302899. Grafik pengujian skenario 2 didapatkan *fitness* tertinggi pada pengujian keenam dengan nilai *fitness* 1,75177874. Grafik pengujian skenario 3 didapatkan *fitness* tertinggi pada pengujian pertama dengan nilai *fitness* 1,759761321. Setelah didapatkan individu dengan nilai *fitness* tertinggi pada setiap skenario selanjutnya akan dipetakan posisi dari internet *gateway* dan node relay pada area implementasi WSN menggunakan fitur plotting pada python (`matplotlib`).



Gambar 5.14 Gambar Plotting Relay Skenario 1



Gambar 5.15 Gambar Plotting Relay Kenario 2



**Gambar 5.16 Gambar Plotting relay Skenario 3**

Setelah dipetakan dapat dilihat penyebaran relay dan internet *gateway* dari skenario 1, skenario 2, dan skenario 3 yang terbagi jadi beberapa tier. Koordinat dari relay dapat dilihat pada lampiran. Aturan pemetaan pada penelitian ini adalah internet *gateway* sebagai tier 0 yang ditetapkan sedari awal lalu diikuti relay yang langsung terhubung dengan internet *gateway* sebagai tier 1 lalu diikuti relay berikutnya yang terhubung sebagai tier 2 dan seterusnya. Pada skenario 1 internet *gateway* berada pada koordinat (45,45), setelah algoritma genetika dijalankan didapatkan relay pada tier1 yang mengitari internet *gateway* (360 derajat) lalu akan diikuti oleh relay selanjutnya pada tier 2 sampai tier terakhir. Pada skenario 2 internet *gateway* berada pada koordinat (0,0), setelah algoritma genetika dijalankan didapatkan relay pada tier1 sebanyak 2 relay dari arah timur laut (kuadran 2) lalu akan diikuti oleh relay selanjutnya pada tier 2 hingga tier terakhir. Pada skenario 3 internet *gateway* berada pada koordinat (120,0), setelah algoritma genetika dijalankan didapatkan satu relay pada tier 1 dari arah barat laut (kuadran 1) lalu akan diikuti oleh relay selanjutnya pada tier 2 hingga sampai tier terakhir. Untuk konektivitas dari sesama relay dan relay terhadap internet *gateway* sudah dipastikan terhubung merujuk dari rumus *fitness* yang dirancang pada subbab 4.2 dengan nilai *fitness* skenario 1 sebesar 1,745302899, *fitness* skenario 2 sebesar 1,759761321, dan *fitness* skenario 3 sebesar 1,759761321.

Setelah didapatkan koordinat relay yang sudah dioptimasi menggunakan metode heuristik menggunakan algoritma genetika, selanjutnya akan dibandingkan hasilnya dengan metode deterministik. Hasil penyebaran relay menggunakan metode deterministik dapat dilihat pada subbab 4.3. Jika dilihat dari





ketentuan *fitness* jika jarak antar 2 perangkat sama dengan radius cakupan ( $r$ ) maka *fitness* bernilai 1 dan jika jarak antar perangkat lebih dari 0,8 radius cakupan maka *fitness* bernilai 1. Hasil optimasi posisi relay dengan metode heuristik menggunakan algoritma genetika jika dibandingkan dengan metode deterministik hasilnya lebih bagus menggunakan deterministik. Hal tersebut dapat dibuktikan dari setiap relay pada metode deterministik memiliki konektivitas dengan nilai *fitness* 1 terhadap relay disekitarnya yang jika dijumlahkan dari nilai *fitness* yang didapatkan dari hubungan antar  $r$  dan juga jarak antar relaynya lebih dari 0,8 radius akan menghasilkan nilai *fitness* yang lebih besar dibandingkan nilai *fitness* optimasi dengan menggunakan algoritma genetika. Dilihat dari segi waktu peletakkan node relay menggunakan metode deterministik lebih baik karena untuk menghasilkan koordinat relay menggunakan algoritma genetika lebih lama daripada menggunakan metode deterministik.

## BAB 6 PENUTUP

Pada bab 6 sudah dipaparkan bagaimana proses dan hasil pengujian dari pengujian konvergensi, pengujian ukuran populasi, dan pengujian rasio lalu dilakukan pengujian akhir dan analisis perbandingan hasil peletakkan relay menggunakan metode heuristik dan metode deterministik. Setelah pengujian selesai dilakukan berikutnya adalah mengambil kesimpulan dari penelitian ini yang menjawab rumusan masalah pada bab 2. Untuk penelitian lanjutan akan diberikan juga saran yang berupa masukkan apa saja yang bisa ditambahkan pada penelitian di masa depan.

### 6.1 Kesimpulan

Dari hasil dan analisa pada penelitian optimasi peletakkan node relay pada jaringan WSN yang sudah dilakukan dapat diambil kesimpulan berdasarkan rumusan masalah adalah:

1. Pengimplementasian algoritma genetika untuk mengoptimasi posisi relay pada jaringan WSN diawali dengan menghitung jumlah relay dan internet *gateway* yang dibutuhkan. Pada penelitian ini yang dioptimasi adalah relay dan internet *gateway* statis maka dari sejumlah relay tersebut dibentuk menjadi kromosom dengan koordinat dari relay sebagai gen. Selanjutnya ditentukan fungsi *fitness* untuk mengukur nilai keoptimalan sebuah individu, fungsi *fitness* disesuaikan dengan constraint yang ingin diselesaikan (pada penelitian ini adalah konektivitas). Selanjutnya ditentukan metode pindah silang dan mutasi yang akan diterapkan. Selanjutnya buat method acak karena gen dalam kromosom nilainya acak. Selanjutnya lakukan pengujian konvergensi, ukuran populasi, dan rasio untuk mendapatkan parameter yang paling optimal. Terakhir lakukan pengujian dengan parameter optimal dari hasil pengujian dan didapatkan kromosom dengan nilai *fitness* terbaik yang berisi koordinat relay yang paling optimal.
2. Kinerja dari algoritma genetika pada penelitian ini bisa menghasilkan koordinat relay dengan nilai *fitness* untuk skenario 1 sebesar 1,745302899 dengan konvergensi global terjadi pada generasi ke 752 dan waktu yang dibutuhkan sebanyak 2025 detik. Pada skenario 2 didapatkan *fitness* sebesar 1,759761321 dengan konvergensi global terjadi pada generasi ke 846 dan waktu yang dibutuhkan sebanyak 1405 detik. Pada skenario 3 didapatkan *fitness* sebesar 1,759761321 dengan konvergensi global di generasi ke 424 dan waktu yang dibutuhkan sebanyak 3100 detik. Parameter rasio mutasi yang digunakan pada penelitian ini lebih besar karena mutase memberikan eksplorasi yang lebih luas dibandingkan pindah silang. Dapat disimpulkan bahwa algoritma genetika cukup bagus untuk mengoptimasi posisi relay karena *fitness* yang didapatkan mendekati sempurna namun waktu yang dibutuhkan sangat lama namun tidak sebgus menggunakan metode deterministik.



Algoritma genetika dirasa cukup handal untuk menyelesaikan permasalahan optimasi node relay pada jaringan terencana. Hasil yang didapatkan tidak bagus bukan dikarenakan tidak cocoknya algoritma genetika untuk kasus ini, hasil yang tidak bagus disebabkan oleh adanya parameter jarak antar *tier* yang tidak dimasukkan pada perhitungan nilai *fitness*.

## 6.2 Saran

Dari kesimpulan penelitian yang sudah dilakukan dapat diketahui bahwa algoritma genetika yang digunakan dirasa sudah optimal. Untuk penelitian lanjut yang bisa dilakukan untuk menyempurnakan penelitian ini ada saran yang didapatkan dari pengujian. Saran dari penelitian ini adalah:

1. Parameter cakupan, *minimum cost*, dan *maximum life-time* adalah hal yang dirasa penting untuk implementasi WSN dan bisa dijadikan sebagai constraint pada penelitian selanjutnya.



## DAFTAR REFERENSI

- Abassy, M. B. et al., 2010. Performance Analysis of Sensor Placement Strategies on a Wireless Sensor Network. *Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM)*, pp. 609-617.
- Abidin, H. Z. & Din, N. M., 2012. PROVISIONING WSN COVERAGE VIA MINIMAX BASED. *IET International Conference on Wireless Communications and Applications*, pp. 1-4.
- Abidin, H. Z. & Din, N. M., 2013. WSN Sensor Node Placement Approach using Territorial Predator Scent Marking Algorithm (TPSMA). *Malaysia International Conference on Communications (MICC)*, pp. 1-4.
- Abidin, H. Z. & Din, N. M., 2013. WSN Sensor Node Placement Approach using Territorial Predator Scent Marking Algorithm (TPSMA). *Malaysia International Conference on Communications (MICC)*, pp. 1-4.
- Alia, O. M. & Al-Ajouri, A. M., 2017. Maximizing Wireless Sensor Network Coverage With Minimum Cost Using Harmony Search Algorithm. *IEEE Sensors Journal*, Volume 17, p. 14.
- Alzaid, H., 2009. A Wireless Sensor Networks test-bed for the Wormhole Attack. *International Journal of Digital Content Technology and its Applications 3*, Volume III.
- Anton, H., 1994. *Elementary Linear Algebra*. 7 ed. s.l.:John Wiley & Sons.
- Asheer, S. & Kumar, S., 2015. Relay node placement strategies in a Wireless Sensor Networks with mobile sink. *IEEE International Conference on Advanced Communication, Control and Computing Technologies*, pp. 882-885.
- Azharudin, M. & Jana, P., 2015. A GA-based approach for fault tolerant relay node placement in wireless sensor networks. *International Conference on Computer, Communication, Control, and Information Technology*.
- Aziz, N. A. B. A., Mohammed, A. W., Sagar, B. S. D. & Alias, M. Y., 2007. Particle Swarm Optimization And Voronoi Diagram For Wireless Sensor Networks Coverage Optimization. *International Conference on Intelligent And Advanced Systems*, pp. 961-965.
- Berger, M., 1987. *Geometry I.. Universitext*.
- Bin, H., Tang, J. & Xue, G., 2014. Fault-tolerant relay node placement in wireless sensor network: formulation and approximation. *Workshop on High Performance Switching and Routing*, pp. 246-250.
- Chakrabarty, K., Iyengar, S. S., Qi, H. & Cho, E., 2002. Grid Coverage For Surveillance And Target Location In Distributed Sensor Networks. *IEEE Transactions On Computers*, Volume 51, pp. 1448-1453.



Chazelle, B., Guibas, L. J. & Lee, D. T., 1985. Power Of Geometric Duality. *BIT*, Volume 25, pp. 76-90.

De La Concepcion, A., Stefanelli, R. & Trincherio, D., 2014. Adaptive wireless sensor networks for high-definition monitoring in sustainable agriculture. *IEEE Topical Conference on Wireless Sensors and Sensor Networks*, pp. 667-69.

Deif, D. & Gadallah, Y., 2013. Classification of Wireless Sensor Networks Deployment Techniques. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, pp. 1-22.

Deif, D. & Gadallah, Y., 2016. Reliability assesment of wireless sensor network deployments. *IFIP Wireless Days*, pp. 1-6.

Desiana, E., 2016. Performace Algoritma Genetika (GA) Pada Penjadwalan Mata Pelajaran. *Jurnal Nasional dan Teknologi Jaringan*, Volume 1, p. 1.

Deyab, T. M., Baroudi, U. & Selim, S. Z., 2011. Optimal Placement Of Heterogenous Wireless Sensor And Relay Nodes. *International Conference Wirelsss Communications And Mobile Computing (IWCMC)*, Issue 65-70.

Diveev, A. & Bobr, O., 2016. Variational Genetic Algorithm For NP-Hard Scheduling Problem. *International Symposium INTELS'16*, pp. 52-58.

Gen, M. & Cheng, R., 1997. *Genetic Algorithms and Engineering Design*. New York: John Wiley & Sons, Inc.

Hamdi, M. & Mhiri, S., 2015. Dynamic mesh router placement for connectivity maximization in wireless mesh networks. *International Conference on Information and Communication Technology and Accessibility*.

Handayani, R., 2010. *Penerapan Algoritma Genetika Untuk Optimasi Jadwal Mata Kuliah Pada Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Sumatera Utara*. Medan: Universitas Sumatera Utara.

Hao, B., Tang, J. & Xue, G., 2004. Fault-tolerant relay node placement in wireless sensor networks: formulation and approximation. *Workshop on High Performance Switching and Routing*, pp. 246-250.

Ingle, M. R. & Bawane, N., 2011. An Energy Efficient Deployment Of Nodes In Wireless Sensor Network. *electronics Computer Technology (ICECT)*, pp. 307-3011.

Kershner, R., 1939. The Number Of Circles Covering A Set. *American Journal Of Mathematics*, pp. 665-671.

kurniawan, r., 2015. *Analisis Kinerja Jaringan Komputer Nirkabel Di Sekolah Tinggi Ilmu ekonomi Widya Wiwaha Yogyakarta*. Yogyakarta: Universitas negeri Yogyakarta.

Lim, S., 2015. *Genetika Algoritma*. [Online] Available at: [https://www.academia.edu/13045448/Genetika Algoritma](https://www.academia.edu/13045448/Genetika_Algoritma) [Diakses 25 Agustus 2019].



Lin, C., Chen, T. & Chin, H., 2016. Adaptive router noe placement with gateway positions and QoS constraints in dynamic wireless mesh network. *Journal of Network and Computer Applications*, Issue 74, pp. 7149-164.

Lin, C., Chen, T. & Jhong, S., 2015. Wireless mesh router placement with constraints of gateway positions and QoS. *International Conference on Heterogeneous Networking for Quality, Reliability, Security, and Robustness*, p.72.

M. Viera, M., Coelho. Junior, C., da Silva. Junio, D. & da Mata, J., 2003. Survey on Wireless Sensor Network Devices. *Wireless Personal Communications*, 2(101), pp. 1019-1055.

Ma, C., Liang, W., Zheng, M. & Sharif, H., 2016. A Connectivity-Aware Approximation Algorithm for Relay Node PLacement in Wireless Sensor Networks. *IEEE Sensors Journal*, 2(16), pp. 515-528.

Mahmudy, W. F., 2016. *Modul Algoritma Evolusi - Semester Ganjil 2015-2016*. [Online] Available at: [wayanfm.lecturer.ub.ac.id](http://wayanfm.lecturer.ub.ac.id) [Diakses 23 Agustus 2019].

Mampentzidou, I., Karaspitoli, E. & Economides, A., 2012. Basic guidlines for deploying Wireless Sensor Networks in agriculture. *International Congress on Ultra Modern Telecommunications and Control Systemsand Workshops*, pp. 864-869.

Niculescu, D., 2004. Positioning in ad hoc sensor network. *IEEE Network*, 4(18), pp. 24-29.

Nitesh, K. & Jana, P. K., 2014. Relay Node Placement Algorithm in Wireless Sensor Network. *IEEE International Advance Computing Conference (IACC)*, pp. 1-5.

Ogryczak, W., 1997. On The Lexicographic Minimax Approach To Location Problems. *European Journal Of Operational Research*, Volume 100, pp. 566-585.

Ramuna, M. D. T. & Mahmudy, W. F., 2015. Optimasi Persediaan Barang dalam Produksi Jilbab Menggunakan Algoritma Genetika. *Repository jurnal Mahasiswa PTIK Universitas Brawijaya*, 5(14), p. 3.

Romoozi, M., Vahidipour, M. & Maghsoodi, S., 2010. Genetic Algorithm For Energy Efficient And Coverage-Preserved Positoning In Wireless Sensor networks. *Intelligent Computing And Cognitive Informatics (ICICCI)*, pp. 22-25.

Saharuna, Z., Widyawan & sumaryono, S., 2012. Simulasi Deployment Jaringan Sensor Nirkabel Berdasarkan Algoritma Particle Swarm Optimization. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, 1(3).

Sarwesh, P., Shet, N. V. & Chandrasekaran, K., 2015. Energy-efficient network achitecture for IoT applications. *Internet of Things*, pp. 119-144.

Setyawan, S. R. D., 2018. *kompas.com*. [Online] Available at: <https://ekonomi.kompas.com/read/2018/02/19/161115126/tahun->



2017-pengguna-internet-di-indonesia-mencapai-14326-juta-orang

[Diakses 10 Juli 2019].

Setyawan, T., Cholissodin, I. & Tibyani, 2019. Implementasi Algoritme Genetika Dalam Penjadwalan Akademik Sekolah. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(7), pp. 6633-6638.

Sharma, R., Rangga, V. & Angra, S., 2015. Placement of relay nodes in wireless sensor network. *International Conference on Green Computing and Iternet of Things*, pp. 999-1004.

Shinji, S. et al., 2016. Node Placement in Wireless Mesh Networks: A Coparison Study of WMN-SA and WMN-PSO SImulation System. *IEEE*, p. 5.

Sitepu, A. C., 2018. *Repository Jurnal Universitas Sumatera utara*. [Online] Available at:

<http://repositori.usu.ac.id/bitstream/handle/123456789/6633/130803063.pdf?sequence=1&isAllowed=y>

[Diakses 27 agustus 2019].

Talbi, E.-G., 2009. *Metaheuristic: from design to implementation*. Hoboken, New Jersey: Joh Wiley & Son, Inc.

Talgin, A., Shakarami, V., Sheikholeslam, F. & Chatraei, A., 2014. Aerial Node Placement in Wireless Sensor Networks Using Fuzzy K-Means Clustering. *International Conference on e-Commerce in Developing Countries: With Focus on e-Trust*, pp. 2-7.

Tang, J., Hao, B. & Sen, A., 2006. Relay Node Placement In Large Scale Wireless Sensor Networks. *Computer Communications*, Issue 29, pp. 490-501.

Tobing, R., 2010. *Sistem Simulasi Penjadwalan Kuliah Dengan Menggunakan Algoritma Genetika*. Medan: Universitas Sumatera Utara.

USU, 2010. *Repository Jurnal Universitas Sumatera Utara*. [Online] Available at:

<http://repository.usu.ac.id/bitstream/handle/123456789/27332/Chapter%2011.pdf?sequence=3&isAllowed=y>

[Diakses 27 agustus 2019].

Wang, G., Huang, L., Xu, H. & Li, J., 2008. Relay node placement for maximizing network lifetime in wireless sensor network. *International Conference on Wireless Communications, Networking, and Mobile Computing*, pp. 1-5.

Wang, Z., Shao, X., LV, C. & Chen, J., 2013. Relay node placement algorithm based on grid in wireless sensor network. *International Conference on Instrumentation and Measurement*, pp. 278-283.

Widada, A. A. H., Amron, K. & Baktiar, F. A., 2019. Optimasi Penempatan Relay Node Pada Wireless Sensor Networks Menggunakan Algoritma Particle Swarm Optimization (PSO). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(7), pp. 7310-7318.



Widada, A. A. H., Amron, K. & Baktiar, F. A., 2019. Optimasi Penempatan Relay Node Pada Wireless Sensor Networks Menggunakan Algoritma Particle Swarm Optimization (PSO). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(7), pp. 7310-7318.

Yu, X., Huang, W., Lan, J. & Qian, X., 2012. A novel virtual force approach for node deployment in wireless sensor network. *IEEE International Conference on Distributed Computing in Sensor System*, pp. 359-363.





## LAMPIRAN A SOURCE CODE

```

Chromosome.py
import copy
import random
from itertools import chain
import math

class Chromosome:
    #fix
    #rad adalah jarijari
    def __init__(self, xIGw, yIGw, RadIGw, gen, xMax, yMax, Rad):
        self.xIGw=xIGw
        self.yIGw=yIGw
        self.radIGw=RadIGw
        self.gene = gen
        self.xmax = xMax
        self.ymax = yMax
        self.rad=Rad
        self.fitness1=0
        self.fitness2=0
        self.fitness = self.calcfit()
        print(self.gene)

    def euclid(self, x, y):
        return math.sqrt(((x[0]-y[0])*(x[0]-y[0]))+((x[1]-
y[1])*(x[1]-y[1])))

    def calcfit(self):
        fitness=0
        fitness_1=0
        fitness_1_count=0
        #mencatat sambungan node

```



```

#-1 berarti tidak terkoneksi dengan IGw
#n bernilai sesuai tier yang tersambung 0=IGw
#print("ketik",self.gene)
before=[-2 for i in range(int(len(self.gene)/2))]
after=[i for i in range(int(len(self.gene)/2))]
#untuk mencatat index
notTaken=[i for i in range(int(len(self.gene)/2))]
#variable untuk menyimpan tier dengan indeks
tier=[[ ] for i in range(int(len(self.gene)/2))]
# i adalah index tier
for i in range(int(len(self.gene)/2)):
    if(i==0):
        #j adalah index relay
        for j in range(int(len(self.gene)/2)):
            S=self.euclid([self.xIGw,self.yIGw,self.gene[j*2:(j*2)+2])
            if(S<=self.radIGw):
                before[j]=-1
                tier[i].append(j)
                notTaken.remove(j)
                fitness+=S/self.radIGw
            print("belum diambil",notTaken)
            print("tier yang terisi",tier)
        else:
            #tier setelahnya masih masalah
            for posj,j in enumerate(notTaken):
                terdekat=-1
                indterdekat=-1
                print("tier sekarang adalah",i)
                jarak=[self.euclid(self.gene[j*2:(j*2)+2],self.gene[k*2:(k*2)+2])
                for k in tier[i-1]]
                print(jarak)
                for k in range(len(jarak)):
                    if(terdekat<jarak[k]<=self.rad):
                        terdekat=jarak[k]
                        indterdekat=k

```



```

#print(terdekat, indterdekat)
if(terdekat!= -1):
    #print("masuk masuk")
    before[j]=tier[i-1][indterdekat]
    tier[i].append(notTaken[posj])
    notTaken.remove(notTaken[posj])
    fitness+=terdekat/self.rad
    if not tier[i]:
        break
    #print("kosong")
    #print("fitness ", fitness)
    #print(before)
    #print(after)
    #print(tier)
    for i in range(len(tier)):
        for j in range(len(tier[i])):
            if (j<(len(tier[i])-1)):
                for k in range(j+1, len(tier[i])):
                    fitness_1_count+=1
                    if (self.euclid(self.gene[tier[i][j]]*2:(tier[i][j]*2)+2), self.gene[
                    tier[i][k]*2:(tier[i][k]*2)+2])>(0.8*self.rad)):
                        fitness_1+=1
                    print(tier[i], len(tier[i]))
                    print("tier ke", i, fitness_1, fitness_1_count)
                    if fitness_1_count==0:
                        fitness_1_count=1
                    print("fitness kedua", fitness_1, fitness_1_count)
                    self.fitness1=fitness/(len(self.gene)/2)
                    self.fitness2=(fitness_1/fitness_1_count)
                return
            (fitness/(len(self.gene)/2))+(fitness_1/fitness_1_count)

def mutate(self):
    childgene=copy.deepcopy(self.gene)
    for i in range(len(childgene)):
        if(i%2==0):
            childgene[i]=
            childgene[i]+(random.random()*self.xmax)

```



```

if (childgene[i]>self.xmax):
    number=childgene[i]//self.xmax
    if (childgene[i]%self.xmax==0):
        number-=1
    childgene[i]-=self.xmax*number
elif (i%2==1):
    childgene[i]=
childgene[i]+(random.random()*self.ymax)
if (childgene[i]>self.ymax):
    number=childgene[i]//self.ymax
    if (childgene[i]%self.ymax==0):
        number-=1
    childgene[i]-=self.ymax*number
    childgene[i]=round(childgene[i],2)
#print("hasil mutasi",childgene)
return
Chromosome(self.xIGw,self.yIGw,self.radIGw,childgene,self.xmax,s
elf.ymax,self.rad)
# fix
#uniform crossover
def mate(self, partner):
    childgene1=copy.deepcopy(self.gene)
    childgene2=copy.deepcopy(partner.gene)
    for i in range(len(childgene1)):
        childgene1[i]=round(childgene1[i]+(random.random()*((childgene2[i]
]-childgene1[i])),2)
        childgene2[i]=round(childgene2[i]+(random.random()*((childgene1[i]
]-childgene2[i])),2)
    if (i%2==0):
        if (childgene1[i]>self.xmax):
            number=childgene1[i]//self.xmax
            if (childgene1[i]%self.xmax==0):
                number-=1
            childgene1[i]-=(self.xmax*number)
        if (childgene2[i]>self.xmax):
            number=childgene2[i]//self.xmax
            if (childgene2[i]%self.xmax==0):

```



```

        number-=1
        childgene2[i]==self.xmax*number
    elif(i%2==1):
        if(childgene1[i]>self.ymax):
            number=childgene1[i]//self.ymax
            if(childgene1[i]%self.ymax==0):
                number-=1
            childgene1[i]==self.ymax*number
        if(childgene2[i]>self.ymax):
            number=childgene2[i]//self.ymax
            if(childgene2[i]%self.ymax==0):
                number-=1
            childgene2[i]==self.ymax*number
    #print("hasil xover", childgene1, childgene2)
    return
    Chromosome(self.xIGw, self.yIGw, self.radIGw, childgene1, self.xmax,
self.ymax, self.rad),
    Chromosome(self.xIGw, self.yIGw, self.radIGw, childgene2, self.xmax,
self.ymax, self.rad)

@staticmethod
def generateRandom(xIGw, yIGw, RadIGw, numRelay, xMax, yMax, Rad):
    f = lambda x: random.randint(0, xMax)
    g = lambda x: random.randint(0, yMax)
    gene=list(chain.from_iterable((f(x), g(x)) for x in
range(numRelay)))
    return
    Chromosome(xIGw, yIGw, RadIGw, copy.deepcopy(gene), xMax, yMax, Rad)

```

Population.py

```

import copy
import Chromosome as chro
import math
import random

class Population:

```



```

def __init__(self, Psize, Cr, Mr, xIGw, yIGw, RadIGw,
numRelay, xMax, yMax, Rad):
    self.popsize = Psize
    self.cr = Cr
    self.mr = Mr
    self.pop = []

    for i in range(0, self.popsize):
        self.pop.append(chro.Chromosome.generateRandom(xIGw, yIGw, RadIGw,
numRelay, xMax, yMax, Rad))
        self.pop = self.sorting(copy.deepcopy(self.pop))

    def sorting(self, data):
        for i in range(len(data)):
            for j in range(i+1, len(data)):
                if (data[i].fitness < data[j].fitness):
                    data[i], data[j] = copy.deepcopy(data[j]), copy.deepcopy(data[i])
        return data

    def evolve(self):
        descendant = []
        cros = math.floor((self.cr * self.popsize) / 2)
        muta = math.floor(self.mr * self.popsize)
        i = 0
        while i < cros:
            par1 = random.randint(0, self.popsize - 1)
            par2 = par1
            while par1 == par2:
                par2 = random.randint(0, self.popsize - 1)
            #print("posisi parent", par1, par2)
            kid = self.pop[par1].mate(self.pop[par2])
            descendant.append(kid[0])
            descendant.append(kid[1])
            i += 1
        i = 0
        while i < muta:

```



```

descendant.append(self.pop[random.randint(0,
self.popsize - 1)].mutate())
i += 1
descendant += self.pop[:i]
self.pop = sorted(descendant, key=lambda x: x.fitness,
reverse=True)[:self.popsize]

```



## LAMPIRAN B HASIL PENGUJIAN KONVERGENSI SKENARIO 1

Generasi ke	Pengujian pertama	Pengujian kedua	Pengujian kesembilan	Pengujian kesepuluh
1	1,595756	1,59089	1,593485	1,618516
2	1,607372	1,616812	1,593556	1,626907
3	1,607372	1,616812	1,611159	1,655355
4	1,607372	1,616812	1,611159	1,655355
5	1,607372	1,625774	1,633177	1,655355
6	1,60895	1,625774	1,633177	1,655355
7	1,613046	1,625774	1,633177	1,655355
8	1,613046	1,625774	1,659714	1,655355
9	1,613046	1,625774	1,659714	1,655625
10	1,646213	1,625774	1,659714	1,655625
990	1,703504	1,696072	1,687352	1,711961
991	1,703504	1,696072	1,687352	1,711961
992	1,703504	1,696072	1,687352	1,711961
993	1,703504	1,696072	1,687352	1,711961
994	1,703504	1,696072	1,687352	1,711961
995	1,703504	1,696072	1,687352	1,711961
996	1,703504	1,696072	1,687352	1,711961
997	1,703504	1,696072	1,687352	1,711961
998	1,703504	1,696072	1,687352	1,711961
999	1,703504	1,696072	1,687352	1,711961
1000	1,703504	1,696072	1,687352	1,711961





## LAMPIRAN C HASIL PENGUJIAN KONVERGENSI SKENARIO 2

Generasi ke	Pengujian pertama	Pengujian kedua	...	Pengujian kesembilan	Pengujian kesepuluh
1	1,500724	1,549057		1,594639	1,523603
2	1,503044	1,549057		1,594639	1,523603
3	1,583149	1,549057		1,594639	1,59797
4	1,583149	1,549057		1,594639	1,59797
5	1,583149	1,549057		1,594639	1,59797
6	1,602506	1,549057		1,629754	1,59797
7	1,602506	1,549057		1,629754	1,617103
8	1,602506	1,627603		1,632952	1,617103
9	1,602506	1,627603		1,632952	1,617103
10	1,602506	1,627603		1,632952	1,617103
990	1,704752	1,75654		1,725665	1,680639
991	1,704752	1,75654		1,725665	1,680639
992	1,704752	1,75654		1,725665	1,680639
993	1,704752	1,75654		1,725665	1,680639
994	1,704752	1,75654		1,725665	1,680639
995	1,704752	1,75654		1,725665	1,680639
996	1,704752	1,75654		1,725665	1,680639
997	1,704752	1,75654		1,725665	1,680639
998	1,704752	1,75654		1,725665	1,680639
999	1,704752	1,75654		1,725665	1,680639
1000	1,704752	1,75654		1,725665	1,680639



### LAMPIRAN D HASIL PENGUJIAN KONVERGENSI SKENARIO 3

Generasi ke	Pengujian pertama	Pengujian kedua	...	Pengujian kesembilan	Pengujian kesepuluh
1	1,530727	1,505397		1,520734	1,550253
2	1,551079	1,539048		1,537151	1,567863
3	1,567956	1,592418		1,567579	1,584691
4	1,606783	1,592418		1,567579	1,584691
5	1,606783	1,646081		1,567579	1,586402
6	1,606783	1,646081		1,586618	1,587991
7	1,606783	1,646081		1,586618	1,609605
8	1,606783	1,646081		1,586618	1,609605
9	1,606783	1,646081		1,586618	1,620412
10	1,606783	1,646081		1,586618	1,620412
990	1,69622	1,720733		1,661919	1,700416
991	1,69622	1,720733		1,661919	1,700416
992	1,69622	1,720733		1,661919	1,700416
993	1,69622	1,720733		1,661919	1,700416
994	1,69622	1,720733		1,661919	1,700416
995	1,69622	1,720733		1,661919	1,700416
996	1,69622	1,720733		1,661919	1,700416
997	1,69622	1,720733		1,661919	1,700416
998	1,69622	1,720733		1,661919	1,700416
999	1,69622	1,720733		1,661919	1,700416
1000	1,69622	1,720733		1,661919	1,700416



## LAMPIRAN E HASIL PENGUJIAN UKURAN POPULASI SKENARIO 1

Ukuran populasi	Pengujian pertama	Pengujian kedua	Pengujian ketiga	Pengujian keempat	Pengujian kelima
50	1,689608	1,709559	1,686242	1,766312	1,677718
75	1,710513	1,727224	1,713917	1,694634	1,696989
100	1,71896	1,717442	1,698355	1,74518	1,708082
125	1,729875	1,688522	1,692629	1,741371	1,733011
150	1,758627	1,722903	1,737825	1,705816	1,734459
175	1,700006	1,720089	1,741832	1,685653	1,778054
200	1,712065	1,721304	1,689229	1,71442	1,744884

Ukuran populasi	Pengujian keenam	Pengujian ketujuh	Pengujian kedelapan	Pengujian kesembilan	Pengujian kesepuluh
50	1,686958	1,707599	1,70561	1,697485	1,696532
75	1,678749	1,697421	1,735537	1,715972	1,739604
100	1,688926	1,721141	1,73957	1,71619	1,715503
125	1,731339	1,718248	1,699605	1,712816	1,695484
150	1,720919	1,737439	1,747462	1,752329	1,726801
175	1,747826	1,739119	1,72851	1,724582	1,750357
200	1,726094	1,767806	1,740972	1,771765	1,719256



## LAMPIRAN F HASIL PENGUJIAN UKURAN POPULASI SKENARIO 2

Ukuran populasi	Pengujian pertama	Pengujian kedua	Pengujian ketiga	Pengujian keempat	Pengujian kelima
50	1,69109	1,710962	1,782869	1,739731	1,78667
75	1,689285	1,691471	1,678182	1,684281	1,714799
100	1,723397	1,686798	1,71084	1,705917	1,68669
125	1,729446	1,668617	1,721581	1,73963	1,757547
150	1,735235	1,688503	1,719949	1,747581	1,748742
175	1,772375	1,679507	1,753955	1,756183	1,795604
200	1,718791	1,703051	1,736243	1,71276	1,675581

Ukuran populasi	Pengujian keenam	Pengujian ketujuh	Pengujian kedelapan	Pengujian kesembilan	Pengujian kesepuluh
50	1,741283	1,72567	1,715504	1,732277	1,786479
75	1,74398	1,710496	1,70358	1,680993	1,719255
100	1,695266	1,745654	1,724098	1,721736	1,695659
125	1,728592	1,69488	1,74528	1,7434	1,731896
150	1,766716	1,695987	1,731159	1,733565	1,708493
175	1,763432	1,753542	1,764568	1,780121	1,766934
200	1,756787	1,770095	1,685174	1,710697	1,693767



## LAMPIRAN G HASIL PENGUJIAN UKURAN POPULASI SKENARIO 3

Ukuran populasi	Pengujian pertama	Pengujian kedua	Pengujian ketiga	Pengujian keempat	Pengujian kelima
50	1,682362	1,721646	1,692379	1,701933	1,710687
75	1,672599	1,681591	1,7712	1,707715	1,714311
100	1,713666	1,71583	1,661635	1,708733	1,71493
125	1,731234	1,685159	1,707228	1,708335	1,713254
150	1,734691	1,700636	1,711472	1,732816	1,709408
175	1,759735	1,746158	1,673775	1,770995	1,764402
200	1,735522	1,692425	1,716953	1,720014	1,736911

Ukuran populasi	Pengujian keenam	Pengujian ketujuh	Pengujian kedelapan	Pengujian kesembilan	Pengujian kesepuluh
50	1,710095	1,676037	1,665851	1,674783	1,704919
75	1,703015	1,693149	1,731145	1,708764	1,731957
100	1,67951	1,711534	1,690816	1,735847	1,698845
125	1,724535	1,689852	1,735719	1,734182	1,727973
150	1,698085	1,71705	1,704424	1,755589	1,719793
175	1,733788	1,73684	1,748926	1,7036	1,710519
200	1,702437	1,7459	1,723826	1,747997	1,671395



## LAMPIRAN H HASIL PENGUJIAN RASIO SKENARIO 1

Uji ke	{0,1;0,9}	{0,2;0,8}	{0,3;0,7}	{0,4;0,6}	{0,5;0,5}
1	1,706403	1,739588	1,696257	1,717956	1,74494
2	1,767576	1,704935	1,710167	1,696954	1,740565
3	1,722808	1,700983	1,71745	1,720038	1,711068
4	1,716963	1,700265	1,726305	1,735481	1,712773
5	1,682843	1,698409	1,733025	1,730257	1,75313
6	1,751985	1,749227	1,736934	1,766548	1,705527
7	1,702226	1,738179	1,739293	1,731511	1,746409
8	1,709995	1,661397	1,741224	1,713544	1,726229
9	1,701673	1,7384	1,708709	1,712681	1,732537
10	1,708684	1,749969	1,711687	1,705545	1,696479

Uji ke	{0,6;0,4}	{0,7;0,3}	{0,8;0,2}	{0,9;0,1}
1	1,713694	1,706898	1,740243	1,696483
2	1,731474	1,769112	1,705055	1,710848
3	1,709805	1,724037	1,70634	1,717502
4	1,727993	1,717183	1,70071	1,726941
5	1,756809	1,71056	1,699362	1,733329
6	1,73225	1,75238	1,749315	1,736966
7	1,755713	1,702474	1,743053	1,739431
8	1,784532	1,710641	1,752716	1,741225
9	1,712245	1,702259	1,73895	1,711354
10	1,745325	1,712413	1,750336	1,712144



## LAMPIRAN I HASIL PENGUJIAN RASIO SKENARIO 2

Uji ke-	{0,1;0,9}	{0,2;0,8}	{0,3;0,7}	{0,4;0,6}	{0,5;0,5}
1	1,762368	1,65263	1,702335	1,76272	1,724453
2	1,714341	1,670537	1,77301	1,744686	1,710906
3	1,653865	1,735671	1,666448	1,69742	1,753633
4	1,672955	1,750263	1,647775	1,706895	1,750263
5	1,692077	1,683526	1,713673	1,752641	1,683526
6	1,647323	1,652988	1,650191	1,700241	1,708741
7	1,743416	1,673543	1,731635	1,743416	1,692338
8	1,69493	1,667799	1,704631	1,700266	1,667799
9	1,667168	1,710032	1,689352	1,739977	1,724683
10	1,671859	1,675054	1,688295	1,725572	1,713163

Uji ke-	{0,6;0,4}	{0,7;0,3}	{0,8;0,2}	{0,9;0,1}
1	1,702458	1,711402	1,704011	1,74179
2	1,77301	1,70684	1,731521	1,67446
3	1,708975	1,763457	1,719729	1,748882
4	1,670602	1,74092	1,734422	1,681002
5	1,726551	1,752249	1,702142	1,672642
6	1,658573	1,751777	1,726979	1,723814
7	1,731635	1,732436	1,75515	1,704798
8	1,719593	1,666256	1,665822	1,689874
9	1,719593	1,757611	1,674093	1,71538
10	1,738206	1,706986	1,741803	1,701499



### LAMPIRAN J HASIL PENGUJIAN RASIO SKENARIO 3

Uji ke-	{0,1;0,9}	{0,2;0,8}	{0,3;0,7}	{0,4;0,6}	{0,5;0,5}
1	1,641711	1,66153	1,678167	1,715851	1,718474
2	1,636829	1,664061	1,682688	1,722659	1,694582
3	1,656703	1,704011	1,751067	1,696225	1,715357
4	1,644255	1,671953	1,642254	1,645728	1,7215
5	1,66076	1,649642	1,666601	1,717238	1,698939
6	1,645747	1,664327	1,678253	1,702455	1,662711
7	1,65092	1,677728	1,643178	1,736996	1,718746
8	1,665999	1,651853	1,647142	1,672604	1,713984
9	1,650479	1,636082	1,708071	1,681499	1,752943
10	1,649448	1,674027	1,729124	1,71921	1,722977

Uji ke-	{0,6;0,4}	{0,7;0,3}	{0,8;0,2}	{0,9;0,1}
1	1,683136	1,773117	1,757447	1,722975
2	1,700933	1,728287	1,661122	1,646883
3	1,721802	1,727137	1,706016	1,735045
4	1,735562	1,689083	1,708111	1,698705
5	1,733482	1,739492	1,721152	1,695211
6	1,704223	1,680344	1,743298	1,675181
7	1,732298	1,736391	1,68827	1,732474
8	1,674586	1,723694	1,71265	1,739906
9	1,752576	1,724529	1,70956	1,709998
10	1,741683	1,720392	1,777204	1,701571





## LAMPIRAN K HASIL PENGUJIAN FINAL

Uji ke-	Skenario 1	Skenario 2	Skenario 3
1	1.7453028986940922	1.6549662617790537	1.7597613207782954
2	1.694231794518425	1.7241261185816597	1.7241993937729347
3	1.720634826968058	1.7063375102337512	1.689414149228333
4	1.695801793323343	1.6783253608971886	1.7547178271240753
5	1.7189919917378416	1.7017167143388532	1.7357125340820678
6	1.7467346792172167	1.7517787396332671	1.701198581463821
7	1.694231794518425	1.7081289207552341	1.7393946439152326
8	1.6944683886534795	1.6966550434551944	1.712846882879792
9	1.7078399262817578	1.6859213837969398	1.7262072902846481
10	1.7069115633198404	1.713113303554377	1.7297818545320012



## LAMPIRAN KOORDINAT RELAY

Skenario 1		
Relay	X	Y
1	28.31	13.28
2	54.84	36.11
3	19.56	2.01
4	19.14	65.76
5	37.99	65.19
6	52.4	18.04
7	55.64	57.62
8	78.26	57.8
9	22.1	54.26
10	48.66	48.18
11	84.26	1.97
12	21.47	73.05
13	67.1	16.7
14	76.23	20.48
15	16.25	36.36
16	29.59	42.19
17	40.67	30.64
18	46.73	15.43
19	84.17	41.43
20	64.96	41.41
21	69.55	51.29
22	18.31	56.77
23	23.74	61.93
24	69.9	21.36
25	58.13	44.14
26	2.52	37.01
27	16.33	4.5



28	39.26	67.75
29	70.79	46.46
30	50.14	0.85
31	1.41	21.57
32	11.09	16.16
33	64.58	12.23
34	84.03	27.44
35	48.95	70.71
36	40.34	33.14
37	3.56	70.67
38	61.04	21.46
39	16.04	76.79
40	60.64	48.05
41	79.36	65.44
42	29.7	67.7
43	74.76	1.55
44	10.05	59.65
45	43.77	54.68

Skenario 2		
Relay	x	y
1	39.73,	4.09
2	43.43,	60.61
3	23.62,	23.44
4	20.8,	58.69
5	13.44,	7.03
6	5.36,	22.64
7	27.79,	8.21
8	25.86,	57.22
9	59.29,	25.55



10	4.24,	38.9
11	55.71,	1.84
12	25.92,	50.64
13	28.02,	64.89
14	0.24,	54.83
15	14.45,	43.63
16	52.53,	70.43
17	31.14,	38.92
18	45.46,	53.3
19	47.63,	69.74
20	44.19,	65.36
21	11.65,	59.05
22	43.9,	17.28
23	57.14,	59.1
24	10.89,	69.28
25	0.41,	13.27
26	53.68,	60.95
27	34.44,	58.3
28	50.38,	8.91
29	35.08,	19.85
30	38.4,	32.72
31	54.9,	23.17

Skenario 3		
Relay	X	Y
1	50.1,	20.78
2	77.36,	45.0
3	100.9,	36.3
4	69.08,	70.09
5	112.98,	63.88



6	102.26,	12.02
7	96.31,	10.37
8	108.92,	40.93
9	19.73,	0.82
10	26.86,	22.67
11	83.08,	70.24
12	111.7,	33.01
13	90.5,	37.25
14	75.26,	60.62
15	1.55,	51.21
16	99.73,	44.78
17	61.65,	61.84
18	22.83,	47.06
19	29.47,	71.12
20	89.32,	61.47
21	3.98,	50.78
22	78.1,	46.08
23	23.81,	8.03
24	8.92,	31.95
25	43.52,	58.92
26	22.31,	38.8
27	48.51,	16.68
28	84.77,	31.74
39	14.74,	32.23
31	114.44,	68.06
32	41.3,	8.35
33	60.59,	71.57
34	38.92,	46.4
35	116.11,	75.02
36	114.85,	50.41
37	78.2,	46.85



38	96.27,	71.1
39	63.29,	52.46
40	62.09,	51.68
41	8.71,	27.33
42	8.71,	27.33
43	0.42,	26.43
44	107.3,	28.95
45	57.45,	46.95
46	108.87,	23.25
47	45.62,	1.39
48	85.51,	75.94
49	68.99,	0.08
50	26.22,	44.15
51	56.67,	50.92
52	19.04,	40.45
53	47.67,	74.74
54	83.17,	67.6
55	59.36,	64.78
56	106.99,	63.3
57	43.2,	25.41
58	93.68,	43.8
59	0.26,	23.33