

**OPTIMASI PENEMPATAN AKSES POINT BERDASARKAN NILAI
MINIMUM PATH LOSS MENGGUNAKAN ALGORITMA
GENETIKA (STUDI KASUS: GEDUNG D DAN GAZEBO PTIIK
UNIVERSITAS BRAWIJAYA)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Gita Widigdo Wicaksono

NIM:115060807111021



PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2015

PERSETUJUAN

OPTIMASI PENEMPATAN AKSES POINT BERDASARKAN NILAI MINIMUM PATH LOSS (STUDI KASUS: GEDUNG D DAN GAZEBO PTIIK UNIVERSITAS BRAWIJAYA)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer



Disusun Oleh :

Gita Widigdo Wicaksono

NIM: 115060807111021

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Imam Cholissodin, S.Si, M.Kom

NIK: 85071916110268

M. Tanzil Furqon, S.Kom., MCompSc

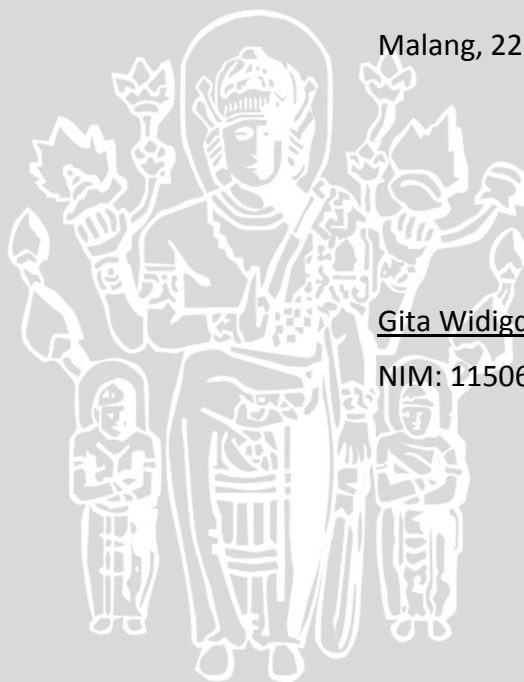
NIP: 198209302008011004

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 pustaka.

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, 22 Januari 2016



Gita Widigdo Wicaksono

NIM: 115060807111021

KATA PENGANTAR

Dengan nama Allah SWT Yang Maha Pengasih Lagi Maha Penyayang. Segala puji dan syukur penulis panjatkan atas kehadiran-Nya, karena hanya atas rahmat, pertolongan dan kasih sayang-Nya penulis dapat menyelesaikan skripsi dengan judul “Optimasi Penempatan Akses Point Berdasarkan Nilai Minimum Path Loss Menggunakan Algoritma Genetika (Studi Kasus: Gedung D dan PTIIK Universitas Brawijaya)” dengan baik. Shalawat serta salam atas junjungan nabi besar kita Nabi Muhammad SAW beserta keluarga dan para sahabat sekalian.

Melalui kesempatan ini, penulis menyampaikan rasa hormat dan terimakasih yang sebesar-besarnya kepada pihak-pihak yang telah memberikan bantuan kepada penulis dalam penyelesaian skripsi ini. Pihak-pihak tersebut antara lain :

1. Keluarga dari penulis terutama kedua orang tua penulis yang telah memberi motivasi, kasih sayang serta dukungan moril dan materil.
2. Bapak Imam Cholissodin, S.Si, M.Kom dan Bapak Muhammad Tanzil Furqon, S.Kom, M.CompSc selaku dosen pembimbing I dan dosen pembimbing II yang telah banyak memberikan bimbingan, ilmu dan saran dalam penyusunan skripsi ini.
3. Bapak Sabriansyah Rizqika Akbar, ST., M.Eng. selaku dosen pembimbing akademik yang telah memberikan bimbingan, ilmu dan saran kepada penulis.
4. Seluruh teman-teman angkatan 2011, terutama untuk teman-teman kuliah TIF-E 2011 terimakasih atas segala bantuannya selama menjadi mahasiswa serta dukungannya dalam menyelesaikan skripsi ini.
5. Teman-teman dekat penulis Yoke, Pandu, Sandi, Wiki, Riski, Elha, Angga, Julian, Ega, Johan, Riko, Erwin, Yoga, Ahsan, Valerian, Itto, Umar, Lutvi dan teman-teman yang lainnya terimakasih atas dukungannya mulai dari sebelum menjadi mahasiswa hingga memperoleh gelar Sarjana Komputer.

Semoga jasa dan amal baik kita semua mendapatkan balasan dari Allah SWT. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari sempurna. Oleh karena itu, kritik dan saran untuk kesempurnaan skripsi ini, senantiasa penulis harapkan dari berbagai pihak.

Malang, 22 Januari 2016

Penulis
wicaksono.gita@gmail.com

ABSTRAK

Gita Wdiddo Wicaksono. 2016. : Optimasi Penempatan Akses Point Berdasarkan Nilai Minimum Path Loss Menggunakan Algoritma Genetika (Studi Kasus: Gedung D dan Gazebo PTIIK Universitas Brawijaya). Skripsi Program Studi Informatika/Illmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya. Dosen Pembimbing : Imam Cholissodin, S.Si, M.Kom dan M. Tanzil Furqon, S.Kom.,McompSc.

Dalam merancang infrastruktur WLAN (*Wireless Local Area Network*) peran AP (*Access Point*) sangat dibutuhkan. AP digunakan untuk menghubungkan perangkat *wireless* dengan perangkat jaringan berkabel (*wired network*). AP menggunakan sinyal radio sebagai perantara komunikasi antar perangkat jaringan, sehingga AP membutuhkan lingkungan yang mendukung agar sinyal tidak terhalang sepenuhnya. Untuk menyesuaikan letak AP dengan tata letak bangunan sehingga mendapatkan kualitas sinyal baik terlebih dahulu dilakukan survei lapangan. Pada survei ini nantinya operator akan melakukan pengujian kualitas sinyal pada suatu titik ruangan. Apabila tata letak ruangan yang kompleks dan jumlah ruangan yang banyak, tentunya proses survei ini akan memakan banyak waktu dan juga biaya. Selain itu, redaman sinyal yang ditimbulkan oleh halangan yang ada pada bangunan patut dipertimbangkan. Untuk menghasilkan WLAN yang baik AP harus mampu mencakup *user* secara maksimal dengan redaman sinyal yang minim. Seiring berkembangnya teknologi, proses optimasi berdasarkan kecerdasan buatan telah banyak dikembangkan. Salah satu metode optimasi yang sering digunakan adalah algoritma genetika. Dengan metode tersebut optimasi akan dapat diterapkan terkait permasalahan penempatan AP. Penelitian ini bertujuan untuk membangun sistem optimasi penempatan lokasi AP berdasarkan nilai minimum *path loss* menggunakan algoritma genetika. Lokasi yang digunakan dalam penelitian adalah ruang robotik, ruang HMIF, dan juga gazebo PTIIK UB. Hasil dari pengujian didapatkan ukuran populasi terbaik bermula dari penggunaan jumlah populasi sebesar 20, banyak generasi 80, kombinasi *cr* dan *mr* 0.9 0.1, dan 2 AP. Dari semua hasil pengujian didapatkan hasil rata-rata nilai *fitness* tertinggi sejumlah 0,5208.

Kata Kunci : Algoritma Genetika, *Access Point*, Sistem Optimasi, Rata-rata Nilai *Fitness*.

ABSTRACT

Gita Wdidgdo Wicaksono. 2016. : Access Point Placement Optimization Based on Path Loss Minimum Value Using Genetic Algorithm (Case Study : Building D and Gazebo PTIIK Brawijaya University). Skripsi Program Studi Informatika/Illmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya. Dosen Pembimbing : Imam Cholissodin, S.Si, M.Kom dan M. Tanzil Furqon, S.Kom., McompSc.

In designing the infrastructure of WLAN (Wireless Local Area Network) the role of AP (Access Point) is needed. AP is used for connecting wireless devices to a wired network devices. AP using radio signals as a medium of communication between network devices , so the AP requires a supportive environment for the signal to not be blocked entirely. To adjust the location of the AP with the layout of the building in order to get a better signal quality then the field survey must be done first. At this survey operator will be testing the quality of the signal at any point in the room. If the room layout is too complex and the amount of room is too much , of course, the survey process will take a lot of time and costs. In addition, signal attenuation caused by the obstacles that exist on the building should be considered. To produce a good WLAN AP must be able to cover the maximum user with minimal signal attenuation. As the development of technology , process optimization based on artificial intelligence have been developed. One method of optimization that is often used is a genetic algorithm. With the optimization method, AP placement issues will be resolved. This study aimed to establish the location of the AP placement optimization system based on the minimum value of path loss using genetic algorithms. The location was used in this study is the robotics room, room of HMIF , and also gazebo PTIIK UB. The results obtained from testing the best population size start from the use of a population of 20 , the best generations size is at 80 , the combination cr mr is 0.9 and 0.1 , and then we used 2 AP. Of all the test results showed the average value of fitness highest number is 0.5208.

Keywords : Genetic Algorithm, Access Point, Optimization System, Average Fitness Value.

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian pustaka	5
2.2 WLAN (<i>Wireless Local Area Network</i>).....	11
2.2.1 Teknologi WLAN.....	12
2.2.2 Jaringan <i>Wireless</i>	12
2.3 Pengertian penyelesaian terbaik dan optimasi	16
2.4 Algoritma genetika.....	16
2.4.1 Dasar-dasar algoritma genetika	17
2.4.2 Proses algoritma genetika.....	18
2.4.2.1 Inisialisasi kromosom.....	18
2.4.2.2 <i>Generate</i> populasi awal.....	19
2.4.2.3 Representasi kromosom.....	19
2.4.2.4 Perhitungan nilai <i>fitness</i>	20
2.4.2.5 Proses <i>crossover</i>	20

2.4.2.6 Mutas.....	21
2.4.2.7 Seleksi.....	21
2.4.2 Proses pemetaan alami ke proses komputer.....	22
BAB 3 METODOLOGI	24
3.1 Studi literatur	24
3.2 Identifikasi masalah	25
3.3 Pegumpulan dan analisa data.....	25
3.4 Perancangan sistem.....	25
3.5 Implementasi sistem.....	26
3.6 Pengujian	26
3.7 Pengambilan kesimpulan.....	26
BAB 4 PERANCANGAN.....	27
4.1 Perancangan sistem.....	27
4.1.1 <i>Generate</i> populasi awal.....	28
4.1.2 <i>Crossover</i>	29
4.1.3 Mutasi.....	30
4.1.4 Menghitung nilai <i>fitness</i>	31
4.1.5 Seleksi	32
4.2 Perhitungan Manual	33
4.2.1 Inisialisasi parameter	35
4.2.2 <i>Generate</i> populasi awal	35
4.2.3 Menghitung nilai <i>fitness</i>	37
4.2.4 Seleksi.....	40
4.2.5 <i>Crossover</i>	40
4.2.6 Mutasi	41
4.2.7 <i>Random injection</i>	41
4.2.8 Status koneksi	41
4.3 Skenario Pengujian	45
4.3.1 Uji coba banyak generasi	45
4.3.2 Uji coba jumlah populasi.....	45
4.3.3 Uji coba kombinasi prob. <i>crossover</i> dan mutasi	46
4.3.4 Uji coba jumlah AP	46

4.4 Perancangan <i>user interface</i>	47
BAB 5 IMPLEMENTASI	51
5.1 Spesifikasi sistem	51
5.1.1 Spesifikasi perangkat keras	51
5.1.2 Spesifikasi perangkat lunak	51
5.2 Implementasi algoritma pada program	52
5.2.1 Implementasi proses <i>input</i> data	52
5.2.2 Implementasi perhitungan algoritma genetika	54
5.2.2.1 Implementasi representasi kromosom	54
5.2.2.2 Implementasi proses <i>Crossover</i>	56
5.2.2.3 Implementasi proses mutasi	59
5.3 Implementasi antarmuka	63
5.3.1 Implementasi halaman <i>home</i>	63
5.3.2 Implementasi halaman <i>input</i>	63
5.3.3 Implementasi halaman hasil	64
BAB 6 PENGUJIAN DAN ANALISIS	66
6.1 Hasil dan analisis skenario pengujian banyak generasi	66
6.2 Hasil dan analisis skenario pengujian jumlah populasi	67
6.3 Hasil dan analisis skenario pengujian kombinasi <i>Cr</i> dan <i>Mr</i>	69
6.4 Hasil dan analisis skenario pengujian jumlah AP	70
BAB 7 Penutup	72
7.1 Kesimpulan	72
7.2 Saran	72
DAFTAR PUSTAKA	74

DAFTAR TABEL

Tabel 2.1 Tabel kajian pustaka	5
Tabel 2.2 Contoh representasi kromosom	19
Tabel 2.3 Contoh penyilangan	21
Tabel 2.5 Pemetaan Proses Alamiah ke Proses Komputasi	23
Tabel 4.1 <i>Generate</i> Populasi Awal	36
Tabel 4.2 Detail kromosom	36
Tabel 4.3 Hasil Perhitungan $pL(d0)$	37
Tabel 4.4 Koordinat <i>User</i>	38
Tabel 4.5 Hasil Perhitungan Nilai Fitness Tiap Kromosom	39
Tabel 4.6 Kromosom Dengan Nilai <i>Fitness</i> Terbaik	40
Tabel 4.7 Proses <i>Crossover</i>	40
Tabel 4.8 Hasil <i>Crossover</i>	40
Tabel 4.9 Proses Mutasi	41
Tabel 4.10 Hasil <i>Random Injection</i>	41
Tabel 4.11 Ilustrasi Status Koneksi <i>User</i>	42
Tabel 4.12 Ilustrasi Status Koneksi Dengan 3 <i>User</i>	43
Tabel 4.13 Rancangan Uji Coba Banyaknya Generasi	45
Tabel 4.14 Rancangan Uji Coba Ukuran Populasi	46
Tabel 4.15 Rancangan Uji Coba <i>Crossover Rate</i> dan <i>Mutation Rate</i>	46
Tabel 4.16 Rancangan Uji Coba Jumlah <i>Access Point</i> (AP)	47
Tabel 5.1 Spesifikasi perangkat keras	51
Tabel 5.2 Spesifikasi perangkat lunak	51
Tabel 6.1 Hasil skenario pengujian banyak generasi penentuan lokasi AP	66
Tabel 6.2 Hasil skenario pengujian ukuran populasi penentuan lokasi AP	68
Tabel 6.3 Hasil skenario pengujian kombinasi <i>Cr</i> dan <i>Mr</i>	69
Tabel 6.4 Hasil skenario pengujian jumlah AP	70

DAFTAR GAMBAR

Gambar 2.1 Infrastruktur WLAN	11
Gambar 2.2 Contoh Panjang Gelombang	13
Gambar 2.3 Ilustrasi LOS	14
Gambar 2.5 Arsitektur Algoritma Genetika.	18
Gambar 3.1 Metodologi penelitian.....	24
Gambar 4.1 Diagram Alir Algoritma Genetika Pada Sistem.....	27
Gambar 4.2. Diagram Alir Membuat Populasi Awal	28
Gambar 4.3 Diagram Alir Proses <i>Crossover</i>	30
Gambar 4.4 Diagram Alir Proses Mutasi.....	31
Gambar 4.5 Diagram Alir Perhitungan Nilai <i>Fitness</i>	32
Gambar 4.6 Diagram Alir Proses Seleksi	33
Gambar 4.7 Ilustrasi Letak <i>User</i>	34
Gambar 4.8 Ilustrasi <i>Encoding</i> Gen Lokasi Pada Kromosom 1 AP ke-1	37
Gambar 4.9 Perancangan antarmuka halaman <i>home</i>	47
Gambar 4.10 Perancangan halaman <i>input</i>	48
Gambar 4.11 Perancangan halaman detail perhitungan	49
Gambar 4.12 Perancangan halaman hasil perhitungan.....	49
Gambar 5.1 Implementasi proses <i>input</i> data	53
Gambar 5.2 Implementasi representasi kromosom	55
Gambar 5.3 Implementasi proses <i>crossover</i>	59
Gambar 5.4 Implementasi proses mutasi	62
Gambar 5.5 Antarmuka halaman home	63
Gambar 5.6 Implementai halaman <i>input</i>	63
Gambar 5.7 Implementasi halaman hasil detail perhitungan	64
Gambar 5.8 Implementasi hasil perhitungan	64

DAFTAR LAMPIRAN

A.1 Foto ruangan dan letak user	75
A.2 Form Kuesioner	77



BAB 1 PENDAHULUAN

1.1 Latar belakang

Access Point (AP) merupakan salah satu perangkat yang digunakan dalam membangun infrastruktur *Wireless Local Area Network* (WLAN). WLAN adalah jaringan komunikasi nirkabel (tanpa kabel) yang meliputi wilayah kecil seperti, kampus, gedung perkantoran, rumah, dan juga sekolah. AP dalam infrastruktur WLAN berperan sebagai perangkat yang menerima dan mengirim sinyal. AP menggunakan frekuensi radio untuk dapat berkomunikasi dengan perangkat *wireless* lainnya, sehingga dapat menghubungkan perangkat tersebut dengan infrastruktur jaringan berkabel yang menyediakan fasilitas internet.

Sinyal *wireless* adalah sinyal yang mudah sekali terhalang oleh suatu benda padat baik itu berupa dinding ataupun benda-benda padat yang lain. *Path loss* merupakan suatu keadaan dimana sinyal *wireless* akan semakin melemah seiring bertambahnya jarak yang ditempuh dalam suatu ruang. Oleh karena itu penempatan AP dalam membangun infrastruktur WLAN merupakan faktor yang penting. Dengan penempatan AP yang terencana dengan baik akan meningkatkan cakupan transmisi dan juga menghasilkan kinerja WLAN yang optimal (Ungurawanich, 2005).

Pemasangan AP pada tempat dengan variasi tata letak dan halangan yang kompleks memerlukan adanya survei guna menentukan lokasi dari AP. Hal ini tentunya akan memakan waktu dan juga biaya.

Selain itu akan sulit menentukan lokasi AP yang dapat menghasilkan kualitas sinyal yang baik. Kualitas sinyal yang baik dapat dilihat dari nilai SNR. SNR adalah *Signal-to-noise Ratio*, yang merupakan perbandingan antara sinyal dan gangguan yang ada. Semakin tinggi nilai dari SNR maka kualitas sinyal semakin baik. SNR yang kecil berarti terdapat banyak gangguan yang mampu menurunkan kekuatan sinyal (Maskuriwong, Vara, dan Nacho, 2003).

PTIIK Universitas Brawijaya telah menerapkan WLAN dalam infrastruktur jaringan komputer yang digunakan di dalamnya. Penerapan WLAN ini digunakan sebagai sistem komunikasi data dan juga sebagai penghubung *user* ke jaringan LAN PTIIK UB. Gedung D dan gazebo PTIIK UB merupakan tempat dengan *traffic network* yang tinggi. *User* pada gedung D dan gazebo PTIIK UB mencapai 400 sampai 800 *user* pada hari perkuliahan, sedangkan AP pada area tersebut hanya berjumlah 8 AP saja. 1 AP mampu melayani maksimal sebanyak 20 *user*. Selain itu *bandwidth* yang disediakan pada masing-masing AP hanya sebesar 1MBps. Hal ini menyebabkan AP *overload* dan menghasilkan kualitas jaringan internet yang buruk. Pada gedung D dan gazebo PTIIK UB juga belum dilakukan optimasi dalam penempatan AP berdasarkan *path loss*, sehingga terdapat kemungkinan adanya lokasi AP yang masih kurang optimal dan menyebabkan penurunan performa jaringan. Berdasarkan uraian di atas, maka dibutuhkan suatu sistem

berbasis komputasi cerdas untuk optimasi penentuan lokasi AP berdasarkan nilai minimum *path loss*.

Optimasi merupakan suatu proses menyelesaikan masalah supaya berada pada kondisi yang paling menguntungkan dari sudut pandang tertentu (Zuchri, 2013). Optimasi dapat dilakukan dengan berbagai metode, diantaranya adalah Brute force, Greddy, Dynamic Programming dan Monte Carlo. Beberapa metode optimasi juga mengadopsi proses alamiah atau disebut dengan Evolutionary Computation, salah satu metode tersebut adalah Algoritma Genetika. Algoritma genetika merupakan metode optimasi berbasis populasi yang didasarkan pada teori Charles Darwin (Barbosa, Marco, dan Meirelles, 2012).

Pada penelitian sebelumnya telah dilakukan proses optimasi penempatan AP yang membandingkan metode *Greedy Search Heuristic* (GSH) dan juga Algoritma Genetika (Barbosa, Marco, dan Meirelles, 2012). Penelitian tersebut dilakukan pada 3 bangunan. Bangunan yang pertama memiliki luas 60m x 11m, bangunan kedua 60m x 18m, dan bangunan ketiga memiliki luas 400m x 400m. Kedua metode menghasilkan area cakupan AP sebesar 100%. Hal ini berarti setiap area bangunan telah ter-cover oleh sinyal AP. Penelitian pada bangunan pertama dengan metode GSH menghasilkan nilai *fitness* sebesar 0.083333, sedangkan pada metode Algoritma Genetika menghasilkan nilai *fitness* sebesar 0.090909. Jumlah AP yang diperhitungkan pada bangunan pertama menggunakan metode GSH sebanyak 12 AP, sedangkan dengan metode Algoritma Genetika sebanyak 11 AP. Pada bangunan kedua nilai *fitness* dengan metode GSH sebesar 0.042245, dan metode Algoritma Genetika sebesar 0.057155. Jumlah AP yang digunakan pada metode GSH sebanyak 23 AP, sedangkan pada metode Algoritma Genetika sebanyak 17 AP. Pada bangunan ketiga nilai *fitness* dengan metode GSH sebesar 0.012282, sedangkan dengan metode Algoritma Genetika menghasilkan nilai *fitness* sebesar 0.018367. Jumlah AP yang digunakan metode GSH sebanyak 51 AP, dan jumlah AP yang digunakan dalam Algoritma Genetika sebesar 50 AP. Penelitian tersebut membuktikan bahwa Algoritma Genetika lebih unggul dari metode GSH. Selain menghasilkan nilai *fitness* yang lebih tinggi, juga menghasilkan jumlah penggunaan AP yang lebih sedikit.

Penelitian selanjutnya adalah tentang optimasi penempatan AP berdasarkan nilai *path loss*. Pada penelitian ini membandingkan Algoritma Genetika dengan metode *Brute-Force*. Penelitian dilakukan pada area dengan luas 10 m x 10 m. Hasil penelitian ini adalah Algoritma Genetika memiliki nilai rata-rata *path loss* yang sedikit lebih baik dari metode *Brute-Force* yaitu sebesar 89.74474, sedangkan metode *Brute-Force* sebesar 89.53809. Hal ini membuktikan bahwa metode Algoritma Genetika merupakan metode yang handal dalam melakukan proses optimasi. Algoritma genetika merupakan algoritma pencarian yang efisien. Untuk ruang pencarian yang besar, algoritma genetika hanya membutuhkan waktu yang sedikit untuk melakukan komputasi dengan solusi yang menjanjikan dibandingkan dengan metode pencarian yang lain seperti integer programming, dynamic programming, branch and bound dan lain sebagainya (Ungurawanich, 2005).

1.2 Rumusan masalah

Dari pemikiran pada bagian latar belakang maka rumusan permasalahan yang akan diselesaikan dalam penulisan ini yaitu:

1. Bagaimana mengimplementasikan algoritma genetika untuk optimasi penempatan AP berdasarkan nilai minimal path loss?
2. Bagaimana nilai evaluasi dari hasil implementasi algoritma genetika pada permasalahan optimasi penentuan lokasi AP berdasarkan nilai minimum path loss?

1.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Mengetahui implementasi algoritma genetika pada penempatan AP berdasarkan nilai minimum path loss
2. Mengetahui nilai evaluasi dari penerapan algoritma genetika pada optimasi penempatan lokasi AP berdasarkan nilai minimum path loss.

1.4 Manfaat

Manfaat dari penelitian ini adalah :

1. Mengoptimalkan infrastruktur WLAN pada gedung D dan gazebo PTIIK UB berdasarkan nilai minimum path loss.
2. Mendapatkan tata letak AP optimal yang menghasilkan kualitas sinyal yang baik.
3. Membantu dalam pembuatan infrastruktur WLAN baru yang mempertimbangkan path loss dalam penempatan AP.
4. Meminimalkan waktu dan biaya dalam penentuan lokasi AP saat membangun infrastruktur WLAN yang baru.

1.5 Batasan masalah

Dari permasalahan yang ada maka diperlukan adanya sebuah batasan untuk memudahkan dalam melakukan penyelesaian masalah. Batasan-batasan yang digunakan yaitu:

1. Data penelitian yang digunakan adalah data tentang lokasi, frekuensi, dan juga transmit power dari AP pada gedung D dan gazebo PTIIK Universitas Brawijaya.
2. Perhitungan lokasi optimal AP didasarkan pada nilai minimum *path loss*.
3. Metode pemetaan yang digunakan menggunakan metode *Grid Point* dua dimensi.

1.6 Sistematika pembahasan

Sistematika Penelitian ini menggunakan kerangka penulisan sebagai berikut:

BAB I : PENDAHULUAN

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan jadwal penelitian.

BAB II : LANDASAN KEPUSTAKAAN

Menguraikan kajian pustaka dan dasar teori yang berkaitan dengan metode algoritma genetika dalam optimasi penempatan lokasi AP.

BAB III : METODOLOGI PENELITIAN

Membahas metodologi yang akan digunakan dalam penelitian yang terdiri dari studi literatur, analisis kebutuhan, pengumpulan data, perancangan sistem, implementasi sistem, pengujian sistem, dan pengambilan kesimpulan.

BAB IV : PERANCANGAN

Membahas tentang perancangan sistem penentuan lokasi AP berdasarkan nilai minimum path loss dengan metode algoritma genetika.

BAB V : IMPLEMENTASI

Membahas tentang implementasi sistem penentuan lokasi AP berdasarkan nilai minimum path loss dengan metode algoritma genetika.

BAB VI : PENGUJIAN DAN ANALISIS SISTEM

Membahas tingkat akurasi sistem dan analisa hasil terhadap sistem penentuan lokasi AP berdasarkan nilai minimum path loss dengan metode algoritma genetika.

BAB VII : PENUTUP

Berisi kesimpulan berdasarkan hasil penelitian yang telah dibangun serta memberikan saran yang berguna bagi pengembangan sistem.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini akan menjelaskan mengenai kajian pustaka dan dasar teori yang menunjang skripsi tentang Optimasi Penempatan Akses Point Berdasarkan Nilai Minimum Path Loss dengan Menggunakan Algoritma Genetika. Pada Kajian Pustaka akan membahas mengenai penelitian yang telah dilakukan sebelumnya dan menjadikannya sebagai perbandingan guna menunjang penulisan skripsi ini. Dasar teori akan membahas mengenai metode Algoritma Genetika, infrastruktur WLAN, serta faktor-faktor yang mempengaruhi kekuatan sinyal dari AP.

2.1 Kajian pustaka

Kajian pustaka merupakan pembahasan mengenai penelitian yang telah dilakukan sebelumnya. Penelitian pada skripsi ini mengenai optimasi penempatan AP berdasarkan nilai minimum *path loss* menggunakan Algoritma Genetika. Penelitian-penelitian sebelumnya digunakan dalam memberikan acuan, baik dari segi efektifitas metode atau sebagai acuan penulisan skripsi ini.

Terdapat empat penelitian yang akan di kaji dalam bab ini. Penelitian tersebut mengenai optimasi infrastruktur WLAN yang berfokus pada penempatan AP, sehingga menghasilkan infrastruktur WLAN yang sesuai dengan kebutuhan *user* dan juga optimal. Beberapa penelitian tersebut menggunakan Algoritma Genetika yang merupakan suatu metode berbasis Algoritma Evolusi yang mengadopsi proses ilmiah dan teori evolusi Charles Darwin (Barbosa, Marco, dan Meirelles, 2012). Dari salah satu penelitian tersebut menggunakan metode empiris dan teoritis, dimana kedua metode tersebut akan dibandingkan terkait hasil optimasi pada penelitian tersebut.

Berikut ini merupakan kajian pustaka mengenai penelitian sebelumnya yang berkaitan tentang optimasi penempatan AP pada infrastruktur WLAN dengan menggunakan metode Algoritma Genetika dan juga metode beberapa metode lainnya seperti yang ditunjukkan pada Tabel 2.1. Keempat penelitian tersebut menjadikan dasar penulis untuk mengambil judul Optimasi Penempatan Akses Point Berdasarkan Nilai Minimum Path Loss dengan Metode Algoritma Genetika.

Tabel 2.1 Tabel kajian pustaka

No.	Judul	Objek	Metode	Hasil
1	Wireless LAN Design Considering Average Path Loss Minimization (Ungurawa	Melakukan optimasi penempatan AP dengan mempertimbangan nilai rata-rata <i>path loss</i> menggunakan Algoritma	Melakukan pengkodean terhadap solusi permasalahan. Solusi tersusun dari gen berupa lokasi AP, <i>frequency channel</i>, dan <i>power level AP</i>.	Algoritma Genetika terbukti mampu menyelesaikan optimasi penempatan AP berdasarkan nilai minimum <i>path loss</i>

	nich, 2005).	Genetika. Input: <i>Input</i> pada penelitian tersebut adalah jumlah AP yang digunakan, batas populasi, probabilitas <i>crossover</i>, dan juga probabilitas mutasi.	• Pembangkitan populasi awal yang tersusun dari solusi yang telah dikodekan. Pembangkitan populasi awal dilakukan secara acak. • Melakukan proses penyilangan (<i>crossover</i>). Proses penyilangan dilakukan oleh dua kromosom yang dipilih secara acak. • Perhitungan nilai <i>fitness</i> pada tiap individu dalam populasi. Nilai <i>fitness</i> ini ditentukan dari nilai minimal <i>path loss</i> pada tiap kromosom. • Melakukan proses mutasi. Mutasi dilakukan berdasarkan probabilitas mutasi yang telah ditentukan. • Mengulang langkah 3 sampai 5 hingga didapati nilai <i>fitness</i> terbaik.	dengan baik. Hal ini dibuktikan dengan perolehan nilai rata-rata <i>path loss</i> dan juga penggunaan AP yang lebih sedikit dari metode Brute Force.
2	Access point design with Genetic Algorithm (Barbosa, Marco, dan Meirelles, 2012).	Objek: Optimasi terhadap penempatan AP sehingga memiliki cakupan dan jumlah <i>user</i> yang maksimal Input: dimensi	Langkah dan metode: • Menentukan lokasi AP menggunakan metode <i>Greedy Search Heuristic</i> (GSH). • <i>Generate</i> kromosom berdasarkan lokasi AP. • Menyusun populasi	Hasil: Pada penelitian ini membandingkan optimasi AP dengan objek cakupan dan jumlah <i>user</i> mencapai maksimal.

		ruangan yang akan digunakan optimasi penempatan AP, jumlah <i>user</i> pada ruangan tersebut, probabilitas <i>crossover</i>, dan probabilitas mutasi.	secara acak dari kromosom – kromosom yang telah terbentuk. • Melakukan evaluasi dengan fungsi objektif berupa perhitungan jumlah AP berdasarkan jumlah <i>user</i> dan juga luas area. • Melakukan proses seleksi dengan metode <i>tournament</i>. • Melakukan penyilangan terhadap kromosom berdasarkan nilai probabilitas yang telah ditentukan. • Melakukan mutasi berdasarkan nilai probabilitas yang telah ditentukan. • Ulangi langkah ke 4 sampai 7 hingga sesuai dengan syarat berhenti.	Optimasi dilakukan dengan dua metode yaitu, GSH dan Algoritma Genetika. Hasilnya Algoritma Genetika memiliki rata-rata cakupan dan jumlah <i>user</i> lebih baik dari pada metode GSH.
3	Optimasi Penataan Sistem Wi-Fi di PENS-ITS dengan menggunakan Algoritma Genetika (Kartika, 2010).	Objek: Menentukan letak AP sehingga menghasilkan area cakupan yang maksimal berdasarkan level daya. Input: Dimensi target area, probabilitas <i>crossover</i>, probabilitas mutasi, ukuran populasi, jumlah	Langkah dan Metode: • Menentukan level daya dari AP yang telah terpasang pada <i>target area</i>. • Generate kromosom berdasarkan lokasi AP. • Melakukan evaluasi dengan fungsi objektif, yang merupakan fungsi perhitungan <i>coverage area</i>. Fungsi tersebut	Hasil: Optimasi penempatan AP dengan metode Algoritma Genetika pada penelitian ini menghasilkan penempatan AP dengan area cakupan sebesar 91,67% dari <i>service area</i>.

		generasi, dan level daya dari setiap AP.	memiliki batasan berupa fungsi jarak yang dihitung berdasarkan level daya pemancar. • Mengulang langkah dua sampai tiga hingga seluruh area ter-cover atau hingga nilai coverage area tidak berubah.	
4	Optimasi Area Jaringan Nirkabel Dalam Ruangan (Studi Kasus: PTIIK Universitas Brawijaya) (Widyaningsih, 2012).	Objek: Menentukan letak AP yang memaksimalkan cakupan sinyal, menentukan <i>channel</i> AP sehingga tidak <i>overlapping</i>, dan juga kinerja jaringan nirkabel. Input: Kriteria pada model teoritis aspek propagasi, yaitu - Coverage yang dapat dilayani - Jumlah pemancar (AP) - Perhitungan <i>Free Space Loss</i> - Received <i>Signal Strength</i> (RSSI)	Langkah dan Metode: • Melakukan perhitungan MAPL (<i>Maximum Allowed Path Loss</i>) sehingga mendapatkan jarak maksimal agar <i>user</i> dan AP dapat terhubung. • Menentukan kekuatan sinyal berdasarkan kekuatan pemancar dan kapasitas <i>user</i>. • Melakukan perhitungan <i>user</i> aktif maksimal yang dapat dilayani oleh 1 AP • Menerapkan perhitungan satu sampai tiga pada area layanan yang telah ditentukan	Hasil: Hasil dari penelitian ini dibagi menjadi dua kelompok. Kelompok pertama merupakan perhitungan metode teoritis berdasarkan aspek propagasi, sedangkan kelompok kedua merupakan perhitungan metode teoritis berdasarkan aspek kapasitas <i>user</i>. Perhitungan dengan metode teoritis dan empiris berdasarkan aspek propagasi menghasilkan output berupa perbaikan dengan topologi baru yang

				disertai <i>display contour</i> jaringan yang dapat mencakup seluruh area yang direncanakan.
--	--	--	--	--

Sumber : (Ungurawanich, 2005) , (Barbosa, Marco, dan Meirelles, 2012) , (Kartika, 2010) , (Widyaningsih, 2012).

Narongchai Ungurawanich dan Naruemon Wattanapongsakorn dari fakultas teknik jurusan teknik komputer King Mongkut's University of Technology Thonburi Thailand telah melakukan penelitian yang berjudul Wireless LAN System Design Considering Average Path Loss Minimization. Penelitian ini membahas tentang penerapan optimasi dalam membangun jaringan WLAN berdasarkan nilai terkecil *path loss* dengan menggunakan metode Algoritma Genetika. Kromosom yang digunakan dalam metode ini terdiri dari tiga gen yang berasal dari Akses Point (AP), yaitu lokasi AP, frekuensi AP, dan juga *power level* dari AP. Penelitian ini menggunakan metode *Grid Segment* untuk memetakan lokasi AP pada bangunan yang dijadikan sebagai studi kasus.

Penelitian ini dilakukan pada area dengan luas 21 m x 35, dengan 33 *user* dan *bandwidth* sebesar 25.75 Mbps. Pada area tersebut telah dilakukan perencanaan infrastruktur WLAN dengan menggunakan tiga AP dan menghasilkan rata-rata nilai *path loss* sebesar 95.85435. Kemudian Algoritma Genetika diterapkan pada area yang sama dengan jumlah AP yang sama, hasilnya rata-rata nilai *path loss* yang diperoleh lebih kecil dari perencanaan infrastruktur sebelumnya yakni sebesar 95.39104. Penerapan Algoritma Genetika juga dilakukan dengan menggunakan 4 AP pada area yang sama dan menghasilkan solusi yang lebih baik dari sebelumnya, dengan rata-rata nilai *path loss* sebesar 91.15715. Dari hasil tersebut dapat disimpulkan bahwa dengan metode Algoritma Genetika diperoleh infrastruktur WLAN dengan rata-rata *path loss* yang lebih rendah. Selain itu infrastruktur yang dibangun dengan metode ini juga memenuhi kriteria penting seperti meminimalkan gangguan sinyal , kapasitas *bandwidth* yang memadai, dan juga nilai SNR yang kecil. Nilai SNR yang tinggi menandakan kualitas sinyal yang dihasilkan minim gangguan.

Pada penelitian kedua yang dilakukan oleh Marco Antonio S. Barbosa dan Maury Meirelles Gouvea Jr. menerapkan Algoritma Genetika dalam proses optimasi penempatan AP yang memaksimalkan cakupan area dan jumlah *user* dengan menggunakan jumlah AP seminimal mungkin. Penelitian ini dilakukan pada tiga bangunan yang berbeda. Bangunan pertama dan kedua merupakan bangunan nyata dengan dimensi 60m x 11m, dan 60m x 18m. Bangunan ketiga merupakan bangunan *hypothetical* (anggapan sementara) dengan dimensi yang

lebih besar yaitu 400m x 400m. Hasil optimasi menggunakan algoritma genetika akan dibandingkan dengan metode GSH.

Penelitian pada bangunan pertama dengan metode GSH menghasilkan fungsi objektif sebesar 0.083333 dengan rata-rata cakupan AP sebesar 100%, jumlah AP sebanyak 12, dan jumlah *user* yang tidak terhubung sebesar 0. Sedangkan dengan menggunakan Algoritma Genetika menghasilkan fungsi objektif sebesar 0.090909 dimana memiliki area cakupan AP sebesar 100%, jumlah AP yang digunakan sebanyak 11, dan jumlah *user* yang tidak bisa terhubung sebanyak 0. Penelitian pada bangunan kedua dengan menggunakan metode GSH menghasilkan fungsi objektif sebesar 0.042245 dengan cakupan area AP sebesar 100%, jumlah AP yang digunakan sebanyak 23, dan jumlah *user* yang tidak dapat terhubung sebanyak 0. Optimasi yang dilakukan dengan Algoritma Genetika menghasilkan fungsi objektif sebesar 0.057155 dengan area cakupan AP 100%, jumlah AP yang digunakan sebanyak 17 dan jumlah *user* yang tidak bisa terhubung sebanyak 0. Pada bangunan ketiga optimasi yang dilakukan menggunakan metode GSH menghasilkan fungsi objektif sebesar 0.012282 dengan arean cakupan AP 100%, jumlah AP yang digunakan sebanyak 51 AP, dan jumlah *user* yang tidak dapat terhubung sebanyak 0. Sedangkan jika menggunakan Algoritma Genetika menghasilkan fungsi objektif sebesar 0.018367 dengan area cakupan AP 100%, jumlah AP yang digunakan sebanyak 50 AP, dan jumlah *user* yang tidak dapat terhubung sebanyak 0.

Penelitian selanjutnya dari Kurnia P. Kartika, Tri Budi Santoso, dan Nur Adi Siswandari yang melakukan penelitian tentang optimasi penataan sistem Wi-Fi dengan metode Algoritma Genetika. Pada penelitian ini memiliki fungsi objektif berupa area cakupan Wi-Fi yang dihitung berdasarkan nilai *level* daya dari AP. *Level* daya ini didapatkan dari database yang berisi *level* daya dari AP yang telah terpasang sebelumnya.

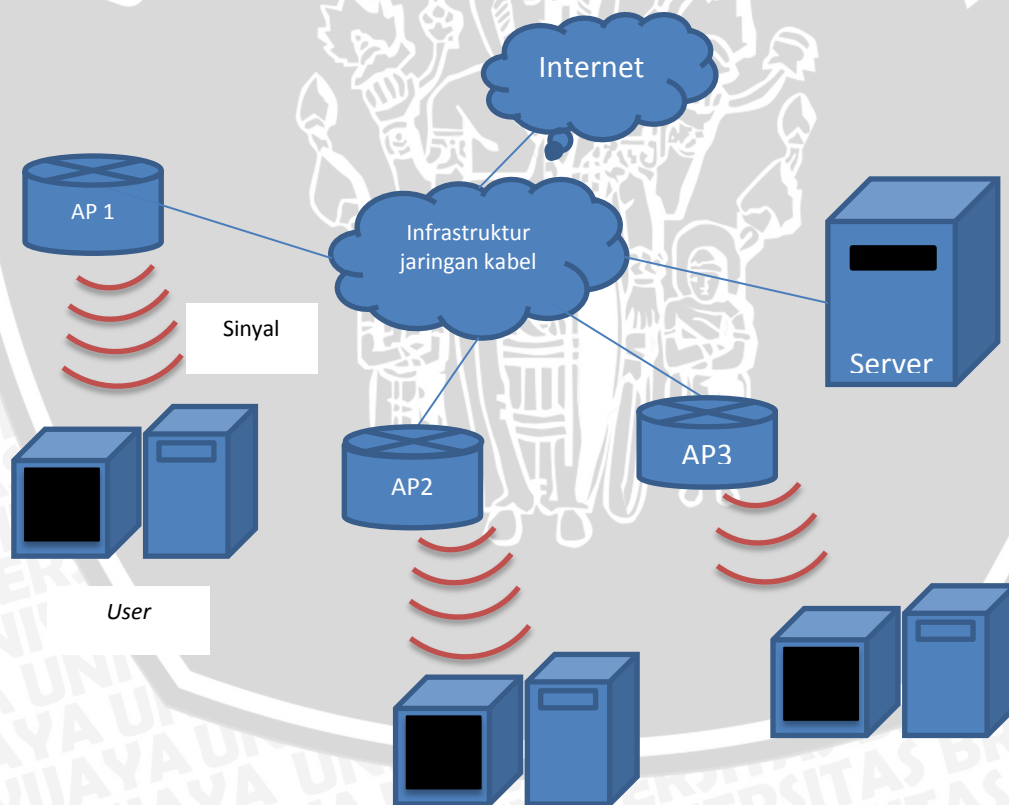
Pada penelitian ini dilakukan dengan dua perbandingan dimana proses Algoritma Genetika menggunakan lokasi AP yang telah ditentukan dan menggunakan lokasi AP secara *random*. Pemodelan dengan penentuan lokasi awal AP secara *random* dilakukan sebanyak 4 kali dengan kenaikan iterasi yang bervariasi. Hasil yang didapatkan dari pemodelan lokasi AP yang ditentukan dan pemodelan dengan lokasi AP *random* adalah sama, yaitu menghasilkan cakupan area AP sebesar 91.67%.

Penelitian keempat adalah penelitian dari Bakti Widyaningsih. Pada penelitian ini membahas tentang optimasi jaringan nirkabel dengan metode empiris dan teoritis. Metode empiris merupakan metode dengan mengamati secara langsung untuk memperoleh data lapangan yang sebenarnya. Metode teoritis dalam penelitian ini adalah pengukuran aspek propagasi jaringan nirkabel yang meliputi jumlah pemancar (AP), *free space loss*, RSSI (*Received Signal Strength*), cakupan area yang dapat dilayani, serta pengukuran terhadap penghalang (tembok, pintu, lantai). Pada penelitian ini terdapat dua macam aspek yang perlu diperhitungkan yaitu aspek kriteria model teoritis dan juga aspek kapasitas *user*.

Penelitian dilakukan pada lima gedung yang berbeda. Pada bangunan pertama menghasilkan estimasi AP sebanyak 2 buah dengan menerapkan *channel* 1 dan 6, *power* 18dbm serta RSSI yang dihasilkan sebesar 50dbm. Pada bangunan kedua, estimasi AP sebanyak 5 buah kemudian menggunakan *channel* 1, 10, 6, 2, 11, *power* 18dbm, dan RSSI sebesar 55dbm. Pada bangunan ketiga, estimasi AP yang dibutuhkan adalah 6 buah, dimana masing-masing AP menggunakan *channel* 1, 7, 6, 2, 11 dengan *power* 18dbm dan rata-rata RSI sebesar -55dbm. Pada gedung ke empat menggunakan estimasi AP sebanyak 5 buah, masing-masing *channel* yang digunakan adalah 6, 1, 11, 10, 5 *power* yang digunakan adalah 18dbm, dengan rata-rata RSI sebesar -50dbm. Gedung kelima, estimasi AP yang digunakan sebanyak 4 buah dengan *channel* 1, 6, 11, 7, kemudiann *power* yang digunakan sebesar 18dbm dan rata-rata RSSI sebesar -55dbm.

2.2 WLAN (Wireless Local Area Network)

WLAN adalah salah satu pengembangan dari LAN, dimana WLAN merupakan bentuk sebuah jaringan lokal dengan media transmisi berupa gelombang radio.



Gambar 2.1 Infrastruktur WLAN

Seperti yang terlihat pada **Gambar 2.1**, pada umumnya WLAN digunakan sebagai titik distribusi pada tingkat *end-user*. WLAN merupakan suatu sistem

komunikasi yang dapat digunakan sebagai perpanjangan atau sebagai alternatif pengganti jaringan kabel (Widyaningsih, 2012). Dengan media transmisi tersebut WLAN melakukan komunikasi dan menyalurkan data dari satu perangkat ke perangkat lainnya, sehingga mampu memberikan fleksibilitas dalam penggunaannya. Dalam pembangunan infrastruktur WLAN terdapat beberapa komponen, salah satunya adalah *Access Point*(AP).

AP berperan untuk menerima dan menyalurkan data, dengan AP cakupan wilayah WLAN menjadi lebih luas. Pada tingkat backbone, sejumlah AP akan tetap terhubung dengan media kabel. WLAN bertujuan sebagai media alternatif dalam menjangkau pengguna yang tidak terlayani oleh jaringan kabel, serta untuk mendukung pergerakan pengguna yang dinamis.

2.2.1 Teknologi WLAN

Dalam aplikasi WLAN frekuensi yang umum dipergunakan adalah 2.4 Ghz dan 5.8 Ghz. Frekuensi tersebut secara internasional masuk dalam wilayah license exempt yang berarti bebas lisensi, sehingga dapat dipergunakan bersama-sama oleh publik.

Standar yang digunakan oleh sebagian besar teknologi WLAN adalah IEEE 802.11 (a/b/g). Standar tersebut sama dengan standar yang digunakan oleh jaringan Wifi. Dimana standar ini menentukan batasan kapasitas layanan yang dihasilkan. Pada 802.11b memiliki kapasitas maksimal 11 Mbps, sedangkan pada 802.11g memiliki kapasitas maksimal 20 Mbps. Kedua standar tersebut bekerja di frekuensi 2.4 Ghz. Sementara standar 802.11a bekerja pada frekuensi 5.8 Ghz [AJI-06]. Perangkat dengan basis standar ini memiliki pita frekuensi yang lebih luas dan modulasi yang lebih baik sehingga mampu melewati data dengan kapasitas 54 dan 108 Mbps dan juga mampu menampung jumlah pengguna yang lebih banyak.

2.2.2 Jaringan *wireless*

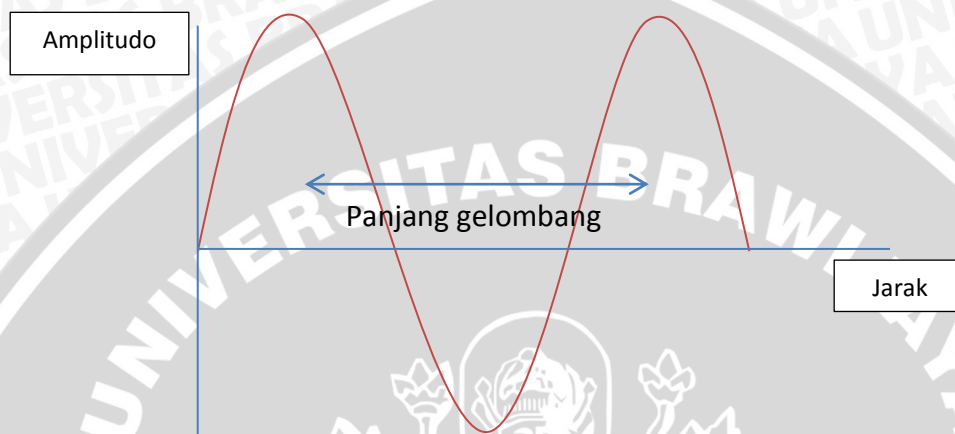
Jaringan *Wireless* atau yang biasa disebut dengan WiFi memiliki faktor-faktor yang harus diketahui, faktor-faktor tersebut diantaranya adalah panjang gelombang, frekuensi, *amplitudo*, EIRP (*Effective Isotropic Radiated Power*), redaman, LOS (*Line Of Sight*), *free space path loss*, dan pergerakan sinyal.

1. *Access Point* (AP)

Access Point (AP) adalah perangkat keras yang digunakan untuk menghubungkan perangkat nirkabel dengan infrastruktur jaringan menggunakan gelombang radio, atau dengan teknologi Wi-Fi. AP memiliki berbagai macam spesifikasi dikarenakan untuk memenuhi kebutuhan jaringan pada tata letak bangunan. Pada penerapannya AP juga digunakan sebagai perangkat yang memperluas cakupan sinyal dari suatu jaringan, selain itu AP juga menggunakan *router* sehingga dapat mengirimkan paket data sekaligus membagi dan mendistribusikan IP *Address*. AP juga memiliki kemampuan untuk membagi beban yang diterima dari *user* menuju ke AP lainnya.

2. Panjang Gelombang (Wavelength)

Panjang Gelombang merupakan satuan jarak dari pola gelombang yang berulang, yang diukur dari satu titik unit ke titik yang sesuai di unit berikutnya. Panjang gelombang berbanding terbalik dengan frekuensi gelombang. Dengan kata lain, semakin pendek panjang gelombang, akan memiliki frekuensi yang besar. Contoh dari panjang gelombang dapat dilihat pada **Gambar 2.2**.



Gambar 2.2 Contoh Panjang Gelombang

3. Frekuensi (Hz)

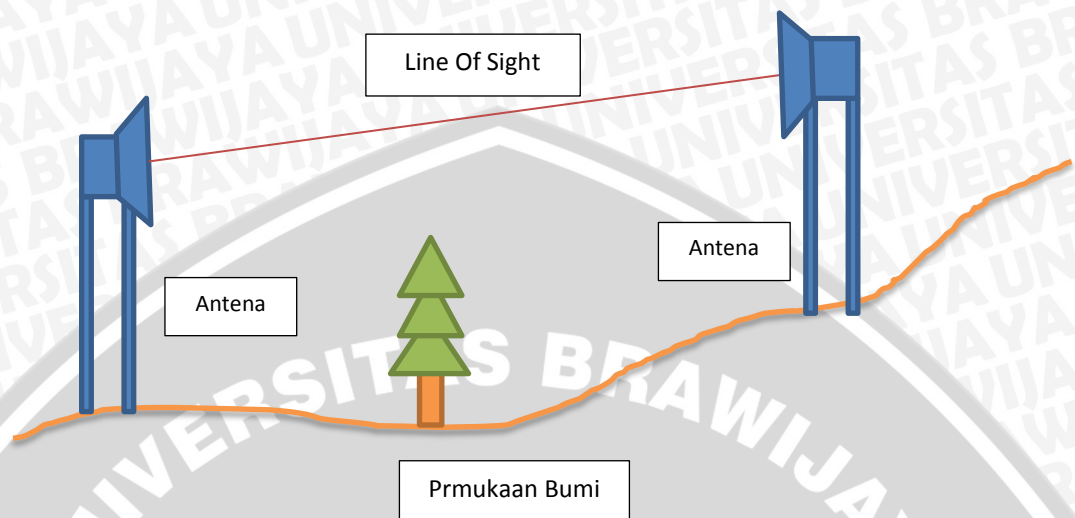
Frekuensi merupakan ukuran jumlah perulangan suatu kejadian. Pada gelombang, frekuensi menunjukkan intensitas sinyal dapat terlihat atau muncul dalam satu detik. Beberapa fakta tentang frekuensi yaitu :

- 1 cycle = 1 Hz
- Frekuensi tinggi memiliki jangkauan jarak lebih pendek
- Ketika sinyal muncul 1x dalam 1 detik = 1 Hz
- 10x dalam 1 detik = 10 Hz
- 1 juta x dalam 1 detik = 1 MHz
- 1 miliar x dalam 1 detik = 1 GHz

4. LOS

Line Of Sight merupakan jalur antara antenna yang saling berhubungan, dimana jalur tersebut harus terbebas dari adanya halangan. *Line Of Sight* juga dapat diartikan sebagai jarak pandang dari antenna yang saling berhubungan. Hal ini berarti bahwa antena-antena tersebut harus dapat saling terlihat, seperti pada **Gambar 2.3**. Agar dapat terlihat masing-masing dari antenna tidak boleh terhalang oleh sesuatu yang mampu menghalangi jarak pandang dari antenna-antena yang lain. Halangan berupa benda padat seperti beton, kayu, bahkan manusia dapat berdampak timbulnya redaman pada sinyal. Redaman tersebut dapat

melemahkan kekuatan sinyal, sehingga kualitas sinyal yang diharapkan tidak dapat terpenuhi.



Gambar 2.3 Ilustrasi LOS

5. Free Space Path Loss

Free Space Path Loss adalah suatu fenomena menurunnya kekuatan sinyal pada jalur LOS antara antena yang saling berhubungan. Penurunan kekuatan sinyal ini tidak disebabkan oleh adanya halangan pada jalur LOS, melainkan kekuatan sinyal melemah dengan sendirinya seiring dengan semakin jauhnya jarak yang ditempuh sinyal dari sumber sinyal dipancarkan.

Berkurangnya kekuatan sinyal dapat dianalogikan seperti gelombang yang tercipta saat kita menjatuhkan batu pada permukaan air. Gelombang yang semakin me jauh dari sumbernya lama kelamaan akan semakin melemah dan menghilang. Sama halnya dengan sinyal, gelombang elektromagnetik yang dipancarkan dari sumbernya juga akan melemah dan pada akhirnya akan menghilang.

Nilai *path loss* pada jarak antara *transmitter* dan *receiver* dapat ditentukan dengan rumus sebagai berikut:

$$pL_{ij} = pL(d_0) + 10n_0 \log \left(\frac{d_{ij}}{d_0} \right) + K\sigma \quad (2-1)$$

Keterangan :

- pL_{ij} = *path loss* yang terjadi pada jarak antara *user i* dan AP *j*.
- $pL(d_0)$ = *path loss* jarak terjauh jangkauan sinyal AP
- i = himpunan *user*.
- j = himpunan AP.
- n_0 = merupakan *path loss* eksponen yang bernilai 3.02 [PRO-02].
- d_{ij} = jarak antara *user i* dengan AP *j*.

d_0 = jarak terjauh yang mampu dijangkau oleh sinyal AP.
 $K\sigma$ = merupakan *shadow fading margin* yang bernilai 10 [PRO-02].

Untuk menentukan *path loss* pada jarak referensi dapat dilakukan dengan rumus berikut :

$$pL(d_0) = 10 \log \left[\left(\frac{4\pi d_0 f_j}{3 \times 10^8} \right)^2 \right] \quad (2-2)$$

Keterangan :

$pL(d_0)$ = *path loss* jarak terjauh jangkauan sinyal AP.
 f_j = frekuensi yang digunakan oleh AP j.
 d_0 = jarak terjauh yang mampu dijangkau oleh sinyal AP.

6. Capacity and coverage area

Kapasitas dan cakupan dari suatu AP merupakan suatu komponen yang penting dalam merencanakan infrastruktur WLAN yang baik. Tata letak dan desain suatu bangunan yang bermacam-macam menuntut WLAN untuk dapat menyesuaikan sehingga mampu memberikan kualitas jaringan yang baik pada *user*. Untuk dapat menyesuaikan dengan ruangan, WLAN dapat menggunakan berbagai macam AP dengan jenis jangkauan dan cakupan sinyal yang berbeda-beda. Pada penelitian ini menggunakan beberapa batasan (*constraints*) terkait dengan kapasitas dan jangkauan dari AP yang digunakan, seperti yang ditunjukkan pada persamaan berikut ini:

$$\sum_{j=1}^n u_{ij}(p_j - pL_{ij}) \geq P_R \quad (2-3)$$

Keterangan :

u_{ij} = merupakan variabel yang menyatakan *user i* terhubung dengan AP j atau tidak.
 P_R = merupakan batasan kekuatan sinyal yang diterima oleh *receiver* yakni sebesar -80 dBm.
 p_j = merupakan power level yang digunakan oleh AP j.
 pL_{ij} = *path loss* yang terjadi pada jarak antara *user i* dan AP j.
 n = jumlah AP yang digunakan.
 j = himpunan AP.

$$10 \log \left[\sum_{j=1}^n (1 - u_{ij}) \log^{-1}(p_j - pL_{ij}) \right] \leq P_I \quad (2-4)$$

Keterangan :

u_{ij} = merupakan variabel yang menyatakan *user i* terhubung dengan AP j atau tidak.
 n = jumlah AP yang digunakan.
 j = himpunan AP.
 p_j = merupakan power level yang digunakan oleh AP j.
 pL_{ij} = *path loss* yang terjadi pada jarak antara *user i* dan AP j.

$$\sum_{i=1}^m d_i \times u_{ij} \leq \beta C_{AP} \quad (2-5)$$

Keterangan :

- d_i = kebutuhan kecepatan jaringan yang dibutuhkan *user*.
- u_{ij} = merupakan variabel yang menyatakan *user* i terhubung dengan AP j atau tidak.
- β = merupakan efektifitas kapasitas AP yang ditetapkan sebesar 90% atau 0,9.
- C_{AP} = kapasitas AP.
- m = jumlah *user*.

$$\sum_{j=1}^n g_{hj} \geq 1 \quad (2-6)$$

Keterangan :

- n = jumlah AP.
- j = himpunan AP.
- g_{hj} = *grid* lokasi AP j.

Persamaan (2-3) merupakan batasan yang menyatakan bahwa kekuatan sinyal yang diterima harus lebih besar dari batas sensitivitas *wireless terminal*. Persamaan (2-4) digunakan untuk mendeteksi gangguan yang terjadi pada sinyal. Persamaan (2-5) digunakan untuk memastikan jumlah *traffic demand* dari *user* tidak melebihi kapasitas AP. Persamaan (2-6) menyatakan bahwa suatu *grid point* pada ruangan mampu menerima sinyal radio yang dipancarkan *transmitter*.

2.3 Pengertian penyelesaian terbaik dan optimasi

Pada dasarnya penyelesaian terbaik bersifat relatif, tergantung dari permasalahan yang dihadapi, metode penyelesaian, dan toleransi yang diizinkan. Setiap permasalahan juga memiliki beberapa solusi dalam menanganinya, sehingga manusia perlu untuk memutuskan solusi terbaik dari suatu permasalahan. Untuk itulah diperlukan optimasi.

Optimasi merupakan suatu proses yang berhubungan dengan penyesuaian masukan, pemilihan karakteristik peralatan, proses matematis dan pengujian yang dilakukan untuk menemukan keluaran optimum (Haupt, 2004). Dengan optimasi manusia dapat menentukan solusi yang paling tepat dalam menyelesaikan suatu permasalahan.

2.4 Algoritma genetika

Algoritma Genetika merupakan suatu metode heuristik yang dikembangkan berdasarkan prinsip genetika dan proses seleksi alamiah Teori Evolusi Darwin. Metode optimasi dikembangkan oleh John Holland pada tahun 1960-an dan dipopulerkan oleh David Goldberg pada tahun 1980-an (Haupt, 2004). Proses pencarian penyelesaian atau proses terpilihnya sebuah penyelesaian dalam

algoritma ini berlangsung sama seperti terpilihnya suatu individu untuk bertahan hidup dalam proses evolusi.

Dalam Teori Evolusi Darwin, suatu individu tercipta secara acak kemudian berkembang biak melalui proses reproduksi sehingga terbentuk sekumpulan individu sebagai suatu populasi. Setiap individu dalam populasi mempunyai tingkat kebugaran yang berbeda-beda. Tingkat kebugaran ini menentukan seberapa kuat individu untuk tetap bertahan hidup dalam populasinya. Individu dengan kebugaran yang buruk memiliki kemungkinan yang lebih besar untuk mati.

Pada teori ini dikenal adanya proses seleksi alam yang mempertahankan individu dengan tingkat kebugaran yang tinggi untuk tetap bertahan hidup. Sebagai contoh sederhananya adalah hewan jerapah. Menurut teori ini awal mula hewan jerapah berleher pendek, dikarenakan seleksi alam jerapah jenis ini punah. Jerapah dengan leher yang lebih panjang memiliki kebugaran yang lebih baik, sehingga dapat bertahan hingga saat ini. Jerapah yang sekarang kita ketahui telah terpilih secara alamiah dalam proses evolusi yang berlangsung dari generasi ke generasi.

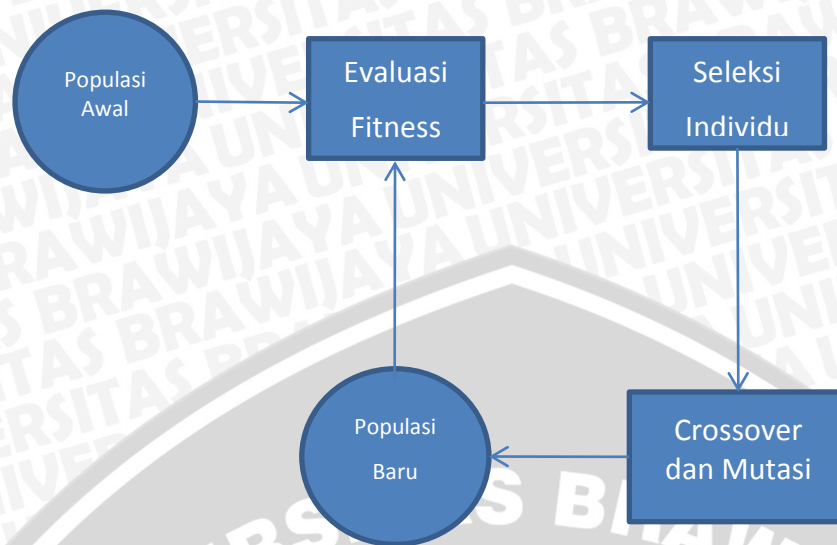
Demikian juga dalam proses pencarian yang berlangsung dalam Algoritma Genetika. Pencarian dimulai dengan memunculkan individu secara acak yang disebut dengan kromosom. Kromosom merupakan gambaran dari suatu kemungkinan solusi yang nantinya akan diperiksa nilai kebenarannya. Layaknya dalam teori evolusi, kromosom tersebut akan akan dinilai tingkat kebugarannya. Kromosom dengan tingkat kebugaran yang tinggi akan terpilih untuk bertahan di populasinya.

Tahap selanjutnya adalah, kromosom-kromosom yang terpilih diantaranya akan melakukan proses reproduksi dengan metode penyilangan atau *crossover*. Dari proses reproduksi akan dihasilkan kromosom baru, dimana sebagian kecil dari kromosom tersebut akan mengalami mutasi.

Kromosom-kromosom terpilih yang tidak melakukan reproduksi akan digabungkan dengan kromosom-kromosom hasil reproduksi, sehingga akan menghasilkan populasi yang baru. Proses akan diulangi hingga sejumlah generasi dicapai. Penyelesaian ditentukan dari kromosom dengan tingkat kebugaran tertinggi pada generasi terakhir.

2.4.1 Dasar-dasar algoritma genetika

Algoritma Genetika merupakan metode pencarian yang didasarkan pada proses evolusi alamiah [TUR-98]. Secara umum Algoritma Genetika dapat dikerjakan melalui serangkaian tahapan, seperti yang ditunjukkan pada **Gambar 2.5**.



Gambar 2.5 Arsitektur Algoritma Genetika.

Algoritma Genetika berawal dari pembentukan populasi awal secara acak yang terdiri dari individu-individu dengan sifat yang sesuai dengan gen-gen pada kromosomnya. Selanjutnya dilakukan perhitungan fitness dari setiap individu. Kemudian individu-individu tersebut akan diseleksi untuk melakukan proses reproduksi sehingga menghasilkan individu yang baru. Sifat dari individu baru tersebut terbentuk dari sifat kedua induknya. Namun terdapat kemungkinan dimana muncul sifat-sifat baru yang merupakan dampak dari adanya mutasi genetik.

Seleksi alamiah merupakan evaluasi yang akan memilih individu dengan kemampuan yang lebih dalam mempertahankan diri pada populasi. Penyelesaian yang di dapat dari Algoritma Genetika sama halnya dengan keturunan yang mempunyai kemampuan mempertahankan diri paling tinggi yang dicari dalam seluruh populasi dari awal hingga akhir.

2.4.2 Proses algoritma genetika

Algoritma Genetika terdiri dari beberapa komponen utama, diantaranya adalah inisialisasi kromosom, *generate* populasi awal, representasi kromosom, perhitungan nilai *fitness*, *crossover*, mutasi, dan seleksi.

2.4.2.1 Inisialisasi kromosom

Inisialisasi merupakan proses dalam pembentukan kromosom-kromosom yang akan menyusun populasi. Populasi terdiri dari solusi-solusi yang dinyatakan dalam bentuk kromosom. Berbeda dengan proses alamiah, pada proses inisialisasi kromosom ini terdapat batasan ukuran populasi. Ukuran populasi ditentukan untuk membatasi perhitungan agar cakupan tidak terlalu besar.

2.4.2.2 Generate populasi awal

Pembentukan populasi awal merupakan pembentukan yang dari susunan individu yang dibangkitkan secara acak atau melalui prosedur tertentu untuk mencari penyelesaian yang optimal.

2.4.2.3 Representasi kromosom

Kromosom merupakan kumpulan solusi yang membentuk sebuah populasi. Untuk melakukan perhitungan terhadap solusi-solusi tersebut diperlukan representasi kromosom, sehingga sesuai dengan proses optimasi yang akan dilakukan. Representasi kromosom yang sesuai dapat berdampak pada kualitas solusi yang dihasilkan oleh perhitungan algoritma genetika. Pada penulisan skripsi ini menggunakan teknik *binary encoding*, dimana setiap kromosom terdiri dari susunan bilangan biner, sehingga setiap kromosom memiliki 2 nilai yaitu 0 dan 1.

Contoh representasi kromosom yang digunakan ditunjukkan pada **Tabel 2.2** :

Tabel 2.2 Contoh representasi kromosom

Kromosom	Gen			Kromosom setelah di kodekan dalam biner
	<i>Location</i>	<i>Frequency Channel</i>	<i>Power Level</i>	
1.	23	2.437	24	0010111 01 10
2.	36	2.462	20	0100100 10 10
3.	95	2.412	20	1011111 00 10
4.	104	2.437	15	1101000 10 00

Pada **Tabel 2.2** ditunjukkan bahwa 1 kromosom terdiri dari 3 gen, yaitu gen lokasi AP, frekuensi, dan juga *power level*. Dari setiap gen tersebut akan dikodekan ke dalam bilangan biner. Perlu diketahui sebelumnya bahwa setiap *grid* pada ruangan akan diberikan ID tertentu. Sehingga 1 ID *grid* tersebut berisi 1 tentang koordinat titik pada suatu ruangan. Untuk proses pengkodean lokasi AP menjadi bilangan biner adalah dengan mengacak ID *grid* pada ruangan yang memungkinkan untuk dapat dipasang AP. Setelah didapatkan ID tersebut kemudian rubah ID tersebut menjadi bilangan biner. Selanjutnya adalah frekuensi dan *power level*. Pada kedua gen ini masing-masing memiliki 3 pilihan yaitu, untuk frekuensi 2.412, 2.437, dan 2.462, sedangkan untuk *power level* 24, 20, dan 15. Ketiga pilihan tersebut akan dirubah menjadi pilihan ke 0, 1, dan 2. Setelah dirubah sedemikian rupa, selanjutnya adalah merubah bilangan tersebut menjadi bilangan biner.

2.4.2.4 Perhitungan nilai *fitness*

Pada setiap generasi, kromosom-kromosom akan mengalami proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness* (kebugaran). Bentuk fungsi *fitness* bergantung pada masalah yang akan dioptimasi. Nilai *fitness* suatu kromosom akan menunjukkan kualitas kromosom dalam populasi. Semakin besar nilai *fitness* (jika fungsi minimalisasi maka semakin kecil nilai *fitness*), maka semakin besar kemungkinan kromosom tersebut dipertahankan dalam populasi selanjutnya (Zuchri, 2013).

Objective utama dari penulisan skripsi adalah untuk menentukan lokasi AP dengan nilai rata-rata *path loss* yang paling sedikit. Rumus dari fungsi *fitness* yang digunakan dalam penulisan skripsi ini ditunjukkan pada persamaan berikut ini:

$$\frac{1}{\min\left\{\frac{\sum_{j=1}^n u_{ij} pL_{ij}}{m}\right\}} \quad (2-7)$$

Keterangan :

- i = himpunan *user* (*Receiver*).
- j = himpunan Akses Point (*Transmitter*).
- pL_{ij} = merupakan nilai *path loss* berdasarkan jarak antara *user* i dengan AP j .
- u_{ij} = merupakan variabel yang menyatakan *user* i terhubung dengan AP j atau tidak.
- m = merupakan jumlah *user*.
- n = merupakan jumlah AP yang digunakan.

2.4.2.5 Proses *crossover*

Pada proses alamiah terdapat mekanisme reproduksi yang biasa disebut dengan mekanisme penyilangan (*crossover*). Pada proses penyilangan tersebut memungkinkan anak (*offspring*) akan mewarisi sifat dari kedua induknya (*parents*). Mekanisme tersebut juga diterapkan dalam Algoritma Genetika dan disebut sebagai operator penyilangan (*crossover*) [ZAI-13]. Proses penyilangan akan dilakukan pada setiap individu dengan nilai kemungkinan (probabilitas) yang telah ditentukan sebelumnya. Probabilitas *crossover* (*crossover rate*) itu sendiri merupakan nilai perbandingan jumlah kromosom yang diharapkan akan mengalami perkawinan silang terhadap jumlah kromosom dalam suatu populasi.

Penyilangan bisa dilakukan dengan memilih posisi pemotongan secara acak atau ditentukan. Pada makalah ini posisi penyilangan dipilih secara acak. Kromosom anak yang pertama dibentuk dari bagian awal kromosom induk pertama dan bagian akhir kromosom induk kedua. Sedangkan anak kedua dibentuk dari bagian awal kromosom induk kedua dan bagian akhir kromosom induk pertama. Ilustrasi mengenai metode penyilangan ini dijelaskan pada **Tabel 2.3**.

Pada ilustrasi tersebut operasi mutasi masih belum dikenakan. P1 dan P2 merupakan posisi pemotongan yang diterapkan pada penyilangan pertama dan kedua. Lokasi pemotongan tersebut dihitung dari sebelah kanan ke kiri. Pada ilustrasi, misal diperoleh nilai P1= 6 dan P2=2. Metode penyilangan pada ilustrasi ini menggunakan metode satu titik. Metode ini biasanya digunakan untuk representasi kromosom biner.

Tabel 2.3 Contoh penyilangan

Crossover	Penyilangan 1	Penyilangan 2
Proses Penyilangan	0010101 01 10 0100110 10 10	1011111 00 00 1101000 10 10
Hasil Penyilangan	0010111 01 10 0100100 10 10	1011111 00 10 1101000 10 00

2.4.2.6 Mutasi

Setiap generasi yang dihasilkan dari proses reproduksi generasi sebelumnya memiliki kemungkinan untuk mengalami perubahan gen keturunannya. Perubahan gen tersebut disebut dengan mutasi. Proses mutasi dilakukan pada setiap gen dengan probabilitas mutasi (*mutation rate*) yang telah ditentukan. Probabilitas tersebut yang menentukan terpilihnya suatu gen untuk dilakukan mutasi.

Pada penulisan skripsi ini representasi kromosom menggunakan kode biner sehingga untuk proses mutasi menggunakan metode cut point dimana pilih salah satu gen kromosom secara acak kemudian nilai gen kromosom diganti (Mahmudy, 2013), jika nilai gen kromosom 1 menjadi 0 atau nilai gen kromosom 0 menjadi 1, seperti pada **Gambar 2.6**.

2.4.2.7 Seleksi

Seleksi digunakan untuk memilih individu-individu manakah yang akan melakukan proses penyilangan dan mutasi. Seleksi digunakan untuk mendapatkan calon induk yang baik. "Induk yang baik akan menghasilkan keturunan yang baik". Semakin tinggi nilai *fitness* suatu individu, maka semakin besar kemungkinannya untuk terpilih.

Metode dalam melakukan seleksi ada beberapa macam, antara lain adalah sebagai berikut :

1. Mesin *Roulette*

Metode seleksi dengan menggunakan prinsip roulette, di mana setiap sektor dari roulette berisi nilai peluang dari setiap individu yang akan dipilih

2. *Rank* seleksi

Metode *roulette* memiliki keterbatasan jika fitness dari setiap individu yang digunakan mempunyai nilai dengan perbedaan yang tinggi. Metode *rank* seleksi memberikan *rank* sendiri terhadap setiap nilai *fitness*. Nilai *fitness* dari setiap kromosom diurutkan kemudian diberi nilai peluang sesuai dengan kelompok urutannya.

3. *Steady state*

Metode *steady state* memungkinkan setiap anggota populasi dirubah setiap saat. Perubahan ini terjadi pada proses mutasi dan *crossover*.

4. Turnamen

Metode ini didasarkan pada kompetisi untuk memilih n individual yang digunakan sebagai parent. Individu dengan nilai fitness tertinggi dikelompokkan dalam sebuah group yang akan terpilih dan individu lainnya dibuang.

5. Elitism

Dalam metode ini, kromosom terbaik digunakan sebagai individu terpilih untuk populasi berikutnya. Metode elitism dengan cepat dapat meningkatkan performance dari algoritma genetika.

2.4.3 Pemetaan proses alamiah ke proses komputasi

Dalam Algoritma Genetika diperlukan suatu proses untuk memetakan proses alamiah yang menjadi acuan komputasi kedalam sebuah algoritma. Komputasi tahap awal berdasarkan proses ilmiah adalah menyusun kemungkinan-kemungkinan penyelesaian kedalam sebuah populasi. Jika dalam proses alamiah kromosom mengandung ciri-ciri unik suatu individu, maka dalam Algoritma Genetika kromosom merupakan representasi penyelesaian masalah yang masih berbentuk simbol.

Pada pembentukan populasi awal dilakukan secara acak, sedangkan untuk populasi selanjutnya akan dibentuk oleh operator-operator Algoritma Genetika (Operator Reproduksi, Operator *Crossover*, Operator Mutasi) selama beberapa generasi. Penerapan operator-operator tersebut adalah membentuk populasi baru yang terdiri dari kromosom-kromosom yang bertahan dari generasi sebelumnya, dan sebagian lainnya merupakan kromosom yang dilahirkan dari kromosom generasi sebelumnya.

Proses seleksi alam yang terjadi pada proses ilmiah juga diterapkan pada Algoritma Genetika. Proses seleksi ini terjadi pada setiap generasi dengan menggunakan suatu fungsi yang disebut dengan fungsi *fitness*. Bentuk dari fungsi *fitness* bergantung pada jenis masalah yang akan dilakukan optimasi. Semakin besar nilai fungsi *fitness*, maka semakin besar pula kemungkinan kromosom untuk dapat bertahan dalam populasi.

Dalam Algoritma Genetika proses pembentukan anak dari induknya mirip dengan proses reproduksi yang dikenal dengan istilah penyilangan (*crossover*). Penyilangan memungkinkan anak (*offspring*) untuk mewarisi sifat dari kedua

induknya (*parents*). Namun terkadang suatu makhluk hidup dapat mengalami proses mutasi sehingga mengubah sifat-sifat genetiknya. Begitu pula pada penerapan Algoritma Genetika, terdapat kemungkinan kromosom hasil dari proses *crossover* mengalami mutasi. Kemungkinan ini ditentukan oleh salah satu operasi Algoritma Genetika yaitu operasi mutasi.

Berbeda proses alamiah, pada Algoritma Genetika terdapat batasan dalam ukuran populasi. Dalam Algoritma Genetika ukuran populasi pada setiap generasinya adalah tetap. Populasi generasi selanjutnya dibentuk dengan cara menyeleksi kromosom induk dan kromosom anak berdasar nilai *fitness*-nya.

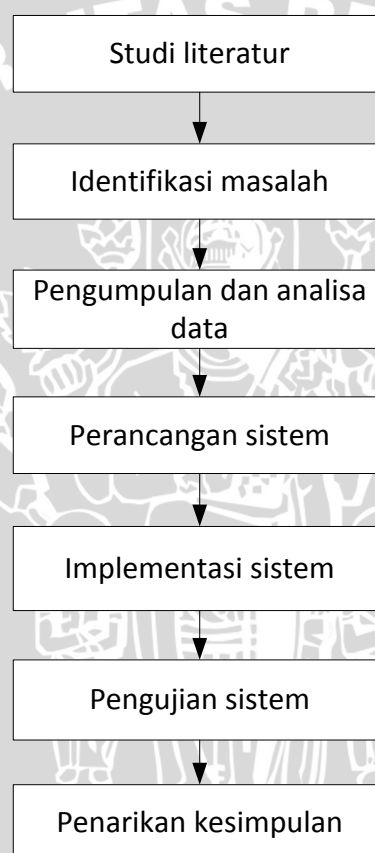
Secara singkat, pemetaan proses alamiah ke dalam proses komputasi Algoritma Genetika dapat dirangkum dalam **Tabel 2.5**

Tabel 2.5 Pemetaan Proses Alamiah ke Proses Komputasi

Proses Alamiah	Proses Komputasi
Individu	Penyelesaian masalah
Populasi	Himpunan penyelesaian
<i>Fitness</i>	Kualitas Penyelesaian
Kromosom	Kode/representasi penyelesaian
Gen	Bagian dari representasi penyelesaian
Pertumbuhan	Pengkodean representasi penyelesaian
Penyilangan	Operator Genetika
Mutasi	Operator Genetika
Seleksi Alam	Seleksi masalah berdasarkan kualitasnya

BAB 3 METODOLOGI

Pada bab metodologi penelitian ini membahas mengenai metode yang digunakan dalam penelitian optimasi penempatan akses point berdasarkan nilai minimal *pathloss* dengan menggunakan metode Algoritma Genetika. Metode penelitian berguna dalam mendukung peneliti agar penelitian dapat berjalan secara runtut dan terarah. Metode penelitian menjelaskan tahapan penelitian secara umum. Metode yang dilakukan pada penelitian ini terdiri dari beberapa tahapan, yaitu studi literatur, identifikasi masalah, pengumpulan dan analisa data, perancangan sistem, implementasi sistem, pengujian sistem, dan penarikan kesimpulan. Tahapan dari penelitian tersebut diilustrasikan pada **Gambar 3.1**.



Gambar 3.1 Metodologi penelitian

3.1 Studi literatur

Studi Literatur merupakan penelusuran literatur yang bertujuan untuk mempelajari konsep dasar teori yang digunakan untuk menunjang penelitian yang berkaitan dengan optimasi penempatan akses point berdasarkan nilai minimum *path loss* dengan menggunakan metode algoritma genetika. Penelitian literatur dapat bersumber dari jurnal, buku, internet, penelitian sebelumnya, dan juga bimbingan dari dosen pembimbing. Penyusunan dasar teori dilakukan

setelah mendapatkan beberapa referensi yang tepat untuk mendukung penulisan penelitian ini.

3.2 Identifikasi masalah

Tahap ini merupakan tahap awal dalam pembuatan suatu aplikasi. Tahap ini merupakan tahapan untuk mengkaji dan membatasi masalah yang akan diimplementasikan dalam aplikasi, mengumpulkan berbagai kebutuhan yang diperlukan sistem dan mengumpulkan data yang berkaitan dengan aplikasi yang akan dibuat. Identifikasi ini merupakan tahap awal yang akan menentukan keberhasilan dari tahap-tahap selanjutnya. Permasalahan yang menjadi latar belakang penelitian ini adalah menentukan lokasi optimal dari AP sehingga mampu menghasilkan kualitas sinyal yang baik dan cakupan sinyal yang maksimal.

3.3 Pengumpulan dan analisa data

Pengumpulan data spesifikasi AP dan tata letak AP pada gedung D PTIIK UB yang bersumber dari BPTIK (Badan Pengembangan Teknologi Informasi dan Komunikasi) UB PTIIK UB. Data yang diperoleh adalah data real berupa spesifikasi AP dan juga letak AP pada gedung D PTIIK UB saat ini. Lingkup pengumpulan data ini hanya sebatas pada area gedung D PTIIK UB dan gazebo PTIIK UB saja.

3.4 Perancangan sistem

Perancangan sistem bertujuan merancang urutan tahapan yang nantinya akan diterapkan pada tahap implementasi aplikasi. Pada tahap perancangan sistem, perancangan sistem optimasi penempatan lokasi AP berdasarkan nilai minimum *path loss* menggunakan algoritma genetika. Kebutuhan yang digunakan dalam perancangan sistem ini adalah sebagai berikut :

1. Spesifikasi Kebutuhan Hardware
 - Laptop Lenovo Thinkpad Edge 20C5A00Q01
 - RAM 4GB
 - DDR3
 - VGA 4GB
2. Spesifikasi Kebutuhan Software
 - Microsoft Windows 7 sebagai sistem operasi yang digunakan
 - PHP merupakan bahasa pemrograman untuk menjalankan aplikasi sistem ini
 - MySql sebagai sistem manajemen database
 - XAMPP

Kebutuhan data yang diperoleh dari BPTIK (Badan Pengembangan Teknologi Informasi dan Komunikasi) Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.

3.5 Implementasi sistem

Mengimplementasikan hasil analisis dan perancangan sistem dengan membangun sebuah program dengan menggunakan bahasa PHP. Aplikasi diimplementasikan dengan mengacu pada tahap perancangan yang dilakukan sebelumnya. Pada tahap ini akan diperlukan beberapa hal antara lain *input*, proses dan output yang dihasilkan oleh aplikasi.

Output dari sistem adalah lokasi, frekuensi, serta *power level* dari sejumlah AP dengan nilai *path loss* terendah .

3.6 Pengujian

Setelah tahap implementasi selesai dilakukan maka hasil yang diperoleh (berupa aplikasi) tentu harus diuji dan dievaluasi secara keseluruhan untuk menguji dan menemukan kesalahan yang mungkin terjadi guna memperoleh hasil yang maksimal.

Pengujian sistem adalah melakukan pengujian data AP yang ada terhadap sistem yang sudah dibangun. Pengujian yang digunakan adalah pengujian validasi.

3.7 Pengambilan kesimpulan

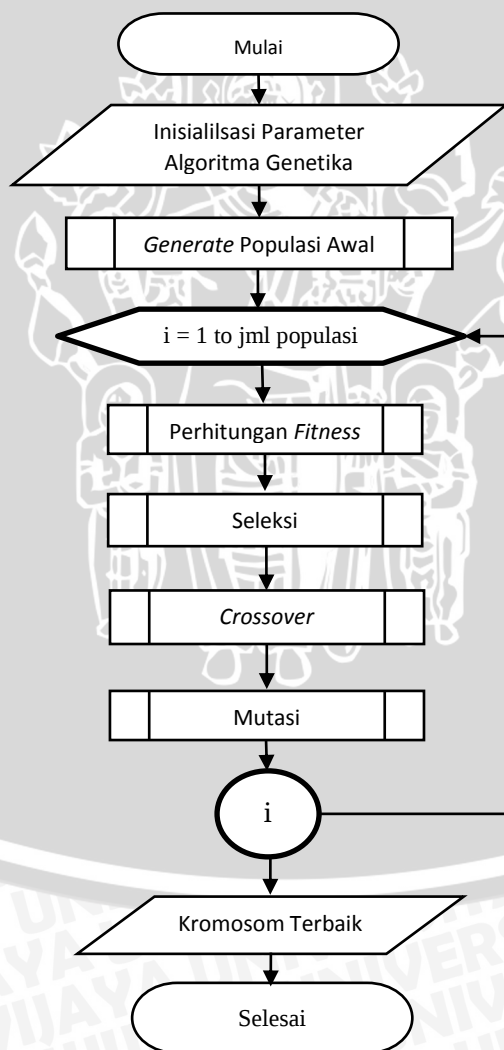
Penarikan kesimpulan dapat dilakukan setelah semua tahapan sebelumnya dilakukan, yang meliputi perancangan, implementasi, dan pengujian sistem. Kesimpulan diambil dari hasil pengujian berserta analisisnya terhadap sistem yang telah dibangun. Kemudian tahapan akhir yang dilakukan adalah penulisan saran. Penulisan saran dimaksudkan agar penelitian berikutnya mampu memperbaiki kekurangan yang terjadi, menyempurnakan penulisan, serta memberikan pertimbangan untuk pengembangan sistem selanjutnya.

BAB 4 PERANCANGAN

Bab perancangan ini terdiri atas beberapa fase, seperti Perancangan Sistem, Perhitungan Manual, Perancangan Antar Muka. Dalam tahap perancangan sistem akan dijelaskan mengenai penerapan proses Algoritma Genetika. Pada tahap perhitungan manual merupakan tahap perhitungan metode Algoritma Genetika dengan *input* jumlah AP dan jumlah populasi, serta *output* berupa kromosom dengan nilai *fitness* terbaik yang berisikan gen lokasi, frekuensi, *power level* AP. Pada perancangan antar muka, berisi rancangan antarmuka dan sistem pada web.

4.1 Perancangan sistem

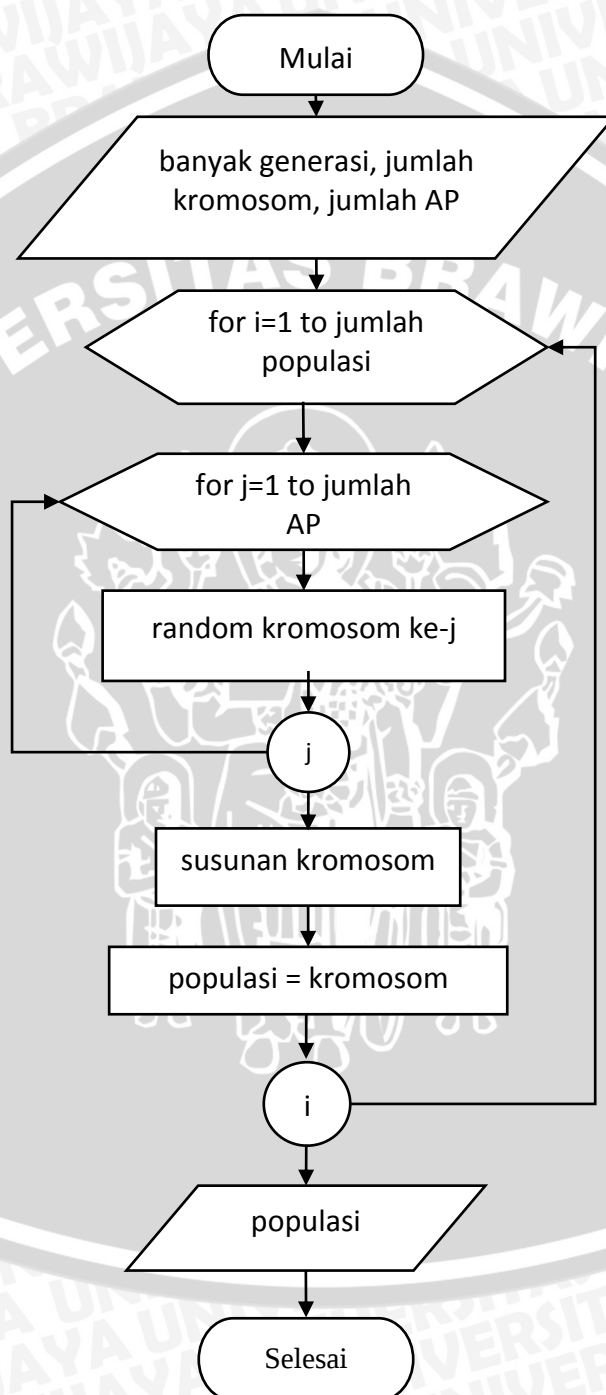
Berdasarkan analisis yang telah dilakukan, maka pada subbab ini akan dibahas mengenai proses pada sistem ini. Proses Algoritma Genetika terdiri dari beberapa tahapan seperti yang dapat dilihat pada **Gambar 4.1**.



Gambar 4.1 Diagram Alir Algoritma Genetika Pada Sistem

4.1.1 Generate populasi awal

Pada tahap pertama, dilakukan proses pembuatan kromosom dan pembentukan populasi awal sebanyak jumlah populasi yang telah ditentukan dengan panjang kromosom yang sama dengan jumlah tujuan yang diinginkan. Ditunjukkan dalam **Gambar 4.2**.



Gambar 4.2. Diagram Alir Membuat Populasi Awal

4.1.2 Crossover

Pada tahap *Crossover* ini sebanyak 70% kromosom dari populasi sebelumnya dengan nilai fitness terbaik akan dipilih untuk melakukan proses *Crossover*.

Dalam metode *Crossover* pada penelitian ini dilakukan dengan memilih titik potong pada kromosom kemudian menukarkan nilai gen *parent* satu dengan *parent* lainnya (Mahmudy, 2013). Berdasarkan **Gambar 4.3** langkah-langkah proses *Crossover*, yaitu:

- Titik potong *Crossover* ditentukan secara acak pada gen ke-4 untuk P_1 dan P_2 . Gen yang terpilih dipilih secara *random* dari segmen kromosom yang ada, dimana segmen pertama terdiri dari 7 bit biner gen untuk nomor *grid* lokasi AP, segmen kedua dan ke tiga terdiri dari 2 gen selanjutnya untuk gen frekuensi dan *power level*.

P_1

0	0	0	0	1	1	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---

P_2

0	1	1	1	0	1	1	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---

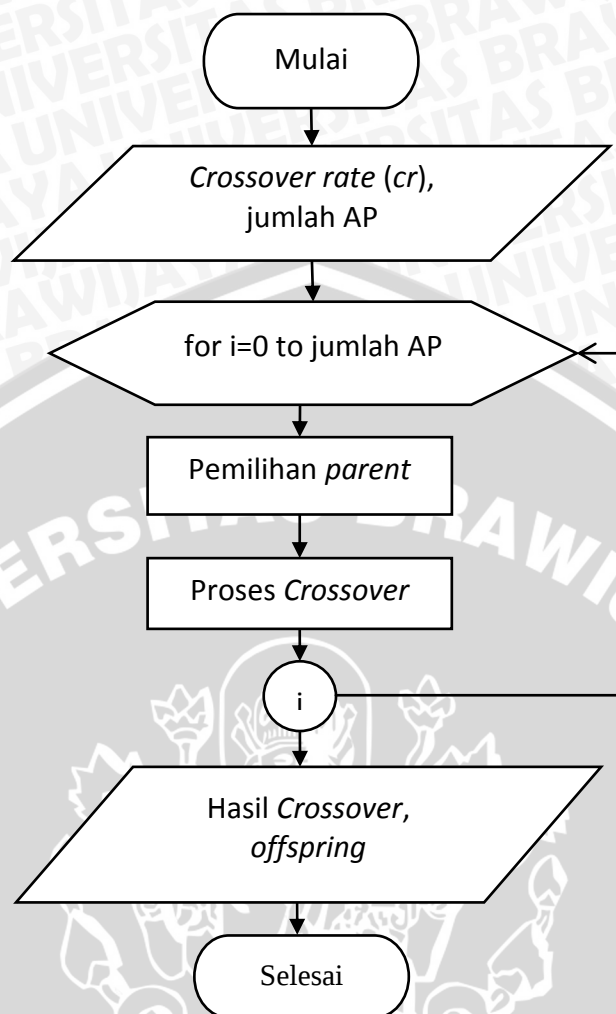
- Tukar nilai gen P_1 dan P_2 sesuai dengan titik potong yang telah ditentukan sebagai hasil dari *Crossover*, *offspring*.

C_1

0	0	0	0	0	1	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---

P_2

0	1	1	1	1	1	1	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---

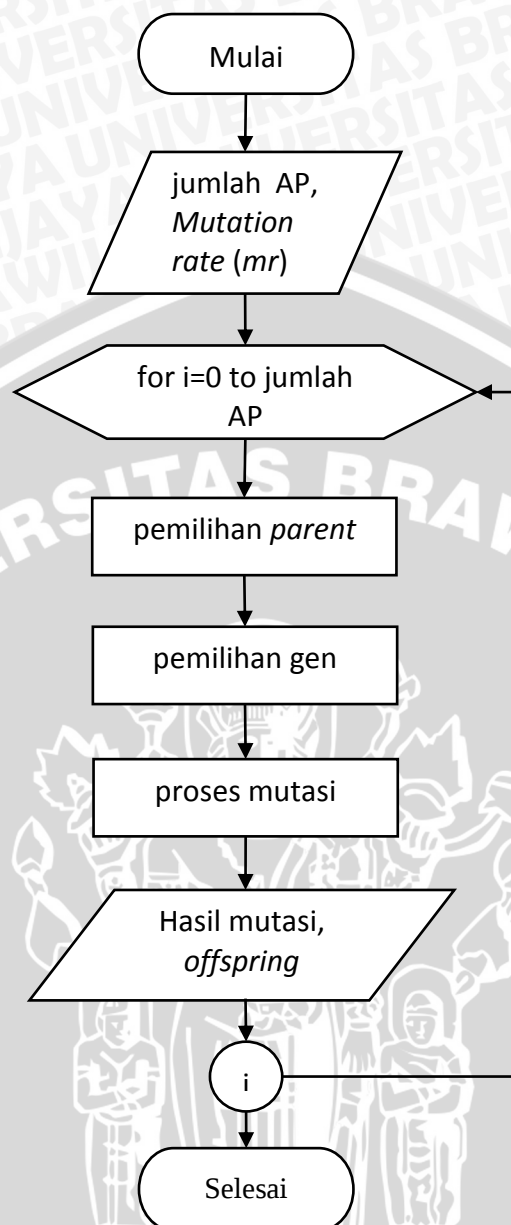


Gambar 4.3 Diagram Alir Proses Crossover

4.1.3 Mutasi

Proses dalam membuat generasi baru yang kedua adalah proses mutasi. Pada proses ini pilih secara acak kromosom kecuali kromosom dengan nilai fitness terbaik. Pemilihan tersebut berdasarkan nilai probabilitas mutasi (Mutation Rate) yang telah ditetapkan sebesar 10%.

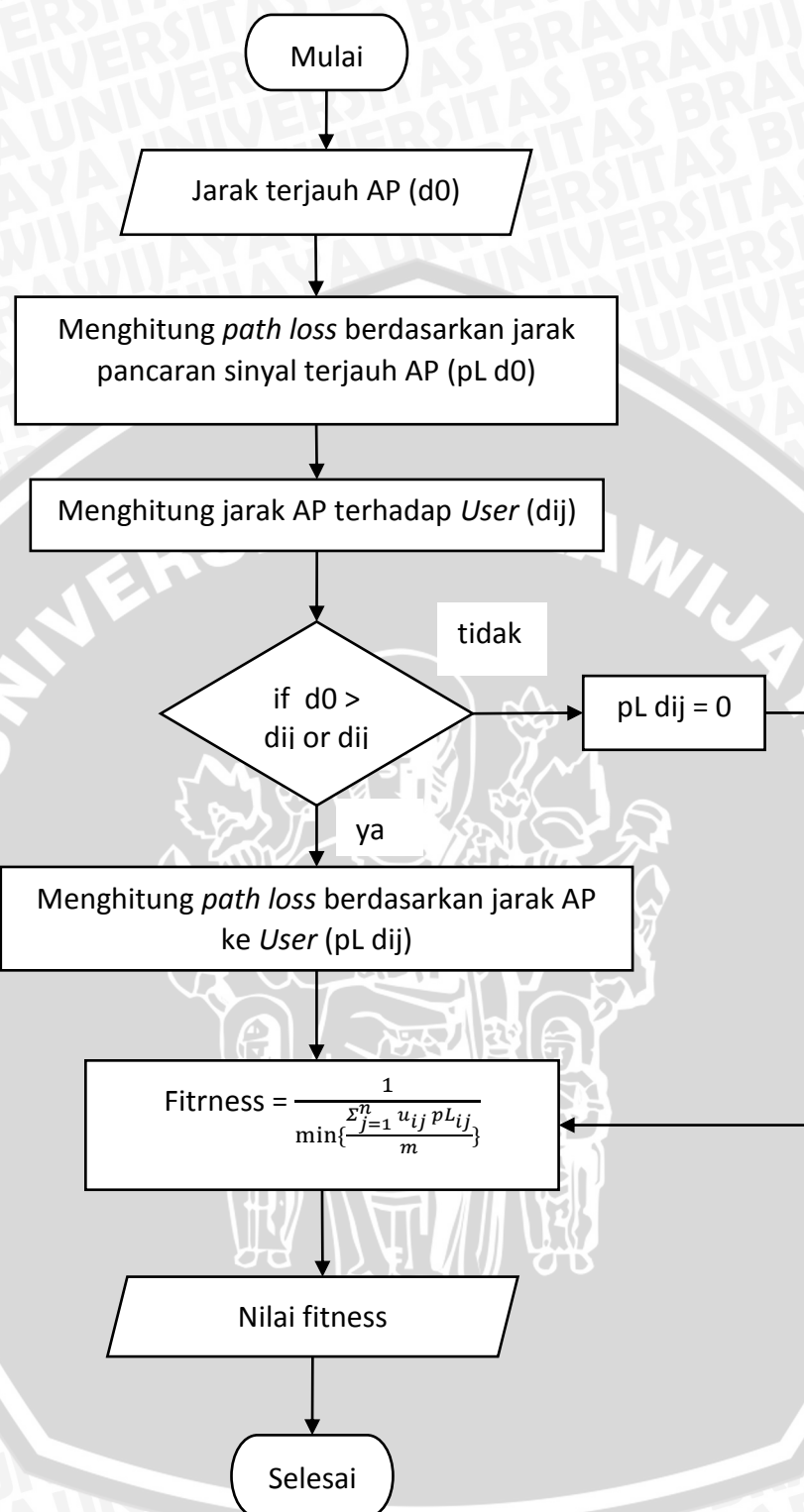
Kemudian kromosom yang dihasilkan dari proses mutasi ini akan digabungkan pada populasi yang telah ada, seperti pada **Gambar 4.4**.



Gambar 4.4 Diagram Alir Proses Mutasi

4.1.4 Menghitung nilai *fitness*

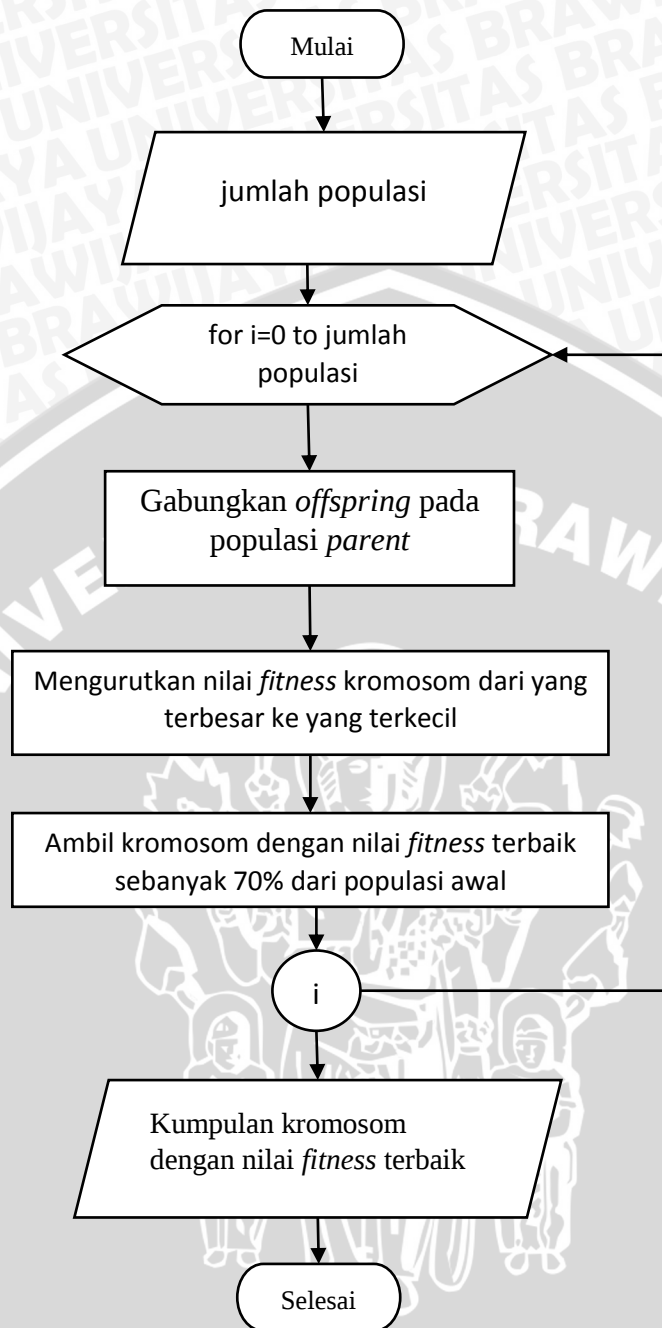
Setelah tahapan reproduksi yang terdiri dari tahap *crossover* dan mutasi akan dilakukan perhitungan nilai *fitness*. Perhitungan nilai tersebut dilakukan berdasarkan persamaan (2-7). **Gambar 4.5** akan menunjukkan tahapan dari proses menghitung nilai *fitness*



Gambar 4.5 Diagram Alir Perhitungan Nilai *Fitness*

4.1.5 Seleksi

Proses seleksi dilakukan dengan memilih 70% kromosom dengan nilai fitness tertinggi. Diagram alir dari proses selection seperti pada **Gambar 4.6**.



Gambar 4.6 Diagram Alir Proses Seleksi

4.2 Perhitungan manual

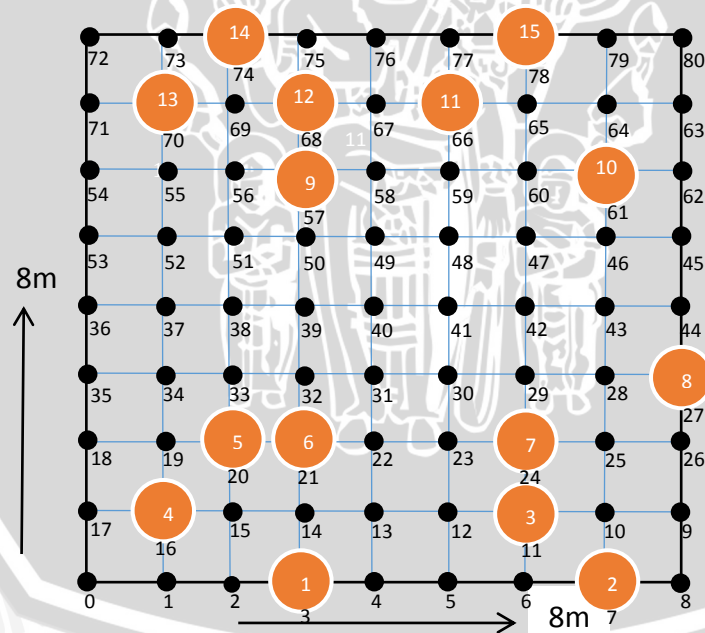
Pada ruangan Robotik PTIIK UB memiliki dimensi sebesar 8m x 8m, dan berdasarkan rata-rata kunjungan pada ruangan tersebut terdapat 15 *user* yang masing-masing akan menempati titik lokasi yang berbeda. Setiap koordinat dari titik tersebut akan diberi kode sebanyak rata-rata jumlah *user* pada ruangan. Untuk memenuhi kebutuhan jaringan *user*, digunakan 2 AP dengan spesifikasi sebagai berikut :

Jenis AP	:	UniFi AP
Frekuensi	:	2.412, 2.437, dan 2.462 GHz
Range	:	122 m
Power Level	:	15 dBm, 20 dBm, dan 24 dBm

Pemetaan ruangan dilakukan dengan metode *grid point*, dimana ruangan tersebut akan dibagi menjadi beberapa lingkup yang lebih kecil. Pada penelitian ini ruangan akan dibagi dengan lingkup berdimensi 1m x 1m. Setiap sudut dari lingkup tersebut akan diberi titik. Titik inilah yang akan menjadi titik lokasi baik lokasi *user* ataupun lokasi AP pada ruangan. Titik atau *point* ini akan diberi nomor sesuai urutannya, yang terhitung mulai dari titik paling kiri (0) bawah dan bertambah dengan pola “ular” sampai titik kanan atas (80) sehingga terdapat 81 titik yang memungkinkan untuk ditempati oleh *user* dan juga AP.

Pada **Gambar 4.7**, lokasi *user* ditentukan berdasarkan pengamatan secara langsung di lapangan, dan juga melalui kuesioner. Satu titik lokasi pada ruangan dapat ditempati oleh satu *user* saja, namun mengingat peletakan lokasi AP yang dilakukan secara acak maka memungkinkan bila dalam satu titik lokasi akan ditempati oleh satu *user* dan satu AP.

Berikut merupakan ilustrasi penerapan metode *grid point* dalam pemetaan ruangan Robotik PTIIK UB :



Gambar 4.7 Ilustrasi Letak User

Keterangan :

- = Grid Point
- = Lokasi User

4.2.1 Inisialisasi parameter

Parameter untuk membentuk populasi awal pada Algoritma Genetika diantaranya adalah jumlah populasi, jumlah kromosom, jumlah gen, probabilitas *crossover*, dan juga probabilitas mutasi.

Dimisalkan inisialisasi parameter awal adalah sebagai berikut:

- Jumlah populasi = 5
- Jumlah gen = 1000
- Probabilitas *Crossover* = 0.9
- Probabilitas Mutasi = 0.1
- Jumlah AP = 2

4.2.2 *Generate* populasi awal

Proses untuk membangkitkan populasi awal dilakukan dengan mengolah gen sedemikian rupa sehingga dapat membentuk suatu kromosom. Proses pengolahan tersebut dinamakan dengan proses *encoding*. Pada penelitian ini terdapat tiga gen yang menyusun satu kromosom, yaitu gen lokasi AP, frekuensi, dan juga *power level*. Ketiga gen tersebut nantinya akan diubah ke dalam bentuk biner dan kemudian disusun sehingga membentuk suatu deretan bilangan biner tertentu. Panjang kromosom ditentukan oleh banyaknya AP yang digunakan serta panjang dari deretan bilangan biner yang digunakan sebagai kode dari gen.

Tahap awal proses *encoding* gen lokasi AP adalah dengan melakukan pemilihan titik secara acak pada *grid point* yang ada pada ruangan. Dalam satu kromosom pemilihan titik ini dilakukan sebanyak jumlah AP yang digunakan. Telah dijelaskan sebelumnya bahwa setiap *grid point* pada ruangan telah memiliki nomor urut. Setelah mendapatkan titik secara *random* pada ruangan, kemudian catat nomor urut titik tersebut, kemudian rubah nomor tersebut ke dalam bilangan biner.

Panjang bilangan biner pada gen lokasi tergantung pada banyaknya *grid point* pada ruangan. Pada penelitian ini terdapat 81 *grid point* pada ruangan, sehingga panjang dari bilangan biner yang digunakan adalah tujuh bit biner. Hal tersebut dikarenakan bilangan biner dari angka 81 adalah 1010001, sehingga jika panjang bilangan biner tersebut kurang dari tujuh maka tidak semua *grid point* pada ruangan akan tercakupi.

Selanjutnya untuk melakukan proses *encoding* gen frekuensi dan juga *power level* terdapat tiga pilihan masing-masing gen yaitu 2.412 GHz, 2.437GHz, 2.467GHz untuk frekuensi dan 24 dBm, 20 dBm, 15 dBm untuk *power level*. Ketiga pilihan memiliki nomor urut bilangan desimal 0, 1, dan 2 yang kemudian dapat dirubah kedalam biner 00, 01, dan 10.

Setelah setiap gen telah dirubah dalam bentuk biner, langkah selanjutnya adalah dengan menyusun gen tersebut sehingga membentuk kromosom. Populasi merupakan kumpulan dari kromosom. Dalam membangkitkan populasi

awal akan dilakukan dengan pemilihan kromosom secara acak sebanyak jumlah populasi yang ditentukan. Dalam penelitian ini jumlah populasi yang digunakan adalah 5 kromosom, seperti yang ditunjukkan pada **Tabel 4.1**.

Tabel 4.1 Generate Populasi Awal

No. Kromosom	AP1	AP2
1	00001111100	00111110011
2	01010110010	01010101010
3	01010100100	01011011110
4	01110110111	01000011111
5	01111000010	00001011100

Pada **Tabel 4.1**, satu kromosom terdiri dari beberapa AP sesuai banyak AP yang digunakan, pada kasus ini menggunakan 2 AP. Pada setiap AP terbagi lagi menjadi 3 segmen gen yaitu, gen lokasi AP, gen frekuensi AP, dan juga gen *power level* AP.

Berikut merupakan ilustrasinya:

$$g_1 f_1 p_1 | g_2 f_2 p_2 | g_3 f_3 p_3 | \dots | g_j f_j p_j$$

Keterangan:

g_j = merupakan gen lokasi AP_j

f_j = merupakan gen frekuensi yang digunakan AP_j

p_j = merupakan gen *power level* AP_j

Tabel 4.2 Detail kromosom

No.	AP1									AP2								
	g_1			f_1			p_1			g_2			f_2			p_2		
1	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1	1
	(7,0)			2,412			24			(4,3)			2,412			15		



Gambar 4.8 Ilustrasi *Encoding* Gen Lokasi Pada Kromosom 1 AP ke-1

Pada **Tabel 4.2** menampilkan segmen-segmen yang terdapat pada kromosom ke-1, kromosom tersebut terbagi menjadi 3 segmen. 7 bit biner pertama merupakan segmen gen lokasi AP, 2 bit biner berikutnya merupakan segmen frekuensi AP, sedangkan 2 bit biner terakhir merupakan segmen *power level* yang digunakan AP.

Gambar 4.8 menjelaskan proses *encoding*. Proses ini merubah bilangan biner tersebut ke dalam bentuk desimal yang dapat digunakan dalam perhitungan. Untuk lokasi AP akan dirubah berdasarkan ID *grid point* dari lokasi tersebut. ID lokasi AP berisi koordinat titik pada ruangan yang memungkinkan untuk menjadi lokasi optimal AP.

4.2.3 Menghitung nilai *fitness*

Nilai *fitness* dihitung dengan menggunakan persamaan (2-7). Untuk menghitung nilai *path loss* antara AP dan *user* gunakan persamaan (2-1) dan (2-2). Berikut merupakan contoh perhitungan nilai *fitness* pada kromosom 1.

Tabel 4.3 Hasil Perhitungan $pL(d_0)$

No.	AP1										AP2											
	g_1							f_1		p_1	g_2							f_2		p_2		
1	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	1	1
	(7,0)							2,412		24	(4,3)							2,412		15		
$pL(d_0)$	81,81211038										81,81211038											

Pada **Tabel 4.3** ditunjukkan kromosom dengan ID bernomor 1 yang terdiri dari 2 AP. Pada AP 1 didapatkan nilai $pL(d_0)$ sebesar 81,81211038. Nilai tersebut didapatkan dari perhitungan dengan menggunakan persamaan (2-2) dengan nilai $d_0 = 122$ m, seperti berikut:

$$pL(d_0) = 10 \log \left[\left(\frac{4\pi d_0 f_j}{3 \times 10^8} \right)^2 \right]$$

$$pL(d_0) = 10 \log \left[\left(\frac{4 \times 3.14 \times 122 \times 2,412 \times 10^9}{3 \times 10^8} \right)^2 \right]$$

$$pL(d_0) = 81,81211038$$

Satuan frekuensi yang digunakan dalam perhitungan diatas menggunakan satuan GHz, sehingga harus dikalikan dengan bilangan 10^9 . Selanjutnya adalah menghitung nilai (pL_{ij}). Untuk dapat menghitung (pL_{ij}) hitung terlebih dahulu jarak AP ke *user*. Berikut ini merupakan tabel korrdinat *user* yang akan digunakan untuk menghitung jarak AP ke *user*.

Tabel 4.4 Koordinat User

Koordinat	X	y
User 1	3	0
User 2	7	0
User 3	6	1
User 4	1	1
User 5	2	2
User 6	3	2
User 7	6	2
User 8	8	3
User 9	3	6
User 10	7	6
User 11	5	7
User 12	3	7
User 13	1	7
User 14	2	8
User 15	6	8

Pada penelitian ini terdapat 15 *user* yang terdapat dalam ruangan Robotik PTIHK UB. Masing-masing letak *user* tersebut ditentukan oleh foto di lapangan ataupun kuesioner. Untuk menentukan jarak antara AP dengan *user* gunakan persamaan berikut ini.

$$(Jarak\ AP\ ke\ User)^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 \quad (4-1)$$

Keterangan :

- x_2 = korrdinat sumbu x lokasi AP
- y_2 = korrdinat sumbu y lokasi AP
- x_1 = korrdinat sumbu x lokasi *user*
- y_1 = korrdinat sumbu y lokasi *user*

Jika pada kromosom 1 didapati lokasi AP 1 terletak pada koordinat (7,0) maka perhitungan jarak AP 1 ke *user* 1 adalah sebagai berikut :

$$\begin{aligned}(\text{Jarak AP ke User})^2 &= (x_2 - x_1)^2 + (y_2 - y_1)^2 \\&= (7 - 3)^2 + (0 - 0)^2 \\&= 4^2\end{aligned}$$

$$\text{Jarak AP ke User} = 4$$

Selanjutnya adalah menghitung nilai *pathloss* antara AP dengan *user* (pL_{ij}). Perhitungan dilakukan pada setiap AP dengan persamaan (2-1), berikut merupakan tahap perhitungan (pL_{ij}) pada kromosom 1 AP ke-1 dan *user* ke-1.

$$pL_{ij} = 81,81211038 + 10 \times 3,02 \log\left(\frac{4}{122}\right) + 10$$

$$pL_{ij} = 47,16447013$$

Kemudian menghitung rata-rata nilai pL_{ij} pada *user* 1, dengan cara menambahkan seluruh nilai pL_{ij} *user* 1 dari setiap AP kemudian dibagi dengan seluruh jumlah *user*.

$$\begin{aligned}\text{Rata-rata path loss pada user 1} &= (pL_{11} + pL_{12})/m \\&= (46,98625523 + 43,90404349)/15 \\&= 6,059353248\end{aligned}$$

Setelah diperoleh nilai rata-rata pL_{ij} , maka langkah selanjutnya adalah memilih nilai rata-rata pL_{ij} dengan nilai terkecil dari setiap *user*. Pada penelitian ini di dapatkan nilai rata-rata pL_{ij} terkecil sebesar 3.183910554, sehingga perhitungan nilai *fitness* dari kromosom 1 berdasarkan persamaan (2-7) adalah sebagai berikut :

$$\begin{aligned}\frac{1}{\min\left\{\frac{\sum_{j=1}^n u_{ij} pL_{ij}}{m}\right\}} &= \frac{1}{3.183910554} \\&= 0,314079175\end{aligned}$$

Semakin tinggi nilai *fitness* dari suatu kromosom, maka semakin besar kromosom tersebut akan dapat bertahan pada populasi. Berikut merupakan tabel hasil perhitungan nilai *fitness* pada tiap kromosom.

Tabel 4.5 Hasil Perhitungan Nilai Fitness Tiap Kromosom

No. Kromosom	Nilai <i>Fitness</i>
1	0,314079175
2	0.205799987
3	0.205547405
4	0.190339519
5	0,182977242

4.2.4 Seleksi

Pada proses seleksi, kromosom-kromosom pada populasi akan diurutkan berdasarkan nilai *fitness* terbaik. Kemudian 70% kromosom dengan nilai *fitness* terbaik dari populasi saat ini akan dipilih untuk menjalani proses *crossover*. Pada **Tabel 4.6** menunjukkan kromosom mana sajakah pada populasi yang dapat melakukan proses *crossover*. Kromosom dengan warna merah, merupakan kromosom yang tidak dapat melakukan proses *crossover*.

Tabel 4.6 Kromosom Dengan Nilai *Fitness* Terbaik

ID Kromosom	Nilai <i>Fitness</i>
P1	0,314079
P2	0,2058
P3	0,205547
P4	0,19034
P5	0,182977

Crossover

Proses *crossover* dilakukan dengan cara memilih 2 kromosom terbaik secara acak dari 70% populasi saat ini. Pada penelitian jumlah populasi yang digunakan sebanyak 5 kromosom, sehingga 70% dari 5 kromosom adalah 3.5, kemudian dibulatkan menjadi 4 kromosom. 4 kromosom dengan nilai tertinggi inilah yang dapat melakukan proses *crossover*.

Selanjutnya adalah dengan memilih titik pertukaran bit biner pada kedua *parent*. Langkah berikutnya adalah menukarkan kedua titik pertukaran dan menyusun kembali kromosom tersebut, sehingga menghasilkan kromosom baru (*offspring*).

Tabel 4.7 Proses *Crossover*

P2	01010110010 01010101010
P1	00001111100 00111110011

Pada **Tabel 4.7** angka yang berwarna merah merupakan gen yang terpilih secara acak yang kemudian akan ditukarkan sehingga membentuk kromosom baru. Pada contoh diatas gen yang terpilih adalah G1=5, dan G2=11 dihitung dari bagian kanan kromosom.

Berikut adalah hasil dari proses *crossover* diatas :

Tabel 4.8 Hasil *Crossover*

P6	01010110010 01010101010
----	-------------------------

Populasi berikutnya tersusun oleh kromosom dengan persentase 20% kromosom dengan nilai terbaik dari populasi saat ini, kemudian 70% dari kromosom terbaik hasil proses *crossover* dan mutasi, serta 10% kromosom yang dibangkitkan secara acak.

4.2.5 Mutasi

Sebelum melakukan proses mutasi, lakukan pengurutan berdasarkan nilai *fitness* terbaik pada populasi saat ini. Tahap selanjutnya adalah dengan melakukan pemilihan kromosom secara acak terhadap populasi berdasarkan nilai probabilitas mutasi sebesar 10 %, namun kromosom dengan *fitness* terbaik tidak boleh terpilih dalam proses mutasi ini.

Pada proses mutasi ini pilih secara acak satu gen lokasi AP kemudian rubah nilai gen tersebut, misalkan nilai gen adalah 0 maka ganti menjadi 1, begitu juga sebaliknya.

Tabel 4.9 Proses Mutasi

P3	01010100100 01011011110
P11	01010110100 01011011111

4.2.6 Random injection

Proses ini merupakan pembangkitan suatu kromosom secara acak sebanyak 10% dari populasi saat ini. *Random injection* digunakan untuk menghindari terjadinya nilai *fitness* yang selalu berulang. Dengan *random injection* diharapkan dapat menambah variasi nilai sehingga tidak terjebak pada nilai yang sama terus menerus.

Untuk proses perhitungannya sendiri sama dengan proses sebelumnya dimana kromosom dibangkitkan secara acak, kemudian di *encoding* dan dihitung nilai *fitness*-nya.

Tabel 4.10 Hasil Random Injection

Random Injection			
Kromosom		Rata-rata pL_{ij}	Nilai <i>Fitness</i>
00101100010	01110100101	5,085162014	0,196650568

4.2.7 Status koneksi

Status koneksi merupakan faktor dalam perhitungan yang digunakan untuk mengasumsikan bahwa *user* sedang terhubung ke suatu AP atau tidak. Status koneksi ini dinyatakan dalam variabel u_{ij} . Pada dasarnya suatu *user* hanya bisa terkoneksi dengan 1 AP saja. Namun pengelolaan *user* seperti ini telah diatur secara otomatis oleh AP, sehingga AP tidak akan mengalami kelebihan beban.

Tabel 4.11 Ilustrasi Status Koneksi User

No.	AP1										AP2											
	g_1							f_1		p_1		g_2							f_2		p_2	
1	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	1	1
	(7,0)							2,412		24		(4,3)							2,412		15	
User 1	Status Koneksi																					
	0										0											

Pada **Tabel 4.11** menunjukkan suatu kasus dimana *user 1* tidak terkoneksi baik ke AP 1 ataupun AP 2. Keadaan ini nantinya akan mempengaruhi perhitungan nilai rata-rata pL_{ij} .

$$pL(d_0) = 10 \log \left[\left(\frac{4\pi d_0 f_j}{3 \times 10^8} \right)^2 \right]$$

$$pL(d_0) = 10 \log \left[\left(\frac{4 \times 3.14 \times 122 \times 2,412 \times 10^9}{3 \times 10^8} \right)^2 \right]$$

$$pL(d_0) = 81,81211038$$

$$(\text{Jarak AP1 ke User})^2$$

$$= (x_2 - x_1)^2 + (y_2 - y_1)^2$$

$$= (7 - 3)^2 + (0 - 0)^2$$

$$= 16$$

$$\text{Jarak AP1 ke User}$$

$$= 4$$

$$(\text{Jarak AP2 ke User})^2$$

$$= (x_2 - x_1)^2 + (y_2 - y_1)^2$$

$$= (4 - 3)^2 + (3 - 0)^2$$

$$= 10$$

$$\text{Jarak AP2 ke User}$$

$$= 3,16227766$$

$$pL_{11} = 81,81211038 + 10 \times 3,02 \log \left(\frac{4}{122} \right) + 10$$

$$pL_{11} = 47,16447013$$

$$pL_{12} = 81,81211038 + 10 \times 3,02 \log \left(\frac{3,16227766}{122} \right) + 10$$

$$pL_{12} = 43,90404349$$

$$\begin{aligned} \text{Rata-rata path loss pada user 1} &= ((u_{11} \times pL_{11}) + (u_{12} \times pL_{12}))/m \\ &= ((0 \times 47,16447013) + (0 \times 43,90404349))/15 \\ &= 0 \end{aligned}$$

Berdasarkan perhitungan diatas dapat di simpulkan bahwa *user* yang tidak terkoneksi sama sekali ke AP, tidak menghasilkan nilai rata-rata *path loss*.

Tabel 4.12 Ilustrasi Status Koneksi Dengan 3 User

No.	AP1										AP2											
	g_1							f_1		p_1		g_2							f_2		p_2	
1	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	1	1
	(7,0)							2,412		24		(4,3)							2,412		15	
Status Koneksi																						
User 1	0										0											
User 2	1										0											
User 3	0										1											

Berdasarkan **Tabel 4.12**, terdapat 3 user, dimana user 1 tidak terkoneksi sama sekali baik di AP 1 ataupun AP 2. Pada user 2 hanya terkoneksi ke AP 1 saja, sedangkan user 3 hanya terkoneksi ke AP 2 saja.

Jika dimisalkan masing-masing dari user tersebut menempati suatu koordinat tertentu pada ruangan dengan detail user 1 (4,3), user 2 (8,2), dan user 3 (5,8). Maka perhitungan nilai *fitness* dari kasus ini adalah sebagai berikut :

$$pL(d_0) \text{ AP 1} = 10 \log \left[\left(\frac{4\pi d_0 f_j}{3 \times 10^8} \right)^2 \right]$$

$$pL(d_0) \text{ AP 1} = 10 \log \left[\left(\frac{4 \times 3.14 \times 122 \times 2,412 \times 10^9}{3 \times 10^8} \right)^2 \right]$$

$$pL(d_0) \text{ AP 1} = \underline{81,81211038}$$

$$pL(d_0) \text{ AP 2} = 10 \log \left[\left(\frac{4\pi d_0 f_j}{3 \times 10^8} \right)^2 \right]$$

$$pL(d_0) \text{ AP 2} = 10 \log \left[\left(\frac{4 \times 3.14 \times 122 \times 2,412 \times 10^9}{3 \times 10^8} \right)^2 \right]$$

$$pL(d_0) \text{ AP 2} = \underline{81,81211038}$$

$$\begin{aligned} (\text{Jarak AP 1 ke User 1})^2 &= (x_2 - x_1)^2 + (y_2 - y_1)^2 \\ &= (7 - 4)^2 + (0 - 3)^2 \\ &= 18 \end{aligned}$$

$$\text{Jarak AP 1 ke User 1} = \underline{4,24264068}$$

$$\begin{aligned} (\text{Jarak AP 1 ke User 2})^2 &= (x_2 - x_1)^2 + (y_2 - y_1)^2 \\ &= (7 - 8)^2 + (0 - 2)^2 \\ &= 5 \end{aligned}$$

$$\text{Jarak AP 1 ke User 2} = \underline{2,23606797}$$

$$\begin{aligned} (\text{Jarak AP 1 ke User 3})^2 &= (x_2 - x_1)^2 + (y_2 - y_1)^2 \\ &= (7 - 5)^2 + (0 - 8)^2 \end{aligned}$$

$$\begin{aligned} \text{Jarak AP 1 ke User 3} &= 68 \\ &= \underline{8,24621125} \end{aligned}$$

$$\begin{aligned} (\text{Jarak AP 2 ke User 1})^2 &= (x_2 - x_1)^2 + (y_2 - y_1)^2 \\ &= (4 - 4)^2 + (3 - 3)^2 \\ &= 0 \end{aligned}$$

$$\text{Jarak AP 2 ke User 1} = \underline{0}$$

$$\begin{aligned} (\text{Jarak AP 2 ke User 2})^2 &= (x_2 - x_1)^2 + (y_2 - y_1)^2 \\ &= (4 - 8)^2 + (3 - 2)^2 \\ &= 17 \end{aligned}$$

$$\text{Jarak AP 2 ke User 2} = \underline{4,12310563}$$

$$\begin{aligned} (\text{Jarak AP 2 ke User 3})^2 &= (x_2 - x_1)^2 + (y_2 - y_1)^2 \\ &= (4 - 5)^2 + (3 - 8)^2 \\ &= 26 \end{aligned}$$

$$\text{Jarak AP 2 ke User 3} = \underline{5,09901951}$$

$$pL_{11} = 81,81211038 + 10 \times 3,02 \log \left(\frac{4,24264068}{122} \right) + 10$$

$$pL_{11} = \underline{47,75865829}$$

$$pL_{12} = 81,81211038 + 10 \times 3,02 \log \left(\frac{2,23606797}{122} \right) + 10$$

$$pL_{12} = \underline{39,35849051}$$

$$pL_{13} = 81,81211038 + 10 \times 3,02 \log \left(\frac{8,24621125}{122} \right) + 10$$

$$pL_{13} = \underline{56,47492807}$$

$$pL_{21} = 81,81211038 + 10 \times 3,02 \log \left(\frac{0}{122} \right) + 10$$

$$pL_{21} = (\text{undefiend}) \text{ dinilai } 0$$

$$pL_{22} = 81,81211038 + 10 \times 3,02 \log \left(\frac{4,12310563}{122} \right) + 10$$

$$pL_{22} = \underline{47,38382222}$$

$$pL_{23} = 81,81211038 + 10 \times 3,02 \log \left(\frac{5,09901951}{122} \right) + 10$$

$$pL_{23} = \underline{52,28276971}$$

$$\begin{aligned} \text{Rata-rata path loss user 1} &= ((u_{11} \times pL_{11}) + (u_{12} \times pL_{12}))/m \\ &= ((0 \times 47,16447013) + (0 \times 0))/3 \\ &= 0 \end{aligned}$$

$$\begin{aligned} \text{Rata-rata path loss user 2} &= ((u_{21} \times pL_{21}) + (u_{22} \times pL_{22}))/m \\ &= ((1 \times 39,35849051) + (0 \times 47,38382222))/3 \\ &= 13,11949683 \end{aligned}$$

$$\begin{aligned} \text{Rata-rata path loss user 3} &= ((u_{31} \times pL_{31}) + (u_{32} \times pL_{32}))/m \\ &= ((0 \times 56,47492807) + (1 \times 52,28276971))/3 \\ &= 17,42758991 \end{aligned}$$

4.3 Skenario pengujian

Untuk melakukan evaluasi terhadap sistem maka dilakukan 5 subjek pengujian, antara lain adalah:

1. Uji coba untuk mengetahui pengaruh banyaknya generasi pada proses optimasi penempatan AP.
2. Uji coba untuk menentukan ukuran populasi yang optimal untuk proses algoritma genetika terhadap permasalahan optimasi penempatan AP.
3. Uji coba untuk mengetahui pengaruh kombinasi Mr dan Cr untuk permasalahan optimasi penempatan AP.
4. Uji coba pengaruh jumlah AP yang digunakan dalam proses optimasi penempatan AP.

4.3.1 Uji coba banyak generasi

Uji coba akan dilakukan terhadap sistem untuk mengetahui pengaruh banyak generasi dalam proses perhitungan optimasi penempatan AP. Uji coba akan dilakukan sebanyak 5 kali dengan generasi mulai dari 10 sampai 100. Banyaknya populasi yang digunakan adalah 5 populasi dengan Cr = 90% dan Mr = 10%. Untuk tabel uji coba bisa dilihat pada **Tabel 4.13**.

Tabel 4.13 Rancangan Uji Coba Banyaknya Generasi

Banyak Generasi	Nilai <i>Fitness</i>					Rata-rata <i>Fitness</i>
	Percobaan ke- <i>i</i>					
	1	2	3	4	5	
10						
20						
30						
40						
50						
60						
70						
80						
90						
100						

4.3.2 Uji coba ukuran populasi

Uji coba akan dilakukan terhadap sistem untuk mengetahui pengaruh kombinasi Cr dan Mr dalam proses perhitungan optimasi penempatan AP. Uji coba akan dilakukan sebanyak 5 kali dengan menggunakan nilai Cr = 90% dan Mr = 10%. Ukuran populasi yang digunakan adalah kelipatan 50 dimulai dari 50 – 250. Untuk tabel uji coba bisa dilihat pada **Tabel 4.14**

Tabel 4.14 Rancangan Uji Coba Ukuran Populasi

Banyak Populasi	Nilai <i>Fitness</i>					Rata-rata <i>Fitness</i>
	Percobaan ke- <i>i</i>					
	1	2	3	4	5	
50						
100						
150						
200						
250						
300						
350						
400						
450						
500						

4.3.3 Uji coba kombinasi probabilitas *crossover* dan mutasi

Uji coba akan dilakukan terhadap sistem untuk mengetahui pengaruh kombinasi Cr dan Mr dalam proses perhitungan optimasi penempatan AP. Uji coba dilakukan sebanyak 5 kali dan digunakan kombinasi Cr = 100% dan Mr = 0%, sampai Cr = 0% dan Mr = 100%. Untuk tabel uji coba bisa dilihat pada **Tabel 4.14**.

Tabel 4.15 Rancangan Uji Coba *Crossover Rate* dan *Mutastion Rate*

Kombinasi		Nilai <i>Fitness</i>					Rata-rata <i>Fitness</i>
		Percobaan ke- <i>i</i>					
Cr	Mr	1	2	3	4	5	
0,9	0,1						
0,8	0,2						
0,7	0,3						
0,6	0,4						
0,5	0,5						
0,4	0,6						
0,3	0,7						
0,2	0,8						
0,1	0,9						

4.3.4 Uji coba jumlah *access point*

Uji coba akan dilakukan terhadap sistem untuk mengetahui pengaruh jumlah AP yang digunakan dalam suatu ruangan terhadap proses perhitungan optimasi penempatan AP. Uji coba akan dilakukan sebanyak 5 kali dengan jumlah AP yang digunakan 1 sampai 5. Untuk tabel uji coba bisa dilihat pada **Tabel 4.15**.

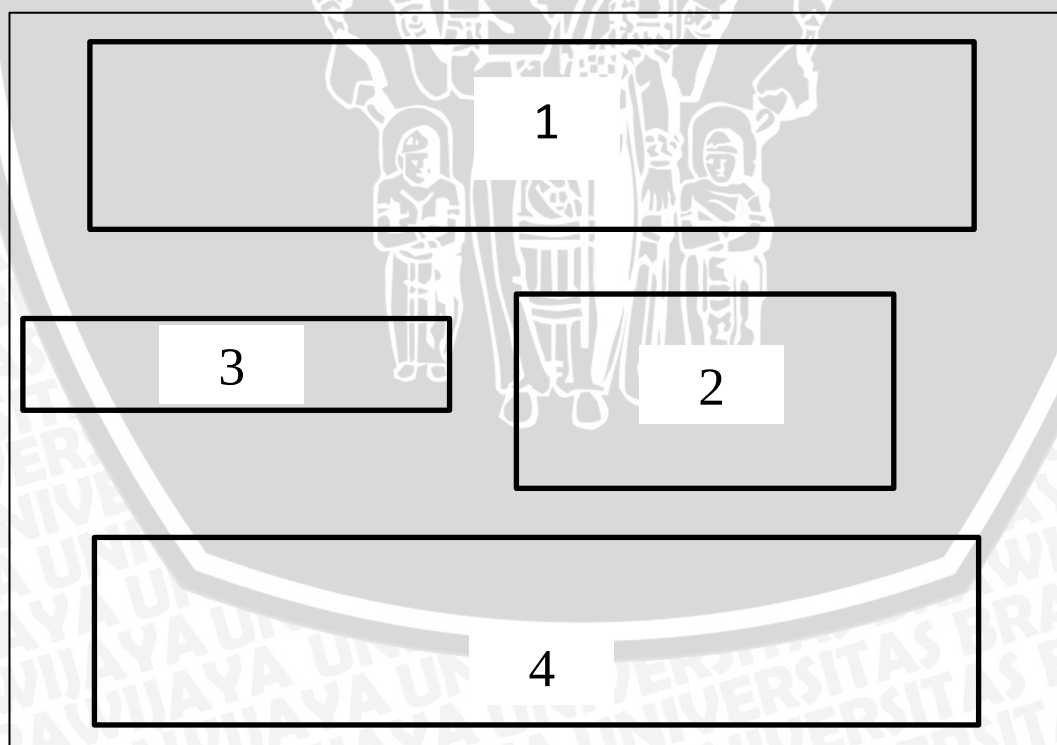
Tabel 4.16 Rancangan Uji Coba Jumlah *Access Point* (AP)

Jumlah AP	Nilai <i>Fitness</i>					Rata-rata <i>Fitness</i>
	Percobaan ke- <i>i</i>					
	1	2	3	4	5	
1						
2						
3						
4						
5						

4.4 Perancangan *user interface*

Sistem optimasi penempatan AP berdasarkan nilai minimum *path loss* menggunakan algoritma genetika memiliki 3 halaman utama yaitu, halaman *home*, halaman *input*, dan juga halaman hasil.

Perancangan antarmuka pada halaman *home* dapat dilihat pada **Gambar 4.9**.



Gambar 4.9 Perancangan antarmuka halaman *home*

Penjelasan dari setiap komponen rancangan halaman *home* sesuai dengan label angka yang diberikan pada **Gambar 4.9**, yakni sebagai berikut :

1. Judul halaman.
2. Logo PTIIK.
3. Pilihan ruangan yang dapat dihitung.
4. Deskripsi umum sistem.

Halaman *Input* data berfungsi untuk menginput data – data yang dibutuhkan oleh sistem. *Input* data berisikan *inputan* untuk data, dan juga tombol *submit*. Perancangan halaman *input* data akan ditunjukkan pada **Gambar 4.10**.

The diagram shows a rectangular frame representing the input page layout. Inside the frame, there are five numbered labels pointing to specific components:

- Label 1 points to a rectangular box at the top left.
- Label 2 points to a rectangular box below label 1.
- Label 3 points to a rectangular box inside a larger container, below label 2.
- Label 4 points to another rectangular box inside the same container as label 3, below label 3.
- Label 5 points to a small rectangular box at the bottom left of the main container.

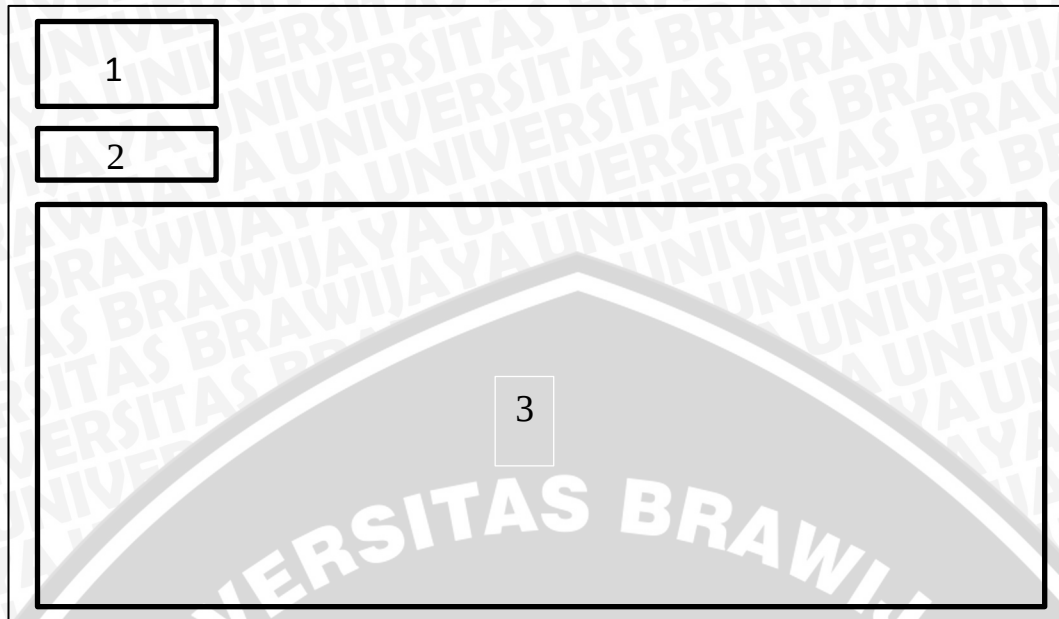
 The background of the frame features a faint watermark of the Brawijaya University logo and name.

Gambar 4.10 Perancangan halaman *input*

Penjelasan dari setiap komponen rancangan halaman *input* sesuai dengan label angka yang diberikan pada **Gambar 4.10**, yakni sebagai berikut :

1. Objek ruangan.
2. Instruksi halaman.
3. Form *inputan* jumlah AP.
4. Form *inputan* jumlah populasi.
5. Tombol *Submit*.

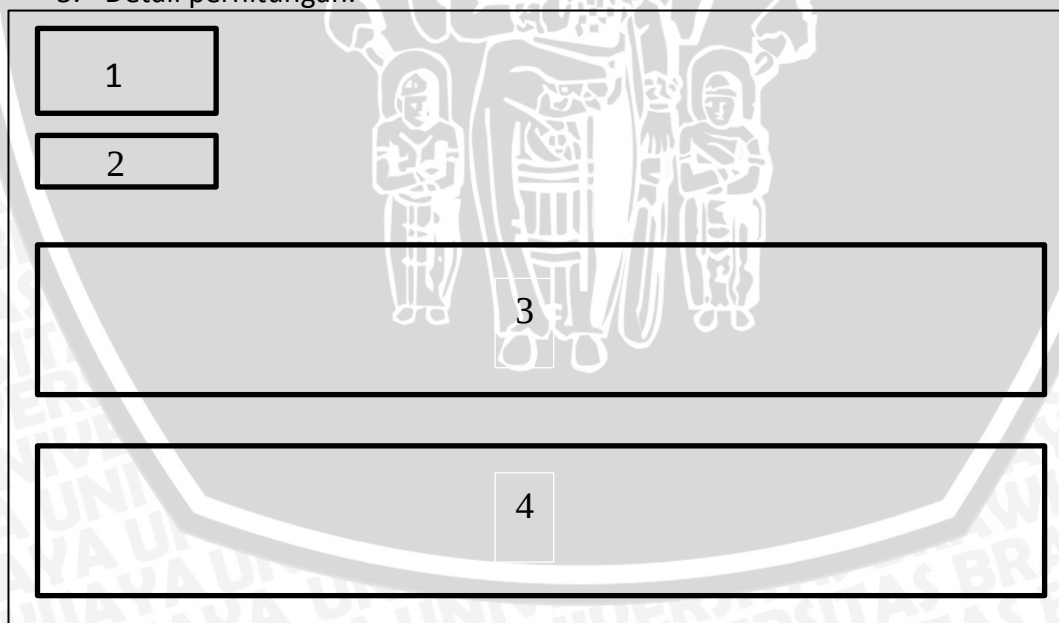
Halaman hasil terdiri dari 2 macam, yaitu halaman hasil dengan detail perhitungan, dan juga halaman hasil perhitungan saja. Perancangan halaman hasil ditunjukkan pada **Gambar 4.11** dan **4.12**.



Gambar 4.11 Perancangan halaman detail perhitungan

Penjelasan dari setiap komponen rancangan halaman *input* sesuai dengan label angka yang diberikan pada **Gambar 4.11**, yakni sebagai berikut :

1. Berisi tentang informasi nama objek ruangan, dimensi ruangan, jumlah AP yang digunakan, dan jumlah populasi yang digunakan.
2. Menunjukkan iterasi ke sekian.
3. Detail perhitungan.



Gambar 4.12 Perancangan halaman hasil perhitungan

Penjelasan dari setiap komponen rancangan halaman *input* sesuai dengan label angka yang diberikan pada **Gambar 4.12**, yakni sebagai berikut :

1. Berisi tentang informasi nama objek ruangan, dimensi ruangan, jumlah AP yang digunakan, dan jumlah populasi yang digunakan.
2. Menunjukkan iterasi ke sekian.
3. Hasil perhitungan AP ke 1.
4. Hasil perhitungan AP ke n (sesuai jumlah AP yang di *inputkan*)



BAB 5 IMPLEMENTASI

Pada bab ini membahas tentang implementasi perangkat lunak berdasarkan hasil yang telah diperoleh dari analisis kebutuhan dan proses perancangan yang telah dibuat. Pembahasan yang akan di jelaskan terdiri dari spesifikasi sistem, batasan – batasan implementasi, implementasi algoritma pada program, implementasi antarmuka, dan implementasi metode.

5.1 Spesifikasi sitem

Hasil analisis kebutuhan dan perancangan perangkat lunak yang telah di bahas pada Bab 4 akan menjadi acuan untuk melakukan implementasi menjadi sistem yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi perangkat keras

Pengembangan sistem dilakukan pada komputer dengan spesifikasi perangkat keras yang akan dijelaskan pada **Tabel 5.1**.

Tabel 5.1 Spesifikasi perangkat keras

Nama Komponen	Spesifikasi
Prosesor	Intel® Core™ i5-4200U CPU@2.50GHz
Memori(RAM)	4000 MB
Harddisk	1000 GB

5.1.2 Spesifikasi perangkat lunak

Spesifikasi perangkat lunak yang digunakan untuk mengembangkan sistem ini, dijelaskan pada **Tabel 5.2**.

Tabel 5.2 Spesifikasi perangkat lunak

Nama	Spesifikasi
Sistem Operasi	Microsoft Windows 7 64-bit
Bahasa Pemrograman	PHP 5.4.4
Tools Pemrograman	PHP designer 8
Server Localhost	XAMPP Version 1.8.0
DBMS	MySQL
Tools DBMS	MySQL Version 5.0.10

5.2 Batasan - batasan implementasi

Batasan Implementasi merupakan batasan proses yang bisa dilakukan oleh sistem sesuai dengan perancangan awal sistem. Batasan – batasan implementasi ini ditampilkan agar penelitian ini memiliki ruang lingkup yang jelas dan tidak

melenceng dalam mengimplementasikan sistem. Beberapa batasan yang ada dalam sistem ini sebagai berikut :

- Sistem Optimasi ini dirancang dan dijalankan dengan menggunakan Web Application.
- Metode yang dipakai dalam menyelesaikan masalah yang digunakan adalah menggunakan metode Algoritma Genetika.
- *Inputan* yang diterima berupa *inputan* AP yang digunakan, dan juga jumlah populasi.
- Output yang akan ditampilkan adalah laporan perhitungan dari setiap kromosom dari populasi, sebanyak iterasi yang di inisialisasikan dalam sistem.
- Spesifikasi AP yang digunakan di sesuaikan dengan AP yang saat ini digunakan pada gedung D dan gazebo PTTIK UB.
- Ruang yang digunakan dalam penelitian ini adalah ruangan robotik, ruangan hinpunan, dan gazebo PTIIK.

5.2 Implementasi algoritma pada program

Sistem ini mempunyai beberapa proses, yaitu proses *input* data dan juga proses perhitungan algoritma genetika.

5.2.1 Implementasi proses *input* data

Proses *input* data merupakan proses awal dari sistem. Proses ini dilakukan untuk memasukkan jumlah AP dan jumlah populasi yang nantinya akan diproses oleh sistem. **Gambar 5.1** merupakan implementasi algoritma proses *input*.

```

1 <div class="content-wrapper">
2   <section class="content-header">
3     <h1>
4       Inputkan
5       <small>Jumlah Akses Point dan Jumlah Populasi dalam
ruangan ini</small>
6     </h1>
7   </section>

8 <!-- Main content -->
9 <section class="content">
10  <div class="row">
11    <div class="col-md-6">
12      <div class="box box-danger">
13        <form action="_proses.php" method="POST">
14          <div class="box-body">
15            <div class="checkbox">
16              <label>
17                <input type="checkbox" name="parent"> Menggunakan
parent yang sudah di tentukan
18              </label>
19            </div>
20          <div class="form-group">

```

```

21 <label>Jumlah Access Point</label>
    <input type="text" class="form-control" name="jml_ap"
22 placeholder="Masukan Jumlah ACESS Point"></input>
23 </div>
24 <div class="form-group">
25 <label>Jumlah Populasi</label>
    <input type="text" class="form-control"
26 name="populasi_awal" placeholder="Masukan Jumlah Populasi
    Awal"></input>
27 </div>
28 <div class="checkbox">
29 <label>
    <input type="checkbox" name="tampil"> Tampilkan
30 perhitungan
31 </label>
32 </div>
33 </div>
34 <input type="hidden" name="p" value="<?php echo $p?>">
35 <input type="hidden" name="l" value="<?php echo $l?>">
    <input type="hidden" name="id_area" value="<?php echo
36 $id_area?>">
37 <div class="box-footer">
    <button type="submit" class="btn btn-
38 primary">Submit</button>
39 </div>
40 </form>
41 </div>
42 </div>
43 </div>
44 </section>
    </div>

```

Gambar 5.1 Implementasi proses *input* data

Berikut merupakan penjelasan **Gambar 5.1** :

- Baris 3 - 6 merupakan kode program yang digunakan untuk menampilkan tulisan *inputkan* jumlah akses *point* dan jumlah populasi dalam ruangan ini.
- Baris 14 - 18 merupakan kode program untuk menampilkan *checkbox* fitur penentuan parent.
- Baris 19 - 22 merupakan kode program untuk membuat form *inputan* jumlah AP.
- Baris 23 - 26 merupakan kode program untuk membuat form *inputan* jumlah populasi.
- Baris 27 - 31 merupakan kode program untuk menampilkan *checkbox* dari fitur tampilkan perhitungan.
- Baris 33 - 35 merupakan kode program untuk mendapatkan *inputan* dari program berupa panjang ruangan, lebar ruangan, dan juga id ruangan.
- Baris 37 merupakan kode program untuk membuat tombol *Submit*.

5.2.2 Implementasi perhitungan algoritma genetika

Proses perhitungan algoritma genetika merupakan garis besar jalannya program sistem optimasi ini. Proses algoritma genetika terdiri dari proses representasi kromosom, *generate* populasi awal, proses *crossover*, dan proses mutasi.

5.2.2.1 Implementasi representasi kromosom

Proses perhitungan representasi kromosom diawali dengan proses *grid point*. Proses ini terdapat pada fungsi *koor*. Fungsi ini melakukan proses pemberian id terhadap setiap titik yang terdapat pada ruangan. Titik ini terdapat pada setiap jarak 1m pada ruangan. Selanjutnya adalah fungsi merubah bilangan biner ke desimal agar dapat diterapkan pada perhitungan sistem, fungsi ini diterapkan pada setiap segmen gen pada kromosom, yakni gen lokasi AP, frekuensi, dan juga *power level* AP. Implementasi fungsi dapat dilihat pada **Gambar 5.2**.

```

1  function koor ($p='', $l='')
2  {
3      /*
4      *KOORDINAT
5      *NB: P=x & L=y
6      */
7      for($x=0 ; $x<=$l ; $x++){
8          for($y=0 ; $y<=$p ; $y++){
9              if($x % 2 == 1){
10                 for($z=$p ; $z >= 0 ; $z--){
11                     $temp_y[] = $z;
12                 }
13                 $koor[] = array('x' => $temp_y[$y],
14                                'y' => $x
15                                );
16             } else {
17                 $koor[] = array('x' => $y,
18                                'y' => $x
19                                );
20             }
21         }
22     }
23
24     return $koor;
25 }
26
27 function frek ()
28 {
29     /*
30     *FREKUENSI
31     */
32     $frek = array(
33         array('biner' => '00', 'frek' => '2.412'),
34         array('biner' => '01', 'frek' => '2.437'),

```

```

35         array('biner' => '10', 'frek' => '2.462'),
36         array('biner' => '11', 'frek' => '2.462')
37     );
38
39     return $frek;
40 }
41
42 function dt_frekuensi ()
43 {
44     $ret = array(
45         '00' => '2.412',
46         '01' => '2.437',
47         '10' => '2.462',
48         '11' => '2.462'
49     );
50     return $ret;
51 }
52
53 function pwr ()
54 {
55     /*
56
57     *POWER
58     */
59     $pwr = array(
60         array('biner' => '00', 'pwr' => '24'),
61         array('biner' => '01', 'pwr' => '20'),
62         array('biner' => '10', 'pwr' => '15'),
63         array('biner' => '11', 'pwr' => '15')
64     );
65
66     return $pwr;
67 }
68
69 function dt_pwr ()
70 {
71     $ret = array(
72         '00' => '24',
73         '01' => '20',
74         '10' => '15',
75         '11' => '15'
76     );
77     return $ret;
78 }

```

Gambar 5.2 Implementasi representasi kromosom

Berikut merupakan penjelasan **Gambar 5.2** :

- Baris 1 - 24 merupakan kode program yang digunakan untuk mencari titik koordinat AP pada ruangan.

- Baris 26 - 39 merupakan kode program untuk inisialisasi array frek, yang berisi angka biner beserta *value* yang dimiliki angka biner tersebut.
- Baris 41 - 50 merupakan kode program yang digunakan untuk menerjemahkan nilai desimal ke *value* frekuensi yang digunakan dalam sistem.
- Baris 52 - 66 merupakan kode program untuk inisialisasi array pwr, yang berisi angka biner beserta *value* yang dimiliki angka biner tersebut.
- Baris 68 - 77 merupakan kode program yang digunakan untuk menerjemahkan nilai desimal ke *value power* yang digunakan dalam sistem.

5.2.2.2 Implementasi proses *crossover*

Proses *crossover* yang digunakan dalam algoritma ini diawali dengan pemilihan dua parent secara acak dari 70% populasi dengan nilai *fitness* terbaik. Langkah selanjutnya adalah dengan menentukan titik pertukaran antar 2 *parent*, jumlah titik yang digunakan disesuaikan dengan banyaknya jumlah AP.

```

1 function hitung_crossover ($source_crossover,
2 $jml_populasi, $jml_ap, $p, $l, $dt_user, $parent=FALSE)
3 {
4     $ret='';
5     $frek = $this->dt_frekuensi();
6     $pwr = $this->dt_pwr();
7     $koor = $this->koor($p,$l);
8     $count_koor = count($koor);
9     $jml_CO = round((90 * $jml_populasi) / 100);
10
11     $array_CO = array(
12         array('r1' => '1', 'r2' => '0', 'biner ditukar' =>
13             array('10', '5')), array('r1' => '2', 'r2' => '0', 'biner
14             ditukar' => array('6', '3')), array('r1' => '3', 'r2' =>
15             '2', 'biner ditukar' => array('1', '9')), array('r1' =>
16             '1', 'r2' => '3', 'biner ditukar' => array('6', '1')),
17             array('r1' => '3', 'r2' => '3', 'biner ditukar' =>
18                 array('6', '10')),
19             );
20     for($CO = 0 ; $CO < $jml_CO ; $CO++){
21         //inisialisai recursive +=
22         foreach($dt_user as $k => $v){
23             $jml_dij_CO[$CO][$k] = 0;
24         }
25         $r1 = $parent ? $array_CO[$CO]['r1'] : rand(0,
26             (count($source_crossover) - 1));
27         $r2 = $parent ? $array_CO[$CO]['r2'] : rand(0,
28             (count($source_crossover) - 1));
29
30         for($AP = 0 ; $AP < $jml_ap ; $AP++){
31             $biner_join[$CO]['biner_1'][$AP] = str_split($this->
32                 add_0(decbin($source_crossover[$r1]['detail'][$AP]

```

```

32 ey_lokasi')) .
33 $source_crossover[$r1]['detail']['AP'].$AP]['key_frekuensi']
34 . $source_crossover[$r1]['detail']['AP
35 '.$AP]['key_power']);
36 $biner_join[$CO]['biner_2']['AP '.$AP] = str_split($this-
37 >add_0
38 (decbin($source_crossover[$r2]['detail']['AP'].$AP]['key_lok
39 asi'])) .
40 $source_crossover[$r2]['detail']['AP'].$AP]['key_frekuensi']
41 . $source_crossover[$r2]['detail']['AP
42 '.$AP]['key_power']);
43 //tukarkan biner random ke-
44 $rand_tukar_biner = $parent ? $array_CO[$CO]['biner
45 ditukar'][$AP] : rand(0,10);
46 $tukarkan_dngn_ini = $biner_join[$CO]['biner_2']['AP
47 '.$AP][$rand_tukar_biner];
48
49 $new_biner = '';
50 foreach($biner_join[$CO]['biner_1']['AP '.$AP] as $k =>
51 $v){
52     if($k == $rand_tukar_biner){
53         $new_biner .= $tukarkan_dngn_ini;
54     } else {
55         $new_biner .= $v;
56     }
57 }
58 $new_biner_CO = $new_biner;
59
60 $key_frekuensi = substr($new_biner_CO, 7, 2);
61 $key_pwr = substr($new_biner_CO, 9, 2);
62 $lokasi_baru = bindec(substr($new_biner_CO, 0, 7));
63 $frekuensi_baru = $frekuensi[$key_frekuensi];
64 $pwr_baru = $pwr[$key_pwr];
65
66 //ubah lokasi jika melebihi kombinasi koordinat
67 if($lokasi_baru > ($count_koor-1)){
68     $modulus = $lokasi_baru % $count_koor;
69     $lokasi_baru = $modulus;
70 }
71
72 $ret[$CO]['AP '.$AP]['biner yang ditukar'] =
73 $rand_tukar_biner;
74 $ret[$CO]['AP '.$AP]['biner'] = $new_biner_CO;
75 $ret[$CO]['AP '.$AP]['key_lokasi'] = $lokasi_baru;
76 $ret[$CO]['AP '.$AP]['lokasi'] =
77 $skoor[$lokasi_baru]['x'].','.$skoor[$lokasi_baru]['y'];
78 $ret[$CO]['AP '.$AP]['key_frekuensi'] = $key_frekuensi;
79 $ret[$CO]['AP '.$AP]['frekuensi'] = $frekuensi_baru;
80 $ret[$CO]['AP '.$AP]['key_power'] = $key_pwr;
81 $ret[$CO]['AP '.$AP]['power'] = $pwr_baru;
82 $ret[$CO]['AP '.$AP]['path_d0'] = 10 * log10(pow((4 * 3.14

```

```

83 * 122 * $frek_baru * pow(10,9)) / (3 * pow(10,8)), 2));
84
85 foreach($dt_user as $k => $v){
86 $ret[$CO]['AP '.$AP]['ap_user']['user
87 '.$v['x'].'.', '.$v['y']] = sqrt(pow(($v['x'] -
88 $kooor[$lokasi_baru]['x']),2) + pow(($v['y'] -
89 $kooor[$lokasi_baru]['y']), 2));
90
91 $ret[$CO]['AP '.$AP]['path_dij']['user
92 '.$v['x'].'.', '.$v['y']] = $kooor[$lokasi_baru]['x'] ==
93 $v['x'] && $kooor[$lokasi_baru]['y'] == $v['y'] ? '0' :
94 $ret[$CO]['AP '.$AP]['path_d0'] + 10 * 3.02 *
95 log10($ret[$CO]['AP '.$AP]['ap_user']['user
96 '.$v['x'].'.', '.$v['y']] / 122) + 10;
97 //ini recursive harus inisialisai dulu di atas
98 $jml_dij_CO[$CO][$k] += $ret[$CO]['AP
99 '.$AP]['path_dij']['user '.$v['x'].'.', '.$v['y']]};
100 }
101 foreach($dt_user as $k => $v){
102 $ret[$CO]['rata-rata']['user '.$v['x'].'.', '.$v['y']] =
103 $jml_dij_CO[$CO][$k] == '0' ? '0' : 1 /
104 ($jml_dij_CO[$CO][$k] / count($dt_user));
105 }
106 $ret[$CO]['max_val']['val'] = max($ret[$CO]['rata-rata']);
107 $ret[$CO]['max_val']['detail']['identity'] = 'CROSSOVER';
108 $ret[$CO]['max_val']['detail']['user'] =
109 array_search($ret[$CO]['max_val']['val'], $ret[$CO]['rata-
110 rata']);
111
112 for($AP = 0 ; $AP < $jml_ap ; $AP++){
113
114 $ret[$CO]['max_val']['detail']['AP '.$AP]['PARENT 1'] =
115 $this->add_0(decbin($source_crossover[$r1]['detail']['AP
116 '.$AP]['key_lokasi'])) .
117 $source_crossover[$r1]['detail']['AP
118 '.$AP]['key_frekuensi'] .
119 $source_crossover[$r1]['detail']['AP '.$AP]['key_power'];
120 $ret[$CO]['max_val']['detail']['AP '.$AP]['PARENT 2'] =
121 $this->add_0(decbin($source_crossover[$r2]['detail']['AP
122 '.$AP]['key_lokasi'])) .
123 $source_crossover[$r2]['detail']['AP
124 '.$AP]['key_frekuensi'] .
125 $source_crossover[$r2]['detail']['AP '.$AP]['key_power'];
126 $ret[$CO]['max_val']['detail']['AP '.$AP]['biner'] = $this-
127 >add_0(decbin($ret[$CO]['AP '.$AP]['key_lokasi'])) .
128 $ret[$CO]['AP '.$AP]['key_frekuensi'] . $ret[$CO]['AP
129 '.$AP]['key_power'];
130 $ret[$CO]['max_val']['detail']['AP '.$AP]['biner yang
131 ditukar'] = $ret[$CO]['AP '.$AP]['biner yang ditukar'];

```

```

124 $ret[$CO]['max_val']['detail']['AP '.$AP]['key_lokasi'] =
125 $ret[$CO]['AP '.$AP]['key_lokasi'];
126 $ret[$CO]['max_val']['detail']['AP '.$AP]['lokasi'] =
127 $ret[$CO]['AP '.$AP]['lokasi'];
128 $ret[$CO]['max_val']['detail']['AP '.$AP]['key_frekuensi']
129 = $ret[$CO]['AP '.$AP]['key_frekuensi'];
130 $ret[$CO]['max_val']['detail']['AP '.$AP]['frekuensi'] =
131 $ret[$CO]['AP '.$AP]['frekuensi'];
132 $ret[$CO]['max_val']['detail']['AP '.$AP]['key_power'] =
133 $ret[$CO]['AP '.$AP]['key_power'];
134 $ret[$CO]['max_val']['detail']['AP '.$AP]['power'] =
135 $ret[$CO]['AP '.$AP]['power'];
136 }
137 }
138 return $ret;
139 }
140 }

```

Gambar 5.3 Implementasi proses *crossover*

- Baris 10 - 17 merupakan kode program yang digunakan untuk menentukan parent dan juga titik potong dari kromosom parent.
- Baris 24 - 27 merupakan kode program untuk memilih parent secara acak dari *source crossover*.
- Baris 29 - 42 merupakan kode program yang digunakan menyusun kromosom yang akan melakukan proses *crossover*.
- Baris 44 - 64 merupakan kode program untuk proses *crossover*.
- Baris 67 - 70 merupakan kode program untuk mengatasi jika lokasi AP berada di luar *grid point*.
- Baris 82 - 83 merupakan kode program untuk menghitung *path loss* d0.
- Baris 85 - 89 merupakan kode program untuk menghitung jarak AP ke setiap *user*.
- Baris 91 - 94 merupakan kode program untuk menghitung *path loss* dij.
- Baris 100 - 103 merupakan kode program untuk menghitung nilai *fitness*.

5.2.2.3 Implementasi proses mutasi

Proses mutasi merupakan proses perubahan nilai gen dari suatu kromosom. Parent dari proses mutasi ini dipilih dari 10% kromosom dengan nilai terbaik pada populasi. Berikut merupakan implementasi fungsi mutasi dapat dilihat pada **Gambar 5.3**.

```

1 function hitung_mutasi($source_mutasi, $jml_ap, $p, $l,
2 $dt_user, $parent=FALSE)
3 {
4 $ret = '';
5 $frek = $this->dt_frekuensi();
6 $pwr = $this->dt_pwr();
7 $koor = $this->koor($p,$l);
8 $count_koor = count($koor);
9
10 $array_M = array(
11 array(6, 10)
12 );
13
14 foreach($source_mutasi as $k => $v){
15 //inisialisai recursive +=
16 foreach($dt_user as $kk => $vv){
17 $jml_dij_M[$k][$kk] = 0;
18 }
19
20 for($AP = 0 ; $AP < $jml_ap ; $AP++){
21 //tukarkan 0->1 1->0 biner random ke-
22 $r_mutasi = $parent ? $array_M[$k][$AP] : rand(0,10);
23
24 $biner[$k]['AP '.$AP]['yang di mutasi'] = $r_mutasi;
25
26 $biner[$k]['AP '.$AP]['biner'] = str_split($this->
27 add_0(decbin($source_mutasi[$k]['detail']['AP
28 '.$AP]['key_lokasi'])). $source_mutasi[$k]['detail']['AP
29 '.$AP]['key_frekuensi'] . $source_mutasi[$k]['detail']['AP
30 '.$AP]['key_power']);
31
32 $new_biner = '';
33 foreach($biner[$k]['AP '.$AP]['biner'] as $kk => $vv){
34 if($kk == $r_mutasi){
35 if($vv == '0'){
36 $new_biner .= '1';
37 } elseif($vv == '1'){
38 $new_biner .= '0';
39 }
40 } else {
41 $new_biner .= $vv;
42 }
43 }
44
45 $biner[$k]['AP '.$AP]['new_biner'] = $new_biner;
46
47 $key_frekuensi = substr($new_biner, 7, 2);
48 $key_pwr = substr($new_biner, 9, 2);

```

```

41 $lokasi_baru = bindec(substr($new_biner, 0, 7));
42 $frek_baru = $frek[$key_fre];
43 $pwr_baru = $pwr[$key_pwr];

44 //ubah lokasi jika melebihi kombinasi koordinat
45 if($lokasi_baru > ($count_koor-1)){

46     $modulus = $lokasi_baru % $count_koor;

47     $lokasi_baru = $modulus;

48 }
49 $ret[$k]['AP '.$AP]['awal biner'] = implode($biner[$k]['AP
50 '.$AP]['biner']);
51 $ret[$k]['AP '.$AP]['array yang di mutasi'] = $r_mutasi;
52 $ret[$k]['AP '.$AP]['biner'] = $new_biner;
53 $ret[$k]['AP '.$AP]['key_lokasi'] = $lokasi_baru;
54 $ret[$k]['AP '.$AP]['lokasi'] =
55 $koor[$lokasi_baru]['x'].'.'.$koor[$lokasi_baru]['y'];
56 $ret[$k]['AP '.$AP]['key_frekuensi'] = $key_fre;
57 $ret[$k]['AP '.$AP]['frekuensi'] = $frek_baru;
58 $ret[$k]['AP '.$AP]['key_power'] = $key_pwr;
59 $ret[$k]['AP '.$AP]['power'] = $pwr_baru;
60 $ret[$k]['AP '.$AP]['path_d0'] = 10 * log10(pow((4 * 3.14 *
61 122 * $frek_baru * pow(10,9)) / (3 * pow(10,8)), 2));

62 foreach($dt_user as $kk => $vv){

63     $ret[$k]['AP '.$AP]['ap_user']['user
64     '.$vv['x'].'.'.$vv['y']] = sqrt(pow(($vv['x'] -
65     $koor[$lokasi_baru]['x']),2) + pow(($vv['y'] -
66     $koor[$lokasi_baru]['y']), 2));

67     $ret[$k]['AP '.$AP]['path_dij']['user
68     '.$vv['x'].'.'.$vv['y']] = $koor[$lokasi_baru]['x'] ==
69     $vv['x'] && $koor[$lokasi_baru]['y'] == $vv['y'] ? '0' :
70     $ret[$k]['AP '.$AP]['path_d0'] + 10 * 3.02 *
71     log10($ret[$k]['AP '.$AP]['ap_user']['user
72     '.$vv['x'].'.'.$vv['y']] / 122) + 10;

73     //ini recursive harus inisialisai dulu di atas
74     $jml_dij_M[$k][$kk] += $ret[$k]['AP '.$AP]['path_dij']['user
75     '.$vv['x'].'.'.$vv['y']];

76 }

77 }

78 foreach($dt_user as $kk => $vv){

79     $ret[$k]['rata-rata']['user '.$vv['x'].'.'.$vv['y']] =

```

```

80 $jml_dij_M[$k][$kk] == '0' ? '0' : 1 / ($jml_dij_M[$k][$kk]
81 / count($dt_user));
82 }
83 $ret[$k]['max_val']['val'] = max($ret[$k]['rata-rata']);
84 $ret[$k]['max_val']['detail']['identity'] = 'MUTASI';
85 $ret[$k]['max_val']['detail']['user'] =
86 array_search($ret[$k]['max_val']['val'], $ret[$k]['rata-
87 rata']);
88 $ret[$k]['max_val']['detail']['PARENT'] =
89 $source_mutasi[$k]['val'];
90 for($AP = 0 ; $AP < $jml_ap ; $AP++){
91 $ret[$k]['max_val']['detail']['AP '.$AP]['awal biner'] =
92 $ret[$k]['AP '.$AP]['awal biner'];
93 $ret[$k]['max_val']['detail']['AP '.$AP]['array yang di
94 mutasi'] = $ret[$k]['AP '.$AP]['array yang di mutasi'];
95 $ret[$k]['max_val']['detail']['AP '.$AP]['biner'] =
96 $ret[$k]['AP '.$AP]['biner'];
97 $ret[$k]['max_val']['detail']['AP '.$AP]['key_lokasi'] =
98 $ret[$k]['AP '.$AP]['key_lokasi'];
99 $ret[$k]['max_val']['detail']['AP '.$AP]['lokasi'] =
100 $ret[$k]['AP '.$AP]['lokasi'];
101 $ret[$k]['max_val']['detail']['AP '.$AP]['key_frekuensi'] =
102 $ret[$k]['AP '.$AP]['key_frekuensi'];
103 $ret[$k]['max_val']['detail']['AP '.$AP]['frekuensi'] =
104 $ret[$k]['AP '.$AP]['frekuensi'];
105 $ret[$k]['max_val']['detail']['AP '.$AP]['key_power'] =
106 $ret[$k]['AP '.$AP]['key_power'];
107 $ret[$k]['max_val']['detail']['AP '.$AP]['power'] =
108 $ret[$k]['AP '.$AP]['power'];
109 }
110 }
111 return $ret;
112 }

```

Gambar 5.4 Implementasi proses mutasi

- Baris 21 - 37 merupakan kode program proses mutasi.
- Baris 45 - 48 merupakan kode program untuk mengatasi jika lokasi AP berada di luar *grid point*.
- Baris 60 - 61 kode program untuk menghitung *path loss* d0.
- Baris 62 - 66 merupakan kode program untuk menghitung jarak AP ke setiap *user*.

- Baris 67 - 72 merupakan kode program *path loss* dij.
- Baris 77 - 82 merupakan program untuk menghitung nilai *fitness*.

5.3 Implementasi antarmuka

Contoh struktur skripsi untuk implementatif pembangunan dan non-implementatif eksperimental dapat dilihat pada kedua seksi berikut.

5.3.1 Antarmuka halaman *home*

Halaman *home* merupakan halaman utama pada sistem ini. Di dalam halaman *home* terdapat deskripsi umum tentang sistem. **Gambar 5.5** merupakan implementasi antarmuka halaman *home* yang mengimplementasikan perancangan antarmuka halaman *home* pada Sub Bab 4.3.1.



Gambar 5.5 Antarmuka halaman *home*

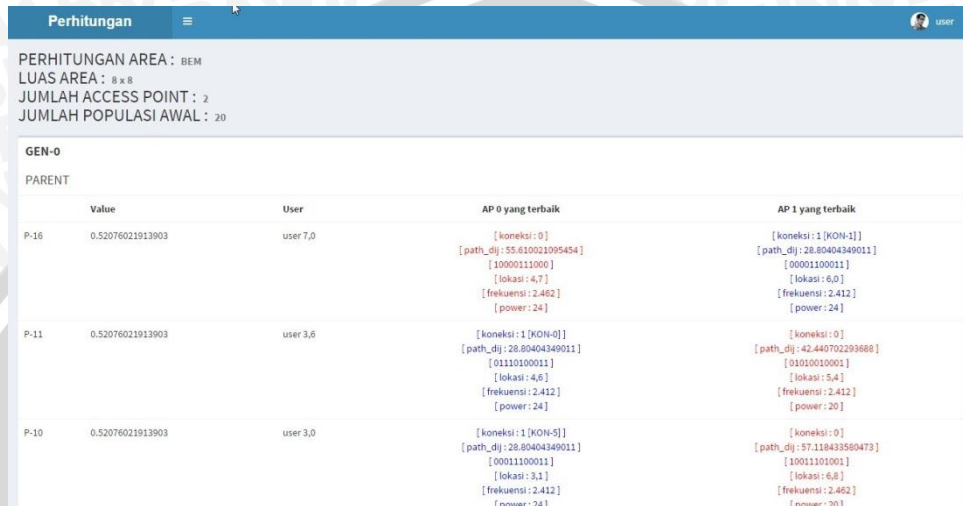
5.3.2 Implementasi antarmuka halaman *input*

Pada halaman *input* pengguna dapat me data berupa jumlah AP, dan juga jumlah populasi. Selain itu pengguna juga dapat memilih fitur seperti tampilkan perhitungan ataupun menggunakan *inputan* yang telah ditentukan. **Gambar 5.6** merupakan implementasi halaman *home*.

Gambar 5.6 Implementai halaman *input*

5.3.3 Implementasi antarmuka halaman hasil

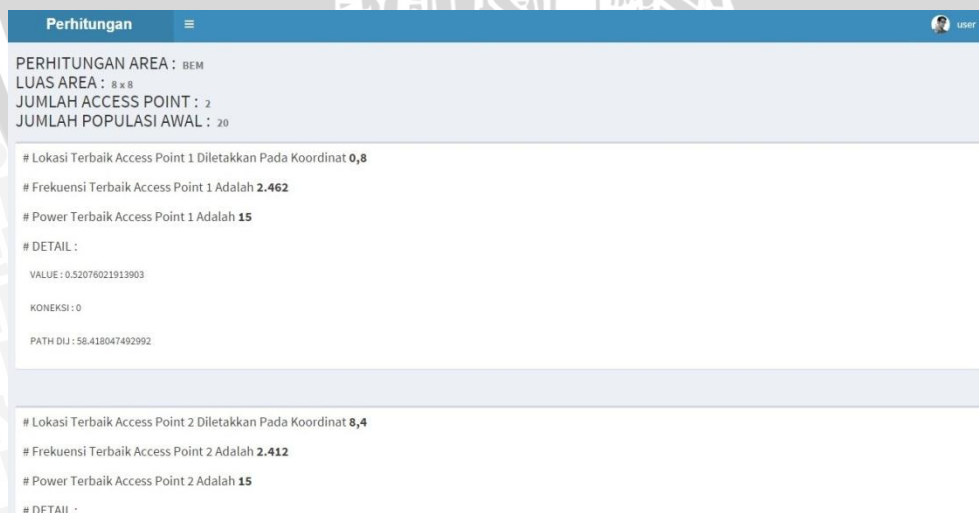
Pada halaman hasil terdapat dua variasi yang dapat dilihat oleh pengguna, yakni versi dengan menunjukkan perhitungan atau versi yang langsung menampilkan nilai terbaik dari perhitungan. **Gambar 5.7** merupakan implementasi halaman hasil dengan detail perhitungan. Dan **Gambar 5.8** merupakan implementasi halaman hasil yang langsung menampilkan nilai optimal dari proses perhitungan.



PARENT	Value	User	AP 0 yang terbaik	AP 1 yang terbaik
P-16	0.52076021913903	user 7,0	[koneksi: 0] [path_dij: 55.610021095454] [10000111000] [lokasi: 4,7] [frekuensi: 2.462] [power: 24]	[koneksi: 1 [KON-1]] [path_dij: 28.80404340011] [00011000011] [lokasi: 6,0] [frekuensi: 2.412] [power: 24]
P-11	0.52076021913903	user 3,6	[koneksi: 1 [KON-0]] [path_dij: 28.80404340011] [01110100011] [lokasi: 4,8] [frekuensi: 2.412] [power: 24]	[koneksi: 0] [path_dij: 42.440702293668] [01010010001] [lokasi: 5,4] [frekuensi: 2.412] [power: 20]
P-10	0.52076021913903	user 3,0	[koneksi: 1 [KON-5]] [path_dij: 28.80404340011] [00011000011] [lokasi: 3,1] [frekuensi: 2.412] [power: 24]	[koneksi: 0] [path_dij: 57.118433580473] [10011010001] [lokasi: 6,8] [frekuensi: 2.462] [power: 20]

Gambar 5.7 Implementasi halaman hasil detail perhitungan

Pada halaman tersebut akan ditampilkan hasil perhitungan dari setiap AP yang digunakan. Detail perhitungan pada halaman tersebut mencakup setiap kromosom dari setiap iterasi. Detail perhitungan yang ditunjukkan adalah nilai *fitness* dari setiap kromosom, kemudian *user* dengan jarak terdekat, setelah itu letak AP beserta nilai yang dimilikinya.



PERHITUNGAN AREA : BEM
LUAS AREA : 8 x 8
JUMLAH ACCESS POINT : 2
JUMLAH POPULASI AWAL : 20
Lokasi Terbaik Access Point 1 Diletakkan Pada Koordinat 0,8
Frekuensi Terbaik Access Point 1 Adalah 2.462
Power Terbaik Access Point 1 Adalah 15
DETAIL :
VALUE : 0.52076021913903
KONEKSI : 0
PATH DIJ : 58.418047492992
Lokasi Terbaik Access Point 2 Diletakkan Pada Koordinat 8,4
Frekuensi Terbaik Access Point 2 Adalah 2.412
Power Terbaik Access Point 2 Adalah 15
DETAIL :

Gambar 5.8 Implementasi hasil perhitungan

Jika pengguna sistem tidak memilih fitur untuk menunjukkan detail perhitungan, maka pada halaman hasil perhitungan akan ditampilkan hasil perhitungan dengan nilai optimal saja. Hasil perhitungan tersebut meliputi jumlah AP yang digunakan dengan mencantumkan lokasi, frekuensi, *power level*, nilai *fitness*, status koneksi, dan *path loss* dij.



BAB 6 PENGUJIAN DAN ANALISIS

Proses pengujian terhadap sistem optimasi penempatan *access point* berdasarkan nilai *path loss* dengan menggunakan algoritma genetika akan dilakukan sesuai dengan skenario pengujian yang telah dijelaskan pada bab 4.3. Skenario pengujian yang dilakukan sebanyak lima skenario, antara lain pengujian pertama adalah pengujian terhadap ukuran banyak generasi, pengujian kedua adalah pengujian terhadap ukuran populasi yang dilakukan dalam proses algoritma genetika, pengujian ketiga merupakan pengujian terhadap nilai *crossover rate* (*cr*) dan *mutation rate* (*Mr*), dan pengujian keempat adalah pengujian terhadap jumlah AP yang digunakan. Pengujian yang dilakukan ini bertujuan untuk mendapatkan nilai *fitness* terbaik dari satu kromosom. Kromosom yang memiliki nilai *fitness* terbaik itulah yang merupakan hasil optimasi dari permasalahan.

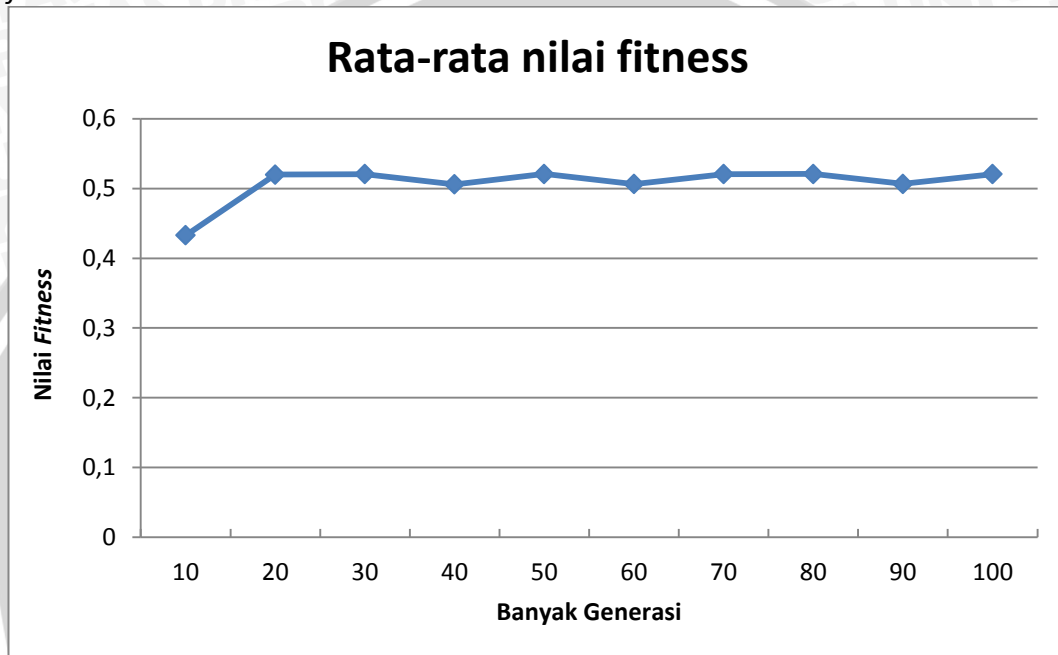
6.1 Hasil dan analisis skenario pengujian banyak generasi

Skenario pengujian pertama akan dilakukan pengujian terhadap banyak generasi. Ukuran generasi yang digunakan dalam pengujian, antara lain 10, 20, 30, 40, 50, 60, 70, 80, 90, dan 100. Ukuran populasi dari hasil pengujian ini akan digunakan untuk skenario pengujian kombinasi nilai *cr* dan *mr* dan jumlah generasi. Pada saat melakukan pengujian ukuran populasi, digunakan generasi sebanyak 10 generasi, 2 AP, dan nilai *crossover rate* - *mutation rate* sebesar 0.9 dan 0.1. Pengujian dilakukan sebanyak 5 kali pada masing-masing ukuran populasi kemudian dihitung rata-rata nilai *fitness*-nya. Hasil skenario pengujian ukuran populasi penentuan penempatan AP ditunjukkan pada **Tabel 6.1**.

Tabel 6.1 Hasil skenario pengujian banyak generasi penentuan lokasi AP

Banyak Generasi	<i>Cr</i> = 0,9 ; <i>Mr</i> = 0,1					Rata-rata nilai <i>fitness</i>
	Percobaan Ke – <i>i</i>					
	1	2	3	4	5	
10	0,5191	0,3811	0,5175	0,3937	0,3534	0,4330
20	0,5191	0,5208	0,5191	0,5208	0,5191	0,5198
30	0,5208	0,5191	0,5208	0,5208	0,5208	0,5205
40	0,4498	0,5208	0,5208	0,5208	0,5175	0,5059
50	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
60	0,5208	0,4498	0,5208	0,5191	0,5208	0,5063
70	0,5208	0,5208	0,5208	0,5191	0,5208	0,5205
80	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
90	0,5208	0,5208	0,5208	0,4498	0,5208	0,5066
100	0,5208	0,5191	0,5208	0,5208	0,5208	0,5205

Banyak generasi yang memiliki rata-rata nilai *fitness* terbesar merupakan banyak generasi yang akan digunakan pada skenario pengujian ukuran populasi. Berdasarkan hasil rata-rata nilai *fitness* pengujian banyak generasi untuk penentuan lokasi AP dapat diketahui rata-rata nilai *fitness* terbesar adalah 0.5208 dengan banyak generasi sebesar 50 dan 80. Jumlah populasi 80 ini akan digunakan dalam skenario pengujian ukuran populasi dalam penentuan lokasi AP. Sedangkan, rata-rata nilai *fitness* terendah terletak pada generasi sebanyak 10. Pada **Gambar 6.1** ditunjukkan grafik yang merepresentasikan rata-rata nilai *fitness*.



Gambar 6.1 Grafik hasil rata-rata nilai *fitness* banyak generasi

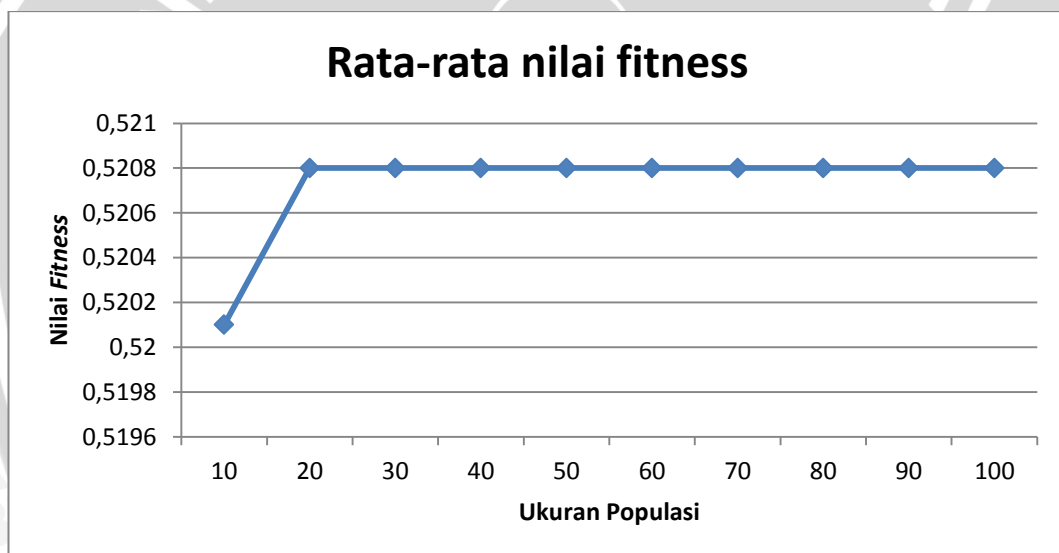
Berdasarkan **Gambar 6.1** dapat diketahui bahwa semakin banyak jumlah generasi yang digunakan maka rata – rata nilai *fitness* akan semakin stabil. Hal ini dikarenakan iterasi yang semakin banyak akan memperluas lingkup pencarian dari algoritma genetika dalam menemukan nilai yang optimal.

6.2 Hasil dan analisis skenario pengujian jumlah populasi

Pada skenario pengujian ini ukuran populasi yang akan diujikan sebanyak 10 nilai ukuran populasi, antara lain 10, 20, 30, 40, 50, 60, 70, 80, 90, dan 100. Berdasarkan pengujian sebelumnya diperoleh jumlah generasi dengan nilai *fitness* terbaik ada pada jumlah 50 dan 80. Pada pengujian kali ini menggunakan jumlah generasi 80, 2 AP, serta nilai *crossover rate-mutation rate* sebesar 0,9 dan 0,1. Dalam pengujian ukuran populasi bertujuan untuk mendapatkan ukuran populasi terbaik dimana dapat menghasilkan nilai akhir yang optimal. Pengujian ukuran populasi ditunjukkan pada Tabel 6.2.

Tabel 6.2 Hasil skenario pengujian ukuran populasi penentuan lokasi AP

Ukuran Populasi	$Cr = 0,9$; $Mr = 0,1$; banyak generasi = 80					Rata-rata nilai <i>fitness</i>
	Percobaan Ke – i					
	1	2	3	4	5	
10	0,5208	0,5208	0,5208	0,5175	0,5208	0,5201
20	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
30	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
40	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
50	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
60	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
70	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
80	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
90	0,5208	0,5208	0,5208	0,4498	0,5208	0,5208
100	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208



Gambar 6.2 Grafik hasil rata-rata nilai *fitness* ukuran populasi

Berdasarkan **Tabel 6.2** diketahui bahwa pada saat menggunakan ukuran populasi 20, rata-rata nilai *fitness* telah mencapai nilai optimal. Hal ini membuktikan bahwa ukuran populasi terbaik bermula pada penggunaan 20 ukuran populasi hingga 100 ukuran populasi, karena pada generasi lebih dari 20 tidak menunjukkan perubahan rata-rata nilai *fitness*. Sebuah grafik pada **Gambar 6.2** merepresentasikan hasil rata-rata nilai *fitness* dari pengujian ukuran populasi untuk optimasi penempatan lokasi AP.

6.3 Hasil dan analisis skenario pengujian kombinasi Cr dan Mr

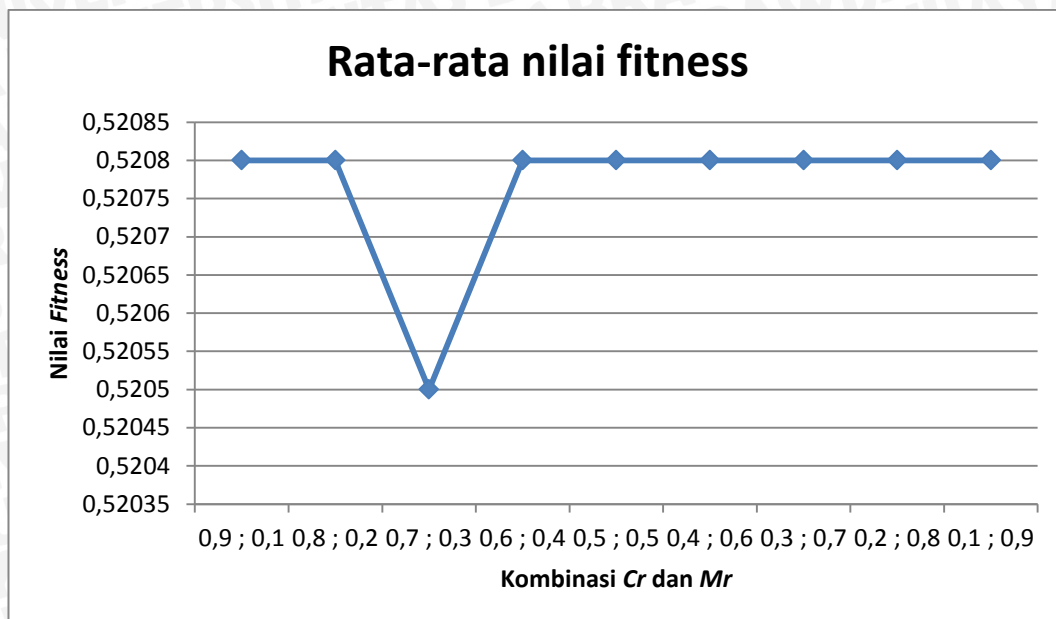
Pada pengujian ini, nilai yang digunakan untuk menguji nilai *crossover rate* dan *mutation rate* adalah 1.0 dan 0.0, 0.9 dan 0.1, 0.8 dan 0.2, 0.7 dan 0.3, 0.6 dan 0.4, 0.5 dan 0.5, 0.4 dan 0.6, 0.3 dan 0.7, 0.2 dan 0.8, 0.1 dan 0.9, serta 0.0 dan 1.0. Pengujian ini bertujuan untuk mendapatkan nilai Cr dan Mr yang tepat sehingga pada proses algoritma genetika menghasilkan hasil akhir yang optimal.

Pada pengujian kombinasi nilai Cr dan Mr digunakan nilai parameter algoritma genetika ukuran populasi sebesar 20, yang didapat dari pengujian ukuran populasi dan jumlah generasi sebanyak 80 generasi. **Tabel 6.3** merupakan hasil dari skenario pengujian dimana setiap percobaan dilakukan 5 kali.

Tabel 6.3 Hasil skenario pengujian kombinasi Cr dan Mr

Kombinasi nilai <i>Cr</i> dan <i>Mr</i>	ukuran populasi = 20 ; jumlah generasi = 80					Rata-rata nilai <i>fitness</i>
	Percobaan Ke – <i>i</i>					
	1	2	3	4	5	
0,9 ; 0,1	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
0,8 ; 0,2	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
0,7 ; 0,3	0,5208	0,5208	0,5191	0,5208	0,5208	0,5205
0,6 ; 0,4	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
0,5 ; 0,5	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
0,4 ; 0,6	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
0,3 ; 0,7	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
0,2 ; 0,8	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
0,1 ; 0,9	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208

Berdasarkan **Tabel 6.3**, diketahui bahwa kombinasi nilai Cr dan Mr sangat sedikit pengaruhnya terhadap perhitungan rata-rata nilai *fitness*. Hal tersebut ditunjukkan dengan minimnya perubahan yang terjadi pada rata-rata nilai *fitness* pada setiap variasi kombinasi Cr dan Mr yang diberikan. Perubahan hanya terjadi pada saat kombinasi Cr dan Mr 0,7 dan 0,3 percobaan ke-3. **Gambar 6.3** akan merepresentasikan perubahan rata-rata nilai *fitness* pengujian kombinasi Cr dan Mr ke dalam sebuah diagram.



Gambar 6.3 Grafik hasil rata-rata nilai *fitness* pengujian kombinasi *Cr* dan *Mr*

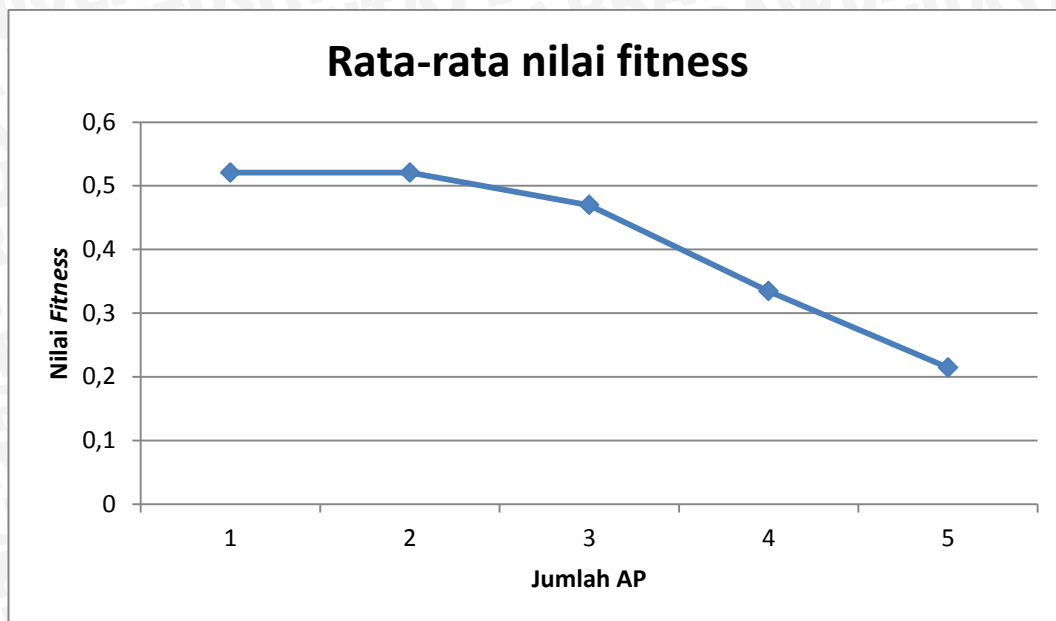
6.4 Hasil dan analisis skenario pengujian jumlah AP

Berdasarkan skenario pengujian jumlah AP, pengujian ini dilakukan dengan masing masing 1 sampai 5 AP. Pengujian ini bertujuan untuk mendapatkan jumlah AP yang dapat menghasilkan rata-rata nilai *fitness* yang optimal.

Pada pengujian jumlah AP digunakan nilai parameter algoritma genetika dengan ukuran populasi sebesar 20, yang didapat dari pengujian ukuran populasi, jumlah generasi sebanyak 80 generasi, dan juga kombinasi *Cr* dan *Mr* sebesar 0.9, dan 0.1. **Tabel 6.4** merupakan hasil dari skenario pengujian yang masing-masing percobaanya dilakukan sebanyak 5 kali.

Tabel 6.4 Hasil skenario pengujian jumlah AP

Jumlah AP	ukuran populasi = 20 ; jumlah generasi = 80					Rata-rata nilai <i>fitness</i>
	Percobaan Ke – <i>i</i>					
	1	2	3	4	5	
1	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
2	0,5208	0,5208	0,5208	0,5208	0,5208	0,5208
3	0,5175	0,5208	0,4486	0,5208	0,3409	0,46972
4	0,5208	0,2989	0,2403	0,2599	0,3519	0,33436
5	0,2054	0,1393	0,1731	0,1731	0,3811	0,2144



Gambar 6.4 Grafik hasil rata-rata nilai *fitness* pengujian jumlah AP

Hasil yang di dapatkan dari **Tabel 6.4** menjelaskan bahwa semakin banyak AP yang digunakan maka akan berdampak pada menurunnya rata-rata nilai *fitness*. Menurunnya rata-rata nilai *fitness* mulai terjadi saat penggunaan lebih dari 3 AP. **Gambar 6.4** merepresentasikan perubahan rata-rata nilai *fitness* pengujian jumlah AP ke dalam sebuah diagram. Penurunan rata-rata nilai *fitness* yang ditunjukkan **Gambar 6.4** dikarenakan semakin banyak jumlah AP akan berakibat pada interferensi sinyal antar AP. Interferensi sinyal merupakan gangguan yang diakibatkan karena sinyal yang saling betumpukan. Interferensi sinyal bisa terjadi jika frekuensi yang digunakan setiap AP sama.

BAB 7 PENUTUP

7.1 Kesimpulan

Kesimpulan yang didapatkan dari hasil perancangan, implementasi dan pengujian adalah :

1. Proses awal yang harus dilakukan dalam mengimplementasikan metode Algoritma Genetika untuk optimasi penempatan AP berdasarkan nilai minimum *path loss* adalah dengan mengumpulkan data berupa denah objek ruangan, lokasi user, dan juga spesifikasi AP. Data tersebut didapatkan melalui pihak BPTIK PTIIK Universitas Brawijaya, berupa peta gedung D dan gazebo PTIIK, serta spesifikasi AP yang digunakan saat ini di bangunan tersebut. Objek ruangan pada gedung D yang digunakan dalam penelitian ini ruangan robotik dan juga himpunan. Proses optimasi menggunakan metode algoritma genetika diawali dengan merepresentasikan data yang ingin diperoleh (lokasi AP, frekuensi, dan *power level*) menjadi suatu kromosom. Kromosom yang telah terbentuk akan dikelompokkan dengan kromosom yang lain sehingga membentuk suatu populasi. Nilai *fitness* ditentukan dari perhitungan nilai *path loss* AP ke *user*. Kromosom dengan nilai *fitness* yang tinggi memiliki peluang lebih banyak untuk melakukan proses reproduksi, yakni proses *crossover*, dan proses mutasi. Generasi selanjutnya akan terbentuk berdasarkan komposisi 20% kromosom terbaik dari populasi sebelumnya, 70% kromosom terbaik dari hasil reproduksi, dan 10% kromosom yang dibangkitkan secara random. Proses akan terus berulang sesuai jumlah generasi yang ditetapkan. Berdasarkan perhitungan yang telah dilakukan didapatkan nilai *fitness* yang optimal sebesar 0.5208.
2. Berdasarkan hasil pengujian banyak generasi, jumlah populasi, kombinasi *Cr* dan *Mr*, dimensi ruangan, serta jumlah AP, didapatkan nilai parameter terbaik yang menghasilkan rata-rata nilai *fitness* sebesar 0.5208. Untuk dapat menghasilkan rata-rata nilai *fitness* yang optimal, nilai parameter banyak generasi yang digunakan sebesar 20 generasi, untuk nilai parameter jumlah populasi sebesar 20 populasi, dan nilai parameter jumlah AP sebesar 2 AP.

7.2 Saran

Peneliti menganggap bahwa penelitian ini masih jauh dari kata sempurna. Oleh karena itu, beberapa saran dari peneliti untuk pengembangan penelitian selanjutnya adalah :

1. Pada penelitian selanjutnya perlu dikembangkan mengenai metode penentuan lokasi user agar dapat menentukan lokasi *user* yang bersifat dinamis.
2. Perlu adanya pengembangan metode *grid point* secara 3D pada objek ruangan sehingga dapat menghasilkan nilai *fitness* yang lebih riil.

3. Perlu adanya database yang menyimpan tentang berbagai macam tipe AP, sehingga sistem tidak hanya terpaku pada satu tipe saja.



DAFTAR PUSTAKA

- Unifi AC, 2016. Spesifikasi AP Unifi, 2016. Tersedia di : http://www.ubnt.com/unifi/unifi_ap/ [Diakses 8 Januari 2016]
- Ungurawanich, N. dan Wattanapongsakorn, N., 2005. *Wireless LAN System Design Considering Average Path Loss Minimization*. Bangkok : Departement of Computer Engineering faculty of Engineering King Mongkut's University of Technology Thonburi.
- Widyaningsih, B., 2012. *OPTIMASI AREA CAKUPAN JARINGAN NIRKABEL DALAM RUANGAN (Studi kasus: PTIIK Universitas Brawijaya)*. Malang : Program Studi Teknik Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
- Supriyatno, A., 2013. *Analisis Kelemahan Keamanan pada Jaringan Wireless*. Semarang : Fakultas Teknologi Informasi, Universitas Stikubank.
- R.L., Haupt dan Haupt, 2004. *Practical Genetic Algorithm*. New Jersey : John Wiley dan Sons, Inc
- E., Turban dan Aronson, 1998. *Decision Support System and Intelligent Systems*. New Jersey : Prentince-Hall Inc.
- J., C., Promak, 2002. *Next Generation Wireless LAN System Design*. In Proceeding MILCOM 2002, Volume 1.
- Zukhri, Zainudin, 2013. *Algoritma Genetikan Metode Komputasi Evolusioner untuk Menyelesaikan Masalah Optimasi*. Yogyakarta :UII.
- Kartika, P., Kurnia, 2010. *Optimasi Penataan Sistem Wi-Fi di PENS-ITS dengan menggunakan Algoritma Genetika*. Surabaya : Kampus ITS.
- Barbosa, S., Antonio, Marco dan Gouvêa, Meirelles, Maury, 2012. *Access Point Design with a Genetic Algorithm*. Brazil : Pontifical Catholic University of Minas Gerais Institute of Exact Sciences and informatics Belo Horizonte.
- Maskuriwong, Kotchakorn, Varavithya, Vara dan Chaiyaratana, Nacho, 2003. *Wireless LAN Access Point Placement using a Multi-Objective Genetic Algorithm*. Bangkok : , Department of Electrical Engineering Research and Development Center for Intelligent Systems King Mongkut's Institute of Technology.

LAMPIRAN A KUESIONER DAN FOTO LOKASI USER

A.1 Foto ruangan dan letak user





A.2 Form Kuesioner

KUESIONER OPTIMASI PENEMPATAN LOKASI AKSES POINT

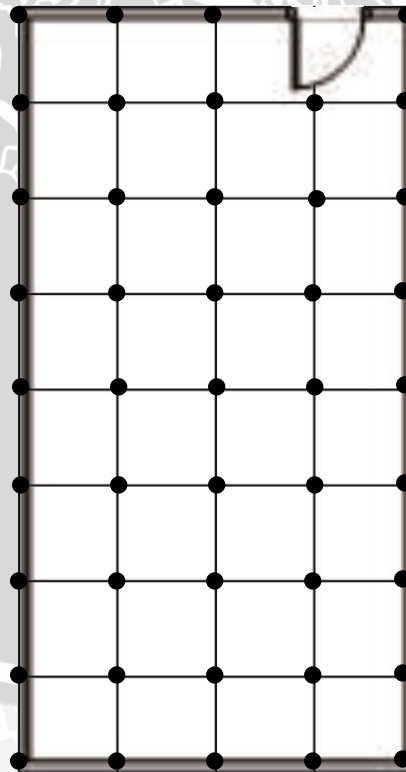
Nama Ruangan :

PETUNJUK PENGISIAN ANGKET

1. Bacalah sejumlah pernyataan di bawah ini dengan teliti.
2. Anda dimohon untuk memberikan penilaian mengenai pengaruh penempatan lokasi Akses Point terhadap kualitas jaringan komputer yang terdapat pada ruangan Anda.
3. Anda dimohon untuk memberikan jawaban sesuai dengan keadaan Anda secara objektif dan dengan melingkari denah ruangan untuk menunjukkan letak pengguna jaringan di ruangan Anda.

1. Sebaran Penempatan Lokasi User pada Ruangan.

Pada persoalan berikut ini Anda akan diminta untuk menunjukkan *spot* atau titik-titik yang menurut Anda sering digunakan oleh Anda ataupun orang lain untuk mengakses internet di dalam ruangan ini. Lingkari titik-titik tersebut pada denah ruangan dibawah ini!



2. Rata-rata pengguna ruangan yang mengakses layanan internet.

Berdasarkan pengalaman dan pengetahuan Anda, berapakah jumlah pengguna ruangan pada saat Anda berada pada ruangan tersebut? Berikan tanda silang (x) pada jawaban yang paling tepat menurut Anda!

- a. 0-5 orang b. 6-10 orang c. 11-15 orang d. 16-20 orang

3. Kualitas jaringan pada ruangan.

Menurut Anda bagaimanakah kualitas jaringan yang terdapat pada ruangan Anda?

Apakah telah memenuhi kriteria Anda? Berikan tanda silang (x) pada jawaban yang paling tepat menurut Anda!

- a. Sangat Buruk b. Buruk c. Cukup d. Baik e. Sangat Baik



A.3 Rekap Hasil Kuesioner Ruang Robotik

Robotik															
No.	user 1	user 2	user 3	user 4	user 5	user 6	user 7	user 8	user 9	user 10	user 11	user 12	user 13	user 14	user 15
1	3.0	7.0	6.1	0.2	3.2	7.2	7.3	1.3	1.4	5.4	1.6	3.7	2.7	1.7	8.8
2	3.0	7.0	6.1	1.1	3.2	6.2	8.3	2.3	1.4	7.4	3.6	5.7	3.7	1.7	6.8
3	3.0	6.0	1.1	6.1	3.2	8.3	1.4	3.6	7.6	7.7	6.7	4.7	3.7	2.7	2.8
4	3.0	1.1	3.2	8.3	2.6	3.6	7.6	5.7	3.7	1.8	2.8	7.8	-	-	-
5	3.6	6.1	1.1	3.2	6.2	7.3	2.3	1.3	8.4	3.6	7.6	5.7	3.7	2.8	6.8
6	1.0	3.0	7.0	6.1	2.1	2.2	3.2	6.2	4.4	7.5	3.6	8.7	3.7	1.7	6.8
7	1.0	7.0	7.1	2.2	3.2	8.3	3.3	3.7	2.7	1.7	3.8	6.8	-	-	-
8	7.1	6.1	2.1	3.2	6.2	8.3	1.3	3.4	3.6	7.6	6.7	5.7	3.7	1.7	2.8
9	8.1	1.1	6.2	3.4	6.5	3.5	2.5	3.6	7.6	6.7	3.7	1.7	2.8	7.8	-
10	2.0	3.0	8.0	7.1	6.1	1.1	3.2	6.2	3.6	5.6	7.6	3.7	1.7	2.8	-

A.4 Rekap Hasil Kuesioner Ruang Himpunan

Himpunan									
No.	user 1	user 2	user 3	user 4	user 5	user 6	user 7	user 8	user 9
1	4.1	3.1	1.4	3.4	3.5	1.6	-	-	-
2	3.1	1.3	3.4	3.5	1.5	1.6	1.7	-	-
3	3.1	1.3	1.4	2.4	2.5	1.5	-	-	-
4	3.1	1.3	1.4	3.4	3.5	1.5	1.6	3.6	-
5	2.1	1.3	1.4	3.4	3.5	1.5	-	-	-
6	1.0	2.0	4.2	0.3	2.4	3.4	5.4	5.5	10.5
7	2.0	1.4	2.4	4.4	1.5	1.6	-	-	-
8	1.0	2.0	4.2	2.3	1.4	4.4	3.5	1.6	-
9	1.0	2.0	2.3	0.3	0.4	2.4	3.5	0.5	-
10	2.0	3.0	1.4	3.5	1.6	0.8	-	-	-

