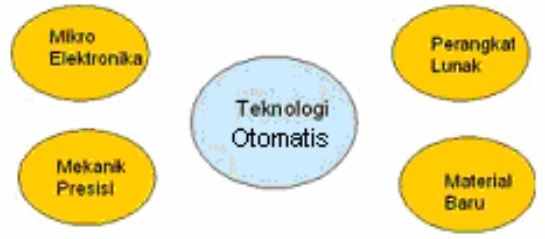
BAB II

LANDASAN TEORI

2.1 Sistem Navigasi Robot Bergerak

Robot adalah satu dari sekian banyak produk Teknologi Otomatis yang didukung oleh 4 disiplin ilmu pengetahuan dan teknologi, yaitu : Teknologi Mikroelektronika, Teknologi Mekanik Presisi, Perangkat Lunak/*Software* dan Teknologi Material seperti yang diperlihatkan dalam Gambar 2.1 di bawah [5].



Gambar 2.1 Teknologi Pendukung Otomatis

Dari keempat disiplin ilmu diatas penulis hanya mengembangkan satu disiplin ilmu yaitu Perangkat Lunak/*Software* dimana penulis hanya membuat simulasi dari sistem navigasi robot bergerak, hal ini yang menjadi dasar pembuatan aplikasi simulasi sistem navigasi robot bergerak dengan menggunakan Algoritma Genetik. Sistem navigasi yang akan dibahas pada tugas akhir ini merupakan bagian dari teknologi Perangkat/Software. Navigasi itu sendiri adalah ilmu pengetahuan tentang bagaimana mengarahkan sesuatu untuk menempuh

suatu jalur tertentu, sehingga sistem navigasi robot bergerak adalah sistem yang mengatur pergerakan robot yang bergerak pada suatu lingkungan tertentu dari posisi awal menuju ke titik tujuan tanpa menabrak penghalang yang ada selama dalam perjalanannya [4].

Pada penentuan sistem navigasi robot seringkali rute yang akan dilalui robot direncanakan secara *off-line* di mana diasumsikan bahwa lingkungan yang akan dilalui robot sudah sangat diketahui dan bersifat stasioner (tetap dan tidak berubah-ubah). Pada kenyataannya lingkungan yang akan dilalui robot mungkin saja berubah-ubah, hal ini mendorong dilakukannya penelitian penentuan rute secara *on-line* di mana robot akan melakukan penginderaan pada lingkungan untuk mengetahui penghalang-penghalang apa saja yang ada pada saat robot menjalani rutenya [4].

2.2 Algoritma Genetik

2.2.1 Konsep Dasar Algoritma Genetik

Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom antar individu organisme. Variasi kromosom ini akan mempengaruhi laju reproduksi dan tingkat kemampuan organisme untuk tetap hidup. Pada dasarnya ada empat kondisi yang sangat mempengaruhi proses evaluasi, yaitu:

1. Kemampuan organisme untuk melakukan reproduksi

2. Keberadaan populasi organisme yang bisa melakukan reproduksi
3. Keberagaman organisme dalam suatu populasi.
4. Perbedaan kemampuan untuk *survive*.

Individu yang lebih kuat (fit) akan memiliki tingkat survival dan tingkat reproduksi yang lebih tinggi jika dibandingkan dengan individu yang kurang fit. Pada kurun waktu tertentu (sering dikenal generasi), populasi secara keseluruhan akan lebih banyak memuat organisme yang fit.

Algoritma genetik pertama kali dikembangkan oleh John Holland dari Universitas Michigan (1975). John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. Algoritma genetika adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom [3].

Algoritma genetik, menggunakan mekanisme seleksi alam dan ilmu genetik. Hal ini bcrarti bahwa istilah-istilah yang terdapat pada algoritma genetik akan bersesuaian dengan istilah-istilah pada seleksi alam dan ilmu genetik. Dalam ilmu genetik kromosom terdiri dari susunan gen-gen. Tiap gen mengandung nilai atau sifat tertentu yang disebut allele, sedangkan posisi gen dalam kromosom disebut locus. Selanjutnya, satu atau beberapa kromosom bergabung membangun paket genetik yang disebut genotif. Interaksi genotif dengan lingkungannya disebut fenotif.

Tabel 2.1 Istilah Ilmu Genetik dan Algoritma Genetik

Ilmu genetik	Algoritma genetik
Kromosom	String
Gen	Karakter
Allele	Nilai karakter
Locus	Posisi dalam string
Genotif	Struktur
Fenotif	Parameter

Algoritma genetik termasuk kedalam algoritma pencarian, proses kerja algoritma ini diinspirasi dari proses teori Darwin yaitu berdasarkan mekanisme seleksi alam dan genetika biologi untuk menentukan struktur-struktur atau individu-individu berkualitas tinggi yang terdapat dalam sebuah domain yang disebut populasi.

Teknik pencarian algoritma genetik dimulai dengan pembangkitan sebuah populasi yang terdiri dari individu-individu. Setiap individu dalam populasi disebut kromosom, yang mewakili suatu solusi dari masalah. Suatu kromosom merupakan sebuah string dari simbol. Kromosom-kromosom tersebut berkembang melalui ulangan berturut-turut, yang disebut sebagai generasi. Pada setiap generasi, kromosom ditentukan nilainya dengan menggunakan beberapa ukuran dari *fitness.* Untuk menghasilkan generasi berikutnya, yaitu kromosom-kromosom baru atau kromosom anak yang disebut *offspring*, dibentuk dengan melakukan proses persilangan dan mutasi sebagai proses reproduksi, juga disebut sebagai operator genetik.

Setelah beberapa generasi, algoritma menuju ke satu titik yaitu kromosom terbaik, yang diharapkan mewakili solusi yang optimal atau suboptimal pada masalah. Dalam satu siklus generasi terdapat dua tahap proses yakni proses seleksi dan rekombinasi.

Tahap seleksi adalah tahap mengevalusi kualitas setiap individu dalam populasi untuk memperoleh peringkat kandidat solusi. Berdasarkan hasil evaluasi tersebut kemudian dipilih individu-individu yang akan mengalami proses rekombinasi. Proses pemilihan ini biasanya dilakukan secara acak di mana individu-individu yang memiliki kemungkinan yang lebih besar untuk dipilih sebagai calon individu baru dari generasi berikutnya.

Tahap rekombinasi meliputi proses genetika untuk memperoleh populasi baru dari calon-calon individu yang didapat pada tahap seleksi. Anggota populasi baru ini diperoleh dengan menerapkan tiga operator genetik yaitu reproduksi, mutasi, dan pindah silang secara acak pada calon-calon individu yang terpilih pada tahap seleksi. Di tahap rekombinasi ini akan diperoleh individu-individu baru yang berbeda dengan individu-individu induknya dan dengan demikian akan diperoleh domain pencarian yang baru.

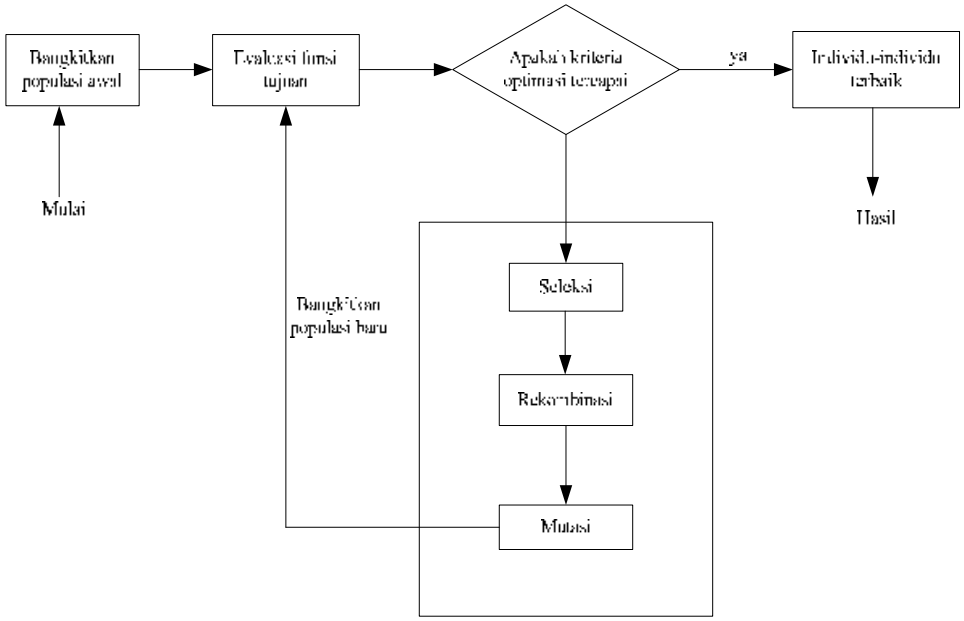
Masalah utama dari algoritma genetik adalah representasi parameter-parameter masalah ke dalam kromosom. Mekanisme dari suatu algoritma genetik sederhana tidak terlalu sulit, hanya mencakup proses penduplikasian string dan pertukaran bagian-bagian dalam string. Meskipun mekanisme ini cukup sederhana, tetapi mempunyai kekuatan untuk menyelesaikan permasalahan optimasi, kekuatan yang terdapat dalam algoritma genetik sederhana ini didukung

oleh tiga operator genetik. Pada reproduksi terjadi proses penduplikasian individu berdasarkan nilai fungsi objektifnya, nilai objektif ini dapat dilihat sebagai suatu keuntungan yang ingin dicapai atau dimaksimalkan. Sementara pertukaran bagian-bagian string dilakukan oleh operator pindah silang dan mutasi [1].

Langkah-langkah dari algoritma genetik sederhana adalah sebagai berikut [1]:

- Inisialisasi, yakni pengisian populasi awal dengan kromosom-kromosom secara acak berdasarkan kriteria tertentu.
- Evaluasi *nilai fitness* yang ada pada populasi.
- Reproduksi dari duplikasi kromosom-kromosom berdasarkan nilai *fitnessnya* sebagai calon anggota populasi berikutnya.
- *Crossover* (pindah silang), yakni proses untuk menghasilkan kromosom-kromosom baru dari "perkawinan" kromosom-kromosom induk untuk menambah keragaman kromosom dalam populasi.
- Mutasi, merupakan operasi yang dilakukan dengan mengubah satu atau lebih nilai yang terdapat pada kromosom.
- Ulangi langkah 2 sampai 5 hingga tercapai solusi atau jumlah generasi maksimum dicapai.

Secara umum, diagram alir algoritma genetik sederhana seperti terlihat pada gambar berikut [3]:



Gambar 2.2 Diagram Alir Algoritma Genetik Sederhana

2.2.2 Struktur Umum Algoritma Genetika

Pada algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin yang dikenal dengan istilah **populasi**. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan poopulasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan istilah **generasi**. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi **fitness**. Nilai fitness dari suatu kromosom akan menunjukan kualitas kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (***offspring***) terbentuk dari gabungan dua kromosom

generasi sekarang yang bertindak sebagai induk (***parent***) dengan menggunakan operator penyilangan (***crossover***). Selain operator penyilangan, suatu kromosom dapat dapat juga dimodifikasi dengan menggunakan operator **mutasi**. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai fitness dari kromosom induk (*parent*) dan nilai fitness dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik [3].

2.2.3 Teknik Penyandian

Teknik penyandian dalam algoritma genetik meliputi penyandian gen dari kromosom [3]. Gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk string bit, pohon, array, bilangan real, daftar aturan, elemen permutasi, elemen program atau representasi lainnya. Demikian juga dengan kromosom, kromosom dapat direpresentasikan dengan menggunakan:

- string bit : 10011, 01101, 11101, dan seterusnya
- bilangan real : 65.65, -67.98, 562.88, dan seterusnya
- elemen permutasi : E2, E10, E5, dan seterusnya
- daftar aturan : R1, R2, R3, dan seterusnya
- elemen program genetika
- struktur lainnya

2.2.4 Inisialisasi Populasi

Dua teknik umum yang digunakan untuk menginisialisasi suatu populasi. Teknik pertama adalah dengan membangkitkan populasi dimana informasi genetik tentang individu yang terterdapat disuatu populasi dapat diakses dari suatu media penyimpanan sekunder. Data ini nantinya akan memberikan titik awal untuk suatu evolusi yang dilaksanakan. Teknik kedua adalah membangkitkan suatu populasi dimana informasi genetik tentang individu disuatu populasi dibangkitkan secara acak [3].

2.2.5 Parameter

Pada algoritma genetik, parameter digunakan untuk memberikan suatu probabilistik (kemungkinan atau peluang) yang berupa presentase. Probabilitas yang ada dalam algoritma ini adalah probabilitas *crossover* dan probabilitas mutasi. Probabilitas tersebut merupakan parameter yang penting dalam algoritma genetik, disamping ukuran populasinya (*population size*) [1].

- Probabilitas *crossover*

  Merupakan presentase dari seringnya melakukan *crossover*. Jika *crossover* tidak ada, maka anak *(offspring)* akan meniru induknya. Tetapi jika *crossover* ada, *offspring* dibuat dari bagian kromosom induk (jika kemungkinan *crossover* adalah 100%, maka semua *offspring* dibuat oleh *crossover*, tetapi jika 0% maka *offspring* dibuat dengan meniru kromosom dari populasi lama secara keseluruhan).

- Probabilitas mutasi

Merupakan bagian dari kromosom yang dimutasikan. Jika mutasi ada, bagian kromosom diubah (jika kemungkinannya 100%, seluruh kromosom akan diubah dan jika 0% tidak ada yang diubah), tetapi jika tidak ada mutasi, *offspring* langsung mengambil dari *crossover* dengan tidak merubah lagi.

- Ukuran populasi (panjang kromosom)

  Merupakan kumpulan kromosom dalam populasi (dalam satu generasi). Setiap generasi ukuran populasi harus sama.Ukuran populasi sebaiknya tidak lebih kecil dari 30 untuk sembarang jenis permasalahan [3].

2.2.6 Komponen-komponen Algoritma Genetik

Pada bagian ini akan diuraikan komponen-komponen pembentuk algoritma genetik yakni operator-operator genetika dasar (reproduksi, pindah silang, mutasi), fungsi *fitness* dan seleksi [1].

2.2.6.1 Reproduksi

Reproduksi merupakan proses penyalinan string individu sesuai dengan nilai *fitness*nya, dalam proses ini setiap individu dalam populasi pada suatu generasi akan diseleksi berdasarkan nilai *fitness*nya. Probabilitas terpilihnya satu individu untuk bereproduksi adalah sebesar nilai *fitness* seluruh individu dalam populasi. Banyaknya salinan satu individu sebanding dengan nilai *fitness* individu tersebut yang berarti bahwa individu dengan nilai *fitness* lebih besar akan direproduksi lebih banyak.

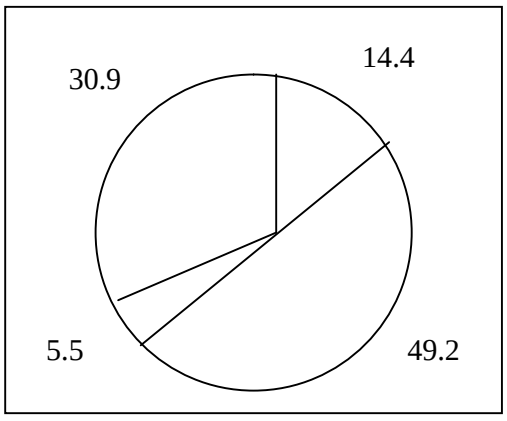
Beberapa motode pemilihan dalam reprodusi, diantaranya adalah sebagai berikut:

a. *Roulette Wheel Method*

Salah satu cara untuk mengimplementasikan proses reproduksi adalah dengan piringan rolet *(roulette wheel).* Piringan rolet ini dibagi menjadi bagian-bagian yang sama dengan jumlah populasi. Tiap individu memiliki bagian dengan ukuran tertentu yang sebanding dengan nilai *fitness*nya. Pemilihan individu untuk mereproduksi dilakukan dengan memutar piringan rolet sebanyak N kali, dimana N adalah ukuran populasi. Proses pemilihan dengan piring rolet ini akan diperjelas dengan contoh yang menggunakan empat buah string seperti yang terlihat pada tabel 2.2 dan gambar 2.3.

Tabel 2.2 Contoh String dan Nilai F*itness*

No	String	Fitness	%total
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
Total		1170	100



Gambar 2.3 Piring Rolet dari Data pada Tabel 2-2

Ukuran besar sektor pada pring rolet ditentukan dengan ukuran besarnya presentase nilai *fitness* dari setiap string sebagai mana yang terlihat pada gambar.

Reproduksi dilakukan dengan memutar string rolet sebanyak jumlah anggota populasi yang diinginkan. Dengan cara ini, string yang memiliki nilai *fitness* yang tinggi memiliki kemungkinan besar untuk dipilih.

b. *Steady-state Method*

Ide utama dari metode ini adalah bahwa bagian terbesar dari kromosom harus dapat bertahan hidup ke generasi berikutnya. Cara kerja dari metode ini adalah dalam setiap generasi dipilih beoerapa kromosom (yang memiliki nilai *fitness* tinggi) untuk menciptakan kromosom baru, kemudian beberapa kromosom dengan nilai *fitness* buruk akan dikeluarkan dan digantikan tempatnya oleh kromosom yang baru diciptakan sebelumnya. Kromosom lain dari populasi akan bertahan ke generasi berikutnya.

c. *Elitism Method*

Metode ini menciptakan populasi baru dimana pada saat terjadi proses pindah silang atau mutasi menyebabkan kemungkinan bahwa kromosom dengan nilai *fitness* terbaik akan hilang. Metode elitisme ini bertujuan untuk mencegah hilangnya kromosom terbaik pada generasi berikutnya. Prosesnya adalah menduplikasi terlebih dahulu kromosom-kromosom terbaik dari populasi tersebut ke generasi berikutnya.

2.2.6.2 Pindah Silang (Crossover)

Operator pindah silang adalah operator yang sangat penting di dalam algoritma genetik. Pindah silang dilakukan atas dua kromosom yang mempunyai dua bagian yang berbeda sifatnya sehingga persilangan harus bisa menanganinya. Tujuan dari pindah silang adalah menambah individu dalam populasi dengan mengawinkan individu pada populasi untuk menghasilkan kromosom anak *(offspring)*. Kromosom anak yang terbentuk akan mewarisi sebagian sifat kromosom induknya.

Metode pindah silang terdiri dari beberapa jenis diantaranya:

a. Penyilangan Satu Titik *(One Point Crossover)*

Pada penyilangan satu titik, posisi penyilangan k (1,2,...,N-1) dengan N=panjang kromosom, diseleksi secara random. Variabel-variabel ditukar dengan antar kromosom secara random pada titik tersebut untuk menghasilkan anak.

Pada metode ini pindah silang dilakukan dengan cara:

- Memasangkan secara acak individu-individu baru hasil reproduksi
- Memilih posisi pindah silang dalam string acak (disebut titik pindah silang), bila individu memiliki panjang sebesar n, diambil secara acak posisi ke-k dari individu dengan k terletak diantara satu sampai panjang string dikurangi satu (1,n-1)
- Melakukan pertukaran bagian string baik sebelah kiri maupun sebelah kanan titik pindah silang dengan bagian dari string pasangan yang telah dipartisi dengan cara yang sama, contoh dari proses pindah silang satu titik dapat dilihat pada gambar 2.4.

Induk A	1	1	0	0	1	|	0	1	1
Induk B	1	1	0	1	1	|	1	1	1
Anak C	1	1	0	0	1		1	1	1
Anak D	1	1	0	1	1		0	1	1

Gambar 2.4 Penyilangan Satu Titik

Dari hasil pindah silang dapat dilihat bahwa anak C mewarisi bagian pertama dari induk A dan mewarisi bagian kedua dari induk B, sedang anak D mewarisi sisanya.

b. Penyilangan Dua Titik *(Two Point Crossover)*

Pada penyilangan dua titik ini proses pindah silangnya sama dengan cara penyilangan satu titik hanya saja titik pertukarannya ada dua tempat, contoh dari proses pindah silang dua titik dapat dilihat pada gambar 2.5.

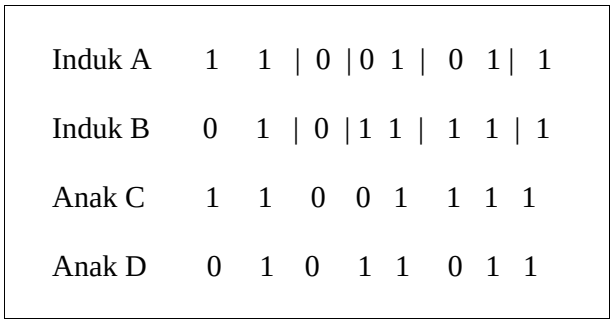
Induk A	1	1	|	0	0	1	|	0	1	1
Induk B	0	1	|	0	1	1	|	1	1	1
Anak C	1	1		0	1	1		0	1	1
Anak D	0	1		0	0	1		1	1	1

Gambar 2.5 Penyilangan Dua Titik

Contoh di atas menunjukkan bahwa kromosom induk terbagi menjadi tiga bagian, anak C akan mewarisi gen bagian pertama dari induk pertama, bagian kedua dari induk kedua dan bagian ketiga dari induk pertama, sedangkan anak lainnya akan mendapatkan sisa bagian yang tidak diwariskan ke anak pertama.

c. Penyilangan Banyak Titik *(Multi Point Crossover)*

Pada penyilangan banyak titik ini posisi penyilangan dengan panjang kromosom diseleksi secara acak dan tidak diperbolehkan ada posisi yang sama serta diurutkan naik. Variabel-variabel ditukar pada titik-titik tersebut untuk menghasilkan anak [3].



Gambar 2.6 Penyilangan Banyak Titik

Contoh di atas menunjukkan bahwa kromosom induk terbagi menjadi banyak bagian, anak C dan D akan bergantian mewarisi setiap potongan dari induknya.

2.2.6.3 Mutasi

Dengan menggunakan operator reproduksi dan pindah silang tersebut dapat terjadi kemungkinan terhapusnya materi yang penting dalam struktur tertentu dan tidak ada cara untuk memperoleh kembali materi yang hilang tersebut. Operator mutasi diperkenalkan sebagai cara untuk mengembalikan materi yang hilang. Melalui mutasi individu baru dapat diciptakan dengan melakukan modifikasi terhadap satu atau lebih nilai gen pada individu yang sama.

Operator mutasi ini memungkinkan untuk melakukan pencarian dalam sembarang daerah dalam ruang persoalan, oleh karena itu peluang melakukan pencarian pada suatu daerah tidak pernah nol, hal ini berfungsi agar operator mutasi dapat mencegah kehilangan total materi genetik setelah reproduksi dan pindah silang.

Proses mutasi suatu kromosom dapat dilakukan dengan cara mengubah nilai pada sebuah atau beberapa gen secara acak berdasarkan peluang mutasinya. Contoh proses mutasi dapat dilihat pada tabel 2.3.

Tabel 2.3 Contoh Proses Mutasi

Kromosom A	11001011
Hasil mutasi	11000011

Contoh diatas menujukkan bahwa kromosom A mengalami mutasi pada gen kelima yakni nilai awal bit 1 menjadi bit 0. Sekali lagi posisi gen ditentukan secara acak untuk proses mutasi.

Terdapat beberapa jenis mutasi diantaranya [1]:

a. Mutasi bilangan real

Pada rnutasi bilangan real, ukuran langkah mutasi biasanya sangat sulit ditentukan. Ukuran yang kecil biasanya sering mengalami kesuksesan, namun adakalanya ukuran yang lebih besar akan berjalan lebih cepat.

Operator mutasi untuk bilangan real ini dapat ditetapkan sebagai berikut:

- variabel yang dimutasi = variabel ± range * delta (+ atau – memiliki probabilitas yang sama
- range = 0.5 * domain variabel; (interval pencarian)
- delta = ? $(a_i * 2^{-I})$; $a_i = 1$ dengan probabilitas 1/m selain itu $a_i = 0$ dengan m = 20

b. Mutasi bilangan biner

Cara sederhana untuk mendapatkan biner adalah mengganti satu atau beberapa nilai gen dari kromosom. Langkah-langkah mutasi adalah sebagai berikut:

- hitung jumlah gen pada populasi (panjang kromosom dikalikan dengan ukuran populasi)
- pilih secara acak gen yang akan dimutasi
- tentukan kromosom dari gen yang terpilih untuk dimutasi
- ganti nilai gen dari kromosom yang akan diganti tersebut

2.2.6.4 Fungsi Fitness

Fitness merupakan istilah yang digunakan dalam ilmu biologi sebagai ukuran kinerja suatu individu untuk tetap bertahan hidup dalam lingkungannya [1]. Dalam algoritma genetik fungsi *fitness* adalah fungsi objektif dari masalah yang akan dioptimasi. Fungsi objektif ini dapat dibayangkan sebagai pengukuran keuntungan *(profit)* yang ingin dimaksimalkan atau pengukuran biaya *(cost)* yang ingin diminimalkan. Setiap masalah yang akan dioptimasi memerlukan pendefinisian fungsi *fitness* tersebut. Individu dengan kinerja yang lebih tinggi akan memiliki *nilai fitness* lebih baik.

Setiap individu dalam populasi memiliki *nilai fitness* tertentu sebagai hasil dari interaksi dengan lingkungannya. *Fitness* merupakan kekuatan penggerak dari seleksi baik seleksi Darwin di alam maupun algoritma genetik. Nilai *fitness* dalam algoritma genetik diperoleh dari fungsi *fitness* permasalahan yang dihadapi. Fungsi *fitness* harus sesuai dengan permasalahan yang akan dioptimasi. Fungsi *fitness* yang ditentukan dengan baik akan menjamin keberhasilan pencarian pada algoritma genetik.

Fungsi *fitness* terdiri dari beberapa jenis, diantaranya [1]:

a. Raw *fitness*

Merupakan pengukuran secara alami terhadap permasalahan. Definisi raw *fitness* tergantung masalah yang dihadapi, untuk masalah yang berbeda dapat dibuat fungsi *fitness* yang berbeda pula. Jenis fungsi ini termasuk yang paling sederhana seperti mendefinisikan fungsi obyektif berupa memaksimumkan pendapatan atau meminimumkan biaya dan lain-lain.

b. Standardized *fitness*

Fungsi ini mengambil nilai dari *raw fitness* sedemikian rupa sehingga nilai terendah memiliki nilai yang lebih baik. Jadi dengan kata lain, individu yang lebih baik pada suatu populasi, nilai *standardized fitness* nya akan lebih kecil. Bila Nilai *raw fitness* lebih rendah maka lebih baik, rumusnya :

$$S(I,t)=r(I,t)$$

Sedangkan bila nilai *raw fitness* lebih tinggi lebih baik, rumusnya :

$$S(I,t) = r_{max}-r(I,t)$$

Dimana S (I,t) adalah *standardized fitness* individu ke-I pada generasi t.

r (I,t) adalah *raw fitness* individu ke-I pada generasi t.

Rmax adalah nilai *raw fitness* maksimum yang mungkin.

c. *Adjust fitness*

Nilai yang digunakan pada perhitungan adjust *fitness* ini dihitung dari *standardized fitness,* yakni :

$$a(I,T)=\frac{1}{1+S(i,t)}$$

Dimana a (I,t) adalah nilai *adjusted fitness* individu ke-I pada generasi t.

S (I,t) adalah *standardized fitness* individu ke-I pada generasi I.

Nilai *fitness* ini terletak diantara 0 dan 1. berbeda dengan *standardized fitness,* nilai adjust *fitness* akan semakin besar untuk individu yang semakin baik pada suatu populasi. Adjust *fitness* bcrguna bagi permasalahan dimana kesalahan yang semakin kecil menunjukkan solusi yang lebih baik sehingga kesalahan sama dengan nol akan diperoleh suatu individu terbaik.