BAB II
KAJIAN PUSTAKA DAN DASAR TEORI

2.1. Kajian Pustaka Algoritma Genetik Pada *Body Repair* Mobil

Algoritma genetika bekerja dari populasi atau sekumpulan kromosom yang merupakan himpunan solusi yang diberi nilai secara acak. Setiap anggota himpunan yang membentuk nilai tertentu dinamakan kromosom. Kromosom dalam suatu populasi berevolusi dalam iterasi yang dinamakan generasi, tiap kromosom dievaluasi berdasarkan *fitness*. Pada algoritma genetika, *fitness* biasanya dapat berupa fungsi objektif dari masalah yang akan dioptimalisasi.

Pada optimasi *body repair* mobil ini *fitness* diambil dari nilai keuntungan, proses awal pemberian nilai random pada tiap kromosom di populasi awal, lalu dilanjutkan dengan proses *crossover* lalu didapatkan *offspring* dan terbentuk populasi baru selanjutnya di proses mutasi dari kromosom yang terpilih. Dari dua proses tersebut, maka terbentuk suatu generasi baru dan dilakukan proses hitung *fitness* dan dilanjutkan dengan proses seleksi yang akan menyaring nilai *fitness* yang tertinggi atau teroptimal.

2.2. Sistem *Body Repair* Mobil

Servis pembetulan pengecatan *body* mobil yang di kenal dengan nama *body repair* mobil, harus yang dikerjakan dengan tenaga-tenaga yang terampil dan terlatih, agar mendapatkan hasil akhir yang sempurna.

Pada sistem *body repair* mobil ini proses perbaikan dilakukan secara keseluruhan oleh tiap-tiap pekerja akan mengerjakan mulai dari pendempulan, pengamplasan, hingga melakukan pemoxyan hingga akhir.

2.3. Harga Pokok *repair*

Biaya *repair* atau harga pokok *repair* merupakan kumpulan dari biaya-biaya yang dikeluarkan untuk membeli dan mengolah bahan dasar sampai selesai. Biaya-biaya tersebut terdiri dari:

a. Biaya Bahan Baku (BBB)

b. Biaya Tenaga Kerja (BTK)
c. Biaya Overhead Bengkel (BOB)

2.3.1. Biaya Bahan Baku

Biaya bahan baku adalah harga perolehan (harga pokok) seluruh materi pokok yang terdapat pada barang jadi. Bahan baku merupakan bagian barang jadi yang dapat ditelusuri keberadaanya.

Pada bidang bengkel body *repair* bahan baku meliputi pembelian bahan baku cat, dempul, kertas gosok dan keperluan lainnya. Dirata-rata total dari biaya bahan baku adalah 75.000 per panel. Barang-barang tersebut harus sudah tersedia ketika proses *repair* dilakukan, bila barang baku kurang atau tidak tersedia maka proses pengerjaan *repair* tidak akan berjalan.

2.3.2. Biaya Tenaga Kerja

Tenaga kerja adalah pekerja yang memiliki kinerja langsung terhadap proses berjalannya pengerjaan body repair mobil, baik menggunakan tenaga fisiknya maupun bantuan mesin. Tenaga kerja memperoleh kontraprestasi yang dikategorikan sebagai biaya tenaga kerja. jadi biaya tenaga kerja adalah semua kontraprestasi yang diberikan kepada tenaga kerja. Pada bidang body repair mobil ini biaya tenaga kerja meliputi biaya-biaya upah proses pengerjaan mobil hingga selesai, dan menjadi suatu ketetapan untuk ongkos kerja 75.000 per panel dan satu hari kerja ditetapkan 8 jam kerja.

2.3.3. Biaya Overhead Bengkel

Biaya overhead bengkel adalah biaya-biaya yang timbul dalam proses pengolahan yang tidak dapat digolongkan dalam biaya bahan baku dan biaya tenaga kerja. Biaya yang termasuk dalam overhead bengkel meliputi biaya tenaga kerja tidak pasti seperti biaya upah pengawas, mekanik mesin. Biaya bahan penolong, meliputi macam-macam bahan yang digunakan dalam proses pengolaan yang kuantitasnya sangat kecil dan sulit ditelusuri pada bahan baku. Juga biaya penyusutan gedung bengkel dan biaya penyusutan mesin. Yang di rata-rata dan ditetapkan oleh bengkel besar BOB adalah 25.000 perpanel.

2.4. Biaya Operasional

Biaya operasional adalah biaya yang dikeluarkan pada saat proses menuju pengerjaan, biaya transpotasi membeli bahan baku,dll. Ditetapkan biaya operasional yang dikeluarkan bengkel adalah 100.000 per hari.

2.5. Keuntungan

Secara garis besar keuntungan adalah hasil atau laba yang diperoleh dari pengerjaan, dalam bidang bengkel *body repair* mobil ini keuntungan dapat dihitung dengan menjumlah uang yang diperoleh dikurangi dengan biaya bahan baku dan biaya tenaga kerja serta biaya-biaya yang digunakan untuk pemrosesan *repair* tersebut .

2.6. Optimasi Waktu

Optimasi menurut kamus besar bahasa Indonesia adalah optimalisasi yang diartikan sebagai pengoptimalan, yaitu proses, cara, perbuatan untuk menghasilkan yang paling baik.

Optimasi waktu digunakan untuk melakukan proses atau perbuatan agar didapat waktu seminimal mungkin, sehingga didapatkan waktu yang paling baik, dengan didapat waktu yang paling baik maka akan terbentuk keuntungan yang paling maksimal

2.7. Algoritma Genetik

Algoritma genetik adalah teknik pencarian yang di dalam ilmu komputer untuk menemukan penyelesaian perkiraan untuk optimisasi dan masalah pencarian. Algoritma genetik adalah kelas khusus dari algoritma evolusioner dengan menggunakan teknik yang terinspirasi oleh biologi evolusioner seperti warisan, mutasi, seleksi alam dan rekombinasi (atau *crossover*)**.**

Algoritma ini ditemukan di Universitas Michigan, Amerika Serikat oleh John Holland (1975) melalui sebuah penelitian dan dipopulerkan oleh salah satu muridnya, David Goldberg.

Algoritma genetik adalah algoritma yang berusaha menerapkan pemahaman mengenai evolusi alamiah pada tugas-tugas pemecahan-masalah (*problem solving*). Pendekatan yang diambil oleh algoritma ini adalah dengan menggabungkan secara acak berbagai pilihan solusi terbaik di dalam suatu kumpulan untuk mendapatkan generasi solusi terbaik berikutnya yaitu pada suatu kondisi yang memaksimalkan kecocokannya atau lazim disebut *fitness* (Nugraha,2008)

2.7.1.Definisi-definisi Dalam Algoritma Genetika

Terdapat beberapa definisi penting dalam algoritma genetika yaitu (lia,2006):

- **Gen**

  Gen merupakan nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom
- **Kromosom / individu**

  Kromosom merupakan gabungan dari gen-gen yang membentuk nilai tertentu dan menyatakan solusi yang mungkin dari suatu permasalahan.
- **Populasi**

  Populasi merupakan sekumpulan individu yang akan diproses bersama dalam satu satuan siklus evolusi.
- **Fitness**

  Fitness menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan.
- **Seleksi**

  Seleksi merupakan proses untuk mendapatkan calon induk yang baik.
- **Crossover**

  Crossover merupakan proses pertukaran atau kawin silang gen-gen dari dua induk tertentu.
- **Mutasi**

  Mutasi merupakan proses pergantian salah satu gen yang terpilih dengan nilai tertentu.
- **Generasi**

Generasi merupakan urutan iterasi dimana beberapa kromosom bergabung.

- **Offspring**

  Offspring merupakan kromosom baru yang dihasilkan setelah melewati suatu generasi

2.7.2 Alur Kerja Algoritma Genetik

Skema dasar dari algoritma genetika, sebagai berikut [1](magdalena,2008)

1. [MULAI]Buat populasi secara random dengan n kromosom (solusi yang pantas terhadap masalah)
2. [FITNESS]Evaluasi nilai *fitness f(x)* dari setiap kromosom x dipopulasi
3. [POPULASI BARU] Buat populasi baru dengan mengulangi langkah berikut sampai populasi baru lengkap

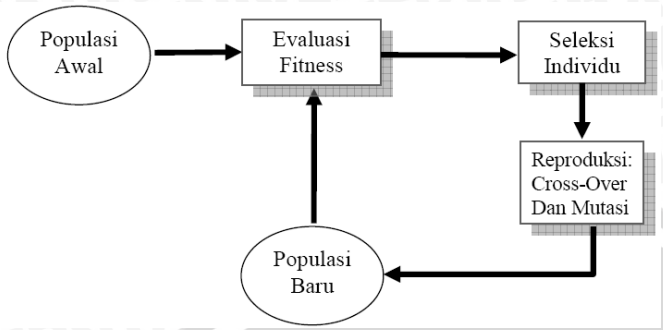
   •[SELEKSI] Pilih dua kromosom *parent* dari populasi berdasarkan nilai *fitness*-nya (nilai *fitness* terbaik biasanya mempunyai kesempatan terbesar untuk dipilih)

   •[CROSSOVER/PERKAWINAN SILANG] Dengan kemungkinan *crossover* maka *crossover* kedua *parent* untuk membentuk *offspring* baru (anak). Jika *crossover* tidak terjadi maka anak adalah sama seperti *parent*nya.

   •[MUTASI] Dengan kemungkinan mutasi maka mutasi *offspring* baru pada tiap bagiannya (posisi dalam kromosom).

   • [PENERIMAAN]Tempatkan *offspring* baru pada populasi baru.
4. [GANTI] Gunakan populasi baru yang telah digenerasi untuk digunakan selanjutnya dalam algoritma.
5. [TES] Jika kondisi telah terpenuhi maka perulangan berhenti dan berikan solusi terbaik dari populasi,
6. [ULANGI]jika kondisi tidak terpenuhi kembali ke langkah 2.

Secara umum, proses/siklus algoritma genetika digambarkan pada gambar 2.1 yang pertama kali diperkenalkan oleh David Goldberg.



Gambar 2.1 Siklus Algoritma Genetika oleh David Goldberg

2.7.3 Pengkodean

Proses Pengkodean (*Encoding*) pada proses pengkodean, gen dapat direpresentasikan dalam bentuk string bit, pohon, array bilangan real, daftar aturan, elemen permutasi, elemen program, atau representasi lainnya yang dapat diimplementasikan untuk operator genetika. (fajar, taufik.2007)

Ada beberapa macam teknik pengkodean yang dapat dilakukan dalam algoritma genetika, diantaranya pengkodean biner (*binary encoding*), pengkodean permutasi (*permutation encoding*), pengkodean nilai (*value encoding*) dan pengkodean pohon (*tree encoding*)

Untuk menyelesaikan suatu permasalahan menggunakan Algoritma Genetika, perlu diketahui beberapa macam encoding guna menentukan operator crossover dan mutasi yang akan digunakan. Encoding tersebut tergantung pada permasalahan apa yang diangkat. Beberapa macam encoding akan dijelaskan di bawah ini (lia,2006)

2.7.3.1 Pengkodean Biner

Pengkodean jenis ini sering digunakan. Kromosom dari pengkodean biner ini berupa kumpulan dari nilai biner 0 dan 1.

Contohnya:

Chromosome1	1101100100110110
Chromosome2	1101011000011110

Dalam pengkodean biner memungkinkan didapatkan kromosom dengan nilai allele yang kecil, tetapi kekurangannya tidak dapat digunakan untuk beberapa permasalahan dan terkadang
diperlukan adanya koreksi setelah proses crossover dan mutasi. Salah satu permasalahan yang menggunakan pengkodean adalah menghitung nilai maksimal dari suatu fungsi.

2.7.3.2 Pengkodean Permutasi

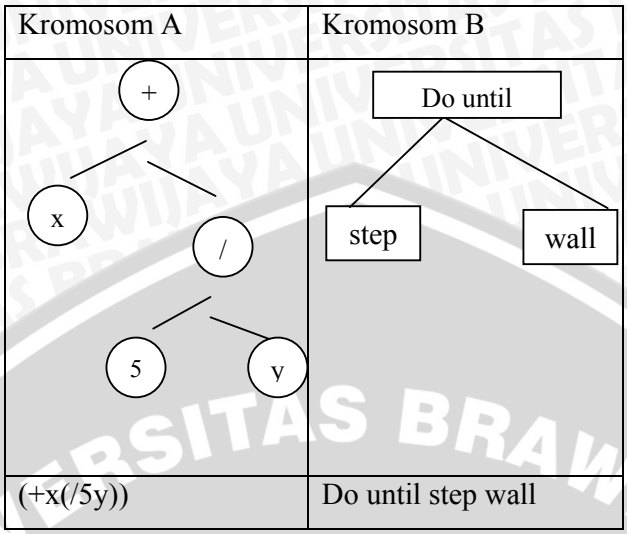
Kromosom dari pengkodean permutasi ini berupa kumpulan dari nilai integer yang mewakili suatu posisi dalam sebuah urutan.Biasanya digunakan pada permasalahan TSP (TravellingSalesman Problem).

2.7.3.3Pengkodean Nilai

Kromosom dari pengkodean nilai berupa kumpulan dari suatu nilai, yang bisa berupa macam-macam nilai sesuai dengan permasalahan yang dihadapi, seperti bilangan real, char atau objek yang lain. pengkodean ini merupakan pilihan yang bagus untuk beberapa permasalahan khusus, biasanya diperlukan metode khusus untuk memproses crossover dan mutasinya sesuai dengan permasalahan yang dihadapi.

2.7.3.4. Pengkodean Pohon/Tree Encoding

Pengkodean pohon biasanya digunakan pada genetic programming. Kromosom yang digunakan berupa sebuah tree dari beberapa objek, seperti fungsi atau command pada genetic programming



Gambar 2.2 Contoh kromosom dengan pengkodean pohon.

2.8 Fungsi *Fitness*

Fungsi evaluasi merupakan masalah yang penting dalam algoritma genetik. Fungsi evaluasi yang baik harus mampu memberikan nilai *fitness* sesuai dengan kinerja kromosom. Pada permulaan optimasi, umumnya nilai *fitness* masing-masing individu mempunyai rentang yang lebar. Seiring dengan bertambahnya generasi beberapa kromosom mendominasi populasi dan menyebabkan nilai *fitness* akan mempunyai rentang yang sempit dan pada akhirnya setelah melewati beberapa generasi, nilai *fitness* akan terkonsentrasi ke suatu nilai tunggal yang dianggap sebagai nilai terbaik. Karena yang akan dihitung adalah nilai optimum dari suatu fungsi, maka fungsi *fitness* yangdipilih umumnya adalah fungsi itu sendiri.

2.9 Operasi Genetik

Didalam Operasi genetik terdapat 2 proses yaitu tukar silang (*crossover*) dan mutasi.

2.9.1 Crossover

Proses tukar silang (*crossover*) berfungsi untuk menghasilkan keturunan dari dua buah kromosom induk yang terpilih. Kromosom anak yang dihasilkan merupakan kombinasi gen-gen yang dimiliki oleh kromosom induk. Probabilitas

tukar silang (pc) menentukan frekuensi dari proses tukar silang yang terjadi dalam suatu populasi. Semakin tinggi nilai pc maka semakin besar kemungkinan dilakukan tukar silang (Kuswara, 2003).

Secara umum, mekanisme tukar silang adalah sebagai berikut (Kuswara, 2003):

1. memilih dua buah kromosom sebagai induk.
2. memilih secara acak posisi dalam kromosom, biasa disebut titik perkawinan silang, sehingga masing-masing kromosom induk terpecah menjadi dua segmen.
3. lakukan pertukaran antar segmen kromosom induk untuk menghasilkan kromosom anak.

Pada pengkodean biner terdapat 4 metode penyilangan yang sering digunakan. Metode tersebut adalah (Lia, 2006):

a. Crossover Satu Titik.

Memilih satu titik tertentu, selanjutnya nilai biner sampai titik crossovernya dari induk pertama digunakan dan sisanya dilanjutkan dengan nilai biner dari induk kedua.

Contoh: **11001**011 + 11011**111** = **11001111**

b. Crossover Dua Titik.

Memilih dua titik tertentu, lalu nilai biner sampai titik crossover pertama pada induk pertama digunakan,dilanjutkan dengan nilai biner dari titik pertama sampai titik kedua dari induk kedua, kemudian sisanya melanjutkan nilai biner dari titik kedua pada induk pertama lagi.

Contoh :**11**001**11 + 11**011**11 = 11011111**

c. Crossover Silang Seragam

Sebuah *mask* penyilangan dibuat sepanjang panjang kromosom secara random yang menunjukkna bit-bit dalam *mask* yang mana induk akan mensuplay anak dengan bit-bit yang ada. Disini, anak_1 akan dihasilkan dari induk_1 jika bit *mask* bernilai 1, dan anak_1 akan dihasilkan dari induk_2 jika bit *mask* bernilai 0. Sedangkan anak_2 dihasilkan dari kebalikan *mask*.

Contoh:

Misalkan ada 2 kromosom dengan panjang 12:

Induk 1: 011100101110

Induk 2: 110100001101

Mask bit:

sampel 1: 100111001101

sampel 2: 011000110010

Setelah penyilangan, diperoleh kromosom baru:

Anak 1: 010100001100

Anak 2: 111100101111

d. Crossover Aritmatik

Suatu operasi aritmetika digunakan untuk menghasilkan offspring yang baru

Contoh: 11001011 + 11011111 = 11001001 (AND)

Pada pengkodean permutasi terdapat *5* metode *crossover* yaitu (Mitsuo, 1997) :

***a. Partial-Mapped Crossover* (PMX)**

PMX seperti halnya pada *crossover* dua titik untuk string biner, namun pada PMX menggunakan prosedur perbaikan khusus untuk menyelesaikan kromosom yg tidak sesuai yg disebabkan oleh crossover dua titik sederhana. Jadi esensinya PMX yaitu *crossover* dua titik ditambah prosedur perbaikan. Langkah-langkahPMX sebagai berikut:

1. Pilih dua posisi sepanjang string secara acak.
2. Tukar dua substring antara orang tua untuk menghasilkan *proto-child.*
3. Tentukan hubungan pemetaan di antara dua bagian substring tersebut.
4. Perbaiki kromosom *proto-child* menggunakan hubungan pemetaan diatas.

Contoh :

1. Parent 1 : 1 2 **3 4 5 6** 7 8 9

Parent 2 : 5 4 **6 9 2 1** 7 8 3

2. *Proto-child 1* : 1 2 **6 9 2 1** 7 8 9

Proto-child 2 : 5 4 **3 4 5 6** 7 8 3

3. 3↔6↔1, 9↔4, 2↔5
4. *Offspring 1* : **3 5 6 9 2 1** 7 8 **4**
   Offsring 2 : **2 9 3 4 5 6** 7 8 **1**

***b. Order Crossover* (OX)**

Metode ini mirip dengan PMX dengan prosedur perbaikan yang berbeda. OX bekerja sebagai berikut:

1. Pilih substring dari *parent* 1 secara acak.
2. Dihasilkan *proto-child* dengan menyalin substring tersebut ke posisi yang sama dengan *parent* 1.
3. Hapus gen-gen yang sudah ada di substring dari *parent* 2.
4. Tempatkan gen yang tidak terhapus pada posisi yang kosong didalam *proto-child* dari kiri ke kanan sesuai dengan urutan pada *parent* 2.

Contoh :

1. *Parent 1* : 1 2 **3 4 5 6** 7 8 9
   Parent 2 : 5 4 6 9 2 1 7 8 3
2. *Proto-child* : - - **3 4 5 6** - - -
3. *Parent 2* : - - - 9 2 1 7 8 -
4. *Offspring* : 9 2 **3 4 5 6** 1 7 8

c. Position-Based Crossover

Metode ini sama seperti pada metode OX dalam hal perbaikan kromosom. Langkah-langkah metode ini sebagai berikut:

1. Pilih beberapa posisi dari *parent* 1 secara acak.
2. Menghasilkan *proto-child* dengan menyalin gen pada posisi yang dipilih tadi dan diletakkan sama seperti pada posisi awal.
3. Hapus gen-gen pada *parent* 2 yang sama dengan yang terpilih pada *parent* 1.
4. Tempatkan gen yang tidak terhapus pada posisi yang kosong didalam *proto-child* dari kiri ke kanan sesuai dengan urutan pada *parent* 2.

Contoh :

1. *Parent 1* : 1 2 **3** 4 **5** **6** 7 **8** 9

Parent 2 : 5 4 6 9 2 1 7 8 3

2. *Proto-child* : - - **3** - **5** **6** - **8** -
3. *Parent 2* : - 4 - 9 2 1 7 - -
4. *Offspring* : 4 9 **3** 2 **5** **6** 1 **8** 7

d. Order-Based Crossover

Metode ini sangat mirip seperti pada metode *Position-Based Crossover*. Langkah-langkah metode ini sebagai berikut:

1. Pilih beberapa posisi dari *parent* 1 secara acak.
2. Hapus gen-gen pada *parent* 2 yang sama dengan yang terpilih pada *parent* 1.
3. Menghasilkan *proto-child* dengan menyalin gen-gen *parent* 2 yang tersisa dan diletakkan sama seperti pada posisi awal.
4. Tempatkan gen-gen *parent* 1 yang terpilih pada posisi yang kosong didalam *proto-child* dari kiri ke kanan sesuai dengan urutan pada *parent* 1.

Contoh :

1. *Parent 1* : 1 2 **3** 4 **5** **6** 7 **8** 9

   Parent 2 : 5 4 6 9 2 1 7 8 3
2. *Parent 2* : - 4 - 9 2 1 7 - -
3. *Proto-child* : - 4 - 9 2 1 7 - -
4. *Offspring* : **3** 4 **5** 9 2 1 7 **6** **8**

***e. Cycle Crossover* (CX)**

CX mirip dengan metode dengan *Position-Based Crossover*. Perbedaannya adalah bahwa gen-gen dari *parent* 1 tidak dipilih secara acak tetapi dipilih berdasarkan siklus yang dihasilkan antar *parent*. Langkah-langkah metode CX sebagai berikut:

1. Cari siklus yang dihasilkan antar *parent*.
2. Salin gen-gen *parent* 1 yang termasuk dalam siklus sebagai *proto-child* dengan posisi yang sesuai dari *parent* 1.
3. Tentukan gen-gen *parent* 2 yang tersisa yang tidak termasuk dalam siklus.
4. Tempatkan gen-gen *parent* 2 yang tersisa pada posisi yang kosong didalam *proto-child* dari kiri ke kanan sesuai dengan urutan pada *parent* 2.

Contoh :

1. *Parent 1* : **1 2** 3 **4 5** 6 7 8 **9**
   Parent 2 : **5 4** 6 **9 2** 3 7 8 **1**
   Siklus : 1→5→2→4→9→1
2. *Proto-child* : **1 2** - **4 5** - - - **9**
3. *Parent 2* : - - 6 - - 3 7 8 -
4. *Offspring* : **1 2** 6 **4 5** 3 7 8 **9**

2.9.2 Mutasi

Mutasi merupakan cara lain untuk membuat variasi populasi. Sama seperti *crossover*, mutasi juga mempunyai probabilitas mutasi (pm) untuk menentukan tingkat mutasi yang terjadi. Proses mutasi dilakukan untuk menghindari solusi-solusi dalam populasi mempunyai nilai lokal optimum.

Pada pengkodean permutasi terdiri dari 4 metode sebagai berikut (Mitsuo,1997) :

1. **Inversion Mutation**

Metode bekerja dengan memilih dua posisi dalam kromosom secara acak dan kemudian membalikkan substring di antara dua posisi tersebut.

Contoh :

a. Memilih dua posisi secara acak : 1 2 **3 4 5** 6
b. Membalikkan substring tersebut: 1 2 **5 4 3** 6

2. **Insertion Mutation**

Metode bekerja dengan memilih sebuah gen secara acak dan kemudian gen tersebut disisipkan dalam posisi yang acak.

Contoh :

a. Memilih sebuah gen secara acak : 1 2 3 **4** 5 6
b. Mensisipkan dalam posisi acak: 1 2 **4** 3 5 6

3. **Displacement Mutation**

Pada metode ini dilakukan dengan memilih beberapa gen secara acak dan kemudian disisipkan secara acak pula.

Contoh :

a. Memilih beberapa gen secara acak : 1 2 **3** 4 5 **6**
b. Mensisipkan dalam posisi acak : 1 **6** 3 4 6 **3**

4. **Reciprocal Exchange Mutation**

Metode ini dilakukan dengan memilih dua posisi gen secara acak dan saling menukar posisi untuk masing-masing gen tersebut.

Contoh :

a. Memilih dua posisi gen secara acak: 1 2 **3** 4 **5** 6
b. Menukar menukar posisi untuk masing-masing gen : 1 2 **5** 4 **3** 6

2.9.3 Seleksi

Pada proses Seleksi merupakan proses terakhir setelah melakukan crossover dan mutasi. Seleksi dilakukan guna mendapatkan calon induk yang baik. Seleksi dipengaruhi oleh nilai fitness. Hal ini mempunyai tujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang mempunyai nilai fitness terbaik. Semakin tinggi nilai fitness suatu individu maka semakin besar kemungkinan individu tersebut untuk terpilih, dan sebaliknya.

Probabilitas Seleksi

Probabilitas seleksi dari kromosom dihitung berdasarkan nilai *fitness*-nya. Langkah-langkah yang diperlukan dalam menghitung probabilitas sebagai berikut (Kuswara, 2003):

1. Menghitung nilai *fitness* dari setiap individu.
2. Menjumlahkan nilai *fitness* dari semua individu yang terdapat dalam populasi.

$$TotFit = \sum_{i=1}^{m} Fitness(i)$$

Keterangan :

TotFit = jumlah nilai *fitness*

m = jumlah individu dalam populasi

3. Menghitung probabilitas terpilihnya setiap individu (pi) dengan membagi *fitness* masing-masing individu dengan total nilai *fitness* populasi.

$$pi = \frac{Fitness}{TotFit}$$

4. Menghitung probabilitas kumulatif untuk setiap individu dalam populasi (*qi*).

$$qi = \sum_{i-1}^{j} pi$$

Mekanisme Seleksi

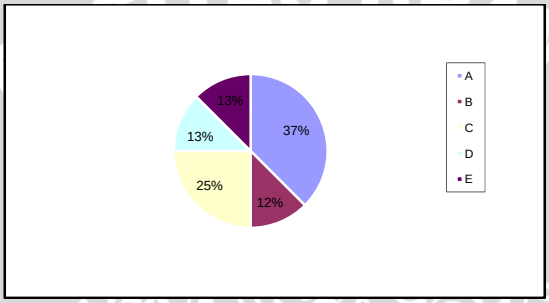
Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai fitness. Seleksi mempunyai tujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang mempunyai nilai *fitness* terbaik. Beberapa jenis seleksi yang umum dipakai adalah (Yohan, 2007):

a. **Seleksi Roda *Roulette***

Roulette *wheel selection*. Metode ini diajukan oleh John Holland. Ide dasarnya adalah untuk menentukan proporsi probabilitas seleksi atau probabilitas *survival* pada tiap kromosom sesuai dengan nilai *fitness*-nya. Individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap segmen individu memiliki ukuran yang sama dengan ukuran *fitness*-nya. Sebuah bilangan random dibangkitkan dan individu yang memiliki segmen dalam kawasan bilangan random tersebut akan terseleksi. Proses ini diulang hingga diperoleh sejumlah individu yang diharapkan.Metode seleksi roda *roulette* juga sering dikenal dengan nama *stochastic sampling with replacement*. Pada metode ini induk dipilih berdasarkan nilai *fitness*nya, semakin besar nilai *fitness* maka akan semakin besar kemungkinannya untuk terpilih menjadi induk. Diandaikan semua kromosom diletakkan pada sebuah roda *roulette*, besarnya kemungkinan bagi setiap kromosom adalah tergantung dari nilai *fitness*nya seperti pada contoh berikut:

Tabel 2.1 Contoh populasi dengan 5 kromosom

Kromosom	*Fitness*
A	15
B	5
C	10
D	5
E	5



Gambar 2.3 Probabilitas suatu kromosom dalam roda *roulette*

Pada gambar 2.3 merupakan contoh dalam satu populasi terdiri dari lima kromosom. Pada tiap kromosom memiliki nilai *fitness* yang berbeda-beda. Pada gambar 2.4 dapat diketahui probabilitas terpilihnya masing-masing kromosom untuk menjadi induk. Pada kromosom A memiliki nilai *fitness* 15 dan nilai tersebut nilai *fitness* tertinggi pada populasi tersebut. Sehingga kromosom A memiliki probabilitas terbesar untuk terpilih menjadi induk.

b. Seleksi Rangking

Pada metode seleksi roda *roulette* akan bermasalah saat terdapat perbedaan *fitness* yang jauh. Sebagai contoh, jika *fitness* kromosom yang terbaik adalah 90% dari semua roda *roulette* dapat menyebabkan kromosom lain memiliki kesempatan yang kecil untuk dapat terpilih (Nia, 2006).

Proses dimulai dengan merangking atau mengurutkan kromosom di dalam populasi berdasarkan *fitness*nya kemudian memberi nilai *fitness* baru berdasarkan urutannya. Kromosom dengan nilai terburuk akan memiliki *fitness* baru nilai 1, terburuk kedua bernilai 2 dan begitu seterusnya, sehingga kromosom yang

memiliki *fitness* terbaik akan memiliki nilai *fitness* N, dimana N adalah jumlah kromosom di dalam populasi. Seperti dapat dilihat pada gambar 2.5 (Nia, 2006).

Tabel 2.2 Keadaan sebelum dirangking

Kromosom	*Fitness*
A	15
B	5
C	10
D	5
E	5

Tabel 2.3 Keadaan setelah dirangking

Kromosom	*Fitness*	*Fitness* baru
B	5	1
D	5	2
E	5	3
C	10	4
A	15	5