

PERTEMUAN 17

PROSES PADA ALGORITMA GENETIKA

A. Tujuan Pembelajaran

Setelah menyelesaikan pada pertemuan ini, mahasiswa mampu menerapkan Proses pada algoritma genetika . Sub materi pada pertemuan ini yaitu:

1. Komponen algoritma genetika
2. Proses seleksi
3. Proses Regenerasi
4. Perkembangan Metode Penjadwalan
5. Contoh Pemakaian algoritma genetika

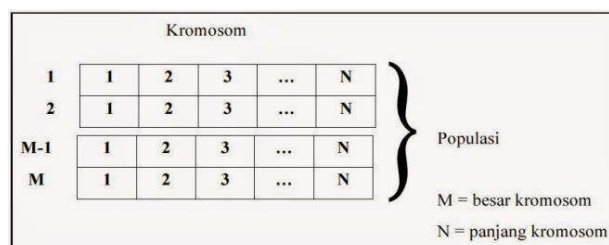
B. Uraian Materi

1. Komponen Algoritma Genetika

Komponen algoritma genetika yang perlu diketahui yaitu teknik pengkodean, individu, pembangkitan populasi, fungsi fitness dan seleksi.

a. Teknik Pengkodean

Pada proses pengkodean, gen dipresentasikan dalam bentuk string bit, tree, array, elemen permutasi dan program, bilangan real, atau representasi lainnya yang bisa diterapkan untuk operator genetika. Teknik pengkodean ada beberapa macam yang bisa dilakukan dalam genetic Algorithm, contohnya binary encoding, permutation encoding, value encoding, dan yang terakhir tree encoding.



Gambar 17.1 Teknik pengkodean algoritma genetika

b. Membangkitkan Populasi

Kebanyakan metode optimasi klasik memasukkan urutan deterministik dan komputasi berdasarkan gradien atau turunan dengan orde lebih tinggi dari fungsi objektif. Metode ini diterapkan pada titik tunggal dalam *space search*. Pendekatan *point to point* ini dapat menyebar pada lokal optimal. Algoritma genetika menampilkan *mutiple directional search* dengan menjaga populasi dari solusi potensial (Berlianty & Arifin 2010). Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak atau melalui *procedure* tertentu. Ukuran untuk populasi tergantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian dilakukan pembangkitan populasi awal.

Teknik dalam pembangkitan populasi awal ini ada beberapa cara, diantaranya adalah sebagai berikut:

- 1) Seperti pada metode *random seach*, pencarian solusi dimulai dari suatu titik uji tertentu. Titik uji tersebut dianggap sebagai alternative solusi yang disebut sebagai populasi.
- 2) *Random Generator* adalah melibatkan pembangkitan bilangan random untuk nilai setiap gen sesuai dengan representasi kromosom yang digunakan.
- 3) Pendekatan tertentu (memasukan nilai tertentu kedalam gen) Cara ini adalah dengan memasukan nilai tertentu kedalam gen dari populasi awal yang dibentuk.
- 4) Permutasi Gen Cara ini adalah penggunaan permutasi josephus dalam permasalahan kombinatorial seperti TSP.

c. Fungsi fitness

Suatu individu atau kromosom dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performasinya. Fungsi yang digunakan untuk mengukur nilai kecocokan atau derajat optimalitas suatu kromosom disebut dengan *fitness function*. Nilai yang dihasilkan dari fungsi tersebut menandakan seberapa optimal solusi yang diperoleh. Nilai yang dihasilkan oleh fungsi fitness merepresentasikan seberapa banyak jumlah persyaratan yang dilanggar, sehingga dalam kasus

penjadwalan perkuliahan semakin kecil jumlah pelanggaran yang dihasilkan maka solusi yang dihasilkan akan semakin baik.

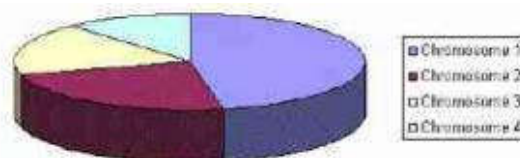
2. Proses Seleksi

Operasi seleksi dilakukan dengan memperhatikan fitness dari tiap individu, manakah yang dapat dipergunakan untuk generasi selanjutnya. Seleksi ini digunakan untuk mendapatkan calon induk yang baik, semakin tinggi nilai fitnessnya maka semakin besar juga kemungkinan individu tersebut terpilih. Terdapat beberapa macam cara seleksi untuk mendapatkan calon induk yang baik, diantaranya adalah seleksi roulette wheel, steady state, tournament dan rank. Proses seleksi yang biasa digunakan adalah mesin roulette (roulette wheel).

Beberapa penjelasan tentang keempat metode seleksi di atas adalah sebagai berikut :

a. *Roulette Wheel*

Calon induk yang akan dipilih berdasarkan nilai fitness yang dimilikinya, semakin baik individu tersebut yang ditunjukkan dengan semakin besar nilai fitnessnya akan mendapatkan kemungkinan yang lebih besar untuk terpilih sebagai induk. Misalkan saja roulette wheel merupakan tempat untuk menampung seluruh kromosom dari tiap populasi, maka besarnya tempat dari roulette wheel tersebut menunjukkan seberapa besar nilai fitness yang dimiliki oleh suatu kromosom, semakin besar nilai fitness tersebut, maka semakin besar pula tempat yang tersedia. Ilustrasinya terlihat pada Gambar 17.2.



Gambar 17.2 Ilustrasi seleksi dengan *Roulette Wheel*

b. *Steady State*

Metode ini tidak banyak digunakan dalam proses seleksi karena dilakukan dengan mempertahankan individu yang terbaik. Pada setiap generasi, akan dipilih beberapa kromosom dengan nilai fitnessnya yang terbaik sebagai induk, sedangkan kromosom-kromosom yang memiliki nilai fitness terburuk akan

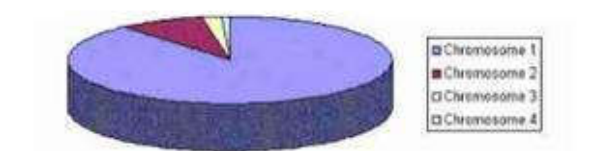
digantikan dengan offspring yang baru. Sehingga pada generasi selanjutnya akan terdapat beberapa populasi yang bertahan.

c. *Tournament*

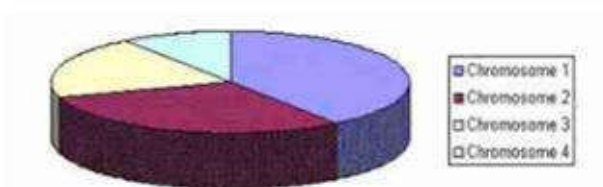
Dalam metode seleksi *tournament* sejumlah individu dipilih secara acak dan kemudian menentukan fitnessnya. Kebanyakan metode seleksi ini digunakan pada *binary*, dimana hanya dua individu yang dipilih.

d. *Rank*

Seleksi ini memperbaiki proses seleksi yang sebelumnya yaitu roulette wheel karena pada seleksi tersebut kemungkinan salah satu kromosom mempunyai nilai fitness yang mendominasi hingga 90% bisa terjadi, sehingga nilai fitness yang lain akan mempunyai kemungkinan yang sangat kecil sekali untuk terpilih. Sehingga dalam seleksi rank, dilakukan perumpamaan sesuai dengan nilai fitnessnya, nilai fitness terkecil diberi nilai 1, yang terkecil kedua diberi nilai 2, dan begitu seterusnya sampai yang terbagus diberi nilai N (jumlah kromosom dalam populasi). Nilai tersebut yang akan diambil sebagai presentasi tepat yang tersedia. Ilustrasinya dapat dilihat seperti pada Gambar 17.3 dan 17.4 berikut :



Gambar 17.3 Metode *rank selection* (situasi sebelum ranking)



Gambar 17.4 Metode *rank selection* (situasi setelah ranking)

3. Proses Regenerasi

Dalam proses regenerasi ini dilakukan tiga buah proses utama yang dipilih secara acak untuk setiap generasi. Namun pemilihan secara acak ini berdasarkan persentase tertentu. Ketiga proses tersebut adalah mutasi, kawin silang, atau reproduksi. Dari ketiga proses ini persentase kemungkinan proses tersebut dijalankan terhadap suatu generasi adalah sama. Karena masing-masing proses mempunyai kemungkinan menghasilkan gen terbaik. Sekalipun dalam proses regenerasi tidak dibawa sifat gen induknya, namun ada kemungkinan menghasilkan gen terbaik. Kemudian dilakukan proses seleksi dan pengulangan proses regenerasi sejumlah generasi.

a. Proses Mutasi

Mutasi juga merupakan salah satu operator penting dalam algoritma genetika selain *crossover*. Metode dan tipe mutasi yang dilakukan juga tergantung pada *encoding* dan permasalahan yang diangkat. Berdasarkan *encoding*nya terdapat beberapa macam, diantaranya adalah sebagai berikut :

1) *Binary Encoding*

Melakukan inversi pada bit yang terpilih, 0 menjadi 1 dan sebaliknya, 1 menjadi 0.

Contoh : 11001001 => 10001001

2) *Permutation Encoding*

Memilih dua nilai dari gen dan menukarnya.

Contoh : (1 2 3 4 5 8 9 7) => (1 8 3 4 5 6 2 9 7)

Beberapa operator mutasi telah diciptakan untuk representasi permutasi, seperti metode *inversion*, *insertion*, *displacement*, dan *reciprocal exchange mutation*.

a) *Inversion Mutation*

Inversion mutation memilih dua posisi dalam sebuah kromosom dengan cara acak dan kemudian menginversikan substring di antara dua posisi tersebut.

b) *Insertion Mutation*

Insertion Mutation memilih sebuah gen dengan cara acak dan

memasukkan ke dalam kromosom dengan cara acak pula.

c) *Displacement Mutation*

Displacement Mutation memilih sebuah sub/sekelompok gen dengan cara acak kemudian memasukkan ke dalam kromosom dengan cara acak.

d) *Reciprocal Exchange Mutation (REM)*

Reciprocal Exchange Mutation memilih dua posisi secara acak, kemudian menukar dua gen dalam posisi tersebut.

3) *Value Encoding*

Menentukan sebuah nilai kecil yang akan ditambahkan atau dikurangkan pada

salah satu gen dalam kromosom.

Contoh : (1.29 5.68 2.86 4.11 5.55) => (1.29 5.68 2.73 4.22 5.55)

4) *Tree Encoding*

Node yang terpilih akan diubah. Karena proses mutasi juga merupakan salah satu operator dasar dalam algoritma genetika, sehingga sama dengan *crossover*, mutasi juga memerlukan *probabilitas* dengan proses yang sama seperti pada probabilitas *crossover*. Individu dengan nilai *probabilitas* yang lebih kecil dari *probabilitas* yang telah ditentukan yang akan melewati proses mutasi. Nilai *probabilitas* mutasi ini menunjukkan seberapa sering gen tertentu dari kromosom yang telah diproses dengan *crossover* akan melewati mutasi. Jika tidak ada proses mutasi, maka offspring yang dihasilkan akan sama dengan hasil individu setelah proses *crossover*, tanpa ada perubahan sedikitpun. Proses mutasi ini biasanya dilakukan untuk mencegah terjadinya local optimum, proses mutasi ini sebaiknya tidak terlalu sering dilakukan karena proses algoritma genetika akan cepat berubah menjadi *random search*. Pada probabilitas mutasi, jika terlalu rendah akan mengakibatkan banyak gen yang berguna tidak sempat untuk dimanfaatkan dan jika terlalu besar akan menyebabkan *offspring* kehilangan sifat dari induknya dan tidak akan dapat memanfaatkan lagi proses evolusi alamiah.

b. Proses Kawin Silang

Proses kawin silang (crossover) adalah salah satu operator penting dalam algoritma genetika, metode dan tipe *crossover* yang dilakukan tergantung dari *encoding* dan permasalahan yang diangkat. Ada beberapa cara yang bisa digunakan untuk melakukan *crossover* sesuai dengan *encodingnya* yang dijelaskan sebagai berikut:

1) *Binary Encoding*

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 : 11001011 + 11011111 = 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 : 11001011 + 11011111 = 11011111

c. *Crossover Uniform*

Nilai biner yang digunakan dipilih secara *random* dari kedua induk.

Contoh : 11001011 + 11011101 = 11011111

d. *Crossover Aritmatik*

Suatu operasi aritmetika digunakan untuk menghasilkan *offspring* yang baru.

Contoh : 11001011 + 11011111 = 11001001 (AND)

2) *Permutation Encoding*

Memilih satu titik tertentu, nilai permutasi sampai titik *crossover* pada induk pertama digunakan lalu sisanya dilakukan scan terlebih dahulu, jika nilai

permutasi pada induk kedua belum ada pada *offspring* nilai tersebut ditambahkan.

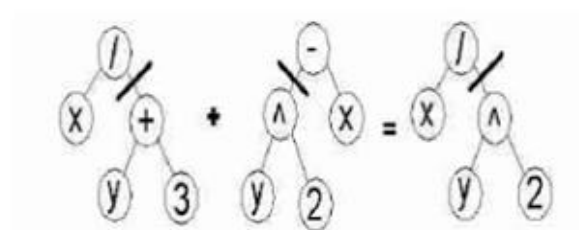
Contoh : $(123456789) + (453689721) = 12345689$

3) Value Encoding

Semua metode *crossover* pada *binary crossover* bisa digunakan.

4) Tree Encoding

Memilih satu titik tertentu dari tiap induk, dan menggabungkan *tree* dibawah titik pada induk pertama dan *tree* di bawah titik pada induk kedua.



Gambar 17.5 Contoh *tree encoding* pada *crossover*

c. Kondisi Berhenti

Offspring merupakan kromosom baru yang dihasilkan setelah melalui proses-proses di atas. Kemudian pada *offspring* tersebut dihitung fitnessnya apakah sudah optimal atau belum, jika sudah optimal berarti *offspring* tersebut merupakan solusi optimal, tetapi jika belum optimal maka akan diseleksi kembali, begitu seterusnya sampai terpenuhi kriteria berhenti.

4. Perkembangan Metode Penjadwalan

Sekarang ini banyak ditemukan metode dan algoritma-algoritma yang dibuat untuk tujuan memecahkan persoalan-persoalan yang ada. Kemudian pada perkembangannya metode atau algoritma tersebut mulai diterapkan untuk memecahkan persoalan penjadwalan, antara lain algoritma semut atau *Ant Colony Optimization (ACO)* dengan pendekatan *Max Min Ant System (MMAS)*, *Taboo Search*, dan teknik pewarnaan graf (*Coloring Graph*).

a. Ant Colony Optimization

Ant Colony Optimization (ACO) terinspirasi oleh koloni-koloni semut dalam mencari makan. Semut-semut tersebut meninggalkan zat (pheromone) di jalan yang mereka lalui. Algoritma ACO ini merupakan algoritma pencarian berdasarkan probabilitas berbobot, sehingga butir pencarian dengan bobot yang lebih besar akan berakibat memiliki kemungkinan terpilih yang lebih besar pula.

b. Tabu Search

Tabu Search adalah salah satu metode metaheuristik yang dipergunakan untuk memecahkan permasalahan-permasalahan optimasi global. Tabu Search merupakan suatu teknik optimasi yang menggunakan short-term memory untuk menjaga agar proses pencarian tidak terjebak pada nilai optimum lokal. Metode ini menggunakan *Tabu List* untuk menyimpan sekumpulan solusi yang baru saja dievaluasi. Selama proses optimasi, pada setiap iterasi solusi yang akan dievaluasi akan dicocokkan terlebih dahulu dengan isi Tabu List untuk melihat apakah solusitersebut sudah ada pada *Tabu List*. Apabila solusi tersebut sudah ada, maka akan dievaluasi lagi pada iterasi berikutnya. Kemudian bila sudah tidak ada lagi solusi yang menjadi anggota *Tabu List*, maka nilai terbaik yang baru saja diperoleh merupakan solusi yang sebenarnya.

c. Coloring Graph

Teknik pewarnaan graf merupakan salah satu subjek yang menarik dan terkenal dalam bidang *graf*. Teori-teori mengenaiya telah banyak dikembangkan dan berbagai algoritma dengan kelebihan dan kelemahan masing-masing telah dibuat untuk menyelesaikannya. Aplikasi dari teknik ini juga telah banyak diterapkan di berbagai bidang, salah satunya adalah membuat jadwal. Perencanaan jadwal disini khususnya diterapkan pada pekerjaan-pekerjaan atau hal-hal yang saling terkait, misalnya hal-hal yang berlangsung pada waktu yang sama, atau pekerjaan yang menggunakan sumber daya yang sama, dan sebagainya. Teknik pewarnaan graf akan membuat jadwal kerja yang dapat menghasilkan hasil yang maksimum dengan cara yang paling efisien.

5. Contoh pemakaian algoritma genetika

Pada contoh pemakaian algoritma genetika kali ini yaitu menggunakan sebuah

persamaan, dimana persamaan tersebut terdiri dari variabel a,b,c dan d. Dengan persamaan sebagai berikut: $a + 2b + 3c + 4d = 20$. Langkah penyelesaian persamaan terdiri dari beberapa tahap yaitu:

a. Pembentukan kromosom

Pembentukan kromosom ini terdiri dari beberapa gen yaitu a,b,c,d. Variabel a terdiri dari nilai 1-20. Variabel b,c,d terdiri dari nilai 0-10.

b. Inisialisasi

Pada proses ini dilakukan dengan memberikan nilai gen pada tahap awal. Pada tahap ini akan diberikan permisalan sebanyak 6 populasi yaitu:

Kromosom1= [14;07;05;10] = sebagai[a;b;c;d]

Kromosom2= [04;03;10;05] = sebagai[a;b;c;d]

Kromosom3= [12;06;05;06] = sebagai[a;b;c;d]

Kromosom4= [22;03;12;08] = sebagai[a;b;c;d]

Kromosom5= [03;06;05;11] = sebagai[a;b;c;d]

Kromosom6= [22;07;09;013] = sebagai[a;b;c;d]

c. Evaluasi Kromosom

Evaluasi kromosom dilakukan dengan menginput variabel a,b,c,d pada persamaan berikut sebagai fungsi objektif.

$$\text{fungsiobjektif(kromosom)} = (a + 2b + 3c + 4d) - 20$$

- 1) Fungsiobjektif melalui kromosom 1

$$= ((14 + 2*7 + 3*5 + 4*10) - 20)$$

$$= ((14 + 14 + 15 + 40) - 20)$$

$$= (63 - 20) = 43$$
- 2) Fungsiobjektif melalui kromosom 2

$$= ((4 + 2*3 + 3*10 + 4*15) - 20)$$

$$= ((4 + 6 + 30 + 60) - 20)$$

$$= (100 - 20) = 80$$
- 3) Fungsiobjektif melalui kromosom 3

$$= ((12 + 2*6 + 3*5 + 4*6) - 20)$$

$$= ((12 + 12 + 15 + 24) - 20)$$

$$= (63 - 20) = 43$$
- 4) Fungsiobjektif melalui kromosom 4

$$= ((22 + 2*3 + 3*12 + 4*8) - 20)$$

$$= ((22 + 6 + 36 + 32) - 20)$$

$$\begin{aligned}
 &= (96 - 20) = 76 \\
 5) \text{ Fungsiobjektif melalui kromosom 5} &= ((3 + 2 \cdot 6 + 3 \cdot 5 + 4 \cdot 11) - 20) \\
 &= ((3 + 12 + 15 + 44) - 20) \\
 &= (74 - 20) = 54 \\
 6) \text{ Fungsiobjektif melalui kromosom 6} &= ((22 + 2 \cdot 7 + 3 \cdot 9 + 4 \cdot 3) - 20) \\
 &= ((22 + 14 + 27 + 12) - 20) \\
 &= (75 - 20) = 55
 \end{aligned}$$

$$\begin{aligned}
 \text{Sehingga average dari ke enam nilai kromosom yaitu} &= (43+80+43+76+54+55)/6 \\
 &= 351 / 6 = 58.5
 \end{aligned}$$

d. Seleksi kromosom

Tahap selanjutnya yaitu seleksi kromosom dengan fungsi fitness = $(1 / (1 + \text{fungsi_objektif}))$

Sehingga perhitungan menjadi berikut:

$$\begin{aligned}
 1) \text{ fitness1} &= 1 / 44 = 0.0227 \\
 2) \text{ fitness2} &= 1 / 81 = 0.0123 \\
 3) \text{ fitness3} &= 1 / 44 = 0.0227 \\
 4) \text{ fitness4} &= 1 / 77 = 0.0129 \\
 5) \text{ fitness5} &= 1 / 55 = 0.0181 \\
 6) \text{ fitness6} &= 1 / 56 = 0.0178 \\
 7) \text{ total nilai fitness} &= 0.0227 + 0.0123 + 0.0227 + 0.0129 + 0.0181 + 0.0178 \\
 &= 0.1065
 \end{aligned}$$

Langkah selanjutnya mencari kemungkinan atau probabilitas : " **$P[i] = \text{fitness}[i] / \text{total_fitness}$** "

$$\begin{aligned}
 P1 &= 0.0227 / 0.1065 = 0.2131 \\
 P2 &= 0.0123 / 0.1065 = 0.1154 \\
 P3 &= 0.0227 / 0.1065 = 0.2131 \\
 P4 &= 0.0129 / 0.1065 = 0.1211 \\
 P5 &= 0.0181 / 0.1065 = 0.1699 \\
 P6 &= 0.0178 / 0.1065 = 0.1671
 \end{aligned}$$

Langkah selanjutnya yaitu mencari nilai komulatif dengan proses dibawah ini:

$$C1 = 0.2131$$

$$C2 = 0.2131 + 0.1154 = 0.3285$$

$$C3 = 0.2131 + 0.1154 + 0.2131 = 0.5416$$

$$C4 = 0.2131 + 0.1154 + 0.2131 + 0.1211 = 0.6627$$

$$C5 = 0.2131 + 0.1154 + 0.2131 + 0.1211 + 0.1699 = 0.8326$$

$$C6 = 0.2131 + 0.1154 + 0.2131 + 0.1211 + 0.1699 + 0.1671 = 0.9997$$

Langkah selanjutnya yaitu membangkitkan R dalam range 0-1. Jika $R[k] < C[1]$ maka pilih kromosom 1 sebagai induk, selain itu pilih kromosom ke-k sebagai induk dengan syarat $C[k-1] < R < C[k]$. Dengan permisalan

$$R1 = 0.301$$

$$R2 = 0.484$$

$$R3 = 0.209$$

$$R4 = 0.722$$

$$R5 = 0.598$$

$$R6 = 0.401$$

Apabila Angka acak pertama $R[1]$ adalah lebih besar dari $C[1]$ dan lebih kecil daripada $C[2]$ maka pilih Kromosom[2] sebagai kromosom pada populasi baru, sehingga menjadi:

$$\text{Kromosom1} = \text{Kromosom2}$$

$$\text{Kromosom2} = \text{Kromosom2}$$

$$\text{Kromosom3} = \text{Kromosom1}$$

$$\text{Kromosom4} = \text{Kromosom5}$$

$$\text{Kromosom5} = \text{Kromosom2}$$

$$\text{Kromosom6} = \text{Kromosom3}$$

Kromosom baru hasil proses seleksi :

$$\text{Kromosom1} = 14;03;10;05$$

$$\text{Kromosom2} = 14;03;10;05$$

$$\text{Kromosom3} = 14;07;05;10$$

$$\text{Kromosom4} = 03;06;05;11$$

$$\text{Kromosom5} = 14;03;10;05$$

$$\text{Kromosom6} = 12;06;05;06$$

C. Soal Latihan/Tugas

1. Carilah nilai komulatif dari variabel gen berikut ini:

Kromosom1= [16;09;07;12] = sebagai[a;b;c;d]

Kromosom2= [06;05;12;07] = sebagai[a;b;c;d]

Kromosom3= [14;08;07;08] = sebagai[a;b;c;d]

Kromosom4= [24;05;14;10] = sebagai[a;b;c;d]

Kromosom5= [05;08;07;13] = sebagai[a;b;c;d]

Kromosom6= [24;09;11;15] = sebagai[a;b;c;d]

2. Carilah nilai komulatif dari variabel gen berikut ini:

Kromosom1= 18;11;09;14= sebagai[a;b;c;d]

Kromosom2=08;07;14;09= sebagai[a;b;c;d]

Kromosom3= 16;10;09;10 = sebagai[a;b;c;d]

Kromosom4= 26;07;16;12 = sebagai[a;b;c;d]

Kromosom5=07;10;09;15]= sebagai[a;b;c;d]

Kromosom6=26;11;13;17= sebagai[a;b;c;d]

3. Carilah penerapan algoritma genetika dalam kehidupan sehari-hari?

D. Referensi

Basuki, Achmad. 2003. Algoritma Genetika : Suatu Alternatif Penyelesaian
Permasalahan Searching, Optimasi dan Machine Learning. Surabaya : Politeknik
Elektronika Negeri Surabaya PENS – ITS.

Kusumadewi, Sri. 2003. Artificial Intelligence – Teknik dan Aplikasinya. Yogyakarta :
Graha Ilmu