BAB III

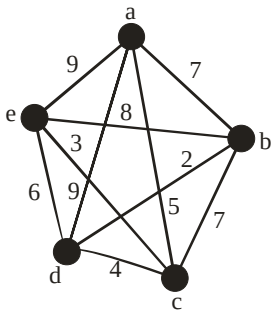
IMPLEMENTASIALGORITMA GENETIK DAN ACS PADA PERMASALAHAN TRAVELLING SALESMAN PROBLEM

3.1 *TRAVELLING SALESMAN PROBLEM*

Sebelum membahas pencarian solusi *Travelling Salesman Problem* menggunakan algoritma genetik dan *ant colony system*, akan terlebih dahulu dibahas mengenai definisi *Travelling Salesman Problem.*

Masalah *Travelling Salesman Problem (*TSP*)* muncul dari permasalahan pedagang keliling. Jika diketahui sejumlah kota dengan ketentuan satu kota dengan yang lainnya saling berhubungan, diketahui jarak yang identik dengan biaya yang harus dikeluarkan, bagaimana cara mengunjungi semua kota tepat satu kali dan kembali ke kota awal dengan biaya termurah.

Dalam model matematika permasalahan ini dijabarkan menjadisebuah graf *G*dengan*n*buah simpul, dan graf *G*merupakan graf lengkap berbobot tak berarah. Dengan solusi berupa jalur terpendek dari simpul *a* melalui seluruh simpul pada graf *G*tepat sekali dan kembali ke simpul *a*. Dengan kata lain akan dicari siklus Hamilton pada graf *G*, dengan bobot terkecil. Pada permasalahan ini terdapat $\frac{(n-1)!}{2}$solusi yang mungkin. Sebagai contoh diilustrasikan pada gambar (3.1), (3.2) dan tabel (3.1).



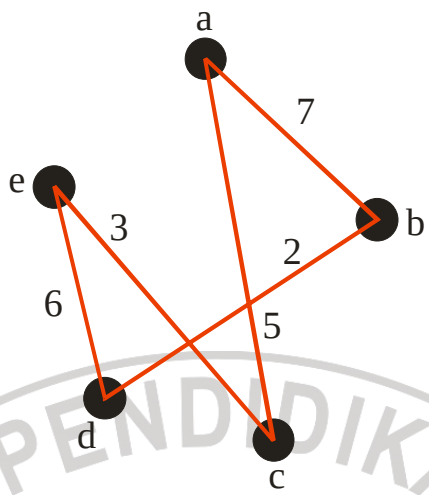
Gambar 3.1 Graf Lengkap Berbobot Tak Berarah *G*

Pada gambar (3.1) menunjukkan sebuah graf lengkap dengan 5 (lima) buah simpul dan setiap sisinya memiliki bobot. Bobot pada graf tersebut dapat direpresentasikan pada tabel (3.1).

Tabel 3.1

	A	**B**	**C**	**D**	**E**
a	**0**	**7**	**5**	**9**	**9**
b	**7**	**0**	**7**	**2**	**8**
c	**5**	**7**	**0**	**4**	**3**
d	**9**	**2**	**4**	**0**	**6**
e	**9**	**8**	**3**	**6**	**0**

Dari data bobot pada graf tersebut akan dicari siklus Hamilton dengan bobot terkecil dan diambil *a* sebagai simpul awal. Dengan menggunakan algoritma *greedy* siklus Hamilton dengan bobot terkecil yang terbentuk adalah *a-b-d-e-c-a.* Siklus Hamilton tersebut diilustrasikan pada gambar 3.2.



Gambar 3.2 Siklus Hamilton dari Graf *G*

3.2 ALGORITMA GENETIK

Algoritma yang pertama kali dikembangkan oleh John Holland dari Universitas Michigan pada tahun 1975 ini adalah algoritma pencarian yang didasarkan atas mekanisme evolusi biologis.Algortima Genetik merupakan simulasi dari proses evolusi Darwin dan operasi genetik atas kromosom.“Keuntungan penggunaan algoritma genetik sangat jelas terlihat darikemudahan implementasi dan kemampuannya untuk menemukan solusi yangbagus (bisa diterima) secara cepat untuk masalah-masalah berdimensi tinggi” (Abdullah, 2008:10)

Ada beberapa istilah yang penting dalam algoritma genetika, diantaranya adalah:

- Gen : Mewakili elemen-elemen yang ada dalamsebuah solusi.

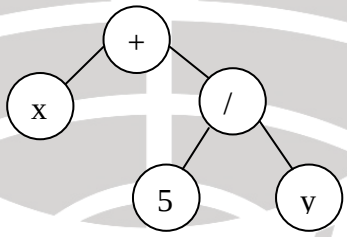
- Populasi : Sejumlah solusi yang mungkin. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan generasi.
- Generasi : Iterasi yang dilakukan untuk menentukan populasi berikutnya.
- Kromosom : Individu yang terdapat dalam satu populasi. Kromosom ini merupakan solusi yang masih berbentuk simbol.
- *Allela* : Nilai yang berada dalam gen
- *Locus* : Letak suatu gen berada dalam suatu kromosom
- Fungsi *Fitness* : Alat ukur dalam dalam proses evaluasi yang dilalui kromosom. Nilai fitness akan menunjukkan kualitas dari suatu kromosom dalam populasi tersebut.
- *Offsprings* : Anak (generasi berikutnya) yang terbentuk dari gabungan 2 kromosom generasi sekarang yang bertindak sebagai induk (parent) dengan menggunakan operator penyilangan (*crossover*) maupun operator mutasi.

Dalam algoritma genetik solusi yang mungkin dapat direpresentasikan ke dalam beberapa bentuk, seperti yang telah dijelaskan oleh Abdullah (2008:15-16) yaitu:

1) *Binary Encoding*
   Encoding jenis ini sering digunakan. Kromosom dari *binary encoding* iniberupa kumpulan dari nilai biner 0 dan 1.
   Contohnya:
   Chromosome1 1101100100110110
   Chromosome2 1101011000011110

Dalam Binary Encoding memungkinkan didapatkan kromosom dengan nilai*allela* yang kecil, tetapi kekurangannya tidak dapat digunakan untukbeberapa permasalahan dan terkadang diperlukan adanya koreksi setelahproses *crossover* dan mutasi. Salah satu permasalahan yang menggunakan*encoding* adalah menghitung nilai maksimal dari suatu fungsi.

2) *Permutation Encoding*
   Kromosom dari *permutation encoding* ini berupa kumpulan dari nilai *integer*yang mewakili suatu posisi dalam sebuah urutan. Biasanya digunakan padapermasalahan TSP (*Travelling Salesman Problem*).
   Contohnya:
   *Chromosome 1*1 4 7 9 6 3 5 0 2 8
   *Chromosome 2*9 3 2 5 8 1 6 0 4 7
3) *Value Encoding*
   Kromosom dari *value encoding* berupa kumpulan dari suatu nilai, yang bisaberupa macam-macam nilai sesuai dengan permasalahan yang dihadapi,seperti bilangan *real*, *char* atau objek yang lain. *Encoding* ini merupakanpilihan yang bagus untuk beberapa permasalahan khusus, biasanyadiperlukan metode khusus untuk memproses *crossover* dan mutasinya sesuaidengan permasalahan yang dihadapi.
   Contohnya:
   Chromosome 1 A B E D B C A E D D
   *Chromosome 2*N W W N E S S W N N
4) *Tree Encoding*
   Tree Encoding biasanya digunakan pada *genetic programming*. Kromosomyang digunakan berupa sebuah *tree* dari beberapa objek, seperti fungsi atau*command* pada *genetic programming*.



Gambar 3.3 Contoh Kromosom pada *Tree Encoding*

Dibangkitkan populasi awal yaitu sejumlah solusi yang mungkin dari permasalahan sesuai dengan *encoding* yang sesuai dengan permasalahan. Populasi awal ini dibangkitkan secara acak. Dalam algoritma genetik, solusi yang mungkin ini disebut kromosom. Kromosom ini terdiri dari gen-gen yang memiliki nilai yaitu *allela*.

Setelah itu setiap kromosom tersebut dievaluasi nilai *fitness*-nya, hal ini diperlukan untuk langkah berikutnya, yaitu seleksi kromosom. Pada permasalahan TSP nilai *fitness* bisa didapatkan dengan cara menjumlahkan jarak antara titik yang terdapat pada kromosom.

Semakin baik nilai *fitness* sebuah kromosom maka semakin besar probabilitas kromosom tersebut terpilih sebagai induk untuk tahap berikutnya yaitu *crossover* (kawin silang).Karena dengan induk yang memiliki nilai *fitness* baik diharapkan menghasilkan anak yang baik pula. Pada permasalahan TSP yang diinginkan adalah kromosom dengan *fitness* yang lebih kecil akan memiliki probabilitas menjadi induk lebih besar oleh karena itu diperlukan nilai invers *fitness* dari setiap kromosom.

$$Q(i)=1/fitness(i)$$

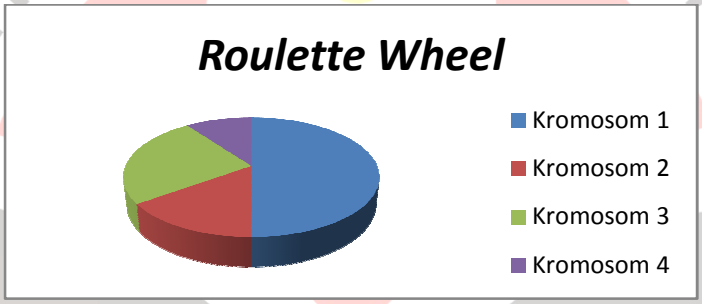
Kemudian dari nilai invers tersebut dicari nilai probabilitas dengan cara:

$$P(i) = Q(i) / \sum_{i=1}^{jumlah\ kromosom} Q(i)$$

Setelah didapatkan nilai probabilitasnya dihitung nilai komulatif probabilitasnya untuk setiap kromosom. Nilai komulatif inilah yang dijadikan acuan untuk melakukan proses seleksi. “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 *roulette wheel*”(Abdullah, 2008:18).

1) *Roulette wheel*

Metode *roulette whell* merupakan metode yang biasa digunakan dalam proses seleksi kromosom pada algoritma genetik.Proses *roulette wheel* dapat diilustrasikan sebagai roda yang menampung semua nilai probabilitas *fitness* dari setiap kromosom. Semakin besar nilai probabilitasnya semakin besar pula area dalam roda tersebut sehingga ketika roda diputar *fitness* yang paling baik dan memiliki nilai probabilitas paling besar lebih besar kemungkinannya untuk terpilih.



Gambar 3.4*Roulette Wheel*

Wilayah kromosom pada roda tersebut direpresentasikan dengan nilai komulatif probabilitas *fitness* (*C(i)*). Nilai tersebut dapat dihitung dengan cara:

Probabilitas kromosom 1= 50%=0,5

Probabilitas kromosom 2=15%=0,15 nilai komulatif : 0,5+0,15=0,65

Probabilitas kromosom 3=25%=0,25 nilai komulatif : 0,65+0,25=0,9

Probabilitas kromosom 4=10%=0,1 nilai komulatif : 0,9+0,1=01

Setelah itu metode ini bekerja dengan cara membangkitkan bilangan acak (*R(i)*)antara [0-1] sebanyak jumlah kromosom. Kemudian kromosom yang akan menjadi induk jika memenuhi syarat:

R(i)<*C(i)* untuk kromosom 1 dan

C(i-1)<R(i)<C(i) untuk kromosom lainnya. (Persamaan 3.1)

2) *Steady State*

Metode ini bekerja dengan cara mempertahankan beberapa kromosom yang memiliki nilai *fitness* terbaik dan mengganti kromosom lainnya dengan *offspring* yang baru.

3) Tournament

Dalam metode seleksi *tournament* sejumlah n individu dipilih secara acakdan kemudian menentukan *fitness*-nya. Kebanyakan metode seleksi ini digunakanpada *binary*, dimana hanya dua individu yang dipilih.

4) *Rank*

Metode ini mirip dengan metode *Roulette Wheel* namun metode ini tidak menggunakan nilai komulatif probabilitas untuk melakukan seleksi. Pada metode ini setiap kromosom diberikan nilai baru sesuai dengan nilai *fitness*-nya. Kromosom dengan nilai *fitness* terburuk diberi nilai 1, kromosom yang lebih baik diberi nilai 2 begitu seterusnya hingga kromosom dengan nilai *fitness* terbaik diberi nilai N (jumlah kromosom dalam populasi).

Setelah terpilih kromosom yang akan menjadi induk maka diantara induk tersebut dilakukan *crossover*. Pada proses *crossover* terdapat parameter probabilitas *crossover*, di mana nilainya berkisar antara [0,1], kemudian bangkitkan bilangan acak antara [0,1). Proses *crossover* akan dilakukan jika bilangan acak bernilai kurang dari probabilitas *crossover*, namun pada kasus ini parameter probabilitas *crossover* bernilai 1 sehingga semua proses *crossover* dilakukan. Proses *crossover* sendiri bekerja dengan cara menukar salah satu gen

di induk pertama dengan gen di induk lainnya yang belum terdapat pada kromosom awal dengan membangkitkan bilangan acak [1,n-1] sebanyak jumlah induk dengan *n* adalah jumlah kota untuk memilih gen mana yang akan ditukar. Namun pertukaran ini tetap melihat urutan gen yang terdapat pada kromosom asal.

Setelah dihasilkan kromosom baru dari hasil *crossover* maka proses selanjutnya adalah mutasi. Proses ini bekerja dengan cara membangkitkan sejumlah bilangan acak antara 1-panjang total gen untuk menunjukkan nomor gen yang akan dimutasi.Banyaknya gen yang dimutasi berdasarkan parameter probabilitas mutasi ($\tilde{n}m$), yaitu

$$\tilde{n}m * panjang\ total\ gen \qquad \text{(Persamaan 3.2)}$$

Panjang total gen sendiri adalah jumlah gen dalam satu kromosom dikali banyaknya kromosom dengan tidak mengikutsertakan kota awal dan kota akhir. Gen yang ditunjuk oleh bilangan acak tadi bertukar posisi dengan gen sesudahnya.

Setelah semua proses selesai maka akan dievaluasi solusi sementara dari satu iterasi yang telah menghasilkan populasi baru, jika dianggap telah optimal maka iterasi berhenti sebaliknya jika dianggap belum optimal maka proses diulangi dimulai dari proses menghitung nilai *fitness*.

3.3 *ANT COLONY SYSTEM* (ACS)

ACS merupakan pengembangan terbaru dari *Ant Colony Optimization (ACO).* Berbeda dengan algoritma genetik yang didasarkan mekanisme evolusi

biologis,ACS didasarkan kepada prilaku koloni semut dalam mendapatkan makanannya. Koloni semut ini memanfaatkan zat yang bernama *pheromone* untuk saling berkomunikasi, mereka meninggalkan *pheromone* pada rute-rute yang mereka lalui*pheromone* yang dimanfaatkan untuk menemukan rute terpendek “Semutmampu mengindera lingkungannya yang kompleks untuk mencarimakanan dan kemudian kembali ke sarangnya dengan meninggalkan zat*pheromone* pada rute-rute yang mereka lalui” (Leksono, 2009:22).

Pheromone adalah zat kimia yang berasal dari kelenjar endokrindan digunakan oleh makhluk hidup untuk mengenali sesama jenis,individu lain,kelompok, dan untuk membantu proses reproduksi. Berbedadengan hormon, *pheromone* menyebar ke luar tubuh dan hanya dapatmempengaruhi dan dikenali oleh individu lain yang sejenis (Leksono, 2009:22).

Seiring waktu, *pheromone* akan menguapdan akan mengurangi kekuatan daya tariknya. Lebih cepat setiap semutpulang pergi melalui rute tersebut, maka *pheromone* yang menguap lebihsedikit. Begitu pula sebaliknya jika semut lebih lama pulang pergi melaluirute tersebut, maka *pheromone* yang menguap lebih banyak.

Koloni semut dapat menemukan rute terpendek antara sarangdan sumber makanan berdasarkan *pheromone* pada lintasan yang telahdilalui. Semakin banyak semut yang melalui suatu lintasan, maka akansemakin jelas bekas *pheromone*-nya. Hal ini akan menyebabkan lintasanyang dilalui semut dalam jumlah sedikit, semakin lama akan semakinberkurang kepadatan semut yang melewatinya, atau bahkan akan tidakdilewati sama sekali. Sebaliknya lintasan

yang dilalui semut dalam jumlahbanyak, semakin lama akan semakin bertambah kepadatan semut yangmelewatinya, atau bahkan semua semut akan melalui lintasan tersebut.

ACS memanfaatkan semut sebagai agen untuk menyelesaikan algoritma ini. Pertama kali hitung nilai *tour* terbaik awal (C^{nn})dengan metode *nearest neigberhoud heuristic*. Kemudian sejumlah *m* semut ditempatkan di sejumlah titik awal secara acak. Karena pada permasalahan TSP titik awal telah ditentukan maka penempatan semut diacak di titik lainnya selain titik awal. Kemudian semut-semut tersebut bergerak ke titik berikutnya masing-masing. Semut-semut tersebut menentukan titik mana yang akan dikunjungi berikutnya dengan menggunakan aturan transisi status.

$$s = \begin{cases} \arg\max\{\tau_{ru}[\eta_{ru}]^{\beta}\}, u \epsilon J_r^k & jika\ q \leq q_0 \\ \frac{[\tau_{rs}]^{\alpha}[\eta_{rs}]^{\beta}}{\sum_{u \epsilon J_r^k}[\tau_{ru}]^{\alpha}[\eta_{ru}]^{\beta}} & jika\ tidak \end{cases}$$

(Persamaan 3.3)

Dimana τ_{rs}adalah jumlah *pheromone* yangterdapat antara sisi *r* dengan *s* sedangkan η_{rs} adalah invers jarak antara titik *r* dengan titik *s* sedangkan *u* adalah himpunan titik-titik yang belum dilalui semut. Simbol lainnya adalah *q*merupakan bilangan random [0,1], di mana $q_0(0 \leq q_0 \geq 1)$ adalah sebuah parameter pembanding bilangan random.Parameter lainnya adalah α yaitu parameter yang mengontrol bobot relatif dan β yaitu parameter pengendali jarak di mana α,β>0.

Setiap semut memodifikasi jumlah *pheromone* pada setiap sisi yang dilaluinya dengan menggunakan *Local Pheromone Update*. Setelah semua semut menyelesaikan *tour*-nya maka dihitung jarak *tour* yang telah diselesaikan. *Tour*

yang memiliki jarak tependek akan dipilih sebagai solusi sementara. Sedangkan untuk penguapan jumlah *pheromone* selama sisi tidak dilalui oleh semut dihitung dengan menggunakan *Global Pheromone Update*.Iterasi ini diulang terus sampai memenuhi kriteria berhenti tertentu yang telah ditetapkan sebelumnya.

1. *Local Pheromone Update*

$$\tau_{rs} \leftarrow (1-\xi)\tau_{rs} + \xi\tau_0$$

$$\tau_0 = \frac{1}{nC^{nn}} \quad \text{(Persamaan 3.4)}$$

2. *Global Pheromone Update*

$$\tau_{rs} \leftarrow (1-\rho)\tau_{rs} + \rho\Delta\tau_{rs}^{bs}$$

$$\Delta\tau_{rs}^{bs} = \begin{cases} \frac{1}{C^{bs}}, & jika\ (r,s) \epsilon\ lintasan\ terbaik\ keseluruhan \\ 0, & jika\ tidak \end{cases}$$

(Persamaan 3.5)

Di mana lintasan terbaik keseluruhan adalah lintasan yang terpilih oleh aturan transisi status. Sedangkan ρ adalah parameter penguapan global, yang memiliki nilai $0<\rho<1$, ξ adalah parameter penguapan lokal dengan nilai $0<\xi<1$, dan C^{bs} adalah panjang dari lintasan terbaik keseluruhan yang didapat dari sebuah semut.