

BAB II

KAJIAN PUSTAKA

Pada bab kajian pustaka berikut ini akan dibahas beberapa materi yang meliputi graf, permasalahan optimasi, model matematika dari objek wisata di Yogyakarta, dan algoritma genetika *single point crossover*, materi tersebut akan dijadikan landasan dalam bab selanjutnya.

A. Teori Graf

1. Istilah Dalam Graf

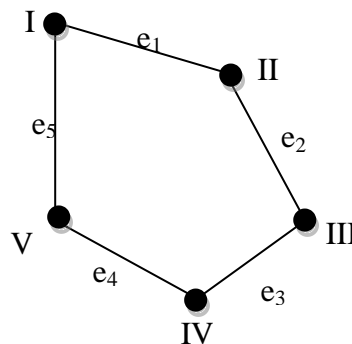
Sebuah graf G yang terdiri atas himpunan tak kosong dari objek-objek yang disebut simpul dan daftar pasangan tak berurut dari elemen-elemen tersebut, yang disebut rusuk. Himpunan simpul-simpul dari graf G disebut simpul, dinyatakan dengan $V(G)$ dan himpunan rusuk-rusuknya disebut himpunan rusuk G , dinyatakan dengan $E(G)$. Jika k dan l adalah simpul dari G , maka sebuah rusuk kl atau lk dikatakan menghubungkan k dan l (Robin J. Wilson & John J. Watkin, 1990: 10).

Suatu graf (Mardiyono, 1996: 1) dapat dipandang sebagai kumpulan dari titik yang disebut simpul dan rusuk yang menghubungkan 2 simpul yang disebut dengan sisi. Sisi yang menghubungkan suatu simpul dengan simpul itu sendiri dinamakan gelang atau *loop*, sedangkan dua sisi yang menghubungkan dua buah simpul yang sama dinamakan sisi ganda.

Menurut Chartrand dan Lesniak (1984: 4), Graf G adalah pasangan himpunan (V, E) dengan V adalah himpunan tidak kosong dan berhingga dari

objek-objek yang disebut simpul dan B adalah himpunan penghubung simpul-simpul berbeda di G dinotasikan dengan $E(G)$. Graf G dapat dinotasikan dengan $E(G)$. Graf G dapat dinotasikan $G = (A, B)$, dengan $V = \{v_1, v_2, v_3, \dots, v_n\}$, dan $E = \{e_1, e_2, e_3, \dots, e_m\}$ $E = \{e_1, e_2, e_3, \dots, e_m\}$.

Graf $G(A, B)$, adalah kumpulan atau pasangan dua himpunan. Himpunan V yang elemennya disebut simpul atau titik, atau vertex, atau point, atau node, sedangkan Himpunan E yang merupakan pasangan tak terurut dari simpul, disebut ruas atau rusuk, atau sisi, atau edge, atau line. (Robin J. Wilson & John J. Watkin, 1990: 8) Banyaknya simpul disebut order Graf G , sedangkan banyaknya sisi disebut ukuran Graf G



Gambar 2.1 Graf (G)

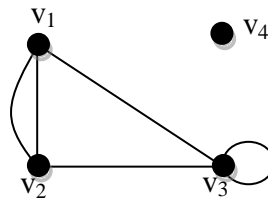
Pada Gambar di atas adalah graf dengan :

$$V = \{I, II, III, IV, V\},$$

$$E = \{(I, II), (II, III), (III, IV), (IV, V), (V, I)\}$$

$$= \{e_1, e_2, e_3, e_4, e_5\}$$

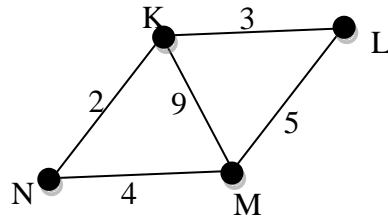
Menurut Mardiono (1996: 12) derajat simpul v_i adalah banyaknya ujung sisi yang hadir pada simpul v_i . Derajat minimum (terkecil) dan derajat maksimum (terbesar) pada graf G dengan himpunan simpul $V(G)$ berturut-turut dilambangkan dengan $\delta(G)$ dan $\Delta(G)$. Berikut contoh derajat simpul pada graf G .



Gambar 2.2 Derajat simpul graf F

Pada gambar 2.2 diketahui bahwa $d(v_1) = d(v_2) = 3$, karena banyaknya ujung sisi yang hadir pada simpul v_1 dan v_2 masing-masing sebanyak 3 buah. Untuk $d(v_3) = 4$, karena pada ujung sisi yang hadir pada v_3 sebanyak 4 buah (sebuah *loop* mempunyai derajat simpul 2) sedangkan pada v_4 karena tidak ada sisi yang hadir di simpul itu maka $d(v_4) = 0$. Sehingga graf G di atas mempunyai $\delta(G) = 0$ dan $\Delta(G) = 4$. Jika $d(v_i) = d$, untuk setiap $v_i \in V(G)$ atau $\delta(G) = \Delta(G) = d$, maka dapat dikatakan G adalah graf teratur berderajat d , disingkat teratur- d .

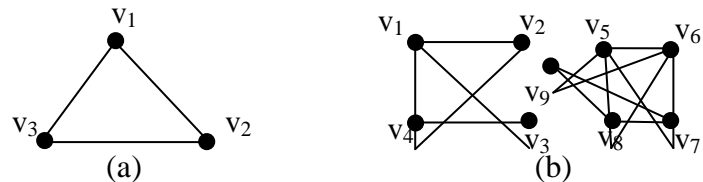
Graf berbobot adalah graf yang setiap ruasnya memiliki nilai/bobot berupa bilangan non negatif (Dwi Yuni Astuti: 12). Bobot pada graf biasanya dinotasikan dengan w_{ij} dimana i dan j sebagai simpul yang berpasangan yang membentuk rusuk.



Gambar 2.3 Graf (H) Berbobot

2. Keterhubungan

Suatu graf G dikatakan terhubung jika terdapat sepasang simpul v_i dan v_j terdapat lintasan antara v_i dan v_j pada graf G . Graf yang tidak terhubung dinamakan graf tak-terhubung (*disconnected graph*).

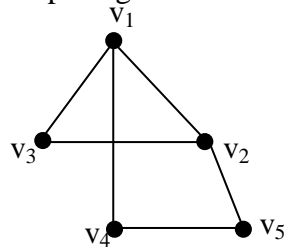


Gambar 2.4 (a) gambar graf (K) terhubung, (b) graf (L) tak-terhubung

Pada gambar 2.4 (a) merupakan graf terhubung, karena terdapat lintasan yang menghubungkan setiap pasang simpul pada graf tersebut. Sedangkan untuk gambar 2.4 (b) merupakan graf tak-terhubung karena tidak terdapat lintasan yang menghubungkan v_3 dengan v_8 maupun v_2 dengan v_5 .

Keterhubungan suatu graf G dinotasikan dengan $k(G)$. Keterhubungan $k(G)$ adalah banyaknya simpul minimum yang dapat menghilangkan hasil keterhubungan graf atau simpul tunggal pada graf K_1 . Dapat dikatakan graf G memiliki k keterhubungan jika $k(G) \geq k$. Sebagaimana ditunjukkan pada gambar 2.4 (Nizhizeki dan Rahman, 2004: 22).

Dua buah simpul v_i dan v_j dikatakan terhubung jika terdapat lintasan dari v_i ke v_j . Misalkan ditunjukkan pada gambar berikut ini (Munir, 2010: 371).



Gambar 2.5 simpul terhubung dari Graf N

a. Jalan (*Walk*)

Sebuah perjalanan dengan panjang k pada sebuah graf G adalah rangkaian terurut dari k rusuk pada graf G dengan bentuk:

$$uv, vw, wx, \dots, yz ;$$

Walk tersebut dinyatakan dengan uv, vw, wx, \dots, yz atau dengan kata lain *walk* antara u sampai z (Robin J. Wilson & John J. Watkin, 1990 : 34). Contoh *walk* pada graf N (pada gambar 2.5) adalah $(v_2, v_1, v_3, v_1, v_4, v_5, v_2)$.

b. Jejak (*Trail*)

Jejak adalah *walk* dengan semua rusuk dalam barisan adalah berbeda (Munir, 2009: 370). Contoh *trail* pada graf N adalah $(v_3, v_2, v_1, v_4, v_5, v_2)$.

c. Lintasan (*Path*)

Jika seluruh rusuk (tidak harus seluruh simpul) pada sebuah trayek berbeda, maka trayek tersebut disebut trail (jejak). Sedangkan jika simpul-simpulnya berbeda jejak tersebut disebut lintasan (Robin J. Wilson & John J. Watkin, 1990 : 35). Contoh Lintasan pada graf N adalah $(v_3, v_1, v_2, v_5, v_4)$.

d. Sikel

Trayek tertutup pada graf G adalah sebuah rangkaian terurut rusuk-rusuk G dalam bentuk:

$$kl, lm, mn, \dots, pq, qk;$$

Jika seluruh rusuknya berbeda, maka trayek tersebut disebut trail tertutup.

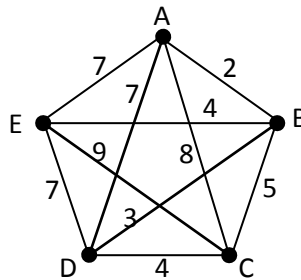
Jika rusuk k, l, m, n, \dots, p, q seluruhnya berbeda, maka trail tersebut disebut sikel (Robin J. Wilson & John J. Watkin, 1990: 35). Contoh pada graf N adalah $(v_2, v_3, v_1, v_4, v_5, v_2)$.

B. Travelling Salesman Problem

Travelling salesman problem (TSP) merupakan suatu masalah yang mudah dideskripsikan namun sulit untuk diselesaikan, yaitu masalah bagaimana menentukan jarak terpendek dalam perjalanan melewati titik-titik tertentu di mana satu titik harus dilalui satu kali dan hanya boleh dilalui satu kali saja dan perjalanan harus berakhir dengan kembali ke titik pertama (Hoffman dan Wolfe, 1985: 1-16). Misalkan $d(a,b)$ merupakan jarak dari kota a ke kota b dan seorang salesman ingin melakukan perjalanan dengan biaya yang paling minimum, dengan total biaya adalah jumlah bobot dari setiap rusuk yang dilalui atau lintasannya. Masalah TSP merupakan bentuk khusus dari masalah transportasi. TSP pertama kali diperkenalkan oleh Rand pada tahun 1948. TSP dikenal sebagai salah satu permasalahan optimasi klasik yang berat untuk dipecahkan secara konvensional. Penyelesaian eksak terhadap persoalan ini akan melibatkan algoritma yang mengharuskan untuk mencari kemungkinan semua solusi yang ada.

TSP memerlukan data berupa jarak, waktu, ataupun biaya, disesuaikan dengan kebutuhan. Dalam melakukan suatu perjalanan TSP semua tempat yang harus dilewati semuanya dikunjungi tepat satu kali. Tetapi pada skripsi ini sales diasumsikan sebagai wisatawan yang ingin berkunjung ke semua tempat wisata yang ada di Yogyakarta, dan kota diasumsikan sebagai objek wisata di Yogyakarta.

Dalam penentuan jalur yang paling optimum dengan memodelkannya sebagai graf tak berarah dan graf berbobot. Berikut ini adalah contoh dari TSP yang direpresentasikan ke dalam graf B .



Gambar 2.6 Graf B Berbobot

Dilihar dari Graf B berbobot tersebut dapat diperoleh lintasan tertutup dari simpul A kembali lagi ke A . Terdapat 24 lintasan tertutup pada graf B yaitu $A-B-C-D-E-A$, $A-C-B-D-E-A$, $A-D-B-C-E-A$, ..., dan $A-E-C-D-B-A$ sehingga banyaknya lintasan (s) dapat diperoleh dengan

$$s = (n-1)! \quad (2.1)$$

Selain graf berbobot graf B juga graf tak berarah, maka $w(A,B) = w(B,A)$ sehingga banyaknya lintasan berubah menjadi

$$s = \frac{(n-1)!}{2} . \quad (2,2)$$

Karena lintasan

$$A-B-C-D-E-A = A-C-B-D-E-A,$$

$$A-B-C-D-E-A = A-D-B-C-E-A,$$

dst. sampai $A-E-C-D-B-A = A-B-D-C-E-A$.

jadi banyaknya rute setelah menggunakan rumus 2.2 adalah 12 lintasan. Dalam skripsi ini salesman tersebut dianalogikan sebagai wisatawan yang akan mengunjungi objek-objek wisata yang membentuk jalur wisata.

C. Algoritma

Algoritma adalah urutan langkah-langkah yang dinyatakan dengan jelas dan tidak rancu untuk memecahkan suatu masalah (jika ada pemecahannya) dalam rentang waktu tertentu. Artinya, setiap langkah harus dapat dikerjakan dan mempunyai efek tertentu (Wahid, 2004, hal:2). Pengertian untuk algoritma agak mirip kepada kata resep, proses, metode, teknik, prosedur, rutin, kecuali bahwa kata “algoritma” itu sendiri memiliki arti sesuatu yang sedikit berbeda ada 5 komponen utama dalam algoritma yaitu *finiteness*, *definiteness*, *input*, *output* dan *effectiveness* (Knuth, 1973, hal: 4).

1. *Fiteness*

Sebuah algoritma harus selalu berakhir setelah sejumlah langkah berhingga.

2. *Definiteness*

Setiap langkah dari sebuah algoritma harus didefinisikan secara tepat, tindakan yang di muat harus dengan teliti dan sudah jelas ditentukan untuk setiap keadaan.

3. *Input*

Sebuah algoritma memiliki nol atau lebih masukan, sebagai contoh, banyaknya masukan diberikan di awal sebelum algoritma mulai.

4. *Output*

Sebuah algoritma memiliki satu atau lebih keluaran, sebagai contoh, banyaknya keluaran memiliki sebuah hubungan yang ditentukan terhadap masukan.

5. *Effectiveness*

Pada umumnya sebuah algoritma juga diharapkan untuk efektif.

D. Algoritma Genetika

Algoritma Genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi suatu organisme. 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 terdapat 4 kondisi yang mempengaruhi proses evaluasi, yaitu (Sri Kusuma Dewi & Hari Purnomo,2005:231):

1. kemampuan organisme melakukan reproduksi
2. keberadaan populasi organisme yang melakukan reproduksi
3. keberagaman organisme dalam suatu populasi
4. perbedaan kemampuan untuk bertahan hidup.

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 (disebut juga generasi), populasi secara keseluruhan akan lebih banyak memuat organisme yang fit (Sri Kusumadewi dan Hari Purnomo, 2005:231).

Algoritma genetika dapat diaplikasikan untuk menemukan solusi untuk menemukan masalah optimasi. Dalam hal ini adalah untuk menemukan jarak terpendek dari objek wisata di Yogyakarta. Keuntungan menggunakan algoritma genetika terlihat dari hasil yang bagus dan cepat untuk menyelesaikan masalah dengan dimensi tinggi atau jumlah titik yang sangat banyak.

1. Struktur Umum Algoritma Genetika

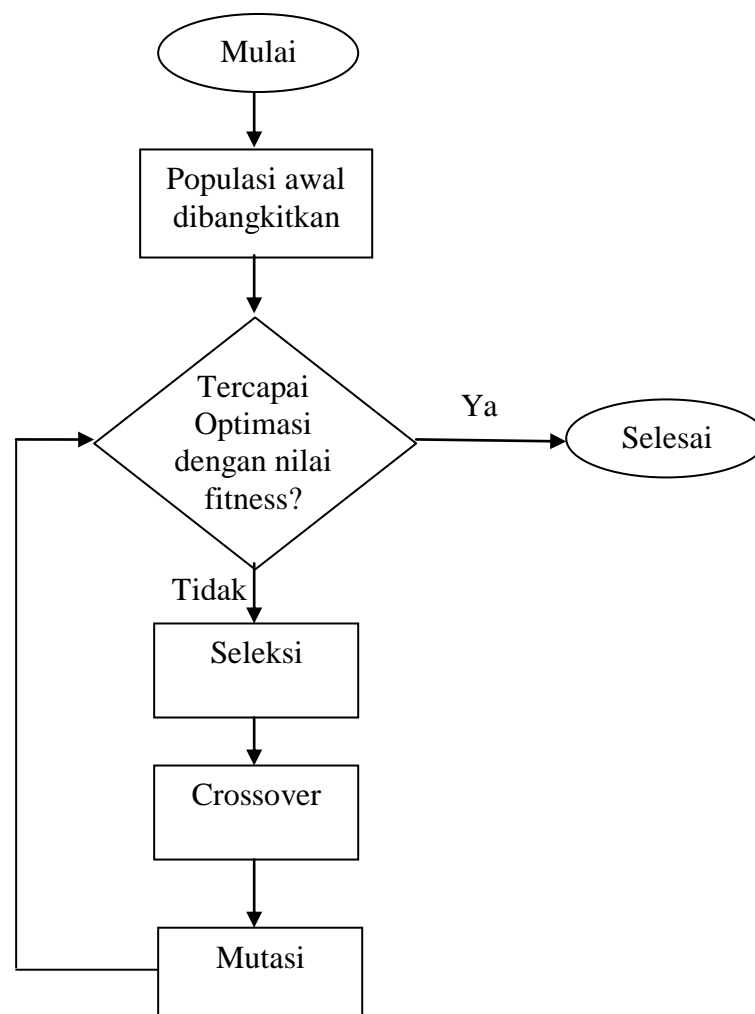
Dalam Struktur umum algoritma genetika, perhitungan dilakukan secara bersama-sama untuk beberapa solusi yang mungkin disebut dengan Populasi. Individu yang terdapat di dalam suatu populasi disebut dengan kromosom. Populasi awal ditentukan atau dipilih secara *random* atau acak, populasi setelahnya merupakan hasil dari evolusi kromosom-kromosomnya dengan melalui proses iterasi yang disebut dengan generasi. Proses evaluasi dilakukan dengan menggunakan fungsi fitness.

Dalam suatu populasi, kualitas dari suatu kromosom dapat dilihat dari nilai fitness. Generasi selanjutnya dapat disebut dengan anak (*off-spring*) yang terbentuk dari penggabungan 2 kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*). Selain operator penyilangan suatu kromosom dapat juga dimodifikasi dengan menggunakan

operator mutasi. Populasi dari generasi baru dibentuk dengan cara menyeleksi nilai fitness dari kromosom induk dan nilai fitness populasi anak serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam populasi) tetap. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik.

2. Skema Algoritma Genetika

Tahapan-tahapan algoritma genetika secara umum dapat dilihat pada skema berikut ini.



Gambar 2.7 Skema Algoritma Genetika

Pada skema diatas akan tercapai optimasi berdasarkan nilai fitnessnya. Jadi ketika populasi awal dibangkitkan ada kemungkinan bahwa hasil tersebut sudah tercapai optimasi. Oleh karena itu tidak perlu melalui iterasi selanjutnya.

3. Komponen-komponen utama Algoritma Genetika (*Single Point Crossover*)

Untuk Menyelesaikan Masalah TSP

a. Teknik Penyandian

Ada bermacam-macam teknik Penyandian yang dapat dilakukan dalam algoritma genetika. Beberapa teknik-teknik Penyandian itu antara lain adalah *binary encoding*, *permutation encoding*, *value encoding* serta *tree encoding*. Teknik Penyandian yang digunakan pada skripsi ini adalah *permutation encoding*.

Pada *permutation encoding*, kromosom-kromosom adalah kumpulan angka yang mewakili posisi dalam sebuah rangkaian. Pada penyelesaian masalah TSP, kromosom mewakili urutan kota-kota sebagai jalur distribusi. Jadi apabila satu kromosom berbentuk sebagai berikut $O_1 = (k_1, k_2, k_3, \dots, k_n)$ berarti jalur perjalanan bergerak dari kota bernomor 1 ke 2 dst hingga kota ke- k_n .

b. Membangkitkan Populasi Awal

Membangkitkan populasi awal adalah membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu (Kusumadewi, 2003:102). Dengan menggunakan *random generator* akan dipilih suatu individu dengan memilih secara acak antara bilangan 1 sampai dengan jumlah gen hingga terbentuk suatu kromosom. Kromosom-kromosom yang sudah dibangkitkan akan menjadi sebuah populasi. Populasi inilah yang disebut dengan ppopulasi awal.

Untuk masalah TSP, kromosom tersebut berupa jalur yang mungkin dilewati oleh sales. Banyaknya rute yang mungkin diperoleh dapat dihitung menggunakan rumus permutasi (2.2) berikut ini

$$s = \frac{(n-1)!}{2} . \quad (2.2)$$

c. Evaluasi Nilai *Fitness*

Setiap kromosom pada populasi dihitung nilai *fitness*nya berdasarkan fungsi *fitness*. Nilai *fitness* suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti.

Pada proses penyeleksian digunakan suatu parameter yang disebut kesesuaian atau *fitness*. *Fitness* digunakan untuk menentukan seberapa baik kromosom akan bertahan hidup. Penentuan berapa besar nilai *fitness* suatu kromosom berdasarkan fungsi *fitness* yang didefinisikan tersendiri. Untuk menghitung nilai *fitness* dalam Skripsi ini, yang dibutuhkan untuk inisialisasi data awal adalah jarak antar objek wisata di DIY. Pada masalah optimasi fungsi *fitness* yang digunakan adalah

$$f = \frac{1}{x} . \quad (2.3)$$

Jika x merupakan nilai individu, maka semakin kecil nilai x , nilai *fitness* akan semakin besar. Tetapi akan menjadi kendala jika nilai x bernilai 0. Itu akan mengakibatkan nilai *fitness*nya menjadi tak berhingga. Solusinya dengan

menjumlah nilai x dengan sebuah bilangan yang sangat kecil sehingga nilai *fitness*nya menjadi

$$f = \frac{1}{(x+k)} \quad , \quad (2.4)$$

dengan k adalah bilangan yang sangat kecil.

d. Seleksi

Proses seleksi adalah proses yang memegang peranan penting dalam algoritma genetika. Proses seleksi ini digunakan agar hanya kromosom-kromosom yang berkualitas yang dapat melanjutkan peranannya dalam proses algoritma genetika berikutnya. Seleksi akan menentukan individu-individu yang akan dipilih untuk dilakukan rekombinasi dan bagaimana offspring terbentuk. Langkah awal dalam seleksi adalah menentukan nilai fitness. Masing-masing individu akan menerima kemungkinan reproduksi yang tergantung pada nilai objektifnya sendiri terhadap nilai objek dari semua individu tersebut. Nilai fitness tersebut akan digunakan pada tahap-tahap berikutnya.

Proses penyeleksian yang digunakan disini adalah *roulette-wheel selection*. Metode *roulette-wheel selection* sangat mudah diimplementasikan dalam pemrograman. Metode seleksi ini merupakan metode yang sederhana, dan sering juga dikenal dengan nama *stochastic sampling with replacement*. Berikut ini adalah langkah-langkah dari *roulette-wheel selection* (Kusumadewi, 2003:105):

- 1) menghitung nilai fitness dari masing-masing individu (f_i dimana I adalah individu ke-1 s/d ke- n),
- 2) menghitung total fitness semua individu dengan rumus (2.4),

- 3) menghitung probabilitas setiap individu tersebut. Nilai *fitness* setiap individu diperoleh dari rumus 2.4 kemudian nilai *fitness* tersebut dijumlah menghasilkan nilai total *fitness*. Peluang dicari dengan membagi nilai *fitness* masing-masing individu dengan nilai total *fitness* semua individu. Rumus untuk Peluang seleksi didapatkan dengan rumus (2.5)

$$Ps[i] = \frac{f(i)}{\sum f(i)} \quad . \quad (2.5)$$

Dengan $Ps[i]$ adalah peluang individu ke- i dan $f(i)$ adalah nilai *fitness* individu ke- i , $i = 1, 2, 3, 4, \dots, n$;

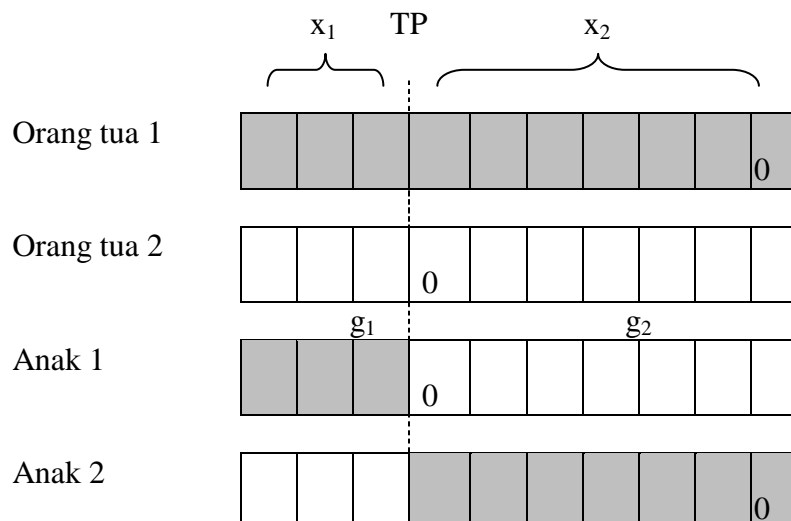
- 4) membangkitkan bilangan random berdasarkan banyaknya populasi pada generasi,
- 5) menentukan individu yang terpilih sebagai induk berdasarkan letak bilangan random yang dihasilkan, misalkan dalam satu populasi terdapat 7 individu dengan masing-masing nilai fitness $f(1)=0.02$, $f(2)=0.15$, $f(3)=0.21$, $f(4)=0.06$, $f(5)=0.09$, $f(6)=0.18$, dan $f(7)=0.01$, kemudian nilai total *fitness* dari ketujuh individu tersebut adalah 0,72. Peluang seleksi masing-masing individu dihitung menggunakan rumus (2.5) sehingga diperoleh $Ps[1]=0.027$, $Ps[2]=0.208$, $Ps[3]=0.292$, $Ps[4]=0.083$, $Ps[5]=0.125$, $Ps[6]=0.25$, dan $Ps[7]=0.014$. Berikutnya akan dicari peluang kumulatif $C[i]$ dari $Ps[i]$ didapatkan $C[1]=0.028$, $C[2]=0.236$, $C[3]=0.528$, $C[4]=0.611$, $C[5]=0.736$, $C[6]=0.986$, $C[7]=1$. Untuk mencari individu yang akan digunakan sebagai induk maka ditentukan bilangan acak $[0,1]$.

Individu yang terpilih sebagai induk dapat diketahui sesuai letak bilangan acak yang akan dihasilkan dalam peluang kumulatif individu.

e. Rekombinasi Dengan *Single-point Crossover*

Tahap ini merupakan salah satu tahap paling penting dalam algoritma genetika yaitu penyilangan (*crossover*). Kromosom akan mendekati solusi yang bagus dengan melalui proses penyilangan dua buah kromosom.

Dalam suatu populasi sangat kecil, suatu kromosom dengan gen-gen yang mengarah ke solusi akan cepat menyebar ke kromosom-kromosom lainnya. Untuk mengatasi masalah ini digunakan suatu aturan bahwa penyilangan hanya bisa dilakukan dengan suatu Peluang tertentu p_c . Artinya, penyilangan bisa dilakukan hanya jika suatu bilangan $\text{random}[0,1)$ yang dibangkitkan kurang dari p_c yang ditentukan. Pada umumnya p_c diatur mendekati 1, misalnya 0,8 (Suyanto, 2005:14). Jika peluang suatu individu memungkinkan untuk pindah silang maka akan ditentukan nilai TP yaitu Titik perpotongan untuk membagi gen individu yang terpilih yang kemudian akan dipindah silang. Gambaan tentang pindah silang adalah sebagai berikut:

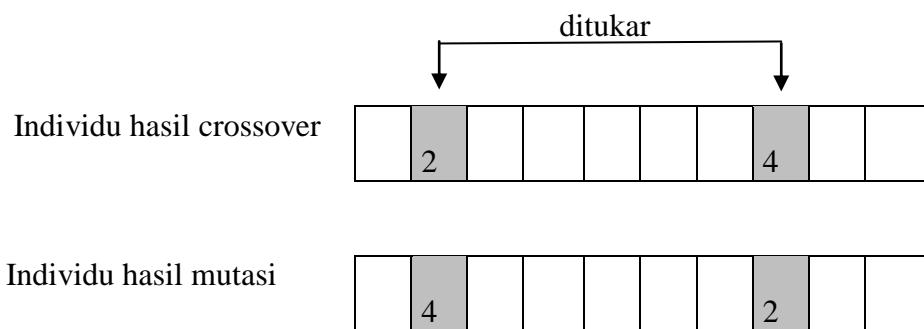


Gambar 2.8 Contoh *Single-point Crossover*

f. Mutasi

Setelah melalui tahap rekombinasi, pada *offspring* dapat dilakukan mutasi. Pada *offspring* dimutasi dengan menambahkan nilai random yang sangat kecil, dengan peluang yang rendah. Peluang mutasi (P_m) adalah peluang dari jumlah semua gen pada populasi yang termutasi. Peluang mutasi adalah untuk mengendalikan banyaknya gen baru yang akan dievaluasi kembali. Peran mutasi adalah untuk menggantikan gen yang hilang dari populasi karena proses seleksi. Proses mutasi ini dilakukan pada individu hasil dari *crossover*.

Teknik mutasi yang digunakan adalah teknik *swapping mutation*. Teknik ini diawali dengan memilih dua bilangan acak kemudian gen yang berada pada posisi bilangan acak pertama ditukar dengan gen yang berada pada bilangan acak kedua. Dalam probabilitas tertentu (Suyanto, 2005:24).



Gambar 2.9 Contoh Mutasi

g. Elitism

Elitism adalah proses untuk mempertahankan supaya individu yang mempunyai nilai *fitness* terbesar tetap ada selama evolusi (Kusumadewi, 2003: 112). Pemilihan individu yang akan di seleksi akan dilakukan secara random sehingga individu dengan nilai *fitness* tertinggi tidak selalu terpilih, karena bisa terjadi penurunan nilai *fitness* karena proses pindah silang.

h. Pembentukan Populasi Baru

Pembentukan populasi baru untuk membuat populasi yang berbeda dengan populasi awal. Pembentukan populasi baru ini diperoleh dari keturunan-keturunan baru hasil mutasi ditambah dengan individu terbaik setelah dipertahankan oleh proses *elitism*.

Setelah diperoleh populasi baru kemudian akan diulang kembali langkah-langkah algoritma genetika sampai diperoleh nilai *fitness* yang konvergen.