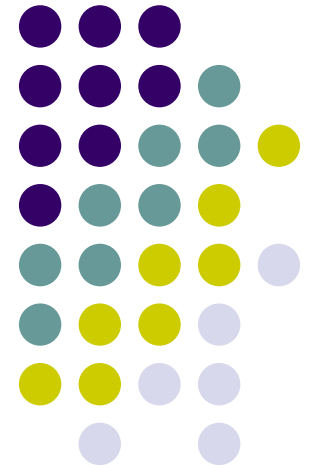


Strategi Menggunakan Algoritma Genetika

Achmad Basuki

Politeknik Elektronika Negeri Surabaya PENS-ITS

2003

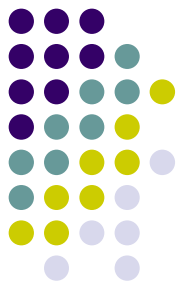




Agenda

- Memahami permasalahan apa yang membutuhkan algoritma genetika
- Strategi mendefinisikan Individu
- Strategi menentukan bentuk kromosom
- Strategi menentukan fitness
- Strategi menentukan seleksi
- Strategi menentukan proses cross-over
- Strategi menentukan mutasi

Mengapa Menggunakan Algoritma Genetika



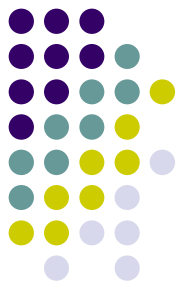
- Algoritma genetika digunakan untuk menyelesaikan permasalahan searching dan optimasi yang mempunyai kompleksitas tinggi yang banyak terjadi dalam dynamic programming seperti TSP dan Knapsack Problem.
- Algoritma genetika dapat menghindari keadaan lokal optimum yang baik.
- Pengembangan model parallel computing dapat dengan mudah dilakukan.

Ciri-Ciri Permasalahan Yang Membutuhkan Algoritma Genetika



- Mempunyai fungsi tujuan optimalisasi non linear dengan banyak kendala yang juga non linear.
- Mempunyai kemungkinan solusi yang jumlahnya tak berhingga.
- Membutuhkan solusi “real-time” dalam arti solusi bisa didapatkan dengan cepat sehingga dapat diimplementasi untuk permasalahan yang mempunyai perubahan yang cepat seperti optimasi pada pembebanan kanal pada komunikasi seluler.
- Mempunyai multi-objective dan multi-criteria, sehingga diperlukan solusi yang dapat secara bijak diterima oleh semua pihak.

Strategi Menentukan Individu



- Individu adalah suatu komposisi nilai yang menyatakan solusi dari suatu permasalahan.
- Permasalahan optimasi adalah suatu permasalahan yang mempunyai banyak solusi dan harus bisa ditentukan solusi mana yang dikatakan optimal.
- Berdasarkan definisi individu algoritma genetika dibedakan menjadi 2 macam yaitu *static genetic algorithm* (ukuran individu sama untuk satu populasi) dan *dynamic genetic algorithm* (ukuran individu tidak sama untuk satu populasi).

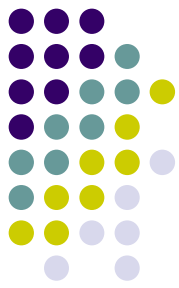
Definisi Individu Untuk Menentukan Nilai Maksimal Fungsi $F(x,y,z)$



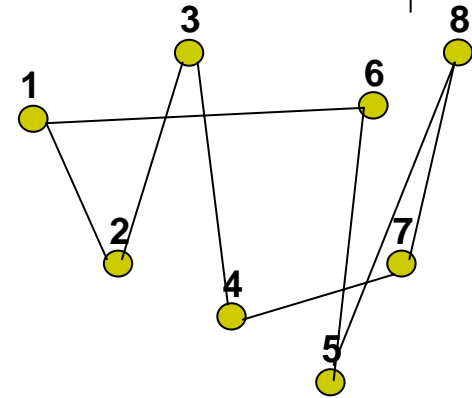
- Nilai fungsi $F(x,y,z)$ sangat tergantung pada nilai (x,y,z) . Artinya solusi yang dicari adalah nilai (x,y,z) yang menyebabkan nilai $F(x,y,z)$ maksimum.
- Individu didefinisikan sebagai nilai (x,y,z) dan dituliskan dengan:

$$S = \{(x, y, z) \mid x, y, z \in R\}$$

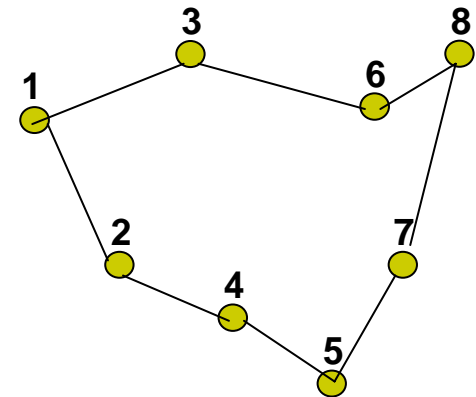
Definisi Individu Untuk TSP



- TSP (Traveling Salesman Problem) adalah suatu permasalahan dimana seorang sales harus mengunjungi N kota dengan jarak yang paling pendek, dengan syarat satu kota hanya dikunjungi satu kali.
- Solusi TSP adalah jalur yang melewati semua kota dan jaraknya paling pendek.
- Individu untuk TSP didefinisikan sebagai jalur atau urutan nomor kota yang dikunjungi. Misalkan untuk 8 kota salah satu jalur yang mungkin adalah : **1-2-3-4-7-8-5-6** atau **1-2-4-5-7-8-6-3**
- Perhatikan bahwa tidak ada nomor yang sama, dan setiap nomor 1 s/d ada.



1-2-3-4-7-8-5-6

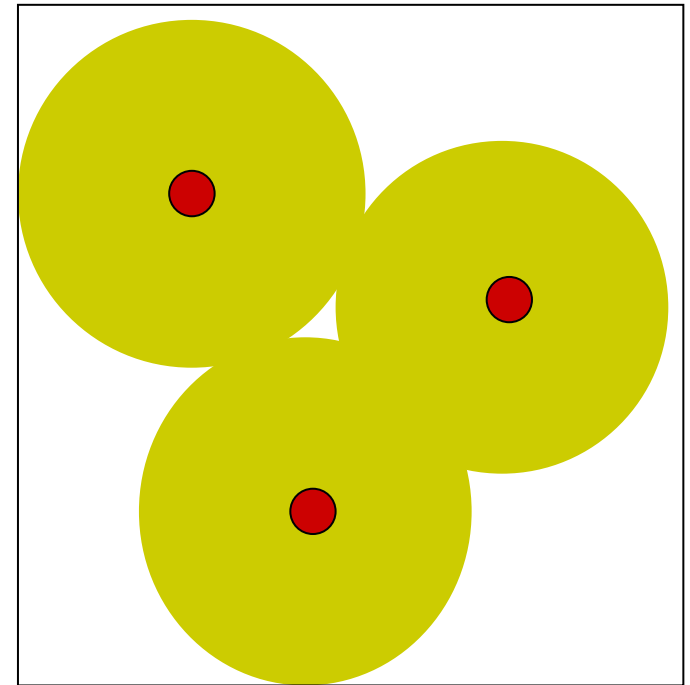


1-2-4-5-7-8-6-3

Definisi Individu Untuk Permasalahan Penentuan Lokasi RBS yang Optimal

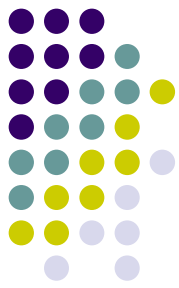


- Dalam satu area yang berukuran $N_1 \times N_2$, akan dipasang N buah RBS. Permasalahannya adalah dimana lokasi RBS-RBS tersebut agar covering areanya maksimal atau blank areanya minimal.
- Solusi dari permasalahan ini adalah lokasi N buah RBS yang dapat memaksimalkan covering area.
- Individu didefinisikan sebagai lokasi N buah RBS dan ditulis dengan:



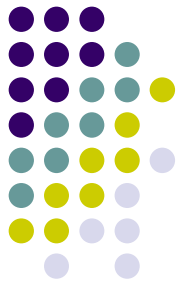
$$S = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_N, y_N) \mid 0 < x_k \leq N_1, 0 < y_k \leq N_2\}$$

Macam-Macam Bentuk Kromosom



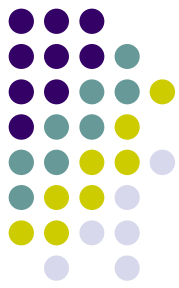
- Kromosom biner, yaitu kromosom yang disusun dari gen-gen yang bernilai 0 dan 1. Kromosom ini adalah model standar dalam algoritma genetika.
- Kromosom float, yaitu kromosom yang disusun dari gen-gen dengan nilai pecahan. Gen integer dapat digolongkan dalam bentuk kromosom ini.
- Kromosom string, yaitu kromosom yang disusun dari gen-gen yang bernilai string (simbol).
- Kromosom Kombinatorial, yaitu kromosom yang disusun dari gen-gen yang dinilai dari urutannya.

Bentuk Kromosom Biner



- Kromosom biner merupakan model yang sederhana dengan tingkat keberhasilan yang tinggi.
- Permasalahan dengan parameter yang jumlah dan range nilainya tertentu sebaiknya menggunakan bentuk kromosom biner, seperti mencari nilai optimal fungsi transedental.
- Jumlah gen pada kromosom biner menunjukkan tingkat ketelitian yang diharapkan.

Bentuk Kromosom Biner Untuk Mencari Nilai Maksimal Fungsi $F(x,y,z)$



- Solusi dari permasalahan menentukan nilai maksimal fungsi $F(x,y,z)$ adalah nilai kombinasi (x,y,z) . Misalkan $x=0.4$, $y=0.36$ dan $z=0.7$ ditulis dengan $[0.4 \ 0.36 \ 0.7]$
- Setiap nilai dikonversikan menjadi nilai biner m bit, bila setiap nilai $X=(x,y,z)$ mempunyai range $[\min(X) \ \max(X)]$ maka m bit 0 menyatakan nilai min dan m bit 1 menyatakan nilai max, sehingga konversinya adalah:

$$X = (\max(X) - \min(X)) \frac{\text{biner}(X)}{2^m - 1}$$

- Misalkan semua dibatasi $[0 \ 1]$ maka model konversi 8 bit per-gen digambarkan sebagai berikut:

00000000	00000000	00000000	→	(0, 0, 0)
00000100	01001001	11000000	→	(4/255, 73/255, 192/255)
11111111	11111111	11111111	→	(1, 1, 1)



Implementasi Dengan Matlab

- Bila dalam populasi terdapat Npop individu, maka untuk membangkitkan individu dapat dilakukan dengan menggunakan perintah:

```
%Membangkitkan individu  
Npop=10;  
individu=floor(rand(Npop,24)*2);
```

- Hasil dari perintah tersebut adalah matrik biner yang ukurannya 10x24.

Konversi Biner Ke Float (Dekoding)



- Sebelum menghitung nilai fungsi $F(x,y,z)$, perlu suatu proses dekoding yang mengkonversikan kombinasi biner menjadi (x,y,z) .
- Implementasinya dalam Matlab adalah sebagai berikut:

```
%Konversi biner ke decimal
Xmin=0;
Xmax=1;

xi=individu(Npop,1:8);
xs=num2str(xi);
x=bin2dec(xs)*(Xmax-Xmin)/255;

yi=individu(Npop,9:16);
ys=num2str(yi);
y=bin2dec(ys)*(Xmax-Xmin)/255;

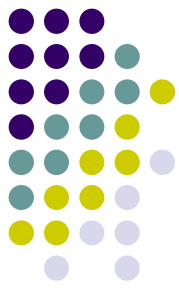
zi=individu(Npop,17:24);
zs=num2str(zi);
z=bin2dec(zs)*(Xmax-Xmin)/255;
```



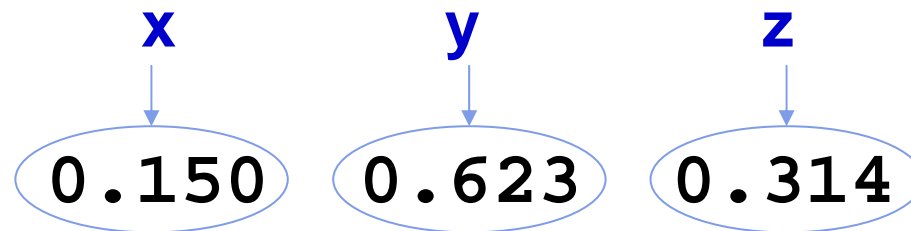
Bentuk Kromosom Float

- Kromosom float merupakan model yang jumlah parameteranya banyak. Tingkat keberhasilan dari bentuk kromosom ini rendah dalam kecepatan (jumlah generasi).
- Model cross-over dan mutasi pada kromosom float ini sangat berbeda dengan model cross-over dan mutasi pada kromosom biner. Sehingga perlu strategi khusus didalam menentukan model cross-over dan mutasi.
- Range nilai [min max] menjadi tidak penting dalam bentuk kromosom float ini.

Bentuk Kromosom Float Untuk Mencari Nilai Maksimal Fungsi $F(x,y,z)$



- Solusi dalam permasalahan nilai optimal fungsi $F(x,y,z)$ adalah nilai (x,y,z) sehingga kromosomnya adalah kombinasi nilai x , y dan z sebagai berikut:



- Dalam Matlab, penentuan individu dapat dituliskan sebagai berikut:

```
Npop=10;           %jumlah populasi
%Membangkitkan individu
individu=rand(Npop,3)
```

Konversi Individu Ke Solusi (Dekoding)



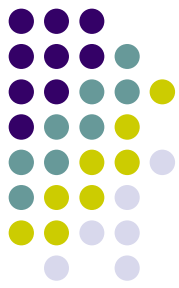
- Individu didefinisikan :

$$\textit{individu}=[x \ y \ z]$$

- Untuk mendapatkan nilai x,y,z tinggal menuliskan lokasi kolom array dari individu.

```
%Dekoding  
x=individu(:,1);  
y=individu(:,2);  
z=individu(:,3);
```

Strategi Menentukan Nilai Fitness



- Nilai fitness merupakan suatu ukuran baik tidaknya suatu solusi yang dinyatakan sebagai satu individu, atau dengan kata lain nilai fitness menyatakan nilai dari fungsi tujuan.
- Algoritma genetika mempunyai tujuan untuk memaksimalkan nilai fitness atau mencari nilai fitness maksimal.
- Untuk permasalahan minimalisasi, nilai fitness adalah inversi dari nilai minimal yang diharapkan. Proses inversi dapat dilakukan dengan:

$$Fitness = A - F(X) \quad \text{atau} \quad Fitness = \frac{A}{F(X) + \varepsilon}$$

Dimana A = konstanta yang ditentukan

X = individu (kromosom)

ε = bilangan kecil yang ditentukan untuk menghindari pembagi nol atau $F(X)=0$

Nilai Fitness Untuk Mencari Nilai Maksimal $F(x,y,z)$



- Tujuannya adalah mencari nilai maksimal $F(x,y,z)$, maka nilai fitnessnya adalah nilai fungsi $F(x,y,z)$ itu sendiri.
- Implementasi pada matlab untuk fungsi:

$$F(x, y, z) = x^2 \cos(z) + y^2 \sin(z) - xy$$

```
%Menghitung Fitness  
fitness=x.^2.*cos(z)+y.^2.*sin(z)+x.*y;
```

Nilai Fitness Untuk Mencari Nilai Minimal $F(x,y,z)$



- Tujuannya adalah mencari nilai minimal $F(x,y,z)$, maka nilai fitnessnya adalah nilai inversi dari fungsi $F(x,y,z)$ itu sendiri.
- Implementasi pada matlab untuk fungsi:

$$F(x, y, z) = x^2 \cos(z) + y^2 \sin(z) - xy$$

```
%Menghitung Fitness  
F=x.^2.*cos(z)+y.^2.*sin(z)+x.*y;  
Fitness=1/(F+1);
```

$$Fitness = \frac{A}{F(X) + \epsilon}$$

$$Fitness = A - F(X)$$

```
%Menghitung Fitness  
A=10;  
F=x.^2.*cos(z)+y.^2.*sin(z)+x.*y;  
Fitness=A-F;
```

Strategi Menentukan Seleksi



- Seleksi dengan Mesin Roulette, model seleksi ini merupakan model yang paling mudah dan paling besar variansinya. Munculnya individu superior sering terjadi pada model ini, sehingga perlu strategi lain untuk menangani hal ini.
- Seleksi dengan Turnamen, model ini merupakan model yang variansinya kecil sehingga kemungkinan munculnya individu superior dapat dikurangi.


Implementasi Seleksi Untuk Mencari Nilai Maksimal $F(x,y,z)$



```
%Seleksi
induk=individu;
Jfitness=sum(fitness); %Total fitness
%prosentase fitness
prosenfitness=floor(100*fitness/Jfitness);
%Pembuatan rolet
k=0;
for i=1:Npop
    for j=1:prosenfitness(i)
        k=k+1;
        rolet(k)=i;
    end
end
End
%Pemilihan
for i=1:Npop
    r=floor(k*rand)+1;
    pilih=rolet(k);
    induk(i,:)=individu(i,:);
end
```

Rank

- Salah satu cara untuk mengurangi kemungkinan terjadinya individu superior yang menyebabkan terjebak ke optimum lokal adalah menggunakan Rank
- Pembagian area pada piringan mesin roulette ditentukan oleh ranking dari individu. Individu dengan fitness paling kecil diberi area 1, setelahnya 2, 3 dan seterusnya, individu yang fitnessnya tertinggi diberi area N.

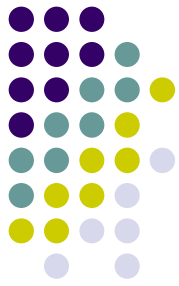


Individu 1 : fitness = 0.154
Individu 2 : fitness = 0.500
Individu 3 : fitness = 0.141
Individu 4 : fitness = 0.220

Individu 1 : pfitness = 2
Individu 2 : pfitness = 4
Individu 3 : pfitness = 1
Individu 4 : pfitness = 3

Individu 1 : prosentase area = 20
Individu 2 : prosentase area = 40
Individu 3 : prosentase area = 10
Individu 4 : prosentase area = 30

Perbandingan Tanpa Rank Dan Dengan Rank



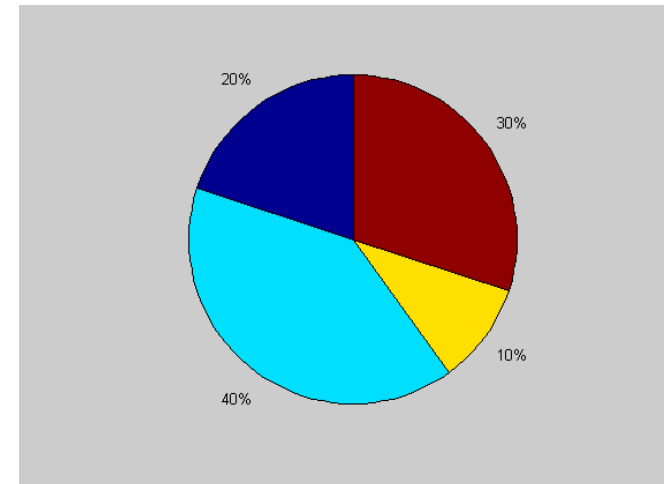
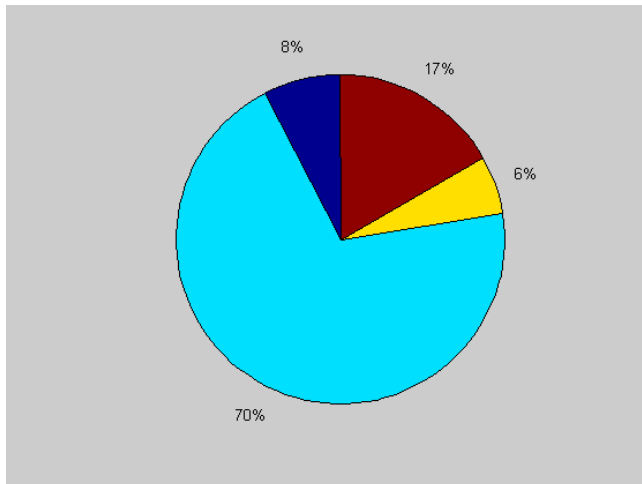
tanpa rank

Individu 1 : fitness = 0.054
Individu 2 : fitness = 0.500
Individu 3 : fitness = 0.041
Individu 4 : fitness = 0.120

dengan rank

Individu 1 : prosentase area = 7.55
Individu 2 : prosentase area = 68.93
Individu 3 : prosentase area = 5.73
Individu 4 : prosentase area = 16.78

Individu 1 : prosentase area = 20
Individu 2 : prosentase area = 40
Individu 3 : prosentase area = 10
Individu 4 : prosentase area = 30



Strategi Penentuan Cross Over

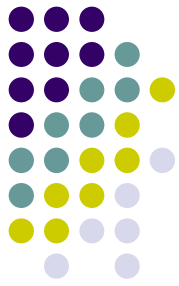


- Penentuan Cross-Over harus menjamin bahwa solusi masih berada di ruang solusi.

$$individu(k) \in S$$

- Cross over untuk kromosom biner dilakukan dengan menggunakan pertukaran gen antar induk.
- Cross over untuk kromosom float dilakukan dengan pertukaran gen atau pertukaran aritmatika antar induk.
- Probabilitas cross over yang baik berada pada kisaran 0.5 sampai dengan 0.95

Permasalahan Cross-Over Pada TSP



INDUK	1	3	4	2	7	8	5	6
	3	6	5	7	4	2	1	8
	1	3	4	7	4	2	5	6
	3	6	5	2	7	8	1	8
	1	3	4	4	5	6	5	6
	3	6	5	5	6	4	1	8

terpilih $r1=4$ dan $r2=6$

Pertukaran gen secara langsung, menghasilkan individu yang keluar dari ruang solusi

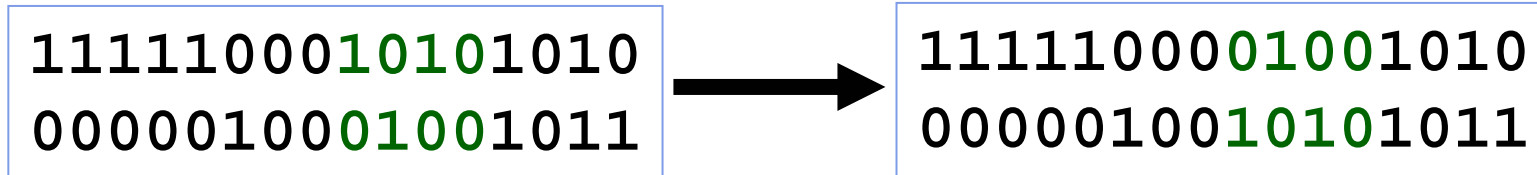
Pertukaran gen secara aritmatika, menghasilkan individu yang keluar dari ruang solusi

Diperlukan teknik Cross-Over yang lain agar dapat menjamin solusi tetap berada di dalam ruang solusi

Cross Over Untuk Mencari Nilai Maksimal Fungsi $F(x,y,z)$



- Bila menggunakan kromosom biner, cross over dapat dilakukan dengan pertukaran gen secara langsung dari kedua induk.



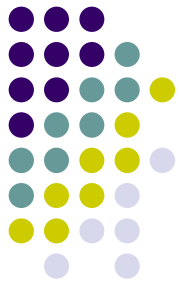
- Bila menggunakan kromosom float, cross over dapat dilakukan dengan pertukaran aritmatika.

$$a(n, k) = r.a(n, k) + (1 - r).a(n, k + 1)$$

$$a(n, k + 1) = (1 - r).a(n, k) + r.a(n, k + 1)$$

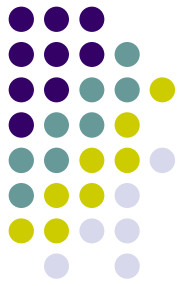
Dimana r adalah bilangan acak 0 s/d 1

Implementasi Cross Over Dengan Kromosom Biner



```
%Cross Over
anak=induk;
m=floor(Npop/2);
probCO=0.85;      % Probabilitas Cross Over
for i=1:m
    p=rand;
    if p<probCO
        r1=floor(24*rand)+1;
        r2=floor((24-r1)*rand)+r1;
        anak(2*i-1,r1:r2)=induk(2*i,r1:r2);
        anak(2*i,r1:r2)=induk(2*i-1,r1:r2);
    end
end
end
```

Implementasi Arithmetic Cross Over Dengan Kromosom Float



```
%Cross Over
anak=induk;
m=floor(Npop/2);
probCO=0.85;      % Probabilitas Cross Over
for i=1:m
    p=rand;
    if p<probCO
        r1=floor(24*rand)+1;
        r2=floor((24-r1)*rand)+r1;
        r=rand;
        anak(2*i-1,r1:r2)=r*induk(2*i-1,r1:r2)+(1-r)*induk(2*i,r1:r2);
        anak(2*i,r1:r2)=(1-r)*induk(2*i-1,r1:r2)+r*induk(2*i,r1:r2);
    end
end
```



Strategi Menentukan Mutasi

- Pada kromosom biner, mutasi dilakukan dengan mengubah gen biner 0 menjadi 1 dan 1 menjadi 0.

$$a(k) = 1 - a(k)$$

- Pada kromosom float ada dua macam mutasi yang banyak dilakukan yaitu *random mutation* dan *shift mutation*.
- Random mutation adalah mengganti gen yang termutasi dengan nilai acak.

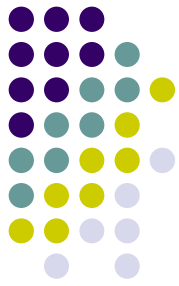
$$a(k) = \text{random}$$

- Shift mutation adalah menggeser nilai gen termutasi sebesar ε , dimana ε adalah bilangan kecil yang ditentukan.

$$a(k) = a(k) + \varepsilon$$

- Probabilitas mutasi yang baik berada pada kisaran 0 sampai dengan 0.3. Probabilitas mutasi yang terlalu kecil menyebabkan terjebak dalam optimum lokal, dan probabilitas mutasi yang terlalu besar menyebabkan konvergensi sulit didapatkan.

Implementasi Mutasi Untuk Kromosom Biner



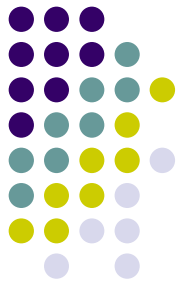
```
%Mutasi
probMut=0.1; %Probabilitas mutasi
for i=1:Npop
    p=rand;
    if p<probMut
        r=floor(rand*24)+1;
        anak(i,r)=1-anak(i,r);
    end
end
%offspring
individu=anak;
```

Implementasi Mutasi Acak Untuk Kromosom Float



```
%Mutasi
probMut=0.1; %Probabilitas mutasi
for i=1:Npop
    p=rand;
    if p<probMut
        r=floor(rand*24)+1;
        anak(i,r)=rand;
    end
end
%offspring
individu=anak;
```

Implementasi Mutasi Shift Untuk Kromosom Float



```
%Mutasi
probMut=0.1; %Probabilitas mutasi
for i=1:Npop
    p=rand;
    if p<probMut
        r=floor(rand*24)+1;
        anak(i,r)=anak(i,r)+0.2*rand-0.1;
    end
end
%offspring
individu=anak;
```