

Kecerdasan Buatan

Studi Kasus Algoritma Genetika

Oleh Politeknik Elektronika Negeri Surabaya
2021



Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

Konten

- Algoritma Genetika Untuk Mencari Nilai Maksimal Fungsi $F(x)=e^{-2x}.\sin(3x)$
- Algoritma Genetika Untuk Mencari Kata Secara Acak
- Traveling Salesman Problem

Tujuan Instruksi Umum

Mahasiswa memahami filosofi Kecerdasan Buatan dan mampu menerapkan beberapa metode Kecerdasan Komputasional dalam menyelesaikan sebuah permasalahan, baik secara individu maupun berkelompok/kerjasama tim.

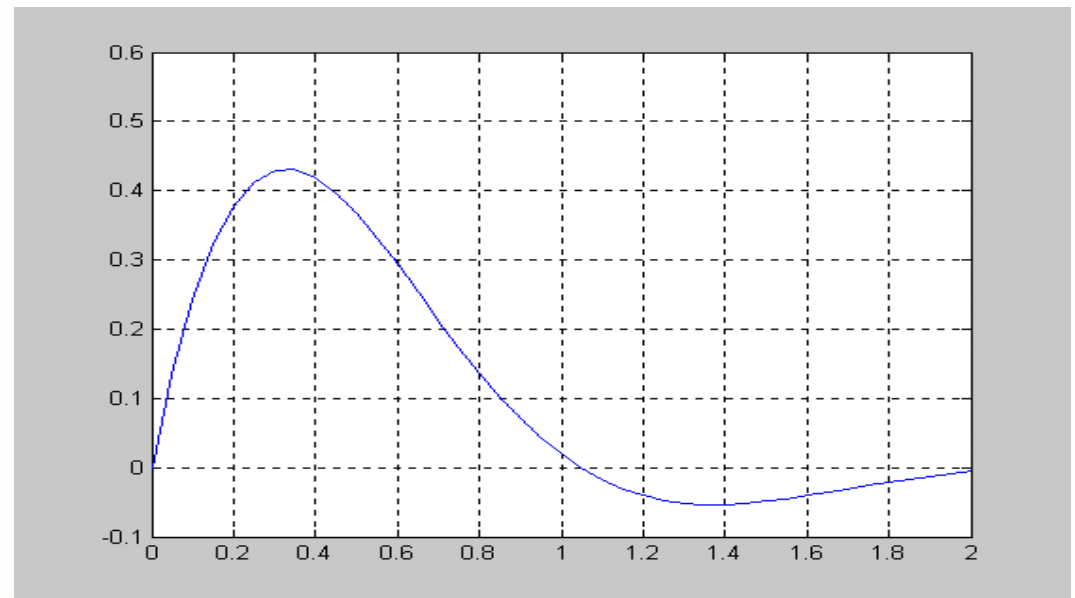
Tujuan Instruksi Khusus

- Mengetahui konsep Algoritma Genetika
- Mengetahui proses dalam Algoritma Genetika
- Mengetahui penerapan Algoritma Genetika

Algoritma Genetika Untuk Mencari Nilai Maksimal Fungsi

$$F(x)=e^{-2x}.\sin(3x)$$

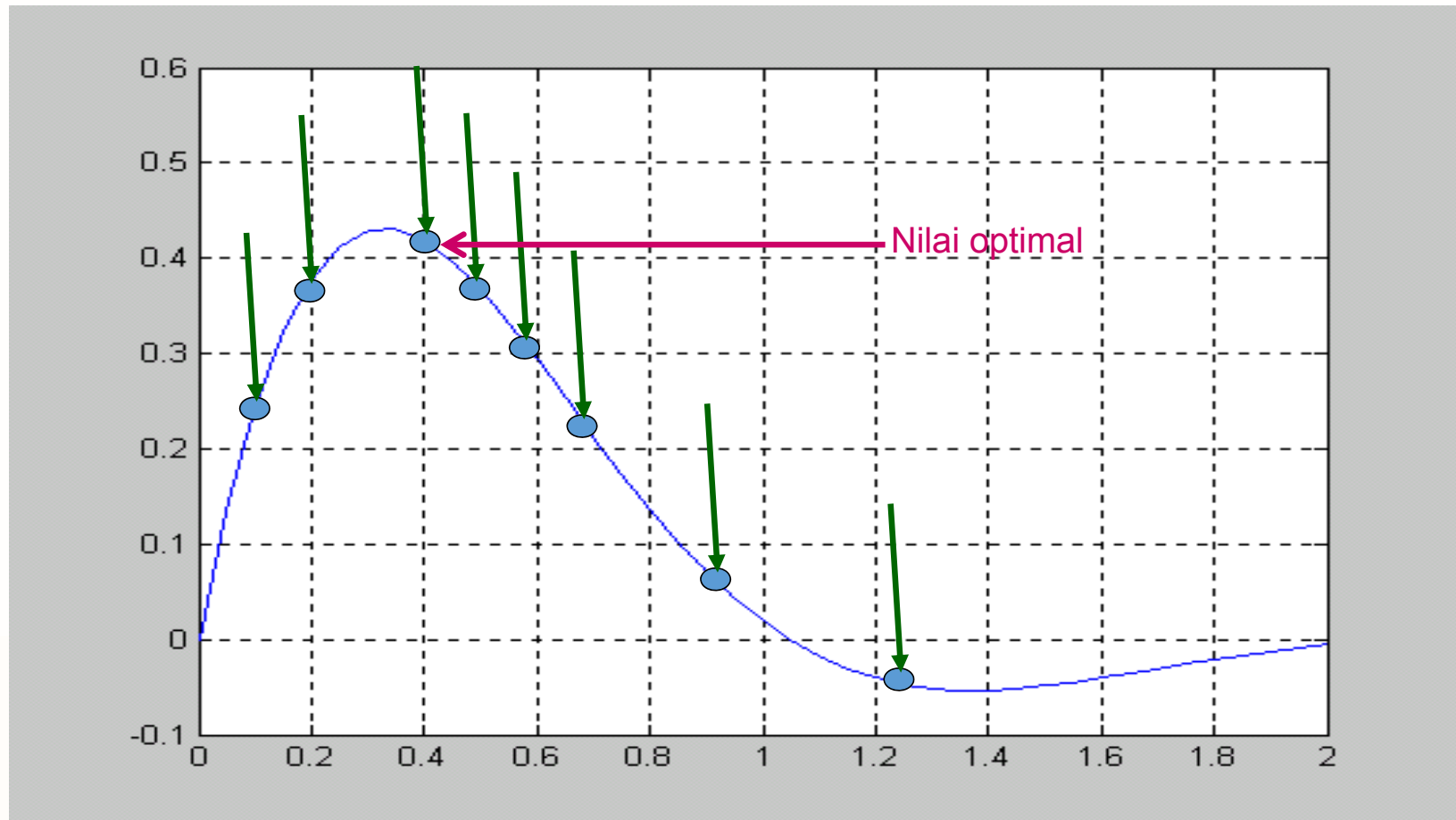
Individu menyatakan nilai x , dalam mendefinisikan nilai x sebagai individu, dapat digunakan nilai biner atau nilai float. **Pada algoritma genetika dasar digunakan nilai biner.** Fungsi di atas bila digambarkan akan menjadi:



Dari gambar terlihat bahwa penyelesaian berada pada nilai $0 < x < 1$. Jadi dengan menggunakan 8 bit biner didefinisikan:

00000000 berarti 0
11111111 berarti 1

Contoh Proses Algoritma Genetika Dalam Searching dan Optimasi



Definisi Individu

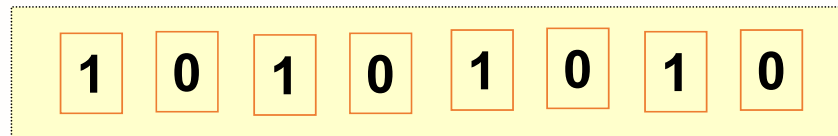
Individu dinyatakan dalam 8 gen biner, dengan batas 0 sampai dengan 1, berarti 1 bit setara dengan 2^{-8} .

Sebagai contoh:

$$10001001 = (128+8+1)/256 = 0.5352$$

$$00110100 = (4+16+32)/256 = 0.2031$$

$$01010010 = (2+16+64)/256 = 0.3203$$



← Individu

Fungsi Fitness :



Fungsi fitness adalah fungsi $f(x)$, karena yang dicari adalah nilai maksimum.

Membangkitkan Populasi Awal

Membangkitkan sejumlah individu, misalkan satu populasi terdiri dari 10 individu, maka dibangkitkan 10 individu dengan 8 gen biner yang dibangkitkan secara acak.

10010000 -- 0.56250 -- 0.32244

01001110 -- 0.30469 -- 0.43060

01100110 -- 0.39844 -- 0.41933

10111101 -- 0.73828 -- 0.18266

11101000 -- 0.90625 -- 0.06699

11110010 -- 0.94531 -- 0.04543

00110011 -- 0.19922 -- 0.37778

11111100 -- 0.98438 -- 0.02616

10000111 -- 0.52734 -- 0.34828

10001011 -- 0.54297 -- 0.33702

→ Individu maksimum

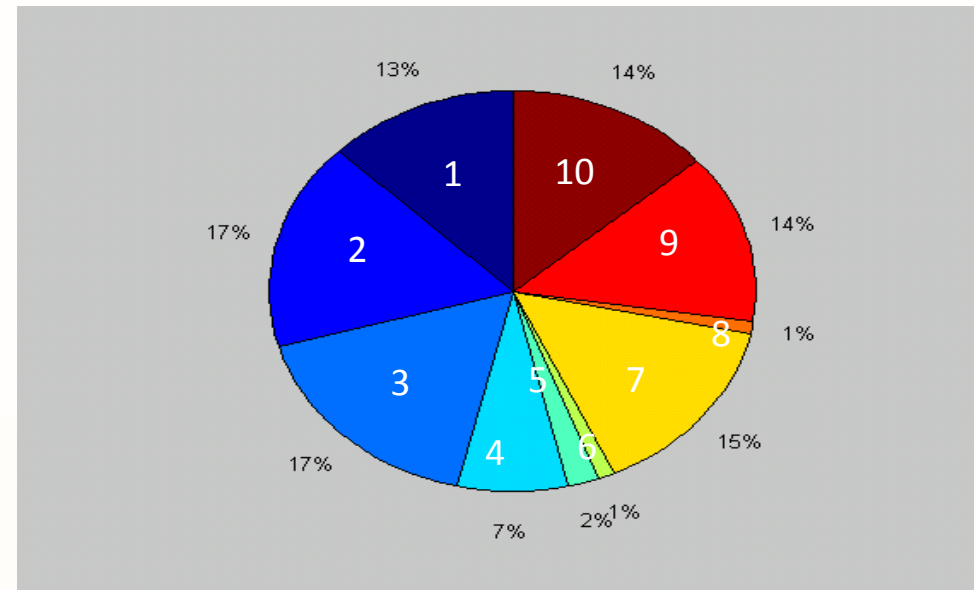
Seleksi

Seleksi adalah proses pemilihan calon induk, dalam proses seleksi ini terdapat beberapa metode yang bisa digunakan antara lain:

- Mesin Roulette (Roulette Wheel)
- Competition dan Tournament.

Dalam contoh ini digunakan Mesin Roullete yang memang metode paling dasar dan model acaknya uniform.

Seleksi dilakukan dengan menggunakan prosentase fitness setiap individu, dimana setiap individu mendapatkan luas bagian sesuai dengan prosentase nilai fitnessnya.



Cross-Over

Cross-Over (Perkawinan Silang) merupakan proses mengkombinasikan dua individu untuk memperoleh individu-individu baru yang diharapkan mempunyai fitness lebih baik. Tidak semua pasangan induk mengalami proses cross-over, banyaknya pasangan induk yang mengalami cross-over ditentukan dengan nilai probabilitas cross-over.

0 0 1 **1 1 0 0** 1 -- 0.22266 ← induk 1

1 0 0 **1 1 0 1** 0 -- 0.60156 ← induk 2

0 0 1 **1 1 0 1** 1 -- 0.23050 ← anak 1

1 0 0 **1 1 0 0** 0 -- 0.59382 ← anak 2

Fitness

0.3968

0.2921

0.4022

0.2982

Mutasi Gen

Mutasi gen adalah proses penggantian gen dengan nilai inversinya, gen 0 menjadi 1 dan gen 1 menjadi 0. Proses ini dilakukan secara acak pada posisi gen tertentu pada individu-individu yang terpilih untuk dimutasikan. Banyaknya individu yang mengalami mutasi ditentukan oleh besarnya probabilitas mutasi.

0 0 1 1 1 **0** 0 1 -- 0.22266 ← induk

0 0 1 1 1 **1** 0 1 -- 0.23828 ← induk

Fitness
0.3968
0.4070

Contoh Hasil Algoritma Genetika

Generasi ke 1 :

10100111 -- 0.65234 -- 0.25127
01000110 -- 0.27344 -- 0.42328
01001110 -- 0.04297 -- 0.43060
01110110 -- 0.46094 -- 0.39076
10111001 -- 0.72266 -- 0.19488
10001111 -- 0.55859 -- 0.32540
10001000 -- 0.53125 -- 0.34550
10010011 -- 0.57422 -- 0.31348
00111011 -- 0.23047 -- 0.40214
10000011 -- 0.51172 -- 0.35913

Generasi ke 2 :

10000000 -- 0.50000 -- 0.36696
10001010 -- 0.53906 -- 0.33987
01001110 -- 0.04297 -- 0.43060
10010111 -- 0.58984 -- 0.30132
10000011 -- 0.51172 -- 0.35913
10000011 -- 0.51172 -- 0.35913
01001110 -- 0.30469 -- 0.43060
10001000 -- 0.53125 -- 0.34550
10111101 -- 0.73828 -- 0.18266
01000010 -- 0.25781 -- 0.41715

Contoh Hasil Algoritma Genetika

Generasi ke 3 :

01001110 -- 0.30469 -- 0.43060

01001110 -- 0.30469 -- 0.43060

10000011 -- 0.51172 -- 0.35913

10001000 -- 0.53125 -- 0.34550

11001110 -- 0.80469 -- 0.13301

01001110 -- 0.04297 -- 0.43060

10001010 -- 0.53906 -- 0.33987

10000011 -- 0.51172 -- 0.35913

01001110 -- 0.30469 -- 0.43060

10000011 -- 0.51172 -- 0.35913

Generasi ke 4 :

00001110 -- 0.05469 -- 0.14641

11001000 -- 0.78125 -- 0.15005

01001110 -- 0.30469 -- 0.43060

10000011 -- 0.51172 -- 0.35913

01001110 -- 0.30469 -- 0.43060

01001110 -- 0.30469 -- 0.43060

01000011 -- 0.26172 -- 0.41885

10001110 -- 0.55469 -- 0.32833

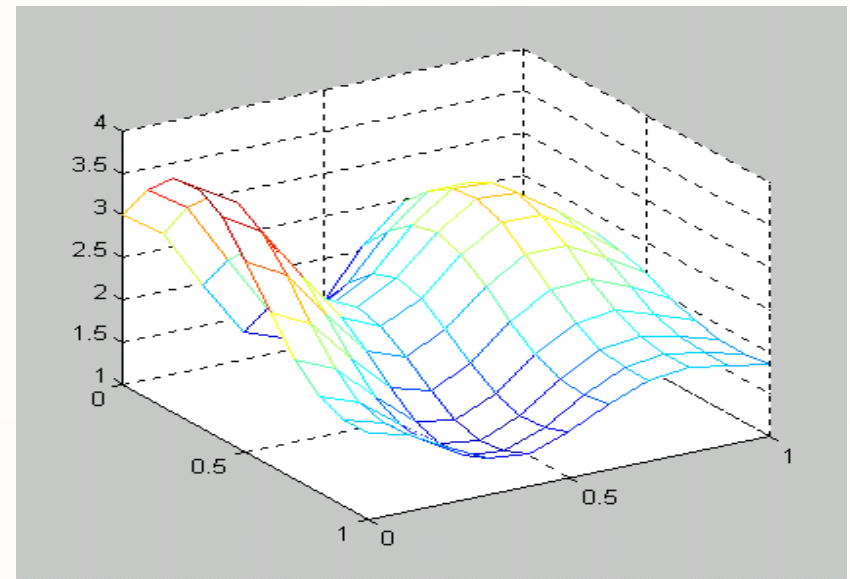
00001110 -- 0.05469 -- 0.14641

01001110 -- 0.78906 -- 0.43060

Algoritma Genetika Untuk Menentukan Nilai Maksimal Fungsi 2 variabel bebas

$$f(x, y) = 2 + e^{-(x^2 + y^2)} \{ \sin(4x) + \cos(8y) \}$$

- Penyelesaian berupa pasangan nilai (x,y), sehingga individu didefinisikan sebagai pasangan (x,y).
- Dalam hal ini digunakan gen float untuk penyederhanaan sistem, karena gen biner akan menyebabkan besarnya ukuran kromosom.
- Fungsi fitness adalah fungsi f(x,y).



Pembangkitan Populasi Awal

Populasi awal dapat dibangkitkan dengan membangkitkan sejumlah pasangan (x,y) secara acak.

x	y	fitness
0.66682	0.98088	2.02703
0.68314	0.29875	1.86538
0.87539	0.15460	2.00401
0.50769	0.93574	2.15746
0.46789	0.67386	1.45168
0.24484	0.42791	2.21924
0.56603	0.83281	1.90247
0.76072	0.17132	1.99177
0.34517	0.44064	1.86459
0.44755	0.75244	1.77778
0.51579	0.61550	1.23893
0.15734	0.25417	3.55771

Operator Dalam Algoritma Genetika

- Seleksi menggunakan roulette wheel.
- Cross-Over menggunakan arithmetic cross-over dengan definisi:

$$anak [1] = r.induk [1] + (1 - r).induk [2]$$

$$acak [2] = r.induk [2] + (1 - r).induk [1]$$

dimana r adalah bilangan 0 s/d 1

- Mutasi menggunakan random mutation dimana gen yang dimutasi diacak kembali.



Hasil Algoritma Genetika

Generasi 1:

x	y	fitness
0.49263	0.67386	1.41828
0.73599	0.17132	1.99971
0.46789	0.21433	2.53206
0.26464	0.17132	3.33107
0.56603	0.83281	1.90247
0.56603	0.83281	1.90247
0.70012	0.17132	2.02239
0.21795	0.25417	3.37522
0.15734	0.25417	3.55771
0.42702	0.25417	2.59217
0.17084	0.25417	3.52082
0.74723	0.17132	1.99537

Generasi 2:

x	y	fitness
0.32399	0.21433	3.08441
0.40854	0.17132	2.75335
0.21732	0.25417	3.37731
0.15797	0.25417	3.55604
0.56603	0.30491	2.00455
0.17084	0.78207	2.39452
0.17084	0.19691	3.65879
0.73599	0.22859	1.99249
0.35642	0.25417	2.85825
0.91416	0.83281	1.89165
0.66069	0.17132	2.06394
0.15734	0.25417	3.55771

Hasil Algoritma Genetika

Generasi 3:

x	y	fitness
0.40784	0.22363	2.73759
0.17084	0.72976	2.19665
0.85135	0.29176	1.89196
0.56603	0.26732	2.13788
0.40854	0.17132	2.75335
0.17084	0.78207	2.39452
0.15734	0.25417	3.55771
0.15797	0.25417	3.55604
0.17084	0.19691	3.65879
0.76096	0.83281	1.82495
0.40854	0.17132	2.75335
0.40854	0.17132	2.75335

Generasi 4:

x	y	fitness
0.27182	0.14624	3.25968
0.17084	0.19691	3.65879
0.15765	0.32497	3.16152
0.16415	0.29516	3.33433
0.37024	0.25417	2.80481
0.27987	0.17132	3.27160
0.26467	0.17335	3.33281
0.17084	0.19488	3.66005
0.17415	0.17132	3.64583
0.17714	0.36596	2.82353
0.17182	0.36596	2.83666
0.17947	0.17132	3.62975

Hasil Algoritma Genetika

Generasi 5:

x	y	fitness
0.17164	0.19047	3.65955
0.17084	0.17584	3.65923
0.16415	0.21711	3.65133
0.64027	0.24422	2.05715
0.96166	0.19499	2.09101
0.17155	0.42238	2.43739
0.17084	0.19488	3.66005
0.17415	0.49995	2.00790
0.17084	0.19488	3.66005
0.37024	0.25933	2.78706
0.16415	0.17137	3.67512
0.17164	0.22974	3.60074

Generasi 6:

x	y	fitness
0.17129	0.69742	2.07593
0.17164	0.45035	2.26035
0.21598	0.22974	3.46123
0.16415	0.17137	3.67512
0.30561	0.21997	3.14860
0.28211	0.20465	3.26279
0.28211	0.22974	3.21996
0.28211	0.22974	3.21996
0.16415	0.19227	3.68061
0.17129	0.17188	3.65484
0.32762	0.22974	3.04113
0.28211	0.22974	3.21996

Algoritma Genetika Untuk Mencari Kata Secara Acak Word Matching

- Sebuah kata ditentukan sebagai target, misalnya: 'GENETIKA'.
- Bila setiap huruf diberi nilai dengan nilai urut alfabet, maka targetnya bisa dinyatakan sebagai besaran numerik :

Target=[7 5 14 5 20 9 11 1]

- Komputer akan membangkitkan kata dengan jumlah huruf yang sama dengan target secara acak, terus-menerus hingga diperoleh kata yang sama dengan kata target.

Nilai Fitness

- Nilai fitness adalah nilai yang menyatakan baik tidaknya suatu solusi (individu).
- Nilai fitness ini yang dijadikan acuan dalam mencapai nilai optimal dalam algoritma genetika.
- Algoritma genetika bertujuan mencari individu dengan nilai fitness yang paling tinggi.

Definisi Fitness dalam Word Matching

- Nilai fitness adalah inversi dari perbedaan antara nilai kata yang muncul (individu) dan target yang ditentukan.
- Individu dalam kasus ini adalah satu kata yang muncul dari proses acak.
- Misalnya kata yang muncul : AGHSQEBC atau [1 7 8 19 17 5 2 3] dan targetnya GENETIKA atau [7 5 14 5 20 9 11 1] maka, nilai perbedaannya:

Target = GENETIKA = [7 5 14 5 20 9 11 1]

Kata = AGHSQEBC = [1 7 8 19 17 5 2 3]

$$g_i - t_i = |1-7| + |7-5| + |8-14| + |19-5| + |17-20| + \\ |5-9| + |2-11| + |3-1|$$

$$= 6+2+6+14+3+4+9+2 = 46$$

$$\text{Fitness} = (26)(8) - 46 = 208-46 = 162$$



Definisi Fitness

- Fitness didefinisikan sebagai:

$$fitness(k) = (jumlahgen * 26) - \left(\sum_{i=1}^n g_i - t_i \right)$$

Dimana : g_i adalah gen ke i dari individu
 t_i adalah target ke i

Pembangkitan Populasi Awal

Populasi awal dibangkitkan dengan cara membangkitkan semua huruf dalam sejumlah kata (individu) yang dibangkitkan.

4 6 6 16 8 6 1 11 - D F F P H F A K >> Fitness = 83

-

-

-

dst



Seleksi Dengan Mesin Roulette

$$Pr(h_i) = \text{Fitness}(h_i) / \sum_{j=1}^n \text{Fitness}(h_j)$$

Individu 1: fitness = 10 %

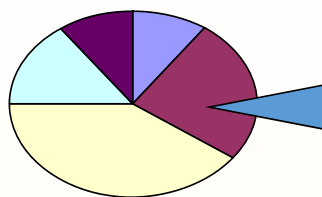
Individu 2: fitness = 25 %

Individu 3: fitness = 40 %

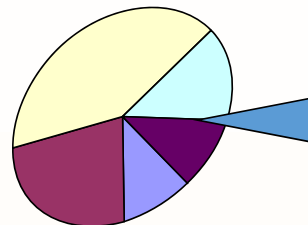
Individu 4: fitness = 15%

Individu 5: fitness = 10%

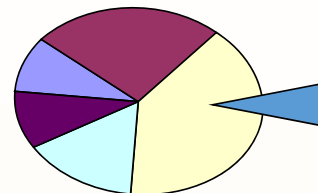
Individu terpilih



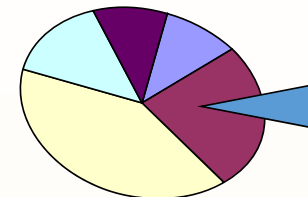
Individu 2



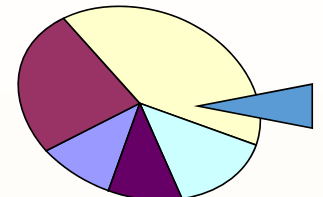
Individu 4



Individu 3



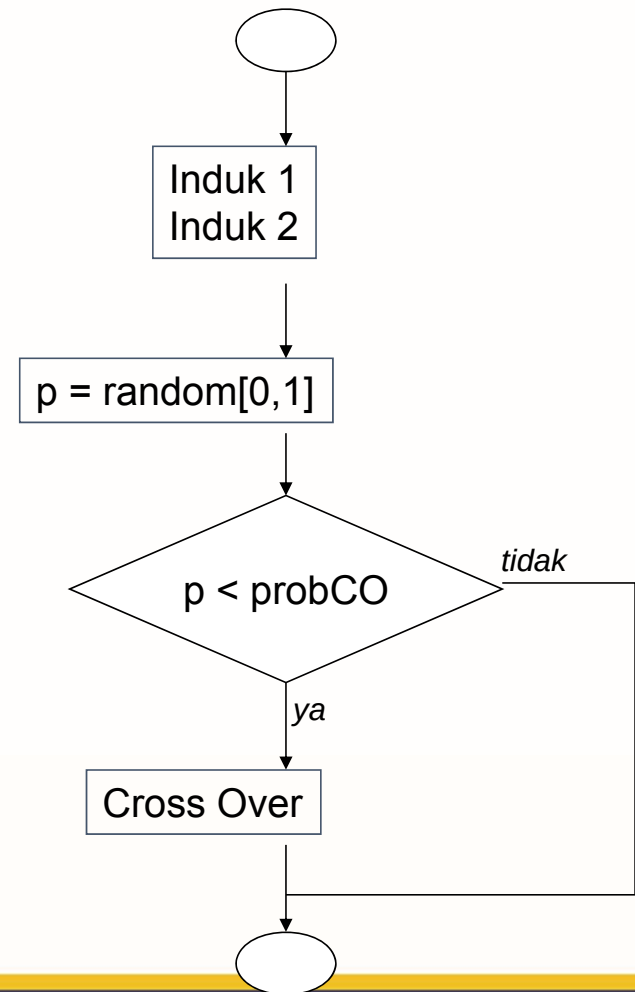
Individu 2



Individu 3

Cross Over

- Cross Over (Pindah Silang) merupakan salah satu operator dalam algoritma genetika yang melibatkan dua induk untuk menghasilkan keturunan yang baru.
- Cross over dilakukan dengan melakukan pertukaran gen dari dua induk secara acak.
- Macam-macam Cross-Over yang banyak digunakan antara lain: pertukaran gen secara langsung dan pertukaran gen secara aritmatika.
- Proses cross over dilakukan pada setiap individu dengan probabilitas cross-over yang ditentukan.



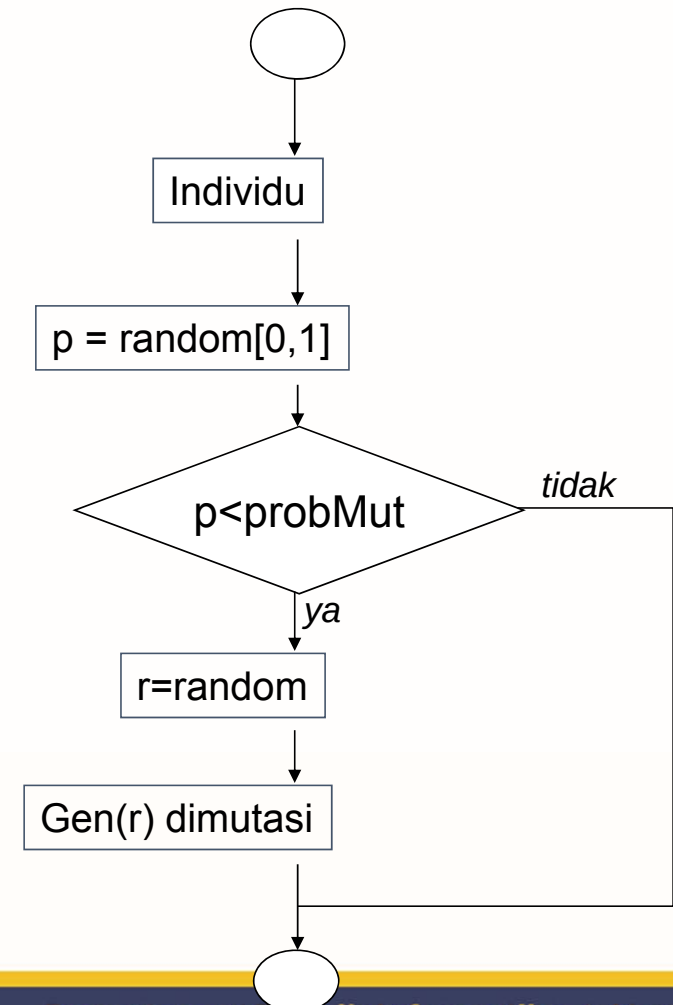
Cross-Over pada Word Matching

Cross-over pada kasus Word Matching memakai Cross Over pertukaran langsung.

8	5	15	15	24	6	11	1	--	H	E	O	O	X	F	K	A
5	22	14	11	19	23	10	2	--	E	V	N	K	S	W	H	B
8	5	14	11	19	6	11	1	--	H	E	N	K	S	F	K	A
5	22	15	15	24	23	10	2	--	E	V	O	O	X	W	H	B

Mutasi Gen

- Mutasi Gen merupakan operator yang menukar nilai gen dengan nilai inversinya, misalnya gennya bernilai 0 menjadi 1.
- Setiap individu mengalami mutasi gen dengan probabilitas mutasi yang ditentukan.
- Mutasi dilakukan dengan memberikan nilai inversi atau menggeser nilai gen pada gen yang terpilih untuk dimutasikan.



Mutasi Geser

- Mutasi dilakukan dengan mengacak kembali nilai pada range nilai tertentu (1-26 pada kasus Word Matching) dari gen yang dimutasikan.
- Perubahan nilai mutasi adalah pergeseran nilai yaitu:

$$g_i + k \text{ atau } g_i - k$$

dimana k : nilai pergeseran

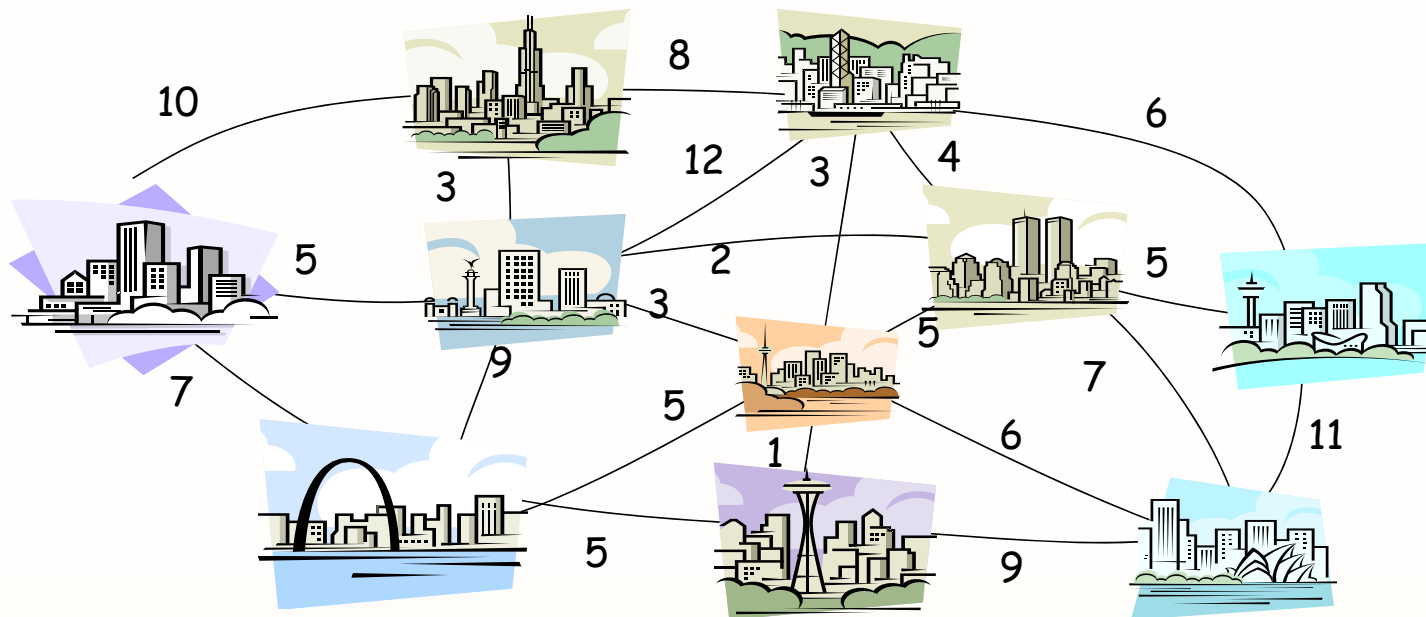
8 5 14 11 19 6 11 1 - - H E N K S F K A

8 5 15 11 19 6 11 1 - - H E O K S F K A

Hasil Algoritma Genetika

4	6	6	16	8	6	1	11	---	D	F	F	P	H	F	A	K
7	6	7	9	21	10	3	9	---	G	F	G	I	U	J	C	I
7	6	7	7	20	7	3	9	---	G	F	G	G	T	G	C	I
7	5	7	6	21	7	3	9	---	G	E	G	F	U	G	C	I
7	5	13	5	20	9	4	9	---	G	E	M	E	T	I	D	I
7	5	14	5	20	10	5	9	---	G	E	N	E	T	J	E	I
7	5	14	5	20	10	6	9	---	G	E	N	E	T	J	F	I
7	5	14	5	20	10	5	9	---	G	E	N	E	T	I	E	I
7	5	14	5	20	9	6	9	---	G	E	N	E	T	I	F	I
7	5	14	5	20	9	6	8	---	G	E	N	E	T	I	F	H
7	5	14	5	20	9	7	9	---	G	E	N	E	T	I	G	I
7	5	14	5	20	9	10	7	---	H	E	N	E	T	I	J	G
7	5	14	5	20	9	10	8	---	G	E	N	E	T	I	J	H
7	5	14	5	20	9	11	9	---	G	E	N	E	T	I	K	I
7	5	14	5	20	9	11	8	---	G	E	N	E	T	I	K	H
7	5	14	5	20	9	11	5	---	G	E	N	E	T	I	K	E
7	5	14	5	20	9	11	4	---	G	E	N	E	T	I	K	D
7	5	14	5	20	9	11	3	---	G	E	N	E	T	I	K	C
7	5	14	5	20	9	11	2	---	G	E	N	E	T	I	K	B
7	5	14	5	20	9	11	1	---	G	E	N	E	T	I	K	A

Traveling Salesman Problem



Constraint

- Tiap kota hanya dilalui satu kali dalam tiap lintasan yang dimungkinkan & lintasan berbentuk *cyclic*
- Lintasan yang ditempuh adalah lintasan yang terpendek

Pengkodean untuk TSP

Dengan representasi permutasi / representasi order

3 - 2 - 5 - 4 - 7 - 1 - 6 - 9 - 8

Fungsi Fitness

$$totaldist = \sum_{i=1}^{city_size} dist(c_i - c_{i+1}) + dist(c_1 - c_{city_size})$$

$$nilaifitness = \frac{jumlahindividu}{totaldist + 0.1}$$

Populasi Awal

- Dengan Kunci Random

Dibangkitkan bilangan random sejumlah gen untuk satu individu, misal:

0.23 0.82 0.45 0.74 0.87 0.11 0.56 0.69 0.78

maka didapatkan kromosom:

6 – 1 – 3 – 7 – 8 – 4 – 9 – 2 – 5

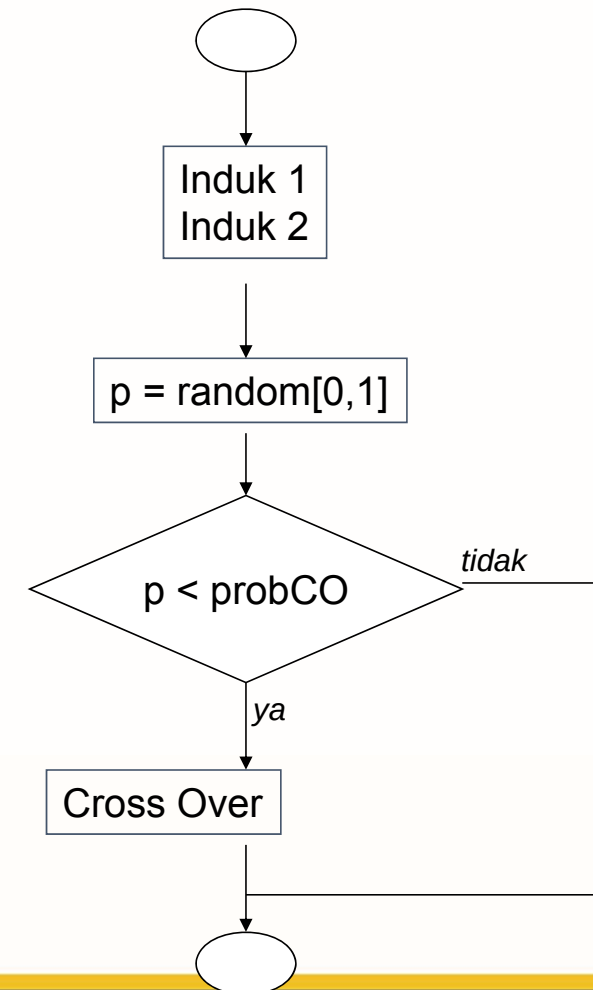
- Dengan Permutasi Josephus

selisih = 5, mulai = 6

6 - 2 - 7 - 3 - 8 - 4 - 9 - 5 - 1

Cross Over

- Cross Over (Pindah Silang) merupakan salah satu operator dalam algoritma genetika yang melibatkan dua induk untuk menghasilkan keturunan yang baru.
- Cross over dilakukan dengan melakukan pertukaran gen dari dua induk secara acak.
- Macam-macam Cross-Over yang banyak digunakan antara lain: pertukaran gen secara langsung dan pertukaran gen secara aritmatika.
- Proses cross over dilakukan pada setiap individu dengan probabilitas cross-over yang ditentukan.



Cross Over

- Untuk permasalahan TSP, Cross Over yang dipakai harus menjaga constraint dari TSP, yaitu nilai gen harus unik.
- Ada beberapa algoritma yang dipakai untuk permasalahan cross over diatas, diantara:
 - Partially Mapped Cross-Over (PMX)
 - Order Cross-Over (OX)
 - Cycle Cross-Over (CX)

Partially Mapped Crossover (PMX)

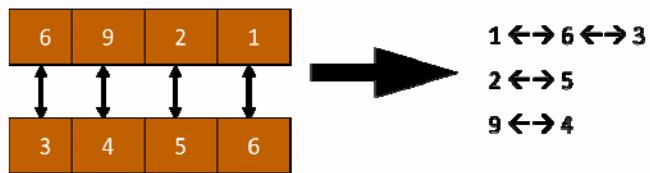
1. Memilih posisi yang akan dilakukan crossover. Tentukan dua posisi pada kromosom dengan aturan acak. Substring yang berada dalam dua posisi ini dinamakan daerah pemetaan.

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

2. Tukar dua substring antar induk untuk menghasilkan anak.

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

3. Menentukan mapping pertukaran di antara dua daerah pemetaan



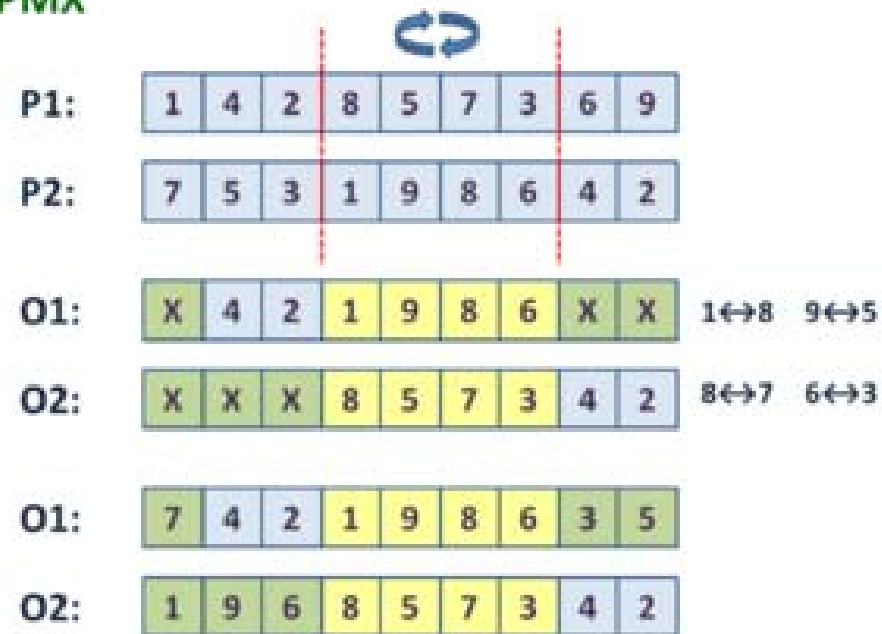
4. Menentukan kromosom keturunan berdasarkan hubungan mapping

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

Contoh lain

Partially Mapped Crossover (PMX)

PMX



Order Crossover (OX)

1. Memilih posisi (substring) yang akan dilakukan crossover secara acak

Induk 1	1	2	3	4	5	6	7	8	9
Induk 2	5	4	6	9	2	1	7	8	3

2. Produksi kromosom anak dengan mengosongkan tempat substring induk 2 pada induk 1

Anak 1			3	4	5		7	8	
Anak 2				9	2	1	7	8	

3. SHR substring pada tempat yang bersesuaian

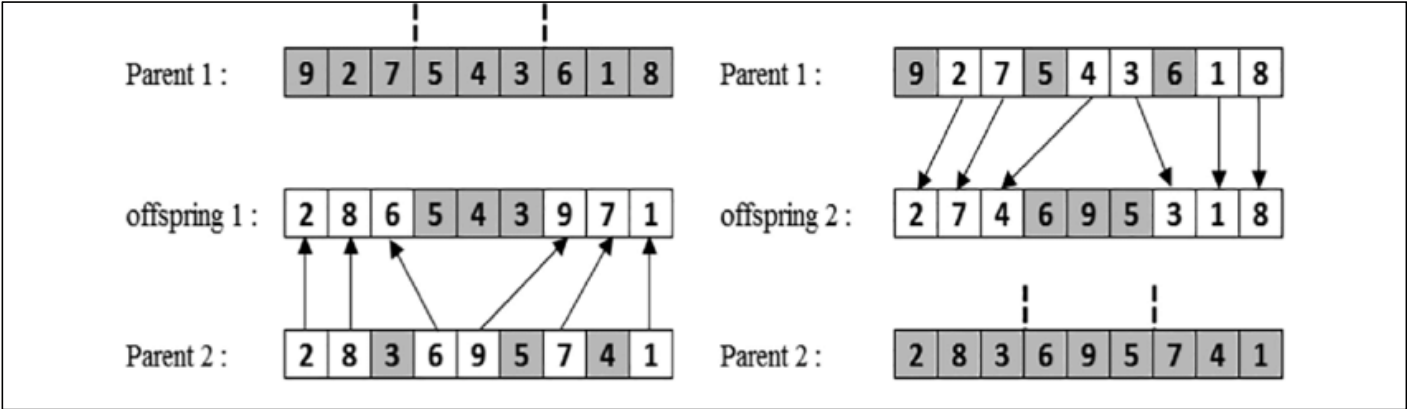
Anak 1	7	8					3	4	5
Anak 2	7	8					9	2	1

- Diciptakan oleh Davis
- Merupakan variasi dari PMX

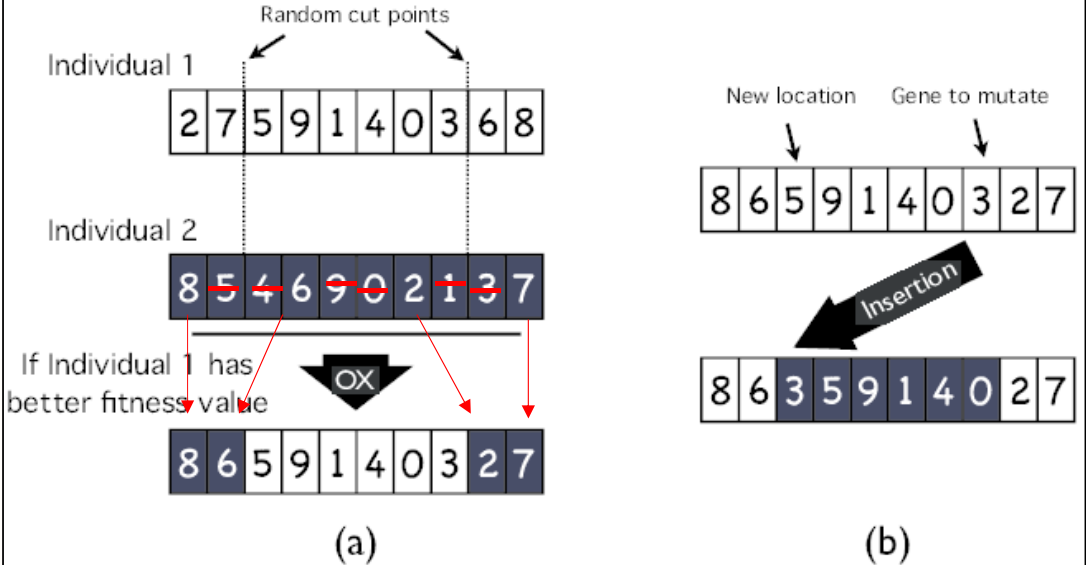
4. Tukar posisi substring antara dua induk

Anak 1	7	8	6	9	2	1	3	4	5
Anak 2	7	8	3	4	5	6	9	2	1

Contoh lain
Order Crossover (OX)



OX



Cycle Crossover (CX)

- Diciptakan oleh Oliver, Smith dan Holland.
- Metode ini mengkopi kota-kota dari satu induk dan memilih kota-kota yang lain dari induk yang lain, dengan mengingat dan menerapkan pola cycle.

1. Tentukan pola *cycle* (asumsi : pola dimulai dari posisi 1)

Induk 1	1	2	3	4	5	6	7	8	9
Induk 2	5	4	6	9	2	3	7	8	1

Pola *cycle* $1 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow 9 \rightarrow 1$

2. Kopi kota-kota dalam *cycle* pada *proto-child*

Proto-child	1	2		4	5				9
-------------	---	---	--	---	---	--	--	--	---

3. Tentukan kota-kota yang diingat dari induk yang lain

Induk 2	5	4	6	9	2	3	7	8	1
---------	---	---	---	---	---	---	---	---	---

Kota yang tidak ada

6	3	7	8
---	---	---	---

3 Mengisikan kota yang diingat dalam keturunan

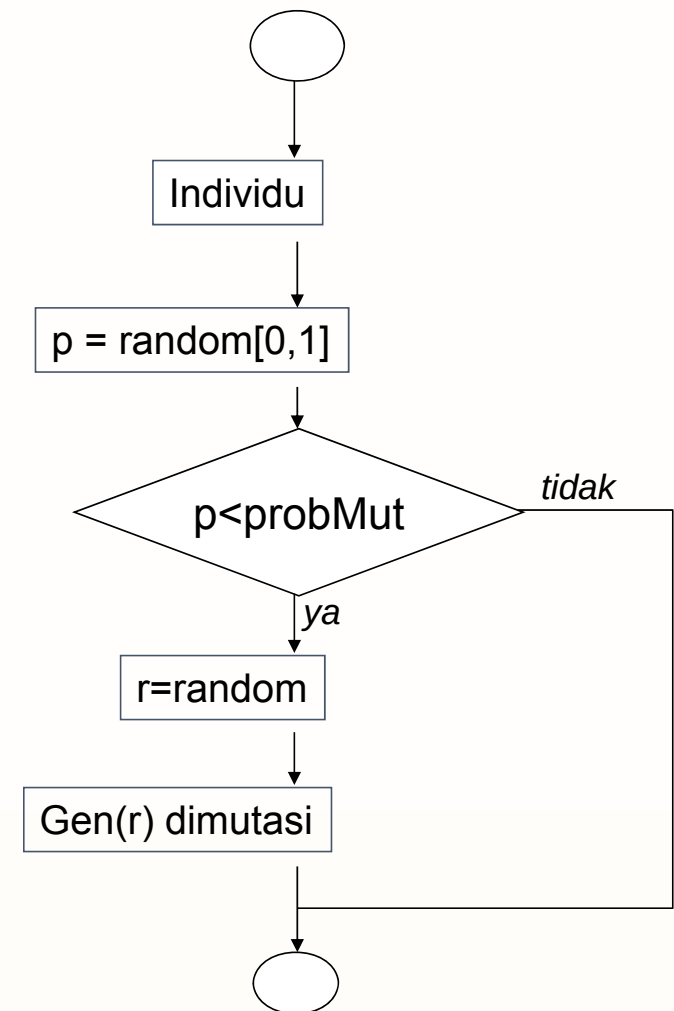
Keturunan 1	1	2	6	4	5	3	7	8	9
-------------	---	---	---	---	---	---	---	---	---

Dengan cara yang sama, diperoleh keturunan 2

Keturunan 2	5	4	3	9	2	6	7	8	1
-------------	---	---	---	---	---	---	---	---	---

Mutasi Gen

- Mutasi Gen merupakan operator yang menukar nilai gen dengan nilai inversinya, misalnya gennya bernilai 0 menjadi 1.
- Setiap individu mengalami mutasi gen dengan probabilitas mutasi yang ditentukan.
- Mutasi dilakukan dengan memberikan nilai inversi atau menggeser nilai gen pada gen yang terpilih untuk dimutasikan.



Inversion Mutation

1. Memilih posisi yang akan dilakukan mutasi

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

2. Invers-kan substring yang sudah dipilih

1	2	6	5	4	3	7	8	9
---	---	---	---	---	---	---	---	---

Insertion Mutation

1. Memilih gen secara acak

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

2. Masukkan ke dalam kromosom dengan posisi secara acak

1	2	6	3	4	5	7	8	9
---	---	---	---	---	---	---	---	---

Reciprocal Exchange Mutation

1. Memilih 2 gen secara acak

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

2. Tukar 2 gen yang sudah dipilih tsb.

1	2	6	4	5	3	7	8	9
---	---	---	---	---	---	---	---	---

Latihan Soal

1. Carilah permasalahan lain yang dapat diselesaikan dengan Algoritma Genetika!
2. Implementasikan program untuk salah satu permasalahan di atas!

Referensi

- Modul Ajar Kecerdasan Buatan, Entin Martiana, Ali Ridho Barakbah, Yuliana Setiowati, Politeknik Elektronika Negeri Surabaya, 2014.
- Artificial Intelligence (Teori dan Aplikasinya), Sri Kusumadewi, cetakan pertama, Penerbit Graha Ilmu, 2003.



bridge to the future

<http://www.eepis-its.edu>