

Suatu Alternatif Penyelesaian Permasalahan Searching,  
Optimasi dan Machine Learning

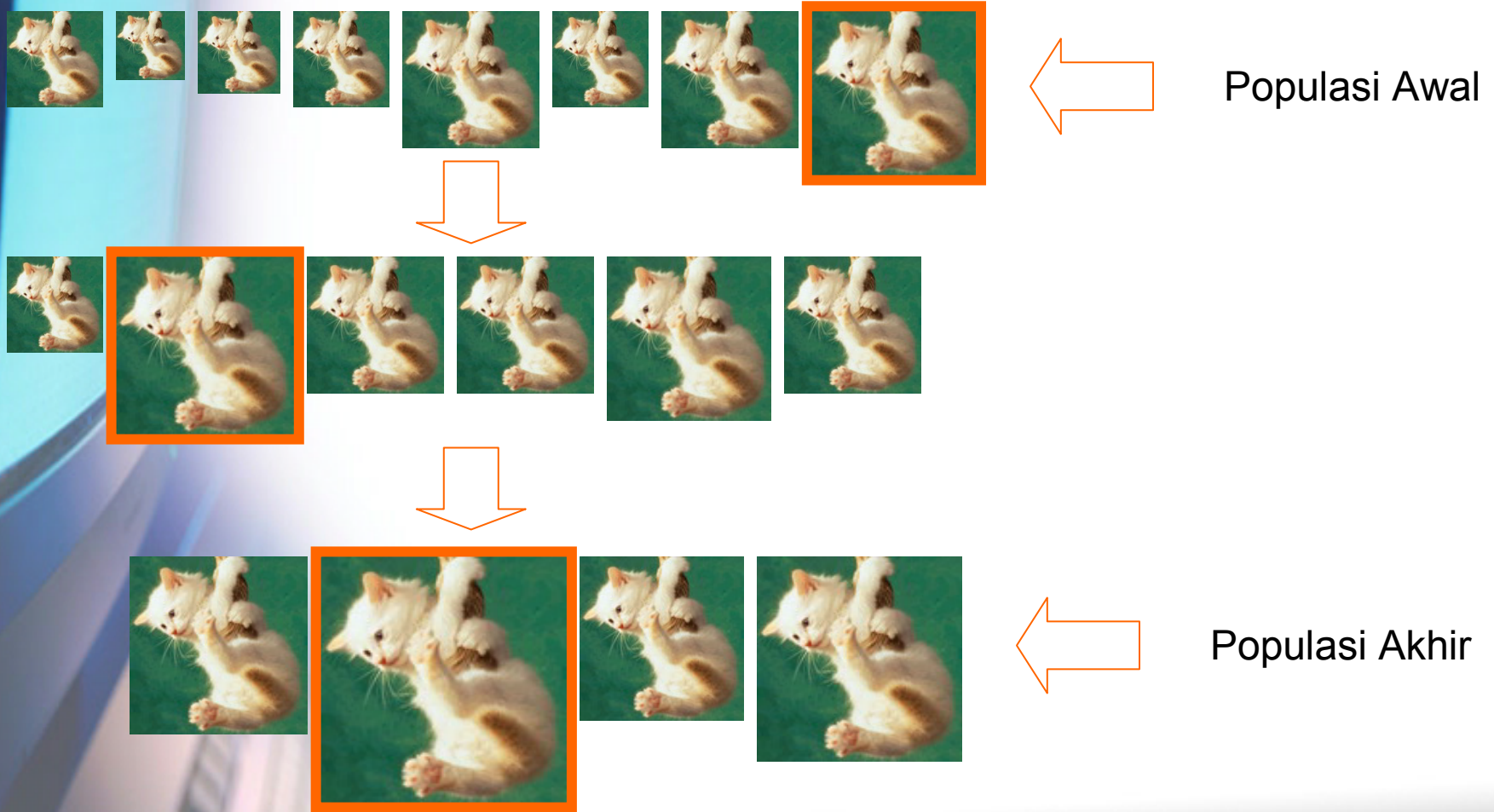
# ALGORITMA GENETIKA



# Algoritma Genetika

- ◆ Algoritma Genetika adalah algoritma yang memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi.
- ◆ Dalam proses evolusi, individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya. **“Hanya individu-individu yang kuat yang mampu bertahan”**.
- ◆ Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembang-biakan. Dalam algoritma genetika ini, proses perkembang-biakan ini menjadi proses dasar yang menjadi perhatian utama, dengan dasar berpikir: **“Bagaimana mendapatkan keturunan yang lebih baik”**.
- ◆ Algoritma genetika ini ditemukan oleh John Holland dan dikembangkan oleh muridnya David Goldberg.

# Proses Algoritma Genetika



# Beberapa Definisi Penting Dalam Algoritma Genetika

- ◆ **Genotype (Gen)**, sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma genetika, gen ini bisa berupa nilai biner, float, integer maupun karakter, atau kombinatorial.
- ◆ **Allele**, nilai dari gen.
- ◆ **Kromosom**, gabungan gen-gen yang membentuk nilai tertentu.
- ◆ **Individu**, menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat
- ◆ **Populasi**, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.
- ◆ **Generasi**, menyatakan satu-satuan siklus proses evolusi.
- ◆ **Nilai Fitness**, menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan.

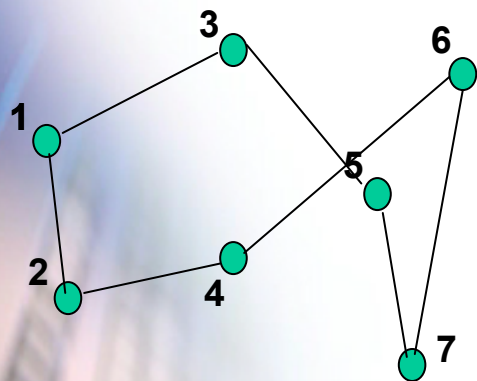


## Hal-Hal Yang Harus Dilakukan Dalam Menggunakan Algoritma Genetika

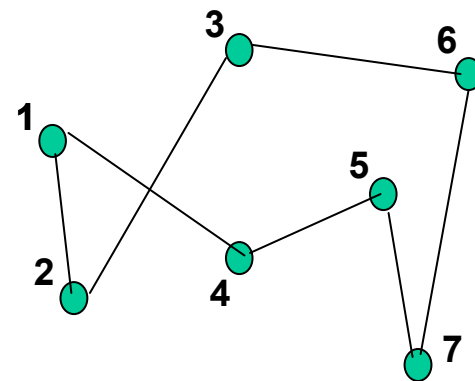
- ◆ **Mendefinisikan individu**, dimana individu menyatakan salah satu solusi (penyelesaian) yang mungkin dari permasalahan yang diangkat.
- ◆ **Mendefinisikan nilai fitness**, yang merupakan ukuran baik-tidaknya sebuah individu atau baik-tidaknya solusi yang didapatkan.
- ◆ Menentukan proses **pembangkitan populasi awal**. Hal ini biasanya dilakukan dengan menggunakan pembangkitan acak seperti *random-walk*.
- ◆ Menentukan proses **seleksi** yang akan digunakan.
- ◆ Menentukan proses **perkawinan silang (cross-over)** dan **mutasi gen** yang akan digunakan.

# Pengertian Individu

- Individu bisa dikatakan sama dengan kromosom, yang merupakan kumpulan gen. Gen ini bisa biner, float, dan kombinatorial.
- Individu menyatakan salah satu solusi yang mungkin. Misalkan dalam TSP individu menyatakan jalur yang ditempuh, dalam penentuan nilai maksimal dari  $F(x,y)$  individu menyatakan nilai  $(x,y)$ .

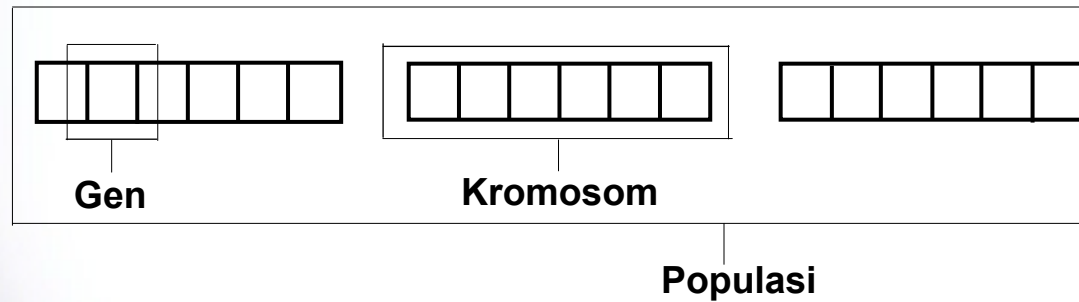


Individu : 1 3 5 7 6 4 2



Individu : 1 2 3 6 7 5 4

# Individu Dalam Algoritma Genetika

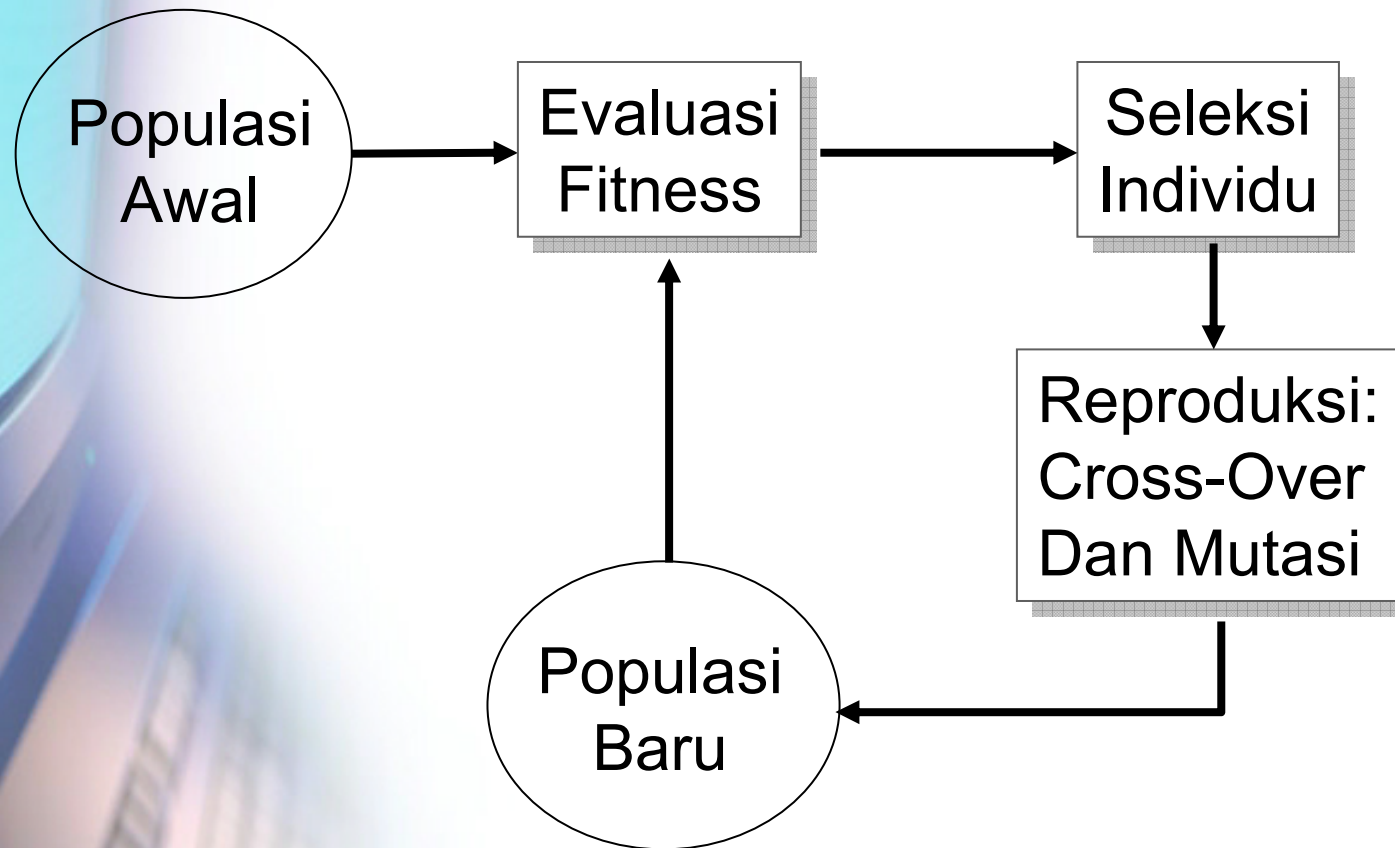


# Nilai Fitness

- 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.
- Dalam TSP, karena TSP bertujuan meminimalkan jarak, maka nilai fitnessnya adalah inversi dari jarak.



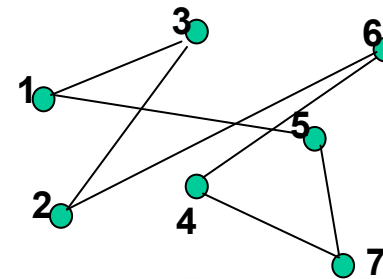
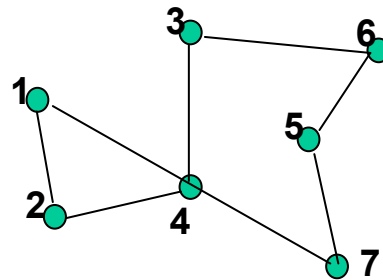
# Siklus Algoritma Genetika



# Membangkitkan Populasi Awal

- Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Syarat-syarat yang harus dipenuhi untuk menunjukkan suatu solusi harus benar-benar diperhatikan dalam pembangkitan setiap individunya.
- Dalam TSP, populasi menyatakan sejumlah solusi (jalur) yang dicari secara acak. Misalkan dalam populasi terdapat 4 individu, maka contoh populasi awal TSP dengan 7 kota adalah sebagai berikut:

1	2	4	3	6	5	7
5	7	4	6	2	3	1
7	2	6	4	5	1	3
4	6	5	7	3	2	1



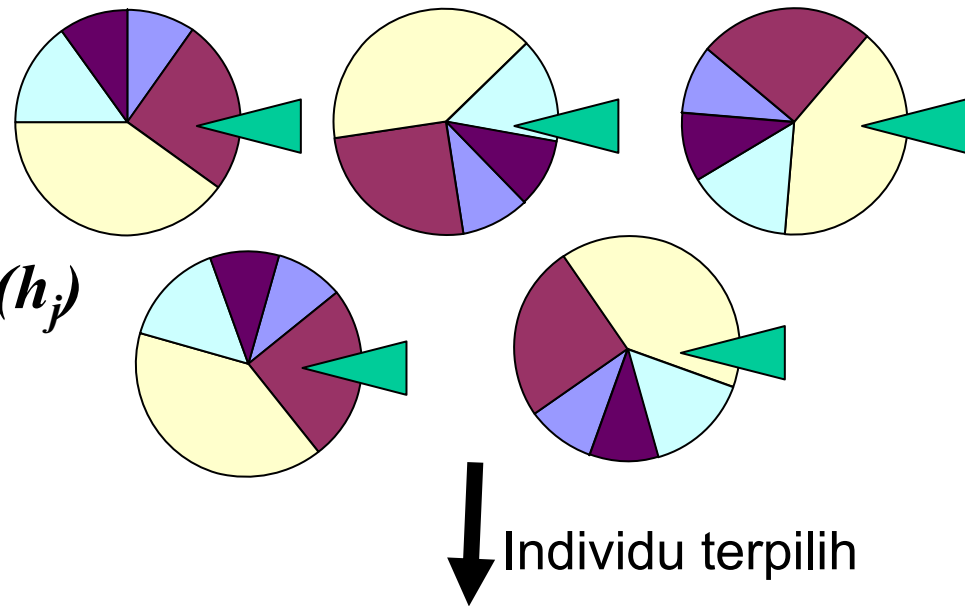
# Seleksi

- Seleksi dilakukan untuk mendapatkan calon induk yang baik. “Induk yang baik akan menghasilkan keturunan yang baik”.
- Semakin tinggi nilai fitness suatu individu semakin besar kemungkinannya untuk terpilih.
- Seleksi dapat dilakukan dengan menggunakan dua macam teknik, yaitu mesin roulette, dan turnamen.

# Seleksi Dengan Mesin Roulette

$$Pr(h_j) = \frac{Fitness(h_j)}{\sum_{j=1}^p Fitness(h_j)}$$

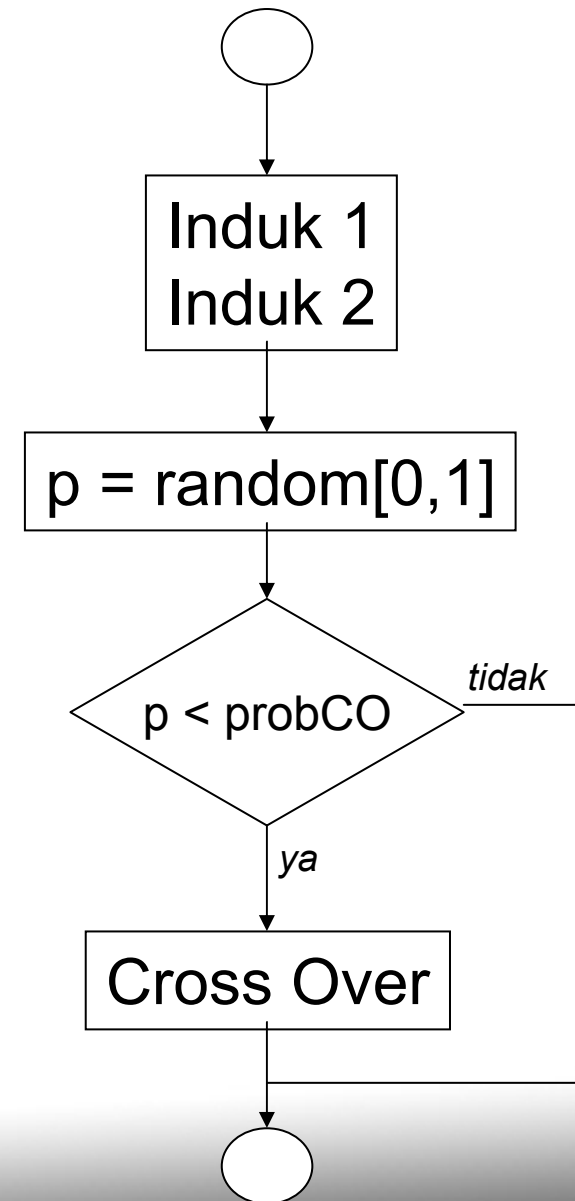
- Individu 1: fitness = 10 %
- Individu 2: fitness = 25 %
- Individu 3: fitness = 40 %
- Individu 4: fitness = 15%
- Individu 5: fitness = 10%



- Individu terpilih
- Individu 2
  - Individu 4
  - Individu 3
  - Individu 2
  - Individu 3

# Cross Over

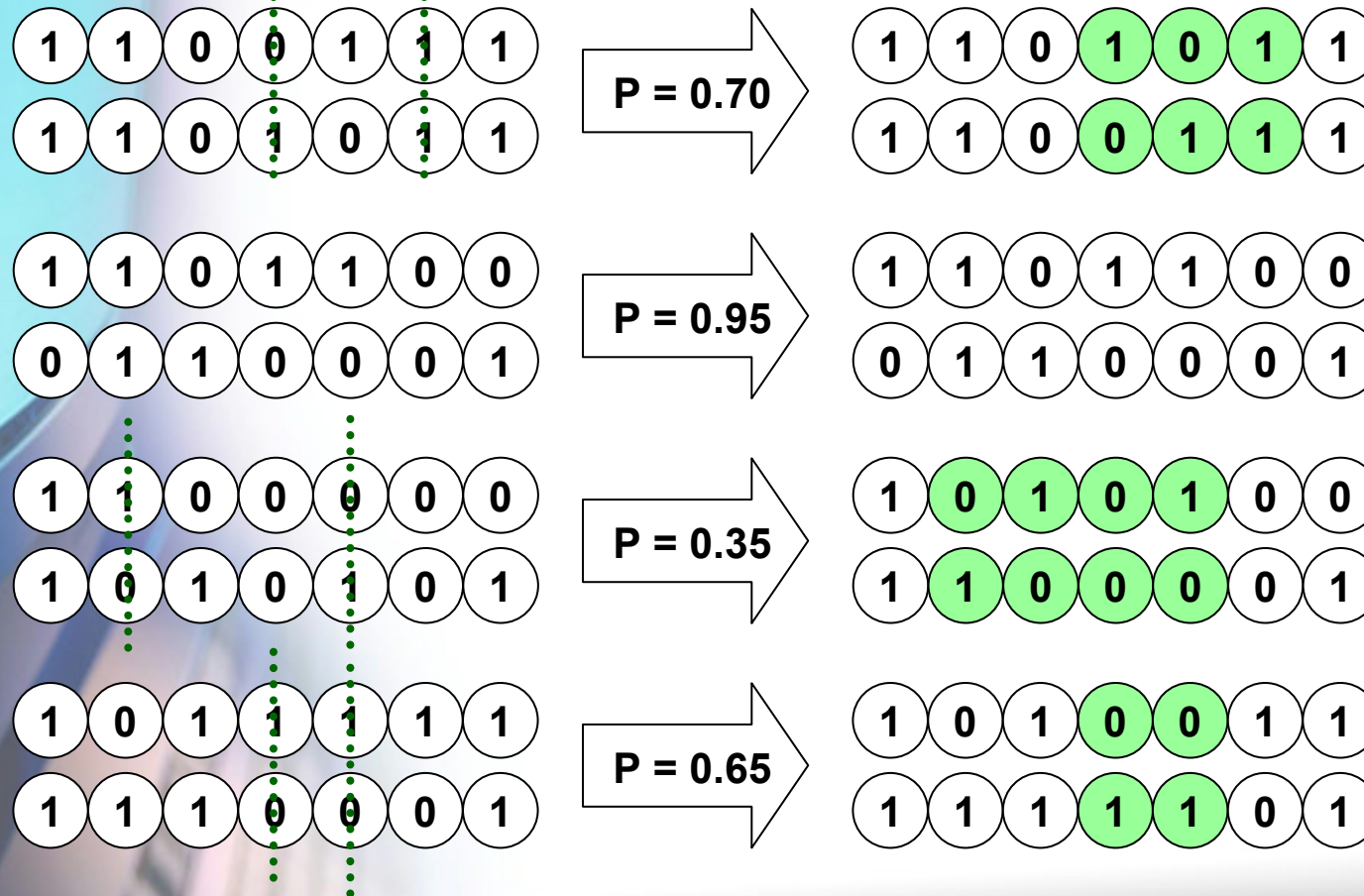
- Cross Over (Pindah Silang) merupakan salah satu operator dalam algoritma genetika yang melibatkan dua induk untuk menghasilkan keturunan yang baru.
- Cross over dilakukan dengan melakukan pertukaran gen dari dua induk secara acak.
- Macam-macam Cross-Over yang banyak digunakan antara lain: pertukaran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





# Cross Over

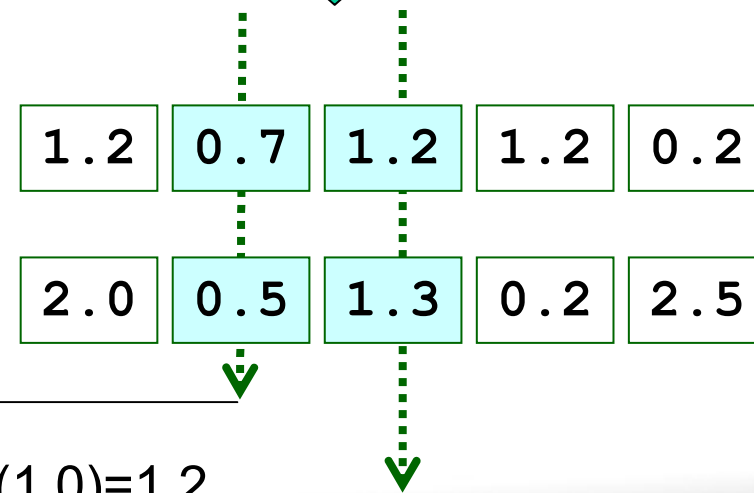
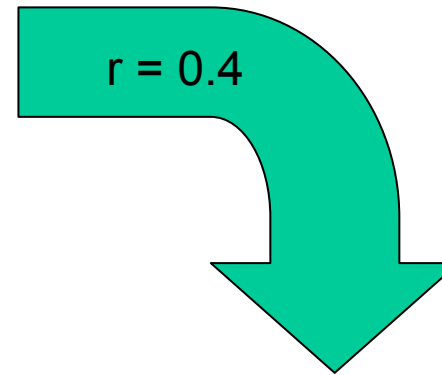
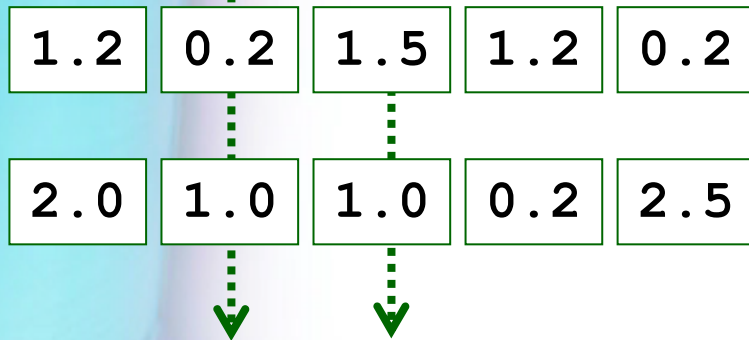
Pertukaran secara langsung



Ditentukan probabilitas Cross-Over = 0.9

# Cross Over

Pertukaran Secara Aritmatika



$$(0.4)(0.2) + (0.6)(1.0) = 0.68$$

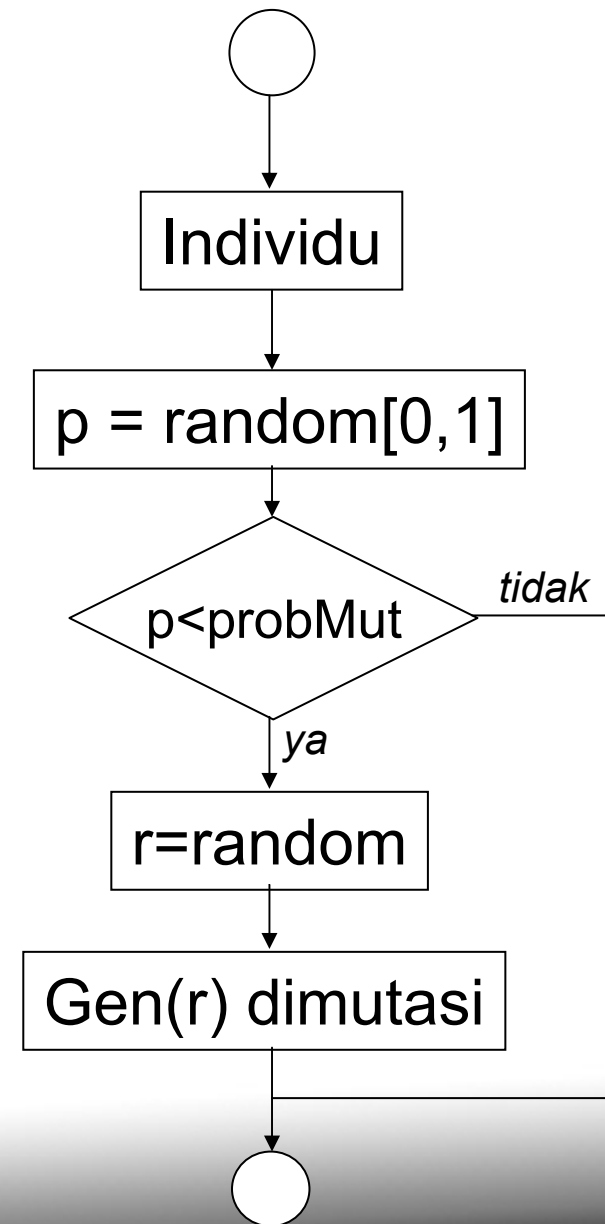
$$(0.4)(1.0) + (0.6)(0.2) = 0.52$$

$$(0.4)(1.5) + (0.6)(1.0) = 1.2$$

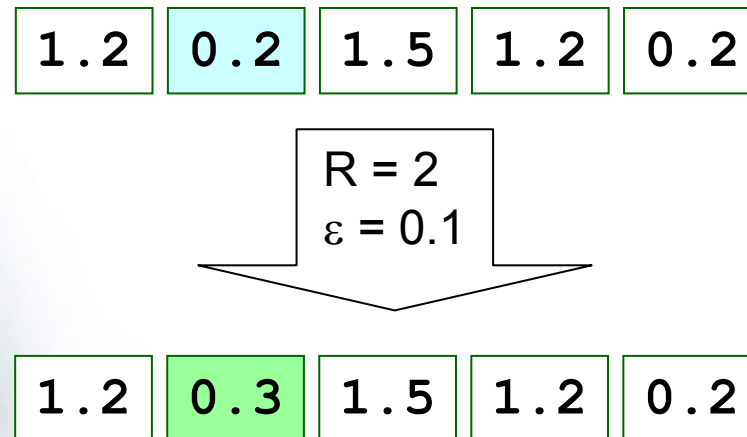
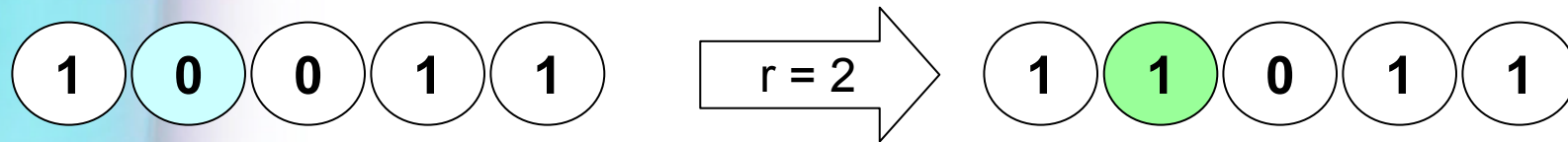
$$(0.4)(1.0) + (0.6)(1.5) = 1.3$$

# 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 Gen

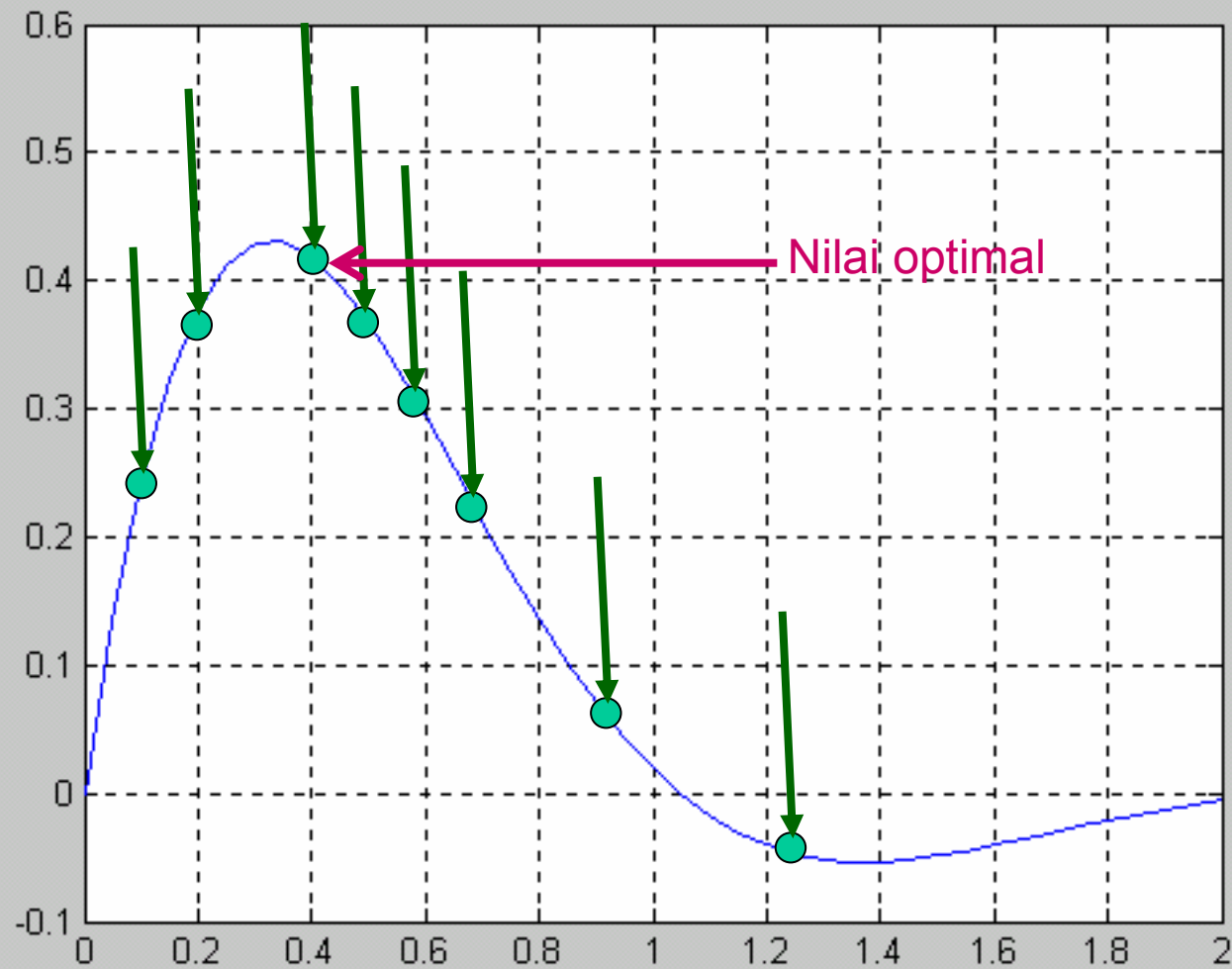


## Hal penting yang harus diketahui dalam pemakaian Algoritma Genetika

- ◆ Algoritma Genetika adalah algoritma yang dikembangkan dari proses pencarian solusi menggunakan pencarian acak, ini terlihat pada proses pembangkitan populasi awal yang menyatakan sekumpulan solusi yang dipilih secara acak.
- ◆ Berikutnya pencarian dilakukan berdasarkan proses-proses teori genetika yang memperhatikan pemikiran bagaimana memperoleh individu yang lebih baik, sehingga dalam proses evolusi dapat diharapkan diperoleh individu yang terbaik.

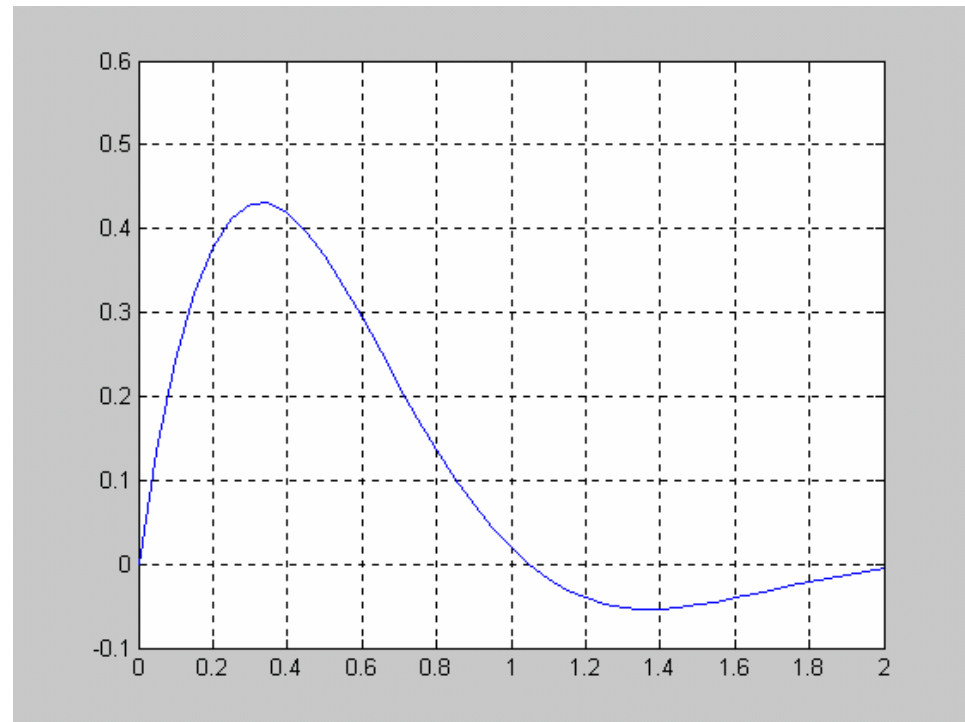


# Contoh Proses Algoritma Genetika Dalam Searching dan Optimasi



# Algoritma Genetika Untuk Mencari Nilai Maksimal Fungsi $F(x)=e^{-2x} \cdot \sin(3x)$

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

**00000000 berarti 0**  
**11111111 berarti 1**

# Definisi Individu

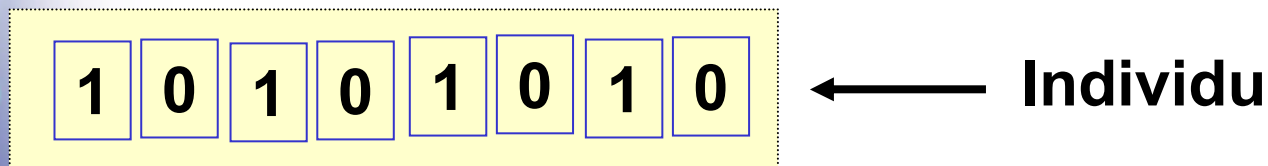
Individu dinyatakan dalam 8 gen biner, dengan batas 0 sampai dengan 1, berarti 1 bit setara dengan  $2^{-8}$ .

Sebagai contoh:

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

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

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



## 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.

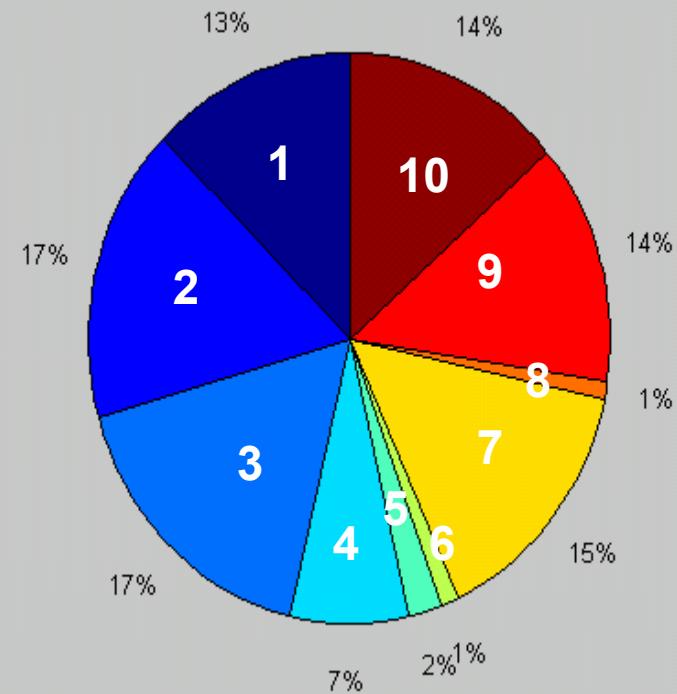
<u>Individu</u>	<u>Fitness</u>
10010000 -- 0.56250 -- 0.32244	
01001110 -- 0.30469 -- 0.43060	
01100110 -- 0.39844 -- 0.41933	
10111101 -- 0.73828 -- 0.18266	
11101000 -- 0.90625 -- 0.06699	
11110010 -- 0.94531 -- 0.04543	
00110011 -- 0.19922 -- 0.37778	
11111100 -- 0.98438 -- 0.02616	
10000111 -- 0.52734 -- 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 prosentasi 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.

	Fitness
0 0 1 1 1 0 0 1 -- 0.22266 ← induk 1	0.3968
1 0 0 1 1 0 1 0 -- 0.60156 ← induk 2	0.2921
0 0 1 1 1 0 1 1 -- 0.23050 ← anak 1	0.4022
1 0 0 1 1 0 0 0 -- 0.59382 ← anak 2	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 -- 0.22266 ← 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

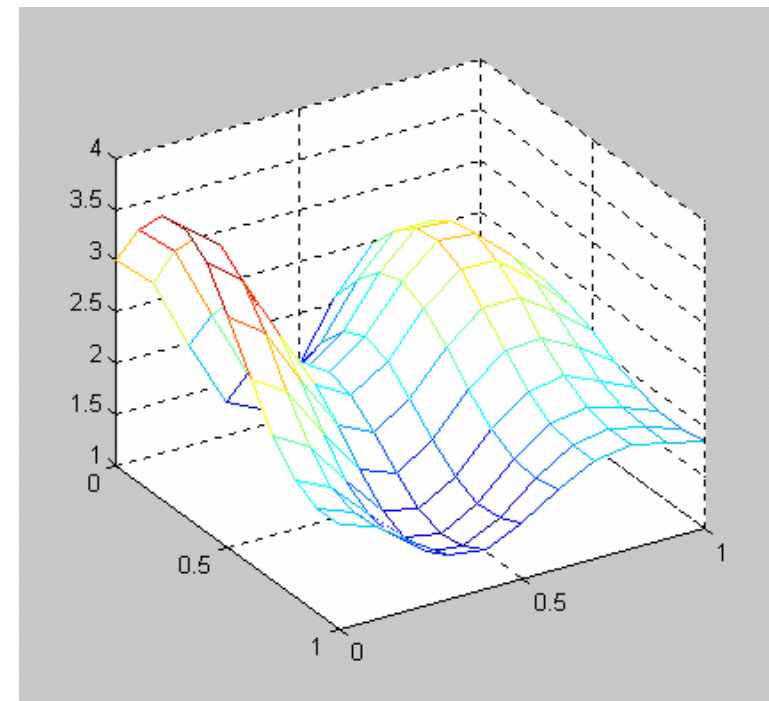
00001110 -- 0.05469 -- 0.14641

**01001110 -- 0.78906 -- 0.43060**

# Algoritma Genetika Untuk Menentukan

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

- 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
<b>0.15734</b>	<b>0.25417</b>	---	<b>3.55771</b>

# 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
<b>0.17084</b>	<b>0.25417</b>	<b>---</b>	<b>3.52082</b>
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
<b>0.17084</b>	<b>0.19691</b>	<b>---</b>	<b>3.65879</b>
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
<b>0.17084</b>	<b>0.19691</b>	<b>---</b>	<b>3.65879</b>
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
<b>0.17084</b>	<b>0.19488</b>	<b>---</b>	<b>3.66005</b>
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
<b>0.16415</b>	<b>0.17137</b>	---	<b>3.67512</b>
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
<b>0.16415</b>	<b>0.19227</b>	---	<b>3.68061</b>
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

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

$$\text{Target}=[2 \ 1 \ 19 \ 21 \ 11 \ 9]$$

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

# Definisi Individu Dan Fitness

- Individu adalah satu kata yang muncul dari proses acak tersebut, misalnya : AGHSQE atau [1 7 8 19 17 5]
- Satu individu mempunyai n gen integer yang setiap gennya menyatakan no urut alfabet.
- Nilai fitness adalah inversi dari perbedaan antara nilai kata yang muncul (individu) dan target yang ditentukan. Misalnya kata yang muncul : AGHSQE dan targetnya BASUKI maka, nilai perbedaannya:

$$E = |1-2| + |7-1| + |8-18| + |19-21| + |17-11| + |5-9|$$
$$= 1+6+10+2+6+4 = 29$$

$$\text{Fitness} = (26)(6) - 29 = 156-29 = 127$$

- Fitness didefinisikan:

$$\text{fitness}(k) = 156 - \sum_n |g(k)_n - t_n|$$

Dimana  $g_n$  adalah gen ke n dari individu ke k dan gen ke n dari target

# Pembangkitan Populasi Awal

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

```
14 20  9  1 17  3  -- N T I A Q C  >> Fitness = 83
  2  5 18  5  6  6  -- B E R E F F  >> Fitness = 127
  8  5 15 15 24  6  -- H E O O X F  >> Fitness = 120
  5 22 14 11 19 23  -- E V N K S W  >> Fitness = 95
19 19  8  6 19  7  -- S S H F S G  >> Fitness = 85
20 16  3 21  8 10  -- T P C U H J  >> Fitness = 103
19 13 12 23 15 10  -- S M L W O J  >> Fitness = 113
15 23  4 16  6 17  -- O W D P F Q  >> Fitness = 88
  5  1  6 19 21 18  -- E A F S U R  >> Fitness = 119
10 12 18  6 17  8  -- J L R F Q H  >> Fitness = 114
10  1  2  8  6 19  -- J A B H F S  >> Fitness = 103
21 18 21 24 26 19  -- U R U X Z S  >> Fitness = 90
```

# Seleksi, Cross-Over & Mutasi

- Seleksi dilakukan dengan menggunakan roulette-wheel.
- Cross-over, dilakukan dengan menukar gen-gen terpilih antar dua induk, seperti pada gen biner.

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

- Mutasi dilakukan dengan mengacak kembali nilai 1-26 dari gen yang dimutasikan.

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

# Hasil Algoritma Genetika

```
5 3 19 19 14 18 --- E C S S N R >> Fitness = 137
5 3 19 19 14 10 --- E C S S N J >> Fitness = 145
5 3 19 19 14 10 --- E C S S N J >> Fitness = 145
5 3 19 19 10 10 --- E C S S J J >> Fitness = 147
5 3 19 24 11 10 --- E C S X K J >> Fitness = 147
5 2 19 19 11 10 --- E B S S K J >> Fitness = 149
5 2 19 22 10 10 --- E B S V J J >> Fitness = 149
5 2 19 22 11 10 --- E B S V K J >> Fitness = 150
5 2 19 22 11 10 --- E B S V K J >> Fitness = 150
5 2 19 21 11 10 --- E B S U K J >> Fitness = 151
5 2 19 21 11 10 --- E B S U K J >> Fitness = 151
5 2 19 21 11 10 --- E B S U K J >> Fitness = 151
1 2 19 21 11 10 --- A B S U K J >> Fitness = 153
2 2 19 21 11 10 --- B B S U K J >> Fitness = 154
2 2 19 21 11 10 --- B B S U K J >> Fitness = 154
2 2 19 21 11 9 --- B B S U K I >> Fitness = 155
2 2 19 21 11 9 --- B B S U K I >> Fitness = 155
2 2 19 21 11 9 --- B B S U K I >> Fitness = 155
2 2 19 21 11 9 --- B B S U K I >> Fitness = 155
2 1 19 21 11 9 --- B A S U K I >> Fitness = 156
```