

# Genetic Algorithm (GA)

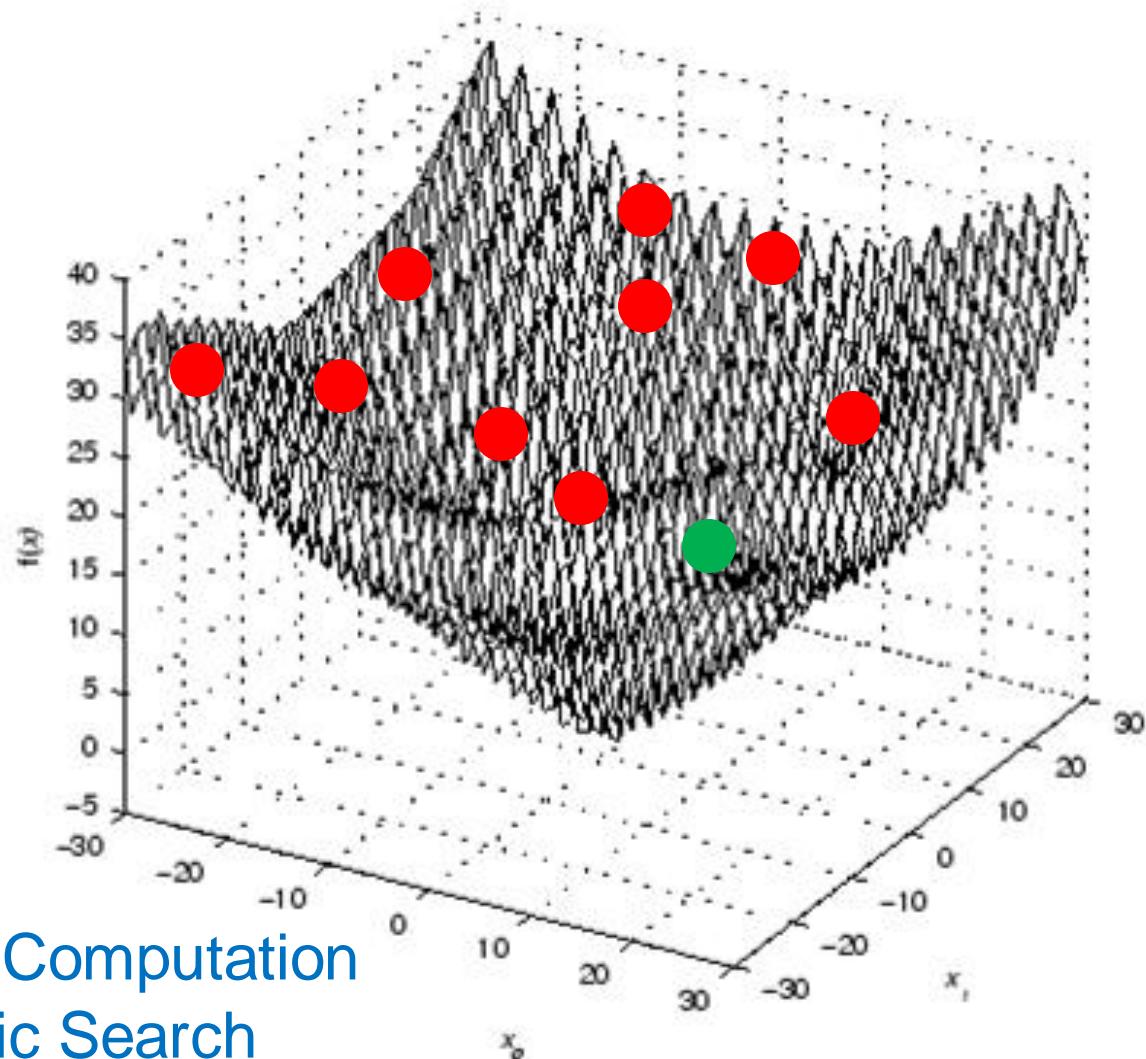
Dr. Suyanto, S.T., M.Sc.  
HP/WA: 0812 845 12345

Intelligence Computing Multimedia (ICM)  
Informatics faculty – Telkom University

# Simple GA

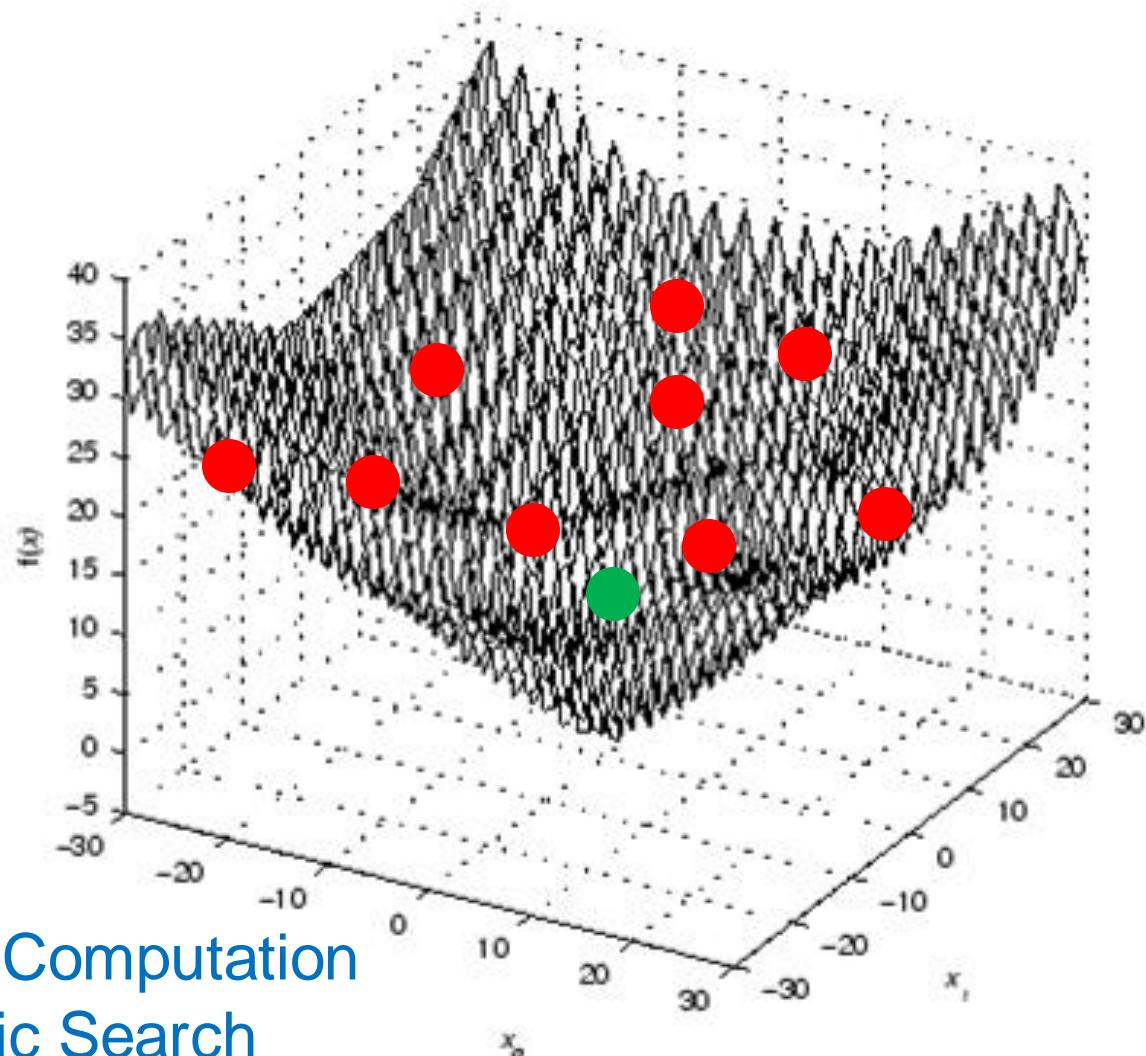
Representasi	Pengkodean biner ( <i>binary encoding</i> )
Seleksi orangtua	Proporsional terhadap nilai <i>fitness</i>
Rekombinasi	<i>N-point</i> atau seragam ( <i>uniform</i> )
Mutasi	Pembalikan bit dengan probabilitas tetap dan bersifat bebas ( <i>independent</i> ) pada masing-masing bit.
Seleksi survivor	Semua individu baru menggantikan semua individu lama ( <i>generational replacement</i> )
Ciri khusus	Lebih menekankan pada rekombinasi

$$f(\vec{x}) = \sum_{i=0}^{D-1} \left( e^{-0.2\sqrt{x_i^2 + x_{i+1}^2}} + 3(\cos(2x_i) + \sin(2x_{i+1})) \right)$$



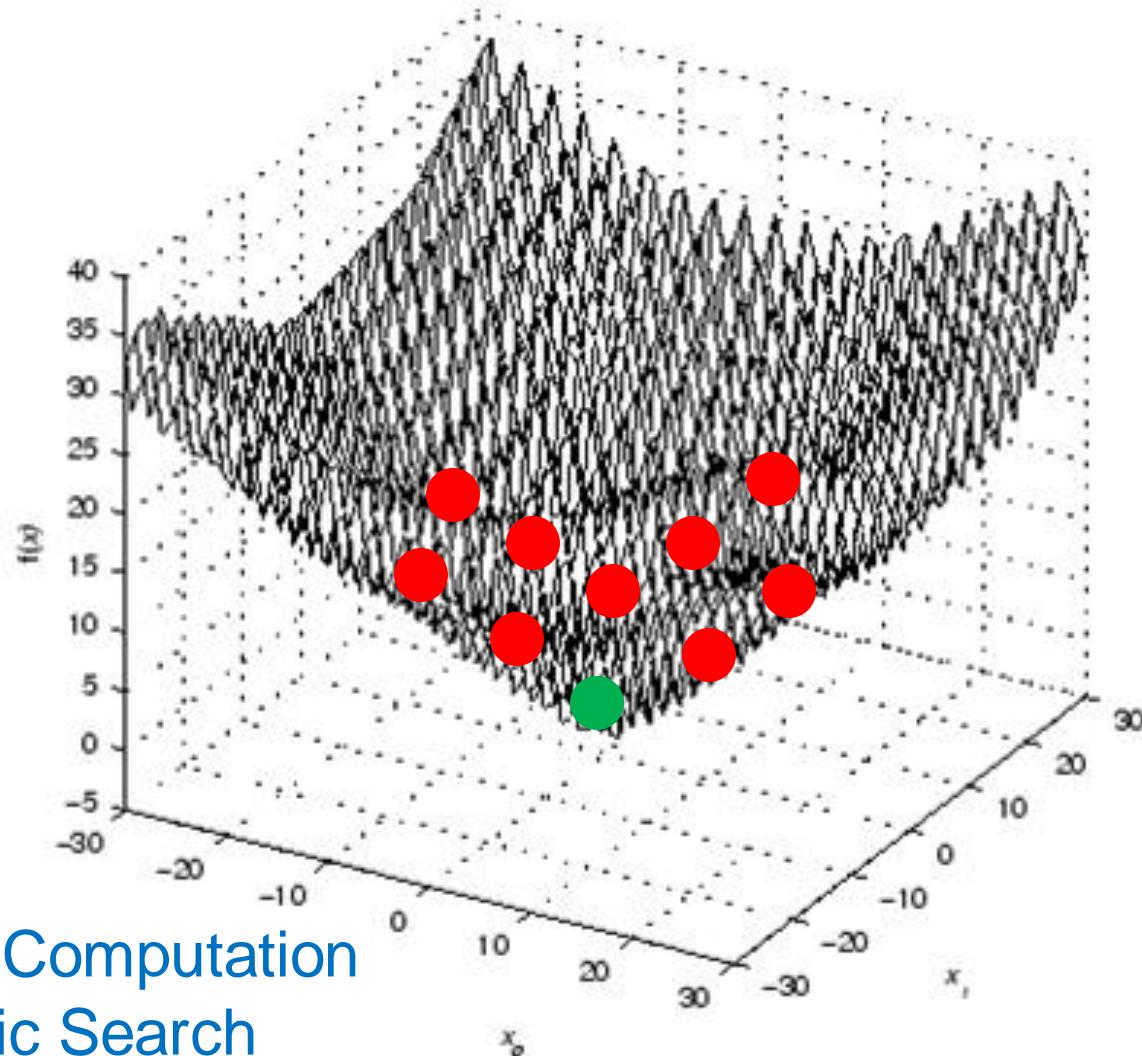
Evolutionary Computation  
Meta Heuristic Search

$$f(\vec{x}) = \sum_{i=0}^{D-1} \left( e^{-0.2\sqrt{x_i^2 + x_{i+1}^2}} + 3(\cos(2x_i) + \sin(2x_{i+1})) \right)$$



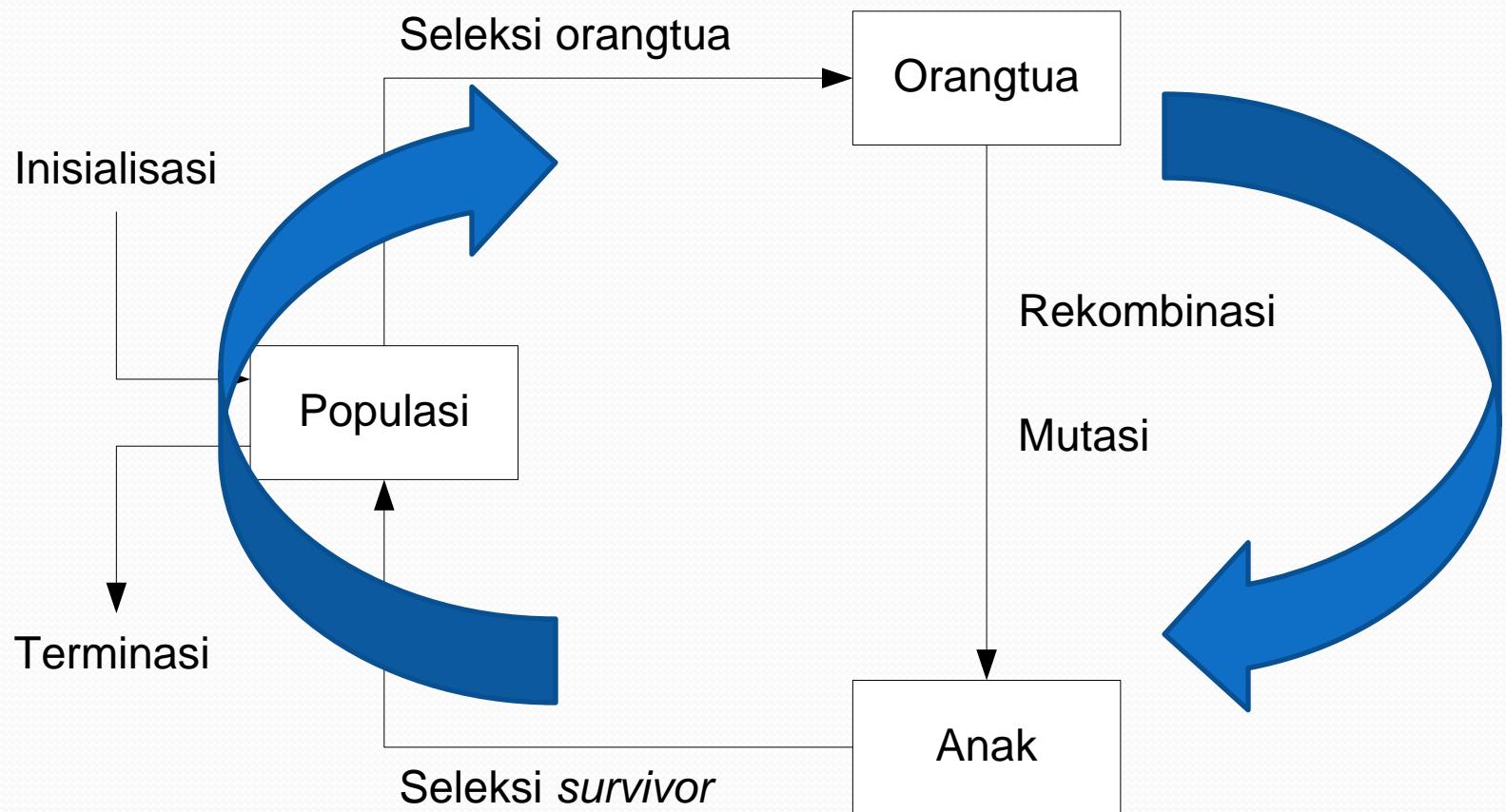
Evolutionary Computation  
Meta Heuristic Search

$$f(\vec{x}) = \sum_{i=0}^{D-1} \left( e^{-0,2} \sqrt{x_i^2 + x_{i+1}^2} + 3(\cos(2x_i) + \sin(2x_{i+1})) \right)$$



Evolutionary Computation  
Meta Heuristic Search

# Skema umum EAs



# Bekerja dengan GA

- Buat skema pengkodean Individu → Kromosom
- Bangun fungsi fitness
- Definisikan operator GA

# Studi kasus: Minimasi fungsi

Nilai minimum  $h = ?$

$$h(x_1, x_2) = x_1^2 + x_2^2$$

$$x_1, x_2 \in [-5,12; 5,12]$$

# Individu

x <sub>1</sub>										x <sub>2</sub>									
1	1	0	1	0	1	0	1	0	0	0	1	0	0	1	1	0	1	0	1
$g_1$										$g_{10}$	$g_{11}$							$g_{20}$	

# *Fitness*

$$f = \frac{1}{(x_1^2 + x_2^2) + 0,01}$$

Jika nilai minimum = 0, nilai maks  $f = ?$

# Generasi 1

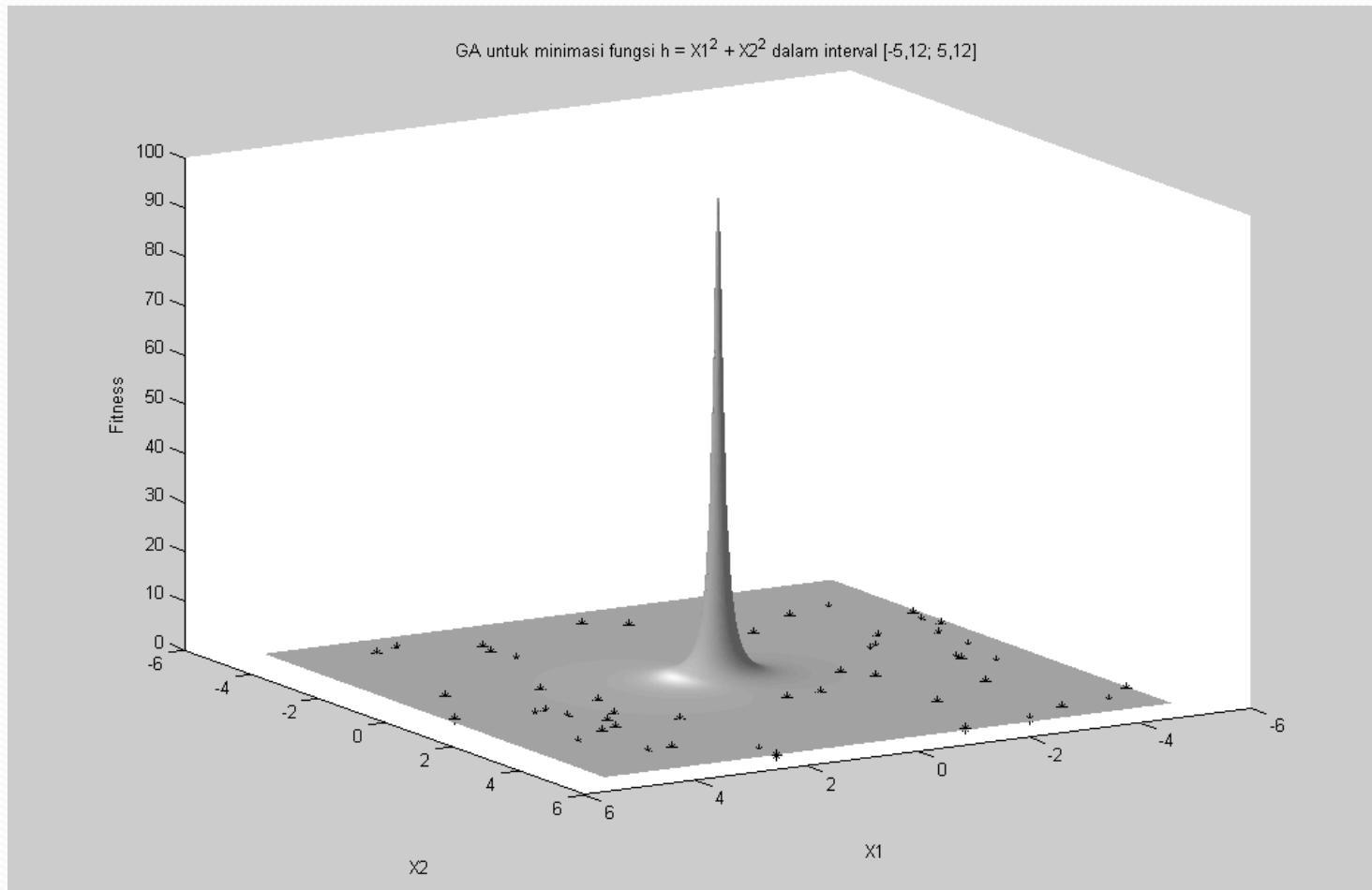
$$x = r_b + \frac{(r_a - r_b)}{\sum_{i=1}^N 2^{-i}} (g_1 \cdot 2^{-1} + g_2 \cdot 2^{-2} + \dots + g_N \cdot 2^{-N})$$

$$x_1, x_2 \in [-5,12; 5,12]$$



No	Genotype	Phenotype		Nilai fitness
	kromosom biner	X1	X2	
1	00010011011001101110	-4.35	1.1	0.049646
2	11001101110001000011	3.11	-4.45	0.033916
3	10110010111111001110	2.03	4.62	0.039254
4	11001110001101111101	3.12	3.81	0.041219
5	11001110101011011001	3.14	2.17	0.068594
6	00101110000110110110	-3.28	-0.74	0.08837
7	01111011111010110010	-0.17	1.78	0.31179
50	11010110011000111011	3.45	0.59	0.081562

# Generasi 1



# Generasi 10

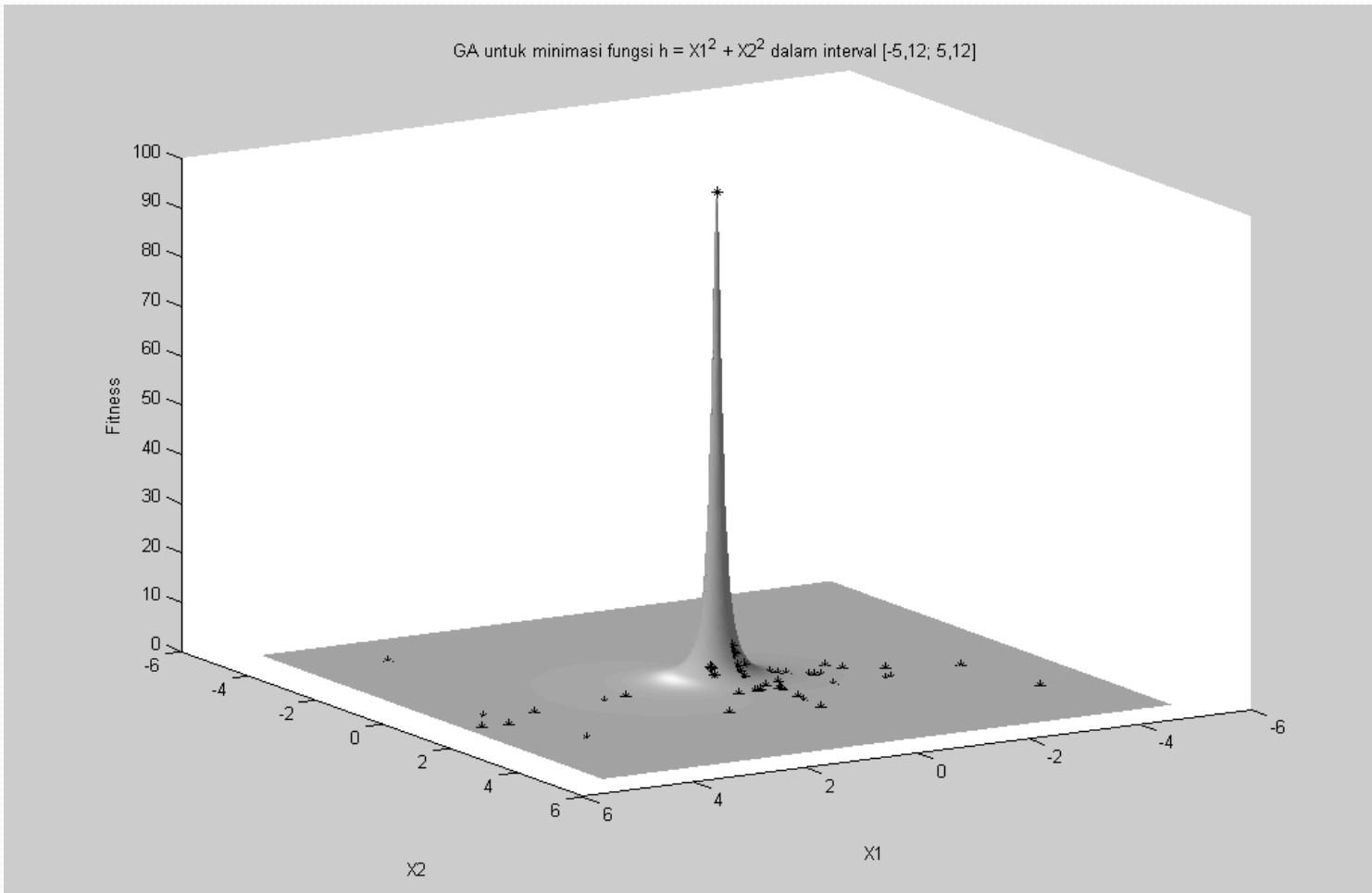
$$x = r_b + \frac{(r_a - r_b)}{\sum_{i=1}^N 2^{-i}} (g_1 \cdot 2^{-1} + g_2 \cdot 2^{-2} + \dots + g_N \cdot 2^{-N})$$

$$x_1, x_2 \in [-5,12; 5,12]$$

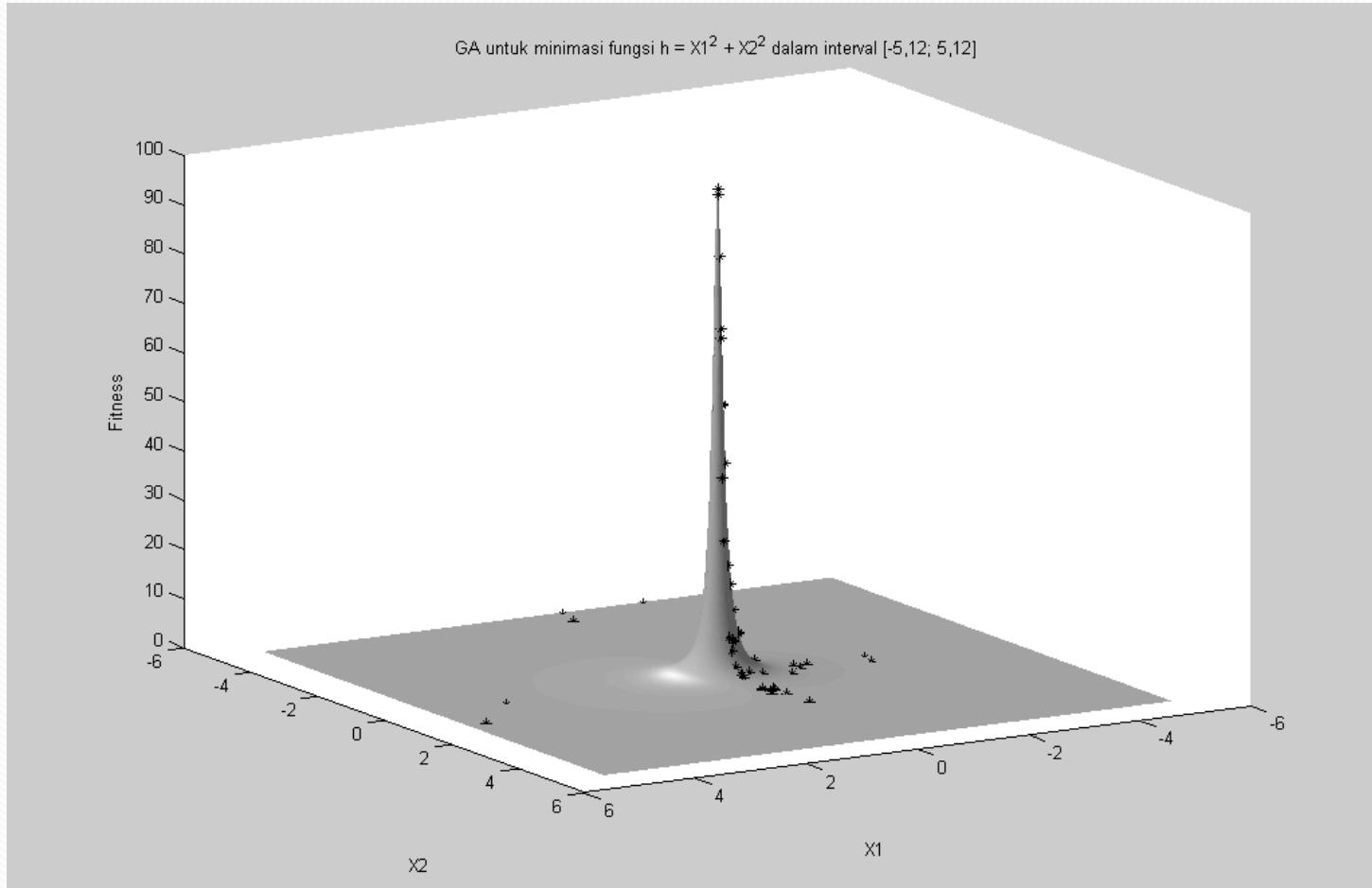


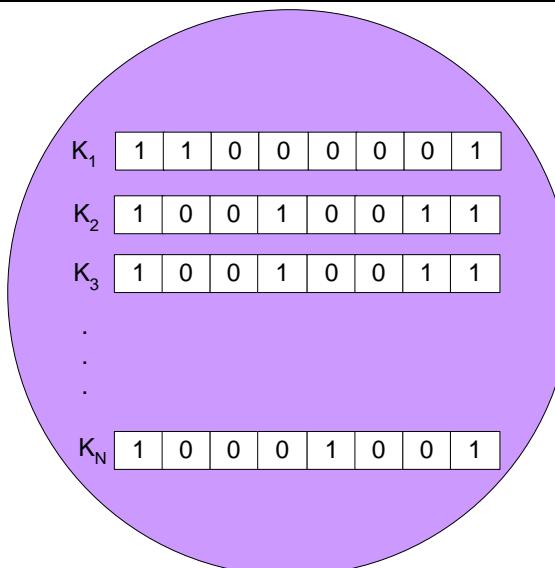
No	Genotype	Phenotype		Nilai fitness
	kromosom biner	X1	X2	
1	01111111110000000000	-0.01	0	99.01
2	01111111110000000000	-0.01	0	99.01
3	01111101010001000001	-3.77	1.03	0.065429
4	01111101011001110001	-2.4	1.2	0.1387
5	01111001111000100001	3.58	0.52	0.076355
6	01011101111000101010	4.83	1.01	0.041053
7	01111101111000100001	-1.38	1.2	0.29812
...				
50	01111101111000000001	-1.93	0.02	0.26772

# Generasi 10

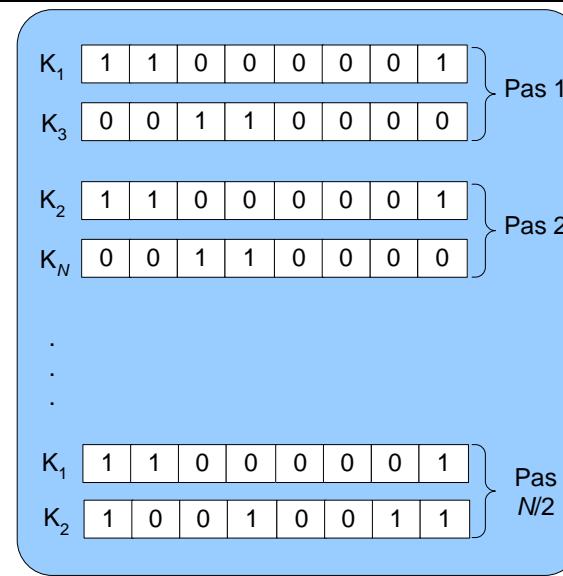
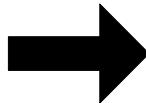


# Generasi 100



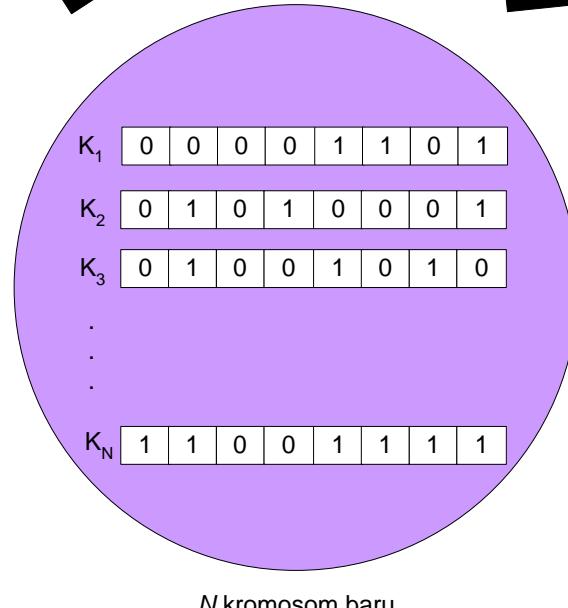


Seleksi orangtua



Penggantian  $N$  kromosom

Rekombinasi dan Mutasi



Seleksi Survivor: *Generational*

A photograph of a large, multi-tiered fountain in a city square. In the center of the fountain is a bronze statue of three figures. The water jets from the fountain are spraying upwards and outwards. In the background, there are several modern skyscrapers, including one with a prominent curved glass facade on the left and a tall, dark building on the right. The sky is blue with some white clouds.

Demo: Simple GA

# Studi kasus: Minimasi fungsi

Nilai minimum  $h = ?$

$$h(x_1, x_2) = x_1^2 + x_2^2$$

$$x_1, x_2 \in [-5,12;5,12]$$

# Komponen GA

- Inisialisasi Populasi ( $N$  kromosom)
- Evaluasi Individu (berbasis fungsi fitness)
- Seleksi Ortu
- Rekombinasi
- Mutasi
- Seleksi Survivor

Implementasinya dalam Matlab?

# Representasi Biner

**function** Populasi = InisialisasiPopulasiBiner(UkPop,JumGen)

Populasi = fix(2\*rand(UkPop,JumGen));

**Misal: UkPop = 5; JumGen = 10;**

1	0	1	1	0	0	1	1	1	0
1	0	1	1	0	0	0	1	0	0
1	0	0	0	1	0	0	1	1	0
1	0	1	0	1	1	0	0	1	1
1	1	1	1	0	0	0	0	1	1

```
function x= DekodeKromosomBiner(Kromosom,Nvar,Ng,Ra,Rb)
```

```
MaxSum = 0;
```

```
for kk=1:Ng,
```

```
    MaxSum = MaxSum + 2^(-kk);
```

```
end
```

```
for ii=1:Nvar,
```

```
    x(ii) = 0;
```

```
for jj=1:Ng,
```

```
    x(ii) = x(ii) + Kromosom((ii-1)*Ng+jj)*2^(-jj);
```

```
end
```

```
    x(ii) = Rb + ((Ra-Rb)/MaxSum) * x(ii);
```

```
end
```

Individu, misal  
X1 dan x2

kode genetik,  
misal biner

# *Fitness*

$$f = \frac{1}{(x_1^2 + x_2^2) + 0,01}$$

Jika nilai minimum = 0, nilai maks  $f = ?$

# Evaluasi Individu

```
function fitness = EvaluasiIndividu(x,BilKecil)
```

```
fitness = 1 / (x(1)^2 + x(2)^2 + BilKecil);
```

```
function Pindex = RouletteWheel(UkPop,LinearFitness)
```

```
JumFitness = sum(LinearFitness);
```

```
KumulatifFitness = 0;
```

```
RN = rand;
```

```
ii = 1;
```

```
while ii <= UkPop,
```

```
    KumulatifFitness = KumulatifFitness + LinearFitness(ii);
```

```
    if (KumulatifFitness/JumFitness) > RN,
```

```
        Pindex = ii;
```

```
        break;
```

```
    end
```

```
    ii = ii + 1;
```

```
end
```

# Rekombinasi Biner

```
function Anak = RekombinasiBiner(Ortu1,Ortu2,JumGen)
```

```
TP = 1 + fix(rand*(JumGen-1));
```

```
Anak(1,:) = [Ortu1(1:TP) Ortu2(TP+1:JumGen)];
```

```
Anak(2,:) = [Ortu2(1:TP) Ortu1(TP+1:JumGen)];
```

# Mutasi Biner

```
function MutKrom = MutasiBiner(Kromosom,JumGen,Pmutasi)
```

```
MutKrom = Kromosom;
```

```
for ii=1:JumGen,
```

```
    if (rand < Pmutasi),
```

```
        if Kromosom(ii)==0,
```

```
            MutKrom(ii) = 1;
```

```
        else
```

```
            MutKrom(ii) = 0;
```

```
        end
```

```
    end
```

```
end
```

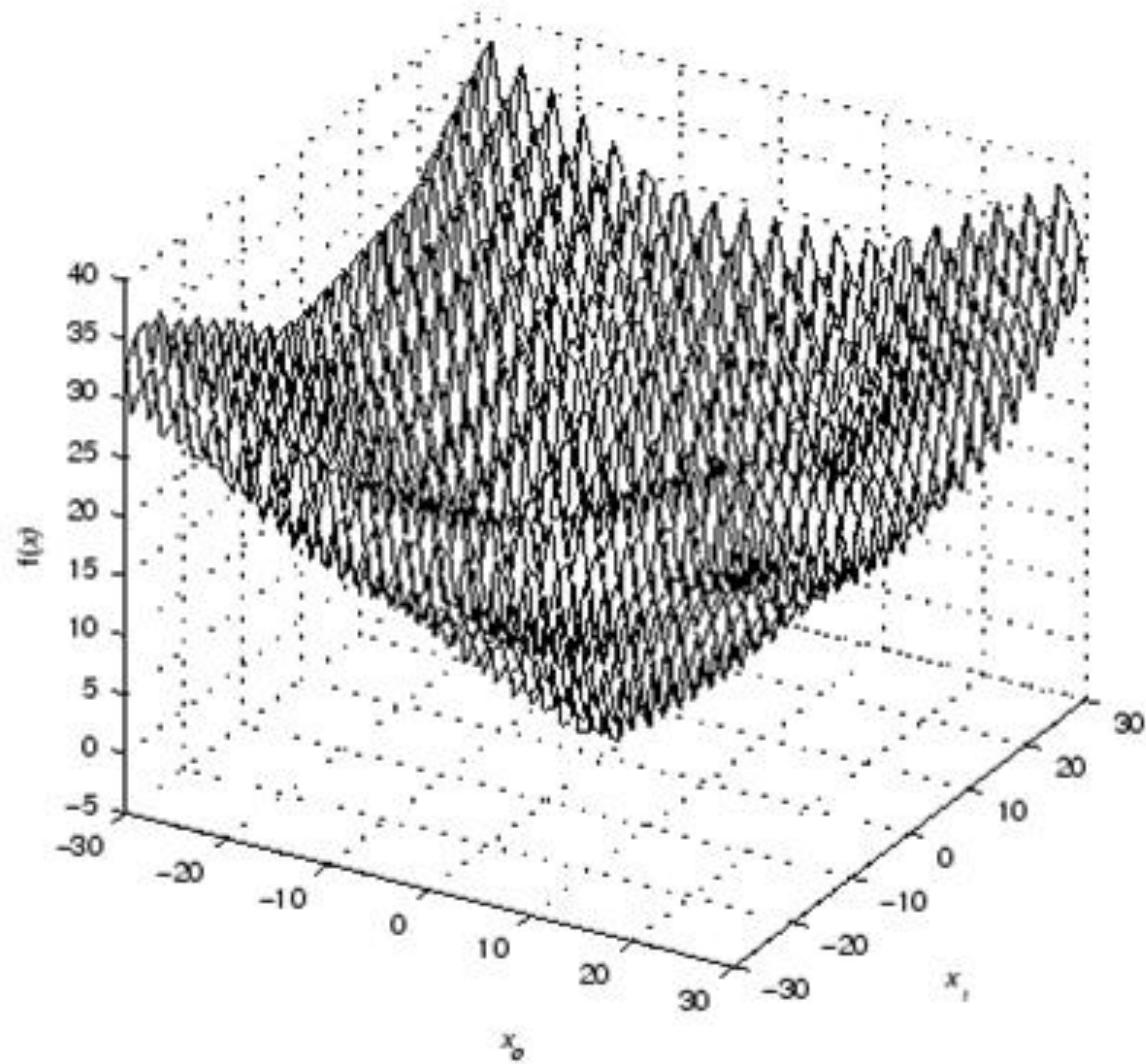
A photograph of a large, multi-tiered fountain in a city square. In the center of the fountain is a bronze statue of three figures. The water jets from the fountain are spraying upwards and outwards. In the background, there are several modern skyscrapers, including one with a prominent curved glass facade on the left and a tall, dark building on the right. The sky is blue with some white clouds.

Demo: Simple GA



# Question?

$$f(\vec{x}) = \sum_{i=0}^{D-1} \left( e^{-0.2\sqrt{x_i^2 + x_{i+1}^2}} + 3(\cos(2x_i) + \sin(2x_{i+1})) \right)$$



# Representasi individu → kromosom

- Representasi Biner
- Representasi Integer
- Representasi Real
- Representasi Permutasi

# Representasi Biner

**function** Populasi = InisialisasiPopulasiBiner(UkPop,JumGen)

Populasi = fix(2\*rand(UkPop,JumGen));

**Misal: UkPop = 5; JumGen = 10;**

1	0	1	1	0	0	1	1	1	0
1	0	1	1	0	0	0	1	0	0
1	0	0	0	1	0	0	1	1	0
1	0	1	0	1	1	0	0	1	1
1	1	1	1	0	0	0	0	1	1

# Representasi Integer

**function** Populasi = InisialisasiPopulasInteger(UkPop,JumGen)

Populasi = fix(9\*rand(UkPop,JumGen));

**Misal: UkPop = 5; JumGen = 10;**

4	3	5	7	4	7	6	6	7	0
7	5	3	3	2	6	7	5	1	6
6	5	1	7	8	1	6	5	3	1
2	2	5	0	6	8	3	0	2	8
3	6	5	5	1	3	6	2	6	0

# Representasi Real

**function** Populasi = InisialisasiPopulasiReal(UkPop,JumGen)

Populasi = rand(UkPop,JumGen);

**Misal: UkPop = 5; JumGen = 10;**

0,405102	0,313349	0,234897	0,84867	0,291679	0,249295	0,314046	0,305683	0,726449	0,61309
0,779813	0,985409	0,605045	0,486053	0,068964	0,196639	0,851981	0,294056	0,585907	0,622961
0,449634	0,197545	0,415529	0,998973	0,177705	0,467867	0,523253	0,481516	0,330538	0,225416
0,332314	0,339029	0,709216	0,031305	0,118845	0,252562	0,546119	0,499327	0,859106	0,752103
0,433246	0,134945	0,926929	0,466493	0,056166	0,188175	0,896785	0,665138	0,588188	0,435401

# Representasi Permutasi

```
function Populasi = TSPInisialisasiPopulasi(UkPop,JumGen)
for ii=1:UkPop,
    [Xval,Ind] = sort(rand(1,JumGen));
    Populasi(ii,:) = Ind;
end
```

**Misal: UkPop = 5; JumGen = 10;**

10	2	9	3	6	4	5	8	1	7
1	5	6	8	4	9	7	3	2	10
6	10	3	4	2	8	1	7	5	9
9	5	8	3	6	2	7	10	1	4
2	1	8	10	4	5	6	9	3	7

```
function x= DekodeKromosomBiner(Kromosom,Nvar,Ng,Ra,Rb)
```

```
MaxSum = 0;
```

```
for kk=1:Ng,
```

```
    MaxSum = MaxSum + 2^(-kk);
```

```
end
```

```
for ii=1:Nvar,
```

```
    x(ii) = 0;
```

```
for jj=1:Ng,
```

```
    x(ii) = x(ii) + Kromosom((ii-1)*Ng+jj)*2^(-jj);
```

```
end
```

```
    x(ii) = Rb + ((Ra-Rb)/MaxSum) * x(ii);
```

```
end
```

Individu, misal  
X1 dan x2

kode genetik,  
misal biner

```
function x = DekodeKromosomInteger(Kromosom,Nvar,Ng,Ra,Rb)

MaxSum = 0;
for kk=1:Ng,
    MaxSum = MaxSum + (9 * 10^(-kk));
end
for ii=1:Nvar,
    x(ii) = 0;
    for jj=1:Ng,
        x(ii) = x(ii) + Kromosom((ii-1)*Ng+jj)*10^(-jj);
    end
    x(ii) = Rb + ((Ra-Rb)/MaxSum) * x(ii);
end
```

```
function x = DekodeKromosomReal(Kromosom,Nvar,Ng,Ra,Rb)  
  
for ii=1:Nvar,  
    x(ii) = Rb + ((Ra - Rb) * Kromosom(ii));  
end
```

# *Fitness*

$$f = \frac{1}{(x_1^2 + x_2^2) + 0,01}$$

Jika nilai minimum = 0, nilai maks  $f = ?$

# Evaluasi Individu

```
function fitness = EvaluasiIndividu(x,BilKecil)
```

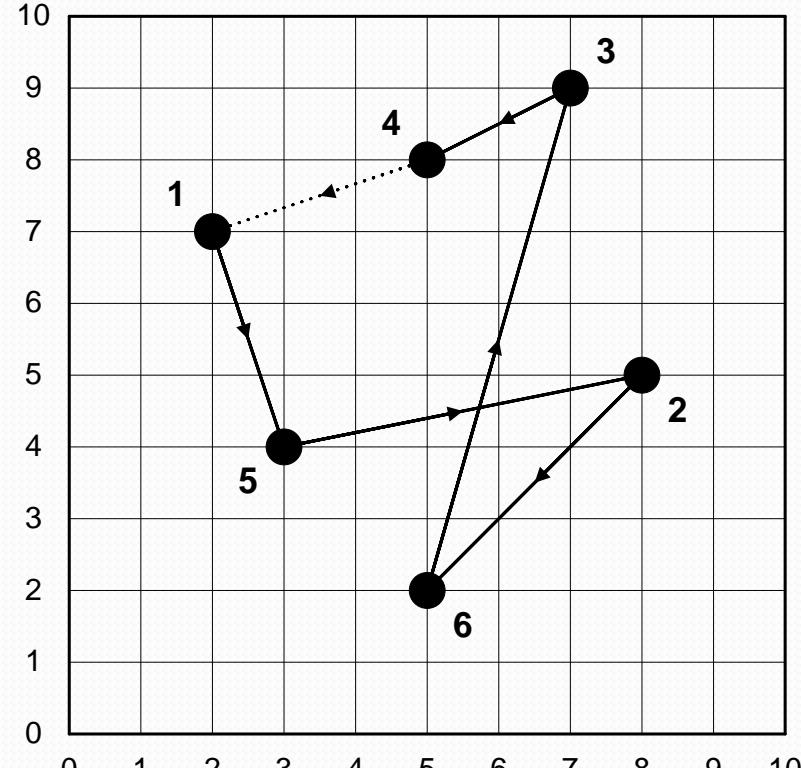
```
fitness = 1 / (x(1)^2 + x(2)^2 + BilKecil);
```

# Kasus TSP

Fitness?

Total biaya kunjungan

Individu: urutan kunjungan semua lokasi



Nilai gen menyatakan nomor lokasi

Posisi gen menyatakan urutan kunjungan

```
function fitness = TSPEvaluasilIndividu(Kromosom,JumGen,XYkota)
```

```
TB = 0;
```

```
for ii=1:JumGen-1,
```

```
    TB = TB + norm(XYkota(Kromosom(ii),:) -  
        XYkota(Kromosom(ii+1),:));
```

```
end
```

```
% Jalur harus kembali ke kota asal
```

```
TB = TB + norm(XYkota(Kromosom(JumGen),:) -  
    XYkota(Kromosom(1),:));
```

```
fitness = 1 / TB;
```

```
function Pindex = RouletteWheel(UkPop,LinearFitness)
```

```
JumFitness = sum(LinearFitness);
```

```
KumulatifFitness = 0;
```

```
RN = rand;
```

```
ii = 1;
```

```
while ii <= UkPop,
```

```
    KumulatifFitness = KumulatifFitness + LinearFitness(ii);
```

```
    if (KumulatifFitness/JumFitness) > RN,
```

```
        Pindex = ii;
```

```
        break;
```

```
    end
```

```
    ii = ii + 1;
```

```
end
```

```

function IndTerpilih = TournamentSelection(UkPop, Fitness, UkTour, ProbTour)

for ii=1:UkTour, % pilih kontestan sebanyak UkTour
    IndTemp(ii) = 1 + fix(rand*UkPop); FitTemp(ii) = Fitness(IndTemp(ii));
end
[M, IndTerbaik] = max(FitTemp); IndLainnya = []; % cari individu terbaik
if IndTerbaik==1,
    IndLainnya = [IndLainnya 2:UkTour];
else
    IndLainnya = [IndLainnya 1:IndTerbaik-1];
    IndLainnya = [IndLainnya IndTerbaik+1:UkTour];
end
if rand < ProbTour, % Individu terbaik terpilih jika memenuhi ProbTour
    IndTerpilih = IndTemp(IndTerbaik);
else
    IndTerpilih = IndTemp(IndLainnya(1+fix(rand*(UkTour-1))));
```

**end**

# Rekombinasi Biner/Integer

```
function Anak = RekombinasiBiner(Ortu1,Ortu2,JumGen)

TP = 1 + fix(rand*(JumGen-1));
Anak(1,:) = [Ortu1(1:TP) Ortu2(TP+1:JumGen)];
Anak(2,:) = [Ortu2(1:TP) Ortu1(TP+1:JumGen)];
```

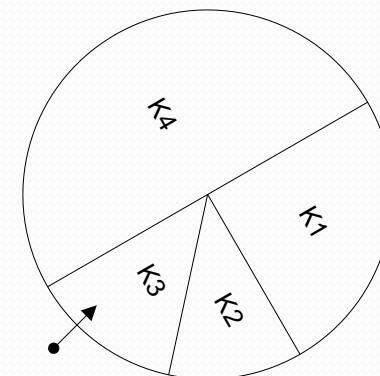
# Seleksi Orangtua

- Fitness Proportionate Selection (FPS)
  - *Roulette wheel*
  - **Baker's SUS** (*Stochastic Universal Sampling*)
- Rank-Based Selection
  - Linear Ranking
  - Non-linear ranking
- Tournament Selection

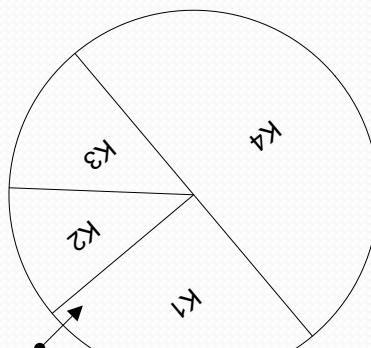
# Fitness Proportionate Selection (FPS)

Metode: *roulette wheel*

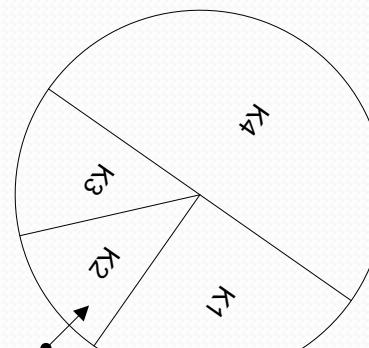
Kromosom	Fitness
K1	2
K2	1
K3	1
K4	4
Jumlah	8



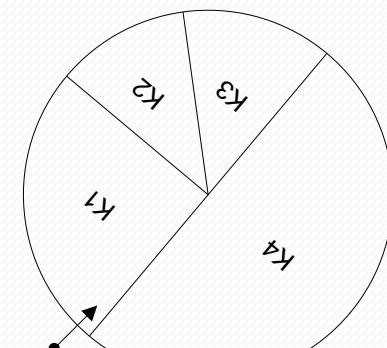
Putaran ke-1



Putaran ke-2



Putaran ke-3

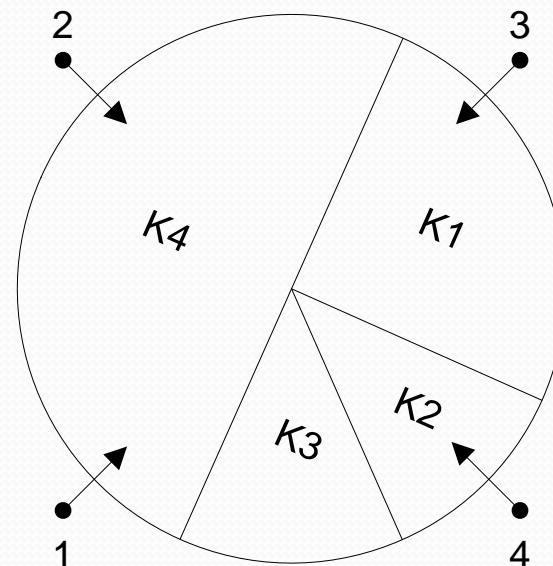


Putaran ke-4

# Fitness Proportionate Selection (FPS)

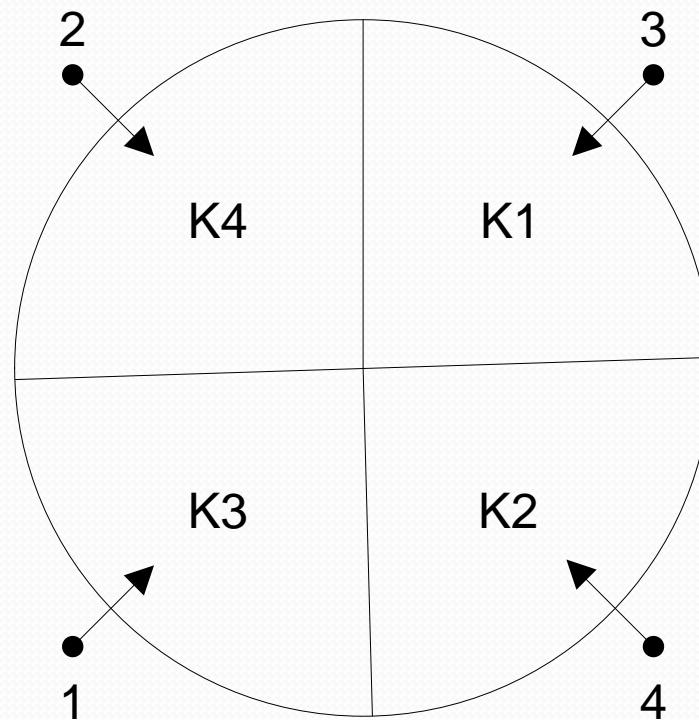
Metode: **Baker's SUS** (*Stochastic Universal Sampling*)

Kromosom	<i>Fitness</i>
K1	2
K2	1
K3	1
K4	4
Jumlah	8



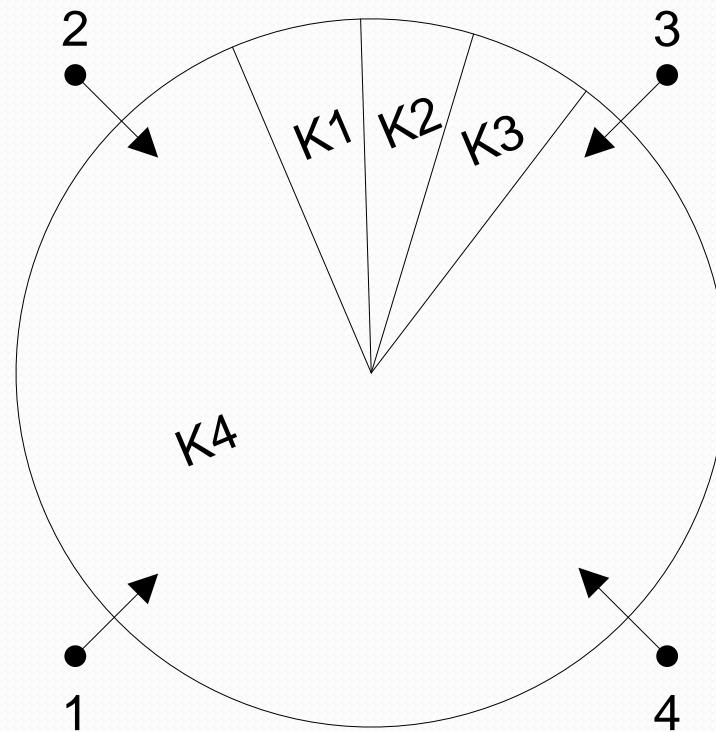
# Kelemahan Baker's SUS

Kromosom	<i>Fitness</i>
K1	1,98
K2	2,01
K3	1,99
K4	2,02
Jumlah	8



# Kelemahan Baker's SUS

Kromosom	<i>Fitness</i>
K1	1
K2	1
K3	1
K4	17
Jumlah	20



# Untuk mengatasi Baker's SUS

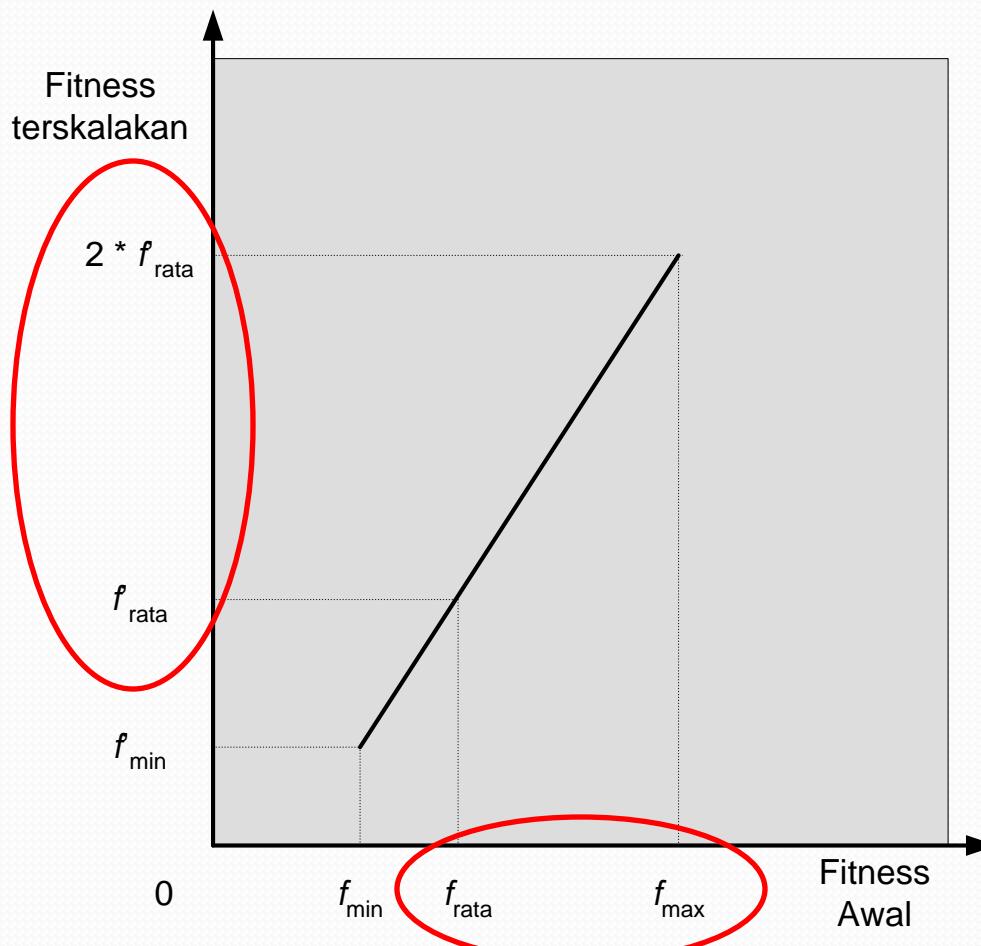
- *Linear Scaling*
- *Window Scaling*
- *Sigma Scaling*

# *Linear Scaling*

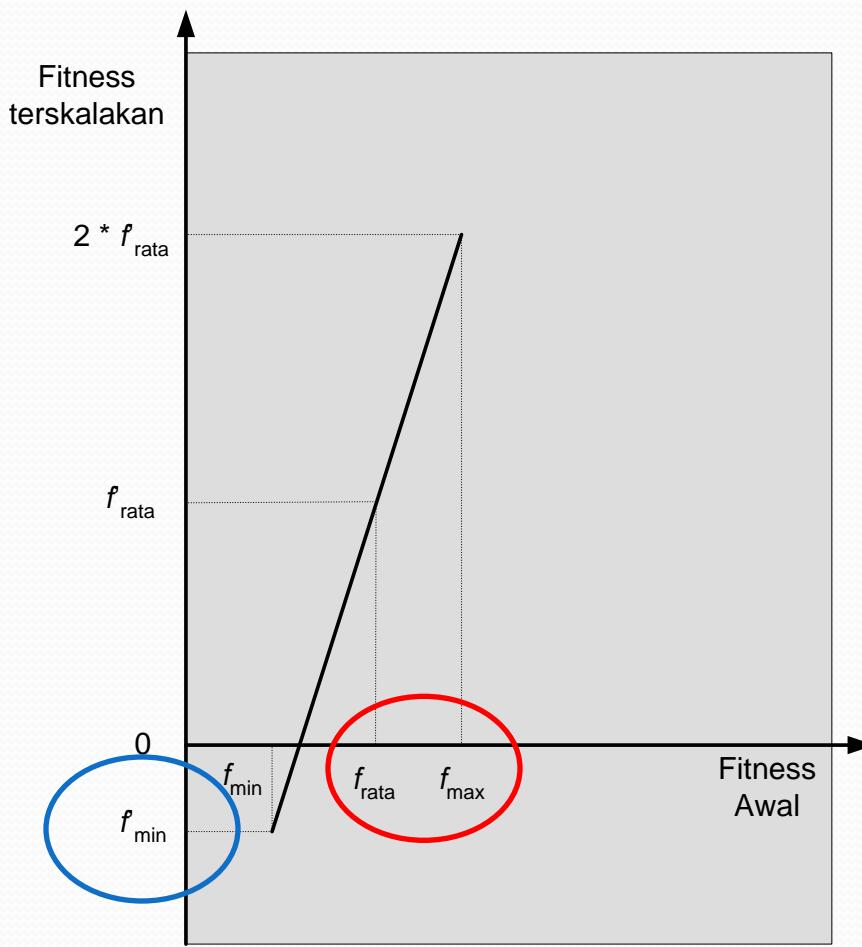
$$f'_i = af_i + b$$

- $f_i$  = nilai *fitness* individu ke-*i*
- $a$  dan  $b$  dipilih sedemikian hingga  $\bar{f}' = \bar{f}$  dan
- $f'_{\max} = C_{mult} * \bar{f}$ , dimana  $C_{mult}$  adalah nilai harapan berapa kali kromosom terbaik terpilih sebagai orangtua
- Untuk ukuran populasi sebesar 50 sampai 100 individu, biasanya  $C_{mult}$  antara 1,2 sampai 2.

# Linear Scaling



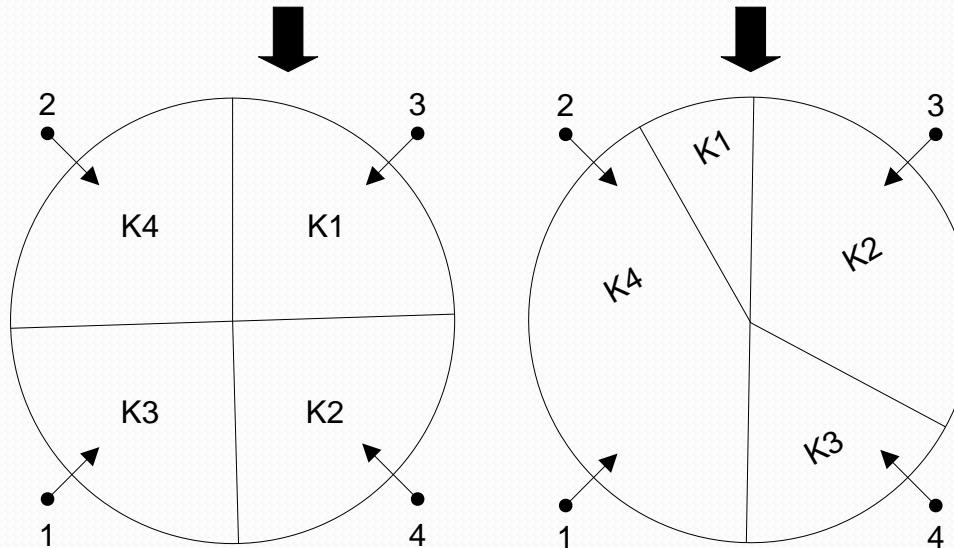
# Kelemahan *Linear Scaling*



# Window Scaling

Misalkan *fitness* terendah selama 5 generasi terakhir = 1,97

Kromosom	<i>f</i>	<i>f'</i>
K1	1,98	0,01
K2	2,01	0,04
K3	1,99	0,02
K4	2,02	0,05
Jumlah	8	0,12



# *Sigma Scaling*

$$f'_i = f_i + (\bar{f} - c * \sigma)$$

- $c$  = bilangan bulat kecil (biasanya sama dengan 2)
- Nilai-nilai  $fitness f$  diskalakan menggunakan rata-rata dan standar deviasi

# Rank-Based Selection

- *Linear Ranking*
- *Non-linear ranking*

# *Linear Ranking (LR)*

- Semua individu yang berada dalam populasi diurutkan berdasarkan nilai fitnessnya secara *ascending*
- Nilai *fitness* hasil perankingan dihitung menggunakan rumus:

$$f'(Pos) = (2 - S) + 2(S - 1) \frac{(Pos - 1)}{(N - 1)}$$

- $S$  adalah ***selective pressure*** (probabilitas terpilih individu terbaik dibandingkan dengan rata-rata probabilitas terpilih semua individu).  $S$  berada dalam interval  $[1, 2]$ .
- $Pos$  adalah posisi individu dalam populasi, dimana individu terburuk (nilai *fitness*-nya paling rendah) berada di posisi 1 dan individu terbaik di posisi  $N$ .

# *Non Linear Ranking (NLR)*

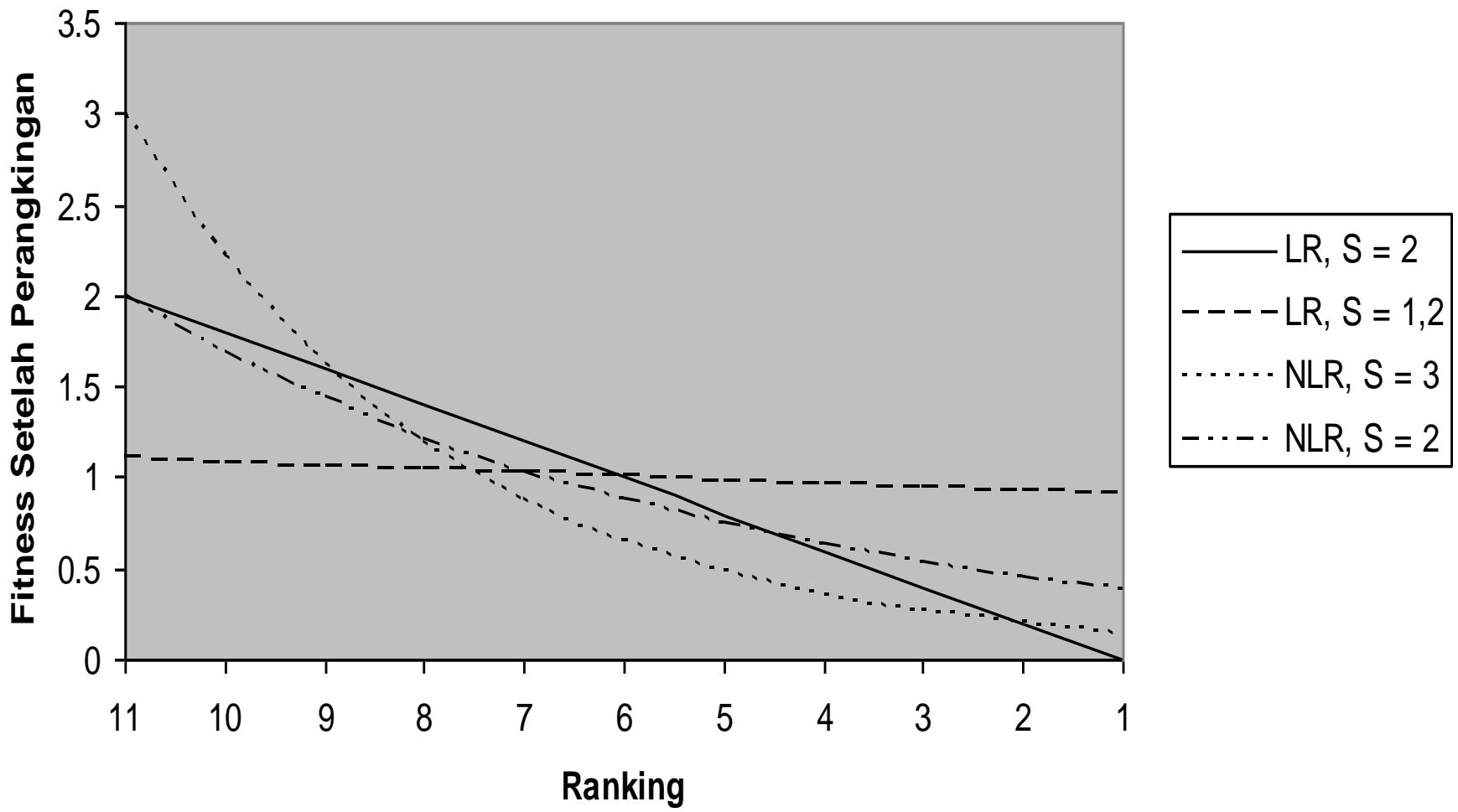
$$f'(Pos) = \frac{NX^{Pos-1}}{\sum_{i=1}^N X^{i-1}}$$

dimana  $X$  adalah akar dari polinomial:

$$0 = (S - N)X^{N-1} + SX^{N-2} + \dots + SX + S$$

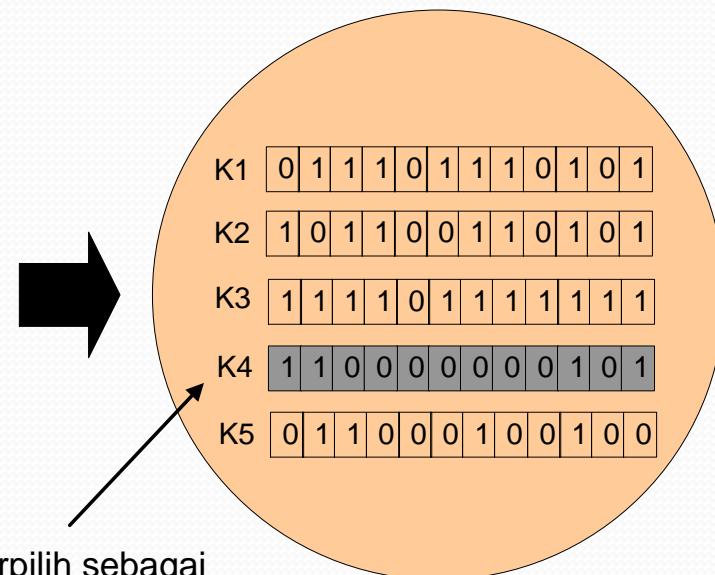
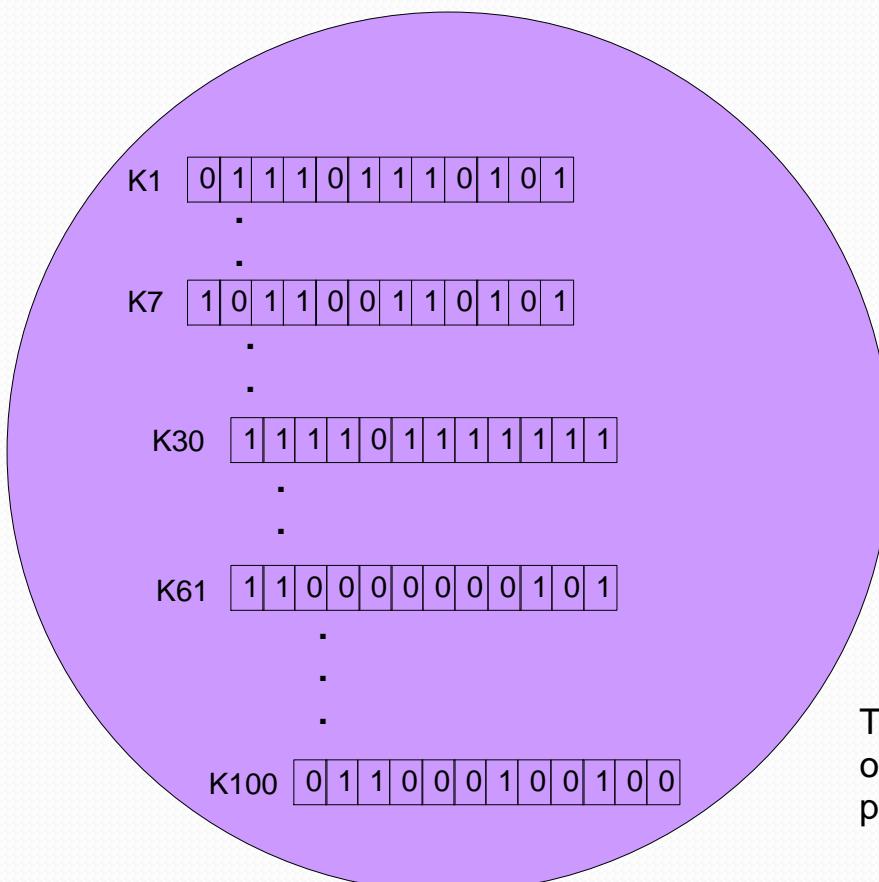
$N$  = ukuran populasi

$S$  = *selective pressure* dalam interval  $[1, N-2]$ .



<b><i>Fitness</i></b>	<b><i>Posisi</i></b>	<b><i>Fitness Perangkingan</i></b>			
		<b>LR, S = 2</b>	<b>LR, S = 1,2</b>	<b>NLR, S = 3</b>	<b>NLR, S = 2</b>
90,11	11	2	1,1	3	2
90,09	10	1,8	1,08	2,21	1,69
90,08	9	1,6	1,06	1,62	1,43
90,06	8	1,4	1,04	1,19	1,21
90,05	7	1,2	1,02	0,88	1,03
89,97	6	1	1	0,65	0,87
89,96	5	0,8	0,98	0,48	0,74
89,95	4	0,6	0,96	0,35	0,62
79,94	3	0,4	0,94	0,26	0,53
79,93	2	0,2	0,92	0,19	0,45
79,91	1	0	0,9	0,14	0,38

# Tournament Selection

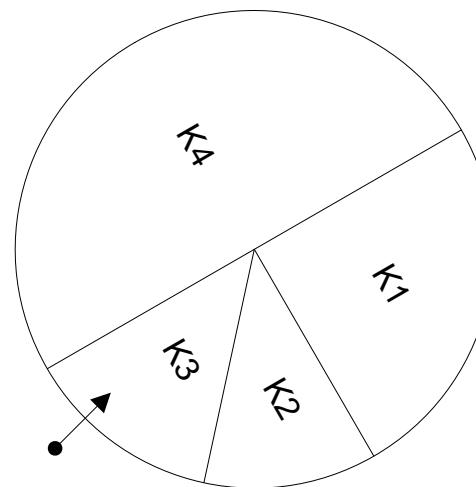


# Tournament Selection

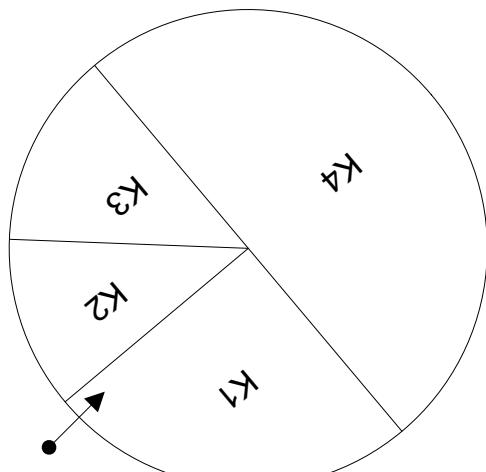
- Banyak cara yang bisa digunakan untuk memilih satu pemenang (orangtua) dari  $k$  kontestan tersebut.
- Ada empat hal yang bisa dijadikan acuan untuk membangun prosedur penentuan pemenang turnamen, yaitu:
  - Perlu perankingan atau tidak?
  - Berapa ukuran sampling  $k$ ?
  - Apakah kromosom yang sudah pernah terpilih sebagai kontestan bisa terpilih lagi?
  - Apakah kontestan terbaik (dengan *fitness* tertinggi) selalu menjadi pemenang (deterministik) atau bergantung pada suatu probabilistik tertentu?

## Metode: roulette wheel

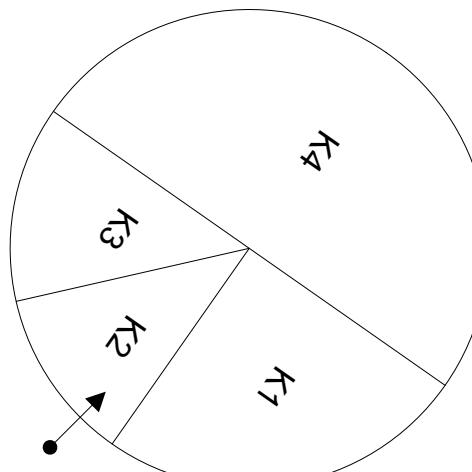
Kromosom	Fitness
K1	2
K2	1
K3	1
K4	4
Jumlah	8



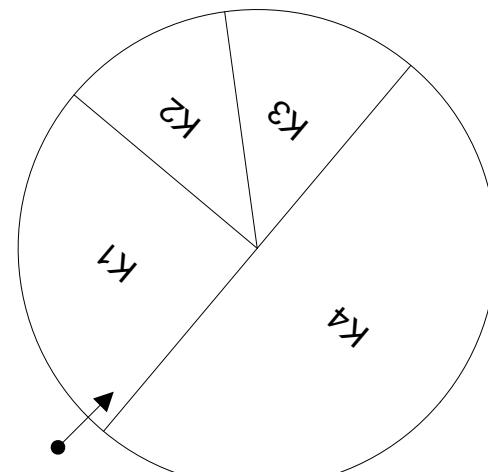
Putaran ke-1



Putaran ke-2



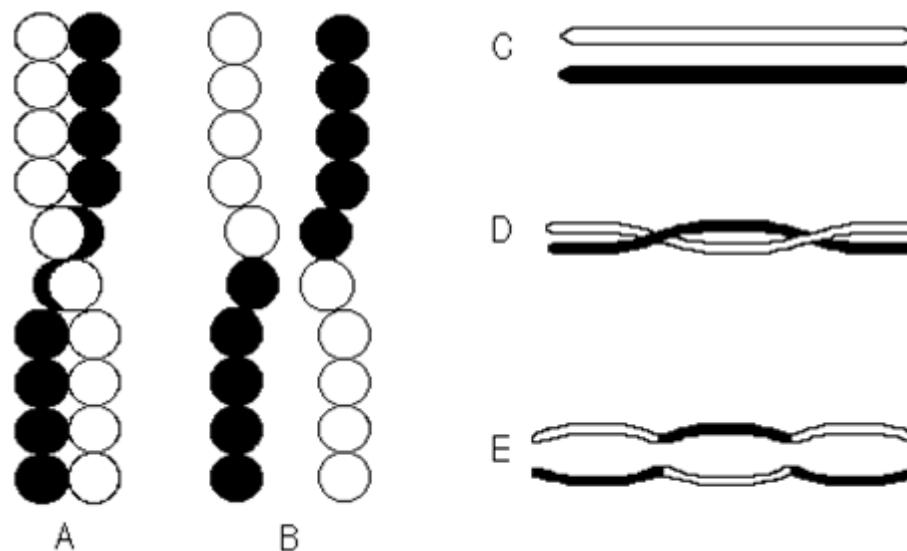
Putaran ke-3



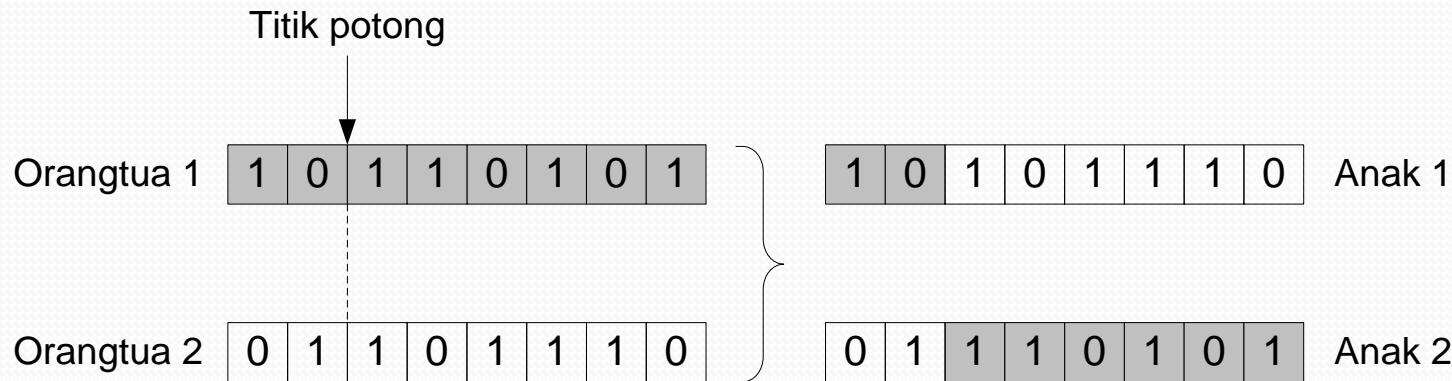
Putaran ke-4

# Rekombinasi

Operator rekombinasi di GA menirukan apa yang terjadi di dunia nyata.



# Rekombinasi



# Rekombinasi

- Pada GA berjenis *generational replacement*, setelah mendapatkan  $N$  (ukuran populasi) kromosom di *mating pool* sebagai orangtua, GA menjalankan operator rekombinasi (atau *crossover/pindah silang*) terhadap pasangan orangtua berdasarkan probabilitas tertentu.
- Banyak metode rekombinasi yang telah diusulkan.
- Masing-masing metode memiliki ciri khusus dan mungkin saja hanya bisa digunakan pada jenis representasi tertentu. Misalnya, rekombinasi untuk representasi permutasi bersifat khusus untuk masalah permutasi.

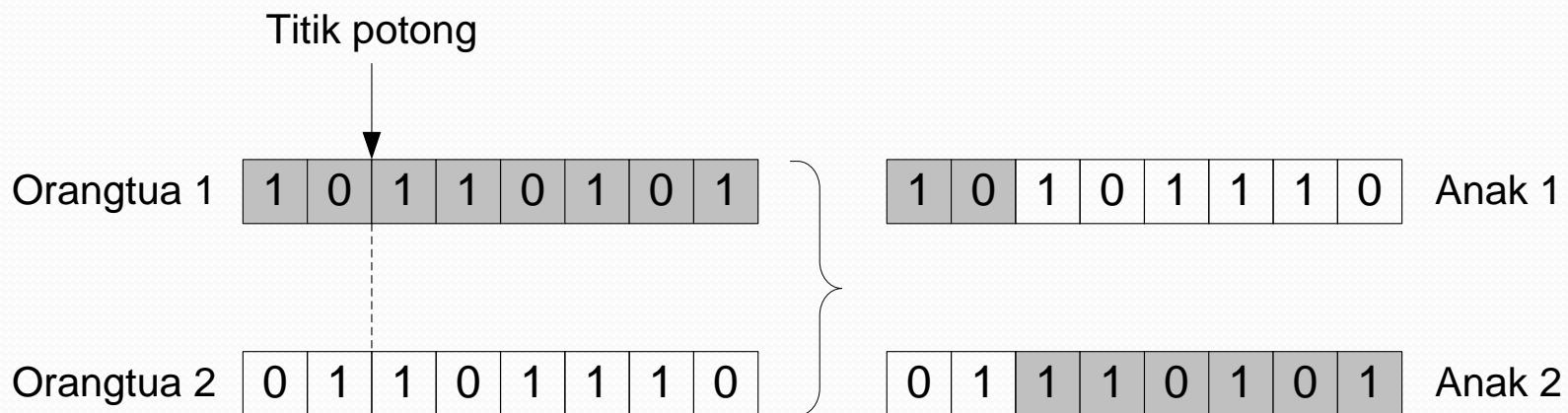
# Rekombinasi

- Rekombinasi untuk representasi Biner
- Rekombinasi untuk representasi Integer
- Rekombinasi untuk representasi Real
- Rekombinasi untuk representasi Permutasi
- Rekombinasi *Path Relinking*
- Rekombinasi *Multi-parent*

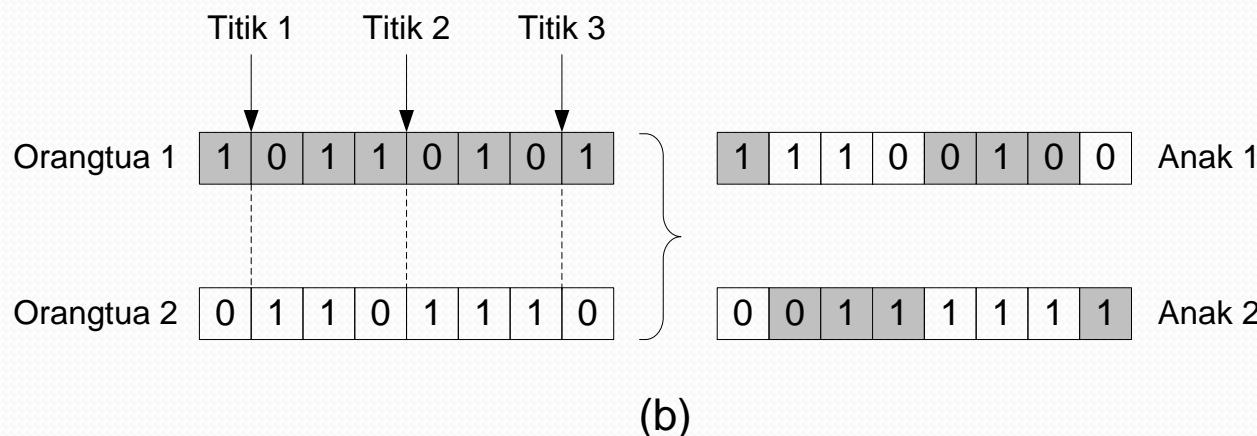
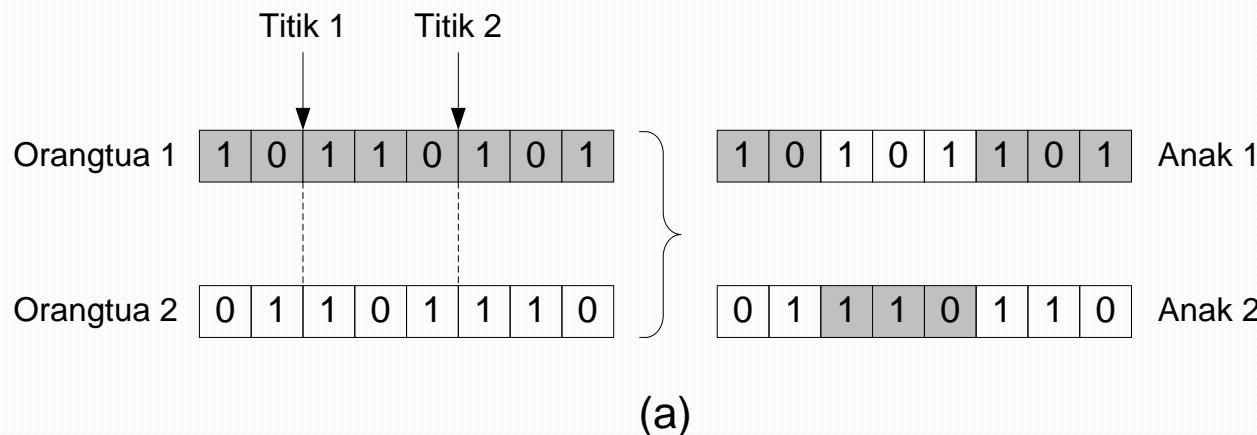
# Rekombinasi untuk Rep. Biner

- Rekombinasi satu titik (1-point crossover)
- Rekombinasi banyak titik (Multipoint crossover)
- Rekombinasi seragam (uniform crossover)

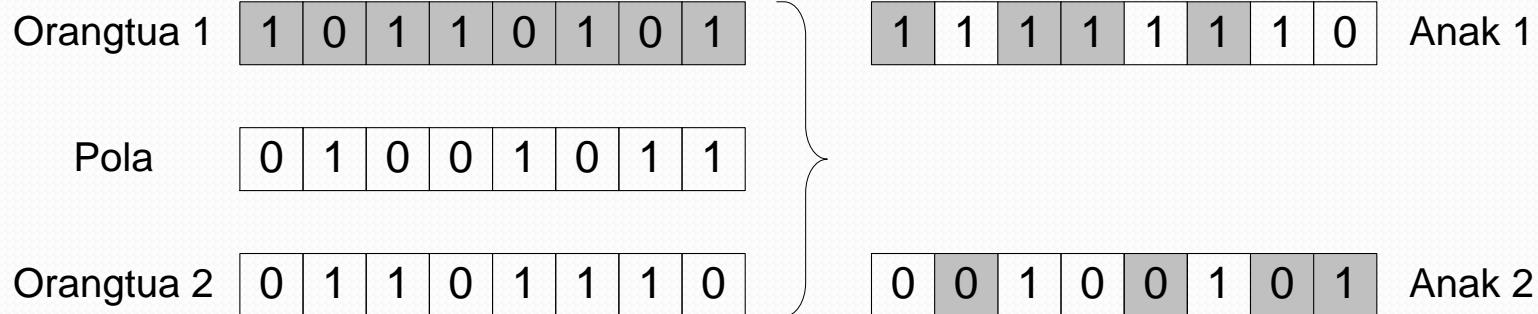
# Rekombinasi satu titik



# Rekombinasi banyak titik



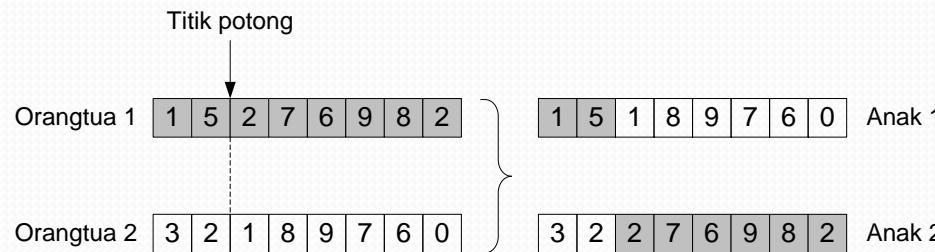
# Rekombinasi Seragam



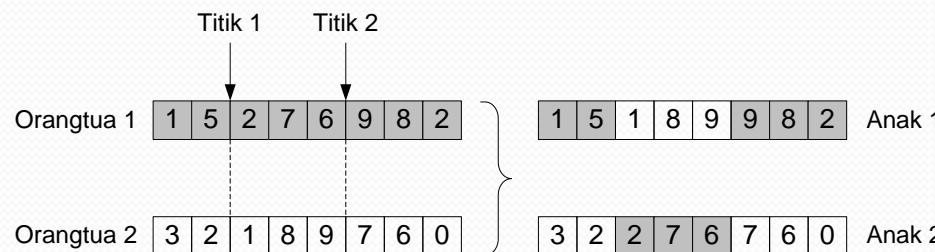
# Rekombinasi untuk Rep. Integer

- Rekombinasi satu titik (1-point crossover)
- Rekombinasi banyak titik (Multipoint crossover)
- Rekombinasi seragam (uniform crossover)

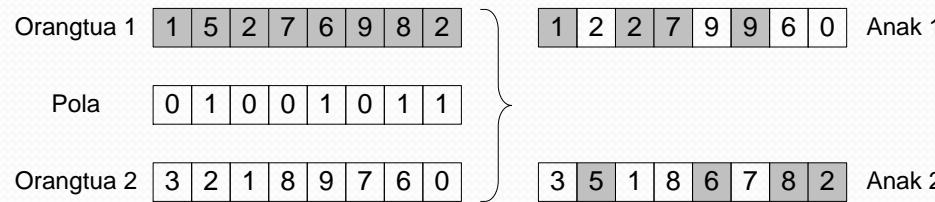
# Rekombinasi untuk Rep. Integer



(a)



(b)



(c)

# Rekombinasi untuk Rep. Real

- Kita bisa memahami bahwa tiga model rekombinasi di atas dapat digunakan untuk representasi biner dan integer.
- Hal ini disebabkan kedua representasi memiliki nilai-nilai gen dalam interval terbatas. Nilai-nilai gen pada representasi biner hanya berupa 0 atau 1.
- Pada representasi integer, mungkin juga hanya ada sedikit variasi nilai untuk setiap gen. Misalnya, representasi integer yang digunakan untuk merepresentasikan bilangan real hanya memiliki 10 kemungkinan nilai pada setiap gennya.
- Bagaimana jika setiap gen bisa memiliki nilai yang sangat bervariasi seperti representasi real? Untuk representasi real, rekombinasi bisa dilakukan dengan dua cara:
  - *discrete*
  - *intermediate*

# Rekombinasi *Discrete*

- Setiap gen pada anak z berasal dari salah satu orangtuanya ( $x, y$ ) dengan probabilitas yang sama,  $z_i = x_i \text{ or } y_i$ .
- Cara pemilihan posisi gen bisa menggunakan rekombinasi banyak titik atau rekombinasi seragam.

# Rekombinasi *Intermediate*

- Memanfaatkan ide pembangunan anak yang berupa kromosom “antara” dari kedua orangtuanya.
- Oleh karena itu, rekombinasi jenis ini disebut juga *arithmetic crossover*.
- Setiap gen pada anak z diperoleh berdasarkan rumus

$$z_i = \alpha x_i + (1 - \alpha) y_i$$

dimana  $0 \leq \alpha \leq 1$ . Parameter  $\alpha$  bisa dibuat konstan (*uniform arithmetical crossover*), variabel (misalnya, bergantung pada usia populasi), atau ditentukan secara acak pada setiap saat.

# Rekombinasi *Intermediate*

Terdapat tiga model *arithmetic crossover*, yaitu:

- *single arithmetic crossover*,
- *simple arithmetic crossover*, dan
- *whole arithmetic crossover*.

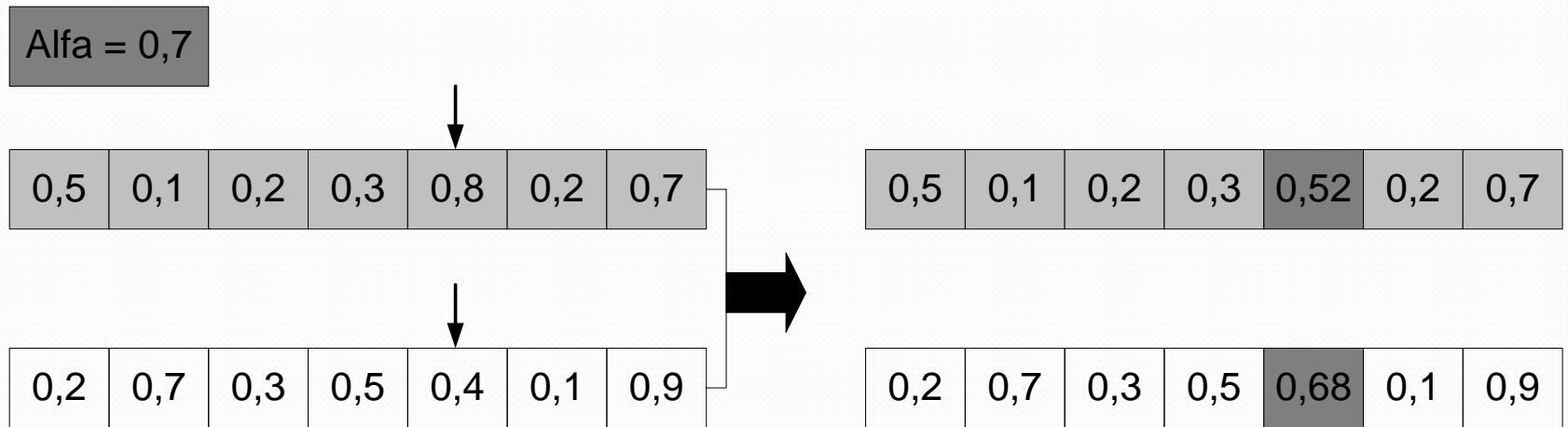
# *Single arithmetic crossover*

- Misalkan dua kromosom orangtua dinyatakan sebagai  $\langle x_1, \dots, x_n \rangle$  dan  $\langle y_1, \dots, y_n \rangle$ .
- Pilih satu gen secara acak, misal  $k$ . Selanjutnya, kedua anak dihasilkan dengan cara:

$$\text{Anak 1 : } \langle x_1, \dots, x_{k-1}, \alpha y_k + (1 - \alpha) x_k, \dots, x_n \rangle$$

$$\text{Anak 2 : } \langle y_1, \dots, y_{k-1}, \alpha x_k + (1 - \alpha) y_k, \dots, y_n \rangle$$

# *Single arithmetic crossover*



$$\text{Anak 1 : } \langle x_1, \dots, x_{k-1}, \alpha y_k + (1 - \alpha)x_k, \dots, x_n \rangle$$

$$\text{Anak 2 : } \langle y_1, \dots, y_{k-1}, \alpha x_k + (1 - \alpha)y_k, \dots, y_n \rangle$$

# *Simple arithmetic crossover*

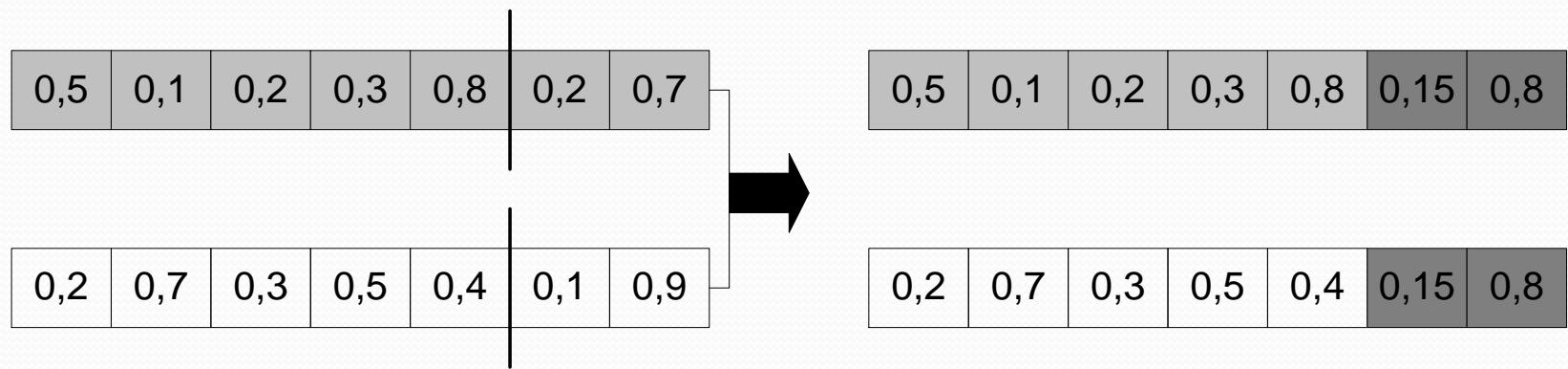
- Misalkan dua kromosom orangtua dinyatakan sebagai  $\langle x_1, \dots, x_n \rangle$  dan  $\langle y_1, \dots, y_n \rangle$ .
- Pilih satu gen secara acak, misal  $k$ . Selanjutnya, kedua anak dihasilkan dengan cara:

Anak 1 :  $\langle x_1, \dots, x_k, \alpha y_{k+1} + (1 - \alpha)x_{k+1}, \dots, \alpha y_n + (1 - \alpha)x_n \rangle$

Anak 2 :  $\langle y_1, \dots, y_k, \alpha x_{k+1} + (1 - \alpha)y_{k+1}, \dots, \alpha x_n + (1 - \alpha)y_n \rangle$

# *Simple arithmetic crossover*

Alfa = 0,5



$$\text{Anak 1 : } \langle x_1, \dots, x_k, \alpha y_{k+1} + (1 - \alpha)x_{k+1}, \dots, \alpha y_n + (1 - \alpha)x_n \rangle$$

$$\text{Anak 2 : } \langle y_1, \dots, y_k, \alpha x_{k+1} + (1 - \alpha)y_{k+1}, \dots, \alpha x_n + (1 - \alpha)y_n \rangle$$

# *Whole arithmetic crossover*

- Misalkan dua kromosom orangtua dinyatakan sebagai  $\langle x_1, \dots, x_n \rangle$  dan  $\langle y_1, \dots, y_n \rangle$ .
- Kedua anak dihasilkan dengan cara:

$$\text{Anak 1: } \alpha \cdot \bar{x} + (1 - \alpha) \cdot \bar{y}$$

$$\text{Anak 2: } \alpha \cdot \bar{y} + (1 - \alpha) \cdot \bar{x}$$

# *Whole arithmetic crossover*

Alfa = 0,5

0,5	0,1	0,2	0,3	0,8	0,2	0,7
-----	-----	-----	-----	-----	-----	-----

0,35	0,4	0,25	0,4	0,6	0,15	0,8
------	-----	------	-----	-----	------	-----

0,2	0,7	0,3	0,5	0,4	0,1	0,9
-----	-----	-----	-----	-----	-----	-----

0,35	0,4	0,25	0,4	0,6	0,15	0,8
------	-----	------	-----	-----	------	-----



$$\text{Anak 1 : } \alpha \cdot \bar{x} + (1 - \alpha) \cdot \bar{y}$$

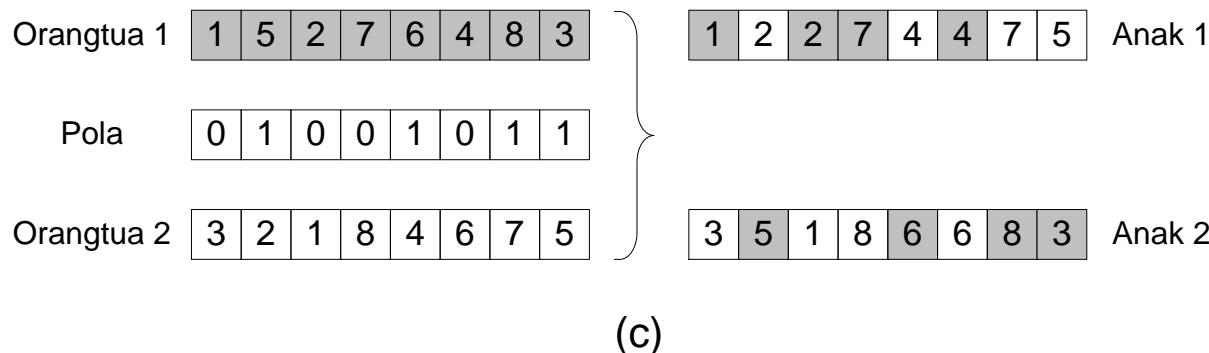
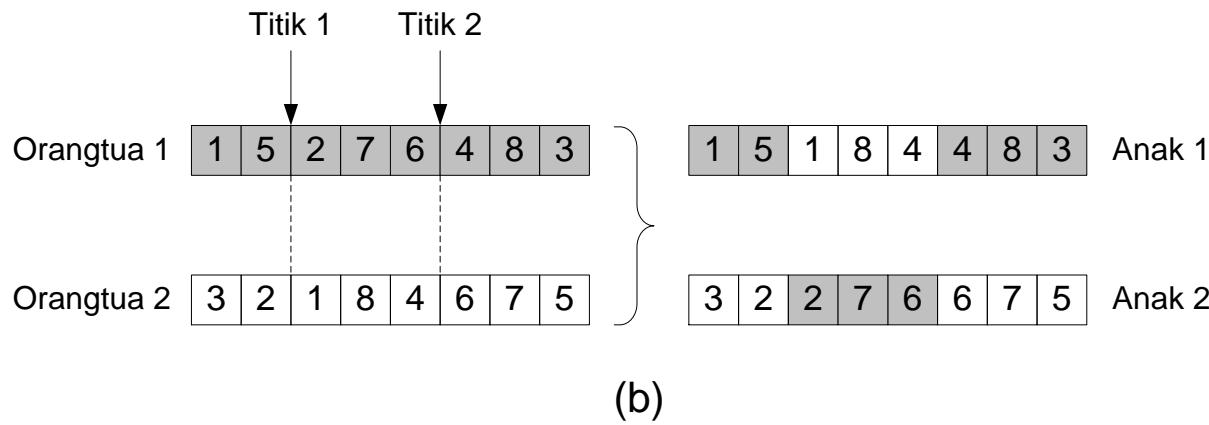
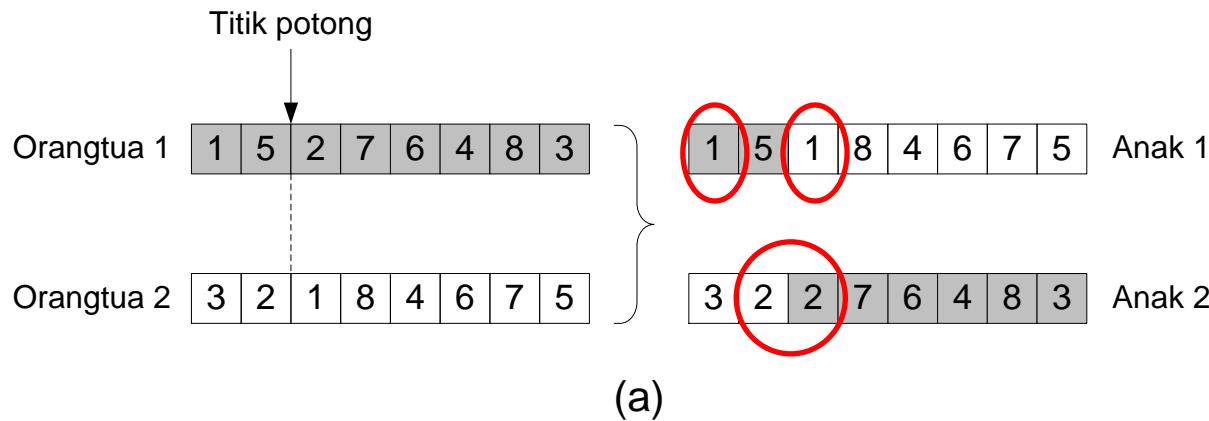
$$\text{Anak 2 : } \alpha \cdot \bar{y} + (1 - \alpha) \cdot \bar{x}$$

# Rekombinasi untuk Rep. Permutasi

- Sesuai dengan namanya, representasi permutasi digunakan untuk masalah-masalah permutasi, seperti *Travelling Salesman Problem* (TSP) misalnya.
- Pada masalah TSP, representasi permutasi memanfaatkan posisi gen sebagai urutan kota. Hal ini menyebabkan metode-metode rekombinasi untuk representasi Biner, Integer maupun Real di atas tidak bisa digunakan untuk representasi ini.
- Mengapa tidak bisa?

# Rekombinasi untuk Rep. Permutasi

- Karena ada kemungkinan anak-anak yang dihasilkan memiliki gen-gen yang **tidak valid**.
- Misalkan, ada dua gen yang bernilai sama dalam suatu kromosom yang berarti ada satu lokasi yang dikunjungi dua kali. Padahal pada masalah TSP ada batasan bahwa setiap lokasi hanya boleh dikunjungi maksimum satu kali.
- Sebaliknya, mungkin saja ada nomor lokasi yang tidak pernah muncul di dalam kromosom, yang berarti ada lokasi yang tidak pernah dikunjungi.



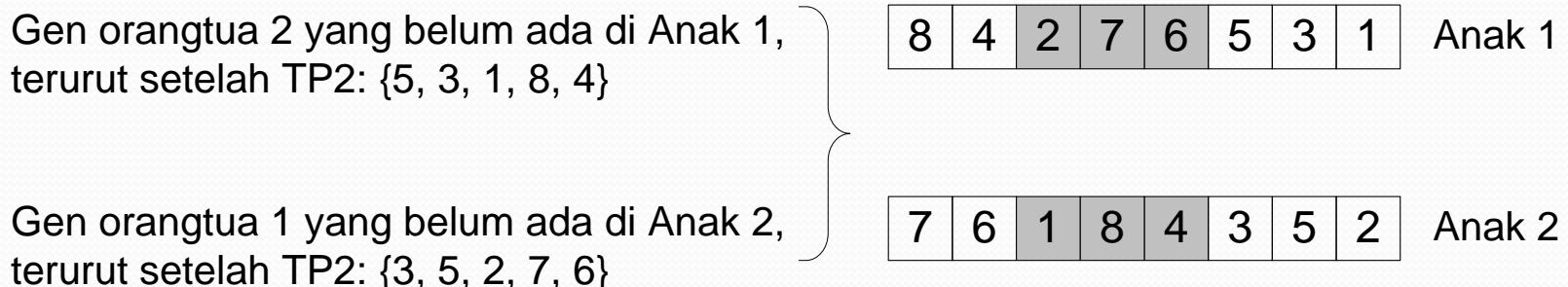
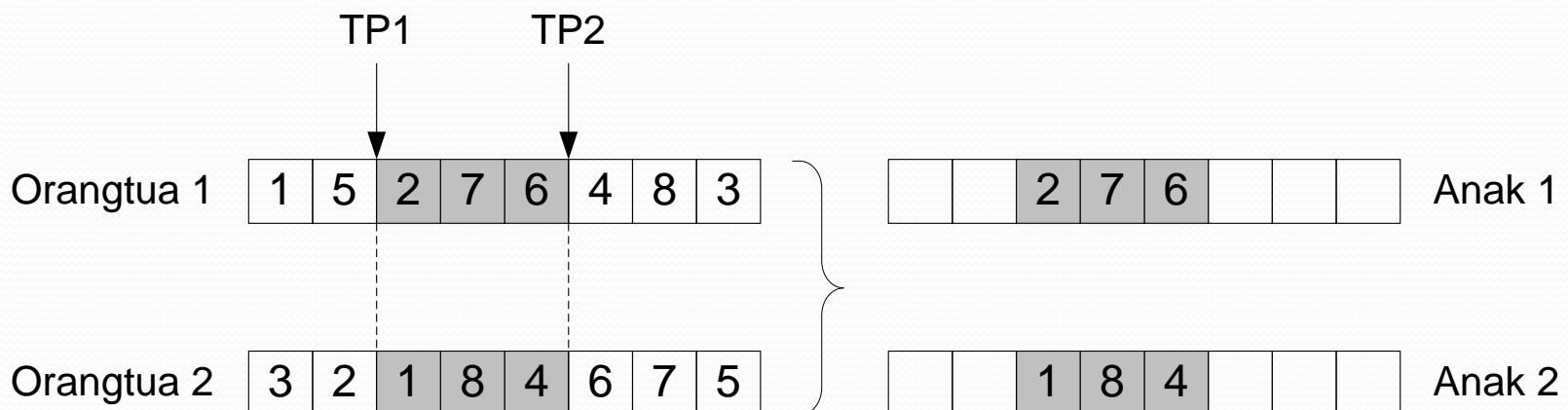
# Rekombinasi untuk Rep. Permutasi

- *Order crossover*
- *Partially mapped crossover*
- *Cycle crossover*
- *Edge Recombination*

# *Order crossover*

- Satu bagian kromosom dipertukarkan
- Tetapi urutan gen secara relatif yang bukan bagian dari kromosom tersebut tetap dijaga.

# *Order crossover*

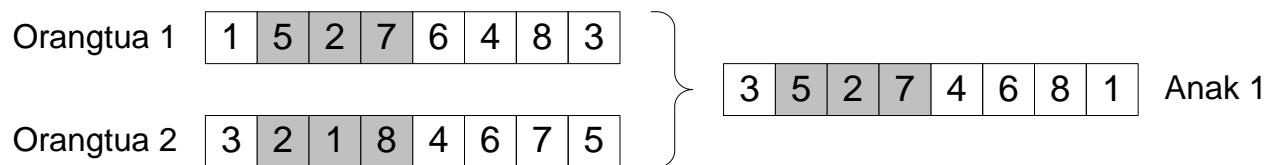
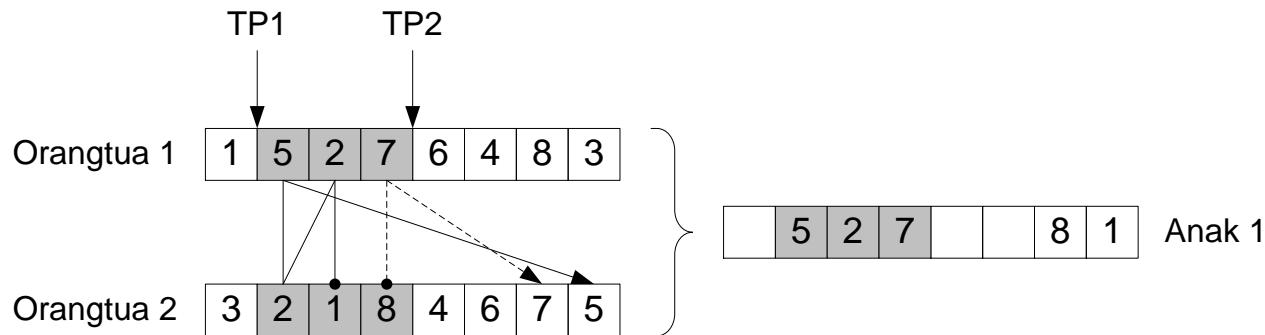


# Algoritma *Order crossover*

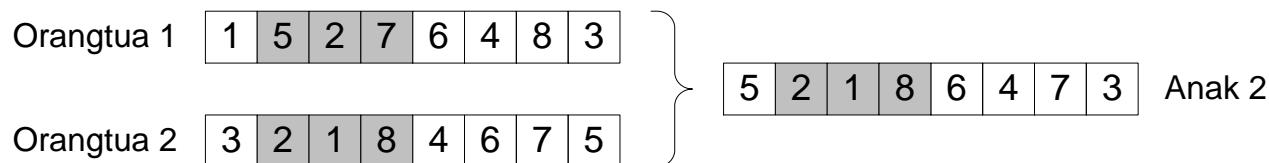
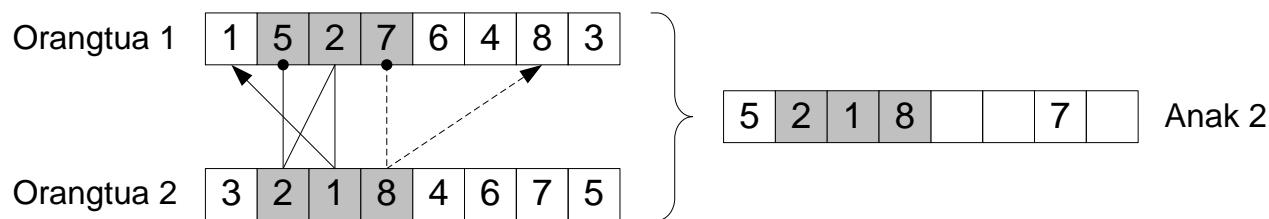
1. Pilih segmen kromosom dari kedua orangtua secara acak dengan cara membangkitkan dua titik potong, TP1 dan TP2.
2. Kopi bagian ini secara searah ke kedua anaknya. Artinya, segmen kromosom Orangtua 1 dikopi ke Anak 1 dan segmen kromosom Orangtua 2 dikopi ke Anak 2.
3. Kopi gen-gen Orangtua 2, yang tidak ada di Anak 1, ke Anak 1 dengan aturan:
  - Mulai dari posisi setelah TP2,
  - Memperhatikan urutan yang ada pada Orangtua 2
  - Kembali ke posisi awal setelah akhir kromosom (*wrapping*)
4. Lakukan langkah 3 dengan cara yang sama terhadap Orangtua 1 untuk menghasilkan Anak 2.

# *Partially mapped crossover*

- Sama dengan metode *order crossover*, metode ini juga mewariskan sebagian gen orangtua secara searah dan sebagian lainnya secara menyilang kepada kedua anaknya.
- Pewarisan sebagian gen secara menyilang dilakukan dengan memanfaatkan posisi-posisi gen kedua orangtuanya yang memiliki nilai sama untuk dilakukan pemetaan (*mapping*).
- Karena pewarisan gen yang menggunakan pemetaan dilakukan hanya pada sebagian gen, maka metode rekombinasi ini dinamakan *partially mapped crossover*.



(a)



(b)

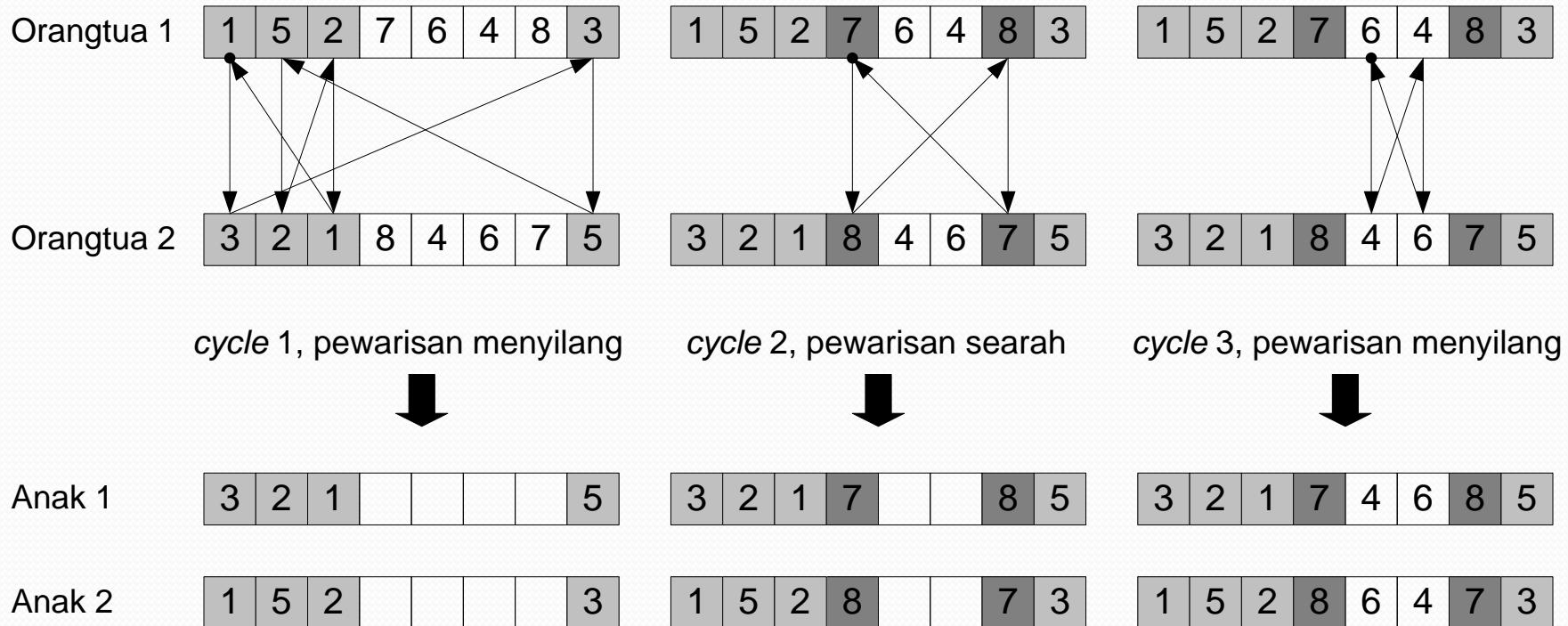
# Algoritma *Partially mapped crossover*

1. Pilih segmen kromosom dari kedua orangtua secara acak dengan cara mebangkitkan dua titik, TP1 dan TP2.
2. Kopi segmen Orangtua 1 ke Anak 1
3. Mulai dari posisi TP1 lakukan pemetaan gen-gen yang ada di segmen Orangtua 2 tetapi tidak ada di segmen Orangtua 1.
4. Wariskan setiap gen tersebut ke Anak 1 pada posisi hasil pemetaan
5. Setelah semua gen di dalam segmen sudah diwariskan ke Anak 1, maka posisi-posisi gen Anak 1 yang masih kosong diisi dengan gen-gen Orangtua 2 pada posisi-posisi yang bersesuaian.
6. Lakukan hal sama pada untuk membangkitkan Anak 2.

# *Cycle crossover*

- Sesuai dengan namanya, algoritma ini mencari siklus-siklus (*cycles*) yang terdapat pada kedua kromosom orangtua dan mewariskan *cycles* tersebut secara menyilang dan searah secara bergantian.
- Artinya, pada *cycle* pertama pewarisan dilakukan secara menyilang. Pada *cycle* ke-2 pewarisan dilakukan searah. Pada *cycle* ke-3 pewarisan kembali dilakukan secara menyilang, dan seterusnya.
- Dengan cara ini akan dihasilkan dua anak yang selalu valid.

# Cycle crossover



# Algoritma *Cycle crossover*

1. Cari cycle pada Orangtua 1 dengan cara:
  - a. Mulai dari posisi pertama Orangtua 1 yang belum diwariskan.
  - b. Buat panah ke posisi yang sama pada Orangtua 2.
  - c. Buat panah ke posisi gen yang bernilai sama pada Orangtua 1.
  - d. Tambahkan gen ini ke cycle.
  - e. Ulangi langkah b sampai d sampai panah kembali ke posisi awal.
2. Wariskan gen-gen orangtua yang berada pada cycle ini kepada kedua anaknya sesuai dengan posisinya dengan cara menyilang, searah, menyilang, searah, dan seterusnya.

# *Edge Recombination*

- Algoritma ini bekerja dengan cara membangun suatu yang mendaftar *edges* (sisi-sisi) gen yang berada pada kedua orangtuanya.
- Untuk mendapatkan *edge* dari suatu gen , kromosom dianggap melingkar. Artinya, gen pada posisi 1 memiliki *edge* yang berupa gen posisi 2 dan gen posisi terakhir.
- Gen pada posisi terakhir memiliki *edge* yang berupa gen posisi sebelumnya dan gen posisi 1.
- Jika *edge* berada pada kedua orangtua, maka *edge* tersebut diberi tanda positif ‘+’ dan disebut sebagai *common edge*.
- Pewarisan gen-gen orangtua dilakukan berdasarkan tabel tersebut.

Orangtua 1

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

Orangtua 2

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

Pencarian edge untuk semua gen pada kedua Orangtua menghasilkan tabel sbb:

Elemen	Edge
1	3, 5, 2, 8
2	5, 7, 3, 1
3	8, 1, 5, 2
4	6+, 8+
5	1, 2, 7, 3
6	4+, 7+
7	2, 6+, 5
8	4+, 3, 1

Tabel harus di-update setelah suatu elemen terpilih. Setelah 3 terpilih, semua edge 3 dihapus dari tabel.

Elemen	Edge
1	5, 2, 8
2	5, 7, 1
3	8, 1, 5, 2
4	6+, 8+
5	1, 2, 7
6	4+, 7+
7	2, 6+, 5
8	4+, 1

...

Pilihan	Elemen terpilih	Alasan	Hasil
Semua	3	Pemilihan secara acak	{3}
8, 1, 5, 2	8	Daftar edge terpendek	{3,8}
4, 1	4	Common edge	{3,8,4}
6	6	Common edge	{3,8,4,6}
7	7	Common edge	{3,8,4,6,7}
2, 5	2	Pemilihan acak	{3,8,4,6,7,2}
5, 1	1	Pemilihan acak	{3,8,4,6,7,2,1}
5	5	Hanya ada satu pilihan	{3,8,4,6,7,2,1,5}

Anak1.  
Anak2  
dihasilkan  
dengan cara  
yang sama.  
Karena acak,  
maka Anak2  
bisa berbeda  
dengan Anak1.

# *Edge Recombination*

- Pada gambar di atas, suatu tabel elemen dan *edge* dibangun berdasarkan susunan gen-gen yang berada pada kedua orangtuanya. Setelah tabel dibangun, suatu elemen awal dipilih secara acak dari semua elemen yang ada (1 sampai 8).
- Misalkan elemen 3 terpilih dan dimasukkan ke hasil. Setelah elemen 3 terpilih, maka tabel harus di-update dengan cara menghapus semua elemen 3 dari daftar *edge* (lihat tabel sebelah kanan).
- Selanjutnya, pilih elemen berikutnya dari daftar *edge* yang ada pada elemen 3, yaitu {8, 1, 5, 2}. Pemilihan elemen dilakukan berdasarkan aturan berprioritas sebagai berikut:
  - Jika ada elemen yang merupakan *common edge*, maka pilih elemen tersebut.
  - Jika tidak ada *common edge*, pilih elemen yang yang memiliki daftar *edge* terpendek (jumlah *edge* paling sedikit).
  - Jika semua elemen memiliki jumlah *edge* yang sama, maka pilih elemen secara acak.

# Algoritma *Edge Recombination*

1. Pilih elemen awal secara acak dari semua elemen yang ada. Kemudian masukkan elemen terpilih ke Hasil.
2. Set elemen terpilih sebagai *current element*
3. Hapus *current element* dari semua daftar edge yang ada di tabel
4. Pilih satu elemen dari daftar edge yang ada di *current element* dengan aturan:
  - a. Jika ada elemen yang merupakan common edge, maka pilih elemen tersebut.
  - b. Jika tidak ada common edge, pilih elemen yang yang memiliki daftar edge terpendek (jumlah edge paling sedikit).
  - c. Jika semua elemen memiliki jumlah edge yang sama, maka pilih elemen secara acak.
5. Jika daftar edge yang ada di *current element* sudah kosong, maka
  - i. Ubah pemilihan elemen terakhir untuk mencari elemen lain yang memiliki daftar edge tidak kosong.
  - ii. Jika perubahan elemen tidak berhasil, pilih elemen baru secara acak.

# Rekombinasi Path Relinking

- Rekombinasi dilakukan dengan membuat banyak anak yang memiliki perbedaan secara berurutan sehingga mirip suatu jalur. Kemudian pilih sejumlah anak yang memiliki *fitness* tertinggi.
- Bagaimana cara membuat anak-anak yang memiliki perbedaan secara berurutan?
- Mudah saja. Buat Anak 1 yang sebagian besar gen-nya (misal 90%) diambil dari Orangtua 1 dan sebagian kecil gen lainnya (10%) diambil dari Orangtua 2. Buat Anak 2 dengan porsi gen yang sedikit berbeda, misalnya 80% dari Orangtua 1 dan 20% dari Orangtua 2.
- Dengan demikian, bisa dikatakan Anak 1 mirip dengan Orangtua 1. Anak 2 mirip dengan Anak 1 dan seterusnya sehingga Anak *n* mirip dengan Orangtua 2.

# Rekombinasi Path Relinking

Orangtua 1

1	0	0	1	0	0	1
---	---	---	---	---	---	---

Orangtua 2

0	0	1	0	1	0	0
---	---	---	---	---	---	---



Anak 1

1	0	0	1	0	0	0
---	---	---	---	---	---	---

Anak 2

1	0	0	1	1	0	0
---	---	---	---	---	---	---

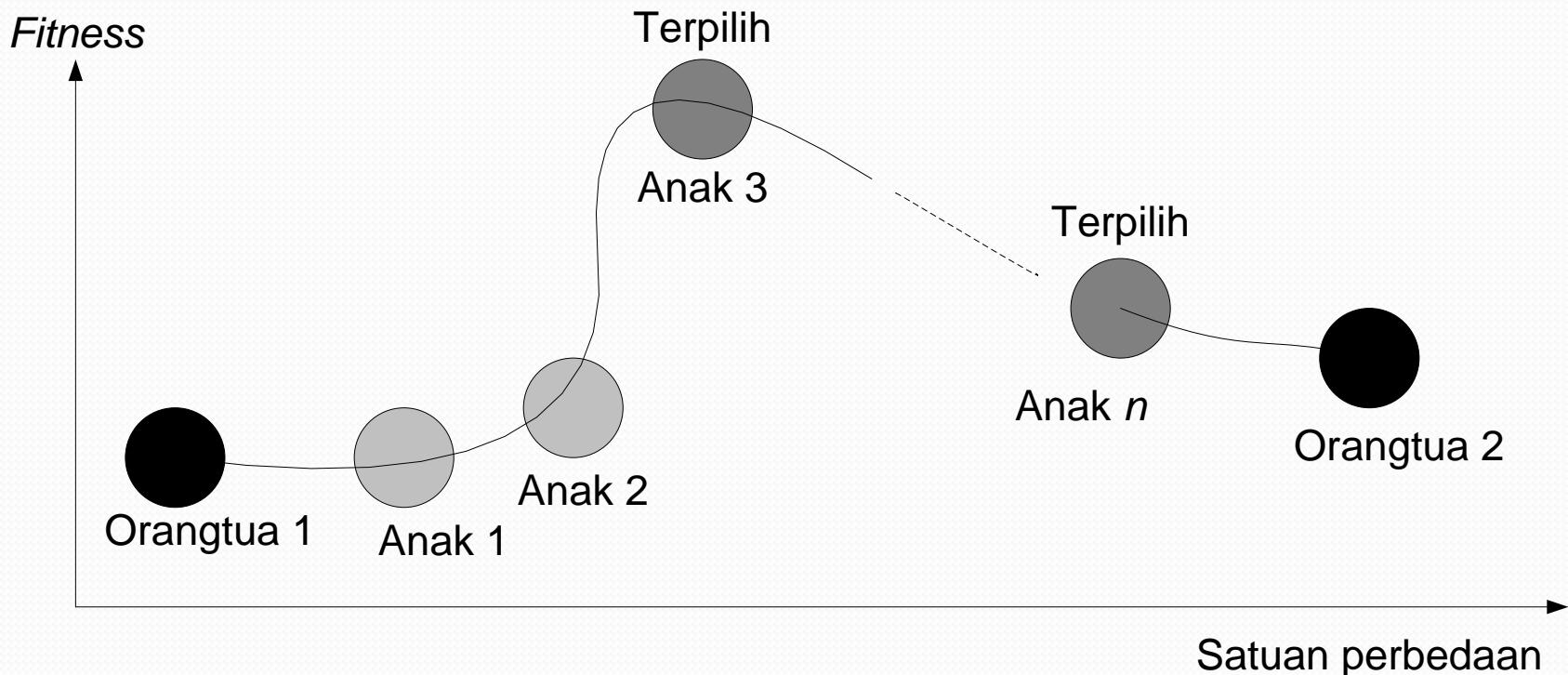
Anak 3

1	0	0	0	1	0	0
---	---	---	---	---	---	---

Anak 4

1	0	1	0	1	0	0
---	---	---	---	---	---	---

# Rekombinasi Path Relinking



# Rekombinasi Multi-parent

- Bagaimanapun, untuk membangun GA kita mungkin tidak perlu terlalu kaku dalam mengadopsi apa yang ada di dunia nyata.
- Mungkin saja penggunaan lebih dari dua kromosom orangtua dalam proses rekombinasi akan memberikan hasil yang lebih baik.
- Hal ini yang disebut sebagai rekombinasi banyak orangtua (*Multi-parent*).

# Rekombinasi Multi-parent

- Bagaimana penerapan rekombinasi *Multi-parent* ini?
- Terdapat tiga pendekatan yang bisa digunakan, yaitu:
  - Berdasarkan frekuensi allele
  - Berdasarkan segmentasi dan rekombinasi
  - Berdasarkan operasi-operasi numerik pada allele bernilai real

# Berdasarkan frekuensi allele

- Pendekatan ini merupakan generalisasi dari rekombinasi seragam (*uniform crossover*).
- Jika pada *uniform crossover* dilakukan pembangkitan pola dengan dua kemungkinan nilai (berdasarkan pelemparan koin), maka pada pendekatan ini pembangkitan pola dilakukan dengan kemungkinan nilai sebanyak jumlah orangtuanya.
- Jika orangtuanya berjumlah tiga, maka pembangkitan pola dilakukan untuk menghasilkan tiga nilai berbeda.

# Berdasarkan frekuensi allele

Orangtua 1 

0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

Orangtua 2 

1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

Orangtua 3 

0	1	1	0	1	1	1	0
---	---	---	---	---	---	---	---

Pola 1 

2	1	3	3	2	1	1	3
---	---	---	---	---	---	---	---

Pola 2 

1	1	3	2	2	1	2	3
---	---	---	---	---	---	---	---

Anak 1 

1	0	1	0	1	1	0	0
---	---	---	---	---	---	---	---

Anak 2 

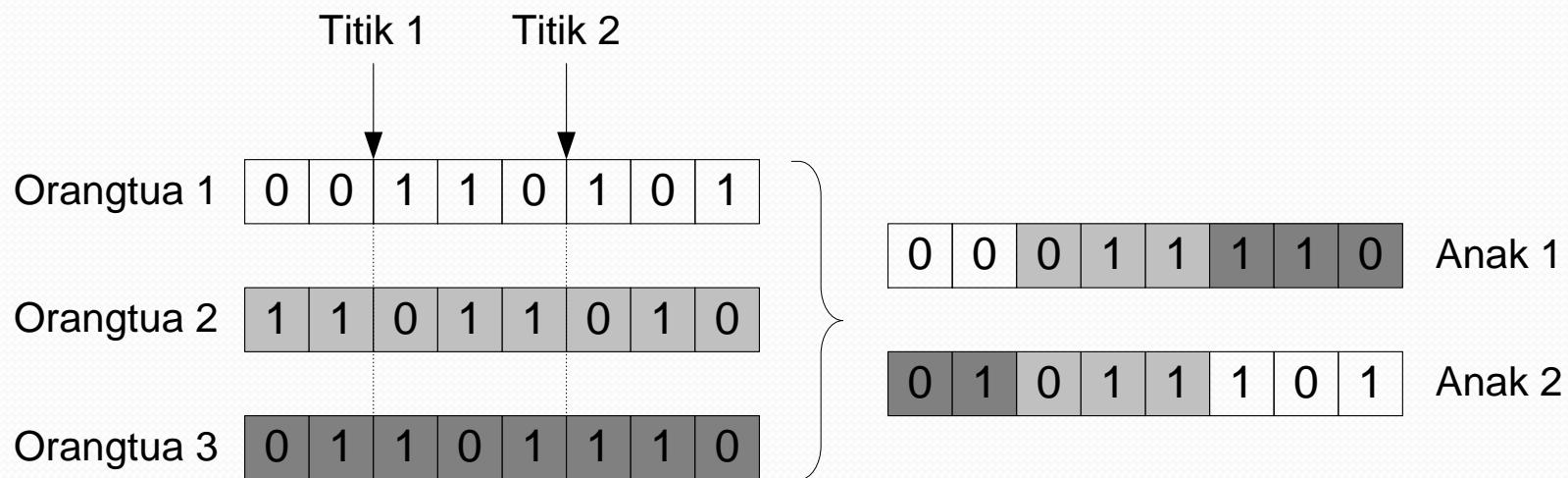
0	0	1	1	1	1	1	0
---	---	---	---	---	---	---	---

Anak 1 dihasilkan berdasarkan Pola 1  
dan Anak 2 berdasarkan Pola 2.

## Berdasarkan segmentasi dan rekombinasi

- Pendekatan ini merupakan generalisasi dari rekombinasi banyak titik (*n-point crossover*).
- Pewarisan gen dilakukan secara diagonal.

# Berdasarkan segmentasi dan rekombinasi



# Berdasarkan operasi-operasi numerik (pada allele bernilai real)

- Pendekatan ini merupakan generalisasi dari rekombinasi *Arithmetic Recombination*.
- Salah satu contohnya adalah rekombinasi pusat massa (*the center of mass crossover*).

# Rekombinasi Biner/Integer

```
function Anak = RekombinasiBiner(Ortu1,Ortu2,JumGen)
```

```
TP = 1 + fix(rand*(JumGen-1));
```

```
Anak(1,:) = [Ortu1(1:TP) Ortu2(TP+1:JumGen)];
```

```
Anak(2,:) = [Ortu2(1:TP) Ortu1(TP+1:JumGen)];
```

# Rekombinasi Real

```
function Anak = RekombinasiReal(Ortu1, Ortu2, JumGen)
```

```
Alfa = 0.8; % Alfa bisa diset sesuai kebutuhan
```

```
% Whole arithmetic crossover
```

```
Anak(1,:) = (Alfa .* Ortu1) + (1 - Alfa) .* Ortu2;  
Anak(2,:) = (Alfa .* Ortu2) + (1 - Alfa) .* Ortu1;
```

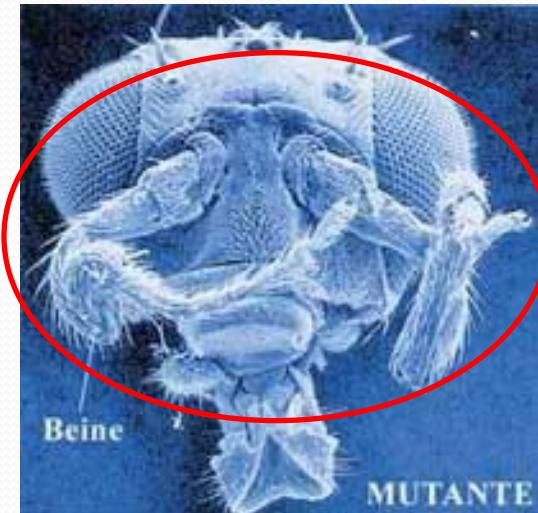
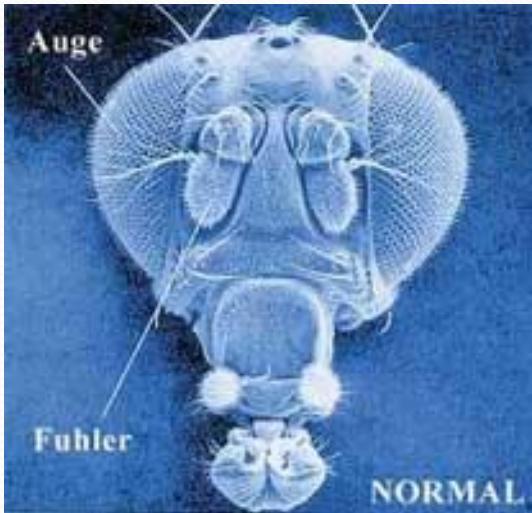
# Mutasi

- Setelah tahap rekombinasi terhadap semua pasangan kromosom pada *mating pool* yang menghasilkan  $N$  (ukuran populasi) kromosom, maka GA menjalankan operator mutasi terhadap setiap kromosom tersebut.
- Banyak metode mutasi yang telah diusulkan.
- Masing-masing metode memiliki ciri khusus dan mungkin saja hanya bisa digunakan pada jenis representasi tertentu.
- Misalnya, mutasi untuk representasi permutasi memerlukan metode khusus yang menjamin kromosom hasil mutasi tetap valid.

# Mutasi

- Mutasi bersifat kecil, acak, berbahaya, dan jarang terjadi. Jika terjadi, kemungkinan besar mutasi itu tidak berguna.
- Empat karakteristik mutasi ini menunjukkan bahwa mutasi tidak dapat mengarah pada perkembangan evolusioner.
- Suatu perubahan acak pada organisme yang sangat terspesialisasi bersifat tidak berguna atau membahayakan.
- Perubahan acak pada sebuah jam tidak dapat memperbaiki, malah kemungkinan besar akan merusaknya atau tidak berpengaruh sama sekali.
- Gempa bumi tidak akan memperbaiki kota, tetapi menghancurnya [RAN88].

# Mutasi: bisa lebih baik?



- Struktur DNA **amat sangat rumit !**
- Perubahan acak (mutasi) **selalu buruk !!**

Kiri: seekor lalat buah (*drosophila*) normal. Tengah: seekor lalat buah dengan kaki tumbuh di kepala (mutasi akibat radiasi). Kanan: Bocah laki-laki korban kecelakaan instalasi nuklir Chernobyl yang mengakibatkan mutasi gen [ADNo7].

# Mutasi pada EAs

- Mutasi untuk representasi Biner
- Mutasi untuk representasi Integer
- Mutasi untuk representasi Real
- Mutasi untuk representasi Permutasi

## Mutasi: bisa lebih baik?

- Bisa. Mengapa?
- Representasi individu pada EAs jauh lebih sederhana
- Mutasi sebagian kecil gen mungkin menghasilkan individu yang lebih baik

# Mutasi untuk Representasi Biner

Kromosom awal

0	1	1	0	1	1	1	0
---	---	---	---	---	---	---	---



$$a \leq P_m$$

Kromosom hasil mutasi

0	0	1	0	1	1	1	0
---	---	---	---	---	---	---	---

Membalik bit: 1 → 0 dan sebaliknya.

# Mutasi untuk representasi Integer

- Membalik nilai integer
- Pemilihan nilai secara acak
- Mutasi *Creep* (Perlahan)

# Mutasi untuk Representasi Integer

Membalik nilai integer

Kromosom awal

3	6	1	0	5	1	2	9
---	---	---	---	---	---	---	---



$$a \leq P_m$$

Kromosom hasil mutasi

3	6	8	0	5	1	2	9
---	---	---	---	---	---	---	---

# Mutasi untuk Representasi Integer

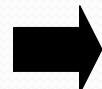
Pemilihan nilai secara acak

Kromosom awal

3	6	1	0	5	1	2	9
---	---	---	---	---	---	---	---



$$a \leq P_m$$



Kromosom hasil mutasi

3	6	5	0	5	1	2	9
---	---	---	---	---	---	---	---

# Mutasi untuk Representasi Integer

## Mutasi *Creep* (Perlahan)

Kromosom awal

3	6	1	0	5	1	2	9
---	---	---	---	---	---	---	---



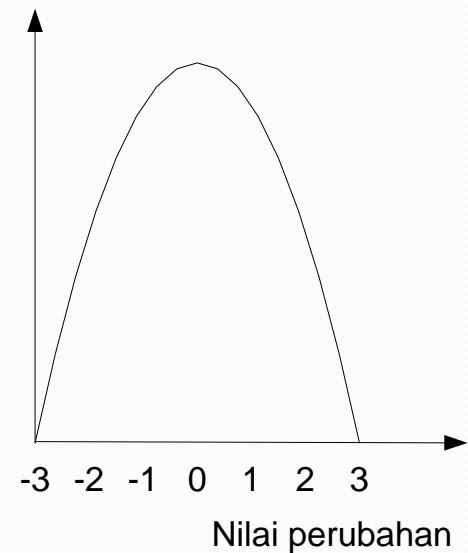
$$a \leq P_m$$

Kromosom hasil mutasi

3	6	2	0	5	1	2	9
---	---	---	---	---	---	---	---



Probabilitas



# Mutasi untuk Representasi Real

- Representasi real memiliki karakteristik yang berbeda dengan biner ataupun integer.
- Nilai-nilai gen pada representasi real bersifat kontinyu sedangkan representasi biner dan integer bersifat diskrit.
- Oleh karena itu, representasi real memerlukan mutasi khusus yang berbeda dengan sebelumnya.

# Mutasi untuk Representasi Real

- Mutasi *Uniform*

Nilai-nilai  $x_i$  didapat dari pembangkitan bilangan secara acak dengan distribusi seragam (*uniform distribution*)

- Mutasi *Non-uniform* dengan distribusi tetap

Mutasi jenis ini paling umum digunakan. Caranya mirip dengan metode *Creep* pada representasi integer. Bedanya di sini digunakan penambahan nilai real (bukan integer).

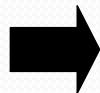
# Mutasi untuk Rep. Permutasi

- Mutasi pada representasi permutasi harus menghasilkan kromosom yang valid.
- Sehingga, proses mutasi dilakukan dengan suatu cara tertentu yang menjamin kromosom hasil mutasi tetap valid.
- Banyak cara yang bisa digunakan, diantaranya adalah:
  - Mutasi pertukaran (*Swap Mutation*)
  - Mutasi penyisipan (*Insert Mutation*)
  - Mutasi pengacakan (*Scramble Mutation*)
  - Mutasi pembalikan (*Inversion Mutation*)

# Mutasi pertukaran (Swap Mutation)

Kromosom awal

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



Kromosom hasil mutasi

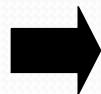
1	8	2	7	6	4	5	3
---	---	---	---	---	---	---	---

Mutasi pertukaran (*Swap Mutation*). Pemilihan dua posisi gen secara acak menghasilkan posisi 2 dan 7. Sehingga gen bernilai 5 dipertukarkan dengan gen bernilai 8.

# Mutasi penyisipan (Insert Mutation)

Kromosom awal

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



Kromosom hasil mutasi

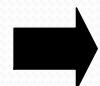
1	5	8	2	7	6	4	3
---	---	---	---	---	---	---	---

Mutasi penyisipan (*Insert Mutation*). Pemilihan dua posisi gen secara acak menghasilkan posisi 2 dan 7. Kemudian gen bernilai 8 disisipkan setelah gen bernilai 5.

# Mutasi pengacakan (*Scramble Mutation*)

Kromosom awal

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

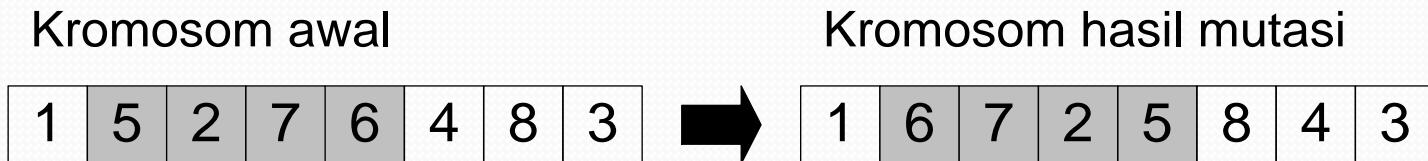


Kromosom hasil mutasi

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

Mutasi pengacakan (*Scramble Mutation*). Pemilihan segmen kromosom menghasilkan {5, 2, 7, 6}. Kemudian pengacakan gen dalam segmen menghasilkan {2, 6, 5, 7}

# Mutasi pembalikan (Inversion Mutation)



Mutasi pembalikan (*Inversion Mutation*). Pemilihan segmen kromosom menghasilkan {5, 2, 7, 6}. Kemudian pembalikan posisi gen dalam segmen tersebut menghasilkan {6, 7, 2, 5}.

# Mutasi Biner

```
function MutKrom = MutasiBiner(Kromosom, JumGen, Pmutasi)

MutKrom = Kromosom;
for ii=1:JumGen,
    if (rand < Pmutasi),
        if Kromosom(ii)==0,
            MutKrom(ii) = 1;
        else
            MutKrom(ii) = 0;
        end
    end
end
```

# Mutasi Integer

```
function MutKrom = MutasiInteger(Kromosom, JumGen, Pmutasi)
```

```
MutKrom = Kromosom;  
for ii=1:JumGen,  
    if (rand < Pmutasi),  
        a = fix(randn * 2);  
        MutKrom(ii) = abs(a + Kromosom(ii));  
        if MutKrom(ii) > 9,  
            MutKrom(ii) = 9;  
    end  
end  
end
```

# Mutasi Real

```
function MutKrom = MutasiReal(Kromosom, JumGen, Pmutasi)
CreepSize = 0.001;
MutKrom = Kromosom;
for ii=1:JumGen,
    if (rand < Pmutasi),
        a = (randn * CreepSize);
        MutKrom(ii) = a + Kromosom(ii);
        if MutKrom(ii) > 1,
            MutKrom(ii) = 1;
        elseif MutKrom(ii) < 0,
            MutKrom(ii) = 0;
    end
end
end
```

# Permutasi: Swap mutation

```
function MutKrom = TSPMutasi (Kromosom, JumGen, Pmutasi)
```

```
MutKrom = Kromosom;
```

```
for ii=1:JumGen,
```

```
    if rand < Pmutasi,
```

```
        TM2 = 1 + fix(rand*JumGen);
```

```
        while TM2==ii,
```

```
            TM2 = 1 + fix(rand*JumGen);
```

```
        end
```

```
        temp = MutKrom(ii);
```

```
        MutKrom(ii) = MutKrom(TM2);
```

```
        MutKrom(TM2) = temp;
```

```
    end
```

```
end
```

Kromosom awal

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



Kromosom hasil mutasi

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

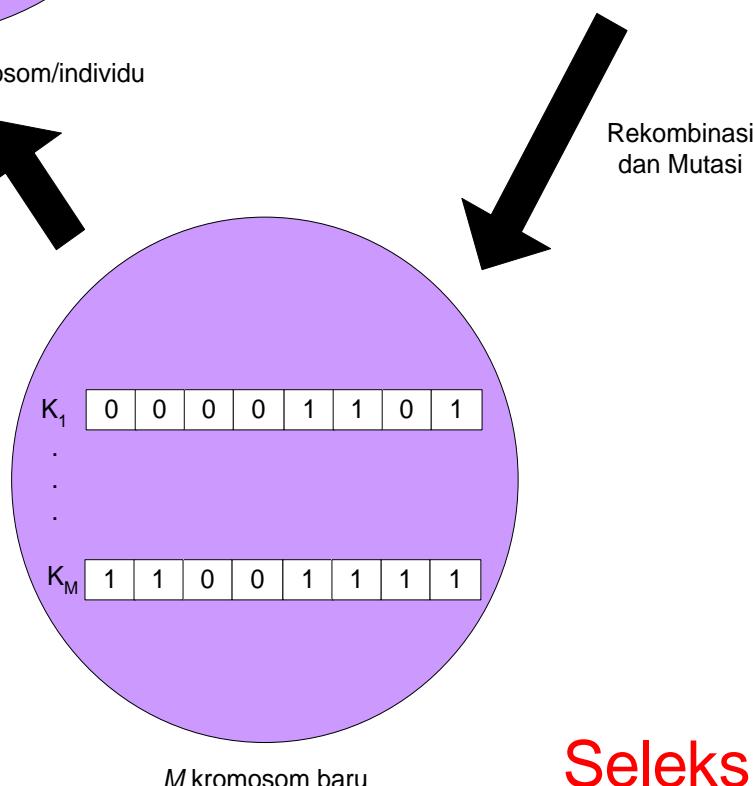
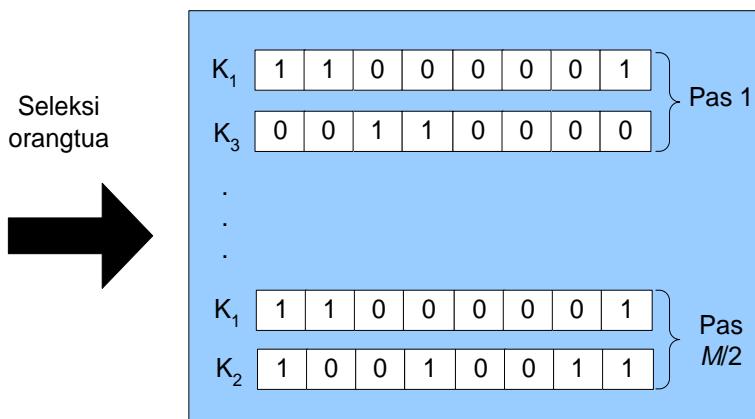
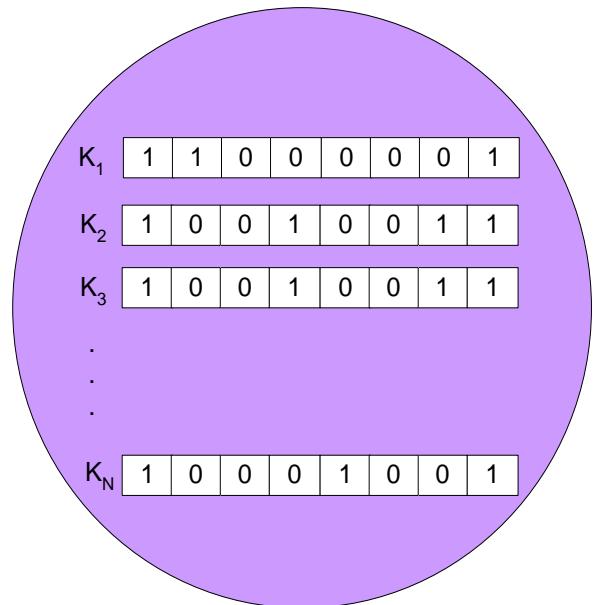
# Seleksi Survivor

- Generational Model

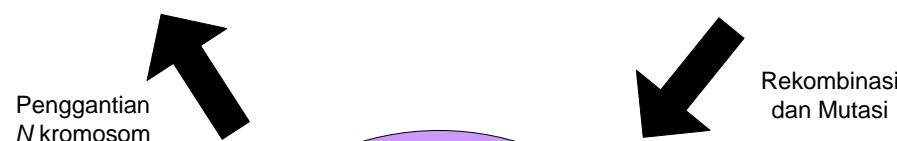
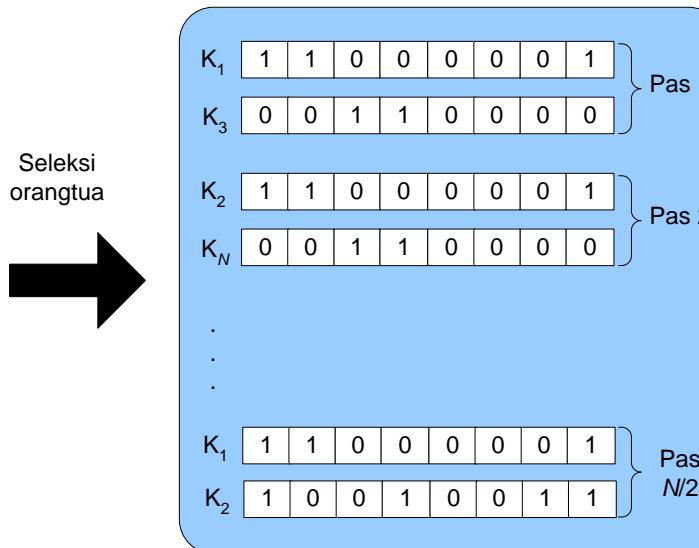
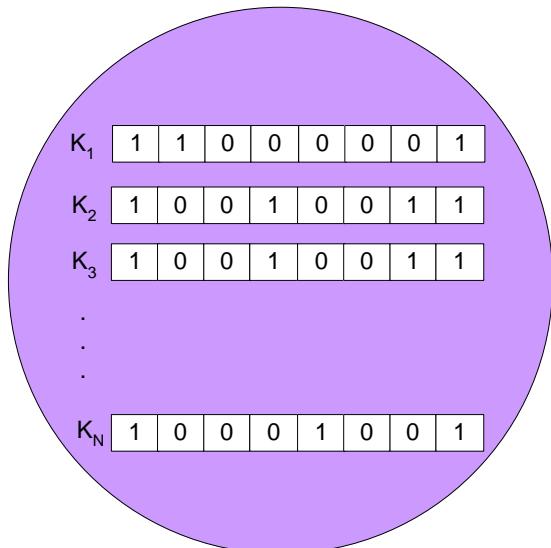
Suatu populasi berukuran  $N$  kromosom/individu pada suatu generasi diganti dengan  $N$  individu baru pada generasi berikutnya. Untuk menjaga kromosom terbaik, digunakan Elitisme.

- Steady State Model

Pada model ini, tidak semua kromosom diganti. Penggantian dilakukan hanya pada sejumlah kromosom tertentu, misal  $M$ . Dimana  $M < N$ .

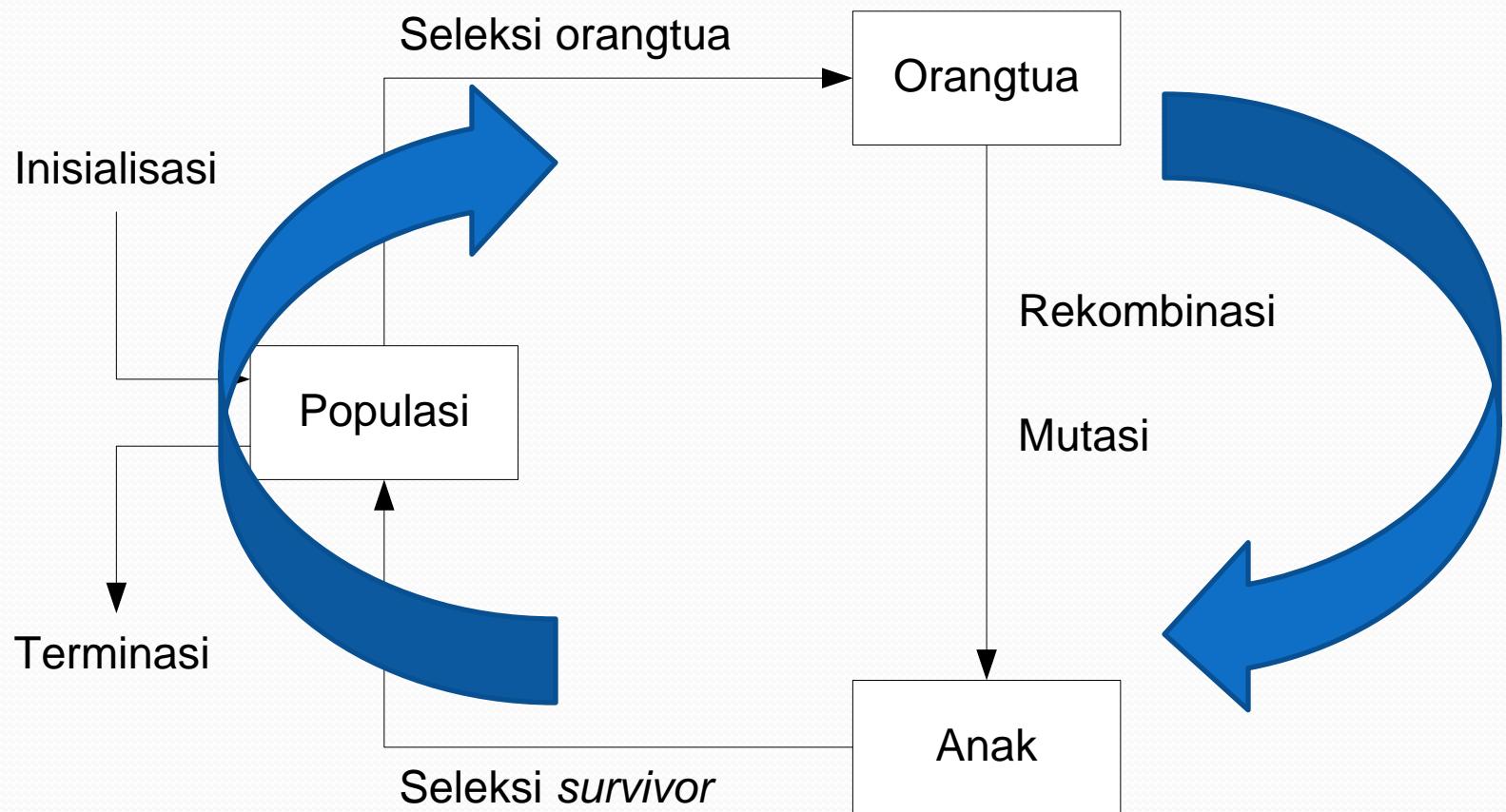


Seleksi Survivor: Steady State



Seleksi Survivor: *Generational*

# Skema umum EAs



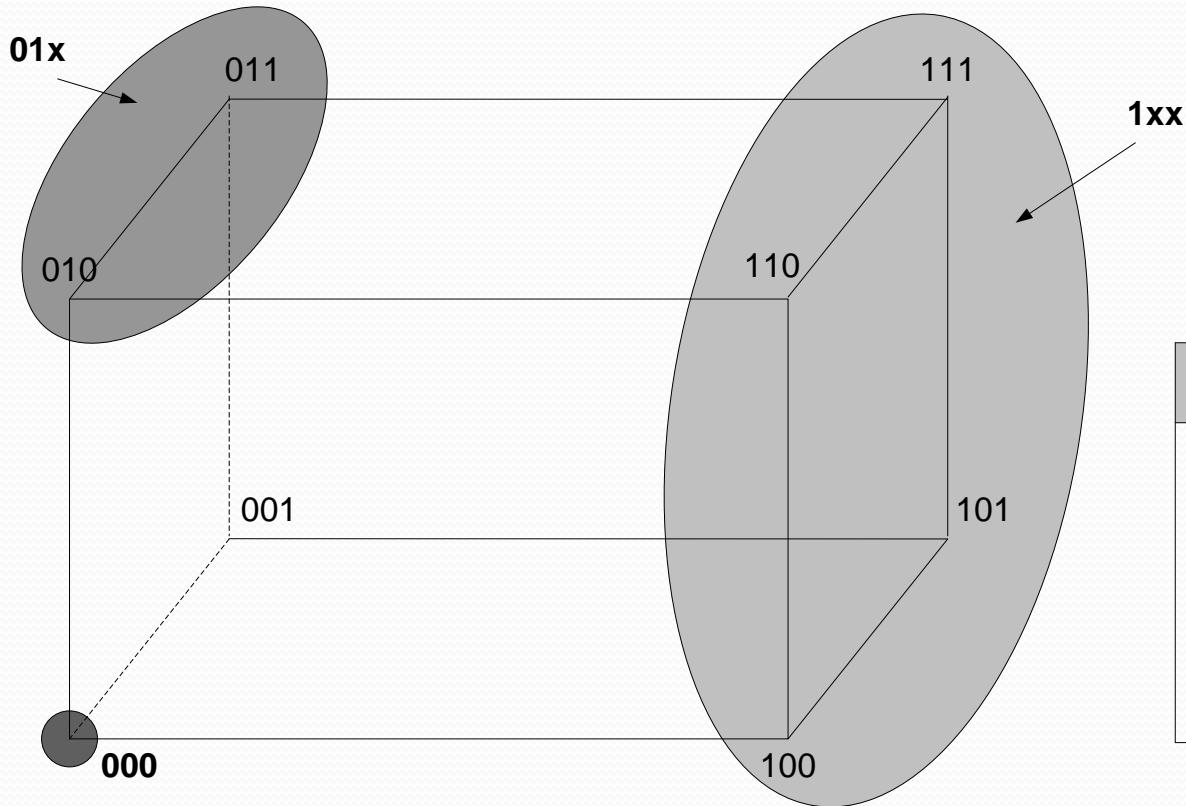
# Pembuktian GA Secara Matematis

- *Schema Theorem*
- Pengaruh seleksi orangtua
- Pengaruh rekombinasi
- Pengaruh mutasi
- Pengaruh rekombinasi dan mutasi

# *Schema Theorem*

- *Schema 1xx0* → kromosom 1000, 1010, 1100, 1110
- *Schema  $S_1 = 1x1xx0xx$*  memiliki
  - ***defining length*** (jarak antara simbol bukan x yang pertama dan terakhir)  $D(S_1) = 6 - 1 = 5$ .
  - ***order*** (jumlah simbol bukan x)  $o(S_1) = 3$ .

# Schema Theorem



*Schema  $S$ , Order  $O$ , dan Defining Length  $D$*

$S$	$O(S)$	$D(S)$
000	3	2
01x	2	1
1xx	1	0

# Pengaruh seleksi orang tua

- Misalkan  $\bar{f}(S)$  menyatakan rata-rata *fitness* dari suatu *schema* S dalam suatu populasi, yang didefinisikan sebagai rata-rata *fitness* dari semua kromosom yang termasuk dalam *schema* S tersebut.
- Dengan menggunakan seleksi orang tua yang proporsional terhadap nilai fitnessnya, maka probabilitas terpilihnya suatu kromosom dengan *fitness*  $f_i$  adalah  $f_i / f$  (dimana  $f = \sum_{i=1}^N f_i$  adalah total *fitness* dari semua  $N$  kromosom dalam populasi tersebut)

# Pengaruh seleksi orang tua

- Selanjutnya, misalkan  $\bar{f}$  menyatakan rata-rata *fitness* dalam populasi dengan  $N$  kromosom, yaitu

$$\bar{f} = f / N$$

- Banyaknya kopi dari *schema S* yang diharapkan pada generasi berikutnya ( $g+1$ ) adalah

$$\Gamma(S, g + 1) = N \frac{\bar{f}(S)\Gamma(S, g)}{f}$$

dimana  $\Gamma(S, g)$  menyatakan jumlah kopi dari *schema S* pada generasi  $g$ .

# Pengaruh seleksi orang tua

- Dengan menggunakan fakta bahwa  $N/f = 1/\bar{f}$ , maka diperoleh persamaan berikut ini

$$\Gamma(S, g + 1) = \frac{\bar{f}(S)}{\bar{f}} \Gamma(S, g)$$

# Pengaruh seleksi orang tua

- Jika suatu *schema* dihubungkan secara konsisten dengan rata-rata *fitness*-nya, yakni  $\bar{f}(S)/\bar{f} = 1 + \alpha > 1$ , dimana  $\alpha$  adalah konstanta, maka jumlah kopi dari *schema* saat ini yang berada dalam populasi akan tumbuh secara **eksponensial** berdasarkan waktu  $k$ .

$$\Gamma(S, g+k) = \Gamma(S, g)(1+\alpha)^k$$

	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	Fitness
k1	1	1	1	1	1	0	0	0	0	0	22
k2	1	1	1	1	1	0	0	0	0	1	18
.	0	1	1	1	1	1	1	0	0	0	15
.	0	0	0	0	0	0	0	0	0	1	15
.	0	0	0	1	1	1	0	1	1	1	10
.	0	0	1	1	1	1	0	0	0	1	10
.	1	0	0	0	1	1	1	1	1	0	5
k8	1	0	0	0	1	1	1	1	0	0	5

$$\bar{f}(S) = \frac{22 + 18}{2} = 20$$

$$\bar{f} = \frac{100}{8} = 12,5$$

$$\Gamma(S, g+1) = \frac{\bar{f}(S)}{\bar{f}} \Gamma(S, g)$$

$$\bar{f}(S) / \bar{f} = 1 + \alpha > 1$$

**S**

	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	Fitness
k1	1	1	1	1	1	0	0	0	0	0	40
k2	1	1	1	1	1	0	0	0	0	1	20
.	0	1	1	1	1	1	1	0	0	0	15
.	0	0	0	0	0	0	0	0	0	1	15
.	0	0	0	1	1	1	0	1	1	1	10
.	0	0	1	1	1	1	0	0	0	1	10
.	1	0	0	0	1	1	1	1	0	0	5
k8	1	1	1	0	0	0	1	1	1	0	5

$$\bar{f}(S) = \frac{40 + 20}{2} = 30$$

$$\bar{f} = \frac{120}{8} = 15$$

$$\Gamma(S, g+1) = \frac{\bar{f}(S)}{\bar{f}} \Gamma(S, g)$$

$$\bar{f}(S) / \bar{f} = 1 + \alpha > 1$$

s

# Pengaruh seleksi orang tua

- *Schema* yang rata-rata *fitness*-nya **lebih tinggi** dibandingkan rata-rata *fitness* semua individu akan **naik secara eksponensial** berdasarkan waktu  $k$ .
- Jumlah *schema* yang rata-rata *fitness*-nya **lebih rendah** dibandingkan rata-rata *fitness* semua individu akan **menurun secara eksponensial juga**.

$$\Gamma(S, g + 1) = \frac{\bar{f}(S)}{\bar{f}} \Gamma(S, g)$$

$$\Gamma(S, g + k) = \Gamma(S, g)(1 + \alpha)^k$$

# Pengaruh Rekombinasi

Probabilitas perusakan, selama rekombinasi, terhadap *schema* dengan *defining length*  $D(S)$  adalah:

$$p_d = \frac{D(S)}{n - 1}$$

dimana  $n$  adalah panjang kromosom.

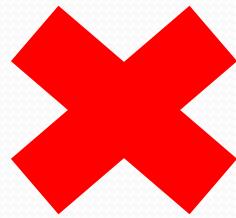
Semakin kecil  $D(S)$  semakin kecil pula  $p_d$ .

# Pengaruh Rekombinasi

Karena rekombinasi terjadi dengan probabilitas  $p_c$  maka *schema S* akan bertahan hidup dengan probabilitas sekitar

Yang di buku salah

$$\frac{1 - p_c D(S)}{(n - 1)}$$



$$1 - p_c \frac{D(S)}{(n - 1)}$$



dimana  $n$  adalah panjang kromosom.

# Pengaruh Mutasi

Probabilitas bahwa *schema S* tidak akan termutasi pada gen-gen yang bukan  $x$  adalah sama dengan

$$(1 - p_m)^{O(S)}$$

dimana  $O(S)$  adalah *order* dari *schema S* (jumlah simbol bukan  $x$  dari *schema S*).

# Pengaruh rekombinasi dan mutasi

Dengan memasukkan pengaruh rekombinasi dan mutasi, jumlah kopi dari *schema S* pada generasi  $g+1$  adalah

$$\Gamma(S, g+1) \geq \frac{\bar{f}(S)}{\bar{f}} \Gamma(S, g) \left(1 - p_c \frac{D(S)}{n-1}\right) (1 - p_m)^{O(S)}$$

$$\approx \frac{\bar{f}(S)}{\bar{f}} \Gamma(S, g) \left(1 - p_c \frac{D(S)}{n-1}\right) (1 - O(S)p_m)$$

$$\approx \frac{\bar{f}(S)}{\bar{f}} \Gamma(S, g) \left(1 - p_c \frac{D(S)}{n-1} - O(S)p_m\right)$$

# Studi kasus: Minimasi fungsi

Nilai minimum  $h = ?$

$$h(x_1, x_2) = x_1^2 + x_2^2$$

$$x_1, x_2 \in [-5,12; 5,12]$$

# Individu

x <sub>1</sub>										x <sub>2</sub>									
1	1	0	1	0	1	0	1	0	0	0	1	0	0	1	1	0	1	0	1
$g_1$										$g_{10}$	$g_{11}$							$g_{20}$	

# *Fitness*

$$f = \frac{1}{(x_1^2 + x_2^2) + 0,01}$$

Jika nilai minimum = 0, nilai maks  $f = ?$

# Generasi 1

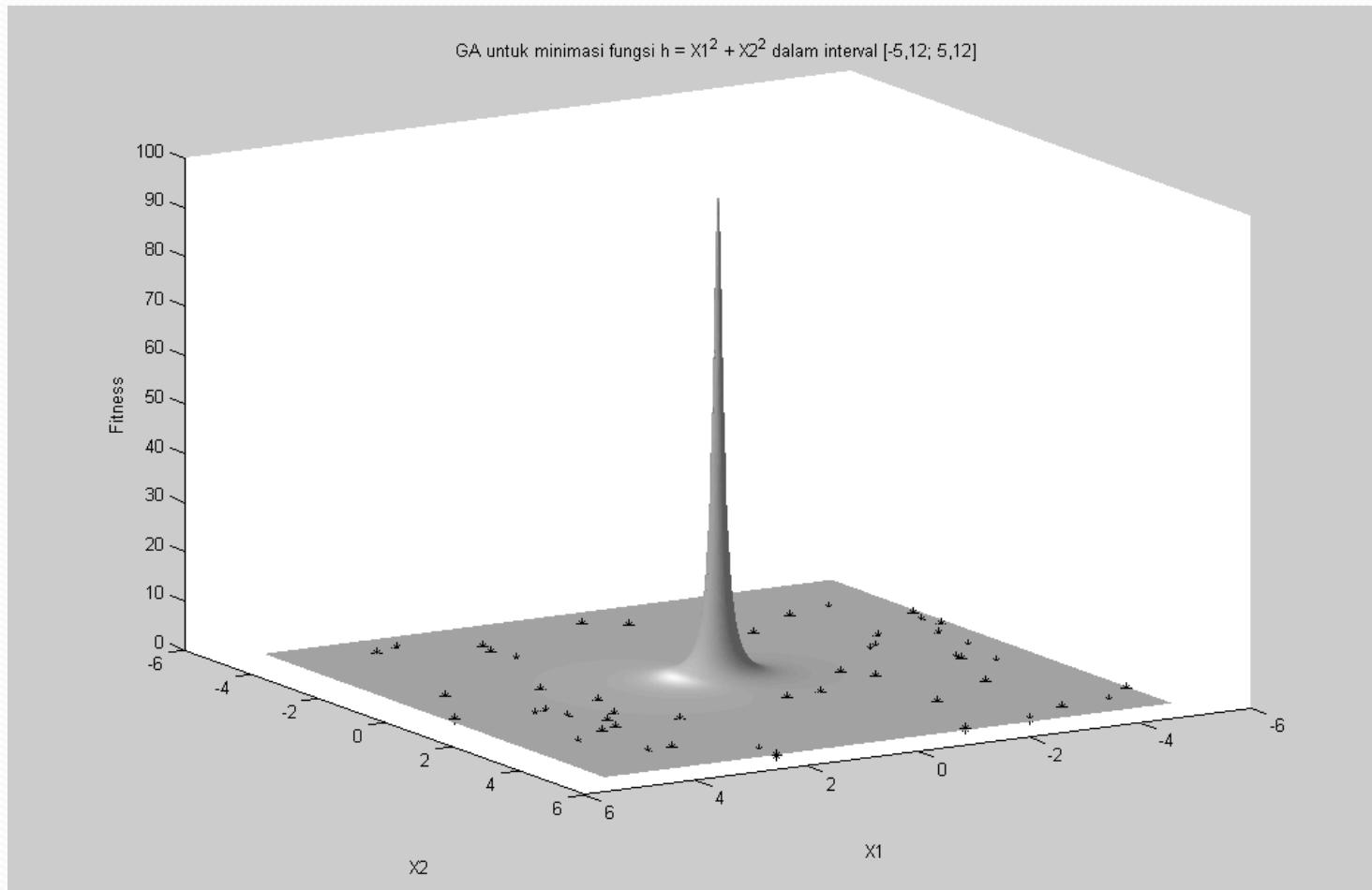
$$x = r_b + \frac{(r_a - r_b)}{\sum_{i=1}^N 2^{-i}} (g_1 \cdot 2^{-1} + g_2 \cdot 2^{-2} + \dots + g_N \cdot 2^{-N})$$

$$x_1, x_2 \in [-5,12; 5,12]$$



No	Genotype	Phenotype		Nilai fitness
	kromosom biner	X1	X2	
1	00010011011001101110	-4.35	1.1	0.049646
2	11001101110001000011	3.11	-4.45	0.033916
3	10110010111111001110	2.03	4.62	0.039254
4	11001110001101111101	3.12	3.81	0.041219
5	11001110101011011001	3.14	2.17	0.068594
6	00101110000110110110	-3.28	-0.74	0.08837
7	01111011111010110010	-0.17	1.78	0.31179
50	11010110011000111011	3.45	0.59	0.081562

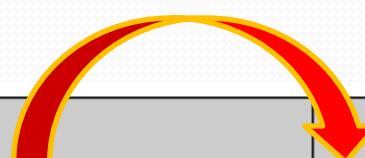
# Generasi 1



# Generasi 10

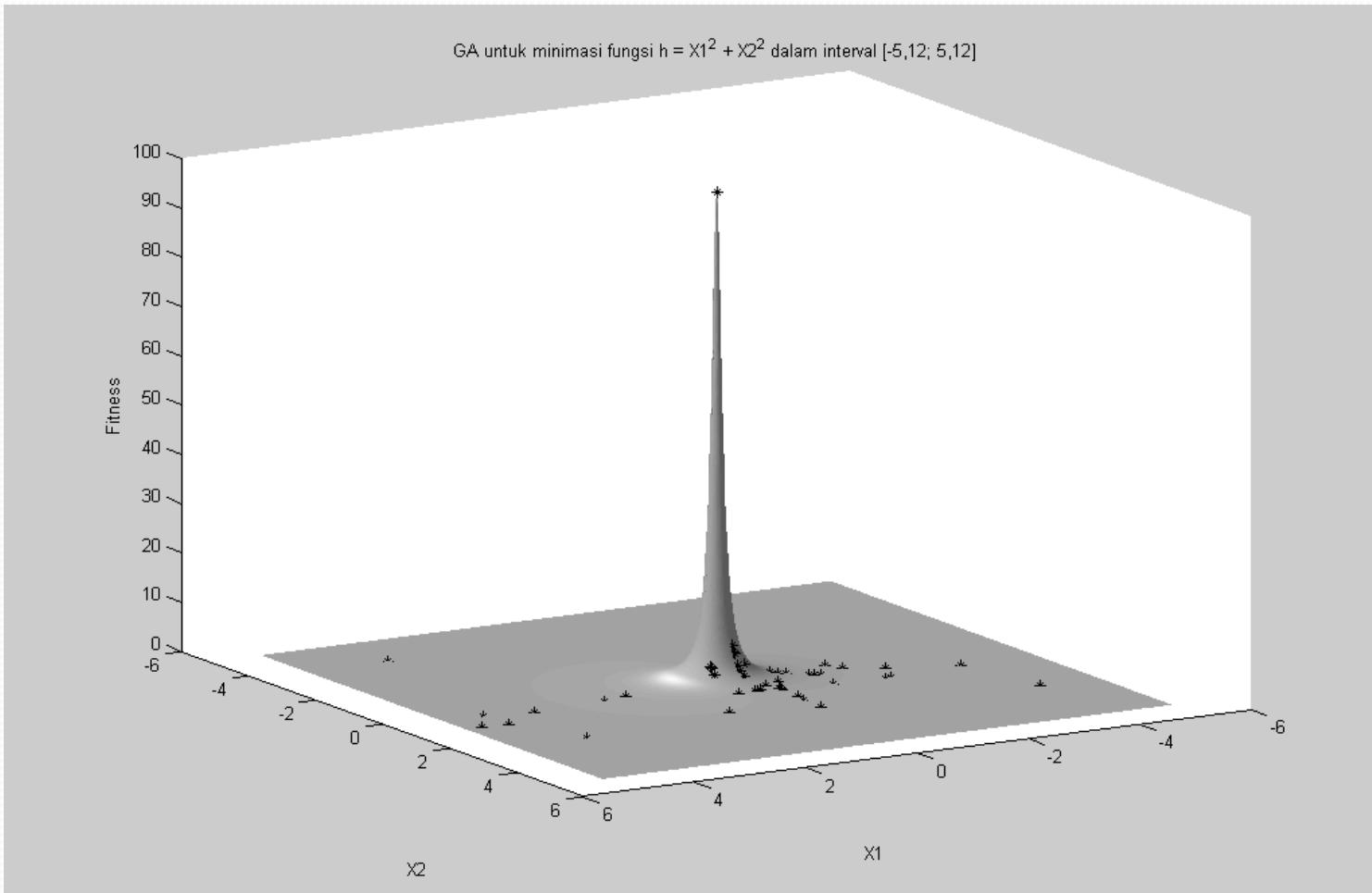
$$x = r_b + \frac{(r_a - r_b)}{\sum_{i=1}^N 2^{-i}} (g_1 \cdot 2^{-1} + g_2 \cdot 2^{-2} + \dots + g_N \cdot 2^{-N})$$

$$x_1, x_2 \in [-5,12; 5,12]$$

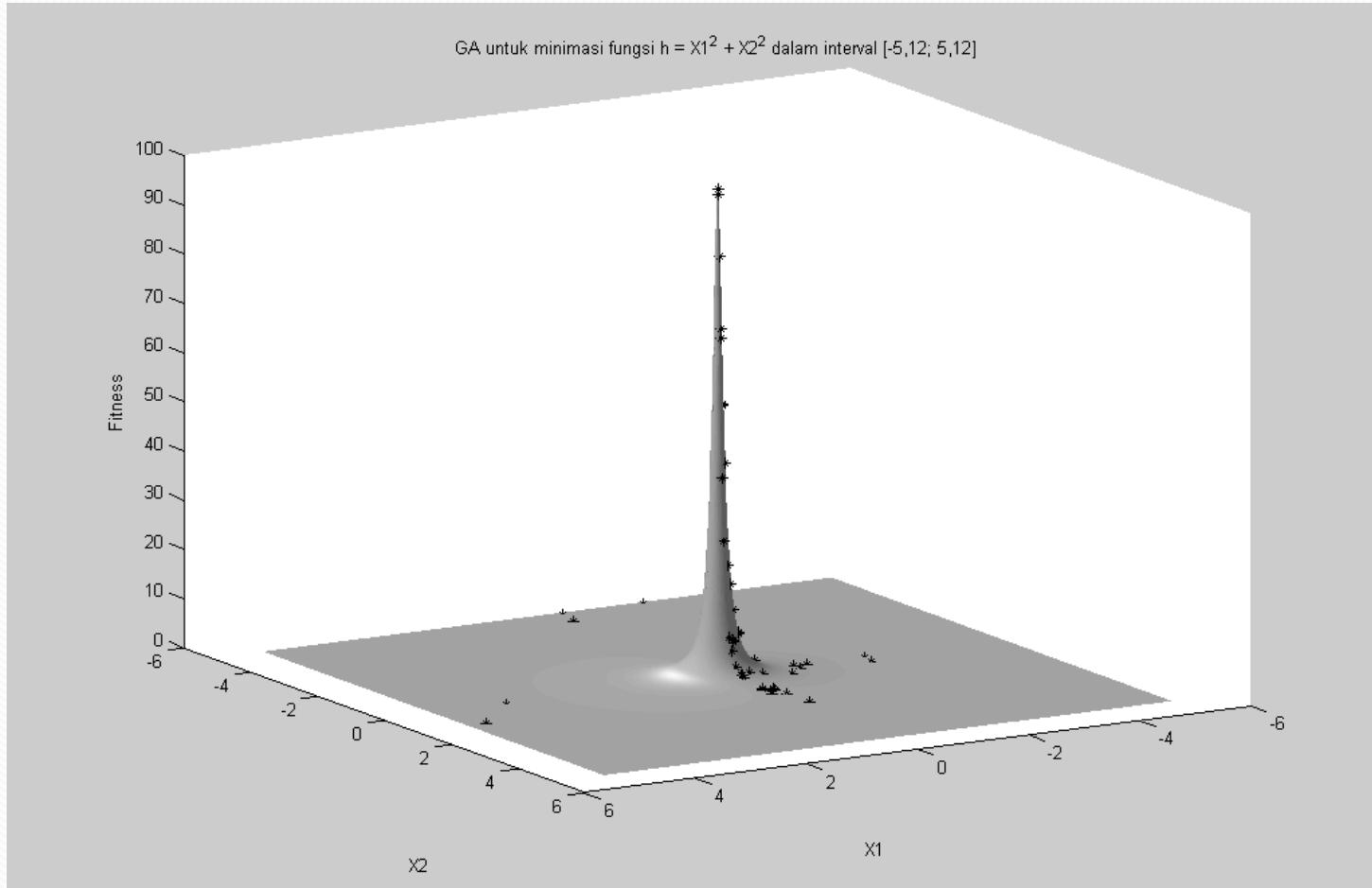


No	Genotype	Phenotype		Nilai fitness
	kromosom biner	X1	X2	
1	01111111110000000000	-0.01	0	99.01
2	01111111110000000000	-0.01	0	99.01
3	01111101010001000001	-3.77	1.03	0.065429
4	01111101011001110001	-2.4	1.2	0.1387
5	01111001111000100001	3.58	0.52	0.076355
6	01011101111000101010	4.83	1.01	0.041053
7	01111101111000100001	-1.38	1.2	0.29812
...				
50	01111101111000000001	-1.93	0.02	0.26772

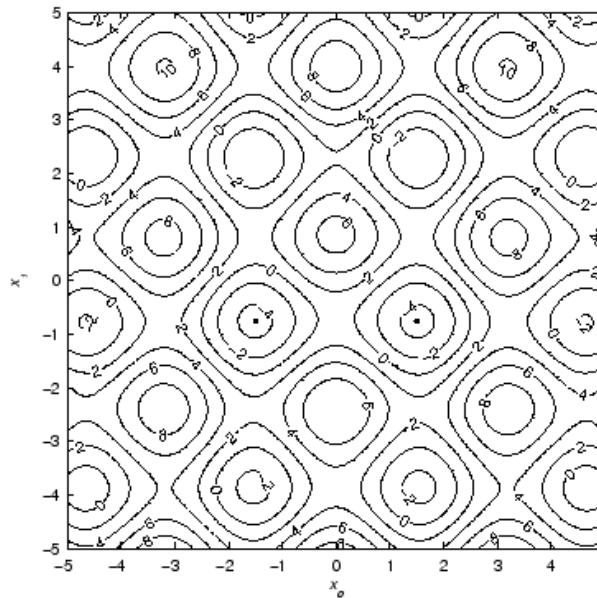
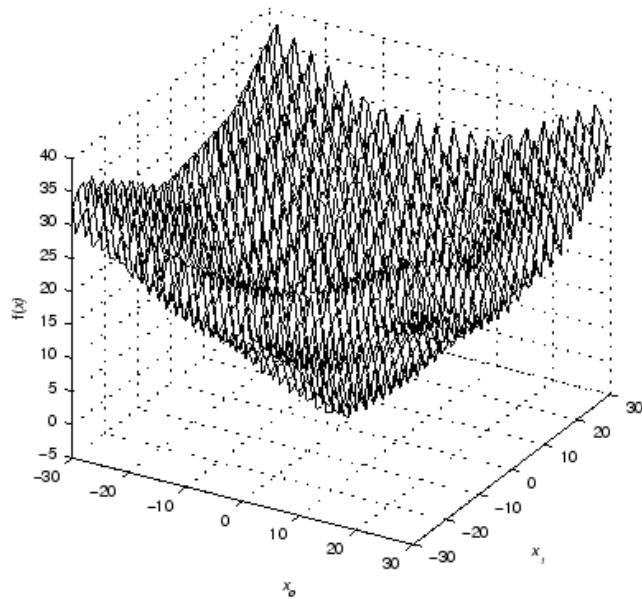
# Generasi 10



# Generasi 100



$$f(\vec{x}) = \sum_{i=0}^{D-1} \left( e^{-0.2} \sqrt{x_i^2 + x_{i+1}^2} + 3(\cos(2x_i) + \sin(2x_{i+1})) \right)$$



- Untuk presisi  $10^{-9}$  → Berapa bit?
- Bisa menggunakan kromosom Real?

# Setting parameter GA

- Tidak ada panduan yang pasti
- Hanya dengan Intuisi
- Umumnya:
  - Representasi kromosom = biner/integer/real/permutasi
  - Jumlah bit per variabel → presisi yang diinginkan
  - Ukuran Populasi = **50 - 100**
  - Probabilitas Crossover (Pc) = **0,8**
  - Probabilitas Mutasi (Pm) =  **$1/NL$  sampai  $1/L$**   
 $N$  = Ukuran Populasi  
 $L$  = Panjang Kromosom (Jumlah Gen)

# Observasi parameter GA

- Minimasi fungsi  $h = x_1^2 + x_2^2$ ,  $x_1$  dan  $x_2$  elemen  $[-10, 10]$
- Fitness =  $1/(x_1^2 + x_2^2 + 0.001)$

- Ukuran Populasi = [50 100 200]
- Jumlah bit = [10 50 90]
- Prob Rekombinasi = [0.5 0.7 0.9]
- Prob Mutasi = [0.5/JumGen 1/JumGen 2/JumGen]

- Jumlah Individu maksimum = 20000 (*fairness*)
- Jumlah running/percobaan = 30 (*valid*)

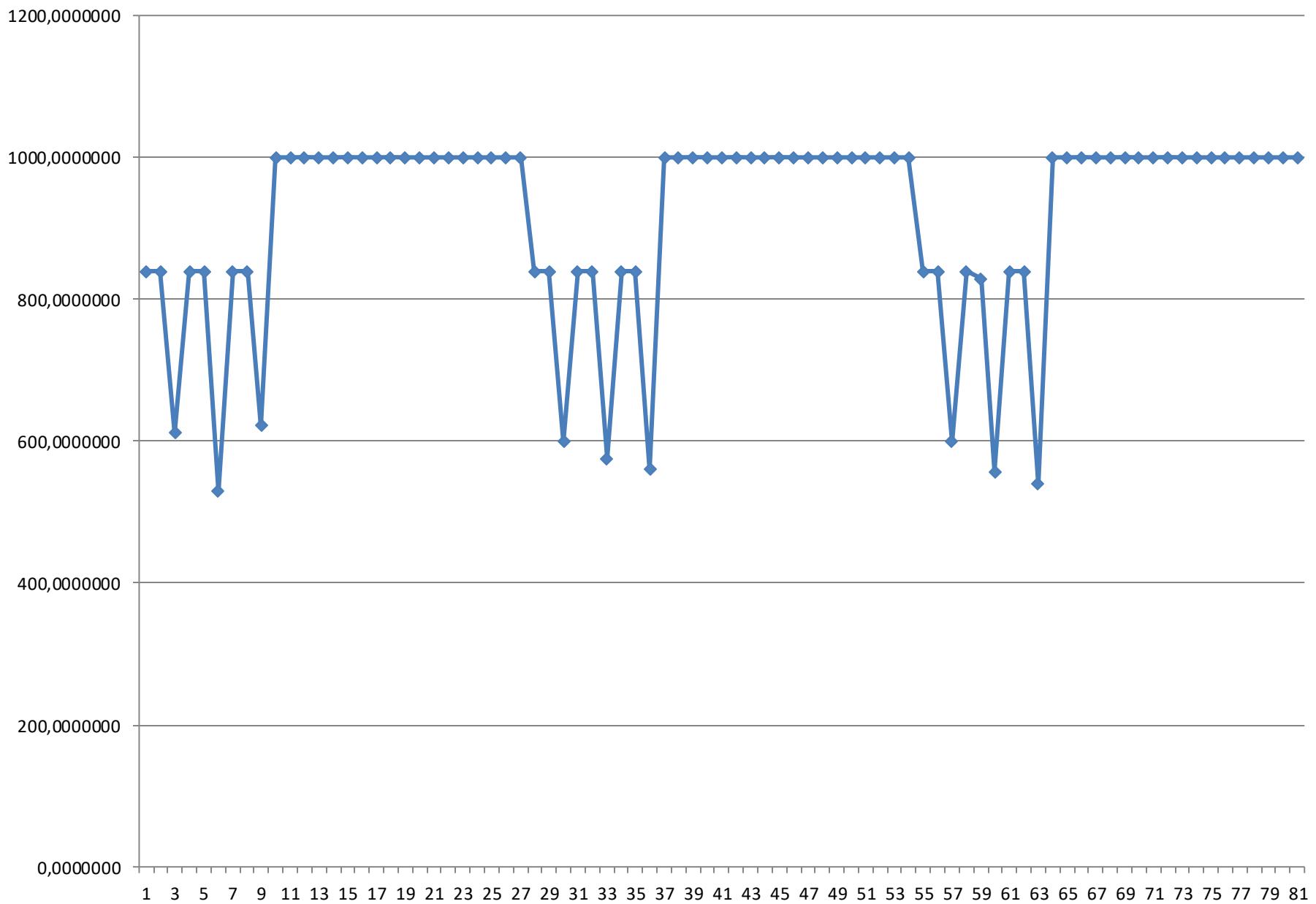
No Observasi	Ukuran populasi	Jumlah bit	Probabilitas Pdh Silang	Probabilitas Mutasi	Rata-rata Fitness terbaik	Rata-rata Jml Individu yang dievaluasi
1	50	10	0,5	0,0250	839,5544749	20000,0000
2	50	10	0,5	0,0500	839,5544749	20000,0000
3	50	10	0,5	0,1000	611,0770624	20000,0000
4	50	10	0,7	0,0250	839,5544749	20000,0000
5	50	10	0,7	0,0500	839,5544749	20000,0000
6	50	10	0,7	0,1000	528,7161733	20000,0000
7	50	10	0,9	0,0250	839,5544749	20000,0000
8	50	10	0,9	0,0500	839,5544749	20000,0000
9	50	10	0,9	0,1000	622,2201392	20000,0000
10	50	50	0,5	0,0050	1000,0000000	8301,6667
11	50	50	0,5	0,0100	1000,0000000	20000,0000
12	50	50	0,5	0,0200	999,9987777	20000,0000
13	50	50	0,7	0,0050	1000,0000000	8013,3333
14	50	50	0,7	0,0100	1000,0000000	20000,0000
15	50	50	0,7	0,0200	999,9982015	20000,0000
16	50	50	0,9	0,0050	1000,0000000	8133,3333
17	50	50	0,9	0,0100	1000,0000000	20000,0000
18	50	50	0,9	0,0200	999,9988782	20000,0000
19	50	90	0,5	0,0028	1000,0000000	8361,6667
20	50	90	0,5	0,0056	1000,0000000	8796,6667
21	50	90	0,5	0,0111	1000,0000000	20000,0000
22	50	90	0,7	0,0028	1000,0000000	8151,6667
23	50	90	0,7	0,0056	1000,0000000	8780,0000
24	50	90	0,7	0,0111	1000,0000000	20000,0000
25	50	90	0,9	0,0028	1000,0000000	7538,3333
26	50	90	0,9	0,0056	1000,0000000	8995,0000
27	50	90	0,9	0,0111	1000,0000000	20000,0000
28	100	10	0,5	0,0250	839,5544749	20000,0000
29	100	10	0,5	0,0500	839,5544749	20000,0000
30	100	10	0,5	0,1000	599,4452769	20000,0000

Paket parameter terbaik untuk kasus di atas

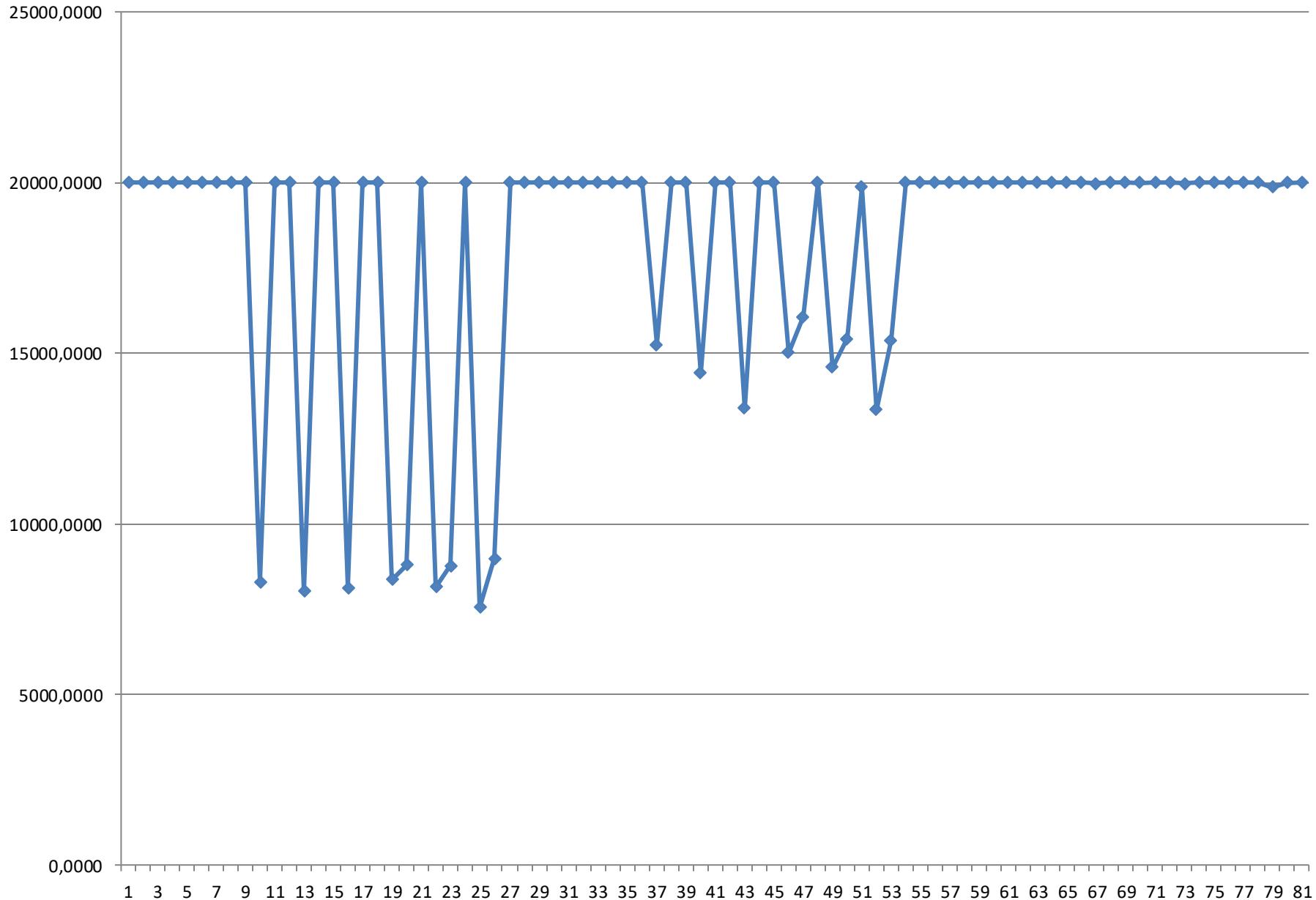
No Observasi	Ukuran populasi	Jumlah bit	Probabilitas Pdh Silang	Probabilitas Mutasi	Rata-rata Fitness terbaik	Rata-rata Jml Individu yang dievaluasi
31	100	10	0,7	0,0250	839,5544749	20000,0000
32	100	10	0,7	0,0500	839,5544749	20000,0000
33	100	10	0,7	0,1000	575,8472869	20000,0000
34	100	10	0,9	0,0250	839,5544749	20000,0000
35	100	10	0,9	0,0500	839,5544749	20000,0000
36	100	10	0,9	0,1000	559,6804844	20000,0000
37	100	50	0,5	0,0050	1000,0000000	15246,6667
38	100	50	0,5	0,0100	1000,0000000	20000,0000
39	100	50	0,5	0,0200	999,9986429	20000,0000
40	100	50	0,7	0,0050	1000,0000000	14416,6667
41	100	50	0,7	0,0100	1000,0000000	20000,0000
42	100	50	0,7	0,0200	999,9988459	20000,0000
43	100	50	0,9	0,0050	1000,0000000	13390,0000
44	100	50	0,9	0,0100	1000,0000000	20000,0000
45	100	50	0,9	0,0200	999,9987118	20000,0000
46	100	90	0,5	0,0028	1000,0000000	15010,0000
47	100	90	0,5	0,0056	1000,0000000	16056,6667
48	100	90	0,5	0,0111	1000,0000000	20000,0000
49	100	90	0,7	0,0028	1000,0000000	14580,0000
50	100	90	0,7	0,0056	1000,0000000	15430,0000
51	100	90	0,7	0,0111	1000,0000000	19860,0000
52	100	90	0,9	0,0028	1000,0000000	13346,6667
53	100	90	0,9	0,0056	1000,0000000	15390,0000
54	100	90	0,9	0,0111	1000,0000000	20000,0000
55	200	10	0,5	0,0250	839,5544749	20000,0000
56	200	10	0,5	0,0500	839,5544749	20000,0000
57	200	10	0,5	0,1000	599,0108676	20000,0000
58	200	10	0,7	0,0250	839,5544749	20000,0000
59	200	10	0,7	0,0500	828,6149185	20000,0000
60	200	10	0,7	0,1000	557,3828866	20000,0000

No Observasi	Ukuran populasi	Jumlah bit	Probabilitas Pdh Silang	Probabilitas Mutasi	Rata-rata Fitness terbaik	Rata-rata Jml Individu yang dievaluasi
61	200	10	0,9	0,0250	839,5544749	20000,0000
62	200	10	0,9	0,0500	839,5544749	20000,0000
63	200	10	0,9	0,1000	539,1055371	20000,0000
64	200	50	0,5	0,0050	1000,0000000	20000,0000
65	200	50	0,5	0,0100	999,9999995	20000,0000
66	200	50	0,5	0,0200	999,9986789	20000,0000
67	200	50	0,7	0,0050	1000,0000000	19966,6667
68	200	50	0,7	0,0100	999,9999997	20000,0000
69	200	50	0,7	0,0200	999,9947933	20000,0000
70	200	50	0,9	0,0050	1000,0000000	19986,6667
71	200	50	0,9	0,0100	999,9999996	20000,0000
72	200	50	0,9	0,0200	999,9939550	20000,0000
73	200	90	0,5	0,0028	999,9999988	19966,6667
74	200	90	0,5	0,0056	999,9999999	20000,0000
75	200	90	0,5	0,0111	999,9999976	20000,0000
76	200	90	0,7	0,0028	999,9999995	20000,0000
77	200	90	0,7	0,0056	1000,0000000	20000,0000
78	200	90	0,7	0,0111	999,9999979	20000,0000
79	200	90	0,9	0,0028	1000,0000000	19866,6667
80	200	90	0,9	0,0056	1000,0000000	19993,3333
81	200	90	0,9	0,0111	999,9999988	20000,0000

## Rata-rata Fitness Terbaik



## Rata-rata Jumlah Individu yg Dievaluasi



# TSP

Masalah TSP dengan 20 kota. Terdapat **121.645 trilyun** solusi yang mungkin.

Solusi 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Solusi 2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	20	19
•																				
•																				
•																				
121.645 trilyun																				
20   2   1   3   4   5   6   7   8   9   10   11   12   13   14   15   16   17   18   19																				

Penyelesaian berbasis **EA** dengan populasi 100 individu

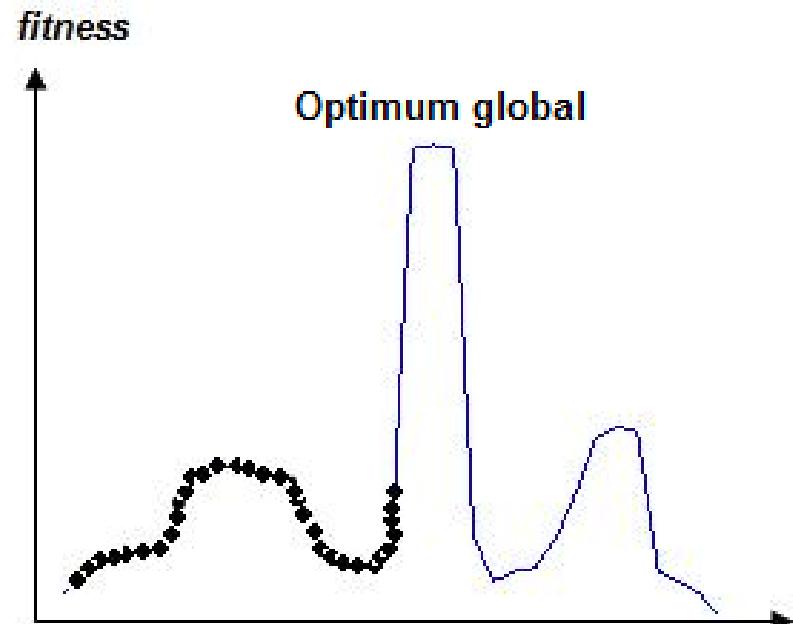
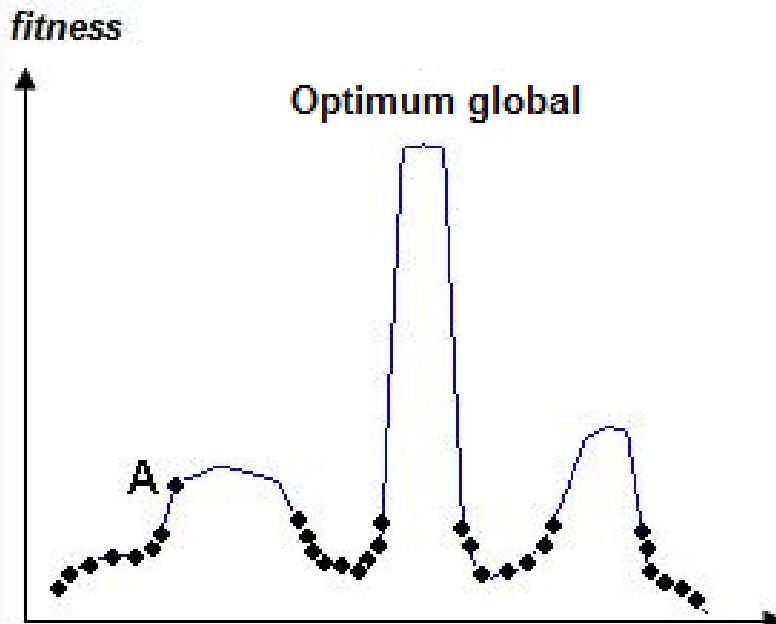
Individu 1	10	2	3	4	5	6	7	8	1	9	11	12	13	14	15	16	17	18	19	20
Individu 2	14	5	8	16	2	17	9	3	7	10	11	12	13	1	18	4	6	20	15	19
•																				
Individu 99	20	5	11	16	3	18	2	6	19	7	1	17	8	15	12	10	14	4	13	9
Individu 100	8	1	3	7	11	6	4	18	9	16	5	12	20	14	17	10	15	2	19	13

Dengan setting parameter yang tepat, GA bisa menemukan solusi secara cepat dan akurat.

# Advanced GA

- Konvergensi Prematur (KP)
- *Gray Coding*
- *Messy Encoding*
- *Fitness Ranking*
- *Island Model*
- *Adaptive GA*
- *Grid-Based Crossover*
- *Grammatical Encoding*

# Konvergensi Prematur (KP)



# Pencegahan KP

- *Gray Coding*
- *Messy Encoding*
- *Fitness Ranking*
- *Island Model*
- *Adaptive GA*

# *Hamming Distance*

Integer

14

0	1	1	1	0
---	---	---	---	---

Biner



Berbeda 1 bit

15

0	1	1	1	1
---	---	---	---	---



Berbeda 5 bit

16

1	0	0	0	0
---	---	---	---	---

# *Binary Coding*

- Jika solusi maksimum yang dicari adalah **10000** (16)
- Individu terbaik saat ini **01111** (15)
- Sampai beberapa generasi berikutnya ternyata individu terbaik tetap **01111** (15).
- Mengapa?
- **01111** → **10000** memerlukan mutasi 5 gen.
- Padahal probabilitas mutasi biasanya dibuat sangat kecil, biasanya  $1/NL$  sampai  $1/L$ , dimana  $N$  adalah ukuran populasi,  $L$  panjang kromosom.
- SOLUSINYA?

# *Gray Coding*

<b>Integer</b>	0	1	2	3	4	5	6	7
<b>Binary coding</b>	000	001	010	011	100	101	110	111
<b>Gray coding</b>	000	001	<b>011</b>	<b>010</b>	<b>110</b>	<b>111</b>	<b>101</b>	<b>100</b>

# *Gray Coding*

individu:  $x_1 = 5$  dan  $x_2 = 3$



$x_1$			$x_2$		
1	0	1	0	1	1

*Binary coding*



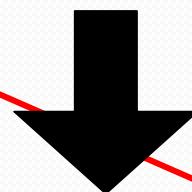
$x_1$			$x_2$		
1	1	1	0	1	0

Kromosom: *Gray coding*

# *Messy Encoding*

***Messy encoding:***

3	0	8	0	2	1	6	1	1	0	4	1	5	1	7	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

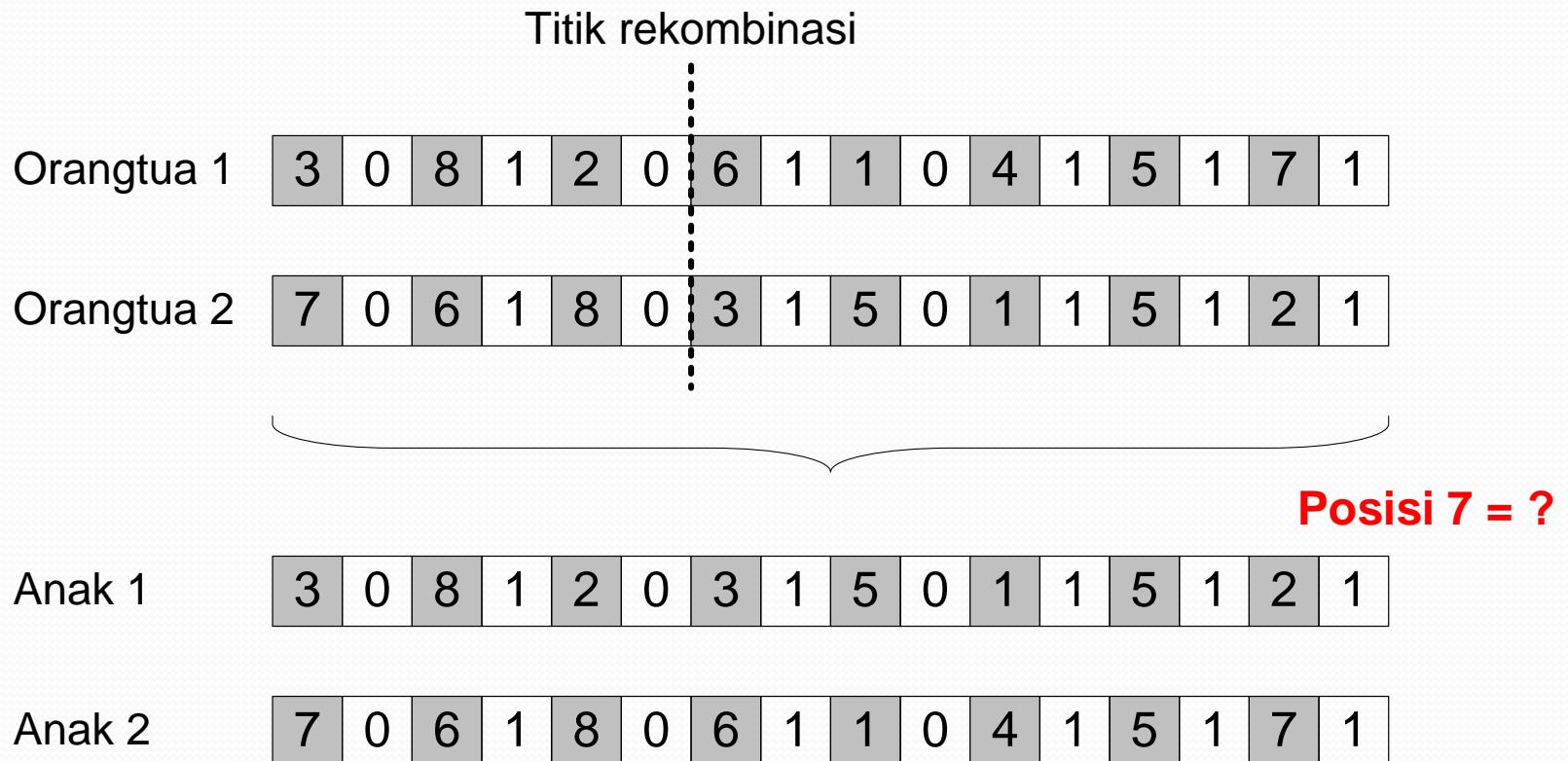


***Binary encoding:***

0	1	0	1	1	1	1	0
---	---	---	---	---	---	---	---

1    2    3    4    5    6    7    8

# Crossover



## ***Messy encoding:***

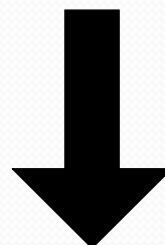
3	0	8	1	2	0	3	1	5	0	1	1	5	1	2	1
3	0	8	1	2	0	3	1	5	0	1	1	5	1	2	1

dipanjangkan  
*n* kali semula

6	0	4	1	8	1	3	1	2	0	5	1	4	1	5	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

## ***Binary encoding:***

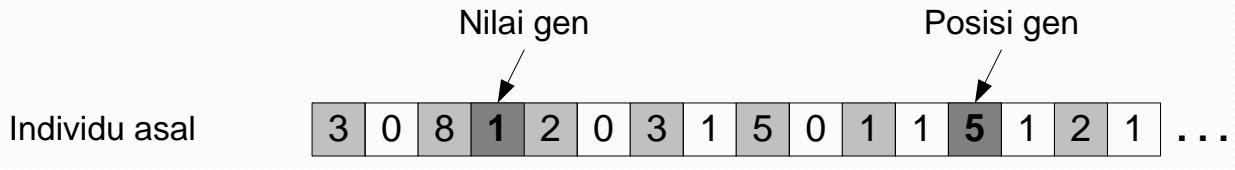
1	0	0	1	0	0	0	1
---	---	---	---	---	---	---	---



Nilai gen untuk posisi 7 = 0  
(dibangkitkan secara acak)

# Mutasi

***Messy encoding:***



***Binary encoding:***

1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---



1	0	0	1	1	0	1	0
---	---	---	---	---	---	---	---

# *Linear Fitness Ranking*

Maksimasi  $h$  dimana  $x_1$  dan  $x_2$  adalah real  $[-2, 2]$ )?

$$h(x_1, x_2) = 100000 + 2x_1 + x_2$$

# *Linear Fitness Ranking*

- Pada fungsi di atas, nilai-nilai  $h$  berada dalam interval 99994 sampai 100006.
- Dengan demikian, semua individu memiliki nilai *fitness* yang hampir sama dalam kisaran 100000.
- Hal ini akan berakibat buruk pada proses seleksi orangtua secara proporsional terhadap *fitness*-nya.
- Bagaimana Solusinya?

# *Linear Fitness Ranking*

$$f_{LR}(i) = f_{\max} - (f_{\max} - f_{\min}) \left( \frac{R(i)-1}{N-1} \right)$$

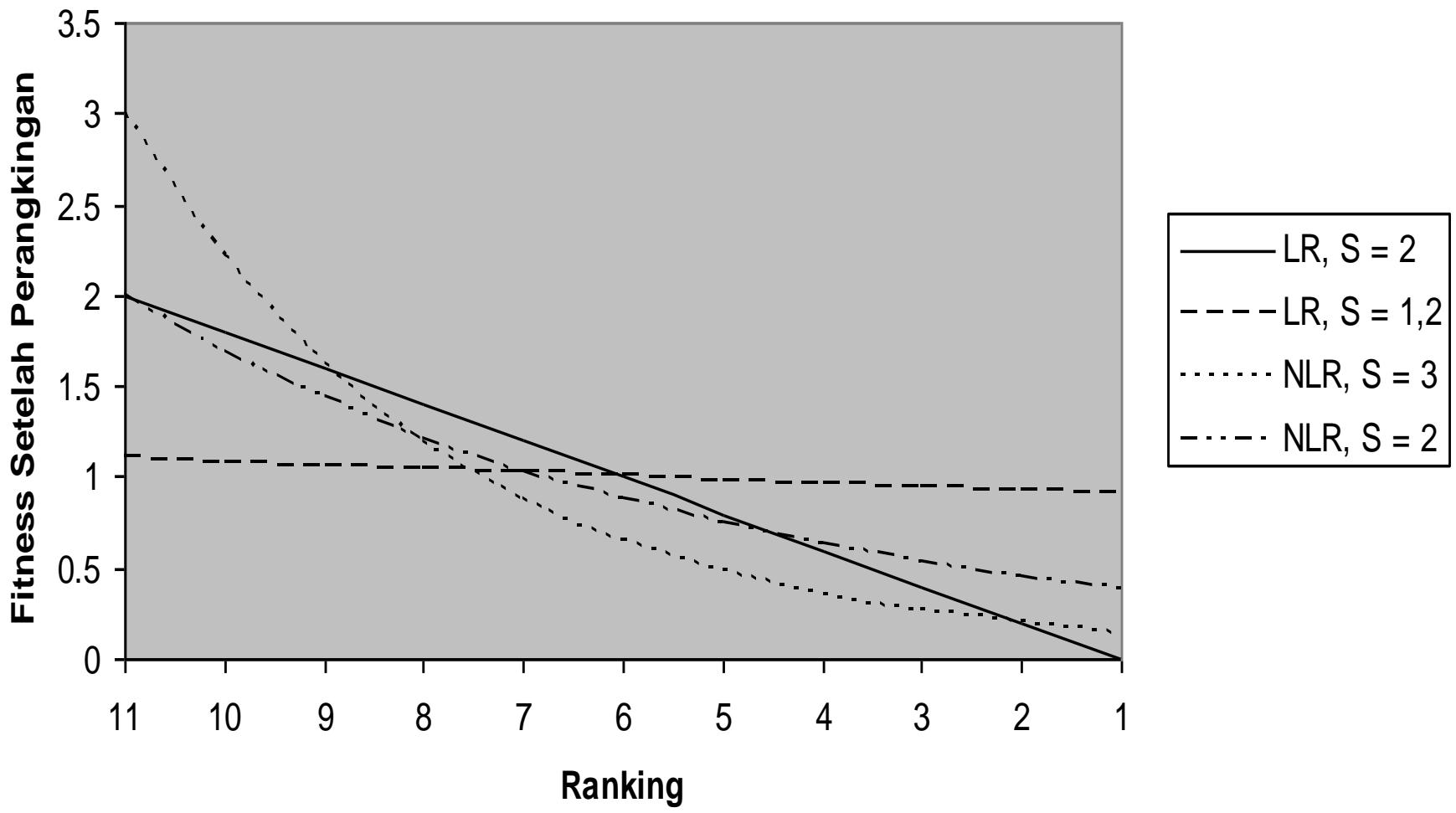
$f_{LR}(i) \rightarrow$  nilai *fitness* individu ke- $i$  yang sudah diskalakan  
 $N \rightarrow$  jumlah individu dalam populasi.

$R(i) \rightarrow$  ranking individu ke  $i$ .

$f_{\min} \rightarrow$  nilai *fitness* terkecil

$f_{\max} \rightarrow$  nilai *fitness* terbesar

<b>Individu ke</b>	<b><i>fitness</i></b>	<b>Ranking <math>R(i)</math></b>	<b><math>f_{LR}</math></b>
1	100004,00	1	100004,00
2	100003,99	2	100003,56
3	100003,98	3	100003,12
4	100003,97	4	100002,68
5	100003,96	5	100002,26
6	100003,95	6	100001,82
7	100003,94	7	100001,38
8	100003,93	8	100000,84
9	100003,92	9	100000,44
10	100000,00	10	100000,00



<b><i>Fitness</i></b>	<b><i>Posisi</i></b>	<b><i>Fitness Perangkingan</i></b>			
		<b>LR, S = 2</b>	<b>LR, S = 1,2</b>	<b>NLR, S = 3</b>	<b>NLR, S = 2</b>
90,11	11	2	1,1	3	2
90,09	10	1,8	1,08	2,21	1,69
90,08	9	1,6	1,06	1,62	1,43
90,06	8	1,4	1,04	1,19	1,21
90,05	7	1,2	1,02	0,88	1,03
89,97	6	1	1	0,65	0,87
89,96	5	0,8	0,98	0,48	0,74
89,95	4	0,6	0,96	0,35	0,62
79,94	3	0,4	0,94	0,26	0,53
79,93	2	0,2	0,92	0,19	0,45
79,91	1	0	0,9	0,14	0,38

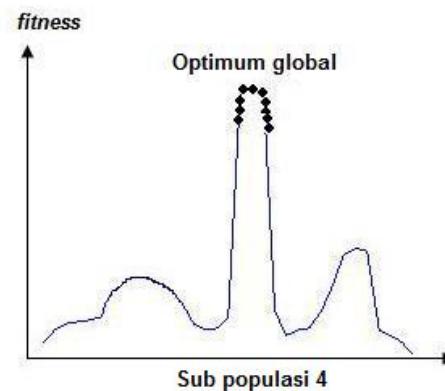
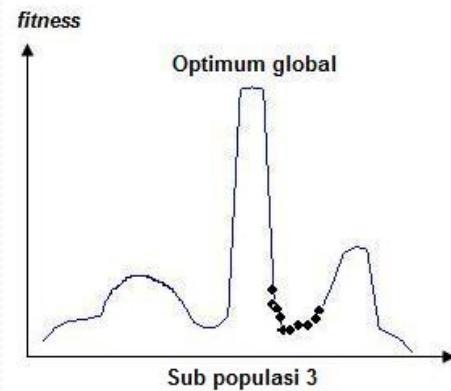
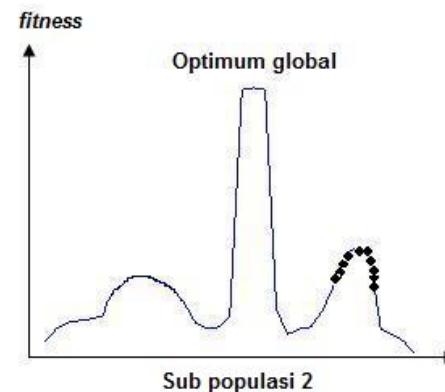
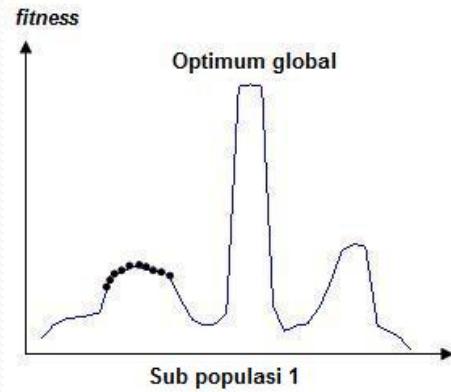
# *Island Models (Sub Population)*

N kromosom dalam satu populasi dibagi menjadi  $N_k$  kelompok. Masing-masing kelompok berisi:

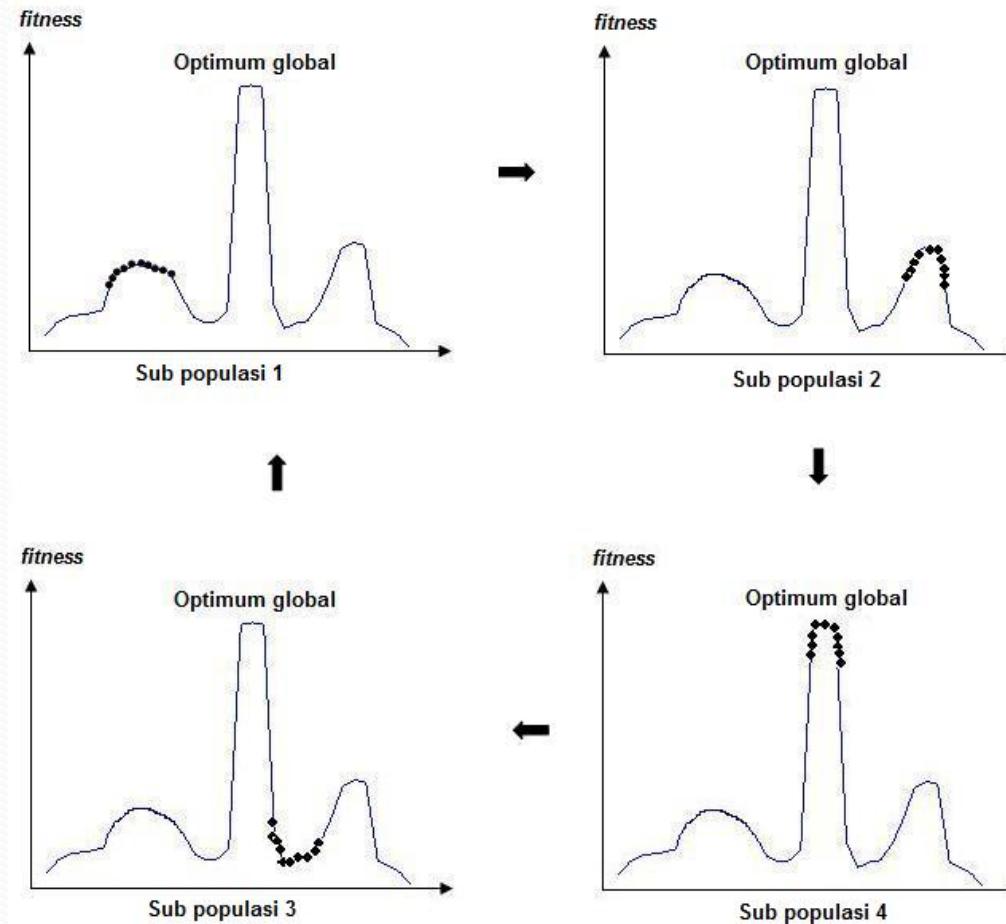
$$v = \frac{N}{N_k}$$

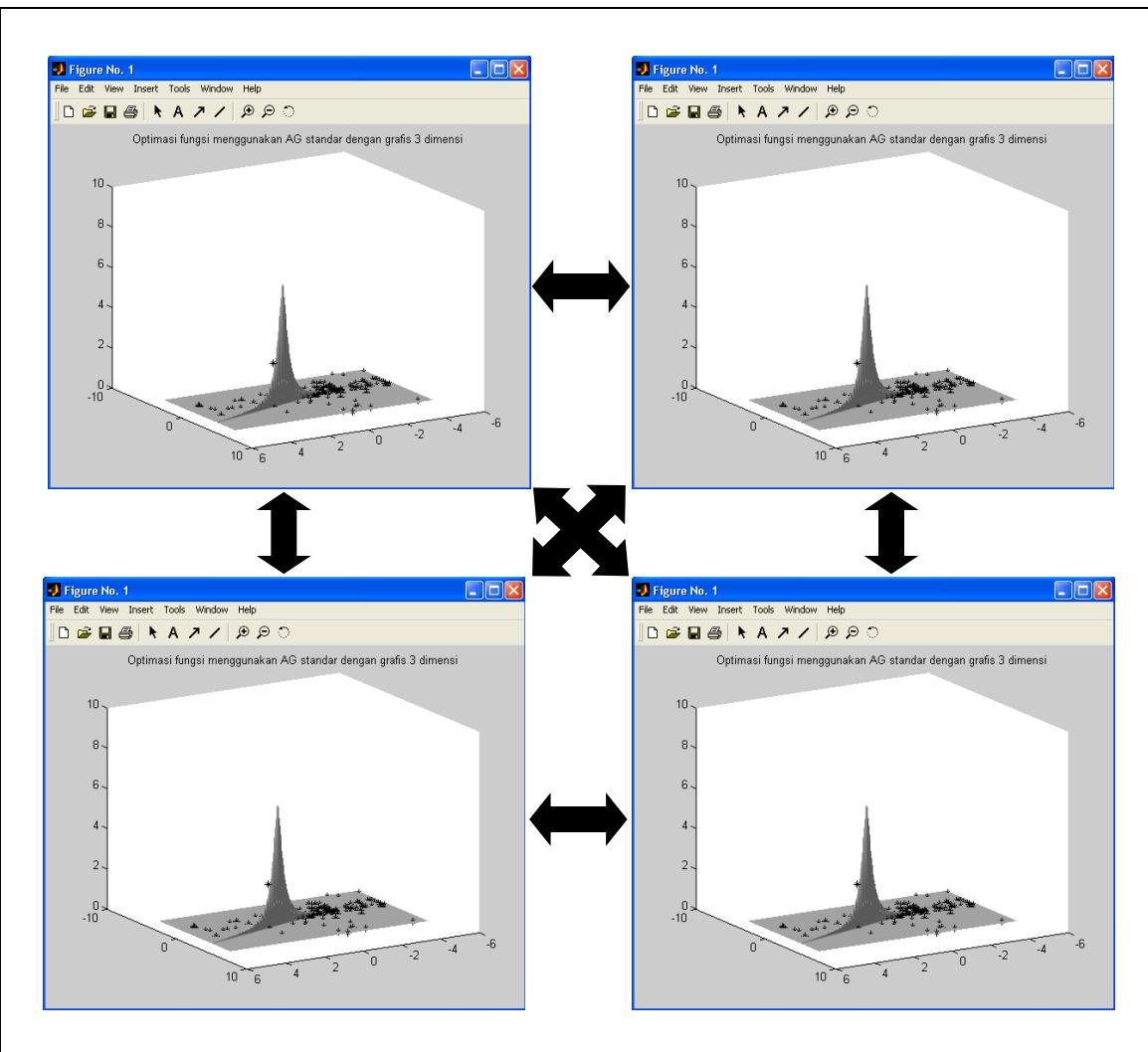
Suatu individu bisa dipindah ke sub populasi lain berdasarkan *tunneling probability*  $p_t$

# *Island model* EAs



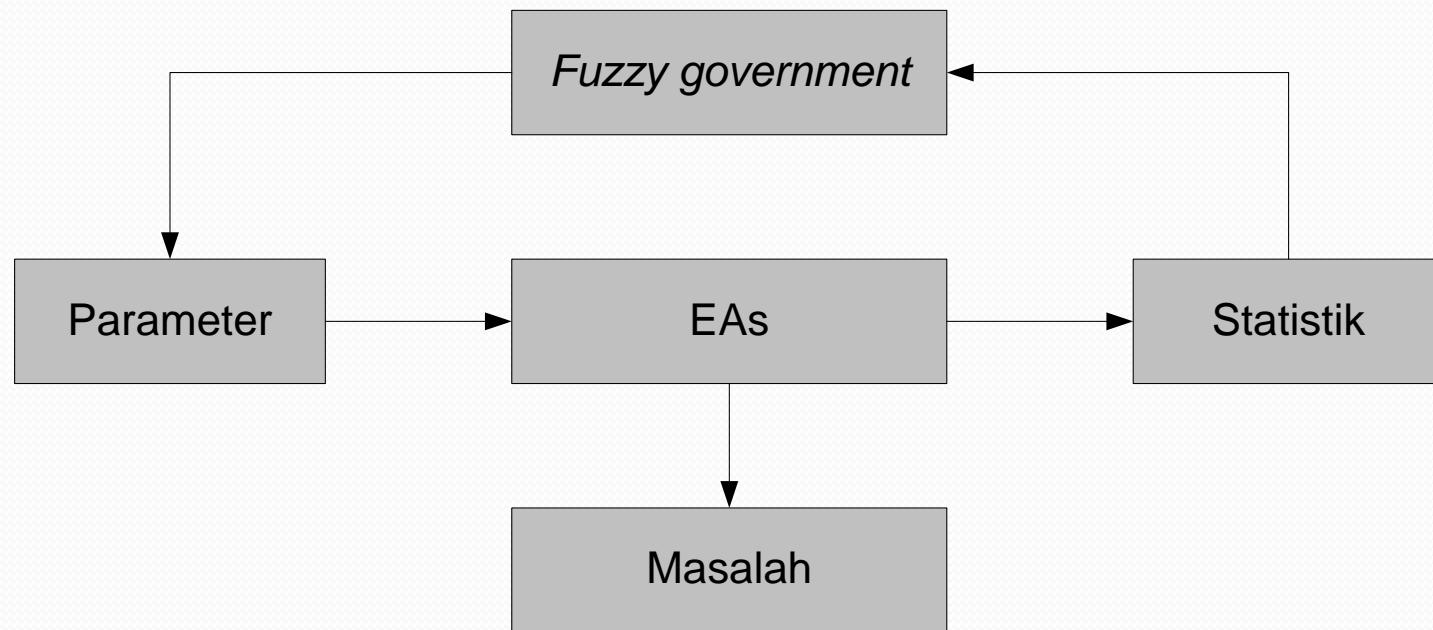
# *Island model* EAs



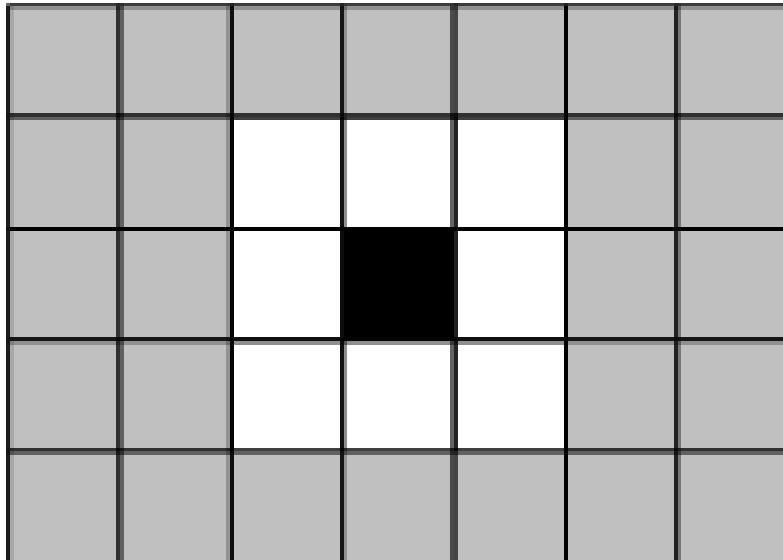


The best  
individual

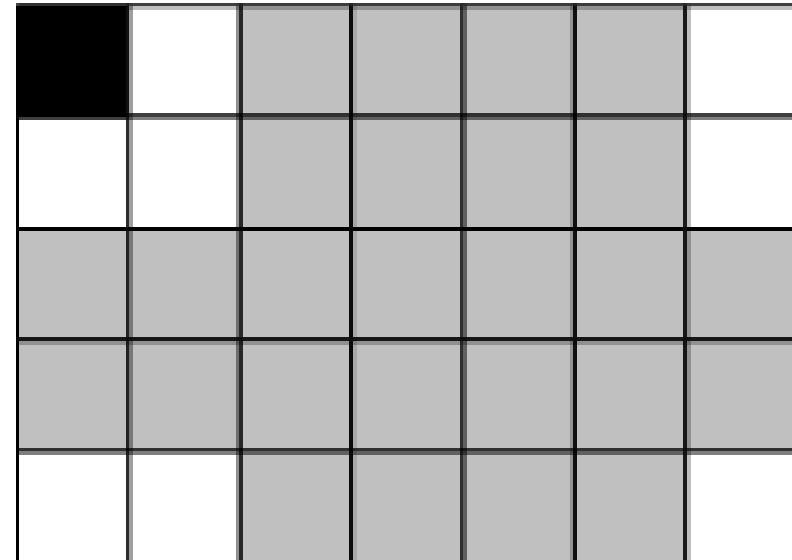
# *Adaptive EAs*



# Grid-based Crossover



(a)



(b)

- Individu-individu diletakkan dalam suatu ***toroidal space***, dimana ujung-ujung kotak tersebut disatukan dan membentuk ruang tiga dimensi seperti bola.
- Individu hanya bisa di-crossover dengan individu2 tetangganya.
- Individu hitam hanya boleh crossover dengan 1 dari 8 individu tetangga.

100	91	92	93	94	95	96	97	98	99	100	91
10	1	2	3	4	5	6	7	8	9	10	1
20	11	12	13	14	15	16	17	18	19	20	11
30	21	22	23	24	25	26	27	28	29	30	21
40	31	32	33	34	35	36	37	38	39	40	31
50	41	42	43	44	45	46	47	48	49	50	41
60	51	52	53	54	55	56	57	58	59	60	51
70	61	62	63	64	65	66	67	68	69	70	61
80	71	72	73	74	75	76	77	78	79	80	71
90	81	82	83	84	85	86	87	88	89	90	81
100	91	92	93	94	95	96	97	98	99	100	91
10	1	2	3	4	5	6	7	8	9	10	1

# GA untuk melatih FFNN (MLP)

- Menemukan weights secara otomatis
- Menggunakan representasi biner
- Bagaimana performansi Advanced GA?

# Masalah 3-Parity

Tabel Kebenaran XOR untuk tiga masukan  $X_1$ ,  $X_2$ , dan  $X_3$ .

$X_1$	$X_2$	$X_3$	$Y$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

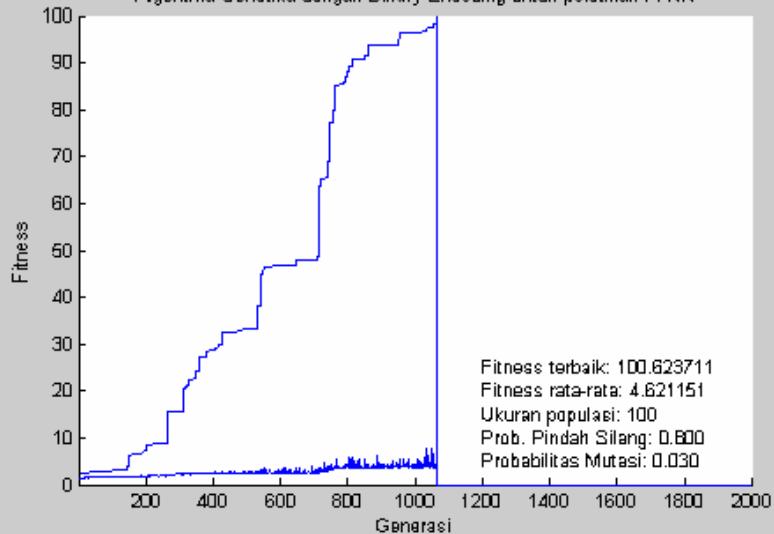
# Fungsi Fitness

$$f = 1/delta$$

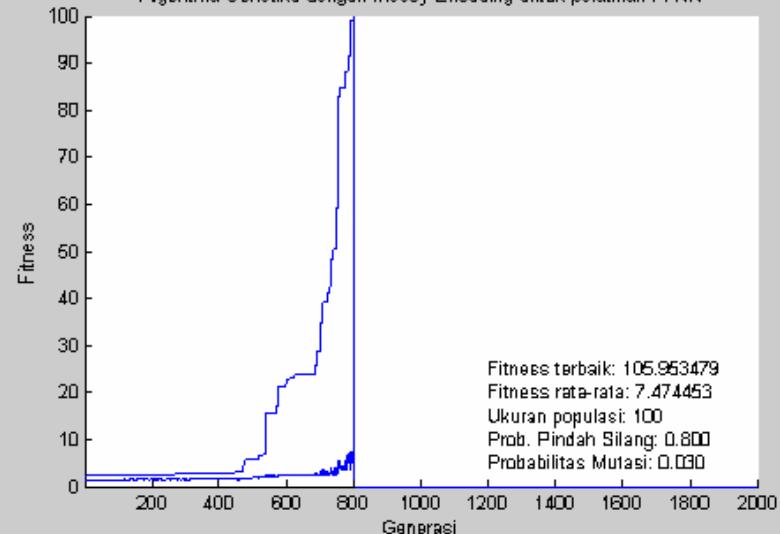
$$\text{delta} = \sqrt{\frac{1}{m} \sum_{i=1}^m (d(i) - y(i))^2}$$

- $m = 2^n = 2^3 = 8$
- $d(i)$  adalah output yang diharapkan.
- Pola masukan pertama,  $X_1 = 0$ ,  $X_2 = 0$ , dan  $X_3 = 0$ , maka  $d(1) = 0$ .
- Pola masukan ke dua,  $X_1 = 0$ ,  $X_2 = 0$ , dan  $X_3 = 1$ , maka  $d(2) = 1$ , dst.
- $y(i)$  adalah output aktual yang dihasilkan.

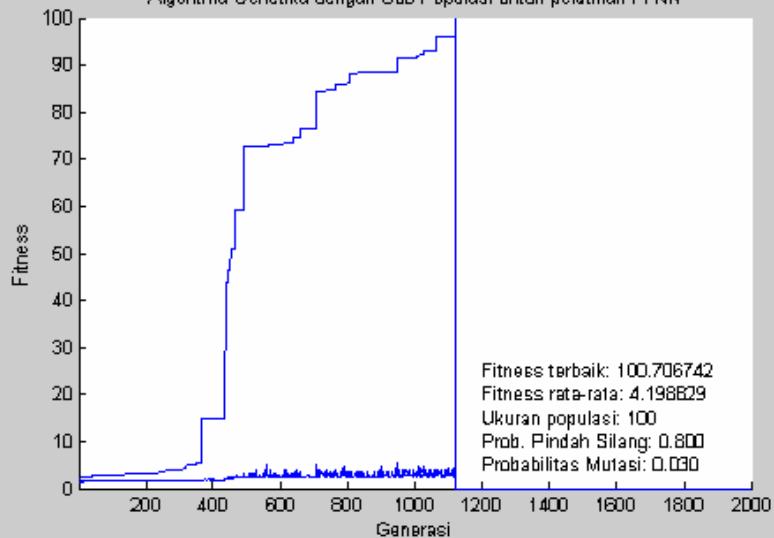
Algoritma Genetika dengan Binary Encoding untuk pelatihan FFNN



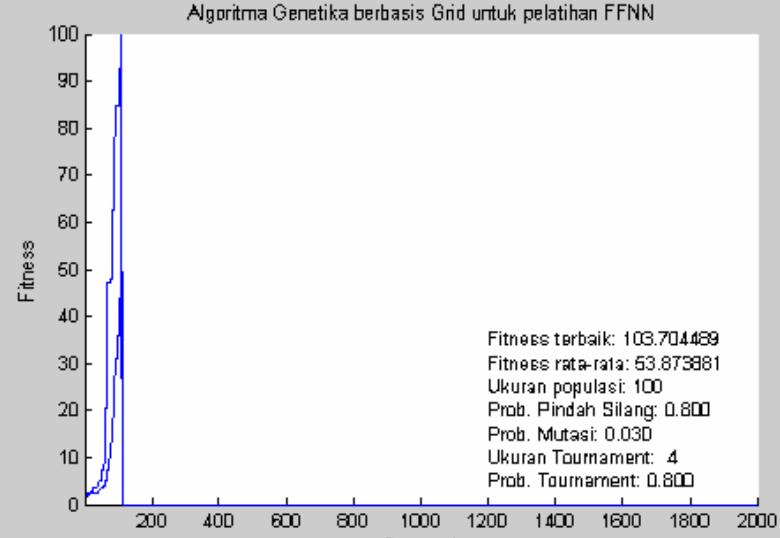
Algoritma Genetika dengan Messy Encoding untuk pelatihan FFNN



Algoritma Genetika dengan Sub Populasi untuk pelatihan FFNN



Algoritma Genetika berbasis Grid untuk pelatihan FFNN



# Grammatical Encoding

- Otak manusia merupakan suatu komputer sangat kompleks yang terdiri dari sekitar  $10^{11}$  elemen komputasi (*neurons*).
- Terdapat sekitar  $10^{14}$  sampai  $10^{15}$  koneksi antar *neurons*, atau sekitar 1000 sampai 10000 koneksi per *neuron*.
- Jika setiap koneksi dikodekan ke dalam kromosom, maka informasi yang mengisi kromosom akan sekitar **10<sup>5</sup> GB**, dimana bobot-bobot sinaptik dikodekan menggunakan hanya 1 *byte*.
- Tetapi, pada kenyataanya ukuran *genome* manusia hanya sekitar **3 GB**.
- Oleh karena itu para peneliti percaya bahwa pengkodean otak manusia bukanlah menggunakan pengkodean langsung, melainkan pengkodean **prosedur** dimana otak dibentuk.

# Grammatical Encoding

- Pada skema ini, kromosom dipandang sebagai kalimat yang diekspresikan menggunakan *grammar* (tata bahasa).
- Ketika sebuah kalimat dibaca (kromosom didekodekan), maka individu dibangkitkan menggunakan *grammar* tsb.
- Contoh: skema Kitano yang digunakan untuk mengkodekan ANN yang berisi maksimum 8 *neurons*

# Skema Kitano

S	ACBA	Aadfb	Bbefd	Dfanp	Bahjm	Ckhgf	...
---	------	-------	-------	-------	-------	-------	-----

$$S \rightarrow \begin{pmatrix} A & C \\ B & A \end{pmatrix}$$

$$A \rightarrow \begin{pmatrix} a & d \\ f & b \end{pmatrix} \quad B \rightarrow \begin{pmatrix} b & e \\ f & d \end{pmatrix}$$

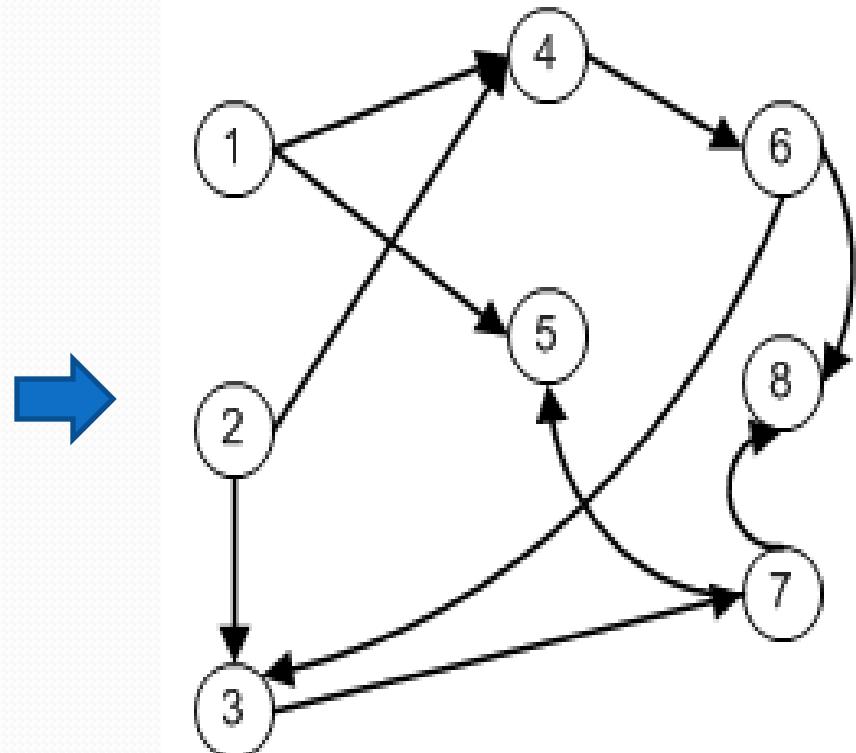
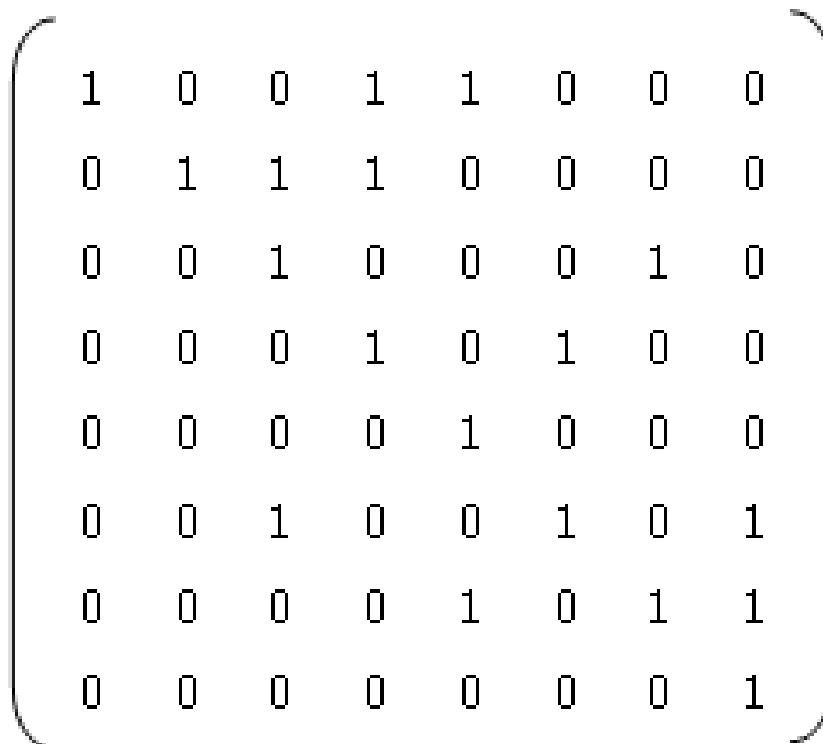
$$a \rightarrow \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, b \rightarrow \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, c \rightarrow \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \dots p \rightarrow \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

# Skema Kitano

S	ACBA	Aadfb	Bbefd	Dfanp	Bahjm	Ckhgf	...
---	------	-------	-------	-------	-------	-------	-----

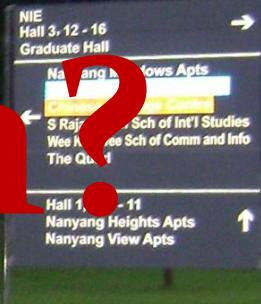
$$\begin{pmatrix} A & C \\ B & A \end{pmatrix} \xrightarrow{\quad} \begin{pmatrix} a & d & k & h \\ f & b & g & f \\ b & e & a & d \\ f & d & f & b \end{pmatrix} \xrightarrow{\quad} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

# Skema Kitano



Demo Program

# Question?



# Evolution Learning?

- Baldwin Effect (1896)
- A new Factor of Evolution
- Learning → mempercepat evolusi?
  - Rusa (belajar) mempercepat larinya
  - Harimau (belajar) mempertajam taringnya
  - Dsb.

# Evolution → Learning?

# Learning → Evolution?



# Hinton & Nowlan's model (1987)

- Kromosom: 20 gen biner
  - 5 gen bernilai 0
  - 5 gen bernilai 1
  - 10 gen bernilai ‘?’ (tidak diketahui 1 atau 0)
- Misalkan
  - Kromosom terbaik: **11111111110000000000**
  - Populasi: **1000** kromosom
  - Setiap kromosom **menebak** gen ‘?’ → **1000** trials
  - **Fitness** =  $1 + ((19^*n)/M)$ 
    - $M$  = jumlah trial maksimum
    - $n$  = sisa trial

# Contoh Populasi

- **Kromosom 1:** 111???001?0100???????
- **Kromosom 2:** 11111?????00000??????
- **Kromosom 3:** 1111100000????????????
- **Kromosom 4:** ????.?0001111100??????
- **Kromosom 5:** 00????00111110??????
- ...
- **Kromosom 1000:** ?000??????0011???111
- **Kromosom target:** **11111111110000000000**

# Operator Evolusi

- **Seleksi ortu** : Roulette Wheel
- **Rekombinasi** : satu titik
- **Mutasi Biner** : terhadap gen 0 atau 1
- **Seleksi survivor:** Generational replacement

# Contoh Trial

**Kromosom target** 11111111110000000000

**Kromosom** 111 ???001 ?0100???????

- **Trial 1** 11100000100100000011
- **Trial 2** 11111000110100001111
- **Trial 3** 11101000100100111100
- ...
- **Trial 1000** 11111100100100000000

# Rekombinasi

## Satu titik potong

Parent 1

110	????????????		0000000
-----	--------------	--	---------

Parent 2

1111111111	????		0???????
------------	------	--	----------

Offspring 1

110	????????????		0???????
-----	--------------	--	----------

Offspring 2

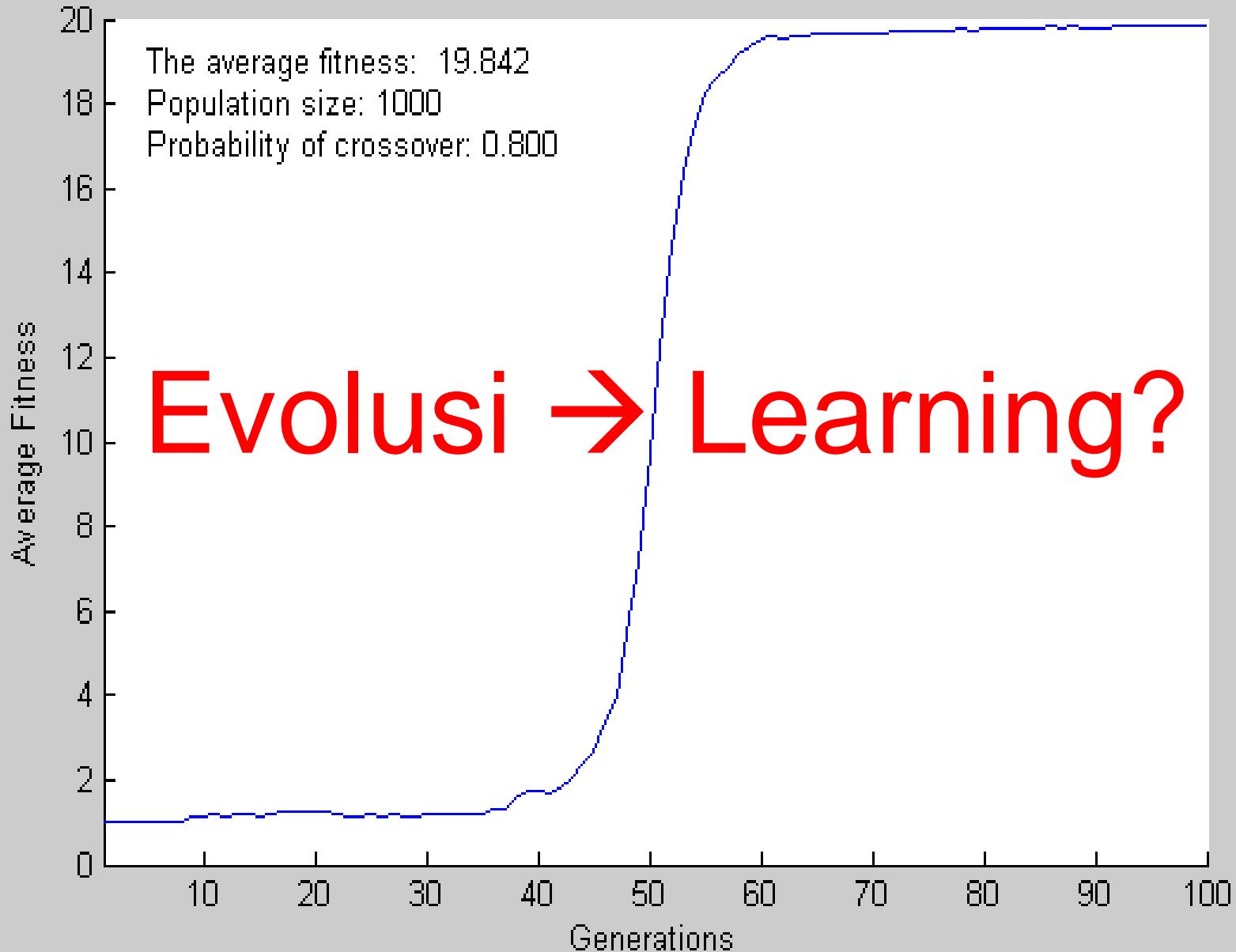
1111111111	????		0000000
------------	------	--	---------

# Mutasi Biner

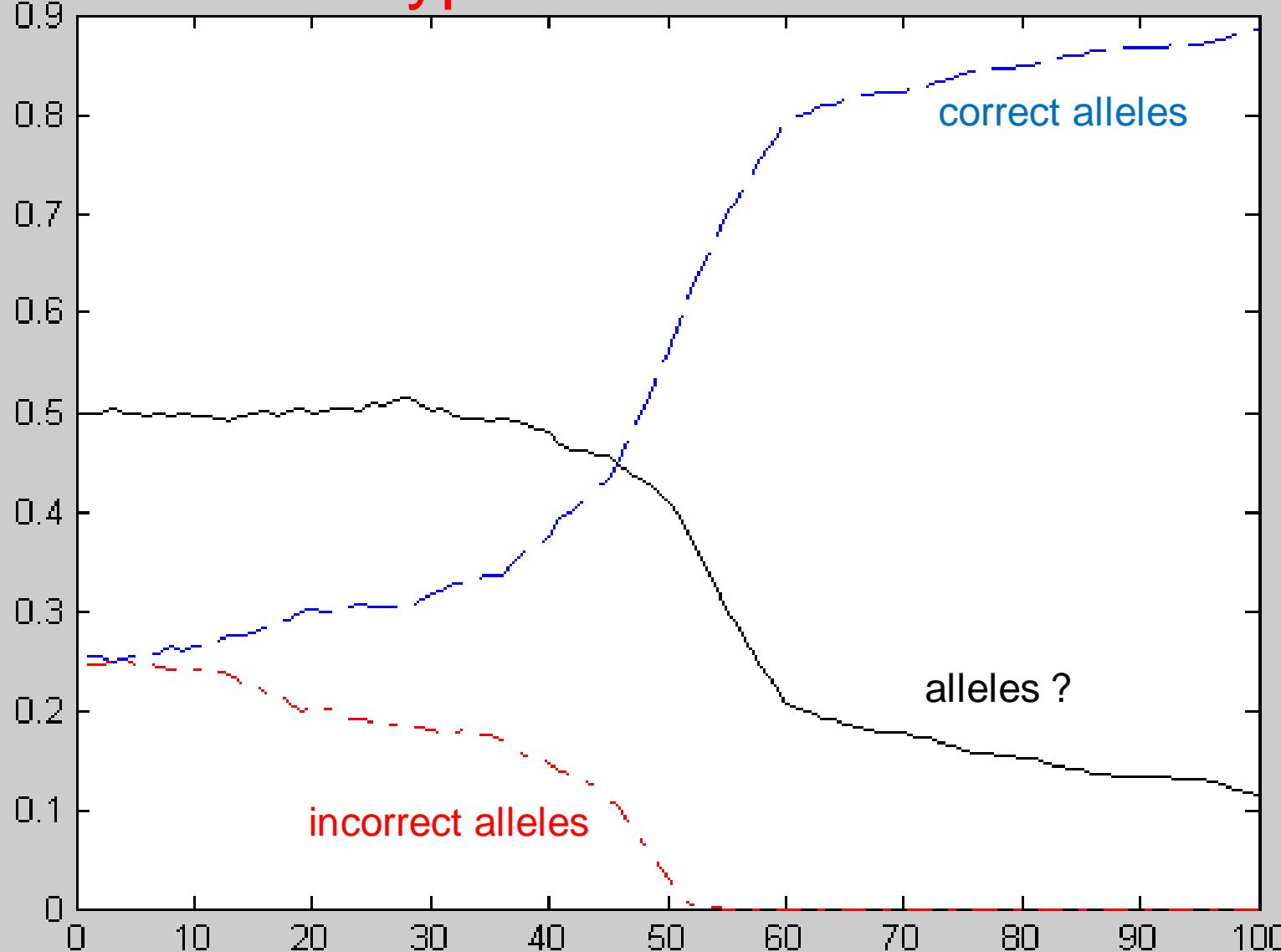
- Mutasi hanya untuk gen 0 atau 1 (tidak utk gen ?)
- 1 dimutasi menjadi 0
- 0 dimutasi menjadi 1

Kromosom                    111111111????1000000

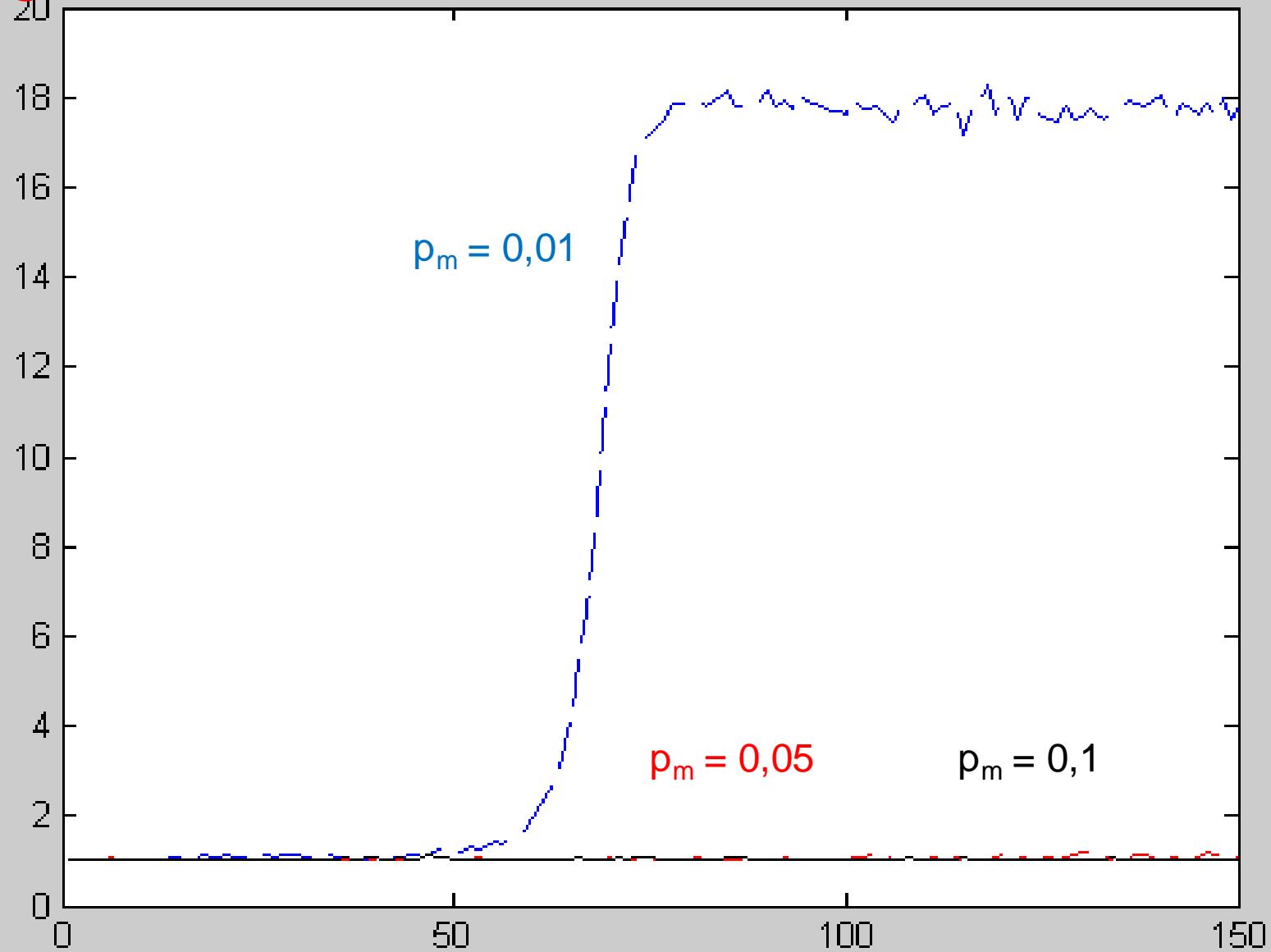
Hasil mutasi                111111111????0000000



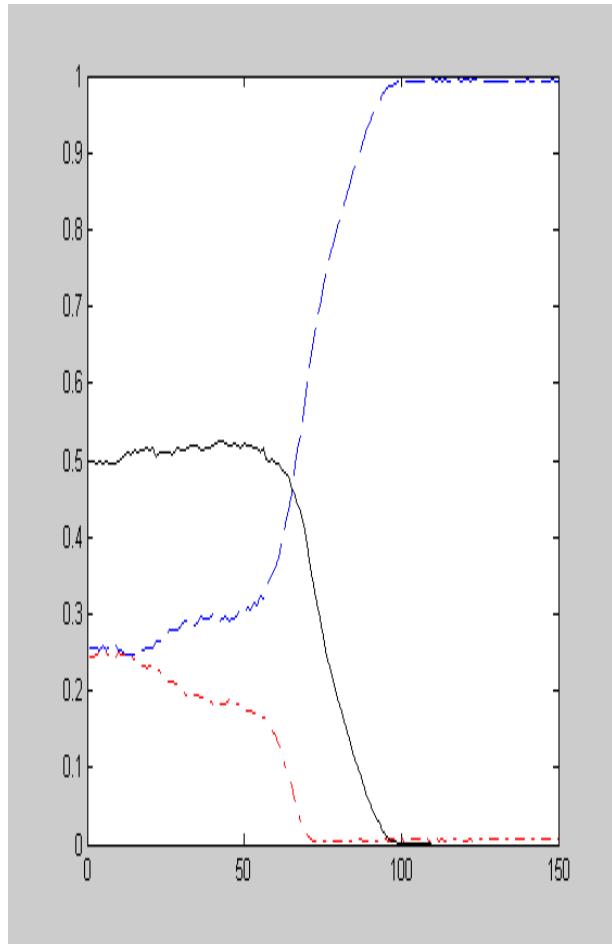
# Dinamika Genotype



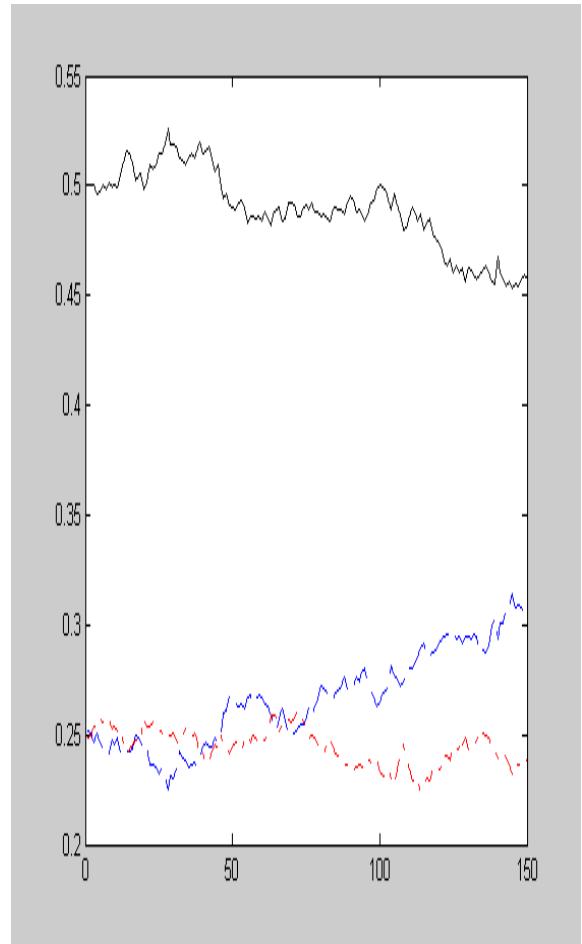
# Pengaruh Mutasi



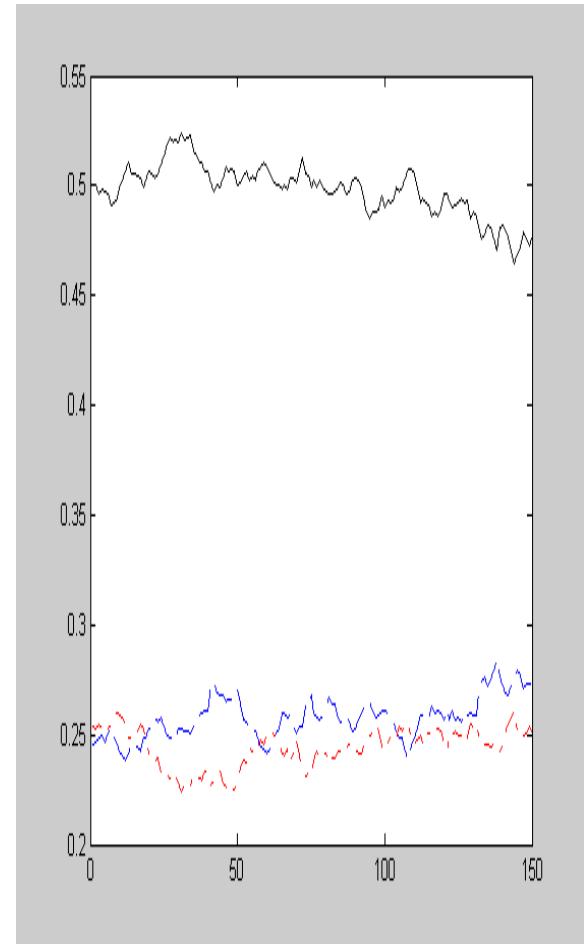
$p_m = 0,01$



$p_m = 0,05$



$p_m = 0,1$

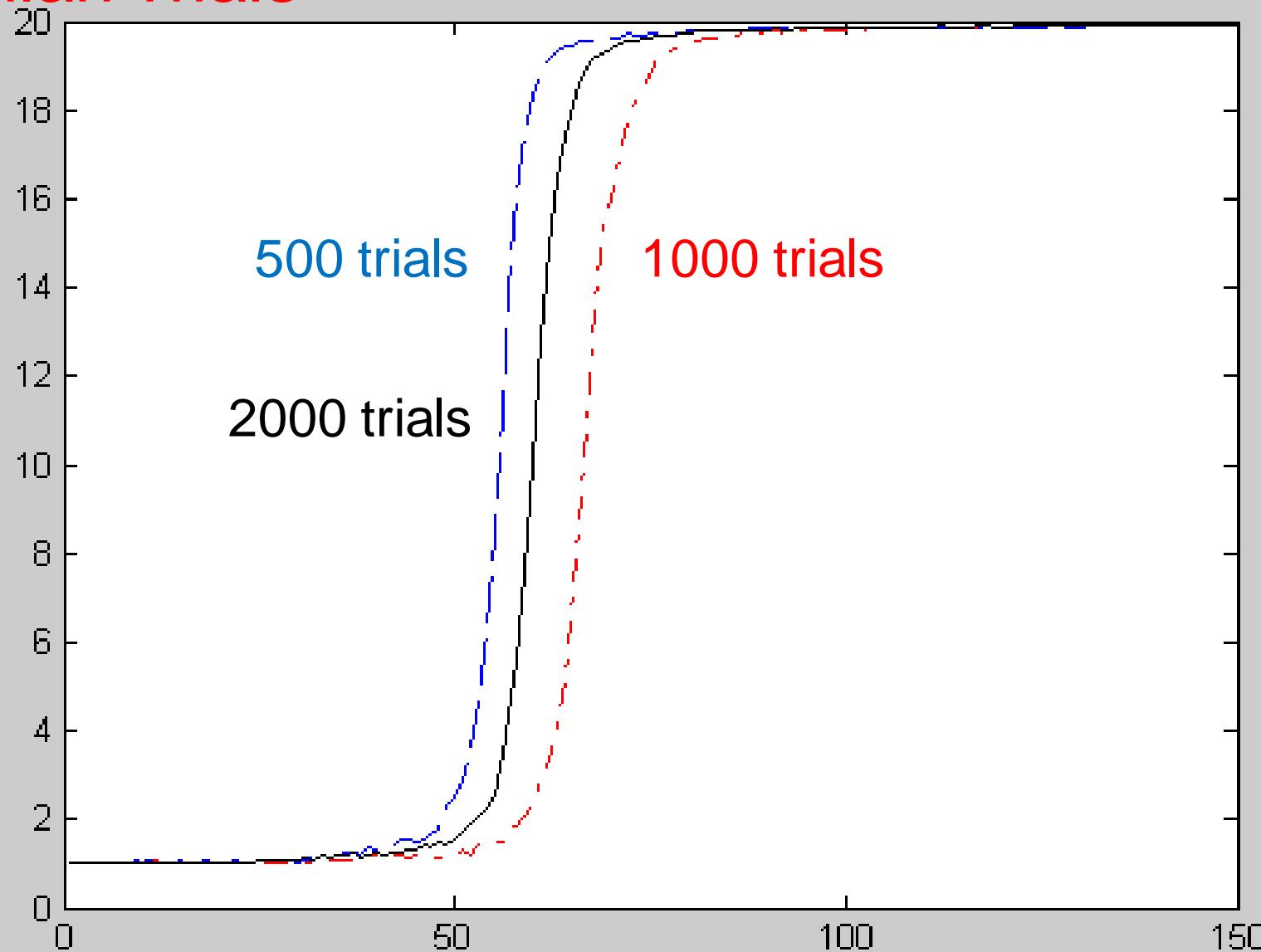


correct alleles

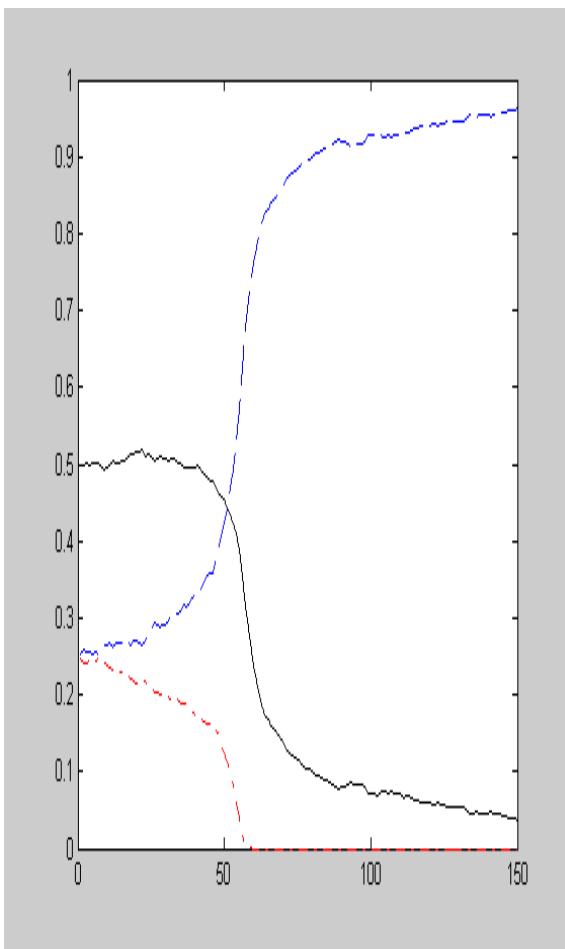
incorrect alleles

alleles ?

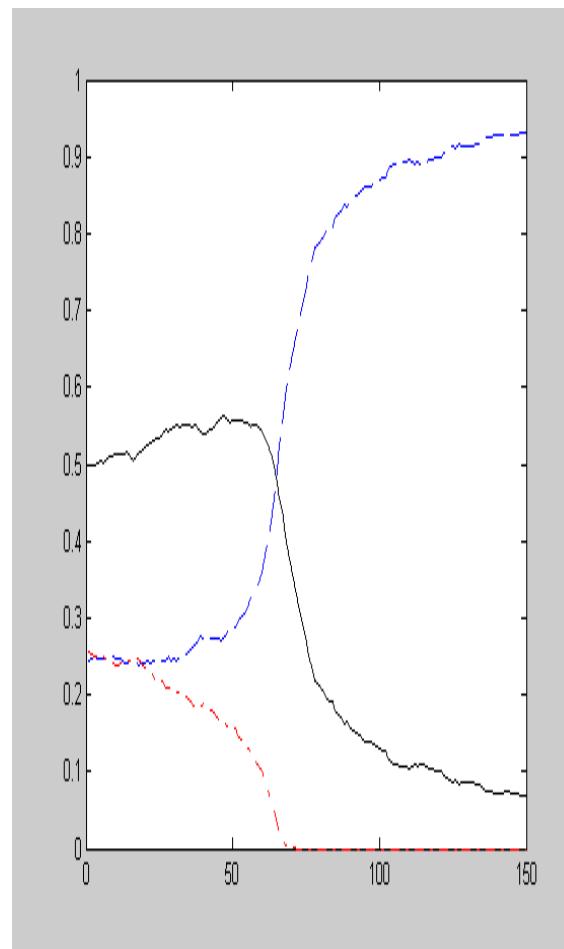
# Jumlah Trials



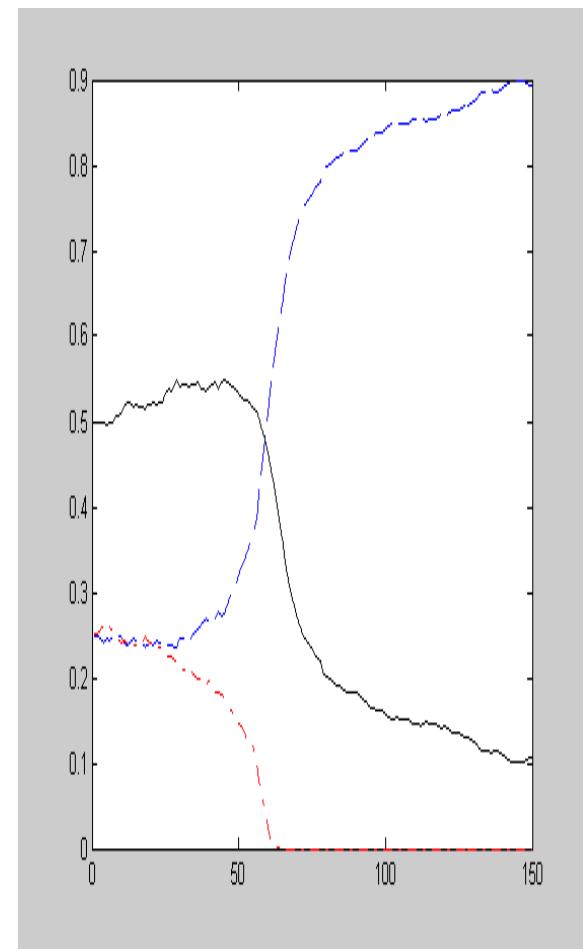
500 trials



1000 trials



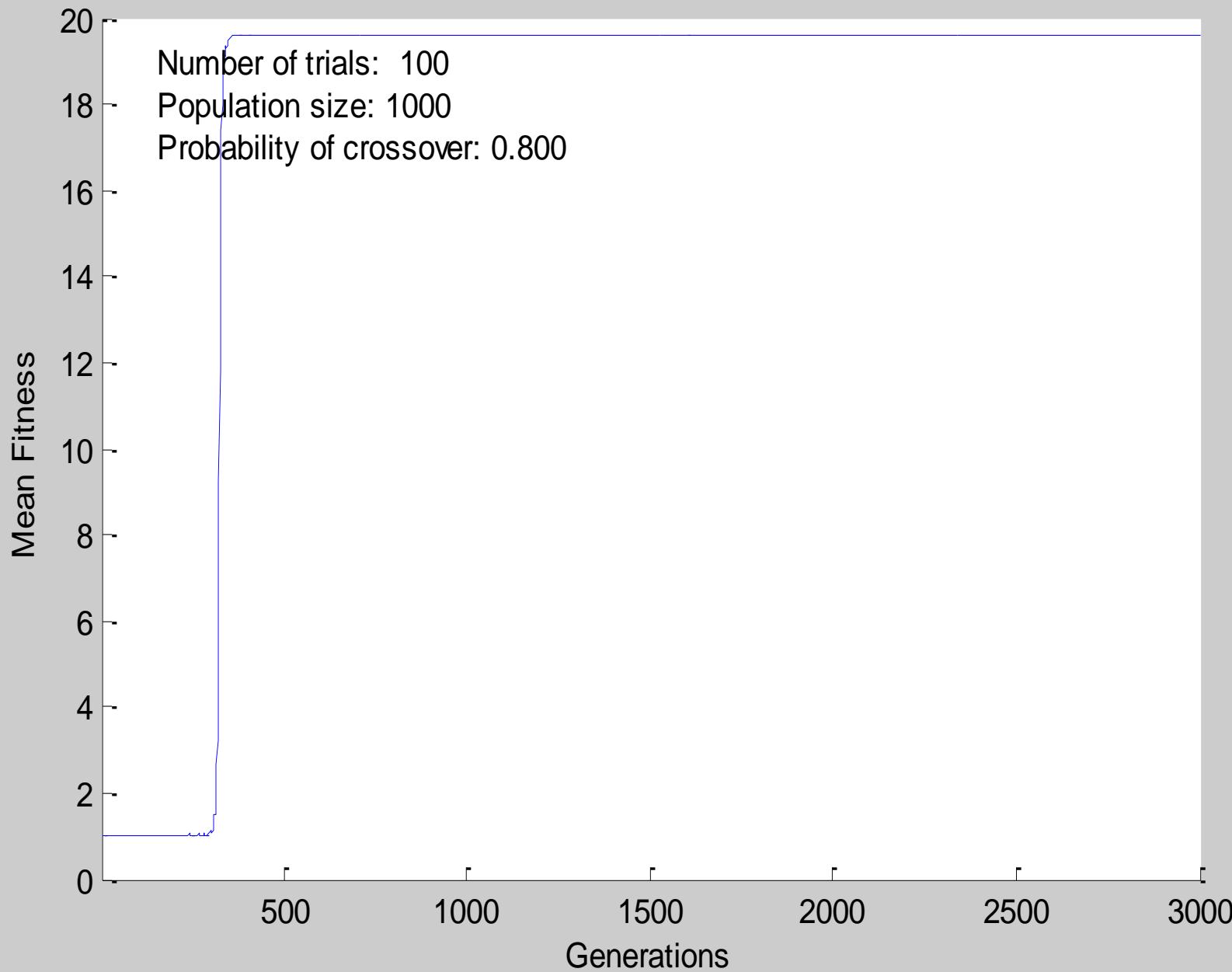
2000 trials

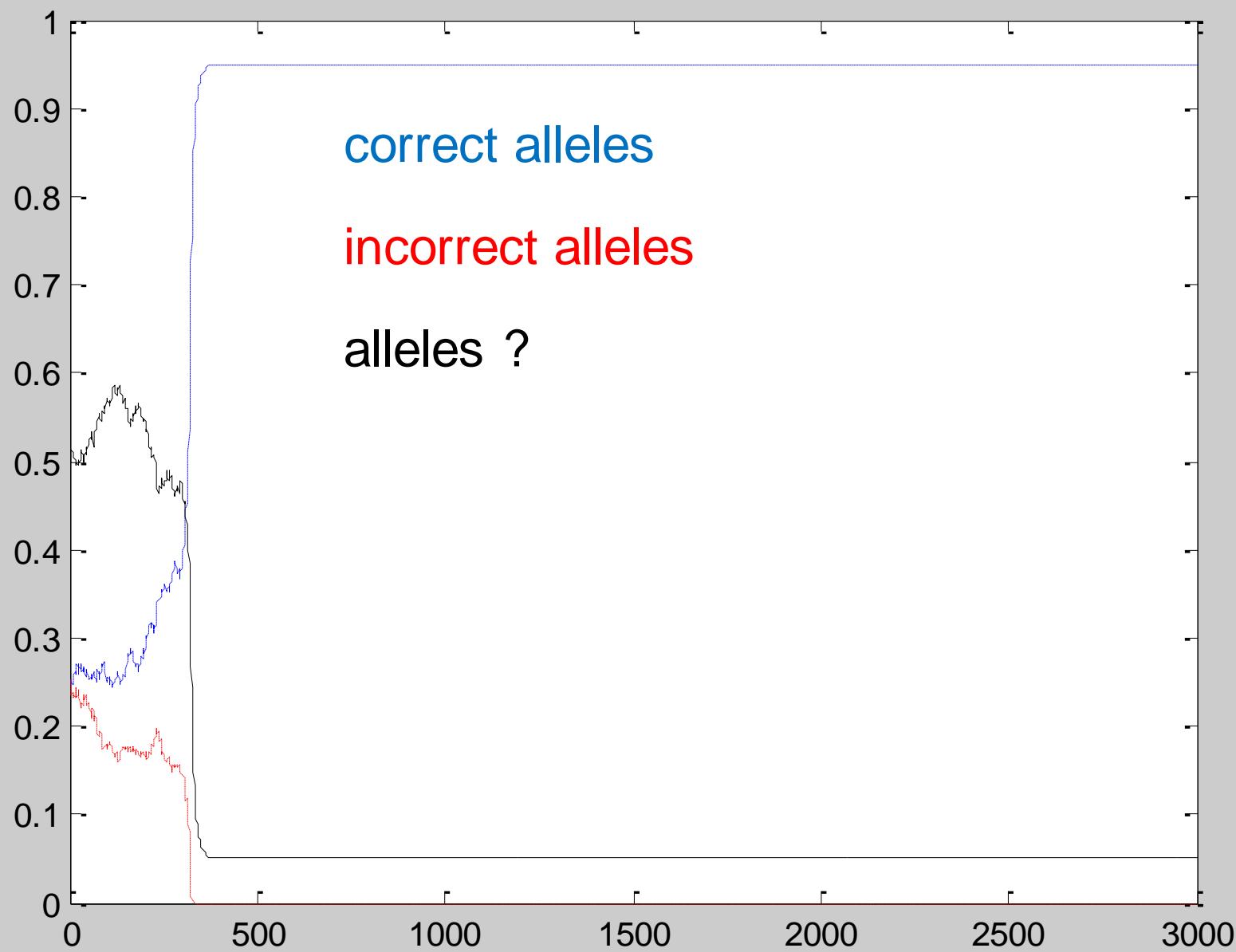


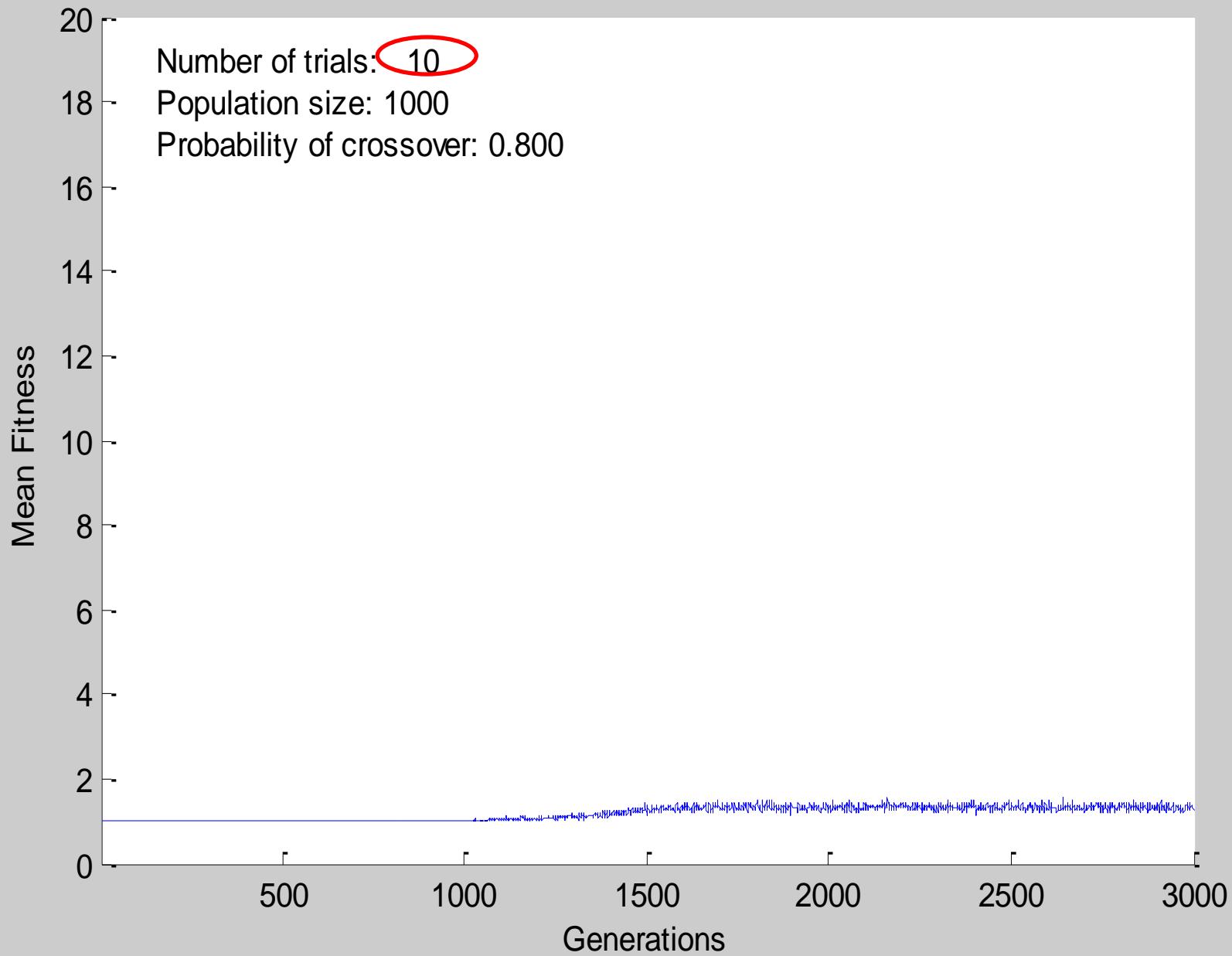
correct alleles

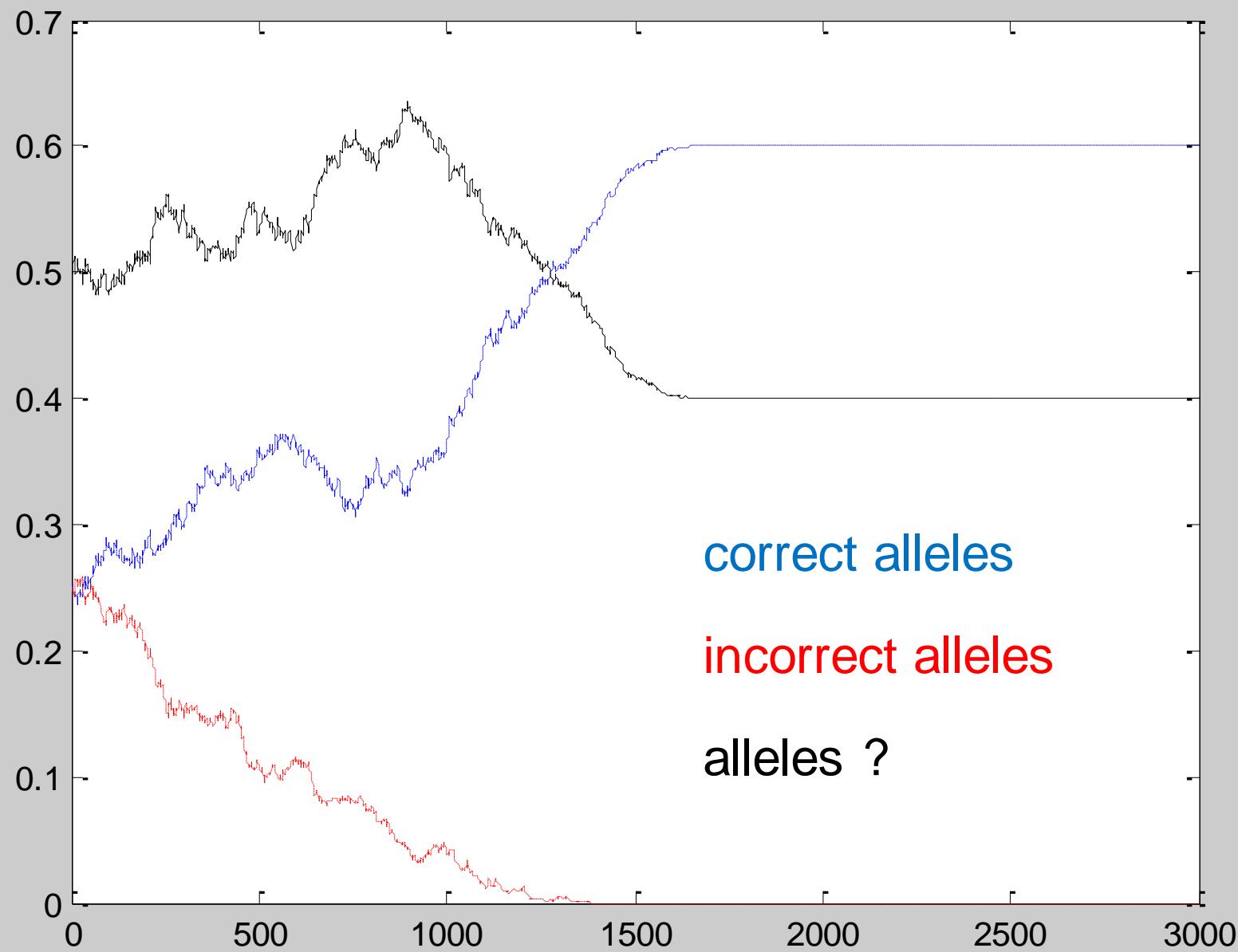
incorrect alleles

alleles ?

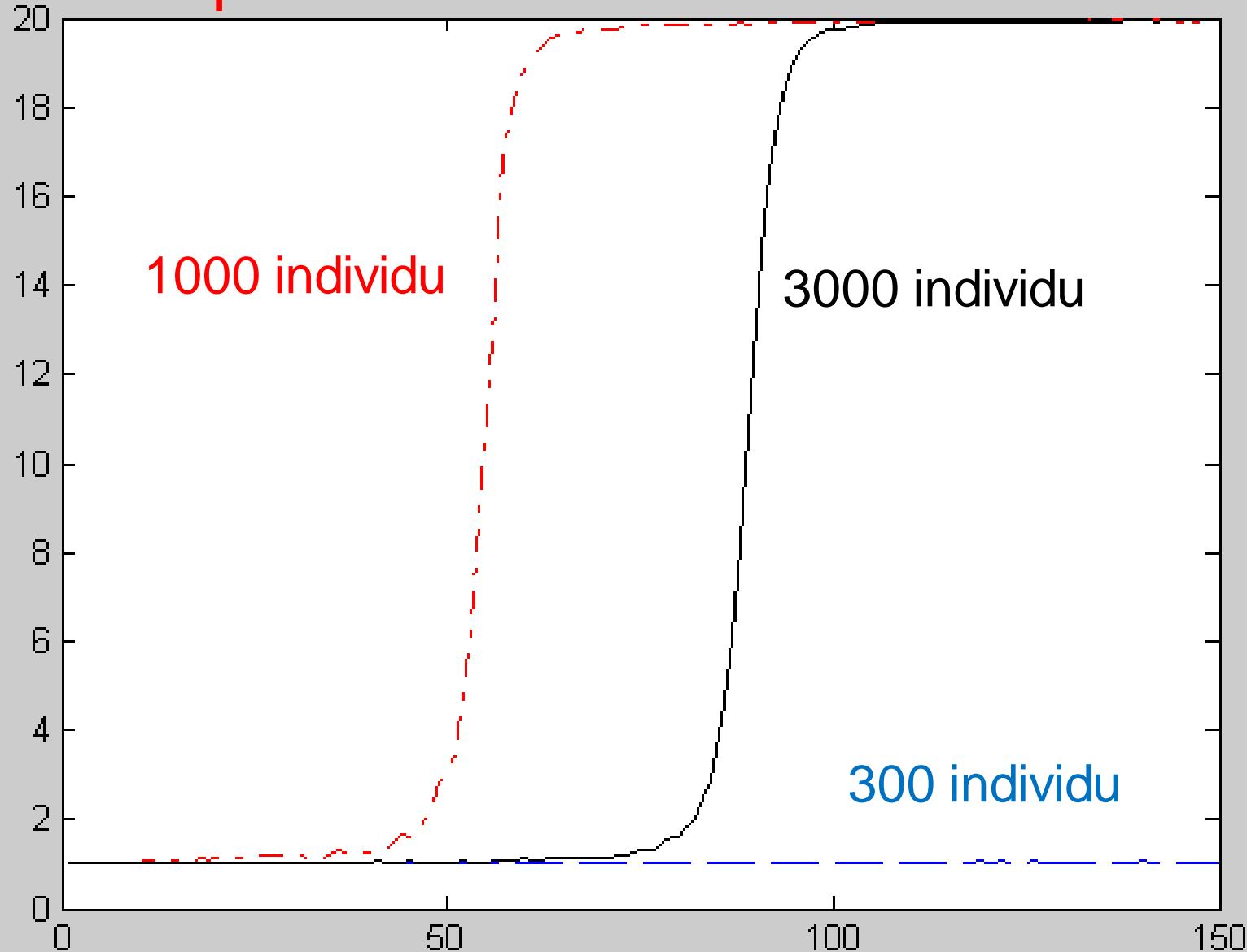




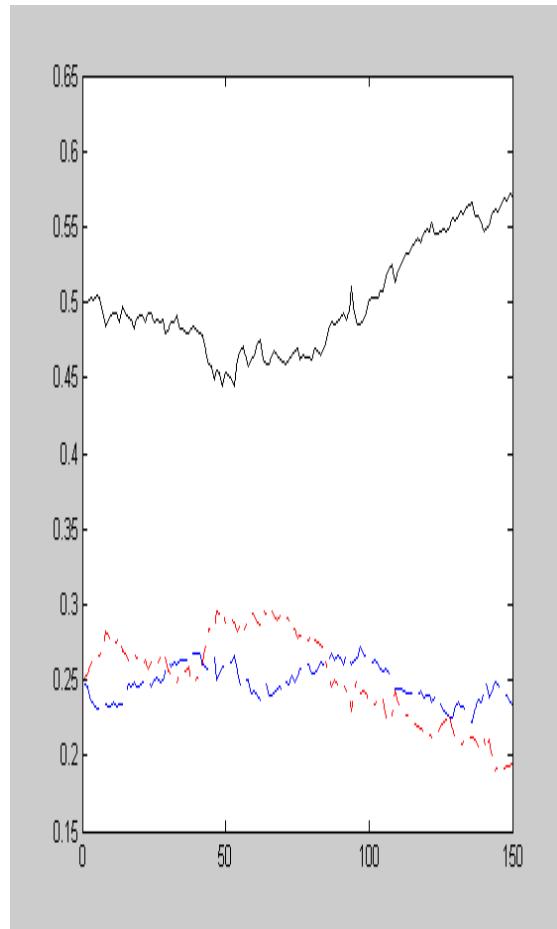




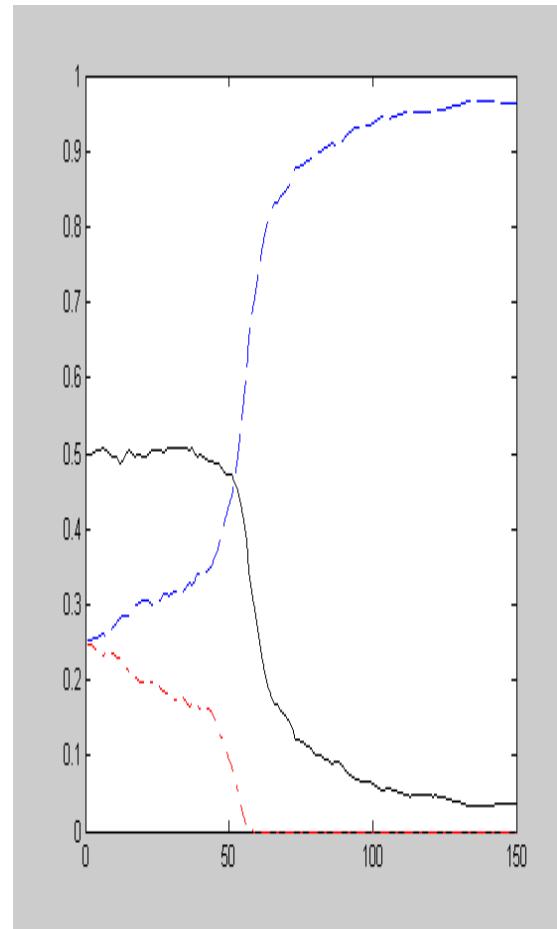
# Ukuran Populasi



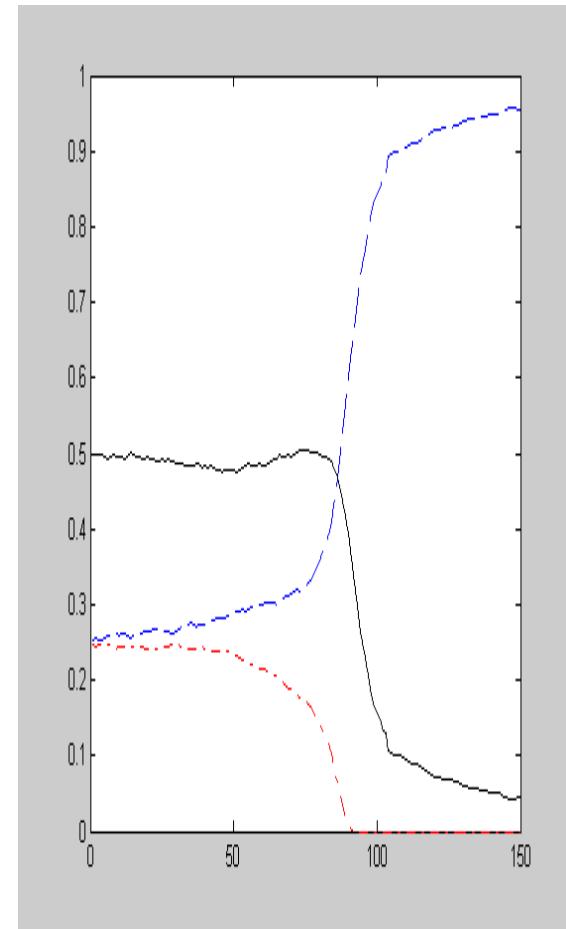
300 individu



1000 individu



3000 individu



correct alleles

incorrect alleles

alleles ?

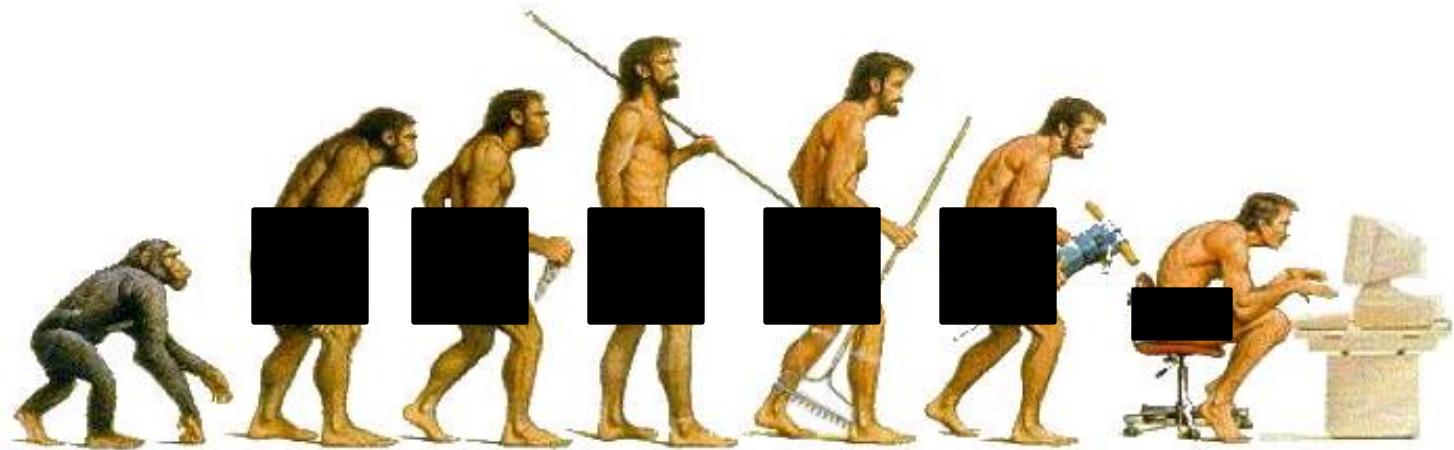
# Evolution → Learning?

# Learning → Evolution?



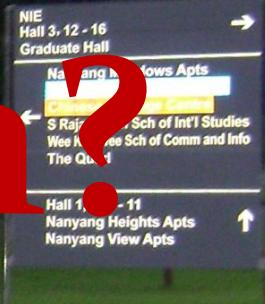
Berlatih terus bisa mempercepat evolusi?

# Evolution



Makin Pintar?

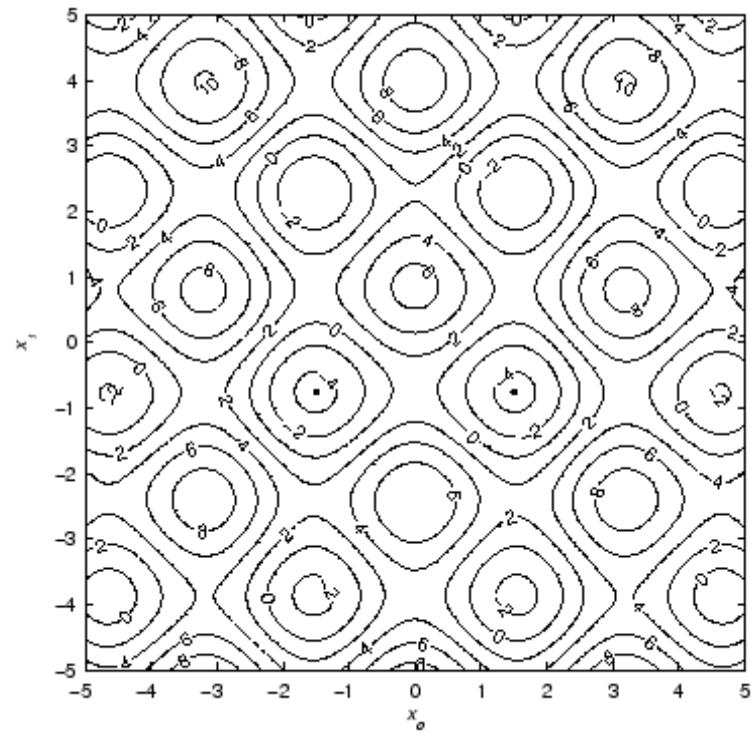
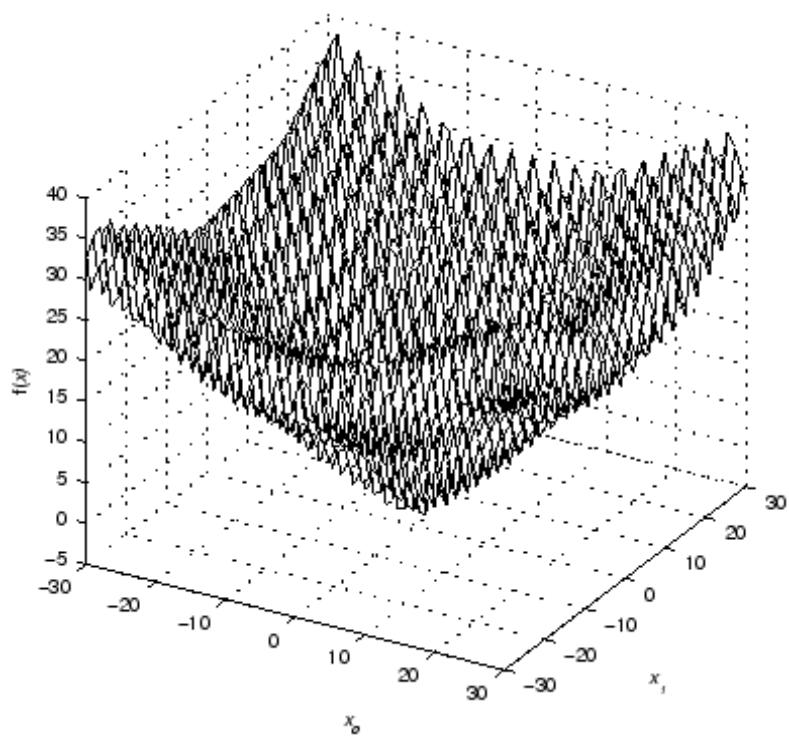
# Question?



# Studi Kasus: Kromosom & Fitness

1. Optimasi Fungsi Kompleks
2. *Graph Bisection*
3. *8-queens*
4. Pemotongan bahan
5. Peramalan Data *Time Series*
6. Penjadwalan kuliah
7. **Your problems**

$$f(\vec{x}) = \sum_{i=0}^{D-1} \left( e^{-0.2} \sqrt{x_{i-1}^2 + x_i^2} + 3(\cos(2x_{i-1}) + \sin(2x_i)) \right)$$



- Untuk presisi  $10^{-9} \rightarrow$  Berapa bit?
- Bisa menggunakan kromosom Integer atau Real?

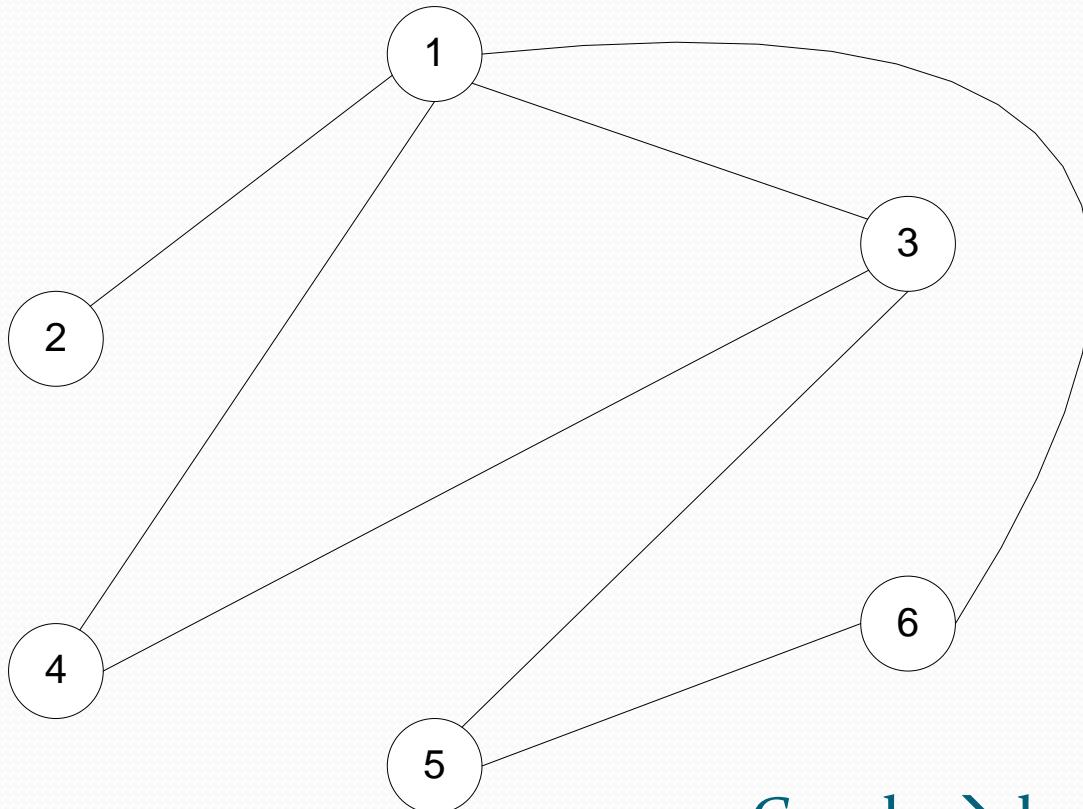
# Fungsi *fitness*

$$f = \frac{1}{(h + a)}$$

$h$  = nilai fungsi yang diminimasi

$a$  = bilangan kecil untuk menghindari pembagian dengan nol

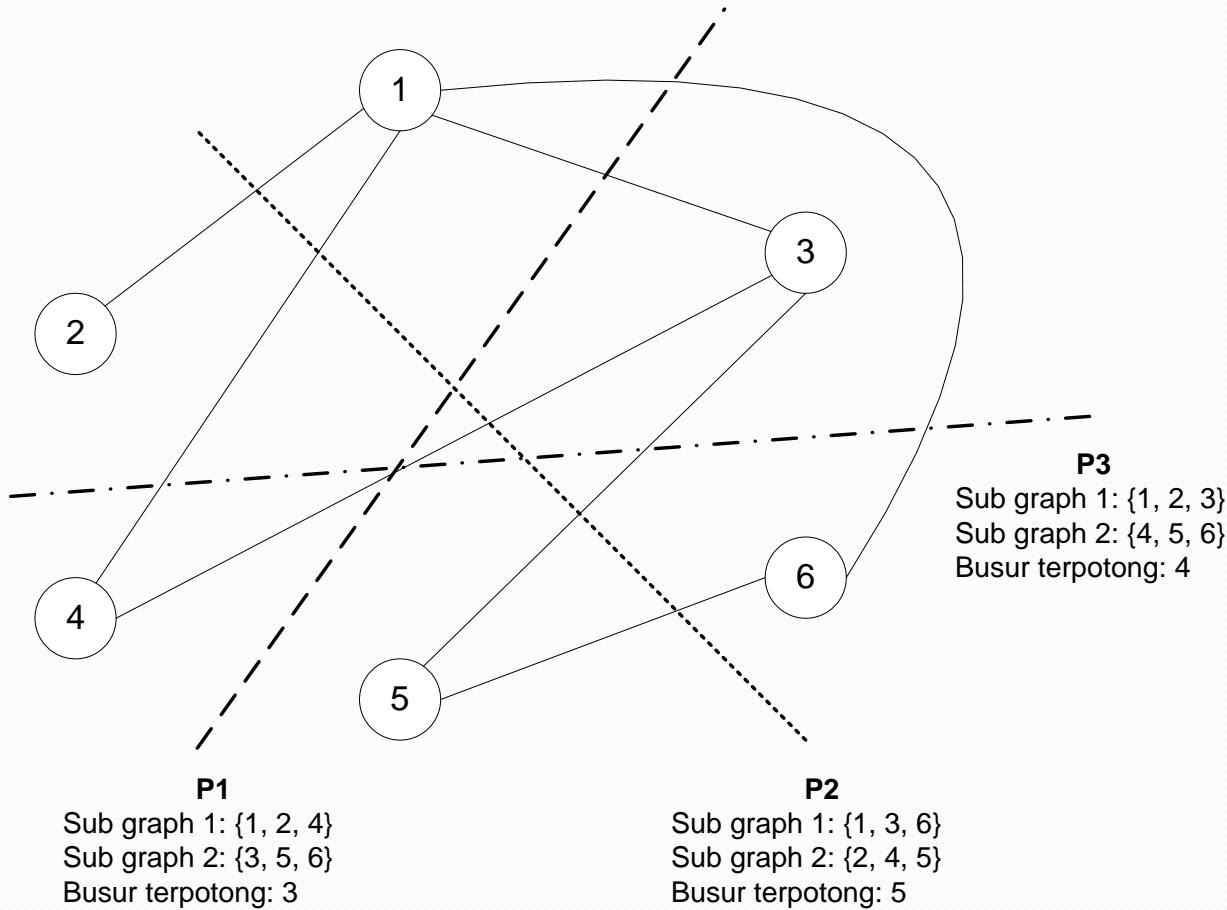
# *Graph bisection*

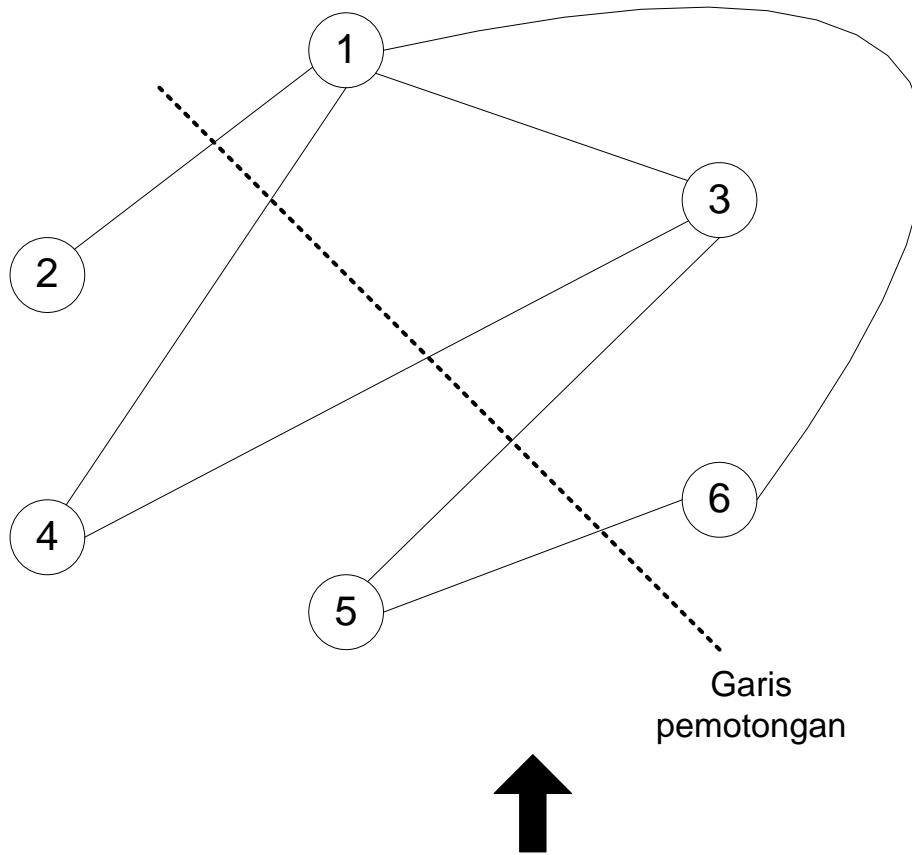


- VLSI design
- Task Scheduling
- Parallel Scientific Computing

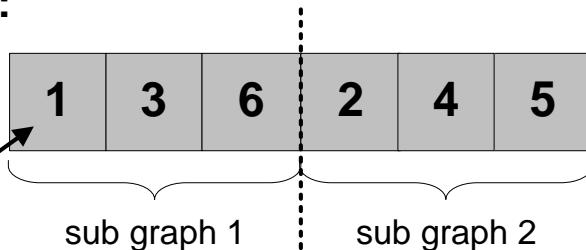
- Graph → dua sub graph sama besar?
- Minimasi busur terpotong

# *Graph bisection*





**Kromosom:**



Nilai gen  
menyatakan  
nomor *node*

Permutasi  
Biner?  
Integer?  
Real?

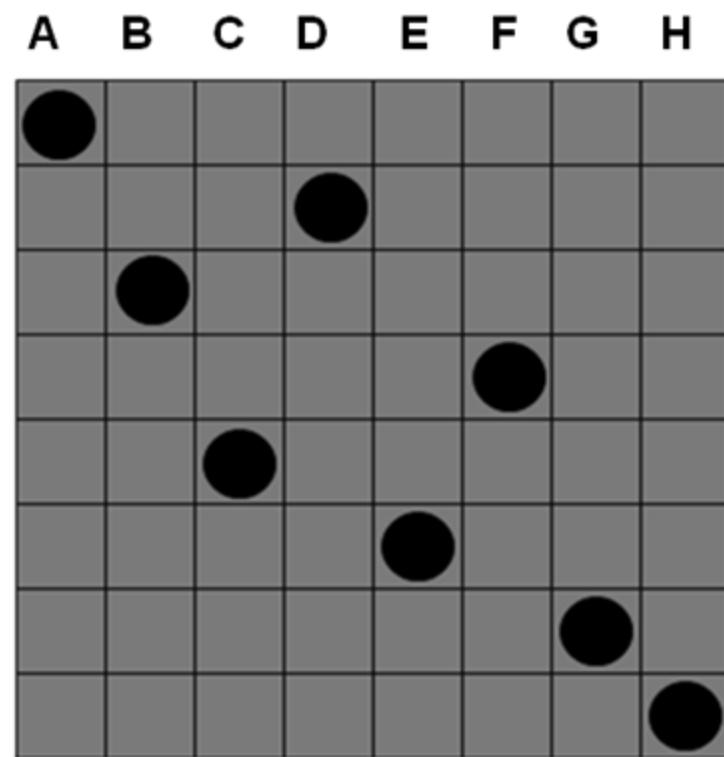
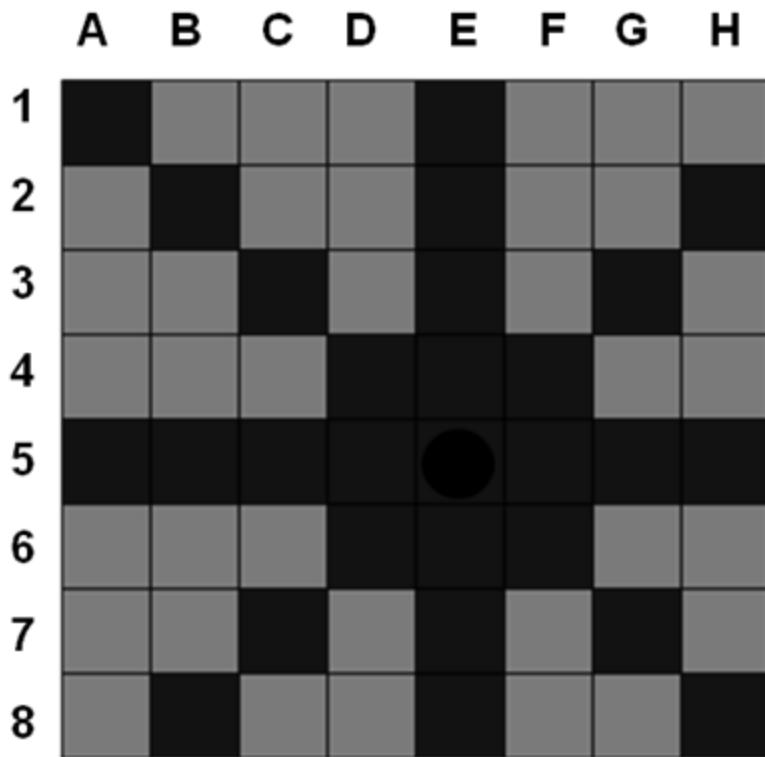
# Fungsi *fitness*

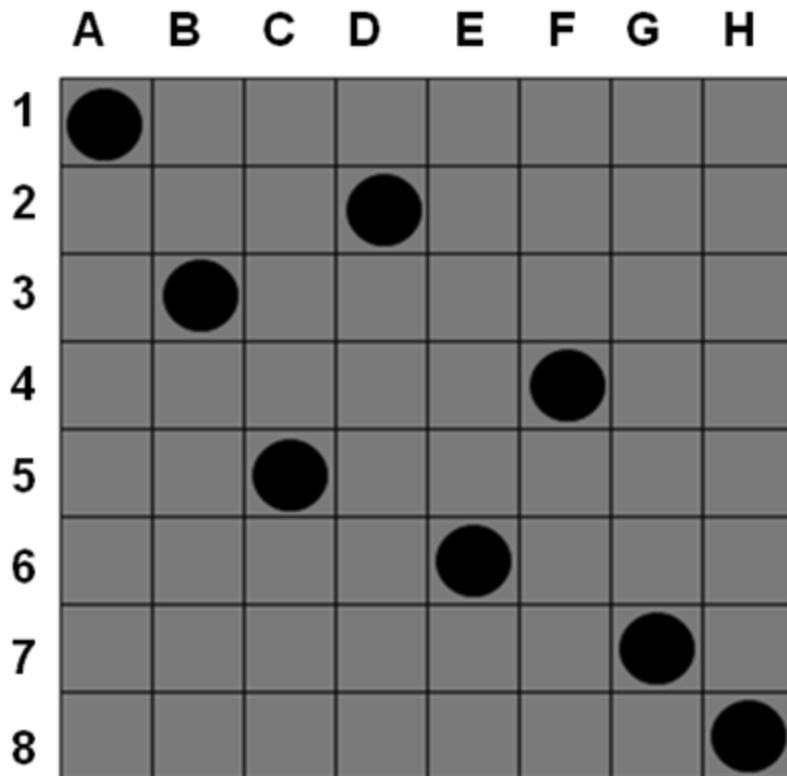
$$f = \frac{1}{(B + a)}$$

$B$  = Jumlah busur yang terpotong

$a$  = bilangan kecil untuk menghindari pembagian dengan nol

# Masalah 8-queens





Kromosom:



1	3	5	2	6	4	7	8
A	B	C	D	E	F	G	H

Nilai gen  
menyatakan  
posisi baris

Permutasi  
Biner?  
Integer?  
Real?

# Fungsi *Fitness*

$$f = \frac{1}{(Q + a)}$$

$Q \rightarrow$  adalah jumlah queen yang saling mengancam  
 $a \rightarrow$  bilangan yang diangap sangat kecil

# Pemotongan bahan

11 cm

Awas !

Kertas ini mahal

15 cm

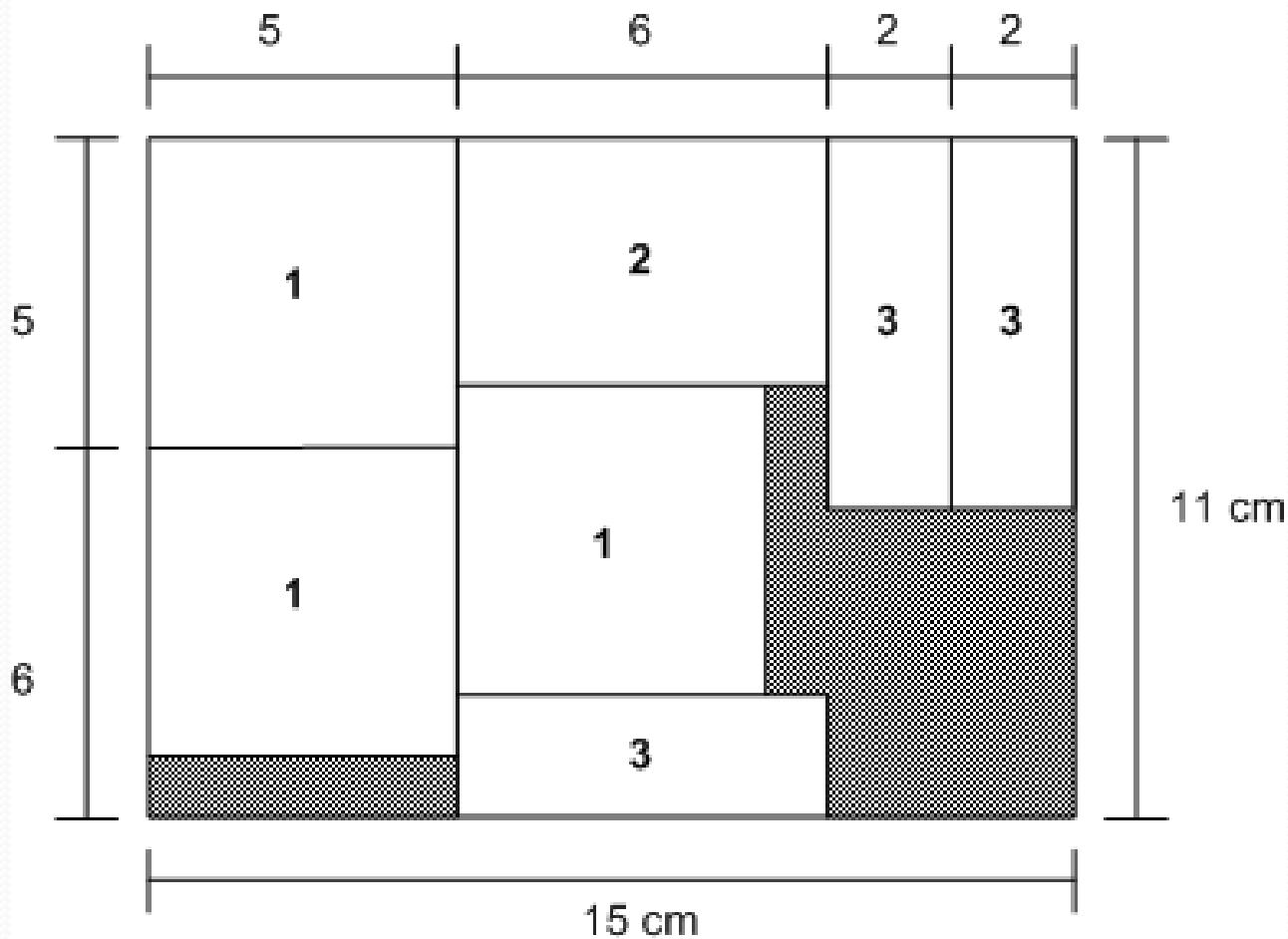
## Order potongan

No.	Ukuran (cm <sup>2</sup> )	Jml
1	5 x 5	3
2	4 x 6	2
3	2 x 6	3

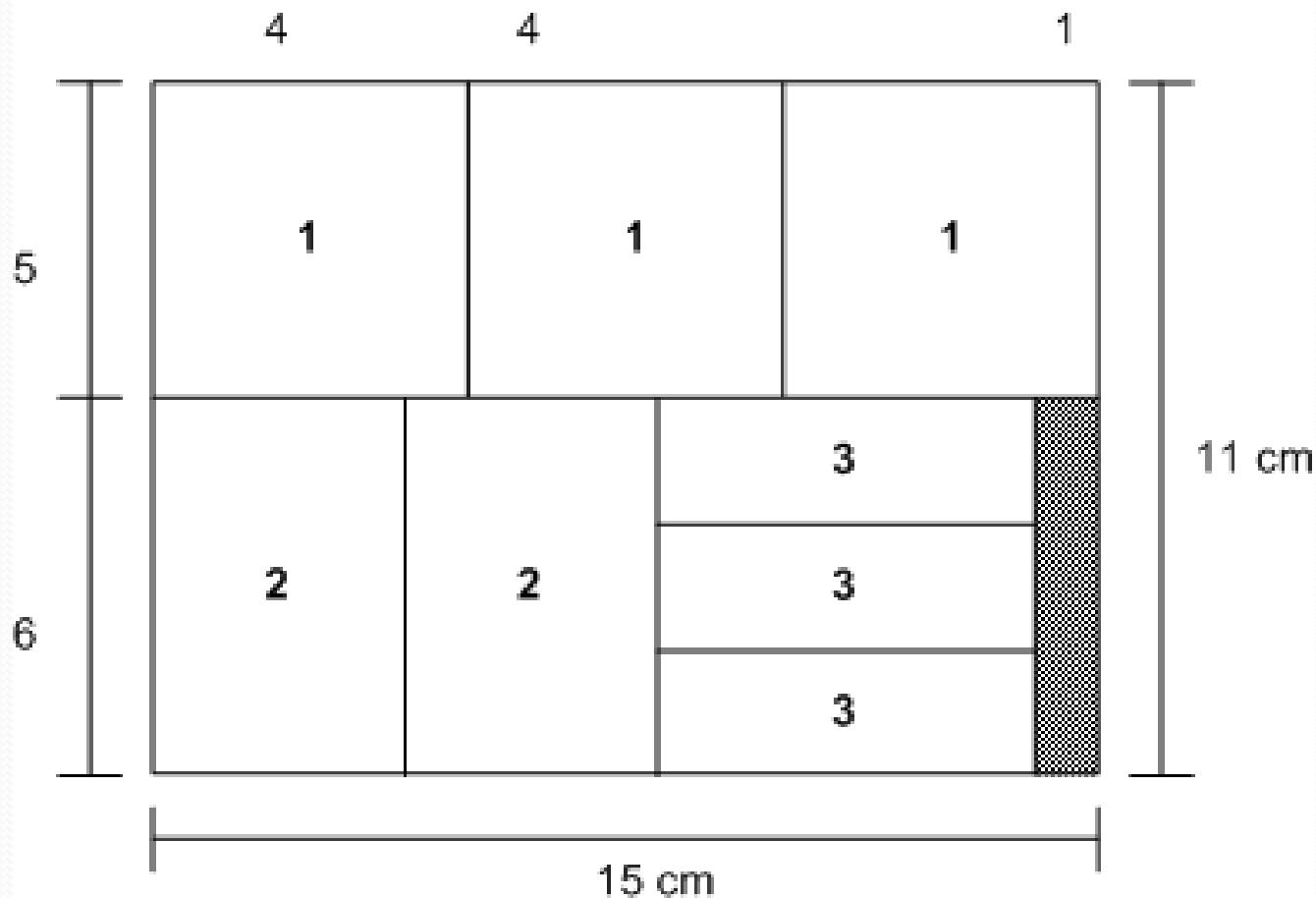
Pola potongan yang memaksimasi order dan meminimasi sisa bahan?

Kasus yang mirip:  
Pemasangan Iklan

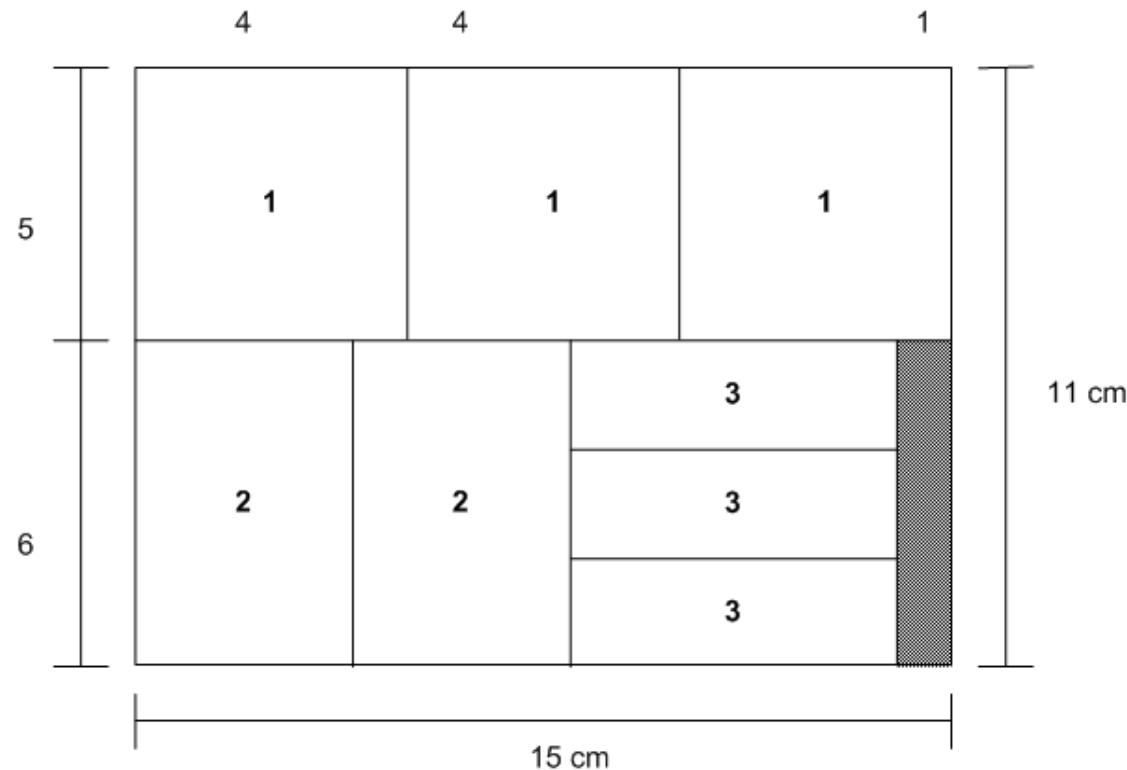
# Pola pemotongan 1



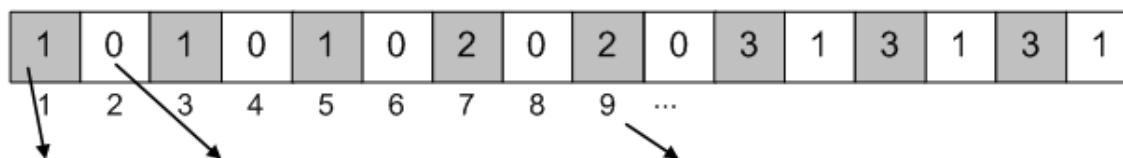
# Pola pemotongan 2



## Pola pemotongan:



## Kromosom:



Nilai gen pada indeks ganjil (warna abu-abu) menyatakan **nomor order**

Nilai gen pada indeks genap (warna putih) menyatakan **orientasi order** untuk nomor order pada gen di sebelah kirinya.

Indeks ganjil menyatakan **urutan pemotongan** dari kiri atas ke kanan bawah. Jadi, urutan pemotongan adalah nomor order pada indeks 1, 3, 5, 7, 9, ...

**Kromosom lain?**

# Fungsi *fitness*

$$f = \frac{P}{(S + a)}$$

$P$  = Permintaan (order) yang terpenuhi

$S$  = Sisa bahan

$a$  = Bilangan kecil untuk menghindari pembagian dengan nol

# Pemotongan Bahan

- **Urutan posisi**
  - Dari kiri atas ke kanan bawah?
  - Dari atas kiri ke bawah kanan?
  - Acak → kromosom perlu posisi baris kolom
- **Bentuk potongan**
  - Segitiga
  - Lingkaran
  - Bintang
- **Bahan**
  - Kain
  - Kulit
  - Kaca

# Peramalan Data *Time Series*

<b>Tanggal</b>	<b>Penjualan (miliar rupiah)</b>
14 Dec 2007	99,9573
13 Dec 2007	99,8459
12 Dec 2007	98,8708
11 Dec 2007	98,7480
10 Dec 2007	98,3897
09 Dec 2007	97,6780
08 Dec 2007	97,3797
...	...
...	...
...	...
03 Jan 2006	90,7597
02 Jan 2006	90,5770
01 Jan 2006	89,3897

# Model Matematis

$$z = a_0 + a_1 y_1 + a_2 y_2 + \dots + a_k y_k$$

# Kromosom

$a_0$	$a_1$	...	$a_k$
0,0371	0,5133	...	0,4911

Real  
Biner?  
Integer?  
Permutasi?

# Fungsi *Fitness*

$$f = \frac{1}{(E + a)}$$

$E \rightarrow$  Tingkat Error (MAPE) untuk semua data histori.  
 $a \rightarrow$  bilangan yang diangap sangat kecil

# Penjadwalan Kuliah

No	Kode MK	Pertemuan ke-	Kode Kelas	Jumlah mhs	Kode Dosen	Status dosen	JFA
1	CS3143	1	IF-33-01	40	SBK	Dalam	Lektor
2	CS2314	1	IF-34-01	35	TWG	LB	Guru Besar
3	CS2314	2	IF-34-01	35	TWG	LB	Guru Besar
...	...	...	...	...	...	...	...
400	CS4923	2	IF-32-03	17	DSS	Dalam	Lektor

# Kromosom

Ruang 1

	SN	SL	RB	KM	JM	SB
07-09	50		7	112	187	
09-11	32	1	101		400	
11-13		3		9		
13-15				6		
15-17	43	21	29	332		15

Gen ke-1

Ruang 36

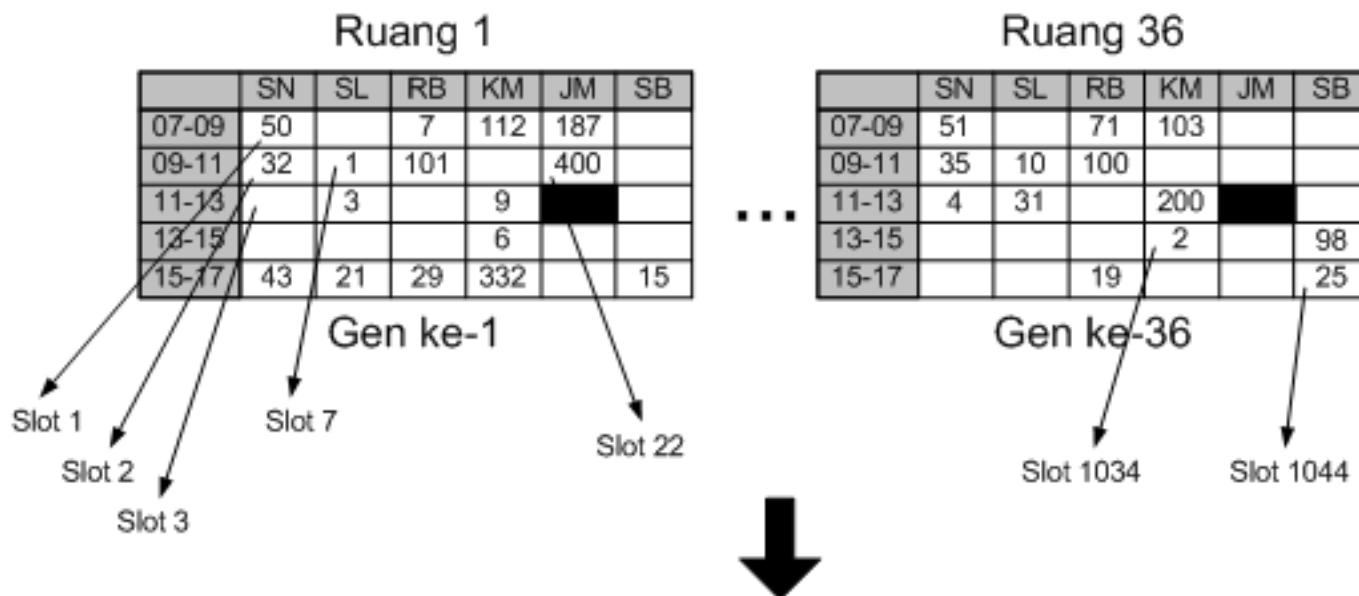
	SN	SL	RB	KM	JM	SB
07-09	51		71	103		
09-11	35	10	100			
11-13	4	31		200		
13-15				2		98
15-17			19			25

Gen ke-36

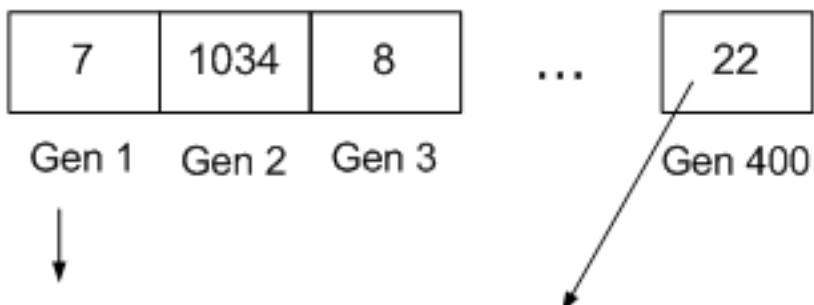
....

Kromosom lain?

## Kromosom dengan 36 gen:



## Kromosom dengan 400 gen:



Posisi gen menyatakan  
nomor urut pertemuan kuliah

Nilai-nilai gen  
menyatakan nomor  
slot

# Batasan

Batasan	Nilai prioritas	Alasan
Tidak bentrok	24	Tidak bentrok merupakan suatu <i>hard constraint</i> (batasan keras) yang sebaiknya tidak dilanggar. Sehingga batasan ini lebih utama dibandingkan harus memenuhi permintaan dosen berjabatan akademik Guru Besar dan Lektor Kepala atau batasan-batasan yang lain. Misal batasan ini diberikan nilai prioritas sebesar 24.
Permintaan Lektor dan Asisten Ahli	4	Kemudian, batasan permintaan dosen berjabatan akademik Guru Besar dan Lektor Kepala diberi nilai prioritas sebesar 12.
Status Dosen	4	Misalkan ketiga batasan yang lainnya diberikan nilai prioritas yang sama, yaitu 4.
Ruang Kuliah	4	

# Pelanggaran Batasan

- $J_a$  = Jumlah jadwal pertemuan yang bentrok kali 24.
- $J_b$  = Jumlah jadwal yang tidak sesuai permintaan dosen berjabatan Guru Besar dan Lektor Kepala dikali 12.
- $J_c$  = Jumlah jadwal yang tidak sesuai permintaan dosen Lektor dan Asisten Ahli dikali 4.
- $J_d$  = Jumlah jadwal yang tidak sesuai ruang kuliahnya dikali 4.
- $J_e$  = Jumlah jadwal yang Tidak Sesuai Status dosen dikali 4.

# Fungsi *Fitness*

$$f = \frac{1}{(B + a)}$$

$$B = J_a + J_b + J_c + J_d + J_e$$

# Operator Evolusi?

- **Seleksi Ortu:**
  - Roullete Wheel, Tournament Selection, Ranking, Scaling, dst.
- **Rekombinasi**
  - Biner, Integer, Real, Permutasi, Gabungan dari rekombinasi tsb.
- **Mutasi**
  - Biner, Integer, Real, Permutasi, Gabungan dari rekombinasi tsb.
- **Seleksi Survivor**
  - Generational atau Steady State?
  - Deterministik atau Probabilistik?
- **EAs Model**
  - Simple Vs Advanced?

A photograph of the National Monument (Monas) in Jakarta, Indonesia. The monument is a tall, white, tapered obelisk topped with a golden spire. It stands on a large, light-colored base. The sky is blue with scattered white clouds. In the foreground, there is a grassy area and a paved walkway. Several people are walking around the base of the monument. In the background, there are other buildings and some industrial structures like towers.

Your cases?

# Question?



# Kasus 1: Pengangkutan Barang

- Diambil dari TA mahasiswa S1-Informatika IT Telkom angkatan 2006: Naufal Mikhdzam Ar Rozi (113061051)
- Mengimplementasikan Algoritma Genetika untuk menemukan solusi optimal
- Melakukan analisis terhadap sistem Algoritma Genetika yang telah dibangun
- Dibangkitkan 3 studi kasus

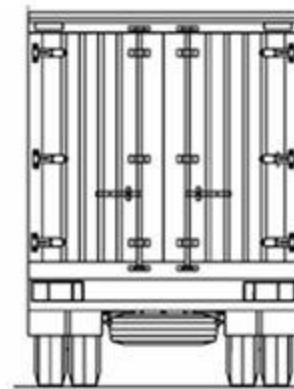
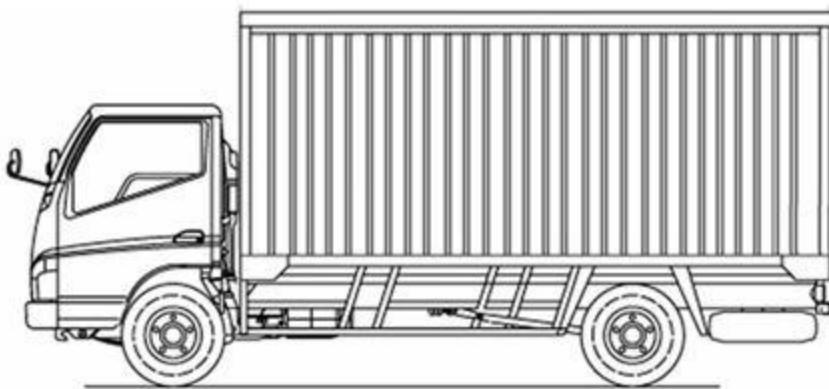
No.	Properti	Keterangan
1.	ID Barang	Identitas dari suatu barang
2.	Panjang	Panjang barang (cm)
3.	Lebar	Lebar barang (cm)
4.	Tinggi	Tinggi barang (cm)
5.	Volume	Volume barang ( $\text{cm}^3$ )
6.	Berat	Berat barang (kilogram)
7.	Kekuatan	Kekuatan barang terhadap beban yang menimpanya (kilogram). Apabila barang memiliki kekuatan 50 kg, maka barang tersebut dapat ditimpa oleh barang-barang dengan maksimal berat total barang yang menimpa adalah 50 kg.
8.	Tingkat Prioritas	Tingkat prioritas barang yang bernilai 1 sampai dengan 5. Tingkat prioritas tertinggi = 5.
9.	Jumlah sisi yang dapat dijadikan alas	Berapa jumlah sisi yang dapat dijadikan alas ketika barang diletakkan (Nilai = 1 sampai 6).
10.	Sisi-sisi yang dapat dijadikan alas	Sisi-sisi mana saja yang dapat dijadikan alas ketika barang diletakkan (Nilai = 1 sampai 6).

## Kasus Tipe A



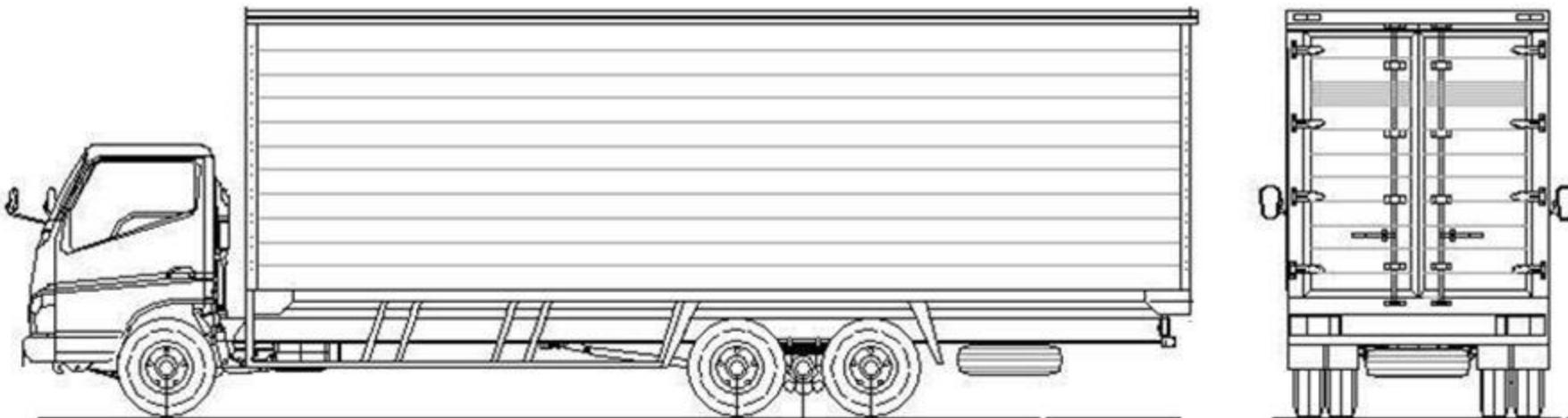
Nama Angkutan	Mitsubishi Colt L300
Panjang	220 cm
Lebar	155 cm
Tinggi	135 cm
Maksimum Beban	1000 kg [6]

## Kasus Tipe B



Nama Angkutan	Mitsubishi Colt Diesel FE 73 HD
Panjang	430 cm
Lebar	197 cm
Tinggi	185 cm
Maksimum Beban	4000 kg [6]

## Kasus Tipe C



Nama Angkutan	Mitsubishi Tronton FE 74
Panjang	710 cm
Lebar	207 cm
Tinggi	218 cm
Maksimum Beban	8000 kg [6]

# Representasi Kromosom

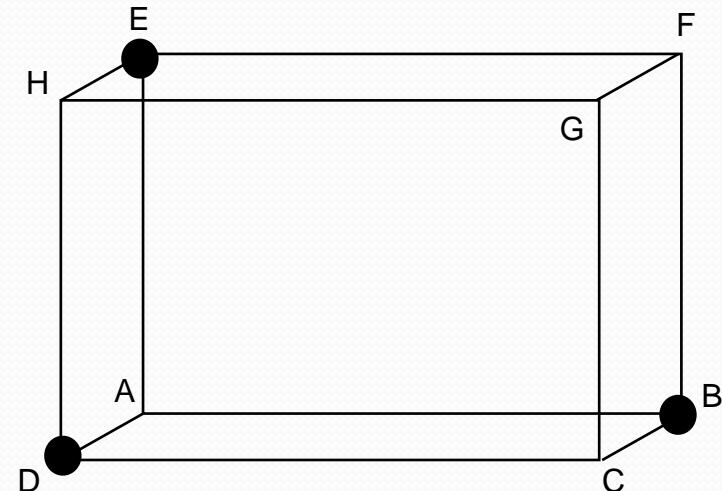
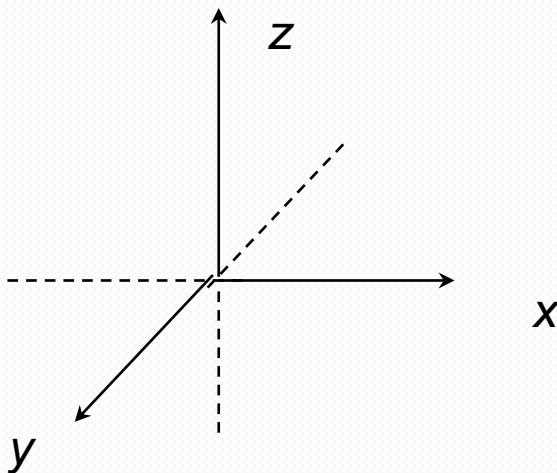
Bagian I

Bagian II

Bagian III

- Bagian I: representasi permutasi yang menggambarkan urutan peletakan barang.
- Bagian II: representasi integer yang menggambarkan sisi barang yang digunakan sebagai alas dalam peletakan.
- Bagian III: representasi biner yang menggambarkan posisi sisi yang dijadikan alas yaitu horizontal atau vertikal

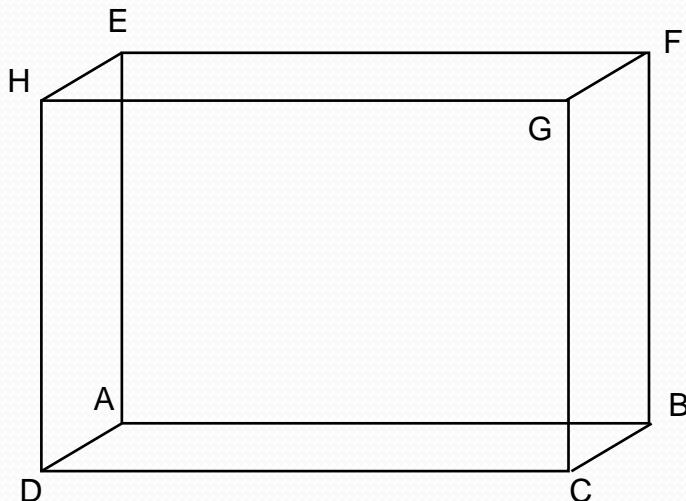
# Peletakan barang



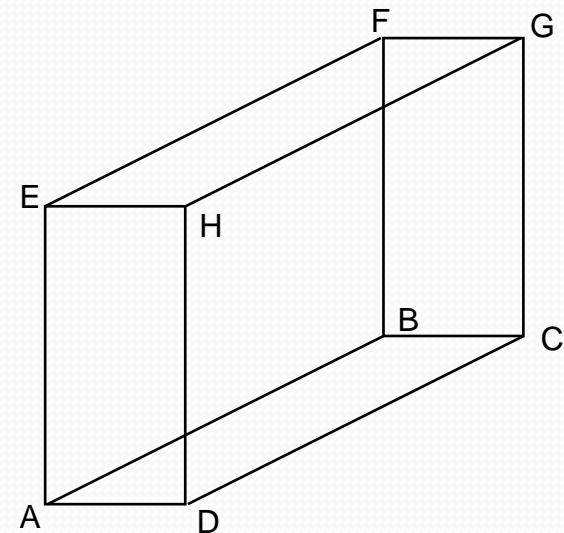
# Penomoran sisi barang

No. Sisi	Sisi
1	ABCD
2	ABFE
3	BCGF
4	CDHG
5	DAEH
6	EFGH

# Posisi peletakan: horizontal/vertikal

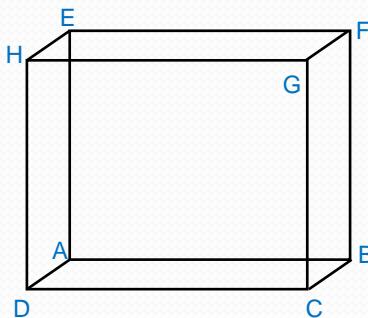


Horizontal

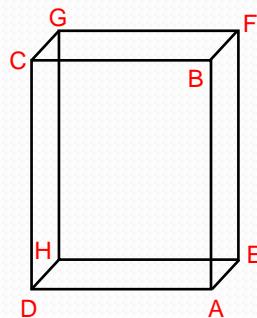


Vertikal

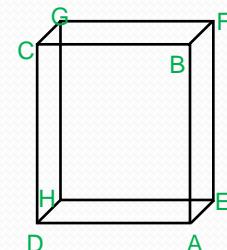
# Individu Vs Kromosom



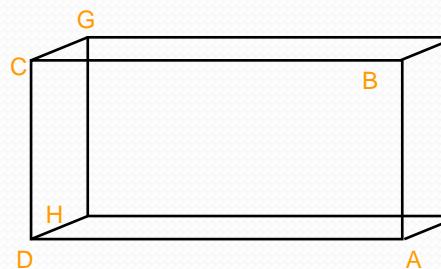
ID = 3



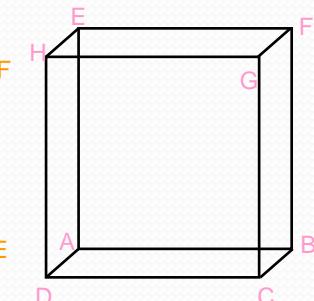
ID = 1



ID = 2



ID = 5

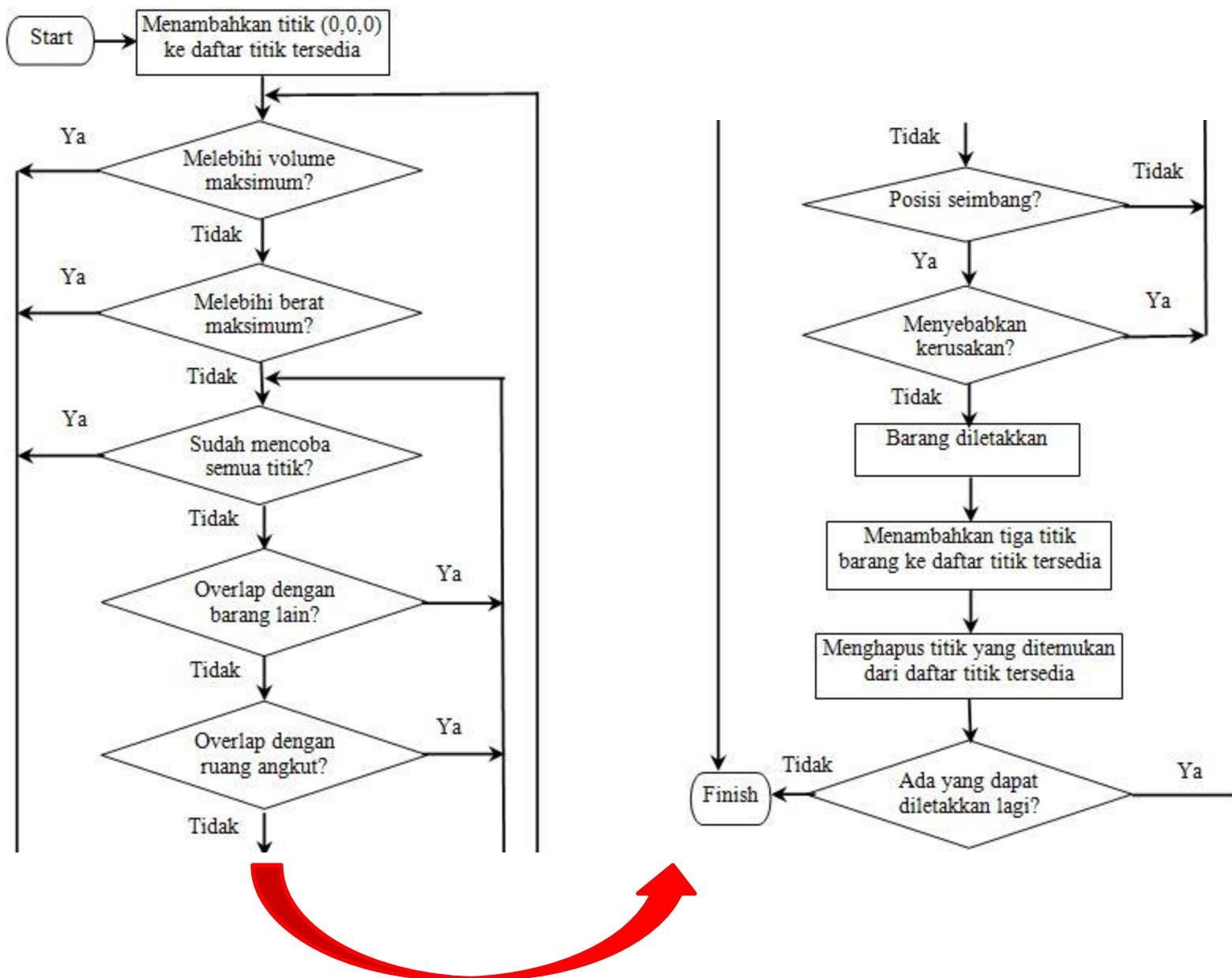


ID = 4



3	1	2	5	4	1	5	5	5	1	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

# Pembangkitan kromosom



# Fungsi Fitness

$$fitness = \frac{Value}{Maximum\ Value} \times 100$$

*Value* : Jumlah biaya dari barang-barang yang terangkut

*Maximum Value*: Jumlah biaya dari semua barang yang ada

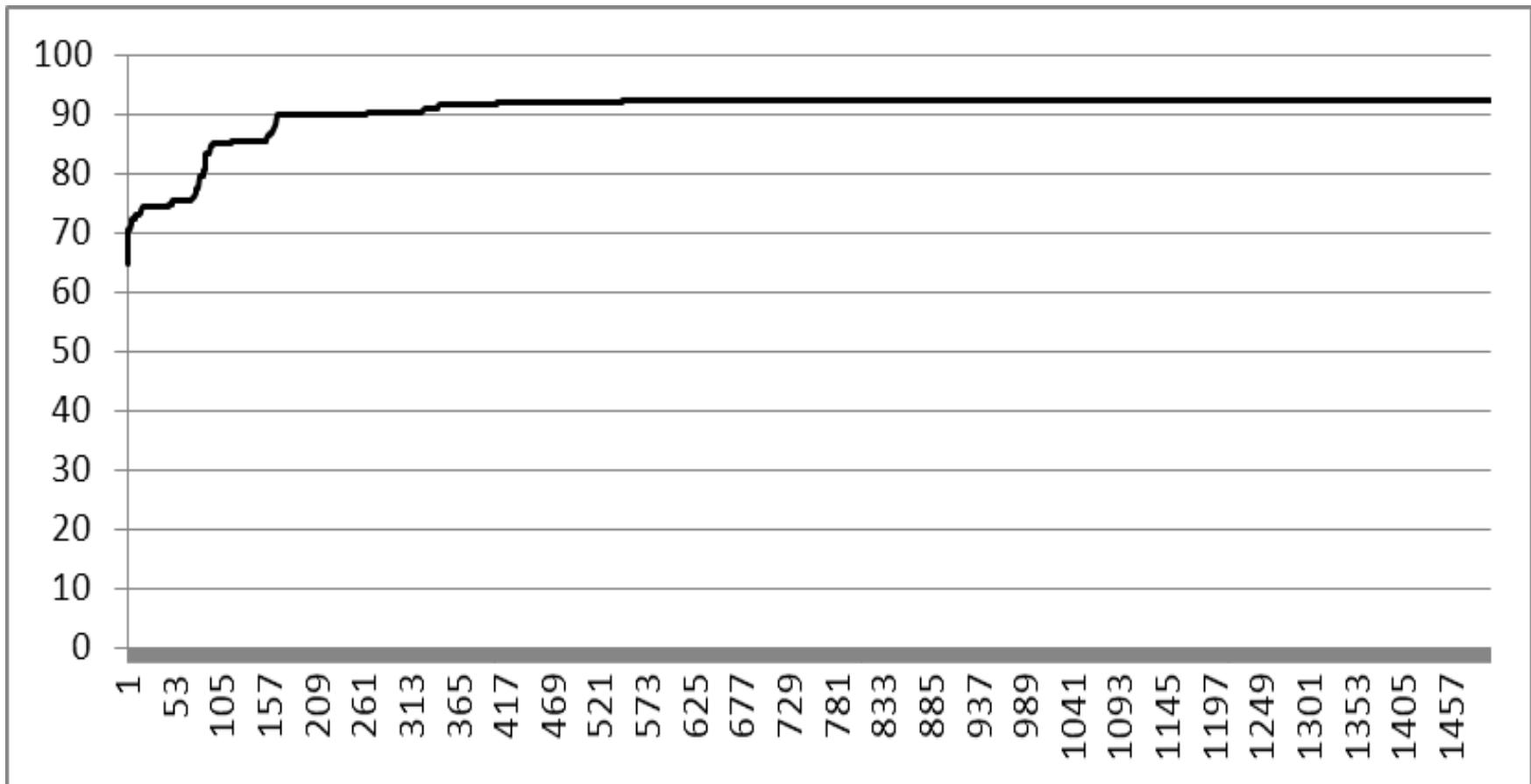
# Biaya pengiriman tiap kilogram untuk tiap tingkat prioritas

Tingkat Prioritas	Biaya (Rp/kg)
1	5000
2	8000
3	15000
4	25000
5	50000

# Operator Evolusi

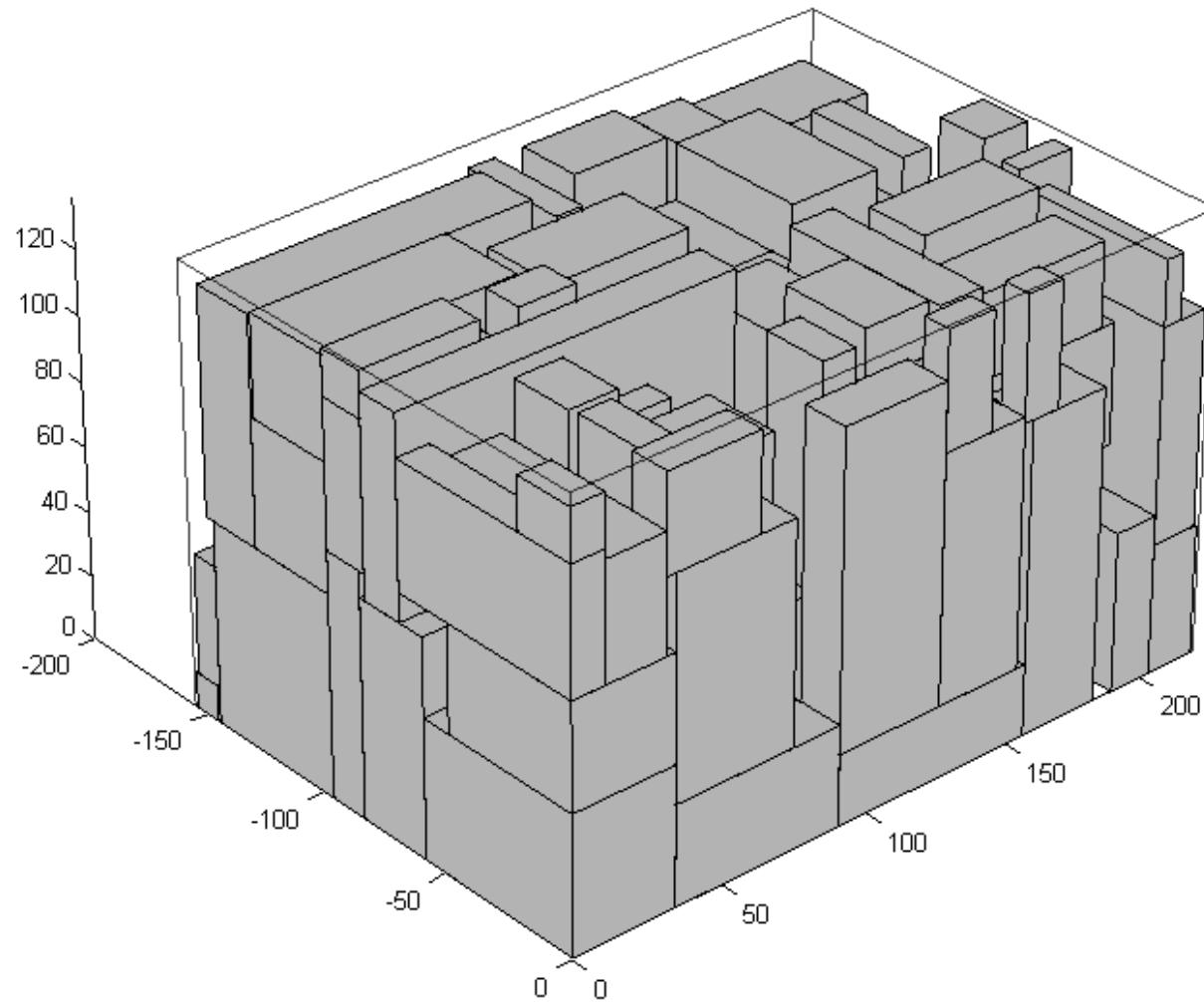
- **Seleksi Orangtua**
  - Tournament Selection
  - Tournament size = 10% dari ukuran populasi
- **Rekombinasi**
  - Bagian I: order crossover
  - Bagian II dan III: uniform crossover
- **Mutasi**
  - Bagian I: swap mutation
  - Bagian II: creep mutation
  - Bagian III: binary mutation
- **Seleksi survivor** : Generational model dengan elitisme

# Hasil Pengujian Kasus A

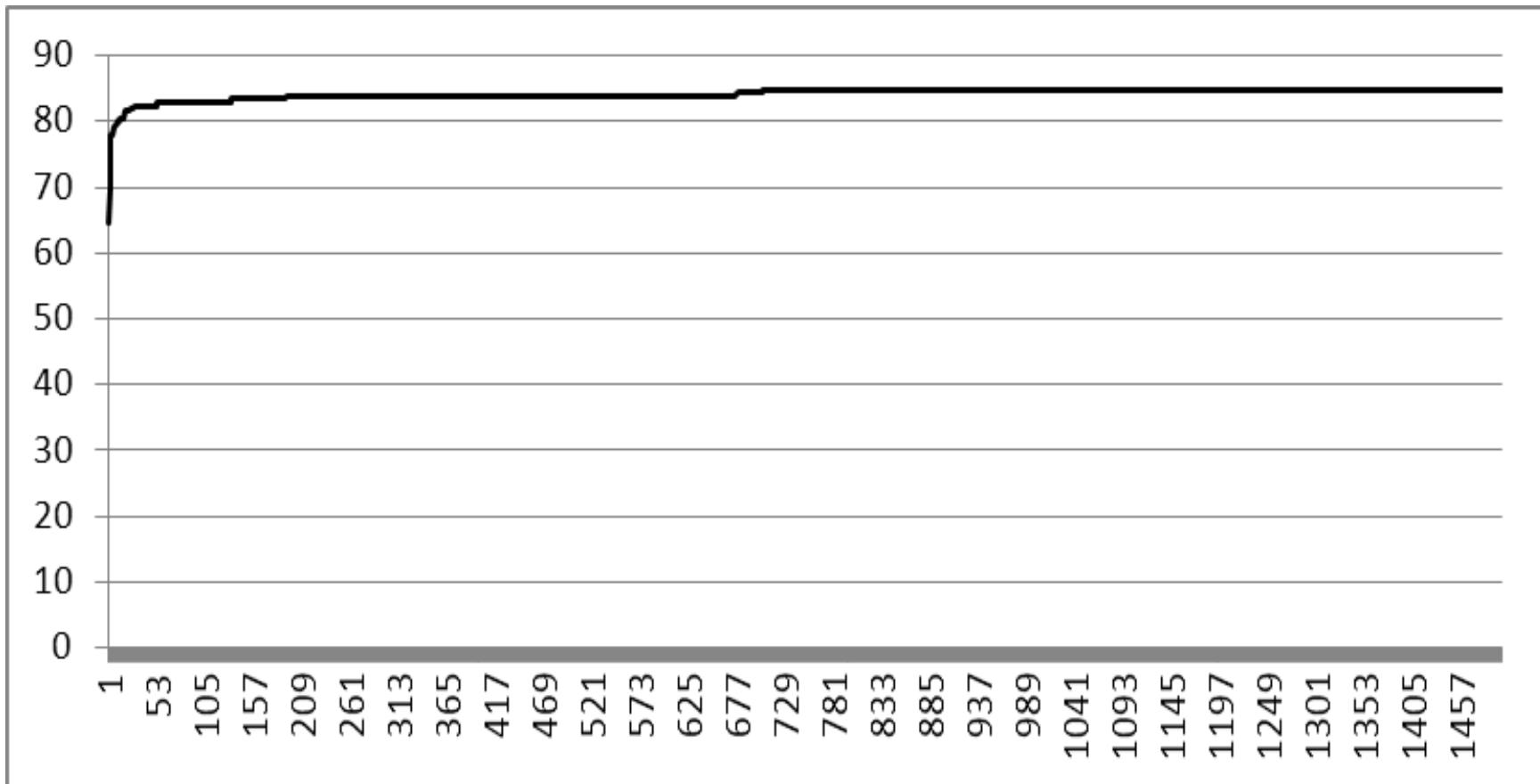


Fitness Terbaik 92.554

# Visualisasi Kasus A

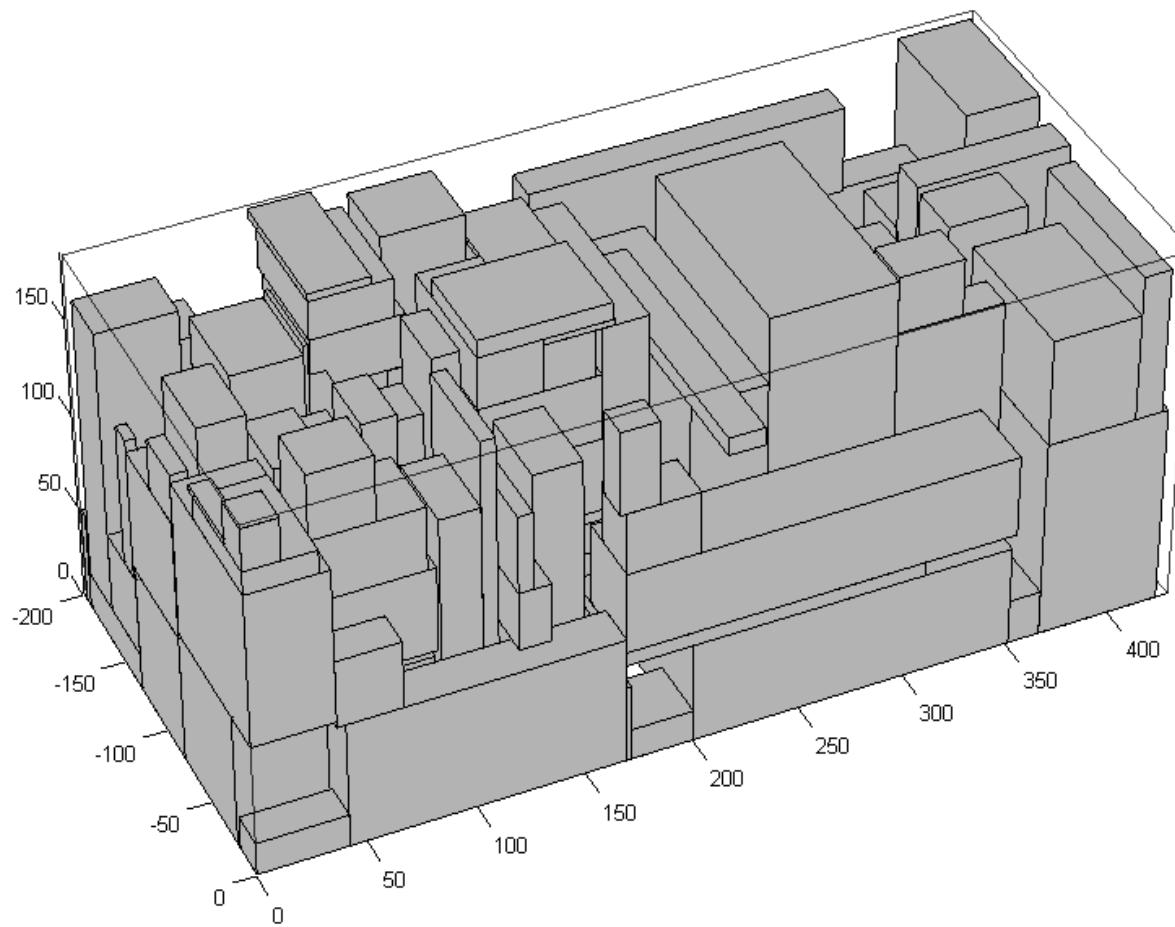


# Hasil Pengujian Kasus B

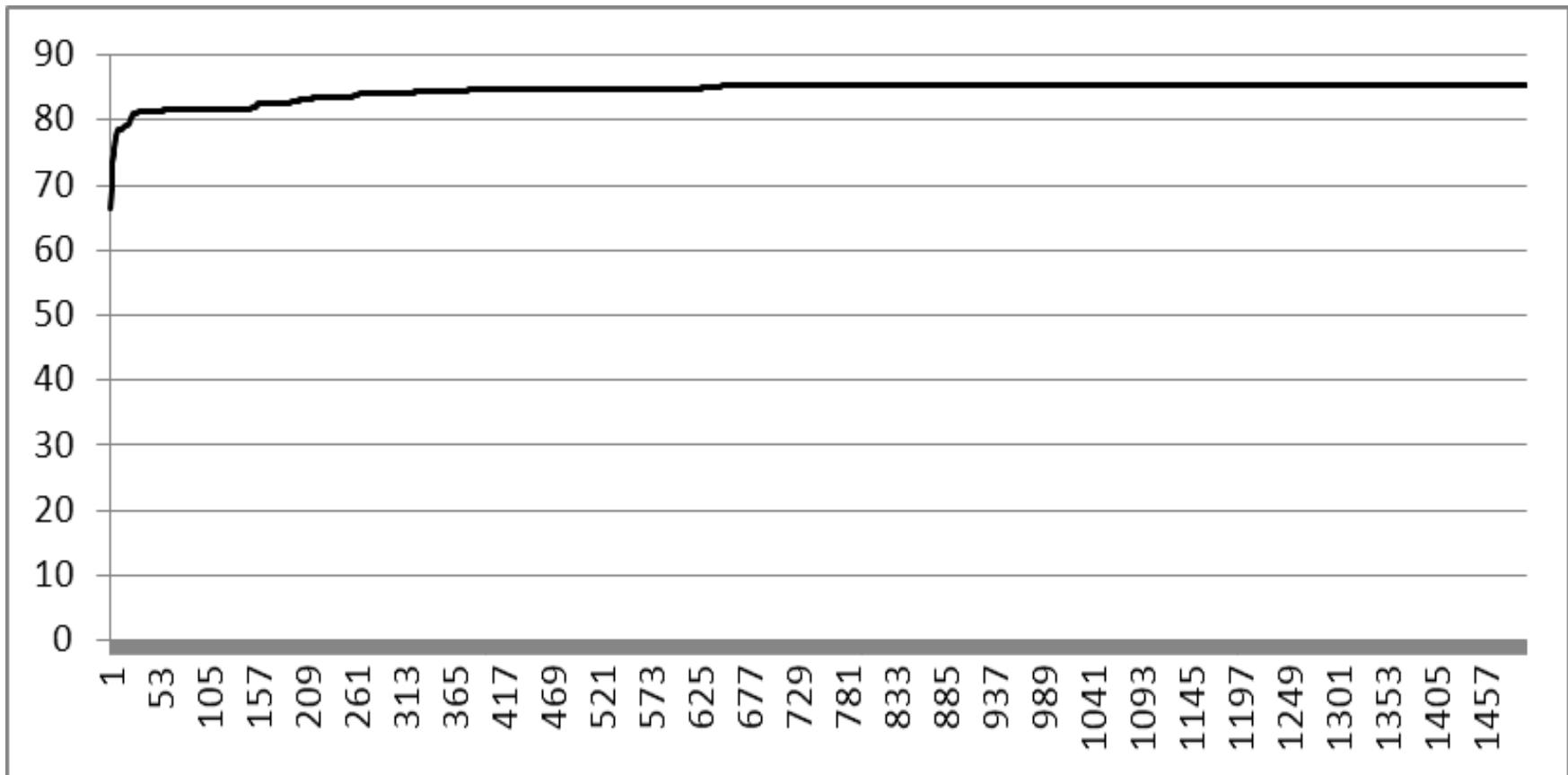


Fitness Terbaik 84.920

# Visualisasi Kasus B

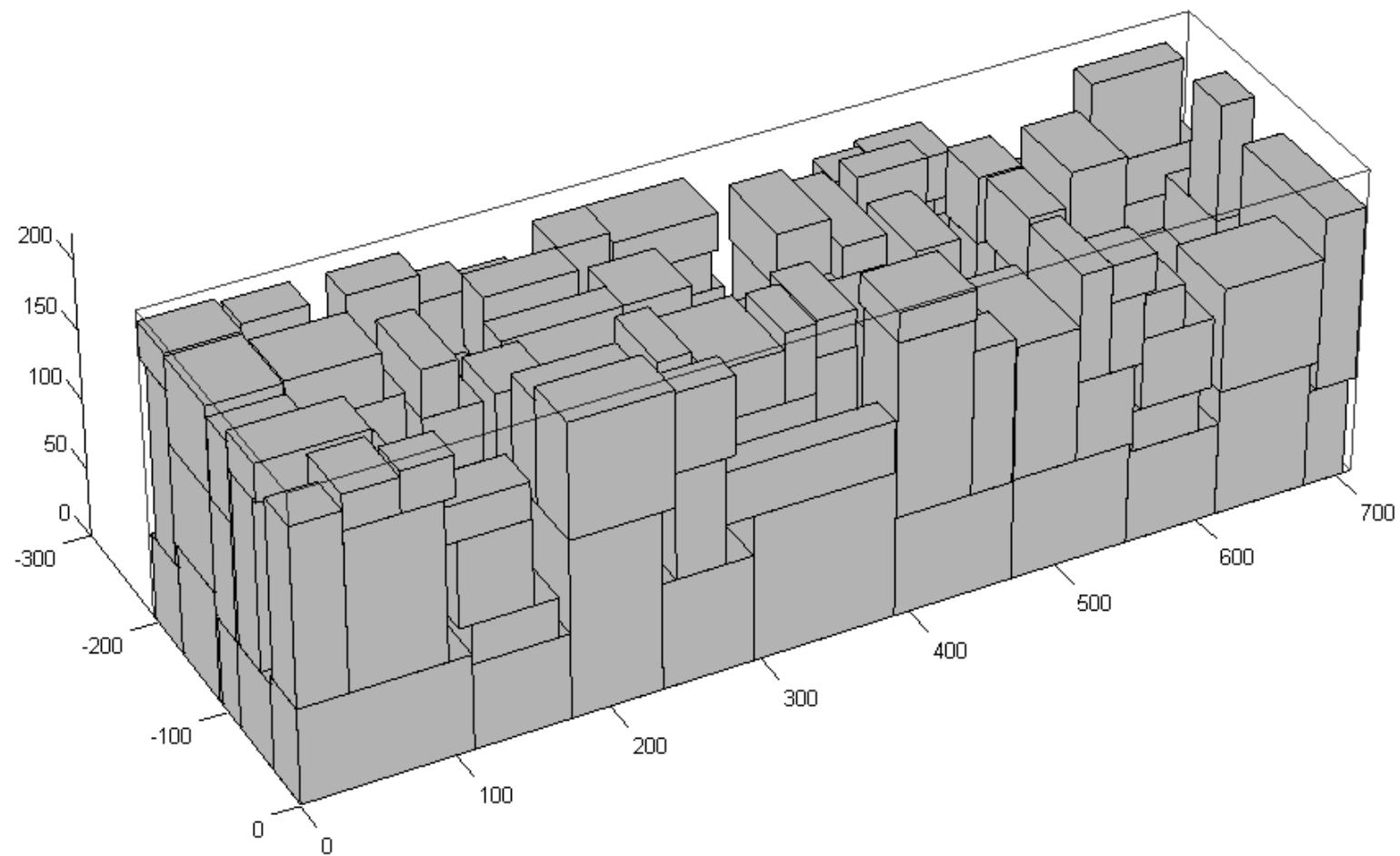


# Hasil Pengujian Kasus C



Fitness Terbaik 85.366

# Visualisasi Kasus C



# Analisa Performansi GA

- Ruang Solusi

$$100! \times 6^{100} \times 2^{100} = 7.7291 \times 10^{265}$$

- Ruang yang dijelajahi 150.000 calon solusi

$$150000 \div (7.7291 \times 10^{265}) = 1.9407 \times 10^{-259}$$

- Performansi GA

- Tipe kasus A: 92.554
- Tipe kasus B: 84.920
- Tipe kasus C: 85.366

# Aplikasi GA

- *Scheduling problems*
- *Chemistry chemical manufacturing*
- *Medicine*
- *Data mining and data analysis*
- *Geometry*
- *Finance and trade*
- *Optimizing distributed protocol*



Question?

# Kesimpulan

- Individu pada GA bisa menggunakan empat representasi berbeda: Biner, Integer, Real, dan Permutasi.
- Operasi Seleksi Orang tua dan Seleksi Survivor tidak bergantung pada representasi individu.
- Rekombinasi dan Mutasi harus dipilih sesuai dengan representasi individu.
- Performansi GA bisa dibuktikan secara matematis meskipun masih sederhana dengan banyak asumsi dan pendekatan.
- Advanced GA sesuai untuk masalah yang sangat kompleks.

# Daftar Pustaka

- [SUYo8] Suyanto, 2008, Evolutionary Computation: Komputasi Berbasis “Evolusi” dan “Genetika”, penerbit Informatika Bandung.
- [RAN88] Ranganathan, B. G, 1988, “Origins?”, Pennsylvania, The Banner Of Truth Trust.
- [ADN07] Adnan Oktar, 2007, "Mekanisme Khayalan Teori Evolusi", [www.evolutiondeceit.com/indonesian/keruntuhan3.php](http://www.evolutiondeceit.com/indonesian/keruntuhan3.php)