

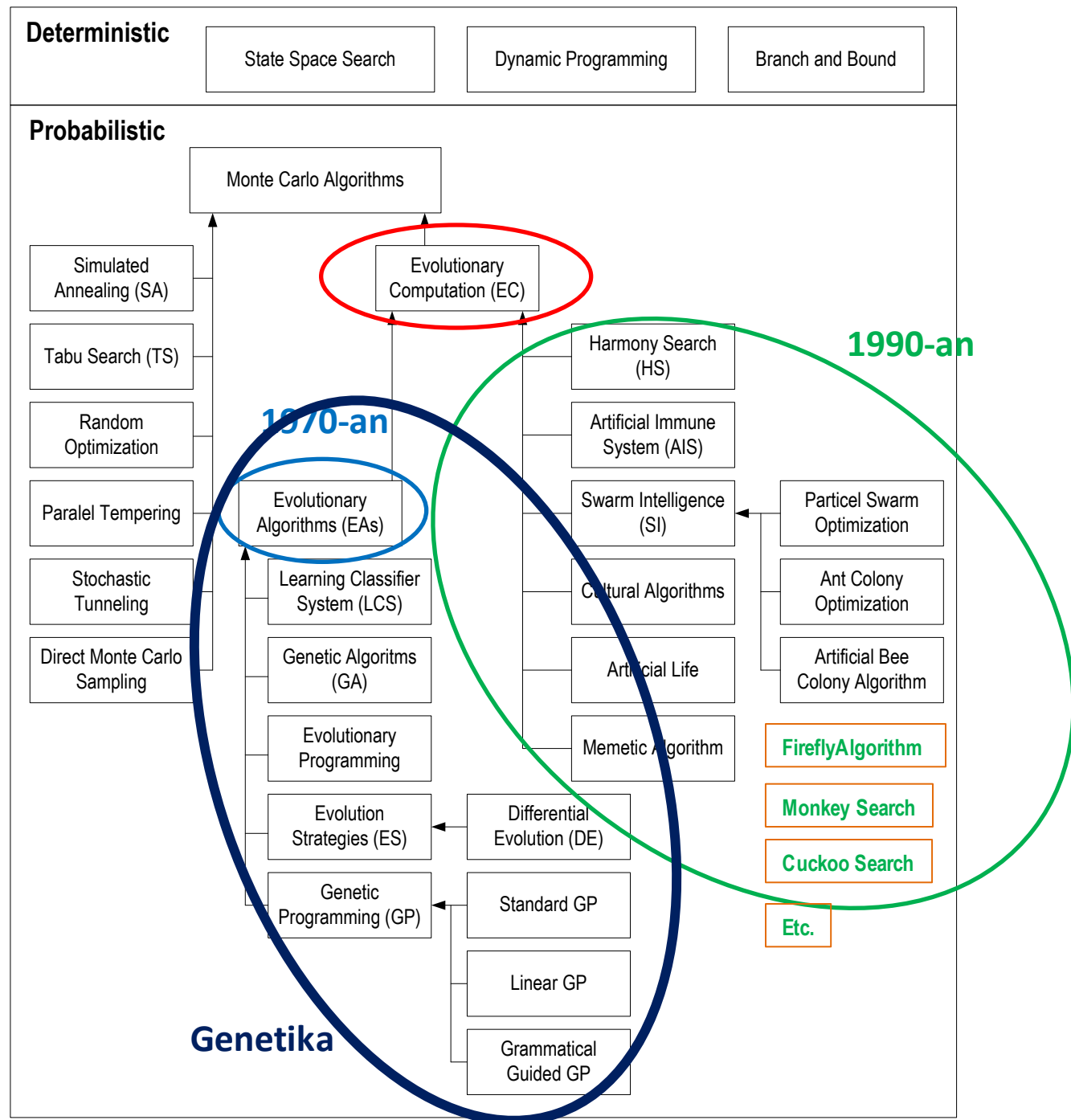
Evolutionary Algorithms (EAs)

Dr. Suyanto, S.T., M.Sc.

HP/WA: 0812 845 12345

Intelligence Computing Multimedia (ICM)
Informatics faculty – Telkom University

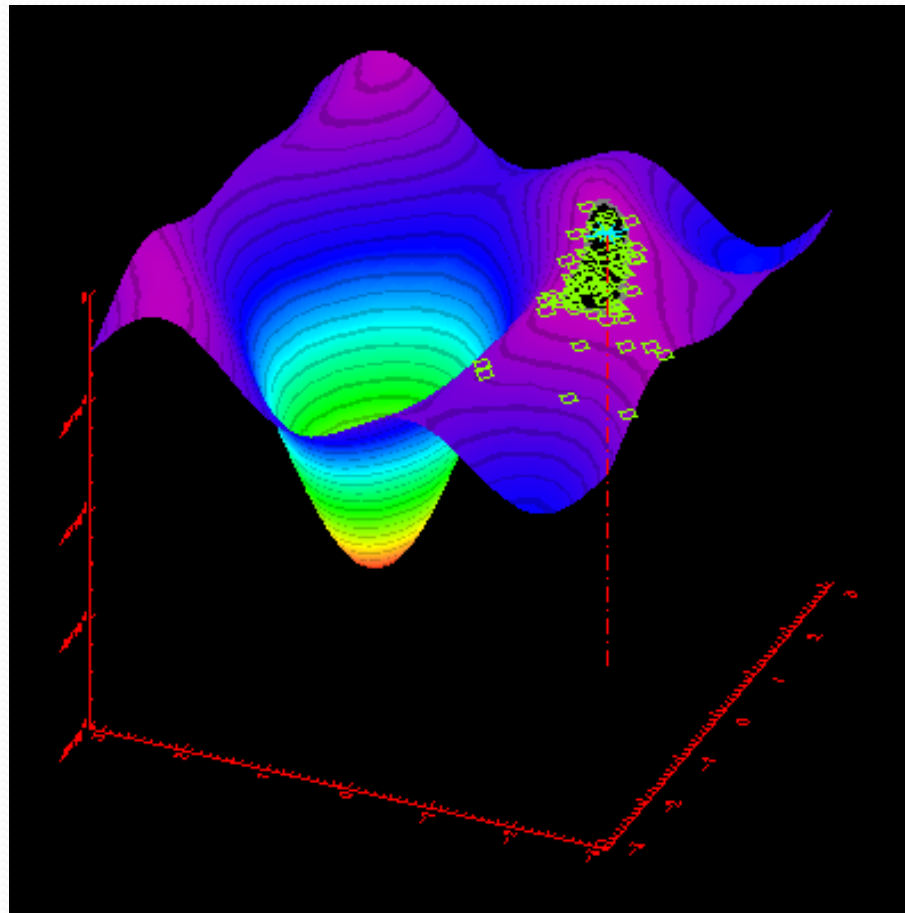
Optimization Algorithms



Swarm, Flock, School, Herd



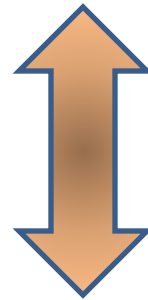
Self Adaptation



Terdapat kriteria tertentu → solusi atau bukan?

Searching

SPECIFIC



Optimization

GENERAL

Fungsi-fungsi objektif → konfigurasinya bagus atau tidak?

- **Searching**: terdapat kriteria tertentu yang menyatakan apakah suatu elemen x_i adalah solusi atau bukan.
- **Optimization**: mungkin tidak terdapat kriteria tersebut, melainkan hanya fungsi-fungsi objektif yang menggambarkan bagus atau tidaknya suatu konfigurasi yang diberikan.
- Fungsi-fungsi objektif bisa memberikan definisi masalah yang lebih umum → Optimization adalah **GENERALISASI** dari Searching.

Masalah Searching atau Optimasi?

1. Navigation System **Searching**
2. Travelling Salesman Problem (TSP) **Optimasi**
3. Vehicle Routing Problem (VRP) **Optimasi**
4. University Course Timetabling Problem **Optimasi**
5. Knapsack Problem **Searching**

Masalah searching bisa diselesaikan dengan algoritma Searching atau Optimasi. Begitu juga sebaliknya.

Apa itu EC?

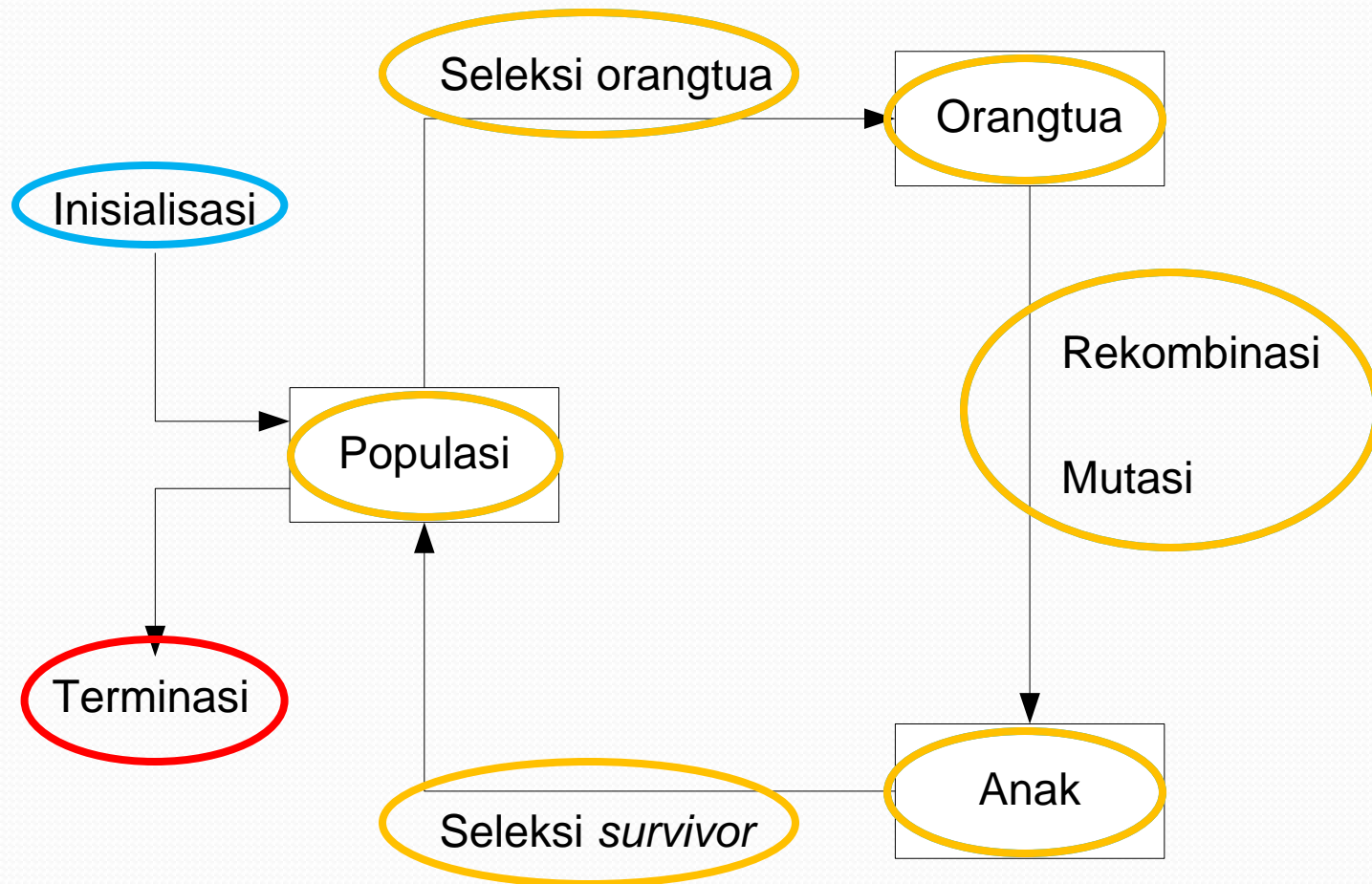
Evolutionary Computation is an abstraction from the theory of **biological evolution** that is used to create **optimization** procedures or methodologies, usually implemented on computers, that are used to solve problems“ [JULo7].

Apa itu EAs?

Evolutionary Algorithms are generic, **population-based meta-heuristic** optimization algorithms that use **biology-inspired** mechanisms like **mutation**, **crossover**, **natural selection** and **survival of the fittest**.

EAs = algoritma2 yang mengimplementasikan abstraksi **EC**

Skema umum EAs



EAs dengan *Steady State Replacement*

Bangkitkan populasi awal, N kromosom

random

Loop untuk N kromosom

 individu = Dekode(kromosom)

 fitness = Evaluasi(individu)

End

Loop sampai Kondisi Berhenti terpenuhi

 Pilih dua kromosom sebagai orangtua P1 dan P2

random

 [anak1, anak2] = Rekombinasi(P1, P2)

random

 [anak1, anak2] = Mutasi(anak1, anak2)

random

 Penggantian(populasi, anak1, anak2)

End

EAs dengan *Generational Replacement*

Bangkitkan populasi awal, N kromosom

random

Loop sampai Kondisi Berhenti terpenuhi

Loop untuk N kromosom

 individu = Dekode(kromosom)

 fitness = Evaluasi(individu)

End

 Buat satu atau dua kopi kromosom terbaik

Loop sampai didapatkan N kromosom baru

 Pilih dua kromosom sebagai orangtua P1 dan P2

random

 [anak1, anak2] = Rekombinasi(P1, P2)

random

 [anak1, anak2] = Mutasi(anak1, anak2)

random

End

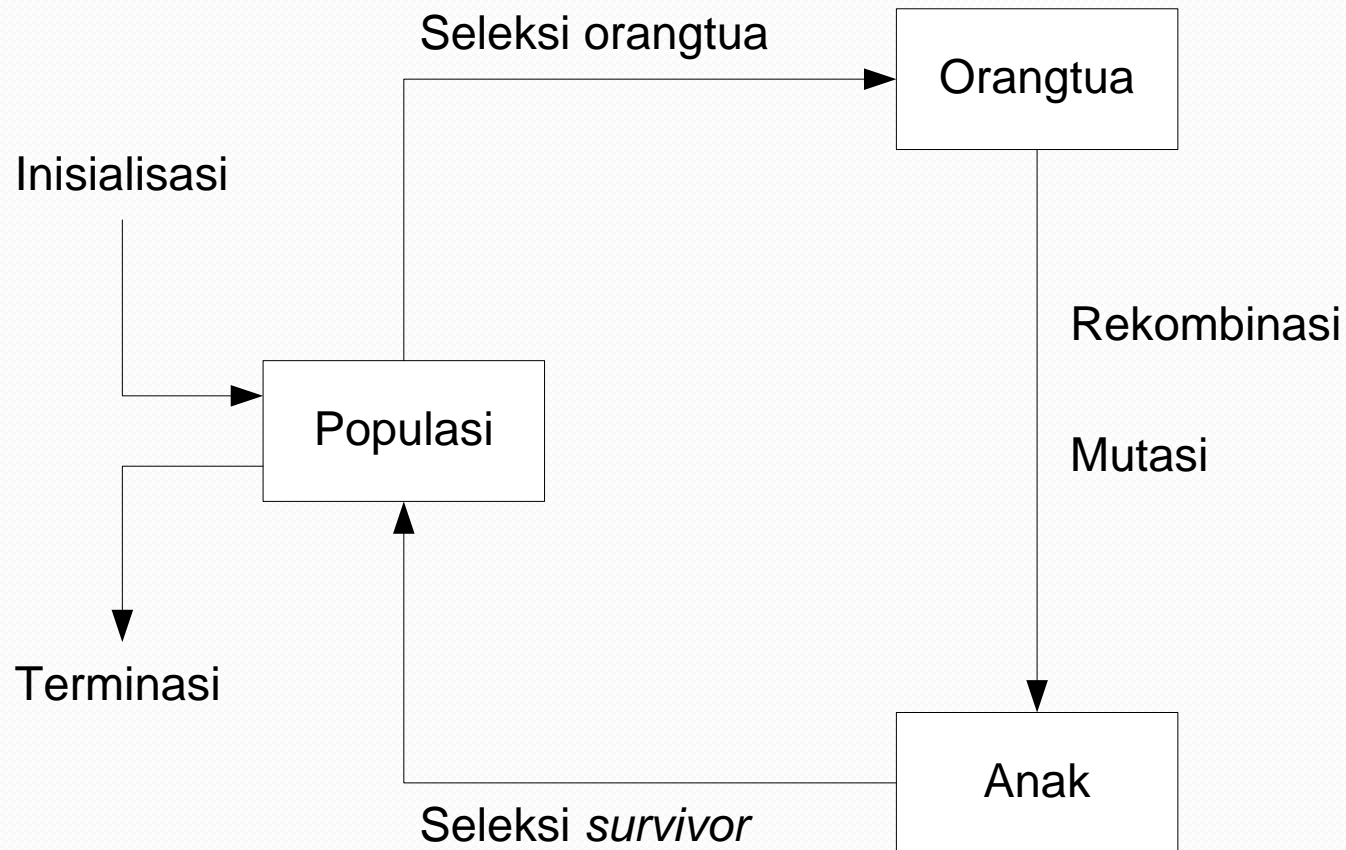
 Ganti N kromosom lama dengan N kromosom baru

End

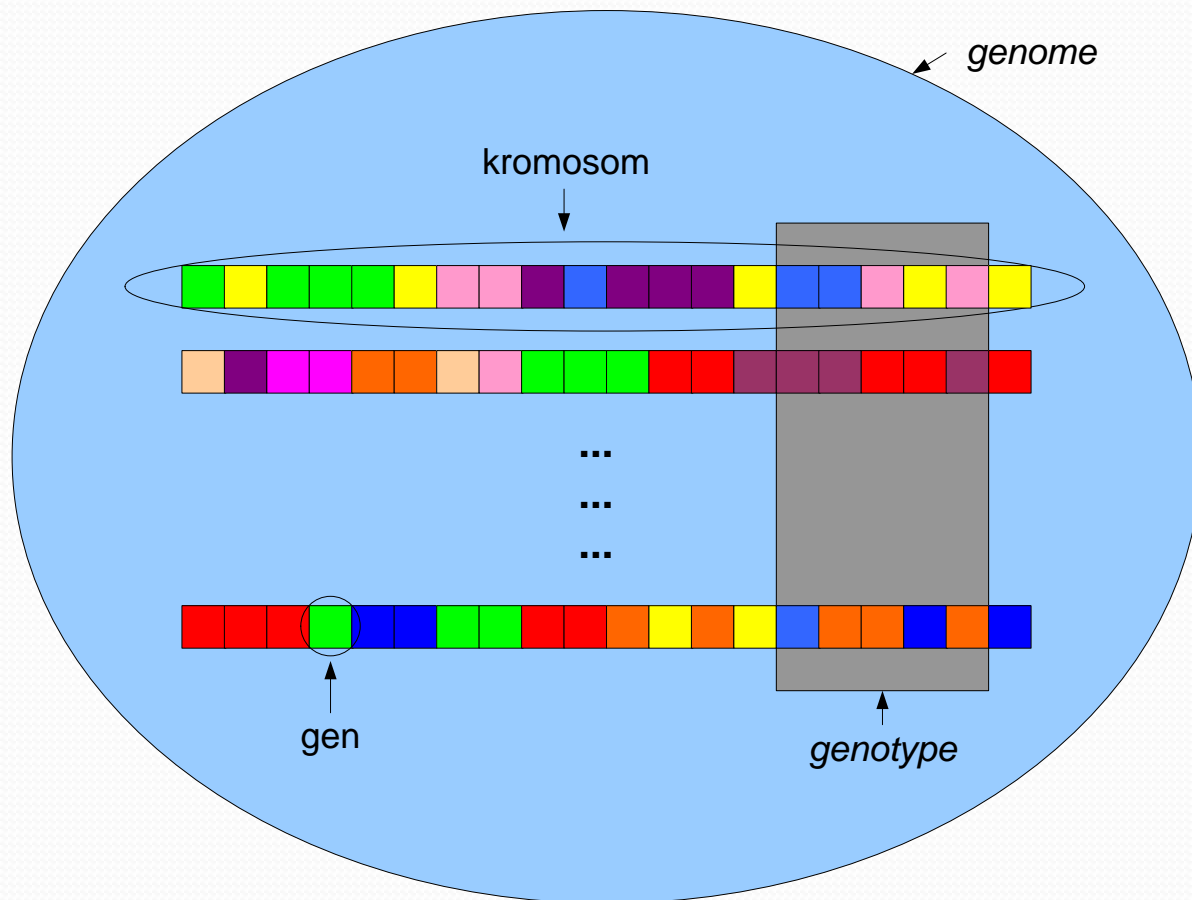
Yang termasuk EAs:

1. Genetic Algorithms (GA): binary strings
2. Evolution Strategies (ES): real-valued vectors
3. Evolutionary Programming (EP): finite state machines
4. Genetic Programming (GP): LISP trees
5. Differential Evolution (DE) → perkembangan ES
6. Grammatical Evolution (GE) → perkembangan GP

Skema umum EAs



Terminologi



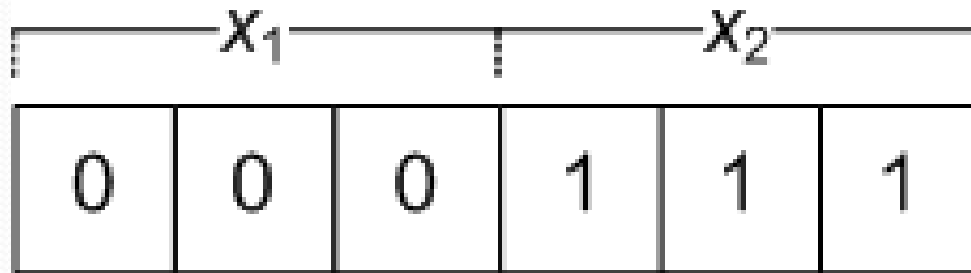
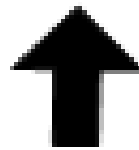
Pengkodean individu → kromosom

- Suatu **Individu** dikodekan ke dalam **Kromosom** dengan cara yang sesuai.
- Empat pengkodean yang umum adalah:
 - Biner
 - Integer
 - Real
 - Permutasi

1. Pengkodean Biner

Rentang Nilai: [-1,2]

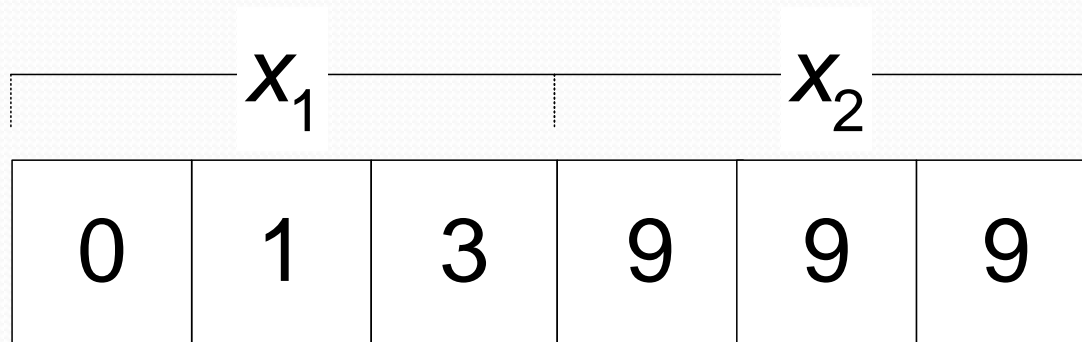
individu: $x_1 = -1$ dan $x_2 = 2$



2. Pengkodean Integer

Rentang Nilai: [-1,2]

individu: $x_1 = -0,96096$ dan $x_2 = 2$



3. Pengkodean Real

Rentang Nilai: [-1,2]

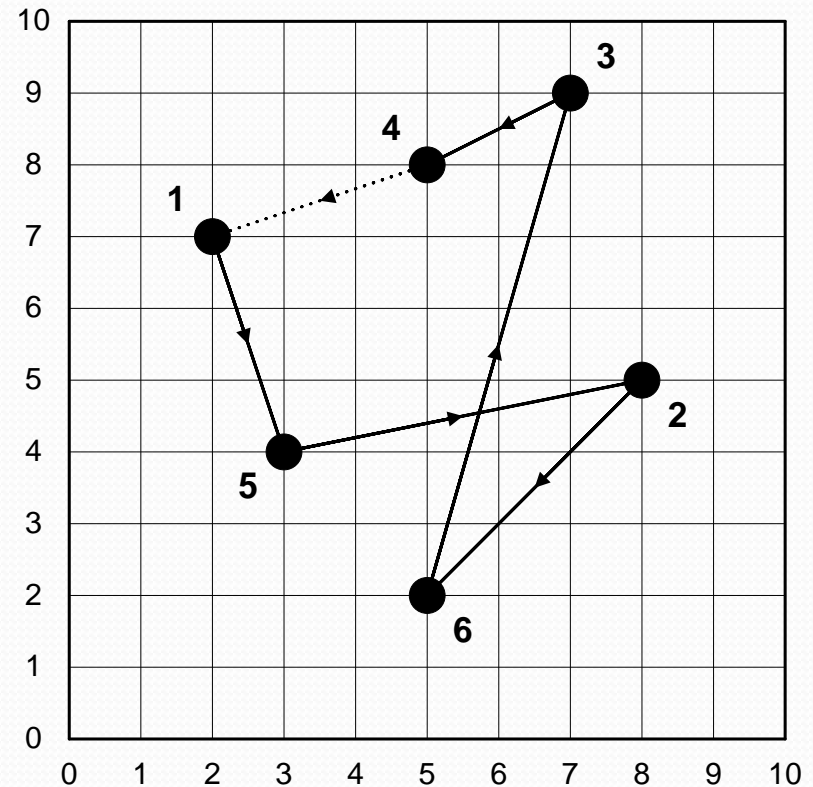
individu: $x_1 = -0,2830$ dan $x_2 = 2$



x_1	x_2
0,2390	1,0000

4. Pengkodean permutasi

Individu: urutan kunjungan semua lokasi



Kromosom:



1 2 3 4 5 6

Nilai gen menyatakan nomor lokasi

Posisi gen menyatakan urutan kunjungan

Pendekodean kromosom → individu

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

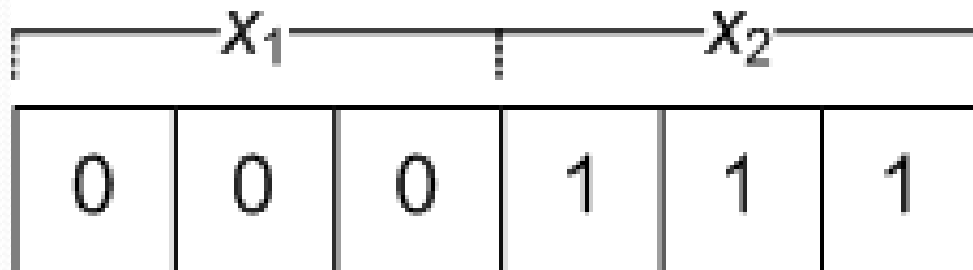
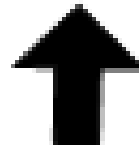
$$x = r_b + \frac{(r_a - r_b)}{\sum_{i=1}^N 9 \cdot 10^{-i}} (g_1 \cdot 10^{-1} + g_2 \cdot 10^{-2} + \dots + g_N \cdot 10^{-N})$$

$$x = r_b + (r_a - r_b)(g_1 + g_2 + \dots + g_N)$$

1. Pengkodean Biner

Rentang Nilai: [-1,2]

individu: $x_1 = -1$ dan $x_2 = 2$



Nilai *Fitness*

- Maksimasi

$$f = h$$

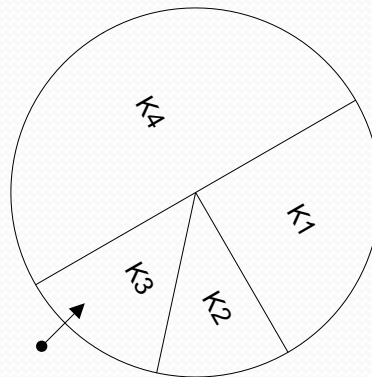
- Minimasi

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

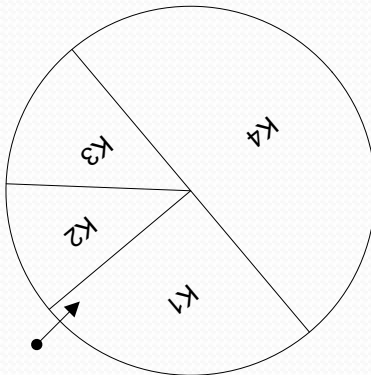
Seleksi Orangtua

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

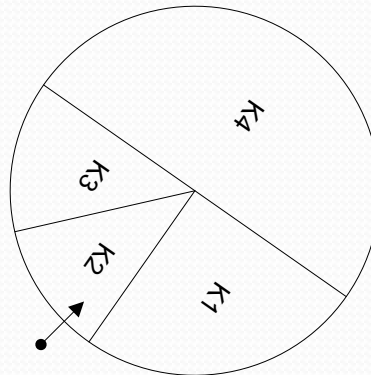
Metode: *roulette wheel*



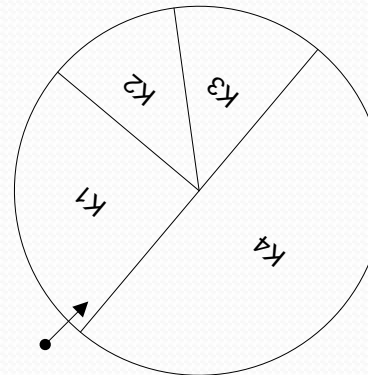
Putaran ke-1



Putaran ke-2

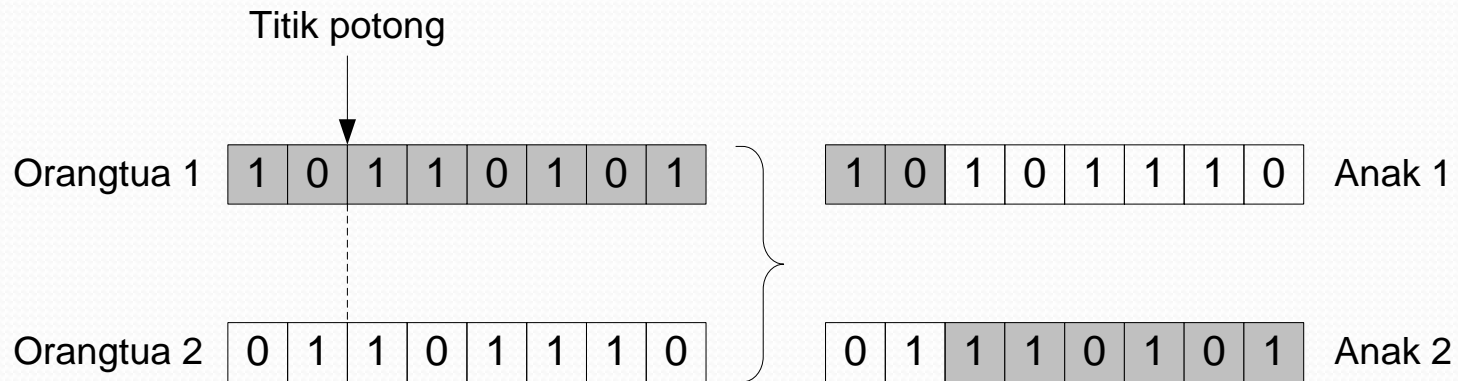


Putaran ke-3

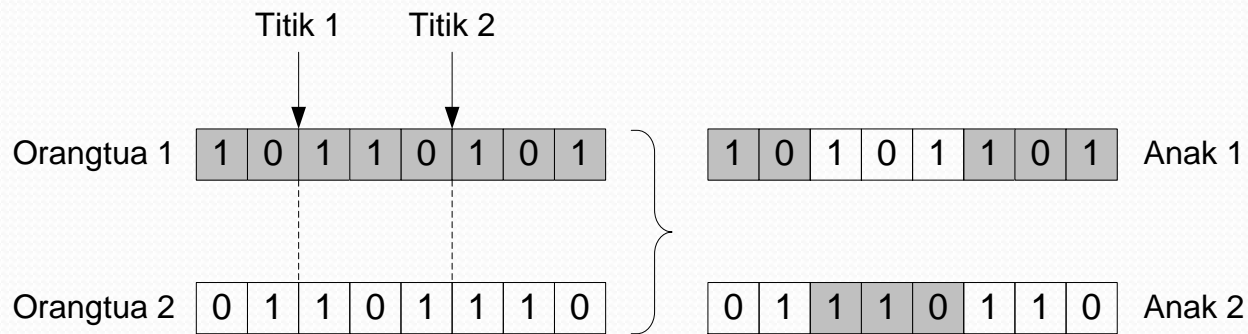


Putaran ke-4

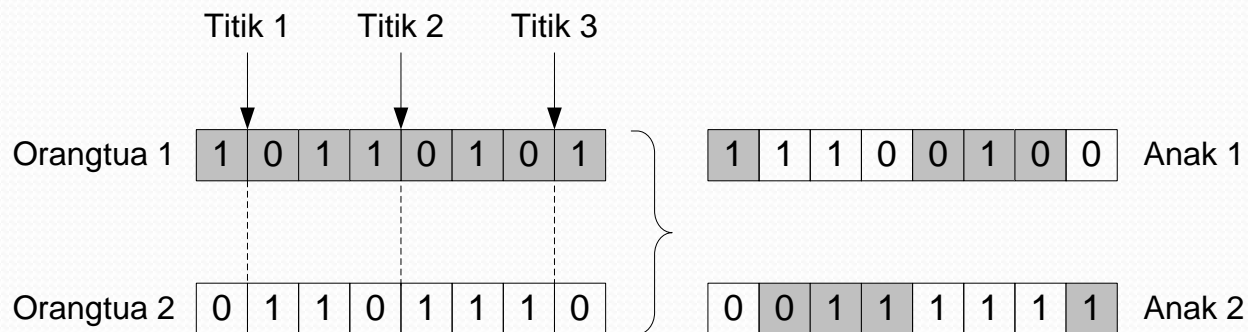
Rekombinasi/Crossover



Rekombinasi

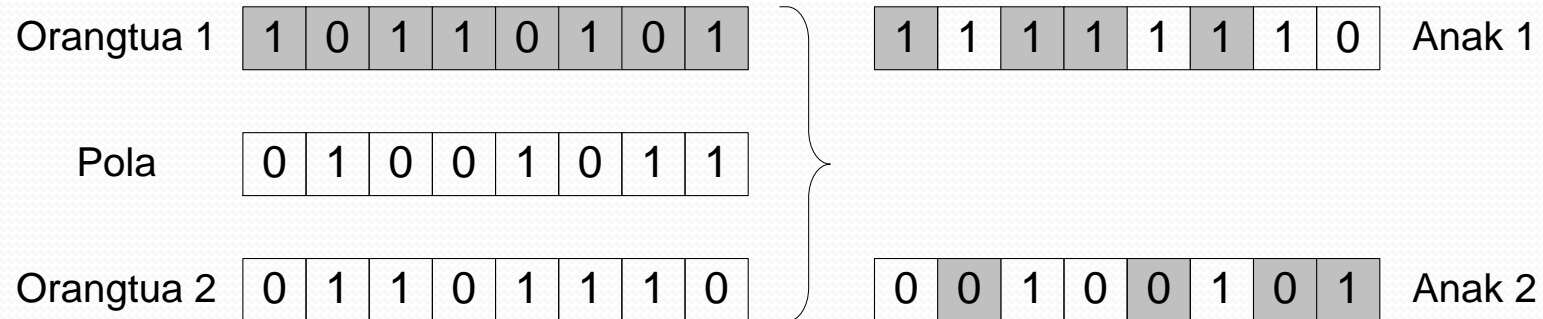


(a)



(b)

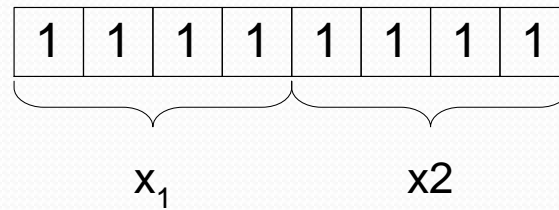
Rekombinasi



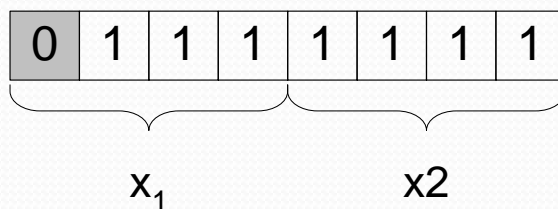
Mutasi

$h = 5x_1 + 2x_2 \rightarrow$ Maksimasi h dimana x_1 & x_2 : integer $[0,15]$

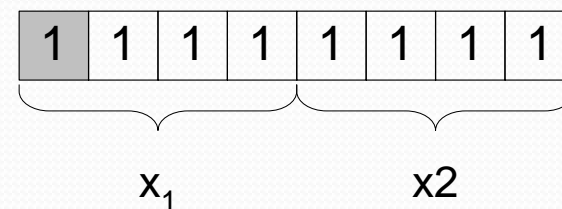
Kromosom yang menghasilkan nilai maksimum

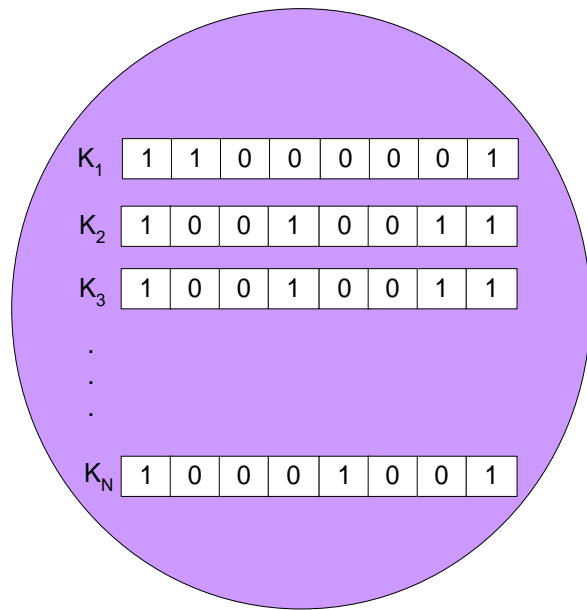


Kromosom awal



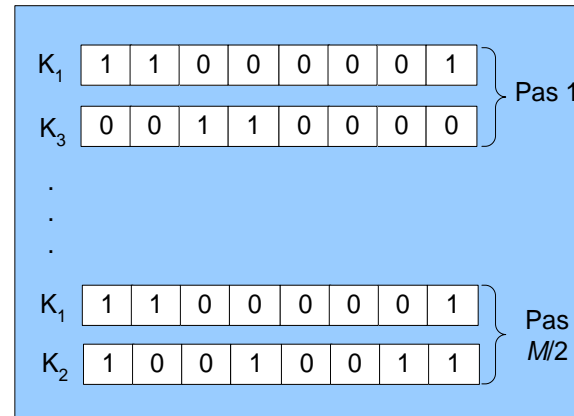
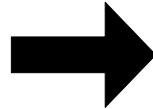
Kromosom hasil mutasi





Generasi 1: Populasi N kromosom/individu

Seleksi
orangtua

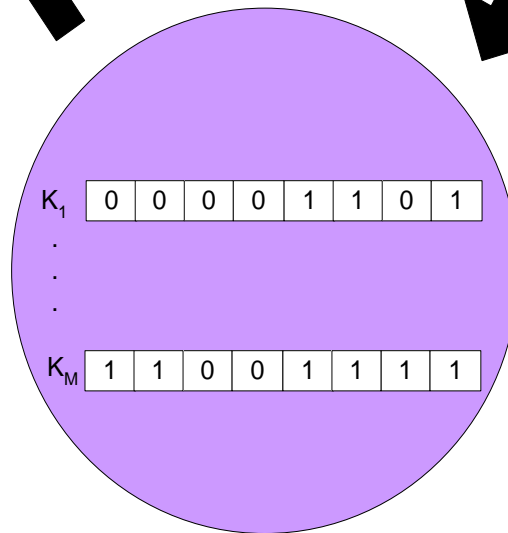
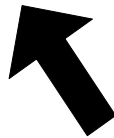


Mating pool: $M/2$ pasang kromosom orangtua

Rekombinasi
dan Mutasi

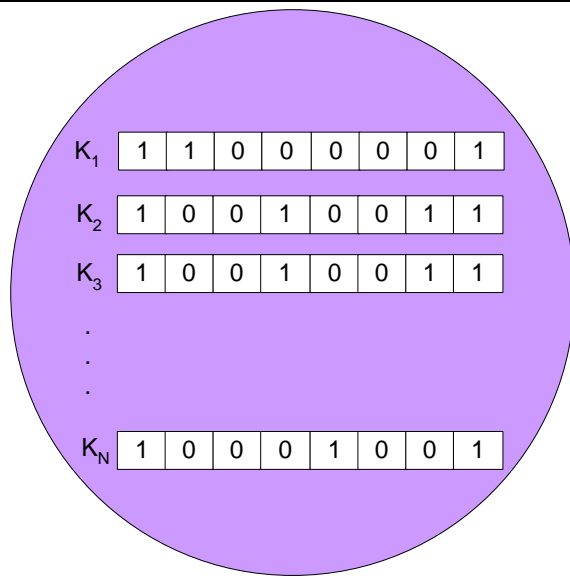


Penggantian
 M kromosom
berdasarkan
metode seleksi
survivor



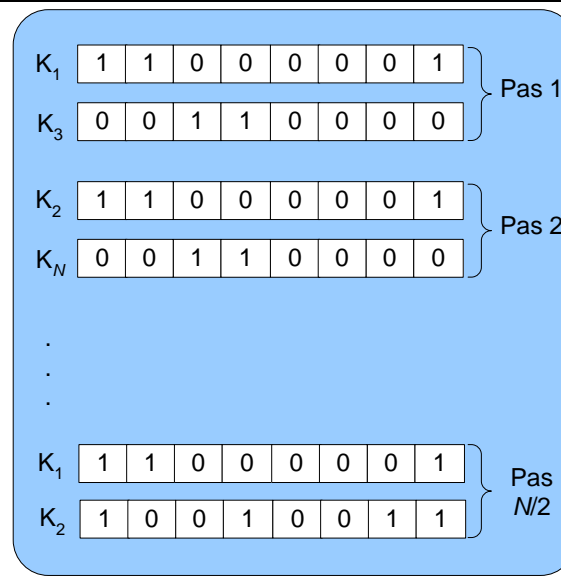
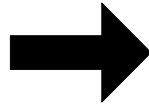
M kromosom baru

Seleksi Survivor: *Steady State*



Generasi 1: Populasi N kromosom/individu

Seleksi
orangtua

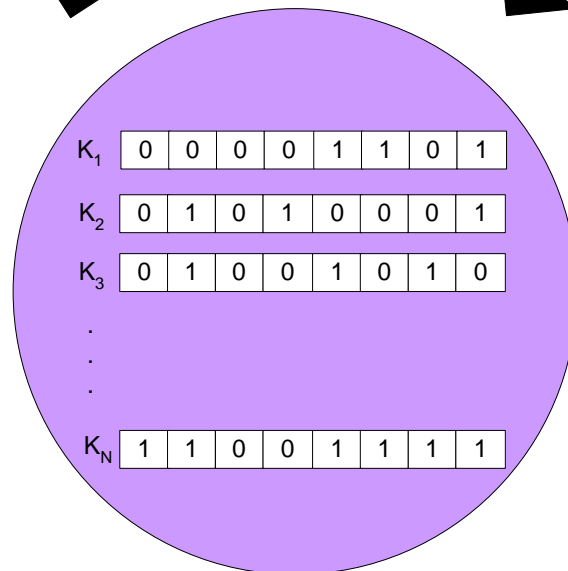


Mating pool: $N/2$ pasang kromosom orangtua

Rekombinasi
dan Mutasi



Penggantian
 N kromosom



N kromosom baru

Seleksi Survivor: *Generational*

Studi kasus 1: Maksimasi Fungsi

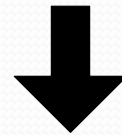
$$h(x_1, x_2) = 7x_1 - 3x_2$$

x_1, x_2 dalam interval $[0, 15]$

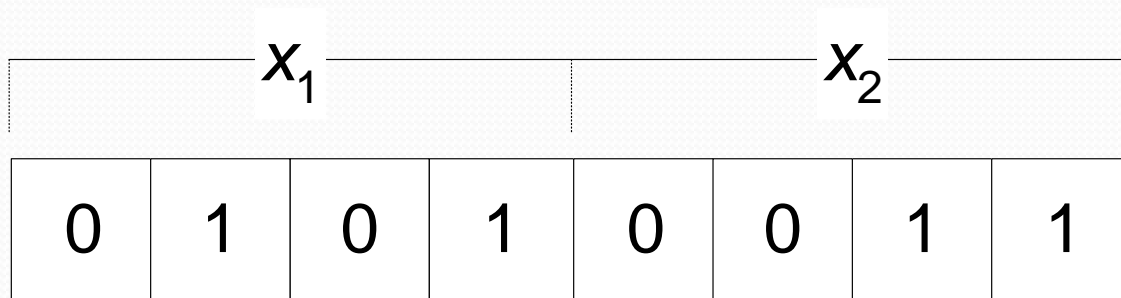
$h = \mathbf{105}$, dimana $x_1 = 15$ dan $x_2 = 0$

Phenotype \rightarrow *Genotype*

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



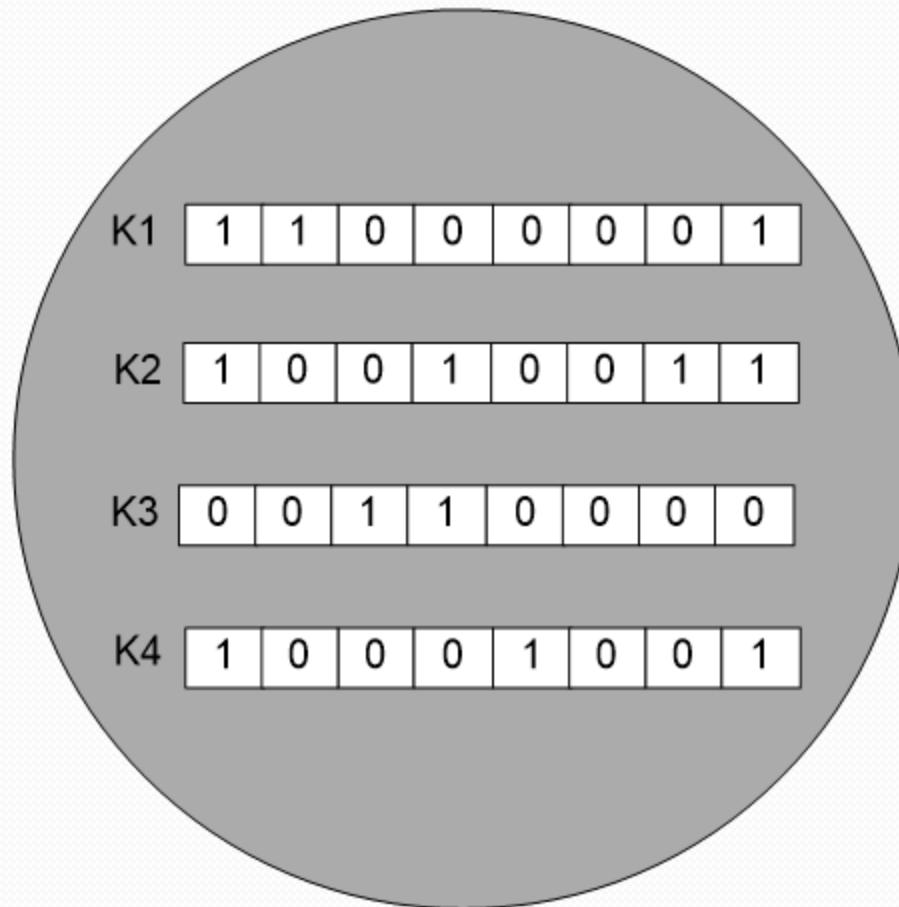
kromosom dengan *binary encoding*



Fungsi Fitness: Maksimasi

$$f = h$$

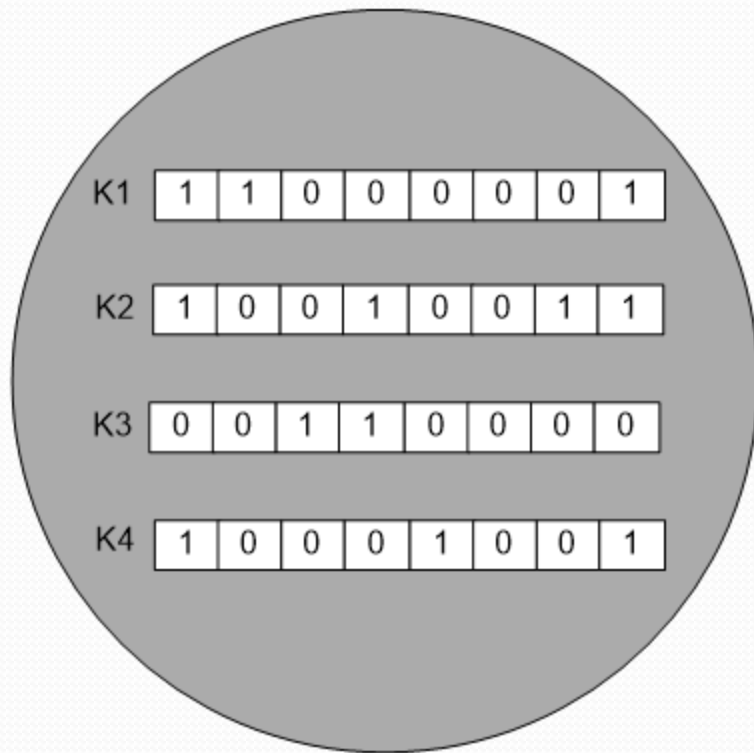
Generasi 1



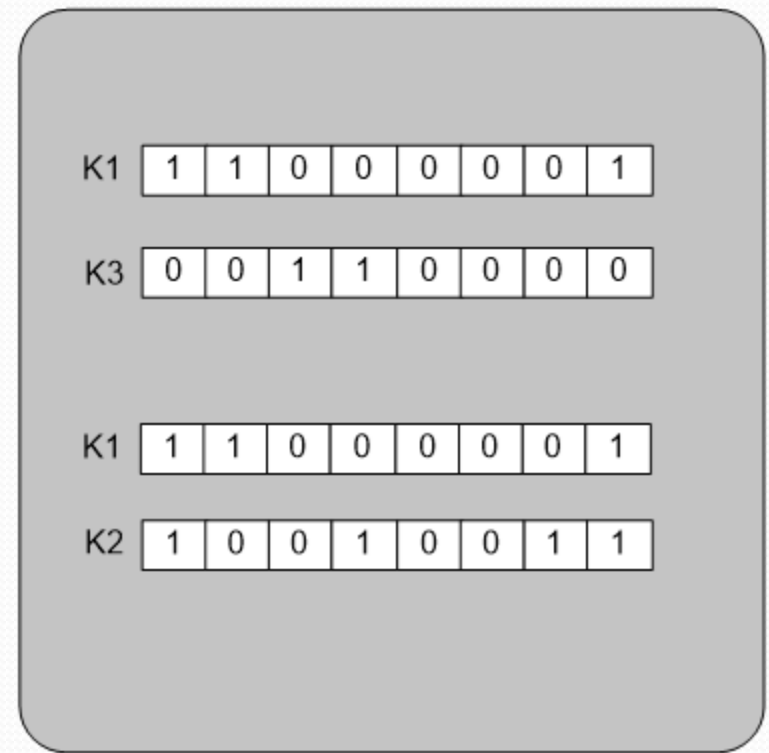
Seleksi Ortu: generasi 1

No	Kromosom	Individu (x_1, x_2)	<i>Fitness</i> ($7x_1 - 3x_2$)	Probabilitas terpilih	Jumlah yang diharapkan di <i>mating pool</i>	Jumlah aktual di <i>mating</i> <i>pool</i>
1	11000001	(12, 1)	81	0,41	1,64	2
2	10010011	(9, 3)	54	0,27	1,08	1
3	00110000	(3, 0)	21	0,11	0,44	1
4	10001001	(8, 5)	41	0,21	0,84	0
Jumlah			197	1,00	4,00	4
Rata-rata			49,25	0,24	1,00	1
Maksimum			81	0,41	1,64	2

Generasi 1



Populasi dengan 4 kromosom: K1 sampai K4



Mating pool: [K1, K3] dan [K1, K2]

Rekombinasi Ortu: generasi 1

No	Kromosom orangtua	Posisi titik rekombinasi	Kromosom anak hasil rekombinasi	Individu anak (x_1, x_2)	<i>Fitness</i> $(7x_1 - 3x_2)$
1	11 000001	2	11110000	(15, 0)	105
2	00 110000	2	00000001	(0, 1)	-3
3	110000 01	6	11000011	(12, 3)	75
4	100100 11	6	10010001	(9, 1)	60
Jumlah					244
Rata-rata					61
Maksimum					105

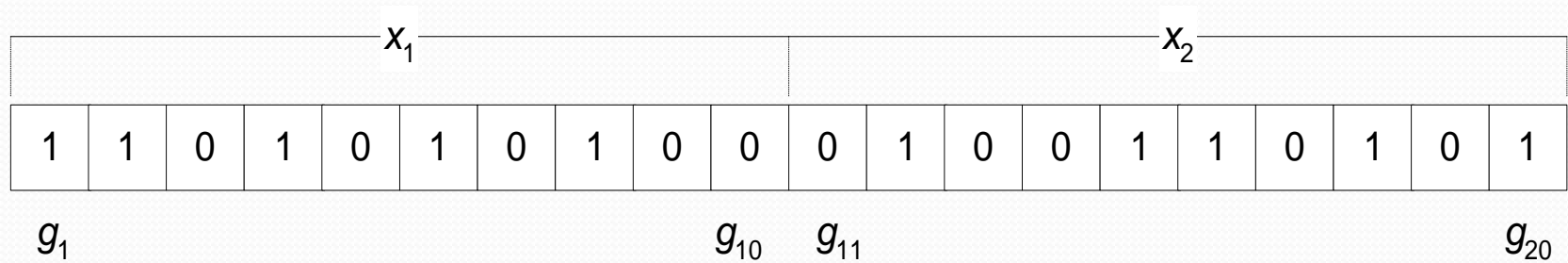
Studi kasus 2: Minimasi

Nilai minimum $h = ?$

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

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

Pengkodean Individu \rightarrow kromosom



Fitness

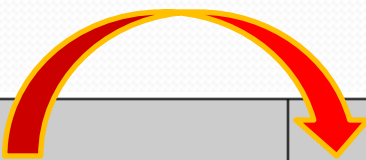
$$f = \frac{1}{(x_1^2 + x_2^2) + 0,01}$$

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

Generasi 1

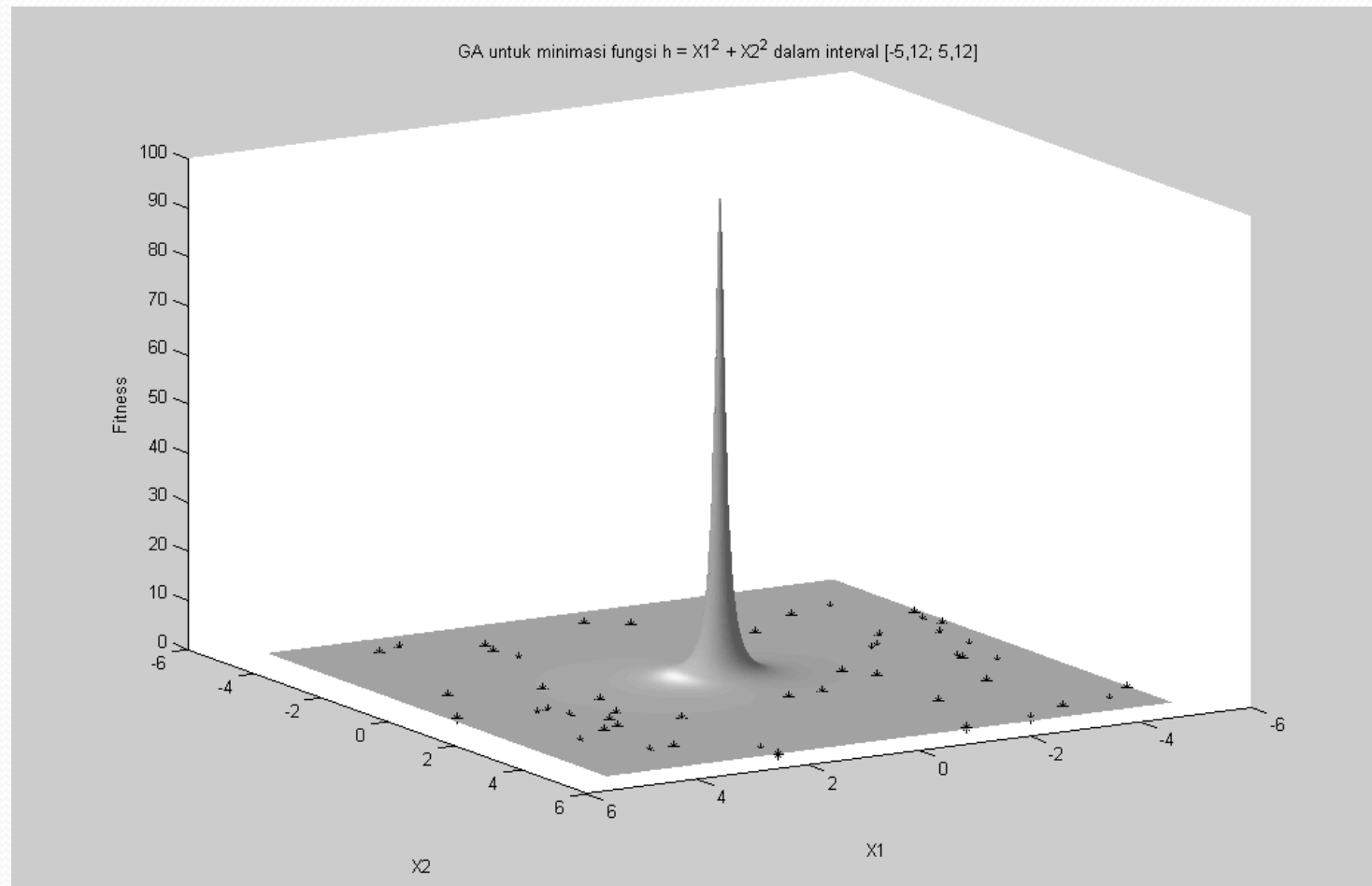
$$x = r_b + \frac{(r_a - r_b)}{\sum_{i=1}^N 2^{-i}} (g_1 \cdot 2^{-1} + g_2 \cdot 2^{-2} + \dots + g_N \cdot 2^{-N})$$

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



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

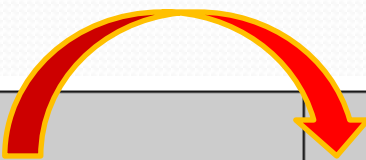
Generasi 1



Generasi 10

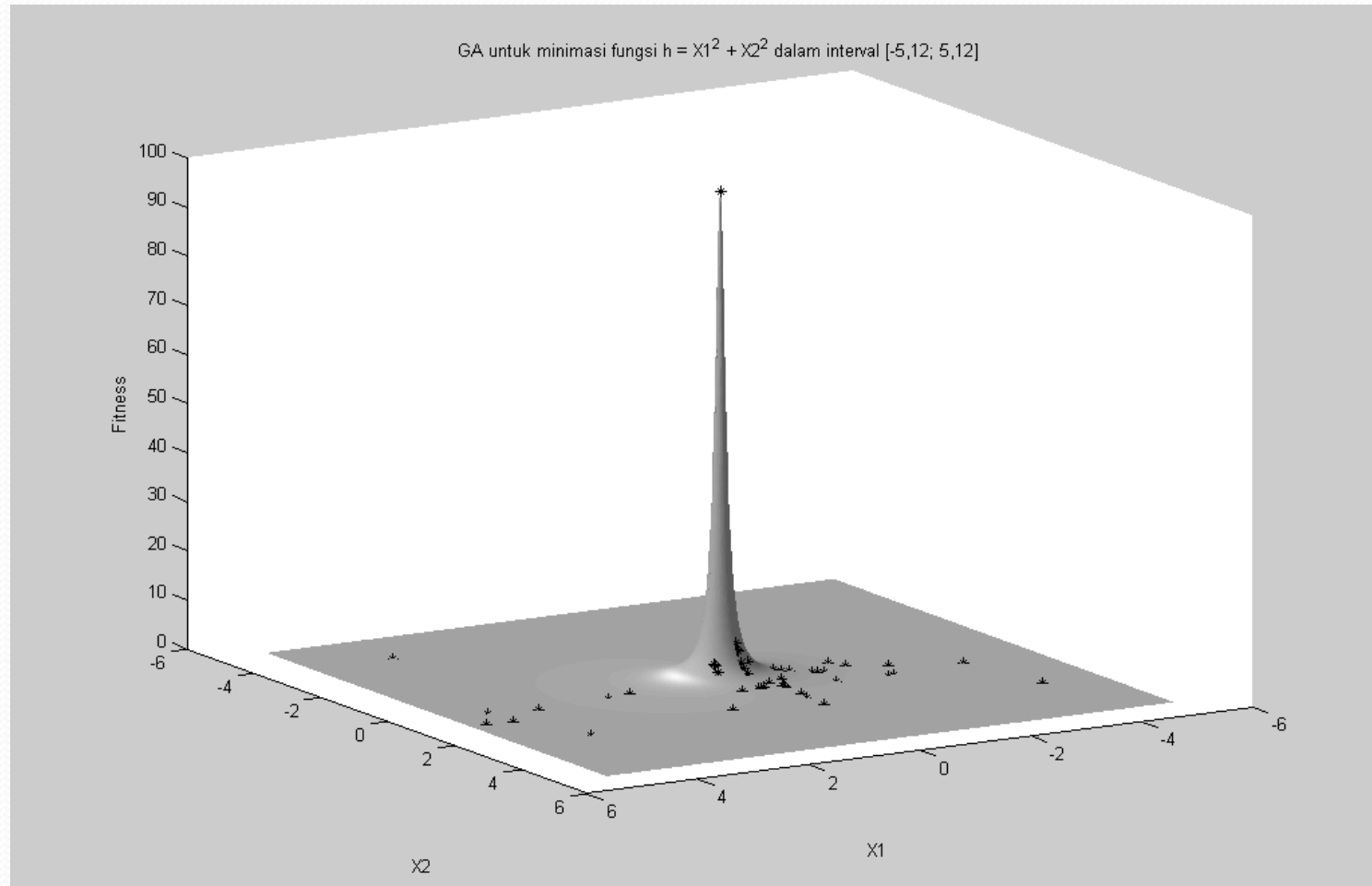
$$x = r_b + \frac{(r_a - r_b)}{\sum_{i=1}^N 2^{-i}} (g_1 \cdot 2^{-1} + g_2 \cdot 2^{-2} + \dots + g_N \cdot 2^{-N})$$

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

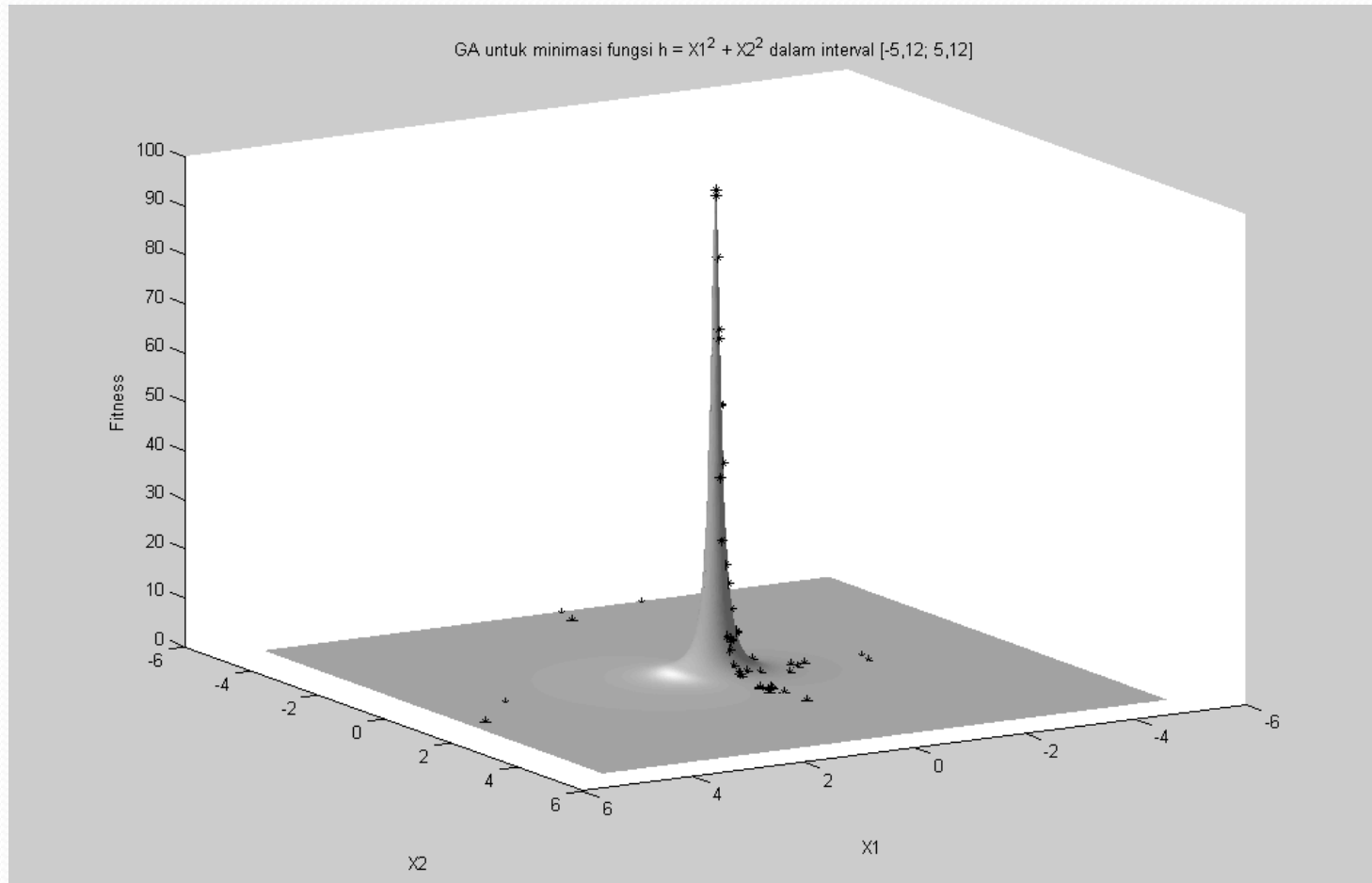


No	Genotype	Phenotype		Nilai fitness
	kromosom biner	X1	X2	
1	0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0	-0.01	0	99.01
2	0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0	-0.01	0	99.01
3	0 1 1 1 1 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1	-3.77	1.03	0.065429
4	0 1 1 1 1 1 0 1 0 1 1 0 0 1 1 1 0 0 0 1	-2.4	1.2	0.1387
5	0 1 1 1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

Generasi 10



Generasi 100

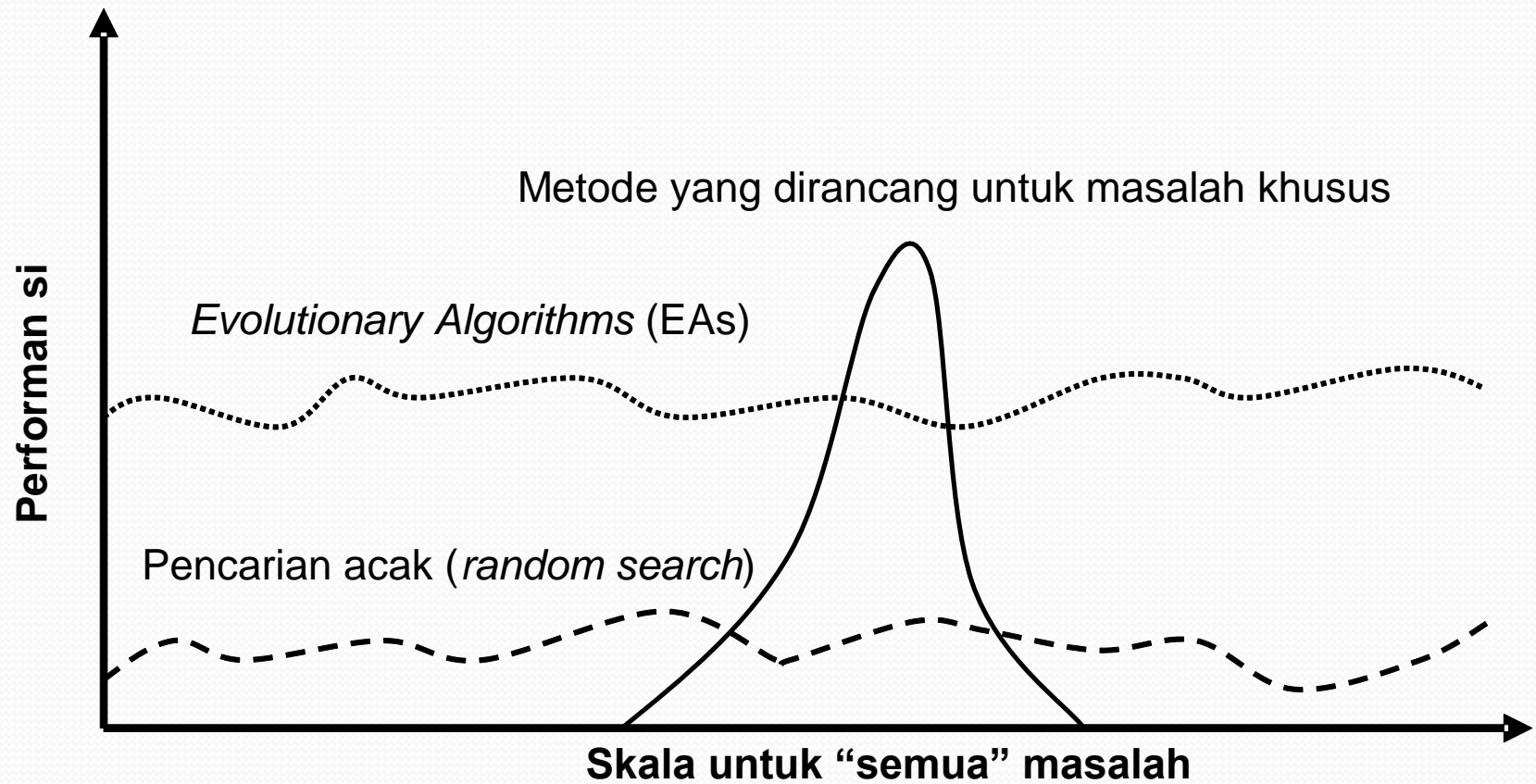


Mengapa EAs?

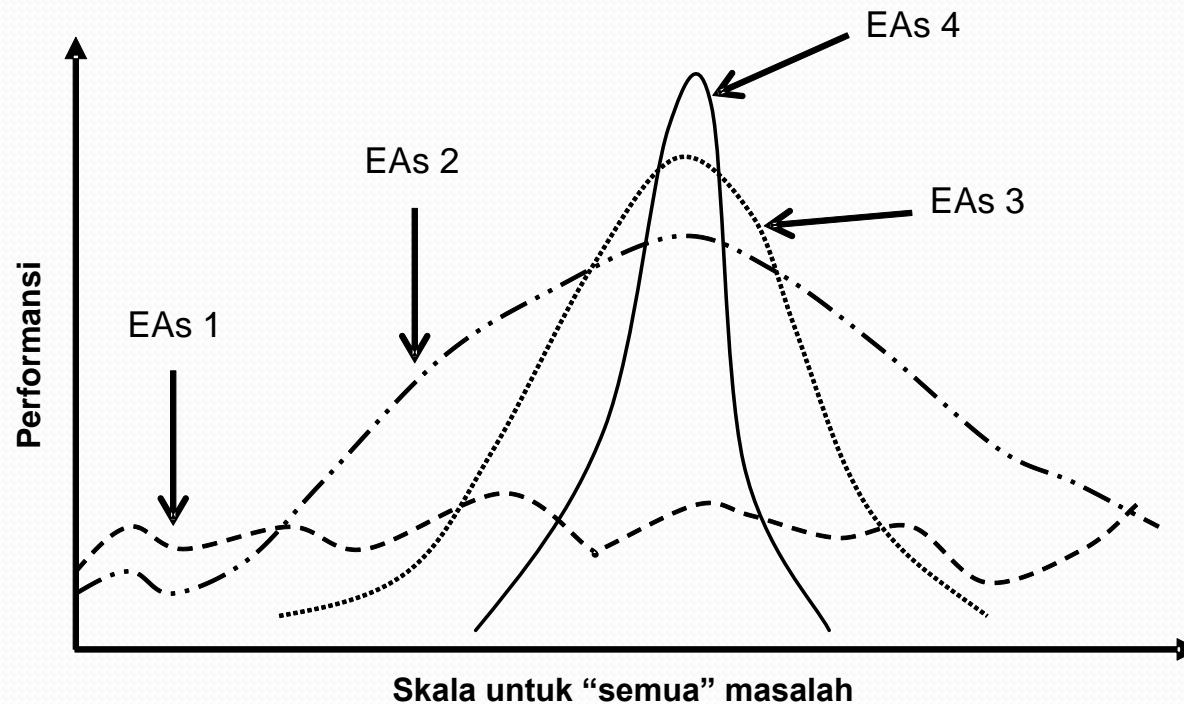
EAs sangat baik untuk permasalahan yang memiliki satu atau lebih ciri-ciri berikut ini:

- Ruang masalah sangat besar, kompleks, dan sulit dipahami;
- Tidak bisa diselesaikan menggunakan metode-metode konvensional;
- Terdapat batasan waktu, misalnya dalam sistem waktu nyata (*real time system*).
- Solusi yang diharapkan tidak harus paling optimal, tetapi 'bagus' atau bisa diterima;
- Kurang atau bahkan tidak ada pengetahuan yang memadai untuk merepresentasikan masalah ke dalam ruang pencarian yang lebih sempit;
- Tidak tersedia analisa matematika yang memadai;

Performansi EAs (Goldberg, 1989)



Performansi EAs (Michalewicz, 1996)



EAs 2 EAs 3, dan EAs 4 adalah EAs yang ditambahkan pengetahuan khusus yang memiliki akurasi lebih baik dibandingkan EAs 1 (tanpa pengetahuan).

Kapan EAs digunakan?

- Jika kita menghadapi masalah TSP untuk *graph* asimetris **10 node**, apakah kita harus menggunakan EAs? Jawabannya mungkin saja tidak perlu karena ada algoritma lain (misal Dijkstra atau dynamic programming) yang performansinya lebih baik.
- Tetapi, jika kita menghadapi masalah TSP untuk *graph* asimetris **1000 node**, apakah kita harus menggunakan EAs? Jawabannya mungkin “ya” karena algoritma lain membutuhkan waktu yang sangat lama.

Kesimpulan

- EAs adalah algoritma-algoritma yang mengimplementasikan abstraksi EC
- Terdapat dua variasi *survivor selection* atau *replacement scheme* , yaitu *Steady State* dan *Generational Replacement*.

Kesimpulan

- Jika pengetahuan yang ditambahkan semakin banyak, maka EAs akan memiliki performansi yang baik untuk berbagai masalah.
- Teori terbaru menyatakan bahwa “**menemukan suatu algoritma yang bisa digunakan untuk semua masalah adalah mustahil**”.

Daftar Pustaka

- [THO96] Thomas Bäck. 1996, “Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms”, Oxford University Press, ISBN: 0195099710, January 1996.
- [THO97] Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors, 1997, “Handbook of Evolutionary Computation”, Computational Intelligence Library. Oxford University Press in cooperation with the Institute of Physics Publishing, Bristol, New York, ringbound edition, ISBN: 0750303921, April 1997.
- [SUYo8] Suyanto, 2008, Evolutionary Computation: Komputasi Berbasis “Evolusi” dan “Genetika”, penerbit Informatika Bandung.
- [JULo7] Julie Leung, Keith Kern, Jeremy Dawson, 2007, “Genetic Algorithms and Evolution Strategies”, presentation slides.