

Evolutionary Programming (EP)

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

HP/WA: 0812 845 12345

Intelligence Computing Multimedia (ICM)
Informatics faculty – Telkom University

Intro

- EP diperkenalkan pertama kali oleh D. Fogel pada era 1960-an di Amerika Serikat
- Pada saat itu EP ditujukan untuk menghasilkan suatu bentuk kecerdasan (*intelligence*), dimana *intelligence* dipandang sebagai suatu tingkah laku yang adaptif (*adaptive behaviour*)
- Salah satu syarat dari *adaptive behaviour* adalah kemampuan memprediksi lingkungan yang dinamis

Intro

- Dengan demikian, **kemampuan memprediksi** adalah kunci menuju *intelligence*
- Bagaimana membangun kemampuan untuk memprediksi?
- D. Fogel mengusulkan EP yang menggunakan *Finite State Machines* (**FSM**) untuk menghasilkan suatu sistem *machine learning* yang memiliki kemampuan memprediksi

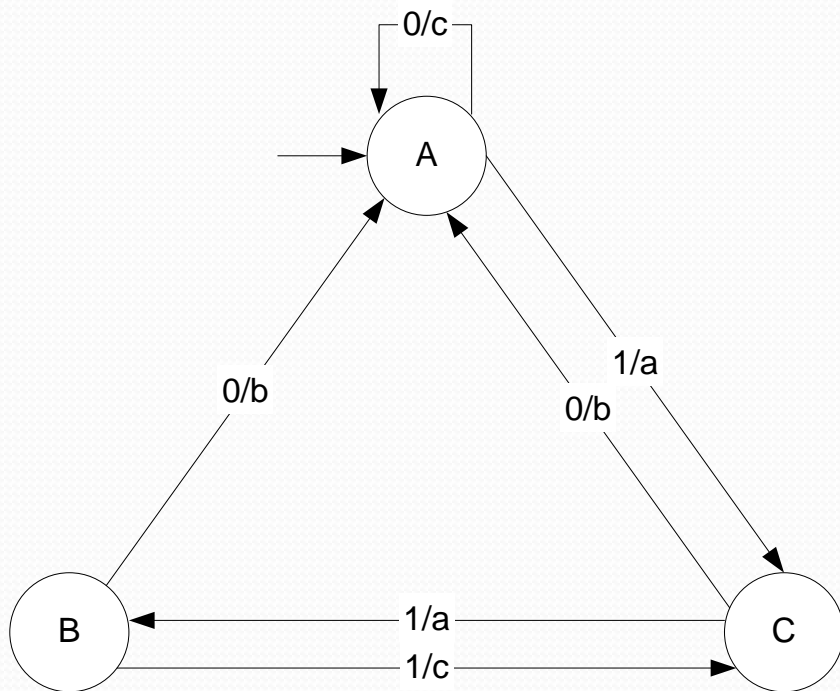
Intro

- FSM adalah mesin dengan sejumlah terbatas keadaan yang terdiri dari:
 - States (keadaan) S
 - Inputs I
 - Outputs O
 - Fungsi transisi $\delta : S \times I \rightarrow S \times O$

Intro

- FSM berfungsi mentransformasikan deretan input ke dalam deretan output.
- Dengan demikian, FSM dapat digunakan untuk memprediksi.
- Misal: “Jika inputnya x , maka outputnya y ”.

FSM



Contoh FSM

- Tiga state $S = \{A, B, C\}$
- Dua nilai input $I = \{0, 1\}$
- Tiga output $O = \{a, b, c\}$
- A adalah *initial state*
- Fungsi transisi digambarkan sebagai garis dengan tanda panah yang menunjukkan perpindahan dari suatu state ke state lainnya dengan menerima suatu input dan memberikan suatu output.

Kriteria FSM yang baik

1. Bisa memetakan sederetan input menjadi sederetan output dengan tingkat akurasi yang tinggi.
2. Memiliki jumlah state yang minimum.

FSM

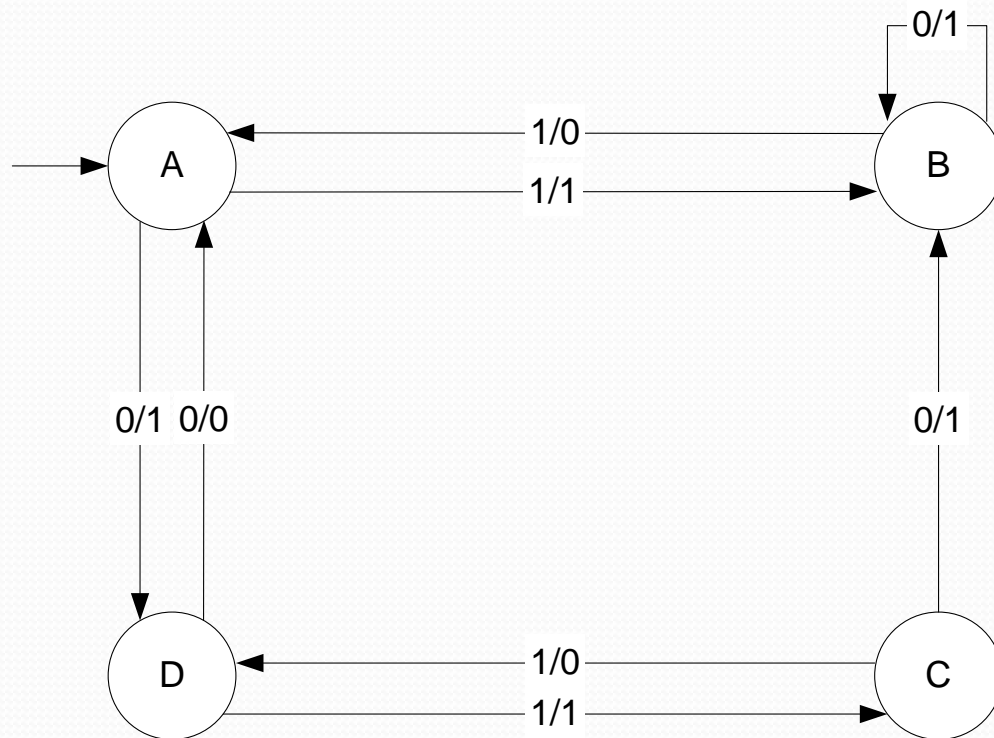
Input : 011010001

Output: 10011110

FSM yang bisa memetakan input-output tsb?

Aplikasi dunia nyata?

Prediksi data time series



FSM

- Pada FSM tersebut, A adalah *initial state*.
- FSM tersebut bisa memetakan setiap I_{i+1} menjadi O_i dengan akurasi 100%.
- Tetapi, mungkin terdapat FSM lain dengan jumlah *state* lebih sedikit dan memiliki akurasi 100% juga.

FSM

- Mengapa memilih FSM dengan minimum state?
- Jumlah state yang minimum tentu saja lebih efisien
- Jumlah state terlalu banyak → mungkin *overfit*

EP

- Pada perkembangannya, EP justru mengarah pada ES
- Kromosom Real
- Sedikit Berbeda dengan ES

Spesifikasi teknis EP

Representasi	Vektor bernilai real
Seleksi orangtua	Deterministik
Rekombinasi	Tidak ada
Mutasi	<i>Gaussian perturbation</i>
Seleksi <i>survivor</i>	Probabilistik ($\mu + \mu$)
Ciri khusus	<i>Self-adaptation</i> pada <i>mutation step sizes</i> (pada Meta-EP)

Representasi Individu

Kromosom yang lengkap pada EP hanya terdiri dari dua bagian, yaitu:

- Variabel objek: x_1, \dots, x_n
- Mutation step sizes: $\sigma_1, \dots, \sigma_n$

Kromosom

$$\langle x_1, \dots, x_n, \sigma_1, \dots, \sigma_n \rangle$$

- n adalah jumlah variabel pada fungsi yang dioptimasi.
- Variabel objek mengkodekan nilai-nilai real secara langsung tanpa konversi.

Seleksi Orangtua

- Pada EP, setiap orangtua menghasilkan satu anak melalui proses mutasi (tanpa rekombinasi).
- Jadi, sejumlah μ orangtua menghasilkan sejumlah μ anak.
- Dengan demikian semua individu dalam populasi pasti terpilih sebagai orangtua tanpa memperhatikan nilai *fitness*-nya.
- Dengan kata lain, seleksi orangtua pada EP bersifat deterministik atau pasti tanpa dipengaruhi *fitness*-nya.
- Hal ini berbeda dengan seleksi orangtua pada GA yang bersifat probabilistik menggunakan metode *roulette wheel*, *tournament*, atau lainnya.

Rekombinasi

- Pada EP tidak terdapat rekombinasi
- Mengapa?
- Karena satu titik pada ruang pencarian berlaku sebagai suatu spesies (bukan sebagai suatu individu seperti pada GA maupun ES), sedangkan rekombinasi tidak mungkin dilakukan antar spesies
- Banyak perdebatan mengenai mutasi dan rekombinasi
- Mana yang lebih penting?
- Apakah keduanya sama-sama penting?
- Bisakah hanya menggunakan salah satu saja?

Mutasi

- Pada EP, mutasi merupakan satu-satunya proses untuk menghasilkan kromosom baru
- Mutasi dilakukan dengan cara mengubah nilai gen dengan menambahkan bilangan random yang dibangkitkan berdasarkan distribusi normal
- Proses mutasinya hampir sama dengan mutasi yang digunakan pada ES

Mutasi

- Suatu kromosom $\langle x_1, \dots, x_n, \sigma_1, \dots, \sigma_n \rangle$ bisa menghasilkan kromosom baru dengan adanya mutasi terhadap σ dan x yang diperoleh menggunakan rumus

$$\sigma'_i = \sigma_i \cdot (1 + \alpha \cdot N(0,1))$$

$$x'_i = x_i + \sigma'_i \cdot N_i(0,1) \quad \begin{array}{l} \sigma'_i = \sigma_i \cdot \exp(\eta \cdot N(0,1) + \tau \cdot N_i(0,1)) \\ x'_i = x_i + \sigma'_i \cdot N_i(0,1) \end{array}$$

dimana α adalah suatu konstanta yang biasanya diset sekitar **0,2**.

Mutasi

- Urutan mutasi merupakan hal yang sangat penting.
- Mutasi terhadap σ harus lebih dulu daripada mutasi terhadap x .
- Jika urutannya dibalik, EP tidak bisa bekerja dengan baik untuk menemukan solusi.

Seleksi *Survivor*

- Misalkan suatu populasi P pada generasi t memiliki μ individu
- Karena seleksi orangtua pada EP adalah deterministik, maka semua individu pasti terpilih menjadi orangtua
- Dengan demikian kejadian ini bisa dilambangkan sebagai $P(t) = \mu$ orangtua

Seleksi *Survivor*

- Selanjutnya, karena setiap orangtua menghasilkan hanya satu anak melalui satu proses mutasi (tanpa rekombinasi), maka dari sejumlah μ orangtua akan menghasilkan sejumlah μ anak
- Sehingga $P'(t)$: μ anak menyatakan suatu populasi bayangan yang berisi sejumlah μ anak
- Pertanyaannya, bagaimana memilih sejumlah μ individu dari gabungan $P(t)$ dan $P'(t)$ untuk hidup pada generasi berikutnya atau $t + 1$?

Seleksi *Survivor*

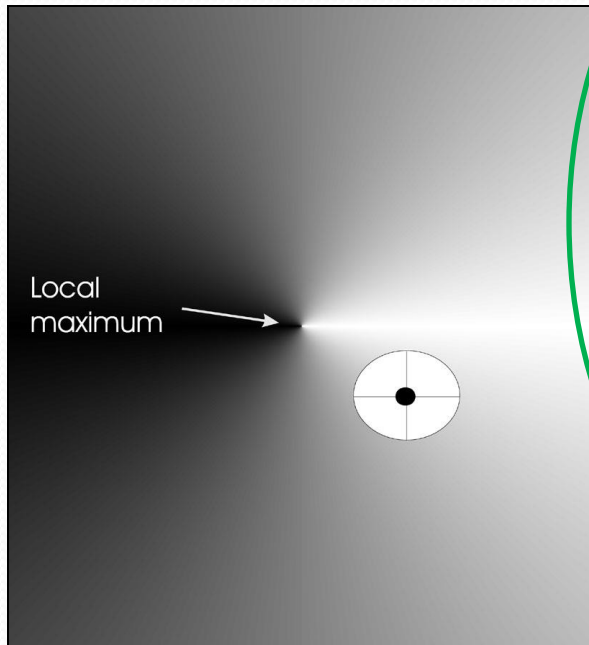
- EP menggunakan *Pairwise Competitions* dalam format *Round-Robin*, dengan cara:
 - Setiap solusi x dari gabungan $P(t) \cup P'(t)$ dievaluasi dengan cara membandingkan x terhadap sejumlah q solusi lain yang dipilih secara acak.
 - Untuk setiap perbandingan, suatu skor "menang" diberikan jika x lebih baik dibandingkan lawannya.
 - Sejumlah μ solusi dengan jumlah "menang" yang paling banyak akan bertahan hidup pada generasi berikutnya ($t + 1$).

Pada cara di atas, parameter q digunakan untuk mengatur *selective pressure*. Biasanya q diset sekitar **10**.

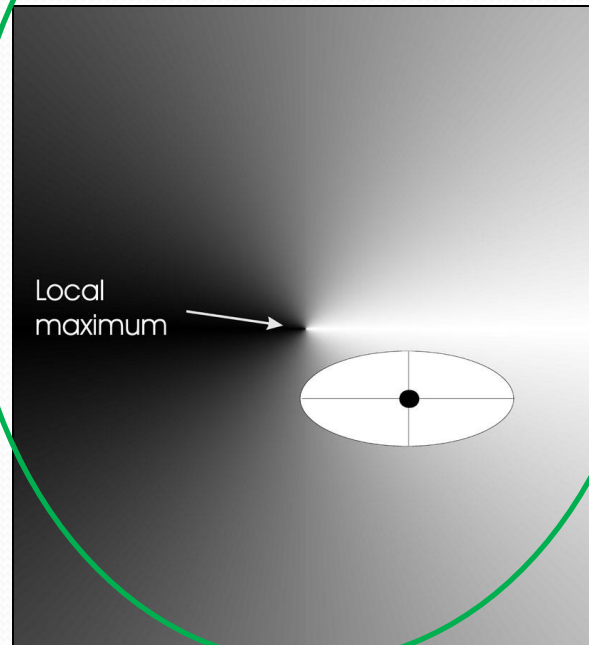
Evolution Strategies

Evolution Programming
(tanpa Rekombinasi)

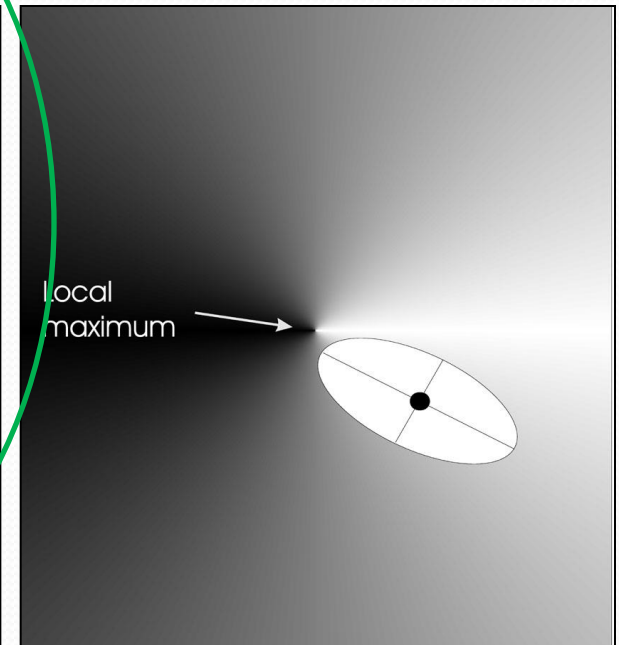
Mutasi tanpa Korelasi
menggunakan satu σ



Mutasi tanpa Korelasi
menggunakan $n \sigma$



Mutasi dengan
Korelasi



$f(x_1, x_2)$

x_2



Congratulations,
it only took you
65298 seconds

Found
It!!!

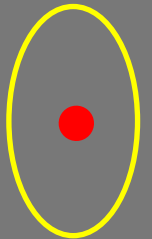
x_1

EP: Mutasi tanpa Korelasi dengan $n \sigma$

Populasi = 1



Global maximum



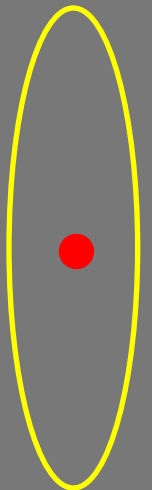
Generasi 1

EP: Mutasi tanpa Korelasi dengan $n \sigma$

Populasi = 1



Global maximum



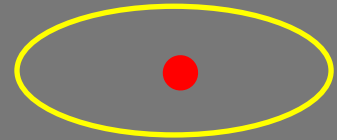
Generasi 2

EP: Mutasi tanpa Korelasi dengan $n \sigma$

Populasi = 1



Global maximum



Generasi 3

EP: Mutasi tanpa Korelasi dengan $n \sigma$

Populasi = 1

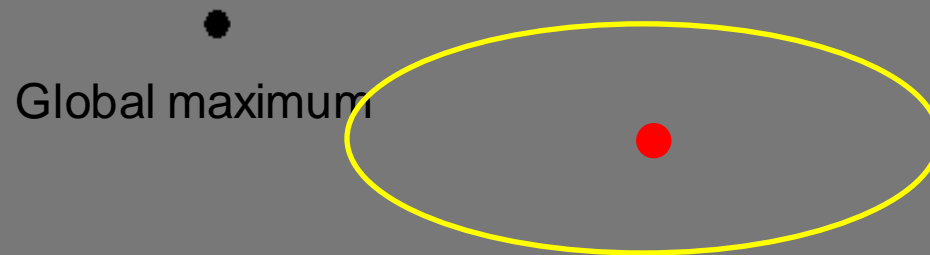


Global maximum

Generasi 10

EP: Mutasi tanpa Korelasi dengan $n \sigma$

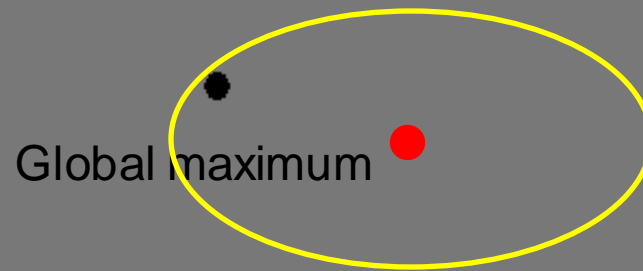
Populasi = 1



Generasi 50

EP: Mutasi tanpa Korelasi dengan $n \sigma$

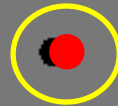
Populasi = 1



Generasi 100

EP: Mutasi tanpa Korelasi dengan $n \sigma$

Populasi = 1



Global maximum

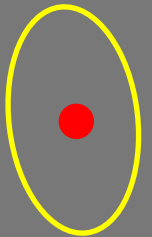
Generasi 200

ES: Mutasi dengan Korelasi

Populasi = 1



Global maximum



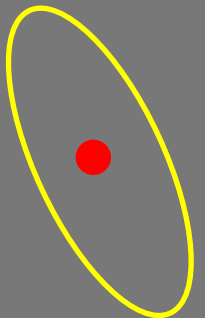
Generasi 1

Mutasi dengan Korelasi

Populasi = 1



Global maximum



Generasi 2

Mutasi dengan Korelasi

Populasi = 1



Global maximum

Generasi 3

Mutasi dengan Korelasi

Populasi = 1

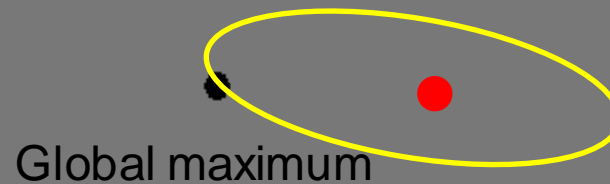


Global maximum

Generasi 10

ES: Mutasi dengan Korelasi

Populasi = 1



Generasi 20

ES: Mutasi dengan Korelasi

Populasi = 1

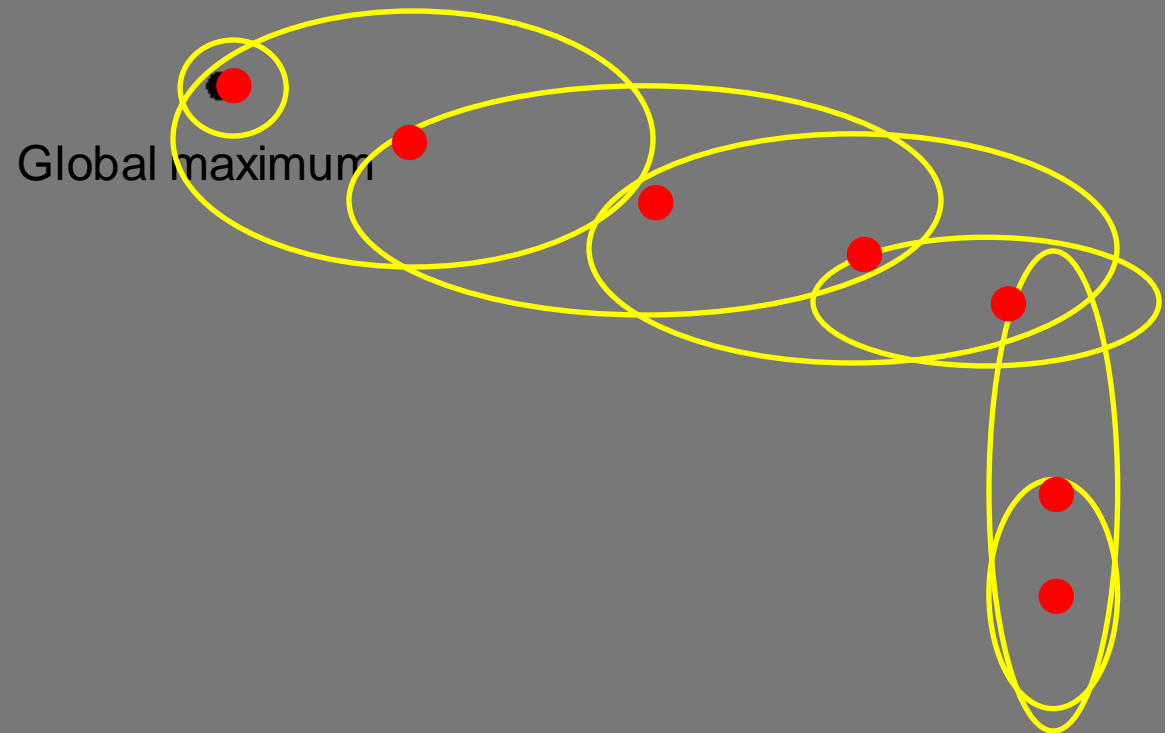


Global maximum

Generasi 50

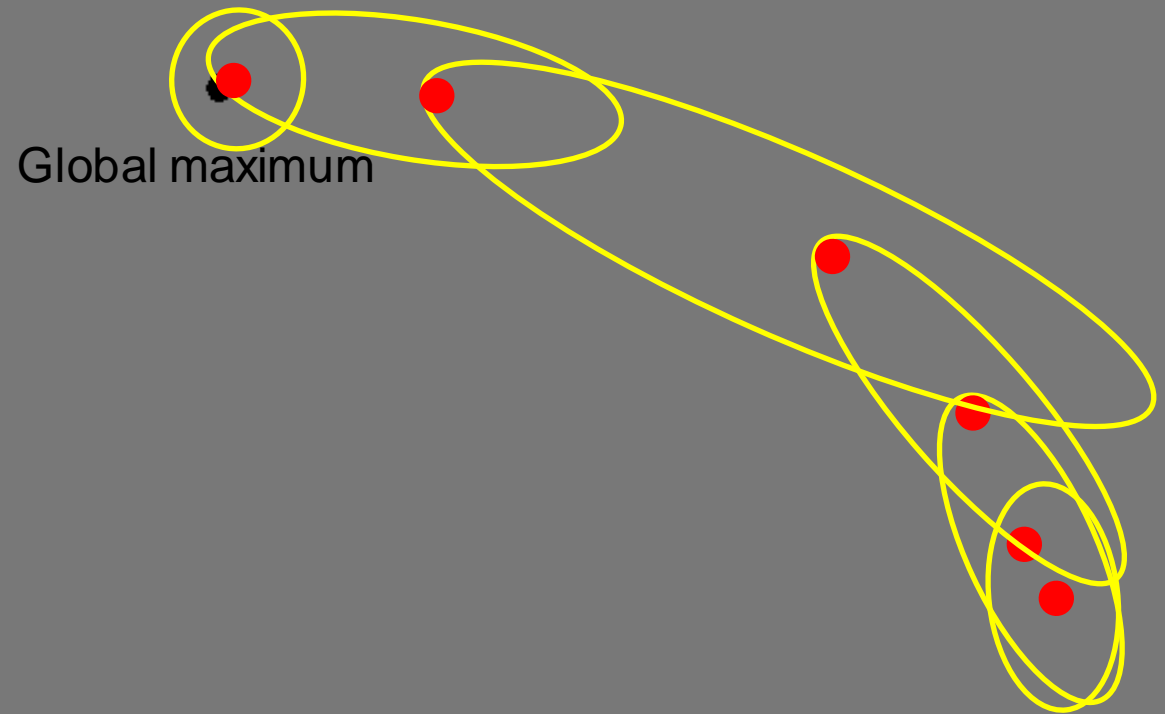
EP: Mutasi tanpa Korelasi dengan $n \sigma$

Populasi = 1

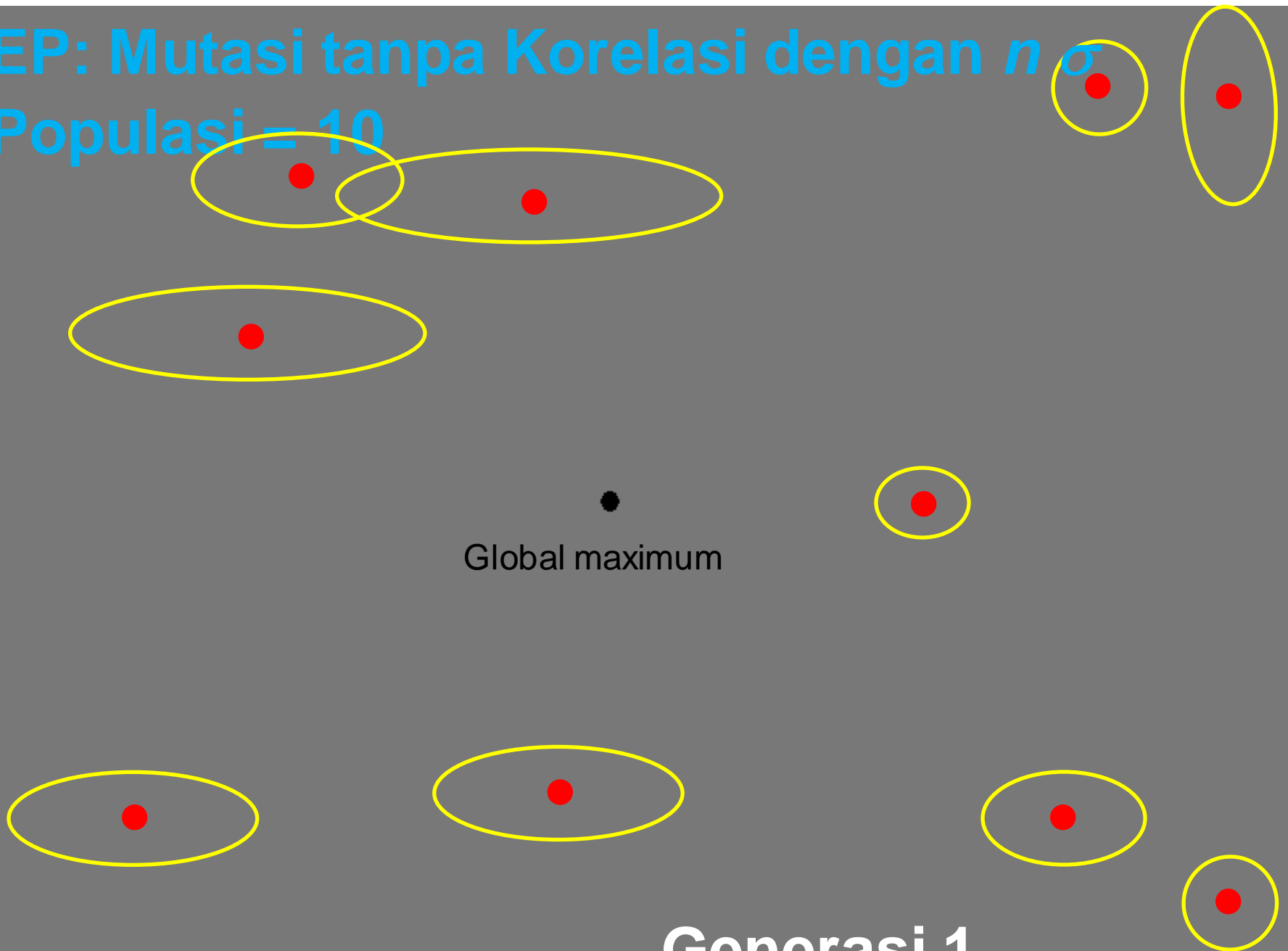


ES: Mutasi dengan Korelasi

Populasi = 1



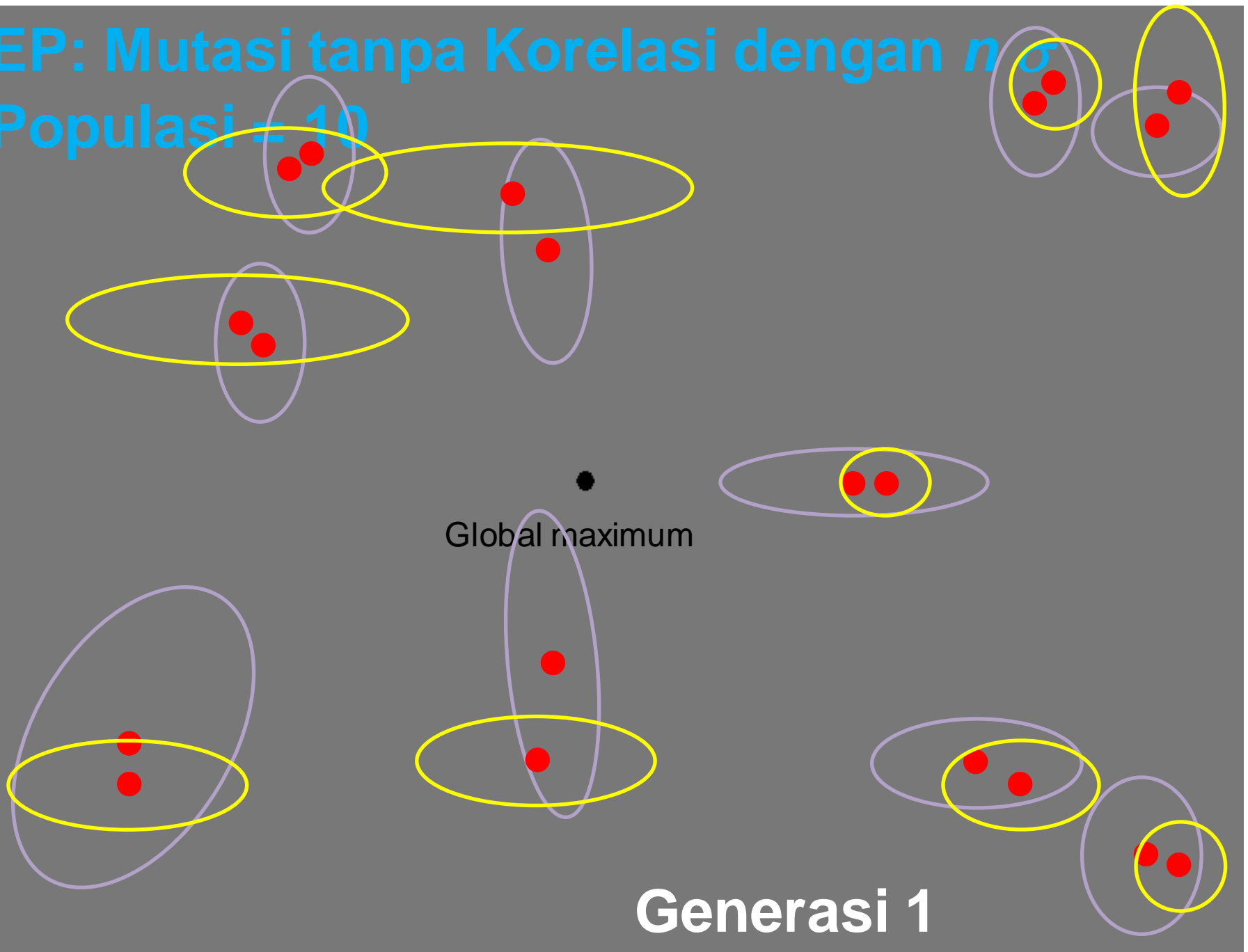
EP: Mutasi tanpa Korelasi dengan $n \sigma$
Populasi = 10



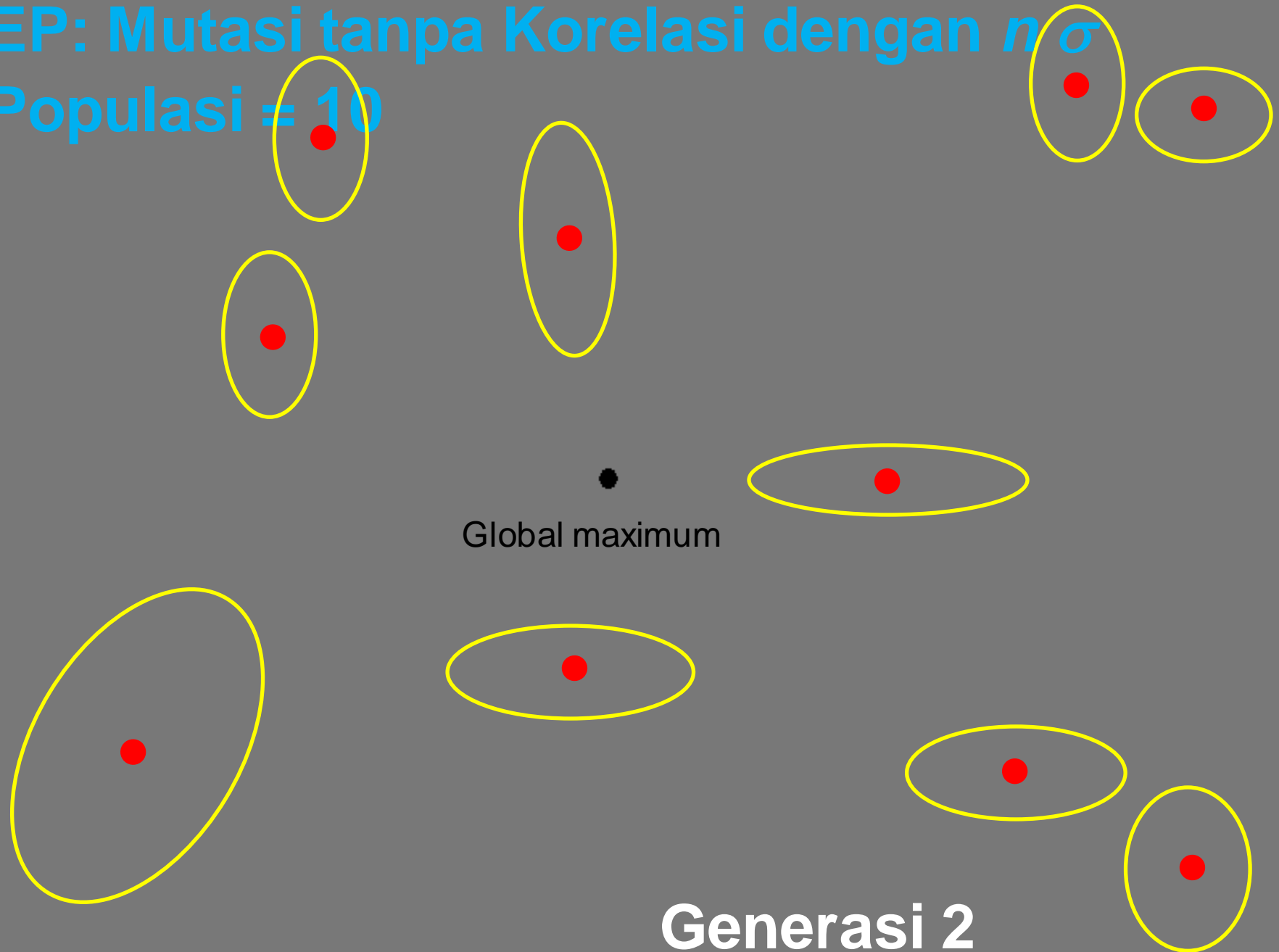
Generasi 1

EP: Mutasi tanpa Korelasi dengan n

Populasi = 10

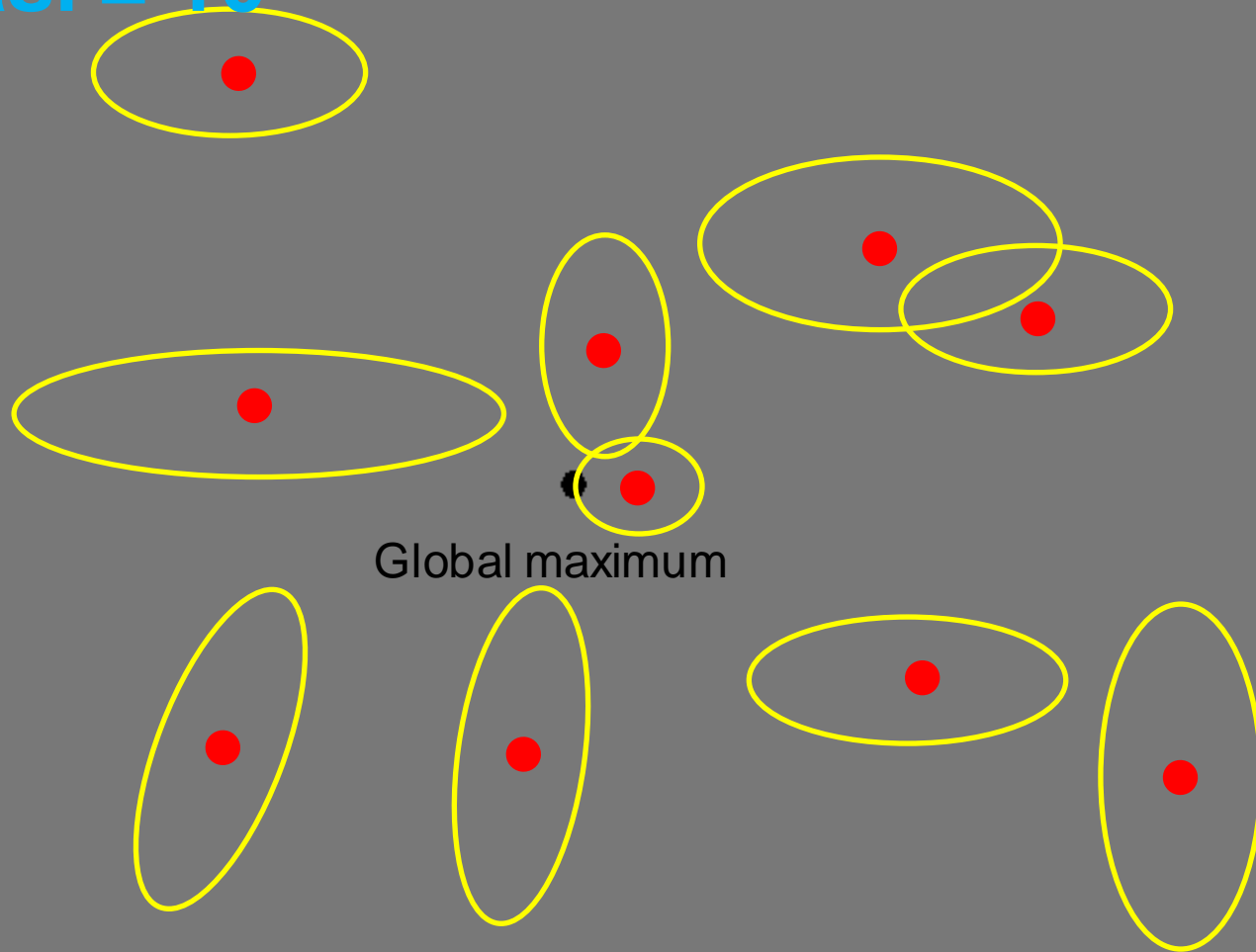


EP: Mutasi tanpa Korelasi dengan $n\sigma$
Populasi = 10



EP: Mutasi tanpa Korelasi dengan $n \sigma$

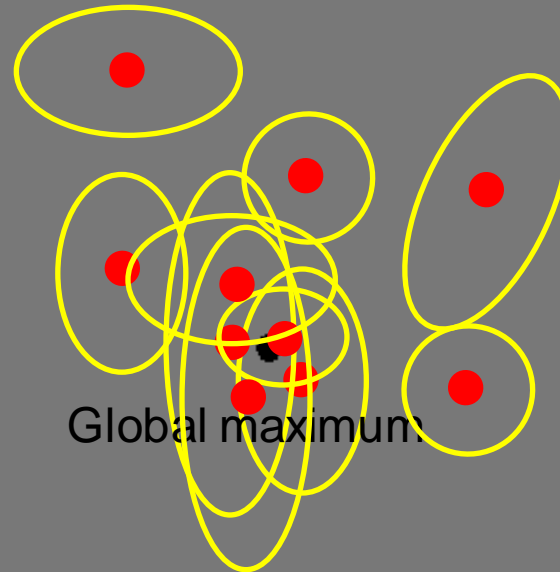
Populasi = 10



Generasi 10

EP: Mutasi tanpa Korelasi dengan $n \sigma$

Populasi = 10



Generasi 50

EP: Mutasi tanpa Korelasi dengan $n \sigma$

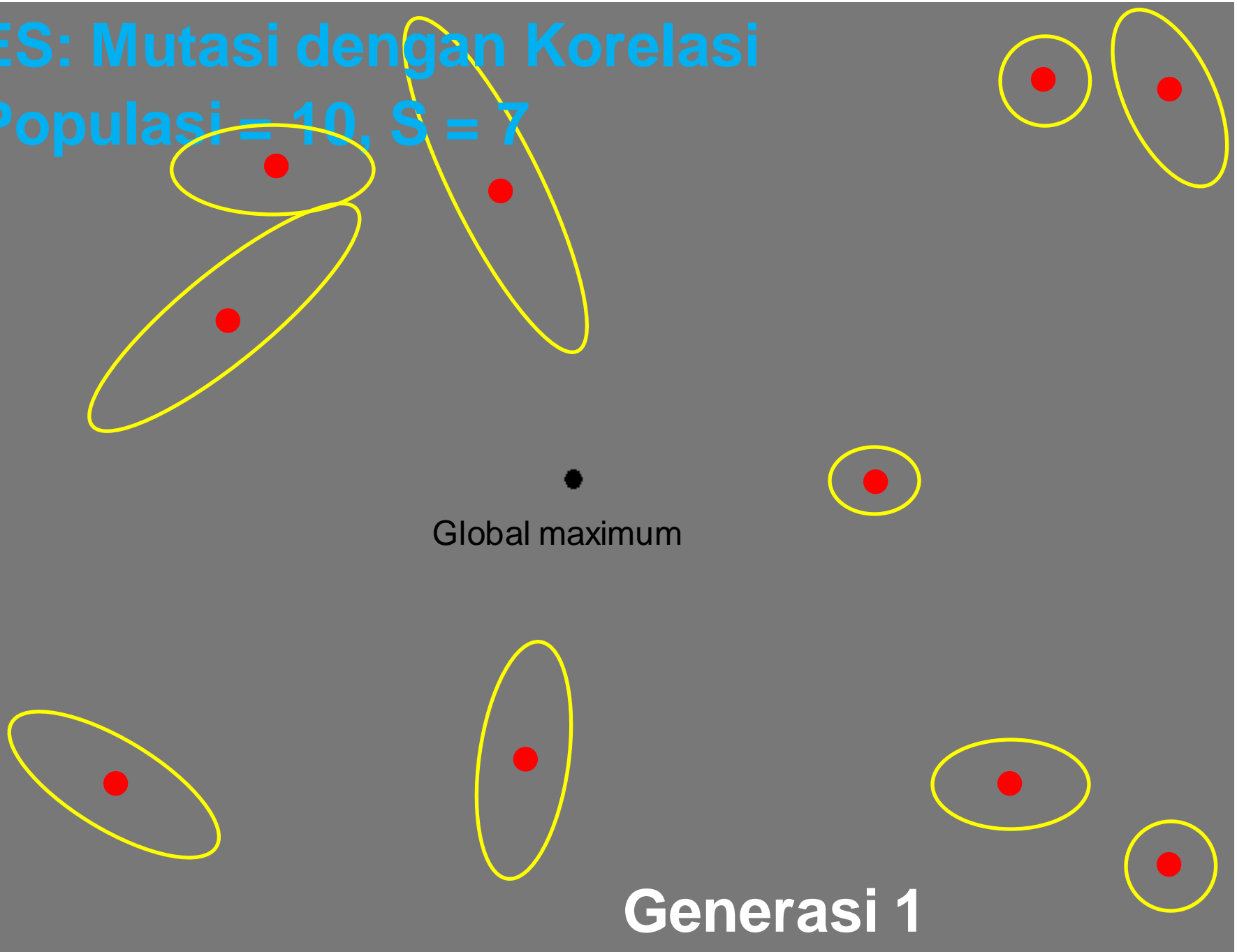
Populasi = 10

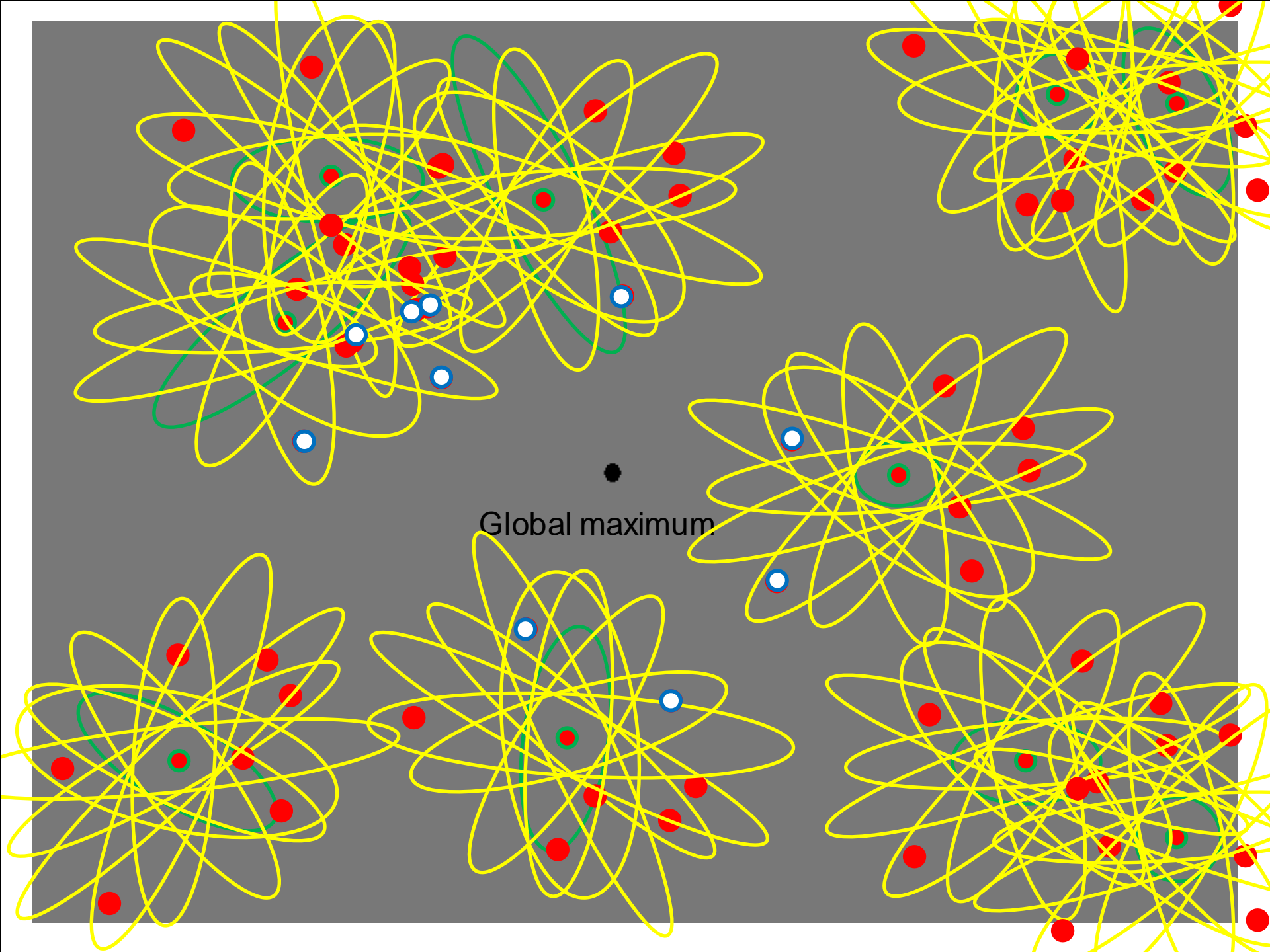


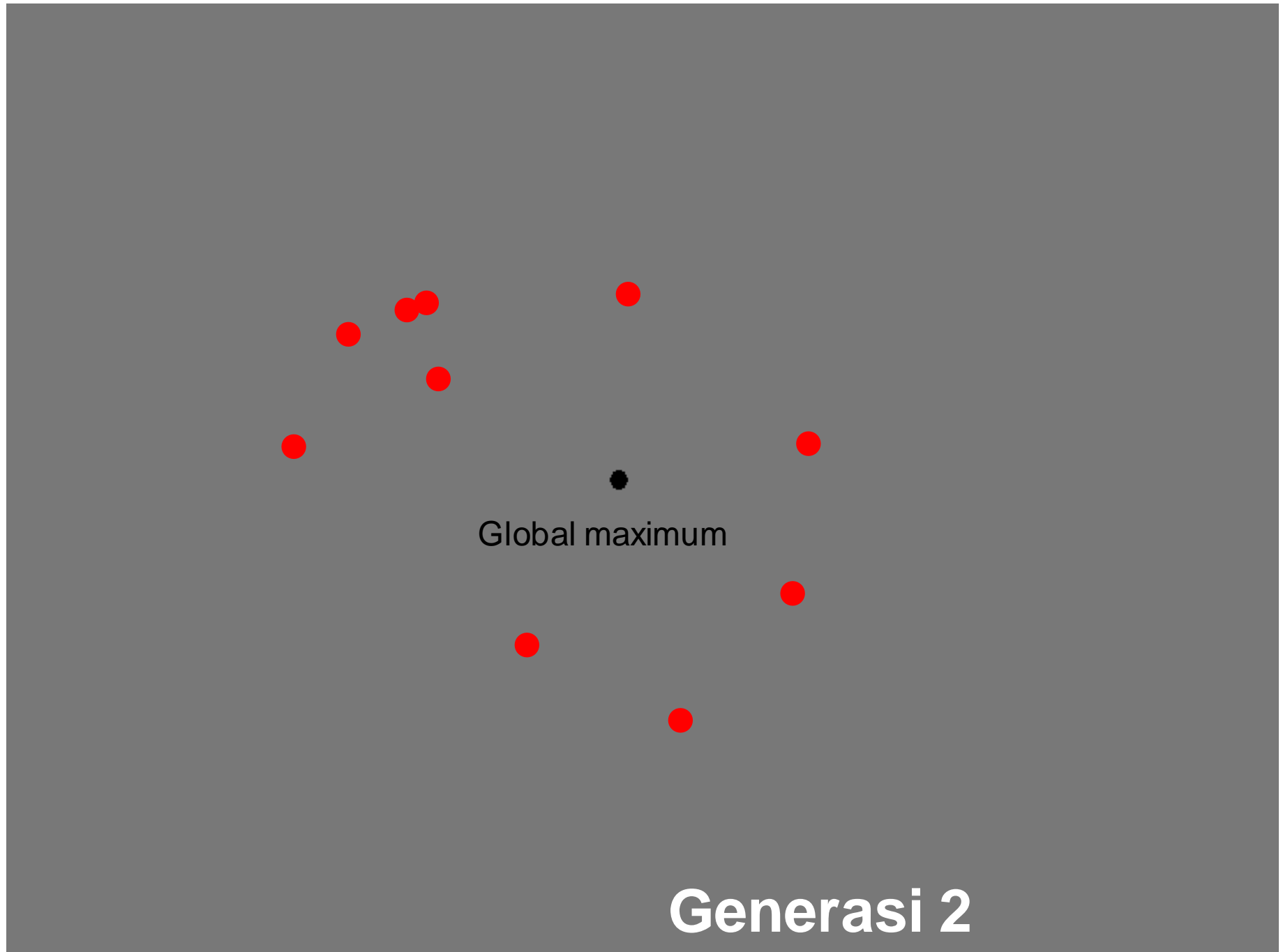
Generasi 100

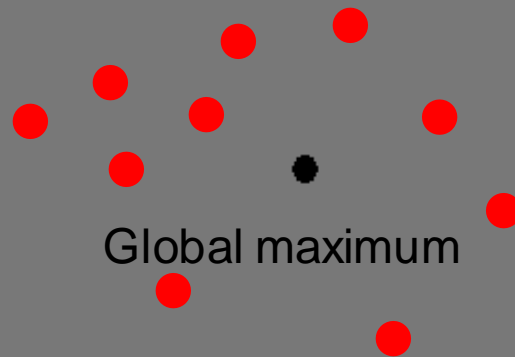
ES: Mutasi dengan Korelasi

Populasi = 10, $S = 7$

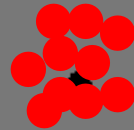








Generasi 10



Global maximum

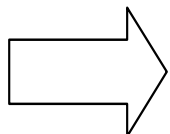
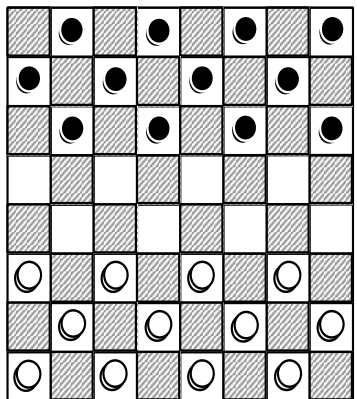
Generasi 20

Aplikasi EP

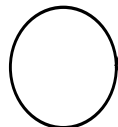
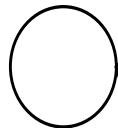
- *evolving game players*
- *training of artificial neural networks*
- *chemistry and biochemistry*
- *electronic and controller design*
- *fuzzy clustering*
- *general constraint optimization*
- *robotic motion control*

Evolving checkers players (Fogel'02)

- Neural nets for evaluating future values of moves are evolved
- NNs have fixed structure with 5046 weights, these are evolved + one weight for “kings”
- Representation:
 - vector of 5046 real numbers for object variables (weights)
 - vector of 5046 real numbers for σ 's
- Mutation:
 - Gaussian, lognormal scheme with σ -first
 - Plus special mechanism for the kings' weight
- Population size 15

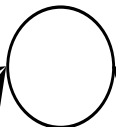


32 Nodes



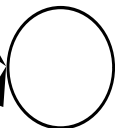
**Input
Layer**

40 Nodes

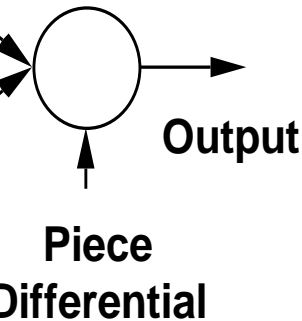


**Hidden Layer
#1**

10 Nodes



**Hidden Layer
#2**



Evolving checkers players (Fogel'02)

- Tournament size $q = 5$
- Programs (with NN inside) play against other programs, no human trainer or hard-wired intelligence
- After 840 generation (6 months!) best strategy was tested against humans via Internet
- Program earned “expert class” ranking outperforming 99.61% of all rated players

Kesimpulan

- Pada awalnya, EP untuk membangun FSM
- Tetapi, dalam perkembangannya EP justru menyerupai ES
- EP hanya menggunakan Mutasi untuk menghasilkan kromosom anak

Daftar Pustaka

- [SUYo8] Suyanto, 2008, Evolutionary Computation: Komputasi Berbasis “Evolusi” dan “Genetika”, penerbit Informatika Bandung.