



**TEKNIK INFORMATIKA**  
FAKULTAS TEKNIK UNIVERSITAS MATARAM

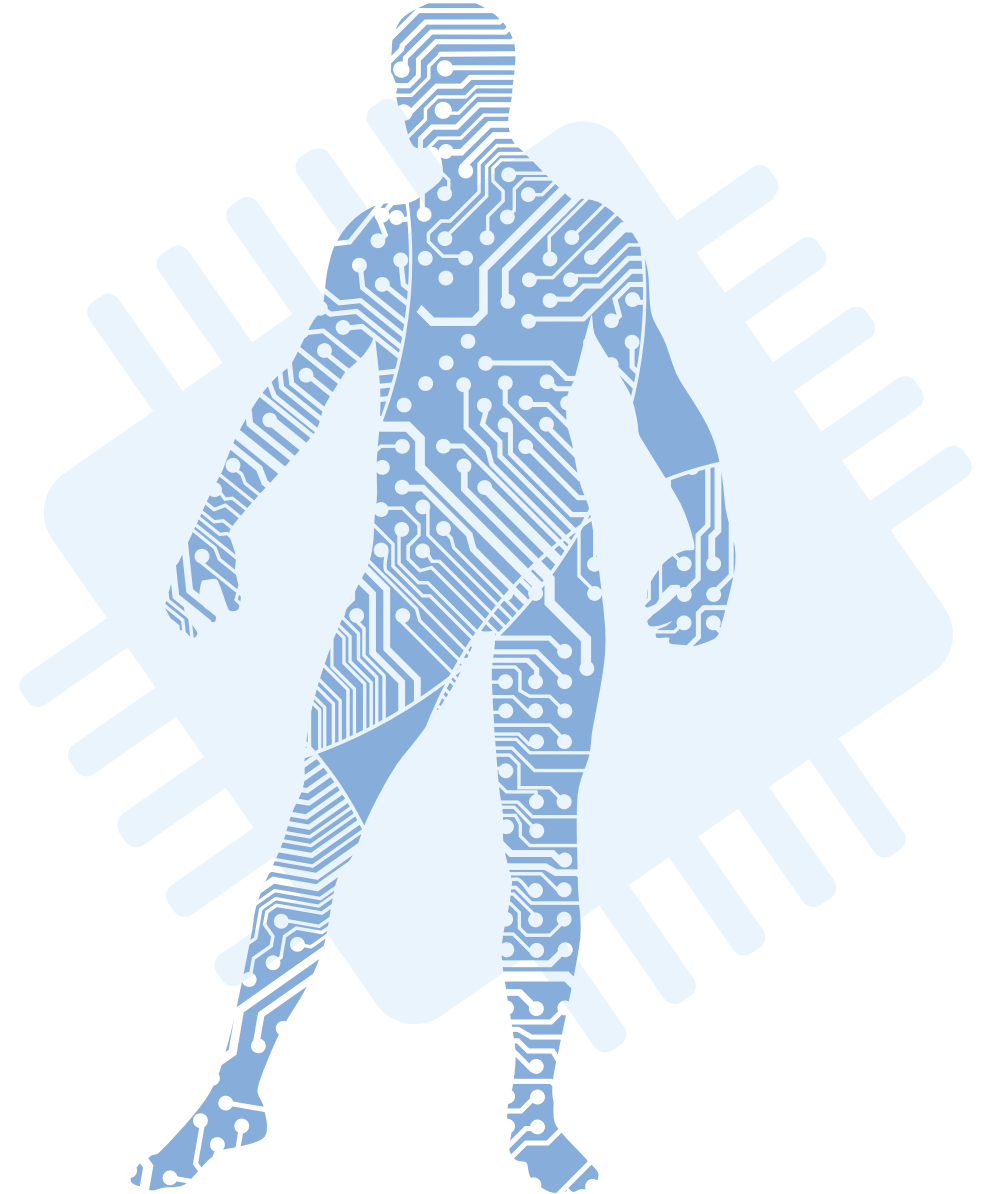


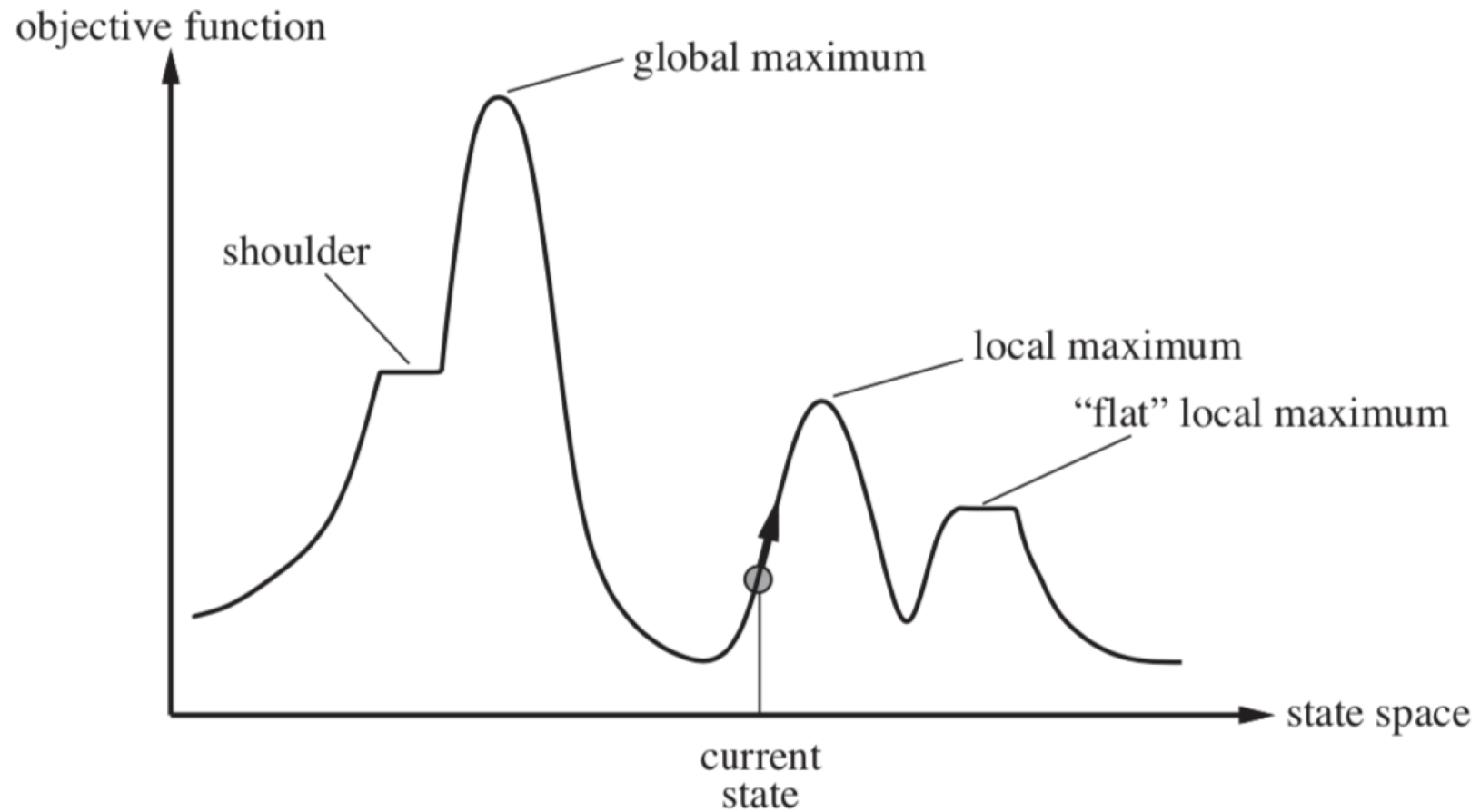
# Heuristic Search (Random Walk)

Ramaditia D

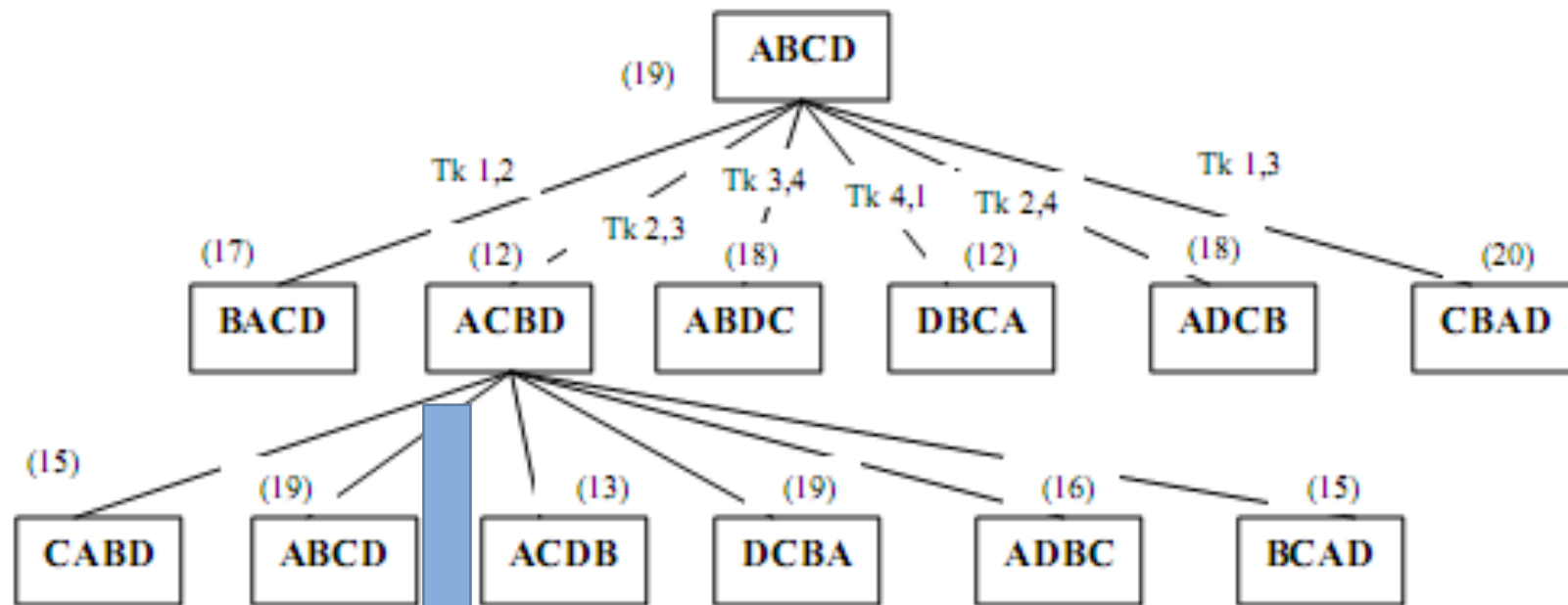
# Outline

- 01** Random-restart HC
- 02** Simulated Annealing
- 03** Genetic Algorithm

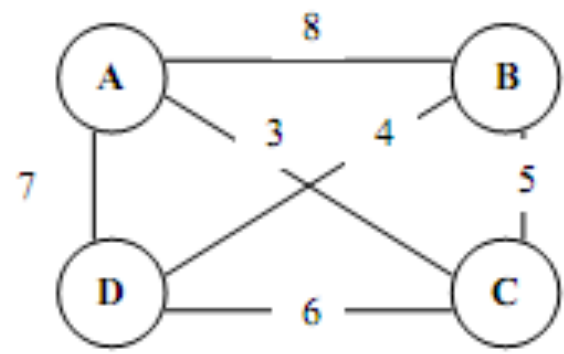




- Random-restart HC merupakan metode pencarian random untuk mengoptimalkan metode pencarian heuristik yang berpotensi menimbulkan hasil yang kurang optimal akibat adanya permasalahan optimum local.
- Secara sederhana, analogi *Random-restart HC* adalah metode *Hill Climbing* yang dilengkapi dengan fitur inisialisasi ulang di tempat random.



Simpan, lalu Random Ulang  
Hingga batas tertentu

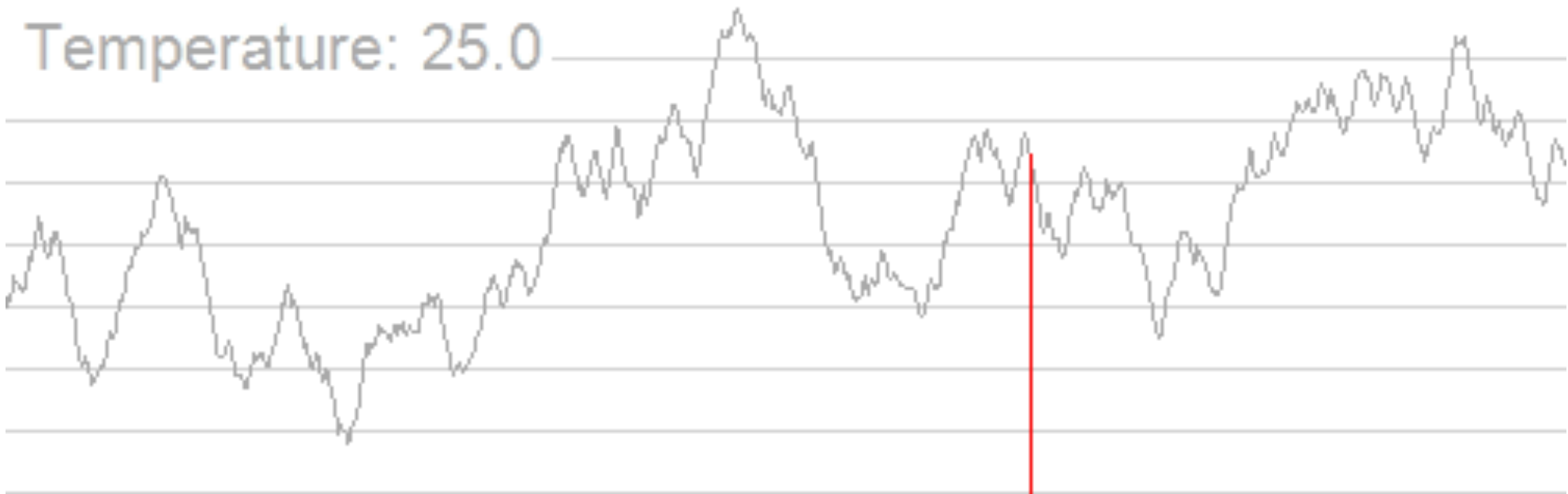


- Merupakan pencarian random variasi dari Random-restart HC, terdapat parameter tertentu di dalam pemilihan tempat pada metode SA.
- Berbasiskan probabilitas dan mekanika statistik, algoritme ini dapat digunakan untuk mencari pendekatan terhadap solusi optimum global dari suatu permasalahan. Masalah yang membutuhkan pendekatan SA adalah masalah-masalah optimisasi kombinatorial, di mana ruang pencarian solusi yang ada terlalu besar, sehingga hampir tidak mungkin ditemukan solusi eksak terhadap permasalahan itu.
- Annealing adalah satu teknik yang dikenal dalam bidang metalurgi, digunakan dalam mempelajari proses pembentukan kristal dalam suatu materi.



# Simulasi SA

7



Simulated annealing searching for a maximum. The objective here is to get to the highest point; however, it is not enough to use a simple hill climb algorithm, as there are many local maxima. By cooling the temperature slowly the global maximum is found.

## Algorithm SIMULATED-ANNEALING

### Begin

*temp* = INIT-TEMP;

*place* = INIT-PLACEMENT;

**while** (*temp* > FINAL-TEMP) **do**

**while** (*result\_criterion* = FALSE) **do**

*new\_place* = CLIMB(*place*);

$\Delta C = \text{COST}(\text{new\_place}) - \text{COST}(\text{place});$

**if** ( $\Delta C < 0$ ) **then**

*place* = *new\_place*;

**else if** ( $\text{RANDOM}(0,1) < e^{-(\Delta C/\text{temp})}$ ) **then**

*place* = *new\_place*;

*temp* = SCHEDULE(*temp*);

**End.**



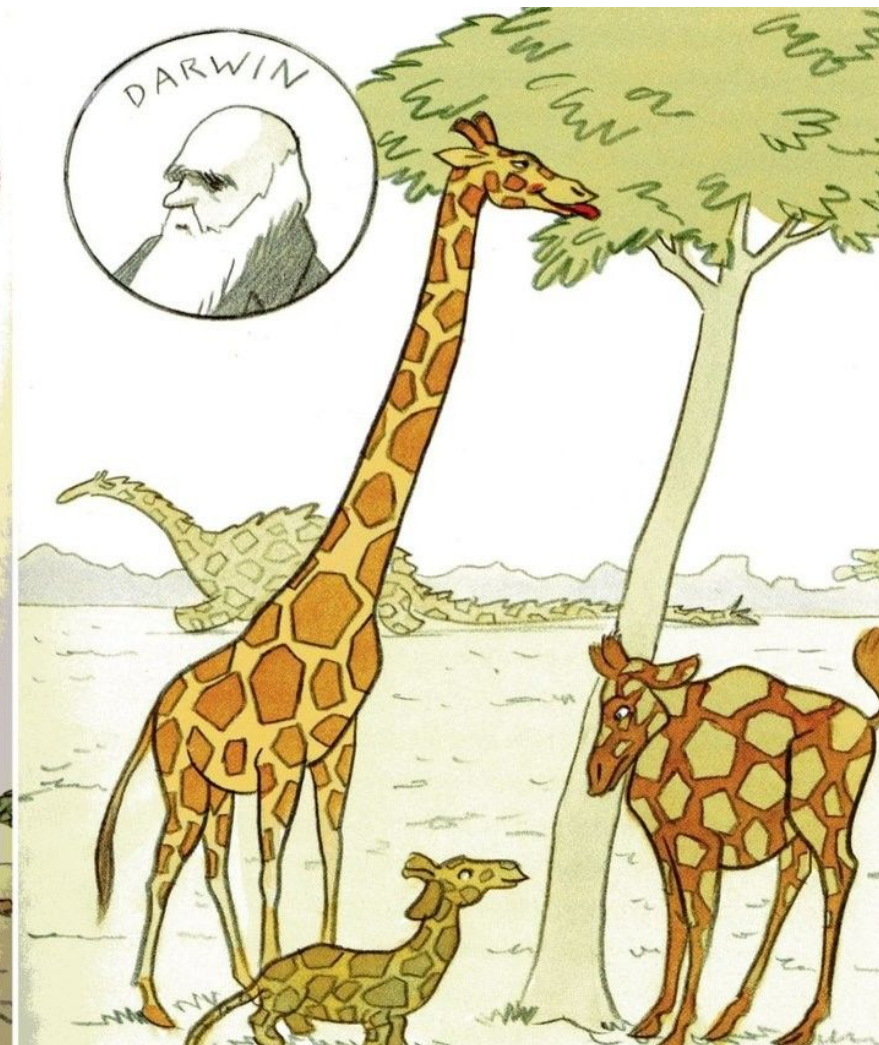
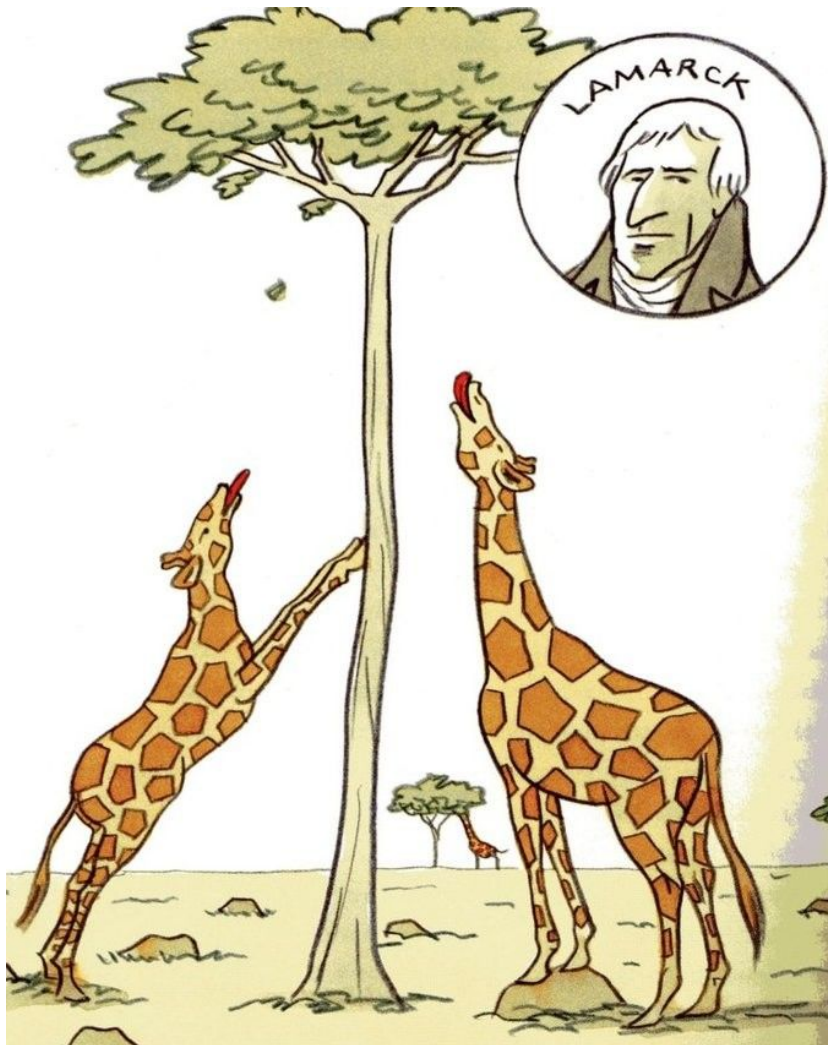


# Algoritme Genetika

- Hill-Climbing Search: berdasarkan nilai objektifnya
- Genetic Algorithm: berdasarkan aturan seleksi alam yang diterapkan pada state collection (sering disebut sebagai populasi)

# A Teori Evolusi

11



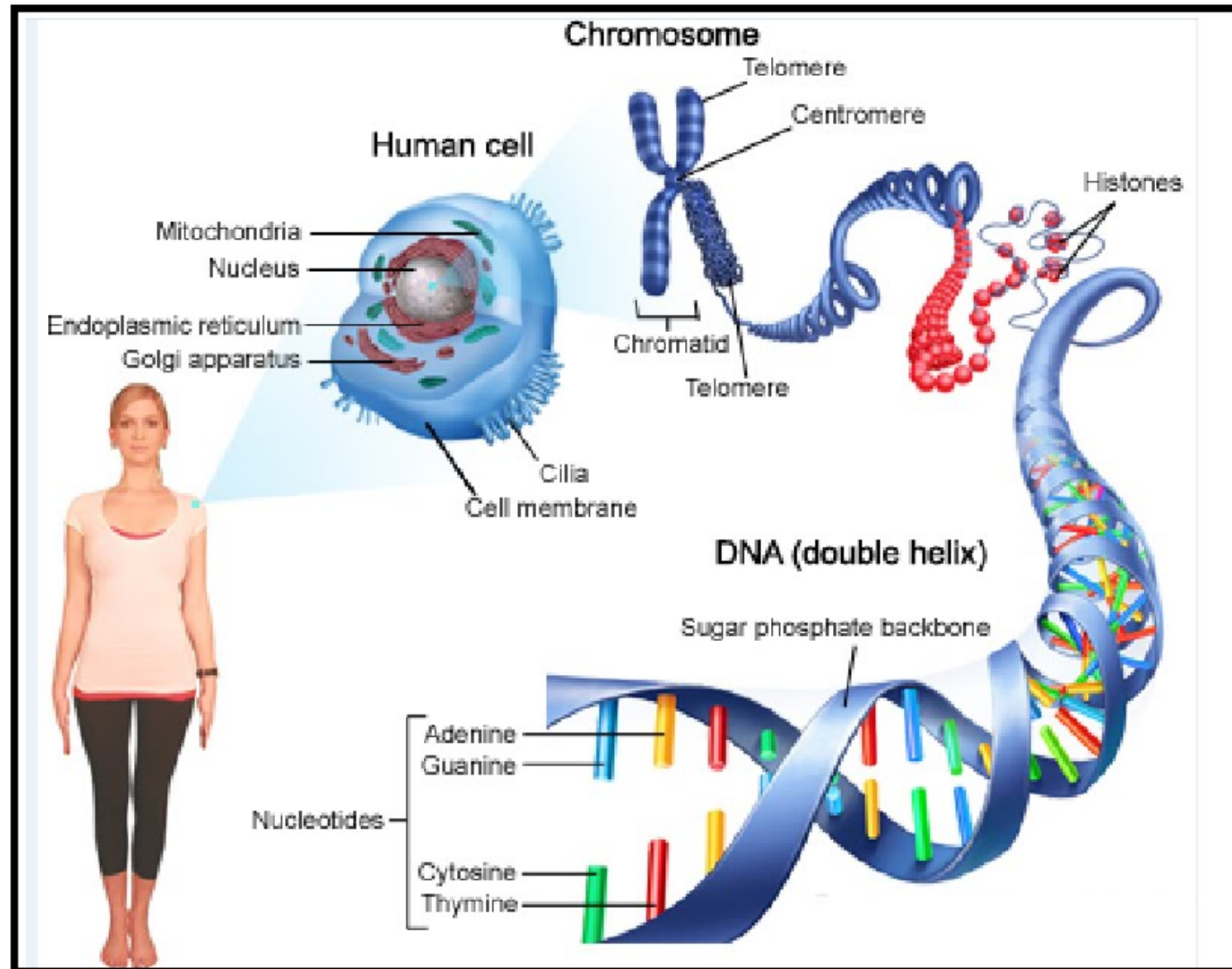


- Ditemukan oleh John Holland dan dikembangkan oleh muridnya David Goldberg
- Algoritme Genetika adalah algoritme yang memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi
- Dalam proses evolusi, individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan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

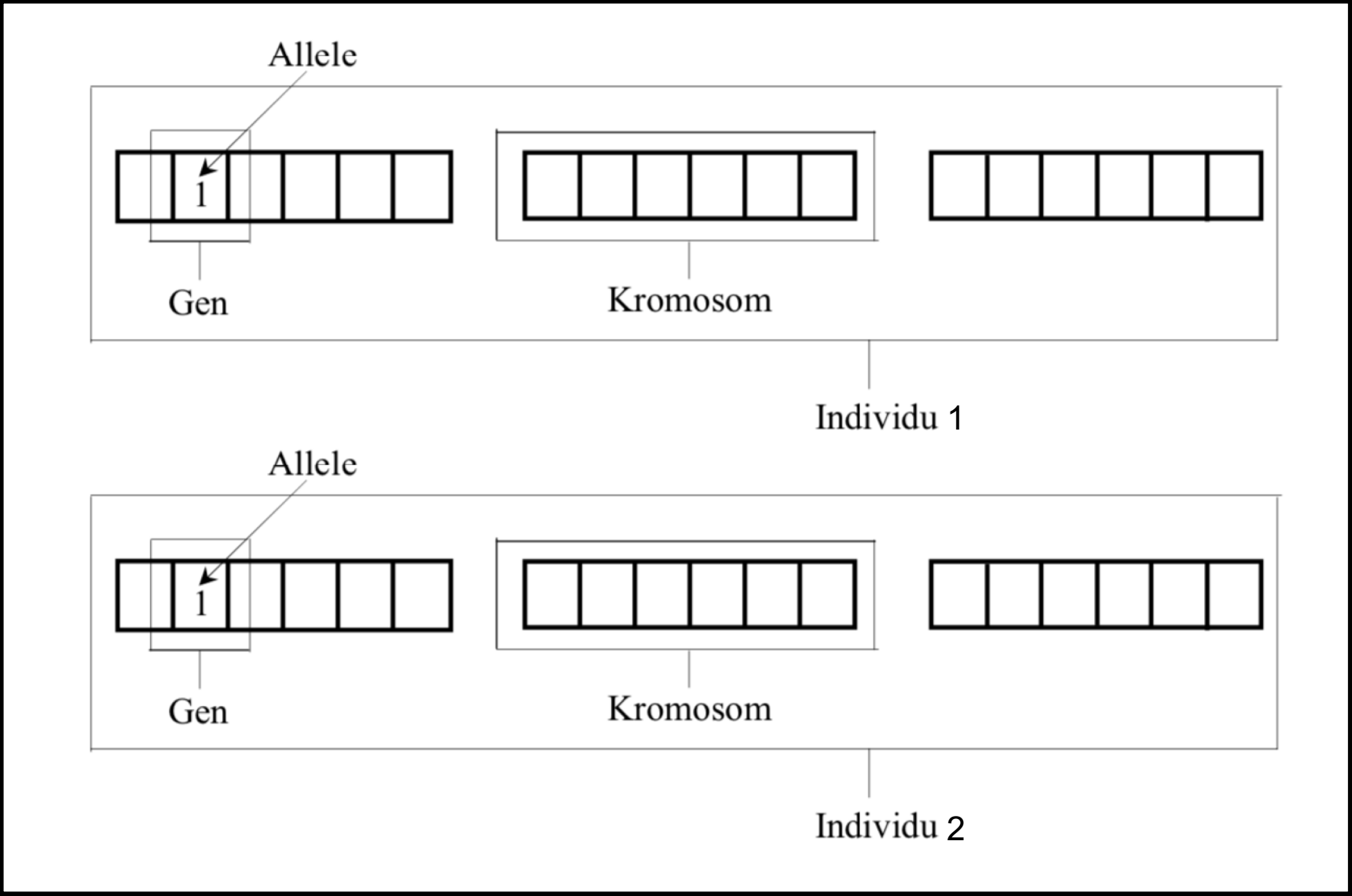


# Genetika Manusia

13



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



**Populasi**

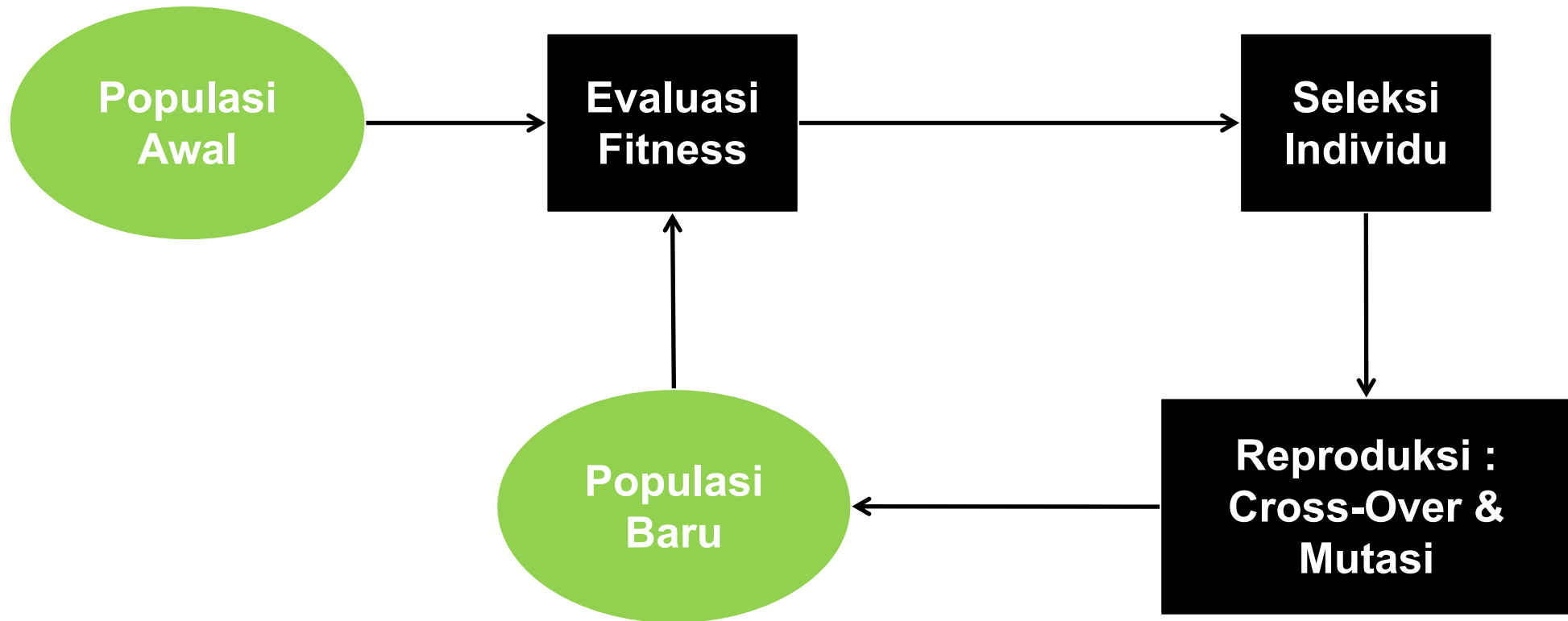
1. Mendefinisikan individu, dimana individu menyatakan salah satu solusi (penyelesaian) yang mungkin dari permasalahan yang diangkat.
2. Mendefinisikan nilai fitness untuk individu (fitness function), yang merupakan ukuran baik-tidaknya sebuah individu atau baik-tidaknya solusi yang didapatkan.
3. Menentukan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





# Siklus Algoritme Genetika

17





# Contoh 1: Optimasi Fungsi

18

- Temukan solusi paling optimum untuk:

$$(X^2 + 1), \text{ dimana nilai } x = \{0, 1, \dots, 31\}$$

- Representasi: kode biner. Contoh: 01101 adalah 13
- Panjang Kromosom: 5 (11111 = 31)
- Populasi Size: 4
- Initial population:

|          |      |
|----------|------|
| 1. 01101 | = 13 |
| 2. 11000 | = 24 |
| 3. 01000 | = 8  |
| 4. 10010 | = 18 |

| No      | Initial Population | x Value | Fitness<br>$f(x) = (x^2 + 1)$ |
|---------|--------------------|---------|-------------------------------|
| 1       | 01101              | 13      | 170                           |
| 2       | 11000              | 24      | 577                           |
| 3       | 01000              | 8       | 65                            |
| 4       | 10010              | 18      | 325                           |
| SUM     |                    |         | 1137                          |
| Average |                    |         | 284.25                        |
| Maximum |                    |         | 577                           |

- Urutan individu berdasarkan fitness: 2 – 4 – 1 – 3
- Misalkan Jumlah seleksi individu adalah 2, maka individu yang akan dipilih adalah individu 2 dan 4



## Cross-Over

| No | Initial Population | Cross-over Point | Offspring (Individu baru) setelah cross-over | x Value | Fitness $f(x) = (x^2 + 1)$ |
|----|--------------------|------------------|--|---------|----------------------------|
| 2  | 11 000             | 2                | 11010 (individu baru nomor 5)                | 26      | 677                        |
| 4  | 10 010             | 2                | 10000 (individu baru nomor 6)                | 16      | 257                        |

## Mutasi (Probabilitas mutasi: 0.2)

| No | Initial Population | Mutation Point | Offspring (Individu baru) setelah cross-over | x Value | Fitness $f(x) = (x^2 + 1)$ |
|----|--------------------|----------------|--|---------|----------------------------|
| 5  | 11010              | 3              | 11110  | 30      | 901                        |
| 6  | 10000              | 3              | 10100  | 20      | 401                        |

Jika angka  $\text{random}(0,1)$  untuk masing-masing gen, dibawah probabilitas mutasi, lakukan mutasi pada gen tersebut

| No           | Populasi Baru    | x Value       | Fitness<br>$f(x) = (x^2 + 1)$ |
|--------------|------------------|---------------|-------------------------------|
| 1            | <del>01101</del> | <del>13</del> | <del>170</del>                |
| 2            | 11000            | 24            | 577                           |
| <del>3</del> | <del>01000</del> | <del>8</del>  | <del>65</del>                 |
| 4            | 10010            | 18            | 325                           |
| 5            | 11010            | 30            | 901                           |
| 6            | 10000            | 20            | 401                           |
| SUM          |                  |               | 2204                          |
| Average      |                  |               | 551                           |
| Maximum      |                  |               | 901                           |

- Jumlah populasi baru harus sama dengan jumlah populasi awal
- Maka, karena ada tambahan 2 individu baru, perlu untuk dilakukan seleksi lagi dan 4 individu dengan nilai fitness tertinggi akan menjadi populasi yang baru.



- Rank-based Fitness Selection
- Boltzmann Selection Elitism
- Steady-State Selection
- Local selection
- Roulette-wheel Selection
- Truncation selection
- Tournament Selection
- Group Selection
- Stochastic universal sampling

- **Crossover**
  - Mengkombinasikan sebagian kromosom parent 1 dengan parent 2

|            |                 |              |
|------------|-----------------|--------------|
| Parent 1   | 1 0 0   0 1 1 1 | (a b)        |
| Parent 2   | 1 1 1   1 0 0 0 | (c d)        |
| Offspring1 | 1001000         | (hasil = ad) |
| Offspring2 | 1110111         | (hasil = cb) |
- **Mutasi**
  - Kromosom hasil perubahan 1 gen / karakter
    - 1 0 0 0 1 1 1 menjadi 1 1 0 0 1 1 1 (perubahan di bit 2)

## **CROSSOVER**

- One-point
- Two-point
- Uniform
- Arithmetic
- Heuristic

## **MUTASI**

- Flip Bit
- Boundary
- Non-Uniform
- Uniform
- Gaussian



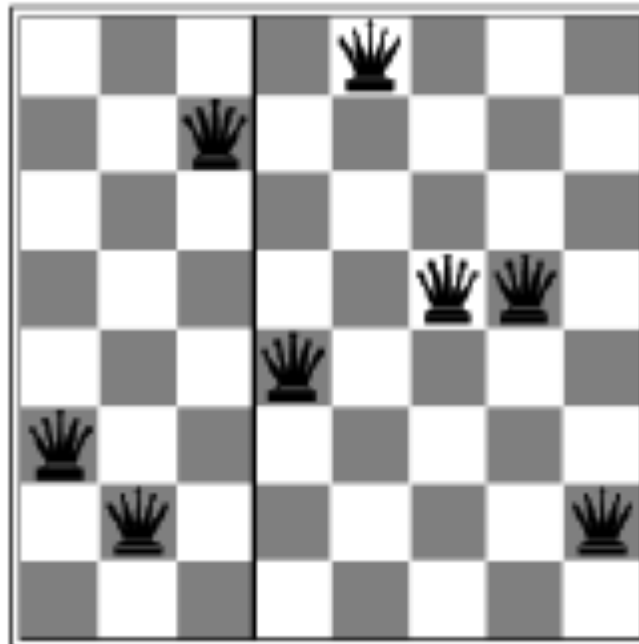




# Persoalan 8-Queen Puzzle

26

- Rancanglah posisi 8 buah ratu pada papan catur sedemikian sehingga tidak terdapat 1 pun ratu yang saling menyerang satu sama lain.



- Nilai Heuristik : Banyak ratu yang saling menyerang.  
Semakin kecil nilai heuristik maka semakin baik.

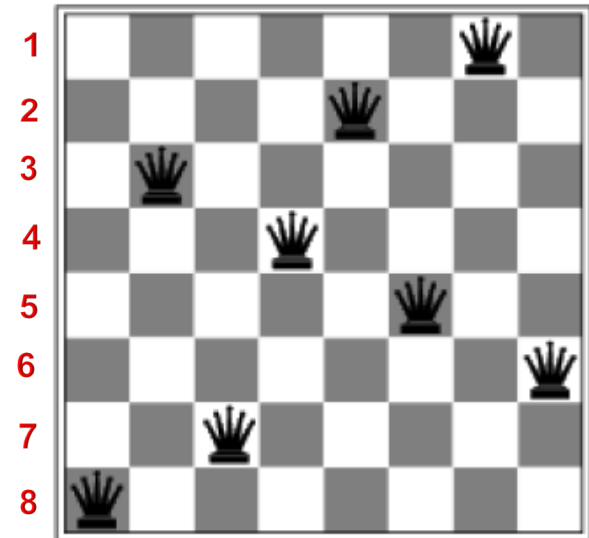
|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| 18 | 12 | 14 | 13 | 13 | 12 | 14 | 14 |
| 14 | 16 | 13 | 15 | 12 | 14 | 12 | 16 |
| 14 | 12 | 18 | 13 | 15 | 12 | 14 | 14 |
| 15 | 14 | 14 | ♚  | 13 | 16 | 13 | 16 |
| ♚  | 14 | 17 | 15 | ♚  | 14 | 16 | 16 |
| 17 | ♚  | 16 | 18 | 15 | ♚  | 15 | ♚  |
| 18 | 14 | ♚  | 15 | 15 | 14 | ♚  | 16 |
| 14 | 14 | 13 | 17 | 12 | 14 | 12 | 18 |

$$h = 17$$

Perhatikan bahwa  
 $h(\text{max}) \rightarrow 8 * 7 / 2 = 28$   
dan  $h(\text{min}) = 0$

- Misalkan nilai fitness function dihitung berdasarkan banyaknya ratu yang **tidak** saling menyerang. Artinya untuk kasus ini, semakin tinggi fitness function maka semakin baik.
- Fitness function  $\rightarrow 28$  – banyak ratu saling menyerang.
- Misalkan suatu individu dinotasikan sebagai deretan 8 angka yang menunjukkan lokasi nomor baris ratu di setiap kolom.
- Contoh : 83742516  $\rightarrow$

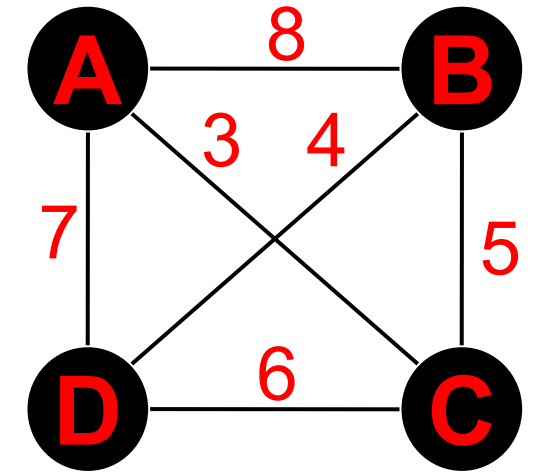
**Fitness Function : 27**





- Persentase fitness function dihitung dari nilai fitness function suatu individu dibagi dengan total nilai fitness function.
- Contoh : individu 24748522 ➔  $24 / (24 + 23 + 20 + 11) = 31 \%$ .
- Ingat, pada kasus ini semakin tinggi fitness function maka semakin baik.
- Pada AG, proses seleksi, cross-over, ataupun mutasi dapat dilakukan berdasarkan pertimbangan apapun yang **dinilai** baik.

- Diketahui persoalan TSP sebagai berikut.
- Gunakan Algoritma Genetika untuk mencari lintasan terpendek persoalan TSP di samping dengan kriteria :
  - 1)FF (Fitness Function) dinyatakan sebagai total cost perjalanan. Artinya semakin rendah semakin baik.
  - 2)AG berhenti jika 3 generasi berturut-turut nilai FF terendah yang diperoleh tidak berubah.
  - 3)Di setiap generasi, hanya dipilih 2 orang tua terbaik dengan proses persilangan menghasilkan 2 anak.
  - 4)Populasi awal adalah [ABCD], [BCDA], [CDAB], dan [DABC].
  - 5)Proses cross-over dilakukan dengan mengambil 1 rute pertama dari Bapak dan 3 rute terakhir dari Ibu.
  - 6)Proses mutasi dilakukan terhadap kota yang muncul 2 kali dalam sebuah individu.



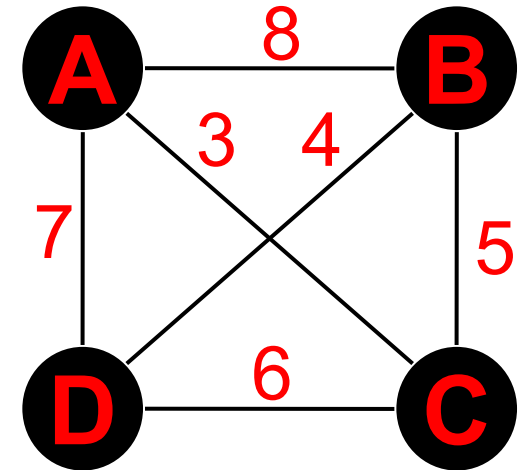
- Populasi Awal :

- Kromosom[1] = [A B C D]
- Kromosom[2] = [B C D A]
- Kromosom[3] = [C D A B]
- Kromosom[4] = [D A B C]

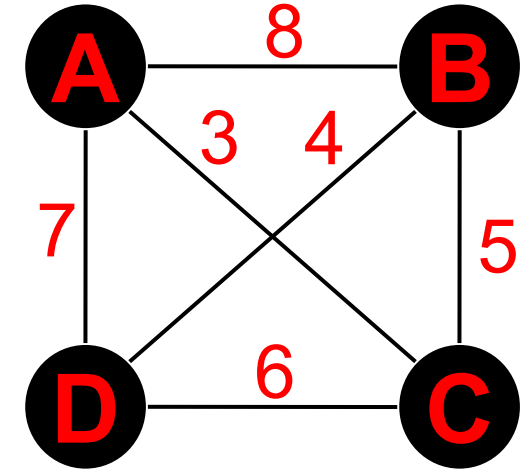
- Evaluasi Fitness:

- $\text{Fitness}[1] = 8 + 5 + 6 = 19$
- $\text{Fitness}[2] = 5 + 6 + 7 = 18$
- $\text{Fitness}[3] = 6 + 7 + 8 = 21$
- $\text{Fitness}[4] = 7 + 8 + 5 = 20$

} Individu Terbaik

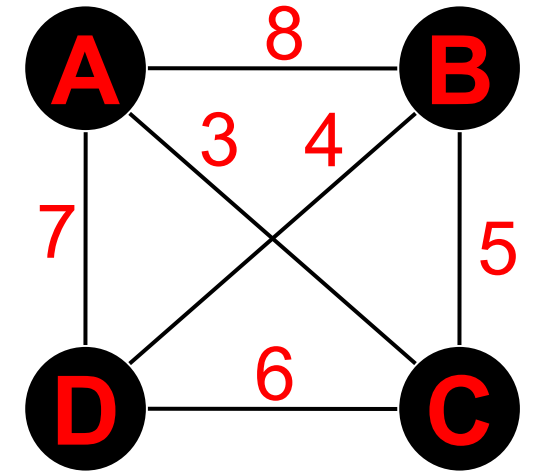


- Seleksi :
  - Kromosom[1] = [A B C D]
  - Kromosom[2] = [B C D A]
- Cross-Over (1 rute awal Bapak + 3 rute akhir Ibu) :
  - Cross-Over[1] = [A C D A]
  - Cross-Over[2] = [B B C D]
- Mutasi
  - Mutasi[1] = [B C D A] atau [A C D B] → Pilih salah satu
  - Mutasi[2] = [A B C D] atau [B A C D] → Pilih salah satu
  - ❑ Karena [B C D A] dan [A B C D] merupakan individu yang sudah pernah muncul sebagai orang tua, maka sebaiknya dipilih Mutasi [1] = [A C D B] dan Mutasi[2] = [B A C D]





- Tahap persilangan selanjutnya :
  - Ulangi lagi dari langkah pemantauan populasi
- Populasi Saat Ini :
  - Kromosom[1] = [A C D B]
  - Kromosom[2] = [B A C D]



Lanjutkan ke tahap selanjutnya, hingga 3 generasi berturut-turut nilai fitness function terendah tidak berubah.