

Praktikum Kecerdasaan Buatan

Algoritma Genetika

Renovita Edelani

Entin Martiana Kusumaningtyas

Yuliana Setiowati

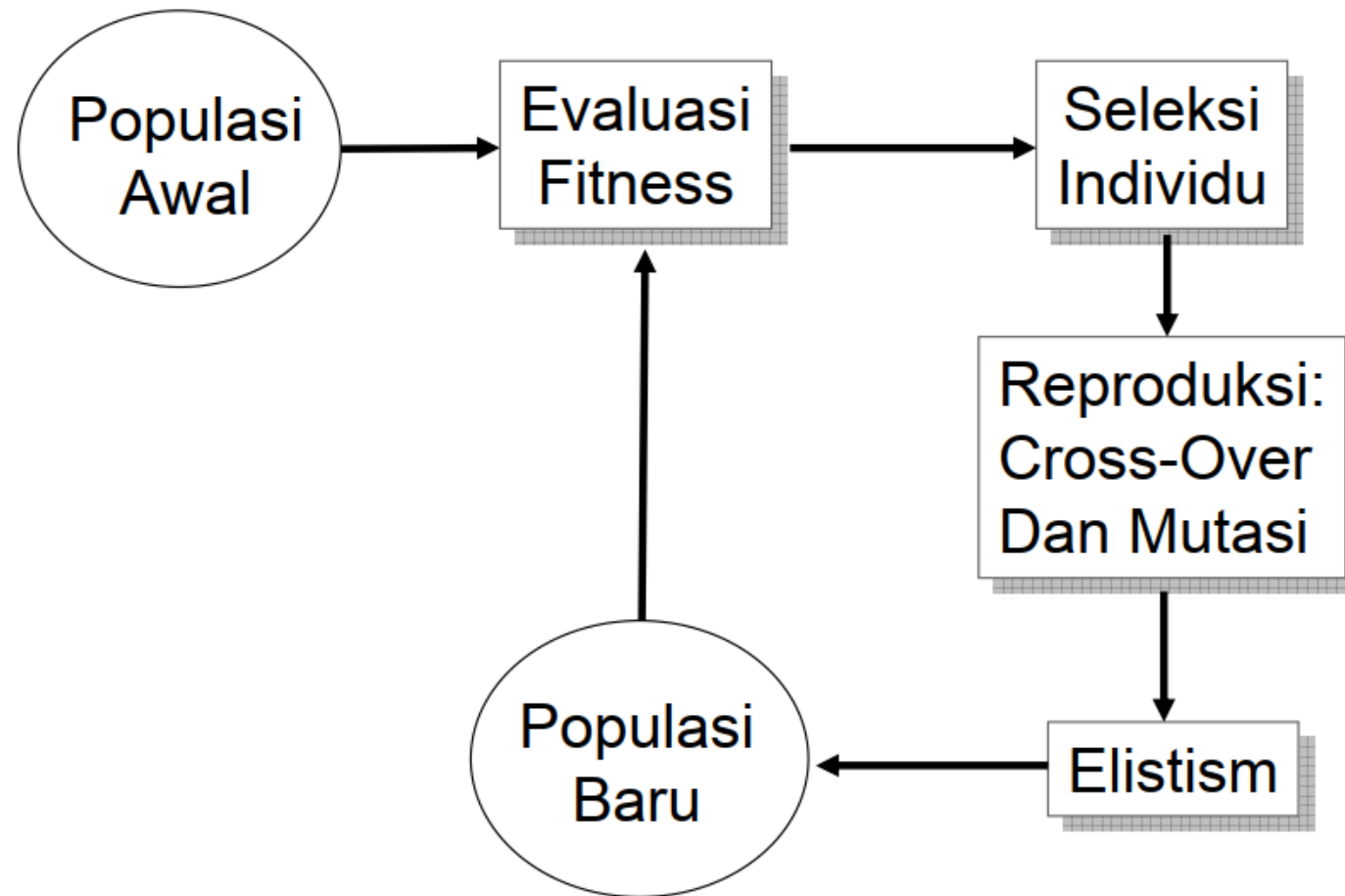
Ali Ridho Barakbah

2023



Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

Siklus Genetic Algorithm



Studi Kasus Genetic Algorithm : Word Matching

Sebuah kata/frasa/kalimat ditentukan sebagai target, misalnya: "GENETIKA"

```
# Target string to be generated  
TARGET = "GENETIKA"
```

Maka Gen yang bisa membentuk kata tersebut adalah:

```
# Valid genes  
GENES = '''abcdefghijklmnopqrs  
tuvwxyzABCDEFGHIJKLMN  
OPQRSTUVWXYZ 1234567890, .-;:_!"#%&/()=?@${[]}'
```

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



1. Pembangkitan Populasi Awal

Kita akan membangkitkan populasi tiap generasi sebanyak 1000 individu

```
# Number of individuals in each generation
POPULATION_SIZE = 1000
```

Dibangkitkan gen-gen dari 1000 individu dengan fungsi create_gnome

```
#create initial population (populasi awal)
for _ in range(POPULATION_SIZE):
    gnome = Individual.create_gnome()
    population.append(Individual(gnome))
```

Fungsi ini membangkitkan gen dengan cara memilih random dari valid genes yang sudah kita tentukan

```
def create_gnome(self):
    ...
    create chromosome or string of genes
    ...
    global TARGET
    gnome_len = len(TARGET)
    return [self.mutated_genes() for _ in range(gnome_len)]
```

2. Evaluasi Fitness

Hitung nilai fitness dari setiap individu dalam populasi, dan urutkan nilai tersebut dari yang paling kecil hingga terbesar

```
#sort the population in increasing order of fitness score
#(Evaluasi Nilai Fitness)
population = sorted(population, key = lambda x:x.fitness)
```

Fungsi fitness yang digunakan pada kasus ini adalah:
Jika gen dari individu \neq gen target maka fitness+=1

```
def cal_fitness(self):
    ...
    Calculate fitness score, it is the number of
    characters in string which differ from target
    string
    ...
    global TARGET
    fitness = 0
    for gs, gt in zip(self.chromosome, TARGET):
        if gs != gt: fitness +=1
    return fitness
```

Contoh:

Generation: 1 String: z lzVIF/ Fitness: 7



3. Seleksi Individu

Proses seleksi individu pada program ini dimana:

- 10% individu dengan nilai fitness paling terbaik (minimum) dari generasi sebelumnya diambil sebagai generasi baru (elitism)
- 90% lainnya diambil dari generasi lama dengan ketentuan sebagai berikut:
 - 50% random sebagai parent 1
 - 50% random sebagai parent 2
 - Setelah itu dilakukan proses Reproduksi → Cross Over dan Mutasi sehingga menghasilkan anak

```
#perform elitism, that mean 10% of fittest population
#goes to the next generation
s=int((10*POPULATION_SIZE)/100)
new_generation.extend(population[:s])

#from 50% of fittest population (Seleksi), Individuals
#will mate (Reproduksi) to produce offspring
s = int((90*POPULATION_SIZE)/100)
for _ in range(s):
    parent1 = random.choice(population[:50])
    parent2 = random.choice(population[:50])
    child = parent1.mate(parent2)
    new_generation.append(child)

population = new_generation
```

4. Reproduksi

- Reproduksi pada program ini tidak terlalu mirip dengan proses Reproduksi pada Teori.
- Pada praktikum, proses reproduksi antara Cross Over dan Mutasi digabung menjadi satu buah fungsi yang sama

prob = random

Jika $\text{prob} < 0.45$ → gen **parent 1** dijadikan gen **child**

Jika $0.45 < \text{prob} < 0.90$ → gen **parent 2** dijadikan gen **child**

Jika $\text{prob} \geq 0.90$ → gen **child** merupakan hasil mutasi

```
#if prob is less than 0.45, insert gene
#from parent 1
if prob < 0.45:
    child_chromosome.append(gp1)

#if prob is between 0.45 and 0.90, insert
#gene from parent 2
elif (prob < 0.90):
    child_chromosome.append(gp2)

#otherwise insert random gene(mutate)
#for maintaining diversity
else:
    child_chromosome.append(self.mutated_genes())
```

5. Elitism

- Elitism yang dipakai pada program ini terjadi pada saat Seleksi Individu, dimana:
- 10% individu dengan nilai fitness paling terbaik (minimum) dari generasi sebelumnya diambil sebagai generasi baru (elitism)

```
#perform elitism, that mean 10% of fittest population  
#goes to the next generation  
s=int((10*POPULATION_SIZE)/100)  
new_generation.extend(population[:s])
```


Tugas

1. Buatlah program python algoritma genetika untuk word matching yang ada pada link ini:
https://colab.research.google.com/drive/1k4r9NY55iUqsm219ScNvxR_G07a2gKVe?usp=sharing
2. Ubahlah Target menjadi nama masing-masing!
3. Jalankan Program! Butuh berapa generasi agar menemukan nama kalian?
4. Ubah banyak Populasi menjadi:
 - jauh lebih kecil seperti 10 dan 100.
 - Lebih besar seperti 10000 dan 100000,Jelaskan yang terjadi?



Tugas

5. Pada proses Reproduksi (mate) ubahlah nilai probabilitas dari proses cross over dan mutasi, dimana fungsi awalnya:

prob = random

Jika prob < 0.45 → gen **parent 1** dijadikan gen **child**

Jika $0.45 < \text{prob} < 0.90$ → gen **parent 2** dijadikan gen **child**

Jika prob ≥ 0.90 → gen **child** merupakan hasil mutasi

Jelaskan hasil yang terjadi?

6. Sama dengan nomor 5 ubahlah nilai probabilitas untuk mutasi sehingga kemungkinan bermutasi jauh lebih besar dari sebelumnya.

Contoh: prob ≥ 0.90 ini berarti kemungkinan mutasi → 0.1

```
#if prob is less than 0.45, insert gene
#from parent 1
if prob < 0.45:
    child_chromosome.append(gp1)

#if prob is between 0.45 and 0.90, insert
#gene from parent 2
elif (prob < 0.90):
    child_chromosome.append(gp2)

#otherwise insert random gene(mutate)
#for maintaining diversity
else:
    child_chromosome.append(self.mutated_genes())
```



Tugas

7. Analisa lah semua percobaan yang dilakukan dan jelaskan apa itu:
 - Generasi, Populasi, Individu, Kromosom dan Gen yang ada pada program pencarian nama masing-masing!
 - Apa hubungan nilai probabilitas cross over dan probabilitas mutasi terhadap output yang didapatkan?
 - Apa itu elitism?



Prosedur Pengumpulan Tugas

- Buatlah laporan dalam slide ppt. laporan terdiri dari screenshot coding, hasil running, Analisa dan jawaban dari pertanyaan.
- Upload file tersebut ke ETHOL
- Simpan laporan dalam file pdf dengan format penamaan:
 - **Untuk D4 IT A:**
AID4ITA_M7_NRP_namadepan.pdf
Deadline upload: Minggu, 16 April Pukul 23.59 WIB
 - **Untuk D3 IT A:**
AID3ITA_M7_NRP_namadepan.pdf
Deadline upload: Selasa, 18 April Pukul 23.59 WIB



Referensi

- Modul Ajar Kecerdasan Buatan, Entin Martiana, Ali Ridho Barakbah, Yuliana Setiowati, Politeknik Elektronika Negeri Surabaya, 2014.
- Artificial Intelligence (Teori dan Aplikasinya), Sri Kusumadewi, cetakan pertama, Penerbit Graha Ilmu, 2003.
- <https://www.geeksforgeeks.org/genetic-algorithms/>