



ALGORITMA GENETIKA

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?



```
# Python3 program to create target string, starting from
# random string using Genetic Algorithm
# https://www.geeksforgeeks.org/genetic-algorithms/
import random

# Number of individuals in each generation
POPULATION_SIZE = 1000

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

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

Generation: 1	String: wT(I	Fitness: 3
Generation: 2	String: kAT(I	Fitness: 2
Generation: 3	String: RATfI	Fitness: 1
Generation: 4	String: RATRI	Fitness: 0

Pada population size yang berjumlah 1000, nama saya ditemukan pada generasi ke-4

```
▶ # Python3 program to create target string, starting from
# random string using Genetic Algorithm
# https://www.geeksforgeeks.org/genetic-algorithms/
import random

# Number of individuals in each generation
POPULATION_SIZE = 10

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

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

Generation: 1252	String: LATRI	Fitness: 1
Generation: 1253	String: LATRI	Fitness: 1
Generation: 1254	String: LATRI	Fitness: 1
Generation: 1255	String: LATRI	Fitness: 1
Generation: 1256	String: LATRI	Fitness: 1
Generation: 1257	String: LATRI	Fitness: 1
Generation: 1258	String: LATRI	Fitness: 1
Generation: 1259	String: LATRI	Fitness: 1
Generation: 1260	String: RATRI	Fitness: 0

Pada population size yang berjumlah 10, nama saya ditemukan pada generasi ke-1260

```

▶ # Python3 program to create target string, starting from
# random string using Genetic Algorithm
# https://www.geeksforgeeks.org/genetic-algorithms/
import random

# Number of individuals in each generation
POPULATION_SIZE = 100

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

# Target string to be generated
TARGET = "RATRI"

```

Pada population size yang berjumlah 100,
nama saya ditemukan pada generasi ke-24

```

➡ Generation: 1   String: &A r_   Fitness: 4
   Generation: 2   String: &A r_   Fitness: 4
   Generation: 3   String: RAA6_   Fitness: 3
   Generation: 4   String: RAAyI   Fitness: 2
   Generation: 5   String: RAAyI   Fitness: 2
   Generation: 6   String: RAAyI   Fitness: 2
   Generation: 7   String: RA RI   Fitness: 1
   Generation: 8   String: RA RI   Fitness: 1
   Generation: 9   String: RA RI   Fitness: 1
   Generation: 10  String: RA RI   Fitness: 1
   Generation: 11  String: RA RI   Fitness: 1
   Generation: 12  String: RA RI   Fitness: 1
   Generation: 13  String: RA RI   Fitness: 1
   Generation: 14  String: RA RI   Fitness: 1
   Generation: 15  String: RA RI   Fitness: 1
   Generation: 16  String: RA RI   Fitness: 1
   Generation: 17  String: RA RI   Fitness: 1
   Generation: 18  String: RA RI   Fitness: 1
   Generation: 19  String: RA RI   Fitness: 1
   Generation: 20  String: RA RI   Fitness: 1
   Generation: 21  String: RA RI   Fitness: 1
   Generation: 22  String: RA RI   Fitness: 1
   Generation: 23  String: RA RI   Fitness: 1
   Generation: 24  String: RATRI   Fitness: 0

```

```
▶ # Python3 program to create target string, starting from
# random string using Genetic Algorithm
# https://www.geeksforgeeks.org/genetic-algorithms/
import random

# Number of individuals in each generation
POPULATION_SIZE = 10000

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

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

```
Generation: 1   String: RA6#7   Fitness: 2
Generation: 2   String: RAGRI   Fitness: 1
Generation: 3   String: RATRI   Fitness: 0
```

Pada population size yang berjumlah 10000, nama saya ditemukan pada generasi ke-3



```
# Python3 program to create target string, starting from
# random string using Genetic Algorithm
# https://www.geeksforgeeks.org/genetic-algorithms/
import random

# Number of individuals in each generation
POPULATION_SIZE = 100000

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

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

Generation: 1	String: RUnRI	Fitness: 1
Generation: 2	String: RATRI	Fitness: 0


Pada population size yang berjumlah 100000, nama saya ditemukan pada generasi ke-2



Analisa

Algoritma genetika dapat beroperasi lebih efisien dan menghasilkan solusi yang baik dengan cepat jika populasi yang dijadikan sampel lebih kecil, seperti 10 atau 100. Akan tetapi, pendekatan ini memiliki risiko bahwa variasi genetik dalam populasi akan berkurang dan individu dengan kualitas yang rendah dapat mendominasi, sehingga solusi yang dihasilkan kurang berkualitas.

Di sisi lain, populasi yang lebih besar, seperti 10.000 atau 100.000, dapat memberikan variasi genetik yang lebih besar dan mencegah populasi terjebak pada solusi lokal yang buruk. Namun, pendekatan ini juga dapat memperlambat waktu komputasi algoritma genetika.



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 < prob < 0.90 → gen **parent 2** dijadikan gen **child**

Jika prob >= 0.90 → gen **child** merupakan hasil mutasi

Jelaskan hasil yang terjadi?

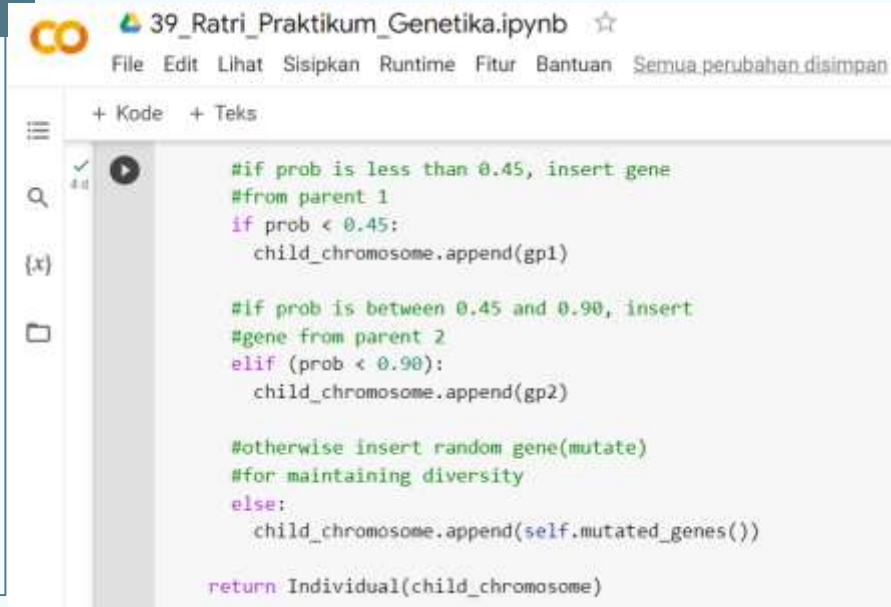
6. Sama dengan nomor 5 ubahlah nilai probabilitas untuk mutasi sehingga kemungkinan bermutasi jauh lebih besar dari sebelumnya.

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())
```



```
39_Ratri_Praktikum_Genetika.ipynb ☆
File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan disimpan

+ Kode + Teks

44 #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())

return Individual(child_chromosome)
```

Jika nilai probabilitas untuk proses mutasi meningkat, maka kemungkinan besar akan terjadi lebih banyak mutasi pada generasi berikutnya. Hal ini dapat mengakibatkan penurunan performa individu terbaik karena banyak gen yang dimutasi secara acak dan dapat menghilangkan gen yang sebelumnya berhasil mencapai fitness score yang baik. Namun, jika nilai probabilitas untuk proses mutasi berkurang, maka kemungkinan besar individu baru akan memiliki banyak gen yang sama dengan gen induk, sehingga dapat menurunkan diversitas populasi dan kecepatan konvergensi.

6. Jika nilai probabilitas untuk proses mutasi meningkat drastis, misalnya menjadi $\text{prob} \geq 0.90$, maka kemungkinan besar akan terjadi lebih banyak mutasi pada setiap generasi. Hal ini dapat meningkatkan diversitas populasi dan kecepatan konvergensi karena akan lebih banyak variasi dalam genotipe populasi. Namun, jika nilai probabilitas untuk proses mutasi terlalu tinggi, misalnya probabilitas mutasi mencapai 1.0, maka kemungkinan besar semua gen dalam populasi akan bermutasi, sehingga populasi akan kehilangan informasi genetik yang sebelumnya berhasil mencapai fitness score yang baik

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?

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

Hubungan probabilitas cross over dan probabilitas mutasi: Probabilitas cross over dan probabilitas mutasi mempengaruhi variasi genetik dalam populasi. Probabilitas cross over menentukan seberapa sering kromosom dalam populasi dicampur dengan cara yang menghasilkan keturunan yang baru, sedangkan probabilitas mutasi menentukan seberapa sering gen dalam kromosom diubah secara acak. Semakin tinggi probabilitas cross over, semakin besar kemungkinan variasi genetik dalam populasi. Sedangkan semakin tinggi probabilitas mutasi, semakin besar kemungkinan variasi genetik dalam individu.

Elitism adalah strategi yang digunakan dalam algoritma genetika untuk mempertahankan individu terbaik dari generasi sebelumnya dan memindahkannya langsung ke generasi berikutnya tanpa modifikasi. Ini dilakukan untuk menghindari kehilangan individu yang memiliki nilai solusi yang tinggi dan untuk mempercepat konvergensi algoritma