

Muhammad Hafidh Raditya

1301184079

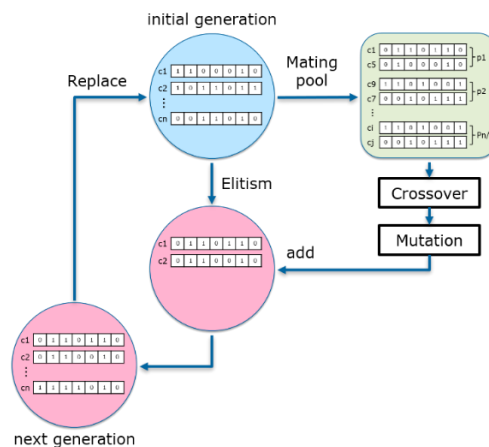
IF-42-03

LAPORAN TUGAS PEMROGRAMAN ALGORITMA GENETIKA

MATA KULIAH PENGANTAR KECERDASAN BUATAN

A. Strategi

Teknik yang saya pakai pada tugas kali ini adalah pembangkitan individu menggunakan integer. Nantinya individu yang sudah dibuat akan digabungkan ke dalam populasi dengan jumlah tertentu tergantung percobaan. Untuk menentukan nilai fitnessnya, saya menggunakan rumus $f=b-h$ dengan $b>1$ dan h adalah fungsi tujuan. Pada tugas ini saya assign $b=10$. Untuk menentukan fitness yang terbaik, populasi saya urutkan terlebih dahulu sesuai nilai fitnessnya. Hal ini saya lakukan agar mudah memilih mana individu terbaik dalam suatu populasi. Individu yang bukan terbaik nantinya akan dilakukan crossover dan mutasi sesuai flowchart dibawah ini:



Untuk kriteria pemberhentian looping generasi baru, saya menambahkan satu variabel yaitu *cekStagnan* yang berguna untuk melihat sudah berapa kali nilai fitness terbaik didapatkan dalam suatu populasi pada setiap kali looping.

Untuk parameter GA yang paling berpengaruh terhadap ditemukannya generasi terbaik, menurut saya adalah parameter untuk menentukan banyaknya jumlah individu di dalam suatu populasi. Hipotesis saya adalah, semakin banyak individu di dalam suatu populasi, maka peluang ditemukannya generasi terbaik akan semakin cepat. Oleh karena itu hipotesis tersebut akan saya buktikan pada analisis berikut ini:

B. Analisis

1. Percobaan pertama

Pada percobaan pertama, saya akan membuat 3 individu yang dibangkitkan secara random di dalam satu populasi. Saya akan menjalankan program sebanyak sepuluh kali run dan menganalisa hasilnya.

PS D:\Belajar Python\coba tugas GA> & GA/main_clone.py" Banyaknya generasi = 73 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0 PS D:\Belajar Python\coba tugas GA> & GA/main_clone.py"	PS D:\Belajar Python\coba tugas GA> & GA/main_clone.py" Banyaknya generasi = 54 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0 PS D:\Belajar Python\coba tugas GA> & GA/main_clone.py"
Banyaknya generasi = 65 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0 PS D:\Belajar Python\coba tugas GA> & GA/main_clone.py"	Banyaknya generasi = 56 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0 PS D:\Belajar Python\coba tugas GA> & GA/main_clone.py"
Banyaknya generasi = 75 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0 PS D:\Belajar Python\coba tugas GA> & GA/main_clone.py"	Banyaknya generasi = 59 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0 PS D:\Belajar Python\coba tugas GA> & GA/main_clone.py"
Banyaknya generasi = 64 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0 PS D:\Belajar Python\coba tugas GA> & GA/main_clone.py"	Banyaknya generasi = 64 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0 PS D:\Belajar Python\coba tugas GA> & GA/main_clone.py"
Banyaknya generasi = 82 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0 PS D:\Belajar Python\coba tugas GA> █	Banyaknya generasi = 73 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0 PS D:\Belajar Python\coba tugas GA> █

Dari screenshot hasil run program sebanyak sepuluh kali, dapat dilihat worst casenya adalah 82 generasi dan best casenya adalah 54 generasi. Namun karena saya menggunakan acuan stagnan sebanyak 50 generasi, artinya worst case yang sebenarnya adalah $82 - 50 = 32$ generasi dan best case yang sebenarnya adalah $54 - 50 = 4$ generasi.

2. Percobaan kedua

Pada percobaan kedua, saya akan membuat 12 individu yang dibangkitkan secara random di dalam satu populasi. Saya akan menjalankan program sebanyak sepuluh kali dan menganalisis hasilnya.

```
PS D:\Belajar Python\coba tu PS D:\Belajar Python\coba tuga
GA/main_clone.py"          GA/main_clone.py"
Banyaknya generasi = 57      Banyaknya generasi = 56
fitness terbaik = 12.0      fitness terbaik = 12.0
nilai minimum -2.0         nilai minimum -2.0
x1 = 2 x2 = 0              x1 = 2 x2 = 0
PS D:\Belajar Python\coba tu PS D:\Belajar Python\coba tuga
GA/main_clone.py"          GA/main_clone.py"
Banyaknya generasi = 54      Banyaknya generasi = 60
fitness terbaik = 12.0      fitness terbaik = 12.0
nilai minimum -2.0         nilai minimum -2.0
x1 = 2 x2 = 0              x1 = 2 x2 = 0
PS D:\Belajar Python\coba tu PS D:\Belajar Python\coba tuga
GA/main_clone.py"          GA/main_clone.py"
Banyaknya generasi = 53      Banyaknya generasi = 54
fitness terbaik = 12.0      fitness terbaik = 12.0
nilai minimum -2.0         nilai minimum -2.0
x1 = 2 x2 = 0              x1 = 2 x2 = 0
PS D:\Belajar Python\coba tu PS D:\Belajar Python\coba tuga
GA/main_clone.py"          GA/main_clone.py"
Banyaknya generasi = 55      Banyaknya generasi = 54
fitness terbaik = 12.0      fitness terbaik = 12.0
nilai minimum -2.0         nilai minimum -2.0
x1 = 2 x2 = 0              x1 = 2 x2 = 0
PS D:\Belajar Python\coba tu PS D:\Belajar Python\coba tuga
GA/main_clone.py"          GA/main_clone.py"
Banyaknya generasi = 54      Banyaknya generasi = 53
fitness terbaik = 12.0      fitness terbaik = 12.0
nilai minimum -2.0         nilai minimum -2.0
x1 = 2 x2 = 0              x1 = 2 x2 = 0
PS D:\Belajar Python\coba tu PS D:\Belajar Python\coba tuga
```

Dari screenshot run program sebanyak sepuluh kali, dapat dilihat worst casenya adalah 60 generasi dan best casenya adalah 53 generasi. Namun karena saya menggunakan acuan stagnan sebanyak 50 kali, artinya worst case yang sebenarnya adalah $60 - 50 = 10$ generasi, dan best case yang sebenarnya adalah $53 - 50 = 3$ generasi.

3. Percobaan ketiga

Pada percobaan ketiga, saya akan membuat 50 individu yang dibangkitkan secara random di dalam satu populasi. Saya akan menjalankan program sebanyak sepuluh kali dan menganalisis hasilnya.

PS D:\Belajar Python\coba GA/main_clone.py" Banyaknya generasi = 53 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0	PS D:\Belajar Python\coba tu GA/main_clone.py" Banyaknya generasi = 53 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0
PS D:\Belajar Python\coba GA/main_clone.py" Banyaknya generasi = 53 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0	PS D:\Belajar Python\coba tu GA/main_clone.py" Banyaknya generasi = 53 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0
PS D:\Belajar Python\coba GA/main_clone.py" Banyaknya generasi = 53 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0	PS D:\Belajar Python\coba tu GA/main_clone.py" Banyaknya generasi = 53 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0
PS D:\Belajar Python\coba GA/main_clone.py" Banyaknya generasi = 53 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0	PS D:\Belajar Python\coba tu GA/main_clone.py" Banyaknya generasi = 53 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0
PS D:\Belajar Python\coba GA/main_clone.py" Banyaknya generasi = 53 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0	PS D:\Belajar Python\coba tu GA/main_clone.py" Banyaknya generasi = 53 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0
PS D:\Belajar Python\coba GA/main_clone.py" Banyaknya generasi = 53 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0	PS D:\Belajar Python\coba tu GA/main_clone.py" Banyaknya generasi = 53 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0
PS D:\Belajar Python\coba GA/main_clone.py" Banyaknya generasi = 53 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0	PS D:\Belajar Python\coba tu GA/main_clone.py" Banyaknya generasi = 53 fitness terbaik = 12.0 nilai minimum -2.0 x1 = 2 x2 = 0

Dari screenshot run program sebanyak sepuluh kali, dapat dilihat kalau worst case dan best case memiliki nilai yang sama yaitu 53 generasi. Namun karena saya menggunakan acuan stagnan sebanyak 50 kali, artinya nilai worst case dan best case sebenarnya yaitu sebanyak $53 - 50 = 3$ generasi.

C. Kesimpulan

Pada percobaan menggunakan 3 individu per populasi didapatkan 32 generasi sebagai worst casenya. Pada percobaan menggunakan 12 individu per populasi, didapatkan 10 generasi sebagai worst casenya. Dan pada percobaan menggunakan 50 individu per populasi, didapatkan 3 generasi sebagai worst case sekaligus best casenya. Maka dapat disimpulkan, kalau semakin besar populasi, maka peluang munculnya generasi terbaik akan semakin cepat.