**LAPORAN TUGAS PEMROGRAMAN 1**

**ALGORITMA GENETIKA**

**IF 43 02**

**GROUP I**



**Disusun oleh :**

1. **Ananda Affan Fattahila (1301194175)**
2. **Maulana Nur (1301190402)**
3. **Zendi Bramantia Alfareza (1301194145)**

**TELKOM UNIVERSITY**

**FAKULTAS INFORMATIKA**

**PRODI S1 INFORMATIKA**

**BANDUNG**

**2021**

# **DAFTAR ISI**

[**DAFTAR ISI** 2](#_Toc67585482)

[**BAB 1** 3](#_Toc67585483)

[**PENDAHULUAN** 3](#_Toc67585484)

[1.1 Persoalan 3](#_Toc67585485)

[**BAB 2** 4](#_Toc67585486)

[**PEMBAHASAN** 4](#_Toc67585487)

[2.1 Desain Kromosom dan Metode Pendekodean 4](#_Toc67585488)

[2.2 Ukuran Populasi 4](#_Toc67585489)

[2.3 Pemilihan orangtua 4](#_Toc67585490)

[2.4 Pemilihan dan teknik operasi genetik (crossover dan mutasi) 4](#_Toc67585491)

[**2.4.1 Crossover (rekombinasi satu titik)** 4](#_Toc67585492)

[**2.4.2 Mutasi Acak** 5](#_Toc67585493)

[2.5 Probabilitas operasi genetik (Pc dan Pm) 5](#_Toc67585494)

[2.6 Metode Pergantian Generasi (Seleksi Survivor) 5](#_Toc67585495)

[2.7 Kriteria Penghentian Evolusi 5](#_Toc67585496)

[2.8 Kode Program 5](#_Toc67585497)

[**2.8.1 Library** 6](#_Toc67585498)

[**2.8.2 Perhitungan fitness** 6](#_Toc67585499)

[**2.8.3 Membuat Populasi** 6](#_Toc67585500)

[**2.8.4 Membuat Himpunan Fitness dari Populasi** 7](#_Toc67585501)

[**2.8.5 Decode** 7](#_Toc67585502)

[**2.8.6 Tournament** 8](#_Toc67585503)

[**2.8.7 Pindah Silang** 8](#_Toc67585504)

[**2.8.8 Mutasi** 9](#_Toc67585505)

[**2.8.9 Mencari Index Minimum Fitness** 9](#_Toc67585506)

[**2.8.10 Seleksi Survivor** 10](#_Toc67585507)

[**2.8.11 Mencari Index Maksimum Fitness** 10](#_Toc67585508)

[**2.8.12 Program Utama** 11](#_Toc67585509)

[2.9 Membuat Grafik Perubahan Fitness Terbaik 12](#_Toc67585510)

[2.10 Link Video Youtube dan Source Code 13](#_Toc67585511)

[**BAB 3** 14](#_Toc67585512)

[**PENUTUP** 14](#_Toc67585513)

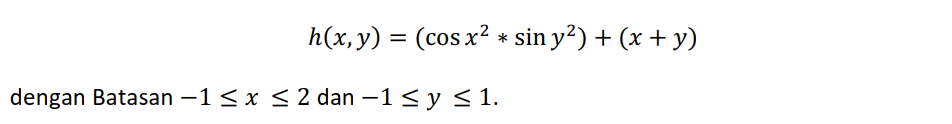
[3.1 Kesimpulan 14](#_Toc67585514)

# **BAB 1**

# **PENDAHULUAN**

## 1.1 Persoalan

Tugas pemrograman 1 merupakan tugas pertama pada mata kuliah pengantar kecerdasan buatan. Pada tugas pemrograman ini kami, ditugaskan untuk melakukan analisis, desain, dan implementasi algoritma *Genetic algorithm* (GA) ke dalam suatu program komputer untuk menemukan **nilai maksimum** dari sebuah fungsi. Fungsi yang dimaksud sebagai berikut :



Terdapat beberapa hal yang harus di observasi dalam pembuatan *Genetic Algorithm* diantaranya adalah Desain Kromosom dan Metode Pengkodean, Ukuran Populasi, Pemilihan orangtua, Pemilihan dan teknik operasi genetik (*crossover* dan mutasi), Probabilitas operasi genetik (Pc dan Pm), Metode Pergantian Generasi (Seleksi *Survivor*) dan Kriteria Penghentian Evolusi

Ada pula beberapa hal yang harus di bangun, diantaranya adalah Dekode kromosom, Perhitungan fitness, Pemilihan orangtua, Crossover (pindah silang), Mutasi dan Pergantian Generasi. Namun kami juga menambahkan sebuah prosedur untuk menampilkan perubahan grafik dimana dia menunjukan angka perubahan nilai fitness terbaik dari tiap generasi.

# **BAB 2**

# **PEMBAHASAN**

## 2.1 Desain Kromosom dan Metode Pendekodean

Pada Algoritma yang kami buat saat ini kami menggunakan representasi biner. Setiap GEN memiliki nilai 0 atau 1. dalam kromosom yang kami rancang memiliki panjang indeks 10, kami memilih 10 indeks di karena kan saat melakukan decode menghasilkan nilai x dan y yang bervariabel, dimana x menjadi 5 indeks dan y menjadi 5 indeks. kami sudah menguji juga bahwa setiap pendekodean tidak akan melebihi batasnya.

## 2.2 Ukuran Populasi

Pada observasi kali ini kami menggunakan populasi dengan jumlah individu sebanyak 20 dan tiap individu kromosom memiliki 10 gen.

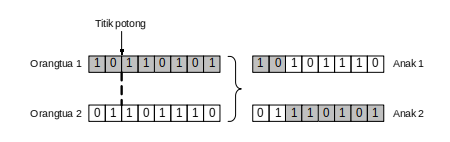
## 2.3 Pemilihan orangtua

Metode yang kami gunakan untuk melakukan pemilihan orang tua adalah tournament selection. Kami memilih tournament selection karena pada penerapannya kami mengambil setengah dari populasi kami yaitu 10 individu sebagai peserta turnamen atau parent, dari 10 individu tersebut akan kami dapatkan dari sorting berdasarkan fitnessnya.

## 2.4 Pemilihan dan teknik operasi genetik (crossover dan mutasi)

### **2.4.1 Crossover (rekombinasi satu titik)**

Pada rekombinasi satu titik, pilih satu titik potong secara acak pada posisi antara 1 hingga G – 1, di mana G adalah jumlah gen dalam kromosom. Misalnya, terpilih titik potongnya adalah posisi T. Selanjutnya, buat Anak 1 dengan gen 1 sampai T diambil dari orangtua 1 dan gen T+1 sampai G diambil dari orangtua 2. Sebaliknya, Anak 2 mendapatkan gen 1 sampai T dari orangtua 2 dan gen T+1 sampai G dari orangtua 1, seperti pada gambar ini.



Pada penerapannya disini kami menggunakan 2 individu atau kromosom yang dimana letak titik potong kami pilih berdasarkan nilai random integer. Setelah kami mendapatkan nilai atau letak titik potong, kami akan mengcopy nilai masing - masing kromosom sebelum titik potong. Saat sudah mencapai letak titik potong maka kromosom yang di copy juga akan berganti untuk mengcopykan gen baru kepada kromosom anak.

### **2.4.2 Mutasi Acak**

Pada metode mutasi ini saya menggunakan metode pemilihan nilai secara acak. pada penerapan pemilihannya menggunakan kondisi jika suatu angka random nilainya lebih kecil dari pada probabilitas yang kami tentukan maka suatu gen dari kromosom yang kami buat akan berubah nilai dari 0 menjadi 1 atau sebaliknya.

## 2.5 Probabilitas operasi genetik (Pc dan Pm)

Untuk probabilitas atau peluang dari operasi genetik ada 2 yaitu crossover dan mutasi. Probabilitas crossover adalah adalah 85% atau 0.85 sehingga individu yang akan di crossover akan lebih banyak menghasilkan populasi. Terhadap proses mutasinya kami menggunakan sebesar 10% atau 0.1 sehingga peluang masing - masing dari tiap kromosom anak berkemungkinan kecil akan terjadi mutasi. Dari 2 probabilitas diatas berkemungkinan akan menghasilkan sebuah generasi baru yang lebih baik atau buruk.

## 2.6 Metode Pergantian Generasi (Seleksi Survivor)

Pada metode pergantian generasi atau seleksi survivor kami menggunakan metode dimana kami akan mempertahankan generasi sebelumnya atau elitisme sebanyak (N - 2), dimana N merupakan banyaknya populasi yang di buat. Dari hasil seleksi orang tua lalu di crossover dan mutasi akan didapatkan populasi anak baru. Populasi anak baru yang telah dibuat akan diambil 2 teratas berdasarkan fitnessnya dan akan menggantikan 2 kromosom atau individu dari populasi sebelumnya sehingga kami mengatakan banyak generasi yang kami pertahankan adalah ( N - 2 ).

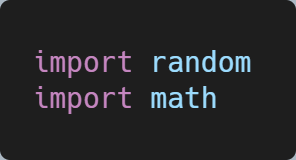
## 2.7 Kriteria Penghentian Evolusi

Pada penghentian evolusi saya menggunakan 1 parameter yaitu ketika sudah sampai pada generasi yang kami minta, dimana kami menggunakan generasi sebanyak 100.

## 2.8 Kode Program

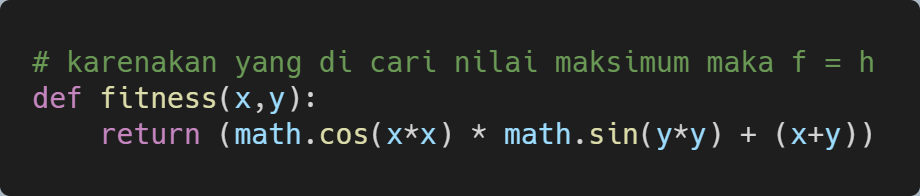
Kami menyusun program untuk algoritma genetika pada tugas pemrograman 1 menggunakan bahasa Python. Berikut kodenya :

### **2.8.1 Library**



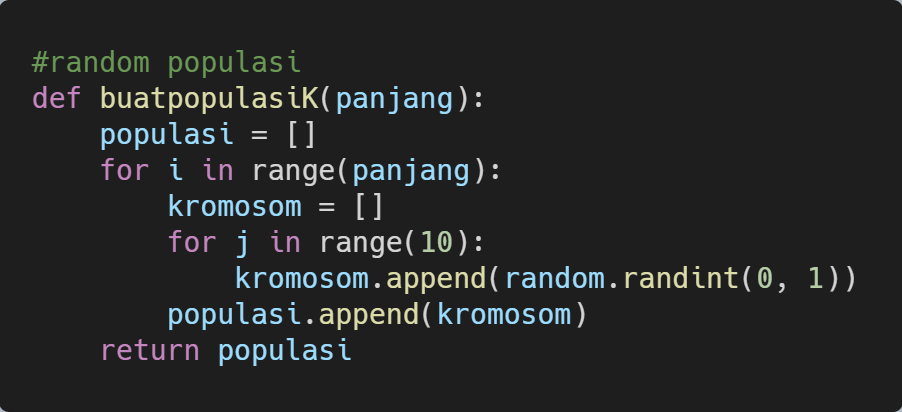
Import random digunakan untuk random angka, Sedangkan import math digunakan untuk menghitung nilai sin cos dan perpangkatan

### **2.8.2 Perhitungan fitness**



Pada prosedur ini kami menerapkan fitness = h karena yang dicari nilai maksimal, dimana rumus yang digunakan untuk mencari h adalah ℎ(𝑥, 𝑦) = (cos 𝑥2 ∗ sin 𝑦2) + (𝑥 + 𝑦) dengan batasan −1 ≤ 𝑥 ≤ 2 dan −1 ≤ 𝑦 ≤ 1.

### **2.8.3 Membuat Populasi**



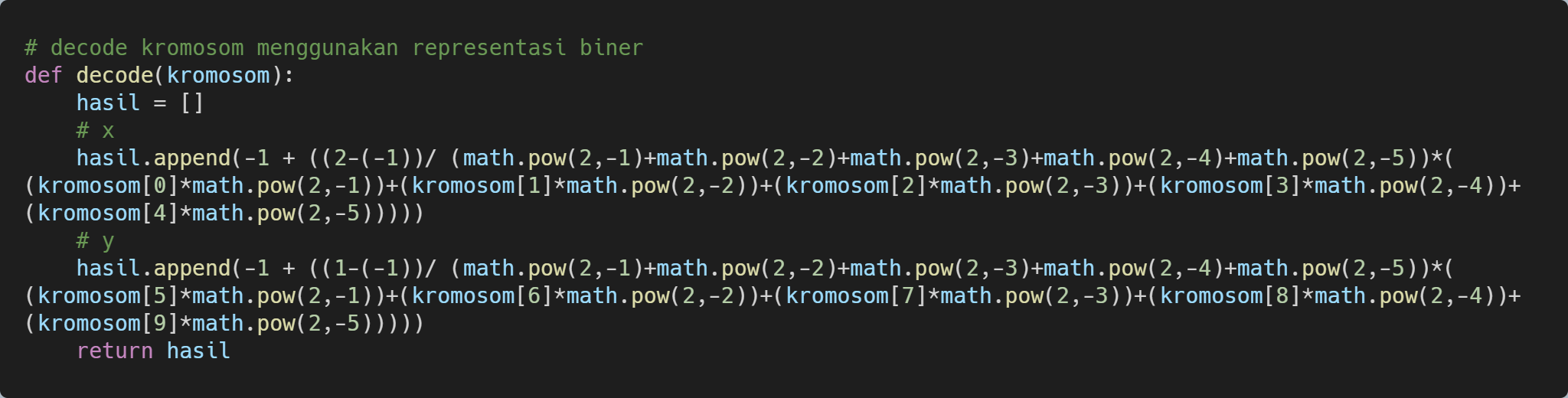
Pada prosedur ini kami menggunakan bantuan library random untuk merandom angka 0 atau 1, dimana angka ini kami susun untuk membuat kromosom yang memiliki gen sebanyak 10

### **2.8.4 Membuat Himpunan Fitness dari Populasi**

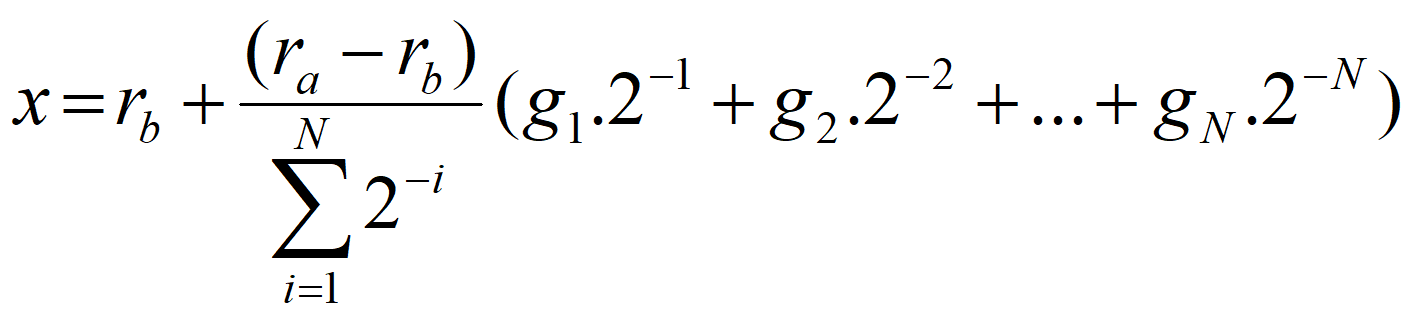


Pada prosedur ini kami menggunakan prosedur decode atau representasi biner untuk mendapatkan nilai x dan y agar bisa menghitung fitness dari masing-masing kromosom di dalam populasi.

### **2.8.5 Decode**

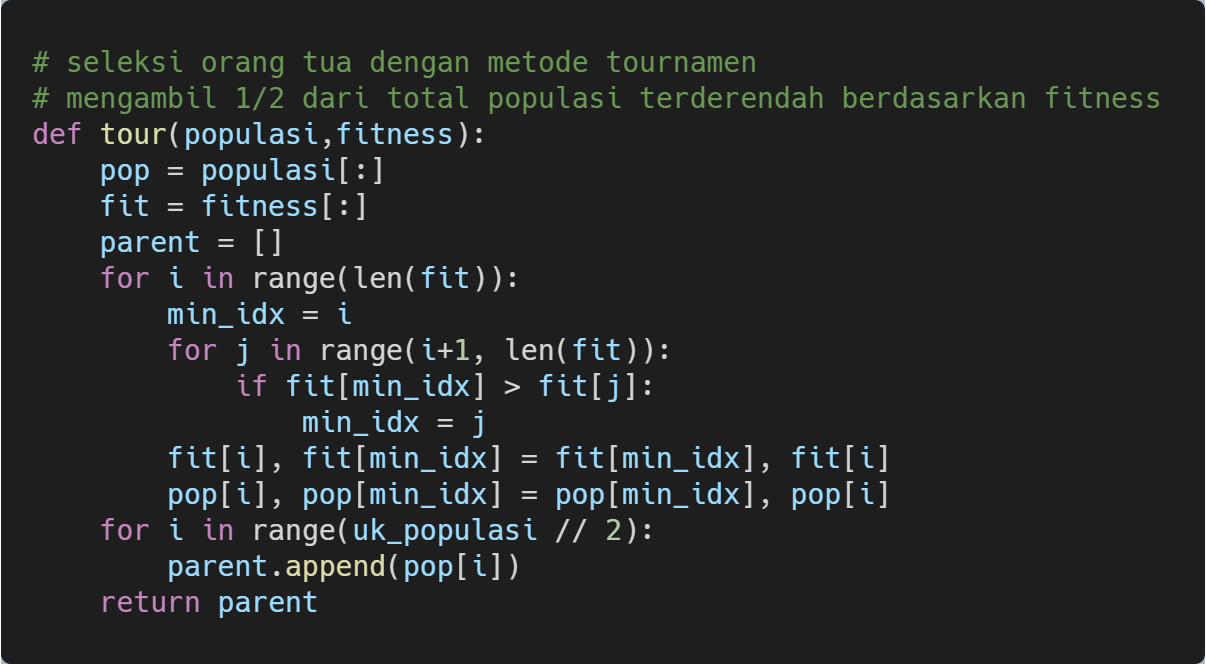


Pada prosedur ini kami menggunakan representasi biner untuk mendapatkan nilai x dan y, dimana rumus representasi biner sebagai berikut :



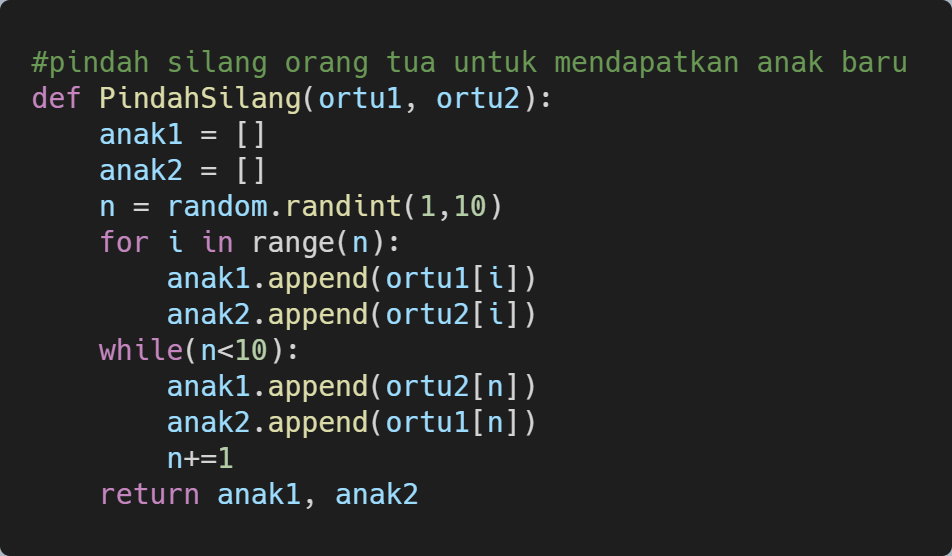
Dimana batas atas untuk nilai x = 2 dan batas atas untuk y = 1 serta untuk batas bawah baik x maupun y sama-sama -1.

### **2.8.6 Tournament**

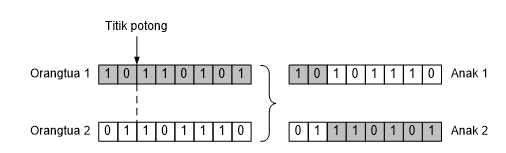


Pada prosedur ini kami akan mengambil setengah dari jumlah populasi dengan menggunakan perbandingan fitness, dimana populasi yang kami ambil yang memiliki fitness dengan nilai menengah kebawah.

### **2.8.7 Pindah Silang**

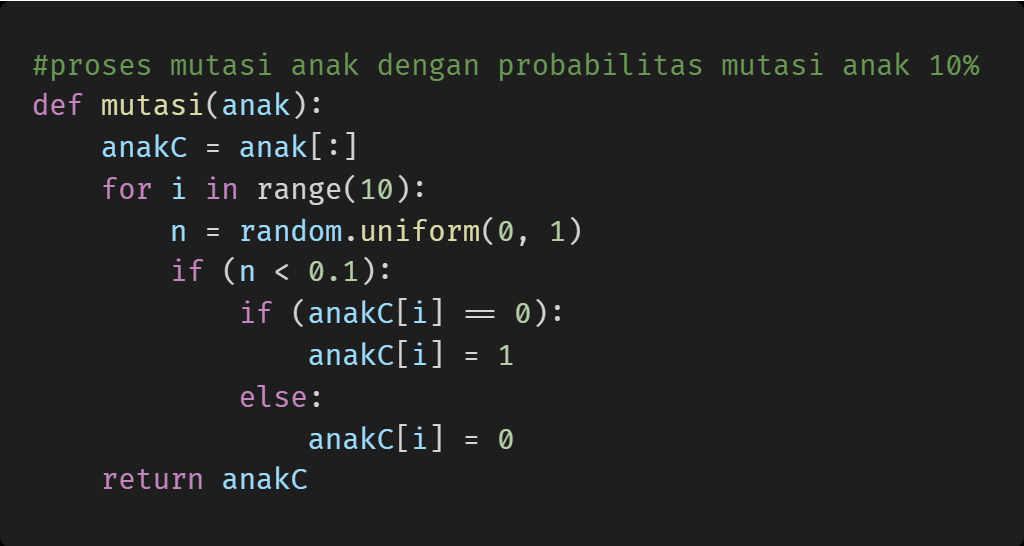


Pada prosedur ini kami menggunakan pindah silang 1 titik dengan gambaran sebagai berikut :



Untuk menentukan titik potong kami menggunakan library random untuk merandom angka dari angka 1 sampai 9.

### **2.8.8 Mutasi**



Pada prosedur ini kami menggunakan probabilitas mutasi sebesar 10% supaya peluang gen dari masing-masing kromosom untuk melakukan mutasi tidak terlalu besar.

### **2.8.9 Mencari Index Minimum Fitness**

|  |
| --- |
|  |

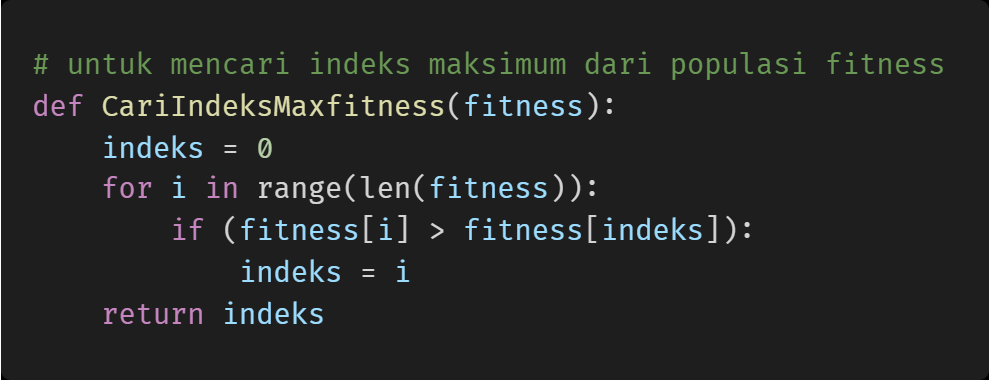
Pada prosedur ini kami mengembalikan nilai indeks letak fitness terkecil.

### **2.8.10 Seleksi Survivor**

|  |
| --- |
|  |

Pada prosedur ini kami menggunakan metode regenerasi yang mempertahankan 18 kromosom terbaik dari populasi sebelumnya (elistisme) dan 2 kromosom yang memiliki fitness terendah akan digantikan oleh populasi kromosom baru yang telah mengalami crossover dan mutasi.

### **2.8.11 Mencari Index Maksimum Fitness**



Pada prosedur ini kami mengembalikan nilai indeks letak fitness terbesar.

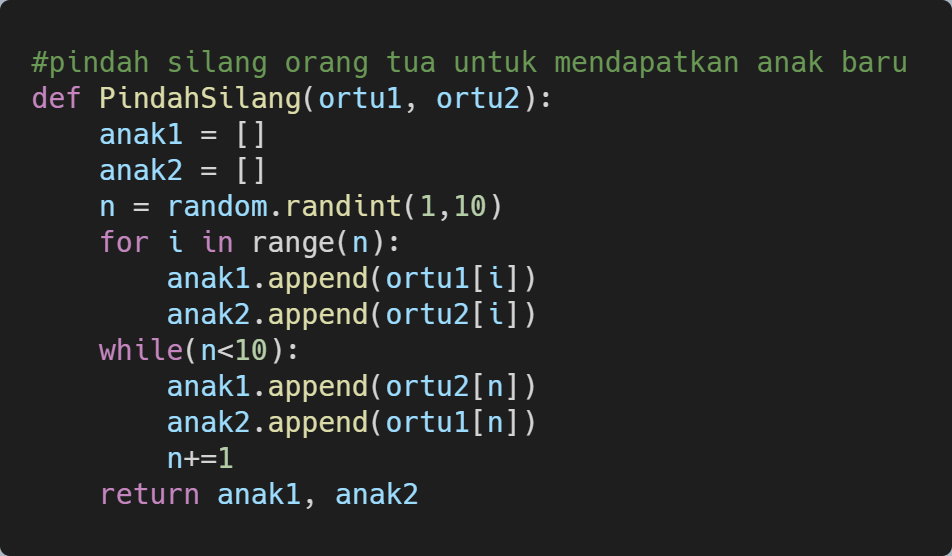
### **2.8.12 Program Utama**



Pada program ini kami menggunakan 20 populasi yang akan dijalankan sebanyak 100 generasi. Dengan urutan pekerjaan :

1. Membuat populasi kromosom.
2. Melakukan pengecekan generasi.
3. Membuat populasi fitness dari populasi kromosom.
4. Mencari kromosom terbaik dari generasi saat ini berdasarkan fitness dan menyimpannya kedalam list best.
5. Melakukan turnamen dengan mengambil setengah dari populasi kromosom yang memiliki fitness menengah kebawah.
6. Masuk kedalam looping untuk melakukan crossover, mutasi dan populasi kromosom baru.
7. Melakukan crossover dengan probabilitas 85%.
8. Melakukan mutasi dengan probabilitas 10%.
9. Memasukkan anak hasil crossover dan mutasi dalam populasi kromosom baru.
10. Kembali ke langkah f sampai semua hasil turnamen telah dimutasi dan di crossover.
11. Buat populasi fitness baru dari populasi kromosom baru.
12. Lakukan regenerasi.
13. Kembali ke langkah b untuk melakukan pengecekan iterasi.
14. Program selesai

## 2.9 Membuat Grafik Perubahan Fitness Terbaik



Prosedur ini digunakan untuk menampilkan grafik perubahan fitness terbaik dari populasi kromosom dari setiap iterasi generasi

## 2.10 Link Video Youtube dan Source Code

Link Google Colab :

<https://colab.research.google.com/drive/1GinpYj6IKUdSqznTUDKTwRJn8E7z6JMr>

Link Video Youtube :

1. Ananda Affan Fattahila (1301194175) : <https://youtu.be/citi33DRrKM>

2. Maulana Nur (1301190402) : <https://youtu.be/GFwnwLnna9o>

3. Zendy Bramantia A (1301194145) : <https://www.youtube.com/watch?v=uq249-ohHSg>

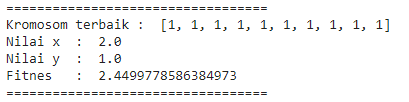
# **BAB 3**

# **PENUTUP**

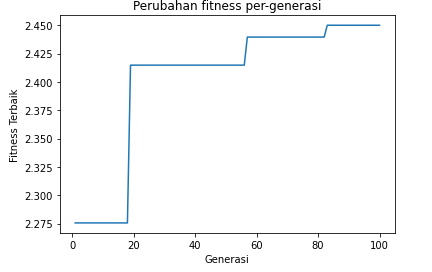
## 3.1 Kesimpulan

Algoritma genetika merupakan teknik pencarian dalam ilmu komputer untuk menyelesaikan optimisasi dan masalah pencarian algoritma ini bersifat evolusioner yang menggunakan teknik pada ilmu biologi seperti warisan (elitisme), mutasi, seleksi alam (seleksi survivor) dan rekombinasi (crossover). Pada tugas ini kami mengimplementasikan algoritma genetika dengan menggunakan bahasa pemrograman python. Desain pada kromosom kami berisikan gen yang memiliki nilai-nilai berupa angka biner yaitu 0 dan 1.

Berdasarkan hal-hal yang telah kami observasi mulai dari desain kromosom, metode pengkodean, ukuran populasi, pemilihan orang tua, teknik operasi genetik, probabilitas operasi genetik, metode pergantian generasi dan kriteria penghentian evolusi. Dari hal-hal yang telah kami sebutkan kami berhasil membangun sebuah program untuk mencari kromosom terbaik dari populasi kromosom yang mengalami evolusi. Salah satu bentuk nilai fitness terbaik atau kromosom terbaik dari program yang kami buat berdasarkan soal tugas pemrograman 1, sebagai berikut :

****

Proses menemukan kromosom terbaik dalam program kami sangatlah random. Oleh karena itu, kami mencoba dengan cara melakukan 100 kali evolusi atau regenerasi untuk mendapatkan kromosom terbaik. Pada setiap iterasi regenerasi kami selalu mencatat kromosom terbaik berdasarkan fitnessnya, jika direpresentasikan pada sebuah grafik, akan menghasilkan grafik seperti di bawah :

****

Namun hasil running dari program kami bisa saja menghasilkan grafik yang berbeda dikarenakan pembuatan populasi secara random.