**PROJECT FINAL**

**ALGORIMA GENETIKA**

Mata Kuliah : Program lanjutan

Dosen Pengampu : Zaitun, S.Si.,M.Mat

Diajukan Untuk Memenuhi Tugas

Oleh

Kelompok 5

**KHIFAYAH NASYWA AL MISRA (231021006) MATEMATIKA**

**NURMADINAH MAHMUD (231021005) MATEMATIKA**

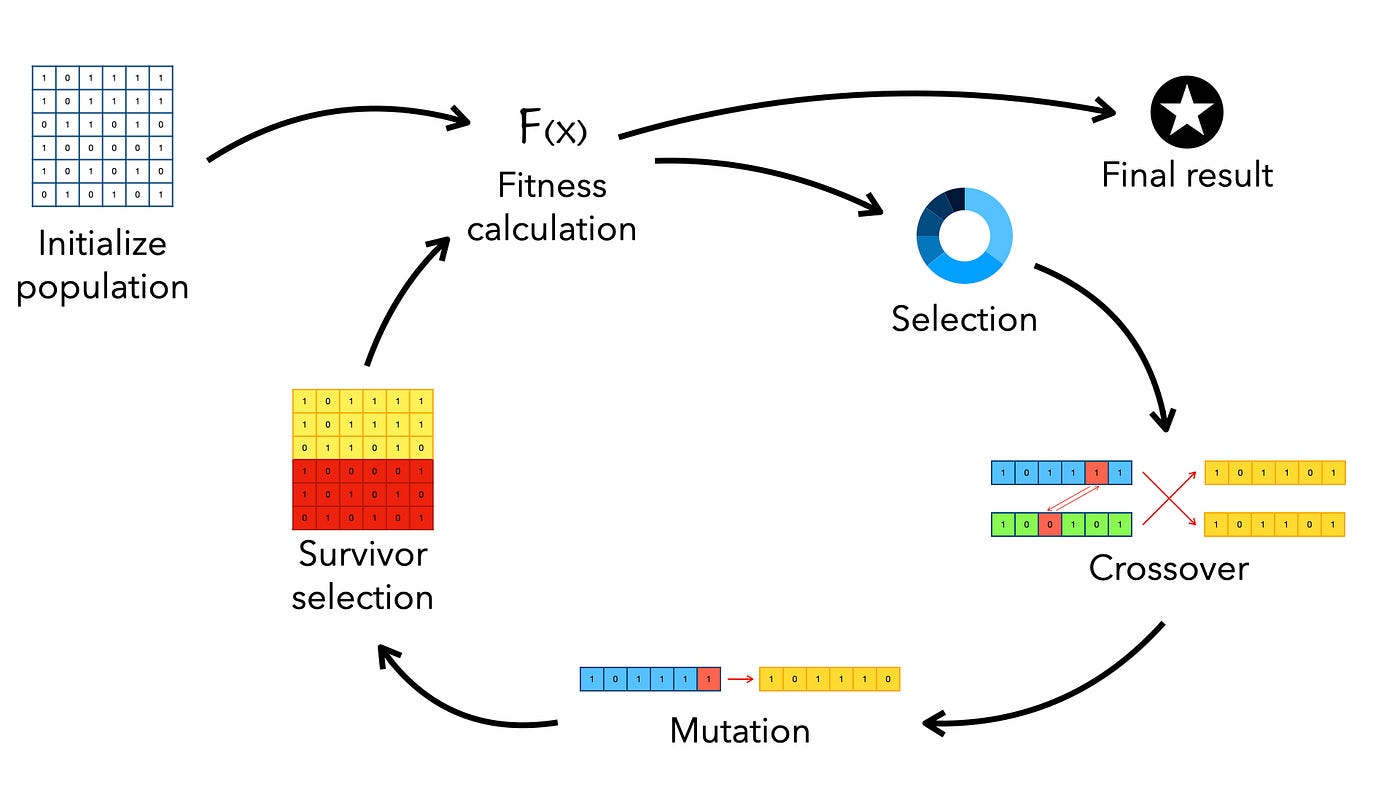
**MICHAEL GERALD**

**INSTITUT TEKNOLOGI BACHARUDDIN JUSUF HABIBIE**

**KOTA PAREPARE**

**2024**

**PENGERTIAN ALGORIMA GENETIKA**

Algoritma genetika didefinisikan sebagai jenis teknik optimasi komputasi yang terinspirasi oleh prinsip seleksi alam dan genetika. Mereka digunakan untuk memecahkan masalah yang kompleks dengan meniru proses evolusi untuk meningkatkan populasi solusi potensial secara berulang.Berikut tahapan algoritma genetika :

CONTOH PROGRAM:

|  |  |
| --- | --- |
| TARGET | PEMROGRAMAN LANJUTAN |
| PANJANG TARGET | 20 |
| POPULASI | 100 |
| GENERASI | 1000 |
| LAJU MUTASI | 0,05 |

**PROSES CROSSOVER DAN MUTASI**

**CROSSOVER :**

Bagian crossover dalam kode ini adalah proses di mana tiga individu terpilih (orangtua) dikombinasikan untuk menghasilkan dua anak baru. Berikut adalah penjelasan spesifik tentang bagaimana bagian crossover bekerja dalam kode:

1. Pemilihan Orangtua

Tiga individu dipilih dari populasi saat ini berdasarkan nilai fitness mereka. Fungsi selection digunakan untuk melakukan seleksi ini. Probabilitas pemilihan setiap individu sebagai orang tua ditentukan berdasarkan nilai fitnessnya. Semakin tinggi nilai fitness individu, semakin besar kemungkinannya dipilih sebagai orang tua.

1. Menentukan Titik Crossover

Tiga titik crossover acak dipilih dari panjang string individu. Dalam kode ini, dua titik crossover acak dipilih menggunakan random.sample(range(1, length), 2). Titik crossover ini digunakan untuk membagi string orang tua dan menggabungkan segmen-segmen dari orang tua yang berbeda untuk menghasilkan anak.

1. Membuat Anak-anak

Dalam implementasi ini, dua anak dihasilkan dari tiga orang tua yang dipilih. Setiap anak dibuat dengan menggabungkan segmen-segmen dari tiga orang tua berdasarkan titik crossover yang telah ditentukan sebelumnya. Berikut adalah bagaimana anak-anak ini dibuat:

* Anak 1 : Bagian awal dari orangtua 1 sampai titik crossover pertama, bagian tengah dari orangtua 2 antara titik crossover pertama dan kedua, bagian selanjutnya diambil dari orang tua 3 antara titik crossover kedua dan ketiga.
* Anak 2 : Bagian awal dari orangtua 2, bagian tengah diambil dari orang tua 3, bagian selanjutnya diambil dari orang tua 1.

Berikut adalah potongan kode yang melakukan proses ini:

def crossover(parents):

    length = len(parents[0])

    crossover\_points = sorted(random.sample(range(1, length), 2))

    children = []

    for i in range(children\_count):

        child = (parents[i % parents\_count][:crossover\_points[0]] +

                 parents[(i + 1) % parents\_count][crossover\_points[0]:crossover\_points[1]] +

                 parents[(i + 2) % parents\_count][crossover\_points[1]:])

        children.append(child)

    return children

Mari kita jabarkan lebih rinci:

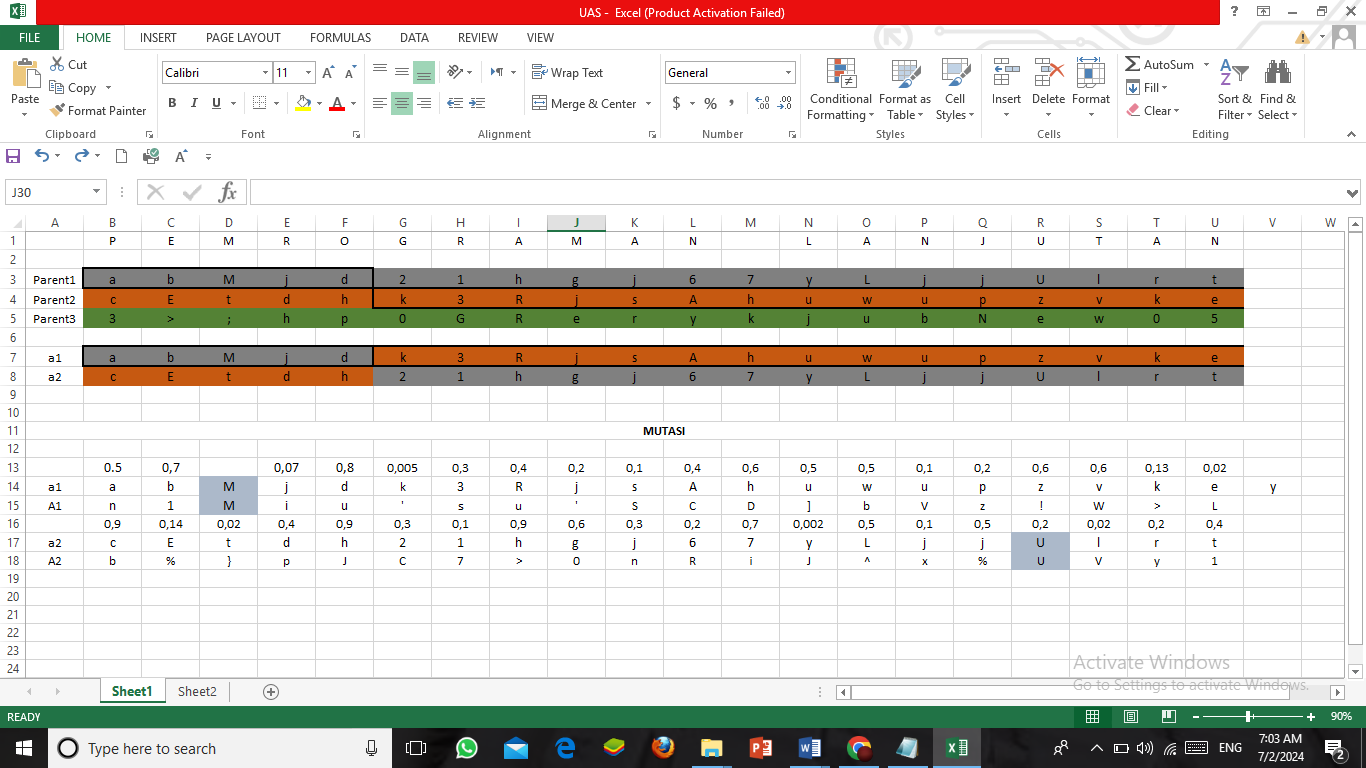
1. Crossover\_Points:Titik crossover adalah posisi dalam string di mana segmen dari orangtua berbeda akan bergabung. Misalnya, jika panjang string adalah 20 dan titik crossover yang dipilih adalah [5, 10, 15], maka string akan dipotong menjadi empat segmen.
2. Loop untuk Membuat Anak-anak

i % parents\_count: Menentukan orangtua mana yang akan digunakan untuk segmen tertentu.

1. Segmen Anak: Setiap anak adalah kombinasi dari segmen orangtua yang berbeda berdasarkan titik crossover yang telah ditentukan.

**MUTASI :**

Contoh konkret Crossovernya:



Dari mutasi diatas, jika nilai random lebih kecil atau sama dengan (<=) 0.05 maka terjadi mutasi dan jika nilai random lebih besar dari 0.05 maka tidak terjadi mutasi.

**HASIL PERCOBAAN**

