**KECERDASAN BUATAN**

**LAPORAN TUGAS PARALEL 1**

**ALGORITMA GENETIKA**

1. Kromosom

Panjang kromosom yang digunakan pada tugas paralel 1 ini sepanjang 8, dengan setiap gen akan diisi dengan angka acak dari 0-10. Dipilih 8 karena untuk menyingkat waktu komputasi. Terdapat pada fungsi chrom() dalam file program IF4102\_1301174354.py.

1. Populasi

Banyak kromosom dalam satu populasi adalah 10. Dipilih 10 karena agar tidak memakan banyak waktu saat proses komputasi. Terdapat pada fungsi populesyen() dalam file program IF4102\_1301174354.py.

1. Fitness

Dalam tugas paralel 1 ini terdapat 2 tahap dalam mencari *fitness.*

Pertama kromosom yang di-*generate* terlebih dahulu dilakukan proses *encode* dengan cara memotong kromosom menjadi 2 bagian yang sama panjang, kemudian masing-masing bagian dimasukan ke fungsi berikut :

Rmax dan Rmin adalah batas atas dan batas bawah tiap x1,x2. Dalam tugas paralel 1 ini batas atas dan batas bawah tiap x adalah -3 ≤ x1 ≤ 3 dan -2 ≤ x2 ≤ 2.

Kemudian langkah selanjutnya adalah memasukan hasil x1,x2 ke dalam fungsi yang akan dicari minimasinya, yaitu :

Setelah mendapat hasil fungsi diatas, masukan hasil fungsi tersebut kedalam rumus berikut :

Dengan a adalah angka positif bebas(dalam tugas ini digunakan 0.01) sehingga didapat *fitness* dari sebuah kromosom. Semua penjelasan fungsi *fitness* diatas tertuang dalam fungsi fitness() dalam file program IF4102\_1301174354.py

1. Parent Selection

Dalam tugas paralel 1 ini digunakan metode *Tournament* untuk proses seleksi. Metode *Tournament* adalah proses pemilihan *parent* dengan cara membandingkan fitness tiap kromosom dengan fitness kromosom lainnya(yang dipilih secara acak) dalam populasi yang sama yang kemudian akan mendapatkan 2 *parent* terbaik dalam sebuah populasi. Terdapat pada fungsi parent\_sel() dalam file program IF4102\_1301174354.py.

1. Crossover

Proses c*rossover* adalah proses untuk menggabung 2 *parent* yang telah diseleksi dengan cara memotong kromosom *parent* dititik tertentu kemudian menukar bagian yang terpotong tersebut ke *parent* lainnya dengan probabilitas yang ditentukan. Dalam tugas paralel 1 ini probabilitas yang digunakan adalah 0.02. Proses ini menghasilkan 2 kromosom baru yang disebut *child*. Terdapat pada fungsi crossover() dalam file program IF4102\_1301174354.py.

1. Mutation

Proses untuk memutasi *child* yang didapat dari proses *crossover* dengan cara mengganti gen kromosom *child* pada index acak dengan angka acak. Terdapat pada fungsi mutation() dalam file program IF4102\_1301174354.py.

1. Generate New Population

Dalam genetika algoritma proses awalnya adalah membuat populasi yang kemudian populasi itu akan dilakukan proses 3-6 hingga dalam populasi baru hanya kromosom terbaik yang akan muncul. Terdapat pada fungsi newpop\_generator() dan Main program dalam file program IF4102\_1301174354.py.

