

DESAIN KROMOSOM

Desain kromosom menggunakan Decision Tree suatu metode klasifikasi paling populer. Konsep Decision Tree mengubah data yang dibuat menjadi decision Tree dan aturan keputusannya. Pohon Keputusan dapat mengeliminasi perhitungan-perhitungan yang tidak diperlukan, karena ketika menggunakan metode pohon keputusan maka contoh diuji hanya berdasarkan kriteria.

Analisis Pemrograman

suatu populasi mempunyai beberapa kromosom dan didalam kromosom tersebut terdapat beberapa rule. file csv disimpan dalam satu folder pengerjaan dan didalam program dengan fungsi `openfile()` yaitu berguna untuk membaca file csv bernama **“data_latih_opsi_1.csv”** yang kemudian ditampung ke dalam bentuk array.

pada fungsi `rule()` menggunakan biner dimana 1 rule mempunyai 15 biner , suatu biner statis yang dapat dipilih sesuai kelipatan 15. kromosom yang dimiliki oleh suatu populasi dapat di *set* sebanyak kromosom yang diinginkan.

suatu panjang kromosom yang akan dibandingkan harus dipecah menjadi dengan masing-masing 1 rule(biner 15) dengan fungsi `splitter()` yang akan memecah tiap kelipatan 15.

dekode pada program ini yang saya gunakan yaitu men-dekode data kromosom yang telah terandom dengan memanggil `spliterrule()` akan mensplit di tiap rule berdasarkan Suhu, waktu, Kodisi langit, kelembapan, Terbang/tidak dan mempunyai banyak kondisi yang akan diubah dari integer menjadi string. Proses dekode dapat diperhatikan pada fungsi `decode()`, di program menggunakan data kromosom diubah menjadi string. Satu rule mempunyai 5 bentuk keadaan yaitu:

1. Suhu(normal,rendah,tinggi).
2. Waktu(pagi,siang,sore,malam).
3. Kondisi langit(berawan,cerah,hujan,rintik).
4. Kelembapan(normal,rendah,tinggi).
5. Terbang/tidak(Ya,tidak).

untuk mencari fitness bandingkan antara file di csv dengan data yang telah di dekode dengan memanggil fungsi `matching()` disitu menggunakan perulangan `for` untuk mencari data yang sama, apabila banyak data tersebut dicocokkan apabila sama disalah satu rule dalam kromosom benar maka menambah satu dan melakukan perulangan selanjutnya.

pemilihan parent dengan menggunakan roulette wheel dan dipanggil dengan fungsi `roulette()` dipanggil dua kali dengan variabel berbeda, maka dapat dua parent.

crossover yang dilakukan dengan mengetahui panjang array pada `parent1` dan `parent2`. setelah itu random bilangan antara `parent1` dan `parent2` maka didapatkan nilainya titik potong dan disimpan pada variabel, gunakan rumus yang terdapat pada ppt terdapat kondisi apabila hasil titik potong diantara rule 1 dan rule 2 maka menggunakan swap multiple pada crossover. apabila crossover di titik rule pertama saja maka tukar sesuai aturan. dan crossover di titik rule 2 maka tukar sesuai aturan juga.

screenshot crossover aturan ke 2:

```
('ss', [1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1])
('ss', [0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1])
```

mutasikan data pilih titik point secara random `parent1` dan `parent2`. probabilitas=0,7 apabila probabilitas lebih kecil dari data randomnya maka mutasi dilakukan, apabila isi dari array tersebut 0 maka diubah menjadi 1 begitupun sebaliknya. generasikan sampai berapa generasi.

```
0.1125
('kromosom terbaik', [[1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1], [1, 0, 0, 1, 0, 0, 1, 0,
0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1]]
[Finished in 0.1s]
```