

Pendekatan Algoritma Genetika Dalam Bidang Learning Untuk Klasifikasi Data

Muhammad Shabri Arrahim Mardi

Alamat : Jl. Perintis No. 46A, Baubau, Sulawesi Tenggara
e-mail: shabri@student.telkomuniversity.ac.id

ABSTRAK

Tulisan ini dimaksudkan untuk memperlihatkan proses kerja Algoritma Genetika pada masalah optimasi yang menggunakan kode biner. Tujuannya adalah menemukan klasifikasi dari sebuah data set. Berdasarkan simulasi yang disusun dalam bentuk program komputer menggunakan Bahasa pemrograman Python diperoleh nilai nilai akurasi sebesar 0.9 (90%) dari data training yang diberikan. Dengan nilai *chromosom* = [1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0].

Kata kunci: Algoritma Genetika, Nilai Minimasi, Fungsi Fitness

1. PENDAHULUAN

Decision tree adalah salah satu metode klasifikasi yang paling populer karena mudah diinterpretasikan oleh manusia. Decision tree digunakan untuk pengenalan pola dan termasuk dalam pengenalan pola secara static. Decision tree dibentuk dari 3 tipe dari simpul:

- Simpul leaf memuat suatu keputusan akhir atau kelas target untuk suatu pohon keputusan.
- Simpul root adalah titik awal dari suatu decision tree.
- Setiap simpul perantara berhubungan dengan suatu pertanyaan atau pengujian.

Dalam laporan ini digunakan data training dengan row data sebanyak 20 dan terdapat 5 atribut dengan 1 atribut yang menajki kelas yang harus ditebak.

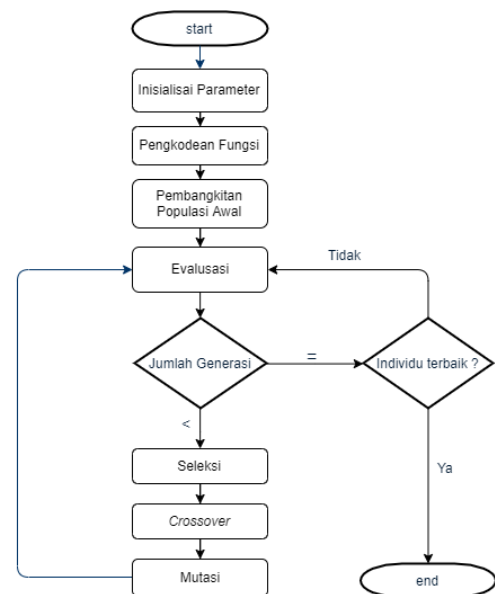
2. METODE

Sebagai algoritma evolusioner, Algoritma Genetika melakukan *learning* dengan meniru evolusi yang terjadi di dunia nyata. Proses diawali dengan membangkitkan populasi yang terdiri dari sekumpulan individu yang selanjutnya disebut kromosom. Kromosom tersusun atas satuan terkecil yang disebut gen, sedangkan posisi gen dalam kromosom disebut lokus. Nilai dari gen disebut allele. Setiap kromosim memiliki nilai ukuran yang disebut nilai *fitness*.

Proses evolusi dilakukan pertama kali dengan menyeleksi kromosom. Seleksi dilakukan berdasarkan nilai *fitness*, kromosom dengan nilai *fitness* yang tinggi memiliki peluang lebih besar terseleksi untuk dilakukan *crossover*. Setelah proses seleksi, akan dipilih 2 individu untuk dilakukan proses penyilangan berdasarkan titik potong yang dipilih secara acak (*crossover*). Berikutnya dilakukan proses *mutasi* yaitu dengan mengganti allele dari suatu lokus (*gen*). Proses ini akan terus berulang sampai mendapatkan populasi dengan kromosom yang dianggap '*sempurna*'.

3. PEMBAHASAN DAN HASIL

Proses pencarian solusi dalam Algoritma Genetika dengan alur sebagai berikut :



Gambar 1 Proses Pencarian Solusi GA

1. Inisialisasi Parameter

Ini inisialisasi parameter dilakukan dengan cara merubah string biner menjadi panjang nilai-nilai sesuai atribut yang berada pada dataset.

Selanjutnya transformasi dari string biner ke dalam nilai-nilai dari atribut yang terdapat pada dataset

Pada kasus ini Panjang bit yang digunakan untuk 5 atribut adalah 15. Maka Panjang string untuk setiap kromosom didalam populasi adalah $n = 15$. Setiap biner akan di split ke 5 bagian menyesuaikan dengan banyak atribut pada data set. Contoh: **1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0** merupakan representasi dari pasangan atribut Suhu sebanyak 3 bit, Waktu sebanyak 4 bit, Kondisi Langit sebanyak 4 bit,, Kelembapan sebanyak 3 bit, Terbang/Tidak sebanyak 1 bit.

2. Pengkodean fungsi

Pada tahap ini fungsi – fungsi yang dibutuhkan didalam program dibangun. Berikut merupakan contoh fungsi *fSelection* pada program :

```

114 def Selection(fitness):
115     i = 0
116     kumfit = 0
117     while i < len(fitness):
118         kumfit = kumfit + fitness[i]
119         if (kumfit/fitness.sum()) > np.random.uniform(0,1):
120             index = i
121             break
122         i=i+1
123     index = i
124     return index

```

3. Pembangkitan populasi awal

Populasi awal dibangun dengan masing – masing panjang himpunan string biner sebesar kelipatan 15. Dan dihitung menjadi 1 buah aturan yang akan digunakan untuk menebak *class* dari data input.

4. Evaluasi

Evaluasi dilakukan dengan cara menghitung nilai *fitness* (akurasi dari masing-masing kromosom dari data training yang diberikan) setiap kromosom/individu. Berikut nilai contoh *fitness* dari setiap individu pada Tabel 1 :

Tabel 2. Nilai Fitness

Kromosom	Nilai Fitness (Nilai Akurasi)
1	0.9
2	1.0
3	0.95

5. Proses Seleksi

Proses seleksi dilakukan dengan menghitung nilai fitness, total fitness, peluang fitness dan peluang kumulatif fitness untuk setiap kromosom/individu. Pemilihan individu yang akan terseleksi dilakukan dengan membangkitkan bilangan random $\text{rand}(0,1)$. Jika $(\text{kumulatiffitness}[i]/\text{fitness.sum}()) > \text{rand}(0,1)$ maka individu ke-i yang terpilih.

6. Crossover

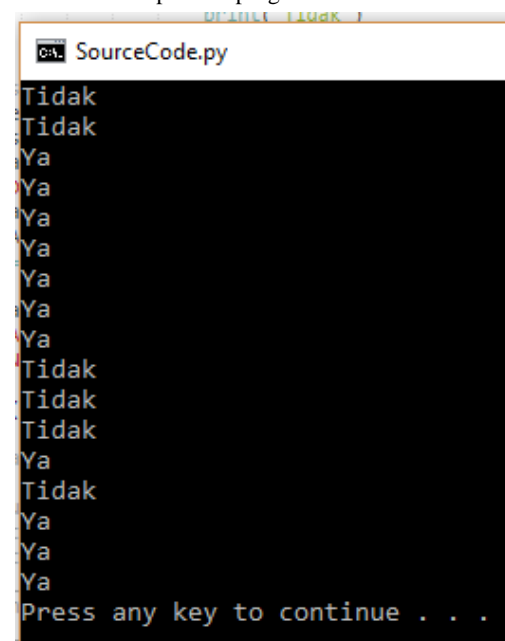
Crossover dilakukan dengan mengambil beberapa titik (*point*) sebagai titik penyilangan untuk menentukan pemotongan kromosom. *Crossover* terjadi jika nilai random yang dibangkitkan tidak lebih dari *cross_rate* yang telah di inisialisai, yaitu sebesar 0.8 (80%). $\text{np.random.rand}() < \text{CROSS_RATE}$.

7. Mutasi

Dalam kasusu ini *mutation_rate* di inialisasi sebesar 1.6, dengan artian sebesar 1.6 pada gen dalam populasi akan mengalami mutasi. Mutasi terjadi jika nilai random yang dibangkitkan tidak lebih dari *mutation_rate*. $\text{np.random.rand}() < \text{MUTATION_RATE}$. Jika gen bernilai 1 akan diganti menjadi 0 dan begitu juga sebaliknya.

Berikut ini merupakan solusi hasil klasifikasi dari 20 row data set uji.

Screenshot output dari program :



4. KESIMPULAN

Algoritma Genetika tidak hanya digunakan untuk mencari sebuah nilai dari suatu fungsi saja (metode : *seraching*), tetapi dapat juga digunakan untuk membangun sebuah system Decision Tree yang digunakan untuk mengklasifikasikan suatu dataset (metode : *learning*).

Semakin banyak algoritma melakukan training dari data, maka akan semakin tinggi akurasi dari algoritma tersebut untuk menentukan kelas dari sebuah inputan data.