**TUGAS UTS**

**MANAJEMEN SUPPORT SISTEM**

**“ALGORITMA GENETIK”**



**Oleh :**

**I Gusti Putu Urip Yogantara**

**1504505053**

**Manajemen Support Sistem (B)**

**JURUSAN TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS UDAYANA**

**2016**

**ALGORITMA GENETIKA**

Algoritma genetika adalah algoritma komputasi yang diinspirasi teori evolusi yang kemudian diadopsi menjadi algoritma komputasi yang biasa digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi. Algoritma ini didasarkan pada proses genetik yang ada dalam makhluk hidup; yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam atau “siapa yang kuat, dia yang bertahan (*survive*)”. Dengan meniru teori evolusi ini, algoritma genetika dapat digunakan untuk mencari solusi permasalahan-pemasalahan dalam dunia nyata.

1. **Hal-Hal yang Harus Dilakukan dalam Algoritma Genetika**
2. Mendefinisikan individu, dimana individu menyatakan salah satu solusi (penyelesaian) yang mungkin dari permasalahan yang diangkat.
3. Mendefinisikan nilai *fitness*, yang merupakan ukuran baik-tidaknya sebuah individu atau baik-tidaknya solusi yang didapatkan.
4. Menentukan proses pembangkitan populasi awal. Hal ini biasanya dilakukan dengan menggunakan pembangkitan acak seperti *random-walk*.
5. Menentukan proses seleksi yang akan digunakan.
6. Menentukan proses perkawinan silang (*cross-over*) dan mutasi gen yang akan digunakan.
   1. **Pengertian Individu**

Individu mernyatakan salah satu solusi yang mungkin. Individu bisa dikatakan sama dengan kromosom yang merupakan kumpulan gen. Gen ini bisa biner, float, dan kombinatorial.

Beberapa definisi penting yang perlu diperhatikan di dalam mendefinisikan individu untuk membangun penyelesaian permasalahan dengan algoritma genetika adalah sebagai berikut.

* 1. Genotype (Gen), sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam sau kesatuan gen yang dinamakan kromosom.
  2. Allele merupakan nilai dari gen.
  3. Kromosom, merupakan gabungan dari gen-gen yang membentuk nilai tertentu.
  4. Individu, menyatakn suatu nilai ataukeadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
  5. Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam siklus proses evolusi.
  6. Generasi, menyatakan satu siklus proses evolusi atau satu iterasi dalam algoritma genetika.

|  |
| --- |
|  |

**Gambar 1.1** Ilustrasi Representasi Penyelesaian Permasalahan dalam Algoritma Genetika

* 1. **Nilai Fitness**

Nilai fitness adalah nilai yang menyatakan baik tidaknya suatu solusi (individu). Nilai fitness ini yang dijadikan acuan dalam mencapai nilai optimal dalam algoritma genetika. Algoritma genetika bertujuan mencari individu dengan nilai fitness yang paling tinggi.

1. **Komponen Utama Algoritma Genetika**

Terdapat 6 komponen utama dalam algoritma genetika. Komponen-komponen tersebut antara lain sebagai berikut.

* 1. **Teknik Pengkodean**

Teknik pengkodean adalah bagaimana mengkodekan gen dari kromosom, dimana gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variabel.

Gen dapat direpresentasikan dalam bentuk bit, bilangan real, daftar aturan, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan untuk operator genetika.

* 1. **Membangkitkan Populasi Awal**

Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Ukuran untuk populasi tergantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian dilakukan pembangkitan populasi awal.

* 1. **Seleksi**

Seleksi digunakan untuk memilih individu-individu mana saja yang akan dipilih untuk proses kawin silang atau mutasi. Seleksi digunakan untuk mendapatkan calon induk yang baik. “Induk yang baik akan menghasilkan keturunan yang baik”. Semakin tinggi nilai fitness suatu individu semakin besar kemungkinannya untuk terpilih.

Terdapat beberapa metode seleksi, yaitu metode mesin roullete, dan turnamen. Adapun penjelasannya sebagai berikut.

* + 1. **Seleksi dengan Mesin Rolette**

Metode seleksi dengan mesin roulette merupakan metode yang paling sederhana dan sering dikenal dengan nama stochastic sampling eith replacement. Cara kerja metode ini adalah sebagai berikut.

1. Dihitung nilai firness dari masing-masing individu (fi , dimana i adalah individu ke-1 s/d ke-n)
2. Dihitung total fitness semua individu.
3. Dihitung probabilitas masing-masing individu.
4. Dari probabilitas tersebut, dihitung jatah masing-masing individu pada angka 1 sampai 100.
5. Dibangkitkan bilangan random antara 1 sampai 100.
6. Dari bilangan random yang dihasilkan, ditentukan individu mana yang terpilih dalam proses seleksi.
   * 1. **Seleksi dengan Turnamen**

Metode seleksi dengan turnamen, ditetapkan suatu nilai tour untuk individu-individu yang dipilih secara random dari suatu populasi. Individu-individu yang terbaik dalam kelompok ini akan diseleksi sebagai induk. Parameter yang digunakan pada metode ini adalah ukuran tour yang bernilai antara 2 sampai N (jumlah individu dalam suatu populasi).

* 1. **Pindah Silang (Crossover)**

Kawin silang (crossover) adalah operator dari algoritma genetika yang melibatkan dua induk untuk membentuk kromosom baru. Pindah silang menghasilkan titik baru dalam ruang pencarian yang siap untuk diuji. Operasi ini tidak selalu dilakukan pada semua individu yang ada. Individu dipilih secara acak untuk dilakukan crossing dengan Pc antara 0.6 sampai dengan 0.95. Jika pindah silang tidak dilakukan, maka nilai dari induk akan diturunkan kepada keturunan.

Prinsip dari *crossover* ini adalah melakukan operasi (pertukaran, aritmatika) pada gen-gen yang bersesuaian dari dua induk untuk menghasilkan individu baru. Proses crossever dilakukan pada setiap individu dengan probabilitas crossover yang ditentukan. Probabilitas *crossover* pada proses *crossover* dapat dilihat pada Gambar 2.1.

|  |
| --- |
|  |

**Gambar 2.1** Probabilitas Crossover

Operator *crossover* bergantung pada representasi kromosom yang dilakukan. Berbagai model *crossover* sesuai dengan representasi kromosom antara lain sebagai berikut.

* + 1. **Crossover Satu Titik**

Crossover satu titik dan banyak titik biasanya digunakan untuk representasi kromosam dalam biner dalam biner. Pada crossover satu titik, posisi crossover k (k=1, 2, ...., N-1) dengan N=panjang kromosom diseleksi secara random. Variabel-variabel ditukar antar kromosom pada titik tersebut untuk menghasilkan anak. Ilustrasi crossover dapat dilihat pada Gambar 2.2.

|  |
| --- |
|  |

**Gambar 2.2** Ilustrasi Crossover Satu Titik.

* + 1. **Crossover Banyak Titik**

Pada crossover banyak titik, m posisi penyilangan ki (k=1, 2, ...., N-1, i=1, 2, ...., m) dengan N= panjang kromosom diseleksi secara random dan tidak diperbolehkan ada posisi yang sama, serta diurutkan naik. Variabel-variabel ditukar anta kromosom pada titik tersebut untuk menghasilkan anak. Ilustrasi crossover banyak titik dapat dilihat pada Gambar 2.3.

|  |
| --- |
|  |

**Gambar 2.3** Ilustrasi Crossover Banyak Titik

* 1. **Mutasi**

Mutasi berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak dapat muncul pada inisialisasi populasi. Kromosom anak dimutasi dengan menambahkan nilai random yang sangat kecil, dengan probabilitas rendah. Peluang mutasi (pm) didefinisikan sebagai persentasi dari jumlah total gen pada populasi yang mengalami mutasi. Peluang mutasi mengendalikan banyaknya gen baru yang akan dimunculkan untuk dievaluasi. Jika peluang mutasi terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Tetapi bila peluang mutasi ini terlalu besar, maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dari induknya, dan juga algoritma kehilangan.

1. **Contoh Penerapan Algoritma Genetika untuk Permasalahan Optimalisasi Distribusi Barang Dua Tahap**
   1. **Metode Penelitian dan Perancangan**

Pada penelitian dalam bidang perangkat lunak, umumnya menggunakan tahapan-tahapan penelitian. Gambar 3.1 adalah tahapan-tahapan penelitian untuk membangun sistem*.*

|  |
| --- |
|  |

**Gambar 3.1** Tahapn-tahapan penelitian

Berdasarkan Gambar 3.1, penjelasan mengenai tahapan penelitian mengenai optimasi distribusi barang dua tahap menggunakan algoritma genetika adalah :

* 1. Mengumpulkan berbagai studi literatur mengenai metode algoritma genetika, jurnal dan literatur lain yang terkait dengan optimasi distribusi barang dua tahap.
  2. Menganalisa dan merancang perangkat lunak untuk optimasi distribusi barang dua tahap dengan menggunakan algoritma genetika.
  3. Mengimplementasikan hasil analisis dan perancangan yang telah dilakukan dalam bentuk perangkat lunak untuk optimasi distribusi barang dua tahap.
  4. Melakukan uji coba terhadap perangkat lunak untuk distribusi dua tahap dengan menggunakan algoritma genetika.
  5. Mengevaluasi hasil dari sistem yang dibangun untuk distribusi dua tahap berdasarkan kondisi nyata.
     1. **Data Penelitian**

Optimasi distribusi barang dua tahap menggunakan data berupa jumlah pabrik, distributor dan agen, data kapasitas persediaan pabrik, data permintaan dari distributor dan agen. Pada sistem ini pengaruh utama pada distribusi berupa banyaknya kapasitas barang yang akan didistribusikan dan biaya yang akan dikeluarkan pada proses pendistribusian barang. Digunakan pabrik sebanyak 2 buah, distributor 5 buah dan agen sebanyak 10. Data distribusi barang dua tahap yang akan digunakan pada penelitian ini berupa tabel yang berisi jumlah produksi, jumlah permintaan, dan tabel biaya distribusi. Data distribusi dua tahap pada penelitian ini dijelaskan pada Tabel 3.1 sampai 3.3 mengenai data kapasitas produksi pabrik dan permintaan distributor dan agen.

**Tabel 3.1** Data Pabrik dan Kapasitas Persediaan Pabrik

|  |  |  |
| --- | --- | --- |
| **Id Pabrik** | **Nama Pabrik** | **Kapasitas Pabrik** |
| 1 | Pabrik 1 | 300 |
| 2 | Pabrik 2 | 400 |

**Tabel 3.2** Data Distributor dan Kapasitas Distributor

|  |  |  |
| --- | --- | --- |
| **Id Distributor** | **Nama Distributor** | **Kapasitas Distributor** |
| 1 | Distributor 1 | 300 |
| 2 | Distributor 2 | 400 |
| 3 | Distributor 3 | 250 |
| 4 | Distributor 4 | 600 |
| 5 | Distributor 5 | 450 |

**Tabel 3.3** Data Agen dan Kapasitas Permintaan Agen

|  |  |  |
| --- | --- | --- |
| **Id Pabrik** | **Nama Agen** | **Permintaan Agrn** |
| 1 | Pabrik 1 | 150 |
| 2 | Pabrik 2 | 175 |
| 3 | Pabrik 3 | 200 |
| 4 | Pabrik 4 | 225 |
| 5 | Pabrik 5 | 200 |
| 6 | Pabrik 6 | 150 |
| 7 | Pabrik 7 | 150 |
| 8 | Pabrik 8 | 300 |
| 9 | Pabrik 9 | 200 |
| 10 | Pabrik 10 | 250 |

Setelah dijabarkan mengenai data produksi dan permintaan, pada distribusi dua tahap dibutuhkan pula data biaya distribusi untuk pengiriman barang dari pabrik menuju distributor dan dari distributor menuju agen. Data biaya dijelaskan pada Tabel 3.4 mengenai data distribusi pabrik ke distributor dan Tabel 3.5 mengenai data distribusi dari distributor ke agen.

**Tabel 3.4** Biaya Distribusi Pabrik ke Distributor

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Node** | **1** | **2** | **3** | **4** | **5** |
| 1 | 27 | 32 | 10 | 17 | 16 |
| 2 | 14 | 12 | 29 | 12 | 29 |

**Tabel 3.5** Biaya Distribusi Distributor ke Agen

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Node | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 20 | 23 | 14 | 2 | 68 | 16 | 21 | 53 | 25 | 12 |
| 2 | 62 | 66 | 61 | 26 | 55 | 67 | 57 | 24 | 13 | 26 |
| 3 | 13 | 59 | 39 | 71 | 66 | 19 | 11 | 11 | 40 | 12 |
| 4 | 8 | 14 | 76 | 51 | 79 | 19 | 54 | 23 | 6 | 61 |
| 5 | 19 | 32 | 61 | 59 | 15 | 11 | 73 | 62 | 47 | 69 |

* + 1. **Perancangan Sistem**

Siklus penyelesaian masalah optimasi distribusi barang dua tahap menggunakan algoritma genetika ditunjukkan dengan diagram alur Gambar 3.2.

|  |
| --- |
|  |

**Gambar 3.2** Flowchart

* 1. **Implementasi**

Implementasi *user interface* terdiri dari halaman input data pabrik, distributor, agen, biaya dan parameter yang digunakan pada proses algoritma genetika. Halaman input parameter ditunjukkan pada gambar 3.3

|  |
| --- |
|  |

**Gambar 3.3** Halamab Input Parameter

Halaman hasil proses algoritma genetika untuk distribusi dua tahap ditunjukkan pada Gambar 3.4.

|  |
| --- |
|  |

**Gambar 3.4** Halaman Hasil

* 1. **Pengujian dan Analisa**

Pengujian sistem yang dilakukan bersifat *sequential,* artinya hasil pengujian terbaikdigunakan untuk pengujian selanjutnya. Pengujian dilakukan pada parameter sebagai berikut :

1. Pengaruh Jumlah *Pop\_Size*

Analisis ini bertujuan untuk menentukan berapa jumlah *pop\_size* yang menghasilkan rata-rata nilai *fitness* terbaik.

1. Pengaruh Kombinasi *Cr* dan *Mr*

Analisis ini bertujuan untuk menentukan kombinasi *cr* dan *mr* tepat untuk menghasilkan rata-rata nilai *fitness* terbaik.

1. Pengaruh Jumlah Generasi

Analisis ini bertujuan untuk mendapatkan generasi terbaik yang menghasilkan rata-rata nilai *fitness* terbaik.

1. Pengaruh penggunaan segmen mutasi Analisis ini bertujuan untuk mengetahui pengaruh penggunaan segmen mutasi untuk menghasilkan rata-rata nilai *fitness* terbaik.
   * 1. **Hasil dan Analisa Pengujian**
        1. **Pengujian Ukuran Populasi**

Pengujian terhadap jumlah populasi digunakan untuk mengetahui pengaruh jumlah populasi terhadap nilai *fitness* yang dihasilkan. Pengujian terhadap pengaruh ukuran populasi dilakukan dengan ukuran populasi sebanyak 20, 40, 60, 80 dan 100 dengan kombinasi *cr* = 0,4 dan *mr* = 0,6 dan jumlah generasi sebanyak 50. Untuk mendapatkan nilai rata-rata *fitness* setiap percobaan akan dilakukan sebanyak 10 kali. Pada gambar 3.5 dapat dilihat grafik kenaikan dan penurunan nilai fitness.

|  |
| --- |
|  |

**Gambar 3.5** Grafik Hasil Uji COba Ukuran Populasi

Berdasarkan grafik hasil pengaruh pengujian ukuran populasi terhadap nilai fitness dari rentang 20 sampai 80 mengalami kenaikan, pada ukuran 100 populasi mengalami penurunan kembali. Pada pengujian ini diperoleh kesimpulan bahwa ukuran populasi yang terlalu kecil maupun terlalu besar tidak akan menghasilkan solusi yang optimal. Karena ukuran populasi yang terlalu kecil mengakibatkan area eksplorasi terlalu lebar sehingga peluang untuk mendapatkan solusi terbaik semakin kecil, sedangkan ukuran populasi yang terlalu besar mengakibatkan area eksplorasi menjadi semakin sempit sehingga memungkinkan terjadi konvergensi dan waktu komputasi juga akan lama. Eksplorasi yang sempit menyebabkan pencarian solusi terbaik tidak berjalan dengan baik dan child yang dihasilkan nilai fitness-nya hampir sama dengan induknya (Mahmudy, 2013).

* + - 1. **Pengujian Pengaruh cd dan mr**

Pengujian terhadap pengaruh kombinasi nilai *crossover rate* (cr) dan *mutation rate* (mr) digunakan untuk mengetahui pengaruh kombinasi dari kedua parameter terhadap hasil yang optimal untuk kasus distribusi barang dua tahap. Pada pengujian ini digunakan jumlah populasi terbaik dari pengujian sebelumnya sebanyak 80 populasi dan jumlah generasi sebanyak 50. Kombinasi parameter reproduksi yang digunakan berkisar 0,1 sampai 0,9 dan masing-masing percobaan dilakukan sebanyak 10 kali yang akan diambil nilai rata-rata *fitness.* Pada gambar 3.6 dapat dilihat grafik kenaikan dan penurunan nilai fitness.

|  |
| --- |
|  |

**Gambar 3.6** Grafik Hasil Uji Coba Kombinasi cr dan mr

Berdasarkan Gambar 3.6 diperoleh rata-rata nilai *fitness* terbesar adalah pada ukuran kombinasi *crossover rate* (*cr*) sebesar 0,6 dan ukuran *mutation rate* (*mr*) sebesar 0,4 dengan rata-rata nilai *fitness* 0,16566. Sedangkan rata-rata nilai *fitness* terkecil adalah pada ukuran kombinasi cr 0,1 dan mr 0,9 dengan rata-rata *fitness* 0,1596. Nilai *fitness* yang dihasilkan pada setiap kombinasi cenderung rendah pada nilai *cr* dan *mr* yang tidak seimbang, yaitu *cr* yang terlalu kecil maupun *mr* yang terlalu kecil. Nilai *cr* yang terlalutinggi dan *mr* yang rendah menyebabkan algoritma genetika tidak akan mampu memperlebar area pencarian, sedangkan nilai *cr* yang rendah dan *mr* yang tinggimenyebabkan algoritma genetika akan bekerja seperti random *search* dan tidak mampu mengekplorasi daerah pencarian secara efektif (Mahmudy, WF, Marlian, RM dan Luong, LHS dalam Pratiwi, 2014).

* + - 1. **Pengujian Pengarug Jumlah Generasi**

Pengujian terhadap maksimum generasi digunakan untuk mengetahui pengaruh banyaknya jumlah generasi yang digunakan terhadap rata-rata nilai *fitness.* Pada pengujian ini digunakan jumlah generasi sebanyak 20, 40, 60, 80,100 dengan jumlah ukuran populasi sebanyak 80 dan kombinasi *cr* sebesar 0,6 dan *mr* sebesar 0,4. Percobaan terhadap pengaruh jumlah generasi dilakukan sebanyak 10 kali untuk diambil nilai rata-rata *fitness,* hasil pengujian ditunjukkan padaGambar 3.7

|  |
| --- |
|  |

**Gambar 3.7** Grafik Hasil Prose Uji Coba Jumlah Generasi

Berdasarkan Gambar 3.7 mengenai pengujian jumlah generasi diperoleh solusi terbaik pada generasi ke 80. Hasil *fitness* tertinggi pada generasi tersebut adalah 0,165652 dan nilai *fitness* terendah dihasilkan ketika menggunakan jumlah generasi yang sedikit pada jumlah generasi 20. Pada grafik yang ditunjukkan oleh Gambar 6.3 generasi 20 sampai dengan 80 mengalami kenaikan rata-rata nilai *fitness* yang signifikan. Jumlah generasi sebanyak 20 memiliki rata-rata nilai *fitness* cenderung sedikit dikarenakaneksplorasi terhadap ruang pencarian cenderung lebih kecil. Sedangkan pada jumlah generasi 80 menghasilkan rata-rata nilai *fitness* yang tinggi karena eksplorasi terhadap ruang pencarian cenderung lebih besar namun dibarengi dengan waktu eksekusi yang lebih lama. Pada generasi ke 100 rata-rata nilai *fitness* cenderung mengalami penurunan karena ruang pencarian cenderung lebih besar dan memungkinkan kromosom mendapatkan *fitness* yang lebih kecil karena sifat genetika adalah random kromosom.

* + - 1. **Pengujian Pengaruh Segment Mutasi**

Pengujian terhadap segmen mutasi digunakan untuk mengetahui perbandingan rata-rata nilai *fitness* berdasarkan segmen mutasi yang digunakan. Pada pengujian segmen mutasi digunakan beberapa parameter terbaik dari pengujian sebelumnya yaitu ukuran populasi sebanyak 80, kombinasi *cr* 0,6 dan *mr* 0,4 serta jumlah generasi 80. Pengujian segmen mutasi diterapkan dengan menggunakan segmen mutasi pada distribusi tahap 1, segmen mutasi pada distribusi tahap 2, menggunakan dua segmen sekaligus dan menggunakan kombinasi antara ketiga jenis segmen. Percobaan pada pengujian segmen mutasi dilakukan sebanyak 10 kali untuk diambil rata-rata nilai *fitness,* hasil pengujian terhadap segmen mutasi ditunjukkan pada Gambar 3.8.

|  |
| --- |
|  |

**Gambar 3.8** Grafik Hasil Uji Coba Segmen Mutasi

Berdasarkan grafik pada Gambar 5.4 mengenai pengujian segmen mutasi terhadap rata-rata nilai *fitness,* menunjukkan bahwa segmen campuran menghasilkan hasil optimal dibandingkan dengan menggunakan segmen yang lain. Hasil rata-rata nilai *fitness* optimal ditunjukkan oleh segmen mutasi campuran dengan hasil 0,166514, sedangkan untuk hasil rata-rata nilai *fitness* minimal ditunjukkan oleh penggunaan segmen mutasi distribusi tahap 1. Hal ini membuktikan bahwa mutasi dengan segmen campuran lebih optimal karena lebih banyak solusi yang dicoba pada setiap segmennya, sedangkan pada mutasi dengan segmen 1 saja menghasilkan hasil yang lebih rendah dibandingkan dengan yang lain mengakibatkan area eksplorasi lebih kecil dikarenakan pada segmen pertama panjang kromosomnya lebih pendek. Sehingga kecenderungan suatu kromosom untuk lebih mirip dengan induknya lebih besar.

**Referensi :**

Sulistiyono Risko. 2016. Penerapan Algoritma Genetika untuk Permasalahan Optimasi Distribusi Barang Dua Tahap. Diakses melalui : <http://openpaper.its.ac.id/download.php/?idf=41> pada tangal 30 Oktober 2016.