

PENERAPAN *ALGORITMA GENETIKA* PADA *KNAPSACK PROBLEM*

Studi Kasus : Optimasi Pemilihan Buah Kemasan Kotak

Oleh :

- 1) Bambang Sunarwan
- 2) Riyadi Juhana

Abstrak

Algoritma genetika adalah algoritma pencarian yang didasarkan atas mekanisme dari seleksi alam yang lebih dikenal dengan proses evolusi. Dalam proses evolusi, individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya. Hanya individu-individu yang kuat yang mampu bertahan. Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembang-biakan. Dalam algoritma genetika, proses perkembang-biakan ini menjadi proses dasar yang menjadi perhatian utama, dengan dasar berpikir: “Bagaimana mendapatkan keturunan yang lebih baik”.

Permasalahan yang akan diselesaikan pada Knapsack Problem ini adalah optimasi pemilihan buah kemasan kotak pada proses distribusi barang untuk menekan biaya pengiriman dan memaksimalkan keuntungan. Adapun parameter-parameter yang digunakan adalah sebagai berikut: a. Hanya terdapat satu item paket buah kemasan kotak yang sejenis dan memiliki profit, tingkat kebutuhan pasar, berat, volume, dan masa awet buah yang akan dipilih b). Berat ataupun volume total dari item buah kemasan kotak yang dipilih tidak boleh melebihi kapasitas dari Knapsack. Parameter-parameter pada algoritma genetika dapat diubah-ubah sesuai keinginan user seperti: ukuran populasi, probabilitas mutasi, dan probabilitas *crossover* d. Sistem akan menampilkan solusi optimal dari hasil pencarian yang berupa terpilih atau tidaknya suatu item untuk bisa dimasukkan ke dalam Knapsack, nilai keuntungan total, berat total, dan volume total dari item yang terpilih

Dari pengembangan dan ujicoba yang telah dilakukan terkait dengan implementasi algoritma genetika pada knapsack problem, dapat disimpulkan beberapa hal, yaitu: 1). Algoritma genetika cukup baik digunakan pada knapsack problem. 2). Peningkatan jumlah populasi akan memberikan nilai *fitness* yang semakin tinggi, yang berarti bahwa solusi optimal yang ditemukan semakin baik. Karena dengan peningkatan jumlah populasi akan memberikan ruang solusi yang semakin banyak.

Kata Kunci: *Algoritma Genetika (AG), Knapsack*

1. PENDAHULUAN

Knapsack Problem merupakan suatu persoalan menarik untuk diteliti dan diimplementasikan pada situasi nyata. Persoalan ini banyak diterapkan pada situasi nyata utamanya pada bidang jasa, sebagai contoh pada permasalahan pengangkutan barang pada peti kemas atau dalam skala kecil pada pengangkutan barang dalam kemasan.

Inti dari pembahasan adalah bagaimana menentukan kombinasi barang yang akan diangkut untuk memperoleh keuntungan maksimal, dengan pertimbangan tidak melebihi kapasitas alat angkut yang digunakan.

2. KNAPSACK PROBLEM

Knapsack Problem merupakan masalah optimasi kombinatorial. Sebagai contoh adalah suatu kumpulan barang masing-masing memiliki berat dan nilai, kemudian akan ditentukan jumlah tiap barang untuk dimasukkan dalam koleksi sehingga total berat kurang dari batas yang diberikan dan nilai total seluas mungkin. Merupakan salah satu dari persoalan klasik yang banyak ditemukan dalam literatur-literatur lama dan hingga kini masih sering ditemukan dalam kehidupan sehari-hari.

Contoh nyata dari *Knapsack Problem* ini misalnya, jika ada seorang pedagang barang kebutuhan rumah tangga berkeliling menggunakan gerobak. Tentu saja gerobaknya memiliki kapasitas maksimum, sehingga ia tidak bisa memasukkan semua barang dagangannya dengan seenak hatinya. Pedagang tersebut harus memilih barang-barang mana saja yang harus ia angkut, dengan pertimbangan berat dari barang yang dibawanya agar tidak melebihi kapasitas maksimum gerobak dan memaksimalkan profit dari barang-barang yang dibawa (Adit, 2009).

Sebuah *Knapsack* memiliki kapasitas total V , dimana terdapat n buah item berbeda yang dapat ditempatkan dalam *knapsack*. Item i memiliki bobot v_i dan profitnya b_i . Jika X_i adalah jumlah item i yang akan dimasukkan dalam *Knapsack* yaitu bernilai 1 dan 0, maka secara umum tujuan yang harus tercapai adalah:

$$\text{maksimalkan } \sum_{i=1}^n B_i \cdot X_i \quad (1)$$

$$\text{dengan batasan } \sum_{i=1}^n v_i \cdot X_i \leq V \quad (2)$$

3. ALGORITMA GENETIKA

Dalam Pemahaman Operation research genetika lebih dikenal dengan proses evolusi dan diketahui bahwa, proses evolusi individu secara terus menerus akan mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya/kebiasaan yang terjadi. Hanya individu-individu yang kuat yang mampu bertahan. Sementara proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembang-biakan/penyesuaian terus menerus secara alami.

Sementara penerapan algoritma genetika, dalam maksud proses perkembang-biakan/perulangan tata peletakan ini merupakan proses dasar yang perlu menjadi perhatian utama, sebagai landasan berpikir: dalam perolehan sistem/optimasi berpikir untuk lebih baik, lebih sesuai, lebih menarik, lebih bernilai, atau dikenal dengan : “Bagaimana mendapatkan keturunan yang lebih baik” (Basuki, 2003).

Secara umum, dalam algoritma genetika terdapat 5 (lima) proses pencapaian /tahapn yaitu: pembentukan populasi awal, perhitungan nilai *fitness*, seleksi, regenerasi (*crossover* dan mutasi), penciptaan populasi baru hasil regenerasi.

4. METODOLOGI

Permasalahan yang akan diselesaikan pada *Knapsack Problem* ini adalah optimasi pemilihan buah kemasan kotak pada proses distribusi barang untuk menekan biaya pengiriman dan memaksimalkan keuntungan. Adapun 4(empat) parameter yang digunakan adalah sebagai berikut: **a**). Hanya terdapat satu item paket buah kemasan kotak yang sejenis dan memiliki profit, tingkat kebutuhan pasar, berat, volume, dan masa awet untuk buah yang akan dipilih **b**). Berat ataupun volume total dari

item buah kemasan kotak yang dipilih tidak boleh melebihi kapasitas dari *Knapsack.c*) Parameter-parameter pada algoritma genetika dapat diubah-ubah sesuai keinginan *user* seperti: ukuran populasi, probabilitas mutasi, dan probabilitas *crossover d*). Sistem akan menampilkan solusi optimal dari hasil pencarian yang berupa terpilih atau tidaknya suatu item untuk bisa dimasukkan ke dalam *Knapsack*, nilai keuntungan total, berat total, dan volume total dari item yang terpilih

4.1 Kodifikasi Kromosom dan Pembentukan Populasi Awal

Kodifikasi kromosom yang digunakan di sini mengadopsi aturan kodifikasi kromosom algoritma genetika, yaitu pengkodean dengan bilangan biner. Bilangan biner ini diperoleh dari yang terpilih atau tidak terpilihnya item secara random. “ Jika item terpilih maka nilainya 1 sedangkan jika tidak terpilih nilainya 0”. Dalam pemrograman hal ini dapat dituliskan dalam bentuk struktur data sebagai berikut.

- 1) Type Populasi
- 2) Kr As String
- 3) Fit As Double
- 4) End Type
- 5) Dim p() As Populasi

Panjang satu kromosom adalah sama dengan jumlah seluruh item yang ada. Sebagai contoh untuk pembentukan kromosom ini, misalkan ada item-item seperti pada :

Tabel 1. Daftar Item

Item	Profit	Weight
Item 1	Rp. 2000	80 kg
Item 2	Rp. 1000	60 kg
Item 3	Rp. 1500	25 kg
Item 4	Rp. 1800	30 kg

Dari Tabel 1 diketahui bahwa ada 4 (empat) item yang akan di pilih yang memenuhi kapasitas maksimum dari *Knapsack*. Berdasarkan Tabel 1 di atas, maka kromosom-kromosom dapat di representasikan sebagai berikut:

- Kromosom[1] = 0111
- Kromosom[2] = 0100
- Kromosom[3] = 0110
- Kromosom[4] = 0101

Urutan nilai 1 dan 0 setiap gen pada setiap kromosom di lakukan secara acak.

4.2 Evaluasi Solusi

Sebelum dilakukan evaluasi fungsi tujuan tiap kromosom, terlebih dahulu perlu dilakukan pengecekan berat dan volume total dari kromosom agar tidak melebihi kapasitas *knapsack*. Untuk kromosom yang memiliki berat dan volume total melebihi kapasitas *knapsack*, maka akan dilakukan perbaikan gen agar tidak melanggar *constraint*. Pada algoritma genetika yang digunakan dalam aplikasi ini, fungsi objektif dapat langsung digunakan nilai *fitness*-nya karena permasalahan yang dihadapi adalah masalah maksimasi fungsi tujuan. Fungsi objektif tersebut adalah:

$$obj = \sum_{i=1}^n \frac{p_i \cdot b_i \cdot q_i}{(1 + bAw_t \times awt_i \cdot q_i)}$$

Dengan *constraint*:

$$0 \leq \sum_{i=1}^n wg_i \cdot q_i \leq WgKnap \tag{3}$$

dan

$$0 \leq \sum_{i=1}^n vol_i \cdot q_i \leq volKnap \tag{4}$$

dimana,

- i* = item buah kemasan kotak ke-i yang tersedia
- n* = jumlah item buah kemasan kotak yang tersedia
- p* = profit dari buah kemasan kotak
- b* = tingkat kebutuhan pasar *q* = terpilih atau tidaknya item ke-i (1 atau 0)
- bAw_t* = bobot pengaruh masa keawetan buah terhadap

awt = nilai optimal fungsi objektif
 = masa awet item buah kemasan kotak
 wg = berat item
 vol = volume item
 $WgKnap$ = berat muatan maksimum yang dapat ditampung oleh *knapsack*
 $volKnap$ = volume muatan maksimum yang dapat ditampung oleh *knapsack* penambahan dengan 1 pada fungsi objektif bertujuan untuk menghindari pembagian dengan 0.

4.3 Pembentukan Generasi Baru

Pembentukan generasi baru dilakukan dengan tiga operasi dasar pada algoritma genetika yaitu dengan seleksi, *crossover* (persilangan), dan mutasi. Pembentukan generasi baru ini bertujuan untuk memperoleh kromosom yang memiliki *fitness* yang paling baik yang merepresentasikan solusi optimal yang diperoleh pada setiap proses

5. HASIL IMPLEMENTASI

Proses input permasalahan dapat dilakukan dengan dua cara, yaitu: (1) dengan memasukkan permasalahan secara manual melalui *form input* permasalahan, dan (2) dengan membuka *file* yang sudah tersimpan. Setelah proses input dilakukan, maka langkah berikutnya adalah mengisikan parameter-parameter yang dibutuhkan dalam *knapsack* seperti: berat maksimum item dan volume maksimum item serta bobot pengaruh keawetan buah pada pencarian solusi optimal. Selain parameter-parameter *knapsack*, yang juga perlu diisikan adalah parameter-parameter untuk algoritma genetiknya.

Ujicoba

Hasil implementasi program yang telah dikembangkan di ujicobakan pada data yang

telah masukkan sebelumnya. Ada 2 (dua) skenario ujicoba yang dilakukan, yaitu: (1) skenario uji coba output algoritma genetika dari suatu kasus, dan (2) skenario jumlah populasi terhadap nilai *fitness* yang dihasilkan.

Tabel 1. Data Ujicoba

No	Nama	Br	Vol	Tkb	Profit	MA
1	Item 1	25	500000	S	5000	2
2	Item 2	35	500000	S	5000	2
3	Item 3	40	500000	S	5000	2
4	Item 4	20	500000	S	5000	2
5	Item 5	15	500000	S	5000	2
6	Item 6	10	300000	S	5000	2
7	Item 7	10	600000	S	5000	2
8	Item 8	10	350000	S	5000	2
9	Item 9	10	550000	S	5000	2
10	Item 10	10	200000	S	5000	2
11	Item 11	10	500000	S	5000	2
12	Item 12	10	500000	R	5000	2
13	Item 13	10	500000	T	5000	2
14	Item 14	10	500000	T	5000	2
15	Item 15	10	500000	S	5000	2
16	Item 16	10	500000	S	5000	2
17	Item 17	10	500000	S	6000	2
18	Item 18	10	500000	S	5000	2
19	Item 19	10	500000	S	7000	2
20	Item 20	10	500000	S	8000	2

Keterangan:

Brt = berat item
 Tkb = tingkat kebutuhan pasar
 S = sedang
 T = tinggi
 R = rendah
 MA = masa awet kemasan buah

Uji Coba 1:

Berikut dijelaskan hasil ujicoba output algoritma genetika dengan parameter: jumlah populasi 20, probabilitas *crossover* 50%, dan probabilitas mutasi 10%. Hasil lengkap dari proses Ujicoba berdasarkan skenario yang diberikan tampak seperti pada

Tabel 2. Hasil Output Algoritma Genetika

No	Kasus	Item digunakan	Output	Uraian
1	Ujicoba output dengan kriteria berat item item bervariasi, sedangkan data lainnya tetap. Kapasitas knapsack yaitu: berat=60 dan volume=2500 000	Item nomor 1-5	1. Item terpilih yaitu item nomor 1,4 dan 5 2. Profit total= 15000 3. Berat total=60 4. Volume total=1.500.000	Item yang diutamakan adalah item yang memiliki berat relative lebih rendah dari yang lainnya
2	Ujicoba output dengan kriteria volume itemitem bervariasi, sedangkan data lainnya tetap. Kapasitas knapsack yaitu: berat=60 dan volume=1000.000	Item nomor 6-10	1. Item terpilih yaitu item nomor 6,8 dan 10 2. Profit total = 15000 3. Berat total = 30 4. Volume total = 8.500.000	Item yang diutamakan adalah item yang memiliki volume relative lebih rendah dari yang lainnya
3	Ujicoba output dengan kriteria Tingkat kebutuhan Pasar bervariasi, Sedangkan data lainnya tetap. Kapasitas knapsack yaitu: berat=60 dan volume=1500.000	Item nomor 11-15	1. Item terpilih yaitu item nomor 11,13 dan 14 2. Profit total= 15000 3. Berat total=30 4. Volume total=1.500.000	Item yang diutamakan adalah item yang memiliki tingkat kebutuhan pasar relative lebih tinggi dari yang lainnya
4	Ujicoba output dengankriteria profit itemitem bervariasi, sedangkan data lainnya tetap. Kapasitas knapsack yaitu: berat=60 dan volume =1500.000	Item nomor 16-20	1. Item terpilih yaitu item nomor 17,19 dan 20 2. Profit total= 21000 3. Berat total=30 4. Volume total=1.500.000	Item yang diutamakan adalah Item yang memiliki Profit relative Lebih tinggi dari yang lainnya

Ujicoba 2:

Berikut ini akan dijelaskan hasil ujicoba output algoritma genetika dengan parameter jumlah populasi yang bervariasi terhadap nilai *fitness* yang dihasilkan. Pada skenario ini digunakan probabilitas *crossover* 50%, dan probabilitas mutasi 1%. Hasil lengkap dari proses ujicoba berdasarkan scenario yang diberikan tampak seperti pada tabel 4.

Tabel 3. Hasil Output Jumlah Populasi Terhadap Nilai *Fitness*

No	Jumlah Populasi	Nilai <i>Fitness</i>
1	5	25333,33
2	10	27000,00
3	15	27333,33
4	20	27333,33

Dari Tabel 3 dapat dilihat bahwa, semakin besar jumlah populasi, maka akan menghasilkan nilai *fitness* yang semakin besar pula.

6. PENUTUP

Dari pengembangan dan ujicoba yang telah dilakukan terkait dengan implementasi algoritma genetika pada *knapsack problem*, dapat disimpulkan beberapa hal, sebagai berikut :

- 1) Algoritma genetika cukup baik digunakan pada knapsack problem
- 2) Peningkatan jumlah populasi akan memberikan nilai *fitness* yang semakin tinggi, yang berarti bahwa solusi optimal yang ditemukan semakin baik. Karena dengan peningkatan jumlah populasi akan memberikan ruang solusi yang semakin banyak.

PUSTAKA

- 1) Hartono, Johanes., 1994, *Pengembangan Algoritma Simulated Annealing pada Tata Letak Fasilitas*. Tugas Akhir Jurusan Teknik Industri Institut Teknologi Bandung.
- 2) Iskandar, Bermawi P. dan Mulyono, Arie., 1997, *Perangkat Lunak "SA-*

- CRAFT PLUS*” untuk Perancangan Tata Letak Pabrik. Jurnal Teknik dan Manajemen Industri, vol. 17, no. 2, hlm. 33 - 43.
- 3) Mulyono, Arie., 1997, *Pengembangan Perangkat Lunak SA-CRAFT (Simulated Annealing Computerized Relative Allocation Facilities Technique) Untuk Perancangan Tata Letak Fasilitas*. Tugas Akhir Jurusan Teknik Industri Institut Teknologi Bandung.
 - 4) Rosenblatt, Meir J., 1986, *The Dynamics of Plant Layout.*, Management Science, 32, 1,hlm. 76-86.
 - 5) Tompkins, James A., et al., 1996, *Facilities Planning*. 2nd edition. New York : John Wiley & Sons, Inc.
 - 6) Urban, T.L., 1993, *A Heuristic for The Dynamics Facility Layout Problem.*, IIE Transaction, 25, hlm. 57-63.

PENULIS

- 1) **Ir. Bambang Sunarwan, MT.**, Staf Pengajar Program Studi Teknik Geologi, Fakultas Teknik, Universitas Pakuan Bogor.
e-mail : narwanbs@yahoo.com
- 2) **Ir. Riyadi Juhana**, Staf pengajar Tetap Program Studi Teknik Industri Fakultas Teknik, Universitas Suryakencana Cianjur.
e-mail : rjoehana@gmail.com