

Evaluasi Kinerja Genetic Algorithm (GA) dengan Strategi Perbaikan Kromosom Studi Kasus: Knapsack Problem

¹Aristoteles, ²Wardiyanto dan ³Aryanti Dwiastuti

¹Jurusan Ilmu Komputer FMIPA Unila

²Jurusan Budidaya Perairan FP Unila

³Jurusan Ilmu Komputer FMIPA Unila

Abstract

Knapsack Problem (KP) is known as one of combinatorial optimization problem that maximizes the amount of value (profit/ priority) of some of the items chosen to put in a sack or a bag without exceeding its capacity. KP has been applied in many fields: cargo loading, budget control, financial management and etc. In addition, KP is also included in NP-Hard problem. Several research have been conducted using some heuristic method to solve it. Some of these methods are Branch and Bound, Greedy Algorithms, Dynamic Programming, and Genetic Algorithms. However, when using GA, the feasibility of chromosome is one of critical issue. Most of optimization problem has constraint functions that will change the feasibility of solution. GA has some strategies to handle this case on the evaluation step. Evaluation not only calculates chromosome's fitness value, but also handles the infeasible chromosomes presence. There're rejecting strategies, repairing strategies, and penalty strategy. This paper develops a strategy GA approach with chromosome repairing strategy. In order to see the performance of the algorithm, we have done numerical experiment with several test problems taken on certain literature. In addition, this study will also give variations in GA parameter to determine the efficiency and effectiveness of the algorithm.

Keywords: *Combinatorial Optimization, Genetic Algorithm (GA), Knapsack Problem, Repairing Strategy*

1 Pendahuluan

Persoalan optimasi merupakan permasalahan matematis yang berkaitan dengan minimalisasi atau maksimalisasi suatu fungsi dengan satu atau beberapa variabel. Umumnya, persoalan optimasi bergantung pada fungsi batasan/ kendala baik berupa persamaan atau pertidaksamaan. Persoalan optimasi sering diaplikasi pada beberapa bidang, seperti Riset Operasi, Ilmu Manajemen, dan desain teknik. Ada beberapa macam persoalan optimasi. Salah satunya adalah persoalan optimasi kombinatorik [1].

Knapsack (KP) merupakan salah persoalan optimasi kombinatorik yang dipelajari secara intensif baik secara teoritis maupun praktis. KP dapat dianggap sebagai persoalan pengelompokan beberapa barang, yaitu barang yang diambil dan yang dibuang. Tujuan utama dari persoalan KP adalah untuk memaksimalkan nilai dari barang-barang yang dimasukkan ke dalam *knapsack/* tas. Diantara beberapa sub-problem dari KP, 0-1 KP merupakan subproblem KP yang cukup intensif dipelajari. 0-1 KP memiliki tiga alasan untuk dipelajari (a) KP dapat dilihat sebagai persoalan integer linear sederhana, (b) KP dapat muncul dengan beberapa sub problem yang lebih kompleks, dan (c) KP dapat merepresentasikan banyak situasi praktikal [2].

0-1 KP juga termasuk ke dalam kelas NP-Hard dimana algoritma penyelesaiannya memiliki kompleksitas nonpolinomial dengan skala ruang solusi yang sangat besar dan peningkatan koefisien secara eksponensial. Secara prinsip, KP dapat diselesaikan dengan perhitungan sederhana. Namun, pada ukuran persoalannya sebenarnya, jumlah solusi layak akan menjadi sangat banyak. Berbagai pendekatan pada metode heuristik, seperti *Dynamic Programming*, *backtracking*, *Branch and Bound* tidak dapat digunakan untuk menyelesaikan persoalan ini. Diantara beberapa metode heuristik

tersebut, *Genetic Algorithm* (GA) merupakan algoritma yang cukup efektif dalam menyelesaikan KP[3].

GA pertama kali diperkenalkan oleh John Holland pada tahun 1975, kemudian GA dipopulerkan oleh beberapa peneliti selanjutnya, seperti Goldberg (1989), Davis (1985 dan 1991), Gen dan Cheng (1997 dan 2000), dan Michalewicz (1994). GA yang memanfaatkan operasi yang terjadi dalam evolusi (seleksi alam dan reproduksi), bekerja dalam populasi kandidat solusi sejumlah n kromosom. Masing-masing kromosom terdiri dari sejumlah gen dengan suatu symbol yang merepresentasikan solusi yang akan diselesaikan. Populasi tersebut akan mengalami seleksi, yaitu proses penentuan kromosom mana yang akan bertahan untuk melakukan reproduksi dan proses modifikasi oleh mutasi. GA dipertimbangkan karena metodenya yang sederhana, operasi yang mudah, syarat yang minimum, dan memiliki perspektif paralel dan global [4].

Namun, ketika menggunakan GA, fungsi kendala/ batasan menjadi suatu hal yang penting. Fungsi tersebut akan membagi ruang solusi. GA memiliki beberapa strategi pada tahap evaluasi untuk menyelesaikan permasalahan ini, yaitu salah satunya dengan strategi perbaikan kromosom. Oleh karena itu, paper ini akan berfokus pada bagaimana kinerja GA dengan strategi perbaikan kromosom. Performa GA akan dievaluasi dengan menggunakan *benchmark test problem* yang telah diketahui nilai optimumnya [5]. Selain itu, untuk melihat, efisiensi algoritma, penelitian ini juga akan mevariasikan nilai probabilitas mutasi dan *crossover*.

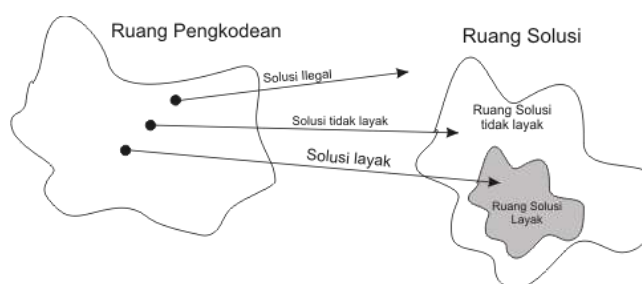


Figure 1 Kromosom Representasi Binari

2 Metodologi

2.1 Knapsack Problem (KP)

0-1 KP merupakan persoalan kombinasi dan optimasi. Persoalan 0-1 KP memaksimalkan total nilai sekumpulan barang yang dimasukkan ke dalam tas yang memiliki kapasitas C . Masing-masing barang tersebut memiliki berat dan nilai. Persoalan ini memiliki berbagai macam aplikasi seperti *budgeting*, *cargo loading*, *cutting stock*, *bin packing*, *network planning*, *network routing*, *parallel scheduling*, dan perencanaan ekonomi.

Secara matematika, 0-1 KP dapat diformulasikan sebagai berikut:

$$\mathbf{max} : z = \sum_{i=1}^n p_i x_i \quad (1)$$

$$\mathbf{s.t.} : \sum_{i=1}^n w_i x_i \leq c \quad (2)$$

$$x_i \in \{0,1\}, i = 1,2,\dots,n \quad (3)$$

dengan p_i adalah nilai item i , w_i adalah berat item i , dan c adalah kapasitas maksimum KP. x_i hanya akan bernilai 0 atau 1. Nilai 1 jika item i terpilih untuk dimasukkan ke dalam tas/ sack dan sebaliknya pada nilai 0 [6-7].

2.2 Algoritma Genetika

GA telah diperkenalkan oleh John Holland (1975), pada bukunya yang berjudul “Adaptation in Natural and Artificial Systems”, sebagai suatu metode pencarian heuristik yang dikembangkan berdasarkan gagasan evolusi seleksi alamiah pada Teori Darwin. GA melakukan apa yang alam lakukan. GA melakukan seleksi untuk berkembang biak (reproduksi: crossover) dan melakukan modifikasi sifat (gen) pada kromosom. GA bekerja dengan memodifikasi masalah ke dalam suatu symbol. GA memiliki lima komponen penting, yaitu (1) representasi gen untuk solusi masalah, (2) cara pembentukan populasi awal, (3) penentuan fungsi evaluasi, (4) operasi genetika, dan (5) nilai parameter GA (probability mutasi dan crossover) [8].

Struktur umum GA adalah sebagai berikut :

```

begin
t ← 0;
inisiasi P(t);
evalusasi P(t);
while (bukankondisi berhenti) do
begin
rekombinasi P(t) menjadi C(t);
evaluasi C(t);
seleksi P(t);
t = t + 1;
end
end

```

2.3 Representasi Kromosom

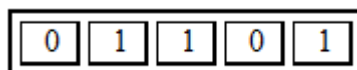


Figure 2 Kromosom Representasi Binari

Representasi Kromosom merupakan hal pertama yang harus dipertimbangkan. Representasi kromosom harus mampu merepresentasikan kandidat solusi serta parameternya. Penentuan representasi yang akan digunakan akan berpengaruh kepada tiap-tiap tahap pada GA. Setiap individu memiliki dua properti, lokasi dan kualitas. Lokasi merupakan nilai representasi suatu individu dan kualitas, merupakan nilai objektifnya.

Kromosom dapat direpresentasikan menjadi beberapa bentuk, seperti binari, matrik, permutasi, dan lain-lain. Pada persoalan 0-1 KP cocok pada penggunaan representasi biner, nilai nol (0) berarti barang tidak terpilih untuk dimasukkan ke dalam *sack* dan sebaliknya nilai satu (1) berarti barang terpilih [2]. Umumnya pembangkitan kromosom dilakukan secara random. Namun, kromosom yang terbentuk belum tentu berupa kromosom layak. 0-1 KP dengan selisih antara berat item dan kapasitas KP yang kecil, akan membuat kromosom yang dibangkitkan sulit mencapai kondisi layak. Oleh karena itu, selain pembangkitan kromosom secara random, paper ini juga menggunakan pembangkitan kromosom secara terarah (*directed*). Pembangkitan kromosom secara *directed* akan diarahkan pada ruang solusi layak. Pembangkitan kromosom *directed* ini hanya digunakan pada evaluasi perbandingan antara GA strategi perbaikan kromosom dan strategi penalti yang dilakukan oleh Riska Malinda. Berikut merupakan prosedur pembangkitan kromosom secara *directed* :

```

Init_pop = zeros(pop_size,genes);
for j = 1 : pop_size
    cap_now = capacity;
    for k = 1 : genes
        index = fix((rand*genes)+1);
        if init_pop[j,index] == 0 &&
            weight[index] <= cap_now then
            init_pop[j,index] = 1;
            cap_now = cap_now - weight[index];
            if cap_now <= 0 then
                break;
            end
        end
    end
end
end
end

```

2.4 Operator Genetika

2.4.1 Seleksi

Proses seleksi akan menjami individu yang siap akan memiliki kesempatan yang besar untuk berkembang biak dan dilakukan operator variasi (*crossover* dan mutasi). Operasi variasi gen akan menukar-nukan nilai gen untuk membangkitkan *offspring*. Ada beberapa metode seleksi, paper ini menggunakan kombinasi metode *roulette wheel* dan *elitism*. *Roulette wheel* akan memberikan peluang terpilih untuk masing-masing kromosom sesuai dengan nilai objektifnya. Kromosom yang memiliki peluang terbesar akan terpilih ke generasi aau tahap selanjutnya. Proses *Elitism* merupakan suatu prosedur untuk menduplikat kromosom yang memiliki nilai *fitness* terbaik. Tujuan *elitism* adalah agar kromosom dengan nilai terbaik akan tetap terpilih dan tidak mengalami kerusakan [4].

2.4.2 Operator Variasi

a. Mutasi

Operasi mutasi merupakan proses modifikasi suatu kromosom terpilih pada suatu gen terpilih. Operasi ini bertujuan untuk meningkatkan keragaman populasi kromosom sehingga pencarian solusi pada ruang solusi global. Metode mutasi yang akan digunakan adalah *flip mutation*, yaitu penukaran atau pembalikan nilai gen yang bernilai satu (1) menjadi nol (0) dan sebaliknya[4].

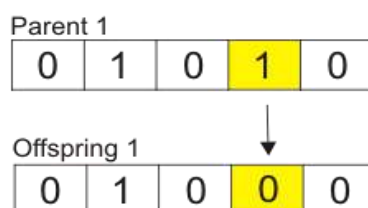


Figure 3 Operasi Mutasi

b. Crossover

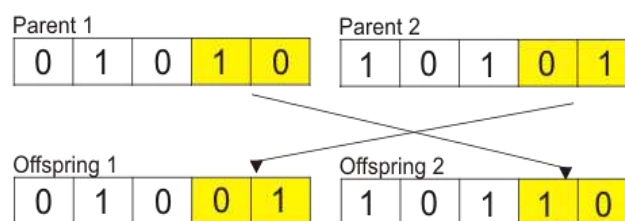


Figure 4 Operasi One-Point Crossover

Proses crossover merupakan suatu proses persilangan dua kromosom sebagai parent terpilih yang akan menghasilkan kromosom turunan. Kromosom turunan yang dihasilkan akan menambah keragaman dari kandidat solusi. Metode *crossover* yang akan digunakan pada penelitian ini adalah metode *one-point crossover*. Metode tersebut dilakukan dengan cara menukar masing masing gen sebelah kanan titik *crossover*. Untuk menemukan titik *crossover* tersebut, pembangkitan bilangan bulat random dilakukan [4].

2.5 Evaluasi kromosom

Evaluasi kromosom merupakan salah satu tahap GA yang menguji kelayakan kromosom terhadap fungsi kendala yang ada pada suatu persoalan. Metode yang umum digunakan adalah metode penskalaan, menghitung nilai fungsi tujuan sebagai nilai fitnessnya. Namun, ketika GA menyelesaikan persoalan optimasi pembatas, seperti KP, suatu variabel juga harus memenuhi beberapa fungsi kendala. Dengan adanya fungsi kendala tersebut, kromosom akan terbagi menjadi dua golongan: solusi layak dan tidak layak. Untuk menangani adanya kromosom yang tidak layak ini, GA memiliki beberapa strategi, diantaranya strategi perbaikan, strategi penalti, dan strategi penolakan [4].

Paper ini menggunakan strategi perbaikan kromosom. Strategi ini membuat kromosom tidak layak menjadi kromosom layak dengan suatu prosedur tertentu. Proses perbaikan kromosom pada persoalan KP adalah mengambil item terpilih dengan rasio $\left(\frac{p_i}{w_i}\right)$ yang paling kecil pada suatu kromosom. Evaluasi perbaikan kromosom akan dilakukan dengan prosedur sebagai berikut:

```

begin
  for i = 1: Pop_size
    %mengecek berat kromosom apakah melebihi kapasitas atau tidak
    if total weight of chromosome i > capacity
      then
        //kromosom belum layak
        Bandingkan rasio masing-masing berat item terpilih dalam kromosom; ubah nilai
        gen menjadi 0;
        hitung ulang total berat masing masing kromosom;
      end
    end
  end
end

```

3 Pembahasan

Penelitian ini dilakukan dengan membangun GA dengan menggunakan strategi perbaikan kromosom. Data yang digunakan didapatkan dari dari *Kpacking* dan terdapat sembilan macam data dengan jumlah item yang berbeda seperti yang ditunjukkan pada Tabel 1 [5]. Penelitian ini dilakukan dengan

menggunakan bahasa pemrograman Matlab R2009a dan di running pada komputer prosesor Inter-Core i3 Semua hasil penelitian didapatkan setelah dilakukan running sebanyak 10 kali.

Table 1 Test Problem

No.	Test Problem	n	Optimum
1	p01	10	309
2	p02	5	51
3	p03	6	150
4	p04	7	107
5	p05	8	900
6	small	40	1149
7	dataset 20_1000	20	4129
8	medium	100	1173
9	p08	24	13549094

Penelitian ini dilakukan evaluasi kinerja GA dengan perbaikan kromosom dengan pembangkitan kromosom secara acak. Selain itu, penelitian ini juga melihat perbandingan efektivitas dan efisiensi dari GA strategi perbaikan kromosom dan strategi penalti dengan pembangkitan kromosom secara terarah dan acak. Table 2 menunjukkan hasil objektif dari GA dengan parameter genetika yang digunakan adalah $pC = 0,2$ and $pM = 0,4$ dengan 500 ukuran populasi dalam 100 generasi.

Table 2 Hasil Komputasi fungsi objektif dan Presentasi Error Hasil Komputasi

No.	Test Problem	Pop size	GA			Error (%)
			Best	Worst	Average	
1	p01	20	309	-	309	0
2	p02	20	51	-	51	0
3	p03	20	150	-	150	0
4	p04	20	107	-	107	0
5	p05	20	900	-	900	0
6	small	20	1149	-	1149	0
7	dataset 20_1000	20	4129	-	4129	0
8	medium	1000	1173	-	1173	0
9	p08	250	13549094	13517399	13537960,6	0

Tabel 3 menunjukkan presentasi error apabila terjadi variasi pada parameter GA. Hasil menunjukkan bahwa GA dapat menghasilkan error sebesar 0,01 pada pM 0,6 dan pM 0,8.

TABLE 3 Presentasi Error Hasil Komputasi dengan variasi parameter GA

No	Test Problem	n	Optimum (Profit)	Pop size	HasilKomputasi (generasi = 100) (%)							
					$pC = 0,2$				$pC = 0,4$			
					0,2	0,4	0,6	0,8	0,2	0,4	0,6	0,8
1	p01	10	309	20	0	0	0	0	0	0	0	0
2	p02	5	51	20	0	0	0	0	0	0	0	0
3	p03	6	150	20	0	0	0	0	0	0	0	0
4	p04	7	107	20	0	0	0	0	0	0	0	0
5	p05	8	900	20	0	0	0	0	0	0	0	0
6	Small	40	1149	20	0	0	0	0	0	0	0	0
7	dataset 20_1000	20	4129	20	0	0	0	0	0	0	0	0
8	Medium	100	1173	1000	0	0	0,01	0,01	0	0	0,01	0,01
9	p08	24	13549094	250	0	0	0	0	0	0	0	0

Pada pembangkitan kromosom biasanya digunakan metode greedy. Namun, pada pembangkitan greedy, kromosom yang dihasilkan tidak tentu akan merepresentasikan kandidat solusi yang layak. Persoalan optimasi dengan fungsi pembatas akan membuat kromosom merepresentasi solusi yang

tidak layak juga. Maka, kromosom yang dibangkitkan akan merepresentasikan solusi yang layak semua, atau setengah layak setengah tidak layak, bahkan semua tidak layak. Oleh karena itu, pada penggunaan GA, Pada penelitian selanjutnya, proses pembangkitan kromosom dilakukan secara random dan directed. Evaluasi ini merupakan hasil perbandingan dengan GA strategi penalti, penelitian yang dilakukan oleh Riska. Figure 5 menunjukkan grafik perbedaan hasil komputasi GA strategi perbaikan kromosom dan penalti.

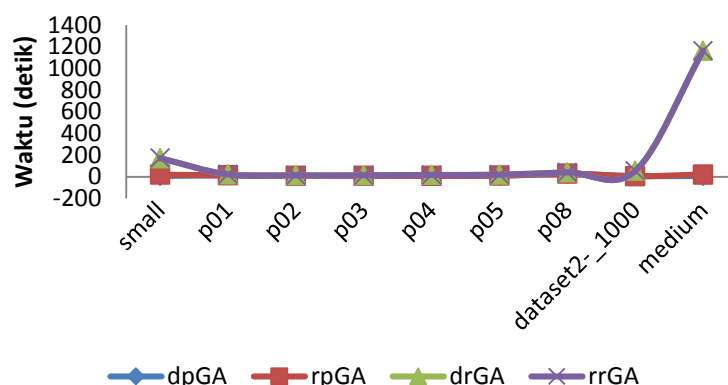


Figure 5 Perbandingan waktu komputasi antara Rsgadan Dpga

4 Kesimpulan

Metode GA dapat mencapai solusi optimal atau mendekati optimal dengan nilai *error* 0 % pada ukuran populasi tertentu pada masing-masing *test problem* dan nilai parameter $p_c = 0.2$ dan $p_m = 0.8$ dengan iterasi 100 generasi. Ukuran populasi bergantung pada banyaknya n data. Semakin banyak n data maka dibutuhkan ukuran populasi yang besar pula untuk mendapatkan hasil yang mendekati atau optimum. Nilai probabilitas mutasi dan *crossover* dapat memengaruhi hasil perhitungan, yaitu semakin besar nilai probabilitas mutasi dan *crossover* semakin tidak optimum pada ukuran populasi tertentu. Metode GA dengan strategi perbaikan kromosom memerlukan waktu yang lebih lama dalam menemukan hasil optimum dari pada waktu yang dibutuhkan strategi penalti. Namun hasil komputasi paling optimum dapat dihasilkan oleh GA strategi perbaikan kromosom.

5 Referensi

- [1] Gen, M. and R. Cheng Genetic Algorithms and Engineering Optimization, John Wiley & Sons, New York, 2000.
- [2] Martello, S. and Toth P., Knapsack Problems, Algorithms and Computer Implementations, John Wiley and Sons, 1990.
- [3] Garg, M. L. and Gupta, S., An Improved Genetic Algorithm Based on Adaptive Repair Operator for Solving the Knapsack Problem. Journal of Computer Science, Vol. 5, No. 8, 2009, pp. 544-547
- [4] Syarif, A., Algoritma Genetika: Teori dan Aplikasi, 2nd Ed., PT Graha Ilmu, 2014.
- [5] <http://kpacking.googlecode.com/svn/trunk/> didownload pada tanggal 23 Januari 2015
- [6] Yu, X. dan Gen, M., Introduction to Evolutionary Algorithms. London: Springer.2010.
- [7] Gupta, M., A Fast and Efficient Genetic Algorithm to Solve 0-1 Knapsack Problem, International Journal of Digital Application and Contemporary Research, 2013, Vol. 1, No. 6