ALGORITMA *NOVEL GLOBAL HARMONY SEARCH* UNTUK MENYELESAIKAN *0-1 KNAPSACK PROBLEM*

Adha Ariutama[1],Sri Mardiyati[2]
[1]Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia, Tangerang
[2]Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia, Jakarta
adha.ariutama@sci.ui.ac.id

Abstrak

0-1 Knapsack Problem adalah permasalahan optimasi dalam menentukan objek dari sekumpulan objek tertentu dimana masing-masing objeknya hanya mempunyai satu unit. Masing-masing objek tersebut mempunyai bobot (*weight*) dan nilai (*profit*) yang dimasukkan ke dalam suatu media penyimpanan yang mempunyai kapasitas tertentu sehingga banyaknya bobot dari objek-objek tersebut tidak melebihi kapasitas dan nilai yang didapatkan maksimum. Salah satu metode untuk menyelesaikan 0-1 *Knapsack Problem* adalah algoritma *Novel Global Harmony Search* (NGHS). Algoritma *Novel Global Harmony Search* (NGHS) merupakan bentuk modifikasi atau pengembangan algoritma dari algoritma *Harmony Search*. Kemudian akan dibandingkan hasil penyelesaian 0-1 KP yang menggunakan algoritma NGHS dengan algoritma *Harmony Search* (HS).

Kata Kunci :0-1 Knapsack Problem (0-1 KP), Algoritma Harmony Search (HS), Algoritma Novel Global Harmony Search (NGHS).

PENDAHULUAN

Dalam kehidupan sehari-hari, masalah optimasi sering dijumpai dalam kehidupan nyata baik disadari maupun tidak. Salah satu contoh masalah optimasi dalam kehidupan di dunia nyata adalah *Knapsack Problem*. *Knapsack Problem* adalah permasalahan optimasi dalam menentukan objek dari sekumpulan objek tertentu yang mempunyai bobot (*weight*) dan nilai (*profit*) yang dimasukkan ke dalam suatu media penyimpanan yang mempunyai kapasitas tertentu sehingga banyaknya bobot dari objek-objek tersebut tidak melebihi kapasitas dan mendapatkan nilai yang maksimum (Pisinger, 1995).

Penyelesaian *Knapsack Problem* dapat diselesaikan dengan metode eksak dan metode heuristik. Metode eksak yang dapat digunakan dalam menyelesaikan {*0,1} Knapsack Problem* adalah *Branch & Bound* oleh Martello pada tahun 1990, dan *cutting plane* oleh Gen & Yu pada tahun 2010, sedangkan metode heuristik yang dapat digunakan dalam menyelesaikan {*0,1} Knapsack Problem* adalah *ant colony optimization* (ACO) oleh Shi pada tahun 2006, *Schema-guiding evolutionary algorithm* (SGEA) oleh Liu pada tahun 2009, *genetic algorithm* (GA) oleh Lin pada tahun 2008, dan *Harmony Search* (HS) oleh geem pada tahun 2001. Pada tahun 2011, muncul algoritma *Novel Global Harmony Search* (NGHS) yang merupakan suatu modifikasi algoritma dari *Harmony Search* (HS) dimana Kekonvergenan algoritma NGHS dalam pencarian solusi optimal lebih baik dari HS (Zou et al 2011). Pada penulisan ini akan digunakan metode algoritma *Novel Global Harmony Search* (NGHS) untuk menyelesaikan masalah {*0,1} Knapsack Problem.*

PEMBAHASAN

0-1 *Knapsack Problem*

{0,1}-*Knapsack Problem* (0-1 KP) adalah KP dimana setiap objek hanya tersedia satu unit. Secara matematis, 0-1 KP dapat diformulasikan sebagai berikut.

maks $\sum_{i\epsilon N} p_i x_i$

dengan kendala :

$$\sum_{i\epsilon N} w_i x_i \leq C \, , i \in N$$

$$x_i \in \{0,1\} \, , i \in N$$

Pada persaman diatas, variabel keputusan x akan bernilai 1 jika item ditempatkan dalam *Knapsack* dan bernilai 0 jika tidak ditempatkan dalam *Knapsack* (Martello & Toth, 1990).

Algoritma *Novel Global Harmony Search*

Algoritma *Novel Global Harmony Search* (NGHS) adalah bentuk modifikasi atau pengembangan algoritma dari algoritma *Harmony Search* yang proses algoritmanya terinsprisasi dari Particle Swarm Optimization (PSO) (Zou et al 2011). berikut adalah perbedaan antara Algoritma *Harmony Search* dengan Algoritma *Novel Global Harmony Search* (Zou et al 2010).

1) Parameter HMCR & PAR tidak termasuk dalam NGHS, diganti dengan parameter *genetic mutation probability* (p_m).
2) Algoritma NGHS memodifikasi tahap improvisasi harmoni baru dari algoritma HS.

Improvisasi Harmoni Baru

Berikut tahapan untuk membuat harmoni baru NGHS yang melibatkan HM awal sebelumnya. Akan dilihat harmoni-harmoni yang mempunyai nilai terbaik dan nilai terburuk yang nantinya diproses melalui improvisasi harmoni baru.

a) Menghitung *adaptive step*

Langkah awal untuk melakukan improvisasi harmoni baru adalah dengan menghitung *adaptive step.*

$$step_i = |x_i^{Best} - x_i^{Worst}| \qquad (1)$$

dimana :

x_i^{Best} merupakan variabel keputusan ke-i pada harmoni terbaik.

x_i^{Worst} merupakan variabel keputusan ke-i pada harmoni terburuk.

b) Melakukan *position updating*

Setelah menghitung $step_i$, selanjutnya akan dilakukan *position updating* dengan ditentukannya nilai variabel keputusan ke-i pada harmoni baru x_i' sebagai berikut :

$$x_i' = x_i^{Best} + r \cdot step_i \qquad (2)$$

dimana r adalah bilangan acak antara 0 dan 1. Setelah didapatkan x_i' dari proses *position updating*, akan diproses untuk tahapan *genetic mutation*.

c) Tahapan *genetic mutation*

Akan dibangkitkan suatu random baru lagi yaitu $rand1$ dan $rand2$ yang diproses sebagai berikut.

jika $rand1 \leq p_m, maka\ x_i' = x_{iL} + rand2 \cdot (x_{iU} - x_{iL})$

$rand1$ dan $rand2$ adalah nilai random antara 0 dan 1

$pm = \frac{2}{N}$ (Zou et al , 2011).

Ketika nilai $rand1 > p_m$ maka x_i' hanya melakukan proses *position updating*

NGHS mengantikan worst harmony (x^{Worst}) pada HM dengan harmoni baru x', meskipun x' lebih buruk dari x^{Worst}.

Prosedur algoritma NGHS :

1) Inisialisasi masalah dan parameter.
2) Pembentukan Harmony Memory.
3) Improvisasi Harmoni baru.
4) Perbarui *Harmony Memory.*
5) Pengecekan Kondisi Berhenti.

Berikut adalah contoh Aplikasi algoritma *Novel Global Harmony Search* pada penyelesaian 0-1 KP. Pandang suatu permasalahan 0-1 KP dimana nilai = $\{9,11,13,15\}$, $w = \{6,5,9,7\}$ dan $C = 20$ yang diformulasikan sebagai berikut :

$$\text{Maks } f = 9x_1 + 11x_2 + 13x_3 + 15x_4 \quad (3)$$

$$\text{Kendala : } g = 6x_1 + 5x_2 + 9x_3 + 7x_4 - 20 \leq 0 \quad (4)$$

$$x_i \in \{0,1\}$$

Untuk penyelesaian masalah diatas, ditetapkan parameter berikut, HMS =5, pm = $\frac{2}{N} = 0{,}5$,NI= 10000 (Zou et al, 2011)

a) Inisialisasi *Harmony Memory*

Dibentuk matriks HM yang entri-entrinya dibentuk secara random berdasarkan *random selection* sebagai berikut :

$$x_i^j = x_{iL} + rand \cdot (x_{iU} - x_{iL})$$

Sehingga matriks HM seperti pada gambar berikut :

$$\text{HM} = \begin{bmatrix} 0{,}6948 & 0{,}3816 & 0{,}4456 & 0{,}6797 \\ 0{,}3171 & 0{,}7655 & 0{,}6463 & 0{,}6551 \\ 0{,}9502 & 0{,}7952 & 0{,}7094 & 0{,}1626 \\ 0{,}0344 & 0{,}1869 & 0{,}7547 & 0{,}1190 \\ 0{,}4387 & 0{,}4898 & 0{,}2760 & 0{,}4984 \end{bmatrix} \quad (5)$$

Selanjutnya dilakukan pembulatan entri-entri pada HM untuk membentuk kandidat HM sebagai berikut.

$$Kandidat\ HM = \begin{bmatrix} H1 \\ H2 \\ H3 \\ H4 \\ H5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (6)$$

Langkah selanjutnya adalah menghitung nilai fungsi tujuan persamaan (3) serta menghitung kendala persamaan (4) berdasarkan nilai-nilai variabel keputusan persamaan (6) Jika terdapat harmoni yang tidak memenuhi syarat 0-1 KP, maka akan dibentuk kembali *random selection* untuk harmoni tersebut.

- Untuk bagian harmoni 2 , didapat nilai fungsi tujuan sebagai berikut.
  $f = 9.0 + 11.1 + 13.1 + 15.1 = 39$ (9)

  Dengan kendala,
  $g = 6.0 + 5.1 + 9.1 + 7.1 - 20 = 1 \nleq 0\ (1 > 0)$ (10)

  Dari persamaan (9) dan persamaan (10), harmoni 2 mempunyai nilai fungsi tujuan sebesar 39, namun tidak memenuhi syarat 0-1 KP. maka akan dilakukan *random selection* kembali untuk harmoni 2 . Berikut adalah harmoni 2 yang sudah diperbarui.
- $x_1^2 = 0 + 0{,}1622(1 - 0) = 0{,}1622.$ Nilai yang mendekati kandidat solusi {0,1}

  untuk $x_1^1 = 0, 1622$ adalah 0

  $x_2^2 = 0 + 0{,}7943(1 - 0) = 0{,}7943.$ Nilai yang mendekati kandidat solusi {0,1}

  untuk $x_1^1 = 0{,}7943$ adalah 1
- $x_3^2 = 0 + 0{,}3112(1 - 0) = 0{,}3112.$ Nilai yang mendekati kandidat solusi {0,1}

  untuk $x_1^1 = 0{,}3112$ adalah 0
- $x_4^2 = 0 + 0{,}5285(1 - 0) = 0{,}5285.$ Nilai yang mendekati kandidat solusi {0,1}

  untuk $x_1^1 = 0, 5285$ adalah 1

Periksa kembali harmoni 2 untuk nilai fungsi tujuan serta memenuhi syarat 0-1 KP dengan cara mensubstitusi nilai-nilai harmoni 2 terhadap fungsi tujuan persamaan (3) dan kendala persamaan (4).

- Untuk bagian harmoni 2 yang sudah diperbarui , akan didapat nilai fungsi tujuan sebagai berikut.
  $f = 9.0 + 11.1 + 13.0 + 15.1 = 26$ (17)

  Dengan kendala,
  $g = 6.0 + 5.1 + 9.0 + 7.1 - 20 = -8 \leq 0$ (18)

  Dari persamaan (17) dan (18), harmoni 2 mempunyai nilai fungsi tujuan sebesar 26, serta memenuhi syarat 0-1 KP.

Masukkan harmoni 2 ke HM sebagai berikut.

$$\text{HM} = \begin{bmatrix} 0{,}6948 & 0{,}3816 & 0{,}4456 & 0{,}6797 \\ \mathbf{0,1622} & \mathbf{0{,}7943} & \mathbf{0{,}3112} & \mathbf{0{,}5285} \\ 0{,}9502 & 0{,}7952 & 0{,}7094 & 0{,}1626 \\ 0{,}0344 & 0{,}1869 & 0{,}7547 & 0{,}1190 \\ 0{,}4387 & 0{,}4898 & 0{,}2760 & 0{,}4984 \end{bmatrix} \quad (19)$$

Selanjutnya dari HM diatas, dibentuk kembali matriks {0,1} yaitu kandidat HM sebagai berikut.

$$kandidat\ HM = \begin{bmatrix} H1 \\ H2 \\ H3 \\ H4 \\ H5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (20)$$

Karena setelah semua harmoni dihitung nilai fungsi tujuan serta semua harmoni sudah memenuhi syarat 0-1 KP. Langkah selanjutnya adalah mensubstitusikan nilai fungsi tujuan untuk masing-masing harmoni seperti pada persamaan (21) dibawah ini.

$$\begin{bmatrix} f(H1) \\ f(H2) \\ f(H3) \\ f(H4) \\ f(H5) \end{bmatrix} = \begin{bmatrix} 24 \\ 26 \\ 33 \\ 13 \\ 0 \end{bmatrix} \quad (21)$$

Semakin besar nilai fungsi tujuan suatu harmoni, semakin baik harmoni tersebut. Terlihat jelas pada persamaan (21) bahwa harmoni 3 merupakan harmoni terbaik. sedangkan harmoni 5 merupakan harmoni terburuk yaitu bernilai nol. Karena harmoni 5 merupakan harmoni yang terburuk, maka harmoni tersebut akan dilakukan tahap tiga, yaitu improvisasi harmoni baru.

b) Improvisasi Harmoni Baru.

Pada tahap ini, harmoni yang terburuk akan diganti dengan harmoni baru. Berikut adalah langkah-langkah pembangkitan harmoni baru.

- Untuk $i = 1$, x_1^{Best} adalah $0{,}9502$, sedangkan x_1^{Worst} adalah $0{,}4387$. harmony baru mempunyai adaptive step dimana,

  $step_1 = |0{,}9502 - 0{,}4387| = 0{,}5115$ (22)

  Misal nilai random r yang diberikan adalah $0{,}8147$.

  $x_1' = 0{,}9502 + 0{,}8147 \times 0{,}5115 = 1{,}3670$ (23)

  Selanjutkan akan dibangkitkan kembali nilai random $(rand1)$ untuk memasuki tahap *genetic mutation*. Misal nilai random yang diberikan adalah $0{,}7715$.

Karena $0{,}7715 > 0{,}5$ maka nilai x_1' persamaan (23) dipertahankan. nilai $x_1' = 1{,}3670$ yang mendekati kandidat solusi {0,1} adalah 1. Seterusnya sampai $i = 4$.

Sehingga didapatkan vektor solusi baru atau harmoni baru $x' = \{1,1,0,0\}$.

Langka selanjutnya adalah menghitung nilai fungsi tujuan pada harmoni baru dan menghitung kendala persamaan (4) agar memenuhi syarat 0-1 KP. Berikut adalah perhitungan nilai fungsi tujuan serta kendala 0-1 KP.

$$f = 9.1 + 11.1 + 13.0 + 15.0 = 20 \quad (24)$$

Dengan kendala,

$$g = 6.1 + 5.1 + 9.0 + 7.0 - 20 = -11 \leq 0 \quad (25)$$

Dari persamaan (24) dan persamaan (25), harmoni baru mempunyai nilai fungsi tujuan sebesar 20, serta memenuhi syarat 0-1 KP. Sehingga tidak perlu lagi melakukan proses impovisasi harmoni baru. Berikut adalah HM yang sudah diperbarui pada bagian harmoni 5

$$\text{HM} = \begin{bmatrix} 0{,}6948 & 0{,}3816 & 0{,}4456 & 0{,}6797 \\ 0{,}1622 & 0{,}7943 & 0{,}3112 & 0{,}5285 \\ 0{,}9502 & 0{,}7952 & 0{,}7094 & 0{,}1626 \\ 0{,}0344 & 0{,}1869 & 0{,}7547 & 0{,}1190 \\ \mathbf{1,3670} & \mathbf{1,0783} & \mathbf{0,4867} & \mathbf{0,4693} \end{bmatrix} \quad (26)$$

Dengan kandidat HM sebaga berikut.

$$kandidat\ HM = \begin{bmatrix} H1 \\ H2 \\ H3 \\ H4 \\ H5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (27)$$

Setelah mendapatkan harmoni baru, maka lanjut proses update *Harmony Memory*.

c) Update *Harmony Memory*

Setelah didapatkan kandidat harmoni baru $\boldsymbol{x}' = (1,1,0,0)$, kandidat harmoni baru langsung mengantikan kandidat harmoni lama. dengan nilai fungsi tujuan untuk kandidat harmony baru adalah sebagai berikut.

$$\begin{bmatrix} f(H1) \\ f(H2) \\ f(H3) \\ f(H4) \\ f(H5) \end{bmatrix} = \begin{bmatrix} 24 \\ 26 \\ 33 \\ 13 \\ \mathbf{20} \end{bmatrix} \quad (28)$$

Dari persamaan (28) diatas, harmoni 4 menjadi harmoni yang terburuk, sehingga harmoni tersebut akan dilakukan tahap 3 untuk improvisasi harmoni baru untuk menggantikan harmoni 4 tersebut dan dilanjutnya iterasi selanjutnya sampai batas NI yang sudah ditetapkan.

Setelah menjalankan beberapa iterasi yang mengakibatkan pergantian harmoni serta kandidat harmoni, akan dapat hasil akhir untuk masing-masing harmoni sebagai berikut

$$\begin{bmatrix} f(H1) \\ f(H2) \\ f(H3) \\ f(H4) \\ f(H5) \end{bmatrix} = \begin{bmatrix} 24 & 24 & 24 & 24 & \cdots & 35 \\ 26 & 26 & 26 & 26 & \cdots & 35 \\ 33 & 33 & 33 & 33 & \cdots & 35 \\ 13 & 13 & 28 & 28 & \cdots & 35 \\ 0 & 20 & 20 & 35 & \cdots & 35 \end{bmatrix} \quad (29)$$

Berdasarkan persamaan (29), selama proses iterasi berjalan, hasil yang didapat adalah nilai fungsi tujuan harmoni menuju ke nilai yang sama yaitu sebesar 35.

d) Pengecekan Kondisi Berhenti

Kondisi berhenti ketika semua harmoni tidak melakukan update harmoni seperti pada persamaan (29) atau mencapai nilai iterasi maksimum (NI) sebesar 10.000.
Solusi optimal yang didapatkan dalam menyelesaikan permasalahan 0-1 KP (N=4) dengan menggunakan algoritma *Novel Global Harmony Search* Adalah $x^* = (1,1,0,1)$ dengan nilai $f(x^*) = 35$.

HASIL SIMULASI

Akan digunakan enam data set dari *0-1 Knapsack Problem* . data 0-1 KP berisi masalah-masalah yang mempunyai dimensi, bobot, nilai dan kapasitas yang berbeda-beda, data 0-1 KP dapat dilihat pada tabel berikut ini.

Tabel 1. Data 0-1 KP untuk Simulasi Program

f	**Dimensi/ Jumlah objek (N)**	**Kapasitas (C)**	**Bobot (w)**	**Nilai (p)**
P_1	10	269	(95,4,60,32,23,72,80,62,65,46)	(55,10,47,5,4,50,8,61,85,87)
P_2	20	878	(92,4,43,83,84,68,92,82,6,44,32,18,56,83,25,96,70,48,14,58)	(44,46,90,72,91,40,75,35,8,54,78,40,77,15,61,17,75,29,75,63)
P_3	4	20	(6,5,9,7)	(9,11,13,15)
P_4	10	60	(30,25,20,18,17,11,5,2,1,1)	(20,18,17,15,15,10,5,3,1,1)
P_5	7	50	(31,10,20,19,4,3,6)	(70,20,39,37,7,5,10)
P_6	5	80	(15,20,17,8,31)	(33,24,36,37,12)

[Sumber: Zou et al 2011]

Data tersebut akan dilakukan simulasi terhadap HS dan NGHS, kemudian hasilnya akan dibandingkan. Dalam penelitian ini akan digunakan beberapa parameter terbaik untuk permasalahan yang digunakan, dan berikut adalah parameternya.

Tabel 2. Parameter Pada Simulasi Program

HMS	5
pm	2/N
HMCR	0.9
PAR	0.3
Bw	1
NI	10000

(Zou et al 2011)

Dari ke enam data tersebut, akan dilakukan simulasi program sebanyak lima puluh kali dengan alur algoritma yang telah disimulasikan dalam perangkat lunak. Output yang akan dikeluarkan dalam simulasi tersebut berupa nilai fungsi tujuan, rata-rata iterasi konvergensi dan rata-rata waktu komputasi. Kemudian didapatkan hasil simulasinya yang akan disajikan pada tabel berikut.

Tabel 3. Hasil Simulasi Program

Masalah	**Jumlah objek**	**Algoritma**	**Nilai fungsi tujuan terbaik**	**Rata-rata iterasi**	**Rata-rata waktu komputasi (dalam detik)**
P_1	10	HS	295	248	0.4053
		NGHS	295	16	0.0013
P_2	20	HS	1024	4082	0,4461
		NGHS	1024	29	0.0021
P_3	4	HS	35	12	0.001003
		NGHS	35	11	0.00081
P_4	10	HS	52	22	0.028111
		NGHS	52	12	0.001333
P_5	7	HS	107	1452	0.467132
		NGHS	107	14	0.000926
P_6	5	HS	130	18	0.001468
		NGHS	130	14	0.001234

Dari hasil simulasi pada tabel (3), jelas bahwa untuk setiap masalah dari P_1 sampai P_6, menghasilkan nilai fungsi tujuan terbaik yang sama. Terlihat jelas juga bahwa algoritma NGHS lebih cepat dalam mencapai kekonvergenannya dibandingkan dengan HS dengan melihat hasil rata-rata iterasi dan waktu komputasinya.

SIMPULAN

Algoritma NGHS dapat diterapkan dalam menyelesaikan 0-1 KP. Pada hasil Tabel (3) , hasil penyelesaian 0-1 KP dengan menggunakan algoritma NGHS lebih baik dibandingkan dengan menggunakan *Harmony Search* HS dimana kekonvergenan dan waktu komputasi algoritma NGHS lebih cepat dibandingkan algoritma HS

DAFTAR PUSTAKA

Geem, Z. W., Kim, J. H., & Loganathan, G.V. (2001). *A New heuristic Optimization Algorithm* : *Harmony Search*, *Simulation* 76(2), 60-68.

Lin, F.T.(2008). Solving the knapsack problem with imprecise weight coefficients using genetic algorithms. *European Journal of Operational Research*, 185 (1) 133–145.

Liu, Y. & Liu, C.(2009). *A schema-guiding evolutionary algorithm for 0–1 knapsack problem.* International Association of Computer Science and Information Technology—Spring Conference, pp. 160–164

Martello, S. & Toth, P. (1990). *Knapsack Problem : Algorithms and Computer Implementations.* Wiley, Chicester, UK.

Pisinger, David. (1995). *Algorithm of Knapsack Problem*. University of Copenhagen. Departemen of Computer Science.

Shi, H.X.(2006). Solution to 0/1 knapsack problem based on improved ant colony Algorithm. *International Conference on Information Acquisition*, pp.1062–1066.

Yu, Xinjie. & Gen, Mitsuo.(2010). Introduction to Evolutionary Algorithm. *Decision Engineering*, page 267.

Zou, D., Gao, L., Li, S., & Wu, J. (2011). Solving 0-1 Knapsack Problem by a Novel Global Harmony Search Algorithm. *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 1556–1564.

Zou, D., Gao, L., Li, S., & Wu, J. (2010). *A novel global harmony search algorithm for reliability problems*. Comput Ind Eng 2010;58(2):307–16.