

- Word Count: 3629

Plagiarism Percentage

12%

sources:

- 1 8% match (Internet from 17-Jan-2018)
<http://jurnalindustri.petra.ac.id/index.php/ind/article/view/20486>
- 2 1% match (publications)
[Marinela Krstinić Nižić, Zvonimira Šverko Grdić, Renata Endres. "Energy Sustainability and Its Impacts on Croatian Tourism", Croatian Economic Survey, 2017](#)
- 3 1% match (Internet from 04-Jun-2017)
<http://puslit2.petra.ac.id/ejournal/index.php/ind/article/download/18964/18635>
- 4 1% match (Internet from 21-Mar-2016)
http://www.jucs.org/jucs_21_4/heuristic_algorithms_for_manufacturing/jucs_21_04_0503_0525_bucki.pdf
- 5 1% match (Internet from 04-Jan-2018)
<https://bazybg.uek.krakow.pl/bazekon/rekord/171426451>
- 6 1% match (Internet from 10-Jun-2015)
http://www.cs.uoi.gr/~kostasp/citations/cita_scopus_papers_2013_09.txt
- 7 < 1% match (Internet from 07-Feb-2017)
<https://www.omicsonline.com/open-access/memetic-harmony-search-algorithm-based-on-multiobjective-differentialevolution-of-evolving-spiking-neural-networks-2090-4908-1000130.php?aid=69097>
- 8 < 1% match (Internet from 13-Nov-2013)
<http://scholar.sun.ac.za/bitstream/handle/10019.1/1644/Vanhee.pdf.txt?sequence=3>

paper text:

Jurnal Teknik Industri, Vol. 19, No. 2, December 2017, 125-132 DOI: 10.9744/jti.19.2.125-132

ISSN 1411-2485 print / ISSN 2087-7439 online

3

Perbandingan Algoritma Simulated Annealing dan Harmony Search dalam Penerapan Picking Order Sequence Tanti

1

Octavia1*, Septianda Angelica1

Abstract: Implementation of mobile rack warehouse is commonly used in manufacturing industry because it can minimize the warehouse area used. Applying picking orders in taking of Stock Keeping Unit (SKU) on mobile rack warehouses could give fast loading order. This research aims to find out which algorithm is better in applying picking order sequence in mobile rack warehouse. The algorithms used are Simulated Annealing (SA) and Harmony Search (HS) algorithm. Both of these algorithms will be compared in terms of the gap with the shortest path method and CPU time. The result shows that the HS algorithm produces a better solution than the SA algorithm with lower CPU time, but the convergence rate of HS is lower than that of SA. HS was able to produce a better solution than the shortest path method of nine cases, while SA only eight cases from fifteen cases. **Keywords:** Picking order sequence, simulated annealing, harmony search.

1

Pendahuluan Penyimpanan memiliki peranan penting dalam industri manufaktur. Gudang sangat dibutuhkan dalam proses penyimpanan agar barang yang disimpan tidak rusak dan tetap terjaga. Pengaturan gudang yang optimal juga dapat meminimalkan penggunaan tempat sehingga dapat meningkatkan kapasitas gudang yang ada. Adapun pengaturan gudang dapat dilakukan seperti mengatur peletakan barang yang disimpan, mengubah tata letak gudang yang ada, serta penggunaan media penyimpanan seperti rak. Penggunaan mobile rack warehouse umumnya cocok digunakan karena dapat menghemat tempat yang tersedia dan dapat bergerak secara otomatis. Dimana, pengambilan stock keeping unit (SKU) yang terletak pada rak dilakukan oleh storage and retrieval vehicle (SRV). Kecepatan dan waktu yang dibutuhkan untuk mengambil SKU sesuai yang diperlukan akan sangat tergantung pada penataan serta relokasi rak yang optimal agar waktu tempuh SRV dalam mengambil SKU semakin optimal. Penataan serta urutan pengambilan (picking order sequence) SKU dengan lokasi yang berdekatan akan mengoptimalkan waktu pengambilan SKU. Dimana pada penelitian ini akan dilakukan optimalisasi kecepatan dan waktu pengambilan SKU dengan menggunakan metode algoritma simulated annealing (SA) and harmony search (HS).

**1 Fakultas Teknologi Industri, Program Studi Teknik Industri,
Universitas Kristen Petra, Jl. Siwalankerto 121-131, Surabaya 60238,
Indonesia. Email: tanti @petra.ac.id * Penulis korespondensi**

3

Gambar 1. Gambaran permasalahan Gambar 2. Contoh penempatan SKU Seperti contohnya sebuah

gudang memiliki empat buah rak dan satu depo (Gambar 1). Picking order yang diberikan berupa nomor SKU yang memiliki demand tertentu dan selanjutnya dikelompokkan ke masing-masing rak. Setiap SKU ditempatkan pada suatu rak saja dan tidak berpindah-pindah. Peletakan SKU pada setiap rak dilakukan secara berurutan, dimana nomor SKU terkecil akan terletak pada nomor rak terkecil (Gambar 2). Pengambilan order yang pertama kali berdasarkan dari rak yang terbuka pertama kali. Rak yang terbuka pertama kali didapatkan dari membangkitkan bilangan acak. Pengambilan order membutuhkan alokasi rak untuk membuka aisle dimana setiap rak hanya memiliki satu aisle saja. Pertama kali SRV akan berada di depo, kemudian SRV akan bergerak mengambil order ke masing-masing rak. Pengaksesan SKU membutuhkan pergerakan rak selama empat meter/menit. SRV akan memiliki kapasitas tertentu, dimana pada penelitian ini SRV memiliki kapasitas 100. Metode Penelitian Pengembangan Model Model yang akan dikembangkan pada penelitian ini terdiri dari: (1) model awal dengan algoritma simulated annealing (SA) dan harmony search (HS); (2) model dengan shortest path (SP). Model awal dikembangkan dengan cara membangkitkan rute pengambilan SKU secara random, sedangkan model shortest path dikembangkan dengan cara mencari rute dengan waktu tempuh terpendek. Metode ini akan digunakan sebagai pembandingan untuk algoritma SA dan HS karena metode shortest path cukup mampu menghasilkan solusi paling minimum meskipun membutuhkan waktu yang cukup lama. Batasan pada penelitian ini adalah hanya menggunakan satu buah SRV dan SRV tidak diperbolehkan mengalami backtracking. Batasan ini ditujukan agar dapat meminimalkan waktu tempuh dari SRV. Adapun total waktu tempuh optimal dilihat pada persamaan berikut ini:
$$T = \sum_{i=1}^n \sum_{j=1}^m t_{ij} \quad \text{dengan kendala: } \sum_{i=1}^n x_{ij} \leq 1, \quad \sum_{j=1}^m x_{ij} = 1, \quad x_{ij} \in \{0, 1\}$$
 dimana: t_{ij} = Waktu tempuh rak ke rak i ke rak j = { 0 1 =

$i \neq k, j \neq l$ (1) (2) (3) (4)

8

Persamaan (2) digunakan untuk membatasi total permintaan yang dibawa oleh SRV tidak melebihi kapasitas. Persamaan (3) dan (4) menunjukkan bahwa semua kemungkinan peletakan rak i pada rute j akan dilakukan sesuai dengan jumlah rak (n) dan rute (m) dan rak i hanya berada pada satu rute j . Jumlah rak dimulai dari angka 2 karena angka 1 menunjukkan depo dimana depo tidak boleh dikunjungi apabila kapasitas SRV masih ada. bernilai 1 jika rak i berada pada rute j dan akan bernilai 0 jika berlaku sebaliknya. Contoh terdapat sebuah rute yaitu (2,3). pada rute ini adalah 21 yang berarti rak 2 berada di rute 1. pada rute ini adalah 32 yang berarti rak 3 berada di rute 2. pada rute ini adalah 23 yang berarti waktu tempuh dari rak 2 ke rak 3. Model Shortest Path Metode ini akan mencari rute dengan waktu tempuh terpendek dengan membandingkan rute mana yang memiliki waktu tempuh paling pendek. Contohnya pada Gambar 3 rak nomor 2 akan diperiksa ke seluruh rute untuk mengetahui rute manakah yang memiliki waktu tempuh terpendek. Dari hasil pencarian, rak nomor 2 memilih rute menuju rak nomor 3 karena rute tersebut memiliki waktu tempuh terpendek. Langkah ini kemudian diulang hingga semua permintaan terpenuhi. Algoritma Simulated Annealing (SA) Simulated Annealing (SA) merupakan algoritma yang digunakan untuk mencari nilai optimal dengan mengontrol nilai dari temperatur (Atmaca dan Ozturk [1]). SA memiliki beberapa parameter yaitu temperatur awal, jumlah penurunan temperatur, derajat penurunan, dan jumlah replikasi setiap temperatur. Adapun penetapan parameter pada penelitian ini mengacu pada Suman dan Kumar [2], Bosen et al [3], dan Yang [4]. Temperatur awal didapatkan dari persamaan berikut ini [4]:
$$T_0 = - \ln(\Delta) \quad (5)$$
 Nilai Δ didapatkan dari pengurangan antara solusi baru dan lama, dimana solusi baru didapatkan dari pengembangan solusi awal dengan menukar dua rute secara random. Misalnya pada solusi awal, hasil yang diperoleh adalah (2,3,4,5), pada solusi baru diperoleh (2,4,3,5). Maka hasil waktu tempuh solusi baru akan dikurangkan dengan hasil waktu tempuh dari solusi lama. Gambar 3. Contoh penerapan metode shortest path START A Penetapan Parameter (HMS, N) dan

Fungsi Memperbarui variabel Tujuan keputusan baru $HMCR_{min} = 0.7$ Menghitung nilai bw pada $HMCR_{max} = 0.99$ iterasi ke n $PAR_{min} = 0.1$ $PAR_{max} = 0.5$ $Bw_{min} = 0.01$ $Bw_{max} = 0.001$ $ak' = ak' \pm rand() \times bw(n)$ Membangkitkan N Harmony Memory (HM) $ak' = aK'$? Y Melakukan improvisasi harmony (Pencarian vektor solusi baru (a')) Membandingkan vektor solusi baru (a') dengan vektor solusi terjelek pada HM Membangkitkan bilangan acak (r) $a' < a_{hm\ worst}$? N a' diabaikan Menghitung HMCR dan PAR pada iterasi ke n Y Memperbarui HM Apakah Membangkitkan variabel $n=n+1$ $r < HMCR(n)$? N keputusan baru dari vektor solusi lama Y Membangkitkan variabel N $n>N$? keputusan baru (ak') untuk dimasukkan pada harmony Y memory HM memiliki vektor solusi terbaik $R < PAR(n)$? N $ak' = ak'$ Y END A Gambar 4.

Flowchart algoritma harmony search SA akan membangkitkan solusi baru dengan dua metode yaitu swap two dan push away. Metode ini sebelumnya sudah dikembangkan oleh Bosen et al. [3]. Pada penelitian ini ditentukan metode swap two yang terjadi sebanyak 90% dan push away sebesar 10%. Contoh dari metode swap two adalah sebagai berikut. Ketika didapatkan solusi awal yaitu (2,4,- 3,5) maka metode swap two akan menukar dua rute secara acak, sehingga solusi baru menjadi (2,3,5,4). Contoh penggunaan dari metode push away adalah pada saat solusi awal didapatkan (2,3,5,4). Metode push away akan meletakkan salah satu rak ke rute paling terakhir sehingga solusi baru menjadi (2,5,4,3). Penentuan rak mana yang akan dipilih berdasarkan hasil random data. Nilai dari Po menurut Suman dan Kumar [2] memiliki kisaran antara 0,5 – 0,95, sedangkan untuk penelitian nilai Po yang digunakan sebesar 0,95. Parameter selanjutnya yaitu derajat penurunan sebesar 0,995 (Bosen et al. [3]), jumlah penurunan temperatur sebanyak 1000 kali, dan jumlah replikasi setiap temperatur sebanyak 5 kali. Algoritma Harmony Search (HS) Harmony search (HS) adalah algoritma yang digunakan untuk mencari nilai optimal yang terinspirasi dari musisi yang mengimprovisasi pitch instrument untuk mendapatkan harmoni yang baik (Zong [5]). Beberapa penelitian dan pengembangan algoritma HS telah dilakukan baik dalam perbaikan parameter algoritma HS itu sendiri maupun perbaikan penerapan algoritma HS dengan algoritma meta-heuristik lain (Alia dan Mandava [6]), misalnya: Ant colony (Yun et al. [7]). Penerapan algoritma HS ini telah banyak dilakukan pada permasalahan traveling salesman problem (Yun et al.[7], Zong et al. [8]); joshop problem (Aulia [9]); maupun four colour mapping (Daham et al. [10]). Penerapan algoritma HS pada ketiga permasalahan di atas telah memberikan hasil yang lebih baik. Beberapa parameter yang dimiliki HS yaitu harmo- ny

memory (HM), harmony memory consideration rate (HMCR), harmony memory size (HMS), pitch adjusting rate (PAR),

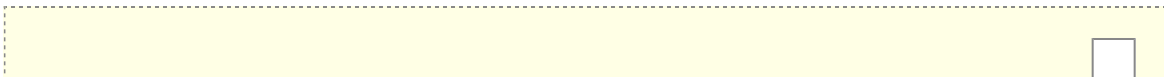
7

dan jumlah iterasi (N). Penetapan parameter pada penelitian ini berdasarkan dari Yang [4], Zong [5], dan Daham et al. [10]. (6) HM berisi vektor–vektor solusi sebanyak HMS, dimana vektor–vektor solusi pada penelitian ini didapatkan secara random. Parameter selanjutnya yaitu HMS sebesar 40, jumlah iterasi sebanyak 1000. Parameter HMCR dan PAR tidak ditetapkan secara pasti tetapi didapatkan melalui persamaan (7): $() = (- \times) () = (- \times)$ (7) (8) Gambar 4 menunjukkan bagaimana algoritma HS dijalankan. HS akan membangkitkan vektor solusi baru berdasarkan dari vektor solusi yang berada di HMS. Hasil dan Pembahasan Simulasi dilakukan dengan menggunakan visual basic application pada aplikasi Microsoft Excel. Simulasi dijalankan menggunakan Laptop dengan spesifikasi prosesor Intel Gen-7 Core i7-7700HQ dan memori DDR4 16 GB 2400Mhz. Simulasi dilakukan pada kasus permasalahan dengan jumlah rak (10,20,30,40,50) dan jumlah order (25,50,100). Setiap kasus direplikasi sebanyak 50 kali. Simulasi dilakukan pada kedua metode dengan mengganti parameter pada model sebanyak tiga kali. Pergantian parameter dilakukan untuk mengetahui pengaruh perubahan parameter terhadap hasil dari fungsi tujuan dan CPU time simulasi tersebut. Parameter Pertama Parameter ini adalah parameter awal untuk kedua

algoritma. Parameter pertama SA adalah sebagai berikut: Temperatur awal Temperatur awal didapatkan dari persamaan (5). Perhitungan nilai Δ dilakukan untuk mengetahui nilai maksimum Δ dari permasalahan (lihat Tabel Tabel 1. Nilai Δ setiap kasus Rak/order 10 20 30 40 50 25 -7,6933 -34,186 -22,559 -39,922 -52,790 50 -11,861 -23,627 -34,921 -31,953 -62,564 100 -11,515 -43,358 -36,697 -79,021 -83,220 Tabel 2. Rata-rata selisih waktu terpendek algoritma SA dan HS dengan shortest path (SP) parameter pertama Jumlah rak Jumlah order 10 25 50 100 20 25 50 100 30 25 50 100 40 25 50 100 50 25 50 100 SA-SP -6,8986 17,069 18,211 47,791 44,749 -3,3398 -17,338 36,762 36,461 84,147 -37,187 -8,8917 -10,444 77,167 0 HS-SP -13,535 0 -2,0783 18,795 39,924 -7,4702 -35,935 12,815 -49,384 18,795 -37,237 -26,021 -30,821 6,0613 0 1). Nilai yang diambil adalah -7,6933 yaitu pada kasus jumlah rak 10 dengan jumlah order 25. Nilai ini akan digunakan pada perhitungan temperatur awal dengan P_0 sebesar 0,95. Temperatur awal yang didapatkan sebesar 149,99. Parameter HS yang pertama adalah sebagai berikut: Harmony Memory Size : 40 HMCR dan PAR didapatkan melalui persamaan (6) dan (7) yang dihitung pada setiap iterasi (n) Jumlah iterasi (N) : 1000 Jumlah penurunan : 1000 Derajat penurunan : 0,995 Replikasi setiap temperatur : 5 Tabel 2 menunjukkan hasil antara kedua algoritma (SA dan HS) dengan metode shortest path. Hasil simulasi menunjukkan bahwa HS mampu mendapatkan hasil yang lebih baik daripada SA. Hal itu ditunjukkan dari 15 kasus yang dikembangkan, HS mampu lebih baik dari metode shortest path sebanyak 8 kasus. SA hanya mampu memiliki hasil yang lebih baik dari metode shortest path sebanyak 6 kasus. Gambar 5 dan 6 menunjukkan bahwa algoritma SA lebih mampu menghasilkan hasil yang lebih konvergen dibandingkan algoritma HS. Hal ini dapat dilihat bahwa SA mulai menghasilkan solusi yang konvergen saat memasuki temperatur kelima atau memasuki iterasi ke 600. HS baru mulai menghasilkan solusi yang konvergen saat memasuki iterasi ke 900. Hal ini berarti SA mampu menghasilkan solusi dengan variasi yang lebih kecil dibandingkan HS. Tabel 3 menunjukkan waktu pengerjaan masing-masing algoritma dalam satuan menit. SA memiliki rata-rata waktu pengerjaan selama 180,7826 menit, sedangkan HS selama 81,1056 menit. Hasil rata-rata ini menunjukkan bahwa HS dapat menghasilkan solusi yang lebih cepat dibandingkan SA. Parameter Kedua Parameter ini ditentukan dengan mengurangi jumlah iterasi untuk masing-masing algoritma. Parameter kedua algoritma SA adalah sebagai berikut : Temperatur awal : 149,99 Jumlah penurunan : 500 Derajat penurunan : 0,995 Replikasi setiap temperatur : 5 Perubahan temperatur SA ini berdasarkan dari hasil konvergensi yang menunjukkan bahwa SA sudah mampu menghasilkan solusi yang konvergen pada saat memasuki iterasi ke 600. Parameter HS yang kedua adalah sebagai berikut: Harmony memory size: 40 HMCR dan PAR didapatkan melalui persamaan (6) dan (7) yang dihitung setiap iterasi (n) Jumlah iterasi (N): 500 HS masih mendapatkan hasil yang lebih baik daripada SA. Hal itu ditunjukkan dari 15 kasus, HS memiliki hasil yang lebih baik dari metode shortest path sebanyak 9 kasus, sedangkan SA sebanyak 5 kasus. Gambar 5. Konvergensi simulated annealing Gambar 6. Konvergensi harmony search Tabel 3. Hasil CPU time parameter pertama (dalam menit) Jumlah rak Jumlah order 10 25 50 100 20 25 50 100 30 25 50 100 40 25 50 100 50 25 50 100 Rata-rata SA 110,1 131,4 134,783 145,25 173,983 181,917 142 181,05 221,067 147,983 215,983 248,04 214,083 213,683 250,417 180,7826 HS 37,433 36,117 42,883 36,117 34,6 30,9 31,9 133,733 134,167 41,85 54 144,05 125,367 175,867 157,6 81,1056 Gambar 7. Konvergensi simulated annealing Gambar 8. Konvergensi harmony search Hasil pergantian parameter menunjukkan bahwa hasilkan solusi yang lebih konvergen daripada parameter pertama. Hal itu dikarenakan jumlah iterasi yang semakin sedikit menunjukkan bahwa jumlah tersebut masih belum cukup untuk mendapatkan hasil yang konvergen. Parameter kedua ini HS lebih baik dalam menghasilkan solusi yang lebih konvergen, walaupun HS menghasilkan solusi yang konvergen pada saat memasuki iterasi ke 470. Tabel 4. Rata-rata selisih waktu terpendek algoritma SA dan HS dengan shortest path (SP) parameter kedua Jumlah rak Jumlah order 10 25 50 100 20 25 50 100 30 25 50 100 40 25 50 100 50 25 50 100 SA-SP -5,533536 -12,42853 -44,4382 15,076138 45,133386 19,273 -42,02161 71,38797 14,1265 56,7382 83,5331 79,289318 37,374504

-15,01927 2,140286 HS-SP -11,49738 -10,4164 -29,0162 6,68741 -44,461196 -27,4406 -31,48758
-25,184357 -21,4100053 27,98565 11,201 28,1444 -21,32427 86,234 104,289925 Tabel 5. Hasil CPU time
parameter kedua (dalam menit) Jumlah rak Jumlah order 10 25 50 100 20 25 50 100 30 25 50 100 40 25
50 100 50 25 50 100 SA 56,133 57,6167 79 70,433 72,4 94,033 86,3833 95,8 111,8 95,433 100,433
130,417 106,75 111,217 155,717 HS 22,7 32,133 38,6333 18,783 19,9167 20,45 21,2833 25,733 20,9667
34,3667 27,6667 38,5 35,9333 29,5333 41,2667 Rata-rata 94,9044 28,52438 Tabel 6. Rata-rata selisih
waktu terpendek algoritma SA dan HS dengan shortest path (SP) parameter ketiga Jumlah rak Jumlah
order SA-SP HS-SP 10 25 50 100 20 25 50 100 30 25 50 100 40 25 50 100 50 25 50 100 -6,2877 -5,9812
-26,613 19,8819 42,3377 -45,152 12,5476 34,4871 -34,091 -15 68,8062 8,68817 -11,008 22,3898 -24,371
-11,49738 -10,4164 -29,0162 6,68741 -44,461196 -27,4406 -31,48758 -25,184357 -21,4100053 27,98565
11,201 28,1444 -21,32427 86,234 104,289925 Tabel 5. menunjukkan SA memiliki rata-rata waktu
pengerjaan selama 94,9044 menit, sedangkan HS selama 28,52438 menit. Hasil rata-rata ini menun-
jukkan bahwa HS masih lebih cepat dalam segi waktu pengerjaan dibandingkan SA. Parameter Ketiga
Parameter ini ditentukan dengan mengganti derajat penurunan dari algoritma SA. Parameter kedua
algoritma SA adalah sebagai berikut : Temperatur awal : 149,99 Jumlah penurunan : 500 Derajat
penurunan : 0,8 Replikasi setiap temperatur : 5 Pergantian derajat penurunan dilakukan untuk mengetahui
penaruh derajat penurunan terhadap hasil yang didapatkan. Parameter HS yang kedua adalah sebagai
berikut: Harmony Memory Size: 40 HMCR dan PAR didapatkan melalui persamaan (6) dan (7) yang
dihitung setiap iterasi (n) Jumlah iterasi (N): 500 Hasil pergantian parameter ketiga menunjukkan bahwa HS
masih mendapatkan hasil yang lebih ba- ik daripada SA. Hal itu ditunjukkan dari 15 kasus, HS memiliki
hasil yang lebih baik dari metode short- est path sebanyak 9 kasus, sedangkan SA sebanyak 8 kasus. SA
menghasilkan solusi yang paling baik pada saat parameter ketiga karena menghasilkan solusi yang lebih
baik daripada metode shortest path paling banyak dibandingkan parameter lainnya. Gambar 9.
Konvergensi simulated annealing Gambar 10. Konvergensi harmony search Tabel 7. Hasil CPU time
parameter ketiga (dalam menit) Jumlah rak Jumlah order SA 10 25 63,8167 50 69,15 100 70,2167 20 25
63,8 50 67,0667 100 72,95 30 25 63,4 50 94,9667 100 105,8 40 25 63,1667 50 112,5 100 127,4 50 25
95,6833 50 19,283 100 165 Rata - rata 83,61332 HS 22,7 32,133 38,6333 18,783 19,9167 20,45 21,2833
25,733 20,9667 34,3667 27,6667 38,5 35,9333 29,5333 41,2667 28,52438 Gambar 9 dan 10 menunjukkan
bahwa SA pada parameter ketiga mampu menghasilkan solusi yang paling konvergen diantara parameter-
parameter sebelumnya. Hal itu ditunjukkan pada saat iterasi ke 235 atau pada saat temperatur $3.15E-21$ SA
sudah menghasilkan solusi yang konvergen. SA mampu menghasilkan solusi yang lebih konvergen
dibandingkan HS. Hal ini berarti SA parameter ketiga mampu menghasilkan solusi dengan varians yang
kecil. Tabel 7 menunjukkan SA memiliki rata-rata waktu pengerjaan selama 83,61332 menit, sedangkan
HS selama 28,52438 menit. Hasil rata-rata ini me- nunjukkan bahwa HS membutuhkan waktu yang lebih
cepat dalam memberikan hasil dibandingkan SA. Simpulan Hasil penelitian menunjukkan algoritma
simulated annealing memberikan rata-rata waktu tempuh yang lebih singkat dibandingkan metode shortest
path dengan rata-rata CPU time sebesar 180,78 menit untuk 8 dari 15 permasalahan. Hasil simulasi juga
menunjukkan algoritma harmony search memberikan rata-rata waktu tempuh yang lebih pendek pada 9
dari 15 permasalahan dibandingkan metode shortest path dengan rata-rata CPU time sebesar 81,106
menit. Algoritma harmony search menghasilkan solusi yang paling baik ketika jumlah iterasi 500 dan 40.
Hasil penelitian juga menunjukkan algoritma har- mony search memberikan solusi yang lebih baik
dibandingkan algoritma simulated annealing dengan CPU time yang lebih rendah, akan tetapi, algoritma
simulated annealing mampu menghasil- kan solusi yang lebih konvergen dibandingkan harmony search.

Daftar Pustaka 1.



Atmaca, E., Ozturk, A., Defining Order Picking Policy: A Storage Assignment Model and a Simulated Annealing Solution in AS/RS Systems, *Applied Mathematical Modelling*, 37, 2012, pp. 5069–5079.

5

2.

Suman, B., and Kumar, P., A Survey of Simulated Annealing as a Tool for Single and Multiobjective Optimization, *Journal of the Operational Research Society*, 57, 2006, pp.1143- 1160.

6

3.

Boysen, N., Briskorn, D., and Emde, S., Sequencing of Picking Orders in Mobile Rack Warehouse, *European Journal of Operational Research*, 259(2017),

1

2016, pp. 293-307. 4.

Yang, X. S., *Nature-Inspired Metaheuristic Algorithms*, United Kingdom: Luniver Press, 2010. 5. Zong, W. G., *Music Inspired Harmony Search Algorithm Theory and Applications*, New York: Springer,

1

2009. 6.

Alia, O. M., Mandava, R., *The Variants of the Harmony Search Algorithm: An Overview*, Springer Science Business Media B.V.,

1

2011. 7.

Yun, H. Y., Jeong, S. J., Kim, K. S., Advanced Harmony Search with Ant Colony Optimization for Solving the Traveling Salesman Problem, *Journal of Applied Mathematics*, 2013,

4

pp. 29-37. 8.

Zong, W. G., Kim, J. H., Loganathan, G. V., *A New Heuristic*

1

Optimization Algorithm: Harmony Search, Simulation Council,

75, 2001, pp. 60–68. 9.

Aulia, I., Nababan, E. B., dan Muchtar, M. A. Penerapan Harmony Search Algorithm dalam Permasalahan Penjadwalan Flow Shop, Jurnal Dunia Teknologi Informasi, 1(1), 2012, pp. 1-

1

7. 10.

Daham, B. F., Mohammed, N. M., and Mohammed, S. K., Modified Harmony Search Algorithm for Solving the Four-Color Mapping Problem, International Journal of Computer Application,

1

91(6), 2014, pp. 975-8887. Octavia et al./ Perbandingan Algoritma Simulated Annealing dan Harmony Search/ JTI,

Vol. 19, No. 2, December 2017, pp.

2

125–132 Octavia et al./ Perbandingan Algoritma Simulated Annealing dan Harmony Search/ JTI,

Vol. 19, No. 2, December 2017, pp.

2

125–132 Octavia et al./ Perbandingan Algoritma Simulated Annealing dan Harmony Search/ JTI,

Vol. 19, No. 2, December 2017, pp.

2

125–132 Octavia et al./ Perbandingan Algoritma Simulated Annealing dan Harmony Search/ JTI,

Vol. 19, No. 2, December 2017, pp.

2

125–132 Octavia et al./ Perbandingan Algoritma Simulated Annealing dan Harmony Search/ JTI,

Vol. 19, No. 2, December 2017, pp.

2

125–132 Octavia et al./ Perbandingan Algoritma Simulated Annealing dan Harmony Search/ JTI,

125–132 125 126 127 128 129 130 131