

BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan proses studi literatur, analisis perangkat lunak, perancangan perangkat lunak, pembangunan perangkat lunak, dan pengujian perangkat lunak, diambil kesimpulan sebagai berikut.

1. Algoritma GWO menyelesaikan MKP dengan membuat kumpulan solusi awal, menentukan tiga solusi terbaik, dan memperkirakan solusi optimal MKP. Kemudian, setiap solusi akan diperbarui berdasarkan perkiraan solusi optimal, dan solusi terbaik baru akan ditentukan jika solusi baru lebih menguntungkan daripada solusi terbaik sebelumnya. Satu solusi terbaik dengan total *profit* tertinggi akan diambil sebagai hasil akhir.
2. Algoritma PECH menggunakan nilai *effective capacity* sebagai nilai efisiensi benda. Nilai *effective profit* merupakan *profit* yang dapat dihasilkan dari nilai *effective capacity*. Benda dengan nilai *effective profit* terbesar akan dimasukkan ke dalam *knapsack*, hingga seluruh benda telah dimasukkan ke dalam *knapsack* atau tidak ada benda lain yang dapat dimasukkan ke dalam sisa kapasitas *knapsack*. Total *profit* dari benda-benda yang dimasukkan ke dalam *knapsack* diambil sebagai hasil akhir.
3. Algoritma GWO dan PECH diimplementasikan ke dalam perangkat lunak. Perangkat lunak memiliki GUI yang dapat menerima *input* pemilihan algoritma dan *input* MKP atau *import dataset*. Perangkat lunak dapat memproses *input* sesuai algoritma yang dipilih, dan menampilkan hasil algoritma pada GUI.
4. Pengujian dilakukan menggunakan GK Dataset, yang berisi 11 buah *dataset* dengan *range* banyaknya benda dan banyaknya dimensi ($n \times m$) dari 100×15 hingga 2500×100 . Pengujian eksperimental mendapatkan hasil nilai parameter optimal untuk algoritma GWO, yaitu batas iterasi maksimal 40 dan ukuran populasi 20. Algoritma GWO dengan parameter optimal memiliki nilai rata-rata pencapaian *profit* tertinggi yang lebih baik sebesar 1.13%, dibandingkan dengan algoritma PECH.

6.2 Saran

Saran penulis untuk penelitian selanjutnya adalah untuk melakukan percobaan terhadap variabel batas iterasi maksimum (G) dan variabel ukuran populasi (s), dengan nilai yang lebih beragam. Hal ini bertujuan untuk menemukan kombinasi nilai kedua variabel yang lebih optimal.

DAFTAR REFERENSI

- [1] Mirjalili, S., Mirjalili, S. M., dan Lewis, A. (2014) Grey wolf optimizer. *Advances in Engineering Software*, **69**, 46–61.
- [2] Tian, X., Ouyang, D., Wang, Y., Zhou, H., Jiang, L., dan Zhang, L. (2023) Combinatorial optimization and local search: A case study of the discount knapsack problem. *Computers and Electrical Engineering*, **105**, 108551.
- [3] Kellerer, H., Pferschy, U., dan Pisinger, D. (2004) Introduction. *Knapsack Problems*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [4] Kellerer, H., Pferschy, U., dan Pisinger, D. (2004) Multidimensional Knapsack Problems. *Knapsack Problems*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [5] Abdel-Basset, M., Abdel-Fatah, L., dan Sangaiah, A. K. (2018) Chapter 10 - metaheuristic algorithms: A comprehensive review. Bagian dari Sangaiah, A. K., Sheng, M., dan Zhang, Z. (ed.), *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications* Intelligent Data-Centric Systems, pp. 185–231. Academic Press.
- [6] Luo, K. dan Zhao, Q. (2019) A binary grey wolf optimizer for the multidimensional knapsack problem. *Applied Soft Computing*, **83**, 105645.
- [7] Akçay, Y., Li, H., dan Xu, S. H. (2007) Greedy algorithm for the general multidimensional knapsack problem. *Annals of Operations Research*, **150**, 17–29.
- [8] Kellerer, H., Pferschy, U., dan Pisinger, D. (2004) Basic Algorithmic Concepts. *Knapsack Problems*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [9] Long, W., Jiao, J., Liang, X., dan Tang, M. (2018) Inspired grey wolf optimizer for solving large-scale function optimization problems. *Applied Mathematical Modelling*, **60**, 112–126.