

SKRIPSI

**PENYELESAIAN PERMASALAHAN *KNAPSACK*
MULTIDIMENSIONAL DENGAN ALGORITMA *GREY WOLF*
OPTIMIZATION DAN ALGORITMA *PRIMAL*
*EFFECTIVE-CAPACITY HEURISTIC***



Wiranata Limantara

NPM: 6181801038

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2023**

UNDERGRADUATE THESIS

**SOLVING MULTIDIMENSIONAL KNAPSACK PROBLEM
WITH GREY WOLF OPTIMIZATION ALGORITHM AND
PRIMAL EFFECTIVE-CAPACITY HEURISTIC ALGORITHM**



Wiranata Limantara

NPM: 6181801038

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2023**

LEMBAR PENGESAHAN

**PENYELESAIAN PERMASALAHAN *KNAPSACK*
MULTIDIMENSIONAL DENGAN ALGORITMA *GREY WOLF*
OPTIMIZATION DAN ALGORITMA *PRIMAL*
*EFFECTIVE-CAPACITY HEURISTIC***

Wiranata Limantara

NPM: 6181801038

Bandung, 5 Juli 2023

Menyetujui,

Pembimbing

**Digitally signed
by Husnul
Hakim**

Husnul Hakim, M.T.

Ketua Tim Penguji

**Digitally signed
by Keenan
Adiwijaya Leman**

Keenan Adiwijaya Leman, S.T.

Anggota Tim Penguji

**Digitally signed
by Luciana
Abednego**

Luciana Abednego, M.T.

Mengetahui,

Ketua Program Studi

**Digitally signed
by Mariskha Tri
Adithia**

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

**PENYELESAIAN PERMASALAHAN *KNAPSACK MULTIDIMENSIONAL*
DENGAN ALGORITMA *GREY WOLF OPTIMIZATION* DAN
ALGORITMA *PRIMAL EFFECTIVE-CAPACITY HEURISTIC***

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 5 Juli 2023



Wiranata Limantara
NPM: 6181801038

ABSTRAK

Multidimensional Knapsack Problem adalah salah satu variasi *Knapsack Problem*. *Multidimensional Knapsack Problem* merupakan permasalahan optimasi pemilihan benda yang dapat dimasukkan ke dalam sebuah penampung yang memiliki batasan daya tampung. Benda yang tersedia memiliki *weight* lebih dari satu dimensi yang menggunakan kapasitas penampung, dan menghasilkan sejumlah *profit*. Pilihan benda-benda yang dimasukkan ke dalam *knapsack* harus menghasilkan total *profit* tertinggi dari seluruh kemungkinan pilihan yang memungkinkan, serta tidak melebihi batasan daya tampung *knapsack*.

Multidimensional Knapsack Problem dapat diselesaikan dengan beberapa jenis algoritma, antara lain *meta-heuristic*, *greedy-type heuristic*, dan lain-lain. Pada skripsi ini, digunakan satu algoritma *meta-heuristic*, yaitu algoritma *Grey Wolf Optimization*, dan satu algoritma *greedy-type heuristic*, yaitu algoritma *Primal Effective-Capacity Heuristic*.

Algoritma *Grey Wolf Optimization* menirukan perilaku berburu yang dilakukan oleh kawanan serigala, dan mengimplementasi hierarki kepemimpinan dalam kawanan serigala. Serigala menyimbolkan sebuah solusi dalam *Multidimensional Knapsack Problem*. Terdapat tiga serigala pemimpin yang dianggap sebagai serigala yang memiliki pengetahuan paling baik terhadap keberadaan mangsa. Tiga serigala pemimpin menyimbolkan tiga solusi *Multidimensional Knapsack Problem* dengan total *profit* tertinggi. Secara iteratif, setiap solusi dalam kumpulan solusi akan diperbarui sehingga mendekati perkiraan solusi optimal yang ditentukan oleh solusi-solusi terbaik. Solusi yang memiliki total *profit* yang lebih tinggi, dibandingkan salah satu solusi terbaik, akan ditetapkan sebagai solusi terbaik baru, menggantikan solusi terbaik sebelumnya. Akhirnya, satu solusi dengan total *profit* yang paling tinggi akan diambil sebagai hasil akhir.

Algoritma *Primal Effective-Capacity Heuristic* merupakan algoritma yang bertipe *greedy*, dengan menggunakan nilai *effective capacity* benda sebagai nilai efisiensi benda. Setiap benda akan menghasilkan *effective profit* berdasarkan nilai *effective capacity* yang dimilikinya. Secara iteratif, benda dengan nilai *effective profit* terbesar akan dimasukkan ke dalam penampung, mengurangi sisa kapasitas yang tersedia, dan menambahkan *profit* ke dalam solusi. Proses berakhir ketika seluruh benda telah dimasukkan ke dalam penampung, atau tidak ada benda yang dapat dimasukkan ke dalam sisa kapasitas penampung.

Skripsi ini bertujuan untuk mengimplementasikan algoritma *Grey Wolf Optimization* dan algoritma *Primal Effective-Capacity Heuristic* untuk menyelesaikan *Multidimensional Knapsack Problem*, serta melakukan pengukuran kinerja kedua algoritma dalam menyelesaikan *Multidimensional Knapsack Problem*. Untuk mencapai tujuan tersebut, dibangun perangkat lunak yang mengimplementasikan kedua algoritma tersebut. Perangkat lunak yang dibangun dapat menerima pilihan algoritma, *input Multidimensional Knapsack Problem* atau *import dataset*, memproses *input* sesuai algoritma yang dipilih, dan menampilkan *output* algoritma.

Hasil implementasi dan pengujian kedua algoritma pada perangkat lunak menunjukkan bahwa algoritma *Grey Wolf Optimization* dengan parameter optimal, memiliki nilai rata-rata pencapaian *profit* tertinggi sebesar 99.37%, lebih baik sebesar 1.13% dibandingkan dengan algoritma *Primal Effective-Capacity Heuristic* yang memiliki nilai rata-rata pencapaian *profit* tertinggi sebesar 98.24%.

Kata-kata kunci: Permasalahan *Knapsack Multidimensional*, Algoritma *Grey Wolf Optimization*, Algoritma *Primal Effective-Capacity Heuristic*, Algoritma *Meta-Heuristic*, Algoritma *Greedy-Type Heuristic*

ABSTRACT

Multidimensional Knapsack Problem is one of the variants of Knapsack Problem. Multidimensional Knapsack Problem is an optimization problem of objects selection that can be put inside a knapsack with capacity boundaries. The given objects has more than one dimension of weight values that consume knapsack's capacity, and yield a profit value. The object selection must yield maximum total of profit of any other selection possibilities, and not exceeding the capacity boundaries.

Multidimensional Knapsack Problem can be solved with several kinds of algorithms, such as meta-heuristic algorithms, greedy-type heuristic algorithms, etc. In this thesis, two algorithms will be used. One meta-heuristic algorithm, which is the Grey Wolf Optimization Algorithm, and one greedy-type heuristic algorithm, which is Primal Effective-Capacity Heuristic Algorithm.

Grey Wolf Optimization Algorithm mimics the hunting behavior in a grey wolf pack, and implements the leadership hierarchy of a grey wolf pack. Wolf represents a solution of Multidimensional Knapsack Problem. There are three levels of the leadership hierarchy, that considered has the best knowledge of the location of the prey. These three leader wolves represent three best solutions of the Multidimensional Knapsack Problem, with the highest total of profit. Iteratively, each of the solution in set is updated towards the estimated optimal solution of Multidimensional Knapsack Problem, which determined by the best solutions. A solution with higher total of profit than one of the best solutions, will determined as the new best solution, replacing the previous best solution. Eventually, one best solution with the highest total of profit will be taken as final result.

Primal Effective-Capacity Heuristic Algorithm is a greedy-typed algorithm, using effective capacity value as the efficiency value of an object. Each object yields an effective profit value based on its effective capacity value. Iteratively, an object with highest effective profit value will be put into the knapsack, consuming the capacity constraints, and add its profit value to the total of profit. This object additions will be done until all object is put inside knapsack, or no objects that fit inside the remaining capacity.

This thesis aims to implement the Grey Wolf Optimization Algorithm and Primal Effective-Capacity Heuristic Algorithm to solve the Multidimensional Knapsack Problem, and do performance measurements of both algorithms in solving the Multidimensional Knapsack Problem. To achieve that goal, a software is built, with the implement of both algorithms. The software accepts an algorithm selection, the Multidimensional Knapsack Problem input or a dataset import, process the inputs with selected algorithm, and display an output of the algorithm.

Implementation and experiment result of the both implemented algorithms shows that the Grey Wolf Optimization Algorithm with optimal parameter, yields an average of highest profit attained with 99.37%, which is 1.13% higher than the Primal Effective Capacity Heuristic Algorithm that yields an average of highest profit attained with 98.24%.

Keywords: Multidimensional Knapsack Problem, Grey Wolf Optimization Algorithm, Primal Effective-Capacity Heuristic Algorithm, Meta-Heuristic Algorithm, Greedy-Type Heuristic

KATA PENGANTAR

Sotthi hotu. Puji dan syukur kepada Tuhan yang Maha Esa dan Sang Tiratana, penulis dapat menyelesaikan skripsi yang berjudul “Penyelesaian Permasalahan *Knapsack Multidimensional* dengan Algoritma *Grey Wolf Optimization* dan Algoritma *Primal Effective-Capacity Heuristic*”. Skripsi ini disusun untuk memenuhi syarat kelulusan di program studi Informatika, di Universitas Katolik Parahyangan.

Penulis menyadari bahwa penyusunan skripsi ini merupakan proses yang panjang dan tidak mudah. Sehingga, tuntasnya skripsi ini merupakan berkat dukungan dan doa restu dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada seluruh pihak yang telah membantu dan mendukung penulis selama proses pengerjaan skripsi ini, dan secara khusus kepada:

- Papa Antony Limantara, Mama Soo Jong Sioe, dan Adik Wijayanata Limantara yang telah memberikan doa dan dukungan, serta toleransi perpanjangan waktu selama proses kuliah dan penyusunan skripsi.
- Dosen Pembimbing, Bapak Husnul Hakim, M.T., yang telah memberikan waktu dan tenaga untuk membantu dan membimbing penulis dalam proses penyusunan skripsi.
- Dosen Penguji, Bapak Keenan Adiwijaya Leman, S.T. dan Ibu Luciana Abednego, M.T., yang telah memberikan saran, kritik, serta masukan yang membangun untuk membuat skripsi ini menjadi lebih baik.
- Teman-teman, Josie Esthaliani, Ame Fedora I. G., Juan Anthonius, dan Rama Fauzi S., yang telah memberikan bantuan, tips, dan trik selama proses penyusunan skripsi.
- Teman-teman, Geraldi Akira Surya, Lucyus Matthew A., Edward Tjahyadi, Shannas Rizqi R., dan Brigitte Nikita R., yang telah bersama-sama berjuang menyusun skripsi masing-masing, dan saling bertukar informasi.
- Teman Kharisma Surya, yang menjadi teman bersama kopi Fore dan Starbucks, yang telah membantu penulis selama proses penyusunan skripsi.
- Seluruh teman dan rekan di Universitas Katolik Parahyangan yang tidak dapat disebutkan satu per satu, yang telah membantu penulis selama berkuliah di Universitas Katolik Parahyangan.

Penulis menyadari bahwa masih terdapat banyak kekurangan pada skripsi ini. Maka, penulis secara terbuka menerima kritik dan saran, serta masukan dari pembaca yang dapat menyempurnakan skripsi ini. Semoga skripsi ini dapat bermanfaat bagi pembaca dan menginspirasi untuk penelitian-penelitian selanjutnya.

Bandung, Juli 2023

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Pembahasan	4
2 LANDASAN TEORI	5
2.1 <i>Knapsack Problem</i>	5
2.2 <i>Multidimensional Knapsack Problem</i>	5
2.3 Algoritma <i>Greedy</i>	6
2.3.1 Algoritma <i>Greedy</i> untuk Menyelesaikan KP	6
2.3.2 Algoritma <i>Greedy</i> untuk Menyelesaikan MKP	7
2.4 Algoritma <i>Grey Wolf Optimization</i>	7
2.5 Algoritma <i>Primal Effective Capacity Heuristic</i>	10
2.6 GK Dataset	10
3 ANALISIS	13
3.1 Deskripsi Masalah	13
3.2 Studi Kasus dengan Algoritma GWO	14
3.2.1 Penghitungan Nilai <i>Pseudo-Utility</i> Benda	14
3.2.2 Pembuatan Populasi Awal	15
3.2.3 Iterasi Pencarian Solusi Optimal Pertama ($t=1$)	16
3.3 Studi Kasus dengan Algoritma PECH	23
3.3.1 Inisialisasi Variabel	23
3.3.2 Iterasi Pertama	23
3.3.3 Iterasi Kedua	25
3.3.4 Iterasi Ketiga	25
3.4 Diagram <i>Flowchart</i> Algoritma GWO	26
3.5 Diagram <i>Flowchart</i> Algoritma PECH	28
3.6 Diagram Aktivitas	29
3.7 Diagram Kelas Sederhana	30
4 PERANCANGAN PERANGKAT LUNAK	33
4.1 Diagram Kelas Lengkap	33
4.2 Penjelasan <i>Input</i> dan <i>Output</i>	41

4.2.1	<i>Input</i> Utama MKP	41
4.2.2	<i>Input</i> Tambahan untuk Algoritma GWO	41
4.2.3	<i>Output</i> Perangkat Lunak	42
4.3	Rancangan <i>Graphical User Interface</i> (GUI)	42
4.3.1	Rancangan Tampilan Halaman Utama	42
4.3.2	Rancangan Tampilan Halaman <i>Error Dialog</i>	43
5	IMPLEMENTASI PERANGKAT LUNAK DAN PENGUJIAN	45
5.1	Hasil GUI	45
5.2	Pengujian Fungsional	45
5.2.1	Penanganan Penggunaan Normal	45
5.2.2	Penanganan Kesalahan <i>Input</i>	46
5.3	Pengujian Eksperimental	48
5.3.1	Percobaan Perubahan Nilai Variabel G	49
5.3.2	Percobaan Perubahan Nilai Variabel s	50
5.3.3	Perbandingan Pencapaian <i>Profit</i> Tertinggi Algoritma GWO dan PECH	50
6	KESIMPULAN DAN SARAN	53
6.1	Kesimpulan	53
6.2	Saran	53
	DAFTAR REFERENSI	55
	A KODE PROGRAM	57
	B HASIL EKSPERIMEN	77

DAFTAR GAMBAR

1.1	Contoh <i>knapsack problem</i>	1
1.2	Contoh <i>multidimensional knapsack problem</i>	1
1.3	Hierarki Kepemimpinan Serigala Abu-abu [1]	2
2.1	Contoh GK <i>Datasets</i>	11
3.1	<i>Flowchart</i> umum perangkat lunak	13
3.2	Diagram <i>flowchart</i> algoritma GWO	27
3.3	<i>Flowchart</i> algoritma PECH	28
3.4	Diagram aktivitas pengguna terhadap perangkat lunak	30
3.5	Rancangan diagram kelas sederhana	31
4.1	Diagram kelas perangkat lunak	34
4.2	Rancangan GUI utama perangkat lunak	43
4.3	Rancangan GUI <i>error dialog</i> perangkat lunak	43
5.1	GUI perangkat lunak yang dihasilkan	46
5.2	Contoh tampilan GUI ketika penggunaan normal	47
5.3	Jendela <i>error-dialog</i> jika terdapat <i>input</i> kosong	47
5.4	Jendela <i>error-dialog</i> jika terdapat <i>input</i> yang tidak valid	48
5.5	Jendela <i>error-dialog</i> jika terdapat <i>input</i> yang tidak lengkap	48

BAB 1

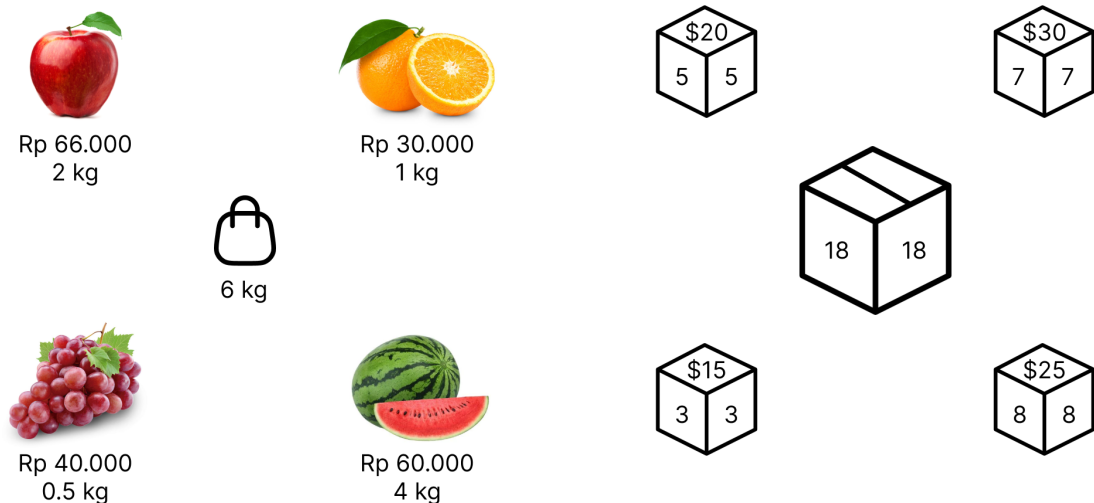
PENDAHULUAN

1.1 Latar Belakang

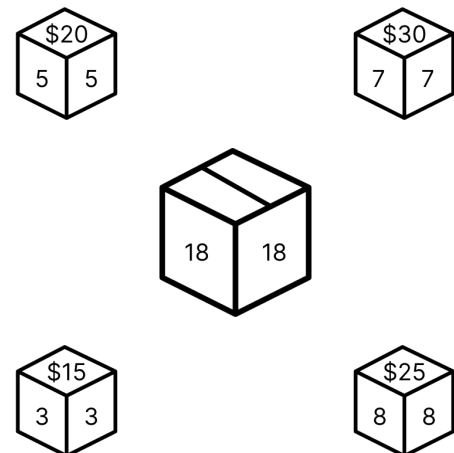
Combinatorial Optimization Problems merupakan masalah pencarian sebuah objek optimal dari himpunan terbatas objek-objek [2]. *Knapsack Problem* (KP) merupakan salah satu permasalahan yang tergolong dalam *Combinatorial Optimization Problems*. Beberapa permasalahan lain yang termasuk dalam *Combinatorial Optimization Problems*, antara lain *Travelling Salesman Problem*, *Minimum Spanning Tree Problem*, dan lain-lain. KP adalah sebuah permasalahan optimasi pemilihan benda-benda yang dapat dimasukkan ke dalam penampung yang memiliki keterbatasan ruang atau daya tampung [3]. Diberikan himpunan benda yang masing-masing memiliki berat (*weight*) dan nilai (*profit*). Sebuah himpunan bagian dari benda-benda tersebut harus ditentukan untuk memaksimalkan nilai yang didapatkan, namun jumlah berat tidak melebihi kapasitas penampung (*capacity*) yang ditentukan.

Secara umum, terdapat dua jenis KP, yaitu 0-1 *Knapsack Problem* dan *Partial Knapsack Problem*. Pada 0-1 *Knapsack Problem*, satu benda hanya dapat dimasukkan ke dalam *knapsack* secara utuh atau tidak dimasukkan sama sekali. Sedangkan, pada *Partial Knapsack Problem*, satu benda dapat dimasukkan ke dalam *knapsack* dengan besaran bagian tertentu, tidak harus utuh atau seluruhnya.

Salah satu contoh dari KP dapat dilihat pada Gambar 1.1. Terdapat 4 jenis buah yang ingin dijual dan dibawa menggunakan sebuah tas. Masing-masing jenis buah memiliki harga jual dan berat. Tas sebagai penampung memiliki kapasitas berat 6 kilogram. Pemilihan jenis buah yang dimasukkan ke dalam tas harus menghasilkan total harga jual tertinggi, dengan jumlah berat buah tidak melebihi kapasitas tas tersebut.



Gambar 1.1: Contoh *knapsack problem*



Gambar 1.2: Contoh *multidimensional knapsack problem*

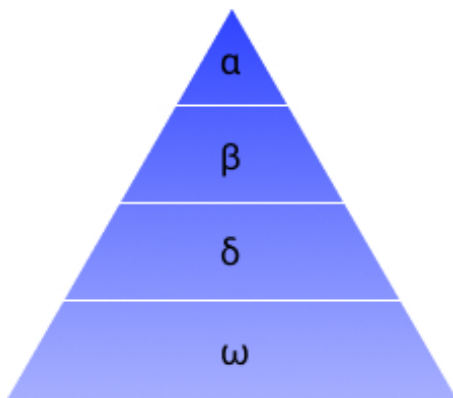
Multidimensional Knapsack Problem (MKP) adalah variasi dari *Knapsack Problem*. Pada MKP, benda di dalam himpunan dapat memiliki lebih dari satu dimensi, misalnya berat dan ukuran benda [4]. Sama halnya dengan tempat untuk memuat benda-benda tersebut, yang dapat memiliki lebih dari satu dimensi kapasitas penampung, bersesuaian dengan jumlah dan jenis dimensi yang dimiliki oleh benda yang diberikan. Dalam kehidupan sehari-hari, MKP dapat diterapkan pada proses pemuatan kargo, pemuatan kontainer, alokasi sumber daya, pemilihan proyek, dan lain-lain.

Gambar 1.2 merupakan salah satu contoh sederhana MKP. Terdapat 4 buah kotak kecil yang ingin dimasukkan ke dalam sebuah kotak besar. Masing-masing kotak memiliki harga barang yang terdapat di dalamnya, dan dua dimensi *weight* yaitu ukuran panjang dan lebar kotak tersebut. Kotak besar sebagai penampung memiliki ukuran panjang dan lebar sebesar 18 satuan. Pemilihan kotak yang dimasukkan ke dalam kotak besar harus menghasilkan total harga tertinggi, dengan jumlah panjang dan lebar tidak melebihi ukuran kotak besar.

MKP memiliki tingkat kompleksitas komputasi yang tinggi (*NP-hard problem*). Beberapa algoritma eksak dapat digunakan untuk menemukan secara pasti solusi optimal MKP, antara lain algoritma *Dynamic Programming*. Namun, pencarian solusi optimal ini memiliki kompleksitas waktu yang tinggi. Contohnya, algoritma *Dynamic Programming* memiliki kompleksitas waktu $O(nc_{max}^m)$ [4]. Variabel n adalah banyaknya benda, c_{max} adalah nilai kapasitas *knapsack* terbesar, dan m adalah banyaknya dimensi *weight* benda. Sehingga, beberapa algoritma *heuristic* dan *meta-heuristic* dikembangkan untuk menyelesaikan MKP dengan solusi yang cukup baik, dalam waktu yang wajar [4].

Algoritma *heuristic* dan *meta-heuristic* menggunakan metode perkiraan/penaksiran untuk mendapatkan hasil yang cukup baik, dalam waktu yang wajar. Algoritma *heuristic* merupakan algoritma yang dapat diterapkan secara efisien terhadap sebuah permasalahan yang spesifik, namun algoritma yang sama tidak dapat diterapkan terhadap permasalahan lainnya (*problem-dependant*) [5]. Sedangkan, algoritma *meta-heuristic* merupakan algoritma yang memberikan kerangka kerja yang umum, dan dapat diterapkan terhadap hampir semua permasalahan optimasi [5].

Pada skripsi ini, diterapkan algoritma *meta-heuristic*, yaitu *Grey Wolf Optimization* (GWO) dan algoritma *heuristic* yang bertipe *greedy*, *Primal Effective-Capacity Heuristic* untuk membantu penyelesaian MKP mendekati solusi yang optimal.



Gambar 1.3: Hierarki Kepemimpinan Serigala Abu-abu [1]

Algoritma GWO termasuk dalam *bio-inspired algorithm*, yang meniru hierarki kepemimpinan dan mekanisme berburu secara berkelompok yang dilakukan serigala abu-abu (*grey wolf*) di alam bebas. Hierarki kepemimpinan tersebut meliputi serigala *alpha*, *beta*, *delta*, dan *omega* [6].

Serigala *alpha* adalah serigala-serigala pemimpin, terdiri dari satu jantan dan satu betina. Para *alpha* bertanggung jawab untuk mengambil keputusan tentang berburu, tempat istirahat, waktu untuk bangun tidur, dan lain-lain. Keputusan ini wajib diikuti oleh seluruh serigala dalam kawanan tersebut. Serigala *beta* menempati tingkat kedua dalam hierarki ini. Para *beta* bertindak sebagai pembantu serigala *alpha* dalam mengambil keputusan dan aktivitas kawanan lainnya. Serigala *beta* tunduk kepada *alpha*, namun juga bertindak sebagai pendisiplin bagi serigala tingkat di bawahnya. Serigala *delta* merupakan bawahan dari *alpha* dan *beta*. Sisa serigala lainnya dalam kawanan merupakan serigala-serigala dengan tingkatan paling rendah, dan disebut serigala *omega*.

Dalam korelasinya dengan penyelesaian MKP, serigala *alpha*, *beta*, dan *delta* merepresentasikan tiga solusi terbaik dari suatu MKP [1]. Sedangkan, *omega* adalah seluruh solusi lainnya dalam ruang solusi. Ruang solusi adalah seluruh himpunan bagian yang berisi kombinasi-kombinasi benda-benda yang dimasukkan, beserta jumlah dimensi dan *profit* yang didapatkan.

Algoritma *Primal Effective-Capacity Heuristic* merupakan algoritma yang bertipe *greedy*, dengan

menggunakan nilai *effective capacity* benda sebagai nilai efisiensi benda [7]. Setiap benda akan menghasilkan *effective profit* berdasarkan nilai *effective capacity* yang dimilikinya. Secara berulang, benda dengan nilai *effective profit* tertinggi akan dimasukkan ke dalam *knapsack*, karena dianggap sebagai benda yang paling menguntungkan untuk dimasukkan ke dalam *knapsack*, pada iterasi tersebut. Benda yang dimasukkan ke dalam *knapsack* akan mengurangi kapasitas *knapsack* sebanyak *weight* yang dimilikinya, dan menambahkan perolehan *profit* sementara. Seluruh proses berakhir ketika seluruh benda telah dimasukkan ke dalam penampung, atau tidak ada benda yang dapat dimasukkan ke dalam sisa kapasitas penampung.

Hasil akhir dari skripsi ini merupakan sebuah perangkat lunak yang dapat menyelesaikan MKP. Perangkat lunak yang dibangun mengimplementasikan algoritma GWO dan PECH untuk menyelesaikan MKP. Kinerja kedua algoritma dibandingkan untuk menentukan algoritma yang dapat mencapai solusi yang lebih baik untuk sebuah MKP. Pengukuran kinerja perangkat lunak dan algoritma dilakukan dengan pengujian fungsional dan eksperimental menggunakan GK Dataset. GK Dataset menyediakan 11 *dataset* dengan variasi banyaknya benda dan banyaknya dimensi ($n \times m$) dari 100×15 hingga 2500×100 .

1.2 Rumusan Masalah

Rumusan masalah yang dibahas pada skripsi ini adalah sebagai berikut.

1. Bagaimana cara kerja algoritma GWO dan PECH untuk menyelesaikan MKP?
2. Bagaimana cara mengimplementasikan algoritma GWO dan PECH untuk menyelesaikan MKP?
3. Bagaimana kinerja algoritma GWO dan PECH untuk menyelesaikan MKP?

1.3 Tujuan

Tujuan yang ingin dicapai dari skripsi ini adalah sebagai berikut.

1. Mempelajari cara kerja algoritma GWO dan PECH untuk menyelesaikan MKP.
2. Mengimplementasikan algoritma GWO dan PECH pada perangkat lunak untuk menyelesaikan MKP.
3. Melakukan pengukuran kinerja algoritma GWO dan PECH pada perangkat lunak yang dihasilkan untuk menyelesaikan MKP.

1.4 Batasan Masalah

Batasan dari skripsi ini adalah permasalahan *Multidimensional Knapsack Problem* yang diselesaikan adalah 0-1 *Multidimensional Knapsack Problem*.

1.5 Metodologi

Metode penelitian yang dilakukan adalah sebagai berikut.

1. Melakukan studi literatur mengenai konsep dan penyelesaian KP.
2. Melakukan studi literatur mengenai konsep dan penyelesaian MKP.
3. Melakukan studi literatur mengenai algoritma GWO untuk menyelesaikan MKP.
4. Melakukan studi literatur mengenai algoritma *Primal Effective-Capacity Heuristic* untuk menyelesaikan MKP.
5. Melakukan studi literatur mengenai pemanfaatan GK Dataset untuk melakukan pengujian perangkat lunak.

6. Melakukan analisis, meliputi analisis deskripsi masalah, analisis studi kasus MKP pada algoritma GWO dan PECH, pembuatan diagram *flowchart* untuk algoritma GWO dan PECH, pembuatan diagram aktivitas perangkat lunak, dan pembuatan diagram kelas sederhana.
7. Melakukan perancangan perangkat lunak, meliputi pembuatan diagram kelas lengkap, menjelaskan *input* dan *output* perangkat lunak, dan perancangan *Graphical User Interface* (GUI).
8. Melakukan implementasi algoritma GWO dan PECH pada perangkat lunak.
9. Melakukan eksperimen dan pengujian perangkat lunak menggunakan GK Dataset.
10. Mengambil kesimpulan hasil eksperimen dan pengujian.
11. Menulis dokumen skripsi.

1.6 Sistematika Pembahasan

Sistematika pembahasan pada skripsi ini adalah sebagai berikut.

- Bab 1 membahas tentang latar belakang masalah, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika pembahasan.
- Bab 2 membahas tentang teori *Knapsack Problem*, *Multidimensional Knapsack Problem*, algoritma-algoritma yang dapat digunakan untuk menyelesaikannya, serta teori tentang GK Dataset yang digunakan untuk pengujian perangkat lunak.
- Bab 3 membahas tentang beberapa analisis perangkat lunak yang dilakukan, yaitu deskripsi masalah, studi kasus untuk algoritma GWO dan PECH, diagram *flowchart* untuk kedua algoritma, diagram aktivitas perangkat lunak, dan diagram kelas sederhana.
- Bab 4 membahas beberapa proses perancangan perangkat lunak, yaitu pembuatan diagram kelas lengkap, penjelasan *input* dan *output* perangkat lunak, dan rancangan GUI.
- Bab 5 membahas implementasi perangkat lunak berupa hasil GUI, serta pengujian fungsional dan eksperimental terhadap perangkat lunak.
- Bab 6 berisi kesimpulan skripsi dan saran untuk penelitian selanjutnya.