

**PENERAPAN *WHALE OPTIMIZATION ALGORITHM*
(WOA) TERHADAP *KNAPSACK PROBLEM***

SKRIPSI

Diajukan untuk memenuhi salah satu syarat guna mencapai gelar
Sarjana dalam bidang ilmu Teknik Industri

Disusun oleh:

Nama : Jane Walisi

NPM : 2015610175



**PROGRAM STUDI SARJANA TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS KATOLIK PARAHYANGAN
BANDUNG
2019**

**FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS KATOLIK PARAHYANGAN
BANDUNG**



Nama : Jane Walisi
NPM : 2015610175
Program Studi : Teknik Industri
Judul Skripsi : PENERAPAN *WHALE OPTIMIZATION ALGORITHM*
(WOA) TERHADAP *KNAPSACK PROBLEM*

TANDA PERSETUJUAN SKRIPSI

Bandung, Januari 2019

**Ketua Program Studi Sarjana
Teknik Industri**

(Romy Loice, S.T., M.T.)

Pembimbing Pertama

(Hanky Fransisotis, S.T., M.T.)

Pembimbing Kedua

(Cynthia P. Juwono, Ir., M.S.)

ABSTRAK

Secara sederhana *knapsack problem* merupakan suatu permasalahan dalam menentukan barang yang hendak dibawa di dalam tas agar memberikan keuntungan maksimal bila kapasitas tas terbatas. Dalam kehidupan sehari-hari, masalah *knapsack* dapat berupa penentuan barang yang hendak dibawa di dalam sebuah tas sekolah dengan batasan berat maksimum muatan yang dapat ditampung tas.

Whale Optimization Algorithm (WOA) merupakan algoritma metaheuristik kelas *swarm based* baru yang terinspirasi dari metode paus *humpback* berburu ikan yang bernama *bubble-net feeding*. *Bubble net feeding* ini merupakan perilaku paus melingkari mangsanya menuju permukaan air menyerupai bentuk spiral logaritmik. WOA memiliki 4 buah parameter yaitu jumlah agen pencari (n), jumlah iterasi maksimum (t), batas atas a (ba_a), dan konstanta b ($b_constant$). Parameter ba_a digunakan untuk menyeimbangkan antara gerakan eksplorasi dan eksploitasi. Sedangkan parameter $b_constant$ merupakan penentu bentuk spiral logaritmik. 9 buah kombinasi parameter WOA digunakan untuk menyelesaikan 9 permasalahan *knapsack*.

Solusi permasalahan *knapsack* yang dikerjakan dengan WOA dibandingkan dengan algoritma *Viral System* (VS) dan *Elephant Herding Optimization* (EHO) untuk kasus pertama sampai ketiga. Selain itu dibandingkan juga dengan algoritma *Harmony Search* (HS), *Artificial Bee Colony* (ABC), dan *Soccer League Competition* (SLC) untuk kasus keempat sampai kesembilan. WOA dapat memberikan solusi yang lebih baik dibandingkan algoritma VS, EHO, HS, dan ABC. Namun solusi permasalahan *knapsack* menggunakan algoritma SLC masih lebih unggul dibandingkan WOA. Selain itu, berdasarkan uji ANOVA, parameter ba_a memiliki pengaruh terhadap nilai solusi.

ABSTRACT

A knapsack problem decides which items should a user take in a capacity-constrained bag to give maximum returns. In everyday life, knapsack problem can be the determination of the item to be carried in a school bag with the maximum weight limit of the bag as the constraint.

The Whale Optimization Algorithm (WOA) is a new swarm-based class metaheuristic algorithm inspired mainly by how humpback whales behave when they prey on fish. This behavior is called bubble-net feeding. In bubble-net feeding, whales circle their prey towards the surface resembling a logarithmic spiral. A WOA has four parameters, which are the number of search agents (n), maximum iteration count (t), an upper limit a (ba_a) and a $b_constant$. The ba_a parameter is used to balance between exploration and exploitation movements, while the $b_constant$ parameter is used to define the shape of the logarithmic spiral. This research used 9 combinations of WOA parameters to solve 9 knapsack problems.

Solutions to the first until the third knapsack problem solved using the WOA are compared to solutions from Viral System (VS) and Elephant Herding Optimization (EHO) algorithms. In addition, it is also compared with Harmony Search (HS), Artificial Bee Colony (ABC), and Soccer League Competition (SLC) algorithms for the fourth until the ninth knapsack problem. Results from the WOA yielded better solutions compared to solutions from the VS, EHO, HS, and ABC algorithms. However, solutions from the SLC algorithm still yielded better results than those from the WOA. Incidentally, ANOVA tests showed that the ba_a parameter influenced the solutions.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi yang berjudul “Penerapan Whale Optimization Algorithm (WOA) terhadap *Knapsack Problem*” dengan tepat waktu. Skripsi ini disusun sebagai salah satu syarat kelulusan untuk mendapatkan gelar sarjana.

Penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah membantu selama proses pembuatan skripsi ini dari awal hingga akhir. Adapun pihak-pihak tersebut yaitu.

1. Ibu Cynthia P. Juwono, Ir., M.S. dan Bapak Hanky Fransiscus, S.T., M.T. selaku dosen pembimbing skripsi. Terima kasih atas bimbingan, ilmu, arahan, maupun saran yang diberikan sehingga skripsi ini dapat disusun dengan baik dan diselesaikan dengan tepat waktu.
2. Bapak Y. M. Kinley Aritonang, Ph.D. dan Bapak Ignatius A. Sandy, S.Si., M.T. selaku dosen penguji sidang proposal. Terima kasih atas ilmu serta saran yang diberikan sehingga skripsi ini dapat disusun dengan baik.
3. Bapak Dr. Sugih Sudharma Tjandra, S.T., M.Si. dan Ibu Titi Iswari, S.T., M.Sc., M.B.A. selaku dosen penguji sidang skripsi. Terima kasih atas ilmu serta saran yang diberikan sehingga skripsi ini dapat disusun dengan baik.
4. Orangtua, saudara-saudara, dan keponakan yang selalu mendoakan, mendukung, dan memberikan motivasi kepada penulis dalam menyelesaikan skripsi ini.
5. Kadima Lukas yang selalu mendoakan, mendukung, memberi masukan dan motivasi kepada penulis dalam menyelesaikan skripsi ini.
6. Demasdika, Kenny S., Alex, Januar, Joice, dan Verrel sesama pejuang skripsi algoritma yang ingin lulus 3,5 tahun yang telah memberikan masukan dan dukungan kepada penulis selama menyelesaikan skripsi ini.
7. Ko Ivan, ko Rainer, dan kak Deva yang telah memberikan masukan dan dukungan kepada penulis.

8. Thoe, Tamara Gita Pramana Putri selaku teman baik penulis yang selalu mendoakan serta memberikan dukungan, motivasi, dan setia menemani penulis dalam menyelesaikan skripsi ini.
9. Natalia dan keluarga yang selalu memberikan dukungan dan motivasi kepada penulis dalam menyelesaikan skripsi ini.
10. Theo, Farrell, Wiri, dan Karla yang selalu memberikan dukungan dan motivasi kepada penulis dalam menyelesaikan skripsi ini.
11. Ighes, Audry, Velyn, Christina, Raka, Marcella, ko Chandra, ko Alvian, ci Steffi, dan Luisa yang selalu mendoakan serta memberikan dukungan dan motivasi kepada penulis dalam menyelesaikan skripsi ini.
12. Seluruh teman TI kelas B angkatan 2015.
13. Seluruh dosen TI UNPAR.
14. Seluruh staf dan pekaya FTI UNPAR gedung 8 dan gedung 10.

Penulis berharap agar skripsi ini dapat berguna dan bermanfaat bagi pembaca. Penulis sadar bahwa masih terdapat banyak kekurangan dalam penyusunan skripsi ini. Maka dari itu, penulis sangat menghargai kritik dan saran yang membangun dari pembaca. Akhir kata, Penulis mengucapkan terima kasih.

Bandung, 16 Januari 2019

Jane Walisi

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	I-1
I.1 Latar Belakang Masalah	I-1
I.2 Identifikasi dan Rumusan Masalah	I-3
I.3 Pembatasan Masalah	I-6
I.4 Tujuan Penelitian	I-6
I.5 Manfaat Penelitian	I-6
I.6 Metodologi Penelitian	I-7
I.7 Sistematika Penulisan	I-9
BAB II TINJAUAN PUSTAKA	II-1
II.1 <i>Encoding</i> dan <i>Decoding</i>	II-1
II.2 Pembaruan Posisi Paus	II-4
II.3 Algoritma <i>Whale Optimization</i>	II-3
II.4 Fungsi Transfer	II-9
II.5 Desain Eksperimen	II-10
BAB III PERANCANGAN ALGORITMA	III-1
III.1 Perhitungan Harga Pokok Penjualan (HPP)	III-1
III.2 Proyeksi Laba Rugi	III-4
II.2.1 Gerakan Mencari Mangsa.....	III-5
II.2.2 Gerakan Melingkari Mangsa	III-8
II.2.3 Gerakan Menyerang Mangsa.....	III-11
III.3 Perbaikan Posisi Paus	III-13
III.4 Rancangan Algoritma WOA terhadap Permasalahan <i>Knapsack</i>	III-14

III.4.1 Notasi Algoritma	III-15
III.4.2 Algoritma Utama.....	III-17
III.4.3 Algoritma Pembuatan Matriks Posisi Awal	III-20
III.4.4 Algoritma Menyerang Mangsa.....	III-23
III.4.5 Algoritma Mencari Mangsa.....	III-24
III.4.6 Algoritma Melingkari Mangsa	III-25
III.4.7 Algoritma Perbaikan Posisi Paus	III-26
III.4.8 Algoritma Fungsi Transfer Matriks Posisi	III-27
III.5 Validasi Algoritma.....	III-31
BAB IV IMPLEMENTASI ALGORITMA	IV-1
IV.1 Verifikasi Program	IV-1
IV.2 Validasi Program	IV-7
IV.3 Penentuan Parameter WOA	IV-12
IV.4 Implementasi <i>Whale Optimization Algorithm</i> pada Kasus	
<i>Knapsack</i>	IV-16
IV.4.1 Kasus <i>Knapsack</i> Pertama.....	IV-17
IV.4.2 Kasus <i>Knapsack</i> Kedua	IV-18
IV.4.3 Kasus <i>Knapsack</i> Ketiga	IV-21
IV.4.4 Kasus <i>Knapsack</i> Keempat.....	IV-23
IV.4.5 Kasus <i>Knapsack</i> Kelima	IV-26
IV.4.6 Kasus <i>Knapsack</i> Keenam.....	IV-28
IV.4.7 Kasus <i>Knapsack</i> Ketujuh	IV-30
IV.4.8 Kasus <i>Knapsack</i> Kedelapan	IV-32
IV.4.9 Kasus <i>Knapsack</i> Kesembilan.....	IV-35
IV.5 Perbandingan Performansi WOA dengan Algoritma Lain ...	IV-37
BAB V ANALISIS	V-1
V.1 Analisis <i>Whale Optimization Algorithm</i> (WOA)	V-1
V.2 Analisis Perancangan Algoritma WOA untuk Kasus	
<i>Knapsack</i>	V-2
V.3 Analisis Penentuan Parameter WOA.....	V-4
V.4 Analisis Hasil Implementasi Algoritma	V-6
BAB VI KESIMPULAN DAN SARAN	V-1
V.1 Kesimpulan	V-1
V.2 Saran	V-2

DAFTAR PUSTAKA
LAMPIRAN
RIWAYAT HIDUP PENULIS

DAFTAR GAMBAR

Gambar I.1	<i>Flowchart</i> Metodologi	I-9
Gambar II.1	Klasifikasi Metode-Metode Optimisasi	II-3
Gambar II.2	<i>Bubble-Net Feedingt</i>	II-4
Gambar II.3	Peran-Peran Paus Bungkok dalam <i>Bubble-Net Feeding</i>	II-5
Gambar II.4	Perpindahan Vektor Posisi	II-6
Gambar II.5	Gerakan Menyerang Mangsa	II-7
Gambar II.6	Mekanisme Pencairan Mangsa	II-8
Gambar II.7	Rumus Fungsi Transfer	II-9
Gambar III.1	Matriks Posisi Paus	III-2
Gambar III.2	<i>Flowchart</i> Utama	III-17
Gambar III.3	<i>Flowchart</i> Pembuatan Matriks Posisi Awal	III-20
Gambar III.4	<i>Flowchart</i> Menyerang Mangsa	III-23
Gambar III.5	<i>Flowchart</i> Mencari Mangsa	III-24
Gambar III.6	<i>Flowchart</i> Melingkari Mangsa	III-25
Gambar III.7	<i>Flowchart</i> Perbaikan Posisi Paus	III-26
Gambar III.8	<i>Flowchart</i> Fungsi Transfer Matriks Posisi	III-29
Gambar IV.1	<i>Input</i> Data Permasalahan <i>Knapsack</i>	IV-1
Gambar IV.2	Program <i>Input</i> Data Parameter WOA	IV-2
Gambar IV.3	Deklarasi Variabel Program	IV-3
Gambar IV.4	Program Algoritma Matriks Posisi Awal	IV-4
Gambar IV.5	Program Pembaruan Posisi	IV-5
Gambar IV.6	Program Transfer Fungsi	IV-6
Gambar IV.7	<i>Error</i> pada <i>Input</i> Jumlah Benda	IV-8
Gambar IV.8	<i>Error</i> pada <i>Input</i> Berat Benda	IV-8
Gambar IV.9	<i>Error</i> pada <i>Input</i> Keuntungan Benda	IV-8
Gambar IV.10	<i>Error</i> pada <i>Input</i> Kapasitas Tas	IV-9
Gambar IV.11	<i>Error</i> pada <i>Input</i> Jumlah Iterasi	IV-9
Gambar IV.12	<i>Error</i> Kedua pada <i>Input</i> Jumlah Iterasi	IV-9
Gambar IV.13	<i>Error</i> pada <i>Input</i> Jumlah Paus	IV-10
Gambar IV.14	<i>Error</i> Kedua pada <i>Input</i> Jumlah Paus	IV-10

Gambar IV.15 <i>Error</i> pada <i>Input</i> Batas Atas a	IV-10
Gambar IV.16 <i>Error</i> Kedua pada <i>Input</i> Batas Atas a	IV-11
Gambar IV.19 Contoh Hasil <i>Run</i> Program.....	IV-11
Gambar IV.20 Grafik Nilai Solusi untuk Jumlah Iterasi maksimum 100	IV-12
Gambar IV.21 Grafik Nilai Solusi untuk Jumlah Iterasi maksimum 100.....	IV-13
Gambar IV.22 Grafik Nilai Solusi untuk Jumlah Iterasi maksimum 300.....	IV-13
Gambar IV.23 Grafik Nilai Solusi untuk Jumlah Iterasi maksimum 400.....	IV-14
Gambar IV.24 Spiral Logaritmik.....	IV-16
Gambar IV.25 ANOVA Multifaktor untuk Kasus Kedua	IV-20
Gambar IV.26 <i>Main Effect Plot</i> Parameter b_a Kasus Kedua	IV-21
Gambar IV.27 ANOVA Multifaktor untuk Kasus Ketiga	IV-23
Gambar IV.28 <i>Main Effect Plot</i> Parameter b_a Kasus Ketiga	IV-24
Gambar IV.29 ANOVA Multifaktor untuk Kasus Keempat	IV-25
Gambar IV.30 <i>Main Effect Plot</i> Parameter b_a Kasus Keempat.....	IV-26
Gambar IV.31 ANOVA Multifaktor untuk Kasus Kelima	IV-28
Gambar IV.32 <i>Main Effect Plot</i> Parameter b_a Kasus Kelima	IV-28
Gambar IV.33 ANOVA Multifaktor untuk Kasus Keenam	IV-30
Gambar IV.34 <i>Main Effect Plot</i> Parameter b_a Kasus Keenam	IV-31
Gambar IV.35 ANOVA Multifaktor untuk Kasus Ketujuh	IV-32
Gambar IV.36 <i>Main Effect Plot</i> Parameter b_a Kasus Ketujuh	IV-33
Gambar IV.37 ANOVA Multifaktor untuk Kasus Kedelapan	IV-34
Gambar IV.38 <i>Main Effect Plot</i> Parameter b_a dan $b_{constant}$ Kasus Kedelapan	IV-35
Gambar IV.39 ANOVA Multifaktor untuk Kasus Kesembilan.....	IV-37
Gambar IV.40 <i>Main Effect Plot</i> Parameter b_a Kasus Kesembilan.....	IV-37

DAFTAR TABEL

Tabel III.1	Kasus Sederhana <i>Knapsack</i>	III-2
Tabel III.2	Perhitungan Berat dan Keuntungan Tas Awal	III-4
Tabel III.3	Perhitungan Berat dan Keuntungan Tas Setelah Pembaruan Posisi	III-13
Tabel III.4	Perhitungan Rasio Keuntungan terhadap Berat Benda	III-13
Tabel III.5	Perhitungan Berat dan Keuntungan Tas Setelah Perbaikan Solusi Paus Ketiga	III-14
Tabel IV.1	Pengamatan Parameter Jumlah Paus.....	IV-15
Tabel IV.2	Kombinasi Parameter WOA	IV-16
Tabel IV.3	Rekapitulasi Ciri-Ciri Kasus <i>Benchmark</i> yang Digunakan	IV-17
Tabel IV.4	Data Kasus <i>Knapsack</i> Pertama.....	IV-17
Tabel IV.5	Rekapitulasi Solusi Implementasi WOA pada Kasus <i>Knapsack</i> Pertama	IV-17
Tabel IV.6	Data Kasus <i>Knapsack</i> Kedua.....	IV-18
Tabel IV.7	Rekapitulasi Solusi Implementasi WOA pada Kasus <i>Knapsack</i> Kedua	IV-19
Tabel IV.8	Hasil Uji ANOVA Kasus Kedua	IV-20
Tabel IV.9	Data Kasus <i>Knapsack</i> Ketiga	IV-21
Tabel IV.10	Rekapitulasi Solusi Implementasi WOA pada Kasus <i>Knapsack</i> Ketiga	IV-22
Tabel IV.11	Hasil Uji ANOVA Kasus Ketiga.....	IV-23
Tabel IV.12	Rekapitulasi Solusi Implementasi WOA pada Kasus <i>Knapsack</i> Keempat	IV-24
Tabel IV.13	Hasil Uji ANOVA Kasus Keempat	IV-26
Tabel IV.14	Rekapitulasi Solusi Implementasi WOA pada Kasus <i>Knapsack</i> Kelima.....	IV-27
Tabel IV.15	Hasil Uji ANOVA Kasus Kelima.....	IV-28
Tabel IV.16	Rekapitulasi Solusi Implementasi WOA pada Kasus <i>Knapsack</i> Keenam	IV-29
Tabel IV.17	Hasil Uji ANOVA Kasus Keenam	IV-30

Tabel IV.18	Rekapitulasi Solusi Implementasi WOA pada Kasus <i>Knapsack</i> Ketujuh.....	IV-31
Tabel IV.19	Hasil Uji ANOVA Kasus Ketujuh	IV-32
Tabel IV.20	Rekapitulasi Solusi Implementasi WOA pada Kasus <i>Knapsack</i> Kedelapan.....	IV-33
Tabel IV.21	Hasil Uji ANOVA Kasus Kedelapan	IV-35
Tabel IV.22	Rekapitulasi Solusi Implementasi WOA pada Kasus <i>Knapsack</i> Kesembilan	IV-36
Tabel IV.23	Hasil Uji ANOVA Kasus Kesembilan	IV-37
Tabel IV.24	Pengaruh Parameter terhadap Solusi Masing-Masing Iterasi	IV-38
Tabel IV.25	Perbandingan Algoritma VS dan EHO terhadap WOA	IV-38
Tabel IV.26	Perbandingan Algoritma HS, ABC, dan SLC terhadap WOA .	IV-39

DAFTAR LAMPIRAN

LAMPIRAN A : *DATA BENCHMARK PROBLEM KE-4, 5, 6, 7, 8, DAN 9*

BAB I

PENDAHULUAN

Pada bab ini diberikan gambaran dari penelitian yang dilakukan. Bab ini terdiri dari latar belakang masalah, identifikasi dan rumusan masalah, serta pembatasan masalah, tujuan dan manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

I.1 Latar Belakang Masalah

Setiap hari semua orang umumnya selalu menghadapi kondisi dimana perlu melakukan pengambilan keputusan. Dalam mengambil keputusan sering dihadapi pula beberapa batasan seperti batasan waktu, biaya, jarak, dan lain-lain. Namun setiap orang pasti menginginkan pengambilan keputusannya merupakan keputusan terbaik. Permasalahan pengambilan keputusan tersebut dapat digambarkan sebagai permasalahan 0/1 *knapsack*.

Secara sederhana *knapsack problem* merupakan suatu permasalahan dalam menentukan barang yang hendak dibawa di dalam tas agar memberikan keuntungan maksimal bila kapasitas tas terbatas. *Knapsack problem* tidak terbatas hanya mengenai penentuan barang yang akan dibawa di dalam sebuah tas, namun juga dapat muncul pada permasalahan bidang ekonomi, teknik, dan bisnis (Bartholdi, III, 2008). Contoh dari *knapsack problem* di bidang bisnis yaitu seorang direktur yang hendak melakukan investasi dengan *budget* yang terbatas dan harus memilih investasi yang paling memberikan keuntungan di antara 100 investasi yang berpotensi (Bartholdi, III, 2008). Keputusan yang dibuat pada *knapsack problem* dalam penelitian ini dinyatakan dalam variabel biner yaitu bernilai 0 atau 1. Variabel dinyatakan bernilai 0 jika barang ditentukan untuk tidak dibawa dalam tas, sedangkan nilai 1 diberikan jika barang ditentukan untuk dibawa dalam tas. *Knapsack problem* seperti ini diberi nama 0/1 *knapsack problem*.

Menurut Runggeratigul (2003) walaupun struktur *knapsack problem* sederhana, namun jika *knapsack problem* diperluas maka *knapsack problem* dapat tergolong ke dalam permasalahan yang *Non-deterministic Polynomial-time Hard* (NP-hard). Perluasan *knapsack problem* dapat berupa penambahan jumlah

barang dan peningkatan kapasitas *knapsack*. Hristakeva dan Shrestha (2005) mencoba membandingkan hasil penyelesaian permasalahan optimasi pada *knapsack problem* dengan menggunakan metode *brute force*, *dynamic programming*, *memory function*, *branch and bound*, *greedy algorithm*, dan *genetic algorithm*. Hasil penyelesaian permasalahan optimasi yang dibandingkan yaitu nilai total penyelesaian *knapsack problem*, rata-rata jumlah operasi dasar, dan memori yang digunakan ketika jumlah barang dan kapasitas *knapsack* (misal *knapsack* pada kasus ini dilambangkan dengan tas) meningkat. Jika jumlah barang meningkat, metode *brute force* tidak dapat digunakan ketika jumlah barang lebih besar dari 25 *item*. Metode *branch and bound*, *dynamic programming*, dan *genetic algorithm* memiliki solusi yang lebih baik dibandingkan *greedy algorithm*. Namun metode *branch and bound* tidak dapat digunakan ketika jumlah barang lebih besar dari 750 *item*. Sedangkan kedua metode lainnya yaitu metode *dynamic programming* dan *genetic algorithm* memberikan hasil yang baik. Ketika jumlah barang bertambah, metode *dynamic programming* memberikan hasil yang lebih baik dibandingkan *genetic algorithm* karena metode *dynamic programming* menggunakan jumlah operasi dan memori yang lebih sedikit dibandingkan dengan *genetic algorithm*. Namun ketika dilakukan perubahan kapasitas *knapsack*, metode *dynamic programming* menggunakan jumlah operasi dan memori yang jauh lebih besar dibandingkan metode *genetic algorithm*. Pertambahan kapasitas *knapsack*, tidak begitu mempengaruhi jumlah operasi dan memori yang dibutuhkan pada metode *genetic algorithm*. *Genetic algorithm* merupakan salah satu algoritma metaheuristik. Bertambahnya jumlah barang dan kapasitas pada suatu masalah *knapsack* akan mempersulit penyelesaian masalah sehingga belakangan ini sering dilakukan penyelesaian permasalahan *knapsack* yang bersifat NP-*hard* dengan menggunakan algoritma metaheuristik. Belakangan ini metode metaheuristik banyak digunakan untuk menyelesaikan permasalahan-permasalahan NP-*hard*. Menurut Blum dan Roli (2003) metaheuristik merupakan metode pendekatan dan biasanya bersifat non-deterministik. Selain itu metode metaheuristik juga bersifat umum sehingga dapat diterapkan dalam berbagai persoalan.

Permasalahan 0/1 *knapsack problem* dapat diselesaikan dengan beberapa metode, beberapa diantaranya yaitu *branch and bound* dan *dynamic programming* oleh Dantzig (1957), *problem-specific heuristic* oleh Dantzig (1957),

tabu search oleh Hanafi dan Freville (1998), dan *genetic algorithm* oleh Michalewicz dan Arabas (1994). Metode *tabu search* dan *genetic algorithm* merupakan salah satu dari algoritma metaheuristik. Mirjalili dan Lewis (2016) menemukan algoritma kelas *swarm-based* baru bernama *Whale Optimization Algorithm* (WOA). WOA terinspirasi dari metode berburu ikan paus *humpback* yang special bernama *bubble-net feeding*. WOA digunakan untuk menyelesaikan masalah optimisasi *knapsack*.

I.2 Identifikasi dan Rumusan Masalah

Knapsack problem merupakan masalah pengalokasian sumber daya dengan batasan kapasitas *knapsack* yang berfokus terhadap maksimasi keuntungan. Ketika permasalahan *knapsack* memiliki pertambahan jumlah sumber daya dan kapasitas *knapsack* yang membesar, kasus *knapsack* menjadi bersifat NP-hard karena semakin sulitnya penyelesaian masalah *knapsack*. Menurut hasil penelitian yang dilakukan Hristakeva dan Shrestha (2005), ketika permasalahan *knapsack* memiliki pertambahan sumber daya dan kapasitas *knapsack* yang membesar, maka metode yang terbaik untuk menyelesaikan permasalahan *knapsack* tersebut adalah metaheuristik.

Knapsack problem sudah pernah diselesaikan dengan algoritma metaheuristik kelas *evolutionary*, *swarm-based*, dan *human-based*. Beberapa algoritma kelas *evolutionary* yang pernah digunakan untuk menyelesaikan masalah optimisasi 0/1 *knapsack* yaitu *memetic algorithm* (Runggeratigul, 2003) dan *viral systems* (Kartika, 2010). Algoritma kelas *swarm-based* yang digunakan untuk menyelesaikan masalah optimisasi 0/1 *knapsack* yaitu *particle swarm optimization* (Liang, Liu, Wang, & Wu, 2010), *novel global harmony search algorithm* (Zou, Gao, Li, & Wu, 2010), *amoeboid organism algorithm* (Zhang, Huang, Hu, Zhang, Mahadevan, & Deng, 2013), *elephant herding optimization* (Rico, 2017), dan *grey wolf optimization* (Yassien, Masadeh, Alzaqebah, & Shaheen, 2017). Sedangkan algoritma kelas *human-based* yang digunakan untuk menyelesaikan masalah optimisasi 0/1 *knapsack* yaitu *soccer league competition* (Moosavian, 2014).

Solusi penyelesaian masalah dengan algoritma metaheuristik memberikan performansi yang baik namun tidak pasti bahwa solusi yang diberikan telah optimal. Wang, Yang, Xu, Niu, Pardalos, dan Fei (2013) membandingkan

penyelesaian masalah 0/1 *knapsack* yang sama menggunakan algoritma *Binary Harmony Search* (BHS), *Adaptive Binary Harmony Search* (ABHS), *Novel Global Harmony Search* (NGHS), dan *Discrete Binary Particle Swarm Optimization* (DBPSO). Berdasarkan hasil yang diperoleh, algoritma ABHS memberikan akurasi pencarian solusi yang lebih baik dibandingkan algoritma lainnya. Kemudian Moosavian (2014) membandingkan penyelesaian masalah 0/1 *knapsack* yang sama dengan Wang, et. al. dengan menggunakan algoritma *Soccer League Competition* (SLC). Penyelesaian masalah 0/1 *knapsack* yang dihasilkan oleh algoritma SLC memberikan solusi yang lebih baik dibandingkan algoritma BHS, ABHS, NGHS, dan DBPSO. Selain itu Rico (2017) menyelesaikan permasalahan 0/1 *knapsack* yang sama dengan Kartika (2010). Solusi permasalahan 0/1 *knapsack* yang diselesaikan dengan algoritma *elephant herding optimization* ternyata memberikan solusi yang lebih baik dibandingkan penyelesaian masalah dengan algoritma *viral systems*. Penyelesaian 0/1 *knapsack* dengan algoritma *elephant herding optimization* yang merupakan algoritma tipe *swarm* memberikan hasil yang lebih baik dibandingkan algoritma *viral systems* yang merupakan algoritma tipe *evolutionary*. Maka dari itu dilakukan penyelesaian masalah 0/1 *knapsack* dengan menggunakan algoritma *swarm-based* baru bernama *whale optimization algorithm* yang diharapkan dapat memberikan solusi yang lebih baik dalam menyelesaikan masalah 0/1 *knapsack*.

Algoritma *whale optimization* merupakan algoritma tipe *swarm* baru yang terinspirasi dari metode berburu ikan paus *humpback* yang spesial (Mirjalili & Lewis, 2016). Ikan paus *humpback* memiliki metode berburu bernama *bubble-net feeding*. Saat berburu ikan paus *humpback* membuat dua gerakan yang diberi nama *upward spiral* (spiral ke atas) dan *double loops* (putaran ganda). Ikan paus *humpback* berenang dari kedalaman 12 meter dan membuat gelembung udara dalam bentuk spiral di sekitar mangsanya, kemudian berenang ke permukaan untuk memakan mangsanya.

Mirjalili dan Lewis (2016) telah menggunakan algoritma *whale optimization* untuk menyelesaikan 6 masalah teknik struktural. Pada kasus-kasus tersebut, performansi algoritma *whale optimization* dibandingkan dengan algoritma metaheuristik lainnya yaitu *particle swarm optimization*, *genetic algorithm*, *ant colony optimization*, dan lain-lain. Solusi yang didapatkan dengan menggunakan algoritma *whale optimization* cukup kompetitif dibandingkan

algoritma metaheuristik lainnya dan teknik konvensional seperti *lagrangian multiplier* dan *branch and bound*. Selain itu Sharawi, Zawbaa, dan Emary (2017) juga membandingkan performansi algoritma *whale optimization* dengan *genetic algorithm* dan *particle swarm optimization* dalam menyelesaikan *feature selection*. Hasilnya algoritma *whale optimization* dapat memberikan kombinasi optimal dengan keakuratan klasifikasi maksimum.

Algoritma *whale optimization* memiliki beberapa parameter internal yaitu jumlah agen pencari (n), jumlah iterasi maksimum (t), dan vektor \vec{A} (Mirjalili & Lewis, 2016). Mirjalili dan Lewis (2016) melakukan penelitian mengenai efek dari nilai parameter terhadap performansi algoritma *whale optimization* pada masalah desain struktural *truss 52* batang. Menurut hasil penelitian tersebut jika jumlah agen pencari dan iterasi semakin besar, maka solusi permasalahan yang didapat semakin mendekati solusi global optimum. Sedangkan nilai parameter \vec{A} yang digunakan untuk memberikan solusi lebih baik adalah penurunan nilai secara linear dari 2 ke 0.

Sebelumnya *0/1 knapsack problem* sudah pernah diselesaikan dengan beberapa algoritma *swarm-based* yaitu *particle swarm optimization*, *elephant herding optimization*, dan lain-lain. Penyelesaian masalah dengan algoritma metaheuristik tidak menjamin dapat memberikan solusi yang optimal. Jika masalah *0/1 knapsack* diselesaikan dengan algoritma baru, maka mungkin saja dapat dihasilkan solusi permasalahan yang lebih baik dibandingkan algoritma yang sudah pernah digunakan sebelumnya. Algoritma *whale optimization* belum pernah digunakan untuk menyelesaikan permasalahan *0/1 knapsack*. Sebagai algoritma *swarm-based* baru, algoritma *whale optimization* diharapkan dapat menghasilkan solusi permasalahan *0/1 knapsack problem* yang lebih baik. Berdasarkan identifikasi masalah yang ada, dapat dirumuskan beberapa masalah yang ada dalam penelitian yang dilakukan:

1. Bagaimana penerapan algoritma *whale optimization* untuk *knapsack problem*?
2. Apa saja parameter yang berpengaruh pada algoritma *whale optimization* untuk *knapsack problem*?
3. Bagaimana perbandingan solusi *knapsack problem* dari penggunaan algoritma *whale optimization* terhadap solusi dari penggunaan algoritma

viral system (Kartika, 2010), *elephant herding optimization* (Rico, 2017), *harmony search*, *artificial bee colony*, dan *soccer league competition* (Moosavian, 2014)?

I.3 Pembatasan Masalah

Pembatasan masalah perlu diberikan untuk memperkecil ruang lingkup penelitian. Ruang lingkup penelitian perlu diperkecil karena adanya keterbatasan kemampuan dan waktu. Batasan yang diberikan pada penelitian yaitu sebagai berikut.

1. Jenis *knapsack problem* yang digunakan yaitu *0/1 knapsack problem*.
2. Kasus yang digunakan bersifat hipotetik dan hanya menggunakan data dari *benchmark problem*.
3. Waktu komputasi yang digunakan untuk mencari solusi menggunakan algoritma *whale optimization* tidak menjadi ukuran performansi.

I.4 Tujuan Penelitian

Dengan mengacu pada identifikasi masalah yang ada, terdapat beberapa tujuan dilakukannya penelitian terhadap permasalahan optimasi *knapsack* menggunakan algoritma *whale optimization* yaitu.

1. Menerapkan algoritma *whale optimization* pada *0/1 knapsack problem*.
2. Mengetahui parameter apa saja yang berpengaruh pada algoritma *whale optimization* untuk *knapsack problem*.
3. Mengetahui perbandingan solusi *knapsack problem* dari penggunaan algoritma *whale optimization* terhadap solusi dari penggunaan algoritma *viral system* (Kartika, 2010), *elephant herding optimization* (Rico, 2017), *harmony search*, *artificial bee colony*, dan *soccer league competition* (Moosavian, 2014).

I.5 Manfaat Penelitian

Berdasarkan tujuan penelitian yang ada, diharapkan penelitian ini dapat memberikan manfaat sebagai berikut.

1. Dapat memberi pengetahuan mengenai penggunaan algoritma *whale optimization* terhadap *knapsack problem*.

2. Dapat memberi solusi *knapsack problem* menggunakan algoritma *whale optimization*.
3. Dapat memberi informasi mengenai performansi algoritma *whale optimization* dalam menyelesaikan 0/1 *knapsack problem* dibandingkan algoritma *viral system*, *elephant herding optimization*, *harmony search*, *artificial bee colony*, dan *soccer league competition* yang sudah pernah menyelesaikan 0/1 *knapsack problem*.

I.6 Metodologi Penelitian

Dalam penelitian mengenai penyelesaian masalah 0/1 *knapsack* dengan algoritma *whale optimization* digunakan metodologi penelitian. Tahapan metodologi yang digunakan dalam penelitian disajikan dalam bentuk *flowchart* yang dapat dilihat pada Gambar I.1.

1. Studi Literatur

Pada tahap ini dilakukan pencarian dan pengumpulan literatur mengenai algoritma *whale optimization* dan 0/1 *knapsack problem*. Literatur yang digunakan dapat berasal dari buku, jurnal, maupun artikel *online*.

2. Identifikasi dan Perumusan Masalah

Pada tahap ini dilakukan penentuan masalah 0/1 *knapsack* yang diteliti. Setelah itu masalah *knapsack* yang telah ditentukan sebelumnya dirumuskan.

3. Pembatasan Masalah

Selanjutnya dilakukan pembatasan terhadap masalah yang diteliti. Pembatasan masalah dilakukan agar ruang lingkup penelitian tidak terlalu luas. Pembatasan masalah perlu dilakukan karena keterbatasan waktu dan kemampuan yang dimiliki dalam menyelesaikan penelitian.

4. Penentuan Tujuan dan Manfaat Penelitian

Pada tahap ini ditentukan tujuan serta manfaat dilakukannya penelitian. Tujuan perlu ditentukan agar penelitian yang dilakukan terarah. Jika tujuan penelitian tercapai, maka penelitian diharapkan bermanfaat bagi pembaca.

5. Perancangan Algoritma *Whale Optimization* untuk Permasalahan 0/1 *Knapsack*

Algoritma *whale optimization* dirancang untuk menyelesaikan permasalahan 0/1 *knapsack*.

6. Validasi Algoritma *Whale Optimization*

Berikutnya algoritma yang telah dirancang divalidasi. Validasi algoritma dilakukan untuk memastikan apakah algoritma yang dibuat telah dapat menyelesaikan masalah *0/1 knapsack* dengan benar.

7. Pembuatan Program Algoritma *Whale Optimization* untuk Permasalahan *0/1 Knapsack*

Setelah algoritma yang dibuat valid dan terverifikasi, dibuat program berdasarkan algoritma yang ada untuk menyelesaikan permasalahan *0/1 knapsack*.

8. Verifikasi Program

Berikutnya program yang telah dirancang diverifikasi. Verifikasi program dilakukan untuk memastikan bahwa program yang dibuat telah sesuai dengan algoritma yang telah dirancang sebelumnya.

9. Implementasi Algoritma

Pada tahap ini program yang telah valid dan terverifikasi digunakan untuk menyelesaikan permasalahan *0/1 knapsack*. Data masalah *0/1 knapsack* berasal dari *benchmark problem* dan bersifat hipotetik. Data yang digunakan pada penelitian telah digunakan sebelumnya oleh Kartika (2010), Rico (2017), dan Moosavian (2014).

10. Pengujian Pengaruh Parameter

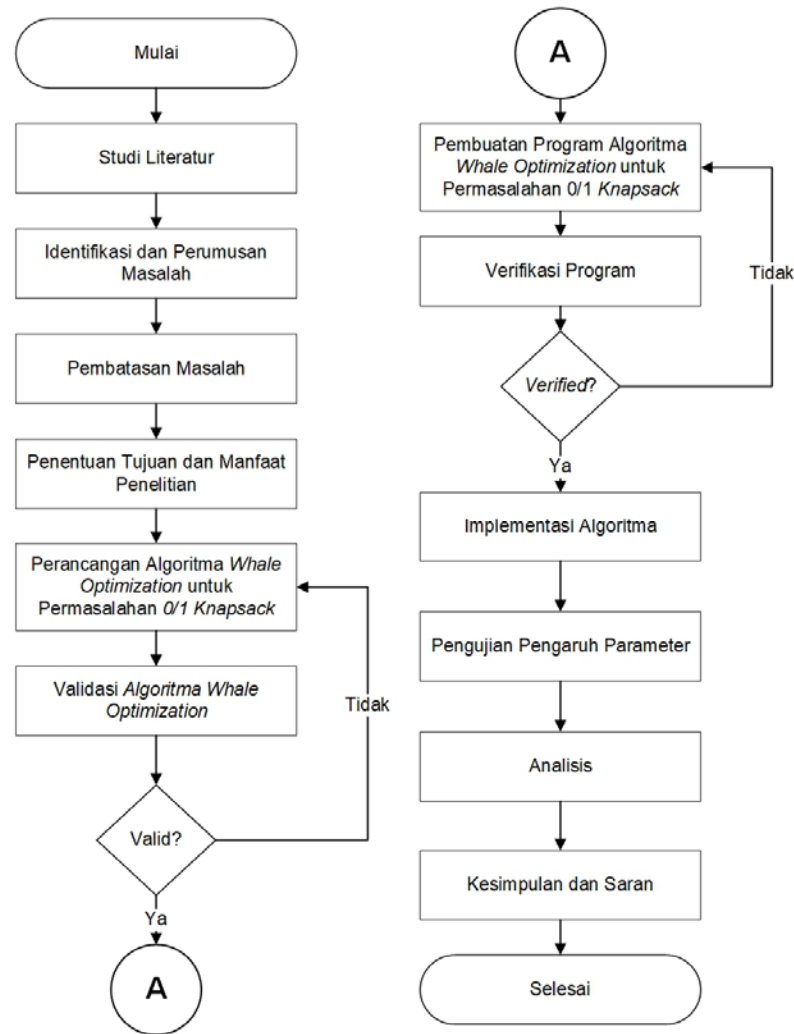
Dilakukan pengujian parameter untuk mengetahui parameter algoritma *whale optimization* apa saja yang memiliki pengaruh pada penyelesaian masalah *0/1 knapsack*.

11. Analisis

Pada tahap ini dilakukan analisis terhadap solusi penyelesaian masalah *0/1 knapsack* menggunakan algoritma *whale optimization* yang dihasilkan. Kemudian solusi yang dihasilkan dibandingkan dengan solusi penyelesaian masalah menggunakan algoritma lain pada penelitian sebelumnya.

12. Kesimpulan dan Saran

Pada tahap ini diberikan kesimpulan dan saran dari penelitian yang dilakukan. Kesimpulan merupakan rangkuman dari analisis untuk menjawab masalah yang telah dirumuskan sebelumnya. Sedangkan saran diberikan sebagai bahan pertimbangan untuk penelitian selanjutnya.



Gambar 1.1 Flowchart Metodologi

I.7 Sistematika Penulisan

Pada subbab ini disajikan sistematika penulisan penelitian. Pada penelitian ini terdapat 6 bab yaitu pendahuluan, tinjauan pustaka, perancangan dan implementasi algoritma, analisis, serta kesimpulan dan saran. Adapun sistematika penulisan penelitian yang digunakan yaitu.

BAB I PENDAHULUAN

Pada bab ini dibahas mengenai gambaran dari dilakukannya penelitian. Bab ini berisi latar belakang masalah, identifikasi dan rumusan masalah, pembatasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini dijabarkan teori-teori yang digunakan dalam penelitian. Tinjauan pustaka yang digunakan yaitu mengenai masalah *knapsack*, desain eksperimen, dan algoritma *whale optimization*.

BAB III PERANCANGAN ALGORITMA

Pada bab ini dilakukan proses perancangan *Whale Optimization Algorithm* (WOA) untuk permasalahan *knapsack*. Pertama-tama ditentukan proses *encoding* dan *decoding*, berikutnya yaitu merancang algoritma WOA untuk permasalahan *knapsack*.

BAB IV IMPLEMENTASI ALGORITMA

Pada bab ini dibahas mengenai implementasi permasalahan *knapsack* dengan menggunakan program. Program dibuat dengan menggunakan *software* MATLAB. Sebelum melakukan implementasi masalah ke dalam program, terlebih dahulu dilakukan verifikasi dan validasi program. Berikutnya dilakukan implementasi WOA terhadap kasus *knapsack* yang digunakan juga oleh Kartika (2010) dan Rico (2017). Pada bab ini juga ditentukan parameter yang mempengaruhi nilai respon (keuntungan).

BAB V ANALISIS

Pada bab ini dilakukan pembahasan mengenai *Whale Optimization Algorithm* (WOA), perancangan algoritma WOA, dan hasil implementasi algoritma WOA.

BAB VI KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan dan saran dari penelitian yang dilakukan. Saran diberikan sebagai masukan untuk dipertimbangkan pada penelitian serupa selanjutnya.