PENERAPAN *DRAGONFLY ALGORITHM* UNTUK MENYELESAIKAN *KNAPSACK SHARING PROBLEM*

SKRIPSI

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

Disusun oleh :

Nama : Stefanus Ivan Laksono

NPM : 2014610052

PROGRAM STUDI TEKNIK INDUSTRI

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS KATOLIK PARAHYANGAN

BANDUNG

2018

FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS KATOLIK PARAHYANGAN
BANDUNG

Nama	: Stefanus Ivan Laksono
NPM	: 2014610052
Program Studi	: Teknik Industri
Judul Skripsi	: PENERAPAN *DRAGONFLY ALGORITHM* UNTUK MENYELESAIKAN *KNAPSACK SHARING PROBLEM*

TANDA PERSETUJUAN SKRIPSI

Bandung, 30 Juli 2018

Ketua Program Studi Teknik Industri

(Dr. Carles Sitompul, S.T., M.T., M.I.M.)

Dosen Pembimbing I

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

Dosen Pembimbing II

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

Program Studi Teknik Industri
Fakultas Teknologi Industri
Universitas Katolik Parahyangan

Pernyataan Tidak Mencontek atau Melakukan Tindakan Plagiat

Saya yang bertanda tangan di bawah ini
Nama : Stefanus Ivan Laksono
NPM : 2014610052

dengan ini menyatakan bahwa Skripsi dengan judul :

"PENERAPAN *DRAGONFLY ALGORITHM* UNTUK MENYELESAIKAN *KNAPSACK SHARING PROBLEM*"

adalah hasil pekerjaan saya dan seluruh ide, pendapat, dan materi dari sumber lain telah dikutip dengan cara penulisan referensi yang sesuai.

Pernyataan ini saya buat dengan sebenar-benarnya dan jika pernyataan ini tidak sesuai dengan kenyataan, maka saya bersedia menanggung sanksi yang akan dikenakan kepada saya.

Bandung, 14 Juli 2018

Stefanus Ivan Laksono
NPM: 2014610052

ABSTRAK

Knapsack sharing problem (KSP) merupakan permasalahan pengalokasian atau pendistribusian sumber daya terbatas kepada pihak-pihak yang membutuhkan. Pada KSP dianalogikan terdapat sebuah tas yang memiliki kapasitas tertentu dan terbagi menjadi beberapa kelas. Tas diisi dengan benda-benda yang memiliki berat dan keuntungan yang berbeda-beda. Tujuannya adalah memaksimasi nilai minimum dari keuntungan yang diperoleh setiap kelas dalam tas. Solusi model KSP berupa kombinasi benda yang dimasukkan ke dalam setiap kelas pada tas. Di dunia nyata, model KSP digunakan untuk menyelesaikan permasalahan pengalokasian dana dan *line balancing* pabrik manufaktur.

Pada penelitian ini, KSP diselesaikan dengan menggunakan *dragonfly algorithm*. *Dragonfly algorithm* merupakan algoritma metaheuristik yang terinspirasi dari perilaku berkerumun yang dimiliki capung. Aktivitas yang dilakukan kerumunan capung tersebut adalah mencari sumber makanan, menghindari pemangsa, menghindari tabrakan dengan capung lain, menyeimbangkan kecepatan dengan capung lain, dan cenderung untuk berada di pusat kerumunan capung. Kelima hal tersebut menjadi faktor penentu posisi capung dalam melakukan pencarian solusi.

Dragonfly algorithm yang telah dirancang diimplementasikan pada 10 kasus *benchmark* KSP dengan 16 kombinasi parameter dan di setiap kombinasi dilakukan 5 replikasi. Dari hasil implementasi diketahui bahwa *dragonfly algorithm* mampu menghasilkan solusi optimal pada 6 kasus. Untuk 4 kasus lainnya, *dragonfly algorithm* menghasilkan solusi dengan penyimpangan terbesar sebesar 0,251% dan penyimpangan terkecil sebesar 0,025% dari solusi optimal. Hasil implementasi tersebut dibandingkan dengan *cat swarm optimization*, *cuckoo search*, dan *tabu search*. *Dragonfly algorithm* memiliki performansi yang lebih baik daripada *cat swarm optimization* dan *cuckoo search* serta memiliki performansi yang sama baiknya dengan *tabu search*. Terdapat 4 parameter *dragonfly algorithm* yang diuji pengaruhnya terhadap performansi algoritma, yaitu batas atas w (ubw), batas bawah w (lbw), batas atas my_c (ubc), dan batas bawah my_c (lbc). Hasil pengujian menunjukkan adanya pengaruh dan interaksi antar parameter terhadap performansi *dragonfly algorithm* di seluruh kasus *benchmark* yang diuji.

ABSTRACT

Knapsack sharing problem (KSP) is a problem of allocating or distributing limited resources to several entities. In KSP, analogous there is a bag that has a certain capacity and is divided into several classes. Bags filled with objects that have different weight and profit. The goal is to maximize the minimum value of the profits from each class in the bag. The KSP model solution is a combination of objects entered into each class on the bag. In the real world, KSP models are used to solve fund allocation problems and factory manufacturing line balancing.

In this research, KSP is solved by using dragonfly algorithm. Dragonfly algorithm is a metaheuristic algorithm that is inspired by the swarming behavior of the dragonfly. The activities of the dragonfly swarm are looking for food sources, avoiding enemy, avoiding collisions with other dragonflies, matching velocity with other dragonflies, and tending to be in the center of mass of the dragonfly swarm. These five activities affect the position of dragonfly in searching for solutions in search space.

The dragonfly algorithm has been implemented in 10 KSP benchmark cases with 16 combinations of parameters and in each combination 5 replications are performed. From the results of the implementation, dragonfly algorithm able to produce optimal solution in 6 cases. For the other 4 cases, the dragonfly algorithm produced the solution with the largest deviation of 0.251% and the smallest deviation of 0.025% of the optimal solution. The results are compared with cat swarm optimization, cuckoo search, and tabu search. Dragonfly algorithm has better performance than cat swarm optimization and cuckoo search and performs equally well with tabu search. There are 4 parameters of the dragonfly algorithm which tested its effect on the performance of the algorithm that is upper limit w (ubw), lower limit w (lbw), upper limit my_c (ubc), and lower limit my_c (lbc). The test results show that the main effect of parameters and interaction between parameters is present on dragonfly algorithm performance in all benchmark cases tested.

KATA PENGANTAR

Puji syukur kepada Tuhan yang Maha Esa karena atas berkat dan rahmat yang diberikan, penulis dapat menyelesaikan laporan skripsi dengan judul "Penerapan *Dragonfly Algorithm* untuk Menyelesaikan *Knapsack Sharing Problem*". Selama proses penyusunan laporan skripsi ini, penulis mendapatkan banyak pengalaman, bantuan, dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Bapak Hanky Fransiscus, S.T., M.T. selaku dosen pembimbing pertama yang telah memberikan motivasi, masukan, dan bimbingan dalam penyusunan laporan skripsi.
2. Ibu Cynthia Prithadevi Juwono, Ir., M.S. selaku dosen pembimbing kedua yang telah memberikan motivasi, masukan, dan bimbingan dalam penyusunan laporan skripsi.
3. Bapak Ignatius A. Sandy, S.Si., M.T. selaku dosen penguji proposal dan sidang skripsi yang telah memberikan berbagai kritik dan masukan dalam penyusunan laporan skripsi.
4. Bapak Fran Setiawan, S.T., M.Sc. selaku dosen penguji proposal dan sidang skripsi yang telah memberikan berbagai kritik dan masukan dalam penyusunan laporan skripsi.
5. Orang tua penulis yang telah memberikan berbagai dukungan agar penulis dapat menyelesaikan laporan skripsi.
6. Adrianus Vincent Djunaidi yang telah memberikan masukan dan selalu mendoakan kelancaran penulis dalam penyusunan laporan skripsi.
7. Seyedali Mirjalili yang telah membantu penulis dalam pembuatan program dengan memberikan *source code* Matlab *dragonfly algorithm*.
8. Deshera Hartanto, Benardus Rogger Sopakuwa, Felick Kurnia, Ray Paulus Tanuel, dan Clarissa Florence yang telah menemani penulis selama penyusunan laporan skripsi.
9. Ranggi Maharani, Julian Anderson, William Wong, Leonardus Andrew, Hardian Gunardi, dan Henry Dharmawan Santosa selaku rekan-rekan mahasiswa yang mengambil topik skripsi algoritma yang telah

memberikan bantuan dan masukan dalam penyusunan laporan skripsi.

10. Dosen Teknik Industri Universitas Katolik Parahyangan atas segala ilmu pengetahuan yang diberikan.
11. Teman-teman penulis yang tidak dapat disebutkan satu per satu yang telah memberikan motivasi dalam proses pembuatan laporan skripsi.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari pembaca agar penulis dapat membuat laporan yang lebih baik di masa yang akan datang.

Bandung, 3 Juli 2018

Stefanus Ivan Laksono

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

Pada bab ini dijelaskan latar belakang masalah, identifikasi dan rumusan masalah, asumsi dan batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

I.1 Latar Belakang Masalah

Pengalokasian sumber daya menjadi salah satu hal penting dalam kehidupan manusia. Manusia tidak terlepas dari penggunaan sumber daya dalam pemenuhan kebutuhan hidup sehari-hari. Sumber daya yang dimaksud dapat berupa makanan, pakaian, listrik, minyak bumi, dan lain-lain. Seiring berjalannya waktu jumlah manusia selalu mengalami peningkatan, namun tidak dengan sumber daya yang tersedia. Sumber daya terus mengalami penurunan dan pada suatu waktu akan terjadi peristiwa kelangkaan. Tentu manusia tidak menginginkan hal tersebut terjadi. Salah satu cara yang dapat dilakukan untuk menangani permasalahan tersebut adalah dengan mengalokasikan sumber daya secara efektif dan efisien. Permasalahan pengalokasian sumber daya ini dapat dimodelkan ke dalam *knapsack problem*.

Knapsack problem merupakan salah satu jenis permasalahan dalam optimasi kombinatorial. *Knapsack problem* dapat dianalogikan sebagai suatu tas yang memiliki kapasitas tertentu dan tas tersebut harus diisi dengan berbagai macam benda. Setiap benda memiliki berat yang berbeda dan memberikan keuntungan yang berbeda pula bagi penggunanya. Pengguna tidak dapat memasukkan semua benda karena adanya batasan kapasitas dari tas. Tujuan utama dari *knapsack problem* adalah menentukan benda mana saja yang harus dibawa agar pengguna memperoleh keuntungan semaksimal mungkin. Walaupun terlihat sederhana, namun *knapsack problem* menjadi kompleks saat jenis benda yang dimasukkan berjumlah sangat banyak.

Knapsack problem banyak mendapat perhatian baik dari ahli teori maupun praktisi. *Knapsack problem* telah dipelajari selama berabad-abad sebagai bentuk paling sederhana dari permasalahan maksimasi (Kellerer, Pferschy, &

Pisinger, 2004). Berdasarkan hasil penelitian yang dilakukan Skiena (1999), *knapsack problem* menduduki posisi ketiga dalam jajaran permasalahan yang paling dibutuhkan implementasinya. Dari sudut pandang praktis, permasalahan ini dapat memodelkan banyak situasi dalam dunia industri, seperti penganggaran dana, pemuatan kargo, pemotongan bahan, dan lain-lain (Martello & Toth, 1990). Dengan adanya *knapsack problem*, permasalahan alokasi sumber daya yang banyak terjadi di dunia nyata dapat terselesaikan dengan efektif dan efisien.

Seperti yang sering terjadi dalam dunia industri, terdapat berbagai macam jenis permintaan konsumen yang harus dipenuhi oleh perusahaan, seperti tingkat urgensi dan prioritas dari pesanan, interval waktu penyelesaian suatu pesanan, paket dengan berat rendah tetapi volumenya tinggi, dan lain-lain (Kellerer et al., 2004). Hal ini menjadi faktor utama berkembangnya variasi dari *knapsack problem*. Perkembangan dari *knapsack problem* antara lain adalah *subset sum problem*, *bounded knapsack problem, unbounded knapsack problem*, *multidimensional knapsack problem*, *multiple knapsack problem*, *multiple-choice knapsack problem*, dan *quadratic knapsack problem* (Kellerer et al., 2004). Selain itu, Brown (1979) juga berkontribusi dalam mengembangkan salah satu variasi *knapsack problem* yang disebut *knapsack sharing problem* (KSP).

I.2 Identifikasi dan Rumusan Masalah

KSP dapat diformulasikan sebagai perkembangan dari *knapsack problem* (Yamada, Futakawa, & Takaoka, 1998). Brown (1979) menyatakan bahwa KSP merupakan permasalahan pengalokasikan sebuah sumber daya dimana setiap penerima dari sumber daya tersebut memiliki fungsi utilitas tertentu. Dalam KSP, terdapat sebuah tas yang memiliki kapasitas tertentu dan terbagi ke dalam beberapa kelas. Tas ini diisi dengan berbagai jenis benda yang memiliki berat dan keuntungan yang berbeda-beda. Tujuan dari KSP adalah memaksimasi nilai minimum dari keuntungan yang diperoleh setiap kelas dalam tas. Solusi model KSP berupa kombinasi benda yang diisi ke dalam setiap kelas dalam tas.

Salah satu contoh ilustrasi KSP dalam dunia nyata adalah pengalokasian dana pembangunan oleh seorang gubernur terhadap wilayah yang dipimpinnya. Gubernur memiliki dana pembangunan yang jumlahnya terbatas. Dana tersebut digunakan untuk 50 kemungkinan jenis proyek dan pelaksanaannya dilakukan di 10 kota. Setiap jenis proyek tentu membutuhkan biaya yang berbeda. Sama

halnya dengan manfaat yang diperoleh warga dengan adanya proyek tersebut. Misalkan biaya pelaksanaan proyek dianalogikan sebagai bobot dan manfaat proyek sebagai keuntungan. Gubernur tersebut ingin menentukan proyek mana saja yang dialokasi ke masing-masing kota. Tujuannya agar kota yang mendapatkan keuntungan dari proyek pembangunan paling sedikit tidak berbeda jauh keuntungannya dengan kota-kota yang lain. Permasalahan ini dapat dimodelkan ke dalam KSP dan solusi yang dihasilkan berupa sejumlah kombinasi proyek yang dialokasikan ke masing-masing kota.

Salah satu metode eksak yang pernah digunakan untuk menyelesaikan KSP adalah *branch and bound* (Yamada et al., 1998). Dalam penelitiannya, Yamada et al. (1998) membandingkan performansi *branch and bound* dengan algoritma *binary search*. Hasilnya, *branch and bound* membutuhkan waktu yang lebih lama daripada algoritma *binary search* untuk dapat memperoleh solusi optimal. Metode *branch and bound* juga tidak mampu menghasilkan solusi optimal pada kasus yang memiliki lebih dari 20.000 variabel dalam rentang waktu yang ditetapkan. Metode eksak lain dalam menyelesaikan KSP adalah algoritma EKSP (Hifi & Sadfi, 2002) dan algoritma *exact tree search* (Hifi & Mhalla, 2010). Penelitian terbaru mengenai metode eksak dalam penyelesaian KSP dilakukan oleh Boyer, Baz, & Elkihel (2011) dengan menggunakan metode *dynamic programming*.

Penggunaan metode eksak memang menghasilkan solusi yang optimal. Namun, KSP merupakan *NP-hard* (*Non-deteministic Polynomial-hard*) (Yamada et al., 1998). Metode eksak membutuhkan waktu yang lama untuk dapat menyelesaikan permasalahan yang bersifat *NP-hard*. Waktu penyelesaian bertambah secara eksponensial seiring dengan meningkatnya ruang lingkup permasalahan yang diteliti. Oleh karena itu, dibutuhkan metode yang dapat memberikan solusi yang baik walaupun belum tentu optimal secara global. Metode tersebut disebut dengan heuristik dan metaheuristik. Heuristik merupakan kriteria, metode, atau prinsip untuk memilih dari antara beberapa alternatif tindakan yang diharapkan paling efektif dalam mencapai tujuan tertentu (Pearl, 1984). Algoritma heuristik yang pernah digunakan untuk menyelesaikan KSP adalah *greedy algorithm* (Yamada & Futakawa, 1997) dan ILPH (*Iterative Linear Programming-based Heuristic*) (Haddar, Khemakhem, Hanafi, & Wilbaut, 2015).

Metaheuristik adalah metodologi umum tingkat tinggi yang dapat digunakan sebagai panduan dasar dalam menyelesaikan permasalahan

optimisasi tertentu (Talbi, 2009). Pada metaheuristik terdapat 2 proses utama, yaitu eksploitasi dan eksplorasi. Eksploitasi menggunakan seluruh informasi yang diperoleh dari permasalahan untuk membantu menghasilkan solusi baru yang lebih baik daripada solusi yang sudah ada (Yang, 2014). Eksploitasi memiliki kecenderungan untuk menghasilkan solusi optimal yang bersifat lokal. Berbeda dengan eksplorasi yang memungkinkan dilakukannya pencarian solusi pada ruang pencarian dengan lebih efisien (Yang, 2014). Dengan adanya ekplorasi, solusi optimal yang diperoleh dapat bersifat global. Algoritma-algoritma yang termasuk dalam metaheuristik adalah *genetic algorithm*, *particle swarm optimization*, *cuckoo search,* dan *dragonfly algorithm*.

Algoritma metaheuristik yang pernah diterapkan pada KSP adalah *tabu search* (Hifi, Sadfi, & Sbihi, 2002), *ant colony system* (Widiapradja, 2013), *cuckoo search* (Angga, 2014), dan *cat swarm optimization* (Herman, 2016). Solusi yang dihasilkan oleh *ant colony system* dan *cat swarm optimization* cukup baik walaupun tidak sebaik algoritma pembandingnya. Berbeda dengan *cuckoo search* dan *tabu search* yang dapat memberikan hasil yang mendekati solusi optimal secara global. Dari 9 kasus KSP yang diujikan, *cuckoo search* menghasilkan solusi optimal pada 4 kasus, sedangkan *tabu search* menghasilkan solusi optimal pada 6 kasus. Hal ini menunjukkan bahwa terdapat kemungkinan bagi algoritma metaheuristik lain untuk dapat menghasilkan solusi yang baik dalam penerapannya pada KSP.

Dragonfly algorithm merupakan salah satu algoritma metaheuristik yang belum pernah digunakan untuk menyelesaikan KSP. Algoritma ini pertama kali dikenalkan oleh Seyedali Mirjalili pada tahun 2015. Algoritma ini terinspirasi dari perilaku berkerumun dari capung yang terbagi menjadi 2 jenis, yaitu *static swarm* dan *dynamic swarm*. Kedua perilaku tersebut menjadi dasar pengembangan karena mirip dengan 2 fase pada optimisasi metaheuristik, yaitu ekploitasi dan eksplorasi. Untuk dapat memperoleh solusi optimal, *dragonfly algorithm* memiliki 5 parameter dalam menentukan posisi individu capung dalam kerumunan, yaitu *separation*, *alignment*, *cohesion*, *attraction to food*, dan *distraction from enemy*. Pengaturan kelima parameter tersebut sangat penting untuk menyeimbangkan kemampuan eksploitasi dan eksplorasi dari algoritma. Oleh karena itu, pada penelitian ini juga diidentifikasi pengaruh parameter terhadap performansinya dalam menyelesaikan suatu permasalahan.

Bila dilihat dari jenisnya, algoritma metaheuristik yang digunakan untuk menyelesaikan KSP mayoritas merupakan *swarm intelligence* (SI). Algoritma yang termasuk ke dalam SI adalah *ant colony system*, *cat swarm optimization*, dan *cuckoo search*. SI merupakan salah satu cabang dari algoritma metaheuristik berbasis populasi yang pengembangannya terinspirasi oleh perilaku berkerumun makhluk hidup. Algoritma SI banyak dikembangkan karena memiliki beberapa kelebihan, seperti sering menggunakan daya ingatnya untuk menyimpan solusi terbaik yang diperoleh sejauh ini, biasanya memiliki parameter yang lebih sedikit, dan mudah diimplementasikan (Mirjalili, Mirjalili, dan Lewis, 2014). *Dragonfly algorithm* sendiri terklasifikasi sebagai algoritma SI sehingga algoritma ini dapat menjadi salah satu alternatif penyelesaian KSP.

Awalnya *dragonfly algorithm* dikembangkan untuk menyelesaikan permasalahan yang memiliki ruang solusi kontinu. Seyedali Mirjalili kemudian mengembangkan versi biner dari *dragonfly algorithm* yang disebut *binary dragonfly algorithm* (BDA) untuk menyelesaikan permasalahan kombinatorial. BDA pernah diterapkan untuk menyelesaikan 0-1 *knapsack problem* (Basset, Luo, Miao, & Zhou, 2017). Berdasarkan penelitian yang dilakukan oleh Basset et al. (2017), BDA menghasilkan solusi dengan tingkat konvergensi yang kuat dan stabil pada 0-1 *knapsack problem*. Dalam penelitian tersebut, BDA juga digunakan untuk menyelesaikan *hard multidimensional knapsack problem* dengan algoritma pembanding *binary cuckoo search*, *quantum inspired cuckoo search*, dan *particle swarm optimization*. Dari 10 kasus yang diujikan, BDA berhasil menghasilkan solusi optimal pada semua kasus.

Berdasarkan identifikasi masalah yang telah dipaparkan sebelumnya, maka dapat dibuat rumusan masalah sebagai berikut.

1. Bagaimana penerapan *dragonfly algorithm* terhadap penyelesaian *knapsack sharing problem*?
2. Bagaimana pengaruh parameter *dragonfly algorithm* terhadap performansinya dalam menyelesaikan permasalahan *knapsack sharing problem*?
3. Bagaimana hasil perbandingan performansi *dragonfly algorithm* dengan *cat swarm optimization* (Herman, 2016), *cuckoo search* (Angga, 2014), dan *tabu search* (Hifi et al., 2002) dalam menyelesaikan permasalahan *knapsack sharing problem*?

I.3 Pembatasan Masalah

Pembatasan masalah yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Permasalahan yang digunakan dalam penelitian ini diambil dari beberapa kasus *benchmark* pada penelitian Hifi & Sadfi (2002).
2. Ukuran performansi algoritma tidak dilihat dari waktu penyelesaian suatu permasalahan karena adanya pengaruh spesifikasi komputer dan *software* yang digunakan dalam pembuatan program.

I.4 Tujuan Penelitian

Berdasarkan identifikasi dan rumusan masalah yang telah dibuat, tujuan yang ingin dicapai melalui penelitian ini adalah sebagai berikut.

1. Menerapkan *dragonfly algorithm* untuk menyelesaikan *knapsack sharing problem*.
2. Mengetahui pengaruh parameter *dragonfly algorithm* terhadap performansinya dalam menyelesaikan *knapsack sharing problem*.
3. Membandingkan performansi *dragonfly algorithm* dengan *cat swarm optimization* (Herman, 2016), *cuckoo search* (Angga, 2014), dan *tabu search* (Hifi et al., 2002) dalam menyelesaikan *knapsack sharing problem*.

I.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat kepada berbagai pihak. Manfaat yang ingin diperoleh melalui penelitian ini adalah sebagai berikut.

1. Mengetahui apakah penerapan *dragonfly algorithm* pada *knapsack sharing problem* dapat memberikan solusi yang baik.
2. Meningkatkan pengetahuan pembaca mengenai *dragonfly algorithm* terutama penerapannya dalam penyelesaian *knapsack sharing problem*.
3. Menambah referensi penelitian yang berhubungan dengan penerapan *dragonfly algorithm* dan studi permasalahan yang termasuk ke dalam *knapsack sharing problem*.

I.6 Metodologi Penelitian

Dalam melakukan suatu penelitian, dibutuhkan suatu metodologi agar penelitian dapat lebih teratur dan sistematis. Gambar I.1 merupakan *flowchart*

metodologi penelitian yang digunakan dalam penerapan *dragonfly algorithm* untuk menyelesaikan *knapsack sharing problem*.

1. Studi Literatur

   Studi literatur merupakan tahap awal dari penelitian ini. Studi literatur dilakukan untuk mengumpulkan informasi-informasi yang berkaitan dengan *knapsack sharing problem* dan *dragonfly algorithm*.

2. Identifikasi dan Perumusan Masalah

   Identifikasi masalah berisi pemaparan permasalahan-permasalahan yang berkaitan dengan *knapsack sharing problem* dan *dragonfly algorithm*. Dari hasil identifikasi tersebut dibuat rumusan masalah yang menjadi fokus penelitian.

3. Pembatasan Masalah

   Pembatasan masalah dilakukan agar penelitian yang dilakukan terfokus pada permasalahan yang diteliti.

4. Penetapan Tujuan dan Manfaat Penelitian

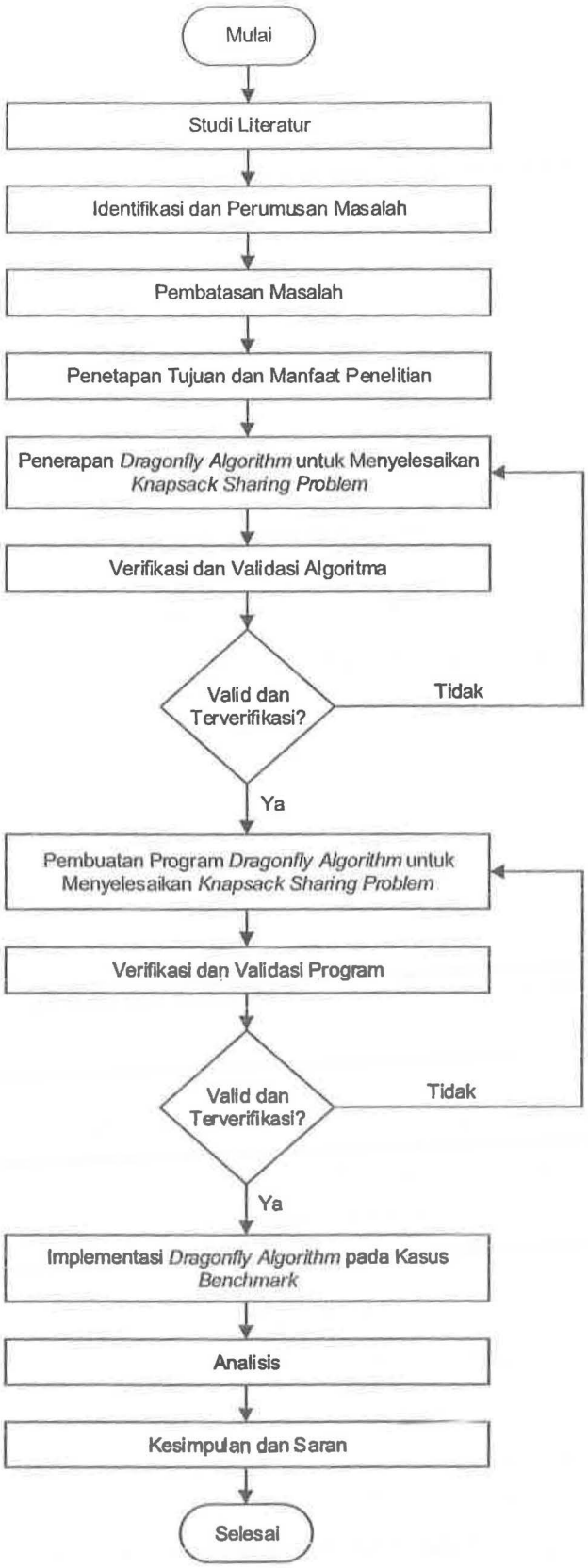
   Pada tahap ini dilakukan penetapan tujuan dan manfaat dari penelitian yang dilakukan. Penetapan ini perlu dilakukan agar penelitian memiliki tujuan dan manfaat yang jelas.

5. Penerapan *Dragonfly Algorithm* untuk Menyelesaikan *Knapsack Sharing Problem*

   Tahapan ini berisikan penerapan *dragonfly algorithm* pada *knapsack sharing problem*. Penerapan algoritma ke dalam permasalahan diawali dengan penetapan metode *encoding* dan *decoding*. *Encoding* merupakan proses representasi solusi suatu permasalahan ke dalam solusi pada algoritma, sedangkan *decoding* merupakan proses penerjemahan solusi dari dunia algoritma menjadi solusi suatu permasalahan.

6. Verifikasi dan Validasi Algoritma

   Tahapan ini berisikan proses verifikasi dan validasi dari *dragonfly algorithm* yang telah diterapkan pada *knapsack sharing problem*. Verifikasi dilakukan dengan memeriksa apakah prosedur pelaksanaan algoritma yang digambarkan dengan bantuan *flowchart* telah sesuai dengan konsep dasar algoritma. Validasi dilakukan untuk memeriksa apakah algoritma dapat menghasilkan solusi yang tepat pada permasalahan yang diujikan.



Gambar I.1 *Flowchart* Metodologi Penelitian

7. Pembuatan Program *Dragonfly Algorithm* untuk Menyelesaikan *Knapsack Sharing Problem*

   Pada tahapan ini dilakukan pembuatan program berdasarkan *flowchart dragonfly algorithm* untuk menyelesaikan *knapsack sharing problem* yang telah terverifikasi dan tervalidasi.

8. Verifikasi dan Validasi Program

   Pada tahap ini dilakukan proses verifikasi dan validasi terhadap program *dragonfly algorithm* untuk menyelesaikan *knapsack sharing problem* yang telah dibuat. Verifikasi dilakukan untuk memastikan apakah program yang dibuat telah sesuai dengan langkah-langkah algoritma pada *flowchart*. Validasi dilakukan untuk memastikan apakah program dapat menghasilkan solusi pada *knapsack sharing problem* tanpa melanggar batasan-batasan yang ada.

9. Implementasi *Dragonfly Algorithm* pada Kasus *Benchmark*

   Pada tahap ini dilakukan implementasi *dragonfly algorithm* terhadap kasus *benchmark*.

10. Analisis

    Analisis dilakukan untuk melihat hasil penerapan *dragonfly algorithm* terhadap penyelesaian *knapsack sharing problem*, pengaruh parameter terhadap performansi *dragonfly algorithm*, dan hasil perbandingan *dragonfly algorithm* dengan algoritma metaheuristik lainnya yang pernah diterapkan untuk menyelesaikan *knapsack sharing problem*.

11. Kesimpulan dan Saran

    Tahap ini berisikan kesimpulan dari hasil penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

I.7 Sistematika Penulisan

Penelitian yang dilakukan disusun dalam sistematika penulisan yang terstruktur dan sistematis. Berikut merupakan sistematika penulisan dari penelitian yang dilakukan.

BAB I PENDAHULUAN

Pada bab ini dijelaskan latar belakang masalah, identifikasi dan rumusan masalah, asumsi dan batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini dijelaskan mengenai dasar teori yang digunakan dalam penelitian. Dasar teori yang digunakan berhubungan dengan *knapsack sharing problem*, *dragonfly algorithm*, dan desain eksperimen.

BAB III PENERAPAN ALGORITMA

Pada bab ini dijelaskan mengenai *encoding* dan *decoding*, langkah kerja *dragonfly algorithm*, penerapan *dragonfly algorithm* untuk menyelesaikan *knapsack sharing problem*, dan verifikasi dan validasi algoritma.

BAB IV IMPLEMENTASI ALGORITMA

Pada bab ini dijelaskan mengenai implementasi *dragonfly algorithm* untuk menyelesaikan *knapsack sharing problem*. Hal-hal yang dibahas meliputi verifikasi dan validasi program, penentuan nilai parameter *dragonfly algorithm*, implementasi *dragonfly algorithm* pada kasus *benchmark*, pengujian pengaruh parameter *dragonfly algorithm*, dan perbandingan performansi *dragonfly algorithm* dengan *cat swarm optimization*, *cuckoo search*, dan *tabu search*.

BAB V ANALISIS

Pada bab ini dijelaskan analisis *encoding* dan *decoding*, analisis algoritma peningkatan dan perbaikan solusi, analisis perpindahan posisi capung, analisis pengaruh parameter *dragonfly algorithm*, dan analisis perbandingan performansi *dragonfly algorithm*.

BAB VI KESIMPULAN DAN SARAN

Pada bab ini dijelaskan mengenai kesimpulan dari hasil penerapan *dragonfly algorithm* untuk menyelesaikan *knapsack sharing problem* dan saran yang diberikan untuk penelitian selanjutnya.