

**PENERAPAN ALGORITMA *SHARK SMELL*
OPTIMIZATION DALAM MENYELESAIKAN
*KNAPSACK SHARING PROBLEM***

SKRIPSI

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

Disusun oleh:

Nama : Riska Sutanto Silvan

NPM : 2014610131



**PROGRAM STUDI TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS KATOLIK PARAHYANGAN
BANDUNG
2018**



**FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS KATOLIK PARAHYANGAN
BANDUNG**



Nama : Riska Sutanto Silvan
NPM : 2014610131
Jurusan : Teknik Industri
Judul Skripsi : PENERAPAN ALGORITMA SHARK SMELL OPTIMIZATION
DALAM MENYELESAIKAN KNAPSACK SHARING PROBLEM

TANDA PERSETUJUAN SKRIPSI

Bandung, Juli 2018

Ketua Jurusan Teknik Industri

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

Pembimbing Pertama

(Alfan, S.T., M.T.)



Jurusan Teknik Industri
Fakultas Teknologi Industri
Universitas Katolik Parahyangan



Pernyataan Tidak Mencontek atau Melakukan Tindakan Plagiat

Saya, yang bertanda tangan di bawah ini,

Nama : Riska Sutanto Silvan

NPM : 2014610131

dengan ini menyatakan bahwa skripsi dengan judul :

“PENERAPAN ALGORITMA SHARK SMELL OPTIMIZATION DALAM MENYELESAIKAN KNAPSACK SHARING PROBLEM”

adalah hasil pekerjaan saya dan seluruh ide, pendapat atau 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, Juli 2018

Riska Sutanto Silvan
2014610131

ABSTRAK

Knapsack Sharing Problem (KSP) merupakan salah satu pemodelan masalah pengalokasian sumber daya yang terbatas. KSP memodelkan masalah dengan tujuan untuk menentukan kombinasi objek terbaik yang hendak dimasukkan ke dalam *knapsack* agar setiap *sub-knapsack* (kelas) memiliki keuntungan yang sama rata dimana pada setiap *sub-knapsack* sudah terdapat objek apa saja yang dapat dipilih.

Penelitian ini menggunakan Algoritma *Shark Smell Optimization* (SSO) untuk menyelesaikan permasalahan *Knapsack Sharing Problem*. Algoritma SSO terinspirasi dari perilaku hiu yang unggul saat memangsa dan kemampuannya untuk menemukan mangsa pada area yang luas dalam waktu yang cepat berdasarkan kemampuan penciumannya yang kuat. Terdapat dua pergerakan pada Algoritma SSO yaitu *forward movement* yang merupakan proses eksplorasi dan *rotational movement* yang merupakan proses eksploitasi.

Algoritma SSO yang telah dirancang diterapkan pada 9 kasus KSP dengan 8 kombinasi parameter yang berbeda. Parameter koefisien kecepatan dan koefisien inersia berpengaruh pada semua kasus, sedangkan parameter pembatas kecepatan hanya berpengaruh pada kasus A05.1, A05C.1, B05C.1 dan B20.1. Selain itu terdapat pula interaksi antar parameter pada semua kasus kecuali kasus A20.1 dan B20.1. Solusi yang diperoleh dari semua kasus mendekati solusi optimal dengan selisih terkecil yaitu sebesar 0,02% dan selisih terbesar yaitu 2,61%. Solusi yang dihasilkan dibandingkan dengan Algoritma *Cat Swarm Optimization* (CSO), Algoritma *Cuckoo Search* (CS) dan Algoritma *Tabu Search* (TS). SSO menunjukkan performansi yang lebih unggul dibandingkan dengan CSO. Sedangkan dibandingkan dengan CS, SSO memiliki performansi yang lebih unggul pada kasus dengan jumlah objek sebanyak 2500, namun lebih buruk pada kasus dengan jumlah objek sebanyak 1000. Dibandingkan dengan TS, SSO menunjukkan performansi yang sama hanya pada 4 kasus, sehingga TS lebih unggul dari SSO.

ABSTRACT

Knapsack Sharing Problem (KSP) is one of the limited resource allocation problem modeling. KSP modeled the problem in order to determine the best combination of objects to be inserted into the knapsack so that each sub-knapsack (class) has the same benefit where each sub-knapsack already has any object to choose from.

This research uses Shark Smell Optimization (SSO) algorithm to solve Knapsack Sharing Problem. The SSO algorithm is inspired by superior shark behavior during prey and its ability to find prey on large areas in quick time based on its strong sense of smell. There are two movements in the SSO algorithm that is forward movement which is the process of exploration and rotational movement which is the process of exploitation.

The designed SSO algorithm is applied to 9 KSP cases with 8 combinations of different parameters. The velocity coefficient and inertial coefficient parameters affect all cases, whereas the speed limiting parameters only affect the cases of A05.1, A05C.1, B05C.1 and B20.1. In addition there is also interaction between parameters in all cases except cases A20.1 and B20.1. The solution obtained from all cases approached the optimal solution with the smallest difference of 0.02% and the largest difference of 2.61%. The resulting solution is compared to the Cat Swarm Optimization Algorithm (CSO), Cuckoo Search Algorithm (CS) and Tabu Search Algorithm (TS). SSO demonstrates superior performance compared to CSOs. While compared to CS, SSO has superior performance in cases with 2500 objects, but worse in cases with 1000 objects. Compared with TS, SSO shows the same performance in only 4 cases, so TS is superior to SSO .

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan penelitian yang berjudul Penerapan Algoritma *Shark Smell Optimization* Dalam Menyelesaikan *Knapsack Sharing Problem*. Selama penyusunan dan pembuatan ini, penulis mendapatkan berbagai dukungan, pengalaman dan juga bimbingan dari berbagai pihak. Maka dari itu pada kesempatan ini, penulis ingin mengucapkan terimakasih kepada berbagai pihak yang telah memberikan baik dukungan maupun motivasi kepada penulis baik secara langsung maupun tidak langsung, yaitu:

1. Bapak Alfian, S.T., M.T. sebagai dosen pembimbing yang telah banyak membantu dan memberikan masukan serta bimbingan dengan penuh kesabaran. Tanpa bimbingan dan nasihat bapak, penelitian ini tidak akan berjalan lancar.
2. Ibu Cynthia Prithadevi Juwono, Ir., M.S. sebagai dosen penguji proposal yang telah memberi banyak masukan dan nasihat pada penelitian ini dengan penuh kesabaran.
3. Ibu Titi Iswari, S.T., M.Sc., MBA sebagai dosen penguji proposal dan dosen penguji sidang skripsi yang telah memberi banyak masukan dan nasihat pada penelitian ini dengan penuh kesabaran.
4. Pak Dr. Sugih Sudharma Tjandra, S. T., M. Si. Sebagai dosen penguji sidang skripsi yang telah memberi banyak masukan dan nasihat pada penelitian ini dengan penuh kesabaran.
5. Papa, mama dan cici yang telah banyak memberikan doa, dukungan dan motivasi kepada penulis setiap saat.
6. Stefanie Puspa dan Aurea Hildani yang telah banyak memberikan dukungan dan motivasi kepada penulis dengan penuh kesabaran.
7. Rainer sebagai teman yang telah banyak membantu dalam proses pembuatan program pada penelitian ini.
8. Seluruh teman-teman :3.
9. Seluruh teman-teman TI 2014 kelas C.
10. Seluruh dosen yang mengajar di TI UNPAR.

11. Seluruh staf dan pekarya FTI UNPAR.
12. Dan semua pihak yang terlibat yang tidak bisa penulis sebutkan satu persatu.

Begitu banyak pengalaman yang penulis dapatkan selama proses pembuatan penelitian ini. Besar harapan penulis agar kelak penelitian ini dapat memberikan manfaat bagi pembaca. Penulis menyadari bahwa penelitian ini masih jauh dari sempurna. Oleh karena itu, penulis sangat terbuka terhadap kritik dan saran yang membangun yang dapat memberikan perbaikan pada penelitian ini sehingga dapat berguna di masa yang akan datang. Akhir kata, penulis ucapkan terima kasih.

Bandung, Juli 2018

Riska Sutanto Silvan

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR TABEL	ix
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	I-1
I.1 Latar Belakang Masalah.....	I-1
I.2 Identifikasi dan Perumusan Masalah.....	I-4
I.3 Pembatasan Masalah.....	I-6
I.4 Tujuan Penelitian.....	I-7
I.5 Manfaat Penelitian.....	I-7
I.6 Metodologi Penelitian	I-7
I.7 Sistematika Penulisan	I-10
BAB II TINJAUAN PUSTAKA	II-1
II.1 <i>Knapsack Sharing Problem</i>	II-1
II.2 Metaheuristik	II-2
II.3 <i>Shark Smell Optimization</i>	II-4
II.3.1 Menemukan Awal Partikel Bau	II-4
II.3.2 Pergerakan Hiu Menuju Mangsa	II-5
II.4 <i>Analysis of Variance (ANOVA)</i>	II-10
BAB III PERANCANGAN ALGORITMA	III-1
III.1 Deskripsi Model <i>Shark Smell Optimization (SSO)</i>	III-1
III.2 Bentuk Kode Solusi	III-3
III.3 Perancangan Algoritma SSO untuk Permasalahan KSP	III-4
III.3.1 Notasi Algoritma.....	III-4
III.3.2 Algoritma Utama <i>Shark Smell Optimization (SSO)</i>	III-7
III.3.3 Algoritma Pencarian Solusi Awal (A)	III-12

III.3.4	Algoritma Cek Kapasitas (B)	III-16
III.3.5	Algoritma Perbaiki Solusi (C)	III-17
III.3.6	Algoritma Perhitungan <i>Objective Function</i> (D)	III-21
III.3.7	Algoritma <i>Forward Movement</i> (E)	III-23
III.3.8	Algoritma <i>Rotational Movement</i> (F)	III-25
III.3.9	Algoritma <i>Position Update</i> (G)	III-28
III.4	Validasi Algoritma	III-30
III.4.1	Pencarian Solusi Awal	III-32
III.4.2	Pergerakan Hiu	III-36
BAB IV	PENERAPAN ALGORITMA	IV-1
IV.1	Verifikasi Program	IV-1
IV.2	Penentuan Parameter SSO	IV-1
IV.3	Penerapan Algoritma SSO pada Kasus <i>Benchmark</i>	IV-5
IV.3.1	Kasus A05.1	IV-5
IV.3.2	Kasus A05C.1	IV-6
IV.3.3	Kasus A20.1	IV-6
IV.3.4	Kasus A30.1	IV-7
IV.3.5	Kasus A50.1	IV-7
IV.3.6	Kasus B05C.1	IV-8
IV.3.7	Kasus B20.1	IV-8
IV.3.8	Kasus B30.1	IV-9
IV.3.9	Kasus B50.1	IV-9
IV.4	Perbandingan Solusi Terbaik	IV-9
IV.5	Hasil Pengujian Parameter SSO	IV-10
BAB V	ANALISIS	V-1
V.1	Analisis Bentuk Kode Solusi	V-1
V.2	Analisis Modifikasi Algoritma	V-2
V.3	Analisis Pengaruh Parameter	V-3
V.3.1	Analisis Pengaruh Koefisien Kecepatan (P1)	V-3
V.3.2	Analisis Pengaruh Koefisien Inersia (P2)	V-4
V.3.3	Analisis Pengaruh Pembatas Kecepatan (P3)	V-5
V.4	Analisis Perbandingan Performansi Algoritma	V-6

BAB VI KESIMPULAN DAN SARANVI-1

VI.1 KesimpulanVI-1

VI.2 SaranVI-2

DAFTAR PUSTAKA

LAMPIRAN

RIWAYAT HIDUP PENULIS

DAFTAR TABEL

Tabel III.1 Contoh Kode Solusi Permasalahan KSP	III-4
Tabel III.2 Data Permasalahan KSP yang Digunakan	III-30
Tabel III.3 <i>Ratio</i> Objek	III-32
Tabel III.4 Matriks Hitung UC Awal ($k=1,s=1$)	III-33
Tabel III.5 Matriks Hitung UC Awal ($k=1,s=2$)	III-34
Tabel III.6 Matriks Hitung UC Awal ($k=1,s=3$)	III-35
Tabel III.7 Matriks Hasil Perhitungan FM ($k=1,s=1$).....	III-37
Tabel III.8 Matriks Hasil Perhitungan RM ($k=1,s=1,m=1$)	III-38
Tabel III.9 Matriks Hasil Perhitungan RM ($k=1,s=1,m=2$)	III-40
Tabel III.10 Matriks Hasil Perhitungan PU ($k=1, s=1$).....	III-41
Tabel III.11 Matriks Hasil Perhitungan FM ($k=1,s=2$).....	III-42
Tabel III.12 Matriks Hasil Perhitungan RM ($k=1,s=2,m=1$)	III-43
Tabel III.13 Matriks Hasil Perhitungan RM ($k=1,s=2,m=2$)	III-44
Tabel III.14 Matriks Hasil Perhitungan PU ($k=1, s=2$).....	III-46
Tabel III.15 Matriks Hasil Perhitungan FM ($k=1,s=3$).....	III-47
Tabel III.16 Matriks Hasil Perhitungan RM ($k=1,s=3,m=1$)	III-48
Tabel III.17 Matriks Hasil Perhitungan RM ($k=1,s=3,m=2$)	III-50
Tabel III.18 Matriks Hasil Perhitungan PU ($k=1, s=3$).....	III-51
Tabel III.19 Matriks Hasil Perhitungan FM ($k=2,s=1$).....	III-52
Tabel III.20 Matriks Hasil Perhitungan RM ($k=2,s=1,m=1$)	III-53
Tabel III.21 Matriks Hasil Perhitungan RM ($k=2,s=1,m=2$)	III-55
Tabel III.22 Matriks Hasil Perhitungan PU ($k=2, s=1$).....	III-56
Tabel III.23 Matriks Hasil Perhitungan FM ($k=2,s=2$).....	III-57
Tabel III.24 Matriks Hasil Perhitungan RM ($k=2,s=2,m=1$)	III-58
Tabel III.25 Matriks Hasil Perhitungan RM ($k=2,s=2,m=2$)	III-59
Tabel III.26 Matriks Hasil Perhitungan PU ($k=2, s=2$).....	III-60
Tabel III.27 Matriks Hasil Perhitungan FM ($k=2,s=3$).....	III-61
Tabel III.28 Matriks Hasil Perhitungan RM ($k=2,s=3,m=1$)	III-63
Tabel III.29 Matriks Hasil Perhitungan RM ($k=2,s=3,m=2$)	III-64
Tabel III.30 Matriks Hasil Perhitungan PU ($k=2, s=3$).....	III-65

Tabel III.31 Hasil Akhir	III-66
Tabel IV.1 Perbandingan Algoritma Dengan Modul Program	IV-1
Tabel IV.2 Hasil Kombinasi Parameter A05.1	IV-6
Tabel IV.3 Hasil Kombinasi Parameter A05C.1	IV-6
Tabel IV.4 Hasil Kombinasi Parameter A20.1	IV-7
Tabel IV.5 Hasil Kombinasi Parameter A30.1	IV-7
Tabel IV.6 Hasil Kombinasi Parameter A50.1	IV-8
Tabel IV.7 Hasil Kombinasi Parameter B05C.1	IV-8
Tabel IV.8 Hasil Kombinasi Parameter B20.1	IV-9
Tabel IV.9 Hasil Kombinasi Parameter B30.1	IV-9
Tabel IV.10 Hasil Kombinasi Parameter B50.1	IV-10
Tabel IV.11 Hasil Perbandingan Solusi Terbaik	IV-10
Tabel IV.12 Rekapitulasi Nilai <i>P-value</i>	IV-11

DAFTAR GAMBAR

Gambar I.1 Metodologi Penelitian SSO untuk KSP	I-9
Gambar II.1 Pergerakan Rotasi Hiu	II-7
Gambar II.2 <i>Flowchart</i> SSO <i>Algorithm</i>	II-8
Gambar III.1 Contoh Bentuk Kode Solusi.....	III-3
Gambar III.2 Algoritma Utama <i>Shark Smell Optimization</i>	III-9
Gambar III.3 Algoritma A (Pencarian Solusi Awal).....	III-14
Gambar III.4 Algoritma B (Cek Kapasitas)	III-17
Gambar III.5 Algoritma C (Perbaikan Solusi)	III-19
Gambar III.6 Algoritma D (Perhitungan OF)	III-22
Gambar III.7 Algoritma E (<i>Forward Movement</i>).....	III-24
Gambar III.8 Algoritma F (<i>Rotational Movement</i>).....	III-27
Gambar III.9 Algoritma G (<i>Position Update</i>).....	III-29
Gambar IV.1 Modul Program pada Algoritma B	IV-2
Gambar IV.2 Verifikasi Program (η).....	IV-3
Gambar IV.3 Verifikasi Program (α).....	IV-3
Gambar IV.4 <i>Main Effect Plot</i> untuk A05.1	IV-12
Gambar IV.5 <i>Interaction Plot</i> untuk A05C.1	IV-13

DAFTAR LAMPIRAN

LAMPIRAN A HASIL *RUNNING* PROGRAM

LAMPIRAN B HASIL UJI ANOVA

LAMPIRAN C *MAIN EFFECT PLOT*

LAMPIRAN D *INTERACTION PLOT*

BAB I

PENDAHULUAN

Bab ini membahas tahap-tahap awal yang dilakukan dalam penelitian. Pembahasan tersebut mengenai latar belakang masalah, identifikasi dan perumusan masalah, pembatasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

I.1 Latar Belakang Masalah

Pengalokasian sumber daya dengan tepat merupakan hal yang penting. Hal tersebut dikarenakan keterbatasan sumber daya yang tersedia. Tak hanya masalah keterbatasan, namun tujuan tertentu yang ingin dicapai seringkali menjadi alasan pentingnya pengalokasian sumber daya. Tujuan yang ingin dicapai misalnya maksimasi keuntungan atau minimasi biaya. Oleh karena itu, perlu adanya perencanaan yang tepat dalam rangka mencapai tujuan tersebut. Salah satu contoh pengalokasian sumber daya adalah permasalahan dalam mengalokasikan investasi. Dalam kehidupan nyata, terdapat berbagai jenis investasi yang dapat kita pilih. Pada pengalokasian investasi, tujuan yang diinginkan adalah memperoleh keuntungan sebesar-sebesar. Maka dari itu, perlu diketahui bagaimana cara yang tepat untuk mengalokasikan sumber daya berupa uang ke dalam investasi yang beragam tersebut agar keuntungan yang diperoleh dapat sebesar-besarnya.

Salah satu pemodelan masalah pengalokasian sumber daya dikenal dengan *knapsack problem*. *Knapsack* dianalogikan sebagai sebuah wadah seperti kantung atau ransel yang memiliki kapasitas tertentu. Lalu *knapsack* akan diisi oleh beberapa objek yang memiliki bobot dan nilai yang berbeda-beda. Maka *knapsack problem* dapat diartikan sebagai kombinasi pemilihan objek yang terbaik yang ingin dimasukkan ke dalam kantung yang memiliki kapasitas tertentu sehingga memberikan hasil yang optimal sesuai dengan tujuan yang ingin dicapai.

Knapsack problem seringkali ditemukan dalam kehidupan sehari-hari seperti penentuan barang apa saja yang sebaiknya dimasukkan ke dalam truk ketika akan berpindah ke kota lain, barang apa saja yang sebaiknya dibawa saat

bepergian menggunakan pesawat terbang, dan sebagainya. *Knapsack problem* terbagi menjadi beberapa jenis yaitu *knapsack sharing problem* (Brown, 1979), *multiple-choice knapsack problem*, *bounded knapsack problem*, *unbounded knapsack problem*, *subset-sum problem*, *change-making problem* (Martello dan Toth, 1990), dan sebagainya.

Knapsack sharing problem membahas mengenai bagaimana cara mengalokasikan sebuah *single resource* (sumber daya tertentu) dimana setiap penerima potensial dari *resource* tersebut memiliki sebuah *utility* atau *tradeoff* (Brown, 1979). *Utility* atau *tradeoff* dapat diartikan sebagai keuntungan yang dapat diberikan oleh suatu objek kepada pembuat keputusan jika objek tersebut terpilih sebagai penerima *resource*. Pada *knapsack sharing problem* terdapat *knapsack* yang terbagi menjadi beberapa bagian (*sub-knapsack*). Pengalokasian ini membagi sumber daya yang terbatas kepada penerima dengan tujuan untuk memaksimalkan nilai minimum dari *sub-knapsack* (fungsi objektif maksimin). Permasalahan ini terjadi ketika sumber daya yang tersedia harus dialokasikan secara adil.

Contoh *knapsack sharing problem* adalah seorang ibu rumah tangga harus memutuskan bagaimana mengalokasikan sejumlah uang yang terbatas ke dalam berbagai tipe makanan yang berbeda dan seorang komandan harus memutuskan bagaimana mengalokasikan sumber daya yang terbatas seperti amunisi kepada pasukan dibawah pimpinannya (Brown, 1979). Seorang ibu rumah tangga ketika membeli makanan memiliki tujuan agar semua jenis nutrisi bagi keluarganya dapat terpenuhi. Misalkan keluarga tersebut terdiri dari ayah, ibu dan 1 anak. Setiap anggota keluarga memiliki keinginan masing-masing mengenai makanan yang ingin disantap. Uang yang dimiliki oleh ibu rumah tangga tersebut digambarkan sebagai sumber daya yang memiliki kapasitas yang terbatas. Nutrisi yang dimiliki oleh makanan digambarkan sebagai keuntungan dari makanan tersebut. Sedangkan harga beli makanan tersebut digambarkan sebagai bobot dari makanan tersebut. Oleh karena itu, ia perlu memutuskan bagaimana cara mengalokasikan uang yang terbatas tersebut dengan sama rata agar tujuan yang diinginkan dapat tercapai. Tujuan tersebut adalah ayah, ibu dan anak mendapatkan asupan gizi yang sama rata baik dari segi karbohidrat dll. Seorang komandan ketika membagikan amunisi kepada pasukan dibawah pimpinannya memiliki tujuan agar semua daerah/kota yang dijaga oleh pasukannya tersebut

memiliki tingkat pengawasan yang sama. Misalkan komandan tersebut mengawasi 2 daerah. Setiap daerah memiliki pasukan masing-masing yang dapat ditugaskan. Jumlah amunisi yang dimiliki oleh komandan digambarkan sebagai kapasitas yang terbatas. Keahlian pasukan dalam menggunakan amunisi untuk melindungi suatu daerah digambarkan sebagai keuntungan. Sedangkan seberapa banyak pasukan dapat membawa amunisi digambarkan sebagai bobot. Oleh karena itu, seorang komandan perlu memutuskan bagaimana cara menentukan pasukan yang ditugaskan di daerah tersebut dengan membawa amunisi sesuai kemampuan mereka sehingga manfaat yang dimiliki oleh setiap daerah sama.

Contoh lain permasalahan KSP dalam dunia nyata adalah *budget sharing* dalam dunia industri. Seorang pemilik perusahaan harus memutuskan pengalokasian sumber daya berupa anggaran yang terbatas untuk divisi-divisi pada perusahaannya seperti produksi, *marketing* dan HRD. Tujuan yang ingin dicapai oleh pemilik perusahaan adalah semua divisi dalam perusahaannya dapat berkembang sama rata dengan pembagian anggaran yang adil untuk semua divisi.

Knapsack sharing problem merupakan permasalahan *NP-hard* (*Non-deterministic Polynomial time hard*) (Yamada dan Futakawa, 1997). Hal tersebut berarti semakin luas ruang lingkup masalah yang diteliti maka penyelesaian dengan metode eksak tidak akan sesuai karena akan memakan waktu yang lama dan memerlukan usaha yang lebih sulit. KSP merupakan *NP-hard* yang termasuk dalam *combinatorial problem*. Oleh karena itu waktu yang dibutuhkan untuk menyelesaikan masalah dengan menggunakan metode eksak akan meningkat secara eksponensial. Salah satu contoh pentingnya waktu adalah ketika terjadi bencana kelaparan pada beberapa daerah. Pemerintah harus memutuskan cara pengalokasian sumber daya yang adil kepada daerah-daerah tersebut dalam waktu yang singkat. Agar semua daerah tersebut dapat memperoleh bantuan sama rata dengan waktu yang cepat. Oleh sebab itu muncullah alternatif metode untuk menyelesaikan *knapsack sharing problem* agar diperoleh penyelesaian yang lebih cepat.

I.2 Identifikasi dan Perumusan Masalah

Knapsack sharing problem memodelkan masalah dengan tujuan untuk menentukan kombinasi objek terbaik yang hendak dimasukkan ke dalam *knapsack* agar dapat memaksimalkan nilai minimum dari fungsi tujuan/*utility sub-knapsack*

dengan batasan kapasitas yang telah ditentukan. *Knapsack sharing problem* dapat memodelkan banyak permasalahan dalam kehidupan sehari-hari. Hal tersebut menunjukkan bahwa model *knapsack sharing problem* bermanfaat sehingga perlu diteliti dan dikembangkan.

Yamada, Futakawa dan Kataoka (1998) menggunakan metode eksak *branch and bound* serta *binary search algorithm* untuk menyelesaikan *knapsack sharing problem*. Penelitian tersebut menunjukkan *binary search algorithm* menghasilkan solusi yang lebih baik dari *branch and bound*. Hifi, Sadfi, dan Sbihi (2002) melakukan penelitian dan membandingkan metode eksak dengan metode lain *metaheuristic*. Penelitian tersebut menunjukkan bahwa metode eksak menyelesaikan masalah kombinatorial lebih lama dibandingkan dengan metode *metaheuristic*. Sehingga, penyelesaian dengan metode *metaheuristic* layak untuk dikembangkan.

Yamada dan Futakawa (1997) melakukan penelitian *knapsack sharing problem* dengan menggunakan *heuristic and reduction algorithm* yang dikenal dengan *greedy algorithm*. Namun metode heuristik tidak menjamin akan dihasilkan nilai yang optimal dan mungkin tidak *feasible* (Rayward-Smith, Osman, Reeves dan Smith, 1996). Penelitian dengan menggunakan metode metaheuristik telah dilakukan seperti *Tabu Search Algorithm* (Hifi, Sadfi, dan Sbihi, 2002), *Cuckoo Search Algorithm* (Angga, 2014) dan *Cat Swarm Optimization* (Herman, 2016). Metaheuristik adalah metodologi umum (*templates*) yang dapat digunakan sebagai panduan strategi dalam merancang heuristik untuk menyelesaikan masalah optimisasi (Talbi, 2009). Kelebihan dari metaheuristik adalah metode ini memberikan solusi yang dapat diterima (*acceptable*) dengan waktu yang masuk akal untuk menyelesaikan masalah yang sulit dan kompleks. Sedangkan kelemahan dari metaheuristik adalah metode ini tidak menjamin diperolehnya hasil yang optimal. Beberapa penelitian yang telah dilakukan terkait dengan metode metaheuristik terhadap permasalahan *knapsack sharing problem* telah disebutkan sebelumnya. Setelah dilakukan perbandingan maka *Tabu Search Algorithm* dan *Cuckoo Search Algorithm* lebih unggul dibandingkan dengan *Cat Swarm Optimization* (Herman, 2016). Walaupun kedua metode tersebut lebih unggul, diketahui masih terdapat kelemahan pada kedua metode tersebut. *Tabu Search Algorithm* memiliki kelemahan yaitu hasil akhir yang diperoleh dapat berupa *local optimum*. Hal tersebut disebabkan oleh proses pencarian algoritma tersebut

bersifat *neighborhood search* (pencarian di daerah yang berdekatan). Sedangkan *Cuckoo Search Algorithm* memiliki kelemahan yaitu tidak ada batasan yang jelas untuk tahap eksplorasi dan eksploitasi serta besar kemungkinan untuk algoritma terjebak dalam *local optimum*. *Local optimum* berarti keoptimalan dari solusi yang dihasilkan hanya berlaku untuk area di sekitarnya. Oleh karena itu, penelitian lebih lanjut perlu dilakukan untuk mendapatkan metode yang lebih baik dari metode yang telah dilakukan sebelumnya.

Shark Smell Optimization (SSO) diperkenalkan pertama kali pada tahun 2014 oleh Abedinia, Amjady dan Ghasemi. SSO dirancang berdasarkan pada kemampuan dari hiu sebagai pemburu yang unggul dalam menemukan mangsa, dimana kemampuan tersebut diperoleh dari penciuman hiu dan pergerakannya menuju sumber bau (Abedinia, Amjady, dan Ghasemi, 2014). Menurut Abedinia et al. (2014), SSO dapat menemukan area-area yang berbeda dari area solusi (kemampuan eksplorasi yang tinggi) serta mencari dalam area-area tersebut dengan resolusi yang tinggi (kemampuan eksploitasi yang tinggi). Sebagai algoritma yang baru diperkenalkan, algoritma SSO belum banyak digunakan untuk memecahkan permasalahan optimasi. Dalam jurnal penelitian Gnanasekaran, Chandramohan, Sathish dan Mohamed (2016), SSO digunakan untuk memecahkan permasalahan optimisasi yaitu *optimal placement of capacitors in radial distribution system*. Penelitian tersebut membandingkan performansi SSO dengan *Gravitational Search Algorithm* (GSA), *Cuckoo Search Algorithm* (CSA), *Harmony Search Algorithm* (HSA) dan *Artificial Bee Colony* (ABC) *Algorithm*. Hasil penelitian menunjukkan bahwa algoritma SSO memiliki performansi yang lebih baik dibandingkan algoritma lainnya. Selain itu, dalam jurnalnya Abedinia et al. (2014) menyelesaikan beberapa fungsi *benchmark* dan membandingkan performansi algoritma SSO dengan metode metaheuristik lainnya seperti *Genetic Algorithm*, *improved Genetic Algorithm*, *Particle Swarm Optimization*, *Harmony Search Algorithm*, *improved Harmony Search Algorithm*, *Honey Bee Optimization* dan lain-lain. Penelitian tersebut menunjukkan bahwa secara keseluruhan algoritma SSO memiliki performansi yang lebih unggul dibandingkan dengan algoritma lain dengan menghasilkan solusi yang memiliki akurasi yang tinggi, *robust* dan efisien dari segi komputasi. Dalam jurnalnya, Jamil dan Yang (2013) mengatakan bahwa *test function* merupakan salah satu hal yang penting untuk membandingkan performansi antar algoritma. Oleh karena itu, dengan kelebihan yang dimiliki oleh

Shark Smell Optimization maka SSO akan digunakan dalam penelitian ini dengan harapan dapat mengatasi kekurangan dari metode atau penelitian sebelumnya dalam menyelesaikan *knapsack sharing problem*.

Dalam pengimplementasian algoritma, ditemui berbagai parameter yang harus ditentukan nilainya dan mungkin berpengaruh terhadap performansi dari algoritma tersebut. Oleh karena itu, perlu diketahui juga bagaimana pengaruh parameter-parameter ini pada performansi yang dihasilkan. Selain itu, untuk menunjukkan adanya perbaikan pada performansi dengan menggunakan metode SSO, maka hasil penelitian ini akan dibandingkan dengan penelitian sebelumnya yaitu *Tabu Search Algorithm* (Hifi et al., 2002), *Cuckoo Search Algorithm* (Angga, 2014) dan *Cat Swarm Optimization* (Herman, 2016).

Pada penelitian ini akan digunakan *Shark Smell Optimization* untuk menyelesaikan masalah *knapsack sharing problem* yang diharapkan akan memberikan solusi yang lebih baik dari algoritma-algoritma yang telah diterapkan sebelumnya. Berdasarkan identifikasi masalah yang telah disebutkan, maka masalah dalam penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana perancangan dan penerapan algoritma *Shark Smell Optimization* yang sesuai dengan *Knapsack Sharing Problem*?
2. Apa parameter yang berpengaruh terhadap performansi algoritma *Shark Smell Optimization*?
3. Bagaimana hasil perbandingan solusi yang diperoleh antara algoritma *Shark Smell Optimization*, *Tabu Search Algorithm* (Hifi et al., 2002), *Cuckoo Search Algorithm* (Angga, 2014) dan *Cat Swarm Optimization* (Herman, 2016) dalam menyelesaikan *Knapsack Sharing Problem*?

1.3 Pembatasan Masalah

Agar penelitian dapat terpusat pada fokus penelitian maka penerapan algoritma *Shark Smell Optimization* pada *Knapsack Sharing Problem* memerlukan adanya pembatasan masalah. Pembatasan masalah yang digunakan adalah sebagai berikut:

1. Kasus yang digunakan untuk penerapan dan perbandingan merupakan kasus *benchmark* yang digunakan juga pada algoritma pembanding.
2. Waktu bukan merupakan ukuran performansi dikarenakan spesifikasi komputer yang berbeda-beda.

I.4 Tujuan Penelitian

Berdasarkan identifikasi dan perumusan masalah yang telah dijelaskan, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang algoritma dengan model *Shark Smell Optimization* untuk menyelesaikan *Knapsack Sharing Problem*.
2. Mengetahui parameter-parameter yang berpengaruh terhadap performansi algoritma *Shark Smell Optimization*.
3. Membandingkan hasil solusi algoritma *Shark Smell Optimization* terhadap solusi yang diperoleh dari *Tabu Search Algorithm* (Hifi et al., 2002), *Cuckoo Search Algorithm* (Angga, 2014) dan *Cat Swarm Optimization* (Herman, 2016).

I.5 Manfaat Penelitian

Penelitian yang dilakukan diharapkan dapat memiliki manfaat bagi beberapa pihak terkait sebagai berikut:

1. Menambah pengetahuan pembaca mengenai penerapan metode *Shark Smell Optimization* dalam *Knapsack Sharing Problem*.
2. Menambah referensi untuk penelitian yang berkaitan dengan algoritma *Shark Smell Optimization* dan *Knapsack Sharing Problem*.

I.6 Metodologi Penelitian

Dalam penelitian ini dirancang sebuah metodologi penelitian yang menunjukkan langkah-langkah penelitian yang akan digunakan. Dimulai dari studi literatur hingga kesimpulan dan saran. *Flowchart* metodologi penelitian terdapat pada Gambar I.1.

1. Studi Literatur Permasalahan Kombinatorial dan Metaheuristik
Studi literatur berkaitan dengan pengumpulan informasi mengenai masalah yang diteliti serta metode yang digunakan untuk penelitian. Pengumpulan informasi ini berupa buku atau penelitian mengenai permasalahan kombinatorial serta metode-metode metaheuristik tersebut.

2. Identifikasi dan Perumusan Masalah

Identifikasi masalah terkait dengan permasalahan-permasalahan yang timbul. Dari identifikasi masalah tersebut, maka ditentukan perumusan masalah agar penelitian lebih terarah dan terfokus.

3. Pembatasan Masalah

Pada tahap ini dilakukan pembatasan masalah terhadap penelitian yang dilakukan. Hal ini dilakukan agar penelitian lebih terfokus pada masalah yang dibahas.

4. Penentuan Tujuan dan Manfaat Penelitian

Tujuan penelitian yang ditentukan menjawab perumusan masalah yang telah ditentukan sebelumnya. Tujuan penelitian memberikan arahan dalam penelitian. Manfaat penelitian berisi kegunaan penelitian ini untuk penelitian berikutnya maupun pihak-pihak yang terkait.

5. Perancangan Algoritma SSO untuk KSP

Pada tahap ini dilakukan perancangan algoritma dengan menggunakan metode *Shark Smell Optimization* dan mengembangkannya dalam rangka penyelesaian *Knapsack Sharing Problem*.

6. Validasi Algoritma

Pada tahap ini dilakukan validasi terhadap rancangan algoritma yang telah dibuat dengan menggunakan metode *Shark Smell Optimization*. Validasi dilakukan untuk memastikan algoritma yang dirancang sesuai dengan langkah yang sebenarnya dan memiliki proses berpikir yang benar.

7. Perancangan Program

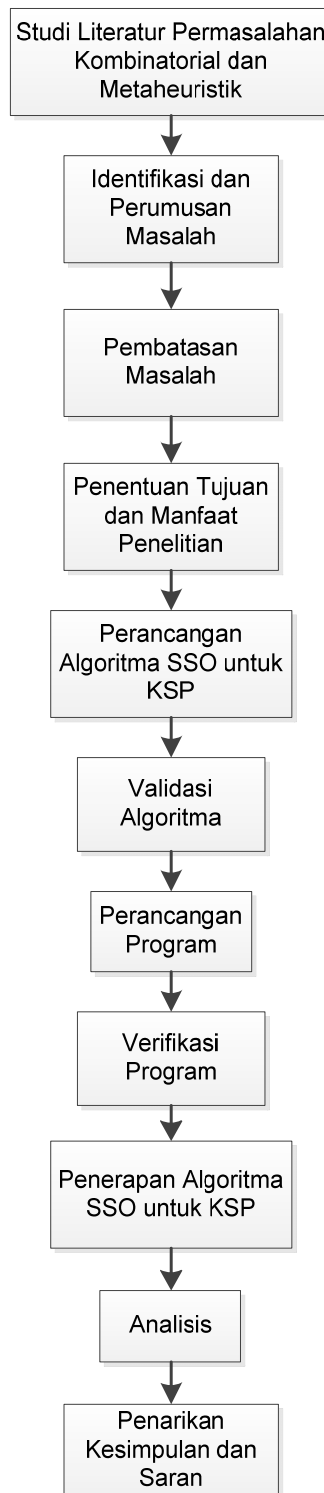
Pada tahap ini, dilakukan perancangan program berdasarkan algoritma yang telah dirancang sebelumnya yaitu algoritma *Shark Smell Optimizatio* dengan menggunakan Netbeans IDE 8.0.2.

8. Verifikasi Program

Pada tahap ini dilakukan verifikasi terhadap program yang telah dibuat sebelumnya. Verifikasi dilakukan untuk memastikan program yang dibuat sesuai dengan program yang telah dirancang.

9. Penerapan Algoritma SSO untuk KSP

Setelah program yang dirancang diverifikasi, maka langkah selanjutnya adalah menerapkan algoritma yang telah dirancang dalam program tersebut secara langsung terhadap *knapsack sharing problem*.



Gambar I.1 Metodologi Penelitian SSO untuk KSP

10. Analisis

Pada tahap ini dilakukan evaluasi terhadap algoritma yang telah dirancang, parameter yang berpengaruh serta hasil perbandingan algoritma SSO dengan algoritma pembandingan.

11. Penarikan Kesimpulan dan Saran

Pada tahap ini dibuat kesimpulan dari penelitian yang telah dilakukan. Selain itu, dibuat pula saran untuk penelitian selanjutnya agar diperoleh hasil yang lebih baik.

I.7 Sistematika Penulisan

Sub-bab ini menjelaskan mengenai sistematika penulisan yang digunakan pada penelitian ini. Sistematika penulisan pada penelitian ini terdiri dari enam bab yaitu sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang masalah yang kemudian lebih difokuskan pada identifikasi dan perumusan masalah. Selain itu, dibahas pula pembatasan masalah yang digunakan dalam penelitian, tujuan penelitian, manfaat penelitian, metodologi penelitian yang digunakan serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi teori-teori yang terkait dan digunakan dalam penelitian. Bab ini berisi pembahasan mengenai permasalahan yang dibahas yaitu *Knapsack Sharing Problem*. Selain itu, terdapat pula pembahasan algoritma yang digunakan yaitu *Shark Smell Optimization*.

BAB III PERANCANGAN ALGORITMA

Bab ini membahas mengenai langkah-langkah pembuatan algoritma SSO untuk permasalahan KSP. Langkah-langkah tersebut disajikan dengan menggunakan diagram alir. Setelah itu, dilakukan validasi terhadap algoritma yang telah dirancang.

BAB IV PENERAPAN ALGORITMA

Bab ini membahas mengenai penerapan algoritma terhadap permasalahan KSP. Penerapan tersebut dilakukan dengan menggunakan program yang dibuat berdasarkan algoritma yang telah

dirancang. Namun, sebelum dilakukan penerapan pada kasus, dilakukan verifikasi program terlebih dahulu.

BAB V ANALISIS

Bab ini berisi analisis keseluruhan dari hasil penelitian. Analisis tersebut meliputi analisis hasil performansi SSO dan parameter yang berpengaruh terhadap performansi SSO. Selain itu, terdapat pula perbandingan solusi dari penyelesaian *Knapsack Sharing Problem* dengan menggunakan algoritma SSO dan algoritma lain.

BAB VI KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang didapat dari hasil penelitian mengenai Penerapan Algoritma *Shark Smell Optimization* Dalam Menyelesaikan *Knapsack Sharing Problem*. Selain itu, terdapat pula saran yang dapat diberikan untuk penelitian selanjutnya.