

# **PENERAPAN *BINARY BAT ALGORITHM* DALAM PENYELESAIAN *KNAPSACK SHARING PROBLEM***

## **SKRIPSI**

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

Disusun oleh:

Nama : Demasdika Adhiwane

NPM : 2015610100



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

## ABSTRAK

Dalam kehidupan nyata permasalahan kelangkaan sumber daya sering terjadi sehingga pengalokasian sumber daya yang baik diperlukan untuk mendapatkan keuntungan yang maksimal dari sumber daya yang terbatas tersebut. Permasalahan alokasi sumber daya dapat dimodelkan dalam *Knapsack Sharing Problem*. *Knapsack Sharing Problem* (KSP) adalah permasalahan pencarian kombinasi benda dengan keuntungan dan berat masing-masing untuk dimasukkan ke dalam wadah dengan kapasitas terbatas dengan tujuan memaksimalkan keuntungan grup benda terkecil. Kapasitas yang terbatas tersebut merupakan sumber daya sedangkan benda merupakan penggunaan sumber daya.

Pada penelitian ini *Binary Bat Algorithm* (BBA) digunakan untuk menyelesaikan permasalahan KSP. *Binary Bat Algorithm* adalah salah satu algoritma metaheuristik yang terinspirasi dari pergerakan dan tingkah laku kelelawar dalam mencari mangsa. Setiap kelelawar memiliki 3 fase pergerakan yang dimulai dari pergerakan dengan mengubah frekuensi, pergerakan *random walk*, dan pergerakan *random search*. BBA dapat secara otomatis menyeimbangkan eksplorasi dan eksploitasinya dengan mengubah parameter, BBA juga menggunakan *v-shaped transfer function* untuk meningkatkan solusi. Pada BBA terdapat 5 parameter yaitu frekuensi maksimal, *loudness* awal, *pulse rate* awal, serta alfa dan gamma yaitu konstanta perubahan *loudness* dan *pulse rate*.

Pada penelitian ini *Binary Bat Algorithm* dirancang untuk dapat menyelesaikan permasalahan KSP dan selanjutnya diterapkan pada 10 kasus *benchmark* KSP. Parameter yang paling berpengaruh adalah *pulse rate* yang berpengaruh pada 8 kasus dan dilanjutkan dengan frekuensi maksimal dan *loudness*. Alfa dan gamma hanya berpengaruh pada 1 kasus. Selain itu terdapat pula interaksi antar parameter. Interaksi yang paling banyak berpengaruh adalah interaksi antara frekuensi maksimal dan *pulse rate* yang berpengaruh pada 6 kasus *benchmark*. Solusi terbaik yang diperoleh BBA dalam setiap kasus dibandingkan dengan 4 algoritma metaheuristik lainnya yaitu *Dragonfly Algorithm* (DA), *Cat Swarm Optimization* (CSO), *Tabu Search* (TS), dan *Shark Smell Optimization* (SSO). BBA berhasil mencapai solusi optimal pada 6 kasus *benchmark* yang diujikan. Pada kasus yang tidak mencapai optimal dihasilkan solusi yang mendekati optimal dengan *error* terbesar 0,85% dan *error* terkecil 0,05% yang masih berada di bawah DA dan TS tetapi berada di atas SSO dan CSO.

## **ABSTRACT**

*In real life the problem of scarcity of resources often occurs so that the allocation of good resources is needed to get the maximum benefit from these limited resources. The problem of resource allocation can be modeled in the Knapsack Sharing Problem. Knapsack Sharing Problem (KSP) is the problem of finding a combination of objects with their own profit and weight to be included in a container with limited capacity with the aim of maximizing the smallest object group profits.*

*In this study Binary Bat Algorithm (BBA) was used to solve KSP problems. Binary Bat Algorithm is one of the metaheuristic algorithms that is inspired by the movement and behavior of bats in finding prey. Each bat has 3 phases of movement which starts from movement by changing frequency, random walk movements, and random search movements. BBA can automatically balance its exploration and exploitation by parameter adjustment, BBA also uses a v-shaped transfer function to improve solutions. BBA has 5 parameters, that is maximum frequency, initial loudness, initial pulse rate, and alpha and gamma, namely loudness change constant and pulse rate.*

*In this study Binary algorithm is designed to solve KSP problems and subsequently applied to 10 KSP benchmark cases. The most influential parameter is the pulse rate which affects 8 cases and followed by maximum frequency and loudness. Alpha and gamma only affect one case. In addition there are also interactions between parameters. The most influential interaction is the interaction between maximal frequency and pulse rate which affects 6 benchmark cases. The best solutions obtained by BBA in each case are compared to the other 4 metaheuristic algorithms namely Dragonfly Algorithm (DA), Cat Swarm Optimization (CSO), Tabu Search (TS), and Shark Smell Optimization (SSO). BBA succeeded in achieving the optimal solution in the 6 benchmark cases tested. In cases that do not reach optimal results, the solution is near optimal with the largest error 0.85% and the smallest error 0.05% which is still below DA and TS but is above SSO and CSO.*

## KATA PENGANTAR

Puji dan syukur kepada Tuhan Yang Maha Esa karena atas berkat dan rahmat yang diberikan dalam menyelesaikan skripsi dengan judul Penerapan *Binary Bat Algorithm* dalam Penyelesaian *Knapsack Sharing Problem*. Dalam pelaksanaan dan proses penelitian serta penyusunan laporan skripsi ini, penulis mendapatkan banyak pengalaman serta bimbingan dari berbagai pihak. Maka dari itu pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah memberikan dukungan dan bantuan secara langsung maupun tidak langsung, yaitu :

1. Orang tua dan kakak penulis yang telah memberi dukungan dan motivasi
2. Bapak Alfian, S.T., M.T. sebagai dosen pembimbing yang telah memberikan motivasi, masukan, dan bimbingan dengan penuh kesabaran dalam penyusunan laporan skripsi ini.
3. Ibu Cynthia Prithadevi Juwono, Ir., M.S. sebagai dosen penguji proposal dan penguji sidang skripsi yang telah memberi banyak masukan dan nasihat pada penelitian ini.
4. Bapak Hanky Fransiscus, S.T., M.T. sebagai dosen penguji proposal dan penguji sidang skripsi yang telah memberi banyak masukan dan nasihat pada penelitian ini
5. Bapak Romy Loice, S.T., M.T. selaku dosen wali, Koordinator Skripsi, serta Ketua Program Studi Teknik Industri UNPAR yang telah memberikan dukungan, kesempatan, dan persetujuan dalam penelitian ini.
6. Dyo, Alex, Januar, Jane, Dian, Kenny, Joice, dan Verrel yang saling mendukung sesama topik algoritma dalam proses penelitian dan penyusunan laporan
7. Seluruh Dosen dan Asisten Perancangan Sistem Terintegrasi yang telah memberikan pengetahuan baru dan pengalaman berharga bagi penulis.
8. Seluruh teman-teman kelas D
9. Seluruh teman-teman TI 2015
10. Seluruh dosen yang mengajar di TI UNPAR

11. Seluruh staf dan pekarya FTI UNPAR
12. Semua pihak lain yang terlibat dalam pembuatan laporan skripsi yang tidak dapat disebutkan satu per satu.

Begitu banyak pengalaman yang penulis dapatkan selama proses penelitian dan penyusunan laporan skripsi ini. Besar harapan penulis agar kelak penelitian ini dapat memberikan manfaat bagi pembaca. Penulis menyadari bahwa masih terdapat ketidaksempurnaan dalam penelitian. Untuk itu penulis dengan senang hati terbuka menerima kritik dan saran yang dapat digunakan sebagai perbaikan laporan ini. Akhir kata, penulis ucapkan terima kasih.

Bandung, Januari 2019

Demasdika Adhiwane

## DAFTAR ISI

<b>ABSTRAK</b> .....	i
<b>ABSTRACT</b> .....	ii
<b>KATA PENGANTAR</b> .....	iii
<b>DAFTAR ISI</b> .....	v
<b>DAFTAR TABEL</b> .....	ix
<b>DAFTAR GAMBAR</b> .....	xi
<b>DAFTAR LAMPIRAN</b> .....	xiii
<b>BAB I PENDAHULUAN</b> .....	I-1
I.1 Latar Belakang Masalah.....	I-1
I.2 Identifikasi dan Rumusan Masalah.....	I-3
I.3 Batasan Masalah.....	I-7
I.4 Tujuan Penelitian.....	I-7
I.5 Manfaat Penelitian.....	I-8
I.6 Metodologi Penelitian.....	I-8
I.7 Sistematika Penulisan.....	I-11
<b>BAB II TINJAUAN PUSTAKA</b> .....	II-1
II.1 <i>Knapsack Sharing Poble</i> m.....	II-1
II.2 Metode Optimasi <i>Knapsack Sharing Problem</i> .....	II-3
II.2.1 Metode Eksak.....	II-3
II.2.2 Metode Heuristik.....	II-4
II.2.3 Metode Metaheuristik.....	II-4
II.2.4 <i>Binary Bat Algorithm</i> .....	II-5
II.3 <i>Design of Experiment</i> .....	II-13
<b>BAB III PERANCANGAN ALGORITMA</b> .....	III-1
III.1 Perancangan <i>Encoding</i> .....	III-1
III.2 Perancangan <i>Decoding</i> .....	III-5
III.3 Perancangan Algoritma.....	III-7
III.3.1 Notasi Algoritma.....	III-7
III.3.2 Algoritma Utama.....	III-9
III.3.3 Algoritma Penentuan Rasio.....	III-12

III.3.4	Algoritma Pembuatan Posisi Awal.....	III-16
III.3.5	Algoritma Perhitungan Berat .....	III-19
III.3.6	Algoritma Perhitungan <i>Fitness</i> .....	III-20
III.3.7	Algoritma Perbaikan Solusi .....	III-21
III.3.8	Algoritma <i>Update</i> Solusi Terbaik .....	III-24
III.3.9	Algoritma Matriks Frekuensi .....	III-27
III.3.10	Algoritma Perpindahan Frekuensi .....	III-27
III.3.11	Algoritma <i>Random Walk</i> .....	III-30
III.3.12	Algoritma <i>Random Search</i> .....	III-33
III.3.13	Algoritma <i>Update</i> Parameter .....	III-36
III.3	Validasi Algoritma.....	III-38
<b>BAB IV</b>	<b>IMPLEMENTASI ALGORITMA</b> .....	IV-1
IV.1	Verifikasi Program Komputer .....	IV-1
IV.2	Implementasi <i>Binary Bat Algorithm</i> .....	IV-12
IV.2.1	Penentuan Nilai Parameter Pengujian <i>Binary Bat Algorithm</i> .....	III-14
IV.2.2	Penerapan <i>Binary Bat Algorithm</i> pada Kasus A05.1 .....	III-22
IV.2.3	Penerapan <i>Binary Bat Algorithm</i> pada Kasus A05C.1.....	III-23
IV.2.4	Penerapan <i>Binary Bat Algorithm</i> pada Kasus A20.1 .....	III-24
IV.2.5	Penerapan <i>Binary Bat Algorithm</i> pada Kasus A30.1 .....	III-25
IV.2.6	Penerapan <i>Binary Bat Algorithm</i> pada Kasus A50.1 .....	III-26
IV.2.7	Penerapan <i>Binary Bat Algorithm</i> pada Kasus B05.1 .....	III-27
IV.2.8	Penerapan <i>Binary Bat Algorithm</i> pada Kasus B05C.1.....	III-28
IV.2.9	Penerapan <i>Binary Bat Algorithm</i> pada Kasus B20.1 .....	III-29
IV.2.10	Penerapan <i>Binary Bat Algorithm</i> pada Kasus B30.1 .....	III-30
IV.2.11	Penerapan <i>Binary Bat Algorithm</i> pada Kasus B50.1 .....	III-31
IV.3	Pengujian Parameter <i>Binary Bat Algorithm</i> .....	IV-12
IV.4	Perbandingan Performansi <i>Binary Bat Algorithm</i> .....	IV-12
<b>BAB V</b>	<b>ANALISIS</b> .....	V-1
V.1	Analisis <i>Encoding</i> dan <i>Decoding</i> .....	V-1
V.2	Analisis Algoritma Perbaikan Solusi .....	V-5
V.3	Analisis Perpindahan Kelelawar .....	V-6
V.4	Analisis Pengaruh Parameter <i>Binary Bat Algorithm</i> .....	V-8
V.5	Analisis Perbandingan Performansi <i>Binary Bat Algorithm</i> .....	V-10

<b>BAB VI KESIMPULAN DAN SARAN</b> .....	V-1
VI.1 Kesimpulan .....	VI-1
VI.2 Saran .....	VI-2
<b>DAFTAR PUSTAKA</b>	
<b>LAMPIRAN</b>	



## DAFTAR TABEL

Tabel II.1 Tabel Pengujian ANOVA <i>Multifactor</i> .....	I-15
Tabel III.1 Contoh Kasus <i>Encoding</i> .....	III-2
Tabel III.2 Contoh Perhitungan Rasio.....	III-2
Tabel III.3 Pengurutan Rasio.....	III-3
Tabel III.4 Pengurutan Kumulatif.....	III-3
Tabel III.5 Perhitungan Nilai <i>Cutoff</i> .....	III-4
Tabel III.6 Penentuan Posisi Awal dan Perhitungan <i>Fitness</i> .....	III-5
Tabel III.7 Posisi <i>Decoding</i> Kelelawar.....	III-6
Tabel III.8 Proses <i>Decoding</i> .....	III-2
Tabel III.9 Matriks <i>Profit</i> Validasi.....	III-38
Tabel III.10 Matriks <i>Weight</i> Validasi.....	III-38
Tabel IV.1 Karakteristik Kasus <i>Benchmark</i> .....	IV-13
Tabel IV.2 Rekapitulasi Iterasi Setiap Kasus.....	IV-20
Tabel IV.3 Rekapitulasi Kombinasi Parameter.....	IV-22
Tabel IV.4 Rekapitulasi Hasil Penerapan pada Kasus A05.1.....	IV-23
Tabel IV.5 Rekapitulasi Hasil Penerapan pada Kasus A05C.1.....	IV-24
Tabel IV.6 Rekapitulasi Hasil Penerapan pada Kasus A20.1.....	IV-25
Tabel IV.7 Rekapitulasi Hasil Penerapan pada Kasus A30.1.....	IV-26
Tabel IV.8 Rekapitulasi Hasil Penerapan pada Kasus A50.1.....	IV-27
Tabel IV.9 Rekapitulasi Hasil Penerapan pada Kasus B05.1.....	IV-28
Tabel IV.10 Rekapitulasi Hasil Penerapan pada Kasus B05C.1.....	IV-29
Tabel IV.11 Rekapitulasi Hasil Penerapan pada Kasus B20.1.....	IV-30
Tabel IV.12 Rekapitulasi Hasil Penerapan pada Kasus B30.1.....	IV-31
Tabel IV.13 Rekapitulasi Hasil Penerapan pada Kasus B50.1.....	IV-32
Tabel IV.14 Rekapitulasi Hasil Pengujian Parameter.....	IV-35
Tabel IV.15 Rekapitulasi <i>Benchmark</i> Algoritma.....	IV-37

## DAFTAR GAMBAR

Gambar I.1 <i>Flowchart</i> Metodologi Penelitian.....	I-10
Gambar II.1 Posisi Awal Kelelawar.....	II-8
Gambar II.2 Arah Perpindahan Dengan Frekuensi.....	II-9
Gambar II.3 Hasil Perpindahan Dengan Frekuensi.....	II-9
Gambar II.4 Arah Perpindahan Dengan <i>Random Walk</i> .....	II-8
Gambar II.5 Hasil Perpindahan Dengan <i>Random Walk</i> .....	II-10
Gambar II.6 Arah Pergerakan Dengan <i>Random Search</i> .....	II-11
Gambar II.7 Posisi Akhir Kelelawar Pada Iterasi 1.....	II-11
Gambar II.8 <i>Sigmoid Transfer Function</i> dan <i>V-shaped Transfer Function</i> .....	II-13
Gambar III.1 <i>Flowchart</i> Algoritma Utama.....	III-10
Gambar III.2 <i>Flowchart</i> Algoritma Rasio.....	III-12
Gambar III.3 <i>Flowchart</i> Algoritma Posisi Awal.....	III-17
Gambar III.4 <i>Flowchart</i> Algoritma Berat.....	III-19
Gambar III.5 <i>Flowchart</i> Algoritma <i>Fitness</i> .....	III-21
Gambar III.6 <i>Flowchart</i> Algoritma Perbaikan Solusi.....	III-22
Gambar III.7 <i>Flowchart</i> Algoritma <i>Update</i> Solusi.....	III-25
Gambar III.8 <i>Flowchart</i> Algoritma Matriks Frekuensi.....	III-28
Gambar III.9 <i>Flowchart</i> Algoritma Perpindahan Frekuensi.....	III-29
Gambar III.10 <i>Flowchart</i> Algoritma <i>Random Walk</i> .....	III-31
Gambar III.11 <i>Flowchart</i> Algoritma <i>Random Search</i> .....	III-34
Gambar III.12 <i>Flowchart</i> Algoritma <i>Update</i> Parameter.....	III-37
Gambar IV.1 Program Utama.....	IV-2
Gambar IV.2 <i>Error Message</i> Kesalahan Parameter.....	IV-3
Gambar IV.3 Program Perhitungan Rasio Bagian 1.....	IV-4
Gambar IV.4 Program Perhitungan Rasio Bagian 2.....	IV-5
Gambar IV.5 Program Pembentukan Posisi Awal.....	IV-5
Gambar IV.6 Program Perhitungan Berat.....	IV-6
Gambar IV.7 Program Perhitungan <i>Fitness</i> .....	IV-6
Gambar IV.8 Program Perbaikan Solusi.....	IV-7
Gambar IV.9 Program <i>Update</i> Solusi Terbaik.....	IV-8

Gambar IV.10 Program Pembuatan Matriks Frekuensi.....	IV-8
Gambar IV.11 Program Perpindahan Frekuensi.....	IV-9
Gambar IV.12 Program <i>Random Walk</i> .....	IV-10
Gambar IV.13 Program <i>Random Search</i> .....	IV-10
Gambar IV.14 Program <i>Update</i> Parameter.....	IV-11
Gambar IV.16 Grafik Konvergensi A05.1.....	IV-15
Gambar IV.17 Grafik Konvergensi A05C.1.....	IV-16
Gambar IV.18 Grafik Konvergensi A20.1.....	IV-16
Gambar IV.19 Grafik Konvergensi A30.1.....	IV-17
Gambar IV.20 Grafik Konvergensi A50.1.....	IV-17
Gambar IV.21 Grafik Konvergensi B05.1.....	IV-18
Gambar IV.22 Grafik Konvergensi B05C.1.....	IV-18
Gambar IV.23 Grafik Konvergensi B20.1.....	IV-19
Gambar IV.24 Grafik Konvergensi B30.1.....	IV-19
Gambar IV.25 Grafik Konvergensi B50.1.....	IV-20

## DAFTAR LAMPIRAN

Lampiran A Waktu <i>Running</i> .....	I-10
Lampiran B Hasil Uji Parameter.....	II-8

# BAB I

## PENDAHULUAN

Pada bagian pertama ini akan dipaparkan mengenai pendahuluan dari penelitian penerapan *binary bat algorithm* untuk menyelesaikan *knapsack sharing problem*. Bagian ini dibagi menjadi beberapa bagian yaitu latar belakang masalah, identifikasi dan rumusan masalah, batasan dan asumsi penelitian, tujuan penelitian, manfaat penelitian, dan metodologi penelitian.

### I.1 Latar Belakang Masalah

Dalam kehidupan nyata, manusia selalu berusaha memenuhi kebutuhan hidupnya. Dalam memenuhi kebutuhan hidupnya manusia memerlukan berbagai sumber daya yang terbatas jumlahnya. Contoh sumber daya tersebut adalah listrik, makanan, energi, uang, dan lain-lain. Setiap penggunaan sumber daya yang terbatas tersebut bukan hanya memberikan keuntungan bagi manusia tetapi juga mengurangi jumlah sumber daya yang tersedia. Penggunaan sumber daya yang tidak bijaksana dapat memicu terjadinya kelangkaan sumber daya. Oleh sebab itu perlu dilakukan pengalokasian sumber daya yang optimal dengan mempertimbangkan hal-hal tersebut. Permasalahan alokasi sumber daya tersebut dapat dimodelkan dalam *Knapsack Problem* (Winston, 2004).

*Knapsack* pada kasus ini diartikan sebagai wadah, tas, atau tempat penampungan yang memiliki kapasitas tertentu. Analogi dasar yang digunakan adalah terdapat sejumlah benda yang memiliki keuntungan tertentu jika dimasukkan ke dalam wadah yang memiliki kapasitas terbatas. Benda tersebut juga memiliki bobot atau berat yang akan mengurangi kapasitas dari wadah. Tujuan dari kasus KP adalah mencari kombinasi dari benda atau *item* yang optimal untuk mendapatkan keuntungan sebesar-besarnya dan tidak melebihi kapasitas dari wadah (Yamada dan Futakawa, 1997). Contoh kasus umum yang dapat dimodelkan dengan menggunakan KP adalah seorang pendaki gunung yang menentukan barang apa saja yang akan dibawa ke dalam ranselnya yang memiliki ruang terbatas untuk mendapatkan manfaat sebesar-besarnya (Martello dan Toth, 1990).

Salah satu permasalahan nyata yang menggunakan KP adalah *capital budgeting*. Pada permasalahan ini sebuah perusahaan ingin melakukan investasi di beberapa proyek. Setiap investasi memberikan keuntungan yang berbeda-beda dan memerlukan jumlah investasi yang berbeda. Jumlah dana yang dimiliki oleh perusahaan untuk diinvestasikan terbatas. Pada kasus ini perusahaan ingin memaksimalkan keuntungan yang didapatkan dengan melakukan investasi di beberapa proyek dengan dana yang terbatas tersebut. Pemilihan proyek investasi yang paling optimal tersebut dapat dimodelkan dalam *Knapsack Problem* (Akpan, Etuk, dan Essi, 2011).

*Knapsack Problem* merupakan permasalahan optimasi kombinatorial dimana pencarian solusi dilakukan pada ruang solusi yang terbatas. Ciri lainnya dari permasalahan kombinatorial memiliki variabel diskrit dan solusi yang terbentuk berupa urutan atau kombinasi dari bilangan *integer* (Papadimitriou dan Steiglitz, 1998). Dalam perkembangannya, muncul permasalahan-permasalahan baru dalam kehidupan sehingga muncul juga variasi-variasi lain dari *Knapsack Problem* untuk menyesuaikan dengan permasalahan baru tersebut. Variasi *Knapsack Problem* tersebut antara lain adalah *Multidimensional Knapsack Problem*, *Bounded Knapsack Problem*, *Change-making Knapsack Problem* (Martello dan Toth, 1990), *Multiple-choice Knapsack Problem*, *Quadratic Knapsack Problem*, dan *Knapsack Sharing Problem* (Brown, 1979).

Pada penelitian ini permasalahan yang akan diselesaikan adalah *Knapsack Sharing Problem* (KSP). KSP merupakan pengembangan dari *Knapsack Problem* dimana *item* terbagi menjadi beberapa kelas. Tujuan dari KSP adalah memaksimalkan keuntungan minimum dari kelas-kelas tersebut. Solusi dari KSP berbentuk kombinasi dari barang yang dimasukkan dari setiap kelas ke dalam *knapsack*. Contohnya adalah bagaimana seorang komandan militer mengalokasikan amunisi dan bahan bakar pada pasukannya (Brown, 1979).

*Knapsack Sharing Problem* adalah permasalahan optimasi yang termasuk dalam *NP-hard problem* (*Non-deterministic Polynomial-time Hard*) yang artinya adalah semakin bertambahnya ruang solusi maka waktu pencarian solusi meningkat secara signifikan. Penyelesaian KSP dengan metode eksak memakan waktu yang sangat panjang, maka dari itu diperlukan metode-metode dan pendekatan heuristik dan metaheuristik untuk menyelesaikan permasalahan KSP ini (Hifi, Sadfi, dan Sbihi, 2002).

## I.2 Identifikasi dan Rumusan Masalah

*Knapsack Sharing Problem* (KSP) merupakan salah satu variasi dari permasalahan *0-1 Knapsack Problem* (KP). Jika pada KP setiap *item* memberikan kontribusi pada keuntungan yang sama, maka pada KSP setiap *item* terbagi ke dalam beberapa kelas dan keuntungan yang didapatkan juga terbagi ke dalam kelas-kelas tersebut. Tujuan utama dari KSP adalah memaksimalkan keuntungan dari kelas yang memiliki keuntungan paling kecil (Brown, 1979).

Contoh permasalahan tersebut dalam dunia industri adalah permasalahan alokasi dana untuk pembelian mesin dan peralatan baru. Pada sebuah pabrik terdapat 3 divisi yaitu divisi *machining*, divisi *casting*, dan divisi *welding*. Pada divisi *machining* digunakan mesin *milling*, *drilling*, dan *turning*. Pada divisi *casting* digunakan peralatan pengecoran seperti *furnace* dan  *mold*. Pada divisi *welding* digunakan mesin *oxyacetylene welding* dan *arc welding*. Divisi-divisi tersebut adalah kelas, sedangkan mesin adalah *item* dalam KSP. Setiap pembelian mesin akan meningkatkan *throughput* produksi divisi tersebut. *Throughput* tersebut disebut juga dengan *profit* dalam KSP. Kemudian muncul permasalahan dimana dana yang tersedia terbatas untuk pembelian mesin. Dalam KSP dana tersebut disebut dengan kapasitas *knapsack* dan harga mesin merupakan berat dari benda. Tujuan penyelesaian permasalahan tersebut adalah mencari kombinasi-kombinasi mesin yang akan dibeli untuk memaksimalkan *throughput* produksi setiap divisi dengan dana yang terbatas. Permasalahan tersebut dapat dimodelkan dalam *Knapsack Sharing Problem* (Herman, 2016).

Metode pemecahan masalah optimasi seperti KSP dapat berupa metode eksak dan heuristik. Metode eksak memberikan hasil optimal, tetapi metode ini akan memerlukan waktu yang sangat lama seiring dengan bertambahnya ruang solusi. Dalam permasalahan KSP terdapat beberapa metode eksak yang telah digunakan untuk menyelesaikan permasalahan tersebut, yaitu *branch and bound* dan *binary search* yang digunakan oleh Yamada, Futukawa, dan Kataoka (1998). Metode eksak lainnya yaitu *tree search algorithm* digunakan oleh Hifi dan Mhalla (2005) dan *dynamic programming* yang digunakan oleh Hifi dan Sadfi (2002).

Metode heuristik mengatasi kelemahan dari metode eksak, yaitu metode heuristik mampu menyelesaikan permasalahan dengan waktu penyelesaian yang lebih cepat dari metode eksak namun tidak ada jaminan bahwa solusi yang dihasilkan adalah optimal. Beberapa metode heuristik yang pernah digunakan

untuk menyelesaikan KSP adalah *greedy algorithm* (Yamada dan Futukawa, 1997) yang memberikan hasil yang baik. Kekurangan dari metode heuristik adalah metode ini merupakan *problem dependent technique* yang artinya satu metode umumnya hanya dapat menyelesaikan satu permasalahan. Contohnya adalah *Dijkstra's Algorithm* yang hanya dapat digunakan untuk menyelesaikan *shortest path problem*. Metode heuristik juga sangat mungkin untuk terjebak di *local optimal solution*. Untuk mengatasi kekurangan ini maka muncul pendekatan metaheuristik.

Metaheuristik adalah metode optimasi yang memiliki tingkatan di atas heuristik. Dibandingkan dengan metode heuristik yang merupakan *problem dependent technique*, metaheuristik adalah *problem independent* yang artinya metode ini tidak bergantung pada permasalahan yang ingin diselesaikan. Metode eksak dapat mendapatkan hasil *global optimal* dengan waktu pencarian yang sangat lama karena metode eksak mencoba semua kombinasi solusi yang memungkinkan mencapai global optimal dan heuristik melakukan pencarian solusi lokal sehingga sangat mungkin untuk terjebak dalam *local optimal*, sedangkan pada metaheuristik terdapat dua prosedur inti yaitu eksplorasi dan eksploitasi. Eksploitasi adalah prosedur mencari solusi di sekitar titik saat ini (*neighborhood*) dengan tujuan meningkatkan solusi sedangkan eksplorasi adalah prosedur mencari solusi ke titik-titik baru pada ruang solusi (Crepinsek, Liu dan Mernik, 2013). Dengan dua prosedur tersebut waktu pencarian solusi dan kemungkinan untuk terjebak dalam *local optimal solution* akan menurun. Metode metaheuristik yang pernah digunakan untuk menyelesaikan permasalahan KSP antara lain adalah *Tabu Search* (Hifi, Sadfi, dan Sbihi, 2002), *Ant Colony System* (Widiapradja, 2013), *Cuckoo Search* (Angga, 2014), *Cat Swarm Optimization* (Herman, 2016), *Shark Smell Optimization* (Sutanto, 2018), dan *Dragonfly Algorithm* (Laksono, 2018).

*Bat Algorithm* (BA) adalah algoritma metaheuristik yang diperkenalkan pertama kali oleh Xin-She Yang pada tahun 2010. BA adalah algoritma metaheuristik yang sudah cukup lama dan terbukti telah menyelesaikan berbagai permasalahan optimasi dengan hasil yang sangat baik. *Bat Algorithm* adalah sebuah algoritma yang terinspirasi dari kemampuan ekolokasi kelelawar dalam mencari mangsa dan membedakan tipe-tipe mangsa dalam kegelapan. Ekolokasi atau disebut juga dengan biosonar adalah metode komunikasi, navigasi, dan deteksi yang digunakan oleh beberapa jenis binatang. Pada proses ekolokasi,



kelelawar memancarkan gelombang suara dengan frekuensi melebihi yang dapat didengar oleh manusia, gelombang tersebut kemudian akan menggema jika bertumbukan dengan rintangan atau mangsa. Kelelawar menggunakan pantulan tersebut untuk mendeteksi dan mengidentifikasi objek. Dari situ kelelawar dapat mencari mangsa di malam hari dan melewati rintangan-rintangan dalam kegelapan total. Prosedur ekolokasi kelelawar inilah yang menjadi dasar perancangan BA.

Pada *Bat Algorithm* posisi setiap kelelawar merupakan representasi dari solusi permasalahan. Kelelawar-kelelawar tersebut akan terbang secara random untuk mencari mangsa. Proses pencarian mangsa tersebut dilakukan dengan merubah frekuensi, *loudness*, dan *pulse emission rate*. Terdapat tiga fase pergerakan kelelawar, fase pertama adalah pergerakan frekuensi dimana kelelawar akan bergerak dengan merubah frekuensi, kecepatan, dan posisi serta membandingkan posisinya dengan solusi terbaik sekarang. Fase berikutnya adalah eksploitasi yaitu pencarian lokal dengan melakukan *random walk* dimana kelelawar akan bergerak secara acak berdasarkan *loudness* dari seluruh kelelawar. Fase terakhir adalah *random search* dimana kelelawar akan bergerak secara acak sepenuhnya pada ruang solusi (Yang, 2010).

Pada algoritma-algoritma yang pernah digunakan sebelumnya seperti *Cuckoo Search* (Angga, 2014) dan *Cat Swarm Optimization* (Herman, 2016) terdapat permasalahan dimana kemungkinan untuk terjebak dalam *local optimal solution* cukup tinggi, karena itu terdapat langkah terakhir dalam BA yaitu proses *update* terhadap *loudness* dan *pulse emission rate* setelah kelelawar berpindah posisi. Seiring dengan iterasi nilai *loudness* akan menurun dan *pulse emission rate* akan meningkat. Semakin tinggi *pulse emission rate* maka frekuensi eksploitasi yang dilakukan akan semakin berkurang dan begitu juga dengan *loudness*, semakin rendah *loudness* maka intensitas eksploitasi akan semakin rendah. Dengan menggunakan kombinasi parameter *loudness* dan *pulse emission rate*, BA dapat menyeimbangkan antara *local search* dan *global search* untuk menghindari terjebak dalam *local optimal solution* (Yang, 2010).

Parameter-parameter BA yaitu *pulse rate*, *loudness*, konstanta *pulse rate*, konstanta *loudness*, dan frekuensi awal ditentukan sebelum algoritma diterapkan. Parameter-parameter tersebut memiliki kemungkinan berpengaruh terhadap performansi algoritma. Oleh sebab itu, selain menerapkan BA pada penyelesaian

KSP dilakukan juga pengujian parameter-parameter BA untuk mengetahui parameter yang berpengaruh pada solusi yang dihasilkan

Permasalahan *Knapsack Sharing Problem* merupakan permasalahan biner. Solusi dari KSP berbentuk angka biner yaitu kombinasi dari angka 1 dan 0 yang menunjukkan apakah barang dimasukkan (1) atau tidak (0) ke dalam wadah. Proses pencarian solusi hanya dapat dilakukan dengan merubah angka 1 menjadi 0 atau sebaliknya. Maka dari itu pada penelitian ini digunakan modifikasi dari BA standar yang mampu menyelesaikan permasalahan kontinyu agar dapat digunakan untuk menyelesaikan permasalahan biner yaitu *Binary Bat Algorithm* atau BBA (Mirjalili, Mirjalili, dan Yang, 2013).

Dalam menerapkan algoritma kontinyu pada permasalahan biner diperlukan *transfer function* untuk mengubah bilangan *real* yaitu posisi menjadi probabilitas. *Transfer function* tersebut dapat berupa *s-shaped* atau *v-shaped*. Mirjalili, Mirjalili, dan Yang (2013) menggunakan *v-shaped transfer function* dalam merancang BBA. Keunggulan dari *v-shaped transfer function* adalah solusi biner tidak dipaksakan untuk setiap dimensi pada kelelawar tetapi setiap dimensi kelelawar akan merubah solusi sebelumnya sesuai dengan posisinya masing-masing dalam bentuk probabilitas sedangkan pada *s-shaped transfer function* terdapat kecenderungan dimana semakin positif posisi kelelawar maka kemungkinan besar solusi biner yang dihasilkan adalah 1 dan jika nilai posisi kelelawar negatif maka solusi biner yang dihasilkan adalah 0.

Yang, Mirjalili, dan Mirjalili (2013) telah melakukan *benchmark* terhadap *Binary Bat Algorithm* untuk beberapa *test function*. Hasil *benchmark* tersebut kemudian dibandingkan dengan hasil *benchmark* algoritma lainnya yaitu *Genetic Algorithm* (GA) dan *Binary Particle Swarm Optimization* (BPSO). Proses *benchmark* ini dilakukan untuk mengetahui performansi dari BBA jika dibandingkan dengan algoritma lainnya dari segi akurasi dan efisiensi. Berdasarkan hasil *benchmark* yang dilakukan oleh Yang, Mirjalili, dan Mirjalili, BBA memiliki performansi dan kecepatan konvergensi yang lebih baik dari BPSO dan GA hampir di semua *test function*. Selain itu BBA yang digunakan dalam penelitian ini juga pernah digunakan untuk menyelesaikan *Multidimensional Knapsack Problem* dan menghasilkan solusi yang lebih baik dari PSO (Sabba dan Chikhi, 2014). Hal ini memungkinkan bagi BBA untuk menyelesaikan *Knapsack Sharing Problem* dengan hasil yang memuaskan juga.

Pada penelitian ini digunakan versi biner dari metaheuristik *Bat Algorithm* yaitu *Binary Bat Algorithm* untuk menyelesaikan permasalahan *Knapsack Sharing Problem*. Hasil performansi BBA tersebut akan dibandingkan dengan algoritma metaheuristik lainnya yang pernah digunakan untuk menyelesaikan KSP. Algoritma yang menjadi pembanding adalah *Tabu Search* (Hifi, Sadfi, dan Sbihi, 2002), *Cat Swarm Optimization* (Herman, 2016), *Shark Smell Optimization* (Sutanto, 2018), dan *Dragonfly Algorithm* (Laksono, 2018). Berdasarkan identifikasi masalah yang dilakukan maka didapatkan rumusan masalah pada penelitian ini adalah sebagai berikut :

1. Bagaimana penerapan *Binary Bat Algorithm* dalam menyelesaikan permasalahan *Knapsack Sharing Problem* ?
2. Bagaimana pengaruh parameter-parameter dari *Binary Bat Algorithm* terhadap performansi *Binary Bat Algorithm* ?
3. Bagaimana perbandingan solusi yang dihasilkan *Binary Bat Algorithm* dengan *Tabu Search* (Hifi, Sadfi, dan Sbihi, 2002), *Cat Swarm Optimization* (Herman, 2016), *Shark Smell Optimization* (Sutanto, 2018), dan *Dragonfly Algorithm* (Laksono, 2018)?

### **I.3 Batasan Masalah**

Batasan masalah pada penelitian penerapan *Binary Bat Algorithm* untuk masalah *Knapsack Sharing Problem* ditentukan sesuai dengan identifikasi dan rumusan masalah. Pembatasan masalah yang digunakan pada penelitian ini adalah:

1. Permasalahan yang akan diselesaikan pada penelitian ini adalah kasus *benchmark* yang umum digunakan dalam menguji performansi algoritma.

### **I.4 Tujuan Penelitian**

Berdasarkan hasil identifikasi dan perumusan masalah pada subbab I.2 maka didapatkan tujuan dari penelitian yang dilakukan adalah :

1. Menerapkan *Binary Bat Algorithm* pada penyelesaian kasus *Knapsack Sharing Problem*.
2. Mengetahui parameter-parameter yang berpengaruh pada performansi *Binary Bat Algorithm*.

3. Membandingkan solusi yang dihasilkan oleh *Binary Bat Algorithm* terhadap solusi yang dihasilkan oleh metode lainnya.

### **I.5 Manfaat Penelitian**

Berdasarkan penelitian yang akan dilakukan terdapat beberapa manfaat-manfaat yang diharapkan dari penelitian ini. Manfaat penelitian tersebut adalah sebagai berikut :

1. Menambah penggunaan *Binary Bat Algorithm* pada kasus *Knapsack Sharing Problem*.
2. Menambah pengetahuan dan wawasan mengenai penerapan *Binary Bat Algorithm* pada penyelesaian *Knapsack Sharing Problem*.
3. Menambah referensi untuk penelitian lain yang berkaitan dengan *Binary Bat Algorithm* dan *Knapsack Sharing Problem*.

### **I.6 Metodologi Penelitian**

Dalam melakukan penelitian penerapan *Binary Bat Algorithm* pada permasalahan *Knapsack Sharing Problem* terdapat metodologi yang digunakan. Pada penelitian ini, metodologi penelitian yang digunakan dapat dilihat pada Gambar I.1 dan diuraikan melalui langkah-langkah berikut :

1. Studi Literatur  
Pada tahap pertama ini dilakukan pencarian dan pengumpulan informasi terkait dengan algoritma yang digunakan yaitu *Binary Bat Algorithm* (BBA) dan *Knapsack Sharing Problem* (KSP) melalui referensi jurnal, *paper*, dan buku.
2. Identifikasi dan Perumusan Masalah  
Pada tahap ini dilakukan identifikasi masalah yaitu permasalahan KSP dan penerapan BBA untuk menyelesaikannya. Selanjutnya dilakukan perumusan masalah berdasarkan identifikasi yang dilakukan.
3. Penentuan Batasan Masalah dan Asumsi  
Setelah didapatkan rumusan-rumusan masalah, maka selanjutnya akan ditentukan batasan dan asumsi penelitian untuk membantu proses penelitian dan agar penelitian tetap terfokus pada tujuan.
4. Penentuan Tujuan dan Manfaat Penelitian

Pada tahap ini ditentukan tujuan yang ingin dicapai dari penelitian yang dilakukan. Selain itu manfaat penelitian juga ditentukan agar penelitian yang dilakukan memiliki maksud yang jelas.

5. Perancangan *Binary Bat Algorithm* Pada Permasalahan *Knapsack Sharing Problem*.

Pada tahap ini dilakukan perancangan algoritma BBA agar sesuai dengan permasalahan yang ingin diselesaikan yaitu KSP.

6. Validasi Algoritma

Algoritma yang telah dirancang harus dilakukan validasi terlebih dahulu sebelum diimplementasikan. Proses validasi dilakukan untuk setiap komponen dan proses dalam algoritma agar algoritma yang dihasilkan benar dan sesuai dengan permasalahan.

7. Pembuatan Program

Langkah selanjutnya setelah algoritma melewati tahap validasi adalah pembuatan program. Pembuatan program dilakukan dengan menggunakan MATLAB.

8. Verifikasi Program

Setelah program telah dibuat, selanjutnya program akan diverifikasi terlebih dahulu sebelum digunakan untuk menyelesaikan permasalahan.

8. Penyelesaian *Knapsack Sharing Problem* Dengan *Binary Bat Algorithm*

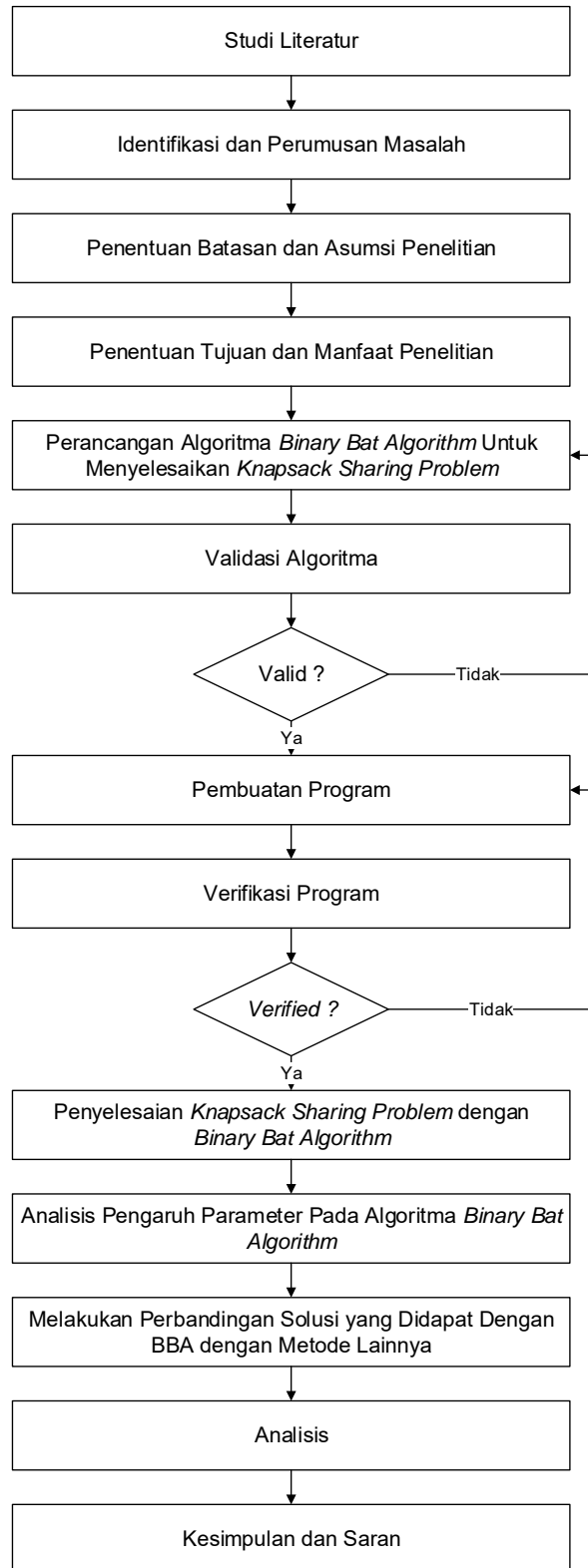
Pada tahap ini, program yang telah dibuat akan digunakan untuk menyelesaikan beberapa kasus *benchmark* KSP untuk mendapatkan solusi yang dihasilkan oleh BBA.

9. Analisis Pengaruh Parameter Pada Algoritma *Binary Bat Algorithm*

Pada tahap ini akan dilakukan pengujian statistik untuk mengetahui pengaruh parameter-parameter pada BBA dan menentukan kombinasi parameter terbaik untuk setiap kasus.

10. Melakukan Perbandingan Solusi yang Dihasilkan BBA dengan Metode Lainnya.

Solusi yang dihasilkan oleh algoritma BBA akan dibandingkan dengan algoritma pembanding. Selain itu akan dihitung juga kedekatan solusi yang dihasilkan BBA dengan solusi optimal setiap kasus.



Gambar I.1 Flowchart Metodologi Penelitian

11. Analisis  
Analisis dilakukan terhadap solusi yang dihasilkan BBA terhadap permasalahan KSP dan juga solusi yang dihasilkan metode lainnya. Pada tahap ini juga dilakukan analisis terhadap hasil pengujian parameter yang telah dilakukan.
12. Kesimpulan dan Saran  
Pada tahap ini dilakukan penarikan kesimpulan yang menjawab tujuan penelitian dan juga saran yang dapat diterapkan untuk penelitian-penelitian selanjutnya.

### **I.7 Sistematika Penulisan**

Dalam menulis sebuah laporan penelitian diperlukan sistematika penulisan untuk mempermudah pembaca. Sistematika penulisan ini disusun dengan urutan:

#### **BAB I PENDAHULUAN**

Bab I berisi tentang latar belakang masalah, identifikasi masalah dan rumusan masalah, batasan dan asumsi penelitian, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

#### **BAB II TINJAUAN PUSTAKA**

Bab II berisi teori-teori yang digunakan dalam penelitian terkait dengan *Knapsack Sharing Problem* yang merupakan permasalahan yang akan diselesaikan dan *Binary Bat Algorithm* yaitu metode penyelesaian yang digunakan.

#### **BAB III PERANCANGAN ALGORITMA**

Bab III berisi perancangan algoritma BBA untuk menyelesaikan KSP mulai dari metode *encoding* dan *decoding* yang digunakan, perancangan algoritma, dan validasi algoritma.

#### **BAB IV IMPLEMENTASI ALGORITMA**

Bab IV berisi pembuatan program dan verifikasi program, penerapan algoritma dalam menyelesaikan KSP dan pengujian parameter-parameter algoritma BBA untuk melihat pengaruh setiap parameter-parameter algoritma.

#### **BAB V ANALISIS**

Bab V berisi analisis tahapan-tahapan yang dilakukan serta evaluasi solusi yang dihasilkan serta analisis perbandingan dengan algoritma lainnya.

**BAB VI KESIMPULAN DAN SARAN**

Bab VI berisi kesimpulan dari penelitian yang menjawab tujuan penelitian dan saran bagi penelitian selanjutnya