

**PENERAPAN ALGORITMA *PIGEON INSPIRED*  
*OPTIMIZATION* PADA *MULTIDIMENSIONAL*  
*KNAPSACK PROBLEM***

**SKRIPSI**

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

Disusun oleh:

Nama : Christian Setiardjo

NPM : 2013610130



**PROGRAM STUDI TEKNIK INDUSTRI  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS KATOLIK PARAHYANGAN  
BANDUNG  
2017**

FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS KATOLIK PARAHYANGAN  
BANDUNG



Nama : Christian Setiardjo  
NPM : 2013610130  
Program Studi : Teknik Industri  
Judul Skripsi : PENERAPAN ALGORITMA *PIGEON INSPIRED*  
*OPTIMIZATION* PADA *MULTIDIMENSIONAL*  
*KNAPSACK PROBLEM*

**TANDA PERSETUJUAN SKRIPSI**

Bandung, 27 Juli 2017

**Ketua Program Studi Teknik**

**Industri**

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

**Pembimbing Pertama**

(Dr. Ir. Ali Sadiyoko, M.T.)

**Pembimbing Kedua**

(Fran Setiawan, S.T., M.Sc.)



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 : Christian Setiardjo

NPM : 2013610130

dengan ini menyatakan bahwa skripsi dengan judul :

**"PENERAPAN ALGORITMA PIGEON INSPIRED OPTIMIZATION PADA  
MULTIDIMENSIONAL KNAPSACK 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, 11 Juli 2017

Christian Setiardjo  
NPM : 2013610130

## ABSTRAK

*Multidimensional Knapsack Problem* atau biasa disingkat MKP merupakan model permasalahan pengalokasian sumber daya dengan beberapa wadah, dimana setiap benda memiliki bobot tertentu di setiap wadahnya. MKP sendiri merupakan pengembangan dari *knapsack problem*. Tujuan MKP adalah mencari keuntungan sebesar-besarnya dengan mempertimbangkan kapasitas dan bobot setiap wadah dan benda.

MKP akan diselesaikan menggunakan *Pigeon Inspired Optimization* atau biasa disingkat PIO dalam penelitian ini. Algoritma PIO merupakan salah satu algoritma metaheuristik yang tergolong dalam *population-based swarm intelligent* yang didasarkan pada tingkah laku burung merpati yang mengetahui letak rumahnya meskipun telah pergi jauh. Algoritma PIO ini memiliki 2 macam *operator*, yaitu *map and compass operator* yang merupakan proses eksplorasi dan *landmark operator* yang merupakan proses eksploitasi.

Perancangan algoritma PIO dilakukan untuk dapat menyelesaikan MKP. Implementasi algoritma PIO terhadap MKP diterapkan terhadap 2 jenis kasus yang memiliki karakteristik yang berbeda menggunakan metode OFAT sebanyak 5 kali replikasi untuk mendapatkan nilai parameter terbaik. Terdapat 2 kombinasi nilai parameter terbaik, yang pertama untuk kasus dengan jumlah *knapsack* yang banyak dan yang kedua untuk kasus dengan jumlah *item* yang banyak. Kombinasi pertama nilai parameter terbaik adalah  $N_{c1max}$  sebesar 1450,  $N_{c2max}$  sebesar 6,  $N_p$  sebesar 450, dan  $R$  sebesar 0.3. Kombinasi kedua nilai parameter terbaik adalah  $N_{c1max}$  sebesar 1150,  $N_{c2max}$  sebesar 4,  $N_p$  sebesar 300, dan  $R$  sebesar 0.8. Hasil implementasi dari kedua kombinasi nilai parameter terbaik Pada penerapan algoritma PIO terhadap MKP untuk 10 kasus yang dibandingkan dengan algoritma *Particle Swarm Optimization*, algoritma *Intelligent Water Drops*, dan algoritma genetika didapatkan hasil yang memuaskan. Penyimpangan terbesar diantara penerapan 10 kasus adalah sebesar 1.03% dari hasil terbaik yang diketahui selama ini dan penyimpangan terkecil diantara penerapan 10 kasus adalah sebesar 0% dari hasil terbaik yang diketahui selama ini. Penyimpangan rata-rata untuk penerapan 10 kasus adalah sebesar 0.27% dari hasil terbaik yang diketahui selama ini.

## **ABSTRACT**

*Multidimensional Knapsack Problem or sometimes abbreviated as MKP is a allocating resources problem model with several container, where every items have its own weight in every container. MKP itself is a improvement of knapsack problem. MKP's goal is to get maximum profit with the consideration of the capacity and the weight of every container and items.*

*MKP will be computed with Pigeon Inspired Optimization or sometimes abbreviated as PIO in this research. PIO algorithm is one of the metaheuristic that classified as population-based swarm intelligent that based of the behaviour of pigeon which knew their home although after gone far away. PIO algorithm have 2 kind of operator, map and compass operator that is exploration proses and landmark operator that is exploitation process.*

*Designing the PIO algorithm is a step to solve MKP. The implementation of PIO algorithm to MKP applied into 2 kind of cases with distinctive characteristic using OFAT method with 5 times replications to get the best parameter values. There are 2 combinations of the best parameter values, the first one are cases that have large amount of knapsacks and the second one are cases that have large amount of items. The first combination of the best parameter values are, 1450 for Nc1max, 6 for Nc2max, 450 for Np, and 0.3 for R. The second combination of the best parameter values are, 1150 for Nc1max, 4 for Nc2max, 300 for Np, and 0.8 for R. The implementation of both best parameter values' combination in the application of PIO algorithm to MKP for 10 cases that compared to Particle Swarm Optimization algorithm, Intelligent Water Drops algorithm, and Genetic algorithm have a satisfying result. The Highest error from the application of 10 cases is 1.03% from the best known solution and the lowest error from the application of 10 cases is 0% from the best known solution. The average error from the application of 10 cases is 0.27% from the best known solution.*

## KATA PENGANTAR

Puji dan Syukur kepada Tuhan Yang Maha Esa atas berkat dan rahmatNya yang membantu penulis dalam menyelesaikan laporan skripsinya yang berjudul *Penerapan Algoritma Pigeon Inspired Optimization Pada Multidimensional Knapsack Problem*. Dalam proses menyusun skripsi ini, penulis mendapatkan banyak bantuan dan bimbingan dari berbagai macam pihak. Melalui kata pengantar ini penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah memberikan banyak bantuan dan bimbingan baik secara langsung dan tidak langsung. Penulis mengucapkan terima kasih kepada :

1. Orang Tua penulis yang tidak hentinya memberikan dukungan, doa, nasihat, dan motivasi kepada penulis. Berkat Orang Tua penulis, penulis mampu menyelesaikan skripsi ini dengan sepenuh hati.
2. Bapak Dr. Ir. Ali Sadiyoko, S.T., M.T. sebagai dosen pembimbing dari penulis yang tidak hentinya memberikan dukungan, bantuan, motivasi, dan masukan dengan penuh kesabaran dalam membimbing penulis menyelesaikan laporan skripsinya.
3. Bapak Fran Setiawan, S.T., M.Sc. sebagai dosen pembimbing dari penulis yang tidak hentinya memberikan dukungan, bantuan, motivasi, dan masukan dengan penuh kesabaran dalam membimbing penulis menyelesaikan laporan skripsinya.
4. Ibu Cynthia Prithadevi Juwono, Ir., M.S. sebagai dosen penguji proposal skripsi yang telah memberikan saran dan masukkan untuk membantu penulis supaya mengerjakan laporan skripsi dengan lebih baik.
5. Bapak Hanky Fransiscus, S.T., M.T. sebagai dosen penguji proposal skripsi yang telah memberikan saran dan masukkan untuk membantu penulis supaya mengerjakan laporan skripsi dengan lebih baik.
6. Marcellina Agnes sebagai orang penting dalam hidup penulis memberikan semangat dan dukungan setiap hari supaya penulis dapat menyelesaikan laporan skripsi dengan tepat waktu.
7. Evan Septiano, Denny Herman, Burky Susanto, David Stanley, dan senior teknik industri UNPAR lainnya yang telah menyelesaikan laporan skripsi

mereka dengan topik algoritma seperti penulis. Penelitian mereka sangat membantu penulis dalam memahami mengenai topik algoritma yang juga merupakan topik penelitian skripsi penulis.

8. Felix Arya, Adrianus Vincent, Arnold Raharja, Deva Nareswara, Agnes Monica, Ricky Tendi, dan Marcellinus Rico sebagai teman-teman dengan topik penelitian skripsi yang sama, yaitu algoritma. Berkat membantu serta memberikan dukungan antar satu sama lain, penulis mampu menyelesaikan laporan skripsi.
9. Seluruh teman dekat penulis baik yang menempuh perjalanan yang sama dalam teknik industri UNPAR maupun tidak yang memberikan banyak dukungan dan motivasi untuk penulis supaya bersemangat dalam menyelesaikan laporan skripsi.
10. Seluruh teman teknik industri UNPAR angkatan 2013 terutama kelas C yang sama-sama berjuang untuk mendapatkan gelar sarjana telah memberikan motivasi dan semangat kepada penulis untuk menyelesaikan laporan skripsi.
11. Seluruh dosen dan staff pada teknik industri UNPAR yang telah membangun karakter penulis supaya dapat menyelesaikan laporan skripsi.
12. Pihak-pihak lainnya yang terlibat dalam pembuatan laporan skripsi penulis baik secara langsung maupun tidak langsung yang belum disebutkan.

Bandung, 11 Juli 2017

Christian Setiardjo

# DAFTAR ISI

<b>ABSTRAK</b> .....	i
<b>ABSTRACT</b> .....	ii
<b>KATA PENGANTAR</b> .....	iii
<b>DAFTAR ISI</b> .....	v
<b>DAFTAR TABEL</b> .....	vii
<b>DAFTAR GAMBAR</b> .....	ix
<b>DAFTAR NOTASI</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 Perumusan Masalah .....	I-3
I.3    Pembatasan Masalah .....	I-5
I.4    Tujuan Penelitian .....	I-5
I.5    Manfaat Penelitian .....	I-6
I.6    Metodologi Penelitian .....	I-6
I.7    Sistematika Penulisan .....	I-8
<b>BAB II TINJAUAN PUSTAKA</b> .....	II-1
II.1 <i>Multidimensional Knapsack Problem</i> .....	II-1
II.2    Metaheuristik .....	II-3
II.3 <i>Pigeon Inspired Optimization</i> .....	II-6
II.4 <i>One-Factor-At-a-Time</i> .....	II-11
II.5    Posisi Penelitian .....	II-12
<b>BAB III PERANCANGAN ALGORITMA</b> .....	III-1
III.1    Model Matematis .....	III-1
III.2 <i>Encoding dan Decoding</i> .....	III-2
III.3    Prosedur <i>Drop/Add Item</i> .....	III-4
III.4    Perhitungan Fungsi Objektif.....	III-8



III.5	Menghitung Fitness $p_t$ , Fitness $B_p$ , Fitness $g$ , $X_{Bpd}$ , dan $X_{gd}$ .	III-9
III.6	Menghitung <i>Map and Compass Operator</i> .....	III-10
III.7	Menghitung <i>Landmark Operator</i> .....	III-12
III.8	Penerapan <i>Pigeon Inspired Optimization</i> Pada <i>Multidimensional Knapsack Problem</i> Secara Keseluruhan.....	III-14
III.9	Verifikasi Program Algoritma.....	III-16
<b>BAB IV IMPLEMENTASI ALGORITMA</b> .....		IV-1
IV.1	Kasus <i>Multidimensional Knapsack Problem</i> .....	IV-1
IV.2	Penentuan Nilai Parameter Untuk Terbaik Melalui <i>One-Factor-At-a-Time (OFAT)</i> .....	IV-5
IV.2.1	Eksperimen OFAT Pada Kasus SENTO1.....	IV-5
IV.2.2	Eksperimen OFAT Pada Kasus WEING7.....	IV-10
IV.3	Implementasi Terhadap Seluruh Kasus .....	IV-16
IV.4	Perbandingan Performansi <i>Pigeon Inspired Optimization</i> , <i>Particle Swarm Optimization</i> , <i>Intelligent Water Drops</i> , dan <i>Genetic Algorithms</i> .....	IV-17
<b>BAB V ANALISIS</b> .....		V-1
V.1	Analisis <i>Pigeon Inspired Optimization</i> .....	V-1
V.2	Analisis Perancangan Algoritma .....	V-2
V.3	Analisis Implementasi Algoritma .....	V-7
<b>BAB VI KESIMPULAN DAN SARAN</b> .....		V-1
VI.1	Kesimpulan.....	V-1
VI.2	Saran.....	V-2

## DAFTAR PUSTAKA

## LAMPIRAN

## RIWAYAT HIDUP PENULIS

## DAFTAR NOTASI

$W_{ij}$	= Bobot <i>item</i> ke- $m$ pada <i>knapsack</i> $n$
$C_j$	= Kapasitas <i>knapsack</i> $n$
$P_i$	= Keuntungan <i>item</i> ke- $m$
$i$	= index <i>item</i>
$j$	= index <i>knapsack</i>
$m$	= jumlah <i>item</i>
$n$	= jumlah <i>knapsack</i>
$N_{c1max}$	= Jumlah iterasi maksimum untuk <i>map and compass operator</i>
$N_{c2max}$	= Jumlah iterasi maksimum untuk <i>landmark operator</i>
$N_p$	= Jumlah merpati
$R$	= Nilai <i>map and compass operator factor</i>
$D$	= Jumlah dimensi
Search range	= batasan untuk menjaga nilai posisi diantara 0-1 saja (batas bawah = 0 dan batas atas = 1)
$HUD_i$	= nilai HUD untuk <i>item</i> ke- $i$
$p$	= Indeks merpati
$d$	= Indeks dimensi
$t$	= Indeks iterasi
$V_{pdt}$	= Kecepatan merpati ke- $p$ pada dimensi ke- $d$ dalam iterasi ke- $t$
$X_{pdt}$	= Posisi merpati ke- $p$ pada dimensi ke- $d$ dalam iterasi ke- $t$
$SF_d$	= Nilai <i>sigmoid function</i> untuk dimensi ke- $d$
repair $p$	= Indeks penanganan <i>constraint</i>
$WSp_j$	= Bobot total merpati ke- $p$ di <i>knapsack</i> $j$ sesuai dengan keputusan <i>item</i> yang diambil
sisa $p_j$	= sisa kapasitas dari merpati ke- $p$ di <i>knapsack</i> $j$
dum $pd$	= Variabel pemuncil nilai HUD berdasarkan kebutuhan (melebihi <i>constraint</i> atau belum memenuhi)
sortdum	= Variabel pemanggil pengurutan nilai HUD untuk variabel dum $pd$

rank pd	= Indeks pemanggil urutan item yang harus ditambahkan /dibuang
repair1	= Variabel pembantu untuk membuang <i>item</i>
repair2	= Variabel pembantu untuk menambahkan <i>item</i>
fitness p(t)	= Nilai <i>fitness</i> dari merpati ke-p pada iterasi ke-t
fitnessB p	= <i>Fitness</i> terbaik merpati ke-p di seluruh iterasi untuk iterasi saat ini
fitnessBS p	= <i>Fitness</i> terbaik merpati ke-p di seluruh iterasi untuk iterasi sebelumnya
fitnessg t	= <i>Fitness</i> terbaik selama ini untuk seluruh merpati pada iterasi ke-t
XBpd	= Posisi terbaik merpati ke-p pada dimensi ke-d dalam seluruh iterasi yang telah dilalui
Xgd	= Posisi terbaik seluruh merpati pada dimensi ke-d dalam seluruh iterasi yang telah dilalui
indexfitnessg	= indeks pencari posisi fitnessg pada vektor fitnessB p pada iterasi ke-0
landmarkrank p1	= Indeks nilai <i>fitness</i> merpati untuk <i>landmark operator</i>
landmarkrank p2	= Indeks urutan merpati untuk <i>landmark operator</i>
LI	= Jumlah merpati pada iterasi tersebut pada <i>landmark operator</i>
Xcdt	= Posisi tengah dari seluruh merpati di dimensi ke-d pada iterasi ke-t
perubah z	= Variabel pemberi penalti pada <i>landmark operator</i>
z	= index variabel perubah
jml i	= Variabel pembantu untuk mencari penalti pada <i>landmark operator</i>

## DAFTAR TABEL

Tabel II.1	Pembagian Subjek Tiap Percobaan .....	II-11
Tabel II.2	Posisi Penelitian .....	II-13
Tabel III.1	<i>Weight per Item per Knapsack</i> .....	III-3
Tabel III.2	<i>Knapsack Capacity</i> .....	III-3
Tabel III.3	<i>Profit per Item</i> .....	III-3
Tabel III.4	Hasil Nilai Acak Posisi Merpati .....	III-3
Tabel III.5	Hasil Pembulatan Posisi Merpati .....	III-4
Tabel III.6	HUD per <i>Item</i> .....	III-5
Tabel III.7	HUD <i>Item</i> Terpilih.....	III-8
Tabel III.8	Hasil perhitungan Fungsi Objektif.....	III-8
Tabel III.9	Ruang Lingkup Kasus-Kasus Verifikasi dan Validasi.....	III-16
Tabel III.10	Hasil Verifikasi.....	III-16
Tabel III.11	Hasil Validasi.....	III-17
Tabel IV.1	Ruang Permasalahan Kasus Perbandingan .....	IV-1
Tabel IV.2	Kombinasi Parameter $N_{c1max}$ OFAT SENTO1 .....	IV-5
Tabel IV.3	Hasil OFAT $N_{c1max}$ SENTO1.....	IV-6
Tabel IV.4	Kombinasi Parameter $N_p$ OFAT SENTO1 .....	IV-7
Tabel IV.5	Hasil OFAT $N_p$ SENTO1 .....	IV-7
Tabel IV.6	Kombinasi Parameter $N_{c2max}$ OFAT SENTO1 .....	IV-8
Tabel IV.7	Hasil OFAT $N_{c2max}$ SENTO1.....	IV-8
Tabel IV.8	Kombinasi Parameter R OFAT SENTO1 .....	IV-9
Tabel IV.9	Hasil OFAT R SENTO1.....	IV-10
Tabel IV.10	Kombinasi Parameter $N_{c1max}$ OFAT WEING7.....	IV-11
Tabel IV.11	Hasil OFAT $N_{c1max}$ WEING7.....	IV-11
Tabel IV.12	Kombinasi Parameter NP OFAT WEING7.....	IV-12
Tabel IV.13	Hasil OFAT NP WEING7.....	IV-13
Tabel IV.14	Kombinasi Parameter $N_{c2max}$ OFAT WEING7.....	IV-14
Tabel IV.15	Hasil OFAT $N_{c2max}$ WEING7.....	IV-14
Tabel IV.16	Kombinasi Parameter R OFAT WEING7 .....	IV-15
Tabel IV.17	Hasil OFAT R WEING7 .....	IV-15

Tabel IV.18 Implementasi Kombinasi Parameter Pertama.....	IV-16
Tabel IV.19 Implementasi Kombinasi Parameter Kedua.....	IV-17
Tabel IV.20 Rekapitulasi Implementasi Seluruh Kasus.....	IV-17
Tabel IV.21 Perbandingan Performansi.....	IV-18
Tabel IV.22 Persentase Perbandingan Performansi .....	IV-18
Tabel IV.23 Persentase Rata-rata .....	IV-19

## DAFTAR GAMBAR

Gambar I.1 <i>Flowchart</i> Metodologi Penelitian.....	I-8
Gambar II.1 Model Optimasi .....	II-3
Gambar II.2 <i>Map and Compass Operator</i> .....	II-3
Gambar II.3 <i>Landmark Operator</i> .....	II-3
Gambar II.4 <i>Flowchart</i> PIO .....	II-4
Gambar II.5 Interaksi Menggunakan OFAT .....	II-12
Gambar III.1 <i>Flowchart Drop/Add Item</i> .....	III-7
Gambar III.2 <i>Flowchart</i> Menghitung Fitness pt, FitnessB pt, Fitnessg, Xpd, dan Xg.....	III-11
Gambar III.3 <i>Flowchart</i> Menghitung <i>Map and Compass Operator</i> .....	III-12
Gambar III.4 <i>Flowchart</i> Menghitung <i>Landmark Operator</i> .....	III-13
Gambar III.5 <i>Flowchart</i> Penerapan PIO Terhadap MKP Secara Keseluruhan .....	III-15

## DAFTAR LAMPIRAN

- LAMPIRAN A *SCRIPT* PROGRAM ALGORITMA
- LAMPIRAN B DATA KASUS VERIFIKASI DAN VALIDASI
- LAMPIRAN C HASIL VERIFIKASI DAN VALIDASI
- LAMPIRAN D DATA KASUS SENTO1
- LAMPIRAN E DATA KASUS SENTO2
- LAMPIRAN F DATA KASUS WEING1
- LAMPIRAN G DATA KASUS WEING2
- LAMPIRAN H DATA KASUS WEING3
- LAMPIRAN I DATA KASUS WEING4
- LAMPIRAN J DATA KASUS WEING5
- LAMPIRAN K DATA KASUS WEING6
- LAMPIRAN L DATA KASUS WEING7
- LAMPIRAN M DATA KASUS WEING8
- LAMPIRAN N KEPUTUSAN PENGAMBILAN *ITEM*

# BAB I

## PENDAHULUAN

Pada bab ini akan dijelaskan mengenai pendahuluan yang dilakukan dalam penelitian ini. Pendahuluan dalam penelitian ini adalah latar belakang penelitian, identifikasi dan rumusan masalah, batasan dan asumsi dalam penelitian, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan laporan skripsi.

### I.1 Latar Belakang Permasalahan

Permasalahan mengenai alokasi sumber daya banyak ditemukan di dunia nyata. Dimulai dari permasalahan-permasalahan sederhana seperti barang-barang apa saja yang ingin dibeli pada saat ke swalayan dan memilih barang-barang apa saja yang akan dibawa ke dalam tas hingga permasalahan-permasalahan dalam dunia bisnis dan industri seperti *capital budgeting* dan *project selection*. Alokasi sumber daya ini menjadi suatu permasalahan dikarenakan sumber daya yang terbatas. Melalui pengalokasian sumber daya yang efektif dan efisien, permasalahan mengenai keterbatasan sumber daya ini dapat terselesaikan. Dalam dunia optimasi permasalahan pengalokasian sumber daya ini biasa termodelkan dalam *knapsack problem*.

*Knapsack problem* merupakan sebuah kasus optimasi kombinatorial. Model *knapsack problem* ini diceritakan sebagai sebuah tas/wadah/*knapsack* dengan kapasitas tertentu yang nantinya akan diisi oleh berbagai macam barang. Setiap barang memiliki bobot/*weight* dan keuntungan/*profit* yang telah ditetapkan. Tujuan akhir dari permasalahan ini adalah menentukan barang apa saja yang diambil untuk memenuhi kapasitas tas dengan keuntungan yang sebesar-besarnya (Martello & Toth, 1990).

Banyak permasalahan yang sering ditemukan sehari-hari dapat dianalogikan dengan model *knapsack problem* ini. Contoh yang paling dekat adalah pada saat ingin melakukan bersih-bersih gudang yang terlalu penuh. Pada kegiatan menyimpan barang dalam gudang, gudang dianalogikan sebagai *knapsack* dan barang-barang yang berada di dalamnya dianalogikan sebagai



benda yang akan masuk ke dalam *knapsack*. Gudang memiliki kapasitas tertentu dan barang-barang memiliki ukuran yang dianalogikan sebagai bobot dari benda dalam model tersebut. Hal itu dikarenakan seluruh barang tidak dapat dimasukkan dalam gudang, maka akan dipilih barang-barang yang jika dimasukkan dapat memenuhi kapasitas gudang serta memiliki keuntungan yang sebesar-besarnya. Keuntungan dalam kasus ini dapat dianalogikan sebagai manfaat/kegunaan dari barang tersebut.

Jika dipelajari lebih dalam, model optimasi *knapsack problem* memiliki banyak perkembangan variasi. Variasi-variasi dari *knapsack problem* ini memiliki perbedaan pada kondisi kasus serta tujuan dari permasalahan. Beberapa contoh variasi dari *knapsack problem* adalah *single knapsack problem*, *multiple knapsack problem*, *bounded knapsack problem*, *multidimensional knapsack problem*, *unbounded knapsack problem*, *knapsack sharing problem*, dan lain-lain (Lin, 1998). *Multidimensional knapsack problem* atau disingkat MKP merupakan model pengembangan dari *knapsack problem* secara umum. MKP merupakan salah satu permasalahan optimasi kombinatorial yang juga termasuk dalam permasalahan *NP-hard* (Labeled, Gherboudj, & Chikhi, 2011). Menurut Chu dan Beasley (1998), MKP merupakan salah satu model *knapsack problem* yang memiliki lebih dari 1 wadah yang setiap barangnya memiliki bobot yang berbeda untuk setiap wadah. Selain itu, setiap barang yang diambil harus masuk kedalam setiap wadah. Setiap keputusan pengambilan barang harus memperhatikan kapasitas setiap wadah dan bobot yang diberikan kepada wadah tersebut. Tujuan utama dari MKP adalah memaksimalkan keuntungan dengan mengambil keputusan barang apa saja yang akan diambil sesuai dengan kapasitas tiap wadah.

Berdasarkan Lorie dan Savage (1955) dalam Drake, Hyde, Ibrahim, dan Ozcan (2014), model MKP pertama kali muncul untuk menghadapi permasalahan *capital budgeting*. Kasus mengenai *capital budgeting* ini kemudian dapat dikembangkan menjadi kasus-kasus yang hampir sama. Salah satu contoh kasus penggunaan MKP adalah penentuan proyek perusahaan yang akan dilaksanakan berdasarkan *budget* yang telah ditentukan. Setiap proyek akan diketahui mengeluarkan berapa banyak uang untuk jumlah tahun tertentu dan diketahui juga keuntungan yang didapat dari tiap proyek tersebut. Dalam kasus ini jumlah tahun adalah analogi dari wadah pada model. Jumlah uang yang

dikeluarkan untuk proyek tersebut selama tahun tertentu menjadi bobot dari model. *Budget* yang dapat dikeluarkan selama tahun tertentu menjadi kapasitas dari wadah pada model. Keuntungan yang didapat dari menjalankan proyek menjadi keuntungan pada model.

Beberapa contoh kasus MKP telah diselesaikan menggunakan metode analitik, seperti *dynamic programming* (Gilmore & Gomory, 1966), *branch and bound* (Shih, 1979), dan lain-lain. Selain itu, juga terdapat beberapa penyelesaian menggunakan metode *metaheuristic* seperti *genetic algorithm* (Khuri, Back, dan Heitkotter, 1994), *Particle Swarm Optimization* (Hembecker, Lopes, dan Godoy Jr., 2007), *Ant Colony Optimization* (Ke, Feng, Ren, & Wei, 2010), *Artificial Bee Colony Algorithm* (Sundar, Singh, & Rossi, 2010), dan lain-lain. Pengerjaan MKP dengan metode analitik ini akan memakan banyak waktu bila ruang lingkup permasalahan yang diteliti juga semakin meluas, karena MKP merupakan permasalahan yang bersifat *NP-hard* atau *Non-deterministic Polynomial-time hard* (Ding, Kang, Hu, dan Wang, 2012). Karena hal itu, maka akan dicoba pengerjaan MKP dengan salah satu algoritma *metaheuristic*, yaitu *Pigeon Inspired Optimization*.

## **I.2 Identifikasi dan Rumusan Masalah**

MKP telah diselesaikan dengan beberapa metode analitik seperti yang telah disebutkan di subbab sebelumnya. Penyelesaian dengan metode analitik ini berguna untuk mendapatkan solusi optimal. Tetapi seperti yang telah disebutkan juga MKP termasuk dalam kasus *NP-hard*, dimana dengan semakin kompleksnya ruang lingkup permasalahan, maka penyelesaian dengan metode analitik akan memakan waktu komputasi yang sangat lama. Selain itu, dapat diketahui juga karena MKP merupakan sebuah permasalahan kombinatorial. Dengan semakin banyak benda pada kasus, maka semakin banyak juga kombinasi solusi yang mungkin terjadi. Karena hal-hal tersebut, diperlukan pendekatan lain untuk mempersingkat waktu pengerjaan.

Pendekatan yang dapat dilakukan untuk menyelesaikan MKP adalah dengan menggunakan metode *heuristic* dan *metaheuristic*. Metode *heuristic* adalah suatu metode yang menjamin solusi yang cukup memuaskan atau mendekati optimal. Metode *heuristic* melakukan perhitungan secara masuk akal tanpa menjamin sebuah solusi yang optimal, namun mungkin juga menghasilkan

solusi yang tidak *feasible* (Rayward-Smith, Osman, Reeves, & Smith, 1996). Metode *metaheuristic* merupakan metode lanjutan yang berbasis dari metode *heuristic* yang dapat menyelesaikan sebuah permasalahan optimasi melalui sebuah pendekatan (Talbi, 2009). Beberapa metode *heuristic* dan *metaheuristic* telah digunakan untuk menyelesaikan masalah MKP seperti yang disebutkan di subbab sebelumnya. Karena sudah banyaknya metode-metode yang digunakan untuk menyelesaikan MKP, maka ada baiknya untuk mencoba metode lainnya untuk dibandingkan apakah hasil yang didapatkan bisa lebih baik.

*Pigeon Inspired Optimization* atau disingkat PIO merupakan sebuah algoritma dengan metode *metaheuristic* yang terinspirasi dari perilaku burung merpati yang memiliki *homing ability* (Duan & Qiao, 2014). Dari zaman dulu merpati sudah terkenal sebagai pengirim surat, karena merpati dapat mengetahui arah kembali pulang walaupun pergi kemanapun. PIO ini pertama kali digunakan untuk menyelesaikan permasalahan *air robot path planning* (Duan & Qiao, 2014). Menurut Li & Duan (2014), PIO merupakan salah satu algoritma yang lebih cepat konvergen dibanding dengan algoritma-algoritma lainnya (*Genetic Algorithm*, *Artificial Bee Colony*, dan lain-lain) pada kasus *Unnamed Aerial Vehicles*. PIO merupakan salah satu algoritma *metaheuristic* yang tergolong dalam *population-based swarm intelligence*. Algoritma-algoritma yang telah disebutkan disubbab sebelumnya seperti *particle swarm optimization*, *ant colony optimization*, dan *artificial bee colony algorithm* juga merupakan algoritma *metaheuristic* yang tergolong dalam *population-based swarm intelligence* seperti PIO. Algoritma-algoritma bertipe *population-based swarm intelligence* tersebut telah banyak diterapkan untuk permasalahan MKP dan menghasilkan hasil yang dapat dibilang baik. Karena itu, PIO juga diharapkan untuk menghasilkan hasil yang baik juga dalam penerapannya terhadap MKP seperti algoritma-algoritma *population-based swarm intelligence* lainnya. Selain itu, PIO juga telah diterapkan ke permasalahan transportasi (Septiano, 2016). Penerapan PIO pada permasalahan transportasi sangat memuaskan. MKP dan permasalahan transportasi memiliki persamaan, yaitu permasalahan bertipe diskrit.

Dalam setiap algoritma *metaheuristic* terdapat beberapa parameter atau nilai-nilai yang ditentukan sebagai *input*. Dalam kasus PIO terdapat 4 buah parameter, yaitu  $N_p$  (jumlah *pigeon*),  $R$  (faktor *map & compass*),  $N_{c1_{max}}$  (jumlah generasi/iterasi maksimal pada operator *map & compass*), dan  $N_{c2_{max}}$  (jumlah

generasi/iterasi maksimal pada operator *landmark*). Nilai dari parameter-parameter tersebut dapat mempengaruhi *output* dari algoritma PIO ini. Sesuai dengan yang telah dijabarkan, berikut adalah rumusan masalah yang ada dalam penelitian ini :

1. Bagaimana penerapan algoritma *Pigeon Inspired Optimization* untuk kasus *Multidimensional Knapsack Problem*?
2. Bagaimana hasil perbandingan solusi dari penggunaan algoritma *Pigeon Inspired Optimization* terhadap algoritma *Particle Swarm Algorithm* (Hembecker, Lopes, dan Godoy Jr., 2007), *Intelligent Water Drops Algorithm* (Shah-Hosseini, 2009), *Genetic Algorithm* (Khuri, Back, dan Heitkotter, 1994), dan solusi terbaik yang diketahui?
3. Berapa nilai parameter terbaik dalam penerapan *Pigeon Inspired Optimization* pada *Multidimensional Knapsack Problem*?

### **I.3 Pembatasan Masalah**

Batasan masalah yang digunakan dalam penelitian kali ini adalah sebagai berikut :

1. Waktu komputasi tidak menjadi ukuran performansi.
2. Permasalahan MKP yang diujikan pada algoritma *Pigeon Inspired Optimization* ini menggunakan kasus *benchmark global* yang sering digunakan dipenelitian lain.

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

Berdasarkan rumusan masalah yang telah disebutkan pada subbab identifikasi dan rumusan masalah, berikut adalah tujuan dalam penelitian ini :

1. Menerapkan algoritma *Pigeon Inspired Optimization* pada kasus *Multidimensional Knapsack Problem*
2. Mengetahui perbandingan solusi algoritma *Pigeon Inspired Optimization* dengan algoritma *Particle Swarm Algorithm* (Hembecker, Lopes, dan Godoy Jr., 2007), *Intelligent Water Drops Algorithm* (Shah-Hosseini, 2009), *Genetic Algorithm* (Khuri, Back, dan Heitkotter, 1994), dan solusi terbaik yang diketahui.
3. Mengetahui nilai parameter terbaik dalam penerapan algoritma *Pigeon Inspired Optimization* pada *Multidimensional Knapsack Problem*.

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

Dengan memenuhi tujuan penelitian, maka manfaat dari penelitian ini adalah sebagai berikut :

1. Menambah pengetahuan dan wawasan terhadap penerapan algoritma *Pigeon Inspired Optimization* dalam penyelesaian kasus *Multidimensional Knapsack Problem*.
2. Dapat menjadi referensi untuk penelitian yang membahas mengenai *Pigeon Inspired Optimization* dan/atau *Multidimensional Knapsack Problem*.

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

Metodologi penelitian merupakan prosedur yang akan dilakukan secara bertahap dalam penelitian. Metodologi penelitian ini dapat dilihat dalam bentuk *flowchart*. Gambar 4 menunjukkan *flowchart* dari metodologi penelitian. Berikut adalah metodologi penelitian yang akan digunakan dalam penelitian ini :

1. Studi Literatur  
Langkah awal dalam penelitian adalah melakukan studi literatur. Studi literatur dilakukan dengan cara mencari referensi-referensi melalui buku/jurnal/media lainnya untuk mengetahui lebih dalam mengenai *Multidimensional Knapsack Problem* dan *Pigeon Inspired Optimization*.
2. Identifikasi dan Perumusan Masalah  
Setelah studi literatur dilakukan, selanjutnya perlu dilakukan identifikasi terhadap permasalahan yang timbul. Dari permasalahan yang sudah diidentifikasi tersebut, kemudian dibuat rumusan masalah yang perlu untuk dicari solusinya.
3. Pembatasan Masalah  
Pembatasan masalah dan pemberian asumsi berguna untuk membuat penelitian yang dijalankan tetap terfokus pada permasalahan yang ingin diselesaikan dan dapat menghasilkan solusi yang sesuai untuk diterapkan pada permasalahan yang diteliti.

4. Penentuan Tujuan dan Manfaat Penelitian

Tujuan dan manfaat penelitian perlu ditentukan terlebih dahulu agar penelitian yang dilakukan dapat dipastikan memiliki tujuan yang jelas dan manfaat yang berguna.

5. Perancangan Algoritma *Pigeon Inspired Optimization* untuk *Multidimensional Knapsack Problem*

Langkah selanjutnya yang perlu dilakukan adalah perancangan dan pengembangan terhadap algoritma yang ingin digunakan, yaitu *Pigeon Inspired Optimization* agar dapat menyelesaikan permasalahan *Multidimensional Knapsack Problem*.

6. Verifikasi Algoritma

Algoritma yang sudah dibuat, kemudian perlu dilakukan verifikasi untuk melihat apakah algoritma yang dirancang dapat menghasilkan solusi untuk permasalahan *Multidimensional Knapsack Problem*.

7. Validasi Algoritma

Algoritma yang telah terverifikasi, kemudian akan dilakukan validasi untuk melihat apakah algoritma yang dirancang dapat menghasilkan solusi yang benar atau sesuai yang diinginkan untuk permasalahan *Multidimensional Knapsack Problem*.

8. Penerapan Algoritma Pada Kasus *Multidimensional Knapsack Problem*

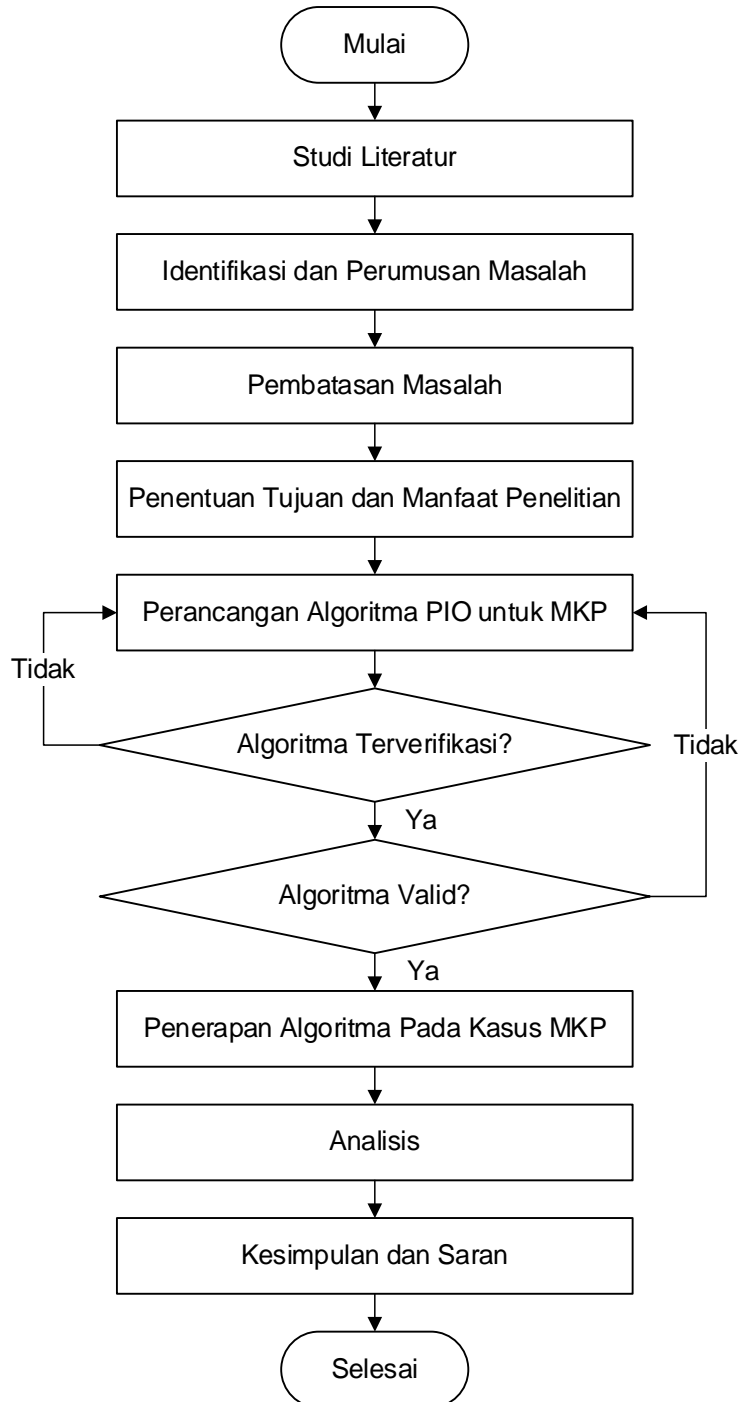
Algoritma kemudian dijalankan menggunakan program untuk seluruh kasus *Multidimensional Knapsack Problem* berdasarkan rancangan yang telah terverifikasi untuk mendapatkan solusi berdasarkan parameter-parameter dari algoritma tersebut.

9. Analisis

Solusi yang muncul dari penerapan algoritma ke program kemudian diteliti untuk melihat berapa nilai parameter yang terbaik serta membandingkan solusi dengan algoritma lain pada kasus-kasus *Multidimensional Knapsack Problem*.

10. Kesimpulan dan Saran

Setelah semua hal diatas telah dilakukan, dibuat kesimpulan untuk menjawab rumusan masalah yang telah dibuat pada awal penelitian. Dibuat juga saran untuk memberikan masukan pada penelitian selanjutnya.



Gambar I.1 *Flowchart* Metodologi Penelitian

## I.7 Sistematika Penulisan

Sistematika penulisan menunjukkan bagaimana penulisan laporan pada penelitian ini. Penulisan laporan dibagi menjadi beberapa bab yang akan dijelaskan selanjutnya. Berikut adalah sistematika penulisan laporan penelitian yang akan digunakan :

## **BAB I PENDAHULUAN**

Dalam bab ini akan dijelaskan mengenai latar belakang penelitian, identifikasi masalah dan rumusan masalah dari penelitian, pembatasan masalah penelitian, tujuan penelitian, manfaat penelitian, metodologi penelitian, serta sistematika penulisan laporan untuk penelitian ini.

## **BAB II TINJAUAN PUSTAKA**

Dalam bab ini akan dibahas mengenai teori dari algoritma yang digunakan dan permasalahan yang digunakan dalam penelitian. Algoritma yang dibahas dalam bab ini adalah *Pigeon Inspired Optimization*. Permasalahan yang dibahas dalam bab ini adalah *Multidimensional Knapsack Problem*. Selain itu, akan dibahas sekilas juga mengenai metaheuristik.

## **BAB III PERANCANGAN ALGORITMA**

Dalam bab ini akan dibahas bagaimana cara merancang algoritma *Pigeon Inspired Optimization* untuk dapat menyelesaikan *Multidimensional knapsack problem*. Kemudian rancangan tersebut akan dibuat program menggunakan *software* matlab yang akan dilakukan verifikasi dalam bab ini.

## **BAB IV IMPLEMENTASI ALGORITMA**

Dalam bab ini akan dilakukan implementasi rancangan algoritma yang telah dibuat pada bab sebelumnya untuk kasus-kasus pembandingan. Setelah itu, akan dibandingkan performansinya terhadap algoritma-algoritma lain dengan kasus-kasus pembandingan yang sama.

## **BAB V ANALISIS**

Dalam bab ini akan dibahas mengenai analisis dari algoritma yang digunakan dalam penelitian, bagaimana perancangan algoritma yang telah dilakukan, dan bagaimana implementasi algoritma yang telah dilakukan.

## **BAB VI KESIMPULAN DAN SARAN**

Dalam bab ini akan dibahas kesimpulan dari penerapan *Pigeon Inspired Optimization* pada *Multidimensional Knapsack Problem* dan saran-saran.



Kesimpulan yang dimaksud disini adalah menjawab tujuan penelitian. Saran-saran diberikan untuk penelitian-penelitian selanjutnya.