



**PENERAPAN ALGORITMA *ARTIFICIAL BEE COLONY*  
PADA PERMASALAHAN *MULTIPLE CONSTRAINTS*  
*BOUNDED KNAPSACK***

**SKRIPSI**

Oleh

**Aulia Nandarema Hayyu  
NIM 121810101066**

**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS JEMBER  
2016**



**PENERAPAN ALGORITMA *ARTIFICIAL BEE COLONY*  
PADA PERMASALAHAN *MULTIPLE CONSTRAINTS*  
*BOUNDED KNAPSACK***

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan Program Studi Matematika (S1)  
dan mencapai gelar Sarjana Sains

Oleh

**Aulia Nandarema Hayyu**  
**NIM 121810101066**

**JURUSAN MATEMATIKA**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS JEMBER**  
**2016**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ayahanda Misdi dan Ibunda Lilik yang selalu memberikan kasih sayang, doa, nasihat serta bimbingan selama perjalanan hidup;
2. Adikku Aulia Nandarema Qayyum yang selalu memberikan dukungan;
3. Guru ngajiku Ibu Masnu'ah Fathul Bary yang senantiasa sabar dalam membimbing;
4. Guru-guru sejak taman kanak-kanak sampai perguruan tinggi yang telah membimbing dan memberikan ilmu yang bermanfaat;
5. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember, SMAN 3 Jember, SMPN 11 Jember, SDN Karangrejo 2, dan TK Diponegoro.

**MOTTO**

Man Jadda Wajada

Jangan takut jatuh, karena yang tidak pernah memanjatlah yang tidak pernah jatuh. Jangan takut gagal, karena yang tidak pernah gagal hanyalah orang-orang yang tidak pernah melangkah. Jangan takut salah, karena dengan kesalahan yang pertama kita dapat menambah pengetahuan untuk mencari jalan yang benar pada langkah kedua.  
(Buya Hamka<sup>\*)</sup>)

---

<sup>\*)</sup> A.K. 2014. *Aku Doamu*. Jakarta: PT. Gramedia

**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

nama : Aulia Nandarema Hayyu

nim : 121810101066

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penerapan Algoritma *Artificial Bee Colony* pada Permasalahan *Multiple Constraints Bounded Knapsack*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 24 Juni 2016

Yang menyatakan,

Aulia Nandarema Hayyu

NIM 121810101066

**SKRIPSI**

**PENERAPAN ALGORITMA *ARTIFICIAL BEE COLONY* PADA  
PERMASALAHAN *MULTIPLE CONSTRAINTS BOUNDED KNAPSACK***

Oleh

Aulia Nandarema Hayyu  
NIM 121810101066

Pembimbing :

Dosen Pembimbing Utama : M. Ziaul Arif, S.Si., M.Sc

Dosen Pembimbing Anggota : Kusbudiono, S.Si., M.Si

**PENGESAHAN**

Skripsi berjudul “Penerapan Algoritma *Artificial Bee Colony* pada Permasalahan *Multiple Constraints Bounded Knapsack*” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas  
Jember

**Tim Penguji:**

Ketua,

Sekretaris,

M. Ziaul Arif, S.Si., M.Sc  
NIP. 198501112008121002

Kusbudiono, S.Si., M.Si  
NIP. 197704302005011001

Anggota I,

Anggota II,

Ahmad Kamsyakawuni, S.Si, M.Kom  
NIP. 197211291998021001

Dian Anggraeni, S.Si., M.Si  
NIP. 198202162006042002

Mengesahkan  
Dekan,

Drs. Sujito, Ph.D.  
NIP.196102041987111001

## RINGKASAN

**Penerapan Algoritma *Artificial Bee Colony* pada Permasalahan *Multiple Constraints Bounded Knapsack***; Aulia Nandarema Hayyu, 121810101066; 2016: 40 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

*Knapsack* merupakan suatu permasalahan bagaimana memilih objek dari sekian banyak dan berapa besar objek tersebut akan disimpan sehingga diperoleh suatu penyimpanan yang optimal dengan memperhatikan objek yang terdiri dari  $n$  objek (1, 2, 3, ...,  $n$ ) dimana setiap objek memiliki bobot ( $w_i$ ) dan nilai *profit* atau keuntungan ( $p_i$ ) dengan memperhatikan juga kapasitas dari media penyimpanan sebesar  $W$ . Salah satu jenis *knapsack* yaitu *bounded knapsack* dimana setiap barang atau objek dapat diambil sebagian atau seluruhnya. Tiap-tiap *knapsack* biasanya terdapat permasalahan salah satunya *multiple constraint* dimana kendala yang didapat lebih dari satu.

Penelitian ini menggunakan data sekunder yang diambil dari skripsi Hadi (2015) dimana data tersebut diambil dari UD. Buah Barokah. Data tersebut terdiri dari nama barang, jumlah kemasan tiap barang, berat, dan keuntungan. Pada penelitian ini menggunakan algoritma *artificial bee colony*. Algoritma tersebut menirukan tingkah laku sekawanan lebah yang mencari sumber makanan. Permasalahan *knapsack* yang telah diterapkan pada algoritma *artificial bee colony* menghasilkan keuntungan yang maksimal sebesar Rp. 3.335.400,- dengan jumlah populasi dan jumlah iterasi 1000 dan 2000.

Semakin besar jumlah iterasi dan jumlah populasi maka semakin mudah mendapatkan solusi optimal yang diinginkan. Oleh karena itu, parameter jumlah populasi dan jumlah iterasi yang sangat berpengaruh terhadap solusi optimal.



## PRAKATA

Puji syukur ke hadirat Allah SWT, atas segala rahmat dan karunia-Nya sehingga penulis menyelesaikan skripsi yang berjudul “Penerapan Algoritma *Artificial Bee Colony* pada Permasalahan *Multiple Constraints Bounded Knapsack*”. Shalawat beserta salam semoga tetap tercurahkan kepada Nabi Muhammad SAW.

Skripsi ini diajukan untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember. Penulisan skripsi ini tidak lepas dari hambatan dan kesulitan, namun berkat bimbingan, bantuan, nasihat, saran, dan kerjasama dari berbagai pihak, akhirnya segala hambatan tersebut dapat diatasi dengan baik. Oleh karena itu, penulis menyampaikan terima kasih kepada:

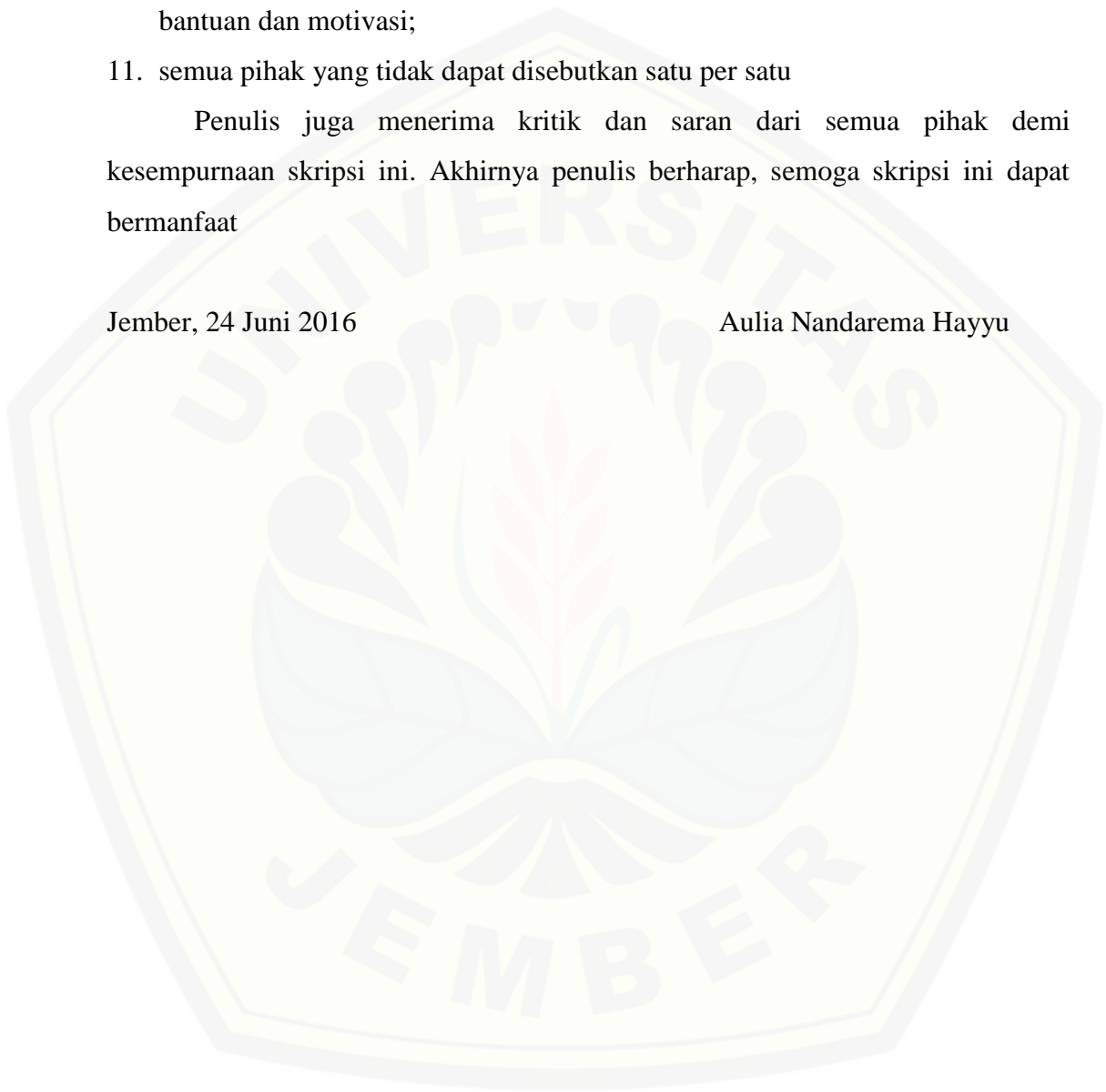
1. M. Ziaul Arif, S.Si., M.Sc., selaku Dosen Pembimbing Utama, Kusbudiono, S.Si., M.Si., selaku Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
2. Kusbudiono, S.Si., M.Si., selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
3. Ahmad Kamsyakawuni, S. Si., M. Kom., selaku Dosen Penguji I dan Dian Anggraeni, S.Si., M.Si., selaku Dosen Penguji II yang telah meluangkan waktu dan memberikan kritik serta saran demi kesempurnaan skripsi ini;
4. dosen dan karyawan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
5. Ayahanda Misdi dan Ibunda Lilik yang selalu memberikan dukungan doa dan senantiasa memberikan semangat;
6. Adikku tercinta Aulia Nandarema Qayyum yang selalu memberikan dukungan;
7. Keluarga besar TPQ Al-Ma’unah yang selalu memberikan motivasi;

8. sahabatku Desi, Reni, Ike, Ninta, Rafika, Nita, dan masih banyak lagi yang selalu memberikan dukungan, motivasi, dan semangat;
9. teman seperjuanganku Anggun dan Erni yang telah membantu analisis;
10. teman-teman BHATICS'12 dan teman-teman SMAN 3 Jember memberikan bantuan dan motivasi;
11. semua pihak yang tidak dapat disebutkan satu per satu

Penulis juga menerima kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat

Jember, 24 Juni 2016

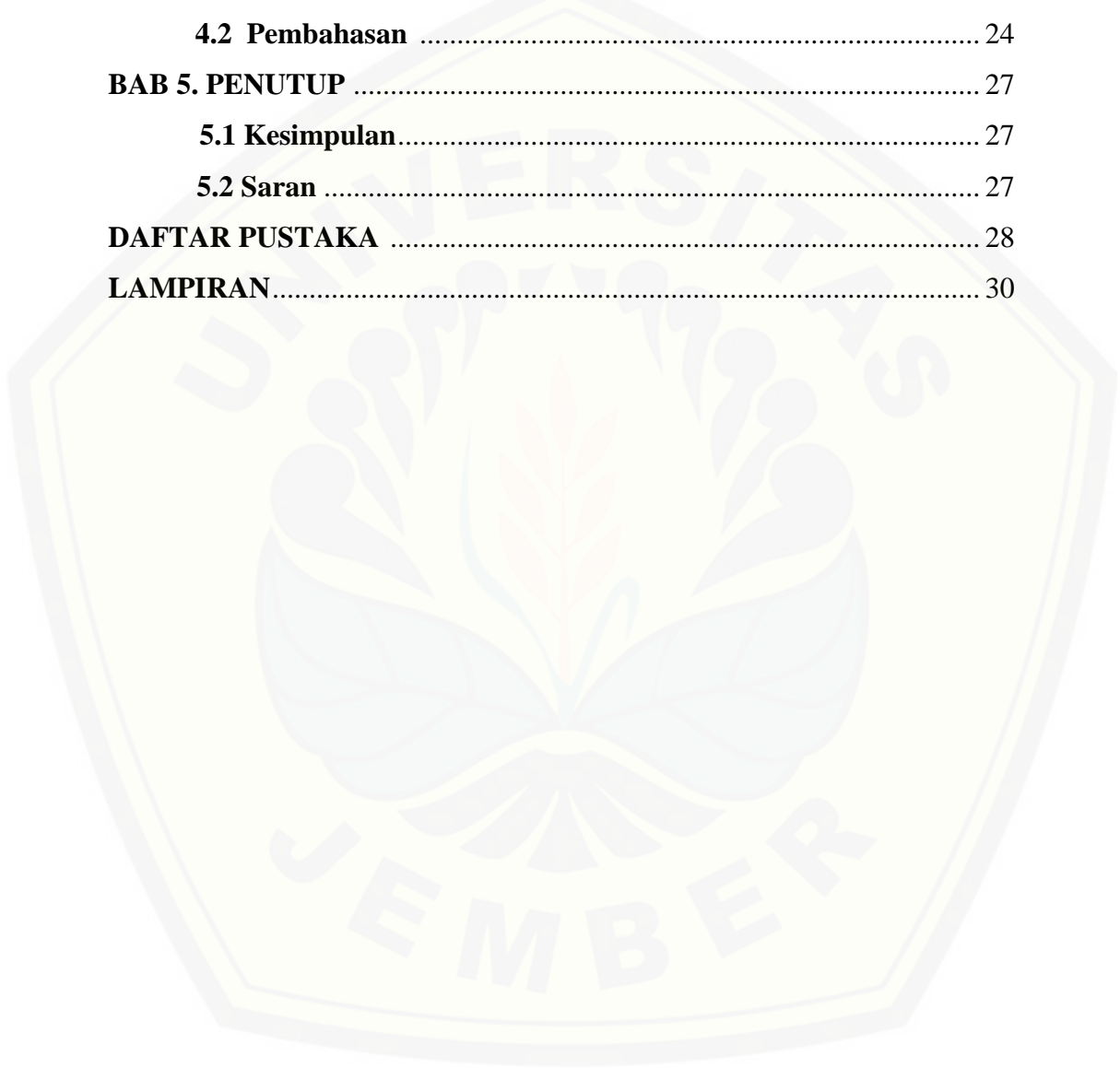
Aulia Nandarema Hayyu



DAFTAR ISI

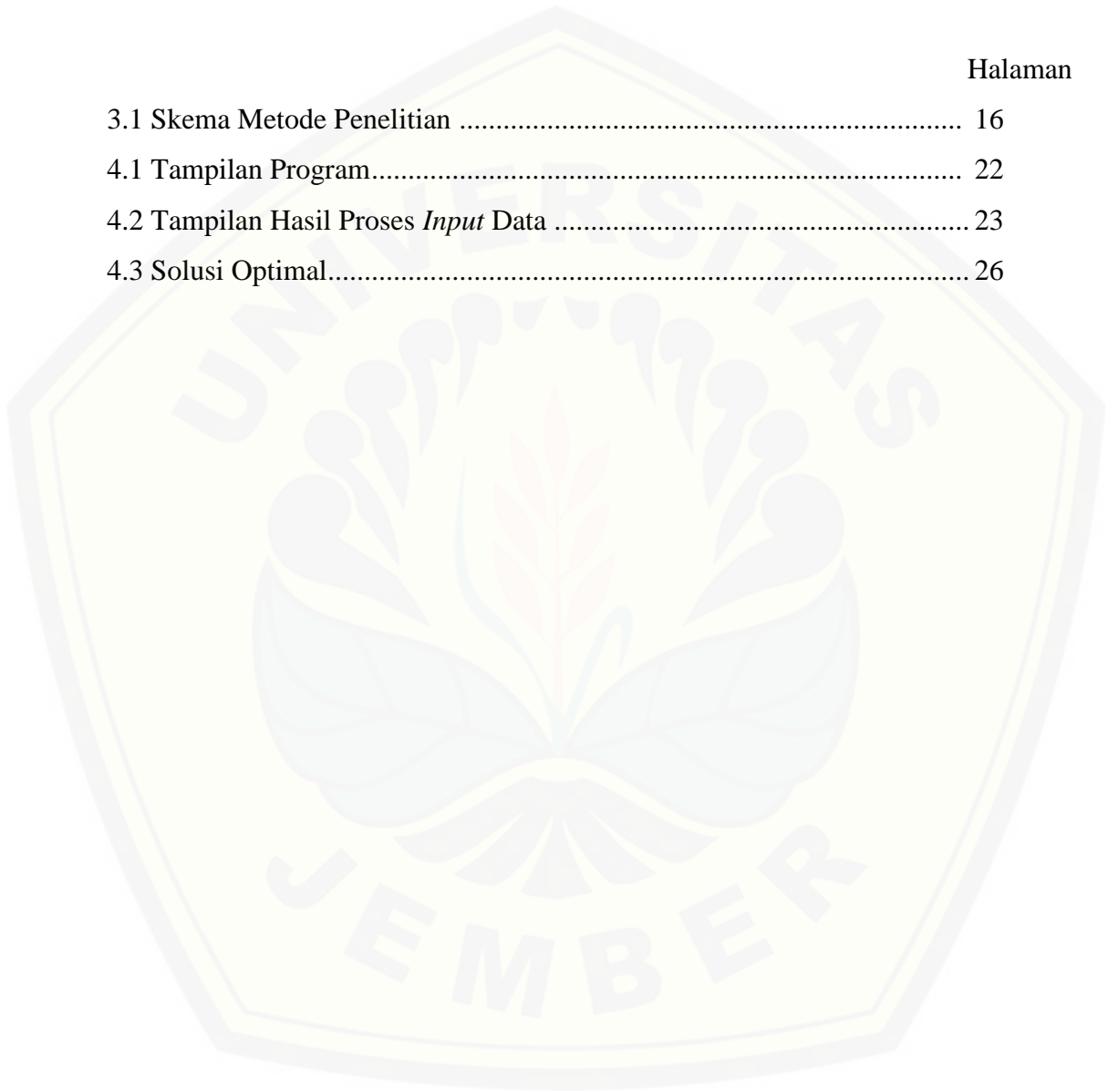
	Halaman
HALAMAN JUDUL .....	i
HALAMAN PERSEMBAHAN .....	ii
HALAMAN MOTTO .....	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN PENGESAHAN.....	vi
RINGKASAN .....	vii
PRAKATA.....	viii
DAFTAR ISI .....	x
DAFTAR GAMBAR .....	xii
DAFTAR TABEL .....	xiii
DAFTAR LAMPIRAN.....	xiv
<b>BAB 1. PENDAHULUAN .....</b>	<b>1</b>
<b>1.1 Latar Belakang .....</b>	<b>1</b>
<b>1.2 Rumusan Masalah .....</b>	<b>3</b>
<b>1.3 Batasan Masalah .....</b>	<b>3</b>
<b>1.4 Tujuan .....</b>	<b>3</b>
<b>1.5 Manfaat .....</b>	<b>3</b>
<b>BAB 2. TINJAUAN PUSTAKA .....</b>	<b>4</b>
<b>2.1 Optimasi .....</b>	<b>4</b>
<b>2.2 Knapsack .....</b>	<b>4</b>
2.2.1 <i>Bounded Knapsack</i> .....	6
2.2.2 <i>Multiple Constraints Knapsack Problem</i> .....	7
<b>2.3 Algoritma .....</b>	<b>9</b>
<b>2.4 Algoritma <i>Artificial Bee Colony</i> .....</b>	<b>9</b>
<b>BAB 3. METODE PENELITIAN .....</b>	<b>14</b>
<b>3.1 Data Penelitian .....</b>	<b>14</b>
<b>3.2 Langkah-Langkah Penelitian .....</b>	<b>15</b>
<b>BAB 4. HASIL DAN PEMBAHASAN .....</b>	<b>17</b>

<b>4.1 Hasil</b> .....	17
4.1.1 Penyelesaian Manual dengan Algoritma <i>Artificial Bee Colony</i> .....	17
4.1.2 Penyelesaian Menggunakan Program dengan Algoritma <i>Artificial Bee Colony</i> .....	21
<b>4.2 Pembahasan</b> .....	24
<b>BAB 5. PENUTUP</b> .....	27
<b>5.1 Kesimpulan</b> .....	27
<b>5.2 Saran</b> .....	27
<b>DAFTAR PUSTAKA</b> .....	28
<b>LAMPIRAN</b> .....	30



**DAFTAR GAMBAR**

	Halaman
3.1 Skema Metode Penelitian .....	16
4.1 Tampilan Program.....	22
4.2 Tampilan Hasil Proses <i>Input</i> Data .....	23
4.3 Solusi Optimal.....	26

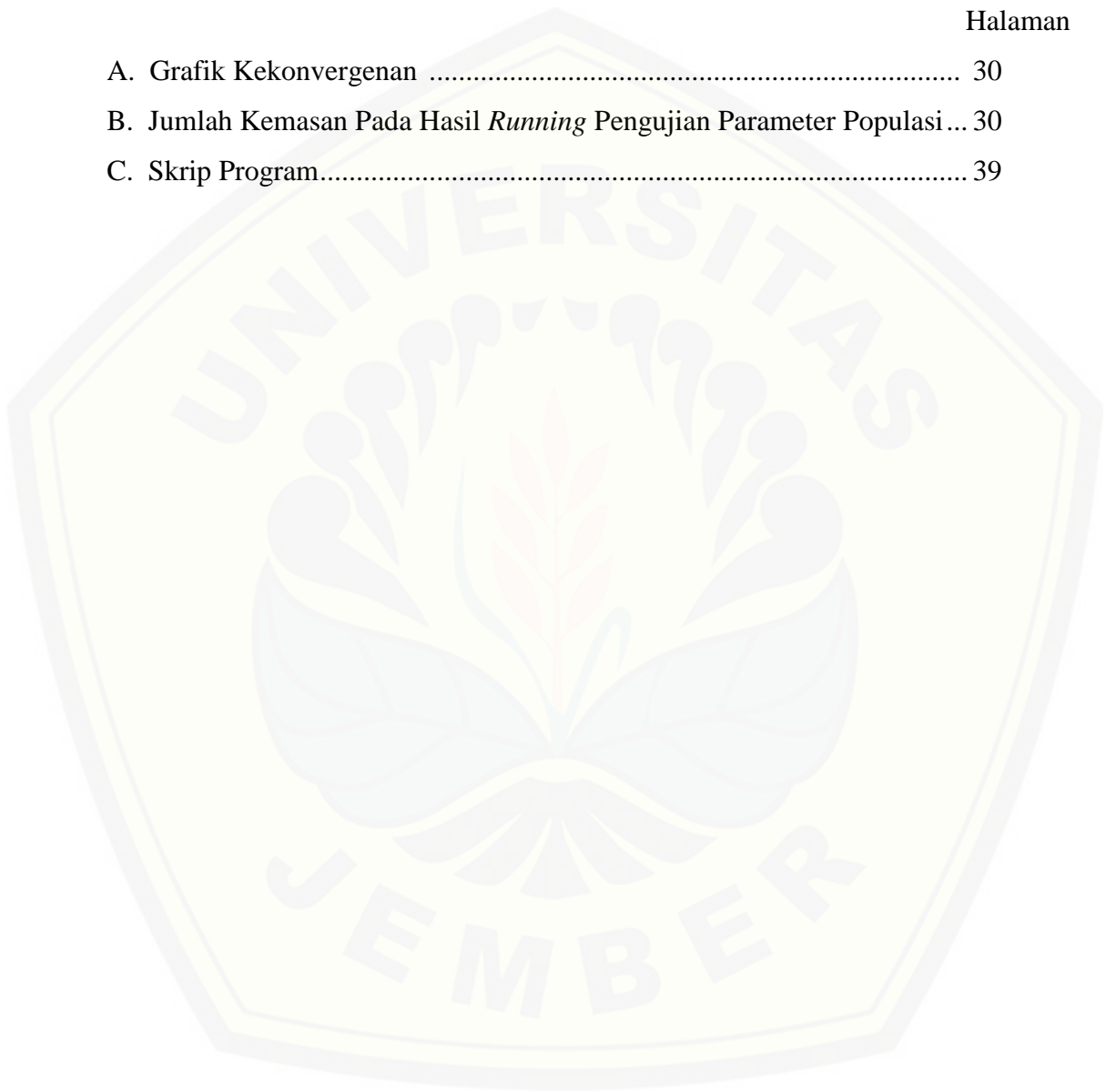


**DAFTAR TABEL**

	Halaman
3.1 Data Penelitian .....	14
4.1 Data Kecil Buah-buahan .....	18
4.2 Data Harga Jual dan Harga Beli.....	18
4.3 Solusi awal .....	19
4.4 Solusi Alternatif .....	19
4.5 Hasil Perhitungan .....	21
4.6 Data Hasil Proses .....	23
4.7 Hasil <i>Running</i> Pengujian Parameter Populasi dan Iterasi.....	25

**DAFTAR LAMPIRAN**

	Halaman
A. Grafik Kekonvergenan .....	30
B. Jumlah Kemasan Pada Hasil <i>Running</i> Pengujian Parameter Populasi...	30
C. Skrip Program.....	39



## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Optimasi adalah salah satu disiplin ilmu dalam matematika yang fokus untuk mendapatkan nilai minimum atau maksimum secara sistematis dari suatu fungsi, peluang maupun pencarian nilai lainnya dalam berbagai kasus. Optimasi sangat berguna di hampir segala bidang dalam rangka melakukan usaha secara efektif dan efisien untuk mencapai target hasil yang ingin dicapai, sehingga optimasi sangat penting dalam persaingan di dunia industri yang sudah sangat ketat di segala bidang yang ada. Banyak kasus optimasi yang membutuhkan solusi untuk mendapatkan dan menghasilkan *output* yang optimal. Beberapa masalah optimasi diantaranya *Travelling Salesman Problem (TSP)*, *Minimum Spanning Tree (MST)*, dan *Knapsack Problem*. Kasus optimasi tersebut membahas tentang masalah pendistribusian, penjadwalan, dan pengemasan.

*Knapsack* merupakan salah satu permasalahan yang seringkali dijumpai ketika hendak memilih sebuah solusi atas suatu permasalahan kombinatorik. *Knapsack* dapat diartikan sebagai media penyimpanan atau kotak pengemasan. Kotak tersebut hanya dapat memuat atau menyimpan beberapa objek dengan ketentuan total objek harus lebih kecil atau sama dengan kapasitasnya. *Knapsack* adalah salah satu permasalahan optimasi yang mendasar dan dapat diselesaikan dengan berbagai cara. Salah satu cara untuk menyelesaikan permasalahan tersebut adalah dengan cara memilih objek dari sekian banyak dan berapa besar objek tersebut akan disimpan sehingga memperoleh hasil yang optimal dari  $n$  objek dimana setiap objek memiliki keuntungan dan bobot dengan kapasitas terbatas.

Perkembangan teknologi saat ini memungkinkan pencarian solusi optimal untuk permasalahan *knapsack* tersebut dilakukan oleh aplikasi komputer karena keterbatasan manusia. Perhitungan manual juga akan semakin sulit jika objek yang dipilih jumlahnya banyak sehingga pertimbangan waktu perhitungan juga diperhatikan. Oleh karena itu, dibutuhkan sebuah metode maupun program



aplikasi metode untuk menyelesaikan masalah *knapsack*. Permasalahan *knapsack* dibagi menjadi empat, yaitu *integer knapsack*, *fractional knapsack*, *bounded knapsack* dan *unbounded knapsack*. Permasalahan ini bertambah kompleks, ketika tiap-tiap pilihan yang ada masing-masing memiliki lebih dari satu dimensi batasan yang lebih dikenal dengan *Multiple Constraints Knapsack Problem*. Pada bidang bisnis ekonomi, tiap-tiap objek memiliki lebih dari satu dimensi batasan sebagai bahan pertimbangan. Kendala yang berhubungan dengan batasan-batasan tersebut adalah adanya keterbatasan pada sumber daya yang dimiliki. Sehingga perlu dilakukan optimasi agar menghasilkan keuntungan yang maksimal.

Beberapa peneliti sudah banyak yang menyelesaikan masalah *knapsack* tersebut diantaranya yang dilakukan oleh Hadi (2015) membahas tentang penerapan algoritma Genetika *Hybrid* pada permasalahan *Bounded Knapsack* dimana algoritma tersebut perpaduan antara algoritma Genetika dan *Tabu Search*. Kemudian oleh Hilviah (2015) membahas tentang penerapan algoritma *Dynamic Programming* dan algoritma *Backtracking* pada permasalahan *Multiple Constraints Knapsack 0-1* dan masih banyak algoritma yang diterapkan dalam kasus optimasi salah satunya *Artificial Bee Colony*.

Algoritma *Artificial Bee Colony* (ABC) merupakan salah satu algoritma metaheuristik. Meskipun relatif baru, banyak peneliti yang melakukan penelitian dan pengujian algoritma ABC. Menurut Karaboga D. (2005) menyatakan bahwa algoritma ABC sangat sederhana dan fleksibel dibandingkan algoritma-algoritma lain kemudian oleh Baykasoglu (2007) menyebutkan bahwa algoritma-algoritma yang terinspirasi oleh lebah memiliki potensi yang sangat menjanjikan untuk pemodelan dan penyelesaian masalah-masalah optimasi kompleks. Untuk memperkuat penelitian tentang algoritma ABC peneliti Sabet (2012) membahas tentang *A Novel Artificial Bee Colony for the Knapsack Problem* dimana algoritma ABC lebih baik dibandingkan dengan algoritma GA dan PSO. Berdasarkan permasalahan diatas, penulis tertarik untuk melakukan penelitian tentang penerapan algoritma *Artificial Bee Colony* dalam menghasilkan masalah optimasi *Multiple Constraints Bounded Knapsack*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang, permasalahan yang akan dibahas pada tugas akhir ini adalah

- a. bagaimana penerapan algoritma *Artificial Bee Colony* dan solusi pada permasalahan *Multiple Constraints Bounded Knapsack*;
- b. apa pengaruh parameter (jumlah populasi dan jumlah iterasi) dalam algoritma *Artificial Bee Colony* dalam pencarian nilai keuntungan maksimal.

## 1.3 Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah tingkat penjualan atau permintaan konsumen diasumsikan sama.

## 1.4 Tujuan

Dari permasalahan yang ada, maka tujuan penulisan tugas akhir ini adalah

- a. menemukan solusi pada permasalahan *Multiple Constraints Bounded Knapsack* dengan menggunakan algoritma *Artificial Bee Colony*;
- b. mengetahui pengaruh parameter (jumlah populasi dan jumlah iterasi) dalam algoritma *Artificial Bee Colony* dalam pencarian nilai keuntungan maksimal.

## 1.5 Manfaat

Manfaat yang diharapkan dalam tugas akhir ini adalah sebagai berikut:

- a. mendapatkan solusi pada permasalahan *Multiple Constraints Bounded Knapsack* dengan menggunakan algoritma *Artificial Bee Colony*;
- b. sebagai rujukan bagi peneliti yang ingin mempelajari lebih jauh tentang algoritma *Artificial Bee Colony* dan *Multiple Constraints Bounded Knapsack*.

## BAB 2. TINJAUAN PUSTAKA

### 2.1 Optimasi

Optimasi ialah salah satu disiplin ilmu matematika yang fokus untuk mendapatkan nilai minimum atau maksimum secara sistematis dari suatu fungsi, peluang, maupun pencarian nilai lainnya dalam berbagai kasus. Untuk dapat mencapai nilai yang optimal baik maksimum atau minimum, secara sistematis dilakukan pemilihan nilai variabel *integer* atau nyata yang akan memberikan solusi optimal.

Kasus yang berkaitan dengan optimasi sangat banyak dalam kehidupan sehari-hari. Berikut ini adalah kasus-kasus yang berkaitan dengan optimasi adalah:

- a. penentuan pemilihan barang pada masalah *knapsack*;
- b. menentukan lintasan terpendek dari suatu tempat ke tempat yang lain;
- c. menentukan jumlah pekerja seminimal mungkin untuk melakukan suatu proses produksi agar pengeluaran biaya pekerja dapat diminimalkan dan hasil produksi tetap maksimal.

### 2.2 *Knapsack*

*Knapsack* merupakan suatu permasalahan bagaimana memilih objek dari sekian banyak dan berapa besar objek tersebut akan disimpan sehingga diperoleh suatu penyimpanan yang optimal dengan memperhatikan objek yang terdiri dari  $n$  objek  $(1, 2, 3, \dots, n)$  dimana setiap objek memiliki bobot  $(w_i)$  dan nilai *profit* atau keuntungan  $(p_i)$  dengan memperhatikan juga kapasitas dari media penyimpanan sebesar  $W$  dan nilai probabilitas dari setiap objek  $(X_i)$ . Masalah *knapsack* merupakan sebuah persoalan yang menarik. Dalam dunia nyata permasalahan *knapsack* ini sering sekali digunakan pada bidang jasa pengangkutan barang seperti pengangkutan peti kemas dalam sebuah media pengangkut. Dalam usaha tersebut, diinginkan keuntungan yang maksimal untuk mengangkut barang yang

ada dengan tidak melebihi kapasitas yang ada. Berdasarkan persoalan tersebut, diharapkan ada suatu solusi yang secara otomatis dapat mengatasi persoalan itu. *Knapsack* adalah permasalahan mengenai optimalisasi kombinatorial dimana kita harus mencari solusi terbaik dari banyak kemungkinan yang dihasilkan (Dimiyati, 2004). Masalah *knapsack* dapat digambarkan sebuah wadah dengan kapasitas ( $W$ ) akan diisi dengan benda sebanyak ( $n$ ), masing-masing benda memiliki keuntungan berupa *value* ( $v$ ) dan berat berupa *weight* ( $w$ ) (Kellerer, 2004). Sehingga masalah *knapsack* dapat digambarkan sebuah proses memilih barang dengan berat dan keuntungan yang maksimal dari sejumlah  $n$  objek yang memiliki kapasitas terbatas agar mendapatkan hasil yang optimal.

Terdapat beberapa variasi dari masalah *knapsack*, yaitu :

a. *Knapsack 0-1 (integer knapsack)*

Setiap barang hanya terdiri dari 1 unit dan objek yang dimasukkan ke dalam media penyimpanan dengan ketentuan diambil atau ditinggalkan.

b. *Knapsack Fractional*

Barang boleh dibawa sebagian saja (unit dalam pecahan). Versi masalah ini menjadi masuk akal apabila barang yang tersedia dapat dibagi-bagi misalnya gula, tepung, dan sebagainya.

c. *Knapsack terbatas (bounded knapsack)*

Setiap barang tersedia sebanyak  $n$  unit (jumlahnya terbatas) dan objek yang dimasukkan ke dalam media penyimpanan dengan ketentuan bisa dimasukkan sebagian atau seluruhnya.

d. *Knapsack tak terbatas (unbounded knapsack)*

Setiap barang tersedia lebih dari 1 unit dan jumlah objek yang dimasukkan ke dalam media penyimpanan macamnya tidak terbatas.

Ada banyak variasi masalah *knapsack* yang muncul dengan jumlah besar dari masalah dasar seperti jumlah kendala, jumlah tujuan atau bahkan medianya. Adapun variasi permasalahan *knapsack* diatas antara lain (Kellerer, 2004);

a. *Multi Objective Knapsack Problem*

Permasalahan yang memiliki fungsi tujuan lebih dari 1 untuk memaksimalkan keuntungannya;

b. *Multidimensional* atau *Multiple Constrains Knapsack Problem*

Permasalahan yang memiliki kendala lebih dari 1 untuk memaksimalkan keuntungannya;

c. *Multi Knapsack Problem*

Permasalahan yang memiliki fungsi objek/media penyimpanan lebih dari 1 dimana semua item harus dikemas untuk memaksimalkan keuntungannya;

d. *Quadratic Knapsack Problem*

Permasalahan yang tujuannya memaksimalkan fungsi objektif dalam bentuk kuadratik untuk kendala kapasitas biner dan linear.

*Knapsack* yang akan dibahas pada skripsi ini adalah *multiple constraints bounded knapsack* dimana tiap item memiliki kendala dengan jumlah unit barangnya terbatas dan jika dimasukkan ke dalam media penyimpanan barang tersebut bisa dimasukkan sebagian atau seluruhnya.

### 2.2.1 *Bounded Knapsack*

*Bounded knapsack* merupakan generalisasi dari *knapsack* 0-1 dimana jumlah dari tiap jenis barang yang tersedia dibatasi. Diberikan  $n$  buah objek yang akan dimasukkan ke dalam media penyimpanan yang memiliki kapasitas daya tampung senilai  $M$ . Tiap objek yang dinotasikan  $j$  memiliki keuntungan  $p_j$  dan bobot/berat  $w_j$  dan sebuah batas atas ketersediaan objek  $m_j$ . Permasalahannya adalah memilih dari sejumlah objek barang  $x_j$  ( $0 \leq x_j \leq m_j$ ) dari tiap jenis objek dengan jumlah keuntungan yang dimaksimalkan tanpa jumlah bobot/berat melebihi daya tampung kapasitas senilai  $M$ .

Secara matematis, permasalahan *bounded knapsack* dapat didefinisikan sebagai berikut:

Fungsi tujuan maksimum

$$z = \sum_{j=1}^n p_j x_j$$

kendala

$$\sum_{j=1}^n w_j x_j \leq M,$$

dimana  $x_j \in \{0, 1, \dots, m_j\}$ ,  $j = 1, \dots, n$

dengan,

$z$  = nilai optimum fungsi tujuan

$p_j$  = keuntungan barang/objek ke- $j$

$x_j$  = variabel keputusan dimana ( $0 \leq x_j \leq m_j$ )

$n$  = banyak barang/objek

$w_j$  = berat barang/objek ke- $j$

$M$  = kapasitas daya tampung/media penyimpanan

fungsi tujuan semua koefisien adalah bilangan bulat positif. Tanpa menghilangkan sifat umumnya, dapat diasumsikan bahwa  $m_j w_j \leq M$  untuk  $j = 1, \dots, n$  sehingga semua jenis barang dapat dimasukkan ke dalam media penyimpanan (Pisinger, 1995).

### 2.2.2 Multiple Constraints Bounded Knapsack (MCBK)

*Multiple constraints knapsack problem* (MCKP) adalah suatu permasalahan *knapsack* yang sering disebut juga dengan nama *multidimensional knapsack problem* (MKP) dimana MCKP ini sendiri merupakan permasalahan optimasi kombinatorial NP-hard yang terdapat pada beragam aplikasi (Puchinger, 2007). Pada MCKP, tiap-tiap objek/barang memiliki batasan lebih dari satu dimensi. Batasan-batasan bisa berupa waktu, biaya, dan pekerja. Tujuan dari masalah ini yaitu memperoleh solusi optimal dengan mengambil kombinasi objek/barang sedemikian hingga semua batasan juga tidak melewati kapasitas yang tersedia (Lamine, 2012). Sedangkan pada MCBK, tiap-tiap objek/barang memiliki batasan

lebih dari satu dimensi dengan objek/barang dapat diambil sebagian atau seluruhnya.

Secara matematik, *multiple constraints bounded knapsack* dapat dirumuskan sebagai formulasi MCBK berikut ini:

Fungsi tujuan maksimum

$$z = \sum_{j=1}^n p_j x_j$$

kendala

$$\sum_{j=1}^n w_{ij} x_j \leq c_i, \quad i = 1, \dots, m,$$

$$x_j \in \{1, \dots, m_j\}, \quad j = 1, \dots, n.$$

dengan,

$z$  = nilai optimum fungsi tujuan

$i$  = sumber

$j$  = barang/objek

$p_j$  = keuntungan barang/objek ke- $j$  dimana ( $j > 0$ )

$x_j$  = variabel keputusan dimana ( $0 \leq x_j \leq m_j$ )

$n$  = banyak barang/objek

$m$  = banyak sumber

$w_{ij}$  = nilai tiap sumber ke- $i$  dimana ( $w_{ij} > 0$ )

$c_i$  = kapasitas untuk setiap sumber ( $c_i > 0$ )

formulasi rumus matematik *multiple constraints bounded knapsack* di atas menggambarkan perolehan nilai optimal ( $z$ ) dari banyaknya barang/objek  $n$  dengan keuntungan  $p_j$  dan banyaknya sumber  $m$  dengan kapasitas  $c_i$ . Variabel keputusan  $x_j$  mengindikasikan *item-item* yang dipilih.

### 2.3 Algoritma

Algoritma merupakan urutan langkah-langkah untuk menyelesaikan masalah yang disusun secara sistematis. Algoritma dibuat dengan tanpa memperhatikan bentuk yang akan digunakan sebagai implementasinya, sehingga suatu algoritma dapat menjelaskan “bagaimana” cara menggunakan fungsi yang dapat diekspresikan dengan suatu program (Tony, 2007). Dalam beberapa konteks, algoritma adalah urutan langkah untuk melakukan pekerjaan tertentu. Pertimbangan dalam pemilihan algoritma adalah algoritma haruslah benar artinya algoritma akan memberikan keluaran yang dikehendaki dari sejumlah masukan yang diberikan. Tidak peduli sebegitu apapun algoritma kalau memberikan keluaran yang salah, pastilah algoritma tersebut bukanlah algoritma yang baik. Algoritma dikatakan efisien jika waktu tempuh (*running time*) dan berapa ruang dalam memori yang dibutuhkan untuk menjalankan algoritma sangat cepat. Meskipun suatu algoritma memberikan hasil yang mendekati optimal tetapi waktu yang dibutuhkan sangat lama maka algoritma tersebut biasanya jarang dipakai (Munir, 2005).

### 2.4 Algoritma *Artificial Bee Colony*

Algoritma ini pertama kali dikenalkan oleh Karaboga pada tahun 2005. Algoritma *artificial bee colony* menirukan tingkah laku kawanan lebah dalam mencari dan mengeksploitasi sumber-sumber makanan secara efisien. Ketika beberapa ekor lebah menemukan sumber-sumber makanan, mereka akan mengundang lebah-lebah lainnya melalui tarian yang dilakukan di lokasi tertentu. Informasi mengenai sumber makanan tersebut terdiri dari tiga hal yaitu arah, jarak dari sarang, kualitasnya (jumlah nektar). Semakin bagus kualitas sumber makanan, semakin lama durasi tarian yang dilakukan sehingga semakin banyak lebah yang mengikuti ajakannya. Jumlah lebah yang pergi menuju beberapa sumber makanan berbeda-beda secara proporsional sesuai dengan kualitas sumber makanan tersebut. Semakin bagus kualitas sumber makanan, semakin banyak pula lebah yang menuju ke sana. Lebah-lebah tidak kesulitan untuk menentukan lokasi sumber makanan karena petunjuknya sudah disampaikan melalui tarian (yang



memberikan informasi arah dan jarak). Ketika sumber makanan sudah habis, maka lebah akan memberitahu lebah lainnya. Dengan demikian, sumber-sumber makanan bisa dihabiskan dalam waktu secepat mungkin dan dengan mengerahkan lebah pencari makan dalam jumlah seminimum mungkin (Suyanto, 2010).

Koloni dari lebah terdiri dari tiga kelompok yaitu lebah pekerja (*employed bees*), lebah penjaga (*onlooker bees*), dan lebah pengintai (*scout bees*). Tugas lebah pekerja yaitu membagikan informasi sumber makanan dan informasi posisi (jarak dan arah) dari sarang kepada lebah penjaga melalui sebuah tarian. Tugas lebah penjaga adalah mengevaluasi dan memutuskan informasi sumber makanan yang diberikan oleh lebah pekerja dengan probabilitas terkait banyaknya nektar. Tugas lebah pengintai adalah mencari sumber-sumber makanan baru di sekitar sarang secara acak. Pertama, setengah bagian dari koloni terdiri dari lebah pekerja dan setengah lagi dari lebah penjaga. Untuk setiap sumber makanan, hanya ada satu lebah penjaga sehingga jumlah lebah pekerja sama dengan jumlah sumber makanan di sekitar sarang. Seekor lebah penjaga yang sudah menghabiskan makanan pada suatu sumber makanan akan menjadi seekor lebah pengintai. Posisi sumber makanan merupakan solusi untuk masalah optimasi dan jumlah nektar dari sumber makanan menyatakan kualitas (*fitness*) dari solusi terkait (Karaboga, 2005).

Performa algoritma *artificial bee colony* dipengaruhi oleh penentuan parameter yang akan digunakan yaitu jumlah solusi dan jumlah iterasi. Karena kedua parameter tersebut memberikan efek yang signifikan terhadap performa algoritma *artificial bee colony*. Semakin besar jumlah solusi dan jumlah iterasinya maka dapat menghasilkan kinerja yang lebih baik karena memiliki ruang pencarian yang semakin besar (Sugioko, 2013).

Langkah-langkah pada algoritma *artificial bee colony* sebagai berikut:

a. Inisialisasi Solusi Awal

Penentuan parameter dilakukan terlebih dahulu sebelum memulai suatu generasi (iterasi). Parameter-parameter tersebut yaitu ukuran jumlah populasi lebah, jumlah lebah pengintai, dan panjang *list* solusi yang akan digunakan

serta kriteria berhenti yaitu jumlah iterasi yang dipakai. Inisialisasi solusi awal menggunakan solusi yang diperoleh secara acak dengan persamaan (2.1),

$$x_{ij} = x_{j \min} + rand(0,1) \cdot (x_{j \max} - x_{j \min}) \quad (2.1)$$

dengan,

- $x_{ij}$  = inisialisasi kemungkinan solusi ke- $i$  dengan parameter ke- $j$ ,
- $x_{j \min}$  = nilai kemungkinan solusi terkecil berdasarkan parameter  $j$ ,
- $x_{j \max}$  = nilai kemungkinan solusi terbesar berdasarkan parameter  $j$ ,
- $rand(0,1)$  = nilai random antara 0 sampai 1,
- $i$  = 1, 2, ...,  $SN$  dengan  $SN$  adalah jumlah kemungkinan solusi (sumber makanan),
- $j$  = 1, 2, ...,  $D$  dengan  $D$  adalah jumlah parameter yang digunakan.

#### b. Menentukan Solusi Alternatif

Tahap penentuan solusi alternatif sering dikenal dengan istilah tahap lebah pekerja (*employed bee phase*). Pada tahap ini dilakukan berdasarkan hasil dari inisialisasi solusi awal yang akan dijadikan acuan sejumlah  $n$  lebah untuk melakukan pencarian sumber nektar, sehingga didapatkan sejumlah  $n$  solusi alternatif dengan menggunakan persamaan (2.2),

$$v_{ij} = x_{ij} + \emptyset_{ij}(x_{kj} - x_{ij}) \quad (2.2)$$

dengan,

- $v_{ij}$  = nilai perluasan kemungkinan solusi ke- $i$  dengan parameter  $j$ ,
- $x_{ij}$  = nilai kemungkinan solusi ke- $i$  dengan parameter  $j$ ,
- $x_{kj}$  = nilai kemungkinan solusi ke- $i$  dengan parameter  $j$ ,
- $k$  = 1, 2, ...,  $SN$ ,
- $\emptyset_{ij}$  = bilangan random antara -1 sampai 1.

Setelah setiap kemungkinan solusi diperluas, akan diaplikasikan *greedy selection* antara nilai kemungkinan solusi  $x_{ij}$  dengan nilai hasil perluasan yaitu  $v_{ij}$ . Jika nilai  $v_{ij}$  lebih besar dari nilai  $x_{ij}$  maka nilai  $v_{ij}$  tersebut akan

dianggap sama dengan nilai  $x_{ij}$  dan nilai percobaan tetap bernilai 0. Jika tidak, nilai  $x_{ij}$  yang disimpan dan nilai percobaan ke- $i$  ditambah dengan 1.

c. Evaluasi Populasi Awal

Setelah setiap kemungkinan solusi diperluas dan dibandingkan dengan nilai awal inialisasi. Tahap selanjutnya adalah menghitung kualitas dari setiap kemungkinan solusi menggunakan fungsi *fitness* sebagai berikut,

$$fitness(x_i) = \begin{cases} \frac{1}{(1 + f(x_i))}, & f(x_i) \geq 0 \\ 1 + |f(x_i)|, & f(x_i) < 0 \end{cases}$$

d. Evaluasi Populasi Alternatif

Evaluasi populasi alternatif atau yang sering disebut dengan istilah tahap lebah penjaga (*onlooker bee phase*) merupakan suatu tahap untuk menghitung nilai *probability* pada setiap kemungkinan solusi dengan menggunakan persamaan (2.3),

$$p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \quad (2.3)$$

dengan,

$p_i$  = nilai *probability*,

$fitness_i$  = nilai *fitness* solusi ke- $i$ ,

$\sum_{i=1}^{SN} fitness_i$  = jumlah nilai *fitness* ke- $i$  sampai  $SN$ .

e. Evaluasi Populasi Akhir

Tahapan evaluasi populasi akhir yang sering disebut dengan istilah pengintaian (*scout bee phase*) merupakan tahapan evaluasi solusi secara keseluruhan. Pada tahap ini, akan diaplikasikan sebuah metode *roollete-whele* yaitu memilih bilangan real secara random antara [0, 1] untuk setiap kemungkinan solusi. Jika nilai  $p_i$  lebih kecil dari bilangan *random* yang ditentukan, maka kemungkinan solusi yang terpilih tersebut merupakan solusi yang sesuai lebah pekerja sebelumnya.

f. Kriteria berhenti

Kriteria pemberhentian yang digunakan untuk kasus itu adalah jumlah iterasi yang telah ditentukan pada awal perhitungan. Jika jumlah iterasi belum terpenuhi maka kembali pada langkah b.



### BAB 3. METODE PENELITIAN

#### 3.1 Data Penelitian

Data yang digunakan dalam penelitian ini adalah data sekunder yang diambil dari skripsi Hadi (2015) yaitu UD. Buah Barokah. UD. Buah Barokah adalah usaha dagang yang bergerak di bidang perdagangan buah-buah pertanian. Data yang diambil adalah data yang berupa harga beli barang, harga jual barang, jumlah kemasan, berat setiap kemasan serta keuntungan.

Tabel 3.1 Data Penelitian

No	Nama Barang	Jumlah Kemasan	Berat (Kg)	Keuntungan (Rp)
1	Anggur	10	0,5	2.300
2	Alpukat	7	5	20.000
3	Apel	12	15	47.000
4	Belimbing	5	10	12.000
5	Buah Naga	11	20	52.000
6	Duku	8	1	2.200
7	Durian	0	1	3.000
8	Jambu Merah	4	10	14.000
9	Jambu Kristal	8	10	18.500
10	Jeruk	9	20	75.000
11	Kesemek	2	3	7.500
12	Mangga	13	15	25.500
13	Manggis	3	5	9.700
14	Melon	5	10	22.000
15	Nanas	7	12	23.500
16	Nangka	14	1	1.850
17	Pepaya	6	20	41.000

No	Nama Barang	Jumlah Kemasan	Berat (Kg)	Keuntungan (Rp)
18	Salak	12	20	112.000
19	Sawo	3	3	6.500
20	Semangka	2	4,5	8.500
21	Strawberry	16	0,5	1.800
22	Anggur Merah	11	0,5	2.500
23	Langsep	4	1	1.250
24	Klengkeng	6	2,5	5000
25	Rambutan	12	5	5.750

### 3.2 Langkah-langkah Penelitian

Langkah–langkah yang akan dilakukan dalam penelitian tentang “Penerapan Algoritma *Artificial Bee Colony* pada permasalahan *Multiple Constraints Bounded Knapsack*”, secara skematik dapat dilihat pada Gambar 3.1. Penjelasan skema pada Gambar 3.1 untuk memperoleh hasil yang diinginkan sebagai berikut:

a. Studi Literatur .

Langkah awal yang dilakukan adalah mencari studi literatur. Studi literatur dilakukan untuk mendapatkan informasi dari buku, jurnal dan skripsi yang terkait tentang materi penerapan algoritma *artificial bee colony* pada permasalahan *multiple constraints bounded knapsack*.

b. Mengambil dan Mengumpulkan Data

Langkah kedua yaitu mengambil dan mengumpulkan data yang diambil dari skripsi Hadi (2015). Tujuannya sebagai objek penelitian untuk menyelesaikan suatu masalah menggunakan algoritma *artificial bee colony*.

c. Mengidentifikasi Data

Langkah ketiga yang dilakukan yaitu menentukan atau menetapkan atribut, keuntungan, bobot, dan lain-lain.

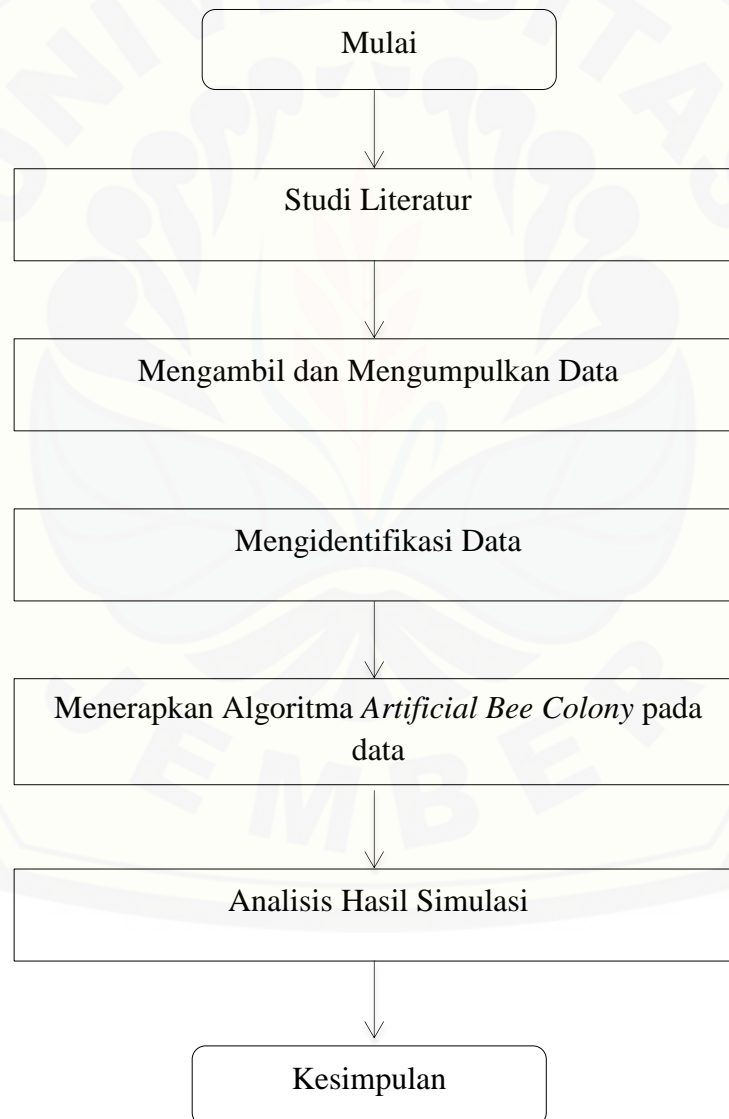
d. Menerapkan Algoritma *Artificial Bee Colony* pada Pemasalahan *Multiple Constraints Bounded Knapsack*

Langkah berikutnya yaitu menerapkan algoritma *artificial bee colony* pada permasalahan *multiple constraints bounded knapsack* dengan cara inisialisasi solusi awal, menentukan solusi alternatif, evaluasi populasi awal, evaluasi populasi alternatif, evaluasi populasi akhir, dan mengecek kriteria pemberhentian.

e. Analisis Hasil Simulasi

Program yang digunakan dalam penelitian ini adalah Matlab R2009a.

f. Kesimpulan



Gambar 3.1 Skema Metode Penelitian

## BAB 5. PENUTUP

### 5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut:

- a. Permasalahan *knapsack* yang telah diterapkan pada algoritma *artificial bee colony* menghasilkan keuntungan yang maksimal sebesar Rp. 3.335.400,- dengan jumlah populasi dan jumlah iterasi 1000 dan 2000.
- b. Semakin besar jumlah iterasi dan jumlah populasi maka semakin mudah mendapatkan solusi optimal yang diinginkan. Oleh karena itu, parameter jumlah populasi dan jumlah iterasi yang sangat berpengaruh terhadap solusi optimal.

### 5.2 Saran

Dalam penelitian ini disarankan untuk peneliti selanjutnya dapat mengaplikasikan algoritma *artificial bee colony* pada permasalahan *multiple constraint unbounded knapsack*. Selain itu, dapat juga dibandingkan dengan algoritma lain seperti algoritma *firefly*, algoritma *cat swarm optimization*.



**DAFTAR PUSTAKA**

- Adil Baykosoglu, Lale Ozbakor, dan Pinar Tapkan. 2007. Artificial Bee Colony Algorithm and Its Application to Generation Assignment Problem. *Swarm Intelligence: Focus on Ant Particle Swarm Optimization*. ISBN 978-3-902613-09-7, pp. 532. Austria: Itech Education and Publishing. <http://natcomp.liacs.nl/SWI/papers/artificial.bee.colony.algorithm/abc.algorithm.and.its.application.to.generalized.assignment.problem.pdf>. [10 Maret 2016].
- Dimiyati, T. dan Ahmad, D. 2004. *Operations Research: Model-model Pengambilan Keputusan*. Bandung: Sinar Baru Agesindo.
- Hadi, I. S. 2015. *Penerapan Algoritma Genetika Hybrid pada Permasalahan Bounded Knapsack*. Tidak diterbitkan. Skripsi. Jember: Universitas Jember
- Hilviah, F. 2015. *Penerapan Algoritma Dynamic Programming dan Algoritma Backtracking pada Permasalahan Multiple Constraints Knapsack 0-1*. Tidak diterbitkan. Skripsi. Jember: Universitas Jember
- Karaboga D. 2005. *An Idea Based on Honey Bee Swarm for Numerical Optimization*. Turkey: Erciyes University, Engineering Faculty, Computer Engineering Departement. [http://mf.erciyes.edu.tr/abc/pub/tr06\\_2005.pdf](http://mf.erciyes.edu.tr/abc/pub/tr06_2005.pdf). [10 Maret 2016]
- Kellerer, H., Pferschy, U., dan Pisinger, D. 2004. *Knapsack Problems*. New York: Springer-Verlag Berlin Heidelberg.
- Lamine, A., Khemakhem, M., dan Chabchoud, H.,. 2012. "Knapsack Problem Involving Dimensions, Demands and Multiple Choice Constraints". *International Journal of Advanced Science and Technology*, Vol 46, 4. Tunisia: University Sfax.
- Munir, R. 2005. *Strategi Algoritmik*. Yogyakarta: Graha Ilmu.
- Pisinger, D. 1995. *Algorithm for Knapsack Problem*. Denmark: University Compenhagen.
- Sa'diyah, N. 2015. *Algoritma Artificial Bee Colony dan Algoritma Differential Evolution Plus untuk Penjadwalan FlowShop*. Tidak diterbitkan. Skripsi. Jember: Universitas Jember.

Sugioko, A. 2013. "Perbandingan Algoritma Bee Colony dengan Algoritma Bee Colony Tabu list dalam Penjadwalan FlowShop". *Jurnal Metris*. **14**: 113-120.

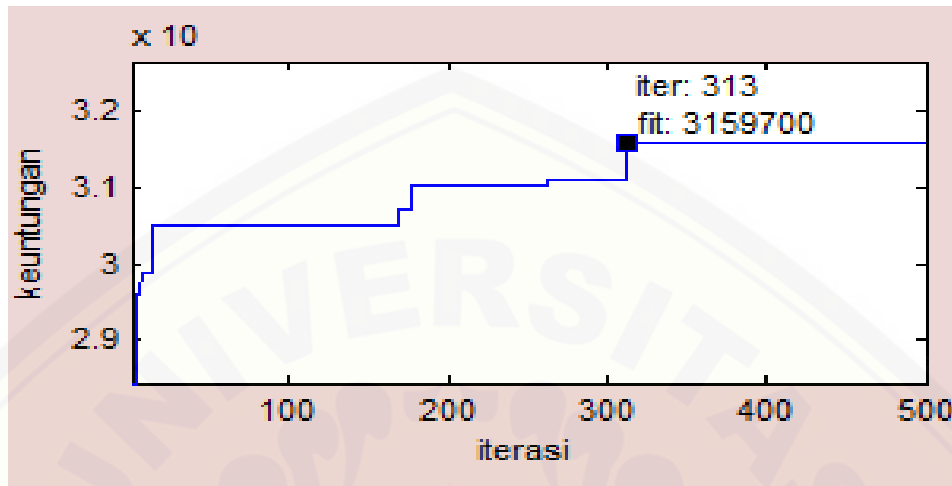
Suyanto, 2010. *Algoritma Optimasi*. Yogyakarta: Graha Ilmu.

Tony, H. 2007. *Algoritma dan Pemrograman*. Seri : Modul Diskusi Fakultas Ilmu Komputer. Surabaya: Universitas Narotama.  
[rincikembang.dosen.narotama.ac.id/files/2011/04/Algo2007.pdf](http://rincikembang.dosen.narotama.ac.id/files/2011/04/Algo2007.pdf). [10 Maret 2016]



LAMPIRAN

LAMPIRAN A. Grafik Kekonvergenan



LAMPIRAN B. Jumlah Kemasan Pada Hasil *Running* Pengujian Parameter Populasi dan Iterasi

- Percobaan pertama

No	Nama Barang	Jumlah Kemasan
1	Anggur	5
2	Alpukat	4
3	Apel	10
4	Belimbing	2
5	Buah Naga	6
6	Duku	3
7	Durian	0
8	Jambu Merah	2
9	Jambu Kristal	5
10	Jeruk	6
11	Kesemek	2
12	Mangga	5
13	Manggis	2
14	Melon	3

No	Nama Barang	Jumlah Kemasan
15	Nanas	1
16	Nangka	7
17	Papaya	3
18	Salak	11
19	Sawo	0
20	Semangka	2
21	Strawberry	8
22	Anggur Merah	2
23	Langsep	2
24	Klengkeng	3
25	Rambutan	10

- Percobaan kedua

No	Nama Barang	Jumlah Kemasan
1	Anggur	7
2	Alpukat	4
3	Apel	6
4	Belimbing	1
5	Buah Naga	9
6	Duku	6
7	Durian	0
8	Jambu Merah	1
9	Jambu Kristal	4
10	Jeruk	9
11	Kesemek	1
12	Mangga	5
13	Manggis	2
14	Melon	2
15	Nanas	2
16	Nangka	7
17	Papaya	2
18	Salak	11

No	Nama Barang	Jumlah Kemasan
19	Sawo	3
20	Semangka	1
21	Strawberry	10
22	Anggur Merah	4
23	Langsep	1
24	Klengkeng	3
25	Rambutan	6

- Percobaan ketiga

No	Nama Barang	Jumlah Kemasan
1	Anggur	7
2	Alpukat	4
3	Apel	7
4	Belimbing	3
5	Buah Naga	7
6	Duku	5
7	Durian	0
8	Jambu Merah	2
9	Jambu Kristal	3
10	Jeruk	6
11	Kesemek	1
12	Mangga	5
13	Manggis	2
14	Melon	2
15	Nanas	3
16	Nangka	8
17	Papaya	4
18	Salak	12
19	Sawo	0
20	Semangka	0
21	Strawberry	11
22	Anggur Merah	8

No	Nama Barang	Jumlah Kemasan
23	Langsep	3
24	Klengkeng	5
25	Rambutan	5

- Percobaan keempat

No	Nama Barang	Jumlah Kemasan
1	Anggur	4
2	Alpukat	4
3	Apel	6
4	Belimbing	2
5	Buah Naga	6
6	Duku	2
7	Durian	0
8	Jambu Merah	2
9	Jambu Kristal	3
10	Jeruk	7
11	Kesemek	2
12	Mangga	7
13	Manggis	0
14	Melon	5
15	Nanas	4
16	Nangka	6
17	Papaya	2
18	Salak	12
19	Sawo	3
20	Semangka	1
21	Strawberry	8
22	Anggur Merah	4
23	Langsep	1
24	Klengkeng	2
25	Rambutan	7

- Percobaan kelima

No	Nama Barang	Jumlah Kemasan
1	Anggur	7
2	Alpukat	5
3	Apel	7
4	Belimbing	0
5	Buah Naga	9
6	Duku	3
7	Durian	0
8	Jambu Merah	1
9	Jambu Kristal	1
10	Jeruk	6
11	Kesemek	0
12	Mangga	5
13	Manggis	0
14	Melon	1
15	Nanas	7
16	Nangka	8
17	Papaya	3
18	Salak	12
19	Sawo	0
20	Semangka	1
21	Strawberry	7
22	Anggur Merah	8
23	Langsep	2
24	Klengkeng	5
25	Rambutan	6

- Percobaan keenam

No	Nama Barang	Jumlah Kemasan
1	Anggur	6
2	Alpukat	6
3	Apel	5

No	Nama Barang	Jumlah Kemasan
4	Belimbing	3
5	Buah Naga	6
6	Duku	4
7	Durian	0
8	Jambu Merah	2
9	Jambu Kristal	5
10	Jeruk	7
11	Kesemek	1
12	Mangga	7
13	Manggis	0
14	Melon	1
15	Nanas	4
16	Nangka	9
17	Papaya	2
18	Salak	12
19	Sawo	2
20	Semangka	2
21	Strawberry	5
22	Anggur Merah	6
23	Langsep	1
24	Klengkeng	0
25	Rambutan	5

- Percobaan ketujuh

No	Nama Barang	Jumlah Kemasan
1	Anggur	3
2	Alpukat	5
3	Apel	9
4	Belimbing	1
5	Buah Naga	7
6	Duku	5
7	Durian	0



No	Nama Barang	Jumlah Kemasan
8	Jambu Merah	1
9	Jambu Kristal	5
10	Jeruk	7
11	Kesemek	2
12	Mangga	6
13	Manggis	2
14	Melon	3
15	Nanas	1
16	Nangka	8
17	Papaya	4
18	Salak	10
19	Sawo	1
20	Semangka	1
21	Strawberry	8
22	Anggur Merah	6
23	Langsep	2
24	Klengkeng	3
25	Rambutan	3

- Percobaan kedelapan

No	Nama Barang	Jumlah Kemasan
1	Anggur	4
2	Alpukat	3
3	Apel	8
4	Belimbing	2
5	Buah Naga	5
6	Duku	3
7	Durian	0
8	Jambu Merah	0
9	Jambu Kristal	5
10	Jeruk	9
11	Kesemek	1

No	Nama Barang	Jumlah Kemasan
12	Mangga	6
13	Manggis	1
14	Melon	2
15	Nanas	4
16	Nangka	7
17	Papaya	3
18	Salak	11
19	Sawo	1
20	Semangka	1
21	Strawberry	6
22	Anggur Merah	5
23	Langsep	0
24	Klengkeng	1
25	Rambutan	7

- Percobaan kesembilan

No	Nama Barang	Jumlah Kemasan
1	Anggur	7
2	Alpukat	4
3	Apel	6
4	Belimbing	3
5	Buah Naga	8
6	Duku	5
7	Durian	0
8	Jambu Merah	0
9	Jambu Kristal	5
10	Jeruk	8
11	Kesemek	0
12	Mangga	7
13	Manggis	0
14	Melon	1
15	Nanas	4

No	Nama Barang	Jumlah Kemasan
16	Nangka	8
17	Papaya	1
18	Salak	12
19	Sawo	3
20	Semangka	0
21	Strawberry	8
22	Anggur Merah	6
23	Langsep	1
24	Klengkeng	3
25	Rambutan	5

- Percobaan kesepuluh

No	Nama Barang	Jumlah Kemasan
1	Anggur	4
2	Alpukat	3
3	Apel	11
4	Belimbing	2
5	Buah Naga	9
6	Duku	2
7	Durian	0
8	Jambu Merah	0
9	Jambu Kristal	1
10	Jeruk	6
11	Kesemek	0
12	Mangga	6
13	Manggis	1
14	Melon	2
15	Nanas	3
16	Nangka	8
17	Papaya	1
18	Salak	12
19	Sawo	1

No	Nama Barang	Jumlah Kemasan
20	Semangka	1
21	Strawberry	9
22	Anggur Merah	7
23	Langsep	3
24	Klengkeng	3
25	Rambutan	7

### LAMPIRAN C. Skrip Program

```

%proses
if mod(pop,1)~=0 || mod(maxiter,1)~=0 || pop<=0 || maxiter<=0
    error('Populasi atau Max Iterasi harus bilangan bulat positif');
else
    n=length(data1);%menghitung banyak jenis barang
    panjang_pop=sum(data1);
    %pop_awal
    for i=1:pop
        bee(i,:)=randi([0 1],1,panjang_pop);
        bobot1(i)=0;fitness1(i)=0;beli1(i)=0;
        for k=1:n
            bobot1(i)=bobot1(i)+sum(bee(i,sum(data1(1:k-1))+(1:data1(k))))*data2(k);
            beli1(i)=beli1(i)+sum(bee(i,sum(data1(1:k-1))+(1:data1(k))))*data4(k);
            fitness1(i)=fitness1(i)+sum(bee(i,sum(data1(1:k-1))+(1:data1(k))))*data3(k);
        end
    end
    %layak=zeros(maxiter,pop);
    itr=1;
    for iter=1:maxiter
        %pop_alternatif
        for i=1:pop
            bee2(i,:)=randi([0 1],1,panjang_pop);
            bobot2(i)=0;fitness2(i)=0;beli2(i)=0;
            for k=1:n
                bobot2(i)=bobot2(i)+sum(bee2(i,sum(data1(1:k-
1))+(1:data1(k))))*data2(k);
                beli2(i)=beli2(i)+sum(bee2(i,sum(data1(1:k-1))+(1:data1(k))))*data4(k);
                fitness2(i)=fitness2(i)+sum(bee2(i,sum(data1(1:k-
1))+(1:data1(k))))*data3(k);
            end
        end
        for i=1:pop
            if bobot2(i)<=MaxBot && beli2(i)<=Modal

```

```

        if fitness2(i)>fitness1(i)
            bee(i,:)=bee2(i,:);
            bobot1(i)=bobot2(i);
            beli1(i)=beli2(i);
            fitness1(i)=fitness2(i);
            %layak(iter,i)=1;
        end
    end
end
%cek kekonvergenan
fitness_layak=[];aa=1;
for i=1:pop
    if bobot1(i)<=MaxBot && beli1(i)<=Modal
        fitness_layak(aa)=fitness1(i);
        aa=aa+1;
    end
end
if ~isempty(fitness_layak)
    [fit_urut ind]=sort(fitness_layak,'descend');
    konver(itr)=fitness_layak(ind(1));
    numiter(itr)=iter;
    itr=itr+1;
end
end
%length(find(layak==1))
solusi_layak=[];bobot_layak=[];fitness_layak=[];beli_layak=[];aa=1;
for i=1:pop
    if bobot1(i)<=MaxBot && beli1(i)<=Modal
        solusi_layak(aa,:)=bee(i,:);
        bobot_layak(aa)=bobot1(i);
        fitness_layak(aa)=fitness1(i);
        beli_layak(aa)=beli1(i);
        aa=aa+1;
    end
end
if ~isempty(fitness_layak)
    [fit_urut ind]=sort(fitness_layak,'descend');
    solusi_akhir=solusi_layak(ind(1),:);
    fitness_akhir=fitness_layak(ind(1));
    bobot_akhir=bobot_layak(ind(1));
    beli_akhir=beli_layak(ind(1));
    nama_barang=dataa{1};
    jml_barang=[];

```