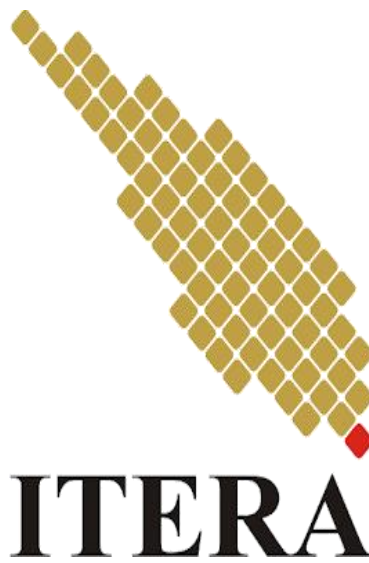


# **Penerapan *Particle Swarm Optimization* (PSO) Pada Knapsack Problem Untuk Optimasi Maksimasi Keuntungan Dalam Sebuah Pengiriman Barang**

*Laporan Akhir Swarm Intelligence*



## **Kelompok 10 (RB)**

Ketua Kelompok	120450006	Arfyani Deiastuti
Anggota 1	120450064	Hanna Septiani
Anggota 2	120450088	Alfianri Mahnihuruk
Anggota 3	120450086	Junpito Salim

## INTRODUCTION

Setiap badan usaha yang ada, baik itu yang berbentuk badan usaha manufaktur ataupun badan usaha jasa, pasti ingin mendapatkan keuntungan. Hal ini memerlukan sistem kerja yang baik dan maksimal. Banyak pihak yang sering kurang mempedulikan setiap aspek yang ada di dalam sistem kerja, salah satunya bagian penyimpanan untuk melakukan pengiriman barang. Barang-barang yang ada seringkali dikemas dengan cara yang asal-asalan. Barang-barang diletakkan begitu saja dalam box tanpa mempedulikan berat dan juga keuntungan yang didapat. Tidak teraturnya barang-barang yang dikemas ini dapat mengakibatkan kerusakan pada barang yang disimpan dan menimbulkan kerugian. Oleh sebab itu dibutuhkannya sebuah optimasi. Optimisasi berguna di hampir segala bidang dalam melakukan usaha untuk mencapai target yang ingin dicapai dengan cara yang efisien. Berdasarkan fungsi tujuannya, optimisasi dibagi menjadi dua jenis, yaitu maksimisasi dan minimisasi. Maksimisasi merupakan optimisasi dengan menggunakan input tertentu untuk mendapatkan keuntungan maksimal, sedangkan minimisasi merupakan optimisasi untuk menghasilkan output tertentu dengan menggunakan input yang paling minimal.

Permasalahan optimisasi tersebut merupakan permasalahan knapsack. Masalah knapsack merupakan suatu permasalahan bagaimana memilih objek dari beberapa objek yang akan dimasukkan ke media penyimpanan dengan setiap objek memiliki bobot dan total bobot dari objek yang dipilih tidak boleh melebihi kapasitas media penyimpanannya, sehingga diperoleh keuntungan yang maksimal. Metode yang dapat digunakan untuk menyelesaikan permasalahan knapsack salah satunya adalah metode meta-heuristic yaitu *Particle Swarm Optimization* (PSO). PSO masuk ke dalam *Swarm Intelligence*.

*Swarm Intelligence* merupakan kecerdasan buatan yang berkaitan dengan kecerdasan dan perilaku agen atau individu secara sederhana dalam menyelesaikan suatu permasalahan dengan cara berkelompok. SI dapat menyelesaikan masalah kontinu maupun diskrit, fleksibel, dan sangat cepat menemukan solusi (mendekati) optimum. Salah satu contoh *Swarm Intelligence* adalah *Particle Swarm Optimization* (PSO). PSO terinspirasi dari kawanan ikan yang berenang dan sekumpulan burung yang terbang mendapat manfaat dari pengalaman anggota lain yang lagi berenang atau terbang bersama. Dengan kata lain bahwa ketika salah satu burung yang terbang secara random dan menemukan sumber makanan maka seluruh kawanan akan terbantu jika salah satu ikan memberitahukan sumber makanan tersebut. Maka seluruh kelompok akan mendapat hasil buruan terbaik jika seluruh anggota saling menginformasikan hasil buruannya.

Dari perumpamaan di atas dapat kita bayangkan jika salah burung akan membantu kita mencari solusi optimal dari suatu ruang dimensi solusi dan solusi yang ditemukan burung tersebut merupakan solusi terbaik dari sejumlah solusi yang ada. Oleh karena itu judul yang diambil untuk melakukan penelitian tentang penerapan *Particle Swarm Optimization* (PSO) Pada Knapsack Problem Untuk Optimasi Maksimasi Keuntungan Dalam Sebuah Penyimpanan Barang.

## PROBLEM STATEMENT

Masalah penyimpanan untuk melakukan pengiriman barang ditemukan pada *brand Brooks Running Shoes* asal Amerika adalah sebuah *brand* olahraga yang memfokuskan diri memproduksi sepatu khusus lari sejak 1914. Ada berbagai macam item yang dijual oleh *brand* ini tentunya dengan jenis dan ukuran yang berbeda-beda. Dengan banyaknya jenis yang diproduksi menimbulkan masalah baru dalam melakukan pengemasan untuk pengiriman barang. Barang yang akan dikirim pastinya akan disimpan ke dalam box-box yang memiliki keterbatasan ruang sehingga tidak memungkinkan menempatkan semua barang. Apabila kapasitas dalam box kurang maksimal akibatnya *brand* akan mengalami kerugian dan apabila kapasitas melewati batas maksimal akibatnya bisa merusak barang karena box tidak dapat menampung dengan baik.

Oleh sebab itu bagaimana petugas harus memasukkan barang semaksimal mungkin ke dalam wadah atau tempat barang tersebut ketika menerima permintaan pengiriman ke konsumen. Jenis barang yang dimasukkan kedalam wadah memiliki berat, harga serta tingkat kepentingan dari barang lain. Petugas akan memilih barang yang sesuai dengan tempat atau wadah dengan pertimbangan berat barang tidak melebihi kapasitas maksimum sehingga dapat mengoptimalkan tempat yang digunakan. Melalui proses analisis ini diharapkan pengiriman jumlah barang dapat dioptimalkan sehingga mendapatkan keuntungan yang sebesar-besarnya.

Perumusan masalah pada penelitian ini yaitu barang yang memiliki profit yang tinggi dengan berat yang lebih ringan itu akan lebih diutamakan untuk dikemas dan dikirim, tidak lupa harus memenuhi kapasitas yang ada. Tujuan dari penelitian ini adalah membantu *brand Brooks Running Shoes* dalam mendapatkan data-data barang yang dapat dikemas dengan profit dan weight yang paling optimal untuk mendapatkan keuntungan yang tinggi dengan Metode *Particle Swarm Optimization* (PSO) pada Knapsack Problem. Batasan masalah yang digunakan adalah analisis ini menggunakan dua variabel dari data yang ada yaitu price sebagai profit dan weight sebagai weight, tanpa adanya penambahan atau pengurangan dari data tersebut. Analisis ini juga menggunakan asumsi bahwa wadah yang digunakan untuk mengemas barang memiliki kapasitas yang sama.

## DATA DESCRIPTION

Pada tugas RBL ini, penulis mengambil data dari <http://brooksrunning.com/> tahun 2020. Data dikumpulkan secara manual menangkap bagian spesifikasi dari setiap halaman sepatu yang berlaku dalam situs *web Books Running*. Data yang digunakan yaitu data barang penjualan sepatu pada *brand Brooks Running Shoes* yang mencakup 26 model dengan rincian sebagai berikut :

- **Name** : Nama sepatu
- **Type** : Tipe (man's/Women's/unisex)
- **Price** : Harga Sepatu (Tahun 2020)

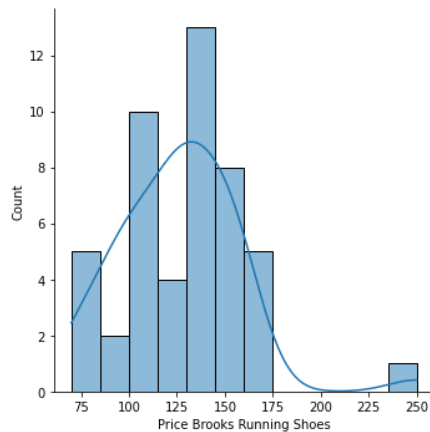
- **Midsole Drop** : Penurunan midsole dalam mm
- **Weight** : Berat dalam gr

Name	Type	Price	Midsole Drop(mm)	Weight
<b>Addiction 14</b>	Men's	130	12	357.2
<b>Adrenaline GTS 20</b>	Men's	130	12	300.5
<b>Anthem 3</b>	Men's	69.95	10	212.6
<b>Asteria</b>	Men's	110	8	235.3
<b>Beast 20</b>	Men's	160	12	331.7
<b>Bedlam 3</b>	Men's	150	8	306.2
<b>Caldera 4</b>	Men's	140	4	283.5
<b>Cascadia 15</b>	Men's	130	8	311.8
<b>Cascadia 15 GTX</b>	Men's	160	8	331.7
<b>Divide</b>	Men's	100	8	292
<b>Dyad 11</b>	Men's	130	10	328.9
<b>Ghost 13</b>	Men's	130	12	286.3
<b>Glycerin 18</b>	Men's	150	10	289.2
<b>Hyperion</b>	Men's	130	10	181.4
<b>Hyperion Tempo</b>	Men's	150	8	207
<b>Hyperion Elite</b>	Unisex	250	8	195.6
<b>Launch 7</b>	Men's	100	10	255.1
<b>Levitate 4</b>	Men's	150	8	292
<b>PureFlow 7</b>	Men's	90	4	252.3
<b>PureGrit 8</b>	Men's	78	4	263.7
<b>Ravenna 11</b>	Men's	110	10	266.5
<b>Revel 4</b>	Men's	100	8	252.3
<b>Ricochet 2</b>	Men's	120	8	275
<b>Ricochet 2 LE</b>	Men's	120	8	275
<b>Transcend 7</b>	Men's	160	10	303.3
<b>Ghost 13</b>	Women's	130	12	249.5
<b>Adrenaline GTS 20</b>	Women's	130	12	266.5
<b>Revel 4</b>	Women's	100	8	224
<b>Glycerin 18</b>	Women's	150	10	255.1
<b>Levitate 4</b>	Women's	150	8	260.8
<b>Bedlam 3</b>	Women's	150	8	269.3
<b>Addiction 14</b>	Women's	130	12	323.2
<b>PureFlow 7</b>	Women's	90	4	212.6
<b>Ravenna 11</b>	Women's	110	10	238.1
<b>Transcend 7</b>	Women's	160	10	269.3
<b>Ricochet 2</b>	Women's	120	8	238.1

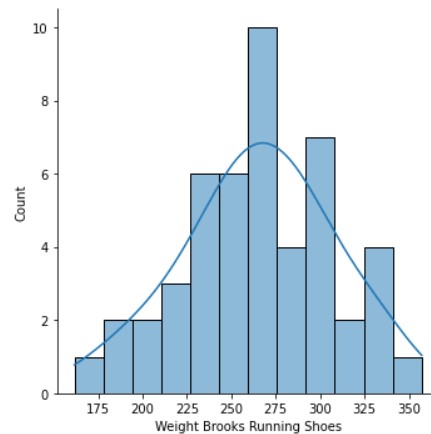
<b>Cascadia 15</b>	Women's	130	8	283.5
<b>Divide</b>	Women's	100	8	260.8
<b>Hyperion Tempo</b>	Women's	150	8	189.9
<b>Caldera 4</b>	Women's	140	4	252.3
<b>Ricochet 2 LE</b>	Women's	120	8	275
<b>Cascadia 15 GTX</b>	Women's	160	8	294.8
<b>Anthem 3</b>	Women's	69.95	10	235.3
<b>Hyperion</b>	Women's	130	10	161.6
<b>Launch 7</b>	Women's	100	10	226.8
<b>Dyad 10</b>	Women's	84.5	10	297.7
<b>Ariel '18</b>	Women's	104	12	334.5
<b>PureGrit 8</b>	Women's	78	4	232.5

**Tabel 1. Data Sepatu Lari Brooks Tahun 2020**

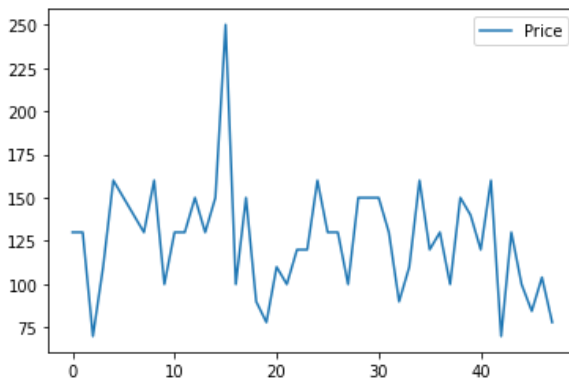
## Visualisasi Data



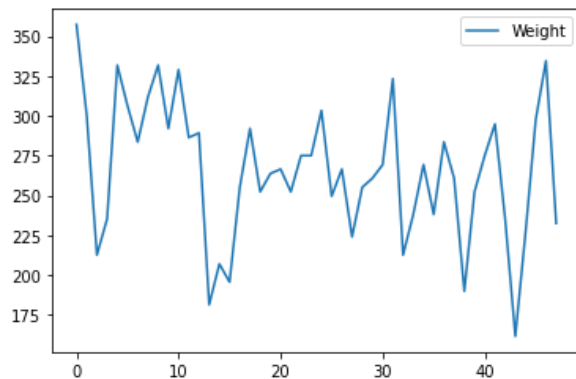
**Gambar 1. Grafik Price Running Shoes**



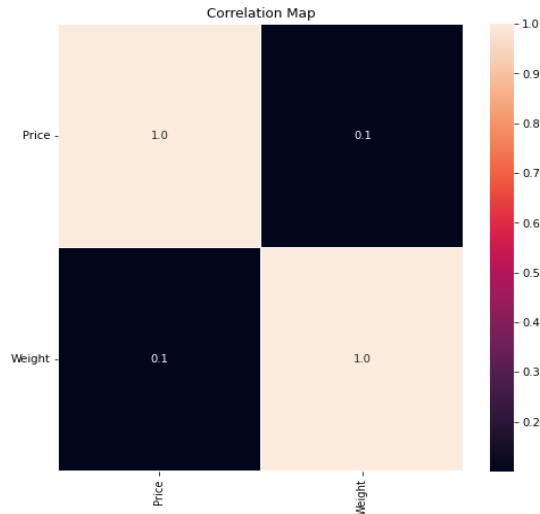
**Gambar 2. Grafik Weight Running Shoes**



**Gambar 3. Plot Data Price**



**Gambar 4. Plot Data Weight**



**Gambar 5. Correlation Map dari Price dan Weight**

### Data Processing

Dari lima variabel yang ada pada data yang digunakan hanya dua variabel yang akan digunakan sebagai parameter yaitu price dan weight dengan jumlah data adalah 48. Dengan parameter profit dan weight yang berbeda, dimana barang tersebut akan ditentukan keuntungan mana yang paling optimal dan akan dilakukan penyimpanan dalam *knapsack* untuk dilakukan pengiriman. Pada kasus ini, diketahui 48 item yang akan dicari keuntungan yang optimal berdasarkan berat dan harga. Berat masing-masing barang dimulai dari 161,6-357,2 gr dengan berat rata-rata adalah 265,36 gr. Harganya juga yang bervariasi yang mulai dengan harga paling murah yaitu \$69,95 dan paling mahal \$250, dengan harga rata-rata adalah \$125,72.

	Price	Weight
count	48.000000	48.000000
mean	125.716667	265.360417
std	31.617932	43.060066
min	69.950000	161.600000
25%	100.000000	237.400000
50%	130.000000	266.500000
75%	150.000000	292.700000
max	250.000000	357.200000

**Tabel 2. Nilai Statistik dari Data**

Profit yang digunakan pada pengolahan data adalah data price barang dalam data. Weight yang digunakan pada pengolahan data adalah data weight (gr) barang dalam data. Nilai K yang digunakan adalah 8000 gr atau 8 kg.

Item	Price	Weight	Density	Profit	Weight	Density	Optimal
1	130	357,2	0,363941769	1	1	0	0
2	130	300,5	0,432612313	1	1	0	0

3	69,95	212,6	0,329021637	0	0	0	0
4	110	235,3	0,467488313	0	0	1	1
5	160	331,7	0,482363582	1	1	1	1
6	150	306,2	0,489875898	1	1	1	1
7	140	283,5	0,49382716	1	1	1	1
8	130	311,8	0,416933932	1	1	0	0
9	160	331,7	0,482363582	1	1	1	1
10	100	292	0,342465753	0	1	0	0
11	130	328,9	0,395256917	1	1	0	0
12	130	286,3	0,454069158	1	1	1	1
13	150	289,2	0,518672199	1	1	1	1
14	130	181,4	0,716648291	1	0	1	1
15	150	207	0,724637681	1	0	1	1
16	250	195,6	1,278118609	1	0	1	1
17	100	255,1	0,392003136	0	0	0	0
18	150	292	0,51369863	1	1	1	1
19	90	252,3	0,356718193	0	0	0	0
20	78	263,7	0,295790671	0	1	0	0
21	110	266,5	0,412757974	0	1	0	0
22	100	252,3	0,396353547	0	0	0	0
23	120	275	0,436363636	1	1	1	1
24	120	275	0,436363636	1	1	1	1
25	160	303,3	0,527530498	1	1	1	1
26	130	249,5	0,521042084	1	0	1	1
27	130	266,5	0,487804878	1	1	1	1
28	100	224	0,446428571	0	0	1	1
29	150	255,1	0,588004704	1	0	1	1
30	150	260,8	0,575153374	1	1	1	1
31	150	269,3	0,556999629	1	1	1	1
32	130	323,2	0,402227723	1	1	0	0
33	90	212,6	0,423330198	0	0	0	1
34	110	238,1	0,46199076	0	0	1	1
35	160	269,3	0,594132937	1	1	1	1
36	120	238,1	0,50398992	0	0	1	1
37	130	283,5	0,458553792	1	1	1	1
38	100	260,8	0,383435583	0	0	0	0
39	150	189,9	0,789889415	1	0	1	1
40	140	252,3	0,554894966	1	0	1	1
41	120	275	0,436363636	0	1	1	1
42	160	294,8	0,542740841	1	1	1	1
43	69,95	235,3	0,297280068	0	0	0	0
44	130	161,6	0,804455446	1	0	1	1
45	100	226,8	0,440917108	0	0	1	1

46	84,5	297,7	0,283842795	0	1	0	0
47	104	334,5	0,310911809	0	1	0	0
48	78	232,5	0,335483871	0	0	0	0
Total Bobot				7932,1	7969,1	7747,8	7960,4
Total Keuntungan				4200	3566,5	4210	4300
Total Barang				29	27	30	31

Tabel 3. Solusi dengan Algoritma greedy pada knapsack

## METHODS

### Particle Swarm Optimization

*Particle swarm optimization* merupakan salah satu cabang dari algoritma evolusi. PSO didasarkan pada perilaku sebuah kawanan burung atau ikan. Dimana sebuah kawanan tidak mempunyai pemimpin untuk menemukan makanan jadi mereka akan menyebar secara acak untuk menemukan letak makanan. Algoritma ini didasarkan dari perilaku sosial organisme ini. Perilaku sosial terdiri dari tindakan individu dan pengaruh dari individu-individu lainnya. Pada algoritma PSO ini, pencarian solusi dilakukan oleh suatu populasi yang terdiri dari beberapa partikel. Populasi dibangkitkan secara random dengan batasan nilai terkecil dan nilai terbesar. Setiap partikel mempresentasikan posisi dan lokasi dari permasalahan yang dihadapi.

Setiap partikel melakukan pencarian solusi yang optimal dengan kecerdasan pengalaman individu itu dengan melintasi dimensi ruang pencarian. Hal ini dilakukan dengan cara setiap partikel melakukan penyesuaian terhadap posisi terbaik dari partikel tersebut (*local best*) dan penyesuaian posisi partikel terbaik dari nilai terbaik dari seluruh kawanan (*global best*) selama melintasi ruang pencarian. Pada setiap iterasi, setiap solusi yang direpresentasikan oleh posisi partikel, dievaluasi kinerjanya dengan cara memasukkan solusi tersebut kedalam *fitness function*. Setiap partikel diperlakukan seperti sebuah titik pada suatu dimensi ruang tertentu. Kemudian terdapat dua factor yang memberikan karakter terhadap status partikel pada ruang pencarian yaitu posisi  $X$  dan kecepatan  $V$  partikel.

#### Partikel :

$X$  : posisi partikel saat ini dalam ruang pencarian

$P$  : posisi solusi terbaik yang ditemukan partikel

$V$  : arah terbang partikel

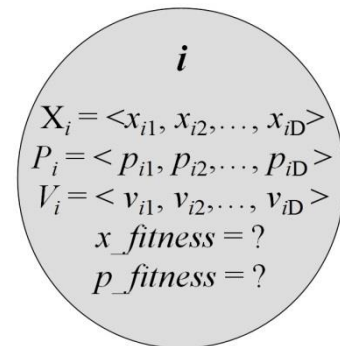
$x\_fitness$  : nilai *fitness* dari vektor  $X$

$p\_fitness$  : nilai *fitness* dari vektor  $P$

#### Update vektor *velocity*, kemudian posisi

$$v_{id} = v_{id} + \varphi_1 \times r \times (p_{id} - x_{id}) + \varphi_2 \times r \times (p_{gd} - x_{id}) \quad (4-1)$$

$$x_{id} = x_{id} + v_{id} \quad (4-2)$$





Dimana  $i$  adalah partikel ke- $i$ ,  $d$  adalah dimensi ke- $d$ ,  $\phi_1$  adalah laju belar (learning rates untuk komponen kognitif (kecerdasan individu),  $\phi_2$  adalah laju belajar komponen social (hubungan sosial antar individu),  $p$  adalah vektor nilai fitness terbaik yang dihasilkan sejauh ini,  $g$  adalah index dari partikel dengan ditness terbaik didalam ketetanggan, dan  $r$  adalah bilangan acak dalam interval  $[0,1]$

## Knapsack Problem

*Knapsack* dapat diartikan sebagai tas, ransel, karung atau kantong. *Knapsack problem* merupakan penyelesaian masalah optimasi kombinasi yang bertujuan untuk mendapatkan nilai maksimal dari barang-barang yang dimasukkan ke dalam *knapsack* atau wadah atau tempat dengan berat seminimal mungkin dan tidak melewati kapasitasnya. Dengan *knapsack problem* kita menentukan bagaimana memilih dari sekian banyak barang yang ada untuk dimasukkan kedalam *knapsack* sehingga mendapat penyimpanan yang optimal dengan mempertimbangkan barang yang terdiri dari  $n$  barang yang memiliki berat ( $W_n$ ) dan nilai profit ( $p_n$ ) dengan memperhatikan juga kapasitas dari tempat penyimpanan ( $W$ ) dan nilai probabilitas dari setiap barang ( $X_n$ ).

Masalah knapsack merupakan permasalahan optimisasi kombinatorial yang diklasifikasikan sebagai NP-complete problem. Tujuannya adalah untuk memperoleh keuntungan maksimum tanpa melebihi kapasitas knapsack dengan objek yang dipilih. Ada beberapa metode yang dikembangkan untuk permasalahan ini, salah satunya adalah metode meta-heuristic. Pada analisis ini, digunakan metode *particle swarm optimization* (PSO) untuk menyelesaikan masalah knapsack.

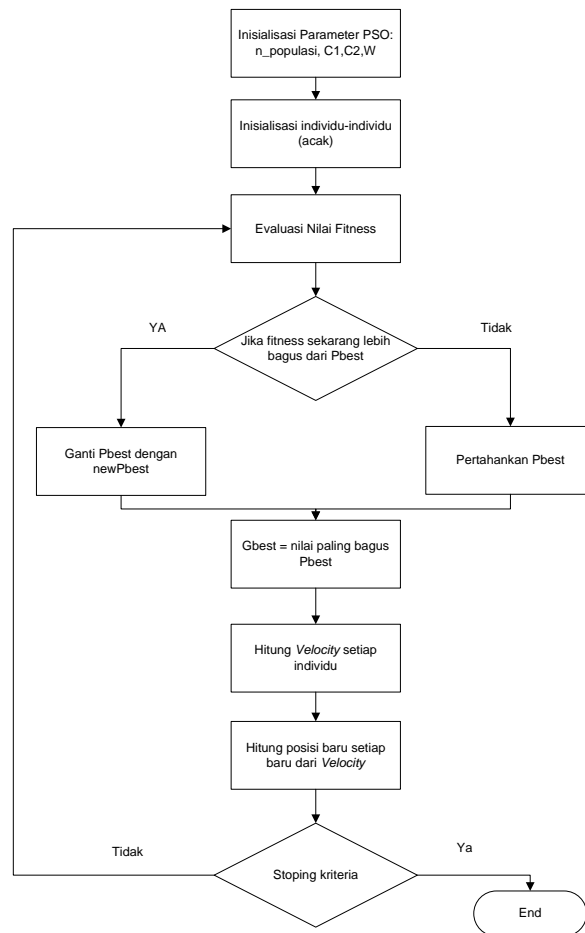
## Algoritma / Pseudocode PSO

```

for setiap partikel
    inisialisasi partikel
end
repeat
for setiap partikel
    hitung nilai fitness
    if nilai fitness baru lebih baik daripada nilai fitness lama maka
        perbarui nilai fitness dari partikel tersebut
    end
end
    pilih partikel dengan nilai fitness terbaik di antara semua partikel tetangganya
    dan simpan nilai fitness terbaik tersebut
for setiap partikel
    hitung velocity partikel menggunakan rumus pada persamaan 4-1
    perbarui posisi partikel menggunakan rumus pada persamaan 4-2
    end
until (kriteriaberkhenti = true)
pascaproses hasil dan visualisasi

```

## Flowchart Metode



**Gambar 6. Flowchart Algoritma PSO**

## Rancangan Metode

Penelitian penerapan *Particle Swarm Optimization* (PSO) pada knapsack problem dilakukan dengan langkah-langkah berikut :

- Studi literatur dilakukan dengan mencari dan mempelajari referensi yang berkaitan dengan permasalahan knapsack serta algoritme PSO pada jurnal internasional dan buku yang berkaitan.
- Mengidentifikasi data dilakukan dengan memilih data yang sesuai dengan permasalahan knapsack. Data yang digunakan mencakup nama barang, harga beli, harga jual dan bobot dari setiap barang, kemudian dicari keuntungan setiap barang yang diperoleh dari selisih harga beli dan harga jual yang telah ditetapkan.
- Menerapkan PSO pada data yang telah teridentifikasi lalu membuat programnya.
- Mengatur kriteria pemberhentian pada kedua algoritme tersebut dengan iterasi sebanyak 100 iterasi.

- e. Membandingkan hasil solusi dari PSO dengan Algorithma Greedy.
- f. Membuat kesimpulan berdasarkan hasil yang diperoleh dari algoritme tersebut.

Metode yang digunakan adalah *Particle Swarm Optimization* (PSO) Pada Knapsack Problem dengan rancangan metode ini yaitu seperti berikut :

1. Inisialisasi parameter PSO (inersia) dan parameter Knapsack problem.
2. Perbarui Pbest dan Gbest.
3. Perbarui kecepatan dan posisi setiap individu menggunakan Persamaan yang ada.
4. Dengan kecepatan baru yang didapat, perbarui posisi setiap individu menggunakan Persamaan yang ada.
5. Tentukan nilai k.
6. Pbest adalah pengabungan dari PbestPSO dan Knapsack problem.
7. Evaluasi nilai fitness dari Pbest.
8. Jika sudah memenuhi kriteria pemberhentian atau maksimum iterasi maka didapatkan nilai individu terbaik, jika tidak maka kembali lagi pada langkah 2.

## RESULTS AND DISCUSSIONS

Digunakan software jupyter notebook, colab, dan juga exel untuk mencari penyelesaian knapsack menggunakan algoritme PSO. Dilakukan percobaan pertama dan kedua menggunakan parameter yang disajikan pada tabel 4 dan 5 berikut :

Parameter	Nilai		
n_individu	50	50	50
n_barang	48	48	48
a	-5	-5	-5
b	5	5	5
w1	2	1	0.5
w2	2	2	2
max_generasi	10	10	10
alpha	3	3	3
beta0	1	1	1
gamma	0.1	0.1	0.1
lim	8000	8000	8000
vmax	100	100	100

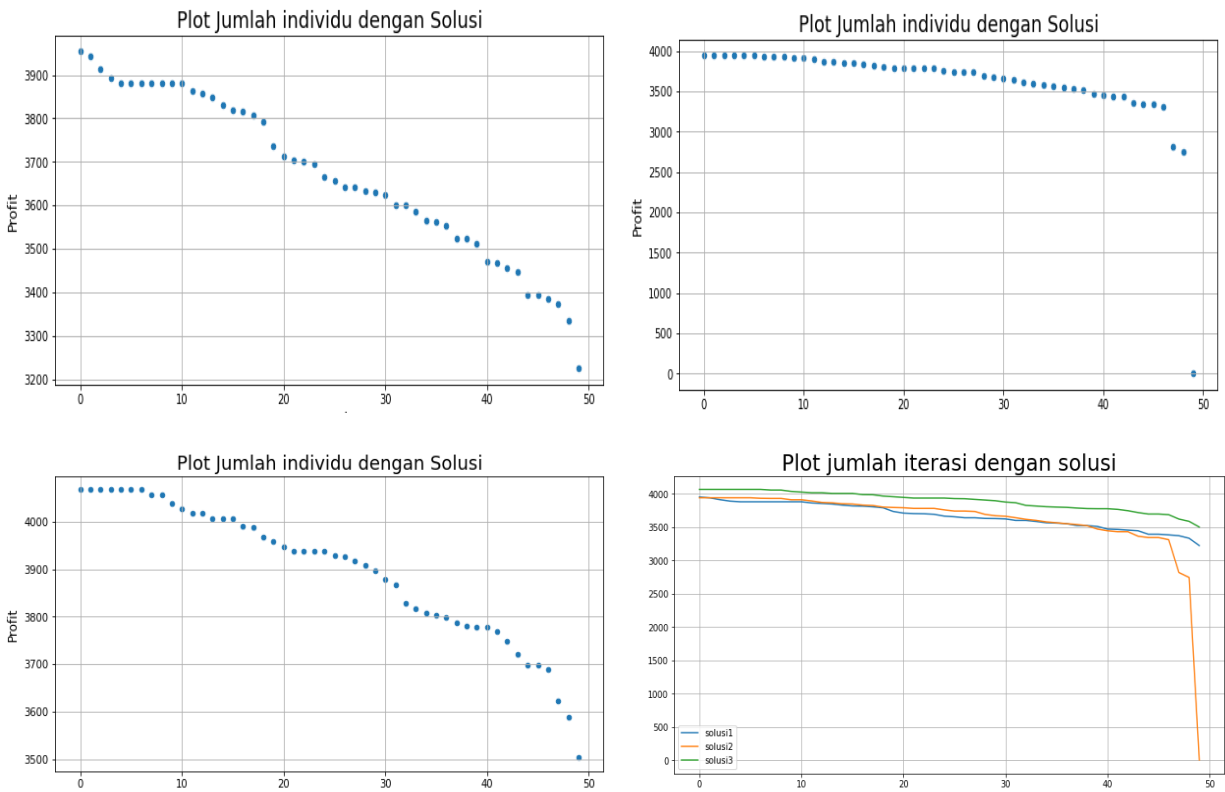
**Tabel 4. Parameter Percobaan Pertama**

Parameter	Nilai		
n_individu	50	50	50
n_barang	48	48	48
a	-5	-5	-5
b	5	5	5
w1	2	2	2
w2	2	1	0.5
max_generasi	10	10	10
alpha	3	3	3
beta0	1	1	1
gamma	0.1	0.1	0.1
lim	8000	8000	8000
vmax	100	100	100

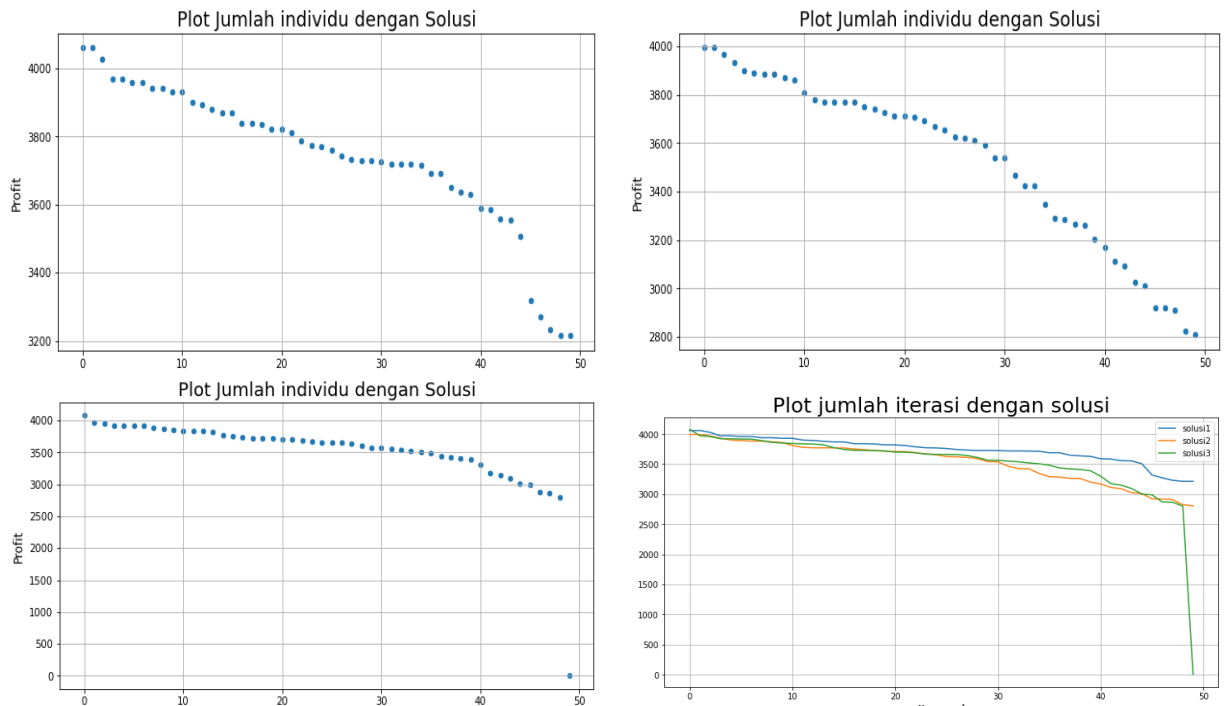
**Tabel 5. Parameter Percobaan Kedua**

Pada percobaan pertama menggunakan parameter pada tabel 1 yaitu melakukan variasi pada nilai w1. Ternyata semakin kecil nilai w1 maka nilai fitness yang didapatkan semakin baik atau berprogress positif. Oleh sebab itu pada percobaan ke 2 pada tabel 2 kita akan melakukan hal yang sama yaitu menurunkan nilai w2 secara perlahan. Dapat disimpulkan bahwa pada awalnya w bernilai 2 dan selanjutnya diturunkan secara bertahap pada setiap percobaan. Ketika w bernilai 2 proses pencarian terjadi secara global ke segala arah. Sementara itu ketika w diturunkan, pencarian data menjadi lebih fokus ke suatu area tertentu

yang potensial. Berikut plot hasil dari percobaan pertama dan kedua.



**Gambar 7. Plot jumlah iterasi dengan masing-masing solusi PSO percobaan 1**



**Gambar 8. Plot jumlah iterasi dengan masing-masing solusi PSO percobaan 2**

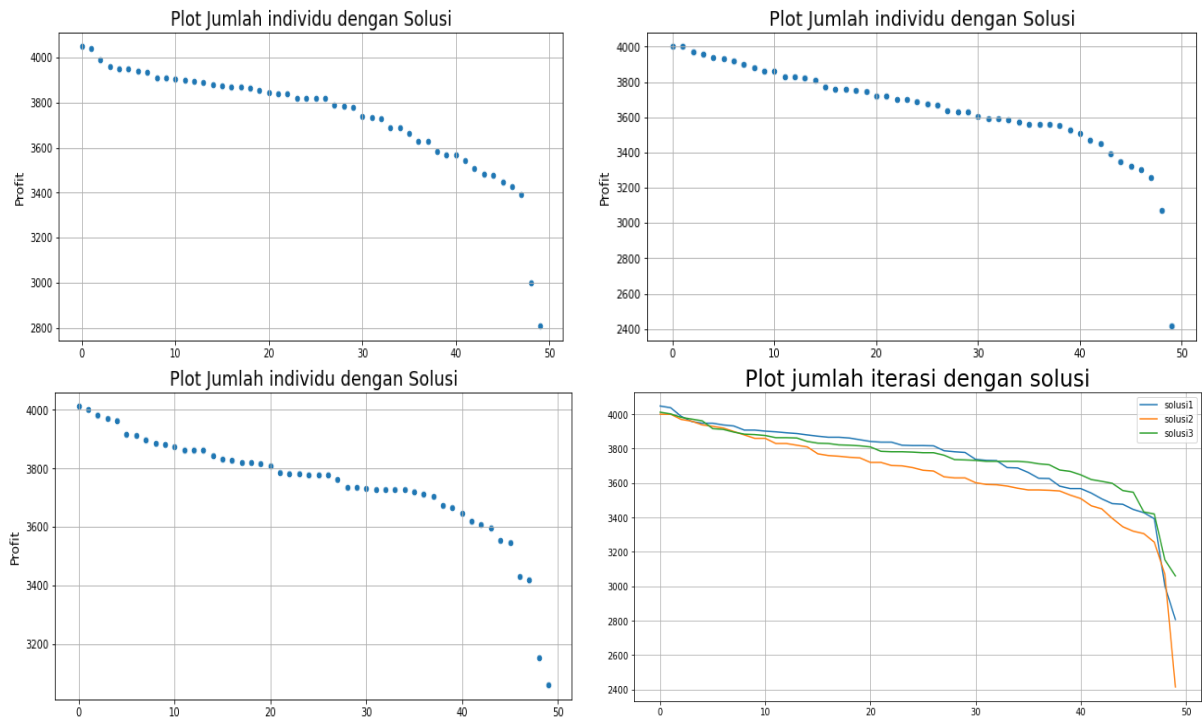
## Modifikasi PSO

$$v_{id} = \omega \times v_{id} + \varphi_1 \times r \times (p_{id} - x_{id}) + \varphi_2 \times r \times (p_{gd} - x_{id}) \quad (4-3)$$

Sehingga pada percobaan ketiga akan melakukan perubahan pada parameter dimana nilai  $w_1$  dan  $w_2$  akan tetap dengan hasil terbaik dari sebelumnya yaitu kisaran 0.5. variasi yang akan dilakukan kali adalah  $v_{max}$  untuk melihat seberapa pengaruh  $v_{max}$  terhadap partikel dan nilai profitnya. Berikut parameter yang digunakan pada tabel 6.

Parameter	Nilai		
n_individu	50	50	50
n_barang	48	48	48
a	-5	-5	-5
b	5	5	5
w1	0.5	0.5	0.5
w2	0.5	0.5	0.5
max_generasi	10	10	10
alpha	3	3	3
beta0	1	1	1
gamma	0.1	0.1	0.1
lim	8000	8000	8000
vmax	100	70	40

Tabel 6. Parameter Percobaan Ketiga



Gambar 9. Plot jumlah iterasi dengan masing-masing solusi PSO percobaan 3

Dari hasil percobaan ketiga didapatkan bahwa vmax dengan nilai 100 lebih baik dibandingkan dengan vmax dengan nilai 70 atau 40. Hal itu terjadi karena kecepatan maksimum mendukung pergerakan pada partikel sehingga semakin baik vmax maka pergerakan partikel semakin baik pula. Setelah mendapatkan pola pada setiap parameter maka dilakukanlah percobaan keempat dengan w1, w2, dan vmax yang tetap namun dengan variasi nilai individu. Berikut parameter yang digunakan pada percobaan keempat ada pada tabel 7.

Parameter	Nilai		
n_individu	100	50	20
n_barang	48	48	48
a	-8	-8	-8
b	8	8	8
max_generasi	10	10	10
alpha	3	3	3
beta0	1	1	1
gamma	0.1	0.1	0.1
lim	8000	8000	8000
w1	0.2	0.2	0.2
w2	0.5	0.5	0.5
vmax	100	100	100

**Tabel 7. Parameter Pada Percobaan Keempat**

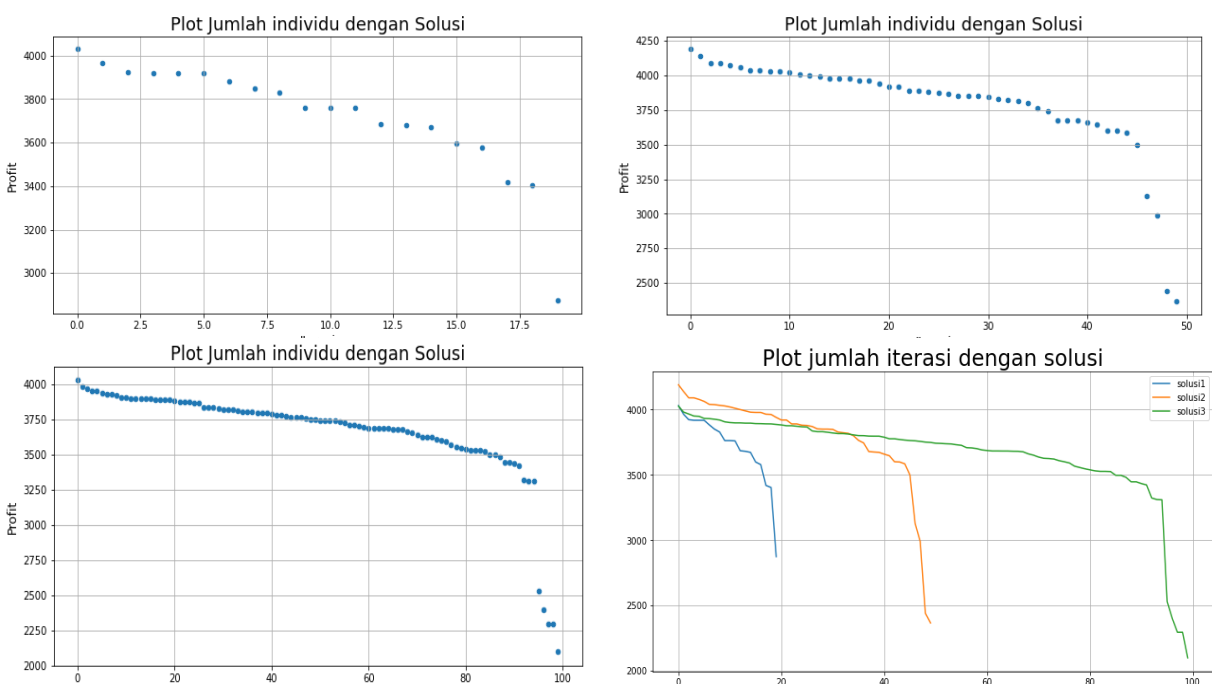
Dengan ketentuan barang yang dipilih ditandai dengan angka (1.0) sedangkan barang yang tidak dipilih ditandai dengan angka (0.0). Sehingga diperoleh nilai fitness pada percobaan keempat hasil ke-1 adalah 4270 dengan weight 7991.6 dan 30 barang. Pada hasil ke-2 adalah 4290 dengan weight 7977.4 dan 31 barang. Pada hasil ke-3 adalah 4114.5 dengan weight 7991.6 dan 31 barang. Dari ketiga solusi dapat disimpulkan bahwa nilai individu terbaik pada kasus ini adalah 50 individu sehingga didapatkan solusi terbaik adalah percobaan ketiga pada hasil ke-2 dengan nilai fitness paling tinggi yaitu 4290. Solusi knapsacknya adalah (0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0). Barang yang diambil adalah barang ke-4, 5, 6, 7, 9, 12-16, 18, 21, 22, 23, 25-31, 33-37, 39-42, dan 44.

Barang	Hasil ke-		
	1	2	3
Barang1	0.0	0.0	1.0
Barang2	1.0	0.0	0.0
Barang3	0.0	0.0	0.0
Barang4	0.0	1.0	1.0
Barang5	1.0	1.0	1.0
Barang6	1.0	1.0	1.0
Barang7	1.0	1.0	1.0

Barang8	1.0	0.0	0.0
Barang9	1.0	1.0	0.0
Barang10	0.0	0.0	1.0
Barang11	1.0	0.0	0.0
Barang12	0.0	1.0	0.0
Barang13	1.0	1.0	0.0
Barang14	1.0	1.0	1.0
Barang15	1.0	1.0	1.0
Barang16	1.0	1.0	1.0
Barang17	0.0	0.0	0.0
Barang18	1.0	1.0	1.0
Barang19	0.0	0.0	1.0
Barang20	0.0	0.0	0.0
Barang21	0.0	1.0	0.0
Barang22	0.0	1.0	0.0
Barang23	1.0	1.0	1.0
Barang24	1.0	0.0	1.0
Barang25	1.0	1.0	1.0
Barang26	1.0	1.0	1.0
Barang27	1.0	1.0	1.0
Barang28	0.0	1.0	1.0
Barang29	1.0	1.0	1.0
Barang30	1.0	1.0	0.0
Barang31	1.0	1.0	1.0
Barang32	1.0	0.0	0.0
Barang33	0.0	1.0	1.0
Barang34	1.0	1.0	1.0
Barang35	1.0	1.0	1.0
Barang36	1.0	1.0	1.0
Barang37	1.0	1.0	1.0
Barang38	0.0	0.0	0.0
Barang39	1.0	1.0	1.0
Barang40	1.0	1.0	1.0
Barang41	0.0	1.0	1.0
Barang42	1.0	1.0	1.0
Barang43	0.0	0.0	0.0
Barang44	1.0	1.0	1.0
Barang45	1.0	0.0	1.0
Barang46	0.0	0.0	1.0

Barang47	0.0	0.0	0.0
Barang48	0.0	0.0	0.0
Profit	4270.0	4290.0	4114.5
Weight	7991,6	7977,4	7991,6
Total Barang	30	31	31

**Tabel 8. Hasil Pada Percobaan keempat**



**Gambar 10. Plot jumlah iterasi dengan masing-masing solusi PSO percobaan 4**

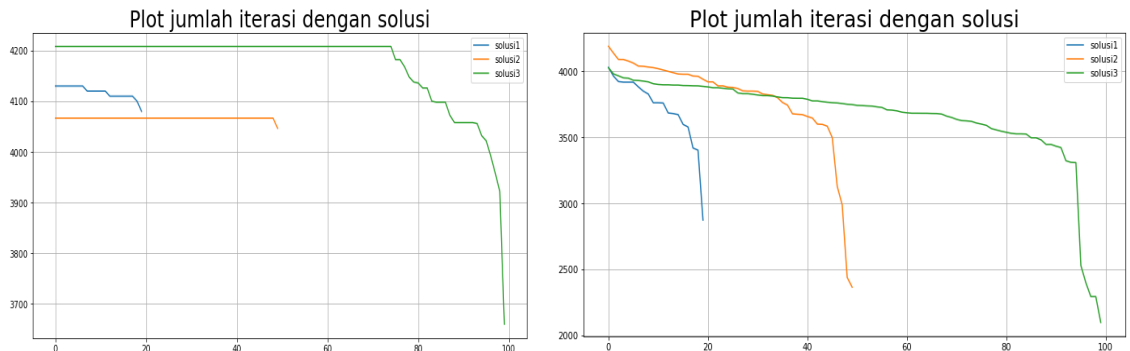
Selain solusi terbaik yang didapatkan ada parameter terbaik yang berperan dalam pencarian solusi. Parameter terbaik adalah parameter yang digunakan pada solusi terbaik yaitu pada percobaan ketiga pada percobaan keempat hasil ke-2 terbukti dengan nilai profit yang paling tinggi. Berikut parameter terbaik yang digunakan pada percobaan kali ini :

Parameter	Keterangan	Nilai
n_individu	Jumlah Individu	50
n_barang	Jumlah Barang	48
a	batas bawah	-8
b	batas atas	8
$\alpha$	alpha	3
$\beta$	beta0	1
max_generasi	Max generasi	100
$\gamma$	gamma	0.1
limit	Kapasitas tempat	8000
Vmax	Kecepatan partikel	100
w1	Bobot inersia 1	0.5
w2	Bobot inersia 2	0.5



**Tabel 9. Parameter Terbaik Pada Percobaan**

### Visualisasi Iterasi



**Gambar 12. Plot solusi PSO pada iterasi ke-1 dan ke-20**

## CONCLUSION

Hasil penyelesaian knapsack problem menggunakan *Particle Swarm Optimization* (PSO) Pada Knapsack Problem dengan data penjualan pada *brand Brooks Running Shoes* menunjukkan bahwa PSO pada knapsack dapat menghasilkan solusi optimal. Namun berdasarkan hasil penelitian ini dapat dilihat bahwa pada kasus yang ada, algoritme greedy mempunyai performa yang lebih baik untuk menyelesaikan permasalahan knapsack 0-1 dibandingkan dengan algoritme *Particle Swarm Optimization* PSO. Pada algoritma greedy menghasilkan nilai fitness 4300 dengan weight 7960.4 serta barang yang dikemas adalah 31 barang. Sedangkan pada PSO menghasilkan nilai fitness terbaik pada percobaan keempat di hasil ke-2 yaitu 4290 dengan weight 7977.4 dan 31 barang.

Dapat diperoleh pemahaman baru bahwa nilai  $w_1$  dan  $w_2$  atau bobot inersia mengalami peningkatan keefektifan saat mengalami pengurangan nilai dari yang lebih tinggi ke nilai yang lebih rendah. Apabila  $v_{max}$  semakin tinggi nilainya maka akan semakin baik partikel tersebut bekerja. Sehingga permasalahan pada *brand Brooks Running Shoes* dapat diselesaikan dengan metode *Particle Swarm Optimization* (PSO) menggunakan Knapsack Problem. Hanya perlu disesuaikan saja pada nilai  $k$  atau kapasitas yang ada untuk mendapatkan nilai keuntungan yang maksimal.

Penelitian ini darisegi algoritma dan metode untuk membuat program pasti tidak 100% baik dan efektif maka dari itu penelitian selanjutnya perlu diperbaiki dan dikembangkan algoritma tersebut supaya lebih baik dan efektif untuk menyelesaikan integer knapsack problem. Masih terbuka bagi peneliti berikutnya untuk menerapkan algoritme meta-heuristic yang lain dan membandingkan hasil dari setiap algoritme tersebut, sehingga didapatkan metode penyelesaian terbaik untuk menyelesaikan permasalahan knapsack 0-1. Peneliti selanjutnya bisa menggunakan permasalahan yang berbeda atau sudah dikembangkan guna menyesuaikan permasalahan pada realita yang ada.

## REFERENCES

1. Referensi ilmiah yang digunakan

A Modified Binary Particle Swarm Optimization for Knapsack Problems, Applied Mathematics and Computation. Journal homepage: [www.elsevier.com/locate/amc](http://www.elsevier.com/locate/amc). (Jagdish Chand Bansal , Kusum Deep)

### Resume :

Judul : A Modified Binary Particle Swarm Optimization for Knapsack Problems

1. knapsack membutuhkan subset dari beberapa item yang diberikan untuk dipilih sedemikian rupa sehingga jumlah keuntungan yang sesuai dimaksimalkan tanpa melebihi kapasitas Ransel. Berbagai jenis Knapsack Problem terjadi, tergantung pada distribusi item dan knapsack:
  - a. Knapsack Problem: setiap item dapat dipilih paling banyak satu kali.
  - b. Bounded Knapsack Problem: jika setiap item dapat dipilih beberapa kali.
  - c. Soal Knapsack Pilihan Ganda: jika item dibagi lagi menjadi beberapa kelas berhingga dan tepat satu item harus diambil dari setiap kelas.
  - d. Soal Knapsack Multidimensi atau Multidimensi: jika kita mempunyai  $n$  benda dan  $m$  knapsack dengan kapasitas tidak harus sama dan knapsack harus diisi secara bersamaan.

2. Binary Particle Swarm Optimization

3. The proposed Modified Binary Particle Swarm Optimization (MBPSO)

### Hasil :

Dalam MBPSO yang diusulkan, Persamaan pembaruan posisi menunjukkan bahwa dalam proses pencarian, perilaku partikel sangat bergantung pada  $V_{max}$  (yaitu, MBPSO yang diusulkan sensitif dengan pilihan parameter  $V_{max}$ ). Oleh karena itu, dilakukan eksperimen untuk mencari nilai  $V_{max}$  yang paling sesuai. Penyesuaian halus ini dilakukan untuk lima masalah tes pertama.

2. Solving Knapsack Problems using Radius Particle Swarm Optimization fuse with Simulated Annealing, (Mudarmeen Munlin).

### Resume :

Jurnal tersebut menyajikan pendekatan baru yang menggabungkan optimalisasi gerombolan partikel radius dan anil simulasi (RPSOSA) untuk menyelesaikan KP. Hasil dari metode yang diusulkan akan dibandingkan dengan optimasi gerombolan partikel radius individu (RPSO) dan anil simulasi (SA).

### Pengaturan parameter :

Ukuran swarm adalah 60, nomor iterasi adalah 4000 atau 240000 evaluasi fungsi, dan sama dan dalamnya kisaran  $[-4, 4]$ ,  $w$  adalah 0,98, laju pendinginan adalah 0,95 dan beku adalah 0,001.

### Hasil :

Optimal lokal sering ditemukan dalam masalah knapsack. Penulis mengusulkan pendekatan baru untuk memecahkan masalah knapsack dengan menggabungkan optimasi radius partikel (RPSO) dan algoritma simulasi anil (SA). Struktur metode memungkinkan keunggulan RPSO yang memiliki pencarian global yang kuat kemampuan dan SA yang memiliki kemampuan pencarian lokal yang kuat untuk diperoleh solusi optimal dengan cepat.

### 3. Sumber data yang digunakan

<https://www.kaggle.com/hannahcollins/2020-brooks-running-shoes>

- desc : Kumpulan data ini mencakup 26 model sepatu Brooks Running terkini yang diperoleh dari <http://brooksrunning.com/>. Data terdiri dari : Nama sepatu, Tipe, Harga (per 29 Agustus 2020), Dukungan, Pengalaman, Permukaan, Penurunan midsole dalam mm, Berat dalam g, Jenis lengkungan, dan Fitur sepatu tambahan.

<https://www.kaggle.com/tizianoparriani/genetic-algorithms-for-0-1-knapsack-problem/notebook>

- desc : Kumpulan 500 data barang yaitu sepatu. Data terdiri dari : Harga Sepatu dan Berat sepatu dalam gram.

## APPENDIX

### 1. Link code untuk percobaan :

<https://colab.research.google.com/drive/13e0CPXmfYsKvixymPPY8DXHq2vzSSxA#scrollTo=znGy6monO2Z7>

### 2. Tabel pembagian tugas.

No	Nama	NIM	Job desk	Deskripsi Tugas
1	Arfyani Deiastuti	120450006	Ketua Kelompok	Mencari data dan mengaplikasikan Algoritma di google colab, introduction, problem statement, Data Description, Methods, Results and Discussion, conclusion, dan PPT.
2	Hanna Septiani	120450064	Anggota 1	Mencari referensi jurnal dan membuat resumenya, conclusion, dan PPT.
3	Alfianri Mahnihuruk	120450088	Anggota 2	Mencari data dan mengaplikasikan Algoritma di google colab, Data Description, Methods, dan PPT.
4	Junpito Salim	120450086	Anggota 3	Mencari data dan mengaplikasikan Algoritma di google colab, introduction, problem statement, dan PPT.

