

[Penerapan Firefly Algorithm dengan Metode Knapsack pada Komoditas Alpukat di Wilayah California]

Laporan Akhir Swarm Intelligence

Kelompok 8

Ketua Kelompok : 120450011 – Fadia Dilla Sabine

Anggota 1 : 120450003 – Fadillah Muhesa Latri

Anggota 2 : 120450029 - Khoirunnisa

Anggota 3 : 120450055 - Lisnurani

INTRODUCTION

RBL atau *Research Based Learning* merupakan suatu tugas untuk merancang atau mengolah dan mendapatkan hasil melalui penerapan dari metode atau ilmu tertentu. Pada *Research Based Learning (RBL)* ini akan dilakukan pengolahan dan analisis data menggunakan algoritma Firefly Algorithm dengan metode Knapsack. Variabel yang digunakan untuk pengolahan data adalah tanggal dan tahun, *average price*, *total volume*, dan *type* (jenis). Seiring dengan perkembangan zaman dan teknologi, khususnya pada sektor perkebunan, pada saat ini membutuhkan suatu solusi untuk mengkategorikan atau menentukan jenis-jenis alpukat unggulan yang memiliki dengan profit atau keuntungannya, sehingga dapat diketahui jenis alpukat mana yang paling sedikit hingga yang paling banyak memberi keuntungan. Penelitian yang kami lakukan ini merupakan penelitian yang sangat menarik karena dapat mengetahui jenis alpukat yang paling banyak memberikan keuntungan di wilayah California.

PROBLEM STATEMENT

Pada laporan RBL (*Research Based Learning*) yang dilakukan, masalah yang akan diselesaikan yaitu permasalahan terkait jenis alpukat di California. Permasalahan tersebut bertujuan untuk mengetahui jenis alpukat di California yang paling banyak membawa keuntungan dan tidak melebihi volume Knapsack yang telah ditentukan. Data yang akan diolah adalah data komoditas alpukat di California dari tahun 2021. Untuk dapat mengolah data tersebut, terdapat beberapa variable yang akan digunakan, yaitu tanggal dan tahun, *average price*, *total volume* dan *type* (jenis) menggunakan algoritma Firefly dengan metode Knapsack.

DATA DESCRIPTION

Pada RBL ini, digunakan data yang berisi mengenai penjualan alpukat di beberapa negara. Namun, data yang akan diambil dan diolah adalah data komoditas alpukat dari wilayah California. Pada data mentah penjualan alpukat di wilayah California, terdapat variable-variable seperti tanggal, *average price*, *total volume*, *small bags*, *total bags*, *large bags*, *small bags*, *Xlarge bags*, *type* (konvensional dan organik), tahun dan wilayah. Namun, data yang akan diolah hanya akan menggunakan tanggal, *average price*, *total volume* dan *type* (jenis). Jumlah fitur data yang dikelola berjumlah 96.

Tanggal	Average Price	Total Volume	Type
01/04/2021	0.991030549	7261522.15	Conventional
01/04/2021	1.648.762.384	333216.18	Organic
01/10/2021	1.133.327.226	6913269.84	Conventional
01/10/2021	1.810.751.702	331679.18	Organic
1/17/2021	1.221.774.373	6564907.82	Conventional
1/17/2021	1.813.524.861	350705.52	Organic

1/24/2021	1.184.074.160	6555971.65	Conventional
1/24/2021	1.771.284.092	357493.91	Organic
1/31/2021	1.193.617.315	6320461.03	Conventional
1/31/2021	1.601.993.396	356459.66	Organic
02/07/2021	0.950597576	9086736.76	Conventional
02/07/2021	1.659.568.910	368847.43	Organic
2/14/2021	1.159.434.749	6076395.06	Conventional
2/14/2021	1.742.132.918	341638.55	Organic
2/21/2021	1.192.972.621	6119428.37	Conventional
2/21/2021	1.622.611.266	354657.81	Organic
2/28/2021	1.153.031.187	6333693.00	Conventional
2/28/2021	1.499.500.494	501557.15	Organic
03/07/2021	1.257.815.400	6129704.88	Conventional
03/07/2021	1.595.021.789	431349.42	Organic
3/14/2021	1.194.553.006	6401275.07	Conventional
3/14/2021	1.707.021.585	349311.21	Organic
3/21/2021	1.287.600.856	5875770.59	Conventional
3/21/2021	1.778.013.860	334908.18	Organic
3/28/2021	1.254.962.220	6217134.06	Conventional
3/28/2021	1.759.633.425	341646.15	Organic
04/04/2021	1.441.819.191	5744292.63	Conventional
04/04/2021	1.777.884.878	379248.92	Organic
04/11/2021	1.361.756.444	5902693.04	Conventional
04/11/2021	1.765.553.497	359824.51	Organic
4/18/2021	1.412.836.305	5886862.75	Conventional
4/18/2021	1.791.969.070	366838.65	Organic
4/25/2021	1.420.226.307	5670058.34	Conventional
4/25/2021	1.822.630.446	373711.83	Organic
05/02/2021	1.150.522.457	6980342.70	Conventional
05/02/2021	1.779.789.015	350868.26	Organic
05/09/2021	1.315.807.039	6184369.12	Conventional
05/09/2021	1.841.056.236	368474.94	Organic
5/16/2021	1.320.280.635	5917100.80	Conventional
5/16/2021	1.816.060.571	351133.39	Organic
5/23/2021	1.275.761.798	5936707.28	Conventional
5/23/2021	1.782.421.117	350232.25	Organic
5/30/2021	1.388.782.388	5935418.70	Conventional
5/30/2021	1.789.221.699	345752.08	Organic
06/06/2021	1.273.816.381	6262807.64	Conventional
06/06/2021	1.873.293.185	337376.96	Organic
6/13/2021	1.327.515.637	5961933.28	Conventional
6/13/2021	1.829.291.086	353109.13	Organic

6/20/2021	1.317.251.452	6509782.09	Conventional
6/20/2021	1.791.796.439	323531.91	Organic
6/27/2021	1.357.793.280	5805434.37	Conventional
6/27/2021	1.845.446.063	282081.97	Organic
07/04/2021	1.413.462.477	6194388.22	Conventional
07/04/2021	1.732.662.414	317385.99	Organic
07/11/2021	1.206.707.033	6363798.46	Conventional
07/11/2021	1.765.940.155	279193.99	Organic
7/18/2021	1.199.959.122	6545351.35	Conventional
7/18/2021	1.917.845.850	322975.41	Organic
7/25/2021	1.324.745.302	5751421.73	Conventional
7/25/2021	1.698.949.665	439169.44	Organic
08/01/2021	1.279.567.984	6063190.30	Conventional
08/01/2021	1.868.103.634	331525.55	Organic
08/08/2021	1.228.550.238	6168481.78	Conventional
08/08/2021	1.940.333.823	324166.74	Organic
8/15/2021	1.368.022.964	5711724.78	Conventional
8/15/2021	1.910.638.844	329857.18	Organic
8/22/2021	1.278.281.052	5795558.96	Conventional
8/22/2021	1.835.193.923	341694.42	Organic
8/29/2021	1.308.969.395	5800961.98	Conventional
8/29/2021	1.952.906.897	313850.24	Organic
09/05/2021	1.221.680.881	6827155.78	Conventional
09/05/2021	1.903.557.089	319705.35	Organic
09/12/2021	1.408.968.480	5530353.21	Conventional
09/12/2021	1.851.289.385	342082.20	Organic
9/19/2021	1.369.312.801	5582003.13	Conventional
9/19/2021	1.870.709.601	304924.74	Organic
9/26/2021	1.374.619.936	5583495.76	Conventional
9/26/2021	1.844.615.422	303937.72	Organic
10/03/2021	1.404.650.322	5342750.48	Conventional
10/03/2021	1.974.513.878	290234.25	Organic
10/10/2021	1.352.535.280	5711161.96	Conventional
10/10/2021	1.917.919.074	282112.12	Organic
10/17/2021	1.219.225.361	6121452.61	Conventional
10/17/2021	1.963.364.153	302646.90	Organic
10/24/2021	1.192.314.553	5759682.58	Conventional
10/24/2021	1.990.725.035	277978.00	Organic
10/31/2021	1.349.447.087	5127369.92	Conventional
10/31/2021	2.037.134.382	292159.28	Organic
11/07/2021	1.135.806.218	6029720.97	Conventional
11/07/2021	1.941.559.315	325146.50	Organic

11/14/2021	1.339.934.545	5285283.58	Conventional
11/14/2021	1.918.552.210	332879.74	Organic
11/21/2021	1.402.693.341	4863668.20	Conventional
11/21/2021	1.943.251.301	302729.40	Organic
11/28/2021	1.414.120.857	4631404.37	Conventional
11/28/2021	1.814.788.318	284916.56	Organic

METHODS

a) Deskripsi Umum Metode RBL

Pada RBL (Research Based Learning) ini, digunakan algoritma Firefly Algorithm dengan metode Knapsack. Firefly Algorithm (atau Algoritma Kunang-Kunang) merupakan algoritma yang digunakan untuk pengotimasian dalam pengambilan keputusan pada suatu data tertentu. Algoritma ini terinspirasi dari tingkah laku kunang-kunang yang menyala dan berkedip. Tujuannya adalah sebagai sinyal untuk dapat menarik kunang-kunang yang lain menuju dirinya. Kunang-kunang yang menyala lebih terang akan dapat menarik kunang kunang yang menyala dengan cahaya kurang terang menuju dirinya (PipTools, 2015). Pada kunang-kunang tersebut akan mengeluarkan suatu cahaya yang disebut intensitas cahaya. Intensitas cahaya tersebut sebanding dengan nilai fungsi sebagai tujuan untuk permasalahan optimalisasi.

Knapsack Problem merupakan suatu masalah mengenai cara menentukan pemilihan barang dari sekumpulan barang di mana setiap barang mempunyai berat dan profit atau nilai (Anisa, 2016). Knapsack merupakan suatu jenis masalah yang populer pada bidang kombinatorial dan optimasi. Permasalahan pada Knapsack adalah bagaimana cara memilih objek agar dapat dimasukkan ke dalam Knapsack sehingga kendala-kendala dapat terpenuhi dan memperoleh suatu keuntungan yang maksimum. Permasalahan Knapsack merupakan permasalahan yang sering dihadapi oleh perusahaan dalam pengelolaan barang (Anisa, 2016).

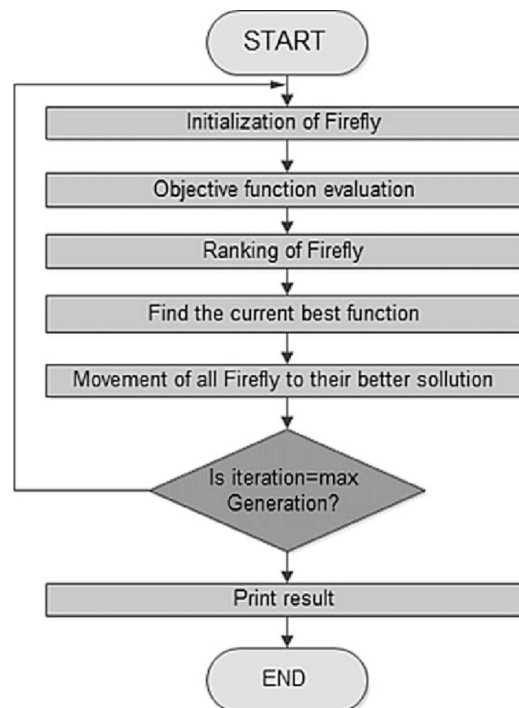
b) Pseudocode yang Digunakan

1. Initialisation max iteration, α , β , γ , n
2. Generate initial population
3. Define the objective function $f(x)$
4. Determine Intensity (I) at cost (x) of each individual determined by $f(x_i)$
5. While ($t < \text{Iter max}$)
 - for $i = 1$ to n

- ```

 for j = 1 to n
 if ($I_j > I_i$)
 Move firefly i towards j in K dimension
 end if
 Evaluate new solutions and update light intensity
 end for j
end for i
Rank the fireflies and find the current best
end while
6. Post process results and visualization
End procedure

```
- c) Penjelasan mengenai modifikasi dari metode yang digunakan
- Firefly algorithm (algoritma kunang-kunang) dibuat untuk menyelesaikan permasalahan yang bersifat continue. Maka, untuk menyelesaikan masalah diskrit (discrete optimization), harus dilakukan diskritisasi.
- d) Flowchart dari metode yang digunakan dalam RBL
- Berikut ini merupakan diagram alir atau flowchart yang digunakan untuk Firefly Algorithm



Gambar 1.1 Flowchart dari Firefly Algorithm

## RESULTS AND DISCUSSIONS

### 1) Solusi terbaik

Pada analisis, dilakukan beberapa percobaan untuk menemukan solusi terbaik.

Pada percobaan pertama divariasikan parameter dengan nilai  $n\_individu$ : 100, 50, 30, 20, 10, 10. Nilai  $a$  dan  $b$  adalah -4 dan 4, nilai  $\alpha$  dan  $\beta$  adalah 1, serta variasi nilai  $\gamma$ : 0.00001, 0.10000, 0.01000, dan 0.00100, menggunakan limit 100. Seperti berikut:

|   | $n\_individu$ | $a$ | $b$ | $\alpha$ | $\beta$ | $\gamma$ | $n\_barang$ | lim | max_generasi |
|---|---------------|-----|-----|----------|---------|----------|-------------|-----|--------------|
| 0 | 100.0         | -4  | 4   | 1        | 1.0     | 0.00001  | 96          | 100 | 3            |
| 1 | 50.0          | -4  | 4   | 1        | 1.0     | 0.10000  | 96          | 100 | 3            |
| 2 | 30.0          | -4  | 4   | 1        | 1.0     | 0.10000  | 96          | 100 | 3            |
| 3 | 20.0          | -4  | 4   | 1        | 1.0     | 0.10000  | 96          | 100 | 3            |
| 4 | 10.0          | -4  | 4   | 1        | 1.0     | 0.01000  | 96          | 100 | 3            |
| 5 | 10.0          | -4  | 4   | 1        | 1.0     | 0.00100  | 96          | 100 | 3            |

Namun, pada percobaan pertama menghasilkan profit 0 yang artinya percobaan pertama bukan merupakan solusi terbaik dari analisis yang dilakukan.

Kemudian, dilakukan percobaan kedua. Pada percobaan kedua, parameter untuk  $n\_individu$ ,  $a$ ,  $b$ ,  $\alpha$ ,  $\beta$ , dan  $\gamma$  sama seperti percobaan pertama. Namun digunakan limit yang berbeda. Pada percobaan kedua, digunakan limit 307.

|   | $n\_individu$ | $a$ | $b$ | $\alpha$ | $\beta$ | $\gamma$ | $n\_barang$ | lim | max_generasi |
|---|---------------|-----|-----|----------|---------|----------|-------------|-----|--------------|
| 0 | 100.0         | -4  | 4   | 1        | 1.0     | 0.00001  | 96          | 307 | 3            |
| 1 | 50.0          | -4  | 4   | 1        | 1.0     | 0.10000  | 96          | 307 | 3            |
| 2 | 30.0          | -4  | 4   | 1        | 1.0     | 0.10000  | 96          | 307 | 3            |
| 3 | 20.0          | -4  | 4   | 1        | 1.0     | 0.10000  | 96          | 307 | 3            |
| 4 | 10.0          | -4  | 4   | 1        | 1.0     | 0.01000  | 96          | 307 | 3            |
| 5 | 10.0          | -4  | 4   | 1        | 1.0     | 0.00100  | 96          | 307 | 3            |

Pada percobaan kedua, diperoleh hasil sebagai berikut:

```
[] hasil[0]
```

| Barang 1 | Barang 2 | Barang 3 | Barang 4 | Barang 5 | Barang 6 | Barang 7 | Barang 8 | Barang 9 | Barang 10 | ... | Barang 88 | Barang 89 | Barang 90 | Barang 91 | Barang 92 | Barang 93 | Barang 94 | Barang 95 | Barang 96 | Profit     |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 141.629097 |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 0.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 124.315293 |
| 1.0      | 1.0      | 1.0      | 1.0      | 0.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 0.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 112.927457 |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |
| ...      | ...      | ...      | ...      | ...      | ...      | ...      | ...      | ...      | ...       | ... | ...       | ...       | ...       | ...       | ...       | ...       | ...       | ...       | ...       | ...        |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |

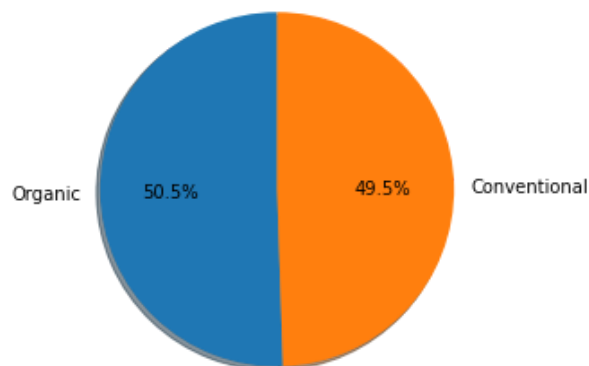
s x 97 columns

Dengan hal ini, maka apabila nilai limit yang diberikan semakin besar, maka profit yang dihasilkan akan semakin besar. Seperti pada percobaan 1 dan 2, dimana ketika nilai limit  $\leq 115$ , profit yang dihasilkan bernilai 0, atau tidak memiliki profit. Sedangkan ketika nilai limit yang diberikan  $\geq 120$ , maka profit yang dihasilkan akan ditampilkan, dan memiliki lebih banyak profit. Oleh karena itu, percobaan kedua adalah kemungkinan dari solusi terbaik.

Untuk bisa mendapatkan jenis alpukat yang paling banyak membawa profit, digunakan percobaan kedua selaku solusi terbaik. Diambil individu teratas dengan profit 141.629097 untuk menampilkan jenis alpukat yang paling banyak membawa profit. Maka, diperoleh ratio atau perbandingan antara alpukat organik berjumlah 48 dan alpukat konvensional berjumlah 47, berikut dengan persentase yang telah divisualisasikan ke dalam pie chart:

```
[] dfres = df[idx_d[:-1]].values
 dfres['Type'].value_counts()
```

```
Organic 48
Conventional 47
Name: Type, dtype: int64
```



## 2) Visualisasi hubungan antara parameter dengan solusi terbaik

Dalam hal ini, parameter yang diobservasi adalah parameter metode, jumlah individu dan maksimal iterasi. Digunakan parameter yang sama dengan parameter pada percobaan kedua, tetapi menggunakan maksimal iterasi dengan nilai 5, maka diperoleh hasil sebagai berikut:

```
[] hasil[4]
```

| Barang 1 | Barang 2 | Barang 3 | Barang 4 | Barang 5 | Barang 6 | Barang 7 | Barang 8 | Barang 9 | Barang 10 | ... | Barang 88 | Barang 89 | Barang 90 | Barang 91 | Barang 92 | Barang 93 | Barang 94 | Barang 95 | Barang 96 | Profit     |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 145.614129 |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |
| 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0      | 1.0       | ... | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 1.0       | 0.000000   |

s x 97 columns



Solusi terbaik:  $lim = 307$  dengan variasi jumlah individu = 100, 75, 50, 20, 10. Dari beberapa percobaan yang telah dilakukan, yaitu:

- Semakin besar maksimal iterasi, maka profit yang dihasilkan dari percobaan 2 sebagai solusi terbaik akan semakin sedikit atau semakin kecil
- Ketika maksimal iterasi diperkecil, maka profit yang dihasilkan dari percobaan 2 sebagai solusi terbaik akan semakin besar

### 3) Visualisasi solusi terhadap iterasi

Berdasarkan percobaan yang dilakukan, untuk mendapatkan solusi terbaik maka maksimal iterasi harus dibuat sekecil mungkin agar profit dapat ditampilkan dan hasil dari profit semakin banyak.

## CONCLUSION

Konsep dari metode knapsack adalah menentukan barang atau pilihan yang harus dimasukkan ke dalam knapsack tanpa melewati limit, tetapi memberi keuntungan maksimum sehingga mencapai hasil yang paling optimal. Pada percobaan analisis yang menggunakan data penjualan alpukat di wilayah California, dilakukan analisis dengan dua percobaan menggunakan parameter yang berbeda. Solusi terbaik didapatkan pada percobaan yang kedua. Maka, kategori jenis alpukat yang paling membawa banyak profit pun diperoleh dari percobaan kedua. Dari 96 data yang ada, diperoleh bahwa alpukat jenis organik membawa profit lebih banyak daripada alpukat jenis konvensional dengan perolehan alpukat organik sebanyak 48, sedangkan sisanya, alpukat konvensional sebanyak 47. Maka, dengan ini program penentuan jenis alpukat yang paling banyak membawa profit di wilayah California berhasil diterapkan dengan Algoritma Firefly Knapsack.

## REFERENCES

Anisa, A. (2016). *Penerapan Algoritma Firefly pada Permasalahan Knapsack 0-1*. Jember.

PipTools. (2015, Juli 8). *Algoritma FA (Firefly Algorithm)*. From PipTools:  
<https://piptools.net/algoritma-fa-firefly-algorithm/>

<https://www.kaggle.com/datasets/valentinjoseph/avocado-sales-20152021-us-centric>

## APPENDIX

1. [https://drive.google.com/file/d/1\\_BybNlvAplaVew995pAv67K6w\\_GfGi6p/view?usp=sharing](https://drive.google.com/file/d/1_BybNlvAplaVew995pAv67K6w_GfGi6p/view?usp=sharing)