

## BAB 4. HASIL DAN PEMBAHASAN

Bab hasil dan pembahasan pada penelitian ini berisi tentang penyelesaian permasalahan produksi di Edmee Outfit menggunakan *Firefly Algorithm* dengan bantuan *software* Phyton .

### 4.1 Data Penelitian

Pengambilan data dilakukan langsung melalui wawancara di toko Edmee Outfit yang berlokasi di Jl. Letjen Suprpto No 38-32, Dabasah, Kecamatan Bondowoso, Kabupaten Bondowoso. Data yang diperoleh merupakan data aktual dari periode produksi bulan Desember 2025. Data lengkap hasil wawancara tercantum pada Lampiran 1. Data tersebut meliputi jenis produk yang diproduksi, persediaan bahan baku kapasitas mesin, harga jual setiap jenis produk, biaya produksi, dan banyaknya produk yang terjual dalam 1 bulan seperti pada Tabel 4.1 sampai dengan Tabel 4.3.

Tabel 4.1 Ketersediaan Bahan Baku Produksi

| Bahan Baku       | Blouse | Tunik | Jaket | Sweater | Celana | Rok | One Set | Persediaan |
|------------------|--------|-------|-------|---------|--------|-----|---------|------------|
| Katun (yard)     | 1,5    | 0     | 0     | 0       | 0      | 0   | 0       | 2000       |
| Viscose (yard)   | 0      | 2     | 0     | 0       | 0      | 0   | 0       | 700        |
| Curduroy (yard)  | 0      | 0     | 1,7   | 0       | 0      | 0   | 0       | 500        |
| Knit (yard)      | 0      | 0     | 0     | 1,6     | 0      | 0   | 0       | 700        |
| Polyester (yard) | 0      | 0     | 0     | 0       | 1,7    | 1,7 | 0       | 1500       |
| Jaguar (yard)    | 0      | 0     | 0     | 0       | 0      | 0   | 3,5     | 1500       |
| Kancing (pcs)    | 5      | 5     | 0     | 5       | 1      | 0   | 4       | 10000      |
| Benang (cone)    | 0,1    | 0,125 | 0,16  | 0,125   | 0,1    | 0,1 | 0,2     | 500        |
| Zipper (pcs)     | 0      | 0     | 1     | 0       | 1      | 1   | 1       | 1500       |

Tabel 4.2 Kapasitas Mesin Produksi

| Mesin  | Durasi Proses per Mesin (jam/unit) |       |       |         |        |      |         | Kapasitas (jam) |
|--------|------------------------------------|-------|-------|---------|--------|------|---------|-----------------|
|        | Blouse                             | Tunik | Jaket | Sweater | Celana | Rok  | One Set |                 |
| Potong | 0,05                               | 0,05  | 0,05  | 0,05    | 0,05   | 0,05 | 0,1     | 328             |
| Jahit  | 0,3                                | 0,35  | 0,4   | 0,5     | 0,4    | 0,3  | 0,6     | 1147            |
| Obras  | 0,15                               | 0,18  | 0,2   | 0,2     | 0,15   | 0,15 | 0,4     | 655             |

Tabel 4.3 Data Produksi Edmee

| Produk  | Biaya Produksi | Harga Jual | Keuntungan | Banyak Terjual | Produksi Maksimum |
|---------|----------------|------------|------------|----------------|-------------------|
| Blouse  | Rp 75.000      | Rp 139.000 | Rp 64.000  | 700            | 3273              |
| Tunik   | Rp 105.000     | Rp 189.000 | Rp 84.000  | 200            | 2727              |
| Jaket   | Rp 105.000     | Rp 179.000 | Rp 74.000  | 150            | 2455              |
| Sweater | Rp 85.000      | Rp 149.000 | Rp 64.000  | 300            | 1966              |
| Celana  | Rp 95.000      | Rp 149.000 | Rp 54.000  | 200            | 2457              |
| Rok     | Rp 90.000      | Rp 139.000 | Rp 49.000  | 250            | 3273              |
| One set | Rp 145.000     | Rp 269.000 | Rp 124.000 | 300            | 1277              |

#### 4.2 Penerapan *Firefly Algorithm* pada Optimasi Produksi menggunakan Python

Implementasi *Firefly Algorithm* pada permasalahan optimasi produksi dilakukan menggunakan bahasa pemrograman Python. Sintak lengkap mengenai penyelesaian permasalahan produksi menggunakan *Firefly Algorithm* dapat dilihat pada Lampiran 2. Seluruh proses komputasi dibangun dalam satu kelas utama, yaitu `class FireflyOptimasiProduksi`, yang mengintegrasikan pembacaan data, evaluasi solusi, penanganan kendala, dan mekanisme iterasi *firefly*. Adapun *library* yang digunakan meliputi `numpy` untuk operasi numerik dan perhitungan vektor, `pandas` untuk membaca serta mengelola data dari excel, `random` untuk pembangkitan komponen acak pada proses eksplorasi, `matplotlib.pyplot` untuk visualisasi konvergensi dan distribusi produksi, serta `time` untuk menghitung durasi eksekusi algoritma. Kemudian tahapan penyelesaian *Firefly Algorithm* untuk optimasi produksi yaitu sebagai berikut.

- 1) Tahap awal dimulai dengan pendefinisian parameter produksi dan parameter algoritma. Input data produksi dilakukan menggunakan fungsi `self.baca_data(file_path)` untuk membaca data dari file Excel yang akan digunakan sebagai batasan kendala. Kemudian model juga memasukkan kendala modal maksimum sebesar Rp350.000.000 sebagai batasan biaya produksi. Selanjutnya metode `inisialisasi_parameter(self)` digunakan untuk mengatur parameter algoritma. Parameter yang digunakan pada penelitian ini ditetapkan berdasarkan kajian literatur dan hasil uji pendahulu,

yaitu  $n = 25$ ,  $\alpha = 0,1$ ,  $\gamma = 0,001$ ,  $\beta_0 = 1$  dan  $I_{max}$  sebanyak 100 iterasi.

Hasil output dari inisialisasi parameter tersebut sebagai berikut.

```
=====
Parameter Algoritma Firefly
=====
Jumlah firefly : 25
Alpha ( $\alpha$ ) : 0.1
Gamma ( $\gamma$ ) : 0.001
Beta0 ( $\beta_0$ ) : 1
Max iterasi : 100
```

- 2) Tahap kedua yaitu membangkitkan populasi awal sebanyak 25 solusi produksi menggunakan sintaks `random.randint(min_val,max_val)` untuk menghasilkan nilai acak yang berada dalam rentang  $[D_j, S_j]$  untuk setiap produk. Fungsi `perbaiki_solusi()` kemudian dipanggil untuk memastikan setiap solusi memenuhi semua kendala produksi sebelum dievaluasi.

```
INISIALISASI FIREFLY AWAL
-----
Firefly 1: blouse (700), tunik (346), jaket (282), sweater (340),
celana (255), rok (572), one_set (300)
Keuntungan: Rp195,490,000, Unit:2795, Status: ☒ FEASIBLE
-----
Firefly 2: blouse (700), tunik (330), jaket (269), sweater (372),
celana (288), rok (465), one_set (300)
Keuntungan: Rp191,771,000, Unit:2724, Status: ☒ FEASIBLE
```

- 3) Total keuntungan setiap solusi produksi dihitung menggunakan fungsi `hitung_keuntungan()` yang mengimplementasikan Persamaan (3.5). Nilai keuntungan terbesar disimpan sebagai solusi terbaik sementara. Berikut merupakan *output* dari pembangkitan populasi *firefly* dengan total keuntungan, biaya, dan keterangan memenuhi semua batasan kendala.

```
=====
🔗 BASELINE (Keuntungan terbesar disimpan sebagai solusi terbaik sementara)
=====
Firefly #21
Kombinasi produk: blouse (738), tunik (343), jaket (281), sweater (429),
celana (300), rok (464), one set (317)
Keuntungan Total: Rp 202,538,000
Total Unit : 2,872
=====
```

- 4) Pergerakan firefly dalam satu iterasi untuk setiap iterasi berada dalam sintak perulangan `for iterasi in range(self.max_iter):`. Apabila solusi produksi  $j$  memiliki keuntungan yang lebih besar dibandingkan solusi produksi  $i$ , maka solusi  $i$  akan bergerak menuju solusi  $j$ . Jarak antar dua solusi dihitung

menggunakan pendekatan jarak Euclidean, sedangkan nilai daya tarik dihitung menggunakan fungsi eksponensial yang dipengaruhi oleh parameter  $\gamma$  dan  $\beta_0$ . Perhitungan jarak dan daya tarik ini diimplementasikan menggunakan operasi numerik dari library `numpy`.

```
Firefly #1 → Firefly #3:
Fitness i: Rp 200,429,000
Fitness j: Rp 201,599,000
Posisi i : [722, 337, 293, 355, 275, 581, 308]
Posisi j : [721, 323, 288, 404, 322, 527, 306]
Perhitungan:
1. Jarak (r) = ||xi - xj|| = 88.0454
2.  $\beta = \beta_0 * \exp(-\gamma * r^2)$ 
   = 1.0000 * exp(-0.0010 * 7752.0000)
   = 0.000430
3. Komponen Daya Tarik:  $\beta * (x_j - x_i)$ 
   = 0.000430 * [-1.00, -14.00, -5.00, 49.00, 47.00, -54.00, 2.00]
   = [-0.0004, -0.0060, -0.0021, 0.0211, 0.0202, -0.0232, 0.0009]
4. Komponen Acak:  $\alpha * (\text{rand} - 0.5)$ 
    $\alpha$  saat ini : 0.1000
   Nilai (rand-0.5):
   [0.2353, 0.3584, 0.0646, 0.4236, -0.0807, -0.1283, -0.4318]
   Komponen acak :
   [6.0538, 9.0556, 1.4881, 7.0575, -1.8219, -3.8797, -4.2192]
```

- 5) Tahapan kelima adalah memperbarui posisi menuju solusi produksi yang memiliki keuntungan lebih besar dengan menambahkan elemen eksplorasi dalam proses pergerakannya. Kemudian dilakukan perbaikan solusi setelah pergerakan untuk semua solusi tetap berada dalam batas yang telah ditetapkan. Fungsi `perbaiki_solusi()` melakukan tiga langkah utama yaitu, pembulatan ke bilangan bulat, penyesuaian batas bawah dan batas atas serta penyesuaian terhadap kendala sumber daya.

```
5. Posisi Baru (float):
   =[728.0533, 346.0496, 294.4859, 362.0786, 273.1983,
     577.0971, 303.7800]
6. Setelah Repair dan Rounding:
   Posisi Baru : [728, 346, 294, 362, 273, 577, 304]
   Fitness Baru: Rp 201,291,000
   Fitness Lama: Rp 200,429,000
    PENINGKATAN: Rp 862,000
```

- 6) Setelah seluruh solusi diperbarui dan diperiksa, nilai total keuntungan dihitung kembali. Kemudian untuk menemukan indeks firefly dengan fitness tertinggi digunakan sintaks `np.argmax(fitness_values)`. Solusi ini kemudian

disimpan sebagai acuan untuk iterasi berikutnya. Berikut ditampilkan contoh *output* dari proses iterasi tersebut.

```

►ITERASI 9:
Best firefly: #19
Keuntungan: Rp 204,562,000
Tidak ada peningkatan
Iterasi tanpa perbaikan: 1
=====
►ITERASI 10:
Best firefly: #19
Keuntungan: Rp 204,562,000
🌟PENINGKATAN di Iterasi 10: 🌟
Firefly #24
Kombinasi produk:
blouse(818), tunik(343), jaket(290), sweater(352), celana(271),
rok(569), one_set(303)
Keuntungan Total: Rp 205,239,000
Selisih peningkatan: Rp +677,000
Total Unit      : 2,946
Status: ☒ SEMUA KENDALA TERPENUHI
=====

```

- 7) Proses perbandingan, pembaruan posisi, perbaikan kendala, dan evaluasi keuntungan ini diulangi hingga mencapai batas maksimum iterasi yaitu 100. Pada akhir proses iterasi, kombinasi jumlah produksi dengan total keuntungan terbesar yang diperoleh selama seluruh iterasi ditetapkan sebagai solusi optimal akhir. Hasil optimasi ditampilkan dalam bentuk rincian jumlah produksi tiap produk, total keuntungan, serta divisualisasikan dalam grafik konvergensi. Hasil akhir dari proses komputasi ditampilkan sebagai berikut.

```

HASIL OPTIMASI PRODUKSI
-----
Firefly ke-21
Konvergen di Iterasi: 97
Waktu: 6.41 detik
SOLUSI OPTIMAL:
blouse(955), tunik(350), jaket(279), sweater(347), celana(300),
rok(416), one_set(346)
Keuntungan Optimal : Rp 212,862,000
Biaya Produksi     : Rp 283,275,000
Unit yang diproduksi: 2,993

```

Hasil optimasi menunjukkan bahwa solusi terbaik diperoleh pada iterasi ke-97 dalam waktu 6,41 detik. Solusi akhir memberikan keuntungan sebesar Rp 212.862.000 dengan total unit yang diproduksi yaitu 2993 item. Berikut rincian rencana produksi optimal disajikan dalam Tabel 4.4 dan Tabel 4.5.

Tabel 4.4 Rencana Produksi Optimal

| No    | Produk  | Jumlah Produksi | Keuntungan/item | Subtotal Keuntungan |
|-------|---------|-----------------|-----------------|---------------------|
| 1     | Blouse  | 955             | Rp 64.000       | Rp 61.120.000       |
| 2     | Tunik   | 350             | Rp 84.000       | Rp 29.400.000       |
| 3     | Jaket   | 279             | Rp 74.000       | Rp 20.646.000       |
| 4     | Sweater | 347             | Rp 64.000       | Rp 22.208.000       |
| 5     | Celana  | 300             | Rp 54.000       | Rp 16.200.000       |
| 6     | Rok     | 416             | Rp 49.000       | Rp 20.384.000       |
| 7     | One set | 346             | Rp 124.000      | Rp 42.904.000       |
| Total |         | 2993            |                 | Rp 212.862.000      |

Tabel 4.5 Pemakaian Sumber Daya

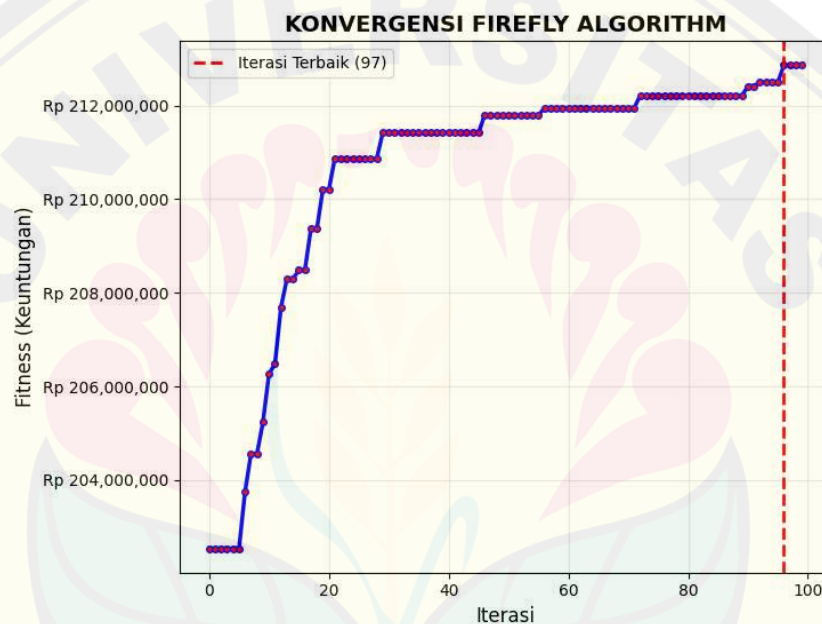
| No | Sumber Daya            | Terpakai       | Tersedia       | Persentase Pemakaian |
|----|------------------------|----------------|----------------|----------------------|
| 1  | Katun                  | 1432,5 yard    | 2000 yard      | 71,6%                |
| 2  | Viscose                | 700 yard       | 700 yard       | 100%                 |
| 3  | Curduroy               | 474,3 yard     | 500 yard       | 94,9%                |
| 4  | Knit                   | 555,2 yard     | 700 yard       | 79,3%                |
| 5  | Polyester              | 1217,2 yard    | 1500 yard      | 81,1%                |
| 6  | Jaguar                 | 1211,0 yard    | 1500 yard      | 80,7%                |
| 7  | Kancing                | 9944 pcs       | 10000 pcs      | 99,4%                |
| 8  | Benang                 | 370,8 cone     | 500 cone       | 74,2%                |
| 9  | Zipper                 | 1341 pcs       | 1500 pcs       | 89,4%                |
| 10 | Kapasitas mesin potong | 167 jam        | 328 jam        | 50,9%                |
| 11 | Kapasitas mesin jahit  | 1146,5 jam     | 1147 jam       | 99,9%                |
| 12 | Kapasitas mesin obras  | 577,2 jam      | 655 jam        | 88,1%                |
| 13 | Biaya produksi         | Rp 283.275.000 | Rp 350.000.000 | 80,9%                |

Berdasarkan Tabel 4.4, tingkat pemanfaatan sumber daya produksi menunjukkan bahwa solusi yang dihasilkan oleh algoritma mampu memanfaatkan sumber daya secara optimal tanpa melampaui batas ketersediaan yang telah ditetapkan. Pemakaian bahan baku berada pada kisaran 71,6% hingga 100% dari total persediaan. Dari sisi penggunaan mesin, tingkat pemakaian mesin potong tercatat sebesar 50,9%, sedangkan mesin jahit mencapai 99,9%, sehingga mesin jahit dapat diidentifikasi sebagai kendala aktif (*binding constraint*) dalam sistem produksi. Selain itu, penggunaan biaya produksi tercatat sebesar Rp283.275.000 dari batas maksimum Rp350.000.000 atau sebesar 80,9%, yang menunjukkan bahwa alokasi modal masih berada di batas kendala biaya yang ditetapkan. Hasil



ini menunjukkan bahwa proses produksi telah memanfaatkan sumber daya secara optimal tanpa melampaui kapasitas yang tersedia.

Setiap hasil yang diperoleh dari tiap iterasi kemudian ditampilkan dalam bentuk grafik. Grafik konvergensi menunjukkan pola peningkatan keuntungan secara bertahap, dimana pada iterasi awal awal, nilai keuntungan meningkat tajam karena algoritma aktif menjelajahi ruang solusi. Kemudian peningkatan berlangsung lebih stabil seiring penyempurnaan solusi. Visualisasi grafik yang merepresentasikan iterasi ke-1 sampai dengan iterasi ke-100 ditunjukkan pada Gambar 4.1.



Gambar 4.1 Visualisasi Grafik *Firefly Algorithm*

### 4.3 Uji Parameter

Pada bagian ini dilakukan pengujian variasi parameter yang digunakan untuk mengevaluasi performa *Firefly Algorithm* dalam menyelesaikan permasalahan optimasi produksi. Parameter yang dianalisis meliputi ukuran populasi awal ( $n$ ),  $\alpha$ ,  $\gamma$ ,  $\beta_0$ , serta jumlah iterasi maksimum ( $Imax$ ). Evaluasi terhadap setiap konfigurasi parameter dilakukan melalui lima kali proses *running* untuk mengamati kestabilan solusi dan kecenderungan nilai fitness yang dihasilkan. Nilai hasil tertentu kemudian digunakan sebagai acuan dalam perbandingan antar konfigurasi parameter, dengan mempertimbangkan kestabilan solusi dan capaian nilai fitness.

Melalui pengujian ini, diharapkan dapat diperoleh konfigurasi parameter yang paling optimal dalam menyelesaikan permasalahan optimasi produksi sehingga menghasilkan keuntungan maksimal dan performa algoritma yang optimal. Rekapitulasi seluruh hasil pengujian parameter dapat dilihat pada Lampiran 3.

### 1. Uji Ukuran Populasi Awal *Firefly* ( $n$ )

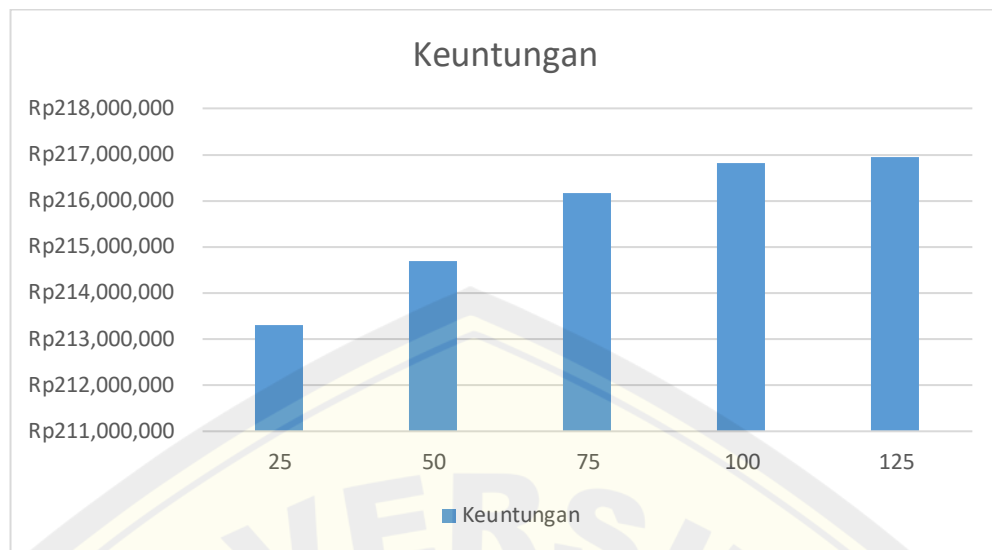
Uji ukuran populasi awal *Firefly* ( $n$ ) dilakukan untuk mengevaluasi pengaruh jumlah individu dalam populasi terhadap kualitas solusi yang dihasilkan, kecepatan konvergensi, serta waktu komputasi. Pada pengujian ini, parameter algoritma lainnya ditetapkan secara konstan pada nilai yang telah digunakan sebelumnya, sedangkan nilai  $n$  divariasikan dari 25 hingga 125. Hasil yang diperoleh dari uji parameter  $n$  dapat dilihat pada Tabel 4.6.

Tabel 4.6 Uji Ukuran Populasi ( $n$ )

| $n$ | $\alpha$ | $\gamma$ | $\beta_0$ | $l_{max}$ | Keuntungan    | Konvergen | Waktu (detik) | Running |
|-----|----------|----------|-----------|-----------|---------------|-----------|---------------|---------|
| 25  | 0,1      | 0,001    | 1         | 100       | Rp213.293.000 | 95        | 7.79          | 4       |
| 50  | 0,1      | 0,001    | 1         | 100       | Rp214.711.000 | 97        | 28.63         | 2       |
| 75  | 0,1      | 0,001    | 1         | 100       | Rp216.174.000 | 97        | 57.05         | 4       |
| 100 | 0,1      | 0,001    | 1         | 100       | Rp216.820.000 | 96        | 117.84        | 3       |
| 125 | 0,1      | 0,001    | 1         | 100       | Rp216.929.000 | 94        | 185.19        | 1       |

Berdasarkan hasil pengujian, terlihat bahwa populasi yang lebih besar mampu menyediakan keragaman solusi awal yang lebih baik, sehingga peluang menemukan solusi dengan keuntungan lebih tinggi menjadi semakin besar. Pada nilai  $n$  yang lebih kecil, keuntungan yang dihasilkan masih relatif rendah. Seiring dengan bertambahnya jumlah populasi hingga  $n = 100$  dan  $n = 125$ , keuntungan meningkat secara bertahap dan mencapai nilai tertinggi pada  $n = 125$ , yaitu sebesar Rp216.929.000. Seiring bertambahnya nilai  $n$ , waktu eksekusi juga meningkat, namun tambahan waktu tersebut masih relatif kecil sehingga tidak memengaruhi kinerja algoritma secara keseluruhan. Visualisasi pengaruh ukuran populasi terhadap keuntungan dapat dilihat pada Gambar 4.2



Gambar 4.2 Grafik Uji Ukuran Populasi ( $n$ )

## 2. Uji Parameter Koefisien Acak ( $\alpha$ )

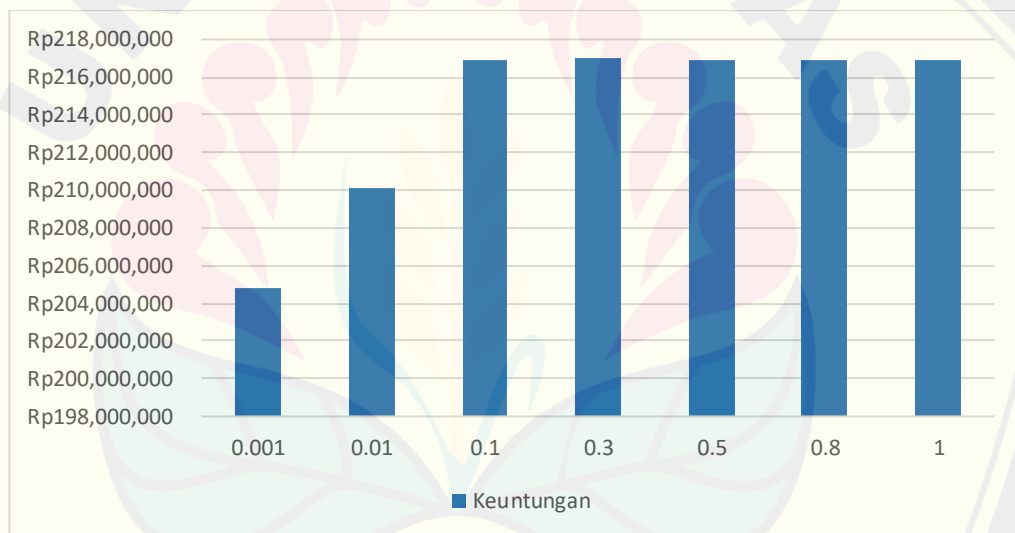
Pengujian selanjutnya yaitu parameter koefisien acak dilakukan dengan memvariasikan nilai  $\alpha$  dari 0,001 hingga 1. Pengujian dilakukan dengan mempertahankan jumlah populasi  $n = 125$  yang telah didapat dari hasil uji parameter sebelumnya. Kemudian untuk nilai  $\beta_0$ ,  $\gamma$  dan  $I_{max}$  ditetapkan konstan sama seperti sebelumnya. Hasil yang diperoleh dari uji parameter  $\alpha$  dapat dilihat pada Tabel 4.7.

Tabel 4.7 Uji Parameter Koefisien Acak ( $\alpha$ )

| $n$ | $\alpha$ | $\gamma$ | $\beta_0$ | $I_{max}$ | Keuntungan    | Konvergen | Waktu (detik) | Running |
|-----|----------|----------|-----------|-----------|---------------|-----------|---------------|---------|
| 125 | 0,001    | 0,001    | 1         | 100       | Rp204.848.000 | 1         | 126.18        | 3       |
| 125 | 0,01     | 0,001    | 1         | 100       | Rp210.116.000 | 88        | 159.56        | 5       |
| 125 | 0,1      | 0,001    | 1         | 100       | Rp216.943.000 | 91        | 170.61        | 5       |
| 125 | 0,3      | 0,001    | 1         | 100       | Rp216.985.000 | 63        | 178.37        | 4       |
| 125 | 0,5      | 0,001    | 1         | 100       | Rp216.868.000 | 41        | 181.79        | 4       |
| 125 | 0,8      | 0,001    | 1         | 100       | Rp216.874.000 | 80        | 193.10        | 1       |
| 125 | 1        | 0,001    | 1         | 100       | Rp216.864.000 | 73        | 204.77        | 3       |

Berdasarkan tabel hasil uji parameter  $\alpha$ , pengujian menunjukkan bahwa variasi nilai  $\alpha$  memberikan pengaruh yang jelas terhadap keuntungan yang dihasilkan oleh algoritma. Pada nilai  $\alpha$  yang kecil, keuntungan masih relatif rendah meskipun waktu eksekusi lebih singkat, yang mengindikasikan bahwa proses

pencarian solusi belum optimal. Ketika nilai  $\alpha$  berada pada rentang menengah, keuntungan yang diperoleh meningkat dan mulai stabil. Pada rentang ini, algoritma mampu menyeimbangkan proses eksplorasi dan eksploitasi dengan baik, sehingga solusi yang diperoleh memiliki kualitas yang lebih tinggi. Nilai keuntungan tertinggi dicapai pada  $\alpha = 0,3$ , yaitu sebesar Rp216.985.000 dengan konvergensi yang relatif cepat dibandingkan nilai  $\alpha$  lainnya. Pada nilai  $\alpha$  yang lebih besar, yaitu mendekati 1, keuntungan yang dihasilkan cenderung stabil namun tidak menunjukkan peningkatan yang signifikan. Hal ini mengindikasikan bahwa tingkat keacakan yang terlalu tinggi dapat menyebabkan pergerakan solusi menjadi kurang terarah, sehingga peningkatan kualitas solusi menjadi terbatas. Visualisasi pengaruh variasi nilai  $\alpha$  terhadap keuntungan dapat dilihat pada Gambar 4.3



Gambar 4.3 Grafik Uji Parameter  $\alpha$

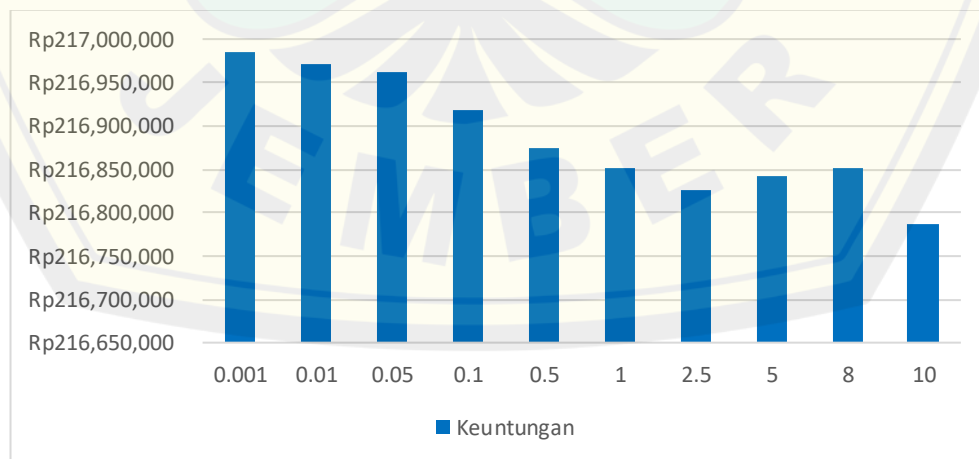
### 3. Uji Parameter Koefisien Penyerapan Cahaya ( $\gamma$ )

Pengujian parameter  $\gamma$  dilakukan setelah diperoleh nilai parameter terbaik dari pengujian sebelumnya, yaitu  $n = 125$  dan  $\alpha = 0,3$ , yang kemudian ditetapkan sebagai nilai tetap, sementara untuk nilai  $\beta_0$  dan  $I_{\max}$  ditetapkan konstan sama seperti sebelumnya. Selanjutnya, nilai  $\gamma$  divariasikan pada rentang 0,001 hingga 10 untuk menganalisis pengaruhnya terhadap kinerja algoritma. Hasil yang diperoleh dari uji parameter  $\gamma$  dapat dilihat pada Tabel 4.8.

Tabel 4.8 Uji Parameter Koefisien Penyerapan Cahaya ( $\gamma$ )

| $n$ | $\alpha$ | $\gamma$ | $\beta_0$ | $I_{max}$ | Keuntungan    | Konvergen | Waktu (detik) | Running |
|-----|----------|----------|-----------|-----------|---------------|-----------|---------------|---------|
| 125 | 0,3      | 0,001    | 1         | 100       | Rp216.985.000 | 54        | 173.54        | 2       |
| 125 | 0,3      | 0,01     | 1         | 100       | Rp216.971.000 | 94        | 174.84        | 5       |
| 125 | 0,3      | 0,05     | 1         | 100       | Rp216.962.000 | 75        | 175.76        | 1       |
| 125 | 0,3      | 0,1      | 1         | 100       | Rp216.918.000 | 55        | 176.93        | 1       |
| 125 | 0,3      | 0,5      | 1         | 100       | Rp216.875.000 | 76        | 174.30        | 3       |
| 125 | 0,3      | 1        | 1         | 100       | Rp216.852.000 | 68        | 174.40        | 5       |
| 125 | 0,3      | 2,5      | 1         | 100       | Rp216.826.000 | 64        | 176.70        | 4       |
| 125 | 0,3      | 5        | 1         | 100       | Rp216.843.000 | 99        | 179.76        | 3       |
| 125 | 0,3      | 7,5      | 1         | 100       | Rp216.852.000 | 77        | 175.39        | 5       |
| 125 | 0,3      | 10       | 1         | 100       | Rp216.786.000 | 77        | 180.32        | 3       |

Hasil pengujian menunjukkan bahwa keuntungan yang dihasilkan relatif stabil pada berbagai nilai  $\gamma$ , dengan nilai tertinggi diperoleh pada  $\gamma = 0,001$  sebesar Rp216.985.000. Pada nilai  $\gamma$  yang semakin besar, keuntungan cenderung mengalami penurunan, meskipun perbedaannya tidak terlalu signifikan. Hasil ini menunjukkan bahwa nilai  $\gamma$  yang kecil memberikan kinerja yang lebih baik karena nilai  $\gamma$  yang kecil mampu mempertahankan pengaruh daya tarik antar solusi, sehingga proses pencarian bisa tetap terarah dan tidak mudah kehilangan potensi solusi yang baik. Visualisasi pengaruh variasi nilai  $\gamma$  terhadap keuntungan dapat dilihat pada Gambar 4.4

Gambar 4.4 Grafik Uji Parameter  $\gamma$

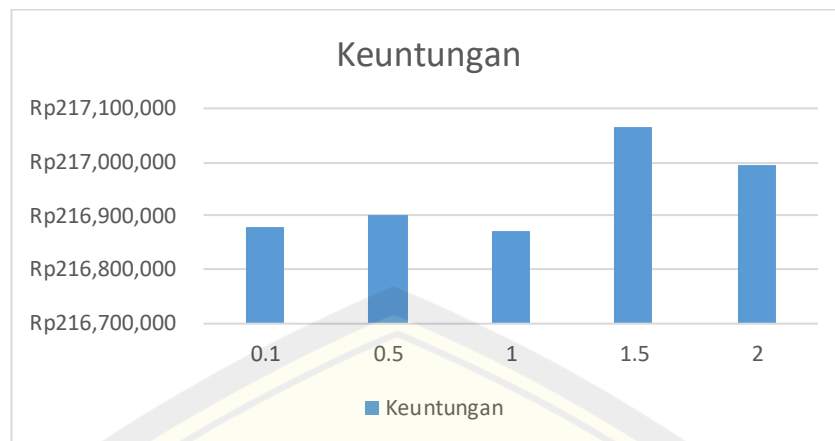
#### 4. Uji Parameter Konstanta Keatraktifan ( $\beta_0$ )

Pengujian konstanta keatraktifan  $\beta_0$  dilakukan sebagai tahap lanjutan setelah didapatkan parameter terbaik dari pengujian sebelumnya, yaitu  $n = 125$ ,  $\alpha = 0,3$ , dan  $\gamma = 0,001$  yang kemudian ditetapkan sebagai nilai tetap, sementara untuk nilai  $I_{max}$  ditetapkan konstan sama seperti sebelumnya. Pada pengujian ini,  $\beta_0$  diuji pada beberapa nilai dalam rentang 0,1 hingga 2. Hasil yang diperoleh dari uji parameter  $\beta_0$  dapat dilihat pada Tabel 4.9.

Tabel 4.9 Uji Parameter Konstanta Keatraktifan ( $\beta_0$ )

| $n$ | $\alpha$ | $\gamma$ | $\beta_0$ | $I_{max}$ | Keuntungan    | Konvergen | Waktu (detik) | Running |
|-----|----------|----------|-----------|-----------|---------------|-----------|---------------|---------|
| 125 | 0,3      | 0,001    | 0,1       | 100       | Rp216.881.000 | 85        | 118.65        | 5       |
| 125 | 0,3      | 0,001    | 0,5       | 100       | Rp216.900.000 | 76        | 120.12        | 3       |
| 125 | 0,3      | 0,001    | 1         | 100       | Rp216.871.000 | 93        | 118.41        | 5       |
| 125 | 0,3      | 0,001    | 1,5       | 100       | Rp217.065.000 | 82        | 119.53        | 1       |
| 125 | 0,3      | 0,001    | 2         | 100       | Rp216.994.000 | 57        | 118.17        | 2       |

Berdasarkan hasil uji parameter konstanta keatraktifan, variasi nilai  $\beta_0$  memberikan pengaruh terhadap kinerja algoritma. Keuntungan tertinggi diperoleh pada  $\beta_0 = 1,5$ , yaitu sebesar Rp217.065.000, yang menunjukkan bahwa nilai  $\beta_0$  pada tingkat menengah-tinggi mampu menyeimbangkan proses eksplorasi dan eksploitasi secara optimal. Di sisi lain,  $\beta_0 = 2$  menghasilkan konvergensi paling cepat dengan waktu komputasi yang relatif singkat, meskipun keuntungan yang diperoleh sedikit lebih rendah. Pada nilai  $\beta_0$  yang lebih kecil, keuntungan yang dihasilkan cenderung stabil dengan kecepatan konvergensi yang sedang. Oleh karena itu,  $\beta_0 = 1,5$  dipilih sebagai nilai parameter yang paling optimal untuk menghasilkan kualitas solusi terbaik pada penelitian ini. Visualisasi pengaruh variasi nilai  $\beta_0$  terhadap keuntungan dapat dilihat pada Gambar 4.5.

Gambar 4.5 Grafik Uji Parameter  $\beta_0$ 

### 5. Uji Jumlah Iterasi Maksimum ( $I_{max}$ )

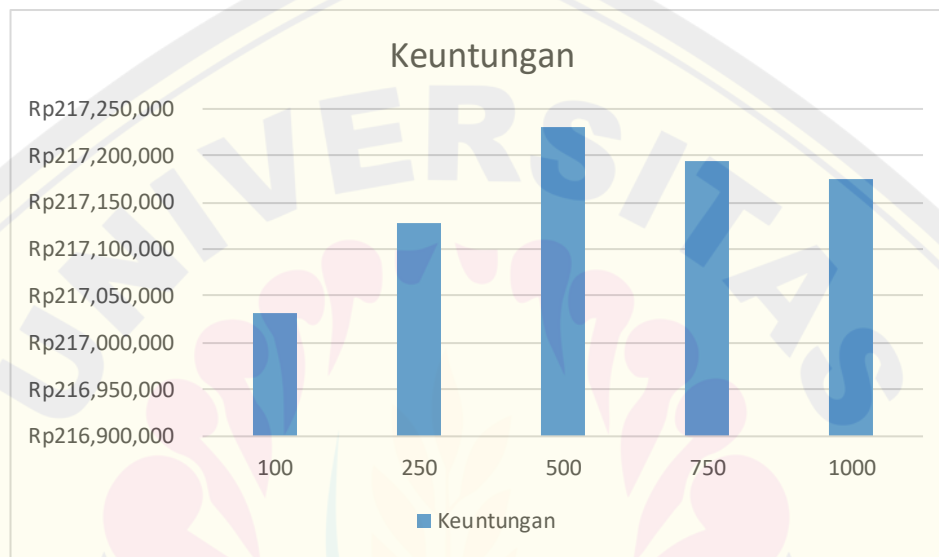
Pengujian jumlah iterasi maksimum ( $I_{max}$ ) dilakukan untuk melihat sejauh mana batas iterasi memengaruhi kemampuan algoritma dalam menemukan solusi terbaik, sekaligus dampaknya terhadap kecepatan konvergensi dan waktu komputasi. Pada pengujian ini, nilai  $I_{max}$  diuji pada beberapa tingkat, yaitu 100, 250, 500, 750, dan 1000, sedangkan parameter lain dipertahankan pada konfigurasi terbaik yang telah diperoleh sebelumnya. Hasil yang diperoleh dari uji jumlah iterasi maksimum dapat dilihat pada Tabel 4.10.

Tabel 4.10 Uji Jumlah Iterasi Maksimum

| $n$ | $\alpha$ | $\gamma$ | $\beta_0$ | $I_{max}$ | Keuntungan    | Konvergen | Waktu (detik) | Running |
|-----|----------|----------|-----------|-----------|---------------|-----------|---------------|---------|
| 125 | 0,3      | 0,001    | 1,5       | 100       | Rp217.031.000 | 75        | 175.08        | 5       |
| 125 | 0,3      | 0,001    | 1,5       | 250       | Rp217.127.000 | 227       | 434.50        | 3       |
| 125 | 0,3      | 0,001    | 1,5       | 500       | Rp217.230.000 | 489       | 899.50        | 2       |
| 125 | 0,3      | 0,001    | 1,5       | 750       | Rp217.195.000 | 331       | 1307.45       | 4       |
| 125 | 0,3      | 0,001    | 1,5       | 1000      | Rp217.174.000 | 495       | 1812.28       | 3       |

Hasil pengujian menunjukkan bahwa peningkatan nilai  $I_{max}$  tidak selalu meningkatkan keuntungan yang diperoleh. Pada  $I_{max} = 100$ , keuntungan yang dihasilkan masih relatif lebih rendah karena proses pencarian solusi berhenti lebih cepat, sehingga algoritma belum sepenuhnya mengeksplorasi ruang solusi. Seiring dengan peningkatan nilai  $I_{max}$  menjadi 250 dan 500, keuntungan yang diperoleh mengalami peningkatan dan mencapai nilai tertinggi pada  $I_{max} = 500$ , yaitu

sebesar Rp217.230.000. Hal ini menunjukkan bahwa penambahan jumlah iterasi memberikan kesempatan yang lebih besar bagi algoritma untuk memperbaiki solusi dan mencapai hasil yang lebih optimal. Pada nilai  $I_{max}$  yang lebih besar, yaitu 750 dan 1000, keuntungan yang dihasilkan cenderung stabil dan tidak menunjukkan peningkatan yang signifikan, meskipun jumlah iterasi dan waktu komputasi bertambah. Visualisasi pengaruh ukuran populasi terhadap keuntungan dapat dilihat pada Gambar 4.6.



Gambar 4.6 Grafik Uji Jumlah Iterasi Maksimum

#### 4.4 Analisis Hasil

Berdasarkan keseluruhan tahapan penelitian yang telah dilaksanakan, dapat diketahui bahwa penyelesaian permasalahan optimasi produksi pakaian menggunakan *Firefly Algorithm* menunjukkan kinerja yang efektif dalam menghasilkan solusi yang optimal. Evaluasi terhadap hasil sebelum dan sesudah proses uji parameter, menunjukkan adanya peningkatan kualitas solusi yang cukup jelas. Hal ini menunjukkan bahwa proses penyesuaian parameter memiliki pengaruh yang signifikan terhadap performa algoritma. Perbandingan hasil optimasi yang diperoleh berdasarkan literatur dengan setelah uji parameter selanjutnya dirangkum dan disajikan pada Tabel 4.11.



Tabel 4.11 Analisis Hasil

| Parameter                 | Berdasarkan Literatur | Berdasarkan Uji Parameter |
|---------------------------|-----------------------|---------------------------|
| $n$                       | 25                    | 125                       |
| $\alpha$                  | 0,1                   | 0,3                       |
| $\gamma$                  | 0.001                 | 0.001                     |
| $\beta_0$                 | 1                     | 1,5                       |
| $Imax$                    | 100                   | 500                       |
| Keuntungan maksimum       | Rp 212.862.000        | Rp 217.230.000            |
| Konvergen                 | Iterasi ke-97         | Iterasi ke-489            |
| Waktu Eksekusi            | 6,41 detik            | 899,50 detik              |
| Jumlah Produksi           | 2993                  | 2975                      |
| Pemakaian Bahan Baku      | 71,6% - 100%          | 68,6% - 99,9%             |
| Penggunaan Mesin          | 50.9% - 99,9%         | 51,8% - 100%              |
| Penggunaan Biaya Produksi | Rp 283.275.000        | Rp 285.655.000            |

Berdasarkan rangkaian uji parameter yang telah dilakukan, parameter koefisien penyerapan cahaya ( $\gamma$ ) merupakan parameter yang paling berpengaruh terhadap kinerja *Firefly Algorithm* dalam menghasilkan solusi optimal. Hasil pengujian menunjukkan bahwa perubahan nilai  $\gamma$ , meskipun tidak menghasilkan perbedaan keuntungan yang sangat besar, secara konsisten memengaruhi arah proses pencarian solusi. Nilai  $\gamma$  yang kecil ( $\gamma = 0,001$ ) mampu mempertahankan daya tarik antar solusi meskipun berada pada jarak yang relatif jauh, sehingga pencarian solusi menjadi lebih terarah dan stabil. Sementara itu, ukuran populasi awal ( $n$ ) dan jumlah iterasi maksimum ( $Imax$ ) berperan penting dalam meningkatkan keragaman solusi awal serta memperluas proses penyempurnaan solusi selama iterasi. Peningkatan nilai  $n$  dan  $Imax$  cenderung meningkatkan keuntungan hingga titik tertentu, tetapi tidak mengubah pola pencarian secara fundamental. Adapun parameter koefisien acak ( $\alpha$ ) dan konstanta keatraktifan ( $\beta_0$ ) berfungsi sebagai pengatur keseimbangan antara eksplorasi dan eksploitasi, di mana nilai menengah menghasilkan kinerja terbaik. Hal tersebut menunjukkan bahwa penyesuaian parameter sesuai dengan karakteristik permasalahan akan mampu meningkatkan performa algoritma secara lebih optimal.

Hasil dari uji parameter algoritma tersebut menunjukkan bahwa kombinasi  $n = 125$ ,  $\alpha = 0,3$ ,  $\gamma = 0,001$ ,  $\beta_0 = 1,5$ , dan  $Imax = 500$  merupakan kombinasi optimal yang menghasilkan keuntungan tertinggi sebesar Rp217.230.000 dengan

total unit yang diproduksi yaitu 2975 unit. Berikut rincian rencana produksi optimal disajikan dalam Tabel 4.12 dan Tabel 4.13.

Tabel 4.12 Rencana Produksi Optimal Setelah Uji Parameter

| No    | Produk         | Jumlah Produksi | Keuntungan/item | Subtotal Keuntungan |
|-------|----------------|-----------------|-----------------|---------------------|
| 1     | Blouse         | 973             | Rp 64.000       | Rp 62.272.000       |
| 2     | Tunik          | 346             | Rp 84.000       | Rp 29.064.000       |
| 3     | Jaket          | 292             | Rp 74.000       | Rp 21.608.000       |
| 4     | Sweater        | 300             | Rp 64.000       | Rp 19.200.000       |
| 5     | Celana         | 200             | Rp 54.000       | Rp 10.800.000       |
| 6     | Rok            | 438             | Rp 49.000       | Rp 21.462.000       |
| 7     | <i>One set</i> | 426             | Rp 124.000      | Rp 52.824.000       |
| Total |                | 2975            |                 | Rp 217.230.000      |

Tabel 4.13 Pemakaian Sumber Daya Setelah Uji Parameter

| No | Sumber Daya            | Terpakai       | Tersedia       | Persentase Pemakaian |
|----|------------------------|----------------|----------------|----------------------|
| 1  | Katun                  | 1459,5 yard    | 2000 yard      | 73%                  |
| 2  | Viscose                | 692 yard       | 700 yard       | 98,9%                |
| 3  | Curduroy               | 496,4 yard     | 500 yard       | 99,3%                |
| 4  | Knit                   | 480 yard       | 700 yard       | 68,6%                |
| 5  | Polyester              | 1084,6 yard    | 1500 yard      | 72,3%                |
| 6  | Jaguar                 | 1491 yard      | 1500 yard      | 99,4%                |
| 7  | Kancing                | 9999 pcs       | 10000 pcs      | 99,9%                |
| 8  | Benang                 | 373,6 cone     | 500 cone       | 74,7%                |
| 9  | Zipper                 | 1356 pcs       | 1500 pcs       | 90,4%                |
| 10 | Kapasitas mesin potong | 170,1 jam      | 328 jam        | 51,8%                |
| 11 | Kapasitas mesin jahit  | 1146,8 jam     | 1147 jam       | 99,9%                |
| 12 | Kapasitas mesin obras  | 592,7 jam      | 655 jam        | 90,5%                |
| 13 | Biaya produksi         | Rp 285.655.000 | Rp 350.000.000 | 81,2%                |