

Sistem Penjadwalan Perangkat Listrik dengan Metode Algoritma *Particle Swarm Optimization* Berbasis Website

1st Muhammad Raihan Ramadhan
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
mraihanramadhan@student.telkomuniversity.ac.id

2nd Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
setiacasie@telkomuniversity.ac.id

3rd Burhanuddin Dirgantoro
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
burhanuddin@telkomuniversity.ac.id

Abstrak—Listrik dapat mempermudah aktivitas masyarakat, namun jika tidak digunakan dengan hati-hati dapat menimbulkan kerugian. Masyarakat harus lebih memperhatikan pasokan listrik dan kebutuhan listrik dengan cara menghemat penggunaan listrik. Di lingkungan masyarakat listrik merupakan salah satu konsumen terbesar dan juga permasalahan terbesar, penggunaan listrik ini menggambarkan aktivitas masyarakat di tempat tinggal. Pada tugas akhir ini membahas tentang sistem penjadwalan perangkat listrik. Penjadwalan perangkat listrik dilakukan dengan cara melakukan optimasi perangkat listrik dalam satu ruangan. Dengan adanya optimasi dapat membuat waktu penggunaan perangkat listrik yang dipakai berkurang. Dengan adanya penjadwalan dapat membatasi waktu penggunaan perangkat listrik. Dalam sistem penjadwalan perangkat listrik, tingkat efisiensi sebuah algoritma optimasi dalam ditentukan dengan menggunakan metode Algoritma *Particle Swarm Optimization* dan sebagai tempat penyimpanan data pengguna digunakan *database*. Nilai fitness terbaik terdapat pada iterasi ke-2 dengan nilai 88.545,663 dimana nilai ini sudah *convergen*, cara memperoleh nilai tersebut dengan cara menguji nilai *fitness* terbaik yang didapatkan dari nilai *fitness* sebelumnya. Pada saat melakukan pengujian fungsionalitas web (pengujian alpha) didapatkan nilai sebesar 100% yang dimana menunjukkan sistem ini berjalan sesuai aturan yang sudah dirancang dengan baik.

Keywords-- algoritma *particle swarm optimization*, perangkat listrik, penjadwalan, *website*.

I. PENDAHULUAN

Listrik dapat mempermudah aktivitas masyarakat, namun jika tidak digunakan dengan hati-hati dapat menimbulkan kerugian. Pemborosan listrik oleh pelanggan bisa disebabkan karena pelanggan tidak memahami konsep penghematan listrik untuk bertahan hidup, seperti permintaan untuk menggunakan energi listrik yang terus meningkat sehingga masyarakat dapat melakukan segala aktivitas dengan baik [1].

Masyarakat harus lebih memperhatikan pasokan listrik dan kebutuhan listrik dengan cara

menghemat penggunaan listrik. Tujuan dari penghematan penggunaan listrik untuk membantu pemerintah mewujudkan kestabilan energi yang lebih baik lagi. Di lingkungan masyarakat listrik merupakan salah satu konsumen terbesar dan juga permasalahan terbesar, penggunaan listrik ini menggambarkan aktivitas masyarakat di tempat tinggal. Namun banyak juga masyarakat yang memaksimalkan penggunaan listrik, seperti masyarakat yang harus kerja dari matahari terbit sampai matahari terbenam, kebiasaan masyarakat seperti itu tidak berdampak besar terhadap penggunaan listrik karena kegiatan yang dilakukan tidak dirumah [2][3].

Penelitian tentang penjadwalan listrik sudah banyak dilakukan, diantaranya "*Particle Swarm Optimization Algorithm for Power Scheduling Problem Using Smart Battery*", adalah menjadwalkan peralatan rumah pintar dengan periode waktu yang tepat, tujuannya untuk mengurangi tagihan listrik dan rasio *Peak-to-average* (PAR) dan meningkatkan tingkat kenyamanan pengguna. Untuk mengoptimalkan penjadwalan, Dengan menggunakan metode Algoritma *Particle Swarm Optimization* (PSO) [4].

Dengan adanya permasalahan ini, maka diperlukan solusi yang dapat mengatasi permasalahan tersebut yaitu diperlukan suatu alat atau sistem yang dapat digunakan secara otomatis untuk mengatur penggunaan listrik yang berlebihan dengan cara menyalakan atau mematikan listrik yang tidak digunakan sehingga agar penggunaan listrik dapat diatur dan terjadi penghematan dalam penggunaan listrik. Oleh karena itu, penelitian berjudul Sistem Penjadwalan Perangkat Listrik dengan Metode Algoritma *Particle Swarm Optimization* Berbasis Website diharapkan dapat membantu pemerintah mengatasi krisis listrik dan dapat membantu menghemat biaya listrik.

II. KAJIAN TEORI

A. Energi Listrik

Energi dipercaya menjadi kunci komponen pada pembangunan ekonomi. Hal ini menciptakan asal daya tenaga yang sangat signifikan untuk setiap negara

[1]. Dalam kehidupan sehari – hari listrik merupakan salah satu energi yang dibutuhkan oleh masyarakat. Oleh karena itu energi listrik merupakan salah satu energi yang cukup mudah untuk diolah menjadi energi lainnya [5]. Semakin lama pertumbuhan penduduk semakin meningkat, begitu juga dengan kebutuhan produksi listrik, banyaknya penggunaan listrik oleh masyarakat, serta penggunaan perangkat listrik seperti barang elektronik yang dipakai akibatnya permintaan konsumsi energi semakin meningkat [6]. Faktor yang mempengaruhi konsumsi listrik yaitu jenis perangkat listrik serta jumlah, dan penggunaan perangkat listrik pada masyarakat. Pembatasan pemakaian energi listrik ini perlu dapat pemahaman lebih terutama pada masyarakat - masyarakat yang menggunakan energi listrik berlebih [7].

B. SQL

SQL mempunyai kepanjangan *Structured Query Language* yang dapat dimanfaatkan untuk mengakses dan menjalankan operasi lain pada basis data relasional. Bahasa pemrograman SQL berstandar ANSI (*American National Standards Institute*) yang dapat digunakan sebagai *query* di berbagai sistem pada *database*, baik *database* sistem *Big Data* modern maupun relasional tradisional [8].

C. MySQL

MySQL adalah program *database* berorientasi layanan yang tidak memerlukan antarmuka pengguna di desktop. Dapat digunakan dalam mode Teks, *Command prompt*, atau *PHPMyAdmin*. *PHPMyAdmin* merupakan aplikasi untuk mengelola server, *database*, dan objek yang berada di dalamnya [9]. Di dalam MySQL terdapat sistem manajemen *database* yang bersifat relational yang artinya data yang dikelola dalam *database* yang akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat. Fungsi MySQL untuk mengelola *database* mulai dari yang kecil sampai dengan yang sangat besar [10].

D. XAMPP

XAMPP adalah sebuah perangkat lunak yang didalamnya terdapat banyak fungsi, XAMPP dapat mendukung banyak sistem operasi yang artinya aplikasi ini bersifat *open source*. XAMPP bisa berfungsi sebagai server lokal atau yang biasa disebut dengan *localhost* dan juga merupakan penggabungan dari beberapa program [11].

E. Python

Python mempunyai fungsi sebagai aplikasi pengembangan perangkat lunak, dalam mendeklarasikan Python tidak ada tipe variabel, parameter, fungsi, atau metode seperti itu dalam penggunaannya. Python sendiri cukup mudah dipahami dalam pemakaian serta dapat diunduh di berbagai macam *platform* karena Python mempunyai sifat *open source*. Karena sifat Python yang mudah

jadi mudah untuk kebutuhan pengembangan perangkat lunak, membuat kode menjadi lebih fleksibel. Python dapat melacak semua jenis nilai pada waktu pemrosesan, file sumber Python saat ini menggunakan ekstensi ".py" dan disebut "modul" [12].

F. Flask

Flask adalah kerangka kerja aplikasi *website* yang ditulis dalam bentuk Python dan *baseband* pada mesin *template* *jinja2* dan *toolkit* *WSGI*. Karena flask menggunakan bahasa pemrograman Python yang fleksibel jadi flask menyediakan *template* sederhana untuk pengembangan web. Setelah di-*import* ke Python, Flask dapat digunakan untuk menghemat waktu membangun aplikasi web. Itu membuat inti tetap sederhana tetapi dapat diperluas [12].

G. Algoritma Particle Swarm Optimization

Algoritma *Particle Swarm Optimization* (PSO) diperkenalkan oleh Kennedy dan Eberhart pada tahun 1995, PSO adalah salah satu teknik evolusioner, populasi pada PSO didasarkan pada penelusuran algoritma dan diawali dengan suatu populasi yang disebut dengan *particle*. *Particle* tersebut bergerak melalui penelusuran ruang dengan *velocity* yang dinamis yang disesuaikan menurut perilaku historisnya. Oleh karena itu, partikel-partikel kecenderungan untuk bergerak ke arah yang lebih baik setelah melewati proses penelusuran. praktik berdasarkan perilaku biologis sosial spesies seperti, burung, penggembalaan dan kawanan. Misalnya, ikan mengambil rute yang sesuai (misalnya untuk makanan), sisa kawanan dapat segera mengikuti, bahkan jika mereka berada di sisi yang berlawanan, jika pencarian ikan memiliki sumber makanan (solusi potensial), mereka tidak memiliki pemimpin dalam kelompok atau kawanan mereka [13].

Pada saat yang sama, kondisi terbaik untuk *swarm* didapat melalui komunikasi antara anggotanya yang sudah memiliki situasi terbaik. Satu partikel yang memiliki kondisi lebih baik berbagi dengan kawanan dan yang lainnya bergerak secara bersamaan ke tempat di mana kondisi atau sumber makanan terbaik ditemukan. Dalam PSO, partikel mewakili solusi potensial. Dengan propertinya, mereka mengeksplorasi dan menggunakan perjalanan gerombolan partikel melalui ruang pencarian *n*-dimensi. Hipotesis ini dikaitkan dengan ruang pencarian dan diunggulkan dengan kecepatan awal sehubungan dengan posisi. PSO memiliki kesamaan dengan algoritma genetika yang mana dengan suatu populasi yang acak dalam bentuk matriks. Tetapi PSO tidak mempunyai operator evolusi yaitu *crossover* dan *mutasi* seperti yang ada pada algoritma genetika[14][15]. Beberapa istilah umum yang biasa digunakan dalam *Particle Swarm Optimization* dapat didefinisikan sebagai berikut:

1. *Swarm*: populasi dari suatu algoritma dengan $x^{(B)}$ dan $x^{(A)}$ secara acak sehingga didapat x_1 ,

x_2, \dots, x_N . Partikel j dan kecepatannya pada iterasi i dinotasikan sebagai $x_j^{(i)}$ dan $v_j^{(i)}$.

- Untuk menentukan nilai fungsi atau *fitness* pada kasus saat ini, ditentukan dengan *fitness* terbaik, dan ditetapkan sebagai *Gbest*.
- Particle*: anggota (individu) pada suatu *swarm*. Setiap *particle* merepresentasikan suatu solusi yang potensial pada permasalahan yang diselesaikan. Posisi dari suatu *particle* adalah ditentukan oleh representasi solusi saat itu.
- Pbest (Personal best)*: posisi *Pbest* suatu partikel yang menunjukkan posisi partikel yang dipersiapkan untuk mendapatkan suatu solusi yang terbaik.
- Gbest (Global best)*: posisi terbaik *particle* pada *swarm*.
- Velocity (vektor)*: vektor yang menggerakkan proses optimisasi yang menentukan arah di mana suatu *particle* diperlukan untuk berpindah yang bertujuan memperbaiki posisinya semula.

H. Antares

Antares adalah suatu platform pemrosesan yang mempunyai daya dan biaya yang cukup rendah dibanding pesaingnya. Kinerja dari Antares ini mempunyai kemampuan yang maksimal walaupun platform yang digunakan pada Antares berukuran kecil. Jalur yang digunakan pada platform Antares akan berkomunikasi secara terenkripsi yang sudah ditransmisikan. Fungsi dari platform Antares ini untuk merancang suatu sistem yang berbasis internet yang bisa digunakan pada sistem yang bergerak di platform *Internet of Things (IoT)* [16][17]. Dengan menggunakan skema *all-data-to-share* platform Antares mempunyai sistem akuisisi data serta dengan menggunakan kabel atau nirkabel platform Antares dapat digunakan untuk melakukan proses operasinya sendiri yang terdapat pada suatu sistem atau yang terhubung dengan modul Antares lain [18].

I. Daftar Tarif Listrik Menurut Golongan

Daftar tarif listrik per kWh dan golongannya sudah ditetapkan oleh Kementerian Energi dan Sumber Daya Mineral (ESDM). Tujuan dibuat daftar tarif listrik per kWh agar masyarakat dapat mengetahui tarif harga per kWh dengan menyesuaikan golongan gedung atau bangunan, serta dapat memberikan subsidi listrik sesuai dengan sasaran. Berikut merupakan daftar tarif harga listrik resmi yang telah ditentukan oleh ESDM dan PLN menurut golongannya [19].

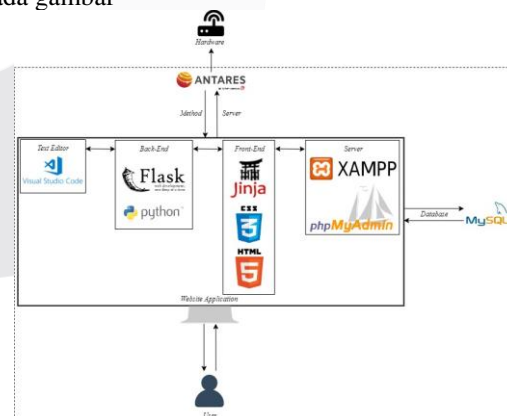
TABEL 1
GOLONGAN TARIF LISTRIK.

Golongan tarif listrik	Batas daya	Biaya pemakaian
R-1/TR	901 – 1.300 VA	Rp1.444,70/kWh
R-1/TR	1.301 – 2.200 VA	Rp1.444,70/kWh
R-2/TR	3.500 VA – 5.500 VA	Rp1.444,70/kWh
R-3/TR	> 6.600 VA	Rp1.444,70/kWh
B-1/TR	1.301 – 5.500 VA	Rp1.100/kWh
B-2/TR	6.600 VA – 200 kVA	Rp1.444,70/kWh
B-3/TM	> 200 kVA	Rp1.114,74/kWh
I-1/TR	1.301 – 2.200 VA	Rp960/kWh
I-1/TR	3.500 – 14.000 VA	Rp1.112/kWh
I-2/TR	14.001 – 200 kVA	Rp972/kWh
I-3P/TM	> 200 kVA	Rp1.114,74/kWh
I-3/TM	> 200 kVA	Rp1.114,74/kWh
I-4/TT	> 30.000 kVA	Rp996,74/kWh
P-1/TR	6.600 – 200 kVA	Rp1.444,70/kWh
P-2/TM	> 200 kVA	Rp1.114,74/kWh
P-3/TR		Rp1.444,70/kWh
L/TR, TM, TT		Rp1.644,52/kWh

III. METODE

A. Arsitektur Aplikasi

Pada arsitektur aplikasi *user* mengakses *website* yang dibuat menggunakan *text editor* yaitu *vscode*, dengan bahasa pemrograman *back-end* yang digunakan, yaitu bahasa pemrograman *python* dan *framework* yang dipakai untuk menyambungkan *back-end* menggunakan *flask*, selanjutnya *front-end* dan *ui* yang digunakan adalah *jinja2*, *css*, dan *html*. Untuk server pada *website* penjadwalan perangkat listrik ini menggunakan *local host* server yang memakai aplikasi *xampp* yang didalamnya terdapat *php* *myadmin*. Dalam sistem ini *database* yang digunakan terhubung pada *mysql*, kemudian *mysql* akan mengirim kembali data ke *website*. Selanjutnya data akan diteruskan ke Antares dan antares akan mengirim data kembali untuk ditampilkan pada *website* agar *user* dapat memonitoring durasi penggunaan listrik, seperti yang ditunjukkan pada gambar



GAMBAR 1
ARSITEKTUR APLIKASI.

B. Analisa Kebutuhan Sistem

Berdasarkan dengan penjelasan gambaran umum sistem, dapat diketahui sistem penjadwalan listrik membutuhkan beberapa aspek yang dapat mendukung implementasi sistem dalam bentuk *website*. Aspek yang dibutuhkan, yaitu Data, Perangkat Lunak, dan Kebutuhan Pengguna.

C. Kebutuhan Data

Berikut adalah data yang dibutuhkan agar sistem dapat berjalan dengan baik:

1. Data perangkat elektronik, data ini yang akan digunakan oleh pengguna, meliputi nama perangkat, daya perangkat, jumlah perangkat, durasi penggunaan, dan prioritas yang akan disimpan kedalam *database*.
2. Data nominal tagihan listrik, data ini yang nantinya diinginkan oleh pengguna dalam hari yang telah ditentukan yang nanti akan dibayarkan, kemudian akan disimpan kedalam *database*.

D. Perangkat Lunak

Perangkat lunak yang digunakan dalam perancangan model, pembuatan *website*, dan simulasi untuk sistem menggunakan perangkat lunak dengan spesifikasi sebagai berikut:

1. Sistem operasi Windows 10 64-bit
2. Web server Apache
3. MySQL
4. Visual Studio Code
5. Flask (Python)

E. Kebutuhan Pengguna

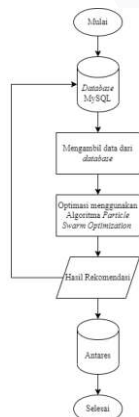
Pengguna yang akan menggunakan sistem penjadwalan perangkat listrik harus memenuhi kriteria sebagai berikut:

1. Mampu menjalankan komputer dengan baik.
2. Mampu menjalankan browser dengan jenis apapun.

F. Perancangan Sistem

Perancangan sistem adalah gambaran dari sistem penjadwalan perangkat listrik yang akan dirancang. Dalam perancangan sistem ini terdapat gambaran diagram alir sistem dan Algoritma *Particle Swarm Optimization*.

1. Diagram Alir Sistem



GAMBAR 2
DIAGRAM ALIR SISTEM.

Diagram alir sistem pada gambar menunjukkan alur kerja dari sistem penjadwalan perangkat listrik. Langkah awal sistem ini dengan

mengambil beberapa data dari pengguna, seperti data gedung, data *device*, dan data tagihan listrik yang diinginkan dalam waktu 30 hari. Setelah mendapatkan data-data mengenai perangkat listrik dari pengguna, selanjutnya Algoritma *Particle Swarm Optimization* akan mengoptimasi data tersebut, kemudian hasil optimasi sistem akan ditampilkan pada *website*. Hasil optimasi tersebut berupa rekomendasi durasi penggunaan perangkat listrik untuk setiap harinya, serta total tagihan listrik tidak melebihi biaya yang dimasukkan oleh pengguna dalam 30 hari. Hasil dari rekomendasi tersebut kemudian akan disimpan dalam *database* serta ditampilkan kembali pada *website*.

2. Use Case Diagram



Gambar 3 Use Case diagram.

Berdasarkan *use case diagram* di atas, dapat diketahui sebagai berikut:

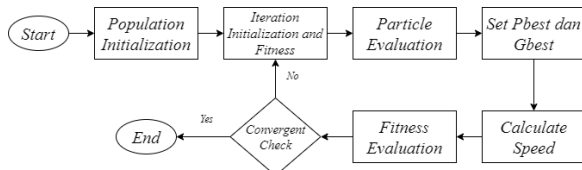
TABEL 2
USE CASE DIAGRAM AKTOR.

No	Aktor	Deskripsi
1	Pengguna	Aktor yang akan menggunakan <i>website</i> penjadwalan perangkat listrik ini.

TABEL 3
USE CASE DIAGRAM DESKRIPSI SISTEM.

No	Tipe Devinsi	Deskripsi
1	Nama	Memasukkan nama gedung
	Proses	Data masukan berupa <i>char</i> berisikan nama gedung.
	Kondisi akhir	Data ditampilkan pada tabel data gedung.
2	Nama	Memasukkan daya gedung
	Proses	Data masukan berupa <i>integer</i> berisikan daya gedung.
	Kondisi akhir	Dokumen telah melewati tahap <i>preprocessing</i> .
3	Nama	Memasukkan data <i>device</i>
	Proses	Data masukan berupa <i>integer</i> , <i>drop down list</i> dan <i>char</i> berisikan nama <i>device</i> , daya <i>device</i> , jumlah <i>device</i> , memilih nama gedung, dan prioritas.
	Kondisi akhir	Data ditampilkan pada tabel data <i>device</i> .
4	Nama	Memasukkan data simulasi algoritma
	Proses	Data masukan berupa <i>integer</i> dan <i>drop down list</i> berisikan memilih nama gedung, durasi (hari), budged, dan memilih metode algoritma, kemudian data diolah menggunakan Algoritma <i>Particle Swarm Optimization</i> .
	Kondisi akhir	Data rekomendasi durasi ditampilkan pada tabel data <i>device</i> .
5	Nama	Mengubah durasi yang diinginkan
	Proses	Data durasi diubah sesuai dengan yang diinginkan oleh <i>user</i> .
	Kondisi akhir	Data yang telah diubah dapat ditampilkan pada tabel data <i>device</i> .
6	Nama	Melihat data waktu
	Proses	Data rekomendasi dikurangi dengan durasi pemakaian <i>device</i> oleh <i>user</i> .
	Kondisi akhir	Data ditampilkan pada tabel monitor penggunaan.

3. Algoritma *Particle Swarm Optimization*



GAMBAR 4
FLOWCHART ALGORITMA PARTICLE SWARM OPTIMIZATION.

Pada pembuatan sistem penjadwalan perangkat listrik memakai metode Algoritma *Particle Swarm Optimization*, metode ini digunakan untuk pemilihan nilai prioritas pada suatu permasalahan. Pada permasalahan kali ini untuk melakukan optimasi penjadwalan pada perangkat listrik. Suatu populasi akan dikatakan konvergen ketika posisi semua partikel menuju nilai yang sama, selanjutnya populasi diperiksa untuk mengetahui solusi yang didapatkan sudah konvergen atau belum, ketika proses populasi sudah konvergen maka proses akan selesai. Akan tetapi jika populasi yang diperiksa belum konvergen maka akan diproses kembali sampai mendapatkan hasil yang konvergen.

1. Inisialisasi Populasi

Untuk inisialisasi populasi, partikel diisi oleh bilangan acak, namun menggunakan ketentuan berdasarkan prioritas yang telah dimasukkan untuk setiap perangkat listrik. Perangkat listrik yang telah dimasukkan akan berisi nilai acak yang terdapat pada setiap partikel dengan ketentuan berikut:

- a. *Device* kelas *High* = nilai acak mulai dari 17 sampai dengan 24.
- b. *Device* kelas *Mid* = nilai acak mulai dari 9 sampai dengan 16.
- c. *Device* kelas *Low* = nilai acak mulai dari 1 sampai dengan 8.

2. Evaluasi Partikel

Pada evaluasi partikel, dilakukan perhitungan *threshold* yang digunakan untuk membatasi penggunaan daya perhari, dengan biaya maksimal yang digunakan Rp.2.700.000,00 dalam sebulan dengan menggunakan rumus (2.1).

$$kWh \text{ per bulan} = \frac{2.700.000}{1.444,7}$$

$$kWh \text{ per bulan} = 1.868,9 \text{ kWh}$$

Perhitungan *threshold* dilakukan menggunakan rumus sebagai berikut:

$$Threshold = \frac{kWh \text{ per bulan}}{Hari} \dots (3.1)$$

$$Threshold = \frac{1.868,9}{30}$$

$$Threshold = 62,2 \text{ kWh}$$

Dibulatkan menjadi:

$$Threshold = 62 \text{ kWh/hari}$$

Pada tahap evaluasi partikel jika total kWh pada partikel melebihi penggunaan

kWh/hari(threshold), akan dilakukan pengurangan 10% disetiap gen pada partikel.

G. Perhitungan Algoritma *Particle Swarm Optimization*

Berikut adalah perhitungan manual dari Algoritma *Particle Swarm Optimization* yang akan dimulai dari menentukan parameter, inisialisasi populasi, evaluasi partikel, menentukan *pbest*, evaluasi nilai fungsi, cek konvergen. Dimana nilai *input* yang akan digunakan sebagai berikut:

TABEL 4
INPUT DEVICE.

Device	Nama Device	Daya	Jumlah	Kelas	Golongan
D1	Kulkas	150 Watt	1	High	1.444,7
D2	Dispenser	350 Watt	1	Mid	1.444,7
D3	Water Heater	350 Watt	1	High	1.444,7
D4	Kipas	80 Watt	6	High	1.444,7
D5	Lampu	5 Watt	25	Mid	1.444,7
D6	AC	450 Watt	3	Mid	1.444,7
D7	Rice Cooker	450 Watt	1	Mid	1.444,7
D8	TV	100 Watt	5	Mid	1.444,7
D9	Komputer	450 Watt	1	Low	1.444,7
D10	Pompa Air	450 Watt	1	Low	1.444,7
D11	Mesin Cuci	100 Watt	1	Low	1.444,7
D12	Printer	20 Watt	1	Low	1.444,7

1. Inisialisasi Populasi

Pada tahap inisialisasi populasi untuk meningkatkan jumlah populasi awal terdapat 12 partikel yang tiap partikel terdapat 12 *device* sehingga partikel yang digunakan sebanyak 144 partikel. Berikut merupakan gambaran untuk populasi awal:

```

Initial population:
[23 23 23 24 22 19 23 23 19 21 19 24]
[24 17 18 20 23 23 19 19 17 19 23 22]
[21 23 21 22 18 21 24 18 24 17 17 18]
[21 24 21 20 21 17 18 20 20 24 22 19]
[16 14 11 12 12 12 16 14 14 10 15 9]
[14 13 13 10 15 16 15 16 16 13 12 16]
[10 12 16 14 12 14 14 9 10 13 12 11]
[16 11 15 14 11 12 10 15 15 11 9 12]
[1 6 3 6 4 7 6 3 7 5 7 6]
[3 4 4 1 6 4 6 8 5 5 3 4]
[6 4 3 3 1 5 7 5 4 2 1]
[8 4 7 3 6 7 4 2 6 4 4 8]
    
```

GAMBAR 5
INISIALISASI POPULASI.

2. Evaluasi Partikel

Setelah mendapatkan populasi awal, setiap partikel akan dilakukan perhitungan nilai *fitness* sebagai berikut:

$$Fitness(i) = \sum_{i=1}^n \frac{V(i) \times W(i) \times JD(i)}{1000} \dots (1)$$

$$Fitness(1) = \frac{23 \times 150 \times 1}{1000} + \frac{24 \times 350 \times 1}{1000} + \frac{21 \times 350 \times 1}{1000} + \frac{21 \times 80 \times 6}{1000} + \frac{16 \times 5 \times 25}{1000} + \frac{14 \times 450 \times 3}{1000} + \frac{10 \times 450 \times 1}{1000} + \frac{16 \times 100 \times 5}{1000} + \frac{1 \times 450 \times 1}{1000} + \frac{3 \times 450 \times 1}{1000} + \frac{6 \times 100 \times 1}{1000} + \frac{8 \times 20 \times 1}{1000}$$

$$Fitness(1) = 65.24$$

Keterangan:

V(i) = Durasi ke-i

W(i) = Watt ke-i

JD(i) = Jumlah Device ke-i

Pada tahap evaluasi partikel jika total kWh pada partikel melebihi penggunaan kWh/hari (threshold), akan dilakukan pengurangan 10% pada partikel. Perhitungan diatas untuk partikel pertama, kemudian proses ini akan dilakukan terhadap setiap partikel.

3. Menentukan P_{best}

Pada tahap menentukan P_{best} akan dilakukan pencarian nilai terbaik dari $X_j(i)$ dan dinyatakan sebagai $P_{best,j}$, dengan nilai fungsi objektif paling rendah $[f[X_j(i)]$ yang ditemui sebuah partikel j pada semua iterasi sebelumnya, Nilai terbaik untuk semua partikel $X_j(i)$ yang ditemukan sampai iterasi ke-I, $G_{best,j}$, dengan nilai fungsi tujuan paling kecil diantara semua partikel untuk semua iterasi sebelumnya, $f[X_j(i)]$. Adapun rumus yang dipakai untuk menghitung nilai P_{best} . Berikut merupakan rumus untuk menentukan P_{best} :

$$V_j(i) = V_j(i-1) + c_1 r_1 [P_{best,j} - x_j(i-1)] + c_2 r_2 [G_{best,j} - x_j(i-1)] \dots (2)$$

Keterangan:

$V_j(i-1)$ = Kecepatan awal

$x_j(i-1)$ = Posisi partikel ke-i

r_1, r_2 = Bilangan random

$C1, C2$ = Koefisien akselerasi

$G_{best,j}$ = Nilai terbaik dari p_{best}

$$V(1) = 0 + (1 \times 0.5)[23 - 23] + (1 \times 0.4)[24 - 23]$$

$$V(1) = 0.4$$

G_{best} yang digunakan disini yaitu, nilai G_{best} high 24, nilai G_{best} mid 16, dan nilai G_{best} low 8. Untuk nilai acak yang digunakan r_1 0.5, r_2 0.4, dan $C1 = C2$ 1. Dimana c_1 dan c_2 masing-masing adalah *learning rates* untuk kemampuan individu (cognitive) dan pengaruh social (group), serta r_1 dan r_2 bilangan random yang berdistribusi dan interval 0 dan 1. Jadi parameters c_1 dan c_2 menunjukkan bobot dari posisi sebuah partikel terhadap posisi dari kelompok (swarm). Nilai dari c_1 dan c_2 biasanya adalah 2 sehingga perkalian $c_1 r_1$ dan $c_2 r_2$ memastikan bahwa partikel – partikel akan mendekati target sekitar setengah selisihnya.

4. Evaluasi Nilai Fungsi

Setelah mendapatkan nilai P_{best} , dilakukan proses *evaluasi nilai fungsi* yang merupakan evaluasi nilai P_{best} yang sudah dihitung sebelumnya.

$$Fitness(i) = \sum_{i=1}^n \frac{V(i) \times W(i) \times JD(i)}{1000} \dots (3)$$

$$Fitness(1) = \frac{17 \times 150 \times 1}{1000} + \frac{17 \times 350 \times 1}{1000} + \frac{17 \times 350 \times 1}{1000} + \frac{17 \times 80 \times 6}{1000} + \frac{16 \times 5 \times 25}{1000} + \frac{16 \times 450 \times 3}{1000} + \frac{11 \times 450 \times 1}{1000} + \frac{16 \times 100 \times 5}{1000} + \frac{6 \times 450 \times 1}{1000} + \frac{6 \times 450 \times 1}{1000} + \frac{6 \times 100 \times 1}{1000} + \frac{8 \times 20 \times 1}{1000}$$

$$Fitness(1) = 65.32$$

Keterangan:

$V(i)$ = Durasi ke-i

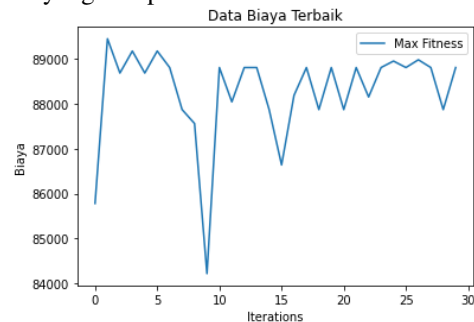
$W(i)$ = Watt ke-i

JD(i) = Jumlah Device ke-i

Proses *evaluasi nilai fungsi* tidak hanya dilakukan satu kali saja, namun dilakukan secara terus menerus hingga nilai acak yang dibangkitkan lebih kecil dari *threshold*. Setelah proses *evaluasi nilai fungsi* selesai, akan dilakukan pengecekan terhadap setiap partikel untuk memastikan berat partikel tidak melebihi *threshold*. Jika partikel melebihi *threshold*, maka nilai partikel tersebut akan di nol kan mulai dari partikel yang paling akhir dari setiap partikel, hingga beratnya sama atau lebih ringan dari *threshold*.

5. Cek Konvergen

Cek apakah solusi yang sekarang sudah konvergen. Jika posisi semua partikel menuju ke satu nilai yang sama, maka ini disebut konvergen. Jika belum konvergen maka langkah menentukan P_{best} diulang dengan memperbarui iterasi = i+1, dengan cara menghitung nilai baru dari P_{best} dan G_{best} . Proses iterasi ini dilanjutkan sampai semua partikel menuju ke satu titik solusi yang sama. Biasanya akan ditentukan dengan kriteria penghentian, misalnya jumlah selisih solusi sekarang dengan solusi sebelumnya sudah sangat kecil. Jumlah iterasi yang sebelumnya sudah sangat kecil. Jumlah iterasi yang sebelumnya 20 ketika dinaikan sampai 30, nilai *fitness* optimal yang didapatkan berada di iterasi 2.



GAMBAR 6
GRAFIK ITERASI.

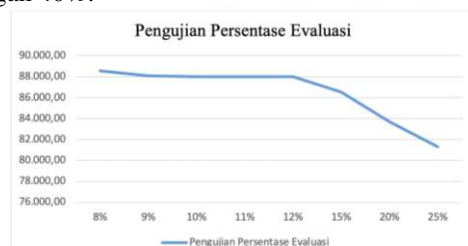
H. *Time Remaining*

Time remaining merupakan pengurangan durasi dari data rekomendasi pada tahap Algoritma *Particle Swarm Optimization*. Pengurangan durasi dilakukan pada saat *user* menggunakan *device*. Memakai *virtual device* untuk memantau durasi pemakaian *device*.

IV. HASIL DAN PEMBAHASAN

A. Pengujian Persentase Evaluasi

Berikut pengujian persentase evaluasi dilakukan untuk mengetahui perbedaan nilai *fitness* terbaik terhadap nilai persentase evaluasi dari 5% sampai dengan 40%.

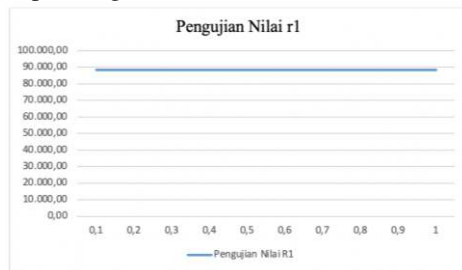


GAMBAR 7
GRAFIK PENGUJIAN PERSENTASE EVALUASI.

Pengujian persentase evaluasi pada grafik dapat disimpulkan bahwa semakin besar persentase evaluasi semakin kecil nilai *fitness* terbaik yang didapatkan sehingga semakin tidak baik untuk diimplementasikan ketika persentase evaluasinya semakin besar.

B. Pengujian Nilai R1

Pengujian ini dilakukan untuk mengetahui perbedaan nilai *fitness* terbaik berdasarkan nilai r_1 dari 0.1 sampai dengan 1.

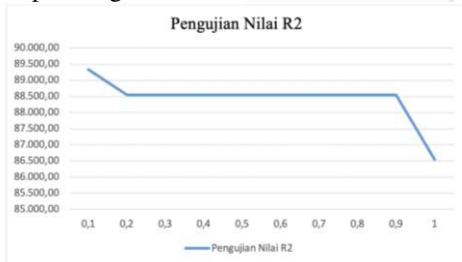


GAMBAR 8
GRAFIK PENGUJIAN NILAI R1

Pengujian nilai r_1 pada tabel 4.2 dan grafik dapat disimpulkan bahwa nilai r_1 tidak berpengaruh terhadap nilai *fitness*, dikarenakan pada rumus (3.4) P_{best} dikurangi dengan X_0 hasilnya 0, 0 dikalikan dengan angka berapapun hasilnya akan tetap 0 sehingga nilai r_1 tidak berpengaruh terhadap nilai *fitness*.

C. Pengujian Nilai R2

Pengujian ini dilakukan untuk mengetahui perbedaan nilai *fitness* terbaik berdasarkan nilai r_2 dari 0.1 sampai dengan 1.



GAMBAR 9
GRAFIK PENGUJIAN NILAI R2.

Pengujian r_2 pada tabel 4.3 dan grafik dapat disimpulkan bahwa semakin besar nilai r_2 terbaik terdapat pada nilai 0,1, dan nilai r_2 terburuk terdapat pada nilai 1, sehingga semakin besar nilai r_2 semakin tidak bagus untuk diimplementasikan.

D. Pengujian Nilai C1 dan C2

Pengujian ini dilakukan untuk mengetahui perbedaan nilai *fitness* terbaik berdasarkan nilai C1 dan C2 dari 0,1 sampai dengan 4.



GAMBAR 10
GRAFIK PENGUJIAN NILAI C1 DAN C2.

Pengujian C1 dan C2 pada tabel 4.4 dan grafik dapat disimpulkan bahwasemakin besar nilai C1 dan C2 maka akan menurunkan nilai *fitness* terbaik, sehingga semakin besar nilai C1 dan C2 semakin tidak bagus untuk diimplementasikan.

E. Pengujian Iterasi

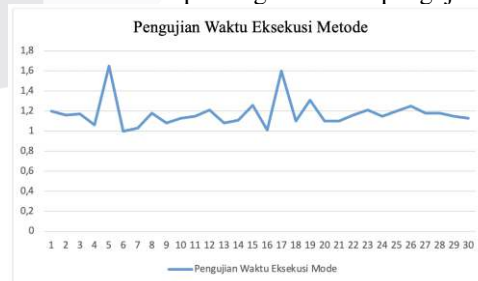


GAMBAR 11
GRAFIK PENGUJIAN ITERASI.

Berdasarkan grafik di atas, nilai *fitness* yang paling optimal didapatkan pada iterasi ke 30, 60, 80, dan 100. Hal tersebut dapat dilihat terjadi penurunan pada iterasi ke 40, 70, dan 90.

F. Pengujian Waktu Eksekusi Metode

Pengujian waktu eksekusi dilakukan untuk mengetahui kecepatan sistem dalam merespon penggunaannya. Dilakukan pengujian respon mengirim dan menerima data antara *database* dan *website* sistem penjadwalan perangkat listrik, dilakukan sebanyak 30 kali untuk mendapatkan nilai waktu respon yang lebih akurat. Berikut merupakan grafik hasil pengujian:



GAMBAR 12
GRAFIK PENGUJIAN WAKTU EKSEKUSI MODE.

Dengan pengujian pada gambar 4.7 didapatkan hasil rata-rata dari eksekusi waktu sebesar 1,17 detik. Waktu eksekusi didapatkan dari *website* yang telah dirancang untuk setiap perhitungan *output* akan menampilkan juga lama waktu proses dari

perhitungan tersebut. Contoh perhitungan waktu eksekusi pada *website* terdapat pada lampiran A.

V. KESIMPULAN

Berdasarkan hasil dari pengujian dan analisis yang telah dilakukan pada Tugas Akhir ini, didapatkan pengujian performa Algoritma *Particle Swarm Optimization*, didapatkan hasil pengujian persentase evaluasi 8%-25% berhasil. Selanjutnya pada pengujian nilai r_1 dan r_2 mendapatkan hasil r_1 dengan *fitness* terbaik pada percobaan 0,1 dengan hasil 61,29 kWh yang dinyatakan berhasil dan r_2 dengan *fitness* terbaik pada percobaan 0,1 dengan hasil 61,835 kWh yang dinyatakan berhasil. Setelah itu pada pengujian nilai C1 dan C2 didapatkan *fitness* terbaik pada nilai 0,5 dengan hasil 61,29 kWh yang dinyatakan berhasil.

REFERENSI

- [1] I. Dincer, "On energy conservation policies and implementation practices," *Int. J. Energy Res.*, vol. 27, no. 7, pp. 687–702, 2003, doi: 10.1002/er.912.
- [2] M. Z. Sasongko and Sucipto, "Desain Prototype IoT menggunakan Bot Telegram Berbasis Text Recognition IoT Prototype Design using Text Recognition Based Telegram Bot," *Res. J. Comput.*, vol. 4, no. 1, pp. 21–27, 2021.
- [3] G. E. Akpan and U. F. Akpan, "Electricity consumption, carbon emissions and economic growth in Nigeria," *Int. J. Energy Econ. Policy*, vol. 2, no. 4, pp. 292–306, 2012.
- [4] S. N. Makhadmeh, A. T. Khader, M. A. Al-Betar, S. Naim, Z. A. A. Alyasseri, and A. K. Abasi, "Particle Swarm optimization Algorithm for Power Scheduling Problem Using Smart Battery," *IEEE Jordan Int. Jt. Conf. Electr. Eng. Inf. Technol.*, pp. 672–677, 2019, doi: 10.1109/JEEIT.2019.8717468.
- [5] D. Prihatmoko, "Pemanfaatan Raspberry Pi Sebagai Server Web Untuk Penjadwalan Kontrol Lampu Jarak Jauh," *J. Infotel*, vol. 9, no. 1, p. 84, 2017, doi: 10.20895/infotel.v9i1.159.
- [6] B. N. Getu and H. A. Attia, "Electricity audit and reduction of consumption: Campus case study," *Int. J. Appl. Eng. Res.*, vol. 11, no. 6, pp. 4423–4427, 2016.
- [7] A. J. Abid and A. Hussein Ali, "Smart Monitoring of the Consumption of Home Electrical Energy," *Int. J. Comput. Trends Technol.*, vol. 47, no. 2, pp. 142–148, 2017, doi: 10.14445/22312803/ijctt-v47p120.
- [8] R. Amornchewin, "The Development of SQL Language Skills in Data Definition and Data Manipulation Languages Using Exercises with Quizizz for Students' Learning Engagement," *IJIE (Indonesian J. Informatics Educ.)*, vol. 2, no. 2, p. 83, 2018, doi: 10.20961/ijie.v2i2.24430.
- [9] B. Rawat, S. Purnama, and M. Mulyati, "MySQL Database Management System (DBMS) On FTP Site LAPAN Bandung," *Int. J. Cyber IT Serv. Manag.*, vol. 1, no. 2, pp. 173–179, 2021, doi: 10.34306/ijcitsm.v1i2.47.
- [10] "Pengertian MySQL | EduChannel Indonesia." <https://educhannel.id/blog/artikel/pengertian-mysql.html> (accessed Jun. 23, 2022).
- [11] "XAMPP Installers and Downloads for Apache Friends." <https://www.apachefriends.org/index.html> (accessed Jun. 23, 2022).
- [12] F. A. Aslam, H. N. Mohammed, J. Musab, and M. Munir, "Efficient Way Of Web Development Using Python And Flask Fankar," *Int. J. Adv. Res. Comput. Sci.*, vol. 6, no. 2, pp. 54–57, 2015.
- [13] S. Sharma and H. M. Pandey, "Genetic Algorithm, Particle Swarm Optimization and Harmony Search: A quick comparison," *Int. Conf. - Cloud Syst. Big Data Eng.*, pp. 40–44, 2016, doi: 10.1109/CONFLUENCE.2016.7508044.
- [14] R. Ashok Bakkiyaraj and N. Kumarappan, "Optimal reliability planning for a composite electric power system based on Monte Carlo simulation using particle swarm optimization," *Int. J. Electr. Power Energy Syst.*, vol. 47, no. 1, pp. 109–116, 2013, doi: 10.1016/j.ijepes.2012.10.055.
- [15] A. A. A. Esmiin and G. Lambert-Torres, "Application of particle swarm optimization to optimal power systems," *Int. J. Innov. Comput. Inf. Control*, vol. 8, no. 3 A, pp. 1705–1716, 2012.
- [16] M. Solehman, F. Azmi, and C. Setianingsih, "Web-Based Flood Warning System Using Decision Tree Method," *Int. Conf. Mechatronics Autom.*, pp. 0–5, 2019.
- [17] T. Nugroho, M. Nasrun, and C. Setianingsih, "Smart Lamp Control Based on User Behavior for Two Lamps Using K-Nearest Neighbour," *Int. Conf. Adv. Mechatronics, Intell. Manuf. Ind. Autom.*, pp. 123–128, 2019, doi: 10.1109/ICAMIMIA47173.2019.9223423.
- [18] C. BIGONGIARI, "Search for Neutrino Emission From Gamma-Ray Sources With the Antares Telescope," *Int. J. Mod. Phys. Conf. Ser.*, vol. 08, no. Hepro Iii, pp. 307–310, 2012, doi: 10.1142/s201019451200476x.
- [19] "Daftar Tarif Listrik Terbaru Kementerian ESDM April – Juni 2021." https://lifepal.co.id/media/daftar-tarif-listrik-terbaru/?utm_campaign=MEDIA_tagihan-listrik_begini-lho-cara-menghitung-tarif-listrik-benar&utm_source=media&utm_medium=inarticle_text&utm_content=begini-lho-cara-menghitung-tarif-listrik-benar (accessed Jun. 23, 2022).