

Optimalisasi Distribusi Energi untuk Mesin Produksi Menggunakan Algoritma Berbasis Perilaku Physarum Polycephalum

Ferdin Arsenarendra Purtadi - 13523117¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

arxenarendra@gmail.com, 13523117@std.stei.itb.ac.id

Abstrak—Algoritma jamur lendir (Slime Mold Algorithm) didasarkan pada perilaku biologis organisme uniseluler Physarum polycephalum, yang mampu menemukan jalur terbaik antara sumber dan tujuan. Algoritma ini digunakan untuk mengoptimalkan distribusi energi ke mesin industri. Algoritma ini menemukan jalur distribusi energi yang efisien dengan menggunakan penguatan jalur berbasis feromon dan penguapan, yang digambarkan dalam bentuk graf berbobot. Hasil simulasi menunjukkan bahwa SMA efektif dalam meminimalkan biaya distribusi serta mampu menyesuaikan diri dengan perubahan permintaan energi. Metode ini menawarkan solusi kreatif untuk manajemen energi dalam jaringan industri.

Kata Kunci—Distribusi Energi, Optimasi Jaringan, Sistem Industri, Slime Mold Algorithm.

I. LATAR BELAKANG

Industri modern saat ini menghadapi masalah besar dalam mengelola distribusi energi ke mesin produksi. Kompleksitas jaringan distribusi, peningkatan konsumsi energi, dan kebutuhan keberlanjutan dan efisiensi operasional menjadi masalah utama. Dalam situasi seperti ini, metode optimasi yang dapat menyeimbangkan biaya dan efisiensi distribusi energi sangat diperlukan.

Jamur lendir (Physarum polycephalum), organisme uniseluler yang dikenal karena kemampuannya menemukan jalur optimal, telah menjadi inspirasi dalam pengembangan algoritma komputasi. Perilaku biologisnya yang menggunakan penguatan jalur berbasis feromon dan penguapan, sangat relevan untuk memodelkan jaringan distribusi energi yang dinamis. Metode ini memungkinkan untuk menemukan jalur distribusi energi yang efisien sambil mempertimbangkan biaya energi, kebutuhan mesin, dan kondisi jaringan umum.

Penelitian-penelitian sebelumnya menunjukkan bahwa algoritma yang terinspirasi dari Physarum polycephalum dapat digunakan untuk optimasi berbagai jaringan. Sebagai contoh, penelitian oleh MDPI [1] mendemonstrasikan bagaimana algoritma ini dapat digunakan dalam distribusi energi listrik untuk mengurangi kehilangan daya dan meningkatkan efisiensi operasional. Selain itu, studi oleh Springer [2] menunjukkan

potensi besar algoritma jamur lendir dalam operasi optimal mikrogrid, yang relevan untuk kebutuhan industri modern.

Makalah ini bertujuan untuk mengeksplorasi penerapan algoritma berbasis jamur lendir dalam mengoptimalkan distribusi energi ke mesin-mesin produksi. Dengan memodelkan jaringan sebagai graf dan menggunakan algoritma ini, diharapkan dapat dihasilkan solusi yang tidak hanya efisien tetapi juga adaptif terhadap perubahan kebutuhan energi di lingkungan industri.

II. DASAR TEORI

A. Graf

Graf adalah struktur matematika yang digunakan untuk memodelkan hubungan antar objek. Dalam konteks distribusi energi, graf digunakan untuk memodelkan hubungan antara sumber energi, pusat distribusi, dan mesin produksi. Graf didefinisikan sebagai pasangan $G=(V,E)$, di mana V adalah himpunan simpul (vertex) yang merepresentasikan entitas (misalnya, sumber energi, mesin produksi) dan E adalah himpunan sisi (edge) yang merepresentasikan hubungan atau jalur antar simpul.

Di dalam graf, terdapat elemen-elemen sebagai berikut:

1. Simpul (Vertices)
Simpul adalah entitas dasar dalam graf yang merepresentasikan objek atau titik. Dalam distribusi energi, simpul dapat menggambarkan sumber energi, substation, atau mesin produksi.
2. Sisi (Edges)
Sisi menghubungkan dua simpul dan merepresentasikan hubungan atau jalur antara keduanya. Dalam konteks distribusi energi, sisi dapat mencerminkan jalur distribusi energi antara sumber energi dan beban.
3. Bobot
Bobot adalah nilai numerik yang diberikan pada sisi. Bobot dapat mencerminkan biaya energi, resistansi, atau jarak distribusi.
4. Path (Lintasan)
Lintasan adalah rangkaian simpul yang dihubungkan oleh sisi, dimulai dari simpul awal dan berakhir di simpul tujuan. Dalam graf energi, lintasan menunjukkan jalur

energi dari sumber ke tujuan tertentu.

5. Sirkuit atau siklus

Siklus adalah lintasan yang dimulai dan berakhir di simpul yang sama tanpa mengulangi sisi. Dalam distribusi energi, siklus mungkin menunjukkan redundansi atau loop energi.

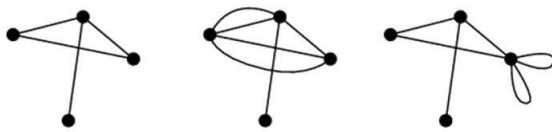
6. Derajat

Derajat simpul adalah jumlah sisi yang terhubung ke simpul tersebut:

- Derajat Masuk (In-Degree): Jumlah sisi yang masuk ke simpul dalam graf berarah.
- Derajat Keluar (Out-Degree): Jumlah sisi yang keluar dari simpul dalam graf berarah.

Graf dibagi menjadi beberapa jenis, yaitu :

1. Graf Tidak Berarah:

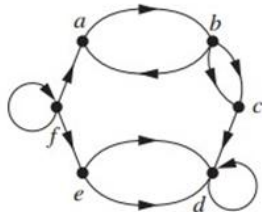


Gambar 1 Graf Tak Berarah

Sumber : [8]

Sisi graf tidak memiliki arah, contohnya hubungan simetris antara node distribusi energi.

2. Graf Berarah:



Gambar 2 Graf Berarah

Sumber : [8]

Setiap sisi graf memiliki orientasi arah, merepresentasikan aliran energi dari sumber ke tujuan.

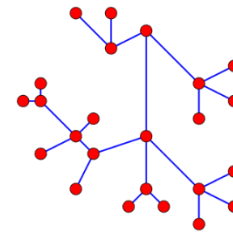
3. Graf Berbobot:

Setiap sisi graf memiliki bobot yang merepresentasikan biaya distribusi energi, resistansi kabel, atau jarak fisik.

Graf dapat direpresentasikan dalam matriks ketetanggaan (Adjacency Matrix), yaitu matriks dua dimensi yang menunjukkan ada atau tidaknya hubungan antara simpul. 1 jika terdapat sisi antara simpul i dan j atau 0 jika tidak ada sisi.

B. Pohon (Tree)

Pohon adalah graf tak-berarah yang terhubung dan tidak memiliki siklus, untuk n simpul, pohon memiliki tepat n-1 sisi. Pohon sering digunakan untuk merepresentasikan struktur hierarki atau jaringan distribusi optimal dalam sistem energi.



Gambar 3 Pohon

Sumber : [11]

Dalam pohon, ada yang dikenal dengan Pohon Merentang Minimum (MST). Pohon Merentang Minimum adalah subgraf dari pohon yang mencakup semua simpul tanpa membentuk siklus serta memiliki total bobot sisinya terkecil. Ada dua algoritma yang biasa digunakan untuk menentukan MST, yaitu algoritma Kruskal dan algoritma Prim.

1. Algoritma Kruskal:

- Memilih sisi dengan bobot terkecil.
- Menambahkan sisi ke pohon selama tidak membentuk siklus.

2. Algoritma Prim:

- Dimulai dari satu simpul.
- Menambahkan sisi dengan bobot terkecil yang menghubungkan simpul baru.

MST digunakan untuk meminimalkan biaya distribusi energi, terutama dalam jaringan besar yang kompleks.

III. ALGORITMA JAMUR LENDIR

Algoritma jamur lendir (Slime Mold Algorithm - SMA) terinspirasi dari perilaku biologis Physarum polycephalum, organisme uniseluler yang mampu menemukan jalur optimal antara sumber makanan dan tujuan. Algoritma ini telah diadaptasi dalam berbagai masalah optimasi seperti distribusi energi, desain jaringan, dan routing.

Tahapan algoritma meliputi :

1. Representasi Graf, di mana:

- Himpunan simpul yang mewakili titik distribusi energi (misalnya, sumber atau beban).
- Himpunan sisi yang mewakili jalur distribusi energi, dengan bobot C_{ij} yang mencerminkan resistansi, jarak, atau biaya.

2. Inisialisasi:

- Tetapkan tingkat awal feromon P_{ij} dan kapasitas tubulus T_{ij} untuk semua jalur.
- Hitung tekanan awal berdasarkan permintaan dan resistansi. Hitung tekanan awal Q_{ij} berdasarkan permintaan D_{ij} dan resistansi R_{ij} :

$$Q_{ij} = \frac{D_{ij}}{R_{ij}}$$

3. Penguatan dan Penguapan Jalur:

- Penguatan feromon pada jalur, penguapan

feromon untuk jalur tidak efisien :

$$P[i][j] \leftarrow P[i][j] + \frac{C_{ij}}{Q_{ij}}$$

- Iterasi dilakukan hingga tingkat feromon stabil atau jalur optimal ditemukan.

Fungsi objektif dari algoritma ini adalah meminimalkan total biaya distribusi energi. Biaya total distribusi energi dihitung sebagai gabungan dari resistansi jalur dan aliran energi pada jalur tersebut. Dalam konteks distribusi energi, biaya total menjadi indikator utama efisiensi jaringan, mencerminkan seberapa optimal energi dapat dialirkan dari sumber ke mesin produksi. Rumus utama untuk menghitung biaya total adalah sebagai berikut :

$$C_{total} = \sum_{\substack{i \in E \\ j \in E}} C(i, j) \cdot Q(i, j)$$

Di mana,

- C_{total} : Total biaya distribusi energi
- Q_{ij} : Aliran energi pada jalur antara i dan j
- R_{ij} : Resistansi jalur antara i dan j
- E : set jalur dalam graf distribusi energi

IV. PEMBAHASAN

A. Penjelasan Program Simulasi

Algoritma jamur lendir (Slime Mold Algorithm - SMA) akan digunakan dalam penentuan jalur terbaik yang dapat dibuat untuk meminimalkan pengeluaran biaya distribusi energi. Sebuah program dalam Python dibuat berdasarkan algoritma jamur lendir untuk mensimulasikan dan menyelesaikan masalah ini, rinciannya adalah sebagai berikut:

1. Inisialisasi

Kelas SlimeMold bertanggung jawab untuk mengelola tingkat feromon, penguapan, penguatan jalur, dan proses pencarian jalur terbaik. Inisialisasi mencakup:

- **pheromone:**
Sebagai indikator preferensi jalur, nilai awal seragam di seluruh jalur, namun akan diperkuat atau diupkan selama simulasi.
- **evaporation_rate:**
Laju penguapan feromon (default: 0.8) yang berfungsi untuk mengontrol pengurangan tingkat feromon di jalur yang jarang digunakan.
- **pheromone_boost:**
Berfungsi sebagai faktor penguatan feromon (default: 2.0) untuk meningkatkan feromon di jalur yang efisien berdasarkan rasio aliran energi terhadap resistansi.

Berikut kode yang diimplementasikan :

```
class SlimeMold:
```

```
def __init__(self, num_nodes,
evaporation_rate=0.8, pheromone_boost=2.0):
    self.num_nodes = num_nodes
    self.pheromone = np.ones((num_nodes,
num_nodes))
    self.evaporation_rate =
evaporation_rate
    self.pheromone_boost =
pheromone_boost
```

- Matriks feromon diinisialisasi secara acak dengan nilai antara 1 dan 5.
- Parameter `evaporation_rate` dan `pheromone_boost` menentukan bagaimana feromon diperbarui selama iterasi.

Feromon awal di semua jalur diatur seragam, tetapi akan diperkuat oleh `pheromone_boost` setiap kali jalur tersebut digunakan untuk mendistribusikan energi. Penguatan ini berbanding lurus dengan efisiensi jalur yang dihitung dari rasio aliran energi terhadap resistansi. Sebaliknya, jalur yang jarang digunakan akan mengalami penguapan feromon yang dikontrol oleh `evaporation_rate`, mengurangi peluang jalur tersebut dipilih di iterasi berikutnya.

2. Simulasi Distribusi Energi

Fungsi `simulate energy distribution` adalah inti simulasi ini. Fungsi ini menjalankan iterasi algoritma SMA untuk memperbarui feromon, menghitung aliran energi dan memilih jalur terbaik.

a. Inisialisasi Matriks Resistansi dan Biaya

Resistansi antar simpul diatur dalam matriks berbobot:

```
for (u, v), r in zip(connections,
resistances):
    costs[u, v] = costs[v, u] = r
```

- Matriks `costs` merepresentasikan resistansi antarsimpul.
- Jika resistansi bernilai negatif atau nol, simulasi dihentikan karena jalur tidak valid

b. Penguapan dan Penguatan Feromon

Pada setiap iterasi akan dilakukan penguapan feromon dan penguatan feromon. Penguapan feromon, feromon pada semua jalur berkurang sesuai `evaporation_rate`:

```
slime_mold.evaporate_pheromone()
```

Penguatan feromon, feromon diperbarui berdasarkan aliran energi:

```
flows[u,v]=
slime_mold.calculate_flow(demand,
```

```

costs[u,v])
    slime_mold.pheromone[u,v]+=
    slime_mold.pheromone_boost*flows[u,v]/
    costs[u, v]

    slime_mold.pheromone[v,u]
    = slime_mold.pheromone[u, v]

```

Matriks biaya (costs) merepresentasikan resistansi antar simpul, yang dihitung dalam satuan ohm. Resistansi ini menentukan efisiensi jalur dalam mendistribusikan energi. Aliran energi (flows) dihitung menggunakan rumus:

$$Q_{ij} = \frac{D_{ij}}{R_{ij}}$$

Jalur dengan resistansi rendah menghasilkan aliran energi yang lebih besar, sehingga diperkuat oleh mekanisme penguatan feromon.

c. Pemilihan Jalur Terbaik

Jalur terbaik dipilih berdasarkan tingkat feromon tertinggi pada setiap langkah ;

```

def get_best_path(self, start, end):
    path = [start]
    current_node = start
    visited = set(path)

    while current_node != end:
        neighbors = [(neighbor,
self.pheromone[current_node, neighbor])]

        for neighbor in
range(self.num_nodes) if neighbor not in
visited]
            next_node = max(neighbors,
key=lambda x: x[1])[0]
            path.append(next_node)
            visited.add(next_node)
            current_node = next_node

    return path

```

- Algoritma memilih jalur dengan tingkat feromon tertinggi hingga mencapai simpul tujuan.
- Jalur optimal secara global dicapai melalui iterasi dan penguatan.

3. Hasil Simulasi

Setelah iterasi selesai dilakukan, hasilnya meliputi:

- Jalur terbaik yang merupakan jalur dari simpul awal ke simpul tujuan dengan tingkat feromon tertinggi.
- Biaya total yang dihitung sebagai:

$$C_{total} = \sum_{\substack{i \in E \\ j \in E}} C(i, j) \cdot Q(i, j)$$

```

total_cost = sum(costs[u, v] * flows[u, v]
for u, v in connections if flows[u, v] > 0)

```

B. Penggunaan dan Analisis Hasil

Penggunaan program yang sudah dibuat dapat dengan memasukkan data input dari user untuk mensimulasikan keadaan produksi industry yang lebih realistis. Pengguna dapat melakukan penyiapan data terlebih dahulu.

Data yang digunakan mencakup:

1. Simpul (Node), sebagai representasi dari :
 - Sumber energi (power plant).
 - Titik distribusi (substation).
 - Beban energi (mesin produksi).
2. Sisi (Edge), sebagai representasi dari :
 - Jalur distribusi yang menghubungkan simpul-simpul tersebut.
 - Resistansi pada setiap sisi dihitung berdasarkan panjang dan material kabel.
3. Permintaan Energi:

Setiap simpul memiliki kebutuhan energi berbeda berdasarkan kapasitas produksinya.

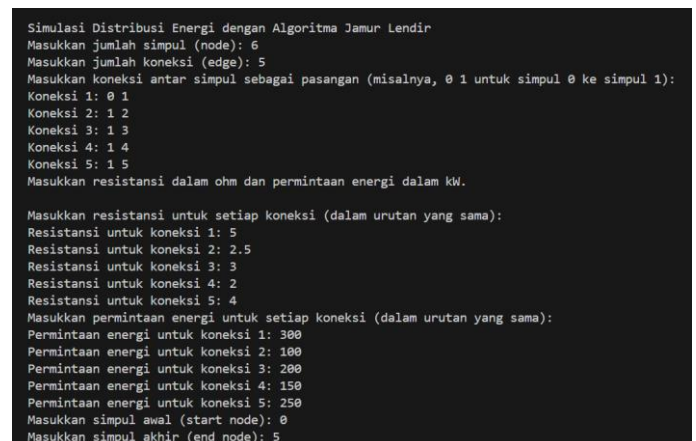
Contoh data yang diinputkan dalam program simulasi :

```

num_nodes = 6
connections = [(0, 1), (1, 2), (1, 3), (1,
4), (1, 5)]
resistances = [5.0, 2.5, 3.0, 2.0, 4.0]
demands = [300, 100, 200, 150, 250]
start = 0
end = 5

```

Dalam program pengguna memasukkan data sesuai dengan tampilan berikut sebelum nantinya program mengoperasikan data-data tersebut.



```

Simulasi Distribusi Energi dengan Algoritma Jamur Lendir
Masukkan jumlah simpul (node): 6
Masukkan jumlah koneksi (edge): 5
Masukkan koneksi antar simpul sebagai pasangan (misalnya, 0 1 untuk simpul 0 ke simpul 1):
Koneksi 1: 0 1
Koneksi 2: 1 2
Koneksi 3: 1 3
Koneksi 4: 1 4
Koneksi 5: 1 5
Masukkan resistansi dalam ohm dan permintaan energi dalam kW.

Masukkan resistansi untuk setiap koneksi (dalam urutan yang sama):
Resistansi untuk koneksi 1: 5
Resistansi untuk koneksi 2: 2.5
Resistansi untuk koneksi 3: 3
Resistansi untuk koneksi 4: 2
Resistansi untuk koneksi 5: 4
Masukkan permintaan energi untuk setiap koneksi (dalam urutan yang sama):
Permintaan energi untuk koneksi 1: 300
Permintaan energi untuk koneksi 2: 100
Permintaan energi untuk koneksi 3: 200
Permintaan energi untuk koneksi 4: 150
Permintaan energi untuk koneksi 5: 250
Masukkan simpul awal (start node): 0
Masukkan simpul akhir (end node): 5

```

Gambar 4 Inputan ke Program

Sumber : Arsip Pribadi

Maka selanjutnya program akan menjalankan simulasi dari data yang telah diinputkan tersebut dan akan menghasilkan hasil sebagai berikut :

Jalur terbaik atau jalur untuk membentuk graf yang optimal :

Jalur Terbaik: [0, 1, 4, 2, 3, 5]

Gambar 5 Jalur Terbaik
Sumber : Arsip Pribadi

Biaya total distribusi yang dihasilkan :

Biaya Total: 1000.00 Rp/kWh (total biaya energi untuk memenuhi permintaan).

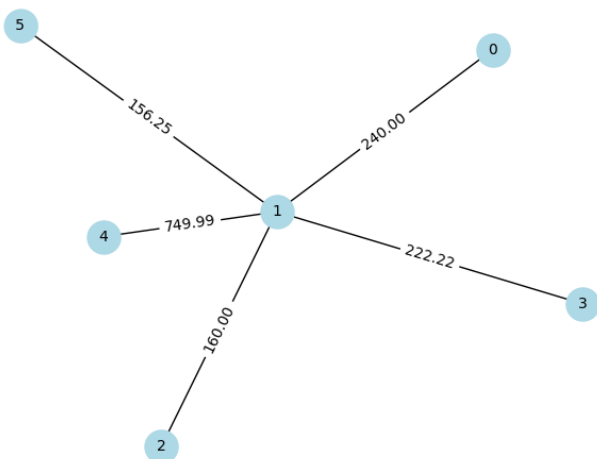
Gambar 6 Biaya Total
Sumber : Arsip Pribadi

Matriks feromon :

```
Matriks Feromon:
[[1.42724769e-05 2.39996589e+02 1.42724769e-05 1.42724769e-05
 1.42724769e-05 1.42724769e-05]
 [2.39996589e+02 1.42724769e-05 1.59997731e+02 2.22219065e+02
 7.49989310e+02 1.56247784e+02]
 [1.42724769e-05 1.59997731e+02 1.42724769e-05 1.42724769e-05
 1.42724769e-05 1.42724769e-05]
 [1.42724769e-05 2.22219065e+02 1.42724769e-05 1.42724769e-05
 1.42724769e-05 1.42724769e-05]
 [1.42724769e-05 7.49989310e+02 1.42724769e-05 1.42724769e-05
 1.42724769e-05 1.42724769e-05]
 [1.42724769e-05 1.56247784e+02 1.42724769e-05 1.42724769e-05
 1.42724769e-05 1.42724769e-05]]
```

Gambar 7 Matriks Feromon
Sumber : Arsip Pribadi

Graf yang terbentuk :



Gambar 8 Graf Hasil Simulasi
Sumber : Arsip Pribadi

Analisis Matriks Feromon

Matriks feromon menunjukkan tingkat kepercayaan algoritma terhadap efisiensi jalur. Jalur dengan tingkat feromon

tinggi, seperti [1 → 4] (749.99), sering digunakan karena efisiensi tinggi. Sebaliknya, jalur seperti [3 → 5], dengan feromon sedang, jalur tersebut digunakan sebagai jalur pendukung dalam distribusi energi. Berikut adalah penjelasan lanjutannya :

Baris 0:

- Jalur dari simpul 0 (sumber energi) ke simpul 1 memiliki tingkat feromon tinggi: 239.996589.
- Jalur lainnya (ke simpul 2, 3, 4, 5) memiliki tingkat feromon sangat rendah ($1.42724769 \times 10^{-5}$), mengindikasikan jalur ini jarang digunakan.

Baris 1:

- Simpul 1 adalah substation utama, sehingga terhubung dengan banyak simpul lain:
 - Jalur [1 → 4] memiliki tingkat feromon tertinggi: 749.989310749.989310749.989310.
 - Jalur [1 → 3] dan [1 → 2] juga memiliki feromon tinggi (222.219065 dan 159.997731), tetapi lebih rendah dibanding [1 → 4].
- Simpul 1 adalah substation utama, sehingga terhubung dengan banyak simpul lain:
- Jalur [1 → 4] memiliki tingkat feromon tertinggi: 749.989310.
- Jalur [1 → 3] dan [1 → 2] juga memiliki feromon tinggi (222.219065 dan 159.997731), tetapi lebih rendah dibanding [1 → 4].

Baris 4:

Jalur dari simpul 4 ke simpul 1 memiliki tingkat feromon sangat tinggi (749.989310), menunjukkan jalur ini sering digunakan.

Baris 5:

Simpul 5 memiliki feromon tinggi untuk jalur [5 → 1], mencerminkan koneksi kembali ke simpul utama setelah energi dialirkan.

Analisis Graf dan Pemilihan Jalur Terbaik

Simpul-simpul utama yang mencakup :

- Simpul 1 adalah substation utama, sehingga terhubung dengan banyak simpul lain.
- Jalur [1 → 4] memiliki tingkat feromon tertinggi: 749.99.
- Jalur [1 → 3] dan [1 → 2] juga memiliki feromon tinggi (222.22 dan 160.00), tetapi lebih rendah dibanding [1 → 4].

Jalur utama dari sumber energi ke simpul substation, yakni [0 → 1], memiliki tingkat feromon 240.00 yang menunjukkan jalur ini sering digunakan untuk memulai distribusi. Kemudian pada jalur [1 → 4], memiliki tingkat feromon tertinggi yaitu 749.99 dalam graf (pembulatan dari matriks feromon). Jalur [1 → 4] menjadi prioritas utama untuk mendistribusikan energi karena efisiensi yang tinggi (resistansi rendah dan permintaan energi

besar di simpul 4). Sementara jalur lain seperti $[4 \rightarrow 2]$, $[2 \rightarrow 3]$, dan $[3 \rightarrow 5]$, merupakan jalur yang melengkapi distribusi energi ke semua mesin produksi, tetapi memiliki tingkat feromon relatif lebih rendah dibanding jalur $[1 \rightarrow 4]$, tetapi cukup untuk dipilih dalam jalur global.

Substation (simpul 1) adalah penghubung utama antara sumber energi (simpul 0) dan mesin produksi lainnya. Simpul ini mendistribusikan energi secara global dengan prioritas pada jalur yang efisien (seperti $[1 \rightarrow 4]$). Meskipun simpul 1 terhubung langsung ke semua mesin produksi, algoritma SMA akan memilih jalur yang efisien secara global, bukan hanya secara lokal. Selain itu, resistansi pada jalur langsung mungkin lebih tinggi dibandingkan jalur yang melibatkan simpul lain (seperti simpul 4). Permintaan energi di simpul 4 lebih besar akibatnya jalur $[1 \rightarrow 4]$ lebih diprioritaskan.

Tingkat feromon mencerminkan seberapa sering jalur digunakan dalam iterasi simulasi. Oleh karena itu, jalur dengan feromon tinggi, seperti $[1 \rightarrow 4]$, menjadi prioritas utama, sedangkan jalur dengan feromon sedang, seperti $[4 \rightarrow 2]$ atau $[3 \rightarrow 5]$, melengkapi rute distribusi energi secara efisien. Jalur terbaik mendapatkan penguatan feromon lebih besar karena sering digunakan. Jalur seperti $[1 \rightarrow 4]$ diperkuat secara signifikan karena efisiensinya dalam distribusi energi. Jalur langsung lain, seperti $[1 \rightarrow 2]$, memiliki tingkat feromon lebih rendah karena jarang digunakan.

Algoritma memilih jalur terbaik berdasarkan kombinasi tingkat feromon dan efisiensi energi secara global. Jalur $[0 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 5]$ mencerminkan prioritas terhadap jalur dengan resistansi rendah (seperti $[1 \rightarrow 4]$) dan jalur dengan permintaan energi besar. Tingkat feromon tinggi memperkuat keputusan ini selama iterasi simulasi.

Analisis Total Biaya (Total Cost)

Dalam konteks distribusi energi industri, total cost menggambarkan pengeluaran energi untuk mengalirkan daya dari substation ke mesin produksi.

Semakin rendah nilai resistansi (R_{ij}) dan semakin efisien distribusi energi (Q_{ij}), maka semakin kecil total cost yang dihasilkan.

Dalam jalur terbaik $[0 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 5]$, biaya total dihitung dengan menjumlahkan kontribusi dari setiap sisi dalam jalur tersebut:

$$\text{Total Cost} = Q_{01} \cdot R_{01} + Q_{14} \cdot R_{14} + Q_{42} \cdot R_{42} + Q_{23} \cdot R_{23} + Q_{35} \cdot R_{35}$$

Hasil simulasi menunjukkan bahwa jalur ini tidak hanya memenuhi kebutuhan energi tetapi juga mengoptimalkan pengeluaran energi.

V. KESIMPULAN

Berdasarkan teori dan implementasi kode yang ada dalam makalah ini, dapat disimpulkan bahwa algoritma jamur lendir (Slime Mold Algorithm - SMA) merupakan solusi efektif untuk mengoptimalkan distribusi energi dalam jaringan industri yang kompleks. Dengan menggunakan representasi graf berbobot, algoritma ini dapat menemukan jalur optimal yang memenuhi kebutuhan energi mesin produksi sambil meminimalkan total

biaya distribusi.

Simulasi yang dilakukan menghasilkan beberapa temuan utama di antaranya adalah efisiensi jalur, yaitu jalur optimal yang dipilih berdasarkan tingkat feromon tinggi mencerminkan efisiensi distribusi energi, dengan jalur prioritas yang memiliki resistansi rendah dan permintaan energi besar, adaptabilitas, yakni bahwa algoritma mampu menyesuaikan jalur distribusi dengan perubahan permintaan energi di simpul, menunjukkan potensi penerapan pada skenario industri yang dinamis, dan aplikasinya untuk industri, algoritma ini relevan untuk kebutuhan manajemen energi di pabrik besar, terutama dalam memastikan distribusi energi yang efisien dengan biaya yang terkendali.

VI. UCAPAN TERIMA KASIH

Pertama-tama, penulis mengungkapkan rasa syukur dan mengaturnya puji yang setinggi-tingginya kepada Allah SWT yang dengan rahmat-Nya, penulis mendapatkan kemudahan dan kelancaran dalam menyelesaikan makalah ini dengan baik.

Penulis mengucapkan terima kasih kepada Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, atas dukungan dan fasilitas yang telah diberikan.

Ucapan terima kasih juga ditujukan kepada dosen pengampu mata kuliah Matematika Diskrit kelas K01, Bapak Dr. Ir. Rinaldi Munir yang dengan bimbingannya selama perkuliahan, penulis mampu menyelesaikan makalah ini dengan baik.

Terakhir, ucapan terima kasih ditujukan kepada orang tua dan rekan-rekan penulis yang memberikan semangat dan dukungan selama pengerjaan makalah ini.

REFERENSI

- [1] A. Adamatzky, "If BZ medium did spanning trees these would be the same as Physarum's," *Physics Letters A*, vol. 373, no. 10, pp. 952–956, 2009.
- [2] D. Tero, R. Kobayashi, dan T. Nakagaki, "A mathematical model for adaptive transport network in path finding by true slime mold," *Journal of Theoretical Biology*, vol. 244, no. 4, pp. 553–564, 2007.
- [3] L. Zhang, Y. Zhao, and Z. Han, "A Novel Slime Mold Algorithm for Network Optimization," in *Proceedings of the International Conference on Computational Intelligence and Communication Networks (CICN)*, 2018, pp. 10–14.
- [4] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA, USA: MIT Press, 2004.
- [5] MDPI, "Graph-Based Energy Distribution Models with Physarum Polycephalum," [Online]. <https://www.mdpi.com>. [Diakses: Des. 25, 2024]
- [6] NetworkX Developers, "NetworkX Documentation," [Online]. <https://networkx.github.io/documentation/stable/>. [Diakses: Des. 25, 2024].
- [7] R. Munir, "Graf: Algoritma DFS dan BFS," [Online]. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/22-Graf-Bagian3-2024.pdf>. [Diakses: Des. 25, 2024].
- [8] R. Munir, "Graf: Definisi, Elemen, dan Representasi," [Online]. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>. [Diakses: Des. 25, 2024].
- [9] R. Munir, "Graf: Struktur dan Operasi," [Online]. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/21-Graf-Bagian2-2024.pdf>. [Diakses: Des. 25, 2024].
- [10] R. Munir, "Pohon: Algoritma Minimum Spanning Tree," [Online]. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/24-Pohon-Bag2-2024.pdf>. [Diakses: Des. 25, 2024].
- [11] R. Munir, "Pohon: Definisi dan Representasi," [Online]. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf>. [Diakses: Dec. 25, 2024].

- [12] Springer, "Microgrid Optimization Using Bio-Inspired Algorithms," [Online]. <https://link.springer.com>. [Diakses: Des. 25, 2024].
- [13] X. Zhang, J. Liu, and Z. Xie, "Energy Optimization Using Slime Mold Algorithms in Industrial Systems," IEEE Transactions on Industrial Informatics, vol. 17, no. 3, pp. 1871–1882, 2021.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 26 Desember 2024



Ferdin Arsenarendra Purtadi 13523117