

**PENENTUAN LINTASAN OPTIMAL DISTRIBUSI BARANG  
MENGUNAKAN *HYPERGRAPH - PARTITIONING* DAN  
ALGORITMA GENETIKA**

**Fitriani Halimatus Sadiyyah**

Program Studi Matematika, Universitas Pendidikan Indonesia, Indonesia

**Kartika Yulianti\***

Program Studi Matematika, Universitas Pendidikan Indonesia, Indonesia  
kartika.yulianti@upi.edu

**Ririn Sispiyati**

Program Studi Matematika, Universitas Pendidikan Indonesia, Indonesia

**ABSTRACT.** *Efficient distribution of goods is critical in logistics management, which requires the selection of optimal distribution paths to achieve delivery targets with minimal total distance. This research combines Hypergraph-Partitioning and genetic algorithm to determine the optimal distribution path of goods to several customers. The Hypergraph-Partitioning divides the goods to be distributed equally to several vehicles, while the genetic algorithm is applied to determine the best distribution path in each partition. The results showed that the Hypergraph-Partitioning method successfully divided 62 customers into two partitions. The first partition serves 31 customers with a total demand of 865 loaves of bread, while the second partition also serves 31 customers with a total demand of 1,035 loaves of bread. The genetic algorithm was then used to find the shortest path for each partition, resulting in an efficient distribution solution.*

**Keywords:** *Genetic Algorithm, Hypergraph-Partitioning, Shortest Path,*

**ABSTRAK.** Distribusi barang yang efisien merupakan kunci dalam manajemen logistik, yang memerlukan pemilihan jalur distribusi optimal untuk mencapai target pengiriman dengan total jarak minimal. Penelitian ini menggabungkan metode *hypergraph-partitioning* dan Algoritma Genetika untuk menentukan jalur distribusi barang yang optimal. *Hypergraph-partitioning* digunakan untuk membagi barang yang akan didistribusikan secara seimbang ke beberapa kendaraan, sementara Algoritma Genetika diterapkan untuk menentukan lintasan distribusi terbaik di setiap partisi. Hasil penelitian menunjukkan bahwa metode *hypergraph-partitioning* berhasil membagi 62 pelanggan ke dalam dua partisi. Partisi pertama melayani 31 pelanggan yang harus dikunjungi dengan total permintaan sebanyak 865 buah roti, sedangkan partisi kedua juga melayani 31 pelanggan yang harus dikunjungi dengan total permintaan sebanyak 1.035 buah roti. Algoritma genetika kemudian digunakan untuk menemukan jalur terpendek untuk masing-masing partisi, sehingga menghasilkan solusi distribusi yang efisien.

**Kata Kunci:** Algoritma Genetika, *Hypergraph-Partitioning*, Lintasan Terpendek.

---

\*Penulis Korespondensi

Info Artikel : dikirim 25 Oktober 2024; direvisi 10 Desember 2024; diterima 18 Desember 2024.

## 1. PENDAHULUAN

Pendistribusian merupakan salah satu aktivitas utama yang dilakukan oleh perusahaan yang bergerak di bidang barang maupun jasa. Dalam proses ini, distributor harus memilih jalur distribusi yang tepat untuk mencapai target pengiriman dengan jarak tempuh terpendek. Mencari lintasan terpendek sangat penting karena dapat meningkatkan efisiensi pengiriman dengan mengurangi jarak tempuh. Hal ini berkontribusi pada penghematan waktu dan bahan bakar, yang pada akhirnya mengurangi biaya operasional (Amroni, Rhohman, & Wulanningrum, 2017).

Berbagai sistem dan algoritma optimasi dapat digunakan untuk menyelesaikan masalah ini. Namun, dalam praktiknya, jumlah dan kapasitas kendaraan biasanya terbatas, sehingga proses pencarian solusi perlu mempertimbangkan kapasitas kendaraan yang ada. Penelitian sebelumnya oleh Rahman (2018) menunjukkan bahwa masalah distribusi barang dengan kendala kapasitas kendaraan dapat diselesaikan menggunakan *Capacitated Vehicle Routing Problem (CVRP)*. CVRP bertujuan untuk menentukan rute optimal bagi setiap kendaraan dari depot ke pelanggan, dengan ketentuan bahwa total permintaan pelanggan harus kurang dari kapasitas kendaraan. Jika permintaan melebihi kapasitas, pelanggan tersebut dilayani oleh kendaraan lain. Meskipun metode ini efektif dalam membagi jumlah barang berdasarkan jarak optimal, CVRP tidak mempertimbangkan keseimbangan muatan di antara kendaraan.

Dalam kasus yang dijumpai pada pabrik roti di Kabupaten Cianjur, distribusi barang ingin diatur secara seimbang antara setiap kendaraan sambil tetap mempertimbangkan jarak antar pelanggan. Menurut Kristianto dan Swanjaya (2020), *Hypergraph-Partitioning* mampu membantu mengelompokkan data berdasarkan jarak terdekat untuk mengatasi masalah tersebut. Dalam penelitian yang dilakukan oleh Kristianto, Swanjaya, dan Pamungkas (2020), *Hypergraph-Partitioning* digunakan untuk mengelompokkan data dengan mempertimbangkan jumlah dan kapasitas kendaraan yang tersedia. Kemudian, algoritma *Branch & Bound* digunakan untuk mencari lintasan paling optimal. Penelitian tersebut

menunjukkan bahwa algoritma tersebut mampu menghasilkan pengelompokan data pelanggan yang seimbang berdasarkan jarak terdekat dan waktu tercepat.

Namun, meskipun metode *Branch & Bound* terbukti lebih unggul dibandingkan beberapa metode lain seperti *Cheapest Insertion Heuristic* (Nababan, 2021), metode ini memiliki kelemahan berupa waktu komputasi yang lama untuk data berukuran besar (Alallah, 2021). Sebaliknya, Algoritma genetika menawarkan waktu komputasi yang lebih stabil, bahkan untuk jumlah data yang besar (Ramadhania & Rani, 2021). Penelitian Saputro dan Wijaya (2006) menunjukkan bahwa waktu pencarian solusi menggunakan algoritma *Branch & Bound* meningkat secara eksponensial seiring bertambahnya jumlah titik pelanggan. Oleh karena itu, Algoritma Genetika lebih cocok digunakan ketika waktu pencarian terbatas namun jumlah kota sangat besar. Hal ini mengindikasikan bahwa Algoritma Genetika lebih efektif dalam menentukan rute terpendek untuk pengiriman barang. Oleh karena itu, dalam penelitian ini, Algoritma Genetika dipilih sebagai metode optimasi untuk menyelesaikan masalah penentuan jalur distribusi barang yang optimal, mengingat data yang digunakan cukup besar yaitu 62 titik pelanggan.

Penelitian ini bertujuan untuk membentuk lintasan yang optimal dengan mempertimbangkan jumlah dan kapasitas kendaraan, membagi data pelanggan ke dalam partisi yang mewakili kendaraan distribusi yang beroperasi, serta menentukan lintasan paling optimal untuk setiap kendaraan agar setiap pelanggan dapat dikunjungi. Tujuan akhir dari penelitian ini adalah untuk mengoptimalkan pembagian muatan pada tiap kendaraan yang beroperasi dan meminimalkan jarak tempuh.

## **2. METODE PENELITIAN**

Data yang digunakan dalam penelitian ini diambil dari pabrik roti yang terletak di Kabupaten Cianjur. Data yang dikumpulkan mencakup lokasi pelanggan dan jarak antar lokasi, jumlah permintaan dari masing-masing pelanggan, serta jumlah dan kapasitas kendaraan yang akan digunakan untuk proses distribusi. Metode penyelesaian yang digunakan pada penelitian ini adalah

*Hypergraph-Partitioning* dan Algoritma Genetika. Pembagian pengiriman untuk setiap kendaraan akan diselesaikan dengan menggunakan *Hypergraph-Partitioning*, selanjutnya metode optimasi untuk penentuan rute yang akan dilalui oleh setiap kendaraan akan diselesaikan dengan menggunakan Algoritma Genetika.

### 2.1 *Hypergraph-Partitioning*

Menurut Kristianto dan Swanjaya (2020), *hypergraph* adalah bentuk *graph* di mana setiap sisinya dapat menghubungkan dua atau lebih simpul yang disebut *hyperedge*. Sedangkan *Hypergraph-Partitioning* adalah proses untuk membagi *hypergraph* menjadi sub-*hypergraph* atau *graph* yang lebih kecil. Proses *partitioning* ini membagi *hypergraph* menjadi beberapa partisi sehingga himpunan partisinya adalah  $P = \{S_1, S_2, S_3, \dots, S_k\}$ , di mana  $S_i$  merupakan himpunan yang berisi simpul-simpul yang dikelompokkan pada  $S_i$  (Kristianto & Swanjaya, 2020). Tujuan dari *Hypergraph-Partitioning* pada penelitian ini adalah untuk memaksimalkan *cut size* dan menjaga keseimbangan kapasitas (*capacity balance*) sehingga diperoleh pembagian pelanggan untuk setiap kendaraan.

Menurut Kristianto dan Swanjaya (2020), tahapan dari *Hypergraph-Partitioning* terdiri dari 3 fase. Fase pertama yaitu *coarsening*, fase ini bertujuan untuk menyederhanakan *hypergraph* dengan mengurangi kompleksitasnya sehingga menghasilkan bentuk partisi awal yang akan digunakan dalam proses *balancing*. Fase kedua yaitu *initial partition/balancing*, fase ini bertujuan mengoptimalkan pembagian graf menjadi dua partisi dengan mempertimbangkan bobot sisi sehingga dapat memaksimalkan jarak antar partisi. Algoritma yang digunakan pada proses *balancing* dalam penelitian ini adalah algoritma *Fiduccia-Mattheyses* (FM), di mana setiap simpul yang telah berada pada partisi awal dievaluasi nilai gain-nya sehingga nilai gain bernilai non-negatif. Nilai gain tiap simpul dihitung dengan rumus berikut:

$$\Delta g = FS - TE \quad (1)$$

di mana  $FS(v)$  merupakan jumlah bobot dari sisi yang menghubungkan simpul  $v$  dengan simpul-simpul yang ada di luar partisi dimana simpul  $v$  berada, dan  $TE(v)$

merupakan jumlah bobot dari *edge* yang menghubungkan simpul  $v$  dengan simpul-simpul yang ada di dalam partisi dimana simpul  $v$  berada.

Terakhir fase *uncoarsening*, pada tahap ini *supernodes* hasil dari fase *coarsening* dibuka kembali sambil mengoptimasi kelompok-kelompok hasil proses *coarsening*, sehingga setiap partisi memiliki jumlah permintaan yang kurang dari kapasitas kendaraan yang ada.

## 2.2 Algoritma Genetika

Menurut Rahman (2018), Algoritma Genetika adalah optimasi yang terinspirasi dari seleksi alam dan gen. Mekanisme pencarian solusi pada Algoritma Genetika didasarkan pada mekanisme sistem natural yakni genetik dan seleksi alam. Algoritma ini bekerja dengan sebuah populasi yang terdiri dari individu-individu, yang masing-masing individu merepresentasikan solusi yang mungkin bagi persoalan yang ada. Dalam kaitan ini, individu dilambangkan dengan sebuah nilai *fitness* yang akan digunakan untuk mencari solusi terbaik dari persoalan yang ada.

Dalam proses pencarian lintasan terpendek menggunakan Algoritma Genetika memiliki langkah-langkah sebagai berikut (Putri, 2014):

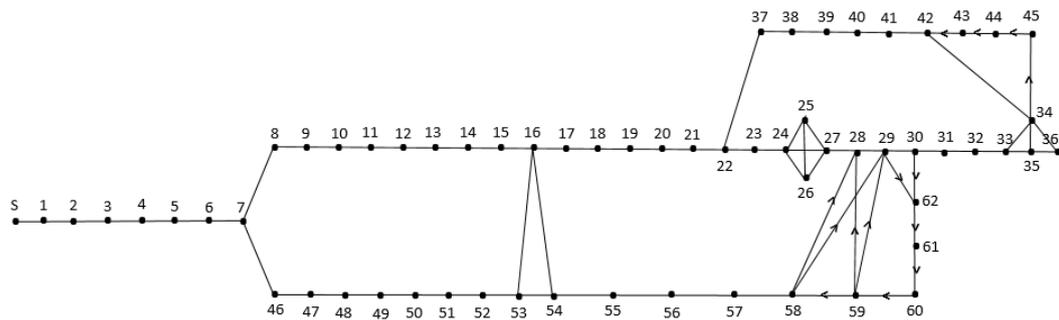
0. Inisialisasi parameter awal meliputi ukuran populasi (*pop-size*), jumlah generasi, laju *crossover* (*crossover rate*, Cr), dan laju mutasi (*mutation rate*, Mr).
1. Bangkitkan populasi awal sebanyak ukuran populasi (*pop-size*).
2. Lakukan *crossover* pada individu yang terpilih sebagai *parent* sesuai dengan Cr yang telah ditetapkan. Metode *crossover* yang digunakan pada penelitian ini adalah *Partially-Mapped Crossover (PMX)*.
3. Lakukan mutasi pada individu yang terpilih sebagai *parent* sesuai dengan Mr yang sudah ditentukan. Metode mutasi yang digunakan pada penelitian ini adalah *Reciprocal Exchange Mutation*.
4. Hitung nilai *fitness* dari setiap kromosom. Nilai *fitness* dihitung dengan rumus berikut:

$$fitness = \frac{1}{total\ jarak} \quad (2)$$

5. Lakukan seleksi elitis untuk memilih individu terbaik berdasarkan nilai *fitness* yang akan dipertahankan dan diteruskan ke generasi berikutnya.
6. Langkah 2 hingga 5 dilakukan sebanyak jumlah generasi. Solusi dengan nilai *fitness* tertinggi diambil sebagai solusi terbaik.

### 3. HASIL DAN PEMBAHASAN

Data yang digunakan pada penelitian ini berasal dari pabrik roti “Bintang” yang berlokasi di Kabupaten Cianjur. Data yang digunakan berasal dari 62 pelanggan di wilayah Cianjur dengan jumlah permintaan sebanyak 1.900 roti. Jumlah ini masih di bawah kapasitas maksimal dua unit motor, yang memiliki total kapasitas 2.200 roti, sehingga pengantaran dapat dilakukan oleh dua motor dengan masing-masing kapasitas 1.100 roti. Graf yang merepresentasi kasus ini ditunjukkan pada Gambar 1 berikut.



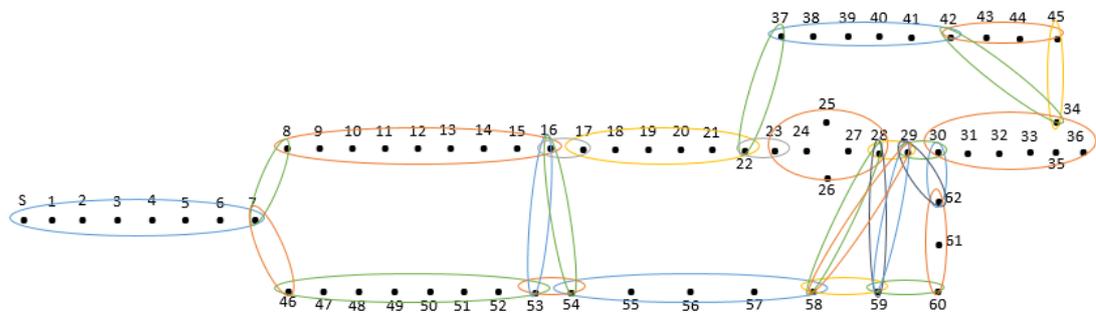
**Gambar 1.** Representasi Kasus dalam Bentuk Graf Berarah

Metode penyelesaian dilakukan dengan menggunakan bantuan *microsoft excel* untuk proses *Hypergraph-Partitioning* dan bahasa pemrograman *python* untuk proses Algoritma Genetika.

#### 3.1 *Hypergraph-Partitioning*

Pada program *Hypergraph-Partitioning*, urutan sisi tidak diperhatikan, sehingga AB dianggap sama dengan BA. Proses *partitioning* ini bertujuan untuk membagi graf menjadi dua partisi tanpa mempertimbangkan urutan pengiriman. Oleh karena itu, graf yang digunakan pada proses ini adalah graf tak berarah dengan bobot yang didasarkan pada jarak terpendek antara titik-titik.

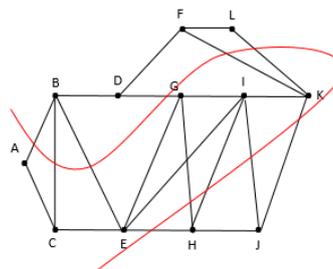
Graf yang merepresentasikan kasus kemudian diubah menjadi *hypergraph* dengan mempertimbangkan kriteria yang akan digunakan dalam *Hypergraph-Partitioning*. Kriteria yang digunakan meliputi maksimasi potongan (*cut size*) dan keseimbangan kapasitas (*capacity balance*). Untuk memenuhi kriteria maksimasi *cut size*, dilakukan penyesuaian dalam representasi *hypergraph* dengan mengelompokkan titik-titik yang hanya memiliki satu jalur dalam satu *hyperedge*. Langkah ini bertujuan agar titik-titik yang pasti akan dilewati secara berurutan dapat dikirimkan oleh kendaraan yang sama. Representasi *hypergraph* dari Gambar 1 ditunjukkan pada Gambar 2 berikut.



**Gambar 2.** Representasi Kasus dalam Bentuk Hypergraph

### 3.1.1 Coarsening

Pada tahap *coarsening*, *hypergraph* disederhanakan menjadi *coarsened hypergraph*. Titik-titik yang berada pada *hyperedge* yang sama, sebagaimana ditunjukkan pada Gambar 2, kemudian digabungkan menjadi *supernode*. Selanjutnya algoritma *Maxima Matching (Greedy)* dijalankan sehingga menghasilkan partisi awal seperti yang ditunjukkan pada Gambar 3 berikut.



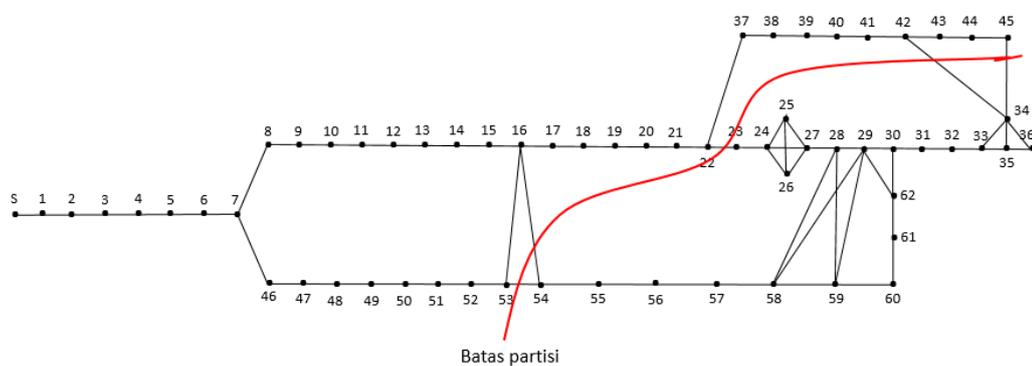
**Gambar 3.** Partisi Awal Hasil Tahap Coarsening

### 3.1.2 *Balancing*

Pada tahap *balancing*, algoritma FM dijalankan, di mana setiap simpul yang telah berada pada partisi awal dievaluasi nilai gain-nya dengan menggunakan rumus (1) sehingga semua nilai gain bernilai non-negatif. Hasil dari proses *balancing* ini diperoleh partisi baru yang lebih baik, di mana setiap titik telah memiliki nilai gain non-negatif. Partisi baru tersebut kemudian akan digunakan sebagai partisi awal pada proses *uncoarsening*.

### 3.1.3 *Uncoarsening*

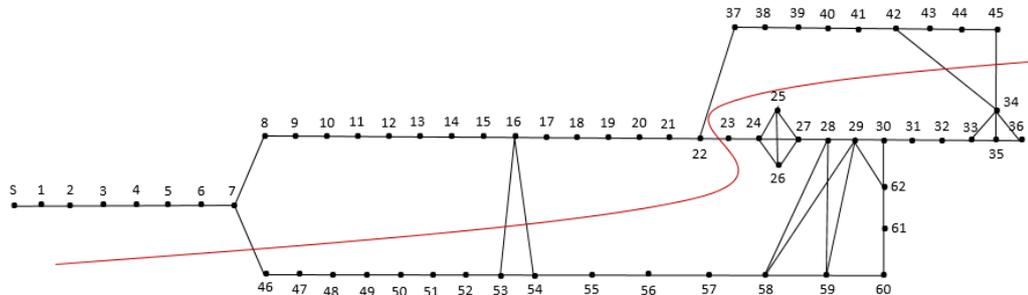
Setelah proses *balancing* selesai, *hypergraph* yang telah disederhanakan (*coarsened hypergraph*) dikembalikan ke bentuk aslinya (*original hypergraph*) sambil mengoptimasi kelompok-kelompok hasil proses *coarsening*, sehingga setiap partisi memiliki jumlah permintaan yang kurang dari kapasitas kendaraan yang tersedia. Hasil dari mengembalikan *coarsened hypergraph* tersebut kemudian menjadi *hypergraph* yang digunakan pada proses *uncoarsening*, di mana titik-titik yang sebelumnya diwakili oleh suatu *supernode*, akan berada pada partisi yang sesuai dengan *supernode* tersebut. Partisi awal *hypergraph* yang digunakan dalam proses ini ditunjukkan pada Gambar 4 berikut.



**Gambar 4.** Partisi Awal Tahap Uncoarsening

Selanjutnya, algoritma FM dijalankan dengan memperhatikan kesesuaian jumlah muatan pada setiap partisi. Hasil partisi yang diperoleh dari proses *Hypergraph-Partitioning* adalah  $P1 = \{1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 37, 38, 39, 40, 41, 42, 43, 44, 45\}$  dan  $P2 = \{23, 24, 25,$

26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62}. Hasil partisi ditunjukkan pada Gambar 3 berikut.



**Gambar 5.** Hasil Partisi

Titik-titik pada partisi pertama (P1) akan dilayani oleh kendaraan pertama, sedangkan titik-titik pada partisi kedua (P2) akan dilayani oleh kendaraan kedua. Kendaraan pertama melayani 31 pelanggan dengan mengangkut muatan sebanyak 865 buah roti, sedangkan kendaraan kedua melayani 31 pelanggan dengan membawa 1.035 buah roti. Hasil *partitioning* ini kemudian digunakan dalam Algoritma Genetika untuk mencari jalur paling optimal.

**3.2 Algoritma Genetika**

Sebelum melakukan proses Algoritma Genetika, dilakukan percobaan untuk penentuan parameter terlebih dahulu. Setelah melakukan pengujian terhadap parameter Algoritma Genetika, penulis menetapkan nilai setiap parameter sebagai berikut: nilai penalti = 6.000, ukuran populasi = 10.000, *crossover rate* = 0,7, *mutation rate* = 0,01, jumlah generasi = 100, dan jumlah Elit = 1.

Solusi terbaik dicari oleh program dengan menghitung nilai fitness dari kromosom pada populasi. Nilai fitness dihitung dengan menggunakan rumus (2), dimana kromosom dengan nilai fitness tertinggi akan dipilih sebagai solusi terbaik. Solusi terbaik yang diperoleh dari program Algoritma Genetika disajikan pada Tabel 1.

**Tabel 1.** Solusi Terbaik Hasil Program Algoritma Genetika

	<b>Solusi Terbaik</b>	<b>Nilai Fitness</b>
<b>Partisi 1</b>	[0, 5, 4, 3, 2, 1, 0]	1,144950767117014e-04
<b>Partisi 2</b>	[0, 6, 7, 8, 18, 3, 2, 1, 15, 12, 13, 14, 17, 16, 11, 10, 9, 4, 5, 0]	2,4165095935430864e-05

Pada partisi pertama, program mencapai konvergensi, menghasilkan solusi terbaik yang konsisten di setiap percobaan. Solusi terbaik yang diperoleh adalah [0, 5, 4, 3, 2, 1, 0]. Titik-titik dalam solusi ini adalah *supernode* yang telah didefinisikan sebelumnya. *Supernode* tersebut kemudian dikembalikan ke bentuk awal, sehingga solusi untuk partisi pertama menjadi [0, 1, 2, 3, 4, 5, 6, 7, 45, 44, 43, 42, 41, 40, 39, 38, 37, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 0] dengan total jarak pengiriman 8.734 meter.

Sebaliknya, pada partisi kedua, program tidak mencapai konvergensi, dan hasilnya berbeda di setiap percobaan. Solusi yang ditampilkan dalam tabel di atas adalah solusi dengan nilai *fitness* terbaik yang berhasil diperoleh, namun solusi ini masih menggunakan sisi yang menghubungkan simpul-simpul yang tidak saling bertetangga. Hal ini menunjukkan bahwa Algoritma Genetika kurang efektif untuk data yang memiliki banyak simpul yang tidak saling bertetangga. Banyaknya simpul yang tidak saling bertetangga membuat fungsi *fitness* menjadi kurang efektif, dan membutuhkan *running-time* lebih lama dibandingkan dengan graf lengkap atau memiliki banyak simpul saling bertetangga. Data set ini mengilustrasikan bagaimana Algoritma Genetika beroperasi dalam graf yang tidak lengkap, di mana pencarian solusi dilakukan secara acak. Penggunaan bilangan acak membuat program ini tidak menjamin solusi optimal. Namun, pemilihan parameter yang tepat dapat membantu dalam menemukan atau mendekati solusi optimal.

#### 4. KESIMPULAN DAN SARAN

*Hypergraph-Partitioning* berhasil diterapkan untuk membagi data pelanggan secara seimbang antara dua kendaraan. Proses ini melibatkan tahapan *coarsening*, *balancing*, dan *uncoarsening*, yang membagi 62 pelanggan secara merata: kendaraan pertama melayani 31 pelanggan dengan 865 buah roti, sedangkan kendaraan kedua melayani 31 pelanggan dengan 1.035 buah roti. Perbedaan muatan yang hanya 170 buah roti menunjukkan efektivitas metode ini dalam pembagian pengiriman.

Selanjutnya diterapkan algoritma genetika untuk menentukan rute terpendek distribusi barang pada setiap partisi. Pada partisi pertama, Algoritma Genetika mencapai konvergensi dan memberikan solusi optimal yang konsisten. Namun, pada partisi kedua, algoritma menghadapi keterbatasan dalam menemukan solusi optimal karena terdapat banyak simpul yang tidak bertetangga. Modifikasi atau tambahan strategi mungkin diperlukan untuk mengatasi tantangan ini dan meningkatkan efisiensi solusi.

### UCAPAN TERIMAKASIH

Penulis berterima kasih kepada pembimbing penelitian, pihak pabrik roti “Bintang”, Universitas Pendidikan Indonesia, serta seluruh pihak yang telah membantu terlaksananya penelitian ini.

### DAFTAR PUSTAKA

- Alallah, A. S. M., *Analisis Metode Branch and Bound pada Optimalisasi Laba Industri Tambang Batu Bata Putih*, Skripsi, Program Studi Matematika Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, 2021.
- Amroni, A. M., Rhozman, F., dan Wulanningrum, R., *Aplikasi Penentuan Rute Optimal Delivery Menggunakan Algoritma Dijkstra*, Proceedings of Seminar Nasional Inovasi dan Teknologi (SEMNASINOTEK), Fakultas Teknik Universitas Nusantara PGRI Kediri, **1**(1) (2017), 211-216.
- Kristianto, Y. H. dan Swanjaya, D., *Graph Clustering pada Pengelompokan Tujuan Distribusi Barang Berdasarkan Matriks Adjacency*, Proceedings of SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi), **4**(3) (2020), 125-130.
- Kristianto, Y. H., Swanjaya, D., dan Pamungkas, D. P., *Implementasi Hypergraph-Partitioning dan Branch & Bound untuk Pembentukan Lintasan Distribusi Barang*, Proceedings of Semnasinotek 2020.

- Nababan, D. O., *Perbandingan Hasil Perolehan Rute Minimum Menggunakan Metode Branch and Bound dan Cheapest Insertion Heuristic di Perjalanan Wilayah Kepulauan Riau*, Disertasi, Universitas Sumatera Utara, 2021.
- Rahman, L. N., *Implementasi Algoritma Genetika pada Capacitated Vehicle Routing Problem*, Tesis, Universitas Pendidikan Indonesia, Bandung, 2018.
- Ramadhania, S. E. dan Rani, S., *Implementasi Kombinasi Algoritma Genetika dan Tabu Search untuk Penyelesaian Travelling Salesman Problem*, AUTOMATA, **2**(1) (2021).
- Saputro, N. dan Wijaya, S., *Perbandingan Kinerja Algoritma Genetik dan Algoritma Branch and Bound pada Travelling Salesman Problem*, Proceedings of Conference: Indonesian Conference on Telecommunications, Bandung, Indonesia, September, 2006.
- Putri, F. B., *Penerapan Algoritma Genetika Untuk Vehicle Routing Problem with Time Window (VRPTW) Pada Kasus Optimasi Distribusi Beras Bersubsidi*, Skripsi, Universitas Brawijaya, Malang, 2014.