

Usulan Wilayah Distribusi Pedagang Kopi Keliling *Brand XYZ* untuk Memperluas Cakupan Permintaan Menggunakan MCLP dan *K-Means*

1st Golden Pratama
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
goldenpratama@student.telkomuniversity.ac.id

2nd Femi Yulianti
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
femiyulianti@telkomuniversity.ac.id

3rd Gisti Ayu Pratiwi
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
gistiayupratiwi@telkomuniversity.ac.id

Abstrak—Tren konsumsi kopi di Indonesia meningkat pesat seiring pergeseeran gaya hidup generasi muda, memunculkan model bisnis inovatif seperti kopi keliling. *Brand XYZ* sebagai salah satu pemain kopi keliling di Bandung menggunakan sepeda listrik untuk distribusi langsung ke konsumen. Namun, penentuan lokasi jualan masih berbasis asumsi individu, menyebabkan cakupan permintaan yang kurang optimal. Penelitian ini bertujuan mengusulkan wilayah distribusi yang dapat memaksimalkan cakupan permintaan menggunakan pendekatan *Maximal Covering Location Problem (MCLP)* dan *K-Means clustering*. Data permintaan diambil dari 20 titik lembaga pendidikan sebagai proksi permintaan tertinggi, dan 40 kandidat lokasi jualan ditentukan berdasarkan visibilitas, aksesibilitas, serta kedekatan dengan titik permintaan. Hasil model MCLP menunjukkan tercapainya cakupan 100% dengan hanya 15 lokasi jualan terpilih. *K-Means* kemudian membagi wilayah menjadi tiga cluster yang seimbang secara spasial (skor silhouette 0,4948), memudahkan alokasi pedagang. Implementasi model ini mendukung keputusan strategis brand dalam memperluas area distribusi berbasis data, meningkatkan efisiensi operasional dan potensi penjualan.

Kata kunci— *Kopi Keliling, MCLP, K-Means, Distribusi, Wilayah Operasional*

I. PENDAHULUAN

Konsumsi kopi di Indonesia terus meningkat dan menjadi bagian dari gaya hidup masyarakat, terutama di kalangan mahasiswa dan generasi muda. Konsep kopi keliling yang memanfaatkan armada sepeda listrik menjadi alternatif distribusi inovatif yang mudah diakses dan ramah lingkungan. *Brand XYZ* sebagai salah satu pelaku kopi keliling di Bandung masih menentukan lokasi jualannya berdasarkan preferensi individu pedagang, sehingga potensi cakupan permintaan tidak dimanfaatkan optimal. Berdasarkan observasi, banyak area dengan potensi permintaan tinggi, khususnya zona pendidikan, belum tersentuh secara sistematis. Penelitian ini bertujuan menyusun usulan wilayah distribusi optimal agar cakupan permintaan dapat dimaksimalkan melalui analisis spasial dan model optimasi.

II. KAJIAN TEORI

A. Distribusi & Transportasi

Distribusi merupakan salah satu kegiatan dalam rantai pasok (supply chain) yang berkaitan dengan kegiatan pemindahan produk dari produsen ke konsumen. Menurut Simchi-Levi dkk. (2008), distribusi adalah proses mengirimkan produk dan layanan ke pelanggan pada waktu dan lokasi yang tepat, yang berkontribusi dalam memberikan nilai kepada pelanggan. Transportasi merupakan salah satu faktor penting dalam kegiatan distribusi yang sangat berperan dalam menentukan efisiensi sistem logistik dan distribusi. Transportasi dalam konteks distribusi juga mendukung kecepatan pelayanan dan memperluas cakupan wilayah layanan ke konsumen akhir.

B. Facility Location Problem (FLP)

Penentuan lokasi fasilitas adalah salah satu permasalahan dalam distribusi yang disebut Facility Location Problem (FLP). Menurut Daskin (2013), FLP merupakan permasalahan yang bertujuan untuk memilih lokasi fasilitas agar dapat meminimalkan biaya atau memaksimalkan cakupan permintaan dengan mempertimbangkan jarak antara fasilitas dan pelanggan. Pemilihan lokasi yang optimal bertujuan untuk mendukung distribusi produk secara lebih efektif, memperpendek jarak tempuh, mengurangi biaya transportasi, serta meningkatkan pelayanan kepada pelanggan.

III. METODE

A. Maximal Covering Location Problem (MCLP)

Salah satu pendekatan dalam teori penentuan lokasi fasilitas adalah *Maximal Covering Location Problem (MCLP)*, yang bertujuan untuk memaksimalkan jumlah permintaan yang tercakup oleh fasilitas yang dibangun dalam batasan jumlah fasilitas dan radius layanan yang ditetapkan. Setelah diperkenalkan pertama kali oleh Church dan ReVelle pada tahun 1974, model ini masih sangat populer di banyak industri, termasuk distribusi barang, layanan darurat, dan penempatan fasilitas *retail*. Secara umum, MCLP digunakan

untuk menjawab pertanyaan berikut: “Dari sejumlah kandidat lokasi fasilitas, lokasi mana saja yang harus dipilih (maksimal n lokasi) agar dapat melayani sebanyak mungkin titik permintaan dalam radius tertentu?” Dalam konteks penelitian ini, MCLP digunakan untuk menentukan lokasi titik penjualan kopi keliling agar dapat menjangkau sebanyak mungkin wilayah target seperti perkantoran, sekolah, dan kampus.

$$x_k = \begin{cases} 1 & \text{jika fasilitas dibuka di titik } k \\ 0 & \text{sebaliknya} \end{cases}$$

A. Kerangka Berpikir

Gambar 1 ini menunjukkan alur keterkaitan antar variabel yang mendukung penjelasan serta penyelesaian pertanyaan penelitian.

$$\text{Maximize} \quad \sum_{i \in I} w_i y_i \quad (1)$$

$$\text{S.T} \quad \sum_{j \in S_i} x_j \geq y_i \quad \forall i \in I \quad (2)$$

$$\sum_{j \in J} x_j \leq P \quad (3)$$

$$x_j + x_k \leq 1 \quad \forall k \in N_j \quad (4)$$

$j \neq k$

$$x_j = 0 \text{ if } d_j^{depot} > D_{max} \quad \forall j \in J \quad (5)$$

$$x_j \in \{0,1\} \quad \forall j \in J \quad (6)$$

$$y_i \in \{0,1\} \quad \forall i \in I \quad (7)$$

Keterangan:

i = titik permintaan

j = titik kandidat lokasi jualan

I = himpunan titik permintaan

J = himpunan titik kandidat lokasi jualan

k = titik kandidat lain yang sedang dibandingkan dengan j

w_i = bobot permintaan di titik i

P = jumlah fasilitas yang ditetapkan

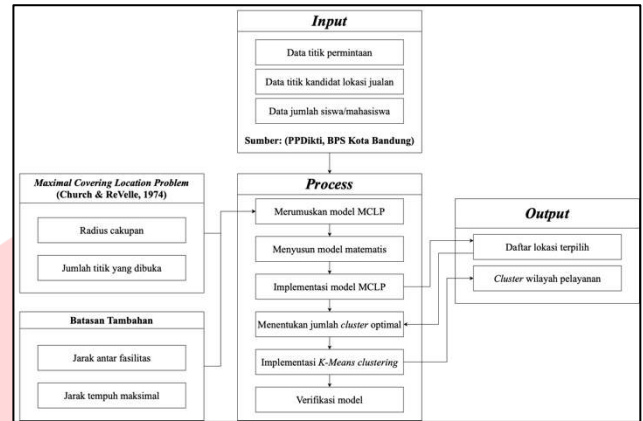
S_i = Himpunan kandidat titik yang terjangkau oleh i

d_j^{depot} = Jarak dari depot ke fasilitas j

D_{max} = Batas jarak maksimum

$$x_j = \begin{cases} 1 & \text{jika fasilitas dibuka di titik } j \\ 0 & \text{sebaliknya} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{jika permintaan di titik } i \text{ tercapai} \\ 0 & \text{sebaliknya} \end{cases}$$

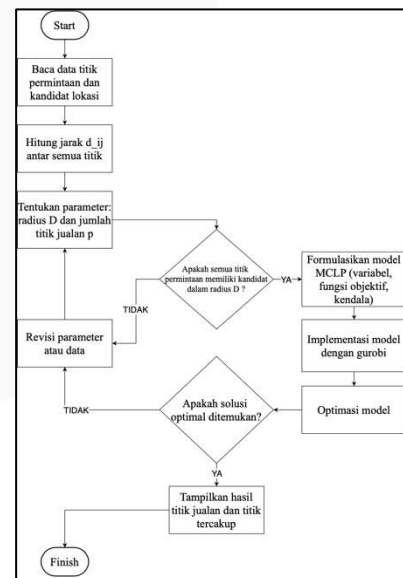


Gambar 1 Kerangka Berpikir

IV. HASIL DAN PEMBAHASAN

A. Alur Proses Pengolahan Data

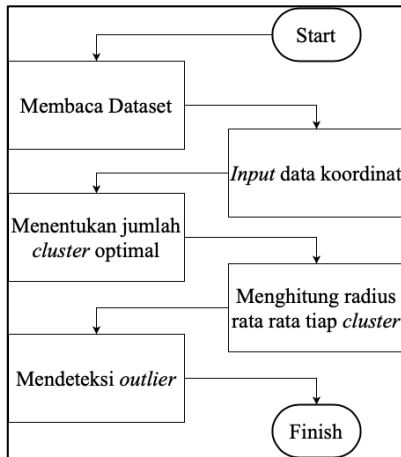
Untuk mempermudah pemahaman tentang cara menyelesaikan masalah dalam menentukan titik jualan yang paling baik dengan menggunakan pendekatan *Maximal Covering Location Problem* (MCLP), di bawah ini disajikan diagram alir yang menunjukkan proses pemodelan yang dilakukan.



Gambar 2 Flowchart MCLP

Setelah menentukan lokasi titik jualan terbaik dari model MCLP, langkah berikutnya adalah melakukan pengelompokan terhadap titik jualan yang telah dipilih. Tujuan dari pengelompokan ini adalah untuk memecah wilayah layanan menjadi beberapa kawasan yang lebih teratur, sehingga penugasan pedagang keliling dapat

dilakukan berdasarkan lokasi yang saling dekat dan mengurangi *overlapping* titik jualan.



Gambar 3 Flowchart K-Means

B. Penyelesaian Permasalahan

Pada tahap ini menjelaskan model matematis MCLP (Church & RaVelle, 1975) menggunakan bahasa Phyton pada *tools* VSCode serta pembagian wilayah menggunakan K-Means dengan penentuan nilai K optimal menggunakan *Silhouette Score*.

Langkah-Langkah Implementasi MCLP

1. input data titik permintaan dan lokasi kandidat

```

1 demand_coords = demand_points_df[['lon', 'lat']].values
2 demand_weights = demand_points_df['demand'].values
3 candidate_sites_data1 = candidate_df[['lon', 'lat']].values
4 candidate_sites_data2 = candidate2_df[['lon', 'lat']].values
  
```

Gambar 4 Inisialisasi titik koordinat

Data ini berasal dari *dataframe* yang masing-masing berisi informasi spasial dalam format *longitude* dan *latitude*. Data yang sudah diambil dari *dataframe* tersebut kemudian disimpan pada variabel '*demand_coords*'. Selain mengambil data titik koordinat, model juga perlu mengambil data *demand* atau bobot permintaan yang ada pada kolom *demand*. Selanjutnya data tersebut disimpan ke dalam variabel '*demand_weights*'.

2. Perhitungan jarak antar titik

```

1 D = distance_matrix(demand_coords, candidate_sites)
  
```

Gambar 5 Perhitungan matriks jarak

Selanjutnya dilakukan perhitungan jarak antara setiap titik permintaan dengan setiap titik kandidat lokasi jualan. Perhitungan menggunakan fungsi '*distance_matrix*' yang akan menghasilkan matriks jarak D.

3. Input parameter radius cakupan serta jumlah titik jualan

```

1 # === Konfigurasi & Jalankan ===
2 K1 = 15
3 K2 = 0
4 radius = 0.00631
5 min_facility_dist = 700
6 depot_coord = (-6.92651, 107.59575)
  
```

Gambar 6 Inisialisasi parameter

Parameter K1 dan K2 merupakan batas jumlah fasilitas yang dibuka atau merupakan nilai *P*. Kemudian ada parameter radius yang diatur sebesar 0,00631, jika dikonversi dalam satuan geografis setara dengan 700 meter. Parameter ini digunakan untuk menentukan cakupan area sebuah titik fasilitas/jualan, selain itu mencegah adanya pemusatan titik fasilitas/jualan yang berdekatan.

4. Apakah semua titik permintaan tercakup dalam radius D?

```

1 coverage = D <= radius
  
```

Gambar 7 Validasi cakupan

Menghitung semua titik permintaan dan kandidat lokasi dalam bentuk matriks yang sudah disimpan dalam variabel D. Selanjutnya dibuat matriks biner '*coverage*' yang dapat menandakan apakah sebuah titik permintaan ada pada radius tertentu dari salah satu lokasi kandidat. Setelah itu dilakukan iterasi untuk seluruh titik permintaan.

5. Formulasi model MCLP

```

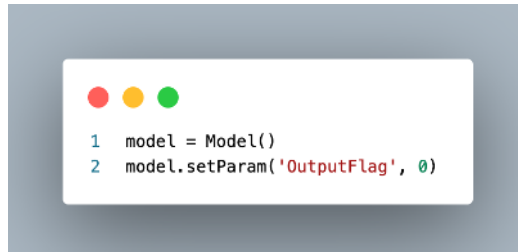
1 x = model.addVars(I, vtype=GRB.BINARY, name="x")
2 y = model.addVars(I, vtype=GRB.BINARY, name="y")
3
4 #constraint 1
5 model.addConstr(quicksum(x[j] for j in range(J)) == K)
6
7 for j in range(J):
8     if not facility_within_depot(j):
9         model.addConstr(x[j] == 0)
10
11 for i in range(I):
12     model.addConstr(quicksum(z[i, j] for j in range(J)) <= 1)
13     model.addConstr(quicksum(z[i, j] for j in range(J)) == y[i])
14
15 #constraint 2
16 for i in range(I):
17     for j in range(J):
18         model.addConstr(z[i, j] <= x[j])
19         if not coverage[i, j]:
20             model.addConstr(z[i, j] == 0)
21
22 for j1 in range(J):
23     for j2 in range(J):
24         if j1 < j2 and facility_distances[j1, j2] < min_facility_dist:
25             model.addConstr(x[j1] + x[j2] <= 1)
26
27 for j in range(J):
28     model.addConstr(quicksum(z[i, j] for i in range(I)) >= x[j])
29
30 #Objective Function
31 model.setObjective(quicksum(demand_weights[i] * y[i] for i in range(I)), GRB.MAXIMIZE)
  
```

Gambar 8 Formulasi model MCLP

Variabel pertama yaitu '*x*', menunjukkan apakah suatu lokasi titik kandidat dipilih atau tidak. Variabel kedua yaitu '*y*', menunjukkan apakah titik permintaan dapat terlayani. Setelah itu ditambahkan adanya beberapa *constraints* atau kendala sesuai dengan struktur model yaitu

jumlah titik yang pilih harus sama dengan nilai P ($K1+K2$) yang sudah ditentukan. . Selanjutnya dilakukan pengecekan terhadap titik fasilitas/jualan, agar hanya titik yang berada dalam radius jangkauan depot yang boleh dipilih. Kemudian, titik permintaan hanya boleh dilayani oleh satu titik fasilitas/jualan saja. Tidak hanya itu, ditambahkan juga adanya batasan jarak antar fasilitas agar menghindari adanya pemilihan dua fasilitas yang terlalu berdekatan. Semua formulasi tersebut ditutup dengan fungsi objektif yang bertujuan untuk memaksimalkan total cakupan permintaan

6. Implementasi model dengan Gurobi



```

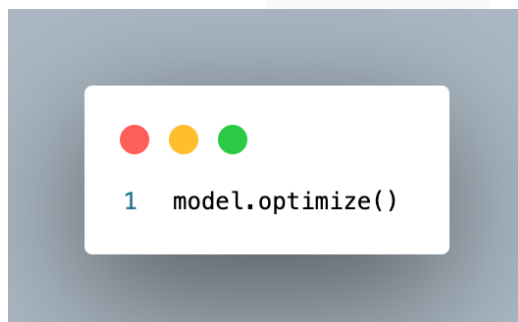
1 model = Model()
2 model.setParam('OutputFlag', 0)

```

Gambar 9 Inisialisasi solver

Objek 'model' dibuat melalui 'model()', yang merupakan konstruktor dari library Gurobi untuk pembentukan model optimasi yang berbasis bilangan bulat. Langkah ini sebagai awal proses komputasi matematis agar dapat menentukan solusi terbaik dari model yang telah dirumuskan sebelumnya.

7. Optimasi Model



```

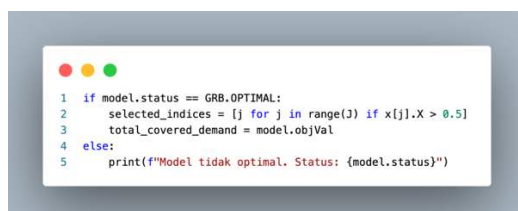
1 model.optimize()

```

Gambar 10 Menjalankan solver

Perintah 'model.optimize()' digunakan untuk menjalankan proses optimasi menggunakan solver Gurobi. Pada prosesnya perintah akan memproses fungsi objektif serta kendala-kendala yang telah didefinisikan sebelumnya serta akan berupaya mencari solusi optimal sesuai dengan tujuan yang telah ditetapkan sebelumnya.

8. Apakah Solusi Optimal?



```

1 if model.status == GRB.OPTIMAL:
2     selected_indices = [j for j in range(3) if x[j].X > 0.5]
3     total_covered_demand = model.objVal
4 else:
5     print(f"Model tidak optimal. Status: {model.status}")

```

Gambar 11 Evaluasi hasil optimasi

Jika hasilnya 'GRB.OPTIMAL' maka indeks fasilitas yang dipilih akan ditentukan dari nilai variabel biner yang lebih besar dari 0,5. Setelah itu nilai 'model.objVal'

menyimpan total permintaan yang berhasil tercakup oleh fasilitas-fasilitas terpilih. Selanjutnya informasi mengenai titik koordinat serta *demand* yang tercakup oleh titik-titik terpilih disimpan dalam 'selected_facilities_info'.

Langkah-Langkah Implementasi K-Means

1. Membaca Dataset



```

1 data = pd.read_csv("Dataset/selected_facilities_with_demand.csv")
2 df_facilities = pd.DataFrame(data)
3 depot = {"lon": 107.59575, "lat": -6.92651}

```

Gambar 12 Pembacaan dataset

Data tersebut ada pada file *selected_facilities_with_demand.csv* yang telah disimpan sebelumnya. Data tersebut dimuat dengan fungsi 'pd.read_csv()' dari library pandas. Setelah data berhasil dibaca, data tersebut diubah menjadi sebuah data *frame* yang diberi nama *df_facilities* yang nantinya digunakan untuk analisis lebih lanjut pada proses klusterisasi.

2. Input Data Koordinat



```

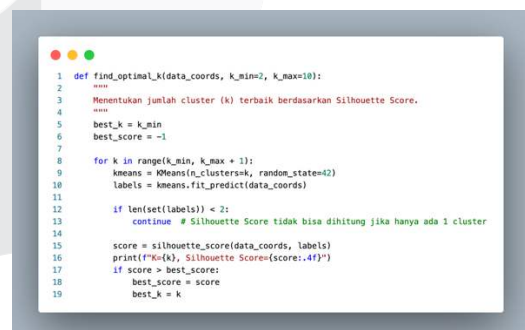
1 coords = df_facilities[['lon', 'lat']].to_numpy()

```

Gambar 13 Ekstraksi titik koordinat

Selanjutnya dilakukan proses ekstraksi setiap titik koordinat dari setiap titik fasilitas/jualan yang sebelumnya telah dimuat. Data titik diambil dari kolom 'lon' dan 'lat' yang menandakan *longitude* dan *latitude*. Kemudian, data tersebut dikonversi menjadi format *array NumPy* menggunakan fungsi 'to_numpy()'. Hasil konversi kemudian disimpan dalam variabel 'coords', tujuannya sebagai input untuk proses klusterisasi *K-means*.

3. Menentukan jumlah cluster optimal



```

1 def find_optimal_k(data_coords, k_min=2, k_max=10):
2     """
3     Menentukan jumlah cluster (k) terbaik berdasarkan Silhouette Score.
4     """
5     best_k = k_min
6     best_score = -1
7
8     for k in range(k_min, k_max + 1):
9         kmeans = KMeans(n_clusters=k, random_state=2)
10        labels = kmeans.fit_predict(data_coords)
11
12        if len(set(labels)) < 2:
13            continue # Silhouette Score tidak bisa dihitung jika hanya ada 1 cluster
14
15        score = silhouette_score(data_coords, labels)
16        print(f"%d, Silhouette Score: {score:.4f}")
17        if score > best_score:
18            best_score = score
19            best_k = k

```

Gambar 14 Penentuan jumlah cluster optimal

Fungsi yang digunakan yaitu metode *silhouette score*, metode ini mengukur seberapa baik setiap titik data berada dalam klasternya jika masing masing titik dibandingkan dengan klaster lain. Proses yang dilakukan dengan mengevaluasi/mencoba berbagai nilai K dalam rentang yang ditentukan. Selanjutnya *silhouette score* dihitung untuk setiap iterasi dan fungsi akan menyimpan nilai K yang memberikan skor tertinggi/optimal.

4. Menghitung radius rata-rata tiap cluster

```

1 if radius is not None:
2     if isinstance(radius, (int, float)):
3         radius = [radius] * n_clusters
4     for i, c in enumerate(centroids):
5         scaled_radius = radius[i] * radius_scale
6         circle = plt.Circle((c[0], c[1]), scaled_radius, color=cmap(i), fill=False,
7                             lw=1.5, linestyle='-', alpha=0.5)
8         ax.add_patch(circle)
    
```

Gambar 15 Visualisasi radius setiap kluster

Fungsi ini akan diaktifkan jika parameter radius memiliki nilai yang bukan *None*. Apabila radius dinyatakan dalam format angka tunggal (baik integer maupun float), maka angka tersebut akan disalin sebanyak jumlah kluster (*n_clusters*) agar setiap kluster mendapatkan radius yang sama. Selanjutnya, dilakukan iterasi untuk setiap *centroid* yang dihasilkan oleh proses *K-Means*, dan di setiap titik pusat kluster tersebut, akan ditambahkan lingkaran dengan radius yang telah disesuaikan oleh faktor '*radius_scale*'.

5. Mendeteksi outlier

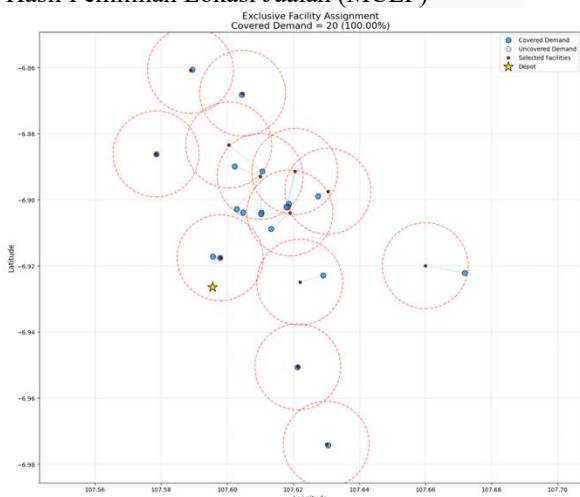
```

1 if outliers.any():
2     ax.scatter(data_coors[outliers, 0], data_coors[outliers, 1],
3               c='black', edgecolor='black',
4               marker='x', s=300,
5               linewidth=2, label='Outliers')
    
```

Gambar 16 Visualisasi outlier

Selanjutnya divisualisasikan juga titik-titik daya yang teridentifikasi sebagai *outlier*, yaitu titik-titik yang secara signifikan berbeda dari pusat kluster. Proses dilakukan dengan fungsi '*outliers.any()*' untuk memastikan bahwa terdapat setidaknya satu data *outlier*. Jika benar ditemukan adanya *outlier* maka akan divisualisasikan menggunakan '*ax.scatter()*' pada grafik.

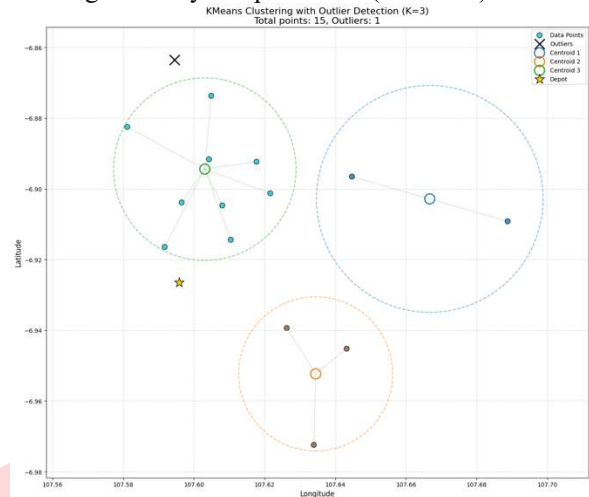
C. Hasil Pemilihan Lokasi Jualan (MCLP)



Gambar 17 Hasil MCLP

Setiap titik berwarna biru melambangkan lokasi permintaan yang tercakup oleh salah satu fasilitas yang dipilih, sedangkan lingkaran merah menandakan jangkauan maksimum dari fasilitas tersebut. Lokasi fasilitas yang dipilih ditandai dengan simbol silang merah, sementara lokasi depot yang menjadi awal distribusi ditunjukkan dengan simbol bintang kuning.

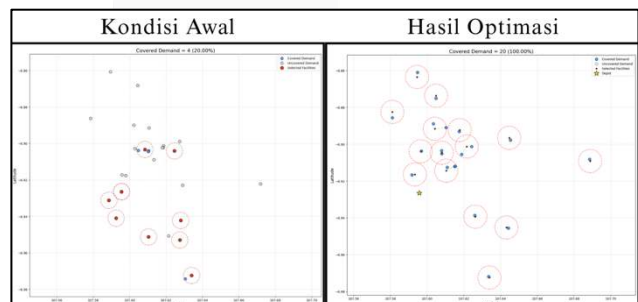
D. Pembagian Wilayah Operasional (K-Means)



Gambar 18 Hasil K-Means clustering

Setiap *cluster* ditandai dengan warna yang berbeda, dilengkapi dengan lingkaran yang melambangkan jarak rata-rata dari pusat menuju titik anggotanya. Titik-titik yang berwarna biru muda, hijau muda, dan oranye muda menunjukkan lokasi yang dikategorikan berdasarkan kedekatannya dengan pusat *cluster*, yang ditunjukkan oleh lingkaran besar dengan warna yang sama. Titik hitam berbentuk silang menunjukkan *outlier*, yaitu lokasi yang jauh dari pusat *cluster* sehingga tidak dapat dimasukkan ke dalam kelompok manapun.

E. Perbandingan Hasil



Gambar 19 Perbandingan dengan kondisi awal

Jika dipetakan dengan koordinat titik permintaan pada area pendidikan potensial, 9 titik penjualan *existing* hanya mencakup 4 titik permintaan atau 20% *covered demand*. Sedangkan pada hasil optimasi, dengan 15 titik penjualan dapat mencakup 20 titik permintaan atau 100% *covered demand*. Pada Tabel dapat dilihat perbandingan total penjualan per *cup* pada titik *existing* dengan estimasi total penjualan pada titik hasil optimasi dengan perhitungan jam kerja yang sama.

Tabel 1 Perbandingan penjualan *cup*/jam

	Titik Existing	Titik Hasil Optimasi (estimasi)
TOTAL	412	1160

Dari perbandingan tersebut dapat dilihat penjualan pada titik *existing* sebesar 412 *cup*/hari dan 1160 *cup*/hari pada titik hasil optimasi. Hal ini menandakan adanya estimasi kenaikan sebesar 182% pada total penjualan. Dengan ini diharapkan titik hasil optimasi dapat memaksimalkan penjualan kopi keliling brand XYZ.

V. KESIMPULAN

Penelitian ini berhasil menyusun usulan wilayah distribusi untuk pedagang kopi keliling brand XYZ dengan memanfaatkan pendekatan Maximal Covering Location Problem (MCLP) dan K-Means clustering. Model MCLP mampu memaksimalkan cakupan permintaan di Bandung dengan radius pelayanan 700 meter, sehingga 20 titik permintaan dapat terlayani secara optimal menggunakan 15 lokasi jualan terpilih. Selanjutnya, pembagian wilayah operasional menggunakan K-Means menghasilkan tiga cluster dengan silhouette score sebesar 0,4948, yang menunjukkan struktur cluster cukup baik dan terpisah jelas. Hasil integrasi kedua metode ini mendukung peningkatan efisiensi distribusi, mengurangi tumpang tindih wilayah operasional, serta mendukung perluasan jangkauan brand XYZ secara lebih terarah dan berbasis data. Penelitian lanjutan dapat mempertimbangkan aspek kapasitas pedagang, variasi permintaan harian, serta variabel dinamis lainnya untuk perbaikan model yang lebih komprehensif.

REFERENSI

- Anwari, A. (2016). Faktor-faktor yang mempengaruhi kebiasaan minum kopi di kalangan mahasiswa. *Jurnal Ekonomi Kreatif Mahasiswa*, 2(1), 55–63.
- AP News. (2023). Indonesia's mobile coffee revolution: How street baristas are changing the game. *AP News*. <https://apnews.com>
- Ardila, D. D., Agustina, T., & Subekti, S. (2019). Saluran dan margin pemasaran kopi Cap Lereng Tancak Kembar di Desa Andungsari Kecamatan Pakem Kabupaten Bondowoso. *SEPA: Jurnal Sosial Ekonomi Pertanian dan Agribisnis*, 15(2), 116–127. <https://doi.org/10.20961/sepa.v15i2.27430>
- Badan Pusat Statistik Kota Bandung. (2023). Jumlah murid SMA menurut kecamatan di Kota Bandung. <https://bandungkota.bps.go.id>
- Ballou, R. H. (2004). *Business logistics/supply chain management: Planning, organizing, and controlling the supply chain* (5th ed.). Pearson Education.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
- Chopra, S., & Meindl, P. (2016). *Supply chain management: Strategy, planning, and operation* (6th ed.). Pearson.
- Church, R., & ReVelle, C. (1974). The maximal covering location problem. *Papers in Regional Science*, 32(1), 101–118. <https://doi.org/10.1007/BF01942293>
- Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches* (4th ed.). SAGE Publications.
- Diestel, R. (2017). *Graph theory* (5th ed.). Springer.
- Evans, J. R., & Lindsay, W. M. (2014). *An introduction to Six Sigma and process improvement*. Cengage Learning.
- Gavrila, A. (2025). Tidak mati akal, melihat Gen Z dalam fenomena bisnis kopi keliling. *GoodStats*. <https://goodstats.id/article/tidak-mati-akal-gen-z-dan-fenomena-bisnis-kopi-keliling-OgliK>
- Gutin, G., & Punnen, A. P. (Eds.). (2002). *The traveling salesman problem and its variations*. Springer.
- Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques* (3rd ed.). Elsevier.
- Ibarra-Rojas, O. J., Delgado, H. E., & Gámez, A. (2021). A multi-criteria maximal covering location model: Application in health care facility planning. *European Journal of Operational Research*, 293(2), 660–676. <https://doi.org/10.1016/j.ejor.2020.02.042>
- Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8), 651–666.
- Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: An introduction to cluster analysis*. Wiley.
- Kementerian Pertanian. (2024). Peluang dan daya saing kopi Indonesia. *Badan Standardisasi Instrumen Pertanian*. <https://tanamanindustri.bsip.pertanian.go.id>
- Kotler, P., & Keller, K. L. (2016). *Marketing management* (15th ed.). Pearson.
- Lee, K., Kim, J., & Yoon, Y. (2020). A robust set covering model for facility location planning with uncertain demands. *Computers & Industrial Engineering*, 139, 106178. <https://doi.org/10.1016/j.cie.2019.106178>
- Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2), 129–137.
- Mona Siahaan. (2025). Produksi dan konsumsi kopi di Indonesia dari 2015 hingga 2024. *Statista*. <https://www.statista.com>

- Otten Coffee. (2023). Fenomena tren kopi keliling: Sudah coba?. *Otten Coffee*. <https://ottencoffee.co.id>
- Santino Coffee. (2024). Data konsumsi kopi di Indonesia terbaru. *Santino Coffee*. <https://santinocoffee.co.id>
- Smith, R., Jones, P., & Zhao, L. (2021). P-median models for service optimization under demand clustering. *Operations Research Letters*, 49(3), 289–296.
- Statista. (2024). Indonesia: Coffee production and consumption from 2015 to 2024. *Statista*. <https://www.statista.com>
- Sunarto. (2004). *Manajemen distribusi*. Ghalia Indonesia.
- Taha, H. A. (2017). *Operations research: An introduction* (10th ed.). Pearson.
- Widhawati, D., Rakhmadi, R. A., & Alhabsyi, F. (2023). Pengaruh konsumsi kopi terhadap kualitas tidur mahasiswa. *Jurnal Psikologi UNIS Tangerang Selatan*, 5(1), 45–52.
- Williams, H. P. (2013). *Model building in mathematical programming* (5th ed.). Wiley.
- Winston, W. L., & Goldberg, J. B. (2004). *Operations research: Applications and algorithms* (4th ed.). Cengage Learning.
- Wolsey, L. A. (2020). *Integer programming*. Wiley.
- Zhao, W., & Snavely, N. (2014). Modeling image appearance from human locations. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3218–3225.
- Díaz, M., Suárez, M., & Iglesias, V. (2021). The missing link between analytics readiness and service firm performance. *International Journal of Information Management*, 57, 102367. <https://doi.org/10.1016/j.ijinfomgt.2020.102367>

