



Penentuan Kluster Koridor TransJakarta dengan Metode Majority Voting pada Algoritma Data Mining

Arief Wibowo¹, Moh Makruf², Inge Virdyna³, Farah Chikita Venna⁴

^{1,2,3,4}Program Studi Magister Ilmu Komputer, Fakultas Teknologi Informasi, Universitas Budi Luhur

¹arief.wibowo@budiluhur.ac.id, ²1911601175@student.budiluhur.ac.id, ³1911601142@student.budiluhur.ac.id,

⁴1911601050@student.budiluhur.ac.id

Abstract

The Covid-19 pandemic has made many changes in the patterns of community activity. Large-Scale Social Restrictions were implemented to reduce the number of transmission of the virus. This clearly affects the mode of transportation. The mode of transportation makes new regulations to reduce the number of passenger capacities in each fleet, for example, TransJakarta services. This study will categorize the TransJakarta corridors before and during the Covid-19 pandemic. The clustering method of K-Means and K-Medoids is used to obtain accurate calculation results. The calculations are performed using Microsoft Excel, Rapid Miner, and Python programming language. The clustering results obtained that using K-Means algorithm before Covid-19 pandemic, an optimum number of clusters is 3 clusters with DBI (Davies Bouldin Index) value is 0.184, and during Covid-19 pandemic, the optimum number of clusters is 2 clusters with DBI value is 0.188. Meanwhile, when using the K-Medoids algorithm before the Covid-19 pandemic, an optimum number of clusters is 3 clusters with the DBI value is 0.200, and during the Covid-19 pandemic, an optimum number of clusters is 4 clusters with the DBI value is 0.190. The final cluster is determined using the majority voting approach from all the tools used.

Keywords: covid-19, transjakarta, clustering, davies bouldin index, majority voting

Abstrak

Pandemi Covid-19 menjadikan banyak perubahan dalam pola aktifitas masyarakat. Pembatasan Sosial Berskala Besar (PSBB) diimplementasi guna menekan angka penularan virus tersebut. Hal ini jelas berpengaruh terhadap moda transportasi. Moda transportasi membuat aturan baru untuk mengurangi jumlah kapasitas penumpang di setiap armada, sebagai contoh layanan TransJakarta. Penelitian ini akan mengelompokkan koridor TransJakarta sebelum dan selama masa pandemi Covid-19. Metode klusterisasi K-Means dan K-Medoids digunakan untuk mendapatkan hasil perhitungan yang akurat. Perhitungan dilakukan menggunakan tools Microsoft Excel, Rapid Miner dan bahasa pemrograman Python. Berdasarkan hasil pengelompokan yang diperoleh dari penggunaan algoritma K-Means sebelum masa pandemi, jumlah *cluster* optimal adalah 3 *cluster* dengan nilai DBI (Davies Bouldin Index) 0,184, dan selama masa pandemi, jumlah *cluster* optimal adalah 2 *cluster* dengan nilai DBI 0,188. Sementara itu, bila menggunakan algoritma K-Medoids sebelum masa pandemi, jumlah *cluster* optimal adalah 3 *cluster* dengan nilai DBI 0,200, dan selama masa pandemi, jumlah *cluster* optimal adalah 4 *cluster* dengan nilai DBI 0,190. Penentuan *cluster* akhir menggunakan pendekatan metode *majority voting* dari semua tools yang digunakan.

Kata kunci: covid-19, transjakarta, klusterisasi, davies bouldin index, pemilihan mayoritas

1. Pendahuluan

Indonesia memiliki sistem transportasi bus cepat atau *Bus Rapid Transit (BRT)* pertama di Asia Tenggara dan Asia Selatan yaitu TransJakarta [1]. TransJakarta dirancang untuk warga Jakarta melalui pelayanan yang cepat, nyaman, dan harga terjangkau. TransJakarta beroperasi sejak tanggal 1 Februari 2004. Menurut situs transjakarta.co.id, hingga saat ini TransJakarta tersedia

38 koridor yang terdiri dari 13 koridor utama dan 25 rute lintas koridor.

Corona Virus Disease 2019 (Covid-19) merupakan penyakit yang disebabkan oleh virus baru yang berasal dari Sars-CoV-2. Kasus pertama Covid-19 ditemukan pada tanggal 31 Desember 2019 di kota Wuhan, Tiongkok. Pada 11 Maret 2020, *World Health Organization (WHO)* menyatakan Covid-19 merupakan

pandemi global. Kasus pertama positif Covid-19 di Indonesia dikonfirmasi pada 2 Maret 2020 [2].

Kasus terkonfirmasi positif Covid-19 di Indonesia semakin meningkat. Untuk mengurangi penyebaran virus Covid-19, pemerintah menerapkan berbagai kebijakan seperti *social distancing* atau *physical distancing* dan kebijakan Pembatasan Sosial Berskala Besar (PSBB). PSBB ditetapkan oleh Menteri Kesehatan RI usulan Gubernur atau Walikota. Jabodetabek menerapkan kebijakan PSBB. Namun kebijakan tersebut dapat mengakibatkan penurunan penumpang angkutan umum seperti TransJakarta. Selama ditetapkan PSBB, TransJakarta tetap beroperasi namun dibatasi jumlah koridor, armada dan intensitas waktu operasi.

Penelitian terdahulu mengenai *clustering* yaitu Adi Supriyatna tentang *clustering* koridor TransJakarta menggunakan algoritma K-Means. Penelitian tersebut diproses melalui *Microsoft Excel* dan *Rapid Miner Studio* menghasilkan 3 *cluster* (jumlah penumpang terbanyak, sedang, dan rendah). Limitasi dari penelitian ini adalah jumlah dataset yang dipakai kurang banyak [3].

Penelitian lain membahas penggunaan metode K-Means untuk *clustering* data penjualan. Data diolah melalui perhitungan manual dan *Rapid Miner*, hasil dari penelitian yaitu menghasilkan 3 *cluster* barang yang dijual yaitu laris, kurang laris, dan cukup laris. Limitasi dari penelitian adalah rentang waktu sampel yang digunakan kurang panjang yaitu selama 3 bulan [4]. Pada kasus kebakaran hutan yang dibahas pada penelitian oleh Sukamto, dkk menggunakan algoritma *Chebysev* K-Means menghasilkan 3 *cluster* (daerah sangat rawan, daerah rawan dan daerah tidak rawan). Pengklasteran pada penelitian ini sudah optimal karena nilai DBI yang diperoleh 0,361. Kekurangan pada penelitian ini adalah tidak menggunakan tools data mining untuk perhitungan sehingga dikhawatirkan perhitungan yang dilakukan tidak akurat [5].

Penelitian yang membahas sektor pertanian mengenai analisa terhadap produktifitas penanaman padi pada 34 provinsi di Indonesia menggunakan metode klusterisasi K-Means. Hasil dari penelitian menunjukkan bahwa metode klusterisasi K-Means mampu memberikan hasil yang optimal dengan menghasilkan 3 klaster, yaitu 4 provinsi dengan produktivitas tinggi, 19 provinsi dengan produktifitas menengah dan 11 provinsi dengan produktifitas rendah. Kekurangan dari penelitian ini adalah tidak adanya evaluasi untuk menentukan jumlah klaster optimal [6].

Penelitian lainnya dari Indra Gunawan tentang *clustering* perkembangan kasus COVID-19 menggunakan algoritma K-Medoids. Pengolahan data pada penelitian tersebut diproses melalui *R Studio*,

Geoda, *Statscan* dan *QGIS* menghasilkan optimum jumlah kelompok sebanyak tiga kategori (tinggi, sedang, rendah). Limitasi dari penelitian adalah hanya menggunakan tiga *variable* yaitu positif, sembuh, dan meninggal, dapat ditambahkan *variable* suspek dan pasien negative [7]. Penelitian klusterisasi juga dilakukan pada sektor kesehatan. Penelitian dari Tri Juninda, dkk menyimpulkan bahwa algoritma K-Medoids dapat melakukan pengelompokan penyakit yang ada di Pekanbaru Riau dengan pengklasteran terbaik dilakukan dengan 4 *cluster*, sehingga proses implementasi pada sistem dan analisis dapat diterapkan berdasarkan penelitian tersebut. Kekurangan pada penelitian ini adalah tidak menggunakan tools data mining untuk perhitungan sehingga dikhawatirkan perhitungan yang dilakukan tidak akurat [8]. Penelitian yang dilakukan oleh Dini Marlina, dkk mengenai pengelompokan pada data sebaran anak cacat yang ada pada Provinsi Riau dengan menggunakan algoritma K-Means dan K-Medoid. Hasil dari komparasi algoritma K-Medoids dan K-Means melalui validitas *Silhouette Coefficient* menghasilkan nilai validitas algoritma K-Medoids sebesar 0.5009, sedangkan nilai validitas algoritma K-Means sebesar 0.1443. Hal ini menunjukkan algoritma K-Medoids lebih baik dibandingkan dengan algoritma K-Means. Limitasi dari penelitian ini adalah ruang lingkup yang digunakan terbatas di provinsi Riau, belum mencakup daerah lain [9].

Penelitian oleh Pramesti, dkk tentang Implementasi metode *K-Medoids Clustering* untuk pengelompokan dalam data potensi kebakaran Hutan/Lahan berdasarkan persebaran titik panas (*Hotspot*), menghasilkan bahwa jumlah *cluster* dan jumlah data berpengaruh terhadap hasil kualitas dari *cluster*. Limitasi dari penelitian ini adalah data yang digunakan terbatas hanya satu bulan, dan perlu dilakukan teknik pemilihan parameter yang digunakan [10]. Pada penelitian oleh Defiyanti, dkk tentang Optimalisasi K-Medoid dalam Pengklasteran Mahasiswa Pelamar Beasiswa dengan *Cubic Clustering Criterion*, membahas tentang jenis data terbaik dalam pengelompokan beasiswa dihitung dengan nilai *Cubic Clustering Criterion (CCC)* menghasilkan tiga buah pengelompokan. Dataset dengan kodifikasi keseluruhan data menempati predikat terbaik dalam keseragaman dalam pengelompokan. Limitasi pada penelitian ini adalah jumlah data yang digunakan sedikit [2].

Pada penelitian ini, penulis mengelompokkan koridor TransJakarta berdasarkan tiga variabel yaitu jumlah penumpang, jarak rute antar koridor, dan jumlah halte yang dilewati. Selama pandemi Covid-19 dan kebijakan PSBB, TransJakarta hanya mengoperasikan 13 koridor utama tanpa mengurangi jumlah halte yang dilewati tiap koridor. Peneliti melakukan studi ini berdasarkan dua hal utama, yaitu adanya perubahan pengaktifan koridor TransJakarta yang mendorong masyarakat untuk

mengganti rute perjalanan yang biasa dilakukan menjadi rute lain. Faktor kedua, berdasarkan perubahan yang terjadi, peneliti ingin mengetahui koridor mana yang mengalami lonjakan penumpang yang signifikan dan koridor mana yang membutuhkan armada lebih dan tambahan jumlah petugas. Dengan demikian maka pertanyaan penelitian yang akan dipecahkan adalah, bagaimana model klasterisasi terhadap koridor-koridor Transjakarta sebelum dan sesudah pandemi Covid-19 terjadi di Jakarta. Hal ini menjadi penting karena klaster yang terbentuk nantinya akan menjadi bagian dari pertimbangan yang dilakukan oleh Badan Pengelola dalam manajemen operasional koridor bus Transjakarta.

Pendekatan algoritma klasterisasi data mining dipilih karena berdasarkan literatur yang diulas, diketahui bahwa metode klasterisasi dengan K-Means dan K-Medoids mampu mencapai hasil yang optimal dalam pembentukan pola klasterisasi.

Studi literatur terdahulu yang membahas tentang pengelompokan koridor TransJakarta belum ada yang mengkomparasi penggunaan algoritma K-Medoids dan K-Means. Hasil dari penelitian sebelumnya, algoritma k-Means dan K-Medoids terbukti memiliki kinerja yang baik, sehingga peneliti ingin menguji kinerja algoritma tersebut dengan data yang tersedia.

Pada penelitian ini, pengelompokan data menggunakan *Rapid Miner*, *Phyton* dan *Microsoft Excel*. Selanjutnya hasil akhir penentuan *cluster* menggunakan *majority voting* berdasarkan semua tools yang digunakan. Pemilihan metode *majority voting* menjadi penting agar pengambilan keputusan menjadi kuat berdasarkan keputusan mayoritas [11].

2. Metode Penelitian

Penelitian ini mengklasterisasi koridor TransJakarta berdasarkan jumlah penumpang, jarak rute koridor dan jumlah halte yang dilewati menggunakan metode *clustering* K-Means dan K-Medoids dalam beberapa tahapan, antara lain Tahap Identifikasi Masalah. Tahapan ini adalah mengidentifikasi bagaimana mengelompokkan koridor TransJakarta menggunakan metode klasterisasi K-Means dan K-Medoids.

Tahap kedua adalah Pengumpulan Data. Tahap ini adalah mencari data yang hendak digunakan, seperti jumlah penumpang, jarak rute koridor, dan jumlah halte yang dilewati. Data diambil adalah data dua periode. Periode pertama yaitu sebelum Covid-19 yaitu bulan Agustus 2019 – Maret 2020. Periode kedua yaitu selama Covid-19 yaitu bulan April 2020 – November 2020. Data tersebut berasal dari situs resmi Jakarta Open Data (data.jakarta.go.id) dan transjakarta.co.id.

Data yang diolah menjadi dua bagian yaitu, 8 bulan sebelum Covid-19 dan 8 bulan selama Covid-19.

Berdasarkan kebijakan pemerintah yang membatasi angkutan umum selama Covid-19, maka peneliti menggunakan 13 koridor utama dengan trayek jenis BRT. 13 Koridor tersebut adalah Koridor 1 (Blok M-Kota), Koridor 2 (Pulogadung 1 – Harmoni), Koridor 3 (Kalideres – Pasar Baru), Koridor 4 (Pulogadung 2 – Dukuh Atas 2), Koridor 5 (Kampung Melayu – Ancol), Koridor 6 (Ragunan – Dukuh Atas 2), Koridor 7 (Kampung Rambutan – Kampung Melayu), Koridor 8 (Lebak Bulus – Harmoni), Koridor 9 (Pinang Ranti – Pulit), Koridor 10 (PGC 2 – Pluit), Koridor 11 (Pulogebang – Kampung Melayu), Koridor 12 (Penjaringan – Tanjung Priok), dan Koridor 13 (CBD Ciledug – Blok M). Tujuan dari penelitian ini adalah diharapkan dapat dijadikan bahan pertimbangan dalam alokasi sumber daya, baik itu armada, personel, maupun jadwal pada tiap-tiap koridor.

Tahap ketiga adalah Pengolahan Data dan Implementasi. Setelah dikumpulkan, selanjutnya data diolah menggunakan metode klasterisasi K-Means dan K-Medoids. Tahapan diproses melalui perangkat lunak aplikasi *spreadsheet*, aplikasi *Rapid Miner*, dan pemrograman *Python*.

Tahap terakhir adalah Evaluasi Hasil Klasterisasi. Pada tahap ini, data dianalisis dalam bentuk *cluster model* dari pengelompokan koridor TransJakarta berdasarkan jumlah penumpang, jarak rute koridor, dan jumlah halte yang dilewati.

Bagian inti dari penelitian ini adalah proses penambangan data yang akan menghasilkan klaster sesuai dengan pengetahuan atau pola yang tersimpan pada data.

Klasterisasi atau *clustering* merupakan salah satu metode pengelompokan data berdasarkan kesamaan sifat data. Apabila data tidak memiliki kesamaan sifat, maka akan dikelompokkan dalam kelompok yang lain [12]. Adapun tujuan dari klasterisasi adalah untuk mempartisi objek dalam sebuah *cluster* (kelompok) yang memiliki kemiripan dibandingkan dengan *cluster* (kelompok) lain [13].

Algoritma K-Means merupakan salah satu metode *clustering* melalui atribut *numeric* [14]. Tahapan dalam implementasi algoritma K-Means, terdiri dari [15] penentuan jumlah *cluster*, selanjutnya adalah menentukan pusat awal *cluster* dengan mengambil data secara acak.

Pada tahap selanjutnya dilakukan perhitungan jarak dengan pusat *cluster* atau *centroid* menurut persamaan 1.

$$d(x, y) = |x - y| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

Dengan demikian dapat dilakukan penempatan data pada *cluster* berdasarkan jarak nilai *centroid* terdekat.

Adapun penentuan pusat *centroid* baru dilakukan dengan indikasi, apabila *centroid* berubah, maka ditetapkan pusat *cluster* yang baru, menggunakan persamaan 2.

$$\text{Pusat Cluster Baru} = \frac{x_1+x_2+x_3+\dots+x_n}{\text{jumlah } x} \quad (2)$$

Dimana $x_1+x_2+x_3+\dots+x_n$ adalah jumlah anggota *cluster*.

Berbeda dengan Algoritma K-Means, Algoritma K-Medoids disebut sebagai varian dari *partitioning around medoids*. *Medoid* adalah titik pusat dari suatu *cluster*. Tujuan dari *medoid* dalam setiap *cluster* adalah untuk mengurangi sensitivitas dari partisi yang dihasilkan sebagai nilai-nilai ekstrim yang ada pada dataset [16]. Perbedaan algoritma K-Means dan K-Medoids adalah K-Medoids menentukan nilai k sebagai objek representative untuk meminimalkan jumlah ketidaksamaan objek data, sedangkan K-Means menentukan jumlah jarak *euclidean distance* untuk objek data [17]. Tahapan dalam implementasi algoritma K-Medoids, dimulai dengan langkah normalisasi hingga iterasi yang akan berakhir saat anggota klaster tidak berubah lagi [10].

Pada tahap normalisasi, dilakukan pada data yang akan diolah menggunakan persamaan 3.

$$\text{Normalized } (x) = \frac{x - \text{MinValue}}{\text{MaxValue} - \text{MinValue}} \quad (3)$$

Tahap selanjutnya adalah menentukan pusat *cluster* sebanyak k (jumlah *cluster*). Perhitungan jarak dari setiap data (objek) ke *cluster* terdekat menggunakan persamaan (1) sehingga objek dapat ditentukan secara acak pada masing-masing *cluster* dengan *medoid* baru. Tahap selanjutnya adalah menghitung jarak setiap objek dengan kandidat *medoid* baru dan menghitung simpangan berdasarkan persamaan 4.

$$S = \sum \text{nilai distance baru} - \sum \text{nilai distance lama} \quad (4)$$

Jika $S < 0$, maka ganti objek dengan data *cluster* untuk memperoleh sekelompok k objek yang baru sebagai *medoid*.

Proses tersebut berulang sampai tidak ada perubahan bentuk *cluster* dan anggota masing-masing *cluster*.

Untuk pemecahan masalah penelitian, penelitian ini mengadopsi pendekatan *Knowledge Discovery In Database (KDD)*, yaitu serangkaian kegiatan yang terdiri dari tahap pengumpulan, pemakaian data lampau yang bertujuan untuk menemukan pola, atau hubungan pada set data yang berukuran besar.

Tahap-tahap KDD meliputi kegiatan-kegiatan yang dimulai dari *Data Selection* hingga Interpretasi atau evaluasi [18]. Tahap *Data Selection* merupakan pemilihan data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam

KDD dimulai. Data hasil seleksi disimpan dalam suatu berkas yang terpisah dari basis data operasional.

Tahap *Pre-processing/Cleaning*, diperlukan sebagai representasi proses pembersihan pada data sebelum pemrosesan dalam kerangka KDD. Tahap pembersihan dilakukan berupa aktifitas seperti mengeliminasi data yang bersifat duplikasi, analisis pada data yang tidak konsisten, serta pekerjaan memperbaiki kondisi kesalahan data, seperti kesalahan cetak (tipografi).

Tahap *Transformation*, merupakan bentuk kegiatan *coding* yang bertujuan agar data agar sesuai untuk pekerjaan-pekerjaan *data mining* selanjutnya. Proses *coding* merupakan proses yang berkaitan dengan jenis atau pola informasi yang akan dicari dalam basis data.

Tahap *Data Mining*, adalah proses mencari pola atau informasi yang menarik dalam data terpilih menggunakan teknik atau metode tertentu.

Tahap *Interpretation/Evaluation*, merupakan tahap untuk mendapatkan hasil penyajian informasi hasil penambangan data. Tahap ini dilakukan agar pengetahuan dapat dimengerti oleh pengguna. Tahap ini termasuk validasi antara pola atau pengetahuan terhadap fakta atau hipotesis.

Dalam proses penentuan keputusan data mining dikenal metode *majority voting*, misalnya dalam proses pemilihan label yang dominan dari setiap hasil penerapan algoritma [19]. Pendekatan yang paling sederhana dan intuitif didasarkan pada aturan *majority voting* yang menetapkan sampel berdasarkan penugasan kelas terbanyak [20]. Aturan *majority voting* menetapkan sampel ke kelas yang terkait dengan frekuensi prediksi tertinggi, kecuali jika pembatasan khusus pada persentase kesepakatan kelas tidak dipenuhi [21].

3. Hasil dan Pembahasan

Langkah-langkah untuk melakukan *data mining* mengikuti aturan KDD (*Knowledge Discovery from Database*), dimulai dari tahap seleksi data hingga evaluasi.

Pada tahap seleksi, diputuskan menggunakan data 8 bulan sebelum masa pandemi Covid-19 (Agustus 2019 sampai dengan Maret 2020) dan 8 bulan awal masa pandemi Covid-19 (April 2020 sampai dengan November 2020). Rentang waktu ini merupakan periode yang unik untuk dianalisis karena pada delapan bulan masa pandemi terjadi fluktuasi penggunaan moda transportasi TransJakarta, berkaitan dengan dimulainya PSBB, pelonggaran dan pengetatan kembali PSBB hingga diberlakukannya PKMM di DKI Jakarta. Data hasil seleksi memiliki 6 atribut, yaitu jenis, kode_trayek, trayek, jumlah_penumpang, panjang_rute, dan jumlah_halte. Berdasarkan tujuan dari penelitian ini,

atribut yang dipilih menjadi 4 atribut yaitu trayek, jumlah_penumpang, panjang_rute, dan jumlah_halte.

Tahap selanjutnya yaitu *Data Preprocessing/Data Cleaning*. Sumber data terdiri dari 4 atribut yaitu trayek, jumlah penumpang, panjang rute, dan jumlah halte. Atribut jumlah_penumpang diperoleh dari data jumlah penumpang yang terdapat dalam website data.jakarta.go.id. Sedangkan atribut panjang_rute adalah panjang rute yang dilalui pada tiap koridor, dan atribut jumlah_halte adalah jumlah halte yang dilalui pada tiap koridor. Dari keseluruhan data yang berjumlah 1.403, beberapa trayek tidak beroperasi selama masa pandemi Covid-19 yang ditandai dengan jumlah penumpang sebanyak 0 atau tidak ada penumpang. Trayek TransJakarta jenis BRT memiliki empat atribut data (trayek, jumlah penumpang, panjang rute, dan jumlah halte), sedangkan trayek lain hanya memiliki 2 atribut (trayek dan jumlah penumpang) dan tidak diperoleh informasi mengenai panjang rute dan jumlah halte. Sesuai dengan tujuan penelitian yaitu mengelompokkan koridor TransJakarta dan ketersediaan data, sehingga data trayek yang dipilih oleh penulis adalah jenis BRT untuk digunakan dalam pemodelan berjumlah 13 yang merupakan koridor utama TransJakarta.

Tahap selanjutnya, proses transformasi data yang dilakukan dengan cara menjumlahkan data jumlah penumpang 8 bulan sebelum masa pandemi Covid-19 menjadi total_penumpang, dan menjumlahkan data jumlah penumpang 8 bulan selama awal masa pandemi Covid-19 menjadi total_penumpang, sehingga diperoleh data yang ditunjukkan pada Tabel 1 dan Tabel 2 berikut.

Tabel 1. Dataset Koridor TransJakarta Sebelum Pandemi

Trayek/ Koridor	Total Penumpang (juta)	Panjang Rute (km)	Jumlah Halte
Koridor 1	19,33	12,90	17
Koridor 2	6,96	24,20	32
Koridor 3	8,44	19,00	16
Koridor 4	5,37	11,85	17
Koridor 5	8,50	13,50	18
Koridor 6	8,19	13,30	20
Koridor 7	8,13	12,80	14
Koridor 8	8,56	26,00	22
Koridor 9	11,95	29,90	27
Koridor 10	6,59	19,40	22
Koridor 11	2,68	15,00	16
Koridor 12	2,19	23,75	25
Koridor 13	6,05	9,30	12

Tabel 2. Dataset Koridor TransJakarta Selama Pandemi

Trayek/ Koridor	Total Penumpang (juta)	Panjang Rute (km)	Jumlah Halte
Koridor 1	5,61	12,90	17
Koridor 2	2,41	24,20	32
Koridor 3	3,06	19,00	16

Trayek/ Koridor	Total Penumpang (juta)	Panjang Rute (km)	Jumlah Halte
Koridor 4	1,77	11,85	17
Koridor 5	2,67	13,50	18
Koridor 6	2,35	13,30	20
Koridor 7	2,76	12,80	14
Koridor 8	2,91	26,00	22
Koridor 9	4,55	29,90	27
Koridor 10	2,26	19,40	22
Koridor 11	0,96	15,00	16
Koridor 12	0,94	23,75	25
Koridor 13	1,90	9,30	12

Tahap utama dari penelitian ini adalah *data mining* yang dilakukan dengan algoritma K-Means dan K-Medoids pada operasionalisasi menggunakan tiga *tools* yaitu perangkat lunak aplikasi *spreadsheet*, aplikasi *Rapid Miner*, dan pemrograman Python.

3.1. Proses *Data Mining* menggunakan perangkat lunak aplikasi *spreadsheet*

Langkah-langkah pada proses klusterisasi dengan algoritma K-Means dijelaskan sebagai berikut:

- a) Menentukan nilai k, yaitu *cluster* yang akan dibentuk. Untuk data sebelum masa pandemi Covid-19 nilai k adalah 3, sedangkan untuk perhitungan data selama pandemi Covid-19 nilai k adalah 2.
- b) Menentukan *centroid* atau pusat *cluster*. *Centroid* dipilih secara acak, sebanyak 3 titik untuk masa sebelum pandemi dan 2 titik untuk masa selama pandemi. Diperoleh titik pusat untuk masing-masing *cluster* seperti ditunjukkan pada Tabel 3 berikut.

Tabel 3. Titik Pusat Awal pada Setiap *Cluster* (K-Means)

	Total Penumpang	Panjang Rute	Jumlah Halte
Sebelum Pandemi			
<i>Cluster</i> 1	19,33	12,9	17
<i>Cluster</i> 2	11,95	29,9	27
<i>Cluster</i> 3	2,19	9,3	12
Selama Pandemi			
<i>Cluster</i> 1	1,77	11,85	17
<i>Cluster</i> 2	2,76	12,8	14

- c) Menghitung jarak setiap objek data terhadap setiap *centroid* yang terbentuk. Dari perhitungan ini akan diketahui objek data tersebut akan berada pada cluster yang mana. Berikut adalah beberapa contoh perhitungan untuk data sebelum masa pandemi terhadap 3 titik pusat. Perhitungan jarak objek data dengan *centroid* pertama:

$$d1 = \sqrt{(19,33 - 19,33)^2 + (12,9 - 12,9)^2 + (17 - 17)^2} = 0$$

Perhitungan jarak objek dengan centroid kedua:

$$d2 = \sqrt{(19,33 - 11,95)^2 + (12,9 - 12,9)^2 + (17 - 27)^2} = 21,06065441$$

Perhitungan jarak objek data dengan centroid ketiga:

$$d3 = \sqrt{(19,33 - 2,19)^2 + (12,9 - 9,3)^2 + (17 - 12)^2} = 18,21984364$$

Berikutnya adalah contoh perhitungan jarak objek data selama pandemi terhadap 2 titik pusat. Perhitungan jarak objek data terhadap *centroid* pertama:

$$d1 = \sqrt{(5,61 - 1,77)^2 + (12,9 - 11,85)^2 + (17 - 17)^2} = 3,97489065$$

Perhitungan jarak objek data terhadap *centroid* kedua:

$$d2 = \sqrt{(5,61 - 2,76)^2 + (12,9 - 12,8)^2 + (17 - 14)^2} = 4,113321049$$

- d) Memilih jarak terdekat setiap objek data dengan *centroid*, dan masukkan objek data tersebut ke dalam *cluster* yang didalamnya terdapat *centroid* tersebut. Hasil perhitungan iterasi pertama untuk data sebelum pandemi dan selama pandemi dapat dilihat pada Tabel 4 berikut.

Tabel 4. Hasil Perhitungan Iterasi Pertama K-Means

Masa	Jumlah Cluster	Nama Cluster	Jumlah Anggota
Sebelum Pandemi	3	Cluster 1	1
		Cluster 2	5
		Cluster 3	7
Selama Pandemi	2	Cluster 1	10
		Cluster 2	3

Setelah semua objek data masuk ke dalam tiap *cluster*, hitung nilai rata-rata atribut pada tiap *cluster* untuk dijadikan *centroid* baru. Berikut adalah beberapa contoh perhitungan *centroid* baru. Misal pada perhitungan iterasi pertama sebelum pandemi pada *cluster 2* terdapat 5 anggota, maka nilai atribut pada tiap objek data dijumlahkan kemudian dibagi jumlah anggota.

$$C21 = (6,96 + 8,56 + 11,95 + 6,59 + 2,19) / 5 = 7,247290$$

$$C22 = (24,2 + 26 + 29,9 + 19,4 + 23,75) / 5 = 24,65$$

$$C23 = (32 + 22 + 27 + 22 + 25) / 5 = 25,6$$

Pada perhitungan iterasi pertama selama pandemi pada *cluster 2* terdapat 3 anggota, maka perhitungan *centroid* baru adalah sebagai berikut:

$$C21 = (3,06 + 2,76 + 1,90) / 3 = 2,572308$$

$$C22 = (19 + 12,9 + 23,75) / 3 = 13,7$$

$$C23 = (16 + 14 + 12) / 3 = 14$$

Keterangan:

- C21 = atribut 1 pada *centroid cluster 2*.
- C22 = atribut 2 pada *centroid cluster 2*.
- C23 = atribut 3 pada *centroid cluster 2*.

Setelah memperoleh *centroid* baru, maka proses (b) hingga (d) akan diulang hingga tiap anggota *cluster* dan

centroid tiap *cluster* tidak mengalami perubahan. Proses berhenti pada iterasi kedua untuk data sebelum pandemi Covid-19 dan proses berhenti pada iterasi keempat untuk data selama pandemi Covid-19.

Langkah langkah pada proses klusterisasi dengan metode *k-medoids* dijelaskan secara rinci dibawah ini:

- a) Melakukan normalisasi pada sampel data yang akan diolah, menggunakan rumus (3) di atas. Hasil dari normalisasi data adalah sebagai berikut:

Tabel 5. Normalisasi Data Sebelum Pandemi

Koridor	Total Penumpang	Panjang Rute	Jumlah Halte
Koridor 1	1,00	0,17	0,25
Koridor 2	0,28	0,72	1,00
Koridor 3	0,36	0,47	0,20
Koridor 4	0,19	0,12	0,25
Koridor 5	0,37	0,20	0,30
Koridor 6	0,35	0,19	0,40
Koridor 7	0,35	0,17	0,10
Koridor 8	0,37	0,81	0,50
Koridor 9	0,57	1,00	0,75
Koridor 10	0,26	0,49	0,50
Koridor 11	0,03	0,28	0,20
Koridor 12	0,00	0,70	0,65
Koridor 13	0,23	0,00	0,00

Tabel 6. Normalisasi Data Selama Pandemi

Koridor	Total Penumpang	Panjang Rute	Jumlah Halte
Koridor 1	1,00	0,17	0,25
Koridor 2	0,31	0,72	1,00
Koridor 3	0,45	0,47	0,20
Koridor 4	0,18	0,12	0,25
Koridor 5	0,37	0,20	0,30
Koridor 6	0,30	0,19	0,40
Koridor 7	0,39	0,17	0,10
Koridor 8	0,42	0,81	0,50
Koridor 9	0,77	1,00	0,75
Koridor 10	0,28	0,49	0,50
Koridor 11	0,00	0,28	0,20
Koridor 12	0,00	0,70	0,65
Koridor 13	0,20	0,00	0,00

- b) Menentukan nilai k, yaitu jumlah *cluster* yang akan dibentuk. Untuk data sebelum pandemi Covid-19 nilai k adalah 3, sedangkan untuk perhitungan data selama pandemi Covid-19 nilai k adalah 4.
- c) Menentukan *medoid* atau pusat cluster. *Medoid* dipilih secara acak, sebanyak 3 titik sebelum masa pandemi, dan 4 titik untuk masa selama pandemi. Diperoleh titik pusat untuk masing-masing *cluster* seperti ditunjukkan pada Tabel 7.

Tabel 7. Titik Pusat Awal pada Setiap Cluster (K-Medoids)

	Total Penumpang	Panjang Rute	Jumlah Halte
Sebelum Pandemi			
Cluster 1	0,57	1,00	0,75

	Total Penumpang	Panjang Rute	Jumlah Halte
Cluster 2	0,28	0,72	1,00
Cluster 3	0,00	0,70	0,65
Selama Pandemi			
Cluster 1	1,00	0,17	0,25
Cluster 2	0,77	1,00	0,75
Cluster 3	0,42	0,81	0,50
Cluster 4	0,00	0,70	0,65

Masa	Jumlah Cluster	Nama Cluster	Jumlah Anggota
		Cluster 2	1
		Cluster 3	9
Selama Pandemi			
	4	Cluster 1	1
		Cluster 2	1
		Cluster 3	6
		Cluster 4	5

Tahap selanjutnya adalah menghitung jarak setiap objek data terhadap setiap *medoid* yang terbentuk. Dari perhitungan ini akan diketahui objek data tersebut akan berada pada *cluster* yang mana. Berikut adalah beberapa contoh perhitungan untuk data sebelum pandemi terhadap 3 titik pusat.

Perhitungan jarak objek data dengan *medoid* pertama:
 $d1 = \sqrt{(1 - 0,57)^2 + (0,17 - 1)^2 + (0,25 - 0,75)^2}$
 $= 1,056680$

Perhitungan jarak objek data dengan *medoid* kedua:
 $d2 = \sqrt{(1 - 0,28)^2 + (0,17 - 0,72)^2 + (0,25 - 1)^2}$
 $= 1,1766606$

Perhitungan jarak objek data dengan *medoid* ketiga:
 $d3 = \sqrt{(1 - 0)^2 + (0,17 - 0,7)^2 + (0,25 - 0,65)^2}$
 $= 1,198921$

Berikutnya adalah contoh perhitungan jarak objek data selama masa pandemi terhadap 4 titik pusat. Perhitungan jarak objek data dengan *medoid* pertama:
 $d1 = \sqrt{(1 - 1)^2 + (0,17 - 0,17)^2 + (0,25 - 0,25)^2} = 0$

Perhitungan jarak objek data dengan *medoid* kedua:
 $d2 = \sqrt{(1 - 0,77)^2 + (0,17 - 1)^2 + (0,25 - 0,75)^2}$
 $= 0,991166$

Perhitungan jarak objek data dengan *medoid* ketiga:
 $d3 = \sqrt{(1 - 0,42)^2 + (0,17 - 0,81)^2 + (0,25 - 0,5)^2}$
 $= 0,895059$

Perhitungan jarak objek data dengan *medoid* keempat:
 $d4 = \sqrt{(1 - 0)^2 + (0,17 - 0,7)^2 + (0,25 - 0,65)^2}$
 $= 0,810464$

- d) Memilih jarak terdekat setiap objek data dengan *medoid*, dan masukkan objek data tersebut ke dalam cluster yang di dalamnya terdapat *medoid* tersebut. Hasil perhitungan iterasi pertama untuk data sebelum pandemi dan selama pandemi dapat dilihat pada Tabel 8.

Tabel 8. Hasil Perhitungan Iterasi Pertama K-Medoids

Masa	Jumlah Cluster	Nama Cluster	Jumlah Anggota
Sebelum Pandemi	3	Cluster 1	3

Setelah semua data masuk ke dalam setiap *cluster*, baik data perhitungan iterasi pertama maupun iterasi berikutnya, jumlahkan nilai jarak terdekat dari data tersebut dan tentukan nilai simpangannya. Sebagai contoh, perhitungan simpangan pada data sebelum pandemi terhadap iterasi pertama dan iterasi kedua:

Jumlah jarak terdekat iterasi pertama
 $= 1,06 + 0 + 0,62 + \dots + 0,98 = 6,96$

Jumlah jarak terdekat iterasi kedua
 $= 0 + 0,45 + 0 + \dots + 0,53 = 4,09$

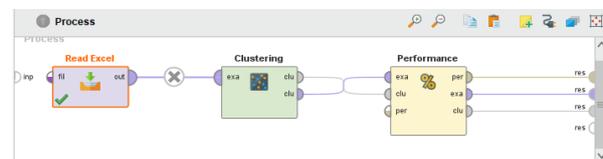
Simpangan dari dua hasil penjumlahan jarak terdekat dari setiap iterasi adalah
 $S = 4,09 - 6,96 = -2,87$

Berdasarkan perhitungan di atas nilai $S < 0$, maka dilakukan perhitungan berikutnya hingga nilai $S > 0$.

Setelah memperoleh *medoid* baru, maka proses (b) hingga (d) akan diulang hingga nilai simpangan yang ditemukan lebih dari nol. Proses berhenti pada iterasi kedua untuk data sebelum pandemi Covid-19 dan data selama pandemi Covid-19.

3.2. Proses Data Mining Menggunakan Aplikasi Rapid Miner

Pemodelan K-Means dengan *Rapid Miner* dibagi menjadi 2 proses yaitu proses klasterisasi koridor TransJakarta sebelum masa pandemi Covid-19 dan selama masa pandemi Covid-19. Proses pemodelan antara data sebelum dan selama pandemi adalah sama, yang membedakan adalah sumber data yang digunakan. Proses pemodelan terlihat pada Gambar 1.



Gambar 1. Pemodelan K-Means dengan *Rapid Miner*

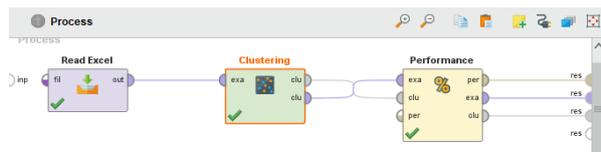
Pada langkah ini dilakukan 4 kali percobaan proses klasterisasi dengan jumlah *cluster* mulai 2, 3, 4, dan 5 *cluster*. Hasil klasterisasi dapat dilihat pada Tabel 9.

Tabel 9. Hasil Pemodelan *Rapid Miner* (K-Means)

Masa	Jumlah Cluster	Nama Cluster	Jumlah Anggota
Sebelum Pandemi	2	Cluster 0	8

Masa	Jumlah Cluster	Nama Cluster	Jumlah Anggota
	3	Cluster 1	5
		Cluster 0	5
		Cluster 1	1
	4	Cluster 2	7
		Cluster 0	7
		Cluster 1	2
		Cluster 2	3
	5	Cluster 3	1
		Cluster 0	6
		Cluster 1	2
		Cluster 2	2
		Cluster 3	2
Selama Pandemi	2	Cluster 0	8
		Cluster 1	5
	3	Cluster 0	5
		Cluster 1	4
		Cluster 2	4
	4	Cluster 0	1
		Cluster 1	7
		Cluster 2	2
		Cluster 3	3
	5	Cluster 0	5
		Cluster 1	2
		Cluster 2	2
Cluster 3		2	
Cluster 4		2	

Pemodelan K-Medoids dengan *Rapid Miner* dibagi menjadi 2 proses yaitu proses klasterisasi koridor TransJakarta sebelum masa pandemi Covid-19 dan selama masa pandemi Covid-19. Proses pemodelan antara data sebelum dan selama pandemi adalah sama, yang membedakan adalah sumber data yang digunakan. Pemodelan terlihat pada Gambar 2.



Gambar 2. Pemodelan K-Medoids dengan *Rapid Miner*

Pada langkah ini dilakukan 4 kali percobaan proses klasterisasi dengan jumlah *cluster* mulai 2, 3, 4, dan 5 *cluster*. Hasil klasterisasi dapat dilihat pada Tabel 10 berikut.

Tabel 10. Hasil Pemodelan *Rapid Miner* (K-Medoids)

Masa	Jumlah Cluster	Nama Cluster	Jumlah Anggota
Sebelum Pandemi	2	Cluster 0	8
		Cluster 1	5
	3	Cluster 0	1
		Cluster 1	7
		Cluster 2	5
	4	Cluster 0	1
Cluster 1		5	

Masa	Jumlah Cluster	Nama Cluster	Jumlah Anggota	
	5	Cluster 2	2	
		Cluster 3	5	
		Cluster 0	1	
	5	Cluster 1	2	
		Cluster 2	3	
		Cluster 3	5	
		Cluster 4	2	
	Selama Pandemi	2	Cluster 0	7
			Cluster 1	6
		3	Cluster 0	1
			Cluster 1	5
			Cluster 2	7
4		Cluster 0	1	
		Cluster 1	1	
		Cluster 2	4	
		Cluster 3	7	
5		Cluster 0	4	
		Cluster 1	4	
		Cluster 2	1	
	Cluster 3	3		
	Cluster 4	1		

3.3. Proses *Data Mining* Menggunakan Pemrograman Python

Pemodelan K-Means dengan pemrograman Python dibagi menjadi 2 proses yaitu proses klasterisasi koridor TransJakarta sebelum masa pandemi Covid-19 dan selama masa pandemi Covid-19. Pada langkah ini dilakukan 4 kali percobaan proses klasterisasi dengan jumlah *cluster* mulai 2, 3, 4, dan 5 *cluster*. Hasil klasterisasi dapat dilihat pada Tabel 11 berikut.

Tabel 11. Hasil Pemodelan Python (K-Means)

Masa	Jumlah Cluster	Nama Cluster	Jumlah Anggota
Sebelum Pandemi	2	Cluster 0	8
		Cluster 1	5
	3	Cluster 0	5
		Cluster 1	7
		Cluster 2	1
	4	Cluster 0	7
		Cluster 1	3
		Cluster 2	1
		Cluster 3	2
	5	Cluster 0	4
		Cluster 1	3
		Cluster 2	1
Cluster 3		2	
Cluster 4		3	
Selama Pandemi	2	Cluster 0	8
		Cluster 1	5
	3	Cluster 0	5
		Cluster 1	7
		Cluster 2	1
	4	Cluster 0	1
		Cluster 1	7

Masa	Jumlah Cluster	Nama Cluster	Jumlah Anggota
		Cluster 2	1
		Cluster 3	4
	5	Cluster 0	1
		Cluster 1	6
Cluster 2		1	
Cluster 3		2	
		Cluster 4	3

Masa	Jumlah Cluster	Nilai DBI (K-Means)	Nilai DBI (K-Medoids)
Sebelum Pandemi	2	0,246	0,243
	3	0,184	0,200
	4	0,270	0,261
	5	0,272	0,247
	Selama Pandemi	2	0,188
	3	0,323	0,202
	4	0,238	0,190
	5	0,291	0,251

Pemodelan K-Medoids dengan pemrograman Python dibagi menjadi 2 proses yaitu proses klasterisasi koridor TransJakarta sebelum masa pandemi Covid-19 dan selama masa pandemi Covid-19. Pada langkah ini dilakukan 4 kali percobaan proses klasterisasi dengan jumlah *cluster* mulai 2, 3, 4, dan 5 *cluster*. Hasil klasterisasi dapat dilihat pada Tabel 12 berikut.

Tabel 12. Hasil Pemodelan Python (K-Medoids)

Masa	Jumlah Cluster	Nama Cluster	Jumlah Anggota	
Sebelum Pandemi	2	Cluster 0	8	
		Cluster 1	5	
	3	Cluster 0	5	
		Cluster 1	4	
		Cluster 2	4	
	4	Cluster 0	4	
		Cluster 1	3	
		Cluster 2	4	
		Cluster 3	2	
	5	Cluster 0	1	
		Cluster 1	3	
		Cluster 2	4	
		Cluster 3	2	
		Cluster 4	3	
	Selama Pandemi	2	Cluster 0	7
Cluster 1			6	
3		Cluster 0	4	
		Cluster 1	3	
		Cluster 2	6	
4		Cluster 0	3	
		Cluster 1	3	
		Cluster 2	3	
		Cluster 3	4	
5		Cluster 0	2	
		Cluster 1	3	
		Cluster 2	2	
		Cluster 3	2	
			Cluster 4	4

Hasil *Cluster* yang terbentuk sebelum dan selama masa pandemi Covid-19 selanjutnya dievaluasi menggunakan *Davies Bouldin Index* (DBI) seperti ditunjukkan pada Tabel 13.

Tabel 13. Nilai DBI

Dari percobaan yang dilakukan, algoritma K-Means untuk data sebelum masa pandemi, jumlah 3 *cluster* menghasilkan kualitas *cluster* yang lebih baik. Hasil evaluasi menunjukkan jumlah 3 *cluster* lebih optimal dengan nilai DBI paling kecil, yaitu sebesar 0,184. Sementara itu, selama masa pandemi, jumlah 2 *cluster* menghasilkan kualitas *cluster* yang lebih baik. Hasil evaluasi menunjukkan jumlah 2 *cluster* lebih optimal dengan nilai DBI paling kecil, yaitu sebesar 0,188.

Untuk algoritma K-Medoids, data sebelum masa pandemi, jumlah 3 *cluster* menghasilkan kualitas *cluster* yang lebih baik. Hasil evaluasi menunjukkan jumlah 3 *cluster* lebih optimal dengan nilai DBI paling kecil, yaitu sebesar 0,200. Sementara itu selama masa pandemi, jumlah 4 *cluster* menghasilkan kualitas *cluster* yang lebih baik. Hasil evaluasi menunjukkan jumlah 4 *cluster* lebih optimal dengan nilai DBI paling kecil, yaitu sebesar 0,190.

Hasil *cluster* optimal yang terbentuk ditunjukkan pada Tabel 14 berikut, indeks *cluster* hasil perhitungan manual disesuaikan dengan aplikasi yang dimulai dengan *cluster* 0.

Tabel 14. Hasil Cluster Sebelum Pandemi

Koridor	K-Means (k=3)			K-Medoids (k=3)			Cluster
	RM*	P*	M*	RM*	P*	M*	
1	1	2	0	0	1	0	0
2	0	0	1	2	2	2	2
3	2	1	2	2	0	2	2
4	2	1	2	1	0	2	2
5	2	1	2	1	1	0	1
6	2	1	2	1	1	2	1
7	2	1	2	1	0	0	1
8	0	0	1	2	2	1	1
9	0	0	1	2	2	1	1
10	0	0	1	2	1	2	1
11	2	1	2	1	0	2	2
12	0	0	1	2	2	2	2
13	2	1	2	1	0	0	1

Keterangan: RM: *Rapid Miner*, P: Python, M: Manual
 Pada Tabel 14 di atas, dengan menggunakan metode *weighted majority voting*, *cluster* yang dihasilkan adalah sebagai berikut:

Cluster 0, sebanyak 1 koridor yaitu Koridor 1. *Cluster 0* tersebut merupakan koridor dengan jumlah penumpang 19,33 juta, dengan panjang rute 12,90 km, dan jumlah halte 17.

Cluster 1, sebanyak 7 koridor yang meliputi Koridor 5, Koridor 6, Koridor 7, Koridor 8, Koridor 9, Koridor 10, dan Koridor 13. *Cluster 1* tersebut merupakan koridor dengan rata-rata jumlah penumpang 8,23 juta, dengan rata-rata panjang rute 17,74 km, dan rata-rata jumlah halte 19,29.

Cluster 2, sebanyak 5 koridor yang meliputi Koridor 2, Koridor 3, Koridor 4, Koridor 11, dan Koridor 12. *Cluster 2* tersebut merupakan koridor dengan rata-rata jumlah penumpang 5,12 juta, dengan rata-rata panjang rute 18,76 km, dan rata-rata jumlah halte 21,20.

Tabel 15. Hasil *Cluster* Selama Pandemi

Koridor	K-Means (k=2)			K-Medoids (k=4)			<i>Cluster</i> Majority Voting
	RM*	P*	M*	RM*	P*	M*	
1	0	0	1	3	3	3	3
2	1	1	0	2	2	0	1
3	0	0	1	3	3	2	0
4	0	0	1	3	1	3	1
5	0	0	1	3	0	3	0
6	0	0	1	3	1	3	1
7	0	0	1	3	0	3	0
8	1	1	0	2	2	2	2
9	1	1	0	0	2	1	1
10	1	1	0	2	3	2	2
11	0	0	1	3	1	3	0
12	1	1	0	2	2	2	2
13	0	0	1	1	0	3	0

Keterangan: RM: *Rapid Miner*, P: Python, M: Manual
 Pada Tabel 15 di atas, dengan menggunakan metode *majority voting*, *cluster* yang dihasilkan adalah sebagai berikut:

Cluster 0, sebanyak 5 koridor yang meliputi Koridor 3, Koridor 5, Koridor 7, Koridor 11, dan Koridor 13. *Cluster 0* tersebut merupakan koridor dengan rata-rata jumlah penumpang 2,27 juta, dengan rata-rata panjang rute 13,92 km, dan rata-rata jumlah halte 15,20.

Cluster 1, sebanyak 4 koridor yang meliputi Koridor 2, Koridor 4, Koridor 6, dan Koridor 9. *Cluster 1* tersebut merupakan koridor dengan rata-rata jumlah penumpang 2,77 juta, dengan rata-rata panjang rute 19,81 km, dan rata-rata jumlah halte 24.

Cluster 2, sebanyak 3 koridor yang meliputi Koridor 8, Koridor 10, dan Koridor 12. *Cluster 2* tersebut merupakan koridor dengan rata-rata jumlah penumpang 2,04 juta, dengan rata-rata panjang rute 23,05 km, dan rata-rata jumlah halte 23.

Cluster 3, sebanyak 1 koridor yang meliputi Koridor 1. *Cluster 3* tersebut merupakan koridor dengan jumlah penumpang 5,61 juta, dengan panjang rute 12,90 km, dan jumlah halte 17.

Ringkasan hasil klasterisasi yang telah dijelaskan di atas, dapat dilihat pada Tabel 16.

Tabel 16. Hasil Majority Voting

	Koridor	Jumlah Penumpang	Panjang Rute	Jumlah Halte
Sebelum Pandemi				
<i>Cluster 0</i>	1	19,33 juta	12,90 km	17
<i>Cluster 1</i>	5, 6, 7, 8, 9, 10, 13	8,23 juta	17,74 km	19,29
<i>Cluster 2</i>	2, 3, 4, 11, 12	5,12 juta	18,76 km	21,20
Selama Pandemi				
<i>Cluster 0</i>	3, 5, 7, 11, 13	2,27 juta	13,92 km	15,20
<i>Cluster 1</i>	2, 4, 6, 9	2,77 juta	19,81 km	24
<i>Cluster 2</i>	8, 10, 12	2,04 juta	23,05 km	23
<i>Cluster 3</i>	1	5,61 juta	12,90 km	17

Dari tabel 16 di atas dapat dilihat bahwa, sebelum masa pandemi, *Cluster 0* adalah koridor 1 (Blok M-Kota) dengan jumlah penumpang tertinggi dengan rute yang pendek dan jumlah halte sedikit. *Cluster 1* adalah koridor-koridor yang sebagian besar berada di wilayah Jakarta Selatan dan sebagian Jakarta Timur, dengan jumlah penumpang sedang, panjang rute sedang, dan jumlah halte sedang. *Cluster 2* adalah koridor-koridor yang sebagian besar berada di wilayah Jakarta Barat dan sebagian Jakarta Timur dengan jumlah penumpang paling sedikit, dengan rute terpanjang, dan jumlah halte terbanyak.

Selama masa pandemi, *cluster 0* adalah koridor-koridor sebagian besar di wilayah Jakarta Timur, dengan jumlah penumpang sedang, panjang rute sedang, dan jumlah halte paling sedikit. *Cluster 1* adalah koridor-koridor yang sebagian besar berada di wilayah Jakarta Selatan, dengan jumlah penumpang sedang, panjang rute sedang, dan jumlah halte terbanyak. *Cluster 2* adalah koridor-koridor yang sebagian besar berada di wilayah Jakarta Barat, dengan jumlah penumpang paling kecil, dengan rute terpanjang, dan jumlah halte sedang. *Cluster 3* adalah koridor 1 (Blok M-Kota) dengan jumlah penumpang terbanyak, dengan rute dan jumlah halte sedang..

4. Kesimpulan

Penerapan metode K-Means dan K-Medoids dalam pengelompokan data koridor TransJakarta dapat

menghasilkan beberapa kelompok koridor sebelum masa pandemi Covid-19 dan selama masa pandemi Covid-19. Hasil klusterisasi dapat dijadikan rujukan bagi manajemen TransJakarta untuk mengatur alokasi sumber daya, baik itu armada, personel, maupun jadwal pada tiap-tiap koridor. Berdasarkan hasil pengelompokan yang diperoleh dari penggunaan algoritma K-Means sebelum masa pandemi, jumlah *cluster* optimal adalah 3 *cluster* dengan nilai DBI 0,184, dan selama masa pandemi jumlah *cluster* optimal adalah 2 *cluster* dengan nilai DBI 0,188. Sementara itu, bila menggunakan algoritma K-Medoids sebelum masa pandemi, jumlah *cluster* optimal adalah 3 *cluster* dengan nilai DBI 0,200, dan selama masa pandemi jumlah *cluster* optimal adalah 4 *cluster* dengan nilai DBI 0,190.

Pada masa mendatang, penelitian ini dapat dikembangkan dengan menambahkan atribut lain misalnya jumlah transaksi, total nilai transaksi, jumlah transaksi tiap-tiap kartu elektronik, dan atribut lainnya, sehingga *cluster* yang dihasilkan dapat lebih optimal. Dengan pengembangan atribut diharapkan pengelola TransJakarta memperoleh analisis yang lebih mendalam, berkaitan dengan pelayanan di halte, penyediaan alat baca kartu elektronik, atau unsur pengelolaan lainnya.

Daftar Rujukan

- [1] Giovanni L. A. and Najid N., 2020. Pengaruh ERP Terhadap Perbaikan Pelayanan Waktu Tunggu Dan Waktu Tempuh Transjakarta Pada Ruas Jalan Sudirman-Thamrin. *JMTS: Jurnal Mitra Teknik Sipil*, vol.3, no. 4, p. 959.
- [2] Defiyanti S., Jajuli M., and Rohmawati N., 2017. Optimalisasi K-MEDOID dalam Pengklasteran Mahasiswa Pelamar Beasiswa dengan CUBIC CLUSTERING CRITERION. *Jurnal Nasional Teknologi dan Sistem Informasi*, vol.3, no. 1, pp. 211–218.
- [3] Supriyatna A., Carolina I., Janti S., and Haidir A., 2020. Clustering Koridor Transjakarta Berdasarkan Jumlah Penumpang Dengan Algoritma K-Means. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol.4, no. September, pp. 682–693.
- [4] Indriyani F. and Irfiani E., 2019. Clustering Data Penjualan pada Toko Perlengkapan Outdoor Menggunakan Metode K-Means. *JUITA: Jurnal Informatika*, vol.7, no. 2, p. 109.
- [5] Sukanto S., Id I. D., and Angraini T. R., 2018. Penentuan Daerah Rawan Titik Api di Provinsi Riau Menggunakan Clustering Algoritma K-Means. *JUITA: Jurnal Informatika*, vol.6, no. 2, p. 137.
- [6] Supriyatna A., Carolina I., Widiati W., and Nuraeni C., 2020. Rice Productivity Analysis by Province Using K-Means Cluster Algorithm. *IOP Conference Series: Materials Science and Engineering*, vol.771, no. 1.
- [7] Gunawan I., Anggraeni G., Rini E. S., and Mustofa Y., 2020. Klusterisasi provinsi di Indonesia berbasis perkembangan kasus Covid-19 menggunakan metode K-Medoids. *Seminar Nasional Matematika dan Pendidikan Matematika (5th SENATIK)*, pp. 301–306.
- [8] Juninda T., Mustasim, and Andri E., 2019. Penerapan Algoritma K-Medoids untuk Pengelompokan Penyakit di Pekanbaru Riau. *Seminar Nasional Teknologi Informasi, Komunikasi dan Industri*, vol.11, no. 1, pp. 42–49.
- [9] Marlina D., Lina N., Fernando A., and Ramadhan A., 2018. Implementasi Algoritma K-Medoids dan K-Means untuk Pengelompokan Wilayah Sebaran Cacat pada Anak. *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi*, vol.4, no. 2, p. 64.
- [10] Pramesti D. F., Lahan, Tanzil Furqon M., and Dewi C., 2017. Implementasi Metode K-Medoids Clustering Untuk Pengelompokan Data Potensi Kebakaran Hutan/Lahan Berdasarkan Persebaran Titik Panas (Hotspot). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol.1, no. 9, pp. 723–732.
- [11] Retno Utari D. and Wibowo A., 2020. Pemodelan Prediksi Status Keberlanjutan Polis Asuransi Kendaraan dengan Teknik Pemilihan Mayoritas Menggunakan Algoritma-Algoritma Klasifikasi Data Mining. *Prosiding Seminar Nasional Teknoka*, vol.5, no. 2502, pp. 19–24.
- [12] Praja B. S., Kusuma P. D., and Setianingsih C., 2019. Penerapan Metode K-Means Clustering Dalam Pengelompokan Data Penumpang Dan Kapal Angkutan Laut Di Indonesia. *e-Proceeding of Engineering*, vol.06, no. 1, p. 1442.
- [13] Nugroho D., Nhita F., and Trantor D., 2016. Prediksi Penyakit Menggunakan Genetic Algorithm (GA) dan Naive Bayes Untuk Data Berdimensi Tinggi Prediction of Disease Using Genetic Algorithm (GA) and Naive Bayes For Data High Dimension. *e-Proceeding of Engineering*, vol.3, no. 2, pp. 3889–3899.
- [14] Ningrat D. R., Maruddani D. A. I., and Wuryandari T., 2016. Analisis Cluster Dengan Algoritma K-Means Dan Fuzzy C-Means Clustering Untuk Pengelompokan Data Obligasi Korporasi. *Jurnal Gaussian*, vol.5, no. 4, pp. 641–650.
- [15] Bastian A., Sujadi H., and Febrianto G., 2018. Penerapan Algoritma K-Means Clustering Analysis Pada Penyakit Menular Manusia (Studi Kasus Kabupaten Majalengka). *Jurnal Sistem Informasi*, vol.14, no. 1, pp. 26–32.
- [16] Sindi S., Ningse W. R. O., Sihombing I. A., Ilmi R.H.Zer F., and Hartama D., 2020. Analisis Algoritma K-Medoids Clustering Dalam Pengelompokan Penyebaran Covid-19 Di Indonesia. *JTI (Jurnal Teknologi Informasi)*, vol.4, no. 1, pp. 166–173.
- [17] Listiyanti D., Syahbana Y. A., and Henim S. R., 2016. Perancangan dan Implementasi Aplikasi Android Penentu Salient Area pada Video dengan Algoritma K-Medoids. *Annual Research Seminar: Computer Science and Information and Communications Technology*, vol.2, no. 1, pp. 96–101.
- [18] Mahartika I. R. and Wibowo A., 2019. Data Mining Klusterisasi dengan Algoritma K-Means untuk Pengelompokan Provinsi Berdasarkan Konsumsi Bahan Bakar Minyak Nasional. *Prosiding Seminar Nasional SISFOTEK (Sistem Informasi dan Teknologi)*, vol.3, no. 1, pp. 87–91.
- [19] Satmoko D. B., Sukarno P., and Jaded E. M., 2018. Peningkatan Akurasi Pendeteksian Serangan DDoS Menggunakan Multiclassifier Ensemble Learning dan Chi-Square Pendahuluan Studi Terkait, vol.5, no. 3, pp. 7977–7985.
- [20] Patil D. R. and Patil J. B., 2018. Malicious URLs detection using decision tree classifiers and majority voting technique. *Cybernetics and Information Technologies*, vol.18, no. 1, pp. 11–29.
- [21] Alotaibi B. and Elleithy K., 2016. A majority voting technique for Wireless Intrusion Detection Systems. *2016 IEEE Long Island Systems, Applications and Technology Conference, LISAT 2016*, no. October 2017.