IDENTIFIKASI HOTSPOT AREA DAN WAKTU PENJEMPUTAN TAKSI MENGGUNAKAN SPASIAL CLUSTERING BERBASIS DENSITAS

Disusun Oleh:

N a m a : Mulia Dea Lestari
NIM : 19523119

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2023

HALAMAN PENGESAHAN DOSEN PEMBIMBING

IDENTIFIKASI HOTSPOT AREA DAN WAKTU PENJEMPUTAN TAKSI MENGGUNAKAN SPASIAL CLUSTERING BERBASIS DENSITAS

TUGAS AKHIR

Disusun Oleh:

N a m a	: Mulia Dea Lestari
NIM	: 19523119

Yogyakarta, 29 Mei 2023

Pembimbing,

(Lizda Iswari, S.T., M.Sc.)

HALAMAN PENGESAHAN DOSEN PENGUJI

IDENTIFIKASI HOTSPOT AREA DAN WAKTU PENJEMPUTAN TAKSI MENGGUNAKAN SPASIAL CLUSTERING BERBASIS DENSITAS

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 29 Mei 2023

Tim Penguji

Lizda Iswari, S.T., M.Sc. _____

Anggota 1

Novi Setiani, S.T., M.T. _____

Anggota 2

Arrie Kurniawardhani, S.Si., M.Kom. _____

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	:	Mulia Dea Lestari
NIM	:	19523119

Tugas akhir dengan judul:

IDENTIFIKASI HOTSPOT AREA DAN WAKTU PENJEMPUTAN TAKSI MENGGUNAKAN SPASIAL CLUSTERING BERBASIS DENSITAS

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 22 Mei 2023

(Mulia Dea Lestari)

HALAMAN PERSEMBAHAN

Alhamdulillahirobbil'alamin, puji syukur kepada Allah SWT atas berkat dan rahmat-Nya telah memberikan kemudahan dan kelancaran dalam proses pembuatan Laporan Tugas Akhir sehingga bisa selesai tepat waktu. Shalawat serta salam kita limpahkan kepada junjungan Nabi Muhammad SAW yang telah membawa umatnya menuju kebajikan.

Saya ucapkan Terima kasih tak terhingga kepada kedua orang tua saya Ibu Samsinar dan Ayah Slamet, yang selalu memberikan dukungan, usaha, nasehat, setiap hari selalu mendoakan saya agar bisa segera selesai menempuh perkuliahan, dan segala bentuk kasih sayang hingga saat ini. Terima kasih juga kepada kakak saya, Chrisna Asry Rahayu yang selalu memberikan semangat, waktu, nasehat, serta doa untuk saya. Terima kasih kepada keluarga besar atas doa nya dan segala kebaikan, yang tidak bisa saya sebutkan satu persatu.

Saya ucapkan terima kasih kepada Ibu Lizda Iswari, S.T., M.Sc., selaku dosen pembimbing yang telah meluangkan waktu untuk membimbing, mengarahkan, memberi saran, pengetahuan baru, motivasi, serta dukungan.

Terima kasih kepada seluruh dosen Informatika UII yang telah memberikan ilmu dan pelajaran hidup selama saya menempuh kuliah disini. Semoga apa yang telah diberikan, menjadi amal jariyah dan berkah.

Terima kasih kepada teman-teman terdekat saya yang telah memberikan dukungan dan semangat, bantuan, waktu selama saya berkuliah di UII sampai selesainya proses penyusunan laporan Tugas Akhir ini.

Last but not least, semoga Allah selalu senantiasa menjunjung kita kedalam kebaikan, serta seluruh ilmu yang diperoleh menjadi berkah kelak bagi diri sendiri maupun orang lain, dan semoga kita diberikan kelancaran, kemudahan, serta tidak mudah menyerah terhadap kesulitan dalam ujian hidup. Aamiin ya robbal alamin

HALAMAN MOTO

“Manusia hanya merencanakan, Tuhanlah yang menentukan”

“Jika Tuhan memberi mu ujian, ingatlah! di setiap ujian selalu ada hikmah dan pesan yang tidak disangka-sangka”

“Everything is not always possible, seperti Kamu tidak bisa menjadikan dirimu sebagai Tuhan”

“Kalau kau hanya mengagumi seseorang, maka kau tidak akan bisa mengalahkannya”
Kise Ryouta

“Jika kamu tidak pernah merasakan kegagalan, artinya dirimu tidak berkembang”

“Kuliah tidak menjamin hidupmu sukses, tetapi dengan kuliah kamu memperbesar peluang untuk bisa jadi lebih baik”

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh

Alhamdulillahirobbil'alamin. Saya panjatkan puji syukur kepada Allah SWT yang Maha Pengasih lagi Maha Penyayang atas berkat dan rahmat-Nya sehingga laporan Tugas Akhir ini dapat selesai dengan baik, dan tepat waktu. Shalawat serta salam kita curahkan kepada Nabi Muhammad SAW dan para sahabatnya. Berkat rahmat dan kuasa-Nya penulis dapat menyelesaikan laporan Tugas Akhir mengenai identifikasi hotspot area dan waktu penjemputan taksi menggunakan spasial clustering berbasis densitas.

Atas penyusunan dan penyelesaian laporan ini, telah banyak pihak yang memberikan bantuan, semangat, bimbingan dan doa. Oleh karena itu, izinkan penulis untuk mengucapkan rasa syukur dan terima kasih sebesar-besarnya kepada:

1. Orang tua tercinta (Ibu Samsinar dan Ayah Slamet) dan satu-satunya kakak ku, yang tidak henti-hentinya mendoakan saya, kasih sayang, nasehat, dukungan, motivasi, dan hal lain selama saya hidup hingga masa perkuliahan bisa diselesaikan dengan baik.
2. Ibu Lizda Iswari, S.T., M.Sc. selaku dosen pembimbing skripsi atas segala waktu, bantuan, dan dukungan yang diberikan selama proses penyusunan laporan ini dibuat hingga selesai.
3. Seluruh dosen Program Studi Informatika Fakultas Teknologi Industri Universitas Islam Indonesia yang telah memberikan ilmu, pengalaman, serta pelajaran hidup yang berharga, semoga berkah dan Allah meberikan balasan kebaikan ini
4. Kepada Insannur Kamil Malik terima kasih telah mendukung, membantu, dan memberi motivasi.
5. Kepada teman-teman terdekatku (Karina Khoiriyah Pertiwi, Aulia Safira Ahda, Fita Maulani Mahfud, Khoiri Rochmanila, Meiyani Oslim, Devi Rizki Dwi Ananda, Rahmatia Sumajayanti, Ervina Novita, Iin Nurintan Safitri) terima kasih telah saling mendukung, memberikan bantuan, motivasi, rasa kekeluargaan yang tercipta selama di Jogja. Semoga kedepannya, kita tetap menjalin tali silaturahmi dengan baik di dunia maupun akhirat, dan kelak bertemu lagi dalam keadaan yang lebih baik dan sukses.
6. Teman-teman Informatika UII Angkatan 2019, atas segala bantuan yang diberikan selama masa perkuliahan.
7. Teman-teman diluar Informatika UII
8. Seluruh pihak yang telah membantu penulis yang tidak bisa disebutkan satu per satu.

Penulis menyadari bahwa penelitian ini masih belum sempurna, dan semoga kedepannya bisa di kembangkan lagi. Penulis mengharapkan, semoga penelitian ini bermanfaat bagi penulis, jasa transportasi, maupun orang lain. Semoga Allah SWT memberikan berkah dan karunia atas kebaikan semua pihak yang terlibat secara langsung ataupun tidak langsung dalam penyelesaiannya tugas akhir ini, Aamiin ya Rabbal alamin.

Yogyakarta, 5 Mei 2023

(Mulia Dea Lestari)

SARI

Taksi adalah segmen industri transportasi yang kompetitif dalam moda transportasi darat dan dianggap sebagai sarana transportasi yang nyaman serta mudah untuk memenuhi kebutuhan individu. Sulitnya mencari taksi pada jam-jam tertentu, ketidakseimbangan antara permintaan dan persediaan taksi, serta lamanya pelanggan harus menunggu taksi adalah beberapa masalah yang mungkin membuat pengoperasian taksi menjadi kurang optimal. Oleh karena itu, diperlukan landasan pengetahuan untuk pengambilan keputusan manajemen strategis guna memaksimalkan layanan taksi. Tujuan dari penelitian ini adalah menganalisis data operasional layanan taksi yang dapat digunakan untuk mengenali wilayah serta waktu-waktu permintaan layanan taksi yang tinggi. Data diolah menggunakan metode *clustering* berbasis densitas DBSCAN dan pengukuran performa pemodelan hasil *clustering* menggunakan metode *silhouette coefficient*.

Penelitian ini mengambil data set yang tersedia secara publik dan terbuka, yaitu data perjalanan taksi di New York City. Adapun area studi berfokus pada wilayah Queens, New York City untuk bulan januari, februari, dan maret 2016. Dalam prosesnya, terdapat beberapa tahapan, yaitu penyaringan *dataset, pre-processing,* proses *clustering*, implementasi, dan pengujian. Hasil menunjukkan bahwa dari bulan januari, februari, dan maret terbentuk cluster yang memiliki pola yang mirip pada area cluster mayor yang berada di wilayah LaGuardia Airport. Kemudian, jumlah penjemputan di setiap jam pada ketiga bulan tidak memiliki pola perbedaan yang signifikan. Selanjutnya, diimplementasikan berbentuk aplikasi website dengan menggunakan framework Streamlit, dan dilakukan pengujian pada aplikasi menggunakan *black box testing*. Dengan demikian, hasil dari penelitian ini dapat dimanfaatkan sebagai bantuan rekomendasi keputusan untuk meningkatkan jasa pelayanan taksi.

Kata kunci: Taksi, Layanan Taksi, *Clustering*, DBSCAN

GLOSARIUM

Clustering	mengelompokkan objek berdasarkan kemiripan satu sama lain
DBSCAN	salah satu metode untuk *clustering* data
Outliers	titik data yang menyimpang atau berbeda dari kumpulan titik data lainnya
Streamlit	*framework* berbasis python untuk implementasi aplikasi web yang interaktif

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Salah satu industri yang bersaing di Indonesia adalah transportasi. Transportasi adalah komponen utama dalam sistem kehidupan, pemerintahan serta kemasyarakatan (Aminah, 2018). Usaha transportasi tidak hanya gerakan barang ataupun orang dari satu tempat ke tempat lain dengan kondisi statis, tetapi juga perlu diusahakan perbaikan dan peningkatan sesuai perkembangan teknologi (Rivai, 2020). Jika dilihat lebih spesifik moda transportasi, menurut penelitian (Athoillah, Firdaus, & Sanim, 2019) taksi merupakan salah satu industri yang bersaing ketat. Menurut *International Association of Public Transport* (UITP) fleksibilitas pelayanan taksi mendorong pertumbuhan dan popularitas di industri secara global (UITP, 2020). Menurut dari portal data Statista pendapatan pada segmen taksi di seluruh dunia pada tahun 2023 diperkirakan mencapai US$330,80 miliar dengan pertumbuhan pendapatan sebesar 14.4% ("Ride-Hailing & Taxi - Worldwide | Statista Market Forecast," 2022). Taksi diakui sebagai moda transportasi yang nyaman (Ulak, Yazici, & Aljarrah, 2020) serta dinilai lebih aman dari kendaraan umum lain (Rizan, Fadillah, & P, 2015). Selain itu, taksi dapat berkontribusi terhadap pengurangan kemacetan lalu lintas, mengurangi polusi udara, serta konsumsi energi (Aminah, 2018; He et al., 2013). Di Indonesia terdapat berbagai macam *brand* taksi yang populer seperti Gojek, Grab, Maxim, Blue Bird, dan sebagainya.

Keberadaan taksi memiliki peran penting dalam penggunaan transportasi umum bagi masyarakat untuk bepergian harian (Jian, Li, & Yu, 2021). Namun dalam pengoperasiannya, pelayanan taksi dihadapkan dengan beberapa masalah dikarenakan meningkatnya pertumbuhan penduduk (Kong, Xia, Wang, Rahim, & Das, 2017). Beberapa masalah dari pelayanan taksi diantaranya: sulitnya menemukan taksi saat jam sibuk (Wang, Zhang, Wang, & Ning, 2018), ketidakseimbangan antara permintaan dan persediaan taksi (Wong, Szeto, & Wong, 2014), tidak mengetahui lokasi pangkalan taksi yang tepat (Qu, Wang, Song, Pan, & Li, 2019), dan lamanya penumpang dalam menunggu taksi. Dari adanya beberapa permasalahan tersebut dapat dikenali bahwa area dan waktu penjemputan taksi merupakan hal yang penting untuk peningkatan pelayanan taksi. Kepuasan pelanggan terhadap pelayanan taksi memengaruhi citra perusahaan sehingga pentingnya bagi perusahaan untuk memperbaiki kualitas pelayanan jasa taksi (Angraini, 2018). Oleh karena itu, identifikasi hotspot area dan

waktu penjemputan taksi penting untuk dikenali sebagai dasar dalam pengambilan keputusan untuk strategi pengelolaan jasa perusahaan taksi sebagai bentuk optimalisasi pelayanan jasa taksi.

Teknologi GPS telah banyak digunakan pada taksi sehingga pengumpulan data perjalanan secara *real time* (Wang et al., 2018). Data yang dikumpulkan dapat diproses dan dianalisis dari data lintasan mentah menjadi pengetahuan yang bermanfaat (Ibrahim & Shafiq, 2019). Pentingnya memahami pola titik perjalanan dari data memberikan peluang yang bagus untuk mendapatkan wawasan dalam mobilitas taksi (Z. Zhou, Yu, Guo, & Liu, 2018) selain itu, dapat memberikan mobilitas yang fleksibel serta nyaman bagi para masyarakat (Cai et al., 2016). Oleh karena itu, pentingnya bagaimana menambang lintasan data untuk meningkatkan layanan taksi (Wang et al., 2018).

Dalam mengidentifikasi area dan waktu penjemputan taksi menggunakan metode clustering berbasis densitas dengan algoritma DBSCAN. Metode *clustering* dalam ranah spasial dapat berguna di berbagai penerapan (Amiruzzaman, Rahman, Islam, & Nor, 2022). *Clustering* adalah proses membagi sejumlah besar data menjadi beberapa kelompok sesuai dengan karakteristik masing-masing kelas, dan algoritma *clustering* yang cocok untuk mengidentifikasi klaster berdasarkan kepadatan data adalah algoritma DBSCAN (Almantara, Aryani, & Swamardika, 2020). Algoritma *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) merupakan pendekatan pengelompokan berbasis lokasi yang digunakan untuk menemukan hubungan dan pola dalam data geografis (Amiruzzaman et al., 2022). DBSCAN adalah algoritma pengelompokkan berbasis kepada daerah kepadatan data tinggi dan menemukan *cluster* dalam data spasial yang mengandung *noise* didalamnya (Almantara et al., 2020).

Selanjutnya, hasil dari *clustering* akan dianalisis secara statistik untuk menghasilkan informasi secara lebih detail mengenai kecenderungan karakteristik dan pola yang ada di area dan waktu penjemputan taksi. Kemudian, akan diimplementasikan dalam bentuk aplikasi website menggunakan framework Streamlit berbasis Python yang bersifat *open-source* untuk membangun aplikasi website secara interaktif di bidang *data science.*

Berdasarkan hasil pemaparan data dan fakta diatas, penelitian ini akan menerapkan metode *clustering* berbasis kepadatan yaitu DBSCAN dalam pemodelan, dan akan diimplementasikan dalam bentuk aplikasi website menggunakan streamlit yang diharapkan dapat digunakan sebagai dasar rekomendasi untuk meningkatkan kualitas layanan taksi serta memberikan pengalaman yang lebih baik bagi para penumpang.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka terdapat rumusan masalah yang akan diselesaikan pada penelitian ini yaitu:

a. Bagaimana mengeksplorasi data perjalanan taksi untuk mengetahui profil data secara statistik, seperti rata-rata durasi perjalanan, rata-rata jarak perjalanan, jumlah permintaan layanan taksi, dan rata-rata kecepatan perjalanan.
b. Bagaimana mengembangkan aplikasi untuk mengenali *cluster* lokasi dan waktu penjemputan menggunakan algoritma *clustering* berbasis densitas DBSCAN.

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini antara lain:

a. Basis pengetahuan berasal dari dataset publik taksi kota New York.
b. Data uji hanya berasal dari dataset publik taksi Queens, kota New York.
c. Data yang digunakan tercatat sejak tanggal 1 Januari hingga 31 Maret tahun 2016 Queens, kota New York.
d. Metode DBSCAN digunakan untuk pengolahan data pada aplikasi yang akan diimplementasikan sebagai bentuk website menggunakan *framework* Streamlit berbasis Python
e. Performa pengukuran hasil clustering menggunakan metode *silhouette coefficient*

1.4 Tujuan Penelitian

Tujuan dari penelitian ini antara lain:

a. Mengetahui profil layanan taksi, seperti rata-rata durasi perjalanan, rata-rata jarak perjalanan, jumlah permintaan layanan taksi, dan rata-rata kecepatan perjalanan
b. Mengetahui *cluster* lokasi yang dijadikan sebagai titik penjemputan penumpang taksi dan waktu permintaan layanan taksi.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat kepada jasa perusahaan taksi antara lain:

a. Dapat dijadikan sebagai dasar rekomendasi strategi pengelolaan dan peningkatan layanan bagi perusahaan taksi swasta.
b. Dapat dijadikan sebagai model untuk meningkatkan pelayanan transportasi publik.

c. Dapat membantu dalam perencanaan transportasi publik

1.6 Metodologi Penelitian

Metodologi dalam tugas akhir ini merupakan tahapan yang dilakukan agar penyusunan lebih terarah. Metodologi yang digunakan antara lain:

a. Penyaringan Dataset

Data yang digunakan pada penelitian tugas akhir ini adalah data yang tersedia di web Kaggle yaitu data perjalanan taksi Queens, di kota New York. Namun, penelitian ini menggunakan dataset tertentu dan akan dilakukan penyaringan dataset. Data tersebut berupa file csv yang akan digunakan untuk pemodelan dan kebutuhan system.

b. Pre-processing

Pre-processing dimulai *feature creation* dan *data cleaning* untuk mempersiapkan data mentah.

c. Proses *Clustering*

Setelah tahap *pre-processing*, kemudian dilakukan proses *clustering* dimulai dari membagi dataset, *feature transformation*, dan pemodelan *clustering* menggunakan metode DBSCAN, dan kemudian dievaluasi menggunakan metric *silhouette coefficient* dengan tujuan mengevaluasi akurasi hasil dari pemodelan.

d. Implementasi

Pada tahap ini adalah mengimplementasikan pemodelan yang telah dibuat dengan menggunakan *framework* Streamlit berbasis Python.

e. Pengujian

Tahap selanjutnya setelah membangun aplikasi adalah mengujinya menggunakan metode *black box*. Pengujian ini bertujuan untuk mengukur kelompok fungsionalitas dari aplikasi dengan kesesuaian tujuan dari yang diharapkan.

1.7 Sistematika Penulisan

Sistem penulisan yang digunakan dalam penelitian ini terbagi ke dalam lima bab yang berisi gambaran dari masalah serta solusinya. Berikut uraian sistematika penulisan pada laporan tugas akhir:

BAB I PENDAHULUAN

Bab ini berisi pembahasan latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi pembahasan mengenai teori-teori yang digunakan untuk mengkaji penelitian, serta tinjauan terhadap penelitian sebelumnya

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang proses tahapan yang dilakukan dalam penelitian, mulai dari mendapatkan data, *pre-processing,* proses *clustering*, implementasi, serta pengujian

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas mengenai implementasi website serta hasil dari semua proses pada tahap metodologi penelitian.

BAB V KESIMPULAN DAN SARAN

Bab ini merupakan bab terakhir yang akan membahas kesimpulan berdasarkan penelitian yang dilakukan dan saran untuk pengembangan penelitian selanjutnya.

BAB II
LANDASAN TEORI

2.1 Data Mining

Data mining adalah proses penggalian data dalam jumlah besar data untuk menemukan pola yang menarik dan berguna untuk mendapatkan pengetahuan, ini dikenal juga dengan istilah *knowledge discovery in databases* (KDD), analisis data/pola, dan ekstraksi pengetahuan (Bharati & Ramageri, 2010 dikutip dalam Prikitiew, n.d., p. 1). Teknik *data mining* digunakan untuk mengidentifikasi pola baru yang sebelumnya belum ditemukan, dan kemudian digunakan untuk membantu keputusan bisnis tertentu (Bharati & Ramageri, 2010). Terdapat berbagai macam teknik dan algoritma dari data mining sebagai berikut (Bharati & Ramageri, 2010):

a. *Clustering*

   Clustering untuk mengidentifikasi objek yang mirip, dapat mengidentifikasi daerah yang padat dan berongga dalam ruang objek, dan dapat belajar mengenai pola distribusi keseluruhan dan hubungan antara variabel data.

b. Klasifikasi

   Teknik *data mining* yang paling banyak digunakan adalah klasifikasi yang menggunakan serangkaian contoh yang dikategorikan sebelumnya untuk membuat model yang dapat mengklasifikasikan sebagian besar *record* data.

c. Prediksi

   Prediksi dapat dilakukan dengan menggunakan teknik regresi. Analisis regresi dapat digunakan untuk memodelkan hubungan antara satu atau lebih variabel bebas dan variabel terikat. Variabel bebas dalam *data mining* adalah karakteristik yang telah diketahui sebelumnya, sedangkan variabel terikat adalah apa yang ingin kita ramalkan.

d. Aturan Asosiasi

   Dalam kumpulan data besar, penemuan kumpulan item yang sering ditemukan menggunakan asosiasi dan korelasi. Kemampuan untuk menghasilkan aturan dengan tingkat kepercayaan di bawah satu merupakan persyaratan untuk algoritma aturan asosiasi. Meskipun begitu, ada kemungkinan aturan asosiasi yang diberikan dalam jumlah besar, dan Sebagian besar dari aturan yang tinggi seringkali memiliki nilai yang kecil.

e. *Neural Networks* (Jaringan saraf)

Neural networks adalah kumpulan unit input/output yang terhubung dan setiap koneksi memiliki bobot. Selama fase pembelajaran, jaringan memodifikasi bobotnya untuk memungkinkan mereka mengantisipasi label kelas yang tepat dari *input tuple*. Jaringan saraf dapat mengekstraksi pola dan mengidentifikasi tren dari data yang kompleks atau ambigu.

2.2 Spatial Data Mining

Spatial data mining adalah proses penemuan yang mengekstraksi pengetahuan umum dari data spasial dengan konsep dasar manipulasi ruang, tampilan data spasial, informasi yang ditemukan serta representasinya (Li, Wang, & Li, 2015). *Spatial data mining* berguna untuk menemukan hubungan antara data spasial ataupun non spasial, khusus *spatial data mining* terdapat pada interaksi dalam ruang yang berakibat basis data geografi adalah spatio-temporal dimana ciri tempat tertentu biasanya terkait satu sama lain dengan ciri lingkungannya (Zeitouni, 2000). Tugas *spatial data mining* yaitu mengidentifikasi kriteria klasifikasi, meringkas data, membuat kelompok objek terkait, mengidentifikasi hubungan dan ketergantungan, dan mengidentifikasi penyimpangan setelah mengidentifikasi tren (Zeitouni, 2000)

2.3 Clustering

Clustering adalah proses membagi sejumlah besar data menjadi beberapa kelompok sesuai dengan karakteristik masing-masing kelas, dan algoritma clustering yang cocok untuk mengidentifikasi klaster berdasarkan kepadatan data adalah algoritma DBSCAN(Almantara et al., 2020). Terdapat beberapa macam metode *clustering* yang telah dikembangkan, menurut (Han & Kamber, 2006) membagi metode *clustering* menjadi dua yaitu metode hierarkis dan metode partisi. Kemudian, menurut (Han & Kamber, 2006) mengkategorikan metode *clustering* menjadi lima kategori utama yaitu: metode partisi, metode hierarkis, metode berbasis densitas, metode berbasis grid, dan pengelompokkan berbasis model. Menurut (Rokach & Maimon, 2005) terdapat beberapa macam metode *clustering* sebagai berikut:

a. Metode Hierarki

   Metode hierarki membentuk *cluster* dengan membagi *instance* secara rekursif dengan cara dekomposisi hierarki. Metode ini terdapat 2 jenis pendekatan yaitu:

   1. *Agglomerative*: setiap objek mewakili sebuah *cluster* sendiri, kemudian *cluster-cluster* tersebut secara urut digabung hingga membentuk *cluster* yang membesar.

2. *Divisive*: semua objek yang sebelumnya merupakan bagian dari satu dipisahkan menjadi *sub-cluster* yang terpisah, dan proses ini diulang sampai struktur *cluster* yang diinginkan tercapai.

b. Metode Partisi

Metode partisi merelokasi *instance* dengan memindahkannya dari satu *cluster* ke *cluster* lain, mulai dari partisi pertama. Metode ini dilakukan dengan menentukan jumlah klaster di awal oleh pengguna, kemudian proses enumerasi dari semua partisi diperlukan untuk mengoptimalkan pengelompokan berbasis partisi.

c. Metode berbasis Kepadatan

Metode berbasis kepadatan dipisahkan dari objek kepadatan rendah oleh daerah yang berdekatan, sering disebut noise (Kriegel, Kröger, Sander, & Zimek, 2011). Tujuan dari metode ini untuk menemukan *cluster* dan distribusi parameter. Metode ini didesain untuk menemukan *cluster* yang bentuknya tidak beraturan dan terdapat *noise*, kemudian dapat digunakan untuk *database* spasial.

d. Metode pengelompokkan berbasis model

Metode ini mencoba untuk mengoptimalkan diantara data yang diberikan dan beberapa model matematika. Selain itu, juga menemukan deskripsi karakteristik setiap kelompok yang mewakili suatu kelas dan metode yang paling sering digunakan yaitu *decision trees* dan *neural networks.*

e. Metode berbasis grid

Metode ini mempartisi ruang menjadi jumlah sel terbatas yang membentuk struktur grid untuk semua pengelompokkan yang dilakukan dan memiliki waktu pemrosesan yang cepat.

f. Metode komputasi lunak

Prinsip dari metode ini untuk mengeksploitasi toleransi untuk ketidakpastian, ketidakakuratan, penalaran, perkiraan, dan kebenaran parsial untuk mencapai kemiripan dengan pengambilan keputusan seperti manusia (Ceryan, 2016). Pada pendekatan *soft clustering* algoritma *fuzzy c-means* adalah yang paling populer dan merupakan pengembangan dari algoritma *k-means*.

2.4 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) adalah algoritma pengelompokan (*clustering*) berbasis kepadatan pertama yang diusulkan oleh (Ester,

Kriegel, & Xu, 1996) yang dirancang untuk menemukan *cluster* data dalam bentuk yang berubah-ubah dengan adanya *noise* dalam basis data dimensi tinggi spasial dan non-spasial. DBSCAN merupakan salah satu contoh teknik *clustering* berdasarkan kepadatan atau densitas yang terkoneksi (Devi, Putra, & Sukarsa, 2015). Jumlah data dalam radius MinPts (Jumlah minimum data dalam radius), data termasuk dalam kategori kepadatan yang diinginkan, dan jumlah data dalam radius hanya mencakup data inti tersebut adalah konsep kepadatan yang dimaksud oleh DBSCAN, kemudian terdiri dari tiga macam istilah yaitu *core* (inti), *border* (batas), *noise* (outlier) setiap data (D. Safitri, Wuryandari, & Rahmawati, 2017). DBSCAN dapat menentukan informasi apa yang harus dianggap sebagai *noise* atau outlier (Bäcklund & Neijman, 2011). Metode clustering DBSCAN mengidentifikasi tetangga (*neighbour*) dari titik spasial yang diberikan dan mengelompokkannya jika mereka dekat satu sama lain dalam jarak tertentu dan memenuhi beberapa persyaratan pengelompokan (Amiruzzaman, Rahman, Islam, & Nor, 2021).

Keuntungan dari algoritma DBSCAN diantaranya (Dang, 2015):

a. Dapat menemukan pengelompokkan yang berbentuk tidak beraturan.
b. Handal dalam mendeteksi *outlier.*
c. Dapat menemukan *cluster* yang dikelilingi oleh *cluster* yang berbeda.
d. Hanya dua titik yang diperlukan dan tidak sensitif dari urutan titik dalam *database.*

DBSCAN dapat menentukan jumlah *cluster* sendiri, sehingga tidak diperlukan bagi kita untuk menentukan jumlah *cluster*, tetapi membutuhkan dua parameter masukan lain yaitu (I Made Suwija Putra, 2018):

a. *Epsilon* yaitu nilai untuk jarak maksimal yang menjadi batas daerah *neighborhood* dari titik (*Epsilon-neighborhood*).
b. *MinPts* yaitu jumlah minimum titik dalam radius Epsilon.

Komputasi dari algoritma DBSCAN sebagai berikut: (Devi et al., 2015 dikuti dalam Fahamsyah, 2020):

1. Tentukan parameter *MinPts* dan *Epsilon* yang akan digunakan.
2. Pilih data awal atau c secara acak.
3. Hitung jarak antara data c terhadap semua titik yang *density reachable* menggunakan *Euclidean distance*. Namun, dalam penelitian ini menggunakan *Haversine distance.* Hal ini dikarenakan, haversine distance lebih cocok digunakan pada basis data spasial karena merupakan pengukuran jarak antara permukaan bumi.

4. Jika titik yang memenuhi epsilon lebih besar dari jumlah minimum(MinPts) maka titik c sebagai titik inti dan terbentuk *cluster*.
5. Jika c adalah *border point*, dan c tidak memiliki titik yang *density reachable*, maka proses akan dilanjutkan ke titik lain.
6. Ulangi langkah 3-4 hingga semua titik diproses.

Menurut (Bäcklund, Hedblom, & Nejiman, 2011) proses komputasi DBSCAN terdiri dari 6 definisi dan 2 lemma, sebagai berikut:

a. Definisi 1: *Epsilon-neighborhood* dari satu titik

$$N_{eps}(A) = \{A \in D | dist(A,B\} < Eps\} \quad (2.1)$$

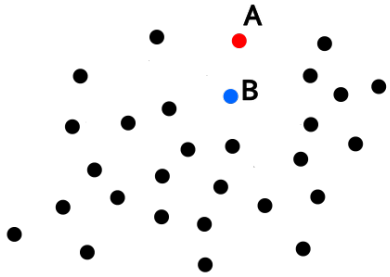
Suatu titik yang menjadi anggota suatu *cluster* memiliki paling sedikit satu titik lain yang lebih dekat dengan titik tersebut daripada jarak nilai *Eps*.

b. Definisi 2: *Directly Density-Reachable*

Terdapat dua titik dalam sebuah cluster yaitu *core points* dan *border points* bisa dilihat pada gambar 2.1.

A: *border point*

B: *core point*



Gambar 2.1 *Border dan core point.*

Eps-neighborhood dari core point cenderung mempunyai titik yang lebih banyak dari border point. Apabila border point memiliki Eps-neighbourhood dari suatu titik point B, maka titik tersebut menjadi bagian dari suatu cluster.

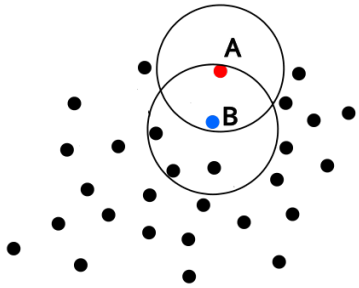
1. $A \in N_{eps(B)}$ (2.2)

   Titik B harus memiliki jumlah titik minimum dalam *eps-neighborhood* agar menjadi *core point*.

2. $\left|N_{eps}(B)\right| \geq MinPts$ (kondisi *core point*) (2.3)

A directly density- reachable dari B

B bukan directly density-reachable dari A



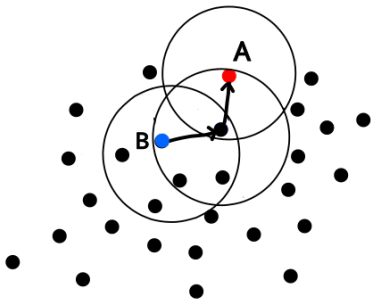
Gambar 2.2 Titik A merupakan *directly density-reachable* dari titik B dan tidak sebaliknya.

c. Definisi 3: *Density-reachable*

Suatu titik A density reachable dari titik B berdasarkan Eps dan MinPts jika terdapat rangkaian titik A_1,A_(2….) A_n, dimana A_1=B, A_n=A dimana A_(i+1) density reachable secara langsung dari A_1. Namun, harus ada core point B dimana keduanya dapat menjadi density reachable.

A *density-reachable* dari B

B bukan *density-reachable* dari A



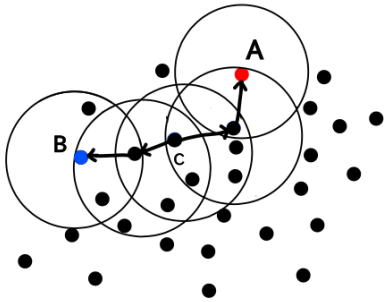
Gambar 2.3 Titik A *density-reachable* dari titik B dan tidak sebaliknya.

d. Definisi 4. *Density-Connected*

Terdapat situasi ketika dua *border point* akan menjadi bagian dari *cluster* yang sama, namun kedua *border point* tidak berbagi *core point* tertentu. Titik-titik tersebut tidak menjadi *density reachable* dari masing-masing titik lain, namun harus ada *core point* B dimana keduanya *density-reachable*. Pada gambar 2.4 ditunjukkan bagaimana *density connectivity* bekerja. Titik A merupakan *density connected* dengan titik B berdasarkan *Eps* dan *MinPts* jika terdapat titik c, sehingga titik A dan B *density reachable* dari c berdasarkan *Eps* dan *MinPts*

A dan B *density connected* satu sama lain oleh c

selain titik didalam lingkaran dianggap *noise* (outlier)



Gambar 2.4 *Density Connectivity.*

e. Definisi 5. Klaster

Jika titik A anggota dari cluster P dan titik B adalah density reachable dari titik A dengan jarak dan jumlah kerapatan titik minimum, maka B adalah anggota cluster P. Dua titik termasuk dalam cluster yang sama dengan P, artinya A density connected dengan B melalui jarak dan jumlah titik dalam jarak yang berikan.

f. Definisi 6: *Noise*

Noise adalah sekumpulan titik dalam *database* yang bukan milik *cluster* manapun.

g. Lemma 1

Sebuah *cluster* dapat dibentuk dari salah satu titik inti (*core point*) dan selalu memiliki bentuk yang sama.

h. Lemma 2

Titik A menjadi titik inti dalam *cluster* P dengan *eps* dan *MinPts.* A sama dengan himpunan O jika himpunan O adalah *densitiy reachable* dari titik p berdasarkan Eps dan MinPts yang sama.

2.5 Haversine Distance

Haversine formula merupakan pengukuran jarak pada bola katakanlah bumi, yang digunakan untuk mengukur jarak antara dua titik permukaan bumi dengan menggunakan latitude dan longitude (Maulana, Solichin, & Syafrullah, 2018). Haversine merupakan metode yang populer dan sering digunakan dalam menganalisis jalur dan bidang, serta mengembangkan aplikasi GIS (*Geographic Information System*) (Upadhyay, 2015). Dengan kata lain, formula ini cocok digunakan untuk data spasial atau berbasis *geo-location*. Pada penelitian (Selvaraj & Sabarish, 2021) menyatakan hasil dari *haversine distance* mengungguli

dari metode lainnya serta jarak menjadi lebih optimal dan mendapatkan hasil akurasi yang lebih baik. *Haversine distance* dapat dirumuskan pada persamaan 2.4:

$$D = 2R.\sin^{-1}\sqrt{sin^2(\frac{y_2 - y_1}{2}) + \cos(y_1).\cos(y_2).sin^2(\frac{x_2 - x_1}{2})} \quad (2.4)$$

Disini diketahui D yaitu jarak, kemudian R merepresentasikan jari-jari bumi yaitu sekitar 6371 km, x1, x2 adalah longitudey1, y2 adalah latitude

2.6 Silhouette Coefficient

Silhouette coefficient digunakan untuk menunjukkan seberapa baik objek dalam suatu *cluster* (Anggara, Sujiani, & Nasution, 2016) dan membantu menentukan jumlah *cluster* yang optimal (Shahapure & Nicholas, 2020). Nilai rata-rata koefisien yang tinggi berarti menunjukkan pengelompokan yang bagus (Shahapure & Nicholas, 2020). Pada persamaan 2.5 merupakan rumus untuk nilai *silhouette coefficient*.

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)} \quad (2.5)$$

Skor nilai *silhouette* mendekati +1 menunjukkan titik data termasuk dalam klaster yang tepat, skor nilai silhouette mendekati 0 menunjukkan bahwa titik data mungkin merupakan bagian dari klaster lain, skor nilai silhouette yang mendekati -1 menunjukkan bahwa titik data berada di klaster yang salah (Shahapure & Nicholas, 2020).

2.7 Streamlit

Streamlit adalah *library* Python yang bersifat *open source* untuk membangun aplikasi website dengan mudah dan interaktif di bidang *data science* dan *machine learning* ("Streamlit Docs," n.d.). Pengguna streamlit tidak perlu memiliki kemampuan *front-end* yang ahli untuk menggunakannya. Streamlit adalah *framework* yang bertujuan untuk implementasi model serta visualisasi pada bahasa Python dengan mudah dan cepat, tetapi memiliki tampilan yang *user-friendly*, dan cukup bagus (Widi Hastomo, Nur Aini, Adhitio Satyo Bayangkari Karno, & L.M. Rasdi Rere, 2022).

2.8 Black Box Testing

Black box testing merupakan pengujian yang berkonsentrasi untuk setiap spesifikasi fungsionalitas pada software (N. Safitri & Pramudita, 2018). Penguji dapat menentukan serangkaian kondisi input dan menjalankan pengujian sesuai dengan spesifikasi fungsional (Hidayat & Putri, 2020). Penguji tidak perlu memiliki pengetahuan mengenai bahasa pemrograman maupun dalam implementasinya serta dilakukan dari sudut pandang pengguna (Nidhra, 2012). Menurut (Chren, n.d.) terdapat beberapa teknik dalam *black box testing* sebagai berikut:

a. Equivalence Partitioning

Dalam pengujian *equivalence partitioning* membagi data input menjadi partisi data test case, dan dibentuk dari input perilaku sistem ditentukan. Setiap kelas mewakili kumpulan status valid atau tidak valid untuk kondisi input.

b. Boundary Value Analysis

Boundary value analysis merupakan teknik pengujian yang berfokus pada pengujian batasan range nilai valid yang valid di sekitar batas minimum dan maksimum pada kondisi input maupun output kelas-kelas yang diidentifikasi dalam spesifikasi.

c. Graph-Based

Pada teknik ini merupakan pengujian yang membuktikan bahwa objek memiliki relasi antara satu sama lain dengan menggunakan grafik.

d. State Transition Table

Teknik ini biasanya disajikan dalam bentuk *state transition diagram*. Pengujian ini berfungsi untuk memeriksa validitas transisi antar status dan transisi-transisi yang tidak termasuk.

Dalam penelitian ini, digunakan teknik *equivalence partitioning* dikarenakan cukup mudah dipahami, dan diimplementasikan. Selain itu, dapat meminimalisir jumlah kasus uji yang harus dijalankan, tanpa mengabaikan keakuratan pengujian, karena pengujian satu nilai perwakilan dari setiap partisi setara, bukan setiap nilai individu.

2.9 Penelitian Terkait

Terdapat berbagai cara yang telah dilakukan pada penelitian sebelumnya untuk terus meningkatkan pelayanan taksi dengan fokus tujuan yang berbeda, seperti dari segi lokasi, waktu, prediksi, dan sebagainya. Pada penelitian ini menganalisis hasil dari *cluster* terhadap wilayah atau area yang akan terbentuk dengan menggunakan metode DBSCAN dan data yang

digunakan berasal dari perjalanan taksi New York City tahun 2016. Selain itu, juga membahas dan menganalisis mengenai profil data perjalanan taksi secara statistik. Hal ini didukung dengan penelitian mengenai data spasial selain taksi yang menerapkan algoritma DBSCAN. Berikut pada tabel 2.1 yang menunjukkan penelitian terkait

Tabel 2.1 Penelitian terkait

Penelitian	Model/Algoritma	Dataset	Tujuan
Penelitian yang menggunakan DBSCAN selain taksi			
(Lei, 2019)	DBSCAN	AIS (*Automatic Information System)* kapal maritim	Mendeteksi area spot yang sering dikunjungi oleh banyak kapal dan mengelompokkan area tempat tinggal kapal selama pelayaran
(Anwar, Hadikurniawati, Winarno, & Supriyanto, 2019)	DBSCAN	Riau, Sumatra, Indonesia	Mengidentifikasi area dengan risiko kebakaran berdasarkan data historis titik hotspot kebakaran hutan
Penelitian yang berkaitan dengan taksi menggunakan metode selain DBSCAN			
(Xu, Rahmatizadeh, Bölöni, & Turgut, 2018)	RNN LSTM-MDN	Taksi New York City	Memprediksi permintaan taksi di setiap area kota berdasarkan permintaan terkini yang dapat membantu untuk meminimalkan waktu tunggu penumpang dan pengemudi serta mengatur armada taksi
(Qu et al., 2019)	TSLM (*Taxi Stand Location Model*)	Taxi di kota China	Mengusulkan strategi lokasi untuk mengatasi masalah lokasi pangkalan taksi diperkotaan yang dapat memberikan referensi ilmiah bagi departemen kota dalam keputusan lokasi pangkalan taksi
(Rodrigues, Markou, & Pereira, 2019)	Time series DL-LSTM, DL-FC (*Fully Connected*)	Taksi New York City	Peramalan permintaan taksi di area acara
(Mangopo & Suprayitno, 2020)	Tidak ada	Taksi Blue Bird	Variasi volume perjalanan dan karakterisik panjang perjalanan taksi bertujuan untuk mengetahui jumlah antaran penumpang dalam sehari dan jam bekerja pengemudi taksi yang efektif

(Rossi, Barlacchi, Bianchini, & Lepri, 2020)	RNN	Taksi di Porto, Manhattan, San Francisco	Memprediksi tujuan lokasi taksi terutama ketika permintaan tinggi. Jika dapat mengetahui dahulu dimana pengemudi mengakhiri perjalanan, itu dapat mengalokasikan sumber daya dengan lebih baik dengan mengidentifikasi taksi mana yang dipanggil.
(Ramadhan, Nur, & Adhinata, 2022)	*Time series* LSTM-RNN	Taksi New York	Mengetahui estimasi durasi perjalanan taksi dalam moda taksi tradisional agar dapat mengatur waktu perjalanan dan menghemat biaya.
Penelitian yang berkaitan dengan taksi menggunakan metode DBSCAN			
(D. Zhou, Hong, & Xia, 2018)	DBSCAN	Taxi di China (kota Kunshan)	Mengidentifikasi titik titik penjemputan dan pengantaran taksi untuk manajemen operasi taksi dan analisis pola perjalanan
(Huang et al., 2021)	DBSCAN+	Taksi kota Huai'an	Pengenalan dan visualisasi hot spot penumpang taksi yang dapat memberikan keputusan untuk perencanaan kota lebih lanjut dan efisiensi lalu lintas

BAB III
METODOLOGI PENELITIAN

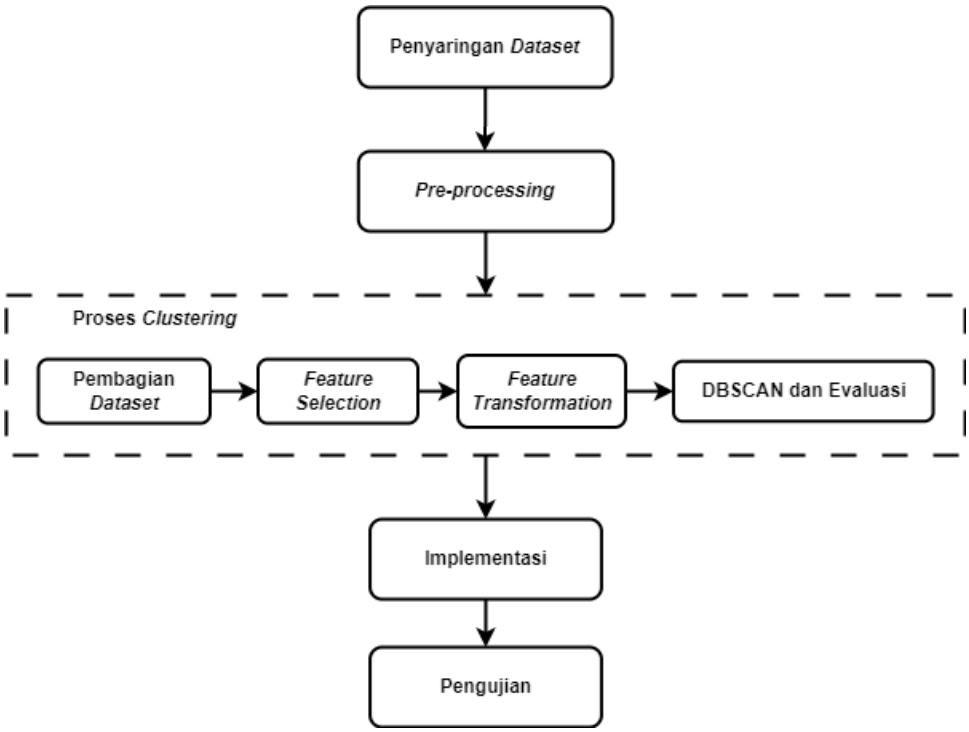
3.1 Sumber Data

Data yang digunakan dalam penelitian ini adalah data sekunder. Pengumpulan data yang dikenal dengan data sekunder adalah data yang telah diperoleh secara tidak langsung atau setelah melalui pencarian yang mendalam, seperti melalui internet, buku, statistik, literatur dan lain-lain (Tanujaya, 2017). Pada penelitian ini, data sekunder yang digunakan berasal dari situs Kaggle yaitu dataset publik perjalanan taksi kota New York.

3.2 Studi Literatur

Strategi menggunakan studi literatur melalui menganalisis sumber tertulis untuk mengungkap berbagai teori yang relevan dengan masalah penelitian dan menjadi landasan untuk membahas temuan penelitian (Handriani, 2019). Kemudian, informasi yang didapat dimanfaatkan sebagai dasar untuk mendukung argumen yang dibuat, dan sebagai landasan pengetahuan untuk penelitian yang dilakukan.

3.3 Metode Penelitian



Gambar 3.1 Tahapan penelitian

3.3.1 Penyaringan *Dataset*

Data yang digunakan dalam penelitian ini tersedia pada situs web Kaggle yaitu *dataset* publik perjalanan taksi kota New York. Dataset yang diambil selama tiga bulan sejak tanggal 1 Januari 2016 hingga 31 Maret 2016 yang hanya berfokus pada wilayah Queens, New York City. Beberapa variabel serta deskripsi dari *dataset* mentah ditunjukkan pada tabel 3.1

Tabel 3.1 Definisi dari variabel

Nama Variabel	Definisi
Id	Kode unik dari setiap perjalanan
Vendor id	Kode yang menunjukkan provider yang terkait dengan record perjalanan
Pickup datetime	Tanggal dan waktu saat meteran diaktifkan
Dropoff datetime	Tanggal dan waktu pada saat meter di nonaktifkan
Passenger count	Jumlah penumpang didalam kendaraan
Pickup longitude	Garis bujur tempat meter ketika diaktifkan
Pickup latitude	Garis lintang tempat meter ketika diaktifkan
Dropoff longitude	Garis bujur tempat meter ketika dinonaktifkan
Dropoff latitude	Garis lintang tempat meter dinonaktifkan
Store and fwd flag	Penanda apakah catatan perjalanan disimpan dalam memori kendaraan sebelum dikirimkan ke vendor atau tidak
Trip duration	Durasi perjalanan dalam hitungan detik.

3.3.2 *Pre-processing*

Pre-processing diekstraksi dari data mentah menjadi data yang bersih dan rapi dan bertujuan untuk meningkatkan kualitas data mentah (Fan et al., 2021). Teknik *preprocessing* dilakukan dengan cara berikut:

a. Feature Creation

Proses *feature creation* yaitu pembuatan variabel dengan cara mengekstrak variabel yang ada menjadi fitur baru, mengkonversi dari variabel numerik menjadi variabel kategori atau yang disebut diskretisasi, dan menghasilkan fitur baru dari hasil pembagian antara dua variabel.

b. Data Cleaning

Data cleaning digunakan untuk proses pembersihan data dengan mengubah atau membuang dari data yang rusak, tidak akurat, inkonsisten serta tidak relevan (Agarwal, 2015)

3.3.3 Proses *Clustering*

Sebelum masuk ke tahap pemodelan, terdapat beberapa langkah proses yang perlu dilakukan diantaranya sebagai berikut:

a. Pembagian Dataset

Penelitian ini membagi dataset menjadi tiga kelompok berdasarkan bulan, yaitu januari, februari, dan maret. Ini dilakukan dengan tujuan untuk mengetahui apakah ada perbedaan atau kesamaan pola kebutuhan taksi di bulan yang berbeda.

b. Feature Selection

Saat proses pemodelan, hanya beberapa variabel yang relevan untuk dilakukan pemodelan. Jika memasukkan variabel yang tidak relevan, maka hasil pemodelan bisa berdampak buruk.

c. Feature Transformation

Data transformation digunakan untuk mengkonversi data menjadi format yang dapat dimengerti (Agarwal, 2015)

d. DBSCAN dan Evaluasi

Pada proses *clustering* metode yang digunakan pada penelitian ini adalah *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN). Proses ini bertujuan untuk mengelompokkan area-area penjemputan taksi. Sebelum dilakukan proses *clustering*, diperlukan penentuan parameter epsilon dan parameter MinPts. Dalam menentukan parameter, dilakukan beberapa percobaan untuk memilih parameter terbaik yang diukur berdasarkan metrik evaluasi. Evaluasi dilakukan bersamaan dengan proses *clustering* untuk menentukan parameter terbaik untuk digunakan dalam pemodelan hingga memperoleh hasil skor evaluasi yang bagus. Evaluasi pada penelitian ini menggunakan metrik *silhouette coefficient* untuk mengetahui akurasi dari hasil pemodelan *clustering*

3.3.4 Implementasi

Pada tahap implementasi merupakan tahap pengembangan aplikasi website berdasarkan pemodelan. Implementasi ini menggunakan *framework* Streamlit berbasis Python yang berguna membangun website secara interaktif dalam pemodelan *machine learning*.

3.3.5 Pengujian

Setelah aplikasi selesai dibuat, dilakukan tahap pengujian yang berguna untuk menentukan kualitas apakah kinerja sistem bekerja seperti yang diharapkan pengguna atau tidak (Hidayat

& Putri, 2020). Metode pengujian pada aplikasi ini menggunakan *black box testing* dengan teknik *equivalence partitioning*. Teknik ini dilakukan menggunakan *test case* untuk melakukan uji coba berdasarkan kelompok fungsinya dan memastikan keluaran sesuai dari yang diharapkan.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Penyaringan *Dataset*

Data yang digunakan berasal dari situs web Kaggle yaitu dataset publik New York City. Pada gambar 4.1 merupakan dataset mentah yang belum di filter. *Dataset* tersebut terdiri dari 1458644 baris dan 11 kolom atau variabel.

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_fwd_flag	trip_duration
2	id2875421	2	3/14/2016 17:24	3/14/2016 17:32	1	-73.98215485	40.76793671	-73.96463013	40.76560211	N	455
3	id2377394	1	6/12/2016 0:43	6/12/2016 0:54	1	-73.98041534	40.73856354	-73.9994812	40.73115158	N	663
4	id3858529	2	1/19/2016 11:35	1/19/2016 12:10	1	-73.97902679	40.7639389	-74.00533295	40.71008682	N	2124
5	id3504673	2	4/6/2016 19:32	4/6/2016 19:39	1	-74.01004028	40.7199707	-74.01226807	40.70671844	N	429
6	id2181028	2	3/26/2016 13:30	3/26/2016 13:38	1	-73.97305298	40.79320908	-73.97292328	40.78252029	N	435
7	id0801584	2	1/30/2016 22:01	1/30/2016 22:09	6	-73.98285675	40.74219513	-73.99208069	40.74918365	N	443
8	id1813257	1	6/17/2016 22:34	6/17/2016 22:40	4	-73.96901703	40.7578392	-73.95740509	40.76589584	N	341
9	id1324603	2	5/21/2016 7:54	5/21/2016 8:20	1	-73.96927643	40.79777908	-73.92247009	40.76055908	N	1551
10	id1301050	1	5/27/2016 23:12	5/27/2016 23:16	1	-73.9994812	40.73839951	-73.98578644	40.73281479	N	255
11	id0012891	2	3/10/2016 21:45	3/10/2016 22:05	1	-73.98104858	40.74433899	-73.97299957	40.78998947	N	1225
12	id1436371	2	5/10/2016 22:08	5/10/2016 22:29	1	-73.98265076	40.76383972	-74.00222778	40.73299026	N	1274
13	id1299289	2	5/15/2016 11:16	5/15/2016 11:34	4	-73.99153137	40.74943924	-73.95654297	40.77062988	N	1128
14	id1187965	2	2/19/2016 9:52	2/19/2016 10:11	2	-73.96298218	40.75667953	-73.98440552	40.7607193	N	1114
15	id0799785	2	6/1/2016 20:58	6/1/2016 21:02	1	-73.95630646	40.76794052	-73.96611023	40.76300049	N	260
16	id2900608	2	5/27/2016 0:43	5/27/2016 1:07	1	-73.99219513	40.72722626	-73.97465515	40.78306961	N	1414
17	id3319787	1	5/16/2016 15:29	5/16/2016 15:32	1	-73.955513	40.76859283	-73.94876099	40.77154541	N	211
18	id3379579	2	4/11/2016 17:29	4/11/2016 18:08	1	-73.99116516	40.75556183	-73.99929047	40.72535324	N	2316
19	id1154431	1	4/14/2016 8:48	4/14/2016 9:00	1	-73.99425507	40.74580383	-73.99965668	40.7233429	N	731
20	id3552682	1	6/27/2016 9:55	6/27/2016 10:17	1	-74.00398254	40.7130127	-73.97919464	40.74992371	N	1317
21	id3390316	2	6/5/2016 13:47	6/5/2016 13:51	1	-73.98388672	40.73819733	-73.99120331	40.72787094	N	251
22	id2070428	1	2/28/2016 2:23	2/28/2016 2:31	1	-73.98036957	40.7424202	-73.96285248	40.76063538	N	486
23	id0809232	2	4/1/2016 12:12	4/1/2016 12:23	1	-73.97953796	40.75336075	-73.96399689	40.76345825	N	652
24	id2352683	1	4/9/2016 3:34	4/9/2016 3:41	1	-73.99586487	40.75881195	-73.99332428	40.74032211	N	423
25	id1603037	1	6/25/2016 10:36	6/25/2016 10:55	1	-73.99355316	40.74717331	-74.00614166	40.70438385	N	1163

Gambar 4.1 *Dataset* mentah

Penelitian ini hanya berfokus pada titik penjemputan di wilayah Queens, New York City dari Januari - Maret 2016. Oleh karena itu, dilakukan penyaringan data sebanyak dua kali, yang pertama dilakukan penyaringan dari variabel “pickup_datetime” untuk mendapatkan data hanya tiga bulan pertama yaitu januari februari, dan maret. *Code* untuk penyaringan data terdapat pada gambar 4.2, dan hasilnya ditunjukkan pada gambar 4.3.

```
df_filter = df.loc[df['pickup_datetime'] < '2016-04-01']
```

Gambar 4.2 *Code* penyaringan data berdasarkan bulan

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_fwd_flag	trip_duration
0	id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.982155	40.767937	-73.964630	40.765602	N	455
1	id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.979027	40.763939	-74.005333	40.710087	N	2124
2	id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.973053	40.793209	-73.972923	40.782520	N	435
3	id0801584	2	2016-01-30 22:01:40	2016-01-30 22:09:03	6	-73.982857	40.742195	-73.992081	40.749184	N	443
4	id0012891	2	2016-03-10 21:45:01	2016-03-10 22:05:26	1	-73.981049	40.744339	-73.973000	40.789989	N	1225
5	id1187965	2	2016-02-19 09:52:46	2016-02-19 10:11:20	2	-73.962982	40.756680	-73.984406	40.760719	N	1114

Gambar 4.3 Hasil dari penyaringan data

Setelah melakukan filter berdasarkan bulan, kemudian dilakukan filter berdasarkan wilayah. Sebelum itu, untuk membuat variabel “suburbs” dilakukan ekstraksi dari variabel “pickup_latitude” dan “pickup_longitude”. Gambar 4.4 merupakan code untuk membuat variabel wilayah.

```
# import module
from geopy.geocoders import Nominatim
#Inisialisasi Nominatim API
geolocator = Nominatim(user_agent="Geocode", timeout=None)
#Membuat data frame baru untuk tabel coordinat
coordinates = pd.DataFrame({'Coordinates':
                            df_jan['pickup_latitude'].astype(str) +","+
                             df_jan['pickup_longitude'].astype(str)})
#Membuat data frame baru untuk locations
locations = pd.DataFrame(columns = ['suburbs_pickup'])
#Looping untuk mengambil informasi alamat spesifik 'suburbs' dari coordinat
dan kemudian disimpan dalam data frame location
for coordinat in coordinates[:]['Coordinates']:
    location = geolocator.reverse(coordinat)
    if location is None:
        locations.loc[len(locations.index)] = ["none"]
    else:
        address = location.raw['address']
        suburbs = address.get('suburb', '')
        locations.loc[len(locations.index)] = [suburbs]
```

Gambar 4.4 Contoh *code* untuk membuat variabel suburbs

Setelah terbentuk variabel “suburbs”, selanjutnya dilakukan penyaringan untuk memilih wilayah Queens. Pada gambar 4.5 ditunjukkan hasil dari penyaringan dari variable “suburbs.

Kemudian setelah proses penyaringan dataset selesai, hasil yang diperoleh sebanyak 724196 baris dan 12 kolom.

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_fwd_flag	trip_duration	suburbs_pickup
0	id2029339	2	2016-01-22 14:13:46	2016-01-22 15:15:21	1	-73.873360	40.774109	-73.958115	40.775558	N	3695	Queens
1	id3579210	2	2016-01-25 21:05:42	2016-01-25 22:01:52	1	-73.782089	40.644650	-73.974243	40.789761	N	3370	Queens
2	id1324382	1	2016-01-07 18:12:52	2016-01-07 18:55:57	1	-73.782349	40.644802	-73.954651	40.789146	N	2585	Queens
3	id0896335	1	2016-01-25 11:07:10	2016-01-25 12:04:58	1	-73.776794	40.645473	-73.958183	40.673565	N	3468	Queens
4	id3929561	1	2016-01-19 16:01:58	2016-01-19 16:48:18	1	-73.790359	40.644024	-73.939026	40.716095	N	2780	Queens

Gambar 4.5 Hasil penyaringan data suburbs

4.2 Pre-processing

4.2.1 *Feature Creation*

Feature creation yaitu pembuatan variabel baru dengan cara mengekstrak data dari variabel yang sudah ada, dan berguna dalam proses pemodelan ataupun analisis. Berikut beberapa *feature creation* yang dilakukan dalam penelitian ini:

a. Pembuatan variabel "pickup_hour", "pickup_day", "pickup_dayname", dan "pickup_monthname"

   Proses pembuatan variabel ini, diekstraksi dari variabel "pickup_datetime" yang sudah ada pada dataset. Pada variable "pickup_hour" menghasilkan waktu pickup, kemudian "pickup_day" merupakan hari dalam urutan angka, "pickup_dayname" yaitu nama hari saat waktu penjemputan, dan "pickup_monthname" yaitu nama bulan. Sebelum membuat variabel tersebut, diperlukan untuk mengubah tipe variabel 'pickup_datetime' yang sebelumnya object menjadi tipe datetime, hal ini diperlukan untuk mengola data yang berkaitan dengan waktu sehingga bisa digunakan dalam pembuatan variabel. Gambar 4.6 adalah code untuk mengubah tipe data menjadi datetime dan Gambar 4.7 merupakan *code* untuk pembuatan variabel tersebut.

```
df_queens['pickup_datetime'] =pd.to_datetime(
                                        df_queens['pickup_datetime'])
```

Gambar 4.6 *Code* untuk mengubah tipe data menjadi datetime

```
df_queens["pickup_hour"] = df_queens['pickup_datetime'].dt.hour
df_queens["pickup_day"] = df_queens['pickup_datetime'].dt.weekday
df_queens["pickup_dayname"] = df_queens['pickup_datetime'].
                                                    dt.day_name()
df_queens["pickup_monthname"] = df_queens['pickup_datetime'].
                                                  dt.month_name()
```

Gambar 4.7 *Code* pembuatan variabel “pickup_hour”, “pickup_day”, “pickup_dayname”, dan “pickup_monthname”.

b. Pembuatan variabel “pickup_groupofday

Proses pembuatan variabel ini dengan cara mengekstrak dari variabel “pickup_day” yang telah dibuat sebelumnya. Variable ini untuk mengetahui jenis hari yaitu hari kerja dan akhir pekan. Pada gambar 4.8 merupakan *code* untuk membuat variabel “pickup_groupofday”.

```
#Membuat fungsi untuk variabel pickup_groupDay
def groupDay(x):
  if x in range(0,5):
    return 'Weekday'
  else:
    return 'Weekend'

#Mengaplikasikan fungsi yang dibuat untuk membuat variabel
df_queens['pickup_groupofday'] =
                              df_queens['pickup_day'].apply(groupDay)
```

Gambar 4.8 *Code* untuk membuat variabel “pickup_groupofday”

c. Pembuatan variabel “pickup_timeofday”

Proses pembuatan variabel ini dengan cara mengekstrak dari variabel “pickup_hour” yang telah dibuat sebelumnya. “pickup_timeofday” merupakan bagian dalam hari yang dikelompokkan menjad 5 bagian. Pada gambar 4.9 merupakan *code* untuk membuat variabel “pickup_groupofday”

```
#Membuat fungsi untuk variabel pickup_timeofday
def time(y):
  if y in range(4,6):
    return 'Dawn'
  elif y in range(6,12):
    return 'Morning'
  elif y in range(12,17):
    return 'Afternoon'
  elif y in range(17,23):
    return 'Evening'
  else:
    return 'Late night'

#Mengaplikasikan fungsi yang dibuat untuk membuat variabel
df_queens['pickup_timeofday'] = df_queens['pickup_hour'].apply(time)
```

Gambar 4.9 *Code* untuk membuat variabel “pickup_timeofday”

d. Pembuatan variabel “distance”

Proses pembuatan variabel ini dengan cara mengekstrak dari variabel “pickup_longitude”, “pickup_latitude”, “dropoff_longitude”, “dropoff_latitude” yang sudah ada pada dataset. Variable “distance” merupakan jarak yang ditempuh dalam perjalanan. Pada gambar 4.10 merupakan *code* untuk membuat variabel “pickup_groupofday”.

```
from math import radians, cos, sin, asin, sqrt

#Membuat fungsi untuk distance haversine
def haversine(long_1, lat_1, long_2, lat_2):
    #Konversi derajat decimal menjadi radians
    long_1, lat_1, long_2, lat_2 = map(radians,
                                       [long_1, lat_1, long_2, lat_2])

    #Rumus haversine
    dh_long = long_2 - long_1
    dh_lat = lat_2 - lat_1
    a = sin(dh_lat/2)**2 + cos(lat_1) * cos(lat_2) * sin(dh_long/2)**2
    c = 2 * asin(sqrt(a))
    r = 6371 # Radius of earth in kilometers. Use 3956 for miles.

    #Determines return value units.
    return c * r

#Mengaplikasikan fungsi yang dibuat untuk membuat variabel
df_queens['distance'] = df_queens.apply(lambda y: haversine(
     y['pickup_longitude'],y['pickup_latitude'],y['dropoff_longitude'],
     y['dropoff_latitude']), axis=1)
```

Gambar 4.10 *Code* untuk membuat variabel “distance”

e. Pembuatan variabel “average_speed

Proses pembuatan variabel “average_speed” dengan cara mengekstrak dari variabel “distance” yang telah dibuat sebelumnya dan “trip_duration_min”. variable ini merupakan kecepatan rata-rata yang ditempuh dalam perjalanan. Pada gambar 4.11 merupakan code untuk membuat variabel “average_speed”.

```
df_queens['average_speed'] = df_queens['distance']/
                                (df_queens['trip_duration']/3600)
```

Gambar 4.11 *Code* untuk membuat variabel “average_speed”

f. Pembuatan variabel “duration_min:

Proses pembuatan variabel “duration_min” dengan cara mengekstrak dari variabel “trip_duration” yang telah ada pada dataset. Pada gambar 4.12 merupakan code untuk membuat variabel “duration_min”. “Duration_min” merupakan durasi perjalanan dari titik penjemputan ke titik pengantaran dalam waktu menit.

```
df_queens['duration_min'] = np.round(df_queens["trip_duration"] / 60)
```

Gambar 4.12 *Code* untuk membuat variabel "duration_min"

Setelah dilakukan pembuatan variabel, dihasilkan sejumlah 9 variabel yang telah dibuat. Hasil dari feature creation dapat dilihat pada gambar 4.13

pickup_hour	pickup_day	pickup_dayname	pickup_monthname	pickup_groupofday	pickup_timeofday	distance	average_speed	duration_min
14	4	Friday	January	Weekday	Afternoon	7.138687	6.955149	62.0
21	0	Monday	January	Weekday	Evening	22.860888	24.421126	56.0
18	3	Thursday	January	Weekday	Evening	21.644570	30.143308	43.0
11	0	Monday	January	Weekday	Morning	15.616084	16.210468	58.0
16	1	Tuesday	January	Weekday	Afternoon	14.879018	19.267793	46.0

Gambar 4.13 Hasil dari *feature creation*

4.2.2 *Data Cleaning*

Data cleaning dilakukan untuk membuang variabel yang tidak relevan atau tidak digunakan untuk analisis dan pemodelan. *Data cleaning* dilakukan dengan menghapus variabel "id", "vendor_id", "dropoff_datetime", "passenger_count", "dropoff_longitude", "dropoff_latitude", "store_and_fwd_flag", "trip_duration" dan "suburbs". Dalam hal ini, pada dataset terdeteksi adanya *outlier* di beberapa variabel, tetapi pada kasus saat ini tidak menghapus *outlier* dikarenakan dasar teori algoritma DBSCAN yaitu handal dalam mendeteksi *outlier*.

```
df_clean = df_queens.drop(['id', 'vendor_id', 'dropoff_datetime',
           'passenger_count','dropoff_longitude','dropoff_latitude',
          'store_and_fwd_flag', 'trip_duration', 'suburbs_pickup'], axis=1)
```

Gambar 4.14 *Code* untuk menghapus variabel

Pada gambar 4.14 merupakan *code* dalam menghapus variabel yang tidak relevan. Pada dataset ini tidak ada *missing value* maupun data yang duplikat, kemudian untuk mengecek apakah data memiliki data yang duplikat dan cara mengatasinya yaitu dihapus dapat dilihat pada gambar 4.15, dan untuk mengecek terdapat *missing value* pada data dan mengatasi missing value dengan dihapus dapat dilihat pada gambar 4.16.

```
#code untuk mengetahui jumlah data yang duplikat
df_clean.duplicated().sum()
#code untuk menghapus jumlah data yang duplikat
df_clean.drop_duplicates(inplace=True)
```

Gambar 4.15 *Code* untuk mengatasi data yang duplikat

```
#code untuk mengetahui jumlah missing value tiap kolom
df_clean.isna().sum()
#code untuk menghapus missing value
df.dropna(axis=0, inplace=True)
```

Gambar 4.16 *Code* untuk mengatasi *missing value*

Dengan demikian, hasilnya ditunjukkan pada gambar 4.17, dan terdapat 12 variabel yang digunakan yaitu “pickup_datetime”, “pickup_longitude”, “pickup_latitude”, “pickup_hour”, “pickup_day”, “pickup_dayname”, “pickup_monthname”, “pickup_groupofday”, “pickup_timeofday”, “distance”, “average_speed”, dan “duration_min”. Dalam melakukan pemodelan *clustering* hanya dibutuhkan 2 variabel saja yaitu “pickup_latitude”, dan “pickup_longitude”. Selain dari 2 variabel tersebut dibutuhkan untuk proses analisis dari hasil *clustering*.

	pickup_datetime	pickup_longitude	pickup_latitude	pickup_hour	pickup_day	pickup_dayname	pickup_monthname	pickup_groupofday	pickup_timeofday	distance	average_speed	duration_min
0	2016-01-22 14:13:46	-73.873360	40.774109	14	4	Friday	January	Weekday	Afternoon	7.138687	6.955149	62.0
1	2016-01-25 21:05:42	-73.782089	40.644650	21	0	Monday	January	Weekday	Evening	22.860888	24.421126	56.0
2	2016-01-07 18:12:52	-73.782349	40.644802	18	3	Thursday	January	Weekday	Evening	21.644570	30.143308	43.0
3	2016-01-25 11:07:10	-73.776794	40.645473	11	0	Monday	January	Weekday	Morning	15.616084	16.210468	58.0
4	2016-01-19 16:01:58	-73.790359	40.644024	16	1	Tuesday	January	Weekday	Afternoon	14.879018	19.267793	46.0

Gambar 4.17 Hasil dari *data cleaning*

4.3 Proses *Clustering*

4.3.1 Pembagian Dataset

Pada penelitian ini dilakukan dengan membagi tiga dataset berdasarkan bulan yaitu, januari, februari dan maret. Hal Ini dilakukan dengan tujuan untuk melakukan proses *clustering* berdasarkan bulan yang berbeda-beda, untuk mengetahui apakah ada perbedaan pola yang terbentuk dari titik penjemputan. *Code* untuk membagi dataset ini bisa dilihat pada gambar 4.18.

```
df_jan = df_clean.loc[df_clean['pickup_datetime'] < '2016-02-01']

df_feb = df_clean.loc[(df_clean['pickup_datetime'] > '2016-02-01') &
                      (df_clean['pickup_datetime'] < '2016-03-01')]

df_mar = df_clean.loc[(df_clean['pickup_datetime'] > '2016-03-01') &
                       (df_clean['pickup_datetime'] < '2016-04-01')]
```

Gambar 4.18 *Code* untuk membagi *dataset* berdasarkan bulan

4.3.2 *Feature Selection*

Sebelum masuk dalam proses *clustering*, dilakukan *feature selection* terlebih dahulu. Tahap ini dilakukan untuk memilih variabel yang relevan untuk dilakukan proses *modelling*. Dalam hal ini, *modelling* dilakukan untuk mengetahui area titik penjemputan. Oleh karena itu, hanya dipilih 2 variabel yang berkaitan yaitu "pickup_longitude" dan "pickup_latitude". *Feature selection* ini dilakukan di setiap dataset yang telah dibagi berdasarkan bulan. Selain dari dua variabel tersebut akan digunakan untuk analisis hasil dari *clustering*. Pada gambar 4.19 merupakan contoh code untuk *feature selection*

```
df_pickup = jan_queens[['pickup_latitude', 'pickup_longitude']]
```

Gambar 4.19 Contoh *code* untuk *feature selection*

4.3.3 *Feature Transformation*

Data yang digunakan berjenis spasial atau geografi yaitu titik longitude dan latitude. Sesuai dari dokumentasi ("Sklearn.Metrics.Pairwise.Haversine_distances," n.d.) metrik *haversine distance* cocok untuk digunakan pada kasus ini karena memberikan perkiraan yang baik tentang jarak antara dua titik di permukaan bumi dan dalam satuan radian. Oleh karena itu, variabel yang telah dipilih dilakukan transformasi dari satuan derajat menjadi satuan radians. Terlihat pada gambar 4.20 *code* untuk mengubah variabel menjadi satuan radians.

```
df_pickup = np.radians(df_pickup)
```

Gambar 4.20 *Code* untuk mengubah variabel menjadi satuan radians

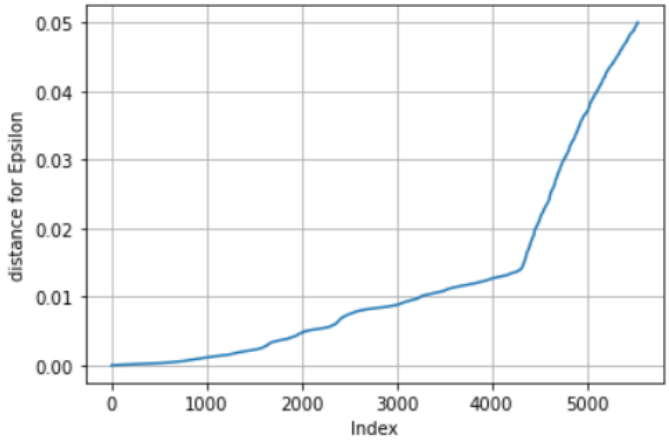
4.3.4 *Density-based Spatial Clustering of Applications with Noise* (DBSCAN) dan Evaluasi

A. Menentukan Parameter

Penelitian ini menggunakan DBSCAN dalam proses *modelling clustering*. Tahap pertama yang perlu dilakukan yaitu, menentukan parameter yang optimal untuk nilai Eps dan MinPts. Dalam hal ini, dilakukan beberapa percobaan untuk memilih parameter yang optimal. Parameter yang dipilih berdasarkan hasil dari evaluasi nilai koefisien *silhouette* yang tertinggi. Seperti di tahap pembagian dataset, menentukan parameter juga dilakukan setiap bulannya.

Sebelumnya, dilakukan percobaan menentukan epsilon dengan menggunakan metode *euclidean distance*. *Euclidean distance* diterapkan dengan digambarkannya sebuah plot/grafik dimana sumbu x merupakan objek dan sumbu y merupakan jarak c tetangga terdekat dibentuk

arah menaik dengan menghitung perbedaan garis pada kemiringan untuk mendapatkan nilai *Eps*, kemudian nilai parameter *Eps* yang optimal didapatkan dari perubahan kemiringan yang signifikan dari objek (Pakuani & Kurniawan, 2021) dikutip dalam (Elbatta, Bolbol, & Ashour, 2012). Terlihat pada gambar 4.21 terbentuk siku di sekitar nilai 0.015.



Gambar 4.21 Hasil visualisasi epsilon menggunakan *euclidean distance*

Kemudian, dilakukan percobaan dengan eps 0.015 dan minpts 4, dan diperoleh hasil *silhouette coefficient* yang jelek yaitu -0.351. Oleh karena itu, menentukan epsilon dengan menggunakan metode *euclidean distance* tidak cocok digunakan pada kasus penelitian ini. Dengan demikian, penelitian ini tidak dilanjutkan untuk menentukan parameter epsilon menggunakan metode *euclidean distance*.

Dalam percobaan uji parameter untuk nilai epsilon di ambil setiap 0.2 kelipatan sekali hingga 1, dikarenakan 0.1 epsilon merepresentasikan jarak 100meter dalam satuan radian, sehingga jarak yang diambil mulai dari 300meter hingga 1kilometer, hal ini dikarenakan jarak tersebut sudah cukup jauh untuk merepresentasikan area hotspot penjemputan taksi. Sedangkan untuk nilai minpts diambil dari angka 10-20, hal ini dikarenakan minpts merupakan representasi dari jumlah minimun titik dalam jangkauan epsilon, dan jumlah minimum tersebut cukup untuk mengetahui area penjemputan.

Terlihat pada tabel 4.1 hasil nilai koefisien silhouette tertinggi yaitu 0.793 pada eps 0.5 dan minpts 20 di bulan januari, kemudian pada bulan februari ditunjukkan pada tabel 4.2 dengan hasil nilai koefisien silhouette tertinggi yaitu 0.806 terdapat pada eps 1 dan minpts 11, dan pada bulan maret ditunjukkan pada tabel 4.3 dengan hasil nilai *silhouette coefficient* tertinggi yaitu 0.807 terdapat pada eps 0.5 dan minpts 20.

Tabel 4.1 Hasil *silhouette coefficient* dari beberapa uji coba parameter bulan januari

Eps / MinPts	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
10	0.534	0.556	0.503	0.560	0.562	0.589	0.608	0.579
11	0.534	0.555	0.676	0.555	0.562	0.582	0.615	0.581
12	0.616	0.554	0.675	0.731	0.738	0.582	0.615	0.612
13	0.597	0.559	0.617	0.730	0.736	0.582	0.616	0.613
14	0.562	0.634	0.616	0.762	0.733	0.585	0.605	0.614
15	0.558	0.634	0.698	0.762	0.767	0.741	0.604	0.616
16	0.571	0.633	0.698	0.697	0.767	0.772	0.604	0.605
17	0.571	0.632	0.698	0.702	0.779	0.772	0.589	0.605
18	0.634	0.633	0.792	0.700	0.779	0.772	0.747	0.749
19	0.633	0.635	0.792	0.700	0.703	0.784	0.748	0.748
20	0.635	0.633	**0.793**	0.700	0.703	0.784	0.786	0.777

Tabel 4.2 Hasil *silhouette coefficient* dari beberapa uji coba parameter bulan februari

Eps / MinPts	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
10	0.590	0.674	0.677	0.760	0.718	0.763	0.803	0.782
11	0.589	0.586	0.790	0.760	0.761	0.763	0.803	**0.806**
12	0.589	0.586	0.790	0.789	0.764	0.803	0.766	0.806
13	0.602	0.636	0.790	0.796	0.763	0.800	0.805	0.803
14	0.593	0.464	0.791	0.796	0.799	0.767	0.805	0.803
15	0.602	0.623	0.797	0.661	0.799	0.769	0.805	0.804
16	0.604	0.617	0.775	0.796	0.705	0.765	0.771	0.804
17	0.612	0.623	0.774	0.797	0.705	0.801	0.766	0.805
18	0.612	0.639	0.774	0.797	0.705	0.697	0.765	0.804
19	0.603	0.639	0.774	0.797	0.696	0.801	0.802	0.748
20	0.553	0.639	0.756	0.796	0.696	0.800	0.802	0.803

Tabel 4.3 Hasil *silhouette coefficient* dari beberapa uji coba parameter bulan maret

Eps / MinPts	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
10	0.567	0.715	0.742	0.541	0.506	-0.338	-0.280	-0.279
11	0.573	0.714	0.724	0.755	0.570	-0.284	-0.278	-0.279
12	0.578	0.602	0.717	0.753	0.769	0.574	0.572	-0.277
13	0.595	0.604	0.716	0.751	0.769	0.769	0.771	0.154
14	0.594	0.619	0.670	0.746	0.763	0.768	0.771	0.148
15	0.567	0.620	0.669	0.747	0.753	0.768	0.770	0.147
16	0.565	0.619	0.787	0.746	0.758	0.770	0.771	0.144
17	0.604	0.638	0.787	0.746	0.756	0.755	0.771	0.770
18	0.604	0.590	0.788	0.673	0.755	0.759	0.771	0.771
19	0.604	0.543	0.806	0.750	0.750	0.757	0.758	0.771
20	0.604	0.590	**0.807**	0.673	0.750	0.757	0.758	0.771

B. Clustering

Setelah ditentukan parameter epsilon dan minpts, maka dilanjutkan proses *clustering* menggunakan algoritma DBSCAN untuk setiap bulan dengan *code* yang dapat dilihat pada gambar 4.22 Kemudian setelah dilakukan *clustering*, terlihat pada table 4.4 menunjukkan hasil clustering pada bulan januari, februari dan maret.

```
kms_per_radian = 6371
#Mengubah epsilon dalam bentuk satuan kilometer
epsilon = eps / kms_per_radian
#Mengaplikasikan algoritma DBSCAN
clusters = DBSCAN(eps=epsilon, min_samples=minpts, algorithm='ball_tree',
                  metric='haversine').fit(df_pickup)
label = clusters.labels_

print(f'Number of clusters found: {len(np.unique(label[label!=-1]))}')
print(f'Number of outliers found: {len(label[label==-1])}')
```

Gambar 4.22 *Code clustering* DBSCAN

Tabel 4.4 Perbandingan hasil *clustering* bulan januari, februari, dan maret

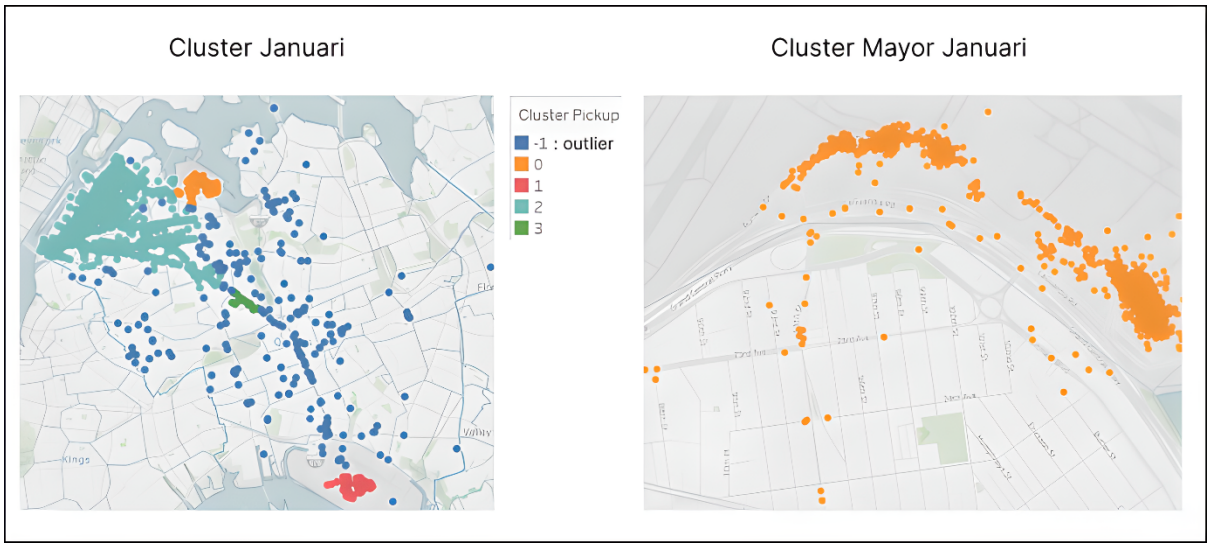
	Januari				**Februari**		**Maret**			
Epsilon	0.5				1		0.5			
MinPts	20				11		20			
Jumlah Cluster	4				2		4			
Jumlah setiap cluster	5351	3808	2203	30	7692	3374	3870	6187	2334	37
Outlier	265				103		273			
Akurasi (SC)	0.79318				0.80603		0.80720			

C. Analisis Hasil *Clustering*

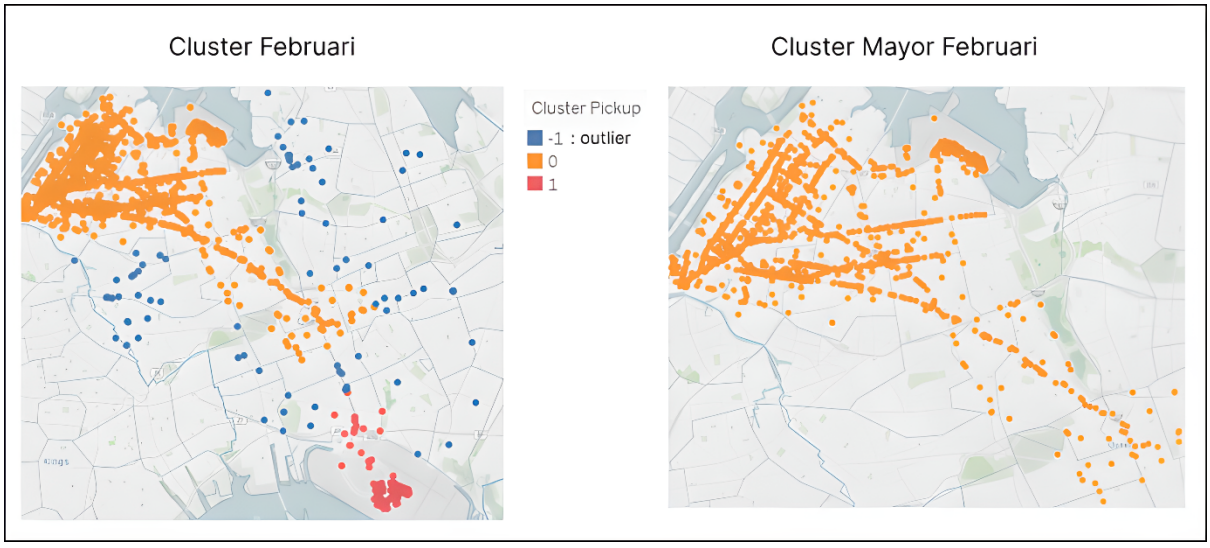
Hasil *clustering* pada setiap bulannya memperoleh jumlah *cluster* yang berbeda-beda. Analisis ini hanya berfokus untuk *cluster mayor* atau yang memiliki jumlah titik penjemputan paling banyak di antara *cluster* lainnya di setiap bulan. *Cluster mayor* di bulan januari berada di *cluster* 0, di bulan februari di *cluster* 0, dan di bulan maret pada *cluster* 1. Hasil *clustering* ditunjukkan dalam bentuk peta karena *dataset* yang dimiliki berbentuk geospasial.

Gambar 4.23 merupakan hasil *clustering* pada bulan januari, *cluster mayor* yang terbentuk terdapat di area Laguardia Airport dengan titik *pickup* sebanyak 5351. Selanjutnya, pada gambar 4.24 menunjukkan hasil *clustering* pada bulan februari, *cluster mayor* yang terbentuk terdapat di area dengan cakupan yang cukup luas yaitu di area LaGuardia Airport, dan disepanjang jalan area lain seperti Steinwey Street, Queens Boulevard, dan jalan lainnya dengan titik yang banyak di sekitar pusat makanan dan perbelanjaan dengan titik *pickup*

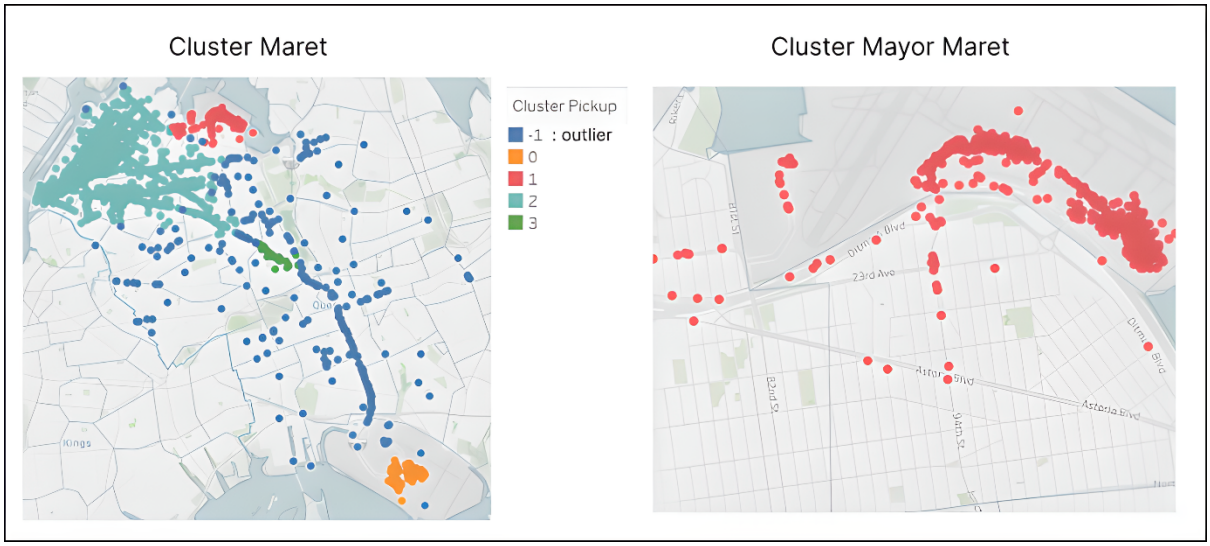
sebanyak 7692. Pada gambar 4.25 menunjukkan hasil *clustering* pada bulan maret, *cluster mayor* yang terbentuk mirip seperti pada bulan februari yaitu terdapat di area Laguardia Airport dengan 6187 titik *pickup.*



Gambar 4.23 Hasil *clustering* plot DBSCAN bulan januari



Gambar 4.24 Hasil *clustering* plot DBSCAN bulan februari



Gambar 4.25 Hasil *clustering* plot DBSCAN bulan maret

Setelah mengetahui area dari hasil *cluster mayor* di setiap bulan, dapat dilakukan analisis statistik untuk melihat karakteristik dari hasil *cluster*. Pada tabel 4.5 merupakan perbandingan antara bulan Januari, Februari, dan Maret untuk *cluster mayor.*

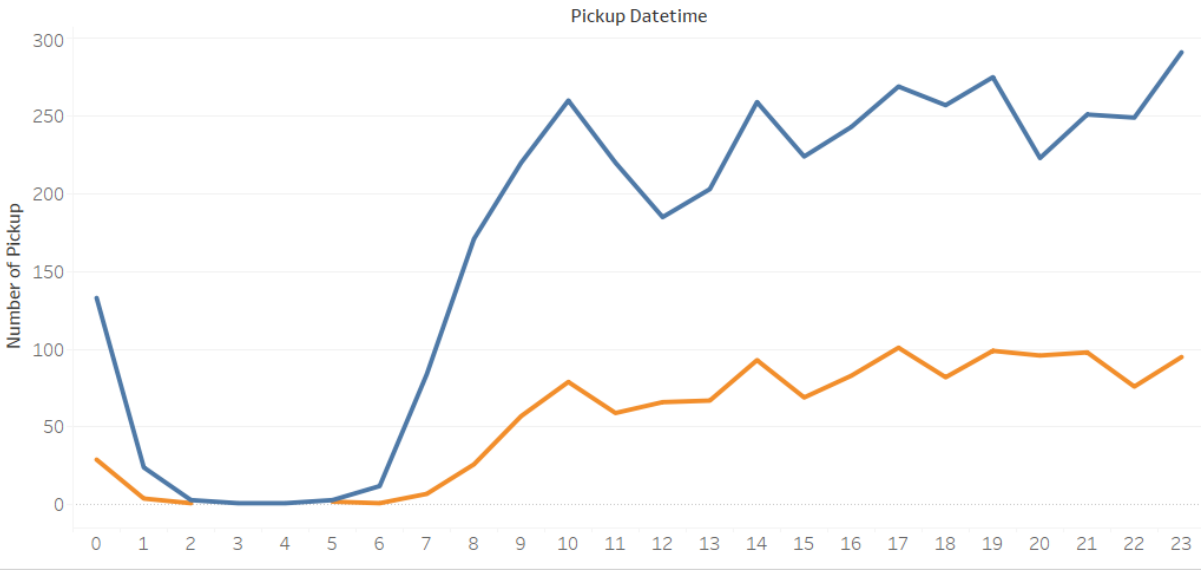
Tabel 4.5 Perbandingan statistik *cluster mayor* pada bulan januari, februari dan maret

	Januari	**Februari**	**Maret**
Rata-rata Jarak Perjalanan	9.8 km	8 km	9.7 km
Rata-rata Durasi Perjalanan	30 menit	29 menit	31 menit
Rata-rata Kecepatan	23 km/jam	23 km/jam	22 km/jam

Seperti yang terlihat pada tabel 4.5 jarak perjalanan berbanding lurus dengan durasi perjalanan, yang berarti semakin jauh jarak perjalanan maka durasi perjalanan pun akan semakin lama. Jika dilihat dari rata-rata kecepatan di setiap bulan cukup rendah hanya berkisar 20 km/jam. Hal ini, dapat dipengaruhi terhadap faktor tertentu seperti padatnya lalu lintas kondisi jalan, dan keterampilan pengemudi. Dengan begitu, dari evaluasi tersebut dapat direkomendasikan bagi perusahaan taksi untuk menggunakan teknologi seperti maps, yang dapat membantu pengemudi untuk memilih rute terbaik agar dapat meminimalkan waktu atau durasi perjalanan, memberikan pelatihan kepada pengemudi untuk meningkatkan keterampilan yang dapat mempercepat perjalanan. Selain bagi perusahaan taksi, dapat direkomendasikan untuk memperbaiki kondisi jalan apabila jalan buruk dikarenakan dapat beresiko bagi keselamatan penumpang, dan jika kondisi jalan bagus akan mempercepat waktu perjalanan.

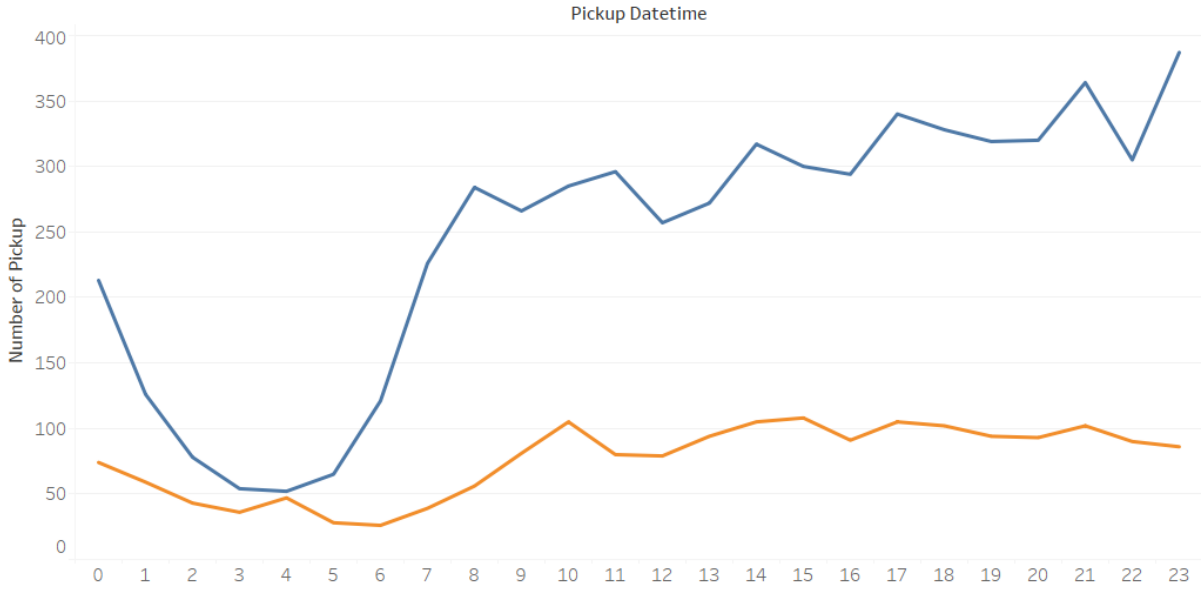
Setelah dilakukan analisis statistik, maka dapat dilakukan analisis berdasarkan waktu penjemputan di setiap jam dengan melihat perbandingan antara hari kerja dan akhir pekan.

Pada setiap grafik yang ditunjukkan di bulan januari, februari dan maret pada gambar 4.26, 4.27, 4.28 terlihat bahwa saat *weekend* jumlah titik penjemputan selalu lebih rendah dari *weekday*. Hal ini dikarenakan, jumlah permintaan taksi berbeda dan pada waktu *weekday* dikarenakan memiliki lebih banyak hari yaitu 5 hari dibandingkan *weekend* yang hanya 2 hari.



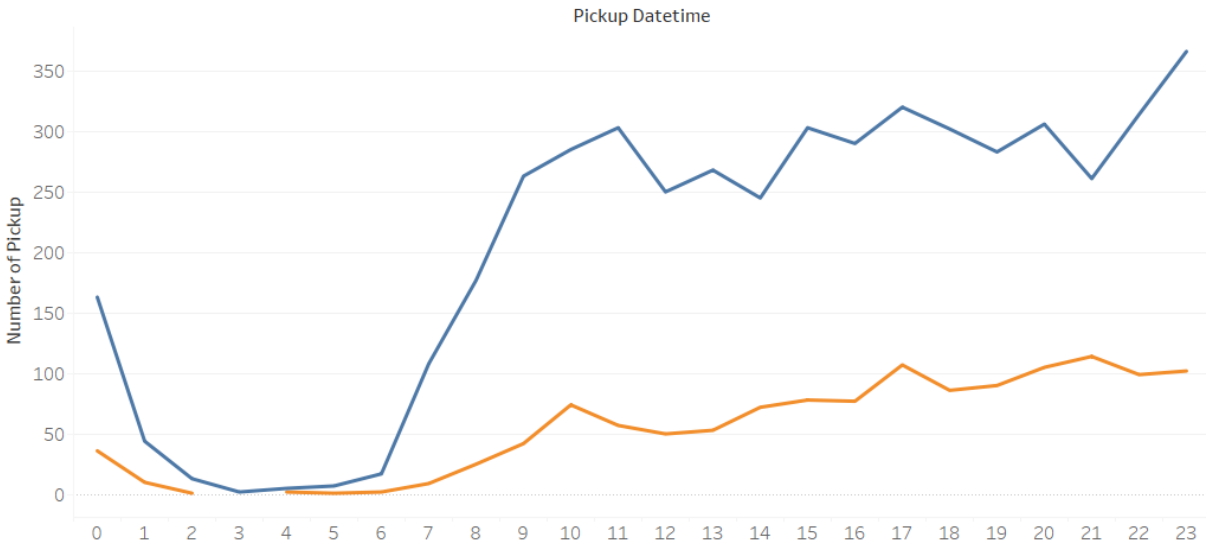
Gambar 4.26 Perbandingan *Weekday* dan *Weekend* berdasarkan waktu penjemputan (jam) bulan januari

Di bulan januari hasil *cluster mayor* terdapat di area Laguardia airport. LaGuardia Airport yaitu bandara yang mengakomodasi layanan penerbangan ke tujuan domestik dan internasional terbatas dan merupakan bandara tersibuk ketiga di wilayah metropolitan New York. Terlihat pada gambar 4.26 pada waktu *weekday* dan *weekend* memiliki pola yang cukup mirip. Permintaan taksi menaik mulai pukul 07.00 hingga 10.00 dan kemudian perlahan menurun tetapi tidak signifikan hingga pukul 12.00 pada saat weekday, dan meningkat lagi hingga pukul 23.00 pada saat *weekday* maupun *weekday*. Dengan demikian, area ini bisa menjadi titik *cluster* penjemputan saat pagi hari hingga larut malam



Gambar 4.27 Perbandingan *Weekday* dan *Weekend* berdasarkan waktu penjemputan (jam) bulan februari

Pada gambar 4.27 hasil *cluster mayor* bulan februari terdapat di area Laguardia Airport dan sekitarnya. Jumlah *pickup* ditunjukkan pada gambar 4.27 pada hari kerja maupun akhir pekan memiliki pola yang mirip, dengan demikian, analisis dapat dilakukan secara bersamaan. Dimulai pada pukul 07.00 hingga 23.00 permintaan taksi termasuk kedalam kategori yang tinggi, meskipun pada jam tertentu mengalami penurunan, tetapi tidak signifikan. Dengan demikian, area ini menjadi titik penjemputan taksi dari pukul 07.00 hingga larut malam.



Gambar 4.28 Perbandingan *Weekday* dan *Weekend* berdasarkan waktu penjemputan (jam) bulan maret

Pada gambar 4.28 hasil *cluster mayor* bulan maret sama seperti bulan janurai yaitu terdapat di area Laguardia Airport. Terlihat bahwa permintaan taksi pada hari kerja maupun akhir pekan memiliki pola yang mirip. Permintaan taksi mulai perlahan naik pada pukul 08.00 dan terus meningkat hingga jam 23.00, meskipun di beberapa jam tertentu menurun tetapi tidak signifikan, dan semakin naik pada sore hari hingga larut malam, kemudian perlahan menurun dari tengah malam hingga fajar.

Dari hasil *clustering* pada bulan januari, februari, dan maret, terbentuk pola yang mirip satu sama lain yaitu ketiganya memiliki *cluster mayor* dengan titik penjemputan taksi terbanyak yang berada di area LaGuardia Airport. Selain itu, perbandingan antara weekend dan weekday di setiap bulannya jika dilihat dari gambar 4.26, 4.27, dan 4.28 memperoleh hasil pola tidak berbeda jauh atau bisa dikatakan mirip. Oleh karena itu, pola yang terbentuk di setiap bulan tidak memiliki perbedaan yang signifikan dari segi lokasi maupun waktu. Dengan demikian, area tersebut dapat menjadi rekomendasi sebagai pangkalan taksi dalam melakukan penjemputan penumpang hal ini dikarenakan hampir di setiap jam memiliki permintaan taksi yang tinggi, dengan begitu dapat mengurangi waktu tunggu penumpang. Waktu penjemputan taksi direkomendasikan dimulai pada pagi hari pukul 07.00 hingga larut malam.

4.4 Implementasi

4.4.1 *Library*

Dalam implementasi pemodelan kedalam aplikasi web Streamlit dibutuhkan beberapa *library* yang ditunjukkan pada gambar 4.29 ***Library* streamlit** merupakan *library* utama yang digunakan untuk membangun aplikasi web Streamlit berbasis python. ***Library* streamlit_option_menu** digunakan untuk membuat dan memilih opsi dalam menu. Library pandas berfungsi untuk memproses dan manipulasi data. ***Library* numpy** dan math digunakan untuk proses sejumlah fungsi matematika. ***Library* plotly.express** berfungsi untuk memudahkan proses visualisasi data. ***Library* sklearn.cluster** digunakan untuk proses pemodelan untuk mengelompokkan data yang tidak berlabel. ***Library* sklearn.metrics** digunakan untuk mengukur performance kinerja pada model.

```
import streamlit as st
from streamlit_option_menu import option_menu
import pandas as pd
import numpy as np
from math import radians, cos, sin, asin, sqrt
```

```
import plotly.express as px
from sklearn.cluster import DBSCAN
from sklearn.metrics import silhouette_score
```

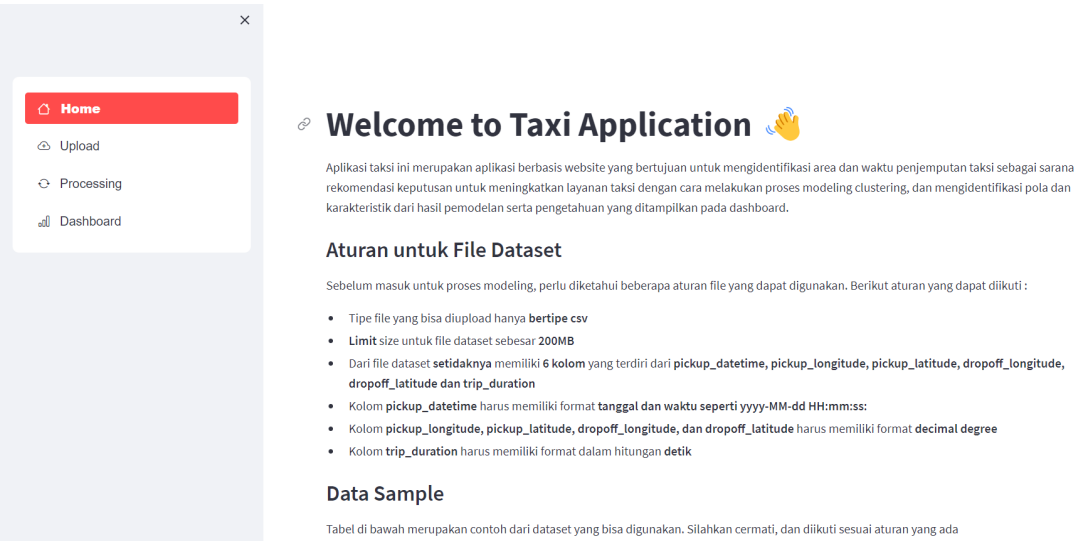
Gambar 4.29 *Library*

4.4.2 Hasil

Aplikasi ini dibentuk atas dasar framework Streamlit menggunakan bahasa pemrograman python. Dalam mengakses aplikasi diperlukan akses jaringan *online* agar bisa dijalankan. Terdapat beberapa fitur pada saat memproses memerlukan waktu yang cukup lama tergantung dari seberapa besarnya data yang ada. Hal ini merupakan salah satu kelemahan Streamlit dalam hal kecepatan.

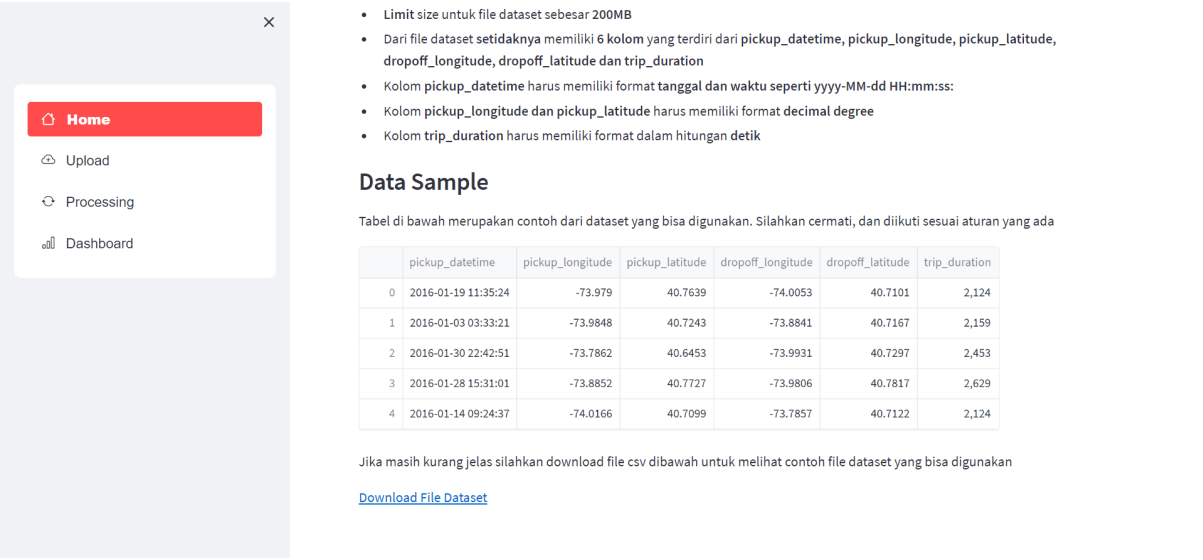
A. Halaman Home

Pada gambar 4.30 merupakan halaman home yang akan ditampilkan pertama kali pada aplikasi dijalankan, dimana terdapat informasi mengenai tujuan dari aplikasi ini serta aturan dalam file dataset yang bisa digunakan. Dalam aplikasi ini bisa digunakan dataset taksi dari kota lain, tetapi harus mengikuti aturan yang telah ditetapkan. Akan tetapi, penelitian ini hanya terbatas menggunakan dataset taksi Queens, New York City.



Gambar 4.30 Halaman *home* 1

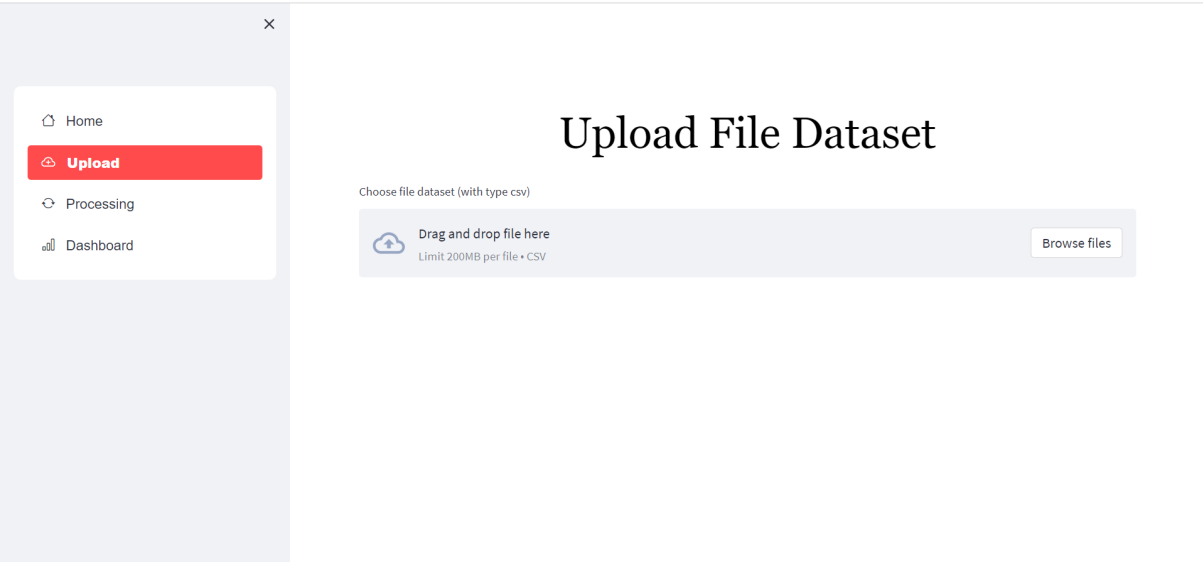
Selanjutnya, pada gambar 4.31 terdapat data sample dan file dataset yang bisa di download sebagai contoh dari dataset yang dapat digunakan agar lebih jelas.



Gambar 4.31 Halaman *home* 2

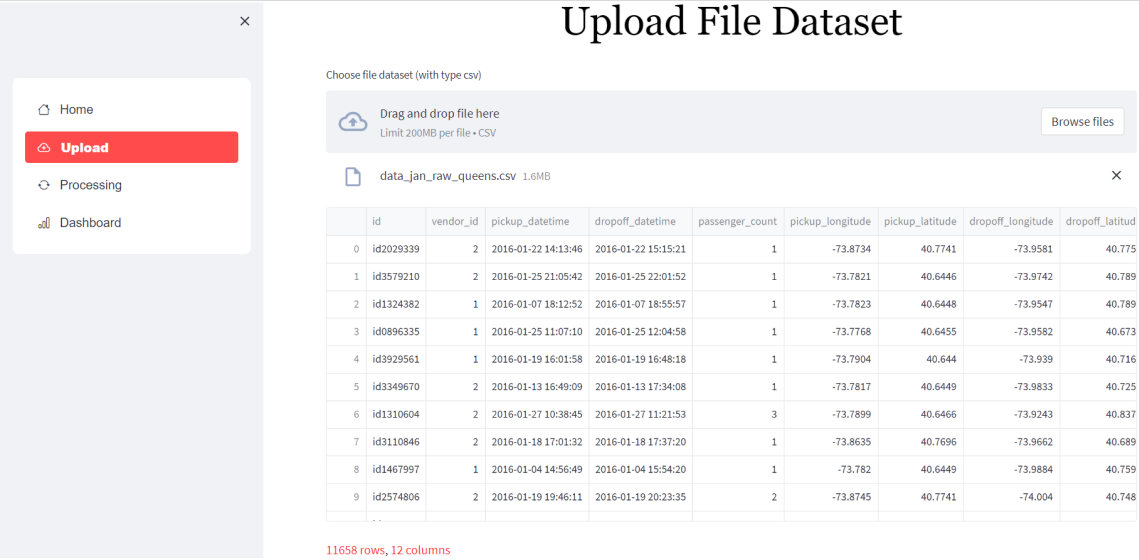
B. Halaman Upload

Pada gambar 4.32 merupakan tampilan dari upload file dataset. Pengguna bisa mengupload file dengan menekan tombol “Browse files”, kemudian memilih file dari file komputer dalam format csv dengan maksimal ukuran file sebesar 200mb. Hal ini merupakan salah satu kelemahan dari Streamlit.



Gambar 4.32 Halaman *upload* 1

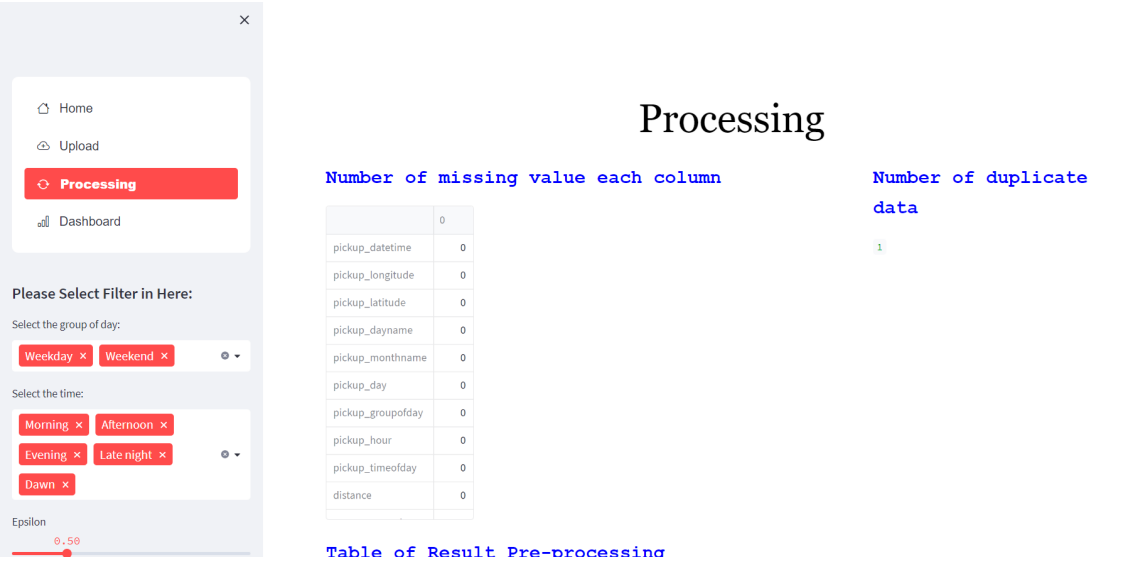
Setelah dataset di upload, kemudian sistem akan menampilkan dataset mentah dari file yang telah diupload berserta informasi mengenai jumlah baris dan kolom ditunjukkan pada gambar 4.33.



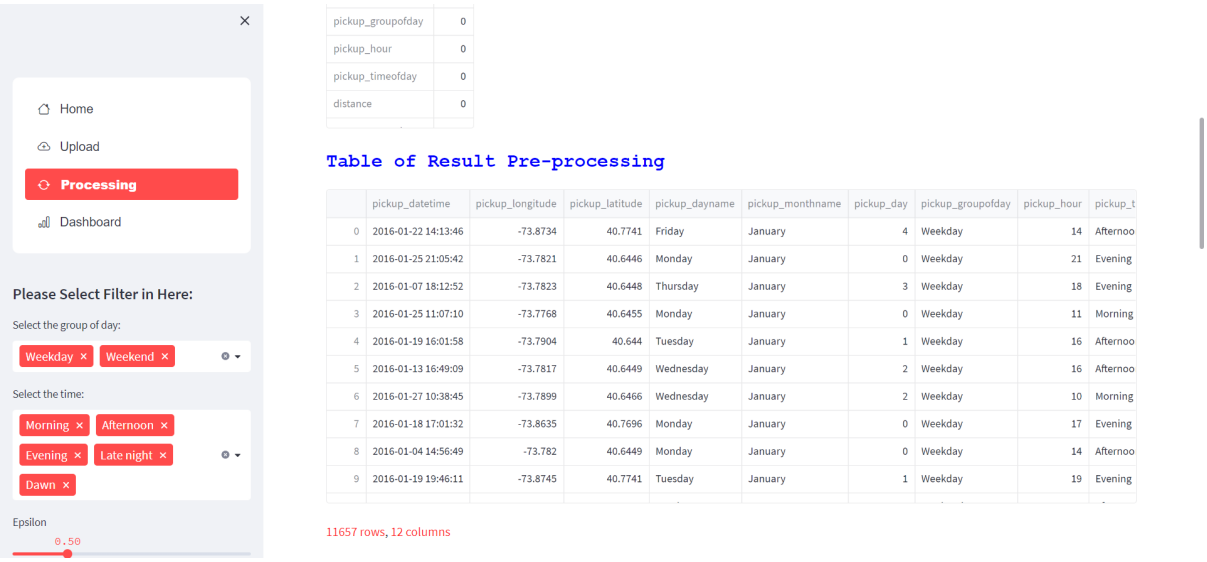
Gambar 4.33 Halaman *upload* 2

C. Halaman *Processing*

Halaman processing menampilkan informasi jumlah missing value setiap kolom beserta jumlah data duplikat pada gambar 4.34, kemudian menampilkan informasi tabel dari hasil *pre-processing* berupa *feature creation* dan *data cleaning* berdasarkan file yang telah di *upload* ditunjukkan pada gambar 4.35. Hal ini berguna kepada pengguna untuk mengetahui informasi yang ada pada dataset, dalam proses *processing* jika terdapat *missing value* ataupun data yang duplikat akan dihapus sehingga jumlah baris berkurang.

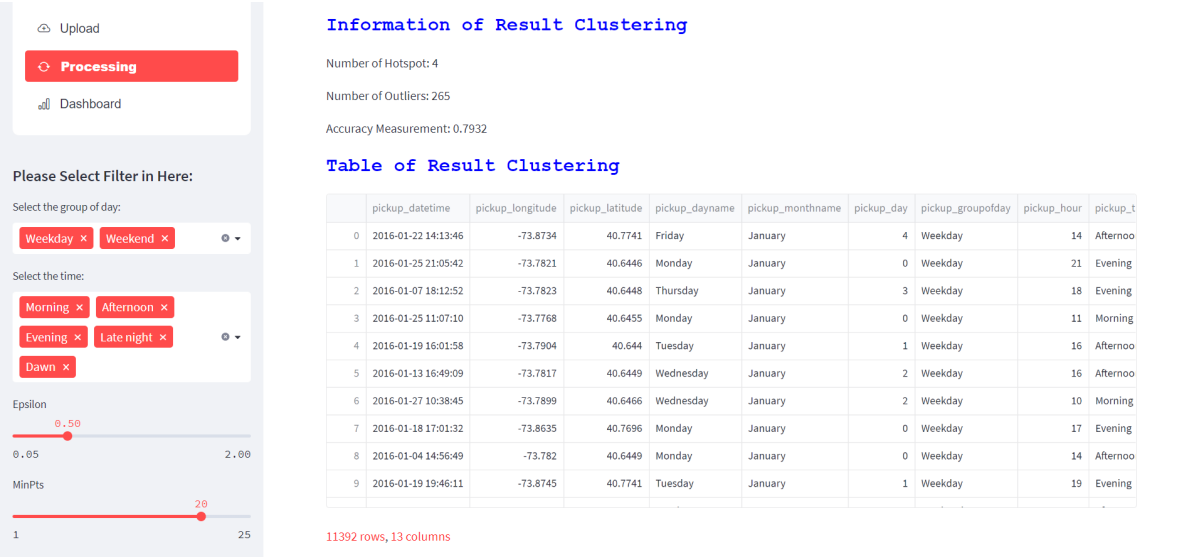


Gambar 4.34 Halaman *processing* 1

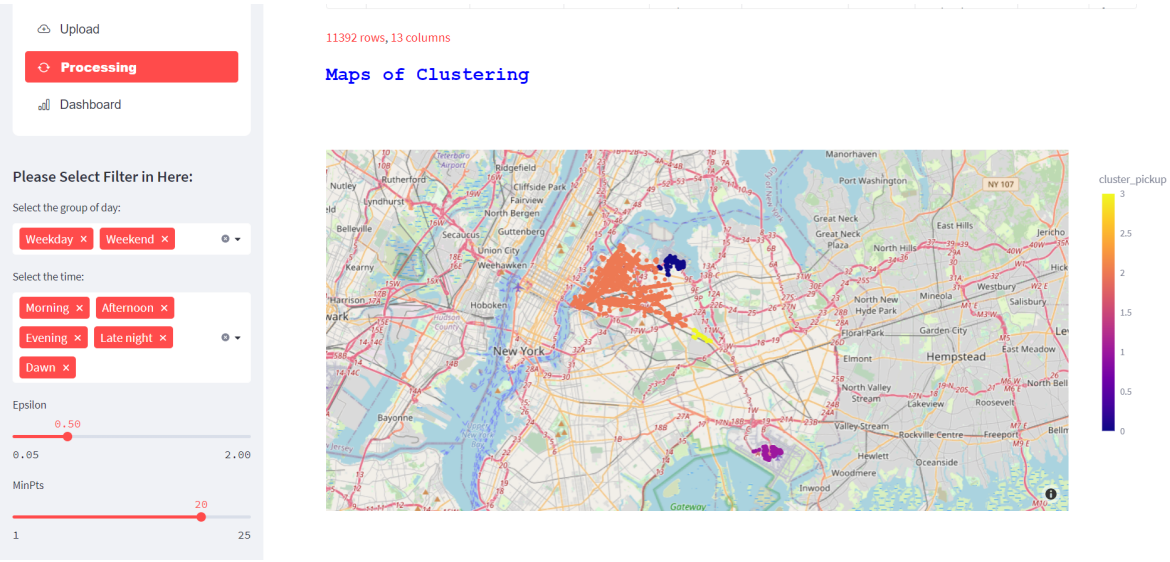


Gambar 4.35 *Hasil processing* 2

Sebelum proses *clustering*, pengguna dapat menentukan varian yang akan dilakukan pemodelan clustering dengan memilih filter pada "Group of Day" atau "Time of Day" yang berada di *sidebar* terlihat pada gambar 4.35. Selanjutnya seperti pada gambar 4.36, proses *clustering* dijalankan pada halaman ini dan akan menampilkan informasi hasil *clustering* berupa jumlah *cluster*, jumlah *outlier*, dan akurasi *silhouette coefficient*. Sebelum menjalankan proses *clustering*, pengguna dapat menetapkan parameter epsilon dan minpts pada *slider*. Setelah menetapkan parameter, maka aplikasi akan menampilkan berupa informasi serta tabel dari hasil *clustering*. Pada gambar 4.37 merupakan hasil clustering dalam bentuk visualisasi maps.



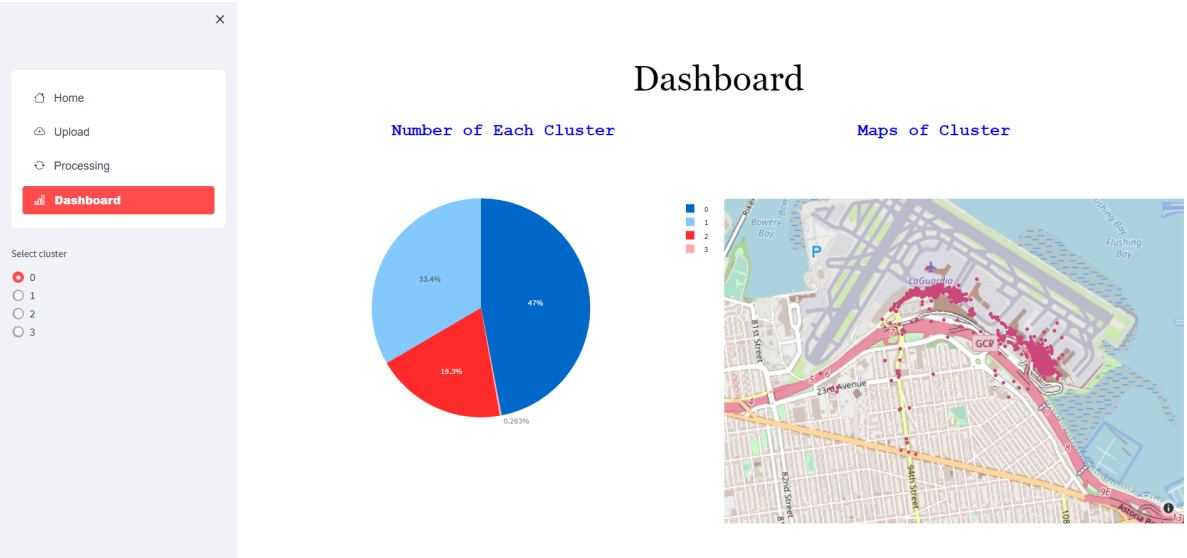
Gambar 4.36 Hasil *processing* 3



Gambar 4.37 *Hasil processing* 4

D. Halaman *Dashboard*

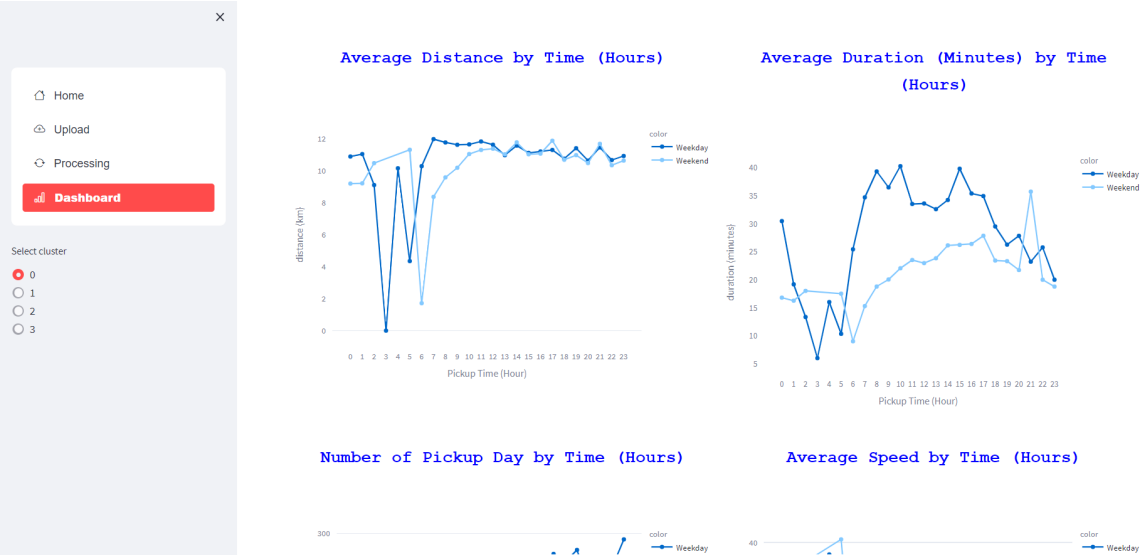
Pada gambar 4.38, 4.39, dan 4.40 merupakan halaman *dashboard* yang menampilkan beberapa grafik. Grafik tersebut diperoleh dari hasil *clustering* yang telah dilakukan sebelumnya. Dalam hal ini bertujuan untuk mempermudah dalam memahami hasil analisis dan menyajikan informasi yang berguna dalam bentuk visualisasi data bagi pengguna.



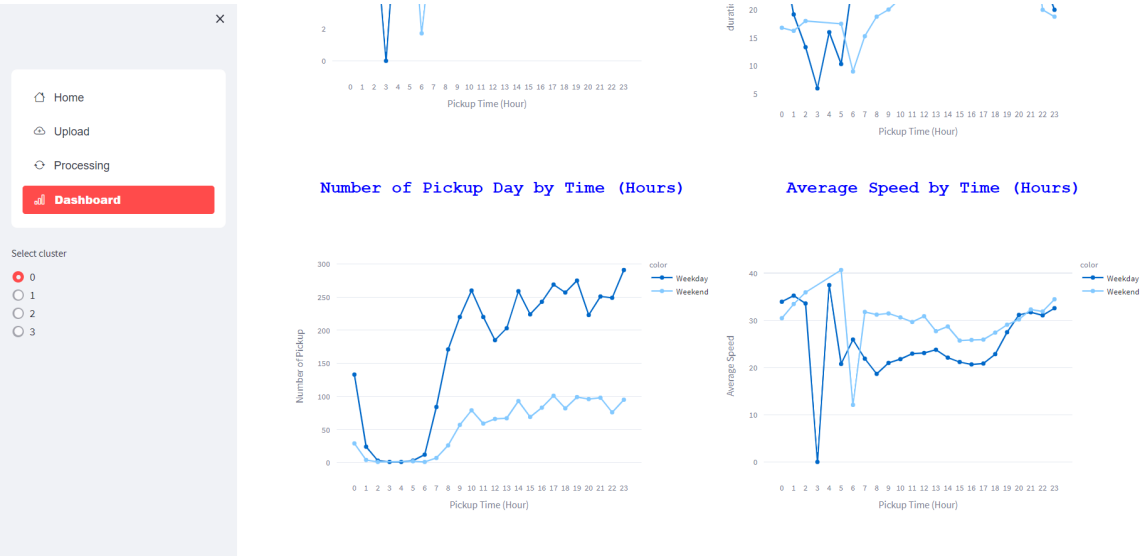
Gambar 4.38 Halaman dashboard 1

Pada halaman dashboard terdapat filter “select cluster” dalam bentuk radio button. Filter tersebut bersifat dinamis tergantung seberapa banyak jumlah hasil dari cluster yang dihasilkan

dari hasil pemodelan. Filter ini berfungsi pada visualisasi “Maps of Cluster”, “Average Distance by Time (Hours)”, “Average Duration (minutes) by Time (Hours)”, “Number of Pickup Day by Time (Hours)”, dan Average Speed by Time (Hours). Visualisasi dengan tipe line chart merupakan informasi berdasarkan waktu di setiap jam. Pengguna dapat memilih cluster berapa yang akan divisualisasikan, pada gambar 4.38 dan 4.39 cluster yang dipilih yaitu cluster 0 dengan begitu, grafik yang terbentuk yaitu cluster yang hanya berasal dari cluster 0.



Gambar 4.39 Halaman dashboard 2



Gambar 4.40 Halaman dashboard 3

4.5 Pengujian

Pengujian aplikasi ini menggunakan *black box testing* dengan teknik *equivalence partitioning* dengan cara menjalankan 16 kasus pengujian terhadap aplikasi. Terdapat 4 kelompok kasus uji yaitu pada halaman *home, upload, processing, dan dashboard* dimana di setiap kelompoknya terdapat beberapa kasus uji. Kemudian, hasil pengujian akan dibandingan dengan hasil yang diharapkan. Pada tabel 4.6 merupakan hasil pengujian pada aplikasi.

Tabel 4.6 Hasil pengujian

No	Skenario	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
Halaman Home					
1	Menampilkan informasi dari halaman home	Menjalankan aplikasi pertama kali	Menampilkan informasi berupa tujuan dari aplikasi, aturan dataset dan contoh data sample	Sesuai	Valid
2	Mendownload file dataset csv	Menekan tulisan "Download File Dataset"	File dataset berhasil di download	Sesuai	Valid
Halaman Upload					
1	Mengunggah file dataset (csv) dan ukuran 200mb	Menekan tombol "Browse files"	Menampilkan tabel data dari file	Sesuai	Valid
2	Mengupload file dataset csv dan ukuran diatas 200mb	Menekan tombol "Browse files"	Muncul status *warning*	Sesuai	Valid
3	Tidak mengunggah file	Mengosongkan file	Tidak error	Sesuai	Valid
4	Mengupload file selain format csv	Menekan tombol "Browse files"	Data tidak diizinkan/diproses	Sesuai	Valid
Halaman Processing					
1	Mengunggah file dataset (csv) dan ukuran 200mb	Menekan tombol "Browse files"	Menampilkan tabel data dari file	Sesuai	Valid

2	Tidak Menampilkan hasil *processing*	Menekan tombol "Processing" pada menu di *sidebar* sebelum file dataset berhasil di upload	Menampilkan status info untuk menambahkan file terlebih dahulu	Sesuai	Valid
3	Menentukan filter hari untuk proses *clustering*	Memilih filter "group of day" *weekday* pada *sidebar*	Baris pada tabel berubah, dan menjalankan proses *clustering* dari awal	Sesuai	Valid
4	Menentukan filter waktu untuk proses *clustering*	Memilih filter "group of time" *morning* pada sidebar	Baris pada tabel berubah, dan menjalankan proses *clustering* dari awal	Sesuai	Valid
5	Tidak memilih filter hari dan waktu	Menghapus filter yang ada pada *sidebar*	Menampilkan status warning dan tidak menampilkan hasil *clustering*	Sesuai	Valid
6	Menentukan jumlah MinPts	Menggeser slider "MinPts" di *sidebar*	Menjalankan proses *clustering* dari awal dan menampilkan hasil *clustering*	Sesuai	Valid
7	Menentukan jumlah epsilon	Menggeser slider "Epsilon" di *sidebar*	Menjalankan proses *clustering* dari awal dan menampilkan hasil *clustering*	Sesuai	Valid
Halaman Dashboard					
1	Menampilkan visualisasi dari hasil *clustering*	Menekan tombol "Dashboard" pada menu di *sidebar* setelah dataset berhasil di *upload*	Menampilkan berbagai visualisasi dari hasil *clustering*	Sesuai	Valid
2	Menentukan filter *cluster* untuk visualisasi	Memilih filter "Select cluster" di *sidebar*	Menjalankan proses visualisasi sesuai dengan *cluster* yang dipilih	Sesuai	Valid
3	Tidak Menampilkan hasil visualisasi dashboard	Menekan tombol "Dashboard" sebelum file dataset berhasil di *upload*	Menampilkan status info untuk menambahkan file terlebih dahulu	Sesuai	Valid

Hasil pengujian black box testing pada 16 kasus uji yang dijalankan pada aplikasi menunjukkan hasil keseluruhan nilai yang valid terlihat pada table 4.6. Dengan demikian, skor dalam pengunjian ini memperoleh sebesar 100%.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini bermanfaat sebagai sarana rekomendasi keputusan untuk meningkatkan layanan taksi, dengan mengidentifikasi area dan waktu penjemputan permintaan taksi yang tinggi di Queens, New York City pada periode Januari hingga Maret 2016, menggunakan metode *clustering* DBSCAN. Dalam penelitian ini, dilakukan pemodelan *clustering* untuk mengelompokkan area berdasarkan kepadatan penjemputan taksi dan kemudian hasil *cluster* dianalisis secara statistik untuk melihat karakteristik dari waktu penjemputan taksi di setiap jam.

Hasil *cluster* yang terbentuk pada bulan januari sebanyak 4, februari sebanyak 2, dan maret sebanyak 4. Performa pengukuran *clustering* pada setiap bulannya memperoleh nilai akurasi yang dikategorikan bagus dan diukur menggunakan metrik *silhouette coefficient.* Dari setiap bulan terbentuk *cluster* mayor atau kepadatan yang tinggi dengan area yang sama yaitu Laguardia Airport, serta pola waktu penjemputan yang tidak berbeda secara signifikan dari ketiga bulan tersebut serta waktu yang direkomendasikan pada saat pagi hari pukul 07.00 hingga larut malam.

Selanjutnya, penelitian ini diimplementasikan ke dalam bentuk aplikasi website. Aplikasi ini berhasil di kembangkan menggunakan streamlit berbasis python yang memungkinkan pengguna untuk melakukan pemodelan *clustering* secara otomatis dan mendapatkan informasi mengenai area dan waktu dari hasil clustering. Kemudian, telah dilakukan pengujian dengan menggunakan metode *black box testing* dengan 16 kasus uji berhasil dilakukan dan memperoleh performa yang baik sebesar 100%.

5.2 Saran

Peneliti menyadari masih terdapat beberapa kekurangan serta batasan yang ada pada penelitian ini, maka dari itu peneliti dapat menyarankan untuk perbaikan penelitian selanjutnya, dapat dilakukan berupa:

a. Mencoba menggunakan metode lain dalam melakukan pemodelan *clustering*
b. Menemukan metode yang optimal dalam menentukan Epsilon dan MinPts dalam algoritma DBSCAN

c. Menemukan potensi lain untuk meningkatkan pelayanan taksi dari variable tertentu.
d. Menggunakan dataset lain yang lebih spesifik, seperti pada perayaan hari raya.

DAFTAR PUSTAKA

Almantara, I. P. S., Aryani, N. W. S., & Swamardika, I. B. A. (2020). Spatial Data Analysis using DBSCAN Method and KNN classification. *International Journal of Engineering and Emerging Technology*, *5*(2), 77–80. https://doi.org/10.24843/IJEET.2020.v05.i02.p013

Aminah, S. (2018). TRANSPORTASI PUBLIK DAN AKSESBILITAS MASYARAKAT PERKOTAAN. *Jurnal Teknik Sipil*, *9*(1), 1142–1155. https://doi.org/10.36448/jts.v9i1.1135

Amiruzzaman, M., Rahman, R., Islam, M. R., & Nor, R. M. (2022, June 27). *Logical analysis of built-in DBSCAN Functions in Popular Data Science Programming Languages*. OSF Preprints. https://doi.org/10.31219/osf.io/ge654

Amiruzzaman, M., Rahman, R., Islam, Md. R., & Nor, R. M. (2021). Evaluation of DBSCAN algorithm on different programming languages: An exploratory study. *2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, 1–6. https://doi.org/10.1109/ICEEICT53905.2021.9667925

Anggara, M., Sujiani, H., & Nasution, H. (2016). *Pemilihan Distance Measure Pada K-Means Clustering Untuk Pengelompokkan Member Di Alvaro Fitness*. *1*(1).

Angraini, P. A. (2018). *STUDI KOMPARATIF PELAYANAN TAKSI ONLINE DAN TAKSI KONVENSIONAL TERHADAP KEPUASAN PELANGGAN DALAM PERSPEKTIF EKONOMI ISLAM (Studi pada Taksi Puspa Jaya di Bandar Lampung)* (Undergraduate, UIN Raden Intan Lampung). UIN Raden Intan Lampung. Retrieved from http://repository.radenintan.ac.id/4975/

Anwar, M. T., Hadikurniawati, W., Winarno, E., & Supriyanto, A. (2019). Wildfire Risk Map Based on DBSCAN Clustering and Cluster Density Evaluation. *Advance Sustainable Science Engineering and Technology*, *1*(1), 0190102. https://doi.org/10.26877/asset.v1i1.4876

Athoillah, A. S., Firdaus, M., & Sanim, B. (2019). Strategi Bersaing Perusahaan Taksi Dalam Menghadapi Perubahan Lingkungan. *CAPITAL: Jurnal Ekonomi dan Manajemen*, *3*(1), 1–15. https://doi.org/10.25273/capital.v3i1.5060

Bäcklund, H., Hedblom, A., & Nejiman, N. (2011). *DBSCAN A Density-Based Spatial Clustering of Application with Noise*.

Bäcklund, H., & Neijman, N. (2011). *TNM 033 2011-1130 1 DBSCAN A Density-Based Spatial Clustering of Application with Noise*. Retrieved from https://www.semanticscholar.org/paper/TNM-033-2011-1130-1-DBSCAN-A-Density-Based-Spatial-B%C3%A4cklund-Neijman/78fd7fd0041f87d37f4fabffeb9d4bec779c0feb

Bharati, M., & Ramageri, B. (2010). Data mining techniques and applications. *Indian Journal of Computer Science and Engineering*, *1*.

Cai, H., Zhan, X., Zhu, J., Jia, X., Chiu, A. S. F., & Xu, M. (2016). Understanding taxi travel patterns. *Physica A: Statistical Mechanics and Its Applications*, *457*, 590–597. https://doi.org/10.1016/j.physa.2016.03.047

Ceryan, N. (2016). *A Review of Soft Computing Methods Application in Rock Mechanic Engineering*. Retrieved from https://www.igi-global.com/chapter/a-review-of-soft-computing-methods-application-in-rock-mechanic-engineering/140384

Chren, J. (n.d.). *Metode pengujian blackbox*. Retrieved from https://www.academia.edu/11980393/Metode_pengujian_blackbox

Dang, S. (2015). Performance Evaluation of Clustering Algorithm Using Different Datasets. *IJARCSMS*, *3*, 167–173.

Devi, A., Putra, I., & Sukarsa, I. (2015). Implementasi Metode Clustering DBSCAN pada Proses Pengambilan Keputusan. *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, 185. https://doi.org/10.24843/LKJITI.2015.v06.i03.p05

Elbatta, M., Bolbol, R., & Ashour, W. (2012). A Vibration Method for Discovering Density Varied Clusters. *ISRN Artificial Intelligence*, *2012*. https://doi.org/10.5402/2012/723516

Ester, M., Kriegel, H.-P., & Xu, X. (1996). *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. 6.

Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques* (2nd ed). Amsterdam ; Boston : San Francisco, CA: Elsevier ; Morgan Kaufmann.

Handriani, D. J. (2019). *Proses Adaptasi Ikatan Mahasiswa Fakfak Di Kota Bandung* (Other, Universitas Komputer Indonesia). Universitas Komputer Indonesia. https://doi.org/10/13.%20UNIKOM_41815217_Dezara%20Judithia%20Handriani_BAB%20V.pdf

He, D., Liu, H., He, K., Meng, F., Jiang, Y., Wang, M., … Wang, Q. (2013). Energy use of, and CO2 emissions from China's urban passenger transportation sector – Carbon

mitigation scenarios upon the transportation mode choices. *Transportation Research Part A: Policy and Practice*, *53*, 53–67. https://doi.org/10.1016/j.tra.2013.06.004

Hidayat, T., & Putri, H. D. (2020). Pengujian Portal Mahasiswa pada Sistem Informasi Akademik (SINA) menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis. *Jutis (Jurnal Teknik Informatika)*, *7*(1), 83–92. https://doi.org/10.33592/jutis.Vol7.Iss1.148

Huang, Z., Gao, S., Cai, C., Zheng, H., Pan, Z., & Li, W. (2021). A rapid density method for taxi passengers hot spot recognition and visualization based on DBSCAN+. *Scientific Reports*, *11*(1), 9420. https://doi.org/10.1038/s41598-021-88822-3

I Made Suwija Putra, S. T. (2018). *ALGORITMA DBSCAN (DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE) DAN CONTOH PERHITUNGANNYA*. Retrieved from http://erepo.unud.ac.id/id/eprint/20097/

Ibrahim, R., & Shafiq, M. O. (2019). Detecting taxi movements using Random Swap clustering and sequential pattern mining. *Journal of Big Data*, *6*(1), 1–26. https://doi.org/10.1186/s40537-019-0203-6

Jian, S., Li, D., & Yu, Y. (2021). Research on Taxi Operation Characteristics by Improved DBSCAN Density Clustering Algorithm and K-means Clustering Algorithm. *Journal of Physics: Conference Series*, *1952*(4), 042103. https://doi.org/10.1088/1742-6596/1952/4/042103

Kong, X., Xia, F., Wang, J., Rahim, A., & Das, S. K. (2017). Time-Location-Relationship Combined Service Recommendation Based on Taxi Trajectory Data. *IEEE Transactions on Industrial Informatics*, *13*(3), 1202–1212. https://doi.org/10.1109/TII.2017.2684163

Kriegel, H.-P., Kröger, P., Sander, J., & Zimek, A. (2011). Density-based clustering. *WIREs Data Mining and Knowledge Discovery*, *1*(3), 231–240. https://doi.org/10.1002/widm.30

Lei, B. (2019). A DBSCAN based Algorithm for Ship Spot Area Detection in AIS Trajectory Data. *MATEC Web of Conferences*, *291*, 01008. https://doi.org/10.1051/matecconf/201929101008

Li, D., Wang, S., & Li, D. (2015). *Spatial Data Mining*. https://doi.org/10.1007/978-3-662-48538-5

Mangopo, M., & Suprayitno, H. (2020). Karakteristik Variasi Panjang Perjalanan dan Volume Perjalanan Taksi Bluebird di Kota Surabaya Berbasis Jam. *Jurnal Aplikasi Teknik Sipil*, 6.

Maulana, A., Solichin, A., & Syafrullah, M. (2018). Penerapan Metode Haversine Pada Sistem Informasi Geografis Untuk Penentuan Lokasi Pembangunan Menara Telekomunikasi Pada Kota Tangerang. *Indonesian Journal on Software Engineering (IJSE)*, *4*(1). https://doi.org/10.31294/ijse.v4i1.6294

Nidhra, S. (2012). Black Box and White Box Testing Techniques—A Literature Review. *International Journal of Embedded Systems and Applications*, *2*, 29–50. https://doi.org/10.5121/ijesa.2012.2204

Pakuani, K. W., & Kurniawan, R. (2021). Kajian Penentuan Nilai Epsilon Optimal Pada Algoritma DMDBSCAN Dan Pemetaan Daerah Rawan Gempa Bumi Di Indonesia Tahun 2014-2020. *Seminar Nasional Official Statistics*, *2021*(1), 991–1000. https://doi.org/10.34123/semnasoffstat.v2021i1.847

Prikitiew, S. (n.d.). *BAB 1 PENGERTIAN DATA MINING DAN FUNGSI-FUNGSI DATA MINING*. Retrieved from https://www.academia.edu/7414635/BAB_1_PENGERTIAN_DATA_MINING_DAN_FUNGSI_FUNGSI_DATA_MINING

Qu, Z., Wang, X., Song, X., Pan, Z., & Li, H. (2019). Location Optimization for Urban Taxi Stands Based on Taxi GPS Trajectory Big Data. *IEEE Access*, *7*, 62273–62283. https://doi.org/10.1109/ACCESS.2019.2916342

Ramadhan, N. G., Nur, Y. S. R., & Adhinata, F. D. (2022). Pendekatan Deep Learning Untuk Prediksi Durasi Perjalanan. *Teknika*, *11*(2), 85–89. https://doi.org/10.34148/teknika.v11i2.460

Ride-hailing & Taxi—Worldwide | Statista Market Forecast. (2022, December). Retrieved March 14, 2023, from Statista website: https://www.statista.com/outlook/mmo/shared-mobility/shared-rides/ride-hailing-taxi/worldwide

Rivai, R. (2020). *Identifikasi Perilaku Penggunaan Dan Persepsi Pengguna Tentang Layanan Pemesanan Dan Pengiriman Makanan Dengan Transportasi Online* (Other, Universitas Komputer Indonesia). Universitas Komputer Indonesia. https://doi.org/10/BAB%20IV%20-%20Unikom%20-%20Riyaldi%20Rivai%20-%2010615004.pdf

Rizan, M., Fadillah, E., & P, A. K. R. (2015). INFLUENCE OF SERVICE QUALITY AND FARE TOWARD CUSTOMER SATISFACTION AND ITS IMPACT ON CUSTOMER LOYALTY OF EXPRESS TAXI IN JAKARTA. *JRMSI - Jurnal Riset Manajemen Sains Indonesia*, *6*(2), 618.

Rodrigues, F., Markou, I., & Pereira, F. C. (2019). Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach. *Information Fusion*, *49*, 120–129. https://doi.org/10.1016/j.inffus.2018.07.007

Rokach, L., & Maimon, O. (2005). Clustering Methods. In O. Maimon & L. Rokach (Eds.), *Data Mining and Knowledge Discovery Handbook* (pp. 321–352). Boston, MA: Springer US. https://doi.org/10.1007/0-387-25465-X_15

Rossi, A., Barlacchi, G., Bianchini, M., & Lepri, B. (2020). Modelling Taxi Drivers' Behaviour for the Next Destination Prediction. *IEEE Transactions on Intelligent Transportation Systems*, *21*(7), 2980–2989. https://doi.org/10.1109/TITS.2019.2922002

Safitri, D., Wuryandari, T., & Rahmawati, R. (2017). METODE DBSCAN UNTUK PENGELOMPOKAN KABUPATEN/KOTA DI PROVINSI JAWA TENGAH BERDASARKAN PRODUKSI PADI SAWAH DAN PADI LADANG. *Jurnal Statistika Universitas Muhammadiyah Semarang*, *5*(1). https://doi.org/10.26714/jsunimus.5.1.2017.%p

Safitri, N., & Pramudita, R. (2018). Pengujian Black Box Menggunakan Metode Cause Effect Relationship Testing. *INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS : Journal of Information System*, *3*(1), 101–110.

Selvaraj, S., & Sabarish, B. (2021). Analysis of distance measures in spatial trajectory data clustering. *IOP Conference Series: Materials Science and Engineering*, *1085*, 012021. https://doi.org/10.1088/1757-899X/1085/1/012021

Shahapure, K. R., & Nicholas, C. (2020). Cluster Quality Analysis Using Silhouette Score. *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, 747–748. https://doi.org/10.1109/DSAA49011.2020.00096

Sklearn.metrics.pairwise.haversine_distances. (n.d.). Retrieved March 25, 2023, from Scikit-learn website: https://scikit-learn/stable/modules/generated/sklearn.metrics.pairwise.haversine_distances.html

Streamlit Docs. (n.d.). Retrieved April 8, 2023, from https://docs.streamlit.io/

Tanujaya, C. (2017). PERANCANGAN STANDART OPERATIONAL PROCEDURE PRODUKSI PADA PERUSAHAAN COFFEEIN. *Jurnal Performa : Jurnal Manajemen Dan Start-up Bisnis*, *2*(1), 90–95. https://doi.org/10.37715/jp.v2i1.441

UITP. (2020). *GLOBAL TAXI BENCHMARKING STUDY 2019* (p. 6). Retrieved from https://cms.uitp.org/wp/wp-content/uploads/2020/11/Statistics-Brief-TAxi-Benchmarking_NOV2020-web.pdf

Ulak, M. B., Yazici, A., & Aljarrah, M. (2020). Value of convenience for taxi trips in New York City. *Transportation Research Part A: Policy and Practice*, *142*, 85–100. https://doi.org/10.1016/j.tra.2020.10.016

Upadhyay, A. (2015, February 11). Haversine formula—Calculate geographic distance on earth. Retrieved April 7, 2023, from https://www.igismap.com/haversine-formula-calculate-geographic-distance-earth/

Wang, X., Zhang, H., Wang, L., & Ning, Z. (2018). A Demand-Supply Oriented Taxi Recommendation System for Vehicular Social Networks. *IEEE Access*, *6*, 41529–41538. https://doi.org/10.1109/ACCESS.2018.2857002

Widi Hastomo, Nur Aini, Adhitio Satyo Bayangkari Karno, & L.M. Rasdi Rere. (2022). Metode Pembelajaran Mesin untuk Memprediksi Emisi Manure Management. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, *11*(2), 131–139. https://doi.org/10.22146/jnteti.v11i2.2586

Wong, R. C. P., Szeto, W. Y., & Wong, S. C. (2014). Bi-level decisions of vacant taxi drivers traveling towards taxi stands in customer-search: Modeling methodology and policy implications. *Transport Policy*, *33*, 73–81. https://doi.org/10.1016/j.tranpol.2014.02.011

Xu, J., Rahmatizadeh, R., Bölöni, L., & Turgut, D. (2018). Real-Time Prediction of Taxi Demand Using Recurrent Neural Networks. *IEEE Transactions on Intelligent Transportation Systems*, *19*(8), 2572–2581. https://doi.org/10.1109/TITS.2017.2755684

Zeitouni, K. (2000, January 1). *A survey of spatial data mining methods databases and statistics point of views.* 491.

Zhou, D., Hong, R., & Xia, J. (2018). *Identification of Taxi Pick-Up and Drop-Off Hotspots Using the Density-Based Spatial Clustering Method* (p. 204). https://doi.org/10.1061/9780784480915.020

Zhou, Z., Yu, J., Guo, Z., & Liu, Y. (2018). Visual exploration of urban functions via spatio-temporal taxi OD data. *Journal of Visual Languages & Computing*, *48*, 169–177. https://doi.org/10.1016/j.jvlc.2018.08.009