

OPTIMASI RUTE BUS: Menggunakan Breadth-First Search (BFS) dan K-Nearest Neighbor (KNN)



Dosen Pengampu
Dr. Elly Matul Imah, M.Kom
Mata Kuliah
“Kecerdasan Artifisial”

DISUSUN OLEH KELOMPOK 5

Anggota:

- 1. Melinda Venta Siburian (23031554124)**
- 2. Siti Muftiah Ummatun (23031554162)**
- 3. Nurma Aulia Azizah (23031554183)**

PROGRAM STUDI SAINS DATA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
TAHUN 2025-2026

Abstract

Aplikasi optimasi rute bus Surabaya ini dirancang untuk memberikan solusi cerdas dalam pengelolaan transportasi publik, khususnya untuk Suroboyo Bus. Aplikasi ini mengimplementasikan algoritma *Breadth First Search (BFS)*, yang memungkinkan perhitungan rute perjalanan bus yang paling efisien dengan meminimalkan jarak dan waktu tempuh antar halte. Algoritma BFS ini berfungsi untuk menemukan rute terbaik antara titik awal dan tujuan bus berdasarkan kondisi lalu lintas. Hal ini memudahkan pengemudi bus untuk memilih rute bus yang memakan waktu dan jarak yang optimal. Aplikasi ini juga dilengkapi dengan fitur prediksi kedatangan bus di halte, yang menggunakan algoritma *K-Nearest Neighbor (KNN)*. Algoritma KNN ini menganalisis data kemacetan lalu lintas untuk memprediksi waktu kedatangan bus, sehingga penumpang dapat lebih akurat memperkirakan waktu kedatangan bus yang ditunggu. Fitur utama lainnya dalam aplikasi ini adalah tampilan informasi waktu tempuh bus, waktu keberangkatan, dan waktu kedatangan di setiap halte. Dengan informasi tersebut, penumpang dapat merencanakan perjalanan mereka dengan lebih baik, mengurangi waktu tunggu, dan memastikan perjalanan yang lebih nyaman dan terorganisir.

Keywords: Breadth-first Search (BFS), KNN, Suroboyo Bus

1. Latar Belakang

Kota Surabaya, sebagai salah satu kota besar dan penting di Indonesia, merupakan pusat aktivitas ekonomi, pendidikan, dan pariwisata. Setiap harinya, ribuan orang datang ke Surabaya untuk bekerja, transit, atau berwisata. Namun, kemacetan lalu lintas yang sering terjadi menjadi masalah besar akibat banyaknya kendaraan pribadi yang beroperasi di kota ini. Salah satu faktor utama yang menyebabkan tingginya penggunaan kendaraan pribadi adalah terbatasnya kualitas dan kuantitas transportasi umum yang tersedia.

Transportasi umum di Surabaya masih terbatas pada layanan untuk perjalanan pulang-pergi ke sekolah atau kantor, dan belum mampu menjangkau seluruh area kota dengan rute yang efisien. Hal ini membuat banyak masyarakat, baik warga lokal maupun wisatawan, memilih kendaraan pribadi sebagai alternatif, yang pada gilirannya memperburuk kemacetan.

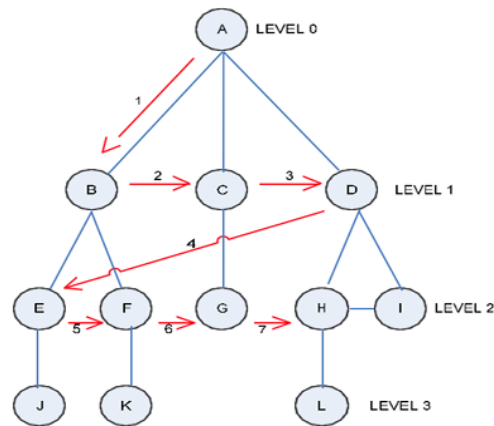
Untuk mengatasi masalah ini, dibutuhkan sistem transportasi publik yang lebih efisien, salah satunya dengan mengoptimalkan rute bus yang ada di Surabaya. Melalui aplikasi yang mengimplementasikan algoritma *Breadth First Search (BFS)* untuk rute perjalanan dan *K-Nearest Neighbor (KNN)* untuk memprediksi kemacetan lalu lintas, masyarakat dapat merencanakan perjalanan menggunakan transportasi umum dengan lebih efisien. Aplikasi ini bertujuan untuk memberikan rekomendasi rute perjalanan yang optimal, menampilkan informasi waktu tempuh, waktu keberangkatan, dan kedatangan bus secara akurat, serta mempermudah peralihan masyarakat untuk lebih memilih transportasi umum daripada kendaraan pribadi. Dengan adanya solusi ini, diharapkan dapat mengurangi kemacetan di Kota Surabaya, meningkatkan kenyamanan bagi masyarakat lokal dan wisatawan, serta mendukung perencanaan kota yang lebih baik dan lebih terintegrasi.

2. LANDASAN TEORI

2.1. Optimasi Secara istilah, optimasi adalah disiplin matematis yang menyangkut proses untuk menemukan sesuatu yang ekstrim (minimum ataupun maksimum) dari angka, fungsi, atau sistem. Para filsuf dan matematikawan kuno di masa lalu menciptakan pondasinya dengan mendefinisikan bagaimana nilai yang optimal dari beberapa permasalahan seperti angka, bentuk geometri optik, fisika, astronomi, kualitas kehidupan manusia dan lain-lainnya. Sedang menurut KBBI, optimasi (optimalisasi) berasal dari kata dasar optimal yang berarti terbaik, tertinggi, paling menguntungkan, menjadikan paling baik, menjadikan paling tinggi, pengoptimalan proses, cara, perbuatan mengoptimalkan (menjadikan paling baik, paling tinggi, dan sebagainya)]. Sedang menurut kamus Oxford, optimasi adalah “The action of making the best or most effective use of a situation or resource.” Jika diterjemahkan dalam Bahasa Indonesia adalah “Sebuah aksi untuk mendapatkan yang terbaik atau paling efektif dari situasi atau sumber daya yang dimiliki”.

2.2. Praproses Data Tahap pra proses data merupakan proses untuk mempersiapkan data mentah agar bisa digunakan untuk proses selanjutnya. Pada umumnya, pra proses data dilakukan dengan cara mengeliminasi data yang tidak sesuai atau mengubah data menjadi bentuk yang lebih mudah diproses oleh sistem. Proses ini diperlukan karena data mentah masih belum siap digunakan untuk proses pengolahan data karena data mentah masih memiliki beberapa kekurangan. Salah satu cara untuk mengubah data mentah menjadi data yang berkualitas adalah dengan melakukan transformasi data.

2.3. Breadth-First Search (BFS) Breadth-first search (BFS) adalah algoritma pencarian graf yang paling sederhana dan merupakan dasar dari beberapa algoritma yang lebih maju. BFS sangat berguna dalam menemukan jalur terpendek pada graf yang tidak memiliki weight atau biaya. Dengan graf $G = (V, E)$ di mana V adalah vertex atau node dan E adalah edges atau jalur dan node awal yaitu s , BFS akan secara sistematis melintasi jalur G untuk menemukan setiap node yang dapat dijangkau dari node s . Algoritma ini menghitung jarak (jumlah jalur terkecil) dari node s ke setiap node yang dapat dicapai dan menghasilkan “breadth-first tree” dengan akar s yang semua jalurnya dapat dijangkau. Setiap jalur v yang dapat dicapai dari *nodes* pada breadth-first tree akan menciptakan rute terpendek dari s ke v , yaitu rute yang memiliki jumlah jalur terkecil. Algoritma ini dapat digunakan dalam graf berarah maupun graf tidak berarah. Untuk mempermudah proses traversal, algoritma BFS menggunakan struktur data queue dalam menyimpan dan menandai node yang telah dikunjungi. Hal ini akan terus berlangsung hingga semua node yang berdekatan telah ditandai. Metode queue yang digunakan ialah first in first out (FIFO).



2.4. Algoritma K-Nearest Neighbor (KNN) adalah metode klasifikasi dalam pembelajaran mesin yang termasuk dalam kategori *supervised learning*, di mana klasifikasi data dilakukan berdasarkan mayoritas kategori dari tetangga terdekatnya. Pada algoritma KNN, data yang ingin diklasifikasikan dibandingkan dengan data pelatihan menggunakan perhitungan jarak, seperti jarak Euclidean atau Manhattan, untuk menentukan seberapa mirip data tersebut dengan data pelatihan. Hasil klasifikasi diperoleh dengan memilih kelas mayoritas dari K tetangga terdekat. Kelebihan dari algoritma KNN adalah kemampuan untuk menangani jumlah data yang besar tanpa memerlukan model pelatihan yang kompleks. Algoritma ini juga mudah diimplementasikan dan cukup fleksibel dalam menghadapi berbagai macam data.

Namun, KNN juga memiliki beberapa kelemahan. Salah satunya adalah kebutuhan untuk menentukan jumlah tetangga terdekat, yang disimbolkan dengan nilai parameter K. Pemilihan nilai K yang tepat dapat mempengaruhi akurasi hasil klasifikasi. Selain itu, algoritma KNN memerlukan perhitungan jarak antara data yang diuji dengan seluruh data pelatihan, yang dapat menghasilkan biaya komputasi yang tinggi, terutama untuk dataset besar. Selain itu, algoritma ini juga sensitif terhadap pemilihan jenis jarak dan atribut yang digunakan untuk perhitungan, yang dapat mempengaruhi kualitas hasil klasifikasi.

$$d_i = \sqrt{\sum_{i=1}^p (x_1 - x_2)^2} \quad (1)$$

Keterangan :

d = jarak

i = variabel data

p = dimensi data

x_1 = sampel data

x_2 = data uji

Proses perhitungan nilai akurasi dapat dilakukan dengan menggunakan persamaan 2.

$$akurasi = \frac{\text{jumlah klasifikasi benar}}{\text{jumlah data uji}} \times 100\% \quad (2)$$

3. Desain Sistem dan Model

Sistem optimasi rute bus ini dirancang untuk meningkatkan efisiensi operasional transportasi umum di Kota Surabaya dengan mengintegrasikan teknologi algoritma untuk menentukan rute optimal dan memprediksi waktu kedatangan bus berdasarkan kondisi lalu lintas. Desain sistem ini terdiri dari beberapa komponen utama, yaitu:

- Pada lapisan data, sistem memanfaatkan data peta rute bus, data historis lalu lintas, serta data real-time kondisi lalu lintas yang disediakan dalam format CSV. Proses *preprocessing* dilakukan untuk mengintegrasikan dan memformat data, seperti penggabungan kolom tanggal dan waktu, serta klasifikasi kondisi lalu lintas (rendah, normal, tinggi, berat) menggunakan *situation mapping*.
- Lapisan pemrosesan menggunakan algoritma BFS untuk menghitung rute optimal antar halte berdasarkan bobot jarak atau waktu tempuh. Untuk memprediksi kemacetan lalu lintas, algoritma KNN diterapkan dengan fitur jam dan menit keberangkatan. Proses prediksi ini dilakukan dengan menghitung jarak Euclidean antara data input dan data *training*, kemudian menentukan kategori lalu lintas berdasarkan mayoritas dari tetangga terdekat (nilai K).
- Model jadwal dinamis dibuat dengan menghitung waktu tempuh antar halte menggunakan data rute yang tersedia, kemudian dikalibrasi berdasarkan prediksi kondisi lalu lintas. Jika lalu lintas ringan, waktu tempuh dikurangi hingga 20%; jika berat, waktu tempuh dapat bertambah hingga 50%. Hasil perhitungan ini ditampilkan dalam bentuk jadwal keberangkatan, tujuan, dan waktu kedatangan, yang disesuaikan secara dinamis.
- *Interface* pengguna memanfaatkan pustaka seperti Streamlit untuk memvisualisasikan aplikasi. Pengguna dapat memilih halte keberangkatan, halte tujuan, serta waktu keberangkatan, kemudian melihat jadwal yang dihasilkan secara interaktif. Dengan desain ini, sistem mendukung perencanaan perjalanan yang lebih efisien, akurat, dan responsif terhadap dinamika lalu lintas kota.

4. Implementasi dan Hasil Performa Model

Proyek ini mengimplementasikan dua komponen utama untuk mengoptimalkan rute perjalanan bus dan memprediksi waktu kedatangan berdasarkan kondisi lalu lintas di Kota Surabaya. Berikut adalah penjelasan detail implementasi dan performa model yang digunakan:

- Implementasi Model KNN untuk Prediksi Lalu Lintas

Model KNN (*K-Nearest Neighbor*) digunakan untuk mengklasifikasikan kondisi lalu lintas berdasarkan data historis. Data lalu lintas yang tersedia diproses terlebih dahulu untuk menggabungkan kolom tanggal dan waktu menjadi format *datetime* yang konsisten.

```
# KNN implementation
def knn_predict(X_train, y_train, X_test, k=3):
    def euclidean_distance(x1, x2):
        return np.sqrt(np.sum((x1 - x2) ** 2))

    predictions = []
    for test_point in X_test:
        distances = [euclidean_distance(test_point, train_point) for train_point in X_train]
        k_indices = np.argsort(distances)[:k]
        k_nearest_labels = [y_train[i] for i in k_indices]
        most_common = max(set(k_nearest_labels), key=k_nearest_labels.count)
        predictions.append(most_common)
    return predictions
```

Implementasi KNN pada kode

Fitur utama yang digunakan dalam prediksi ini adalah jam dan menit keberangkatan, sedangkan target prediksi adalah situasi lalu lintas yang dikategorikan menjadi empat level, dengan pemetaan berikut: 0: "low", 1: "normal", 2: "high", 3: "heavy".

Dataset dibagi menjadi dua bagian: Data Training (80%) digunakan untuk melatih model dan data Testing (20%) digunakan untuk mengevaluasi performa model. Model ini menggunakan nilai parameter $k=13$, yang dipilih berdasarkan pengujian terhadap dataset. Prediksi dilakukan dengan menghitung jarak Euclidean antara setiap titik data baru (query) dan data pelatihan, kemudian menentukan kategori lalu lintas berdasarkan mayoritas kategori dari 13 tetangga terdekat.

Hasil evaluasi menunjukkan bahwa model KNN mencapai tingkat akurasi sebesar 69% pada data uji. Meskipun akurasi ini masih dapat ditingkatkan, model sudah cukup representatif untuk memberikan estimasi lalu lintas yang membantu dalam penghitungan jadwal bus.

- Implementasi BFS untuk Optimasi Route

Algoritma BFS digunakan untuk menemukan rute optimal dari halte keberangkatan ke halte tujuan berdasarkan konektivitas antar halte. BFS bekerja dengan eksplorasi tingkat demi tingkat dari graf halte, memastikan rute yang ditemukan adalah yang paling pendek dalam hal jumlah pemberhentian. Pada aplikasi ini, halte bus dimodelkan sebagai simpul dalam graf berarah, dan hubungan antar halte (serta waktu tempuhnya) dimodelkan sebagai sisi dengan bobot. Saat pengguna memilih halte keberangkatan dan halte tujuan, BFS akan menelusuri graf untuk menemukan semua jalur yang mungkin dan memilih jalur dengan waktu tempuh terpendek. Hasil penerapan algoritma BFS menunjukkan bahwa algoritma ini efisien dalam menemukan rute optimal, terutama pada skenario dengan jumlah halte yang besar. Performa BFS sangat baik karena algoritma ini bekerja secara sistematis dengan memanfaatkan semua hubungan langsung sebelum menjelajahi tingkat berikutnya.

```

# Fungsi Algoritma Genetika untuk menemukan rute optimal
def bfs_shortest_path(G, start, end):
    if start not in G.nodes() or end not in G.nodes():
        raise ValueError("Halte keberangkatan atau tujuan tidak valid")

    queue = [(start, [start])]
    while queue:
        current_node, path = queue.pop(0)
        if current_node == end:
            total_time = calculate_route_time(path, G)
            return path, total_time
        for neighbor in G.neighbors(current_node):
            if neighbor not in path:
                queue.append((neighbor, path + [neighbor]))
    return [], float('inf')

```

Implementasi BFS pada kode

- Hasil performa model

Model prediksi lalu lintas dengan KNN memiliki akurasi sebesar 69% pada data uji, menggunakan parameter $k=13$, dengan data dibagi menjadi 80% data *train* dan 20% data *test*. KNN memanfaatkan fitur waktu (jam dan menit) untuk memprediksi kondisi lalu lintas yang diklasifikasikan menjadi empat kategori: low (0), normal (1), high (2), dan heavy (3). Tingkat akurasi ini menunjukkan bahwa model cukup andal, meskipun ada ruang untuk perbaikan, terutama dalam menangani kategori lalu lintas yang jarang terjadi.

Sementara itu, BFS bekerja dengan efisien dan cepat, serta diintegrasikan dengan hasil prediksi lalu lintas dari KNN untuk memberikan waktu perjalanan yang lebih realistis. Waktu tempuh antar halte disesuaikan berdasarkan prediksi kondisi lalu lintas: kategori low mempercepat waktu tempuh sebesar 20%, kategori normal tidak mengubah waktu, kategori high menambah waktu sebesar 30%, dan kategori heavy menambah waktu sebesar 50%.

Secara keseluruhan, sistem ini mampu memberikan rute optimal serta jadwal perjalanan bus yang adaptif terhadap kondisi lalu lintas. Kombinasi KNN dan BFS menciptakan solusi yang efektif dalam mendukung pengelolaan transportasi umum di Kota Surabaya. Namun, akurasi prediksi lalu lintas yang sebesar 69% masih menunjukkan kebutuhan untuk pengembangan lebih lanjut guna meningkatkan performa sistem, khususnya dalam menangani data lalu lintas yang dinamis dan kompleks.

5. Deployment

Proyek ini diimplementasikan sebagai aplikasi berbasis web dengan menggunakan Streamlit, sebuah framework Python yang dirancang untuk membangun *interface* pengguna secara cepat dan interaktif. Proses deployment dilakukan dengan tujuan agar aplikasi dapat diakses secara publik oleh pengguna, sehingga memungkinkan siapa saja untuk memanfaatkan

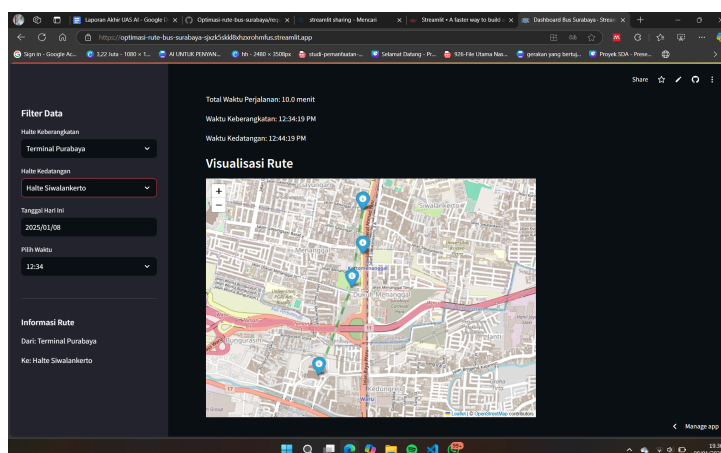
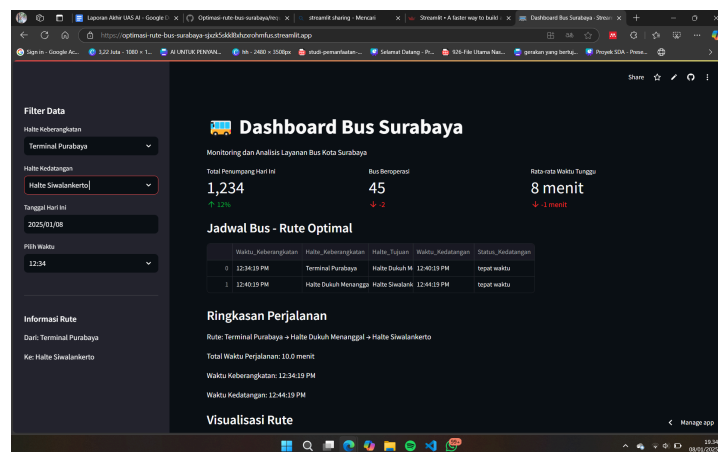
fitur-fitur yang telah dikembangkan, seperti perencanaan rute bus, estimasi waktu perjalanan, dan prediksi kondisi lalu lintas.

Aplikasi ini di-host secara publik melalui layanan cloud yang kompatibel dengan Streamlit, sehingga menghasilkan tautan unik yang dapat diakses menggunakan browser tanpa memerlukan instalasi perangkat lunak tambahan di sisi pengguna. Dengan *interface* yang sederhana dan intuitif, pengguna dapat dengan mudah memilih halte keberangkatan, halte tujuan, serta waktu keberangkatan, untuk mendapatkan rekomendasi rute dan jadwal bus yang optimal.

Berikut tautan deployment nya :

<https://optimasi-rute-bus-surabaya-sjxzk5skkl8xhzhxrohmfus.streamlit.app/>

Berikut tangkapan layar dari *user interface* nya :



6. Kesimpulan

Kesimpulan dari pengembangan aplikasi ini menunjukkan bahwa aplikasi optimasi rute bus berhasil dikembangkan dengan menggunakan algoritma Breadth-First Search (BFS) untuk menentukan rute optimal antar halte dan algoritma K-Nearest Neighbors (KNN) untuk menganalisis serta mengelompokkan situasi kemacetan lalu lintas.

Aplikasi ini berhasil menentukan rute optimal berdasarkan jarak geografis serta menghasilkan jadwal perjalanan bus yang mempertimbangkan situasi lalu lintas. Dengan antarmuka pengguna yang interaktif dan peta visual yang informatif, aplikasi ini memberikan pengalaman pengguna yang intuitif dan mempermudah perencanaan perjalanan.

Secara keseluruhan, aplikasi ini dapat memberikan rekomendasi rute dan jadwal perjalanan yang optimal dan informatif, mendukung kebutuhan pengguna dalam meningkatkan efisiensi perjalanan menggunakan transportasi bus di Kota Surabaya. Hasil performa model menunjukkan bahwa kombinasi algoritma yang digunakan dapat berjalan dengan baik dalam skenario yang diterapkan. Namun, aplikasi ini masih memiliki beberapa keterbatasan, seperti belum terintegrasinya data lalu lintas secara real-time dan tidak mempertimbangkan faktor lain, seperti kepadatan penumpang atau efisiensi biaya perjalanan.

7. Referensi

Wibowo, D. A., & Abdiansah, A., "Implementasi algoritma Fisher-Yates Shuffle dalam pengacakan urutan soal ujian berbasis web," *Jurnal Manajemen Informatika dan Teknik Komputer (MITRANS)*, vol. 6, no. 2, pp. 158-165, Universitas Negeri Surabaya, Surabaya, 2022.

Maruli Tua Harianja, "Komparasi Kinerja Algoritma BFS, Dijkstra, Greedy BFS, dan A* dalam Melakukan Pathfinding," Artikel Penelitian, Fakultas Teknologi Informasi, Universitas Katolik Indonesia Atma Jaya, Jakarta, 2020.

Muhammad Iqbal Imaduddin, "Penerapan Algoritma Genetika Hyper-Heuristic untuk Optimasi Rencana Perjalanan Menggunakan Angkutan Umum sebagai Sebuah Orienteering Problem: Studi Kasus Bis Kota di Kota Surabaya," Skripsi Sarjana, Departemen Sistem Informasi, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember, Surabaya, 2018.

N. R. Aprilia and Yahfizham, "Analisis Studi Literatur Eksplorasi Algoritma Pemrograman Mahasiswa Pendidikan Matematika di Era Society 5.0 dengan Bibliometrik," *Jurnal Arjuna: Publikasi Ilmu Pendidikan, Bahasa dan Matematika*, vol. 1, no. 5, pp. 211-221, 2023. Available.

8. Lampiran

- Github code : [sitimuftiah162/Optimasi-rute-bus-surabaya](https://github.com/sitimuftiah162/Optimasi-rute-bus-surabaya)
- File google colab model KNN :
https://colab.research.google.com/drive/1jFE_46gaaOSiL7cmQftZ3u0Xt8qO5Rge?usp=sharing
- Kontribusi anggota:

- Melinda : mencari dan memproses dataset, membuat kode BFS dan model KNN, menyempurnakan kode, mengerjakan laporan,
- Siti : mencari dan memproses dataset, membuat tampilan UI di streamlit, melakukan deploy streamlit, mengerjakan laporan
- Nurma : mencari dan memproses dataset, membuat tampilan UI di streamlit, mengerjakan laporan