

Manual Book

Website Optimizer SMA

BRT Transjakarta

Tigo Ilhami Fasyah

535220235

DAFTAR ISI

DAFTAR ISI	2
DAFTAR GAMBAR	3
BAB I PENDAHULUAN	4
1.1 Tujuan Aplikasi.....	4
1.2 Target Pengguna	4
1.3 Istilah Penting	5
BAB II MEMULAI APLIKASI	6
2.1 Kebutuhan Sistem.....	6
2.2 Mengakses Aplikasi	6
BAB III ANTARMUKA PENGGUNA	7
3.1 Tampilan Utama.....	7
3.2 Komponen Input.....	9
3.3 Komponen Output.....	11
BAB IV ALUR PENGGUNAAN APLIKASI	15
4.1 Langkah-langkah Pencarian Rute	15
4.2 Membaca Hasil Rekomendasi	15
4.3 Menggunakan Peta Interaktif.....	16
BAB V SKENARIO PENGGUNAAN	17
5.1 Skenario 1: Mencari Rute Seimbang (Blok M – Ancol) dengan MILP	17
5.2 Skenario 2: Mencari Rute Minim Transit (Blok M – Ancol) dengan SMA	18
5.3 Skenario 3: Mencari Rute Tercepat (Blok M – Ancol) dengan SMA	19
BAB VII PENUTUP	22
6.1 Kesimpulan	22

DAFTAR GAMBAR

Gambar 3.1 Halaman Utama Website Optimizer Transjakarta	7
Gambar 3.2 Halaman Visualisasi Rute	8
Gambar 3.3 Komponen Input Dari (Halte Asal)	9
Gambar 3.4 Error Messages Kondisi Halte Tujuan sama Dengan Halte Asal	9
Gambar 3.5 Error Messages Kondisi Halte yang Tidak Sesuai.....	10
Gambar 3.6 Komponen Input Preferensi Rute	10
Gambar 3.7 Komponen Input Jam Keberangkatan.....	11
Gambar 3.8 Komponen Output Ringkasan Perjalanan.....	13
Gambar 3.9 Komponen Output Visualisasi Rute Rekomendasi.....	14
Gambar 5.1 Skenario Pengujian Preferensi Seimbang dengan MILP	17
Gambar 5.2 Visualisasi Skenario Pengujian Preferensi Seimbang dengan MILP	18
Gambar 5.3 Skenario Pengujian Preferensi Minim Transit dengan SMA.....	19
Gambar 5.4 Visualisasi Skenario Pengujian Preferensi Minim Transit dengan SMA	19
Gambar 5.5 Skenario Pengujian Preferensi Paling Cepat dengan SMA.....	20
Gambar 5.6 Visualisasi kenario Pengujian Preferensi Paling Cepat dengan SMA	21

BAB I

PENDAHULUAN

1.1 Tujuan Aplikasi

Aplikasi Optimasi Rute Transjakarta ini dirancang dengan tujuan utama untuk membantu pengguna menemukan rute perjalanan terbaik menggunakan layanan bus Transjakarta. Berbeda dari aplikasi pencarian rute biasa, aplikasi ini memungkinkan pengguna untuk memilih preferensi perjalanan seperti rute paling efisien, paling cepat, atau dengan transit seminimal mungkin dan membandingkan hasil dari dua metode optimasi yang berbeda yaitu:

- **MILP (Mixed-Integer Linear Programming)**
Mencari solusi rute yang optimal secara matematis.
- **SMA (Slime Mould Algorithm)**
Mencari solusi rute yang mendekati optimal dengan potensi waktu pencarian yang lebih cepat.

Dengan demikian, aplikasi ini bertujuan memberikan rekomendasi rute yang tidak hanya valid, tetapi juga paling sesuai dengan kebutuhan dan prioritas setiap pengguna.

1.2 Target Pengguna

Aplikasi ini ditujukan untuk:

- **Penumpang Transjakarta,** Baik pengguna rutin maupun occasional yang membutuhkan bantuan dalam merencanakan perjalanan di jaringan rute Transjakarta yang kompleks.
- **Pengguna Baru Transjakarta,** Memudahkan mereka yang belum familiar dengan sistem koridor dan transit Transjakarta.
- **Perencana Perjalanan,** Siapa saja yang ingin mencari rute Transjakarta dengan kriteria spesifik (misalnya, menghindari transit yang banyak atau mencari waktu tempuh tercepat).
- **Akademisi/Mahasiswa,** Sebagai alat demonstrasi perbandingan antara metode optimasi eksak (MILP) dan metaheuristik (SMA) dalam kasus nyata.

1.3 Istilah Penting

Berikut adalah beberapa istilah yang sering digunakan dalam aplikasi ini:

- **Halte**

Titik pemberhentian resmi bus Transjakarta tempat penumpang naik dan turun.

- **Koridor**

Jalur atau rute spesifik yang dilayani oleh bus Transjakarta, biasanya ditandai dengan nomor atau kode (misalnya, Koridor 1, Koridor 3F, Koridor 6B).

- **Transit**

Proses perpindahan dari satu koridor ke koridor lain di sebuah halte transit untuk melanjutkan perjalanan.

- **Solver**

Algoritma atau metode yang digunakan untuk menghitung dan menemukan rute optimal. Di aplikasi ini terdapat dua pilihan yaitu MILP sebagai metode matematis eksak, dan SMA sebagai metode metaheuristik.

- **Preferensi Rute**

Merupakan Pilihan pengguna untuk memprioritaskan kriteria tertentu dalam pencarian rute, yaitu:

- a. **Paling Efisien dan Seimbang**

Mencari keseimbangan antara waktu tempuh dan jumlah/kemudahan transit.

- b. **Minim Transit**

Memprioritaskan rute dengan jumlah perpindahan koridor paling sedikit.

- c. **Paling Cepat**

Memprioritaskan rute dengan estimasi total waktu tempuh tersingkat.

BAB II

MEMULAI APLIKASI

2.1 Kebutuhan Sistem

Untuk dapat menjalankan dan menggunakan aplikasi ini di komputer Anda, pastikan Anda memiliki:

- **Perangkat**
Komputer desktop atau laptop.
- **Web Browser Modern**
Disarankan menggunakan versi terbaru dari Google Chrome, Mozilla Firefox, Microsoft Edge, atau Safari.
- **Koneksi Internet**
Meskipun aplikasi berjalan secara lokal dan data rute tersimpan di komputer, koneksi internet tetap dibutuhkan agar latar belakang peta (dari Leaflet.js) dapat dimuat dan ditampilkan dengan benar pada fitur visualisasi rute. Proses pencarian rute itu sendiri tidak memerlukan internet.
- **Untuk Menjalankan Server Lokal**
Python (versi 3.x), Django, dan *library* Python lain yang diperlukan (seperti PuLP, NetworkX, GeoPandas, pytz) harus sudah terinstal di komputer jika ingin menjalankan *server* aplikasi ini sendiri.

2.2 Mengakses Aplikasi

Untuk mengakses aplikasi:

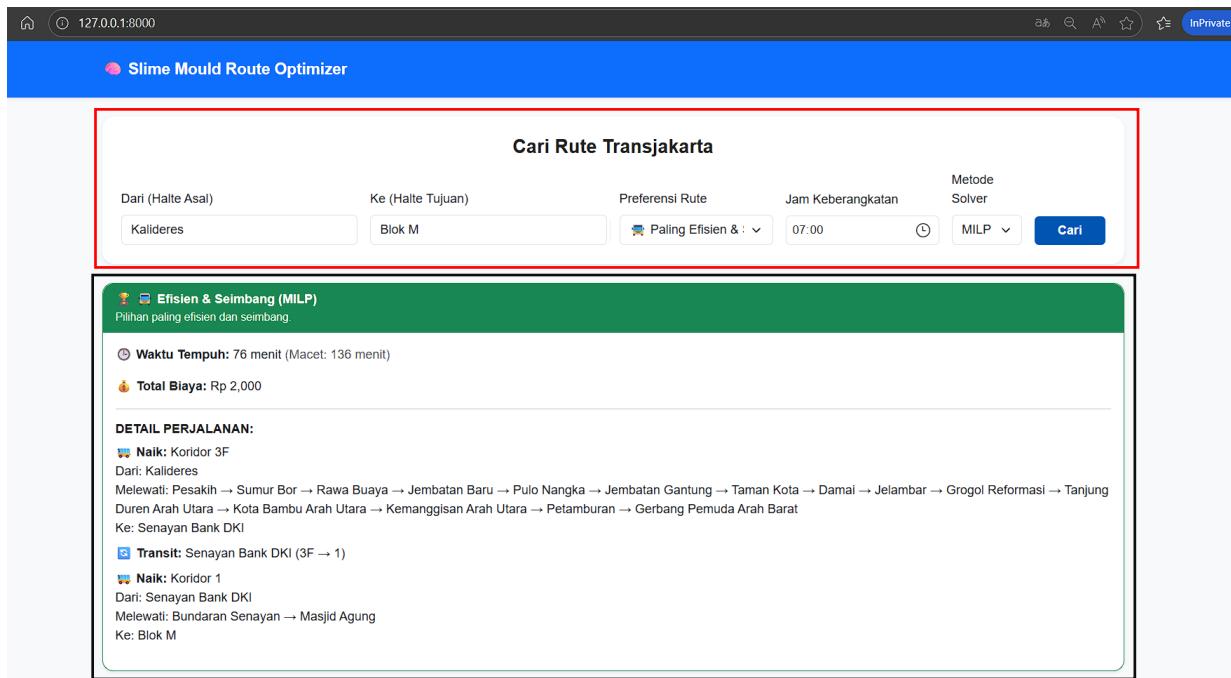
1. Pastikan *server* aplikasi sedang berjalan (dijalankan menggunakan command `python manage.py runserver` di terminal).
2. Buka *web browser*.
3. Masukkan alamat URL berikut pada *address bar*: `http://127.0.0.1:8000/`
4. Tekan Enter. Halaman utama aplikasi akan dimuat.

BAB III

ANTARMUKA PENGGUNA

3.1 Tampilan Utama

Sistem aplikasi web ini dirancang untuk menyediakan antarmuka yang sederhana dan intuitif bagi pengguna dalam mencari rute Transjakarta yang optimal berdasarkan preferensi mereka. Aplikasi ini terdiri dari satu halaman utama yang dinamis, di mana pengguna memasukkan kriteria pencarian dan hasilnya akan ditampilkan di halaman yang sama. Secara garis besar, aplikasi ini memiliki tiga bagian utama yang terlihat pada antarmuka pengguna seperti pada Gambar 3.1 berikut ini:



Gambar 3.1 Halaman Utama Website Optimizer Transjakarta

Keterangan:

a. Form Pencarian Rute

Terletak di bagian atas halaman dan pada Gambar 3.1 pada kotak berwarna merah, bagian ini berisi field input bagi pengguna untuk memasukkan:

- Halte Asal: Titik awal perjalanan (dengan autocomplete).

- Halte Tujuan: Titik akhir perjalanan (dengan autocomplete).
- Preferensi Rute: Pilihan prioritas optimasi (Efisien, Minim Transit, Tercepat).
- Jam Keberangkatan: Waktu rencana keberangkatan (mempengaruhi tarif).
- Metode Solver: Pilihan algoritma optimasi (MILP atau SMA).
- Tombol "Cari": Untuk memulai proses pencarian rute.

b. Panel Hasil Pencarian

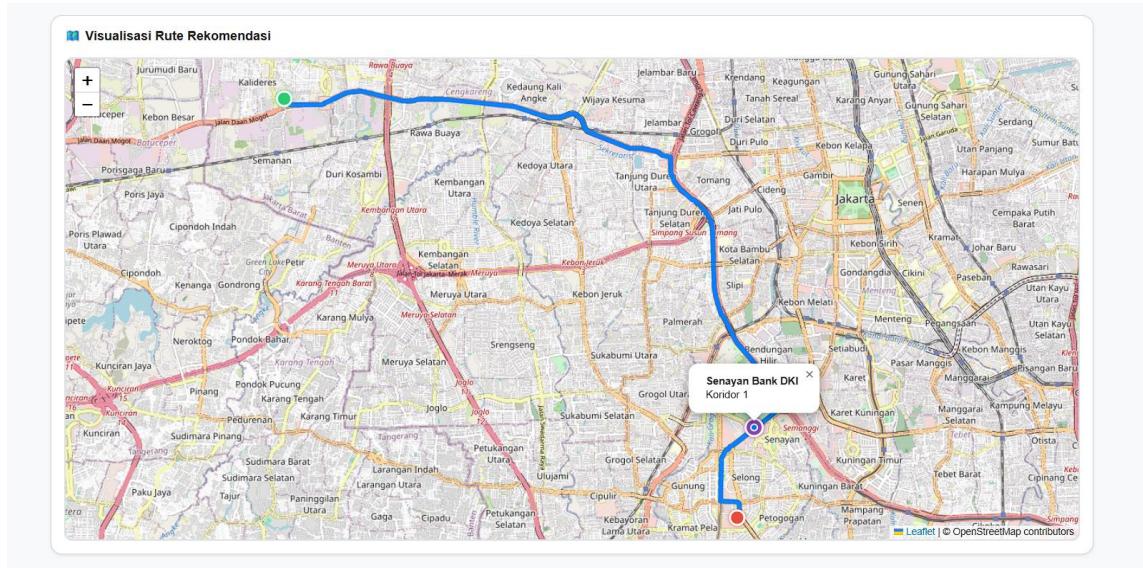
Muncul di bawah form setelah pencarian berhasil yaitu pada Gambar 3.1 pada kotak berwarna hitam. Bagian ini menampilkan:

- Ringkasan Rute: Informasi kunci seperti estimasi waktu tempuh (normal dan saat macet), total biaya, jam berangkat, dan jumlah transit.
- Detail Perjalanan: Langkah-langkah rinci rute yang harus diikuti, termasuk koridor bus, halte naik/turun, halte yang dilewati, dan informasi transit.

c. Visualisasi Peta

Terletak di bawah panel hasil teks, bagian ini menampilkan peta interaktif (menggunakan Leaflet.js) yang menggambarkan jalur rute secara visual, lengkap dengan *marker* untuk halte awal, akhir, dan transit. Hasil visualisasi tersebut dapat dilihat melalui gambar 3.2

Berikut:



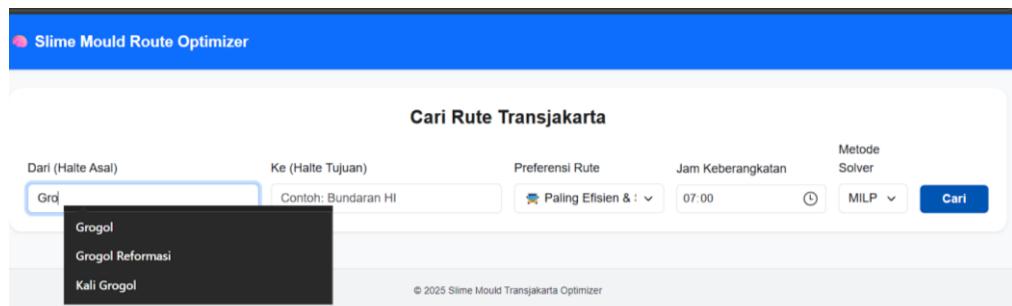
Gambar 3.2 Halaman Visualisasi Rute

3.2 Komponen Input

Bagian ini merupakan area utama bagi pengguna untuk memasukkan detail perjalanan yang diinginkan. Komponen-komponennya adalah sebagai berikut:

- **Dari (Halte Asal)**

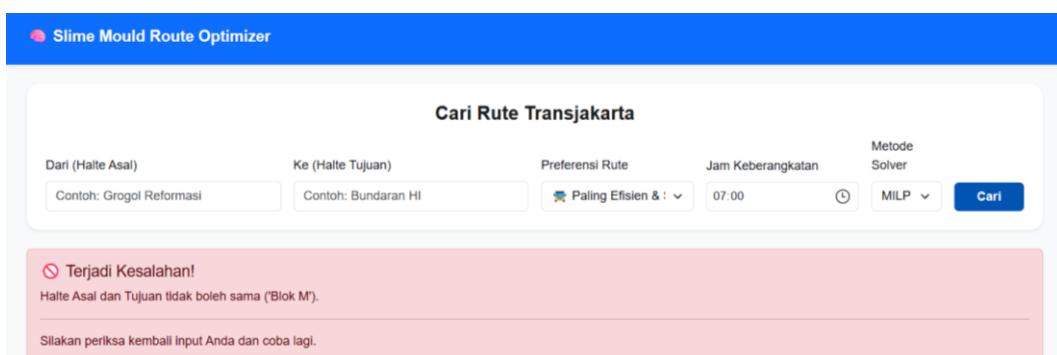
Kotak teks untuk memasukkan nama halte keberangkatan. Saat mulai mengetik (minimal 2 huruf), daftar nama halte yang cocok akan muncul sebagai sugesti (*autocomplete*) untuk membantu memilih halte yang benar. Berikut adalah contoh Input Halte mengetik “Gro” maka akan keluar kemungkinan seperti Grogol Reformasi, Kali Grogol, dan Grogol. Hal tersebut dapat dilihat melalui Gambar 3.3 berikut:



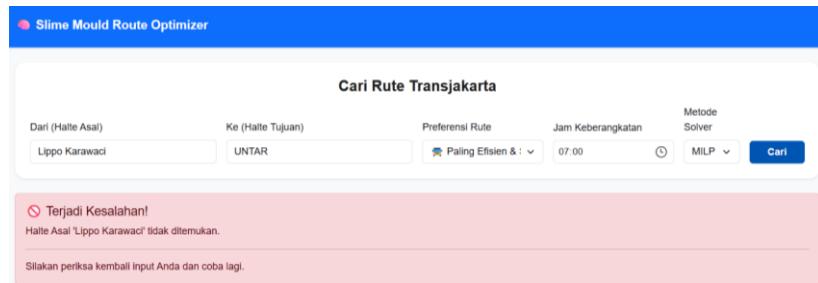
Gambar 3.3 Komponen Input Dari (Halte Asal)

- **Ke (Halte Tujuan)**

Kotak teks untuk memasukkan nama halte tujuan. Fitur *autocomplete* juga tersedia, dan Pastikan nama halte tujuan berbeda dengan halte asal karena jika sama akan ada error messages seperti pada Gambar 3.4 dan juga pastikan halte yang ditulis sesuai dengan halte yang dilalui oleh BRT Transjakarta karena jika tidak akan ada error messages seperti pada Gambar 3.5 berikut:



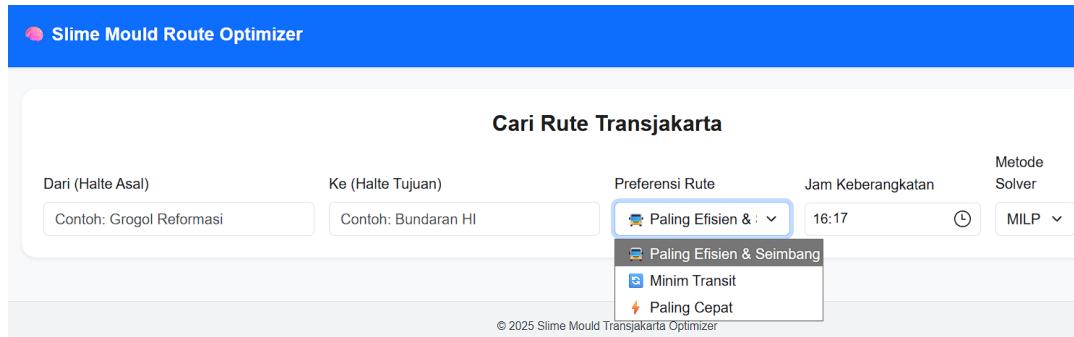
Gambar 3.4 Error Messages Kondisi Halte Tujuan sama Dengan Halte Asal



Gambar 3.5 Error Messages Kondisi Halte yang Tidak Sesuai

- **Preferensi Rute**

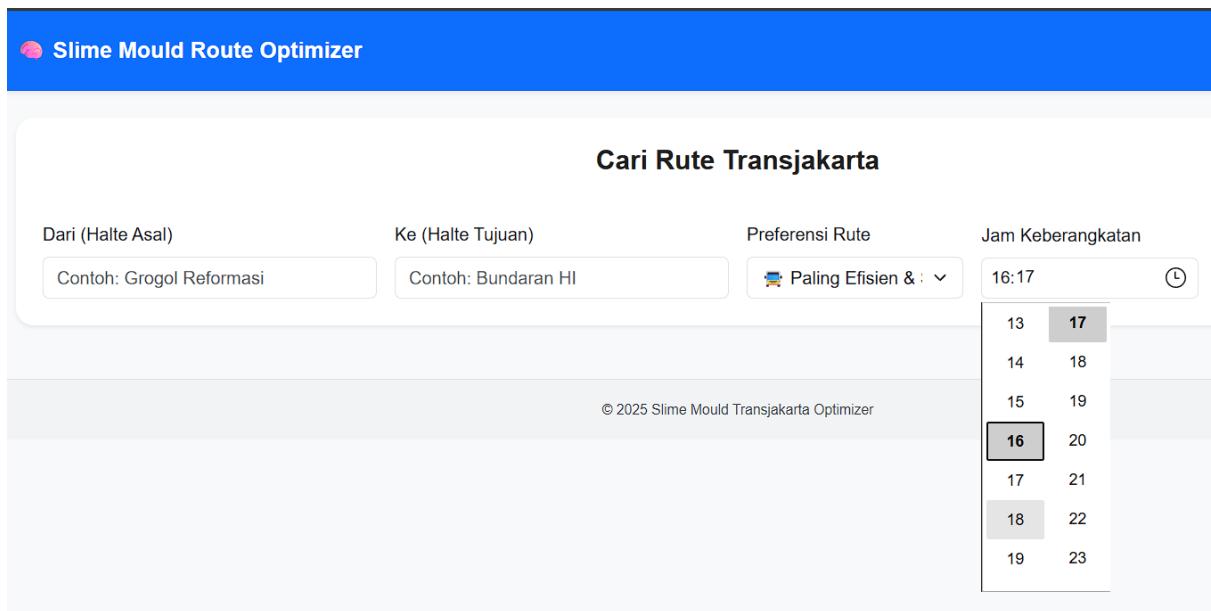
Dropdown untuk memilih prioritas optimasi rute. Pilihan yang tersedia adalah "Paling Efisien & Seimbang" (default), "Minim Transit", dan "Paling Cepat". Pilihan ini akan memengaruhi bobot perhitungan algoritma.



Gambar 3.6 Komponen Input Preferensi Rute

- **Jam Keberangkatan**

Kolom input waktu (HH:MM) untuk menentukan perkiraan waktu mulai perjalanan Anda. Secara *default*, kolom ini akan terisi dengan waktu saat Anda membuka aplikasi, namun Anda dapat mengubahnya. Waktu ini digunakan untuk menentukan tarif dasar, apakah menggunakan tarif ekonomis (Pukul 05.00 – 07.00) atau normal. Kolom Input tersebut dapat dilihat melalui Gambar 3.7 berikut:



Gambar 3.7 Komponen Input Jam Keberangkatan

- **Metode Solver**

Dropdown untuk memilih algoritma optimasi yang akan digunakan. Pilihan yang tersedia adalah "MILP" (metode eksak) dan "SMA" (metode heuristik).

- **Tombol Cari**

Tombol biru di ujung kanan untuk memulai proses pencarian rute setelah semua input diisi.

3.3 Komponen Output

Setelah tombol "Cari" ditekan dan sistem berhasil menemukan rute, hasil rekomendasi akan ditampilkan di bawah form pencarian. Komponen output tersebut mencakup beberapa hal yaitu:

1. Panel Hasil Pencarian (Hasil Rute & Peta)

a. Header Hasil

Menampilkan preferensi rute yang dipilih dan solver yang digunakan (misalnya, Efisien & Seimbang (SMA)).

b. Ringkasan Rute

Menyajikan informasi kunci secara ringkas, yaitu:

- Waktu Tempuh

Estimasi total waktu perjalanan (dalam menit atau jam), seringkali disertai perkiraan saat kondisi macet dalam tanda kurung.

- Total Biaya

Perkiraan tarif dasar Transjakarta berdasarkan jam keberangkatan yang diinput.

- Jam Berangkat

Mengulang jam keberangkatan yang Anda masukkan.

- Jumlah Transit

Menampilkan jumlah perpindahan antar koridor yang diperlukan. Jika 0, akan tertulis "Tidak ada transit".

c. Detail Perjalanan

Memberikan panduan langkah-demi-langkah, yaitu:

- Naik Bus

Menunjukkan koridor bus yang harus dinaiki, halte asal untuk segmen tersebut, halte-halte yang dilewati (via), dan halte tujuan untuk segmen tersebut.

- Transit

Menunjukkan di halte mana Anda perlu transit, serta perpindahan dari koridor asal ke koridor tujuan. Terdapat juga indikator

Secara keseluruhan, panel hasil pencarian ini berisikan tentang output ringkasa perjalanan hasil dari komponen input. Secara lengkapnya kita dapat melihatnya melalui contoh Gambar 3.8 berikut.

The screenshot shows the user interface of the Slime Mould Route Optimizer. At the top, there's a blue header bar with the title "Slime Mould Route Optimizer". Below it is a search form titled "Cari Rute Transjakarta". The search fields are "Dari (Halte Asal)" containing "Kalideres" and "Ke (Halte Tujuan)" containing "Grogol Reformasi". Underneath these are dropdown menus for "Preferensi Rute" (set to "Paling Efisien & Seimbang"), "Jam Keberangkatan" (set to "--:--"), and "Metode Solver" (set to "SMA"). A "Cari" button is located to the right of the search fields.

Below the search form, a green header bar displays the selected preference: "Efisien & Seimbang (SMA)" with the subtext "Pilihan paling efisien dan seimbang".

The main content area contains the following information:

- Waktu Tempuh:** 30 menit (Macet: 54 menit)
- Total Biaya:** Rp 3,500
- Jam Berangkat:** 16:17
- Jumlah Transit:** Tidak ada transit

DETAIL PERJALANAN:

Naik: Koridor 3F
Dari: Kalideres
Melewati: Pesakih → Sumur Bor → Rawa Buaya → Jembatan Baru → Pulo Nangka → Jembatan Gantung → Taman Kota → Damai → Jelambar
Ke: Grogol Reformasi

Gambar 3.8 Komponen Output Ringkasan Perjalanan

2. Visualisasi Peta

a. Peta Interaktif

Di bawah panel teks, terdapat peta (menggunakan Leaflet.js) yang menampilkan jalur rute yang direkomendasikan secara geografis.

b. Jalur Rute

Garis berwarna (umumnya biru atau warna lain) yang mengikuti jalan sesuai koridor yang dilalui.

c. Marker Halte:

- *Marker* khusus (misal, hijau) untuk Halte Asal.
- *Marker* khusus (misal, merah) untuk Halte Tujuan.
- *Marker* khusus (misal, ungu) untuk Halte Transit.
- *Marker* lebih kecil (misal, biru) untuk halte naik/turun di setiap segmen perjalanan.

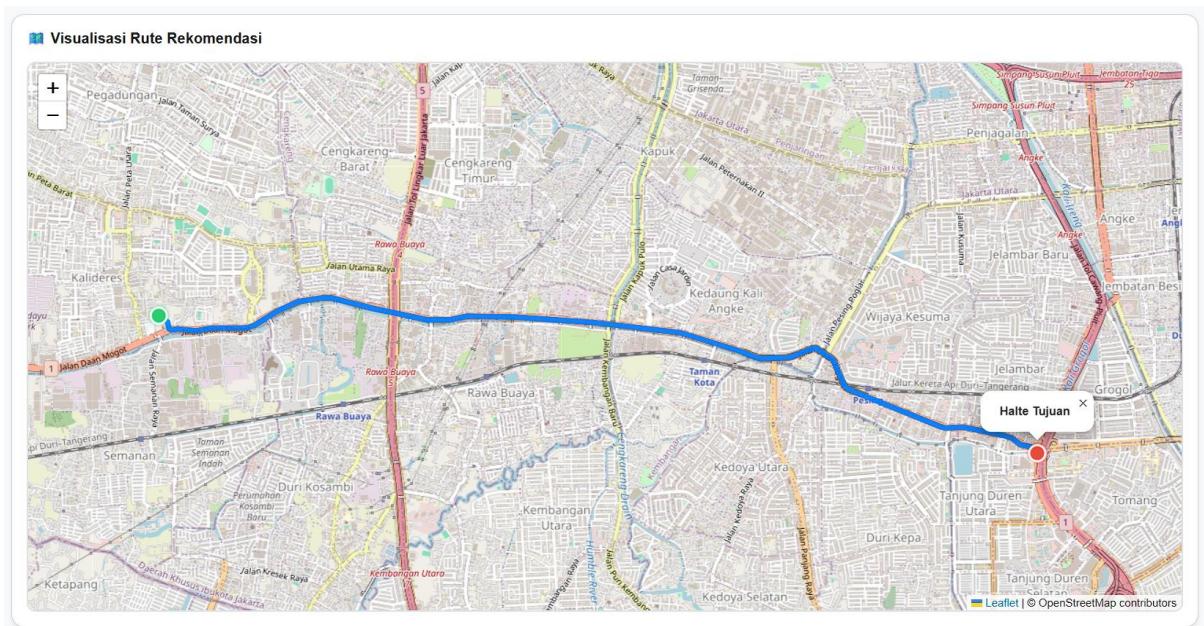
d. Informasi Popup

Mengklik *marker* halte pada peta akan menampilkan *popup* berisi nama halte dan informasi relevan lainnya (seperti koridor).

e. Kontrol Peta

Anda dapat memperbesar (*zoom in*), memperkecil (*zoom out*), dan menggeser (*pan*) peta untuk melihat detail rute.

Komponen output ini dirancang untuk memberikan informasi rute yang lengkap dan mudah dipahami, baik dalam format teks maupun visual. Secara lengkap, komponen output visualisasi dapat kita lihat melalui Gambar 3.9 berikut.



Gambar 3.9 Komponen Output Visualisasi Rute Rekomendasi

BAB IV

ALUR PENGGUNAAN APLIKASI

4.1 Langkah-langkah Pencarian Rute

Berikut adalah alur penggunaan aplikasi untuk mencari rute Transjakarta:

1. **Akses Aplikasi** dengan membuka *web browser* Anda dan akses URL aplikasi sesuai petunjuk di Bab 2. Pastikan halaman utama tampil dengan benar.
2. **Input Halte Asal** pada *form* pencarian, klik kolom "**Dari (Halte Asal)**" dan mulailah mengetik nama halte keberangkatan Anda. Gunakan fitur *autocomplete* (sugesti) yang muncul untuk memilih nama halte yang tepat atau pastikan ejaan sudah benar.
3. **Input Halte Tujuan** yang sama seperti pada halte asal di kolom "**Ke (Halte Tujuan)**". Ingat, halte tujuan tidak boleh sama dengan halte asal.
4. **Pilih Preferensi Rute**: Klik *dropdown* "**Preferensi Rute**" dan pilih salah satu opsi sesuai prioritas Anda, apakah rekomendasi, minim transit, atau tercepat.
5. **Atur Jam Keberangkatan** pada kolom "**Jam Keberangkatan**" menyesuaikan dengan kepergian anda.
6. **Pilih Metode Solver** pada kolom "**Metode Solver**" dan pilih algoritma yang ingin digunakan, apakah dengan MILP atau SMA
7. **Mulai Pencarian** dengan klik tombol "**Cari**". Aplikasi akan memproses permintaan Anda, dan waktu proses dapat bervariasi tergantung pada kompleksitas rute dan *solver* yang dipilih.

4.2 Membaca Hasil Rekomendasi

Setelah proses pencarian selesai dan rute ditemukan, hasil akan ditampilkan di bawah *form*:

1. Periksa Ringkasan Rute
Lihat di bagian ringkasan untuk informasi cepat mengenai estimasi Waktu Tempuh (normal dan perkiraan macet), Total Biaya, Jam Berangkat, dan Jumlah Transit. Perhatikan apakah jumlah transit "Tidak ada transit" atau sejumlah kali tertentu.
2. Pahami Detail Perjalanan langkah demi langkah sebagai berikut:
 - a. Perhatikan Koridor bus yang harus dinaiki.

- b. Catat Halte Naik ("Dari:") dan Halte Turun ("Ke:") untuk setiap segmen perjalanan.
- c. Lihat Halte yang Dilewati ("Melewati:") untuk orientasi selama perjalanan.
- d. Jika ada langkah Transit, perhatikan di Halte mana Anda harus turun dan pindah ke Koridor mana.
- e. Ikuti langkah-langkah ini hingga Anda mencapai langkah terakhir yang menandakan Tujuan Anda.

4.3 Menggunakan Peta Interaktif

Untuk pemahaman visual, gunakan peta di bagian bawah hasil dengan keterangan sebagai berikut:

- a. Peta akan secara otomatis menampilkan keseluruhan rute yang direkomendasikan.
- b. Perhatikan garis berwarna yang menunjukkan jalur bus.
- c. Lihat *marker* untuk halte asal (hijau), tujuan (merah), dan transit (ungu).
- d. Klik pada *marker* halte mana saja untuk melihat nama halte dan informasi terkait.
- e. Gunakan tombol + atau - atau *scroll mouse* untuk memperbesar/memperkecil peta.

Klik dan seret peta untuk menggeser tampilan.

BAB V

SKENARIO PENGGUNAAN

5.1 Skenario 1: Mencari Rute Seimbang (Blok M – Ancol) dengan MILP

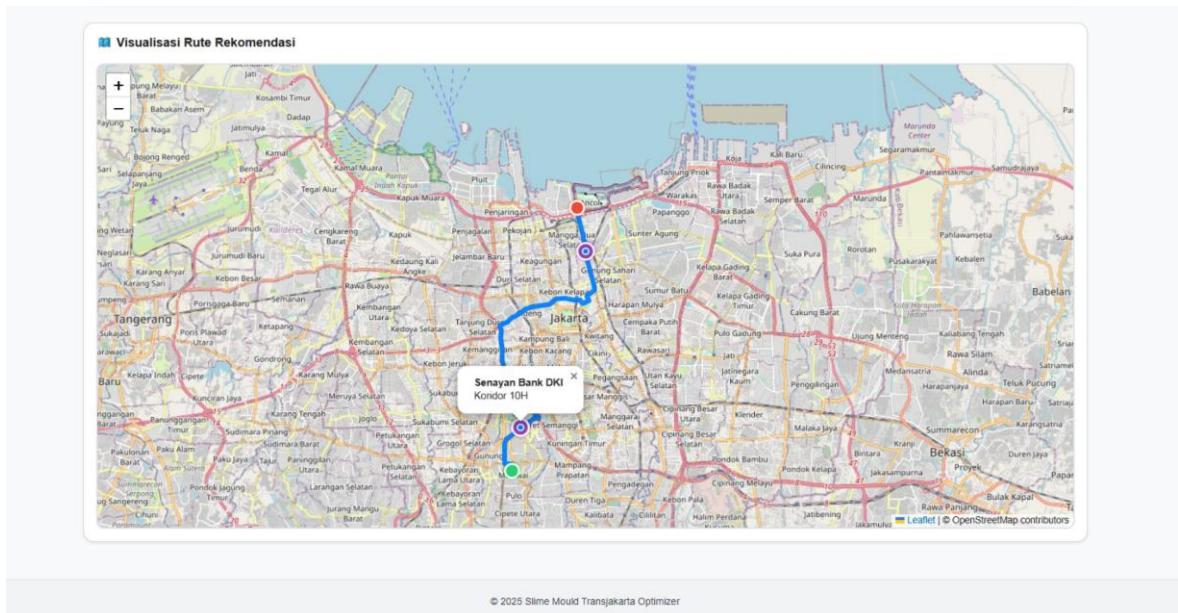
Tujuan: Pengguna ingin mencari cara tercepat untuk pergi dari Blok M ke Ancol dan tidak memikirkan permasalahan transit ataupun yang paling cepat, sehingga hasil terbaik menurut sistem dapat diterima mau bagaimanapun. Berikut adalah deskripsi input pengguna:

- Input Asal = Blok M
- Input Tujuan = Ancol
- Pilih Preferensi = Seimbang
- Jam Keberangkatan = 06.30
- Pilih Solver = MILP

Hasil tersebut dapat kita lihat melalui Gambar 5.1 dan Gambar 5.2 terkait ringkasan perjalanan dan hasil visualisasi dari skenario tersebut.

The screenshot shows a user interface for a route planning application. At the top, there are input fields for 'Dari (Halte Asal)' containing 'Blok M' and 'Ke (Halte Tujuan)' containing 'Ancol'. Below these are dropdown menus for 'Preferensi Rute' (set to 'Paling Efisien & Seimbang'), 'Jam Keberangkatan' (set to '06:30'), and 'Solver' (set to 'MILP'). A blue 'Cari' button is on the right. The main area displays the results under a green header 'Efisien & Seimbang (MILP)'. It lists travel details: Waktu Tempuh: 73 menit (Macet: 132 menit), Total Biaya: Rp 2,000, Jam Berangkat: 07:00, and Jumlah Transit: 2 kali. Below this, a 'DETAIL PERJALANAN:' section provides a detailed breakdown of the route segments, mentioning Koridor 1, 10H, and 5, along with various landmarks like Masjid Agung, Bundaran Senayan, and Jembatan Merah.

Gambar 5.1 Skenario Pengujian Preferensi Seimbang dengan MILP



Gambar 5.2 Visualisasi Skenario Pengujian Preferensi Seimbang dengan MILP

5.2 Skenario 2: Mencari Rute Minim Transit (Blok M – Ancol) dengan SMA

Tujuan: Pengguna ingin rute yang sama (Blok M ke Ancol), tetapi prioritas utamanya adalah menghindari transit yang sulit dan lebih memilih perpindahan yang mudah, meskipun total waktunya mungkin menjadi sedikit lebih lama.

- Input Asal = Blok M
- Input Tujuan = Ancol
- Pilih Preferensi = Minim Transit
- Jam Keberangkatan = 08.00
- Pilih Solver = SMA

Hasil tersebut dapat kita lihat melalui Gambar 5.3 dan Gambar 5.4 terkait ringkasan perjalanan dan hasil visualisasi dari skenario tersebut.

Dari (Halte Asal)	Ke (Halte Tujuan)	Preferensi Rute	Jam Keberangkatan	Metode Solver
Blok M	Ancol	<input checked="" type="checkbox"/> Minim Transit	--::--	SMA
<input type="button" value="Cari"/>				

Minim Transit (SMA)
Pilihan paling efisien dan seimbang

Waktu Tempuh: 68 menit (Macet: 123 menit)

Total Biaya: Rp 2,000

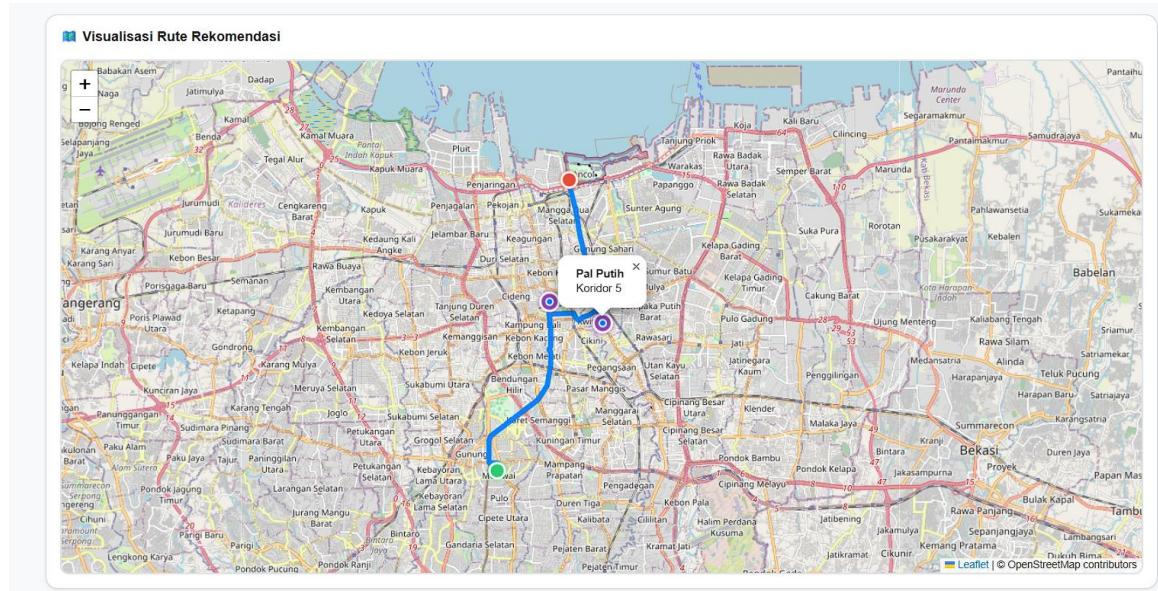
Jam Berangkat: 07:00

Jumlah Transit: 2 kali

DETAIL PERJALANAN:

- Naik:** Koridor 1
Dari: Blok M
Melewati: Masjid Agung → Bundaran Senayan → Senayan Bank DKI → Polda Metro Jaya → Bendungan Hilir → Karet → Dukuh Atas → Tosari → Bundaran HI ASTRA → M.H. Thamrin → Kebon Sirih Arah Utara
Ke: Monumen Nasional
- Transit:** Monumen Nasional (1 → 5C)
- Naik:** Koridor 5C
Dari: Monumen Nasional
Melewati: Balai Kota → Kwitang
Ke: Pal Putih
- Transit:** Pal Putih (5C → 5)
Naik: Koridor 5
Dari: Pal Putih
Melewati: Jaga Jakarta → Lapangan Banteng → Pasar Baru Timur → Jembatan Merah → Gunung Sahari → Pademangan
Ke: Ancol

Gambar 5.3 Skenario Pengujian Preferensi Minim Transit dengan SMA



Gambar 5.4 Visualisasi Skenario Pengujian Preferensi Minim Transit dengan SMA

5.3 Skenario 3: Mencari Rute Tercepat (Blok M – Ancol) dengan SMA

Tujuan: Pengguna ingin mencari cara tercepat untuk pergi dari Blok M ke Ancol dan tidak keberatan jika harus transit di tempat yang agak sulit (jalan jauh/menyeberang JPO), asalkan total waktunya paling minimal.

- a. Input Asal = Blok M

- b. Input Tujuan = Ancol
- c. Pilih Preferensi = Tercepat
- d. Jam Keberangkatan = 15.00
- e. Pilih Solver = SMA

Hasil tersebut dapat kita lihat melalui Gambar 5.5 dan Gambar 5.6 terkait ringkasan perjalanan dan hasil visualisasi dari skenario tersebut.

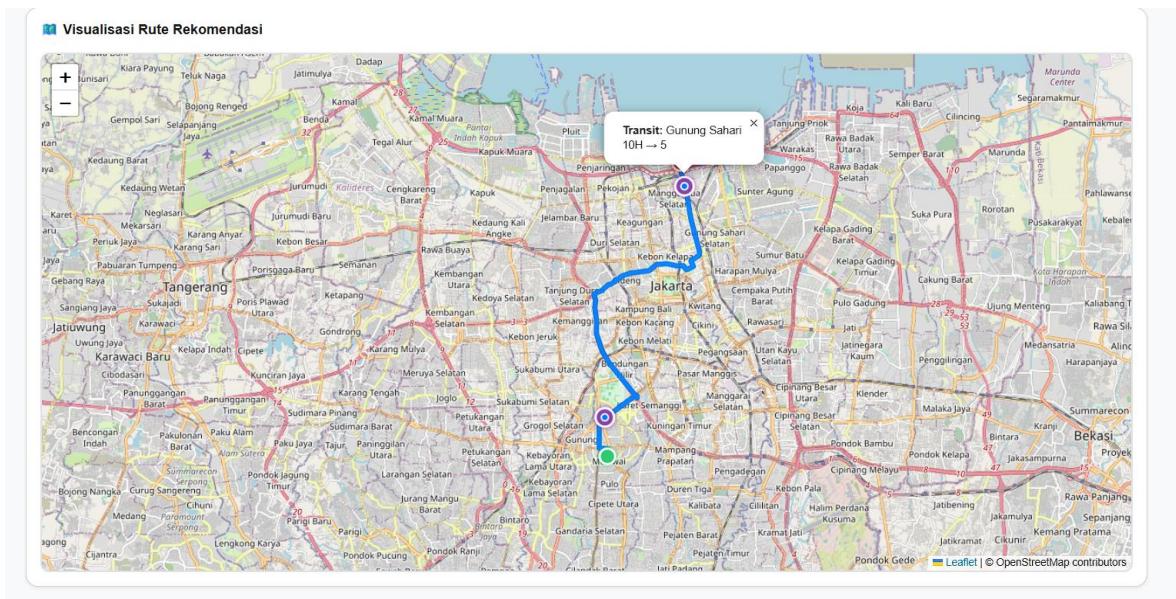
The screenshot shows a travel planning application interface. At the top, there are input fields for 'Dari (Halte Asal)' (Blok M) and 'Ke (Halte Tujuan)' (Ancol), along with dropdown menus for 'Preferensi Rute' (Paling Cepat), 'Jam Keberangkatan' (15.00), and 'Solver' (SMA). A 'Cari' (Search) button is also present. Below the search bar, a green header bar displays the result: 'Paling Cepat (SMA)' and 'Pilihan paling efisien dan seimbang'. The main content area contains the following information:

- Waktu Tempuh:** 63 menit (Macet: 113 menit)
- Total Biaya:** Rp 2,000
- Jam Berangkat:** 07:00
- Jumlah Transit:** 2 kali

DETAIL PERJALANAN:

- Naik:** Koridor 1
Dari: Blok M
Melewati: Masjid Agung
Ke: Bundaran Senayan
- Transit:** Bundaran Senayan (1 → 10H)
- Naik:** Koridor 10H
Dari: Bundaran Senayan
Melewati: Senayan Bank DKI → Gerbang Pemuda Arah Barat → Petamburan → Kemanggisan Arah Utara → Kota Bambu Arah Utara → Tomang Raya → Tarakan → Petojo → Pecenongan → Juanda → Pasar Baru Timur → Jembatan Merah
Ke: Gunung Sahari
- Transit:** Gunung Sahari (10H → 5)
- Naik:** Koridor 5
Dari: Gunung Sahari
Melewati: Pademangan
Ke: Ancol

Gambar 5.5 Skenario Pengujian Preferensi Paling Cepat dengan SMA



Gambar 5.6 Visualisasi kenario Pengujian Preferensi Paling Cepat dengan SMA

BAB VII

PENUTUP

6.1 Kesimpulan

Panduan Pengguna (Manual Book) ini telah menjabarkan secara rinci cara penggunaan Aplikasi Optimasi Rute Transjakarta. Dimulai dari pengenalan tujuan aplikasi, kebutuhan sistem, hingga penjelasan komponen antarmuka, panduan ini dirancang untuk membantu pengguna memanfaatkan fungsionalitas aplikasi secara maksimal.

Fitur utama aplikasi ini adalah kemampuannya untuk memberikan rekomendasi rute tidak hanya berdasarkan satu kriteria, tetapi berdasarkan preferensi perjalanan yang dipilih pengguna ("Paling Efisien", "Minim Transit", atau "Paling Cepat"). Selain itu, aplikasi ini menyediakan dua metode *solver* (MILP dan SMA) yang dapat dipilih.

Melalui alur penggunaan (BAB 4) dan contoh skenario (BAB 5) yang telah disajikan, pengguna diharapkan dapat memahami cara mengoperasikan aplikasi, membaca hasil rekomendasi (termasuk detail perjalanan dan visualisasi peta), serta memahami pesan kesalahan yang mungkin muncul. Dengan demikian, aplikasi ini dapat menjadi alat bantu yang efektif untuk merencanakan perjalanan yang lebih efisien dan sesuai dengan kebutuhan pribadi pengguna di jaringan Transjakarta.