

PROJECT BASED LEARNING

Chick Route - Rute Optimal Delivery Makanan dari Lapak Chicken ke Sekitar Kampus UNS dengan Efisiensi Jarak dan Biaya



Dosen Pengampu: Hasan Dwi Cahyono, S.Kom, M.Kom

Disusun Oleh:

- 1. ANISA DEVINA MAHARANI : L0124002**
- 2. APRILIA ALFA GUSASTI C. : L0124003**
- 3. AYU SANIATUS SHOLIAH : L0124005**

**PROGRAM STUDI INFORMATIKA FAKULTAS
TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET 2025**

DAFTAR ISI

BAB 1.....	4
PENDAHULUAN.....	4
1.1 Latar Belakang.....	4
1.2. Rumusan Masalah.....	5
1.3. Tujuan Proyek.....	5
1.4. Manfaat Proyek.....	5
BAB 2.....	6
DASAR TEORI.....	6
2.1. Traveling Salesman Problem (TSP).....	6
2.2. Algoritma Brute Force.....	6
2.3. Algoritma Dijkstra.....	7
2.4. Graf Berarah dan Tak Berarah.....	7
2.4.1 Graf Tak Terarah (Undirected Graph).....	7
2.4.2 Graf Terarah (Directed Graph).....	8
BAB 3.....	9
3.1 Variabel Proyek.....	9
3.2 Jenis Proyek.....	9
3.3 Teknik Pengumpulan Data.....	11
3.4 Analisis Data.....	11
BAB 4.....	12
HASIL DAN PEMBAHASAN.....	12
BAB 5.....	14
PENUTUP.....	14

DAFTAR PUSTAKA.....	15
LAMPIRAN.....	16

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Efisiensi operasional merupakan faktor krusial dalam layanan logistik, terutama di lingkungan yang dinamis dan padat seperti kawasan universitas. Untuk usaha skala mikro seperti "Lapak Chicken" di lingkungan kampus UNS, kemampuan untuk mengantarkan pesanan dengan cepat dan biaya sehemat mungkin dapat menjadi penentu keberlanjutan bisnis. Berbagai penelitian menunjukkan bahwa optimasi rute pengiriman dapat secara signifikan mengurangi biaya operasional dan meningkatkan efisiensi.

Permasalahan utama muncul dari "momen kebingungan" yang dihadapi oleh pengemudi setiap harinya. Di lingkungan sibuk seperti kampus UNS, dengan puluhan kemungkinan titik pengantaran, menentukan jalur yang benar-benar optimal menjadi sebuah tantangan krusial. Saat ini, realitanya banyak pengemudi yang hanya mengandalkan insting atau pengalaman pribadi. Pendekatan ini seringkali terbukti suboptimal dan tidak mampu beradaptasi secara efisien, yang berujung pada pemborosan waktu, bahan bakar, dan potensi keterlambatan pengantaran.

Untuk mengatasi masalah inefisiensi ini, diperlukan sebuah pendekatan yang sistematis dan berbasis data. Proyek ini mengusulkan penerapan teori graf untuk memodelkan jaringan jalan di kampus UNS. Dengan merepresentasikan setiap lokasi sebagai simpul (node) dan setiap jalan sebagai sisi (edge) yang memiliki bobot jarak, masalah penentuan rute dapat diubah menjadi masalah matematis yang dapat diselesaikan secara komputasi. Penggunaan Algoritma Dijkstra untuk mencari rute terpendek dan pendekatan Traveling Salesman Problem (TSP) untuk menangani rute multi-tujuan adalah metode yang valid dan telah banyak diimplementasikan dalam sistem informasi layanan pengiriman.

Oleh karena itu, tujuan utama dari proyek "Chick Route" ini adalah untuk merancang dan membangun sebuah sistem cerdas yang mampu memberikan rekomendasi rute pengiriman yang terpendek dan paling optimal, baik untuk tujuan tunggal maupun multi-tujuan, sehingga dapat meningkatkan efisiensi operasional secara signifikan.

1.2. Rumusan Masalah

1. Bagaimana merancang sistem yang dapat menentukan rute terpendek dari satu titik awal ke satu titik tujuan di lingkungan kampus UNS?
2. Bagaimana mengoptimalkan rute pengiriman untuk beberapa titik tujuan sekaligus agar total jarak yang ditempuh menjadi minimal?

1.3. Tujuan Proyek

1. Mengimplementasikan algoritma Dijkstra dan Traveling Salesman Problem (TSP) untuk menyelesaikan masalah penentuan rute tunggal dan ganda.
2. Membangun aplikasi fungsional yang mampu memberikan rekomendasi rute optimal beserta estimasi biayanya kepada pengguna.

1.4. Manfaat Proyek

1. Meningkatkan efisiensi operasional pengiriman melalui penghematan waktu dan bahan bakar dengan adanya rute yang optimal.
2. Menyediakan alat bantu pengambilan keputusan yang berbasis data bagi pengemudi, mengurangi ketergantungan pada insting atau perkiraan.

BAB 2

DASAR TEORI

2.1. Traveling Salesman Problem (TSP)

Traveling Salesman Problem (TSP) merupakan salah satu permasalahan klasik dalam bidang optimasi kombinatorial dan teori graf. Masalah ini bertujuan untuk menemukan lintasan atau sirkuit terpendek yang memungkinkan seorang agen atau kurir mengunjungi setiap kota atau titik tujuan tepat satu kali, dan kembali ke titik asal. TSP tergolong dalam kategori *NP-Hard*, yang berarti tidak ada algoritma deterministik yang diketahui mampu menyelesaikan permasalahan ini secara efisien dalam waktu polinomial untuk jumlah titik yang besar.

Permasalahan TSP memiliki berbagai aplikasi nyata dalam dunia logistik, perencanaan rute, pemrosesan chip elektronik, hingga bioinformatika (DNA sequencing). Kompleksitas permasalahan meningkat secara eksponensial terhadap jumlah kota yang harus dikunjungi. Oleh karena itu, berbagai pendekatan telah dikembangkan, mulai dari algoritma eksak seperti Brute Force dan Dynamic Programming, hingga pendekatan heuristik dan metaheuristik seperti Simulated Annealing dan Ant Colony Optimization ([Kothari et al., 2024](#)).

2.2. Algoritma Brute Force

Algoritma Brute Force merupakan metode penyelesaian TSP secara eksak yang bekerja dengan cara mengevaluasi seluruh kemungkinan urutan kunjungan (*permutasi*) dari kota-kota yang ada. Langkah-langkah utama dalam algoritma ini meliputi:

- Menghasilkan semua kemungkinan urutan perjalanan dari titik awal ke seluruh titik lainnya dan kembali ke titik awal
- Menghitung total jarak untuk setiap rute yang mungkin.
- Memilih rute dengan total jarak paling kecil sebagai solusi optimal.

Kompleksitas waktu algoritma ini adalah $O(n!)$, di mana n adalah jumlah titik tujuan. Oleh karena itu, algoritma ini hanya layak digunakan untuk ukuran

masukan yang relatif kecil (biasanya kurang dari 10 titik), karena penambahan satu titik saja dapat menyebabkan lonjakan jumlah kemungkinan rute secara drastis.

Penelitian oleh Ariyanti et al. (2019) menunjukkan efektivitas algoritma Brute Force dalam aplikasi pengiriman pesanan pada skala lokal. Meskipun memiliki waktu komputasi yang lebih tinggi dibanding metode heuristik, hasil yang diberikan adalah solusi optimal yang presisi.

Sementara itu, penelitian oleh Gohil et al. (2022) mengkaji penerapan paralelisasi untuk mempercepat eksekusi algoritma Brute Force, menunjukkan bahwa pendekatan ini tetap relevan apabila dikombinasikan dengan strategi komputasi paralel pada arsitektur multiprosesor.

2.3. Algoritma Dijkstra

Algoritma Dijkstra merupakan metode populer untuk mencari jalur terpendek dari satu titik asal ke titik tujuan dalam graf berbobot. Dijkstra bekerja dengan prinsip relaksasi jarak terpendek: setiap simpul (node) diperiksa, dan jarak terpendek diperbarui jika ditemukan lintasan baru yang lebih pendek.

Algoritma ini cocok untuk graf berbobot positif dan digunakan secara luas dalam sistem navigasi, jaringan komputer, dan perencanaan rute transportasi. Dalam studi oleh Rahmadi et al. (2025), implementasi Dijkstra berhasil digunakan untuk merancang rute optimal dari Jakarta ke Surabaya berdasarkan bobot jarak.

2.4. Graf Berarah dan Tak Berarah

2.4.1 Graf Tak Berarah (Undirected Graph)

Graf tak terarah adalah struktur graf di mana sisi-sisinya tidak memiliki arah tertentu. Hubungan antar simpul dalam graf ini bersifat dua arah; artinya, jika simpul A terhubung ke simpul B, maka secara implisit B juga terhubung ke A. Graf tak terarah banyak digunakan untuk memodelkan sistem dengan hubungan timbal balik, seperti jaringan sosial, sistem komunikasi simetris, dan peta jalan dua arah.

2.4.2 Graf Berarah(Directed Graph)

Graf berarah atau *digraph* adalah graf yang sisi-sisinya memiliki arah tertentu, ditandai dengan panah dari satu simpul ke simpul lainnya. Hubungan ini bersifat satu arah; jika terdapat sisi dari simpul A ke simpul B, belum tentu berlaku sebaliknya. Graf berarah umum digunakan dalam model sistem aliran data, lalu lintas satu arah, serta grafik dependensi dalam algoritma.

Menurut El-Ghoul dan Al-Obaidi (2012), graf berarah dan tak terarah dibedakan berdasarkan orientasi sisi. Dalam graf terarah, setiap sisi diasosiasikan dengan pasangan terurut (v,w) yang menunjukkan arah dari simpul v ke simpul w , sedangkan dalam graf tak terarah, sisi hanya menunjukkan keterhubungan antar simpul tanpa memperhatikan arah aliran data atau hubungan.

BAB 3

METODOLOGI

Proyek ini dilaksanakan secara berkelompok menggunakan laptop masing-masing anggota selama periode bulan Mei 2025. Objek yang menjadi fokus penelitian adalah jaringan jalan di lingkungan Kampus Ketingan Universitas Sebelas Maret (UNS), yang disimulasikan untuk optimasi rute pengiriman dari titik awal "Lapak Chicken". Untuk membangun sistem ini, perangkat lunak yang digunakan meliputi Bahasa Pemrograman Java dengan Java Development Kit (JDK) dan library Swing untuk membangun antarmuka pengguna grafis (GUI), serta Graphviz sebagai perangkat lunak eksternal untuk keperluan visualisasi graf.

3.1 Variabel Proyek

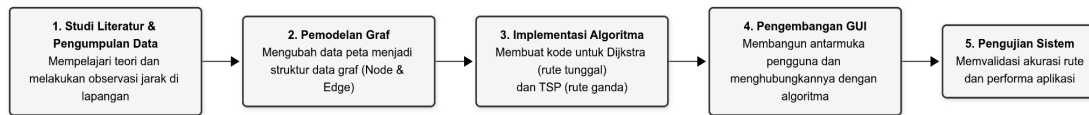
Pada proyek ini memiliki tiga variable diantaranya sebagai berikut:

1. **Variabel Bebas:** Titik tujuan yang dipilih oleh pengguna, baik berupa satu lokasi (tujuan tunggal) maupun beberapa lokasi (tujuan ganda).
2. **Variabel Terikat:** Hasil output dari sistem, yang meliputi rute optimal yang direkomendasikan, total jarak tempuh, dan estimasi total biaya pengiriman.
3. **Variabel Kontrol:** Data graf yang sudah ditentukan (kumpulan lokasi dan jarak antar lokasi), serta parameter perhitungan biaya seperti harga bensin dan ongkir minimum.

3.2 Jenis Proyek

Jenis penelitian yang digunakan adalah Penelitian dan Pengembangan (Research & Development) dengan pendekatan kuantitatif. Pendekatan R&D dipilih karena proyek ini berfokus pada proses untuk menghasilkan dan menguji validitas sebuah produk perangkat lunak baru, yaitu aplikasi "Chick Route". Di dalam proses pengembangan tersebut, pendekatan kuantitatif diterapkan dalam pengukuran dan validasi data, seperti saat menghitung jarak rute (meter), menganalisis efisiensi biaya (Rupiah)

Tahapan Pengembangan Sistem



Gambar 3. 1 Tahapan Pengembangan Sistem

Proyek ini dilaksanakan melalui beberapa tahapan utama sebagai berikut:

3.2.1 Studi Literatur dan Pengumpulan Data

Tahap awal meliputi studi literatur untuk memahami secara mendalam konsep teori graf, Algoritma Dijkstra, dan *Traveling Salesman Problem* (TSP). Selanjutnya, dilakukan pengumpulan data jarak antar lokasi di UNS melalui pendekatan hibrida: menggunakan Google Maps sebagai referensi awal dan dilanjutkan dengan observasi langsung di lapangan untuk validasi dan koreksi data.

3.2.2 Pemodelan Jaringan sebagai Graf

Jaringan jalan kampus yang telah didata kemudian dimodelkan sebagai graf berbobot. Setiap lokasi penting menjadi sebuah **Node**, dan setiap jalan menjadi **Edge** dengan bobot yang merepresentasikan jarak. Struktur data yang digunakan adalah Adjacency List (**Map<Node, List<Edge>>**) karena fleksibilitasnya. Model ini juga mendukung jalur satu arah (**directed**) untuk merepresentasikan kondisi lalu lintas nyata di kampus.

3.2.3 Implementasi Algoritma

Logika inti sistem diimplementasikan dalam dua bagian:

- Algoritma Dijkstra: Diimplementasikan untuk menangani kasus pengiriman ke satu tujuan. Algoritma ini menjamin ditemukannya rute dengan jarak terpendek.
- Algoritma TSP (Brute Force): Diimplementasikan untuk kasus multi-tujuan (dibatasi maks. 6). Untuk meningkatkan efisiensi, diterapkan strategi *caching* di mana jarak antar titik penting dihitung

terlebih dahulu menggunakan Dijkstra dan disimpan, sehingga tidak perlu dihitung berulang kali.

3.2.4 Pengembangan Antarmuka dan Integrasi

Seluruh logika algoritma diintegrasikan ke dalam sebuah antarmuka pengguna grafis (GUI) yang dibangun menggunakan Java Swing. GUI ini menyediakan dua mode operasi ("Satu Tujuan" dan "Multi Tujuan") serta menampilkan output rute dan biaya secara jelas kepada pengguna.

3.2.5 Pengujian Sistem

Pengujian dilakukan dengan menjalankan berbagai skenario, seperti memilih tujuan yang dekat, jauh, atau beberapa tujuan sekaligus. Hasil dari sistem kemudian divalidasi dengan perhitungan manual untuk kasus sederhana guna memastikan akurasi dan keoptimalan rute yang dihasilkan.

3.3 Teknik Pengumpulan Data

Teknik pengumpulan data untuk mengevaluasi sistem ini menggunakan metode Uji Komparatif. Dalam pengujian ini, kami menyiapkan beberapa skenario pengiriman dan membandingkan hasil output dari aplikasi—yaitu rute, total jarak, dan estimasi biaya—dengan hasil dari rute yang ditentukan secara manual. Perbandingan ini bertujuan untuk membuktikan dan mengukur secara kuantitatif tingkat efisiensi yang dicapai oleh sistem "Chick Route" dibandingkan dengan pendekatan konvensional.

3.4 Analisis Data

Analisis data pada proyek ini secara komputasional mengolah data graf untuk menghasilkan rute optimal. Sistem menggunakan Algoritma Dijkstra untuk mencari rute terpendek ke satu tujuan, dan Algoritma TSP (Brute Force) untuk menemukan tur paling efisien saat mengunjungi banyak tujuan. Jarak optimal yang ditemukan kemudian digunakan sebagai dasar analisis biaya untuk menghitung estimasi total ongkos kirim.

BAB 4

HASIL DAN PEMBAHASAN

Berdasarkan metodologi perancangan yang telah diuraikan pada bab sebelumnya, sistem optimasi rute "Chick Route" telah berhasil diimplementasikan dalam bentuk aplikasi desktop fungsional. Bab ini akan memaparkan dan menganalisis secara rinci hasil dari implementasi tersebut, yang mencakup pengujian fungsionalitas untuk fitur rute tujuan tunggal (Dijkstra) dan rute multi-tujuan (TSP), serta analisis performa dari setiap algoritma yang digunakan.

4.1 Hasil

Hasil pengujian sistem dilakukan dengan dua skenario utama, yaitu pengiriman ke satu tujuan dan pengiriman ke beberapa tujuan sekaligus. Pada skenario satu tujuan, sistem mampu menghitung rute terpendek secara otomatis menggunakan algoritma Dijkstra. Sebagai contoh, untuk pengiriman dari Lapak Chicken ke Fakultas Ekonomi dan Bisnis (FEB), sistem memilih rute tercepat melalui gerbang samping (Gerbang FMIPA), dengan total jarak tempuh 1.210 meter. Hal ini berbeda dengan Google Maps yang justru mengarahkan melalui gerbang belakang, menghasilkan jarak tempuh sekitar 1.500 meter. Temuan ini menunjukkan bahwa sistem memiliki pemahaman spasial yang lebih sesuai dengan kondisi sebenarnya di area kampus, karena mempertimbangkan jalur pintas internal yang tidak dikenali oleh peta komersial.

Pada skenario multi tujuan, sistem diuji dengan tiga titik tujuan: Gedung Javanologi, Pascasarjana, dan GOR UNS. Meskipun input dilakukan secara urut, sistem tidak mengikuti urutan tersebut secara langsung, melainkan menghitung kombinasi urutan terbaik menggunakan algoritma TSP (Traveling Salesman Problem) yang telah dioptimalkan dengan algoritma Dijkstra. Hasil penghitungan menunjukkan bahwa urutan optimal adalah mengunjungi Pascasarjana terlebih dahulu, kemudian GOR UNS, dan terakhir Gedung Javanologi. Dengan cara ini, total jarak tempuh

menjadi lebih pendek dibandingkan jika mengikuti urutan input pengguna. Hal ini membuktikan bahwa sistem tidak hanya bekerja untuk satu kasus saja, tetapi juga efektif dalam mengoptimalkan pengiriman ke banyak lokasi dalam satu kali perjalanan.

BAB 5

PENUTUP

5.1 Kesimpulan

Sistem pengiriman berbasis graf yang dikembangkan menggunakan algoritma Dijkstra dan TSP terbukti mampu memberikan solusi yang efisien untuk pengiriman dalam lingkungan kampus. Dalam kasus satu tujuan, sistem mampu mendeteksi rute tercepat berdasarkan kondisi jalur internal yang realistis, bahkan melampaui akurasi Google Maps. Sementara untuk pengiriman multi tujuan, integrasi algoritma Dijkstra dan TSP mampu menentukan urutan pengiriman paling optimal secara otomatis. Hal ini sangat berguna untuk efisiensi bahan bakar, waktu tempuh, dan kenyamanan pengemudi.

5.2 Saran

Ke depannya, kami menyarankan untuk mempelajari dan mengintegrasikan algoritma lanjutan guna meningkatkan efisiensi sistem. Salah satu yang dapat dipertimbangkan adalah algoritma A-Star (A^*) sebagai alternatif Dijkstra dalam pencarian satu tujuan, karena memiliki keunggulan dalam kecepatan melalui pendekatan heuristik. Selain itu, untuk kasus multi-tujuan, kami juga merekomendasikan eksplorasi algoritma heuristik atau aproksimasi TSP, seperti *Nearest Neighbor* atau *Simulated Annealing*, yang lebih efisien dibandingkan pendekatan brute force saat jumlah tujuan semakin banyak. Pengembangan ini diharapkan dapat menjadikan sistem lebih adaptif, responsif, dan tetap optimal dalam berbagai skenario pengiriman.

DAFTAR PUSTAKA

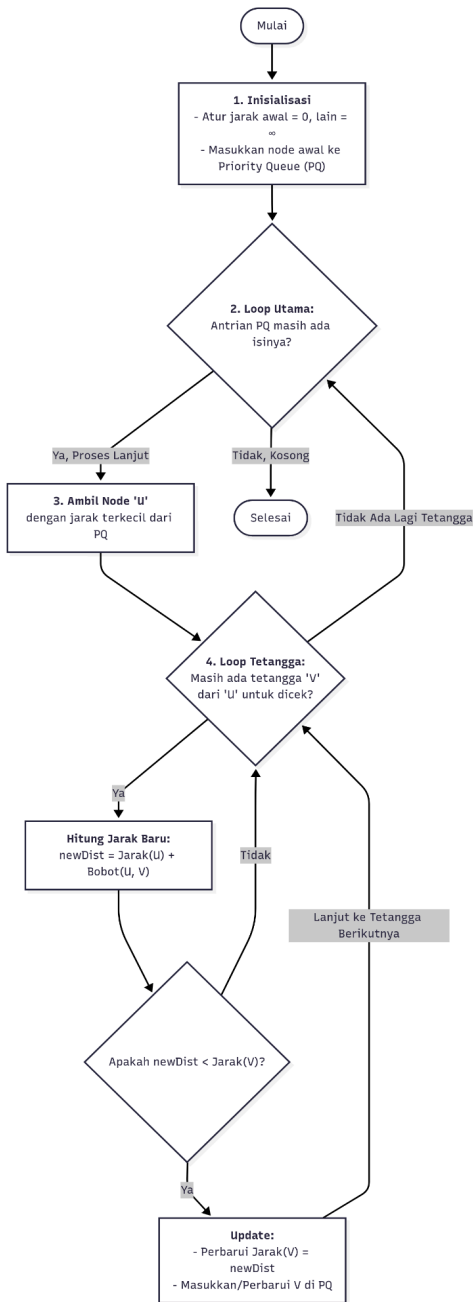
- Kothari, A., Nikhil, V. R. N. S., Rohit, N., & Sahay, A. (2024). *Harnessing meta-heuristic algorithms to optimize travelling salesman problem*. IEEE. <https://doi.org/10.1109/icccnt61001.2024.10726138>
- Ariyanti, I., Ganiardi, M. A., & Oktari, U. (2019). Mobile application searching of the shortest route on delivery order of CV. Alfa Fresh with brute force algorithm. *LOGIC: Journal of Information Technology*, 19(3), 123–132. <https://doi.org/10.31940/logic.v19i3.1437>
- Rahmadi, D., Putri, T. N., Ilmi, D. A., Rohmah, S. M., & Safinka, D. N. (2025). Optimasi jalur perjalanan antara Jakarta dan Surabaya menggunakan algoritma Dijkstra. *Al-Aqlu*, 3(1). <https://doi.org/10.59896/aqlu.v3i1.140>
- Tiong, A. Z. M., Panganiban, C. J. G., Blanco, M. C. R., Regala, R. C., & Cortez, D. M. A. (2022). Enhancement of Dijkstra algorithm for finding optimal path. *International Journal of Research Publications*, 100(1), 1–9. <https://doi.org/10.47119/ijrp1001021620223299>
- El-Ghoul, M. M., & Al-Obaidi, N. M. (2012). Neighborhoods of directed simplex graph. *International Journal of Mathematical Archive*, 3(1), 123–128.

LAMPIRAN

Link Repository Github:

<https://github.com/AyuSaniatusSholihah/Chick-Route---Rute-Optimal-Delivery-Makanan-/tree/main>

Alur Flowchart Algoritma Dijkstra



Penjelasan

1. Mulai & Inisialisasi (Kotak 1)

- Proses dimulai. Langkah pertama adalah persiapan: semua jarak ke setiap lokasi diatur sebagai "tak terhingga", kecuali untuk lokasi awal ("Lapak Chicken") yang jaraknya diatur ke 0. Lokasi awal ini kemudian dimasukkan ke dalam sebuah daftar prioritas (**PriorityQueue**).

2. Loop Utama (Wajik 2)

- Ini adalah titik pengecekan utama. Algoritma akan bertanya, "Apakah daftar prioritas (PQ) masih ada isinya?". Selama masih ada, proses akan terus berjalan. Jika sudah kosong, berarti proses selesai.

3. Ambil Node Terdekat (Kotak 3)

- Jika daftar prioritas masih ada isinya, algoritma akan mengambil node yang memiliki **jarak tercatat paling kecil**. Kita sebut node ini 'U'. Ini adalah langkah "greedy" atau "rakus" dari Dijkstra, karena ia selalu memilih yang paling optimal saat itu.

4. Loop Tetangga (Wajik 4)

- Setelah mendapatkan node 'U', sekarang kita masuk ke loop kedua untuk memeriksa semua tetangga yang terhubung langsung dengan 'U'. Algoritma akan bertanya, "Masih adakah tetangga dari 'U' yang belum dicek?".

5. Hitung & Bandingkan (Kotak F & Wajik G)

- Jika masih ada tetangga ('V'), sistem akan **menghitung jarak baru** ke tetangga tersebut melalui 'U'.
- Kemudian, ia akan **membandingkan**: Apakah jarak baru ini lebih pendek daripada jarak lama yang sudah tercatat untuk 'V'?

6. Update atau Lanjut (Kotak H)

- **Jika Ya:** Jarak ke 'V' akan **diperbarui** dengan jarak baru yang lebih pendek, dan 'V' akan dimasukkan ke dalam daftar prioritas (PQ) untuk diproses nanti.
- **Jika Tidak:** Tidak ada yang dilakukan.
- Setelah itu, alur kembali ke **langkah 4** untuk memeriksa tetangga berikutnya dari 'U'.

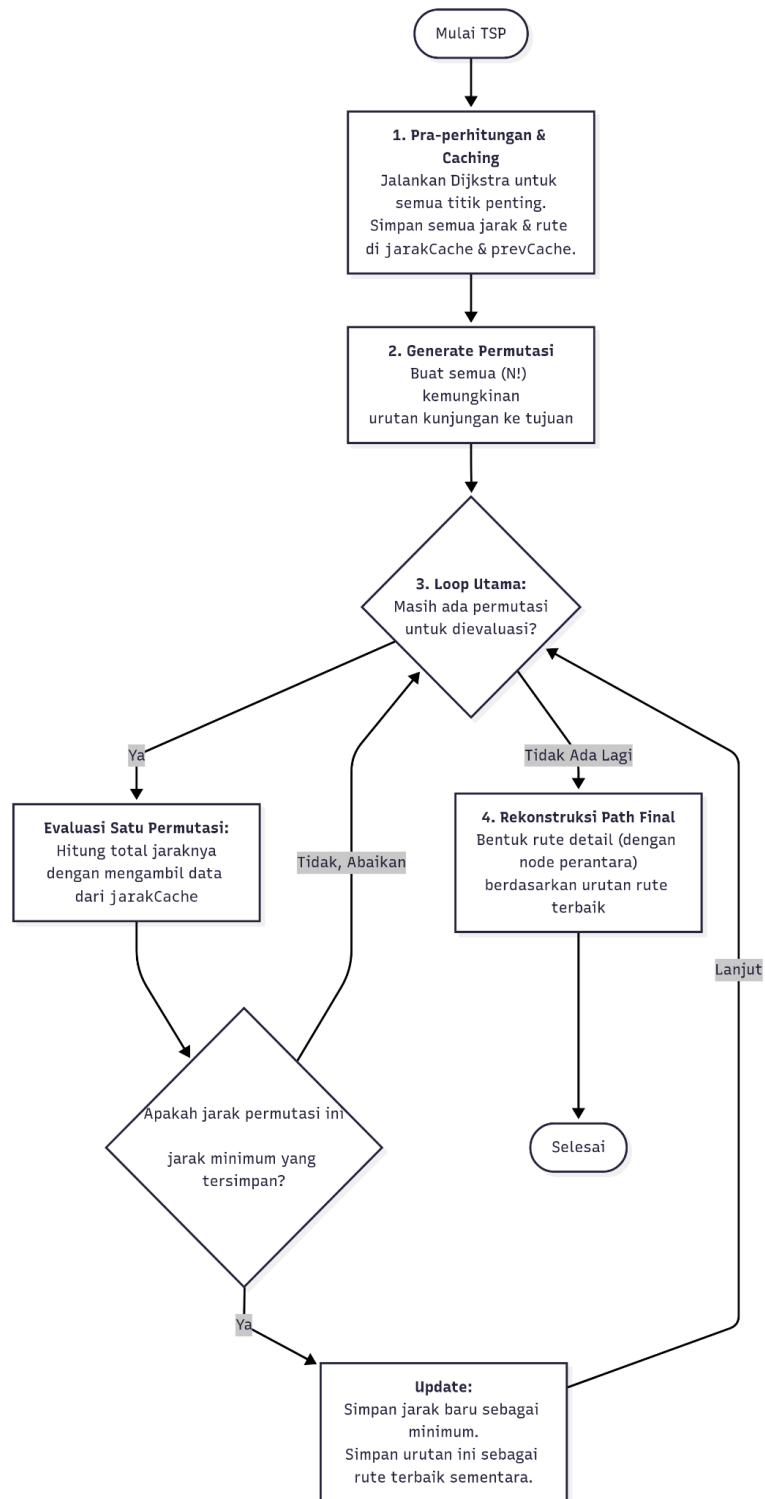
7. **Kembali ke Loop Utama**

- Jika semua tetangga dari 'U' sudah selesai diperiksa, alur akan kembali ke **langkah 2** untuk mengambil node terdekat berikutnya dari daftar prioritas.

8. **Selesai**

- Ketika daftar prioritas sudah kosong, loop utama berhenti. Proses selesai dan kita telah menemukan semua rute terpendek dari titik awal.

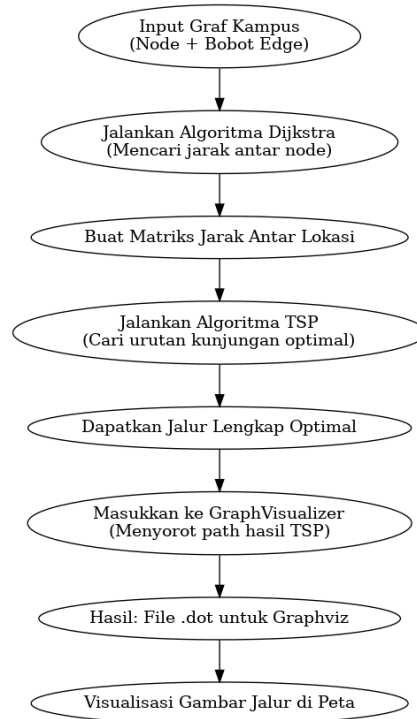
Alur Flowchart TSP



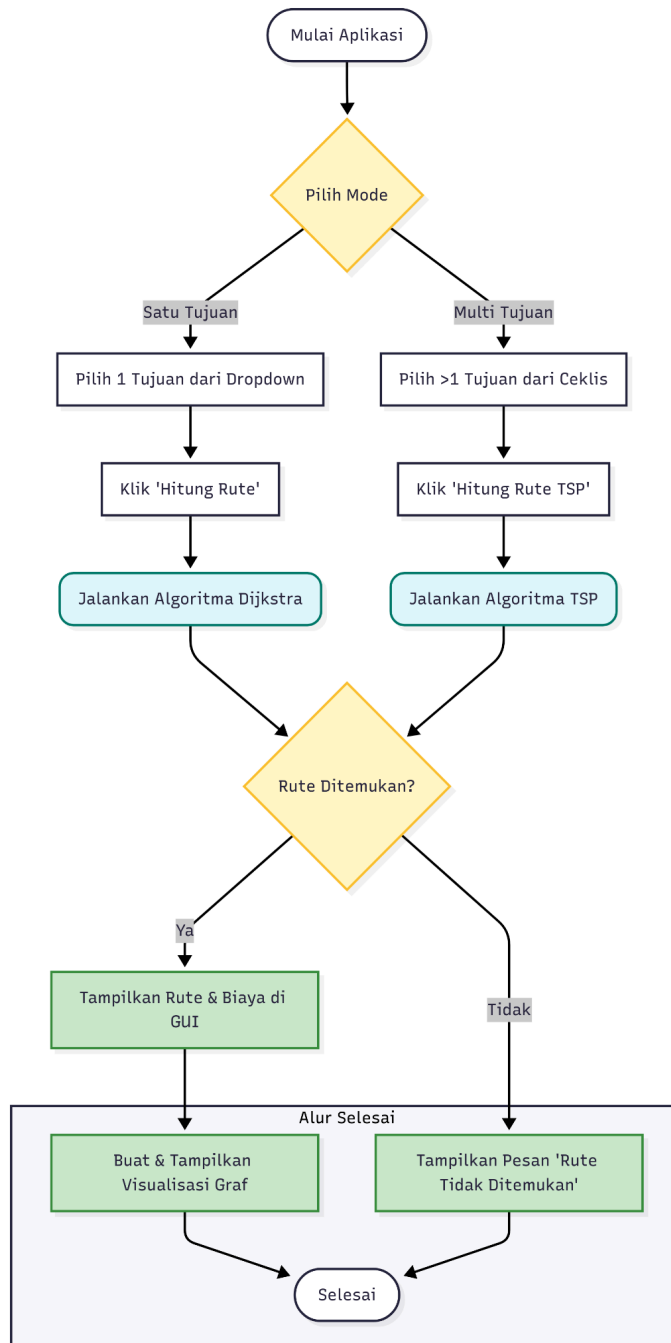
Penjelasan Alur Flowchart TSP

1. Pra-perhitungan & Caching: Sebelum proses utama dimulai, sistem secara cerdas menjalankan algoritma Dijkstra untuk menghitung dan menyimpan jarak terpendek antara semua titik penting (lokasi awal dan semua tujuan) ke dalam sebuah "cache". Ini adalah langkah optimasi agar tidak perlu melakukan perhitungan yang sama berulang kali.
2. Generate Permutasi: Sistem kemudian membuat daftar semua kemungkinan urutan perjalanan yang bisa ditempuh untuk mengunjungi semua tujuan yang dipilih. Ini adalah inti dari pendekatan *Brute Force*, di mana setiap kemungkinan akan dievaluasi.
3. Loop Utama & Evaluasi: Sistem akan melalui setiap kemungkinan urutan satu per satu.
 - Untuk setiap urutan, ia akan menghitung total jaraknya dengan sangat cepat, karena hanya perlu mengambil data dari cache yang sudah dibuat di langkah pertama.
 - Total jarak ini kemudian dibandingkan dengan "jarak minimum" yang pernah ditemukan sejauh ini. Jika lebih pendek, maka sistem akan memperbarui catatannya dan menyimpan urutan saat ini sebagai "rute terbaik sementara".
 - Proses ini diulang sampai semua kemungkinan urutan telah dievaluasi.
4. Rekonstruksi Path Final: Setelah loop selesai dan sistem sudah memiliki urutan kunjungan terbaik, langkah terakhir adalah membentuk jalur yang lengkap dan detail (termasuk jalan-jalan perantara) berdasarkan urutan tersebut untuk ditampilkan kepada pengguna.

Alur Visualisasi Rute Peta



Alur Penggunaan Sistem



Input 1 arah tujuan FEB

Chick Route - Delivery App

Chick Route Delivery

Satu Tujuan

Multi Tujuan (TSP)

Pilih Tujuan Tunggal

Tujuan: Fakultas Ekonomi Bisnis

Hitung Route

Route Optimal

Lapak Chicken

-> Gerbang Samping

-> Fakultas Hukum

-> Fakultas Ekonomi Bisnis

Ringkasan Biaya

Total Jarak Tempuh:

1,210 meter

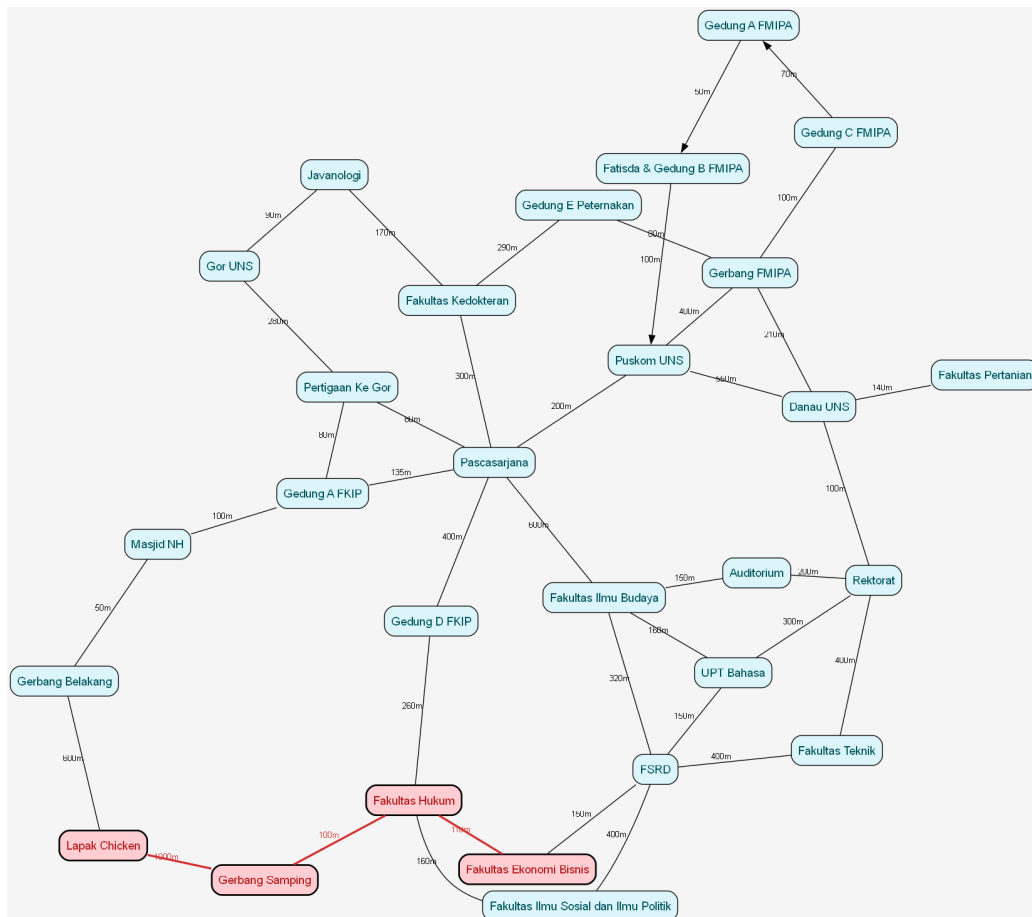
Estimasi Biaya Bensin (PP):

Rp2.622

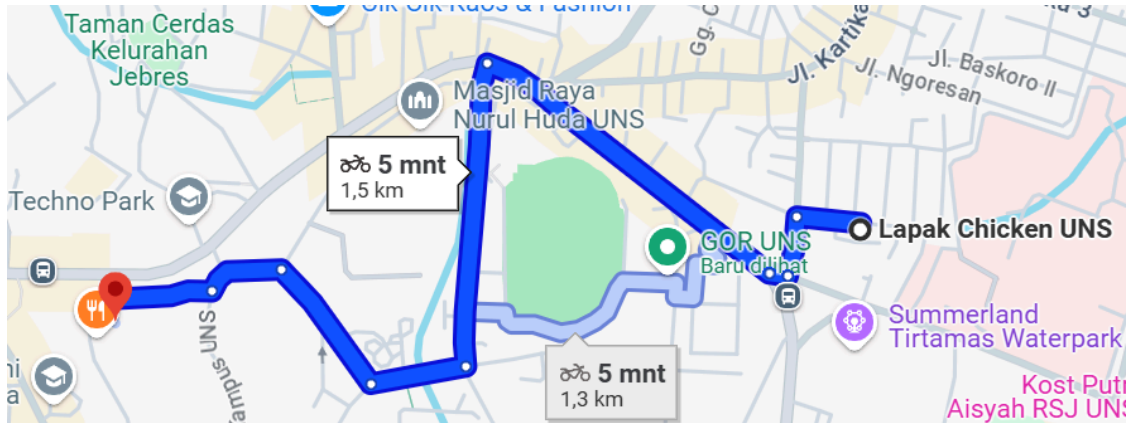
Total Ongkir:

Rp4.622

Visualisasi peta rute ke FEB



Peta google maps dari lapak chicken ke rektorat



Input multi tujuan ke Javanologi, Pascasarjana, GOR UNS

Chick Route - Delivery App

Chick Route Delivery

Satu Tujuan Multi Tujuan (TSP)

Pilih Beberapa Tujuan (Maks 6)

- ☒ Pascasarjana
- ☒ Gor UNS
- ☐ Fakultas Kedokteran

Hitung Route TSP

Route Optimal

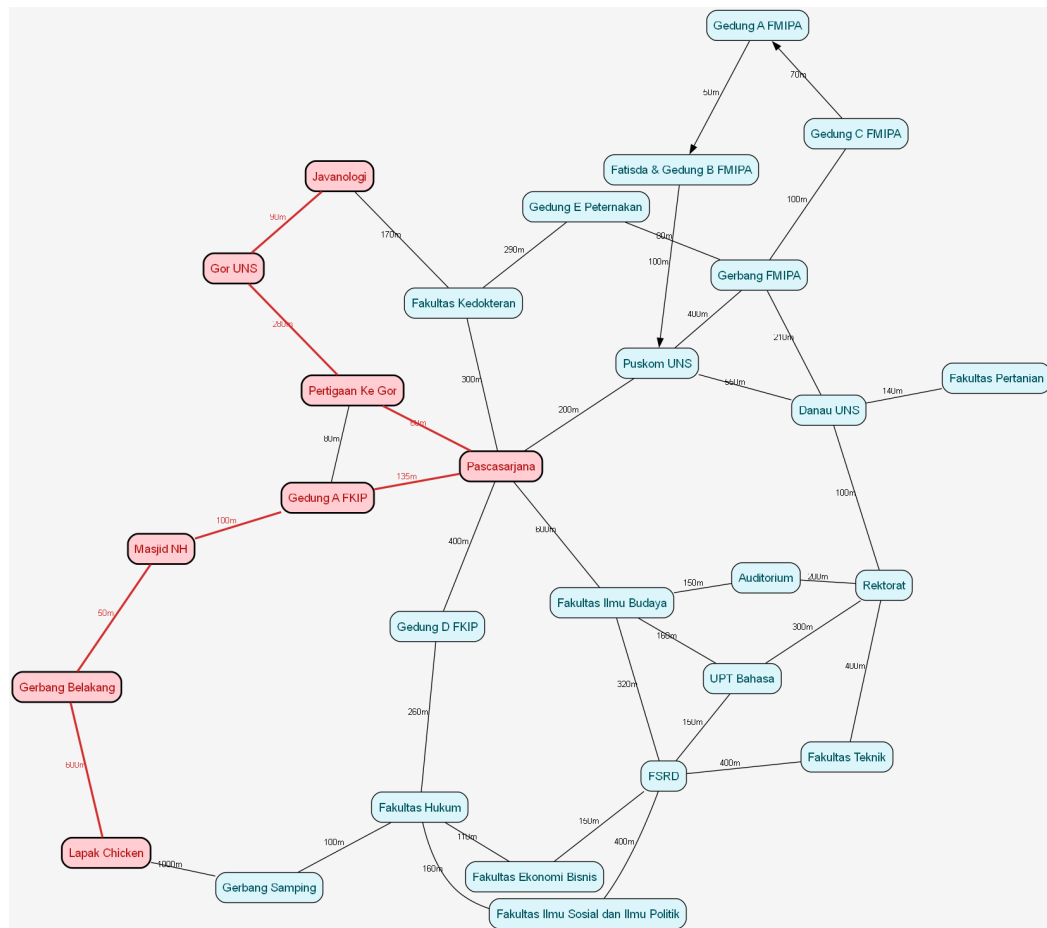
Lapak Chicken

- > Gerbang Belakang
- > Masjid NH
- > Gedung A FKIP
- > Pascasarjana
- > Pertigaan Ke Gor
- > Gor UNS
- > Javanologi

Ringkasan Biaya

Total Jarak Tempuh:	1,315 meter
Estimasi Biaya Bensin (PP):	Rp2.849
Total Ongkir:	Rp4.850

Visualisasi rute ke Javanologi, Pascasarjana, GOR UNS



Input multi tujuan ke gerbang FMIPA, Masjid NH, gedung A FMIPA,

Chick Route - Delivery App

Chick Route Delivery

Satu Tujuan

Multi Tujuan (TSP)

Pilih Beberapa Tujuan (Maks 6)

☐ Gedung U FKIP
 ☐ Pascasarjana
 ☒ Gor UNS

Hitung Rute TSP

Rute Optimal

-> Gerbang Belakng
 -> Masjid NH
 -> Gedung A FKIP
 -> Pertigaan Ke Gor
 -> Gor UNS
 -> Javanologi
 -> Fakultas Kedokteran
 -> Gedung E Peternakan
 -> Gerbang FMIPA
 -> Gedung C FMIPA
 -> Gedung A FMIPA

Ringkasan Biaya

Total Jarak Tempuh:

1,910 meter

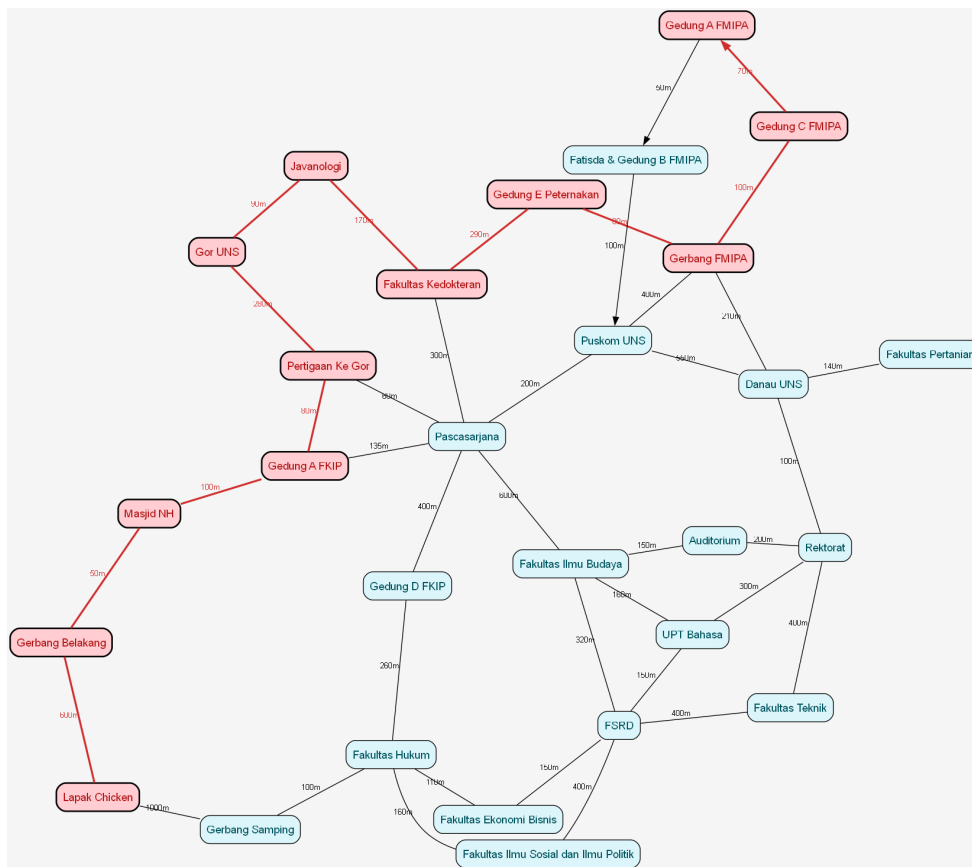
Estimasi Biaya Bensin (PP):

Rp4.138

Total Ongkir:

Rp6.139

Visualisasi rute ke gerbang FMIPA, Masjid NH, gedung A FMIPA,



Input rute ke FH, FIB, FT, FSRD, UPT Bahasa, FISIP

Chick Route - Delivery App

Chick Route Delivery

Satu Tujuan **Multi Tujuan (TSP)**

Pilih Beberapa Tujuan (Maks 6)

☒ Fakultas Hukum

☐ Fakultas Ekonomi Bisnis

☒ Fakultas Ilmu Sosial dan Ilmu Politik

Hitung Rute TSP

Rute Optimal

Lapak Chicken

- > Gerbang Samping
- > Fakultas Hukum
- > Fakultas Ilmu Sosial dan Ilmu Politik
- > FSRD
- > UPT Bahasa
- > Fakultas Ilmu Budaya
- > UPT Bahasa
- > FSRD
- > Fakultas Teknik

Ringkasan Biaya

Total Jarak Tempuh:	2,680 meter
Estimasi Biaya Bensin (PP):	Rp5.807
Total Ongkir:	Rp7.807

Visualisasi rute ke FH, FIB, FT, FSRD, UPT Bahasa, FISIP

