OPTIMASI ALGORITMA *CHEAPEST INSERTION HEURISTIC* DENGAN ALGORITMA *TABU SEARCH* DALAM PENCARIAN RUTE TERPENDEK

SKRIPSI

OLEH:
SILVIYATUS YULIANTI
NIM. 210601110010

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025

OPTIMASI ALGORITMA *CHEAPEST INSERTION HEURISTIC* DENGAN ALGORITMA *TABU SEARCH* DALAM PENCARIAN RUTE TERPENDEK

SKRIPSI

Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)

Oleh
Silviyatus Yulianti
NIM. 210601110010

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025

OPTIMASI ALGORITMA *CHEAPEST INSERTION HEURISTIC* DENGAN ALGORITMA *TABU SEARCH* DALAM PENCARIAN RUTE TERPENDEK

SKRIPSI

Oleh
Silviyatus Yulianti
NIM. 210601110010

Telah Disetujui untuk Diuji

Malang, 14 April 2025

Dosen Pembimbing I	Dosen Pembimbing II
M. Nafie Jauhari, M.Si	Ach. Nasichuddin, M.A
NIPPPK. 19870218 202321 1 018	NIP. 19730705 200003 1 002

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc
NIP. 19741129 200012 2 005

OPTIMASI ALGORITMA *CHEAPEST INSERTION HEURISTIC* DENGAN ALGORITMA *TABU SEARCH* DALAM PENCARIAN RUTE TERPENDEK

SKRIPSI

Oleh
Silviyatus Yulianti
NIM. 210601110010

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)

Tanggal 19 Mei 2025

Ketua Penguji	: Prof. Dr. H. Turmudi, M.Si., Ph.D	.....
Anggota Penguji 1	: Muhammad Khudzaifah, M.Si	.....
Anggota Penguji 2	: M. Nafie Jauhari, M.Si	.....
Anggota Penguji 3	: Ach. Nasichuddin, M.A	.....

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc
NIP. 19741129 200012 2 005

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Silviyatus Yulianti

NIM : 210601110010

Program Studi : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Optimasi Algoritma *Cheapest Insertion Heuristic* dengan Algoritma *Tabu Search* dalam Pencarian Rute Terpendek

Menyatakan dengan sebenarnya bahwa skripsi ini saya tulis benar – benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai tulisan saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar rujukan. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 19 Mei 2025

Silviyatus Yulianti

NIM. 210601110010

MOTO

"*Bersabarlah dalam setiap ujian, bertawakkallah dalam setiap langkah, berprasangkalah baik atas setiap takdir, dan yakinlah bahwa setelah gelap, cahaya Allah akan datang*."

(QS. Al-Baqarah: 153, QS. Al-Ahzab: 3, QS. At-Talaq: 2-3, QS. Al-Insyirah: 6)

PERSEMBAHAN

Bismillahirohmanirohim, dengan memanjatkan puji syukur ke hadirat Allah Swt., Dzat yang Maha Pengasih dan Maha Penyayang, atas limpahan rahmat, taufik, dan hidayah-Nya, setiap langkah dalam perjalanan ini dapat terlewati dengan baik. Dengan penuh rasa hormat dan cinta, skripsi ini penulis persembahkan kepada: Ibunda tercinta, Yuliatin, sosok penuh kasih yang dengan doa, dukungan, dan ketulusan tiada batas telah menjadi cahaya di setiap langkah perjalanan penulis. Ayahanda tercinta, Sutrisno, pilar kekuatan dan keteguhan yang dengan penuh pengorbanan selalu menjadi penopang dalam setiap usaha dan cita-cita penulis. Adik penulis, Ach Diyon Firmansyah, serta kakak penulis, Maudiah Sutriani, yang kehadiran, semangat, dan kebersamaannya menjadi penguat dalam setiap suka dan duka. Seluruh keluarga besar, yang kehangatannya selalu menjadi tempat berpulang dan menemukan makna dari setiap perjuangan. Para dosen yang dengan ilmu, kesabaran, dan dedikasi telah membuka cakrawala pemahaman penulis, serta teman-teman yang telah mengukir kenangan penuh makna, berbagi tawa, semangat, dan kebersamaan dalam setiap langkah perjuangan ini. Teristimewa, penulis tujukan untuk diri penulis sendiri, untuk setiap malam panjang yang dipenuhi doa dan usaha, untuk setiap kegagalan yang mengajarkan arti ketahanan, dan untuk setiap keberhasilan yang membuktikan bahwa ketekunan tidak pernah sia-sia.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan judul "Optimasi Algoritma *Cheapest Insertion Heuristic* dengan Algoritma *Tabu Search* dalam Pencarian Rute Terpendek" sebagai salah satu syarat untuk menyelesaikan studi di Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim.

Proses penyelesaian skripsi ini tidak terlepas dari bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua yang telah memberikan bantuan dan dukungan selama proses ini berlangsung. Dengan kerendahan hati, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim.
2. Prof. Dr. Sri Harini, M.Si, selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
3. Dr. Elly Susanti, S.Pd., M.Sc, selaku ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim.
4. M. Nafie Jauhari, M.Si., selaku dosen pembimbing I, atas bimbingan, arahan, serta saran yang sangat berharga dalam penyusunan skripsi ini.
5. Ach. Nasichuddin, M.A., selaku dosen pembimbing II, atas bimbingan, masukkan, serta arahan dalam penyusunan skripsi ini.
6. Prof. Dr. H. Turmudi, M.Si., Ph.D., selaku dosen ketua penguji, atas kritik dan saran yang membangun selama proses penyusunan skripsi ini.
7. Muhammad Khudzaifah, M.Si., selaku dosen anggota penguji I, atas saran yang membangun selama proses penyusunan skripsi ini.
8. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim, atas ilmu, pengalaman, dan inspirasi yang telah diberikan selama masa studi.

9. Ibu dan bapak tercinta, atas cinta yang tidak mengenal batas, doa yang tiada henti dipanjatkan, serta dukungan yang selalu mengalir tanpa lelah. Tanpa kalian, penulis tidak akan dapat mencapai titik ini. Kalian adalah sumber kekuatan dan inspirasi dalam setiap langkah kehidupan penulis.
10. Kakak dan adek tersayang, atas semangat dan dukungan dalam setiap langkah hidup penulis.
11. Seluruh keluarga besar, atas dukungan dan dorongan semangat yang sangat berarti.
12. Seluruh mahasiswa angkatan 2021 Program Studi Matematika, Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
13. Teman-teman terdekat, atas dukungan, kebersamaan, serta dorongan semangat yang telah diberikan selama masa studi.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, namun dengan penuh keikhlasan, penulis mempersembahkannya sebagai wujud penghargaan dan rasa hormat kepada semua pihak yang telah memberikan kontribusi, dukungan, dan inspirasi kepada penulis. Semoga skripsi ini dapat memberikan manfaat bagi pembaca dan semua pihak yang berkepentingan, serta berkontribusi pada pengembangan ilmu pengetahuan.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 19 Mei 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL ..... **i**
HALAMAN PENGAJUAN ..... **ii**
HALAMAN PERSETUJUAN ..... **iii**
HALAMAN PENGESAHAN ..... **iv**
PERNYATAAN KEASLIAN TULISAN ..... **v**
MOTO ..... **vi**
PERSEMBAHAN ..... **vii**
KATA PENGANTAR ..... **viii**
DAFTAR ISI ..... **x**
DAFTAR TABEL ..... **xii**
DAFTAR GAMBAR ..... **xiii**
DAFTAR LAMPIRAN ..... **xiv**
ABSTRAK ..... **xv**
ABSTRACT ..... **xvi**
مستخلص البحث ..... **xvii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 5
1.3 Tujuan Penelitian 5
1.4 Manfaat Penelitian 6
1.5 Batasan Masalah 6
BAB II KAJIAN TEORI **8**
2.1 Optimasi 8
2.2 Graf 9
2.2.1 Definisi Graf 9
2.2.2 Jenis-Jenis Graf 10
2.2.3 Istilah Dasar Graf Berarah 12
2.3 Rute Terpendek 13
2.3.1 *Traveling Salesman Problem* 14
2.4 Algoritma *Cheapest Insertion Heuristic* (CIH) 15
2.5 Algoritma *Tabu Search* 21
2.6 Pemanfaatan Waktu dalam Perspektif Al-Quran 25
BAB III METODE PENELITIAN **28**
3.1 Jenis Penelitian 28
3.2 Data dan Sumber Data 28
3.3 Langkah-langkah Penelitian 29
BAB IV HASIL DAN PEMBAHASAN **34**
4.1 Memodelkan Masalah dalam Graf Berarah 34
4.2 Penerapan Algoritma *Cheapest Insertion Heuristic* (CIH) 36
4.3 Penerapan Optimasi Algoritma CIH Menggunakan Algoritma *Tabu Search* 44
4.4 Uji Pengaruh Algoritma *Tabu Search* terhadap Kualitas Solusi Algoritma CIH 51
4.5 Efisiensi Waktu dalam Algoritma Optimasi 55

BAB V PENUTUP 58
5.1 Kesimpulan 58
5.2 Saran untuk Penelitian Lanjutan 60
DAFTAR PUSTAKA 61
LAMPIRAN 64
RIWAYAT HIDUP 71

DAFTAR TABEL

Tabel 2.1 Jarak Antar Lokasi 18
Tabel 2.2 Proses Sisipan Pertama 19
Tabel 2.3 Proses Sisipan Kedua 19
Tabel 2.4 Proses Sisipan Ketiga 20
Tabel 2.5 Proses Pencarian Rute Alternatif Iterasi Pertama 23
Tabel 2.6 Proses Pencarian Rute Alternatif Iterasi Kedua 24
Tabel 2.7 Proses Pencarian Rute Alternatif Iterasi Ketiga 24
Tabel 4.1 Data Titik Parkir Kecamatan Sukun 35
Tabel 4.2 Data Jarak Antar Titik dalam Satuan Kilometer 36
Tabel 4.3 Sisipan Pertama Algoritma CIH 37
Tabel 4.4 Sisipan Kedua Algoritma CIH 38
Tabel 4.5 Sisipan Ketiga Algoritma CIH 39
Tabel 4.6 Total Jarak Tempuh Rute yang Dihasilkan Algoritma CIH 43
Tabel 4.7 Proses Optimasi Rute Iterasi Pertama 45
Tabel 4.8 Proses Optimasi Rute Iterasi Kedua 46
Tabel 4.9 Proses Optimasi Rute Iterasi Ketiga 47
Tabel 4.10 Rute dan Total Jarak Tempuh dari Optimasi 50
Tabel 4.11 Perbandingan Jarak Total dalam Satuan Kilometer 52

DAFTAR GAMBAR

Gambar 2.1 Graf Berbobot dengan 5 titik dan 7 Sisi 11
Gambar 2.2 Graf Perjalanan 17
Gambar 2.3 Rute CIH 20
Gambar 2.4 Rute *Tabu Search* 25
Gambar 4.1 Model Masalah dalam Graf Berarah 35
Gambar 4.2 Visualisasi Rute Algoritma CIH dengan Titik Awal Dinas Perhubungan Kota Malang dan Jl. Kasin Jaya 42
Gambar 4.3 Visualisasi Rute Hasil Optimasi CIH menggunakan *Tabu Search* 48

DAFTAR LAMPIRAN

Lampiran 1 Data Jarak Antar Titik dalam Satuan Kilometer 64
Lampiran 2 Kode Python 65
Lampiran 3 *Output Python* Rute Optimasi Algoritma CIH dan *Tabu Search*..... 68
Lampiran 4 Visualisasi Rute Algoritma CIH..... 68
Lampiran 5 Visualisasi Rute Optimasi Algoritma CIH menggunakan *Tabu Search* 69

ABSTRAK

Yulianti, Silviyatus. 2025. **Optimasi Algoritma *Cheapest Insertion Heuristic* dengan Algoritma *Tabu Search* dalam Pencarian Rute Terpendek**. Skripsi. Program Studi Matematika Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) M. Nafie Jauhari, M.Si. (II) Ach. Nasichuddin, M.A.

Kata kunci: *Traveling Salesman Problem*, Optimasi Rute, Algoritma *Cheapest Insertion Heuristic*, Algoritma *Tabu Search*.

Masalah pencarian rute terpendek merupakan salah satu topik penting dalam teori graf dan optimasi kombinatorial, dengan aplikasi luas di bidang logistik, transportasi, dan penjadwalan. Penelitian ini bertujuan untuk meningkatkan kualitas solusi dan efisiensi waktu dalam penyelesaian masalah *Traveling Salesman Problem* (TSP) melalui optimasi algoritma *Cheapest Insertion Heuristic* (CIH) menggunakan algoritma *Tabu Search*. Algoritma CIH digunakan untuk membentuk solusi awal dengan menyisipkan titik berdasarkan bobot minimum, sedangkan *Tabu Search* diimplementasikan untuk mengoptimalkan solusi dengan menghindari jebakan pada solusi lokal melalui mekanisme daftar tabu. Data penelitian berupa jarak antar titik lokasi pengambilan retribusi parkir oleh Dinas Perhubungan Kota Malang di Kecamatan Sukun, yang diperoleh dari *Google Maps*. Evaluasi kinerja algoritma dilakukan dengan membandingkan total jarak tempuh sebelum dan sesudah optimasi, serta dianalisis secara statistik menggunakan uji *Wilcoxon signed-rank test* karena data tidak berdistribusi normal. Hasil penelitian menunjukkan bahwa optimasi algoritma CIH menggunakan algoritma *Tabu Search* secara signifikan menghasilkan rute dengan jarak tempuh lebih pendek dibandingkan penggunaan algoritma CIH secara tunggal. Temuan ini membuktikan bahwa optimasi algoritma CIH menggunakan algoritma *Tabu Search* mampu meningkatkan efektivitas dalam pencarian solusi rute terpendek.

ABSTRACT

Yulianti, Silviyatus. 2025. **Optimization of Cheapest Insertion Heuristic Algorithm with Tabu Search Algorithm in Shortest Route Search**. Thesis. Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) M. Nafie Jauhari, M.Si. (II) Ach. Nasichuddin, M.A

Keywords: Traveling Salesman Problem, Route Optimization, Cheapest Insertion Heuristic Algorithm, Tabu Search Algorithm.

The shortest route finding problem is a significant topic in graph theory and combinatorial optimization, with wide applications in logistics, transportation, and scheduling. This research aims to improve the quality of the solution and time efficiency in solving the Traveling Salesman Problem (TSP) by optimizing the Cheapest Insertion Heuristic (CIH) algorithm using the application of the Tabu Search algorithm. The CIH algorithm constructs an initial solution by inserting points based on minimum weight. At the same time, the Tabu Search algorithm is applied to enhance the solution by avoiding local optima using a tabu list mechanism. The research data, consisting of the distances between parking retribution collection points by the Malang City Transportation Agency in Sukun Sub-district, were obtained from Google Maps. The algorithm performance evaluation is done by comparing the total mileage before and after optimization, and statistically analyzed using the Wilcoxon signed-rank test because the data does not follow a normal distribution. The results showed that optimizing the CIH algorithm using the Tabu Search algorithm significantly resulted in routes with shorter travel distances than using the CIH algorithm alone. This finding proves that optimizing the CIH algorithm with Tabu Search increases the effectiveness of finding the shortest route.

مستخلص البحث

يولي انتي، سيلفيتوس. ٢٠٢٥. **التحسين الأمثل للخوارزمية *Cheapest Insertion Heuristic* مع الخوارزمية *Tabu Search* في إيجاد أقصر الطريق**. البحث العلمي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (١) محمد نافع جوهري، الماجستير. (٢) أحمد ناصح الدين، الماجستير

الكلمات المفتاحية :مشكلة البائع المتجول، تحسين المسار، خوارزمية *Cheapest Insertion Heuristic*، خوارزمية *Tabu Search*.

تعد مشكلة العثور على أقصر طريق من إحدي الموضوعات المهمة في نظرية الرسم البياني والتحسين التوافقي، مع تطبيقات واسعة في الخدمات اللوجستية والنقل والجدولة. هدف هذا البحث إلى تحسين جودة الحل وكفاءة الوقت في *Traveling Salesman Problem* (TSP) من خلال تحسين خوارزمية *Cheapest Insertion Heuristic* (CIH)باستخدام الخوارزمية *Tabu Search*. تم استخدام خوارزمية CIH لتكوين حل الاولي عن طريق إدخال النقاط بناءً على الحد الأدنى للوزن، بينما يتم تنفيذ *Tabu Search* لتحسين الحل عن طريق تجنب الوقوع في فخ الحلول المحلية من خلال آلية *tabu list*. بيانات البحث هي في شكل المسافة بين نقاط تجميع جزاءات مواقف السيارات من قبل وكالة النقل في مدينة مالانج في منطقة سوكون الفرعية، والتي تم الحصول عليها من *Google Maps*. تم تقييم أداء الخوارزمية من خلال مقارنة إجمالي المسافة المقطوعة قبل التحسين وبعده، ويتم تحليلها إحصائيًا باستخدام اختبار *Wilcoxon signed-rank test* لأن البيانات ليست موزعة بشكل طبيعي. أظهرت النتائج أن تحسين خوارزمية CIH باستخدام خوارزمية *Tabu Search* أدى إلى مسار أقصر مسافة مقطوعة بشكل ملحوظ من استخدام خوارزمية CIH وحدها. وثبت هذه النتيجة أن تحسين خوارزمية CIH باستخدام الخوارزمية *Tabu Search* يمكن أن يزيد فقاليتها في إيجاد أقصر حل للمسار.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Rute terpendek adalah konsep dalam teori graf yang merujuk pada rute yang menghubungkan titik awal dengan titik tujuan dalam graf dengan total jarak minimum dibandingkan rute lainnya (Kholidasari et al., 2022). Graf tersebut umumnya terdiri dari titik-titik yang dihubungkan oleh sisi-sisi yang memiliki bobot, merepresentasikan jarak atau biaya perjalanan antar titik. Dalam berbagai aplikasi, pencarian rute terpendek menjadi salah satu permasalahan optimasi yang banyak digunakan, karena dapat membantu dalam penyelesaian masalah rute yang efisien.

Optimasi rute sangat penting dalam berbagai bidang, termasuk logistik, transportasi, dan penjadwalan. Pada sektor logistik, optimasi rute dapat mengurangi biaya operasional dan waktu pengiriman, yang berdampak langsung pada kepuasan pelanggan (Wiyanti, 2013). Pada sektor transportasi, optimasi rute dapat mengurangi kemacetan dan emisi karbon, berkontribusi pada keberlanjutan lingkungan (Suryani et al., 2019). Dalam penjadwalan, optimasi rute memungkinkan alokasi sumber daya yang lebih baik dan peningkatan produktivitas.

Salah satu masalah terkenal dalam optimasi rute adalah *Travelling Salesman Problem* (TSP), di mana tujuan utamanya adalah menemukan rute terpendek yang mengunjungi setiap kota tepat satu kali dan kembali ke kota asal (Aristi, 2014). Salah satu algoritma yang sering digunakan untuk mempercepat pencarian solusi

pada masalah *Traveling Salesman Problem* (TSP) adalah algoritma heuristik. Algoritma tersebut sering kali memberikan solusi yang cukup baik dalam waktu lebih cepat dibandingkan dengan pencarian solusi secara menyeluruh, karena mengurangi kemungkinan solusi yang perlu dievaluasi (Yulianto & Setiawan, 2018).

Cheapest Insertion Heuristic (CIH) adalah salah satu algoritma heuristik yang efektif untuk menyelesaikan masalah TSP (Yulianto & Setiawan, 2018). algoritma CIH bekerja dengan membangun sebuah tur (perjalanan) dengan menambahkan titik-titik baru secara bertahap ke dalam rute yang sudah ada. Proses ini dilakukan dengan memilih titik baru serta sisi yang akan disisipkan untuk menghasilkan nilai penyisipan minimum, sehingga bobot total rute yang dihasilkan menjadi seminimal mungkin (Saleh et al., 2015).

Keunggulan algoritma CIH terletak pada kemampuannya menentukan titik yang akan disisipkan, baik yang berada di luar tur yang sedang dibangun maupun sisi yang sudah ada dalam tur. Algoritma ini dapat memilih titik-titik optimal untuk dimasukkan, sehingga menghasilkan rute yang efisien dan terstruktur (Kusrini & Istiyanto, 2007). Kelemahan algoritma CIH terletak pada penentuan dua titik awal, yang dapat membuat hasil total jarak rute bervariasi. Dalam penelitian yang dilakukan oleh Dayanti (2024), untuk memperoleh rute terpendek melalui algoritma CIH diperlukan 2256 percobaan, menunjukkan bahwa waktu eksekusi meningkat seiring bertambahnya jumlah titik yang harus dikunjungi.

Tabu Search merupakan algoritma metaheuristik yang diterapkan untuk mencari solusi optimal dalam konteks optimasi kombinatorial. Algoritma ini mengeksplorasi solusi tetangga dan menggunakan daftar tabu sebagai mekanisme

untuk mencegah pengulangan solusi yang telah dieksplorasi sebelumnya. Daftar tabu ini menyimpan informasi solusi atau atribut tertentu yang telah dikunjungi, sehingga algoritma dapat menghindari eksplorasi ulang solusi yang sama selama rentang waktu tertentu. Dengan cara ini, *Tabu Search* efektif dalam mencegah terjebak pada siklus solusi lokal dan memperluas pencarian ke area solusi baru (Wang et al., 2017).

Keunggulan utama dari *Tabu Search* terletak pada kemampuannya dalam meningkatkan kualitas solusi melalui strategi intensifikasi dan diversifikasi. Strategi intensifikasi bertujuan untuk mengeksplorasi lebih dalam area sekitar solusi terbaik yang telah ditemukan, sehingga memungkinkan perbaikan solusi secara optimal. Sementara itu, strategi diversifikasi berfungsi untuk memperluas ruang pencarian dengan mengeksplorasi area yang belum dijelajahi, sehingga meningkatkan peluang menemukan solusi yang lebih baik (Sitorus et al., 2014).

Penelitian oleh (Ramadhania & Rani, 2021) menunjukkan bahwa kombinasi algoritma Genetika dan *Tabu Search* (GA-TS) menghasilkan solusi yang lebih optimal dibandingkan penggunaan algoritma Genetika tunggal. Hasil tersebut menunjukkan bahwa pendekatan kombinasi efektif dalam mempercepat pencapaian solusi optimal dengan jarak tempuh yang lebih pendek. Sementara itu, penelitian oleh Sembiring et al. (2023) membuktikan bahwa penerapan Algoritma *Tabu Search* dengan solusi awal dari algoritma *Nearest Neighbor* berhasil mengurangi jarak tempuh distribusi hanya dengan satu iterasi. Penelitian oleh Sari (2013) juga menyimpulkan bahwa kombinasi algoritma Genetika dan *Tabu Search* dalam penjadwalan personalia rumah sakit lebih efisien dibandingkan penggunaan algoritma tunggal.

Berdasarkan hasil dari penelitian-penelitian sebelumnya tersebut, pendekatan penggabungan algoritma menjadi penting dalam menghasilkan solusi yang optimal. Oleh karena itu, dalam penelitian ini, algoritma *Cheapest Insertion Heuristic* (CIH) digunakan sebagai pembentuk solusi awal, kemudian dioptimalkan menggunakan algoritma *Tabu Search*. Pendekatan optimasi ini diharapkan mampu meningkatkan kualitas solusi serta efisiensi waktu komputasi.

Hemat adalah sikap bijaksana dalam menggunakan sumber daya seperti uang, waktu, dan energi, dengan menghindari pemborosan dan memaksimalkan manfaat dari setiap pengeluaran. Gaya hidup hemat sering dikaitkan dengan hidup sederhana dan memprioritaskan kebutuhan daripada keinginan. Dalam pandangan Islam, hemat juga sangat dianjurkan sebagai sikap pertengahan antara kikir dan boros. Islam mengajarkan umatnya untuk menggunakan harta dengan cermat dan tidak berlebihan, serta berbagi dengan orang lain jika ada kelebihan. Sebagai mana dalam satu ayat Al-Quran yang menekankan pentingnya hidup hemat adalah surah Al-Isra' ayat 26-27 (Kemenag, 2024), yang berbunyi:

وَاٰتِ ذَا الْقُرْبٰى حَقَّه وَالْمِسْكِيْنَ وَابْنَ السَّبِيْلِ وَلَا تُبَذِّرْ تَبْذِيْرًا (٢٦) اِنَّ الْمُبَذِّرِيْنَ كَانُوْٓا اِخْوَانَ الشَّيٰطِيْنِ ۗوَكَانَ الشَّيْطٰنُ لِرَبِّه كَفُوْرًا (٢٧)

"*Berikanlah kepada kerabat dekat haknya, (juga kepada) orang miskin, dan orang yang dalam perjalanan. Janganlah kamu menghambur-hamburkan (hartamu) secara boros. Sesungguhnya para pemboros itu adalah saudara-saudara setan dan setan itu sangat ingkar kepada Tuhannya*."

Tafsir surah Al-Isra' ayat 26-27 menyampaikan bahwa perilaku boros atau mubazir merupakan tindakan yang mendekati sifat setan (Izzaturrahim et al., 2020). Perilaku ini tidak hanya melanggar ketentuan Allah, tetapi juga menyerupai karakter setan yang selalu ingkar terhadap Allah (Ali & Rusmana, 2021). Pemborosan berarti menggunakan harta atau sumber daya secara tidak bermanfaat,

dan hal ini dapat menimbulkan kerugian baik bagi individu maupun masyarakat. Para ulama dalam penafsiran ayat ini menekankan bahwa mubazir mencakup tindakan mengeluarkan harta tanpa hak yang tepat atau menggunakan sumber daya secara berlebihan melebihi kebutuhan yang wajar (Izzaturrahim et al., 2020).

Berdasarkan pemaparan di atas, penelitian ini bertujuan untuk meningkatkan kualitas solusi algoritma *Cheapest Insertion Heuristic* (CIH) dengan optimalisasi menggunakan algoritma *Tabu Search*. Dengan pendekatan ini, diharapkan solusi yang dihasilkan lebih optimal dalam menyelesaikan permasalahan pencarian rute terpendek.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang ada pada latar belakang, rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut:

1. Bagaimana implementasi algoritma *Cheapest Insertion Heuristic* (CIH) dalam pencarian rute terpendek?
2. Bagaimana implementasi optimasi algoritma CIH menggunakan algoritma *Tabu Search* dalam pencarian rute terpendek?
3. Seberapa besar pengaruh penerapan algoritma *Tabu Search* terhadap kualitas solusi yang dihasilkan oleh algoritma CIH?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Mengimplementasikan algoritma *Cheapest Insertion Heuristic* (CIH) dalam menyelesaikan permasalahan pencarian rute terpendek.

2. Mengoptimalkan solusi yang diperoleh dari algoritma CIH menggunakan algoritma *Tabu Search*.
3. Menguji pengaruh penerapan algoritma *Tabu Search* terhadap kualitas solusi yang dihasilkan oleh algoritma CIH dalam menyelesaikan permasalahan pencarian rute terpendek.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah :

1. Manfaat Akademis
   a. Menambah wawasan dan pengetahuan di bidang algoritma optimasi, khususnya dalam penerapan CIH dan *Tabu Search*.
   b. Menjadi referensi bagi penelitian selanjutnya yang berkaitan dengan masalah optimasi rute.
2. Manfaat bagi peneliti

   Memperdalam pemahaman tentang algoritma heuristik, khususnya *Cheapest Insertion Heuristic* (CIH) dan *Tabu Search*.

1.5 Batasan Masalah

Batasan masalah yang akan ditetapkan pada penelitian ini adalah sebagai berikut:

1. Data yang digunakan pada penelitian ini adalah data titik lokasi pengambilan retribusi parkir di Kecamatan Sukun yang diperoleh dari Dinas Perhubungan Kota Malang pada bulan Juli tahun 2024, yang digunakan untuk uji coba.

2. Pada data penelitian ini, titik mewakili lokasi jalan, di mana jalan yang dimaksud merupakan jalan dari lokasi titik parkir berada.
3. Penelitian ini hanya akan mempertimbangkan jarak antar titik pada data uji coba, jarak diperoleh dari *Google Maps*.

BAB II

KAJIAN TEORI

2.1 Optimasi

Optimasi merupakan suatu proses sistematis yang bertujuan untuk mencari solusi terbaik dalam suatu sistem, dengan mempertimbangkan kendala dan faktor yang mempengaruhinya (Dully et al., 2022). Optimasi juga dapat diartikan sebagai ukuran keberhasilan suatu aktivitas dalam memenuhi kebutuhan yang telah dirancang. Dalam matematika, optimasi merupakan cabang ilmu yang fokus pada upaya untuk menentukan nilai minimum atau maksimum dari suatu fungsi, probabilitas, atau berbagai parameter lainnya dalam beragam situasi (Ayuwati, 2019). Optimasi dapat diterapkan dalam berbagai konteks, termasuk dalam kehidupan sehari-hari, seperti mengatur waktu, mengelola sumber daya, hingga meningkatkan efisiensi kerja. Proses optimasi melibatkan analisis yang mendalam terhadap variabel-variabel yang terlibat serta pemilihan metode yang tepat untuk mencapai tujuan.

Menurut Kamus Besar Bahasa Indonesia (KBBI), istilah "optimasi" berasal dari kata "optimal" dan merujuk pada proses pencarian nilai terbaik atau paling menguntungkan. Pemahaman ini bergantung pada konteks serta karakteristik suatu sistem. Dengan demikian, optimasi dapat diartikan sebagai upaya untuk menemukan solusi paling efektif dalam menyelesaikan permasalahan. Menurut Saputro et al. (2015), optimasi adalah proses pemilihan solusi terbaik dari sekumpulan alternatif yang ada dengan mematuhi sejumlah batasan yang diberikan. Sebagai contoh, pada masalah penentuan rute untuk mengunjungi sejumlah kota,

terdapat berbagai alternatif rute yang dapat dipertimbangkan sebagai solusi. Oleh karena itu, optimasi dalam berbagai bidang, baik dalam ilmu matematika, ilmu komputer, maupun bisnis, bertujuan untuk meningkatkan efisiensi dan efektivitas proses. Azzahra (2024) menambahkan bahwa solusi optimal dicapai ketika suatu proses atau sistem memenuhi kriteria atau standar yang telah ditetapkan. Oleh karena itu, optimasi tidak hanya fokus pada pencarian hasil terbaik, tetapi juga memastikan bahwa solusi yang diperoleh sesuai dengan batasan dan tujuan sistem yang dianalisis.

Berdasarkan berbagai pandangan tersebut, dapat disimpulkan bahwa optimasi adalah proses pencarian solusi terbaik dari sekumpulan alternatif yang tersedia dengan mempertimbangkan kriteria atau batasan tertentu. Tujuan utama optimasi adalah mencapai kondisi yang paling menguntungkan, baik dalam bentuk maksimisasi keuntungan, minimisasi biaya, maupun pencapaian efisiensi dalam suatu sistem.

2.2 Graf

2.2.1 Definisi Graf

Graf adalah representasi logis dari suatu kejadian, peristiwa, proses, atau berbagai hal lain yang saling berhubungan (Wibisono, 2008). Dalam konteks aktivitas sehari-hari, graf berfungsi untuk merepresentasikan berbagai struktur yang ada. Tujuan utamanya adalah untuk memvisualisasikan objek-objek tersebut sehingga lebih mudah dipahami (Siang, 2004). Beberapa contoh graf yang umum ditemui dalam aktivitas sehari-hari meliputi struktur organisasi, peta, rangkaian listrik, diagram alir untuk pemilihan mata kuliah, dan lain sebagainya.

Dalam teori graf, suatu graf G adalah pasangan himpunan $G = (V, E)$, di mana V adalah himpunan titik-titik dan E adalah himpunan sisi-sisi yang menghubungkan beberapa pasangan titik dalam V. Dengan kata lain, graf G terdiri dari dua himpunan V dan E, di mana V merepresentasikan objek-objek yang disebut titik (*vertex*), dan E merepresentasikan sisi (*edge*) yang menghubungkan dua titik dalam V (Chartrand et al., 2015). Setiap titik dalam graf dapat diberi nama atau label dengan huruf seperti $(a, b, c, d, \dots)$, angka seperti $(1, 2, 3, \dots)$, atau simbol tertentu.

2.2.2 Jenis-Jenis Graf

Jenis-jenis graf dalam teori graf diklasifikasikan berdasarkan karakteristik hubungan antar simpul dan sifat dari setiap sisi yang menghubungkannya. Menurut (Chartrand et al., 2015), berikut adalah beberapa jenis graf yang umum dipelajari dalam matematika:

1. Graf sederhana

   Graf sederhana adalah graf tak berarah yang tidak memiliki gelang (*loop*) atau sisi ganda antara dua titik yang sama. Setiap sisi pada graf sederhana hanya menghubungkan dua titik berbeda, dan untuk setiap pasangan titik, hanya terdapat satu sisi yang menghubungkannya.

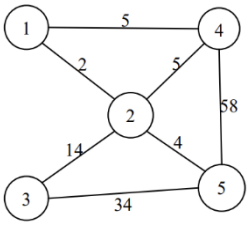
2. Graf lengkap

   Graf lengkap adalah graf sederhana di mana setiap pasangan simpul berbeda terhubung oleh tepat satu sisi. Jika graf lengkap memiliki n titik, maka ia mengandung tepat $\frac{n(n-1)}{2}$ sisi, sehingga seluruh titik pada graf ini saling

berhubungan langsung. Notasi untuk graf lengkap dengan n titik biasanya dituliskan sebagai K_n.

3. Graf Berbobot (*Weighted Graph*)

Graf berbobot merupakan graf di mana setiap sisinya memiliki nilai numerik tertentu yang disebut bobot atau harga dari sisi tersebut. Bobot ini sering kali merepresentasikan metrik tertentu, seperti jarak, biaya, atau waktu, tergantung pada konteks penggunaan graf tersebut. Sebagai contoh, berikut adalah contoh dari graf berbobot:



Gambar 2.1 Graf Berbobot dengan 5 titik dan 7 Sisi

Graf yang ditunjukkan pada gambar 2.1 adalah graf berbobot dengan lima titik dan tujuh sisi. Titiknya adalah 1, 2, 3, 4, dan 5. Bobot pada sisi yang menghubungkan titik 1 dan 2 adalah 2, sedangkan bobot pada sisi yang menghubungkan titik 1 dan 4 adalah 5, dan seterusnya (Saputra, 2013).

4. Graf Berarah (*Directed Graph* atau *Digraph*)

Graf berarah, atau sering disebut sebagai *digraph*, adalah suatu struktur matematis yang terdiri dari dua himpunan, yaitu himpunan titik $V(D)$ yang bersifat terhingga dan tak kosong, serta himpunan busur atau sisi berarah $E(D)$ yang berupa pasangan berurutan dari elemen-elemen dalam $V(D)$. Notasi umum untuk menyatakan sebuah graf berarah adalah $D = (V, E)$. Dalam graf berarah, setiap busur memiliki arah tertentu, yang menunjukkan hubungan dari satu titik ke titik lainnya. Sebagai

contoh, jika a dan b adalah titik, maka pasangan berurutan (a, b) menunjukkan adanya busur dari titik a ke titik b.

2.2.3 Istilah Dasar Graf Berarah

Menurut (Chartrand et al., 2015), dalam graf berarah terdapat beberapa istilah yang sering digunakan untuk menjelaskan elemen-elemen dan konsep-konsep dasar yang membentuk struktur graf berarah. Berikut adalah istilah-istilah tersebut:

1. Bertetangga (*Adjacent*)

   Dalam graf berarah, dua titik disebut bertetangga jika terdapat busur yang menghubungkan keduanya. Jika busur tersebut mengarah dari a ke b, maka titik a disebut tetangga keluar dari titik b, dan titik b disebut tetangga masuk dari titik a. Busur tersebut juga disebut bersisian (*incident)* dengan kedua titik, yaitu a dan b. Relasi bertetangga ini mencerminkan adanya koneksi langsung antara dua titik dalam suatu arah tertentu, yang menjadi dasar struktur hubungan dalam graf berarah.

2. Jalan (*Walk*)

   Jalan adalah urutan dari titik $a_0, a_1, a_2, \dots, a_k$, di mana setiap pasangan berurutan (a_i, a_{i+1}) merupakan busur dalam graf berarah tersebut. Dalam jalan, titik maupun busur yang terlibat tidak harus berbeda, sehingga memungkinkan adanya pengulangan pada titik dan busur.

3. Jejak (*Trails*)

   Jejak adalah jalan yang tidak memiliki pengulangan pada busur. Dalam jejak, setiap busur hanya muncul satu kali, meskipun masih dimungkinkan adanya pengulangan titik.

4. Lintasan Berarah (*Directed Path*)

   Lintasan berarah adalah jalan yang tidak hanya menghindari pengulangan busur, tetapi juga memastikan bahwa setiap titik hanya dilalui satu kali.

5. Siklus Berarah (*Directed Cycle*) atau Sirkuit Berarah (*Directed Circuit*)

   Siklus atau sirkuit merupakan lintasan berarah yang dimulai dan diakhiri pada titik yang sama.

6. Terhubung (*Connected*)

   Dalam sebuah graf berarah dikatakan terhubung (*connected* atau *weakly connected*) apabila graf dasar dari graf berarah tersebut adalah graf terhubung. Graf dasar dari suatu graf berarah diperoleh dengan mengabaikan arah pada setiap sisi dalam digraf tersebut. Selain itu, graf berarah disebut sangat terhubung (*strong* atau *strongly connected*) apabila, untuk setiap pasangan titik u dan v di dalam digraf, terdapat lintasan dari u ke v serta lintasan dari v ke u.

2.3 Rute Terpendek

Rute terpendek merupakan salah satu konsep dasar dalam teori graf yang berkaitan dengan pencarian lintasan minimum antara dua atau lebih titik dalam suatu graf berbobot. Graf berbobot adalah struktur yang terdiri dari titik dan sisi, di mana setiap sisi diberi nilai tertentu yang disebut bobot. Bobot ini dapat

merepresentasikan berbagai aspek seperti jarak, waktu, atau biaya, bergantung pada konteks penggunaannya. Dalam pencarian rute terpendek, tujuan utamanya adalah menemukan lintasan dengan jumlah bobot terkecil dari titik asal menuju titik tujuan (Dayanti, 2024). Rute terpendek secara konseptual didefinisikan sebagai lintasan sederhana tanpa pengulangan titik dengan total bobot paling minimum dari titik awal ke titik akhir dalam graf berbobot (Esanata, 2019). Selain itu, persoalan rute terpendek dapat diklasifikasikan ke dalam beberapa kategori utama, yaitu lintasan terpendek antara dua titik tertentu (*pair shortest path*), lintasan terpendek dari satu titik ke seluruh titik lainnya (*single source shortest path*), lintasan terpendek antara semua pasangan titik (*all pairs shortest path*), serta lintasan terpendek yang melalui titik-titik tertentu (*intermediate shortest path*) (Mukti & Mulyono, 2018).

2.3.1 *Traveling Salesman Problem*

Traveling Salesman Problem (TSP) pertama kali dikemukakan oleh William Rowan Hamilton, matematikawan asal Irlandia dan Thomas Penyngton, matematikawan Inggris pada abad ke-18. TSP diakui sebagai salah satu masalah optimasi kombinatorial yang terkenal dan telah banyak dipelajari hingga saat ini di dalam teori graf. Masalah ini berfokus pada pencarian rute optimal dengan menentukan urutan kunjungan ke semua lokasi, di mana setiap lokasi dikunjungi tepat satu kali sebelum kembali ke lokasi awal (Wiyanti, 2013). Tujuannya adalah menghasilkan rute kunjungan yang efisien dan lengkap, dimulai dari titik awal, kemudian mengunjungi setiap titik lainnya satu per satu, dan akhirnya kembali ke lokasi awal.

Secara matematis, masalah *Traveling Salesman Problem* (TSP) dapat diselesaikan sebagai sebuah *cycle*. Dalam *cycle* ini, setiap tempat diwakili sebagai titik, dan jalan antar tempat diwakili sebagai sisi dengan bobot tertentu yang menunjukkan jarak antar titik yang terhubung. Tujuan utama dari TSP adalah mencari sirkuit Hamilton, yaitu jalur tertutup yang melalui setiap titik tanpa pengulangan, dengan bobot total minimum pada sebuah graf terhubung. Menurut (Wiyanti, 2013), TSP dikelompokkan menjadi 2 jenis, yaitu:

1. TSP asimetris, di mana bobot dari titik 1 ke titik 2 berbeda dengan bobot dari titik 2 ke titik 1. Jika terdapat n titik, maka ukuran ruang pencarian adalah $\frac{n!}{n} = (n-1)!$ kemungkinan jalur.
2. TSP simetris, di mana bobot dari titik 1 ke titik 2 sama dengan bobot dari titik 2 ke titik 1. Jika terdapat n titik, maka ukuran ruang pencarian adalah $\frac{n!}{2n} = \frac{(n-1)!}{2}$ kemungkinan jalur.

2.4 Algoritma *Cheapest Insertion Heuristic* (CIH)

Dalam upaya menemukan solusi cepat untuk masalah *Traveling Salesman Problem* (TSP), teknik heuristik sering digunakan. Teknik ini efektif karena membatasi jumlah solusi yang perlu dipertimbangkan, sehingga proses pencarian menjadi lebih cepat. Untuk masalah TSP, algoritma penyisipan termasuk salah satu pendekatan heuristik yang efisien. Di antara berbagai metode algoritma penyisipan, metode *Cheapest Insertion Heuristic* (CIH) terbukti cukup efektif dan akan menjadi fokus utama dalam penelitian ini (Yulianto & Setiawan, 2018).

Metode *Cheapest Insertion Heuristic* (CIH) menciptakan rute dengan memulai dari rangkaian titik-titik kecil yang memiliki bobot terkecil. Proses ini

dilakukan dengan menambahkan titik-titik baru secara berurutan. Pemilihan titik baru ini diiringi dengan pemilihan sisi terbaik, yang memungkinkan penemuan jarak penyisipan yang terkecil. Selanjutnya, titik baru ini dimasukkan di antara dua titik yang sudah menjadi sisi terpilih.

Menurut (Wiyanti, 2013), algoritma *Cheapest Insertion Heuristic* (CIH) bekerja dengan langkah-langkah berikut dalam bentuk yang lebih sederhana:

1. Penelusuran Awal: Memulai dengan menentukan dua titik awal (i) dan (j).
2. Membentuk *Subtour* Awal: Membentuk subtour awal yang terdiri dari dua titik tersebut dalam bentuk siklus (*cycle*) yaitu (i, j, i).
3. Penyisipan Titik: Menambahkan titik baru ke dalam *subtour* dengan mencari posisi optimal berdasarkan peningkatan nilai jarak terkecil. Nilai sisipan dihitung menggunakan formula:

   $$C_{i,k} + C_{k,j} - C_{j,i}$$

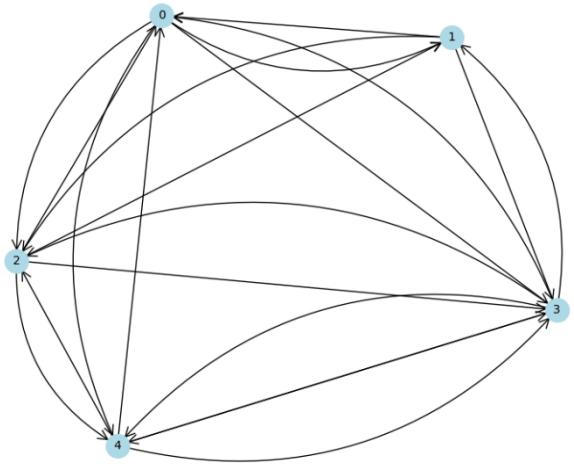
   di mana:

   - $C_{i,k}$ yaitu jarak dari titik i ke titik k
   - $C_{k,j}$ yaitu jarak dari titik k ke titik j
   - $C_{j,i}$ yaitu jarak asli dari titik j ke titik i sebelum penyisipan

   Sehingga diperoleh *subtour* baru yaitu (i, k, j, i).
4. Iterasi: Proses penyisipan titik diulang hingga semua titik tergabung dalam *subtour*. Setiap iterasi mencari posisi terbaik untuk penyisipan berikutnya agar jarak total tetap minimum. Ketika seluruh titik telah tergabung dalam *subtour*, maka *subtour* tersebut memenuhi kriteria sebagai rute yang optimal dalam algoritma CIH

Dengan cara ini, algoritma CIH membangun rute secara efisien, memastikan bahwa titik-titik baru dimasukkan tanpa meningkatkan total jarak secara signifikan.

Berikut contoh penerapan algoritma *cheapest insertion heuristic* pada sebuah kasus. Seorang pegawai perusahaan logistik bertugas untuk mengirim barang di beberapa lokasi. Pegawai tersebut ingin mengoptimalkan perjalanan sehingga memperoleh total jarak tempuh minimum. Kendaraan harus memulai dari kantor pusat (lokasi 0) dan mengunjungi setiap lokasi sekali sebelum kembali ke kantor pusat. Setiap lokasi tersebut harus dikunjungi tepat satu kali dengan tujuan jarak tempuh perjalanan merupakan jarak tempuh paling minimum.



Gambar 2.2 Graf Perjalanan

Gambar 2.2 menggambarkan perjalanan seorang pegawai dalam bentuk graf berarah. Lokasi-lokasi yang akan dikunjungi oleh pegawai tersebut ditunjukkan oleh titik-titik pada graf berarah, sedangkan jalan yang menghubungkan satu lokasi dengan lokasi lainnya ditunjukkan oleh busur pada graf berarah. Arah panah pada busur menunjukkan arah perjalanan dari satu lokasi ke lokasi lainnya. Bobot atau jarak antara titik-titik lokasi tersebut tidak ditampilkan langsung pada gambar, tetapi disajikan dalam Tabel 2.1. Tabel ini memberikan informasi detail mengenai

jarak antara dua lokasi yang dihubungkan oleh sisi-sisi graf, dengan memperhatikan arah perjalanan.

Tabel 2.1 Jarak Antar Lokasi

Dari/Ke (km)	0	1	2	3	4
0	0	12	15	20	22
1	8	0	18	13	24
2	14	17	0	15	11
3	22	16	18	0	15
4	19	23	6	18	0

Tabel 2.1 memperlihatkan jarak antar lokasi yang ditempuh oleh seorang pegawai sebagaimana digambarkan pada Gambar 2.2. Jarak tersebut diukur dalam satuan kilometer. Berikut ini adalah langkah-langkah dalam Algoritma *Cheapest Insertion Heuristic* (CIH):

1. Langkah pertama pada penerapan algoritma CIH dari contoh kasus tersebut yaitu memulai dengan memilih lokasi 0 sebagai titik mulainya perjalanan, dan pilih satu lokasi yang paling dekat dengan lokasi 0 untuk dimasukkan ke dalam rute awal. Berdasarkan Tabel 2.1, lokasi terdekat dari lokasi 0 adalah lokasi 1 dengan jarak 12 km.
2. Kemudian bentuk *subtour* awal antara dua titik awal yaitu $(0, 1, 0)$.
3. Proses penyisipan dilakukan dengan mencari titik baru k yang memiliki bobot sisipan terkecil berdasarkan formula: $C_{i,k} + C_{k,j} - C_{j,i}$. Berikut adalah daftar kemungkinan sisipan berdasarkan Tabel 2.2:

Tabel 2.2 Proses Sisipan Pertama

Busur yang Dipilih untuk Penyisipan	**Titik yang Disisipkan**	**Perhitungan Bobot Sisipan**	**Bobot Sisipan**
$(\mathbf{0,1})$	**2**	$\boldsymbol{c_{0,2} + c_{2,1} - c_{0,1}}$	$\mathbf{15 + 17 - 12 = 20}$
(0,1)	3	$c_{0,3} + c_{3,1} - c_{0,1}$	$20 + 16 - 12 = 24$
(0,1)	4	$c_{0,4} + c_{4,1} - c_{0,1}$	$22 + 23 - 12 = 33$
(1,0)	2	$c_{1,2} + c_{2,0} - c_{1,0}$	$18 + 14 - 8 = 24$
(1,0)	3	$c_{1,3} + c_{3,0} - c_{1,0}$	$13 + 22 - 8 = 27$
(1,0)	4	$c_{1,4} + c_{4,0} - c_{1,0}$	$24 + 19 - 8 = 35$

Berdasarkan pada Tabel 2.2, didapatkan bobot sisipan terkecil adalah 20 km yang diperoleh saat menyisipkan titik 2 di antara titik 0 dan titik 1, sehingga menghasilkan *subtour* barunya adalah $(0, 2, 1, 0)$. Karena masih terdapat titik yang belum termuat dalam *subtour*, maka selanjutnya lakukan pengulangan pada langkah penyisipan.

4. Lakukan pengulangan pada langkah penyisipan menggunakan *subtour* yang baru didapat dari langkah ketiga.

Tabel 2.3 Proses Sisipan Kedua

Busur yang Dipilih untuk Penyisipan	**Titik yang Disisipkan**	**Perhitungan Bobot Sisipan**	**Bobot Sisipan**
(0,2)	3	$c_{0,3} + c_{3,2} - c_{0,2}$	$20 + 18 - 15 = 23$
$(\mathbf{0,2})$	**4**	$\boldsymbol{c_{0,4} + c_{4,2} - c_{0,2}}$	$\mathbf{22 + 6 - 15 = 13}$
(2,1)	3	$c_{2,3} + c_{3,1} - c_{2,1}$	$15 + 16 - 17 = 14$
(2,1)	4	$c_{2,4} + c_{4,1} - c_{2,1}$	$11 + 23 - 17 = 17$
(1,0)	3	$c_{1,3} + c_{3,0} - c_{1,0}$	$13 + 22 - 8 = 27$
(1,0)	4	$c_{1,4} + c_{4,0} - c_{1,0}$	$24 + 19 - 8 = 35$

Berdasarkan pada Tabel 2.3, didapatkan bobot sisipan terkecil adalah 13 km yang diperoleh saat menyisipkan titik 4 di antara titik 0 dan titik 2, sehingga menghasilkan *subtour* barunya adalah $(0, 4, 2, 1, 0)$. Karena masih terdapat titik yang belum termuat dalam *subtour*, maka selanjutnya lakukan pengulangan pada langkah penyisipan.

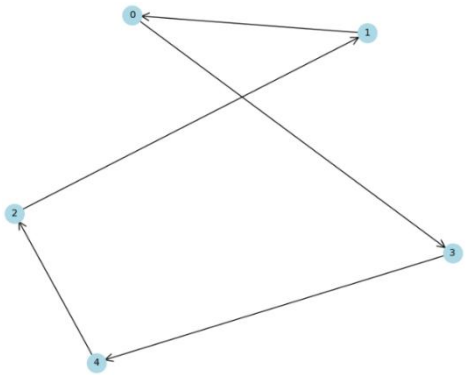
5. Lakukan pengulangan pada langkah penyisipan menggunakan *subtour* yang baru didapat dari langkah ketiga.

Tabel 2.4 Proses Sisipan Ketiga

Busur yang Dipilih untuk Penyisipan	**Titik yang Disisipkan**	**Perhitungan Bobot Sisipan**	**Bobot Sisipan**
$(\mathbf{0,4})$	**3**	$\boldsymbol{c_{0,3} + c_{3,4} - c_{0,4}}$	$\mathbf{20 + 15 - 22 = 13}$
$(4,2)$	3	$c_{4,3} + c_{3,2} - c_{4,2}$	$18 + 18 - 6 = 30$
$(2,1)$	3	$c_{2,3} + c_{3,1} - c_{2,1}$	$15 + 16 - 17 = 14$
$(1,0)$	3	$c_{1,3} + c_{3,0} - c_{1,0}$	$13 + 22 - 8 = 27$

Berdasarkan pada Tabel 2.4, didapatkan bobot sisipan terkecil adalah 13 km yang diperoleh saat menyisipkan titik 3 di antara titik 0 dan titik 4, sehingga menghasilkan *subtour* barunya adalah $(0, 3, 4, 2, 1, 0)$.

Proses penyisipan titik dalam *subtour* berakhir setelah semua titik tergabung dalam rute. Hasil dari proses yang telah dijalankan menunjukkan bahwa rute dengan bobot total minimum yang diperoleh adalah $(0, 3, 4, 2, 1, 0)$ dengan perhitungan dengan total jarak tempuh sebagai berikut: $20 + 15 + 6 + 17 + 8 = 66$ km. Dari langkah-langkah yang telah dijalankan, didapatkan sebuah model graf berarah yang menggambarkan rute terpendek yang diperoleh dengan penerapan Algoritma CIH, sehingga diperoleh jarak minimum perjalanan seorang pegawai perusahaan logistik. Model graf berarah tersebut divisualisasikan dalam Gambar 2.3.



Gambar 2.3 Rute CIH

2.5 Algoritma *Tabu Search*

Algoritma *Tabu Search* adalah salah satu pendekatan metaheuristik yang dikemukakan oleh Fred Glover pada tahun 1986. Algoritma ini dirancang untuk menangani masalah optimasi kombinatorial dengan tujuan menghindari jebakan pada solusi optimal lokal melalui pemanfaatan struktur memori yang disebut Daftar tabu (*Tabu list*) (Naufal Hays, 2017). Dalam penerapannya, Algoritma *Tabu Search* menggunakan metode pencarian lokal (*local search*), di mana setiap iterasi dilakukan dengan menyeleksi solusi terbaik dari himpunan solusi tetangga (*neighborhood*) yang tersedia. Fungsi utama daftar tabu adalah untuk mencatat solusi-solusi yang telah dievaluasi sehingga solusi tersebut tidak dipertimbangkan kembali dalam iterasi berikutnya (Fatmawati et al., 2015).

Algoritma ini terbukti efektif dalam menyelesaikan berbagai permasalahan optimasi berskala besar seperti *Traveling Salesman Problem* (TSP). Dalam proses pencariannya, Algoritma *Tabu Search* menggabungkan beberapa elemen penting, yaitu representasi solusi, fungsi *cost*, neighborhood, *Tabu List*, *aspiration criteria*, dan *long-term memory* (Naufal Hays, 2017 & Ramadhania & Rani, 2021). Dalam aplikasinya, Algoritma *Tabu Search* akan terus melakukan pencarian solusi optimal dengan memperbarui daftar tabu dan mengevaluasi solusi-solusi baru hingga mencapai kriteria pemberhentian yang telah ditetapkan.

Beberapa langkah penting dalam penerapan *Tabu Search* adalah sebagai berikut (Naufal Hays, 2017 & Fatmawati et al., 2015):

1. Inisialisasi Solusi: Memilih solusi awal dan menjadikannya sebagai solusi terbaik sementara.

2. Pencarian Tetangga (*Neighborhood Search*): Membangkitkan solusi tetangga dengan melakukan perubahan kecil pada solusi saat ini, seperti menukar dua titik pada TSP.
3. Pembaruan Daftar Tabu: Solusi yang telah dievaluasi disimpan dalam daftar tabu untuk mencegahnya dipilih kembali dalam iterasi berikutnya.
4. Evaluasi Solusi: Setiap solusi tetangga dievaluasi menggunakan fungsi tujuan, dan solusi dengan nilai terbaik akan dipilih, kecuali jika terdaftar dalam daftar tabu.
5. Kriteria Pemberhentian: Proses iterasi akan dihentikan berdasarkan jumlah iterasi yang telah ditentukan atau ketika tidak ada perbaikan signifikan yang ditemukan.

Algoritma *Tabu Search* memiliki fleksibilitas yang tinggi karena dapat mengingat solusi terbaik sebelumnya dan mencegah terjadinya perulangan yang tidak efisien (Fatmawati et al., 2015).

Dengan menggunakan data yang sama dengan data pada penerapan algoritma CIH, berikut langkah-langkah untuk menentukan solusi rute dengan menerapkan algoritma *Tabu Search*:

1. Langkah pertama adalah memilih solusi awal yang akan digunakan sebagai iterasi pertama (iterasi ke-0). Solusi awal ini ambil secara acak dan dianggap sebagai solusi terbaik sementara. Rute awal yang ambil secara acak yaitu $(0, 2, 1, 4, 3, 0)$ dengan total jarak tempuh sebesar 96 km.

2. Kemudian, Menentukan iterasi berikutnya dan cari solusi alternatif. Solusi alternatif ini didapatkan dengan menukar posisi dua titik lokasi, yang ditunjukkan pada Tabel 2.5.

Iterasi 1

Solusi awal yaitu $(0, 2, 1, 4, 3, 0)$ dengan total jarak tempuh sebesar 96 km

Daftar tabu = { }

Tabel 2.5 Proses Pencarian Rute Alternatif Iterasi Pertama

Pertukaran Posisi	Rute Perjalanan	Jarak Tempuh (km)
(1,2)	$(0, 1, 2, 4, 3, 0)$	81
(2,4)	$(0, 4, 1, 2, 3, 0)$	100
(2,3)	$(0, 3, 1, 4, 2, 0)$	80
(1,4)	$(0, 2, 4, 1, 3, 0)$	84
(1,3)	$(\mathbf{0}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{1}, \mathbf{0})$	**76**
(3,4)	$(0, 2, 1, 3, 4, 0)$	79

Pada iterasi pertama, diperoleh jarak tempuh dengan nilai optimal 76 km yang diperoleh dengan menukar posisi lokasi 1 dengan 3. Jadi, perbaruan daftar tabu-nya adalah {(1,3)}. Selanjutnya, lakukan iterasi kedua dengan metode perhitungan yang sama seperti pada iterasi pertama.

3. Melakukan perulangan pada langkah kedua, untuk perhitungan iterasi kedua yang ditunjukkan pada Tabel 2.6.

Iterasi 2

Solusi awal yaitu $(0, 2, 1, 4, 3, 0)$ dengan total jarak tempuh sebesar 96 km

Daftar tabu = {(1,3)}

Tabel 2.6 Proses Pencarian Rute Alternatif Iterasi Kedua

Pertukaran Posisi	Rute Perjalanan	Jarak Tempuh (km)
(1,3) dan (2,3)	$(0, 3, 2, 4, 1, 0)$	80
(1,3) dan (2,4)	$(0, 4, 3, 2, 1, 0)$	83
(1,3) dan (1,2)	$(\mathbf{0}, \mathbf{1}, \mathbf{3}, \mathbf{4}, \mathbf{2}, \mathbf{0})$	**60**
(1,3) dan (3,4)	$(0, 2, 4, 3, 1, 0)$	68
(1,3) dan (1,4)	$(0, 2, 3, 1, 4, 0)$	89

Pada iterasi kedua, diperoleh jarak tempuh dengan nilai optimal 60 km yang diperoleh dengan menukar posisi lokasi 1 dengan 2. Jadi perbaruan daftar tabu-nya adalah $\{(1,3), (1,2)\}$. Selanjutnya, lakukan iterasi ketiga dengan metode perhitungan yang sama seperti pada iterasi pertama.

4. Lakukan perulangan pada langkah kedua, untuk perhitungan iterasi ketiga yang ditunjukkan pada Tabel 2.7.

Iterasi 3

Solusi awal yaitu $(0, 2, 1, 4, 3, 0)$ dengan total jarak tempuh sebesar 96 km

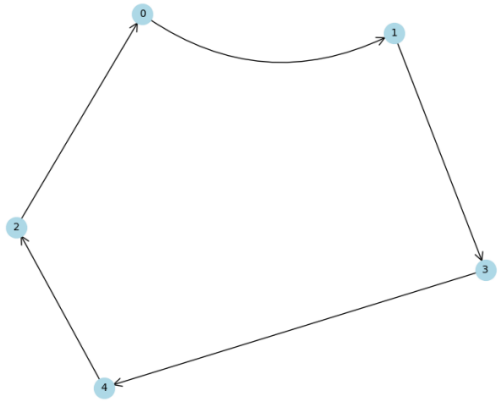
Daftar tabu = $\{(1,3), (1,2)\}$

Tabel 2.7 Proses Pencarian Rute Alternatif Iterasi Ketiga

Pertukaran Posisi	Rute Perjalanan	Jarak Tempuh (km)
(1,3), (1,2) dan (1,4)	$(0, 4, 3, 1, 2, 0)$	88
(1,3), (1,2) dan (3,4)	$(0, 1, 4, 3, 2, 0)$	86
(1,3), (1,2) dan (2,3)	$(0, 1, 2, 4, 3, 0)$	81
(1,3), (1,2) dan (2,4)	$(\mathbf{0}, \mathbf{1}, \mathbf{3}, \mathbf{2}, \mathbf{4}, \mathbf{0})$	**73**

Pada iterasi ketiga, diperoleh jarak tempuh dengan nilai optimal 73 km yang diperoleh dengan menukar posisi 2 dengan 4. Maka, perbaruan daftar tabu-nya adalah $\{(1,3), (1,2), (2,4)\}$.

Berdasarkan perhitungan yang telah dilakukan hingga iterasi ketiga, didapatkan jarak tempuh paling minimumnya adalah 60 km dengan rute $(0, 1, 3, 2, 4, 0)$. Dari langkah-langkah yang telah dijalankan, didapatkan sebuah model graf berarah yang menggambarkan rute terpendek yang diperoleh dengan penerapan Algoritma *Tabu Search*, sehingga diperoleh jarak minimum perjalanan seorang pegawai perusahaan logistik. Model graf berarah tersebut divisualisasikan dalam Gambar 2.4 di bawah ini.



Gambar 2.4 Rute *Tabu Search*

2.6 Pemanfaatan Waktu dalam Perspektif Al-Quran

Pemanfaatan waktu secara umum dapat didefinisikan sebagai cara seseorang mengelola dan menggunakan waktu yang dimiliki untuk mencapai tujuan yang produktif dan bermanfaat. Dalam berbagai aspek kehidupan, baik secara pribadi maupun profesional, optimalisasi penggunaan sangat penting untuk memastikan setiap aktivitas memperoleh hasil yang maksimal. Waktu merupakan sumber daya yang tidak dapat diperbarui, sehingga memanfaatkan waktu dengan baik berarti menghindari hal-hal yang sia-sia dan fokus pada kegiatan yang berkontribusi positif bagi kehidupan seseorang. Dalam pandangan Islam, konsep pemanfaatan waktu juga memiliki dimensi spiritual, di mana manusia diperintahkan

untuk memanfaatkan waktu yang mereka miliki untuk hal-hal yang mendekatkan diri kepada Allah SWT dan bermanfaat bagi sesama manusia.

Dalam Al-Quran, pemanfaatan waktu merupakan salah satu konsep yang sangat ditekankan. Allah SWT menegaskan pentingnya waktu melalui berbagai ayat-Nya, salah satunya adalah dalam surah Al-Ashr. Surah ini, yang terdiri dari hanya tiga ayat, memberikan pelajaran mendalam mengenai bagaimana umat manusia seharusnya memanfaatkan waktu dalam kehidupannya. Allah Subhanahu wa Ta'ala berfirman dalam dalam surah Al-Ashr ayat 1-3 (Kemenag, 2024):

وَالْعَصْرِۙ (١) اِنَّ الْاِنْسَانَ لَفِيْ خُسْرٍۙ (٢) اِلَّا الَّذِيْنَ اٰمَنُوْا وَعَمِلُوا الصّٰلِحٰتِ وَتَوَاصَوْا بِالْحَقِّ ەۙ وَتَوَاصَوْا بِالصَّبْرِ (٣)

"*Demi masa. Sesungguhnya manusia berada dalam kerugian, kecuali orang-orang yang beriman, beramal saleh, saling menasihati dalam kebenaran, dan saling menasihati dalam kesabaran.*"

Tafsir Surah Al-Ashr menegaskan bahwa waktu harus dimanfaatkan untuk hal-hal yang bermanfaat bagi diri sendiri dan orang lain. Waktu yang tidak digunakan dengan baik akan menyebabkan kerugian yang besar (Firdaus, 2022).

Dalam tafsir Quraish Shihab, dijelaskan bahwa waktu lebih berharga daripada harta benda, karena harta yang hilang dapat dicari kembali, sementara waktu yang berlalu tidak dapat dikembalikan (Firdaus, 2022). Pandangan ini mempertegas urgensi memanfaatkan waktu dengan sebaik-baiknya untuk mencapai kesuksesan dunia dan akhirat. Terkait dengan manajemen waktu, Islam memandangnya sebagai bentuk ibadah, di mana manusia harus mengisi waktu mereka dengan amal sholeh dan kegiatan yang mendekatkan diri kepada Allah SWT. Ini sesuai dengan pemahaman dari perspektif pendidikan Islam yang

menyatakan bahwa waktu merupakan sarana bagi manusia untuk beramal, dan amal itu hanya bermakna jika dilakukan dalam waktu yang tepat (Sabri et al., 2022).

Para ulama juga berpendapat bahwa manajemen waktu yang baik melibatkan upaya untuk mengatur segala hal yang terkait dengan kehidupan, baik secara personal maupun dalam konteks organisasi (Sabri et al., 2022). Tafsir Wahbah Zuhayli menekankan bahwa keberhasilan seseorang tidak diukur dari harta atau status, melainkan dari keimanan dan amal sholeh yang dikerjakan dalam bingkai waktu yang telah diberikan (Firdaus, 2022).

Dengan demikian, Surah Al-Ashr dan tafsirnya memberikan pengingat yang kuat bagi umat Islam untuk mengisi waktu mereka dengan kegiatan yang membawa kebaikan, baik secara spiritual maupun sosial. Waktu, yang merupakan nikmat dari Allah, harus dijaga dan dipergunakan sebaik mungkin agar hidup manusia tidak berakhir dengan kerugian.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan pendekatan kuantitatif yang bertujuan untuk menguji pengaruh penerapan algoritma *Tabu Search* terhadap kualitas solusi yang dihasilkan oleh algoritma *Cheapest Insertion Heuristic* (CIH) dalam menyelesaikan permasalahan pencarian rute terpendek. Data yang digunakan berupa jarak antar titik lokasi pengambilan retribusi parkir yang diperoleh dari *Google Maps*, yang menjadi *input* utama dalam algoritma. Dalam konteks ini, Pendekatan kuantitatif ini memungkinkan pengukuran hasil algoritma secara objektif sehingga kualitas solusi dapat dibandingkan dengan jelas. Untuk menganalisis hasil penelitian, digunakan uji-t berpasangan atau *Wilcoxon signed-rank test* apabila data tidak berdistribusi normal sebagai metode analisis statistik, dengan tujuan membandingkan jarak total yang dihasilkan oleh algoritma CIH sebelum dan sesudah dioptimalkan dengan algoritma *Tabu Search*.

3.2 Data dan Sumber Data

Data yang digunakan dalam penelitian ini merupakan data jarak antar titik yang diambil dari *Google Maps*. Data ini mencakup informasi jarak antara titik-titik lokasi pengambilan retribusi parkir di Kecamatan Sukun, Kota Malang. Pengambilan data dilakukan pada bulan Juli 2024 untuk memastikan bahwa informasi yang digunakan adalah aktual dan representatif. Data ini berfungsi

sebagai *input* untuk algoritma dalam proses pencarian rute terpendek serta sebagai data uji coba untuk mengevaluasi kinerja algoritma yang diterapkan.

3.3 Langkah-langkah Penelitian

Langkah-langkah yang diterapkan pada penelitian ini adalah sebagai berikut:

1. Memodelkan Masalah dalam Graf Berarah

   Memodelkan permasalahan sebagai graf berarah berbobot:

   $$D = (V, E)$$

   dengan:

   - $V = \{v_0, v_1, v_2, \dots, v_{n-1}\}$ adalah himpunan titik yang mempresentasikan titik lokasi pengambilan retribusi parkir oleh Dinas Perhubungan Kota Malang di Kecamatan Sukun.
   - $E \subseteq V \times V$ adalah himpunan busur, yaitu pasangan berurutan (v_i, v_j) yang menunjukkan arah dari titik v_i ke v_j.
   - Fungsi bobot dinotasikan dengan $d: E \rightarrow R^+$, yang memetakan setiap busur $(v_i, v_j) \in E$ ke nilai jarak positif antara titik v_i ke v_j.

2. Penerapan Algoritma *Cheapest Insertion Heuristic*

   Algoritma *Cheapest Insertion Heuristic* bekerja dengan langkah-langkah berikut:

   a. Menetapkan titik $v_i \in V$ sebagai titik awal, di mana D graf berarah representasi dari data pada langkah pertama.

   b. Menentukan titik awal $v_j \in V, i \neq j$ untuk membentuk *subtour* awal (*cycle*):

   $$C = (v_i, v_j, v_i)$$

Subtour C adalah sebuah siklus atau *cycle*, yaitu lintasan tertutup yang dimulai dan diakhiri pada titik yang sama.

c. Definisikan $S \subseteq V$ sebagai himpunan titik yang sudah dimasukkan ke dalam *subtour*: $S = \{v_i, v_j\}$.

d. Untuk setiap titik $v_k \in V \backslash S$, dan untuk setiap pasangan berurutan $(v_p, v_q) \in C$, hitung nilai:

$$\Delta d = d(v_p, v_k) + d(v_k, v_q) - d(v_p, v_q)$$

e. Pilih penyisipan simpul v_k yang menghasilkan Δd minimum, lalu perbarui:

$$C \leftarrow (\dots, v_p, v_k, v_q, \dots), \quad S \leftarrow S \cup \{v_k\}$$

f. Ulangi langkah d – e hingga $S = V$. Hasil akhir adalah rute penuh:

$$R_0 = (v_{r_1}, v_{r_2}, \dots, v_{r_n}, v_{r_1})$$

Di mana R_0 adalah solusi awal berbentuk siklus lengkap.

3. Penerapan Optimasi Algoritma CIH Menggunakan Algoritma *Tabu Search*

Setelah memperoleh rute dari penerapan algoritma CIH, langkah selanjutnya adalah mengoptimalkan solusi tersebut menggunakan algoritma *Tabu Search*. Berikut langkah-langkahnya:

a. Menetapkan R_0 sebagai solusi awal pada iterasi ke-$t = 0$:

$$R_t = (v_{r_1}, v_{r_2}, \dots, v_{r_n}, v_{r_1}) \text{ , } t = 0,1,2,3, \dots$$

b. Definisikan daftar tabu sebagai:

$$T_t \subseteq V \times V \text{ , } t = 0,1,2,3, \dots$$

yaitu himpunan pasangan titik yang tidak diizinkan untuk dipertukarkan pada iterasi ke-t. Kapasitas maksimum dari daftar tabu

ditetapkan di awal sebagai parameter $z \in \mathbb{N}$, yang menyatakan jumlah maksimum pasangan yang boleh disimpan dalam T_t, maka $|T_t| \leq z$.

c. Untuk setiap iterasi

i. Bangkitkan solusi tetangga $R' \in N(R_t)$ dari R_t dengan menukar dua titik $v_m, v_n \in V$ yang bukan anggota T_t.

ii. Evaluasi bobot rute tetangga:

$$f(R') = \sum_{i=1}^{n} d(v_{r_i}, v_{r_{i+1}}), \text{ dengan } v_{r_{i+1}} = v_{r_1}$$

iii. Jika $f(R') < f(R_t)$, maka

$$R_{t+1} = R', \;\; T_{t+1} = T_t \cup \{(v_m, v_n)\}$$

iv. Jika $|T_{t+1}| > z$, maka hapus pasangan titik yang paling awal dimasukkan ke dalam daftar tabu.

d. Iterasi dihentikan jika jumlah iterasi mencapai batas maksimum.

e. Solusi akhir R^* dengan nilai $f(R^*)$ minimum dianggap sebagai rute terbaik hasil optimasi.

4. Uji Pengaruh Algoritma *Tabu Search* terhadap kualitas Solusi Algoritma CIH

Langkah ini bertujuan untuk seberapa besar pengaruh dari penerapan algoritma *Tabu Search* terhadap kualitas solusi yang dihasilkan oleh algoritma *Cheapest Insertion Heuristic* (CIH) dalam pencarian rute terpendek. Prosedur pengujian dilakukan melalui tahapan-tahapan berikut:

a. Menyusun tabel perbandingan yang memuat nilai total jarak tempuh dari masing-masing rute yang dihasilkan oleh algoritma CIH dan total jarak tempuh setelah dilakukan optimasi menggunakan algoritma

Tabu Search. Tabel ini juga dilengkapi dengan nilai selisih antara kedua metode pada setiap rute yang diuji.

b. Menghitung selisih jarak tempuh untuk setiap pasangan rute menggunakan rumus berikut:

$$\Delta d_i = d_{CIH,i} - d_{CIH+TS,i}$$

dengan:

- Δd_i menyatakan selisih jarak total pada rute ke-i
- $d_{CIH,i}$ menyatakan total jarak tempuh yang diperoleh dari penerapan algoritma CIH
- $d_{CIH+TS,i}$ menyatakan total jarak tempuh setelah rute tersebut dioptimasi menggunakan algoritma *Tabu Search*.

c. Melakukan uji normalitas terhadap data selisih. Uji normalitas merupakan langkah penting dalam penelitian ini untuk menentukan apakah data selisih tersebut berdistribusi normal. Distribusi normal merupakan asumsi dasar dalam uji statistik parametrik, sehingga hasil uji normalitas akan menentukan metode pengujian statistik yang paling valid untuk digunakan. Metode *Shapiro-Wilk* dipilih karena memiliki sensitivitas tinggi terhadap penyimpangan dari distribusi normal, serta direkomendasikan untuk ukuran sampel kecil hingga sedang ($n < 50$), sebagaimana yang digunakan dalam penelitian ini.

d. Berdasarkan hasil uji normalitas, metode uji hipotesis dipilih sebagai berikut:

   i. Jika data selisih berdistribusi normal, maka digunakan uji-t berpasangan (*paired t-test*) untuk menguji apakah terdapat

perbedaan yang signifikan antara rata-rata jarak tempuh sebelum dan sesudah optimasi.

ii. Jika data selisih tidak berdistribusi normal, maka digunakan uji *Wilcoxon signed-rank test*, yaitu uji non-parametrik yang digunakan untuk menguji nilai selisih dari dua kelompok data berpasangan tanpa asumsi distribusi normal.

e. Hasil dari pengujian statistik tersebut menjadi dasar dalam menarik kesimpulan mengenai efektivitas penerapan algoritma *Tabu Search* dalam meningkatkan kualitas solusi yang diperoleh dari algoritma CIH.

BAB IV

HASIL DAN PEMBAHASAN

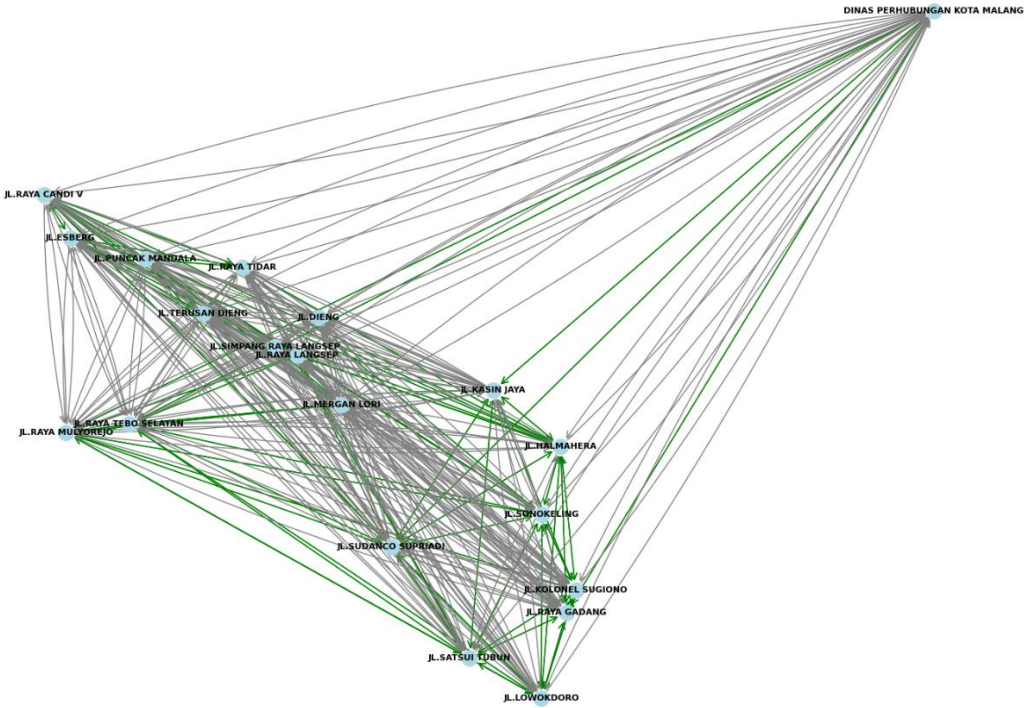
4.1 Memodelkan Masalah dalam Graf Berarah

Memodelkan permasalahan rute pengambilan retribusi parkir oleh Dinas Perhubungan Kota Malang di Kecamatan Sukun dalam bentuk graf berarah. Titik-titik pada graf ini merepresentasikan lokasi titik parkir yang berada di bawah naungan Dinas Perhubungan Kota Malang. Setiap titik dihubungkan oleh busur yang menunjukkan rute potensial untuk pengambilan retribusi parkir. Pemodelan graf ini memfasilitasi visualisasi serta pemahaman yang lebih jelas mengenai struktur dan hubungan antar titik-titik parkir yang ada. Tabel 4.1 menunjukkan data titik parkir di Kecamatan Sukun.

Representasi graf berarah yang menggambarkan titik-titik parkir beserta rute yang mungkin dilalui ditunjukkan dalam Gambar 4.1. Graf tersebut memberikan gambaran mengenai hubungan antar lokasi titik parkir dan rute-rute yang memungkinkan untuk dilalui dalam proses pengambilan retribusi parkir. Pemodelan ini bertujuan untuk mempermudah visualisasi jaringan titik parkir di Kecamatan Sukun, sehingga proses analisis dan perencanaan rute dapat dilakukan secara lebih sistematis dan efisien. Informasi mengenai jarak antar titik parkir tidak dicantumkan langsung dalam gambar, namun disajikan secara rinci pada Tabel 4.2, yang memuat data jarak antar lokasi berdasarkan arah perjalanan dan dinyatakan dalam satuan kilometer.

Tabel 4.1 Data Titik Parkir Kecamatan Sukun

Titik	Lokasi Titik	Longitude	Latitude
v_0	DINAS PERHUBUNGAN KOTA MALANG	112.649940	-7.929576
v_1	JL. DIENG	112.614342	-7.973224
v_2	JL. ESBERG	112.600074	-7.961868
v_3	JL. HALMAHERA	112.628579	-7.991772
v_4	JL. KASIN JAYA	112.624698	-7.983663
v_5	JL. KOLONEL SUGIONO	112.629142	-8.011944
v_6	JL. LOWOKDORO	112.627254	-8.027363
v_7	JL. MERGAN LORI	112.615733	-7.985673
v_8	JL. PUNCAK MANDALA	112.604541	-7.965091
v_9	JL. RAYA CANDI V	112.598321	-7.955917
v_{10}	JL. RAYA GADANG	112.628597	-8.015301
v_{11}	JL. RAYA LANGSEP	112.613175	-7.978666
v_{12}	JL. RAYA MULYOREJO	112.599615	-7.989652
v_{13}	JL. RAYA TEBO SELATAN	112.603611	-7.988647
v_{14}	JL. RAYA TIDAR	112.609802	-7.966275
v_{15}	JL. SATSUI TUBUN	112.623177	-8.021730
v_{16}	JL. SIMPANG RAYA LANGSEP	112.611998	-7.977532
v_{17}	JL. SONOKELING	112.627273	-8.001358
v_{18}	JL. SUDANCO SUPRIADI	112.618482	-8.005944
v_{19}	JL. TERUSAN DIENG	112.607591	-7.972527



Gambar 4.1 Model Masalah dalam Graf Berarah

Tabel 4.2 Data Jarak Antar Titik dalam Satuan Kilometer

Dari/Ke	v_0	v_1	v_2	v_3	⋯	v_{16}	v_{17}	v_{18}	v_{19}
v_0	0	8,4	9,8	8,1	⋯	9,1	9,1	10,0	8,9
v_1	8,3	0	3,0	3,4	⋯	1,4	4,6	4,6	0,9
v_2	9,2	2,5	0	5,8	⋯	3,3	7,1	6,5	1,9
v_3	8,6	4,0	5,8	0	⋯	1,3	3,1	3,1	4,1
⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮
v_{16}	8,8	0,65	2,9	1,3	⋯	0	4,4	3,8	0,95
v_{17}	9,4	4,7	6,9	3,1	⋯	4,3	0	1,9	5,2
v_{18}	10,0	4,1	6,3	3,1	⋯	3,7	1,9	0	4,6
v_{19}	9,0	0,75	1,9	4,1	⋯	0,95	5,4	4,8	0

4.2 Penerapan Algoritma *Cheapest Insertion Heuristic* (CIH)

Untuk menentukan rute terpendek dalam pengambilan retribusi parkir, algoritma CIH diterapkan dengan membangun rute secara bertahap dengan memasukkan titik berdasarkan bobot sisipan terkecil. Berikut ini langkah-langkah penerapannya:

1. Menetapkan titik v_0 sebagai titik awal v_i, karena pegawai yang bertugas untuk mengambil retribusi parkir berangkat dari kantor Dinas Perhubungan Kota Malang.
2. Pilih titik v_j yang memiliki bobot busur terkecil dari titik v_0. berdasarkan Tabel 4.2 diperoleh titik awal v_j adalah v_4. Karena titik i adalah v_0 dan titik v_j adalah v_4, maka bentuk *subtour* awal yaitu (v_0, v_4, v_0). Jadi, $S = \{v_0, v_4\}$.
3. Menyusun tabel yang berisi daftar titik yang dapat disisipkan ke dalam *subtour* beserta nilai bobot sisipannya (Δd). Titik yang akan disisipkan merupakan titik yang belum termuat dalam S, sebagaimana ditampilkan pada Tabel 4.3.

Tabel 4.3 Sisipan Pertama Algoritma CIH

Busur yang Dipilih untuk Penyisipan (v_p, v_q)	Titik yang Disisipkan (v_k)	Bobot Sisipan (Δd)
(v_0, v_4)	v_1	$8{,}4 + 2{,}8 - 7{,}5 = 3{,}7$
(v_0, v_4)	v_2	$9{,}8 + 4{,}6 - 7{,}5 = 6{,}9$
$\boldsymbol{(v_0, v_4)}$	$\boldsymbol{v_3}$	$\mathbf{8{,}1 + 1{,}5 - 7{,}5 = 2{,}1}$
(v_0, v_4)	v_5	$9{,}7 + 4{,}1 - 7{,}5 = 6{,}3$
(v_0, v_4)	v_6	$11 + 5{,}8 - 7{,}5 = 9{,}3$
(v_0, v_4)	v_7	$8{,}6 + 1{,}3 - 7{,}5 = 2{,}4$
(v_0, v_4)	v_8	$9{,}2 + 4 - 7{,}5 = 5{,}7$
(v_0, v_4)	v_9	$10 + 5{,}4 - 7{,}5 = 7{,}9$
(v_0, v_4)	v_{10}	$10 + 4{,}4 - 7{,}5 = 6{,}9$
(v_0, v_4)	v_{11}	$8{,}8 + 1{,}7 - 7{,}5 = 3$
(v_0, v_4)	v_{12}	$12 + 3{,}9 - 7{,}5 = 8{,}4$
(v_0, v_4)	v_{13}	$11 + 3{,}5 - 7{,}5 = 7$
(v_0, v_4)	v_{14}	$8{,}6 + 3{,}4 - 7{,}5 = 4{,}5$
(v_0, v_4)	v_{15}	$11 + 5 - 7{,}5 = 8{,}5$
(v_0, v_4)	v_{16}	$9{,}1 + 2{,}2 - 7{,}5 = 3{,}8$
(v_0, v_4)	v_{17}	$9{,}1 + 2{,}7 - 7{,}5 = 4{,}3$
(v_0, v_4)	v_{18}	$10 + 2{,}9 - 7{,}5 = 5{,}4$
(v_0, v_4)	v_{19}	$8{,}9 + 2{,}9 - 7{,}5 = 4{,}3$
(v_4, v_0)	v_1	$2{,}5 + 8{,}3 - 7{,}5 = 3{,}3$
(v_4, v_0)	v_2	$4{,}7 + 9{,}2 - 7{,}5 = 6{,}4$
(v_4, v_0)	v_3	$1{,}5 + 8{,}6 - 7{,}5 = 2{,}6$
(v_4, v_0)	v_5	$4{,}1 + 10 - 7{,}5 = 6{,}6$
(v_4, v_0)	v_6	$5{,}9 + 12 - 7{,}5 = 10{,}4$
(v_4, v_0)	v_7	$1{,}4 + 8{,}8 - 7{,}5 = 2{,}7$
(v_4, v_0)	v_8	$4{,}1 + 8{,}5 - 7{,}5 = 5{,}1$
(v_4, v_0)	v_9	$5{,}5 + 9{,}2 - 7{,}5 = 7{,}2$
(v_4, v_0)	v_{10}	$4{,}5 + 10 - 7{,}5 = 7$
(v_4, v_0)	v_{11}	$1{,}8 + 9 - 7{,}5 = 3{,}3$
(v_4, v_0)	v_{12}	$3{,}8 + 12 - 7{,}5 = 8{,}3$
(v_4, v_0)	v_{13}	$3{,}4 + 11 - 7{,}5 = 6{,}9$
(v_4, v_0)	v_{14}	$3{,}5 + 8 - 7{,}5 = 4$
(v_4, v_0)	v_{15}	$5{,}0 + 12 - 7{,}5 = 9{,}5$
(v_4, v_0)	v_{16}	$2{,}1 + 8{,}8 - 7{,}5 = 3{,}4$
(v_4, v_0)	v_{17}	$2{,}8 + 9{,}4 - 7{,}5 = 4{,}7$
(v_4, v_0)	v_{18}	$2{,}9 + 10 - 7{,}5 = 5{,}4$
(v_4, v_0)	v_{19}	$3 + 9 - 7{,}5 = 4{,}5$

Berdasarkan pada Tabel 4.3, didapatkan nilai bobot sisipan terkecil adalah 2,1 yang diperoleh saat menyisipkan titik v_3 di antara titik v_0 dan titik v_4,

sehingga menghasilkan *subtour* baru yaitu (v_0, v_3, v_4, v_0). Jadi, $S = \{v_0, v_4, v_3\}$. Karena $S \neq V$, maka selanjutnya lakukan pengulangan pada langkah penyisipan.

4. Melakukan perulangan pada langkah penyisipan dengan menyisipkan titik yang belum termuat dalam S sesuai yang ditunjukkan pada Tabel 4.4.

Tabel 4.4 Sisipan Kedua Algoritma CIH

Busur yang Dipilih untuk Penyisipan (v_p, v_q)	Titik yang Disisipkan (v_k)	Bobot Sisipan (Δd)
(v_0, v_3)	v_1	$8{,}4 + 3{,}4 - 8{,}1 = 3{,}7$
(v_0, v_3)	v_2	$9{,}8 + 5{,}8 - 8{,}1 = 7{,}5$
(v_0, v_3)	v_5	$9{,}7 + 2{,}8 - 8{,}1 = 4{,}4$
(v_0, v_3)	v_6	$11 + 4{,}6 - 8{,}1 = 7{,}5$
(v_0, v_3)	v_7	$8{,}6 + 2{,}7 - 8{,}1 = 3{,}2$
(v_0, v_3)	v_8	$9{,}2 + 5{,}2 - 8{,}1 = 6{,}3$
(v_0, v_3)	v_9	$10 + 6{,}6 - 8{,}1 = 8{,}5$
(v_0, v_3)	v_{10}	$10 + 3{,}2 - 8{,}1 = 5{,}1$
(v_0, v_3)	v_{11}	$8{,}8 + 3{,}2 - 8{,}1 = 3{,}9$
(v_0, v_3)	v_{12}	$12 + 5{,}4 - 8{,}1 = 9{,}3$
(v_0, v_3)	v_{13}	$11 + 5 - 8{,}1 = 7{,}9$
(v_0, v_3)	v_{14}	$8{,}6 + 4{,}6 - 8{,}1 = 5{,}1$
(v_0, v_3)	v_{15}	$11 + 3{,}9 - 8{,}1 = 6{,}8$
(v_0, v_3)	v_{16}	$9{,}1 + 1{,}3 - 8{,}1 = 2{,}3$
(v_0, v_3)	v_{17}	$9{,}1 + 3{,}1 - 8{,}1 = 4{,}1$
(v_0, v_3)	v_{18}	$10 + 3{,}1 - 8{,}1 = 5$
(v_0, v_3)	v_{19}	$8{,}9 + 4{,}1 - 8{,}1 = 4{,}9$
(v_3, v_4)	v_1	$4 + 2{,}8 - 1{,}5 = 5{,}3$
(v_3, v_4)	v_2	$5{,}8 + 4{,}6 - 1{,}5 = 8{,}9$
(v_3, v_4)	v_5	$2{,}8 + 4{,}1 - 1{,}5 = 5{,}4$
(v_3, v_4)	v_6	$4{,}6 + 5{,}8 - 1{,}5 = 8{,}9$
$\boldsymbol{(v_3, v_4)}$	$\boldsymbol{v_7}$	$\mathbf{2 + 1{,}3 - 1{,}5 = 1{,}8}$
(v_3, v_4)	v_8	$5{,}2 + 4 - 1{,}5 = 7{,}7$
(v_3, v_4)	v_9	$6{,}6 + 5{,}4 - 1{,}5 = 10{,}5$
(v_3, v_4)	v_{10}	$3{,}2 + 4{,}4 - 1{,}5 = 6{,}1$
(v_3, v_4)	v_{11}	$3{,}2 + 1{,}7 - 1{,}5 = 3{,}4$
(v_3, v_4)	v_{12}	$5{,}2 + 3{,}9 - 1{,}5 = 7{,}6$
(v_3, v_4)	v_{13}	$4{,}8 + 3{,}5 - 1{,}5 = 6{,}8$
(v_3, v_4)	v_{14}	$4{,}6 + 3{,}4 - 1{,}5 = 6{,}5$
(v_3, v_4)	v_{15}	$3{,}5 + 5 - 1{,}5 = 7$
(v_3, v_4)	v_{16}	$1{,}3 + 2{,}2 - 1{,}5 = 2$
(v_3, v_4)	v_{17}	$3{,}1 + 2{,}7 - 1{,}5 = 4{,}3$

Lanjutan Tabel 4.4

Busur yang Dipilih untuk Penyisipan (v_p, v_q)	Titik yang Disisipkan (v_k)	Bobot Sisipan (Δd)
(v_3, v_4)	v_{18}	$3{,}1 + 2{,}9 - 1{,}5 = 4{,}5$
(v_3, v_4)	v_{19}	$4{,}1 + 2{,}9 - 1{,}5 = 5{,}5$
(v_4, v_0)	v_1	$2{,}5 + 8{,}3 - 1{,}5 = 3{,}3$
(v_4, v_0)	v_2	$4{,}7 + 9{,}2 - 1{,}5 = 6{,}4$
(v_4, v_0)	v_5	$4{,}1 + 10 - 7{,}5 = 6{,}6$
(v_4, v_0)	v_6	$5{,}9 + 12 - 7{,}5 = 10{,}4$
(v_4, v_0)	v_7	$1{,}4 + 8{,}8 - 7{,}5 = 2{,}7$
(v_4, v_0)	v_8	$4{,}1 + 8{,}5 - 7{,}5 = 5{,}1$
(v_4, v_0)	v_9	$5{,}5 + 9{,}2 - 7{,}5 = 7{,}2$
(v_4, v_0)	v_{10}	$4{,}5 + 10 - 7{,}5 = 7$
(v_4, v_0)	v_{11}	$1{,}8 + 9 - 7{,}5 = 3{,}3$
(v_4, v_0)	v_{12}	$3{,}8 + 12 - 7{,}5 = 8{,}3$
(v_4, v_0)	v_{13}	$3{,}4 + 11 - 7{,}5 = 6{,}9$
(v_4, v_0)	v_{14}	$3{,}5 + 8 - 7{,}5 = 4$
(v_4, v_0)	v_{15}	$5{,}0 + 12 - 7{,}5 = 9{,}5$
(v_4, v_0)	v_{16}	$2{,}1 + 8{,}8 - 7{,}5 = 3{,}4$
(v_4, v_0)	v_{17}	$2{,}8 + 9{,}4 - 7{,}5 = 4{,}7$
(v_4, v_0)	v_{18}	$2{,}9 + 10 - 7{,}5 = 5{,}4$
(v_4, v_0)	v_{19}	$3 + 9 - 7{,}5 = 4{,}5$

Berdasarkan pada Tabel 4.4, didapatkan nilai bobot sisipan terkecil adalah 1,8 yang diperoleh saat menyisipkan titik v_7 di antara titik v_3 dan titik v_4, sehingga menghasilkan *subtour* baru yaitu $(v_0, v_3, v_7, v_4, v_0)$. Jadi, $S = \{v_0, v_4, v_3, v_7\}$. Karena $S \neq V$, maka selanjutnya lakukan pengulangan pada langkah penyisipan.

5. Melakukan perulangan pada langkah penyisipan dengan menyisipkan titik yang belum termuat dalam S sesuai yang ditunjukkan pada Tabel 4.5.

Tabel 4.5 Sisipan Ketiga Algoritma CIH

Busur yang Dipilih untuk Penyisipan (v_p, v_q)	Titik yang Disisipkan (v_k)	Bobot Sisipan (Δd)
(v_0, v_3)	v_1	$8{,}4 + 3{,}4 - 8{,}1 = 3{,}7$
(v_0, v_3)	v_2	$9{,}8 + 5{,}8 - 8{,}1 = 7{,}5$
(v_0, v_3)	v_5	$9{,}7 + 2{,}8 - 8{,}1 = 4{,}4$

Lanjutan Tabel 4.5

Busur yang Dipilih untuk Penyisipan (v_p, v_q)	Titik yang Disisipkan (v_k)	Bobot Sisipan (Δd)
(v_0, v_3)	v_6	$11 + 4{,}6 - 8{,}1 = 7{,}5$
(v_0, v_3)	v_8	$9{,}2 + 5{,}2 - 8{,}1 = 6{,}3$
(v_0, v_3)	v_9	$10 + 6{,}6 - 8{,}1 = 8{,}5$
(v_0, v_3)	v_{10}	$10 + 3{,}2 - 8{,}1 = 5{,}1$
(v_0, v_3)	v_{11}	$8{,}8 + 3{,}2 - 8{,}1 = 3{,}9$
(v_0, v_3)	v_{12}	$12 + 5{,}4 - 8{,}1 = 9{,}3$
(v_0, v_3)	v_{13}	$11 + 5 - 8{,}1 = 7{,}9$
(v_0, v_3)	v_{14}	$8{,}6 + 4{,}6 - 8{,}1 = 5{,}1$
(v_0, v_3)	v_{15}	$11 + 3{,}9 - 8{,}1 = 6{,}8$
(v_0, v_3)	v_{16}	$9{,}1 + 1{,}3 - 8{,}1 = 2{,}3$
(v_0, v_3)	v_{17}	$9{,}1 + 3{,}1 - 8{,}1 = 4{,}1$
(v_0, v_3)	v_{18}	$10 + 3{,}1 - 8{,}1 = 5$
(v_0, v_3)	v_{19}	$8{,}9 + 4{,}1 - 8{,}1 = 4{,}9$
(v_3, v_7)	v_1	$4 + 2 - 2 = 4$
(v_3, v_7)	v_2	$5{,}8 + 3{,}9 - 2 = 7{,}7$
(v_3, v_7)	v_5	$2{,}8 + 4 - 2 = 4{,}8$
(v_3, v_7)	v_6	$4{,}6 + 5{,}5 - 2 = 8{,}1$
(v_3, v_7)	v_8	$5{,}2 + 3{,}2 - 2 = 6{,}4$
(v_3, v_7)	v_9	$6{,}6 + 4{,}7 - 2 = 9{,}3$
(v_3, v_7)	v_{10}	$3{,}2 + 4{,}4 - 2 = 5{,}6$
(v_3, v_7)	v_{11}	$3{,}2 + 0{,}9 - 2 = 2{,}1$
(v_3, v_7)	v_{12}	$5{,}2 + 2{,}9 - 2 = 6{,}1$
(v_3, v_7)	v_{13}	$4{,}8 + 2.5 - 2 = 5{,}3$
(v_3, v_7)	v_{14}	$4{,}6 + 2{,}6 - 2 = 5{,}2$
(v_3, v_7)	v_{15}	$3{,}5 + 4{,}5 - 2 = 6$
$\boldsymbol{(v_3, v_7)}$	$\boldsymbol{v_{16}}$	$\mathbf{1{,}3 + 1{,}2 - 2 = 0{,}5}$
(v_3, v_7)	v_{17}	$3{,}1 + 3 - 2 = 4{,}1$
(v_3, v_7)	v_{18}	$3{,}1 + 2{,}4 - 2 = 3{,}5$
(v_3, v_7)	v_{19}	$4{,}1 + 2{,}2 - 2 = 4{,}3$
(v_7, v_4)	v_1	$1{,}7 + 2{,}8 - 1{,}3 = 3{,}2$
(v_7, v_4)	v_2	$3{,}9 + 4{,}6 - 1{,}3 = 7{,}2$
(v_7, v_4)	v_5	$4{,}2 + 4{,}1 - 1{,}3 = 7$
(v_7, v_4)	v_6	$5{,}7 + 5{,}8 - 1{,}3 = 10{,}2$
(v_7, v_4)	v_8	$3{,}2 + 4 - 1{,}3 = 5{,}9$
(v_7, v_4)	v_9	$4{,}7 + 5{,}4 - 1{,}3 = 8{,}8$
(v_7, v_4)	v_{10}	$4{,}6 + 4{,}4 - 1{,}3 = 7{,}7$
(v_7, v_4)	v_{11}	$0{,}95 + 1{,}7 - 1{,}3 = 1{,}35$
(v_7, v_4)	v_{12}	$2{,}9 + 3{,}9 - 1{,}3 = 5{,}5$
(v_7, v_4)	v_{13}	$2{,}5 + 3{,}5 - 1{,}3 = 4{,}7$
(v_7, v_4)	v_{14}	$2{,}7 + 3{,}4 - 1{,}3 = 4{,}8$
(v_7, v_4)	v_{15}	$4{,}7 + 5 - 1{,}3 = 8{,}4$

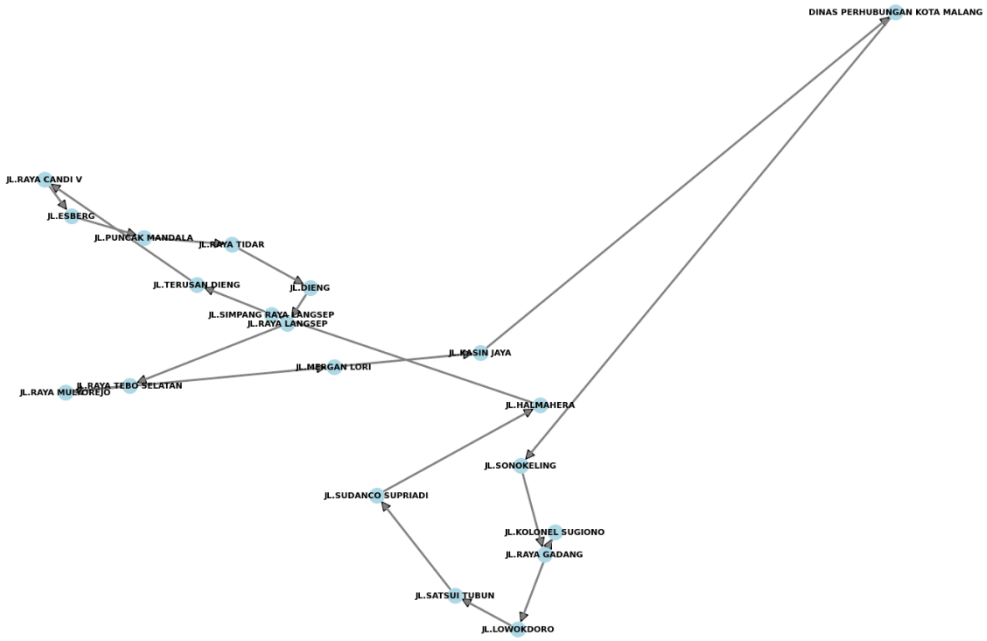
Lanjutan Tabel 4.5

Busur yang Dipilih untuk Penyisipan (v_p, v_q)	Titik yang Disisipkan (v_k)	Bobot Sisipan (Δd)
(v_7, v_4)	v_{16}	$1{,}2 + 2{,}2 - 1{,}3 = 2{,}1$
(v_7, v_4)	v_{17}	$3{,}2 + 2{,}7 - 1{,}3 = 4{,}6$
(v_7, v_4)	v_{18}	$2{,}6 + 2{,}9 - 1{,}3 = 4{,}2$
(v_7, v_4)	v_{19}	$2{,}1 + 2{,}9 - 1{,}3 = 3{,}7$
(v_4, v_0)	v_1	$2{,}5 + 8{,}3 - 7{,}5 = 3{,}3$
(v_4, v_0)	v_2	$4{,}7 + 9{,}2 - 7{,}5 = 6{,}4$
(v_4, v_0)	v_5	$4{,}1 + 10 - 7{,}5 = 6{,}6$
(v_4, v_0)	v_6	$5{,}9 + 12 - 7{,}5 = 10{,}4$
(v_4, v_0)	v_8	$4{,}1 + 8{,}5 - 7{,}5 = 5{,}1$
(v_4, v_0)	v_9	$5{,}5 + 9{,}2 - 7{,}5 = 7{,}2$
(v_4, v_0)	v_{10}	$4{,}5 + 10 - 7{,}5 = 7$
(v_4, v_0)	v_{11}	$1{,}8 + 9 - 7{,}5 = 3{,}3$
(v_4, v_0)	v_{12}	$3{,}8 + 12 - 7{,}5 = 8{,}3$
(v_4, v_0)	v_{13}	$3{,}4 + 11 - 7{,}5 = 6{,}9$
(v_4, v_0)	v_{14}	$3{,}5 + 8 - 7{,}5 = 4$
(v_4, v_0)	v_{15}	$5 + 12 - 7{,}5 = 9{,}5$
(v_4, v_0)	v_{16}	$2{,}1 + 8{,}8 - 7{,}5 = 3{,}4$
(v_4, v_0)	v_{17}	$2{,}8 + 9{,}4 - 7{,}5 = 4{,}7$
(v_4, v_0)	v_{18}	$2{,}9 + 10 - 7{,}5 = 5{,}4$
(v_4, v_0)	v_{19}	$3 + 9 - 7{,}5 = 4{,}5$

Berdasarkan pada Tabel 4.5, didapatkan nilai sisipan terkecil adalah 0,5 yang diperoleh saat menyisipkan titik v_{16} di antara titik v_3 dan titik v_7, sehingga menghasilkan *subtour* baru yaitu $(v_0, v_3, v_{16}, v_7, v_4, v_0)$. Jadi, $S = \{v_0, v_4, v_3, v_7, v_{16}\}$.

Dengan cara serupa, lakukan pengulangan pada langkah penyisipan hingga $S = V$ sehingga terbentuklah rute $(v_0, v_{17}, v_{10}, v_5, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$ yaitu Dinas Perhubungan Kota Malang → Jl. Sonokeling → Jl. Raya Gadang → Jl. Kolonel Sugiono → Jl. Lowokdoro → Jl. Satsui Tubun → Jl. Sudanco Supriadi → Jl. Halmahera → Jl. Simpang Raya Langsep → Jl. Terusan Dieng → Jl. Raya Candi V → Jl. Esberg → Jl. Puncak Mandala → Jl. Raya Tidar → Jl. Dieng → Jl. Raya

Langsep → Jl. Raya Tebo Selatan → Jl. Raya Mulyorejo → Jl. Mergan Lori → Jl. Kasin Jaya → Dinas Perhubungan Kota Malang, dengan total jarak tempuh yang diperoleh adalah $9{,}1 + 1{,}9 + 0{,}4 + 1{,}7 + 1 + 2{,}1 + 3{,}1 + 1{,}3 + 0{,}95 + 2{,}7 + 0{,}8 + 0{,}65 + 0{,}6 + 1{,}2 + 1{,}1 + 2{,}8 + 1{,}1 + 2{,}9 + 1{,}3 + 7{,}5 = 44{,}2$ km.



Gambar 4.2 Visualisasi Rute Algoritma CIH dengan Titik Awal Dinas Perhubungan Kota Malang dan Jl. Kasin Jaya

Dengan menerapkan metode yang serupa, titik awal $\boldsymbol{v_i}$ ditetapkan di Dinas Perhubungan Kota Malang. Setiap titik selain yang telah ditetapkan sebagai titik awal $\boldsymbol{v_i}$ akan dijadikan sebagai titik awal $\boldsymbol{v_j}$. Proses ini melibatkan langkah-langkah untuk mempertimbangkan dan menguji setiap kombinasi titik awal $\boldsymbol{v_i}$ dan $\boldsymbol{v_j}$ hingga ditemukan total jarak tempuh dari rute yang dihasilkan. Total jarak tempuh dari setiap rute yang dihasilkan akan ditampilkan pada Tabel 4.6.

Tabel 4.6 Total Jarak Tempuh Rute yang Dihasilkan Algoritma CIH

No	Titik Awal *i*	Titik Awal *j*	Total Jarak Tempuh Rute (dalam Kilometer)
1	Dinas Perhubungan Kota Malang	Jl.Dieng	42,2
2	Dinas Perhubungan Kota Malang	Jl.Esberg	42,6
3	Dinas Perhubungan Kota Malang	Jl.Halmahera	44,2
4	Dinas Perhubungan Kota Malang	Jl.Kasin Jaya	44,2
5	Dinas Perhubungan Kota Malang	Jl.Kolonel Sugiono	43,8
6	Dinas Perhubungan Kota Malang	Jl.Lowokdoro	43,8
7	Dinas Perhubungan Kota Malang	Jl.Mergan Lori	43,1
8	Dinas Perhubungan Kota Malang	Jl.Puncak Mandala	42,6
9	Dinas Perhubungan Kota Malang	Jl.Raya Candi V	41,9
10	Dinas Perhubungan Kota Malang	Jl.Raya Gadang	43,8
11	Dinas Perhubungan Kota Malang	Jl.Raya Langsep	42,95
12	Dinas Perhubungan Kota Malang	Jl.Raya Mulyorejo	44,05
13	Dinas Perhubungan Kota Malang	Jl.Raya Tebo Selatan	44,05
14	Dinas Perhubungan Kota Malang	Jl.Raya Tidar	42,6
15	Dinas Perhubungan Kota Malang	Jl.Satsui Tubun	43,8
16	Dinas Perhubungan Kota Malang	Jl.Simpang Raya Langsep	42,95
17	Dinas Perhubungan Kota Malang	Jl.Sonokeling	42,45
18	Dinas Perhubungan Kota Malang	Jl.Sudanco Supriadi	44,25
19	Dinas Perhubungan Kota Malang	Jl.Terusan Dieng	45,5

Visualisasi dari 19 rute yang dihasilkan dapat dilihat pada lampiran.

4.3 Penerapan Optimasi Algoritma CIH Menggunakan Algoritma *Tabu Search*

Setelah mendapatkan rute awal dari algoritma CIH, langkah berikutnya adalah mengoptimalkan rute tersebut dengan algoritma *Tabu Search*. Berikut ini langkah-langkahnya:

1. Menetapkan rute dari CIH sebagai solusi awal R_0

   $R_0 = (v_0, v_{17}, v_{10}, v_5, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$ dengan total jarak tempuh sebesar 44,2 km

2. Definisikan daftar tabu $T_0 = \{\}$, dengan kapasitas maksimum sebanyak tiga elemen. Tentukan jumlah iterasi yang diizinkan hanya sebanyak tiga kali.

3. Melakukan proses optimasi rute bertujuan untuk memperbaiki solusi awal yang telah diperoleh sebelumnya dengan membuat tabel yang berisi daftar pertukaran dua titik pada solusi awal. Dua titik yang ditukar merupakan pasangan titik yang belum termuat dalam daftar tabu, sebagaimana ditampilkan pada Tabel 4.7.

   Iterasi 1

   $R_0 = (v_0, v_{17}, v_{10}, v_5, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$ dengan total jarak tempuh sebesar 44,2 km

   $T_0 = \{\}$

Tabel 4.7 Proses Optimasi Rute Iterasi Pertama

Pertukaran Posisi	Rute	Total Jarak Tempuh (dalam Kilometer)
v_{17},v_{10}	$(v_0, v_{10}, v_{17}, v_5, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$	46,3
v_{17},v_5	$(v_0, v_5, v_{10}, v_{17}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$	46,4
⋮	⋮	⋮
v_{17},v_4	$(v_0, v_4, v_{10}, v_5, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_{17}, v_0)$	49
v_{10},v_5	$(v_0, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$	43,5
⋮	⋮	⋮
v_7,v_4	$(v_0, v_{17}, v_{10}, v_5, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_4, v_7, v_0)$	46,6

Berdasarkan Tabel 4.7, pada iterasi pertama diperoleh jarak tempuh rute dengan nilai optimal 43,5 km yang diperoleh dengan menukar posisi titik v_{10} dengan v_5. Jadi, perbaruan daftar tabu $T_1 = \{(v_5, v_{10})\}$. Iterasi pertama menghasilkan rute $R_1 = (v_0, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$.

4. Melakukan pertukaran posisi dua titik yang belum termuat dalam daftar tabu sesuai yang ditunjukkan pada Tabel 4.8.

Iterasi 2

$R_1 = (v_0, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$ dengan total jarak tempuh sebesar 43,5 km

$T_1 = \{(v_5, v_{10})\}$

Tabel 4.8 Proses Optimasi Rute Iterasi Kedua

Pertukaran Posisi	Rute	Total Jarak Tempuh (dalam Kilometer)
v_{17}, v_5	$(v_0, v_5, v_{17}, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$	45,6
v_{17}, v_{10}	$(v_0, v_{10}, v_5, v_{17}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$	46,4
⋮	⋮	⋮
v_9, v_8	$(v_0, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_8, v_2, v_9, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$	43,5
⋮	⋮	⋮
v_7, v_4	$(v_0, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_4, v_7, v_0)$	45,9

Berdasarkan Tabel 4.8, pada iterasi kedua diperoleh jarak tempuh rute dengan nilai optimal 43,5 km yang diperoleh dengan menukar posisi titik v_9 dengan v_8. Jadi, perbaruan daftar tabu $T_2 = \{(v_5, v_{10}), (v_8, v_9)\}$. Iterasi kedua menghasilkan rute $R_2 = (v_0, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_8, v_2, v_9, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$.

5. Melakukan pertukaran posisi dua titik yang belum termuat dalam daftar tabu sesuai yang ditunjukkan pada Tabel 4.9

Iterasi 3

$R_2 = (v_0, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_8, v_2, v_9, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$ dengan total jarak tempuh sebesar 44,2 km

$T_2 = \{(v_5, v_{10}), (v_8, v_9)\}$

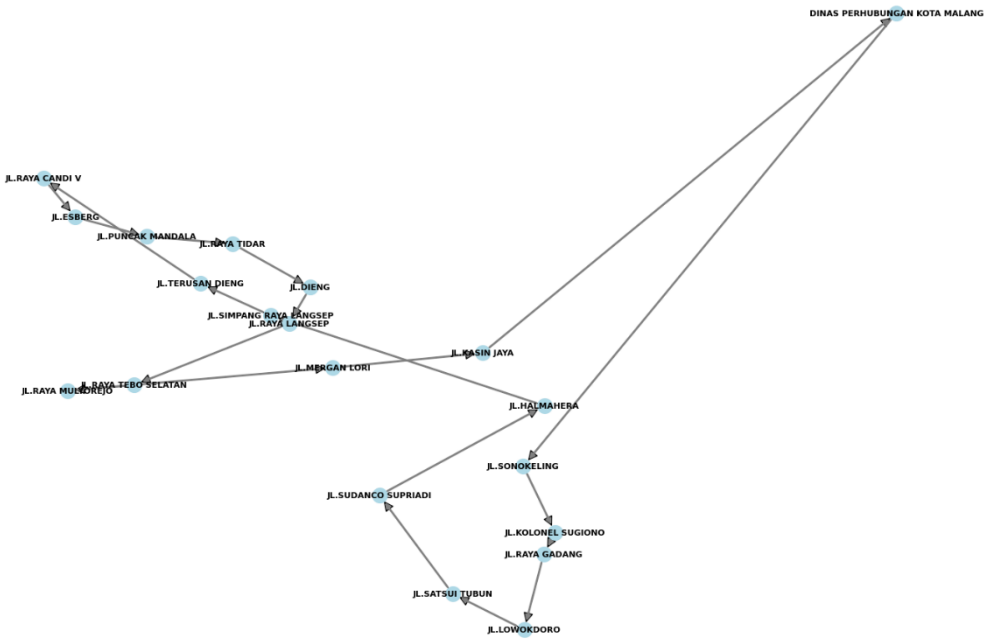
Tabel 4.9 Proses Optimasi Rute Iterasi Ketiga

Pertukaran Posisi	Rute	Total Jarak Tempuh (dalam Kilometer)
v_{17}, v_5	$(v_0, v_5, v_{17}, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_8, v_2, v_9, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$	45,6
v_{17}, v_{10}	$(v_0, v_{10}, v_5, v_{17}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_8, v_2, v_9, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$	46,4
⋮	⋮	⋮
v_{13}, v_{12}	$(v_0, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_8, v_2, v_9, v_{14}, v_1, v_{11}, v_{12}, v_{13}, v_7, v_4, v_0)$	43,5
⋮	⋮	⋮
v_7, v_4	$(v_0, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_8, v_2, v_9, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_4, v_7, v_0)$	45,9

Berdasarkan Tabel 4.9, pada iterasi ketiga diperoleh jarak tempuh rute dengan nilai optimal 43,5 km yang diperoleh dengan menukar posisi titik v_{13} dengan v_{12}. Jadi, perbaruan daftar tabu $T_3 = \{(v_5, v_{10}), (v_8, v_9), (v_{12}, v_{13})\}$. Iterasi ketiga menghasilkan rute $R_3 = (v_0, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_8, v_2, v_9, v_{14}, v_1, v_{11}, v_{12}, v_{13}, v_7, v_4, v_0)$.

Pada iterasi pertama, proses pertukaran dua titik berhasil menghasilkan perbaikan rute dengan total jarak tempuh sebesar 43,5 km. Selanjutnya, pada iterasi kedua dan ketiga, meskipun langkah pertukaran titik tetap dilakukan, tidak diperoleh jarak tempuh yang lebih kecil daripada hasil pada iterasi pertama. Hal ini menunjukkan bahwa solusi optimal telah ditemukan sejak iterasi pertama, dan upaya optimasi lanjutan tidak memberikan peningkatan hasil yang signifikan. Berdasarkan perhitungan yang telah dilakukan hingga iterasi ketiga, didapatkan

jarak tempuh paling minimumnya adalah 43,5 km dengan rute $(v_0, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$ yaitu Dinas Perhubungan Kota Malang → Jl. Sonokeling → Jl. Kolonel Sugiono → Jl. Raya Gadang → Jl. Lowokdoro → Jl. Satsui Tubun → Jl. Sudanco Supriadi → Jl. Halmahera → Jl. Simpang Raya Langsep → Jl. Terusan Dieng → Jl. Raya Candi V → Jl. Esberg → Jl. Puncak Mandala → Jl. Raya Tidar → Jl. Dieng → Jl. Raya Langsep → Jl. Raya Tebo Selatan → Jl. Raya Mulyorejo → Jl. Mergan Lori → Jl. Kasin Jaya → Dinas Perhubungan Kota Malang, yang diperoleh pada iterasi pertama.



Gambar 4.3 Visualisasi Rute Hasil Optimasi CIH menggunakan *Tabu Search*

Sebelum diterapkan pada seluruh rute, proses optimasi dilakukan terlebih dahulu melalui tiga iterasi. Pada iterasi pertama, kedua, dan ketiga, dilakukan pertukaran posisi dua titik untuk mengevaluasi kemungkinan perbaikan terhadap solusi awal hasil algoritma CIH. Setiap iterasi bertujuan untuk memperkecil total jarak tempuh dan menghindari terjebak pada solusi lokal. Dari ketiga iterasi yang

dilakukan, diperoleh hasil optimasi dengan total jarak tempuh yang lebih baik dibandingkan solusi awal. Dengan menerapkan metode yang serupa, setiap rute yang dihasilkan melalui algoritma CIH digunakan sebagai solusi awal. Proses ini melibatkan langkah-langkah untuk mengoptimalkan setiap rute sehingga diperoleh rute baru yang dihasilkan dari penerapan algoritma *Tabu Search*, dengan total jarak tempuh yang diperoleh. Setiap rute yang dihasilkan serta total jarak tempuhnya akan disajikan pada Tabel 4.10.

Tabel 4.10 Rute dan Total Jarak Tempuh dari Optimasi

No	Rute	Total Jarak Tempuh (dalam Kilometer)
1	$(v_0, v_4, v_3, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_{12}, v_{13}, v_7, v_{11}, v_{16}, v_1, v_{19}, v_9, v_2, v_8, v_{14}, v_0)$	41,2
2	$(v_0, v_4, v_3, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_{12}, v_{13}, v_7, v_{11}, v_{16}, v_1, v_{19}, v_{14}, v_8, v_2, v_9, v_0)$	40,9
3	$(v_0, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$	43,5
4	$(v_0, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_7, v_4, v_0)$	43,5
5	$(v_0, v_4, v_3, v_{16}, v_{19}, v_8, v_2, v_9, v_{14}, v_1, v_{11}, v_7, v_{13}, v_{12}, v_{18}, v_{17}, v_5, v_6, v_{15}, v_{10}, v_0)$	43,8
6	$(v_0, v_4, v_3, v_{16}, v_{19}, v_8, v_2, v_9, v_{14}, v_1, v_{11}, v_7, v_{13}, v_{12}, v_{18}, v_{17}, v_5, v_6, v_{15}, v_{10}, v_0)$	43,8
7	$(v_0, v_9, v_2, v_8, v_{14}, v_{19}, v_1, v_{11}, v_{16}, v_3, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_{12}, v_{13}, v_7, v_4, v_0)$	42,15
8	$(v_0, v_4, v_3, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_{12}, v_{13}, v_7, v_{11}, v_{16}, v_1, v_{19}, v_{14}, v_8, v_2, v_9, v_0)$	40,9
9	$(v_0, v_4, v_3, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_{12}, v_{13}, v_7, v_{11}, v_{16}, v_1, v_{19}, v_{14}, v_8, v_2, v_9, v_0)$	40,9
10	$(v_0, v_4, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_7, v_{13}, v_{12}, v_{18}, v_{17}, v_5, v_6, v_{15}, v_{10}, v_0)$	43,8
11	$(v_0, v_3, v_4, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_{12}, v_{13}, v_7, v_{11}, v_{16}, v_1, v_{19}, v_{14}, v_8, v_2, v_9, v_0)$	41,2
12	$(v_0, v_3, v_4, v_{13}, v_{12}, v_{18}, v_{15}, v_6, v_{10}, v_5, v_{17}, v_7, v_{11}, v_{16}, v_1, v_{19}, v_{14}, v_8, v_2, v_9, v_0)$	42,3
13	$(v_0, v_3, v_4, v_{13}, v_{12}, v_{18}, v_{15}, v_6, v_{10}, v_5, v_{17}, v_7, v_{11}, v_{16}, v_1, v_{19}, v_{14}, v_8, v_2, v_9, v_0)$	42,3
14	$(v_0, v_4, v_3, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_{12}, v_{13}, v_7, v_{11}, v_{16}, v_1, v_{19}, v_{14}, v_8, v_2, v_9, v_0)$	40,9
15	$(v_0, v_4, v_3, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_7, v_{13}, v_{12}, v_{18}, v_{17}, v_5, v_{15}, v_6, v_{10}, v_0)$	43,8
16	$(v_0, v_3, v_4, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_{12}, v_{13}, v_7, v_{11}, v_{16}, v_1, v_{19}, v_{14}, v_8, v_2, v_9, v_0)$	41,2
17	$(v_0, v_{17}, v_5, v_{10}, v_6, v_{15}, v_{18}, v_{12}, v_{13}, v_7, v_{11}, v_1, v_{14}, v_8, v_2, v_9, v_{19}, v_{16}, v_3, v_4, v_0)$	41,75
18	$(v_0, v_3, v_4, v_7, v_{16}, v_{19}, v_9, v_2, v_8, v_{14}, v_1, v_{11}, v_{13}, v_{12}, v_{18}, v_{15}, v_6, v_{10}, v_5, v_{17}, v_0)$	43,4
19	$(v_0, v_{14}, v_8, v_2, v_9, v_{19}, v_1, v_{11}, v_{16}, v_7, v_{13}, v_{12}, v_{18}, v_{17}, v_{10}, v_{15}, v_6, v_5, v_3, v_4, v_0)$	43,45

Visualisasi dari 19 rute yang dihasilkan dapat dilihat pada lampiran.

4.4 Uji Pengaruh Algoritma *Tabu Search* terhadap Kualitas Solusi Algoritma CIH

Untuk menguji pengaruh algoritma *Tabu Search* terhadap kualitas solusi algoritma CIH, langkah pertama adalah membuat tabel yang menampilkan hasil jarak total yang diperoleh dari penerapan algoritma CIH dan hasil jarak total setelah penerapan algoritma *Tabu Search* pada rute yang dihasilkan oleh algoritma CIH. Tabel ini akan memberikan gambaran awal mengenai perubahan jarak total yang terjadi setelah algoritma *Tabu Search* diterapkan dan menampilkan hasil perhitungan selisih jarak total sebelum dan setelah penerapan algoritma *Tabu Search*. Selisih jarak total (d_i) dihitung dengan rumus:

$$d_i = d_{CIH,i} - d_{CIH+TS,i}$$

di mana:

$i = 1, 2, 3, \ldots, n$

$d_{CIH,i}$ = Jarak total yang diperoleh dari algoritma CIH ke-i

$d_{CIH+TS,i}$ = Jarak total yang diperoleh dari algoritma CIH yang dioptimasi menggunakan *Tabu Search* ke-i

Tabel 4.11 menunjukkan perbandingan jarak total dalam satuan kilometer yang diperoleh dari algoritma CIH sebelum dan sesudah penerapan algoritma *Tabu Search* pada beberapa rute. Dari tabel tersebut, dapat dilihat selisih jarak total yang terjadi setelah optimasi algoritma CIH menggunakan algoritma *Tabu Search* untuk setiap rute yang diujikan.

Tabel 4.11 Perbandingan Jarak Total dalam Satuan Kilometer

No	CIH (d_{CIH})	CIH + *Tabu Search* (d_{CIH+TS})	Selisih (d)
1	42,2	41,2	1
2	42,6	40,9	1,7
3	44,2	43,5	0,7
4	44,2	43,5	0,7
5	43,8	43,8	0
6	43,8	43,8	0
7	43,1	42,15	0,95
8	42,6	40,9	1,7
9	41,9	40,9	1
10	43,8	43,8	0
11	42,95	41,2	1,75
12	44,05	42,3	1,75
13	44,05	42,3	1,75
14	42,6	40,9	1,7
15	43,8	43,8	0
16	42,95	41,2	1,75
17	42,45	41,75	0,7
18	44,25	43,4	0,85
19	45,5	43,45	2,05

Langkah selanjutnya dalam analisis data adalah melakukan uji normalitas terhadap data selisih. Uji normalitas dilakukan untuk mengetahui apakah data selisih hasil algoritma *Cheapest Insertion Heuristic* (CIH) dan hasil optimasi algoritma CIH menggunakan algoritma *Tabu Search* berdistribusi normal. Pada penelitian ini, uji normalitas dilakukan menggunakan metode *Shapiro-Wilk*, karena metode ini lebih sesuai untuk ukuran sampel kecil hingga sedang ($n < 50$).

Hipotesis yang digunakan dalam uji *Shapiro-Wilk* adalah sebagai berikut:

H_0: Data selisih berdistribusi normal (*p-value* $\geq 0{,}05$)

H_1: Data selisih tidak berdistribusi normal (*p-value* $< 0{,}05$)

Pengujian dilakukan menggunakan bantuan perangkat lunak Python. yang menggunakan pustaka scipy.stats untuk menghitung nilai statistik dan p-value secara akurat. Berdasarkan hasil pengujian, diperoleh nilai statistik *Shapiro-Wilk*

sebesar $0{,}8688$ dan *p-value* sebesar $0{,}0138$. Karena *p-value* < $0{,}05$, maka dapat disimpulkan bahwa H_0 ditolak, sehingga data selisih tidak berdistribusi normal.

Dengan demikian, karena asumsi normalitas tidak terpenuhi, maka uji statistik parametrik seperti uji-t berpasangan tidak dapat digunakan. Oleh karena itu, analisis statistik pada penelitian ini dilanjutkan dengan menggunakan uji non-parametrik, yaitu *Wilcoxon Signed Rank Test*, yang lebih sesuai untuk data berpasangan yang tidak berdistribusi normal.

Uji *Wilcoxon Signed Rank Test* digunakan untuk mengetahui apakah terdapat perbedaan yang signifikan antara dua sampel berpasangan. Dalam konteks penelitian ini, pengujian dilakukan untuk mengetahui apakah terdapat perbedaan antara total jarak tempuh rute yang dihasilkan oleh algoritma *Cheapest Insertion Heuristic* (CIH) dan total jarak tempuh setelah dilakukan optimasi menggunakan algoritma *Tabu Search*.

Adapun hipotesis yang digunakan dalam uji *Wilcoxon Signed Rank Test* adalah sebagai berikut:

H_0: Tidak terdapat perbedaan yang signifikan antara hasil Algoritma CIH dan hasil setelah optimasi menggunakan *Tabu Search*.

H_1: Terdapat perbedaan yang signifikan antara hasil Algoritma CIH dan hasil setelah optimasi menggunakan *Tabu Search*.

Karena tujuan dari penelitian ini adalah untuk mengetahui apakah algoritma *Tabu Search* dapat mengurangi total jarak tempuh secara signifikan, maka digunakan uji satu arah, di mana hipotesis alternatif menyatakan bahwa hasil optimasi lebih baik daripada sebelumnya.

Uji ini dilakukan dengan menghitung selisih antara dua pengukuran pada pasangan data, yaitu selisih antara jarak tempuh dari CIH dan CIH + *Tabu Search*. Nilai absolut dari selisih ini kemudian diberi peringkat. Setelah itu, dilakukan pemisahan peringkat berdasarkan tanda selisih: bertanda positif dan bertanda negatif. Jumlah peringkat dari selisih bertanda positif disebut W^+, dan dari selisih bertanda negatif disebut W^-. Nilai statistik uji *Wilcoxon* dihitung sebagai $W = \min(W^+, W^-)$.

Berdasarkan data pada Tabel 4.11, diketahui bahwa seluruh nilai selisih bernilai positif atau nol, yang menunjukkan bahwa penerapan algoritma *Tabu Search* secara konsisten memberikan solusi yang sama atau lebih baik dibandingkan algoritma CIH. Pasangan data yang memiliki nilai selisih sama dengan nol ($d_i = 0$) dikeluarkan dari perhitungan, sehingga hanya 15 pasangan data yang valid untuk dianalisis. Karena seluruh nilai selisih yang tersisa bertanda positif, maka W^+ adalah jumlah dari seluruh peringkat, dan $W^- = 0$. Dengan demikian, diperoleh $W = \min(W^+, W^-) = 0$.

Berdasarkan tabel distribusi kritis *Wilcoxon* untuk $n = 15$ dan taraf signifikansi $\alpha = 0{,}05$ (uji satu arah), diperoleh nilai W_{tabel} sebesar 25. Karena $W = 0 < W_{tabel} = 25$, maka H_0 ditolak. Artinya, terdapat perbedaan yang signifikan antara total jarak tempuh rute yang dihasilkan oleh algoritma CIH dan total jarak tempuh setelah dilakukan optimasi dengan algoritma *Tabu Search*. Dengan kata lain, optimasi yang dilakukan menggunakan *Tabu Search* memberikan peningkatan kualitas solusi secara signifikan, karena mampu menurunkan total jarak tempuh.

Hasil ini menunjukkan bahwa optimasi algoritma CIH menggunakan algoritma *Tabu Search* merupakan pendekatan yang efektif dalam menyelesaikan masalah pencarian rute terpendek. Optimasi ini mampu mengatasi keterbatasan solusi awal yang dihasilkan oleh CIH dan memberikan hasil yang lebih optimal dari segi total jarak tempuh.

4.5 Efisiensi Waktu dalam Algoritma Optimasi

Efisiensi waktu merupakan elemen penting dalam berbagai disiplin ilmu, termasuk dalam penyelesaian permasalahan optimasi rute dalam teori graf. Pada konteks pencarian rute terpendek, efisiensi waktu berkaitan dengan kemampuan algoritma untuk menemukan solusi optimal dengan jumlah langkah minimal. Prinsip ini selaras dengan ajaran Islam mengenai pemanfaatan waktu secara bijaksana, sebagaimana disebutkan dalam QS. Al-Ashr Ayat 1-3. Ayat tersebut mengingatkan umat manusia akan pentingnya waktu sebagai sumber daya yang tidak dapat diperbarui, sehingga harus dimanfaatkan secara produktif. Dalam konteks teori graf, efisiensi waktu tercermin dalam pemilihan algoritma yang sistematis untuk menghindari eksplorasi jalur yang tidak perlu dan memastikan bahwa setiap langkah dalam pencarian rute membawa perbaikan mendekati solusi optimal.

Pada penelitian ini, algoritma *Cheapest Insertion Heuristic* (CIH) digunakan sebagai pendekatan awal untuk membangun solusi dengan cepat. Algoritma CIH bekerja dengan menambahkan titik ke dalam rute yang sudah ada berdasarkan kriteria jarak tambahan terkecil. Akan tetapi, pendekatan ini memiliki keterbatasan karena hanya berfokus pada optimalisasi lokal, tanpa

mempertimbangkan kemungkinan perbaikan secara global. Sebagai ilustrasi, metode ini dapat menghasilkan rute yang kurang optimal karena mengabaikan struktur keseluruhan solusi.

Untuk mengatasi keterbatasan algoritma CIH, algoritma *Tabu Search* diterapkan guna meningkatkan efisiensi pencarian solusi. Algoritma ini menggunakan daftar tabu untuk menghindari eksplorasi ulang terhadap solusi yang telah diuji sebelumnya. Dengan demikian, algoritma ini memperluas ruang pencarian solusi dan meningkatkan peluang menemukan solusi yang lebih baik secara global. Optimasi algoritma CIH menggunakan algoritma *Tabu Search* terbukti mampu meningkatkan efisiensi waktu pencarian tanpa mengorbankan kualitas solusi.

Konsep efisiensi waktu dalam algoritma ini sejalan dengan ajaran Islam yang menekankan pentingnya manajemen waktu yang baik. QS. Al-Ashr mengajarkan bahwa waktu harus diisi dengan aktivitas yang bernilai, bermanfaat, dan menghindari pemborosan. Dalam optimasi rute, prinsip ini tercermin dalam upaya menghindari eksplorasi jalur yang tidak perlu dan memastikan bahwa setiap langkah dalam pencarian rute membawa perbaikan mendekati solusi optimal.

Hasil penelitian menunjukkan bahwa optimasi algoritma CIH menggunakan algoritma *Tabu Search* menghasilkan rute dengan total jarak tempuh lebih pendek dibandingkan ketika hanya menggunakan CIH. Hal ini menunjukkan bahwa strategi optimasi yang diterapkan mampu meningkatkan efisiensi langkah pencarian solusi tanpa mengurangi kualitas hasil akhir. Dengan demikian, penelitian ini menggambarkan bagaimana konsep manajemen waktu yang diajarkan dalam Islam

dapat diintegrasikan ke dalam strategi ilmiah, khususnya dalam penyelesaian permasalahan optimasi.

Dengan demikian, penelitian ini berkontribusi pada pengembangan solusi untuk permasalahan optimasi rute dalam teori graf melalui penerapan optimasi algoritma yang efisien. Selain itu, penelitian ini juga menjadi bukti bagaimana nilai-nilai Islam tentang efisiensi waktu dapat diimplementasikan secara konkret dalam pendekatan ilmiah untuk menghasilkan solusi yang lebih efektif dan bermanfaat.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai optimasi algoritma *Cheapest Insertion Heuristic* (CIH) dengan algoritma *Tabu Search* dalam pencarian rute terpendek, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Implementasi algoritma *Cheapest Insertion Heuristic* (CIH) diterapkan untuk menyelesaikan permasalahan pencarian rute terpendek pada 20 titik lokasi pengambilan retribusi parkir di wilayah Kecamatan Sukun, Kota Malang. Algoritma ini bekerja dengan membentuk *subtour* awal dari dua titik, kemudian menyisipkan titik-titik lainnya secara bertahap berdasarkan nilai jarak sisipan minimum. Proses ini menghasilkan sebanyak 19 rute dalam bentuk *cycle* yang dimulai dan diakhiri pada titik, yaitu kantor Dinas Perhubungan Kota Malang. Hasil implementasi menunjukkan bahwa CIH mampu menghasilkan solusi awal dengan total jarak tempuh yang relatif efisien. Namun demikian, solusi tersebut masih bersifat sub-optimal secara matematis karena ruang pencarian solusi tidak dijelajahi secara menyeluruh dan algoritma cenderung berhenti pada solusi lokal.
2. solusi awal yang diperoleh dari algoritma CIH kemudian dioptimasi menggunakan algoritma *Tabu Search* melalui pendekatan pertukaran posisi antar titik (*swap*). Optimasi dilakukan terhadap 19 rute dengan menjalankan algoritma *Tabu Search* sebanyak tiga iterasi untuk masing-masing rute. Pada setiap iterasi, solusi tetangga dibangkitkan dan dievaluasi berdasarkan

total jarak tempuh, dengan pengecualian terhadap solusi-solusi yang masuk dalam daftar tabu. Hasil optimasi menunjukkan bahwa perbaikan solusi signifikan umumnya telah tercapai pada iterasi pertama atau kedua, dan dalam beberapa kasus tetap konsisten hingga iterasi ketiga. Hal ini menunjukkan bahwa algoritma Tabu Search efektif dalam memperbaiki solusi awal dari CIH, bahkan dengan jumlah iterasi yang terbatas, sehingga efisien dari sisi waktu dan sumber daya komputasi.

3. Untuk mengukur secara kuantitatif pengaruh penerapan algoritma *Tabu Search* terhadap kualitas solusi yang dihasilkan oleh algoritma *Cheapest Insertion Heuristic* (CIH), dilakukan analisis statistik terhadap total jarak tempuh sebelum dan sesudah proses optimasi. Metode pengujian yang digunakan adalah *Wilcoxon signed-rank test*, yang dipilih berdasarkan hasil uji normalitas yang menunjukkan bahwa data selisih tidak terdistribusi normal. Berdasarkan hasil analisis, diperoleh nilai statistik uji *Wilcoxon* sebesar $W = 0$, yang lebih kecil daripada nilai kritis tabel ($W_{tabel} = 25$) pada taraf signifikansi $\alpha = 0{,}05$. Hasil ini mengindikasikan bahwa terdapat perbedaan yang signifikan antara total jarak tempuh rute yang dihasilkan oleh algoritma CIH dan total jarak tempuh setelah dilakukan optimasi menggunakan algoritma *Tabu Search*. Dari 19 rute yang dianalisis, sebanyak 15 rute mengalami penurunan total jarak tempuh setelah dioptimasi menggunakan *Tabu Search*, dengan selisih maksimum sebesar $2{,}05$ km. Berdasarkan temuan tersebut, dapat disimpulkan bahwa integrasi algoritma Tabu Search terhadap solusi awal CIH memberikan peningkatan kualitas solusi secara signifikan dalam konteks optimasi rute terpendek.

5.2 Saran untuk Penelitian Lanjutan

Penelitian lanjutan sebaiknya dilakukan pada *dataset* yang lebih besar dan kompleks untuk menguji kinerja algoritma *Cheapest Insertion Heuristic* (CIH) dan *Tabu Search* dalam kondisi yang lebih menantang. Pengujian ini akan memberikan pemahaman yang lebih mendalam mengenai skalabilitas dan efisiensi algoritma dalam menangani masalah rute terpendek pada jaringan yang lebih besar. Selain itu, penelitian mendalam mengenai optimasi parameter algoritma *Tabu Search*, seperti ukuran *tabu list*, intensitas eksplorasi, dan kriteria penghentian, perlu dilakukan untuk meningkatkan kinerja dan kualitas solusi yang dihasilkan. Eksperimen dengan berbagai konfigurasi parameter dapat membantu menemukan pengaturan yang paling optimal, sehingga menghasilkan solusi yang lebih efisien dan akurat.

DAFTAR PUSTAKA

Ali, M. H., & Rusmana, D. (2021). Konsep Mubazir dalam Al-Qur'an: Studi Tafsir Maudhu'i. *Jurnal Riset Agama*, *1*(3), 11–29. https://doi.org/10.15575/jra.v1i3.15065

Aristi, G. (2014). Perbandingan Algoritma Greedy, Algoritma Cheapest Insertion Heuristics Dan Dynamic Programming Dalam Penyelesaian Travelling Salesman Problem. *Paradigma*, *XVI*(2), 52–58. https://ejournal.bsi.ac.id/ejurnal/index.php/paradigma/article/view/779

Ayuwati, S. (2019). *Optimasi Biaya Produksi Pada Industri Kerajinan Kain Tenun Buton Menggunakan Metode Karush-Kuhn-Tucker* [Universitas Hasanuddin]. https://repository.unhas.ac.id/id/eprint/5624/2/19_H11113502%28FILEminimizer%29 ... ok 1-2.pdf

Azzahra, S. (2024). Optimalisasi Pengelolaan Sarana dan Prasarana Dalam Meningkatkan Kualitas Pembelajaran Siswa. *Al-Fikri: Jurnal Hukum, Politik dan Bisnis*, *2*(2), 152–165. https://www.jurnal.zarilgapari.org/index.php/faizi/article/view/127

Chartrand, G., Lesniak, L., & Zhang, P. (2015). Graphs & digraphs. In A. Boggess & K. Rosen (Ed.), *Graphs and Digraphs* (Sixth). CRC Press Taylor & Francis Group. https://doi.org/10.1201/b19731

Dayanti, V. (2024). *Efektivitas Parameter Algoritma Cheapest Insertion Heuristic (CIH) dalam Menentukan Rute Terpendek Bus Sekolah Gratis Kota Malang* [Universitas Islam Negeri Maulana Malik Ibrahim Malang]. http://etheses.uin-malang.ac.id/65184/

Dully, chrisye kusuma, Widajanti, E., & Sunarsono. (2022). Analisis Linear Programming Untuk Optimasi Laba. *Jurnal Ekonomi dan Kewirausahaan*, *22*(3), 262–270. https://ejurnal.unisri.ac.id/index.php/Ekonomi/article/view/8430

Esanata, C. V. (2019). Penerapan Metode Djikstra Sebagai Penentuan Rute Terpendek Distribusi Pengiriman Kantor Jne Pusat Kabupaten Jombang. *Jurnal Mahasiswa Teknik Informatika*, *3*(1), 79–84. https://ejournal.itn.ac.id/index.php/jati/article/download/582/550.

Fatmawati, Prihandono, B., & Noviani, E. (2015). Penyelesaian Travelling Salesman Problem Dengan Metode Tabu Search. *Buletin Ilmiah Mat. Stat. Dan Terapannya (Bimaster)*, *04* no. *1*(1), 17–24.

Firdaus. (2022). Konsep Manajemen Waktu dalam Surat al-'Ashr (Kajian Semiotika Al-Qur'an). *JIQTA: Jurnal Ilmu Al-Qur'an dan Tafsir*, *1*(1), 1–18.

Izzaturrahim, B. F., Khambali, & Tsaury, A. M. (2020). Implikasi Pendidikan Q.S. Al-Isra Ayat 26-27 Tentang Larangan Tabdzir Terhadap Upaya Menghindari Perilaku Mubazir. *Prosiding Pendidikan Agama Islam*, *6*(17), 36–39. https://karyailmiah.unisba.ac.id/index.php/pai/article/view/22560

Kemenag. (2024). *Qur'an Kemenag*. Lajnah Pentashihan Mushaf Al-Qur'an.

Kholidasari, I., Zein, A. P., & Sundari, S. (2022). Penerapan Metode Shortest Route Problem Untuk Menentukan Rute Distribusi Produk Gas Lpg 3 Kg Dengan Kriteria Minimasi Biaya Transportasi Di Pt.Www. *Industrika : Jurnal Ilmiah Teknik Industri*, *6*(1), 1–10. https://doi.org/10.37090/indstrk.v6i1.585

Kusrini, & Istiyanto, J. E. (2007). Penyelesaian Travelling Salesman Problem Dengan Algoritma Cheapest Insertion Heuristics Dan Basis Data. *Jurnal Informatika*, *8*(2), 109–114. http://puslit2.petra.ac.id/ejournal/index.php/inf/article/view/16775

Mukti, M. R., & Mulyono. (2018). Menentukan Rute Terpendek Dengan Menggunakan Algoritma Floyd-Warshall Dalam Pendistribusian Barang Pada Pt. Rapy Ray Putratama. *KARISMATIKA: Kumpulan Artikel Ilmiah, Informatika, Statistik, Matematika dan Aplikasi*, *4*(1). https://doi.org/10.24114/jmk.v4i1.11857

Naufal Hays, R. (2017). Implementasi Firefly Algorithm - Tabu Search Untuk Penyelesaian Traveling Salesman Problem. *Jurnal Online Informatika*, *2*(1), 42–48. http://join.if.uinsgd.ac.id/index.php/join/article/view/v2i18/51

Ramadhania, S. E., & Rani, S. (2021). Implementasi Kombinasi Algoritma Genetika dan Tabu Search untuk Penyelesaian Travelling Salesman Problem. *Automata*, *2*(1), 99–106. http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/

Sabri, A., Hidayati, H., Hidayat, R., Husna, R., & Putra, E. F. (2022). Pengelolaan Waktu dalam Pendidikan Islam. *Journal on Education*, *5*(1), 1243–1252. https://doi.org/10.31004/joe.v5i1.746

Saleh, K., Helmi, & Prihandono, B. (2015). Penentuan Rute Terpendek Dengan Menggunakan Algoritma Cheapest Insertion Heuristic (Studi Kasus: PT. Wicaksana Overseas International Tbk. Cabang Pontianak). *Buletin Ilmiah Math. Stat. dan Terapannya (Bimaster)*, *04*(3), 295–304. https://jurnal.untan.ac.id/index.php/jbmstr/article/view/11888/11091

Saputra, C. A. (2013). *Aplikasi Graf dalam Rute Pengiriman Barang*. 1–3. https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Makalah2013/MakalahIF2120-2013-027.pdf

Saputro, H. A., Mahmudy, W. F., & Dewi, C. (2015). Implementasi Algoritma Genetika Untuk Optimasi Penggunaan Lahan Pertanian. *Jurnal Mahasiswa PTIIK*, *5*(12), 12. https://r.search.yahoo.com/_ylt=AwrPplStGbNnDAIAEADLQwx.;_ylu=Y29sbwNzZzMEcG9zAzEEdnRpZAMEc2VjA3Ny/RV=2/RE=1741000365/RO=10/RU=http%3A%2F%2Fwww.wayanfm.lecture.ub.ac.id%2Ffiles%2F2015%2F05%2FJurnalSkripsi-2014-2015-015-Harim-Adi-Saputro.pdf/RK=2/RS=YKkJ8v

Sari, E. D. (2013). Optimasi Penjadwalan Personalia Rumah sakit Berbasis AGENT Menggunakan Kombinasi Algoritma Genetika dan Tabu Search [Universitas Islam Negeri Maulana Malik Ibrahim Malang]. In *Optimasi Penjadwalan Personalia Rumah sakit Berbasis AGENT Menggunakan Kombinasi*

Algoritma Genetika dan Tabu Search. http://etheses.uin-malang.ac.id/7548/

Sembiring, A. C., Lumbanntoruan, I. S., & Jufri, H. B. (2023). Optimalisasi Rute Distribusi Menggunakan Algoritma Tabu Search Dan Nearest Neighbor. *Junal Ilmiah Teknik Industri Prima)*, *6*(2), 50–56.

Siang, J. jek. (2004). *Matematika Diskrit dan Aplikasinya Pada Ilmu Komputer* (2 ed.). Andi.

Sitorus, H. M., Juwono, C. P., & Ciputra, P. (2014). Penerapan Algoritma Tabu Search pada Permasalahan Lintasan Keseimbangan Bentuk U Tipe I dengan Waktu Proses Stokastik. *Inasea*, *15*(1), 15–27. https://research.binus.ac.id/publication/DCDCFBA2-686E-44CF-A0EA-0B9AF8EB5859/penerapan-algoritma-tabu-search-pada-permasalahan-lintasan-keseimbangan-bentuk-u-tipe-i-dengan-waktu-proses-stokastik/

Suryani, F., Natadipura, R. K., & Tunafiah, H. (2019). Optimalisasi Rute Perjalanan Sarana Angkutan Umum Terpadu Bogor-Jakarta. *Ikra-ITH Teknologi*, *3*(2), 63–75. http://journals.upi-yai.ac.id/index.php/ikraith-teknologi/article/view/708

Wang, Y., Wu, Q., & Glover, F. (2017). Effective metaheuristic algorithms for the minimum differential dispersion problem. *European Journal of Operational Research*, *258*(3), 829–843. https://doi.org/10.1016/j.ejor.2016.10.035

Wibisono, S. (2008). *Matematika Diskrit* (A. R. Rachmawati (ed.); 2 ed.). Graha Ilmu.

Wiyanti, D. T. (2013). Algoritma Optimasi Untuk Penyelesaian Travelling Salesman Problem. *Jurnal Transformatika*, *11*(1), 1. https://doi.org/10.26623/transformatika.v11i1.76

Yulianto, E., & Setiawan, A. (2018). Optimasi Rute Sales Coverage Menggunakan Algoritma Cheapest Insertion Heuristic Dan Layanan Google Maps Api. *INTERNAL (Information System Journal)*, *1*(1), 39–54. https://doi.org/10.32627/internal.v1i1.326

LAMPIRAN

Lampiran 1 Data Jarak Antar Titik dalam Satuan Kilometer

Dari/Ke	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
v_0	0	8,4	9,8	8,1	7,5	9,7	11,0	8,6	9,2	10,0
v_1	8,3	0	3,0	3,4	2,8	6,0	7,7	2,0	2,1	3,8
v_2	9,2	2,5	0	5,8	4,6	8,1	9,6	3,9	0,65	0,8
v_3	8,6	4,0	5,8	0	1,5	2,8	4,6	2,0	5,2	6,6
v_4	7,5	2,5	4,7	1,5	0	4,1	5,9	1,4	4,1	5,5
v_5	10,0	5,7	7,9	2,8	4,1	0	1,7	4,0	7,3	8,7
v_6	12,0	7,2	9,4	4,6	5,8	1,7	0	5,5	8,8	10,0
v_7	8,8	1,7	3,9	2,7	1,3	4,2	5,7	0	3,2	4,7
v_8	8,5	1,8	0,65	5,2	4,0	7,5	9,0	3,2	0	1,5
v_9	9,2	3,3	0,8	6,6	5,4	8,9	10,0	4,7	1,5	0
v_{10}	10,0	6,1	8,3	3,2	4,4	0,4	1,3	4,4	7,7	9,1
v_{11}	9,0	0,8	3,0	3,2	1,7	5,1	6,6	0,9	2,3	3,8
v_{12}	12,0	4,0	5,3	5,4	3,9	5,5	6,7	2,9	4,6	6,1
v_{13}	11,0	3,6	4,9	5,0	3,5	5,1	6,2	2,5	4,2	5,7
v_{14}	8,0	1,2	1,2	4,6	3,4	6,8	8,4	2,6	0,6	2,0
v_{15}	12,0	6,2	8,4	3,9	5,0	1,7	1,0	4,5	7,8	9,2
v_{16}	8,8	0,65	2,9	1,3	2,2	5,8	7,3	1,2	2,2	3,7
v_{17}	9,4	4,7	6,9	3,1	2,7	1,6	3,3	3,0	6,3	7,7
v_{18}	10,0	4,1	6,3	3,1	2,9	2,0	3,1	2,4	5,7	7,1
v_{19}	9,0	0,75	1,9	4,1	2,9	6,4	7,9	2,2	1,3	2,7

Dari/Ke	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}	v_{18}	v_{19}
v_0	10,0	8,8	12,0	11,0	8,6	11,0	9,1	9,1	10,0	8,9
v_1	6,4	1,1	4,3	3,9	1,4	6,7	1,4	4,6	4,6	0,9
v_2	8,8	3,0	5,6	5,2	1,2	8,5	3,3	7,1	6,5	1,9
v_3	3,2	3,2	5,2	4,8	4,6	3,5	1,3	3,1	3,1	4,1
v_4	4,5	1,8	3,8	3,4	3,5	5,0	2,1	2,8	2,9	3,0
v_5	0,4	5,0	5,5	5,0	6,7	1,7	5,3	1,6	1,9	6,2
v_6	1,3	6,5	6,7	6,2	8,2	1,0	6,8	3,3	3,1	7,7
v_7	4,6	0,95	2,9	2,5	2,7	4,7	1,2	3,2	2,6	2,1
v_8	7,8	2,3	4,9	4,5	0,6	8,0	2,1	6,3	5,7	1,3
v_9	9,3	3,8	6,4	6,0	2,0	9,3	3,5	7,9	7,1	2,7
v_{10}	0	5,4	5,8	5,4	7,1	1,3	5,7	1,9	2,3	6,6
v_{11}	5,5	0	3,2	2,8	1,8	5,5	0,35	4,0	3,4	1,2
v_{12}	6,8	3,2	0	1,1	4,6	5,7	3,5	5,4	3,5	3,6
v_{13}	6,3	2,8	1,1	0	4,2	5,8	3,1	5,5	3,7	3,2
v_{14}	7,2	1,7	4,9	4,5	0	7,2	2,0	5,7	5,2	1,2
v_{15}	1,3	5,5	5,7	5,8	7,2	0	5,8	3,3	2,1	6,7
v_{16}	6,2	0,7	3,2	2,8	1,6	5,9	0	4,4	3,8	0,95
v_{17}	1,9	4,0	5,4	5,5	5,7	3,3	4,3	0	1,9	5,2
v_{18}	2,4	3,4	3,5	3,7	5,1	2,1	3,7	1,9	0	4,6
v_{19}	6,8	1,3	3,9	3,5	1,2	6,9	0,95	5,4	4,8	0

Lampiran 2 Kode Python

Membuat matriks data

```
import numpy as np
matriks_jarak = np.array([
   [0, 8.4, 9.8, 8.1, 7.5, 9.7, 11.0, 8.6, 9.2, 10.0, 10.0,
8.8, 12.0, 11.0, 8.6, 11.0, 9.1, 9.1, 10.0, 8.9],
   [8.3, 0, 3.0, 3.4, 2.8, 6.0, 7.7, 2.0, 2.1, 3.8, 6.4, 1.1,
4.3, 3.9, 1.4, 6.7, 1.4, 4.6, 4.6, 0.9],
   [9.2, 2.5, 0, 5.8, 4.6, 8.1, 9.6, 3.9, 0.65, 0.8, 8.8, 3.0,
5.6, 5.2, 1.2, 8.5, 3.3, 7.1, 6.5, 1.9],
   [8.6, 4.0, 5.8, 0, 1.5, 2.8, 4.6, 2.0, 5.2, 6.6, 3.2, 3.2,
5.2, 4.8, 4.6, 3.5, 1.3, 3.1, 3.1, 4.1],
   [7.5, 2.5, 4.7, 1.5, 0, 4.1, 5.9, 1.4, 4.1, 5.5, 4.5, 1.8,
3.8, 3.4, 3.5, 5.0, 2.1, 2.8, 2.9, 3.0],
   [10.0, 5.7, 7.9, 2.8, 4.1, 0, 1.7, 4.0, 7.3, 8.7, 0.4, 5.0,
5.5, 5.0, 6.7, 1.7, 5.3, 1.6, 1.9, 6.2],
   [12.0, 7.2, 9.4, 4.6, 5.8, 1.7, 0, 5.5, 8.8, 10.0, 1.3, 6.5,
6.7, 6.2, 8.2, 1.0, 6.8, 3.3, 3.1, 7.7],
   [8.8, 1.7, 3.9, 2.7, 1.3, 4.2, 5.7, 0, 3.2, 4.7, 4.6, 0.95,
2.9, 2.5, 2.7, 4.7, 1.2, 3.2, 2.6, 2.1],
   [8.5, 1.8, 0.65, 5.2, 4.0, 7.5, 9.0, 3.2, 0, 1.5, 7.8, 2.3,
4.9, 4.5, 0.6, 8.0, 2.1, 6.3, 5.7, 1.3],
   [9.2, 3.3, 0.8, 6.6, 5.4, 8.9, 10.0, 4.7, 1.5, 0, 9.3, 3.8,
6.4, 6.0, 2.0, 9.3, 3.5, 7.9, 7.1, 2.7],
   [10.0, 6.1, 8.3, 3.2, 4.4, 0.4, 1.3, 4.4, 7.7, 9.1, 0, 5.4,
5.8, 5.4, 7.1, 1.3, 5.7, 1.9, 2.3, 6.6],
   [9.0, 0.8, 3.0, 3.2, 1.7, 5.1, 6.6, 0.9, 2.3, 3.8, 5.5, 0,
3.2, 2.8, 1.8, 5.5, 0.35, 4.0, 3.4, 1.2],
   [12.0, 4.0, 5.3, 5.4, 3.9, 5.5, 6.7, 2.9, 4.6, 6.1, 6.8,
3.2, 0, 1.1, 4.6, 5.7, 3.5, 5.4, 3.5, 3.6],
   [11.0, 3.6, 4.9, 5.0, 3.5, 5.1, 6.2, 2.5, 4.2, 5.7, 6.3,
2.8, 1.1, 0, 4.2, 5.8, 3.1, 5.5, 3.7, 3.2],
   [8.0, 1.2, 1.2, 4.6, 3.4, 6.8, 8.4, 2.6, 0.6, 2.0, 7.2, 1.7,
4.9, 4.5, 0, 7.2, 2.0, 5.7, 5.2, 1.2],
   [12.0, 6.2, 8.4, 3.9, 5.0, 1.7, 1.0, 4.5, 7.8, 9.2, 1.3,
5.5, 5.7, 5.8, 7.2, 0, 5.8, 3.3, 2.1, 6.7],
   [8.8, 0.65, 2.9, 1.3, 2.2, 5.8, 7.3, 1.2, 2.2, 3.7, 6.2,
0.7, 3.2, 2.8, 1.6, 5.9, 0, 4.4, 3.8, 0.95],
   [9.4, 4.7, 6.9, 3.1, 2.7, 1.6, 3.3, 3.0, 6.3, 7.7, 1.9, 4.0,
5.4, 5.5, 5.7, 3.3, 4.3, 0, 1.9, 5.2],
   [10.0, 4.1, 6.3, 3.1, 2.9, 2.0, 3.1, 2.4, 5.7, 7.1, 2.4,
3.4, 3.5, 3.7, 5.1, 2.1, 3.7, 1.9, 0, 4.6],
   [9.0, 0.75, 1.9, 4.1, 2.9, 6.4, 7.9, 2.2, 1.3, 2.7, 6.8,
1.3, 3.9, 3.5, 1.2, 6.9, 0.95, 5.4, 4.8, 0]
])
```

Menentukan Rute Terpendek menggunakan Algoritma CIH dengan menetapkan titik awal v_i adalah v_0 dan menentukan titik awal $v_j \in V,\ i \neq j$

```
def hitung_jarak_rute(rute, matriks_jarak):
    jarak = 0
    for i in range(len(rute) - 1):
        jarak += matriks_jarak[rute[i], rute[i + 1]]
    jarak += matriks_jarak[rute[-1], rute[0]]
    return jarak

def cheapest_insertion_heuristic(matriks_jarak, titik_awal_i=0,
titik_awal_j=1):
    n = len(matriks_jarak)
    rute = [titik_awal_i, titik_awal_j, titik_awal_i]
    belum_dikunjungi = set(range(n)) - {titik_awal_i,
                       titik_awal_j}

    while belum_dikunjungi:
        jarak_sisipan_terbaik = float('inf')
        indeks_sisipan_terbaik = None
        titik_terbaik = None

        for titik in belum_dikunjungi:
            for i in range(1, len(rute)):
                jarak_sisipan = (matriks_jarak[rute[i - 1],
                titik] + matriks_jarak[titik, rute[i]] -
                matriks_jarak[rute[i - 1], rute[i]])
                if jarak_sisipan < jarak_sisipan_terbaik:
                    jarak_sisipan_terbaik = jarak_sisipan
                    indeks_sisipan_terbaik = i
                    titik_terbaik = titik

        rute.insert(indeks_sisipan_terbaik, titik_terbaik)
        belum_dikunjungi.remove(titik_terbaik)
        jarak_total = hitung_jarak_rute(rute, matriks_jarak)

    return rute, jarak_total

titik_awal_i = 0
for titik_awal_j in range(1, 20):
rute_CIH, jarak_final =
     cheapest_insertion_heuristic(matriks_jarak,
     titik_awal_i=titik_awal_i,
     titik_awal_j=titik_awal_j)
print(f"Titik Awal J: {titik_awal_j}, Rute CIH: {rute_CIH},
Total Jarak: {jarak_final:.2f}")
```

Mengoptimalkan rute yang dihasilkan Algoritma CIH dengan menggunakan Algoritma Tabu Search

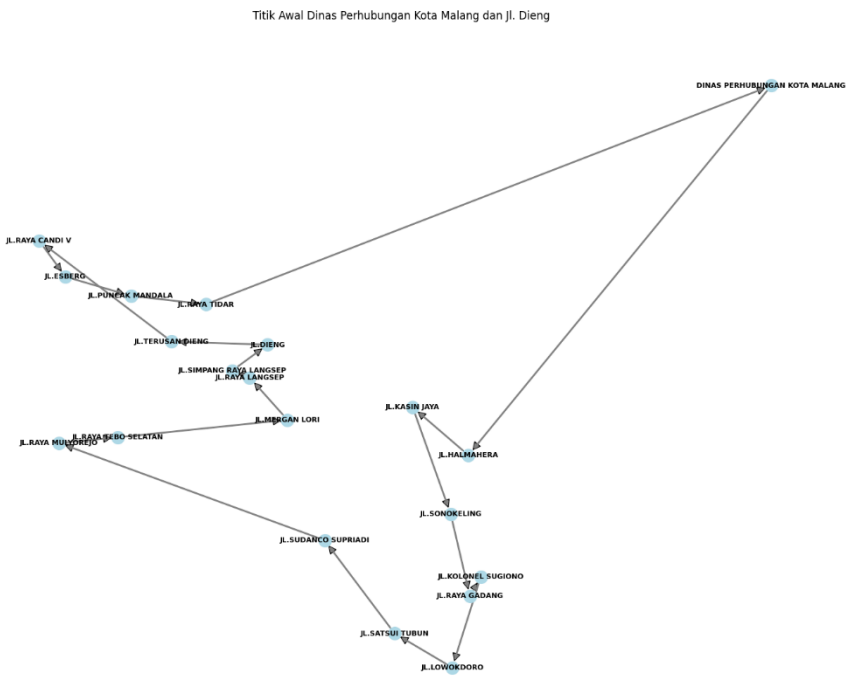
```
def hitung_jarak_rute(rute, matriks_jarak):
    jarak = 0
    for i in range(len(rute) - 1):
        jarak += matriks_jarak[rute[i], rute[i + 1]]
    jarak += matriks_jarak[rute[-1], rute[0]]
    return jarak

def pencarian_tetangga(rute, matriks_jarak):
    tetangga = []
    for i in range(1, len(rute) - 1):
        for j in range(i + 1, len(rute) - 1):
            tetangga_baru = rute.copy()
            tetangga_baru[i], tetangga_baru[j] =
            tetangga_baru[j], tetangga_baru[i]
            jarak = hitung_jarak_rute(tetangga_baru,
            matriks_jarak)
            tetangga.append((tetangga_baru, jarak, i, j))
    return tetangga

def tabu_search(rute_CIH, matriks_jarak, maks_iterasi=3,
ukuran_tabu=3):
    rute_terbaik = rute_CIH.copy()
    jarak_terbaik = hitung_jarak_rute(rute_terbaik,
    matriks_jarak)
    rute_saat_ini = rute_CIH.copy()
    daftar_tabu = []
    rute_terbaik_per_iterasi = [(rute_CIH, jarak_terbaik)]

    for iterasi in range(maks_iterasi):
        tetangga = pencarian_tetangga(rute_saat_ini,
        matriks_jarak)
        kandidat_terbaik = None
        jarak_kandidat_terbaik = float('inf')
        gerakan_terbaik = None

        for tetangga_baru, jarak, i, j in tetangga:
            gerakan = (i, j)
            if gerakan not in daftar_tabu and jarak <
            jarak_kandidat_terbaik:
                kandidat_terbaik = tetangga_baru
                jarak_kandidat_terbaik = jarak
                gerakan_terbaik = Gerakan
        if kandidat_terbaik is not None:
            rute_saat_ini = kandidat_terbaik
            daftar_tabu.append(gerakan_terbaik)
            if len(daftar_tabu) > ukuran_tabu:
                daftar_tabu.pop(0)
                rute_terbaik_per_iterasi.append((rute_saat_ini,
                hitung_jarak_rute(rute_saat_ini,
                matriks_jarak)))
    rute_terpendek, jarak_terpendek =
        min(rute_terbaik_per_iterasi, key=lambda x: x[1])
    return rute_terpendek, jarak_terpendek
```

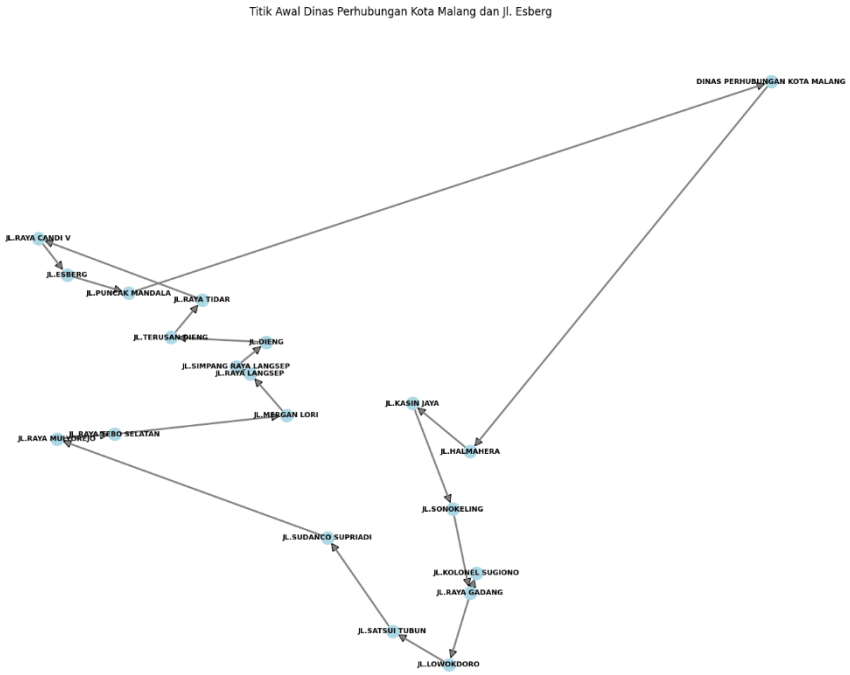
```
titik_awal_i = 0
hasil_tabu_search = []
for titik_awal_j in range(1, 20):
    rute_CIH, jarak_final =
    cheapest_insertion_heuristic(matriks_jarak,
    titik_awal_i=titik_awal_i, titik_awal_j=titik_awal_j)
    print(f"Titik Awal J: {titik_awal_j}, Rute CIH: {rute_CIH},
    Total Jarak CIH: {jarak_final:.2f}")

rute_terbaik, jarak_terbaik = tabu_search(rute_CIH,
matriks_jarak)
hasil_tabu_search.append((titik_awal_j, rute_terbaik,
jarak_terbaik))
print(f"Rute Tabu Search: {rute_terbaik}, Total Jarak:
```

Lampiran 3 *Output Python* Rute Optimasi Algoritma CIH dan *Tabu Search*

https://colab.research.google.com/drive/1ofKTH5rewXyzX_funrrS5zMBUzOIUgd-?usp=sharing

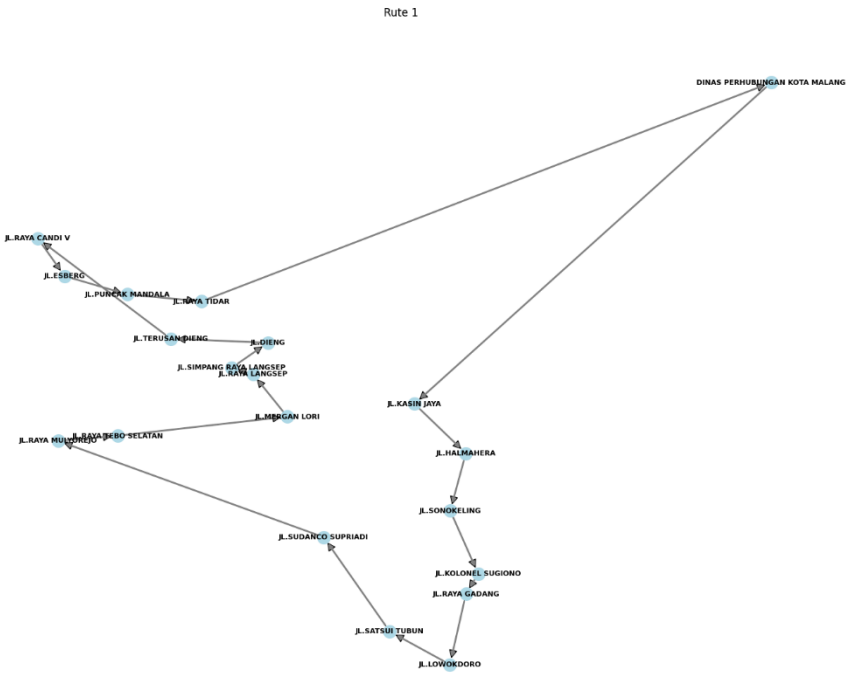
Lampiran 4 Visualisasi Rute Algoritma CIH

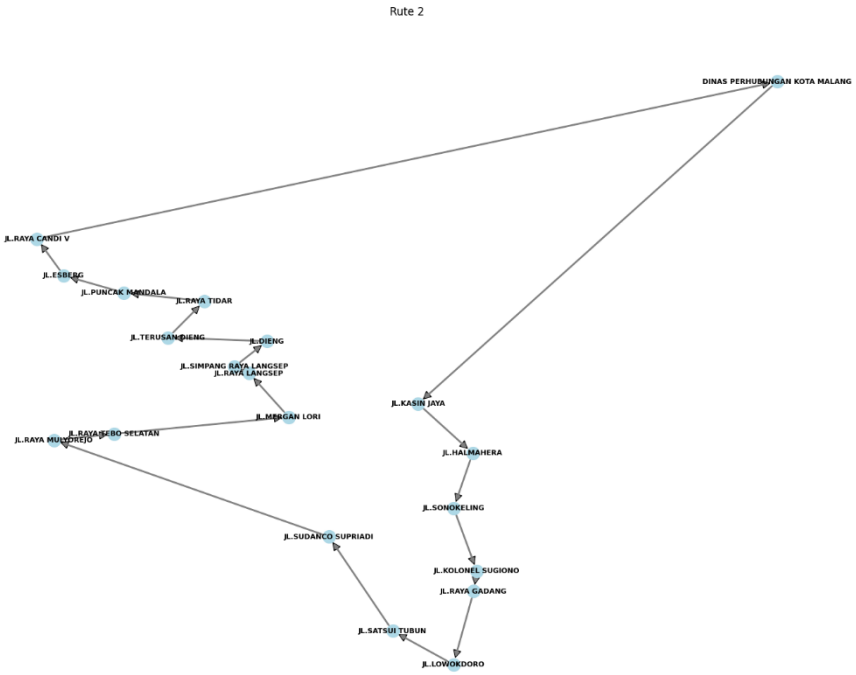




https://bit.ly/VisualisasiRuteAlgoritmaCIH

Lampiran 5 Visualisasi Rute Optimasi Algoritma CIH Menggunakan *Tabu Search*





https://bit.ly/VisualisasiRuteOptimasiAlgoritmaCIHMenggunakanTabuSearch

RIWAYAT HIDUP

Silviyatus Yulianti, yang akrab dipanggil Silvi, lahir di Sidoarjo pada 26 Agustus 2002. Penulis merupakan anak kedua dari tiga bersaudara, putri dari Bapak Sutrisno dan Ibu Yuliatin. Pendidikan awal penulis dimulai di RA Darussalam Keboguyang dan diselesaikan pada tahun 2009, kemudian melanjutkan pendidikan dasar di MI Darussalam Keboguyang hingga tahun 2015. Pendidikan menengah pertama ditempuh di SMPN 2 Jabon dan berhasil diselesaikan pada tahun 2018. Setelah itu, penulis melanjutkan pendidikan menengah atas di SMA Walisongo Gempol dengan jurusan Matematika dan Ilmu Pengetahuan Alam (MIPA) dan lulus pada tahun 2021.

Pada jenjang perguruan tinggi, penulis menempuh studi Strata-1 (S1) di Universitas Islam Negeri Maulana Malik Ibrahim Malang pada Fakultas Sains dan Teknologi dengan program studi Matematika. Selama masa perkuliahan, penulis aktif dalam organisasi DEMA Fakultas Sains dan Teknologi sebagai anggota Departemen Minat dan Bakat selama periode kepengurusan tahun 2022–2023. Selain itu, penulis juga turut berpartisipasi dalam beberapa kepanitiaan kegiatan DEMA Fakultas Sains dan Teknologi pada tahun 2023.

Di luar kegiatan akademik, penulis memiliki pengalaman praktik kerja lapangan di Dinas Perhubungan Kota Malang selama satu bulan, yakni pada tanggal 24 Juni hingga 24 Juli 2024. Selain itu, penulis juga mengikuti kegiatan di luar kampus, seperti mengikuti pelatihan dan seminar.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

Nama : Silviyatus Yulianti
NIM : 210601110010
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Optimasi Algoritma *Cheapest Insertion Heuristic* dengan Algoritma *Tabu Search* dalam Pencarian Rute Terpendek
Pembimbing I : M. Nafie Jauhari, M.Si
Pembimbing II : Ach. Nasichuddin, M.A

No	Tanggal	Hal	Tanda Tangan
1.	17 September 2024	Konsultasi Topik dan Data	1.
2.	4 November 2024	Konsultasi Bab I	2.
3.	11 November 2024	Konsultasi Bab I, II, dan III	3.
4.	14 November 2024	Konsultasi Bab I, II, dan III	4.
5.	28 November 2024	Konsultasi Kajian Agama Bab I dan II	5.
6.	29 November 2024	ACC Bab I, II, dan III	6.
7.	29 November 2024	ACC Kajian Agama Bab I dan II	7.
8.	5 Desember 2024	ACC Seminar Proposal	8.
9.	13 Februari 2025	Konsultasi Revisi Seminar Proposal	9.
10.	20 Februari 2025	Konsultasi Bab IV dan V	10.
11.	6 Maret 2025	ACC Bab IV dan V	11.
12.	6 Maret 2025	Konsultasi Kajian Agama Bab IV	12.
13.	12 Maret 2025	ACC Kajian Agama Bab IV	13.
14.	12 Maret 2025	ACC Seminar Hasil	14.
15.	29 April 2025	Konsultasi Revisi Seminar Hasil	15.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

16.	30 April 2025	ACC Sidang Skripsi	16.
17.	19 Mei 2025	ACC Keseluruhan	17.

Malang, 19 Mei 2025
Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005