**PENERAPAN ALGORITMA GREEDY UNTUK MENENTUKAN RUTE TERPENDEK ANTAR KOTA DARI KOTA KARAWANG KE KOTA MAIOBORO**

Analisis Desain dan Algoritma

Dosen Pengampu: Ultach Enri, M.Kom.

A blue and yellow logo

Description automatically generated

Disusun oleh:

* Aufa Sarifatun Nisa (2210631170010)
* Septian Nur Hidayat (2210631170151)
* Siti Herawati (2210631170153)

4B – Informatika

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS SINGAPERBANGSA KARAWANG**

**2024**

**KATA PENGANTAR**

Puji syukur kami panjatkan ke hadirat Allah SWT, karena atas rahmat dan hidayah-Nya, penulis dapat menyelesaikan tugas mata kuliah Analisis Desain Algoritma dengan judul "Penerapan Algoritma Greedy Untuk Menentukan Rute Terpendek Antar Kota Dari Kota Karawang ke Kota Bandung".

Dalam penyusunan tugas ini, kami mengucapkan terima kasih yang sebesar-besarnya kepada Ibu Ultach Enri, M.Kom. selaku dosen pengampu mata kuliah Analisis Desain Algoritma, atas bimbingan, arahan, dan motivasi yang telah diberikan kepada kami dalam menyelesaikan tugas ini.

Dalam makalah ini, kami mencoba untuk memberikan penjelasan yang komprehensif tentang konsep algoritma Greedy, serta bagaimana algoritma tersebut dapat diterapkan untuk menentukan rute terpendek antara Kota Karawang dan Kota Bandung. Kami juga menyertakan studi kasus yang mengilustrasikan langkah-langkah yang diambil dalam proses penerapan algoritma ini, mulai dari pemetaan kota-kota yang dilalui, hingga perhitungan jarak terpendek berdasarkan kriteria yang ditetapkan.

Kami berharap makalah ini dapat memberikan kontribusi yang berarti dalam pemahaman dan penerapan greedy untuk mencari solusi terbaik. Kami mengucapkan terima kasih atas dukungan dan kontribusi dari berbagai pihak yang telah membantu dalam penyusunan makalah ini. Kritik dan saran yang membangun sangat kami harapkan untuk perbaikan kedepannya. Akhir kata, kami berharap semoga Allah SWT senantiasa memberikan ridha-Nya dan melimpahkan keberkahan dalam setiap langkah perjalanan kita.

Aamiin.

Karawang, 13 Mei  2024

Kelompok 04

# DAFTAR ISI

[KATA PENGANTAR ii](#_Toc168081842)

[DAFTAR ISI iii](#_Toc168081843)

[DAFTAR GAMBAR v](#_Toc168081844)

[GAMBAR TABEL vi](#_Toc168081845)

[BAB 1 PENDAHULUAN 1](#_Toc168081846)

[**1.1 Latar Belakang Masalah 1**](#_Toc168081848)

[**1.2 Batasan Masalah 1**](#_Toc168081849)

[**1.3 Rumusan Masalah 1**](#_Toc168081850)

[**1.4 Tujuan dan Manfaat 2**](#_Toc168081851)

[**1.4.1 Tujuan 2**](#_Toc168081852)

[**1.4.2 Manfaat 2**](#_Toc168081853)

[BAB II TINJAUAN PUSTAKA 3](#_Toc168081854)

[**2.1  Definisi Algoritma 3**](#_Toc168081856)

[**2.2. Definisi Greedy 3**](#_Toc168081857)

[**2.3  Teori Graf 4**](#_Toc168081858)

[**2.4  Jalur Terpendek ((Shortest Path) 4**](#_Toc168081859)

[**2.5 Algoritma Dijkstra 4**](#_Toc168081860)

[**2.6  Google Maps API 4**](#_Toc168081861)

[**2.7  Google Maps (Peta) 5**](#_Toc168081862)

[**2.8 Pemrograman Python 5**](#_Toc168081863)

[**2.9 Flowchart (Diagram Alur) 6**](#_Toc168081864)

[BAB III METODOLOGI 8](#_Toc168081865)

[**3.1 Perencanaan 8**](#_Toc168081867)

[**3.2 Pengumpulan Data 8**](#_Toc168081868)

[**3.3 Analisis kebutuhan 8**](#_Toc168081869)

[**3.4 Perancangan 9**](#_Toc168081870)

[BAB IV HASIL DAN PEMBAHASAN 11](#_Toc168081871)

[**4.1 Implementasi Perhitungan secara Manual 11**](#_Toc168081873)

[**4.2 Hasil Pengujian Jarak 21**](#_Toc168081874)

[**4.3 Implementasi Menggunakan Pemrograman python 21**](#_Toc168081875)

[BAB V PENUTUP 27](#_Toc168081876)

[**5.1 Kesimpulan 27**](#_Toc168081878)

[**5.2 Saran 27**](#_Toc168081879)

[DAFTAR PUSTAKA 28](#_Toc168081880)

# DAFTAR GAMBAR

[3.1. 1Tahap Perancangan 8](#_Toc168293372)

[3.1. 2Flowchart Algoritma Greedy 10](#_Toc168293373)

[4.1. 1 Rute dari Karawang ke Kosambi 11](#_Toc168293395)

[4.1. 2 Rute Kosambi ke Cikampek 12](#_Toc168293396)

[4.1. 3Rute Cikampek ke Pamanukan 13](#_Toc168293397)

[4.1. 4 Rute Pamanukan ke Patrol 14](#_Toc168293398)

[4.1. 5 Rute Patrol ke Jatibarang 15](#_Toc168293399)

[4.1. 6 Rute Jatibarang ke Karanganyar 15](#_Toc168293400)

[4.1. 7 Rute Karanganyar ke Cirebon 16](#_Toc168293401)

[4.1. 8 Rute Cirebon ke Brebes 16](#_Toc168293402)

[4.1. 9 Rute Brebes ke Tegal 17](#_Toc168293403)

[4.1. 10 Rute Tegal ke Pemalang 17](#_Toc168293404)

[4.1. 11 Rute Pemalang ke Kandeman 18](#_Toc168293405)

[4.1. 12 Rute Kandeman ke Sukoredjo 18](#_Toc168293406)

[4.1. 13 Rute Sukoredjo ke Temanggung 19](#_Toc168293407)

[4.1. 14 Rute Temanggung ke Magelang 19](#_Toc168293408)

[4.1. 15 Rute Magelang ke Malioboro 20](#_Toc168293409)

[4.1. 16 Rute Karawang ke Malioboro 20](#_Toc168293410)

[4.2. 1 Graf Rute Karawang hingga ke Malioboro 21](#_Toc168293411)

[4.3. 2Implementasi dalam bentuk codingan python 25](#_Toc168293416)

[4.3. 3 Output dari implemntasi dalm bentuk codingan python 26](#_Toc168293417)

# GAMBAR TABEL

[Tabel 2.9. 1 Rancangan Flowchart (Diagram Alur) 7](#_Toc168292242)

# BAB 1

# **PENDAHULUAN**

## **1.1 Latar Belakang Masalah**

  Transportasi merupakan salah satu aspek penting dalam kehidupan sehari-hari, terutama dalam menghubungkan berbagai kota di Indonesia. Rute yang efisien antara dua kota tidak hanya menghemat waktu, tetapi juga mengurangi biaya operasional dan konsumsi bahan bakar. Kota Karawang dan Kota Malioboro, sebagai dua pusat ekonomi yang sangat penting, membutuhkan rute transportasi yang optimal untuk mendukung kegiatan bisnis dan mobilitas penduduk. Dalam konteks ini, penerapan algoritma Greedy untuk menentukan rute terpendek antara kedua kota tersebut menjadi solusi yang relevan dan efektif.

Algoritma Greedy adalah salah satu metode dalam ilmu komputer yang digunakan untuk menyelesaikan berbagai masalah optimasi, termasuk dalam penentuan rute terpendek. Algoritma ini bekerja dengan cara memilih langkah yang tampak paling optimal pada setiap tahapannya, dengan harapan solusi keseluruhan akan optimal. Dalam kasus penentuan rute terpendek dari Karawang ke Bandung, algoritma Greedy dapat digunakan untuk memilih jalan dengan jarak terpendek secara iteratif dari satu titik ke titik berikutnya hingga mencapai tujuan akhir. Penerapan algoritma ini diharapkan mampu menghasilkan rute yang tidak hanya cepat, tetapi juga hemat biaya dan waktu.

Dengan demikian, pengguna jalan dan penyedia jasa transportasi dapat membuat keputusan yang lebih informasional mengenai rute mana yang sebaiknya diambil. Selain itu, studi ini juga dapat memberikan dasar bagi pengembangan sistem navigasi yang lebih canggih di masa depan, yang tidak hanya mempertimbangkan jarak tetapi juga faktor-faktor lain seperti kondisi lalu lintas dan kualitas jalan. Dengan demikian, penerapan algoritma Greedy untuk menentukan rute terpendek antar kota dari Karawang ke Malioboro dapat menjadi langkah awal menuju optimalisasi jaringan transportasi yang lebih luas.

## **1.2 Batasan Masalah**

Berdasarkan latar belakang diatas, penulisan makalah ini terdapat batasan-batasan masalah. Adapun batasan masalah adalah sebagai berikut :

1. Pencarian dilakukan dengan metode algoritma greedy.
2. Penerapan graf yang digunakan harus memiliki bobot positif

## **1.3 Rumusan Masalah**

Berdasarkan latar belakang masalah diatas, rumusan masalah dalam makalah ini adalah sebagai berikut :

1. Bagaimana menerapkan algoritma greedy dalam pencarian rute terpendek antar kota Karawang ke kota Malioboro.
2. Bagaimana mengimplementasi yang dapat dilakukan algoritma greedy dalam melakukan pencarian rute terdekat antar kota Karawang ke kota Malioboro.

## **1.4 Tujuan dan Manfaat**

### **1.4.1 Tujuan**

Berdasarkan latar belakang di atas, terdapat tujuan penelitian dalam makalah ini. Adapun tujuan penelitian sebagai berikut :

1. Mengetahui proses algoritma greedy dalam menentukan rute terpendek antar kota Karawang ke kota Bandung.
2. Mengoptimalkan rute transportasi antar kota untuk mengurangi kemacetan, menghemat biaya, dan mempercepat perjalanan.

### **1.4.2 Manfaat**

Berdasarkan lataR belakang diatas, terdapat manfaat dari penelitian dalam makalah ini. Adapun manfaat penelitian sebagai berikut :

1. Untuk mendapatkan pengetahuan dari algoritma greedy dalam melakukan rute terpendek antar kota Karawang ke kota Malioboro
2. Dapat digunakan dalam sistem navigasi dan aplikasi peta untuk memberikan rute cepat dan mudah kepada pengguna.

# BAB II

# TINJAUAN PUSTAKA

## **2.1  Definisi Algoritma**

Algoritma adalah metode efektif yang diekspresikan sebagai rangkaian terbatas. Algoritma juga merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat berupa apa saja, dengan syarat untuk setiap permasalahan memiliki kriteria kondisi awal yang harus dipenuhi sebelum menjalankan sebuah algoritma. Algoritma juga memiliki pengulangan proses (iterasi), dan juga memiliki keputusan hingga keputusan selesai (Maulana, 2017).

Algoritma berbeda dengan logaritma. Logaritma merupakan operasi matematika yang merupakan kebalikan dari eksponen atau pemangkatan. contoh logaritma seperti bc=a ditulis sebagai blog a=c (b disebut basis) (Maulana, 2017).

**Sifat –sifat Algoritma**

sifat utama dari algoritma adalah sebagai berikut:

1. Input

Algoritma memiliki input atau kondisi awal sebelum implementasinya, yang dapat berupa nilai variabel yang diambil dari set khusus.

1. Output

suatu algoritma akan menghasilkan output setelah implementasinya,atau algoritma tersebut akan mengubah kondisi awal menjadi kondisi akhir, dimana nilai output diperoleh dari nilai input yang telah diproses melalui algoritma.

1. Definiteness

langkah-langkah yang dituliskan dalam algoritma terdefinisikan dengan jelas jadi mudah dilakukan oleh pengguna algoritma.

1. Finiteness

Setiap algoritma harus memberikan hasil akhir atau keluaran setelah sejumlah langkah yang terbatas dilakukan terhadap setiap kondisi awal atau masukan yang diberikan.

1. Effectiveness

Setiap langkah dalam algoritma dapat dilakukan dalam suatu selang waktu tertentu sehingga pada akhirnya diperoleh solusi yang sesuai yang diharapkan.

1. Generality

langkah-langkah algoritma berlaku untuk setiap set input yang sesuai dengan masalah yang diberikan, bukan hanya untuk set tertentu.

## **2.2. Definisi Greedy**

Algoritma Greedy adalah algoritma untuk menemukan solusi optimal untuk masalah optimasi dengan menyelesaikan masalah langkah demi langkah. Algoritma Greedy merupakan algoritma yang dapat memecahkan masalah dengan membuat pilihan yang tampaknya terbaik pada saat tertentu. Banyak masalah optimasi yang dapat diselesaikan dengan menggunakan algoritma Greedy.

## **2.3  Teori Graf**

Graf adalah bidang ilmu yang memiliki banyak cabang. Seringkali digunakan untuk menunjukkan jaringan, graf dapat menunjukkan banyak struktur dan menyelesaikan banyak masalah. Misalkan jaringan jalan raya digambarkan sebagai graf, dengan kota sebagai simpul atau node, dan jalan yang menghubungkan setiap kota sebagai sisi. Panjang jalan adalah bobot dari sisi, dan dalam beberapa model persoalan, bobot dari suatu sisi dapat negatif. Misalkan simpul mewakili kota, dan sisi mewakili perjalanan yang memungkinkan. Biaya yang dikeluarkan untuk perjalanan yang memungkinkan diwakili oleh setiap sisi, dan jarak yang ditempuh diwakili oleh setiap sisi. Pasangan himpunan (V, E) adalah definisi dari graf G, yang ditulis dengan notasi G = (V,E). Dalam hal ini, V adalah himpunan simpul (vertices atau node) yang tidak kosong, dan E adalah himpunan sisi (edges atau arcs) yang menghubungkan dua simpul. Tidak mungkin bahwa V kosong, tetapi E juga mungkin kosong. Untuk menandai titik-titik pada grafik, Anda dapat menggunakan huruf seperti a, b, c, v, w, dll., dengan bilangan dasar 1, 2, 3, atau kombinasi keduanya.

## **2.4  Jalur Terpendek ((Shortest Path)**

Salah satu masalah dengan optimasi adalah mencari lintasan terpendek di grafik. Untuk pencarian lintasan terpendek, digunakan graf berbobot, yang berarti bahwa setiap sisi dari graf memiliki nilai atau bobot. Nilai atau bobot ini dapat menunjukkan hal-hal seperti jarak antar kota, waktu pengiriman pesan, biaya pembangunan, dan sebagainya. Di sini, asumsi yang digunakan adalah bahwa semua bobot memiliki nilai positif.  Lintasan terpendek adalah jalur yang dilalui dari suatu node ke node lain dengan besar atau nilai pada sisi yang jumlah akhirnya dari node awal ke node akhir paling kecil. Lintasan terpendek adalah lintasan minimum yang diperlukan untuk mencapai suatu tempat dari tempat lain. Lintasan minimum yang dimaksud dapat dicari dengan menggunakan graf. Graf yang digunakan adalah graf yang berbobot yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot.

## **2.5 Algoritma Dijkstra**

Algoritma Dijkstra adalah algoritma greedy yang digunakan untuk memecahkan masalah jarak terpendek untuk sebuah graf berarah dengan bobot-bobot sisi (edge) yang bernilai tak negatif. Ide dasar algoritma ini adalah untuk menemukan nilai biaya yang paling dekat dengan tujuan yang berfungsi pada sebuah graf berbobot, sehingga dapat membantu memberikan pilihan jalur. Algoritma Dijkstra menggunakan graph berarah untuk menemukan rute lintasan terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalnya, jika titik menunjukkan gedung dan garis menunjukkan jalan, Algoritma Dijkstra menghitung semua bobot terkecil yang mungkin dari setiap titik.

## **2.6  Google Maps API**

Google Maps API adalah library JavaScript yang membutuhkan pengetahuan HTML dan JavaScript, serta koneksi Internet yang stabil. Dengan menggunakan API ini, Anda dapat menghemat waktu dan uang dengan membangun aplikasi peta digital yang handal, fokusnya hanya pada menampilkan data.Dengan kata lain, kami hanya membuat data, dan Google memiliki peta, jadi kami tidak perlu membuat peta lokasi atau dunia.

Dalam pembuatan program Google Map API menggunakan urutan sebagai berikut:

1. Memasukkan Maps API JavaScript ke dalam HTML kita.

2. Membuat element div dengan nama map\_canvas untuk menampilkan peta.

3. Membuat beberapa objek literal untuk menyimpan property-properti pada peta.

4. Menuliskan fungsi JavaScript untuk membuat objek peta.

5. Meng-inisiasi peta dalam tag body HTML dengan event onload. Pada Google Maps

API terdapat 4 jenis pilihan model peta yang disediakan oleh Google, diantaranya adalah:

1. Roadmap, ini yang saya pilih, untuk menampilkan peta biasa dimensi
2. Satellite, untuk menampilkan foto satelit
3. Terrain, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai
4. Hybrid, akan menunjukkan foto satelit yang diatasnya tergambar pula apa yang tampil pada roadmap (jalan dan nama kota)

## **2.7  Google Maps (Peta)**

Google Maps adalah sebuah paket lengkap untuk menciptakan suatu rancangan sistem aplikasi yang berkenaan dengan pemetaan atau petunjuk suatu lokasi, kita bisa berkreasi dengan berbagai fitur yang disediakan oleh Google Maps, memanfaatkan layanan Maps Application Programming Interface (API) perancang sistem Aplikasi bisa menghemat waktu karena tidak membuat peta dari awal. Kita berhak menentukan apa-apa saja yang ingin ditampilkan pada Maps dan menentukan lokasi sesuai keinginan kita sebagai pengguna Google Maps API.

Google Maps, yang memiliki berbagai kemampuan dan mudah digunakan, adalah salah satu fasilitas Google yang menawarkan layanan pemetaan wilayah. Layanan informasi bisnis, layanan publik, jalan, lokasi, dan lainnya merupakan komponen tambahan yang mendukung peta tersebut. Google Maps adalah library JavaScript yang membutuhkan pengetahuan HTML dan JavaScript, serta koneksi Internet. Dengan menggunakannya, kita dapat menghemat waktu dan uang dengan membangun aplikasi peta digital, fokus kami hanya pada data yang akan ditampilkan.

## **2.8 Pemrograman Python**

Python dianggap sebagai bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berpusat pada tingkat keterbacaan kode. Bahasa ini digambarkan sebagai bahasa yang menggabungkan kemampuan, potensi, dan dengan fungsionalitas pustaka standar yang besar dan komprehensif, dan sintaksis kode yang sangat jelas. Python adalah bahasa pemrograman umum yang dirancang untuk membuat source code mudah dibaca. Dengan library yang lengkap, programmer dapat membuat aplikasi yang mutakhir dengan menggunakan source code yang tampak sederhana.

## **2.9 Flowchart (Diagram Alur)**

Flowchart berfungsi sebagai representasi simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah; ini memudahkan pengguna untuk melihat bagian-bagian yang terlupakan dalam analisis masalah dan juga membantu berkomunikasi antara pemrogram yang bekerja dalam tim proyek. Flowchart membuat lebih mudah untuk memberi tahu orang lain (bukan pemrogram) jalannya program karena mereka membantu memahami urutan logika yang panjang dan rumit. Adapun jenis – jenis flowchart sebagai berikut : Bagan Alir Sistem, Bagan Alir Dokumen, Bagan Alir Skematik, Bagan Alir Program, Bagan Alir proses.

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
| Mulai              Selesai | Permulaan dan akhir dari suatu program, disebut dengan terminator symbol |
|  | Arah Aliran dari suatu Program., disebut dengan connecting line. |
|  | Proses Penghitungan atau pemrosesan data, disebut dengan processing symbol. |
|  | Input/output dalam format cetak, disebut dengan simbol dokumen. |
|  | Menggambarkan dokumen beserta rangkapnya atau beberapa dokumen, disebut dengan berkas/ multi dokumen. |
|  | Simbol keluar-masuk atau penyambungan proses dalam lembar/halaman yang sama, disebut dengan connector symbol. |
|  | Simbol pemilihan proses berdasarkan kondisi yang ada, disebut dengan decision symbol. |
|  | Simbol penyimpanan ke database atau storage, biasanya disebut dengan database symbol. |
|  | Simbol memasukan data secara manual, disebut dengan simbol manual input. |
|  | Simbol menunjukkan pengolahan secara manual atau operasi manual yang tidak dilakukan oleh komputer, disebut dengan manual operation symbol. |
|  | Simbol yang menyatakan input/output tanpa tergantung, disebut dengan input-output symbol. |

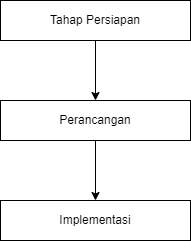
Tabel 2.9. 1 Rancangan Flowchart (Diagram Alur)

# BAB III

# METODOLOGI

## **3.1 Perencanaan**

Pada titik ini, penelitian ini merencanakan pekerjaannya dalam beberapa tahap, yaitu persiapan, desain, dan implementasi. Rencana kerja penelitian ini digambarkan dalam diagram blok berikut.



3.1. 1Tahap Perancangan

1. Pada tahap persiapan dilakukan serangkaian kegiatan sebelum memulai tahap pengumpulan data dan pengolahan data.
2. Untuk membantu peneliti memahami alur atau fungsi rancangan yang akan datang, rancangan penelitian ini melibatkan pengembangan tahapan kebutuhan yang diubah ke dalam diagram blok, flowchart, dan alat grafis lainnya.
3. Pada tahap implementasi, Semua proses yang dirancang dengan baik antara perangkat keras dan perangkat lunak, bersama dengan kode sumber.

## **3.2 Pengumpulan Data**

Pada tahap ini, Menemukan referensi teoritis yang terkait dengan kasus atau masalah saat ini adalah salah satu metode pengumpulan data yang digunakan. Masalah dan landasan teori untuk penelitian dapat diperkuat dengan melakukan penelitian literatur dan tugas akhir tentang penerapan algoritma greedy untuk menentukan rute terpendek di kota Karawang dan Bandung.

## **3.3 Analisis kebutuhan**

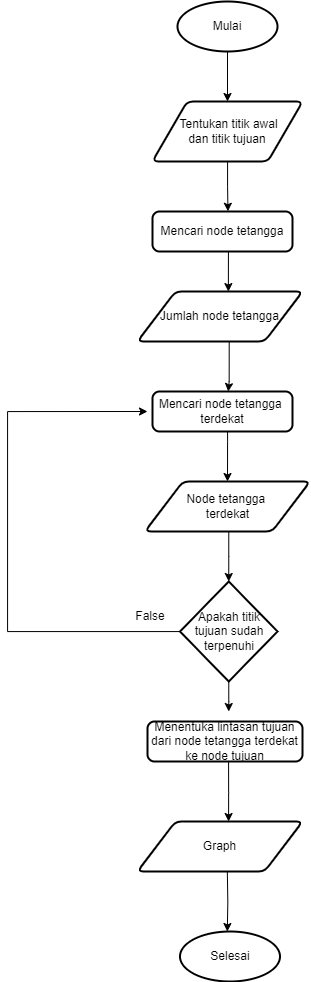
Analisis kebutuhan sistem mengidentifikasi bahan-bahan yang diperlukan dalam sistem yang akan digunakan untuk menambah dan membantu proses pembuatan. Memasukkan titik awal dan jarak kota ke dalam peta yang telah dianalisis adalah salah satu tindakan yang harus dilakukan saat merancang sistem ini. Rute terpendek akan diproses dan ditentukan dengan algoritma greedy. Hasilnya akan ditampilkan, dan algoritma ini akan digunakan untuk mendapatkan hasil yang baik dari grafik yang telah dibentuk.

## **3.4 Perancangan**

Dalam proses pembuatan suatu sistem atau aplikasi, perancangan sistem dilakukan dengan tujuan untuk memberikan gambaran umum tentang aplikasi yang akan dibuat.Tahap awal desain perangkat lunak adalah desain sistem. Tahap ini dilakukan untuk mengetahui kondisi umum sistem.

1. **Flowchart Algoritma Greedy**

Rancangan berikut adalah diagram flowchart yang dirancang oleh algoritma greedy untuk menentukan rute terpendek menuju dari kota Karawang menuju kota Bandung:



3.1. 2Flowchart Algoritma Greedy

# BAB IV

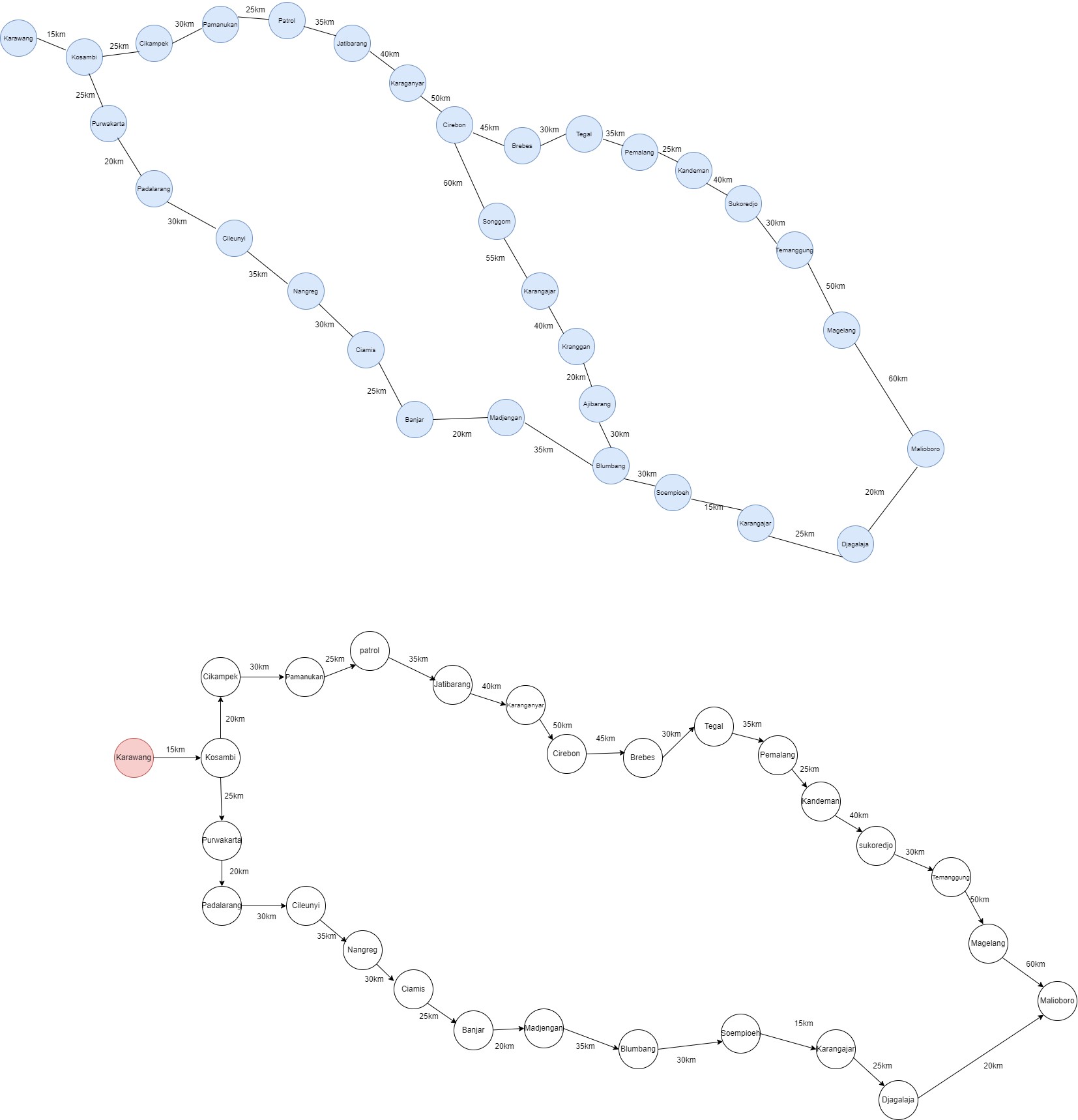
# HASIL DAN PEMBAHASAN

## **4.1 Implementasi Perhitungan secara Manual**

Selanjutnya, algoritma greedy digunakan untuk melakukan perhitungan dari titik awal yang dipilih oleh pengguna, yaitu dari titik awal kota Karawang hingga kota Malioboro. Proses perhitungan dilakukan dengan Algoritma Greedy sebagai berikut:

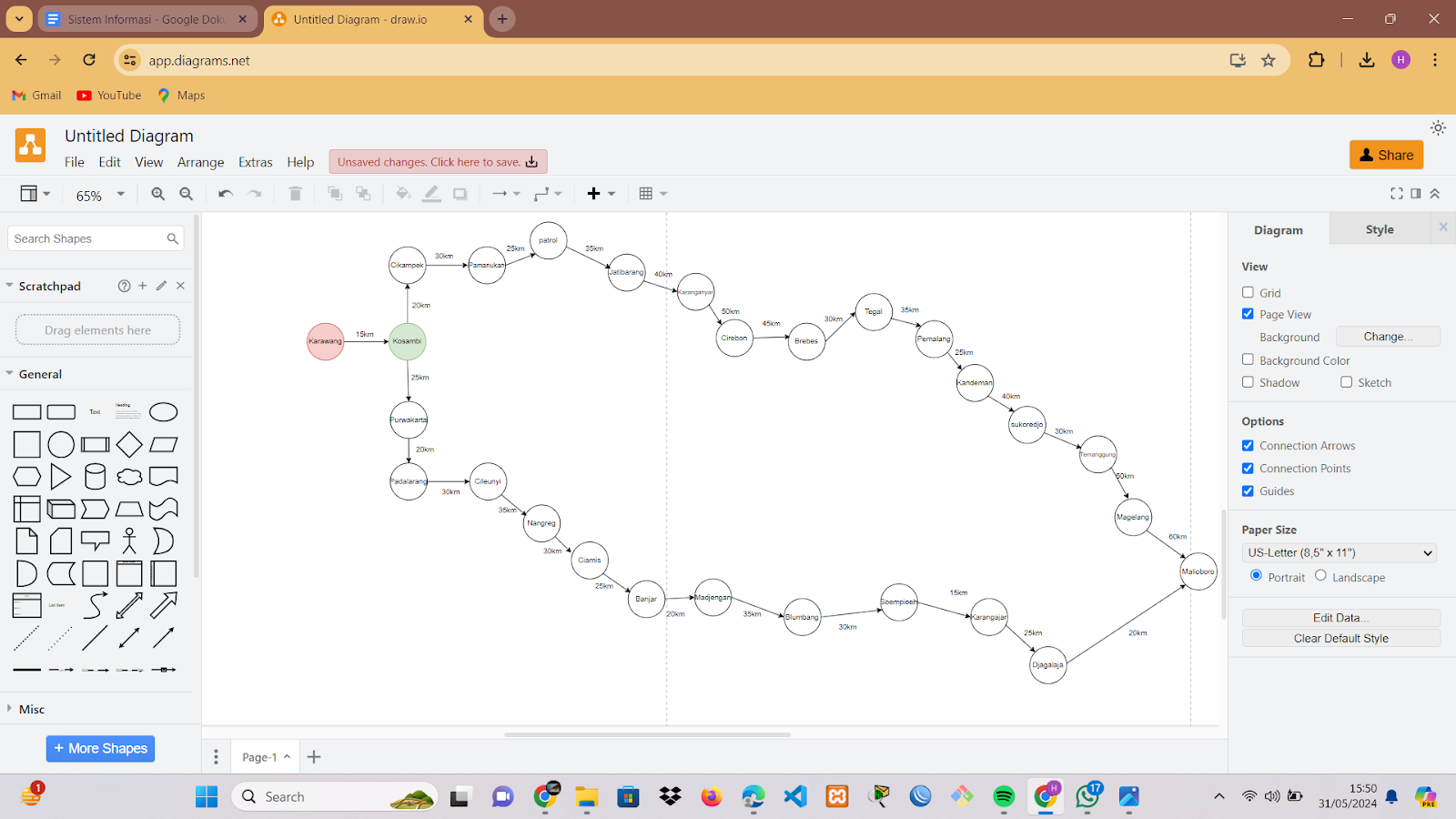
Graf dari noe karawang sampai dengan node Malioboro dan bobot tiap tiap sisi ( bobot menyatakan jarak dari setiap node sat uke node lain) seperti gambar beriku:

Lintasan pertama yang harus dipilih dari kota karawang adalah yang berdekatan yaitu kosambi (sisi karawang dan kosambi) dengan bobot 15km, ditunjukan pada gambar 4.1.1.



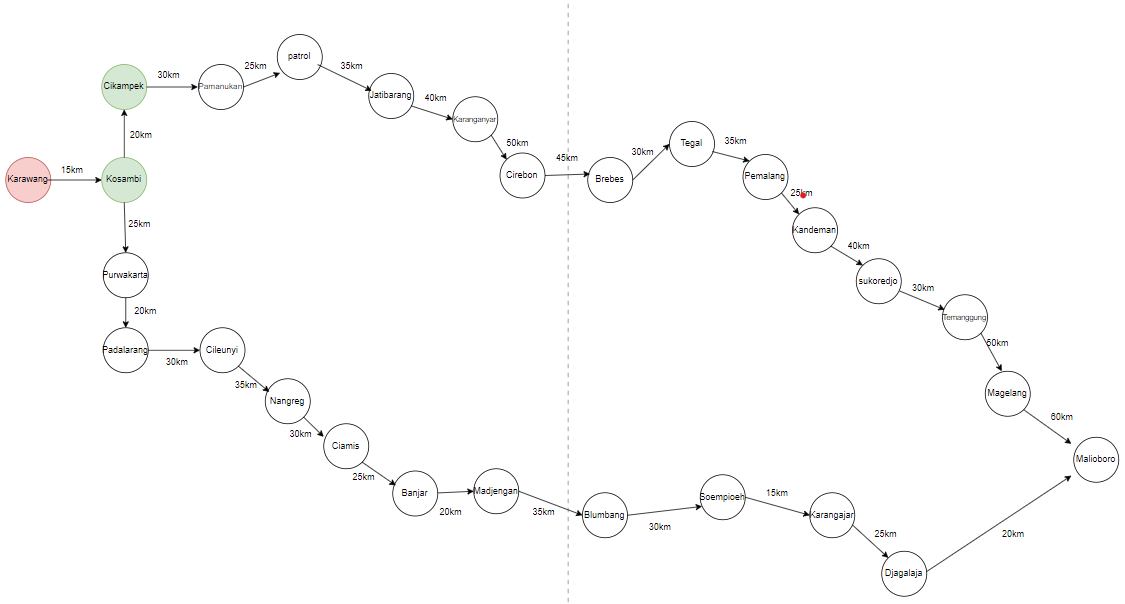
4.1. 1 Rute dari Karawang ke Kosambi

Selanjutnya memilih sisi yang berdekatan dengan Kosambi, yaitu Cikampek dengan total bobot jaraknya menjadi 35km. Seperti pada Gambar 4.1.2.



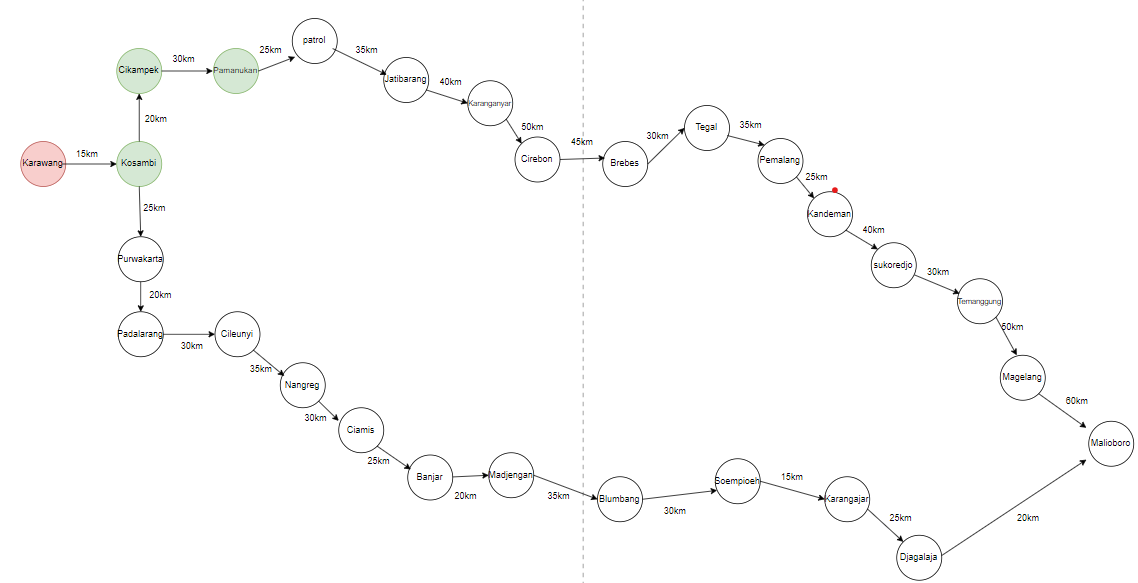
4.1. 2 Rute Kosambi ke Cikampek

Selanjutnya memilih sisi yang berdekatan dengan Cikampek, yaitu Pamanukan dengan total bobot jaraknya menjadi 65km. Seperti pada Gambar 4.1.3.



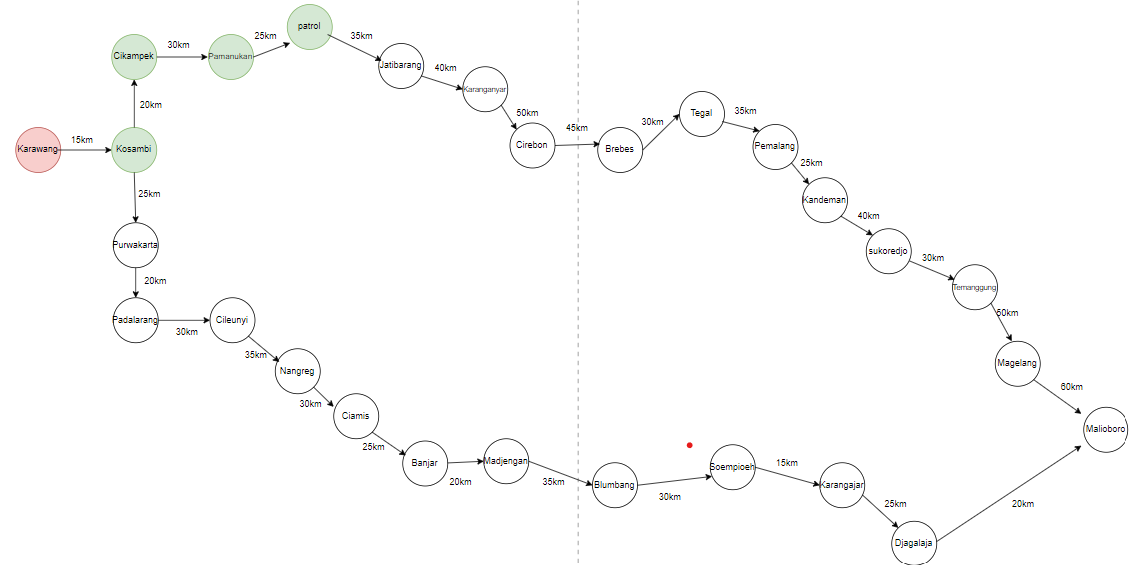
4.1. 3Rute Cikampek ke Pamanukan

Selanjutnya memilih sisi yang berdekatan dengan Pamanukan, yaitu Patrol dengan total bobot jaraknya adalah 90km. Seperti pada Gambar 4.1.4.



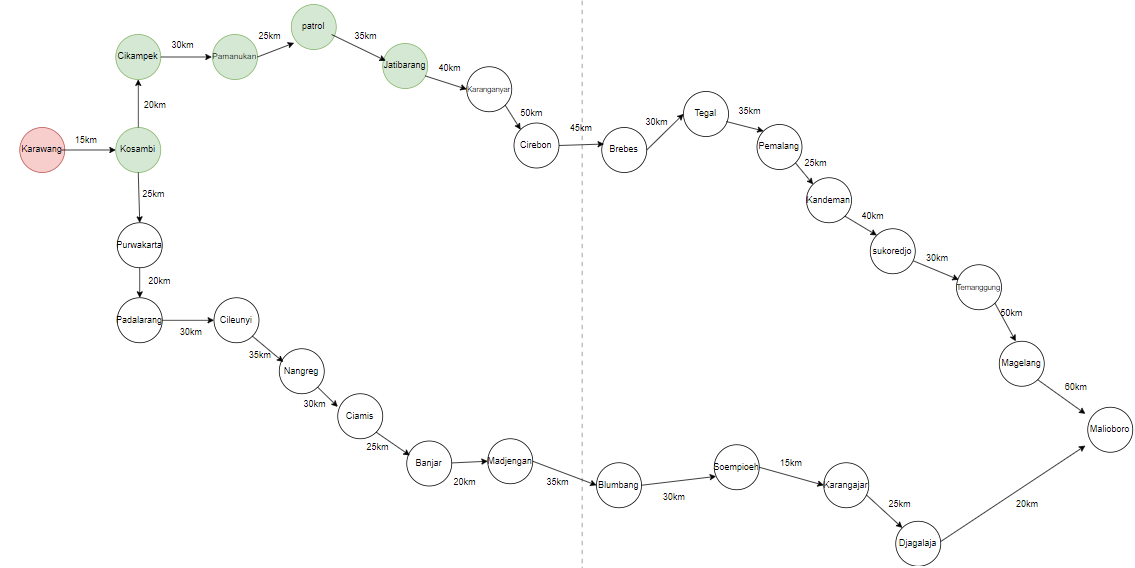
4.1. 4 Rute Pamanukan ke Patrol

Selanjutnya memilih sisi yang berdekatan dengan Patrol, yaitu Jatibarang dengan total bobot jaraknya adalah 125km. Seperti pada Gambar 4.1.5.



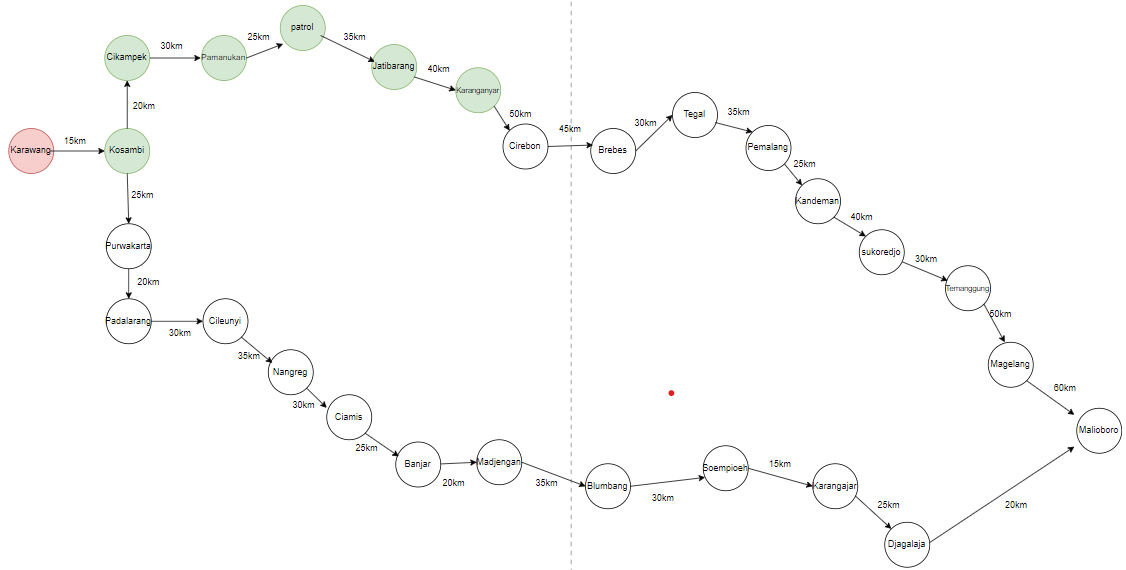
4.1. 5 Rute Patrol ke Jatibarang

Selanjutnya memilih sisi yang berdekatan dengan Jatibarang, yaitu Karanganyar dengan total bobot jaraknya adalah 165km. Seperti pada Gambar 4.1.6.



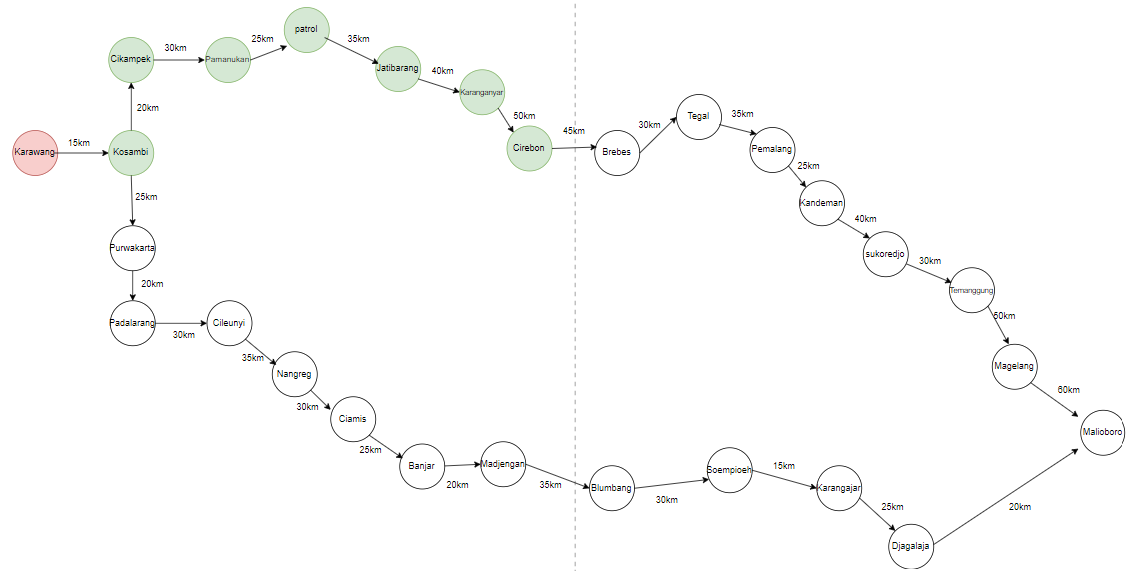
4.1. 6 Rute Jatibarang ke Karanganyar

Selanjutnya memilih sisi yang berdekatan dengan Karanganyar, yaitu Cirebon dengan total bobot jaraknya adalah 215km. Seperti pada Gambar 4.1.7.



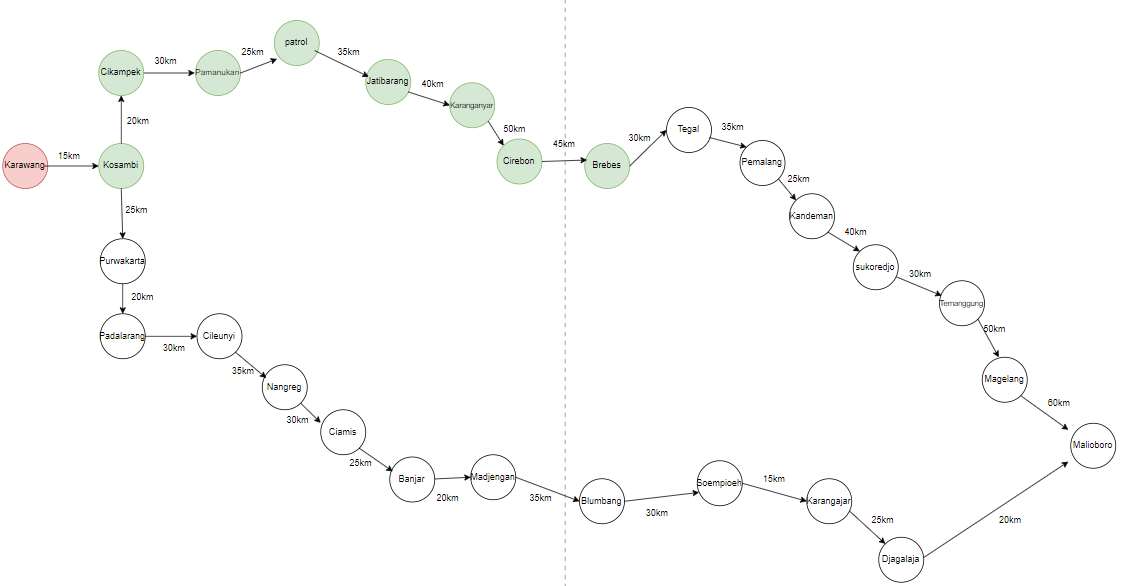
4.1. 7 Rute Karanganyar ke Cirebon

Selanjutnya memilih sisi yang berdekatan dengan Cirebon, yaitu Brebes dengan total bobot jaraknya adalah 260km. Seperti pada Gambar 4.1.8.



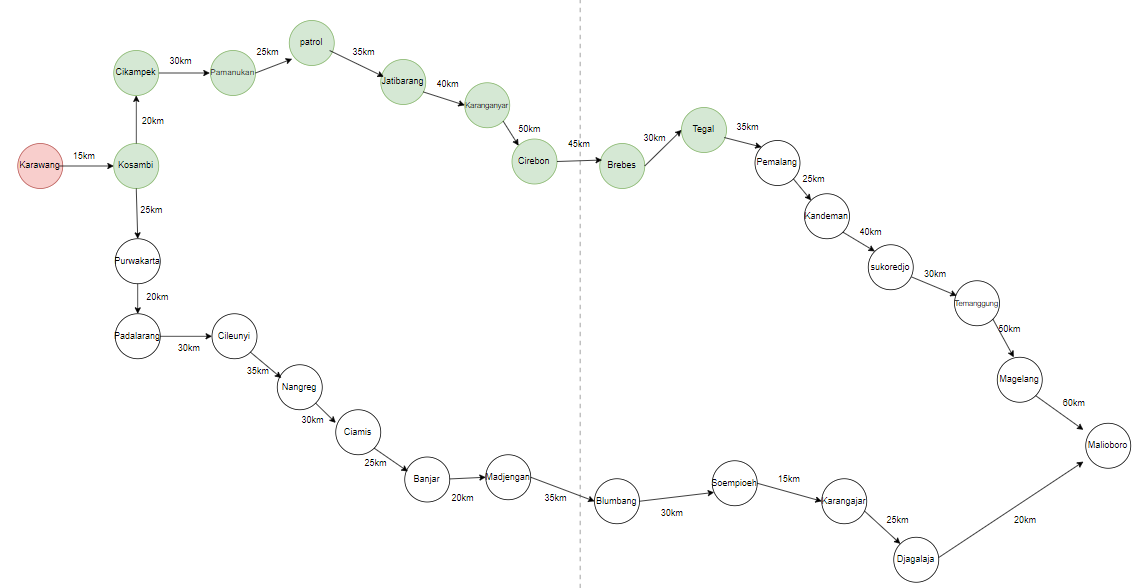
4.1. 8 Rute Cirebon ke Brebes

Selanjutnya memilih sisi yang berdekatan dengan Brebes, yaitu Tegal dengan total bobot jaraknya adalah 290km. Seperti pada Gambar 4.1.9.



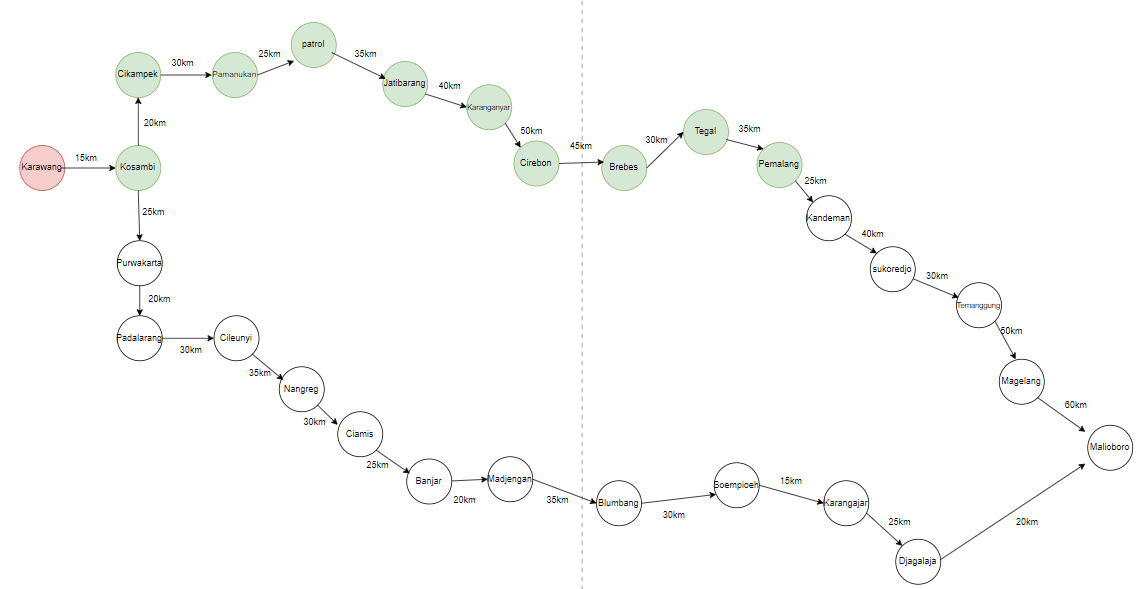
4.1. 9 Rute Brebes ke Tegal

Selanjutnya memilih sisi yang berdekatan dengan Tegal, yaitu Pemalang dengan total bobot jaraknya adalah 325km. Seperti pada Gambar 4.1.10.



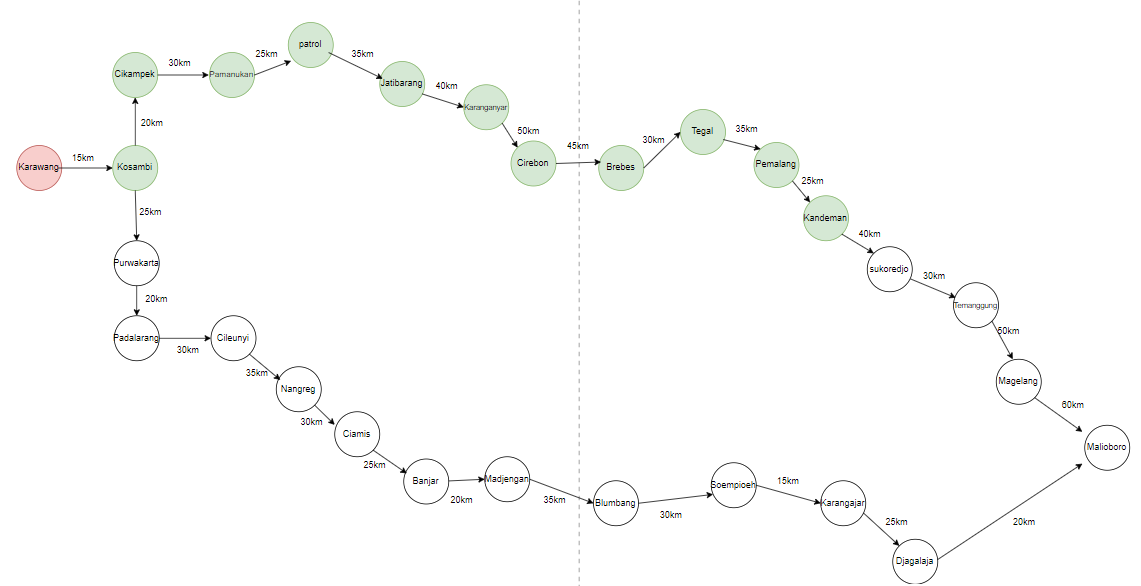
4.1. 10 Rute Tegal ke Pemalang

Selanjutnya memilih sisi yang berdekatan dengan Pemalang, yaitu Kandeman dengan total bobot jaraknya adalah 350km. Seperti pada Gambar 4.1.11.



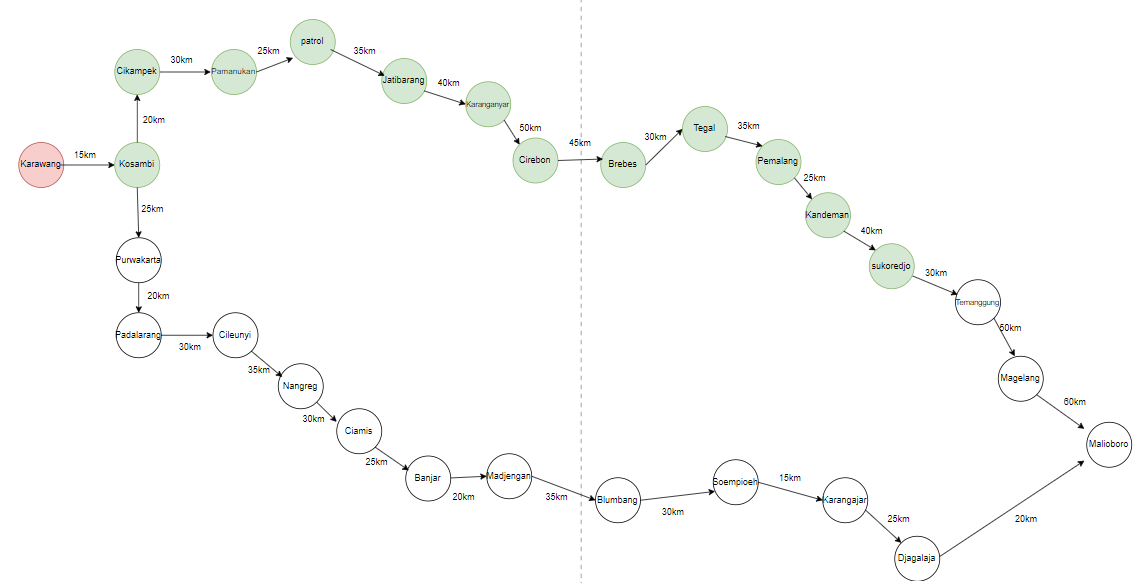
4.1. 11 Rute Pemalang ke Kandeman

Selanjutnya memilih sisi yang berdekatan dengan Kandeman, yaitu Sukoredjo dengan total bobot jaraknya adalah 390km. Seperti pada Gambar 4.1.12.



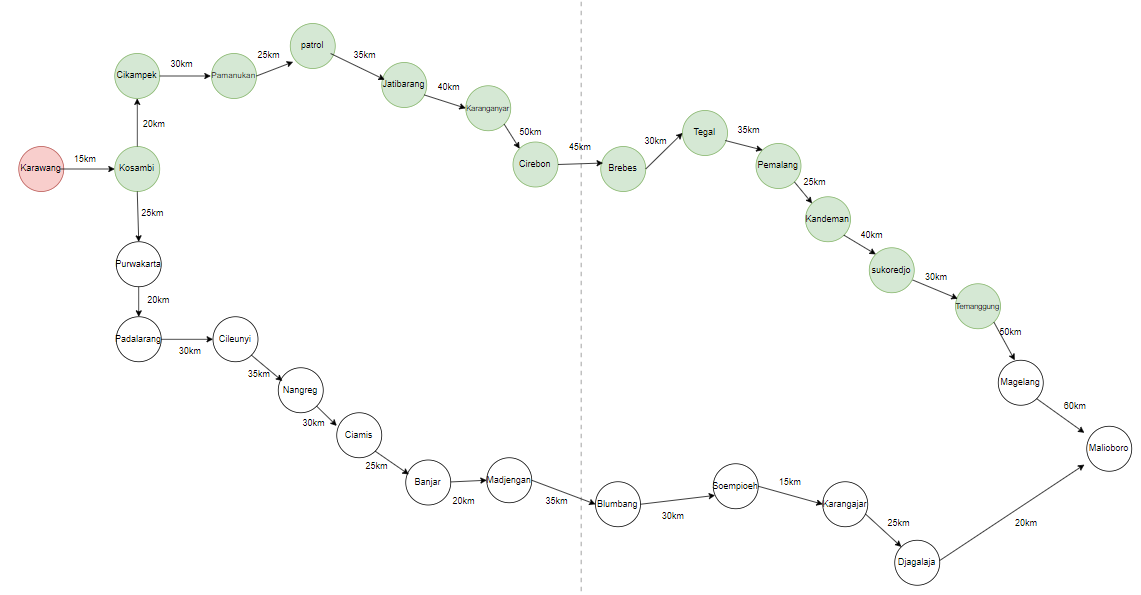
4.1. 12 Rute Kandeman ke Sukoredjo

Selanjutnya memilih sisi yang berdekatan dengan Sukerdjo, yaitu Temanggung dengan total bobot jaraknya adalah 420km. Seperti pada Gambar 4.1.13.



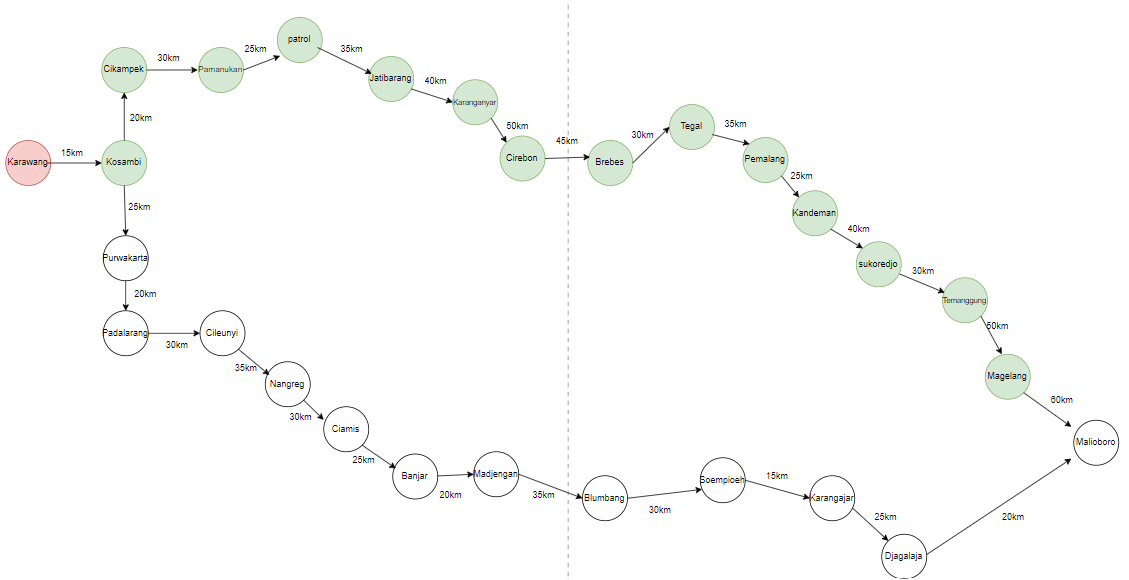
4.1. 13 Rute Sukoredjo ke Temanggung

Selanjutnya memilih sisi yang berdekatan dengan Temanggung, yaitu Magelang dengan total bobot jaraknya adalah 470km. Seperti pada Gambar 4.1.14.



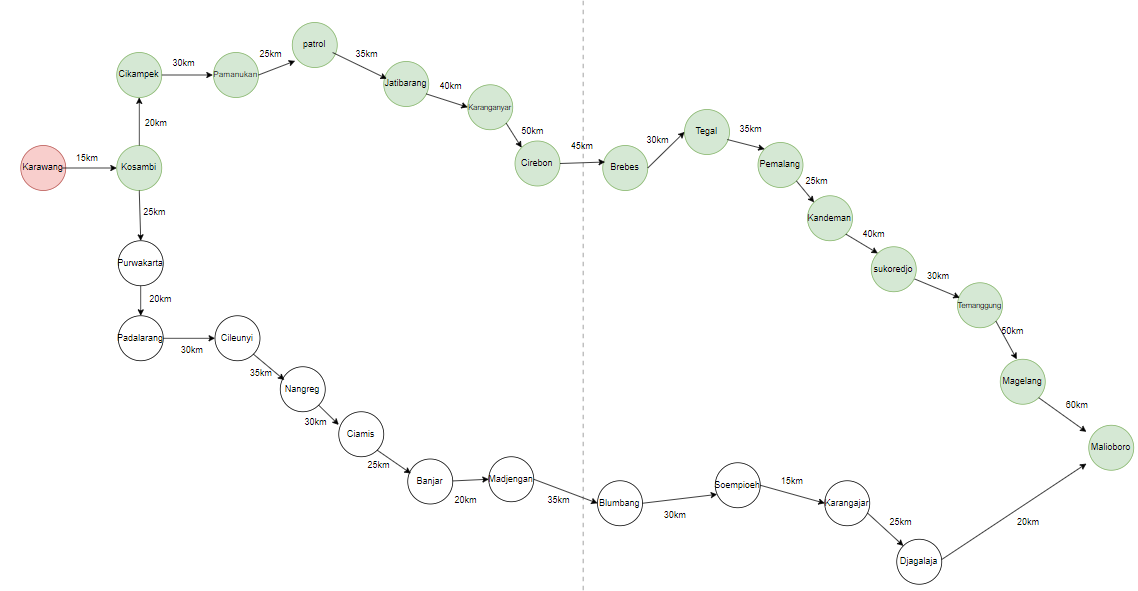
4.1. 14 Rute Temanggung ke Magelang

Selanjutnya memilih sisi yang berdekatan dengan Magelang, yaitu Malioboro dengan total bobot jaraknya adalah 530km. Seperti pada Gambar 4.1.15.



4.1. 15 Rute Magelang ke Malioboro

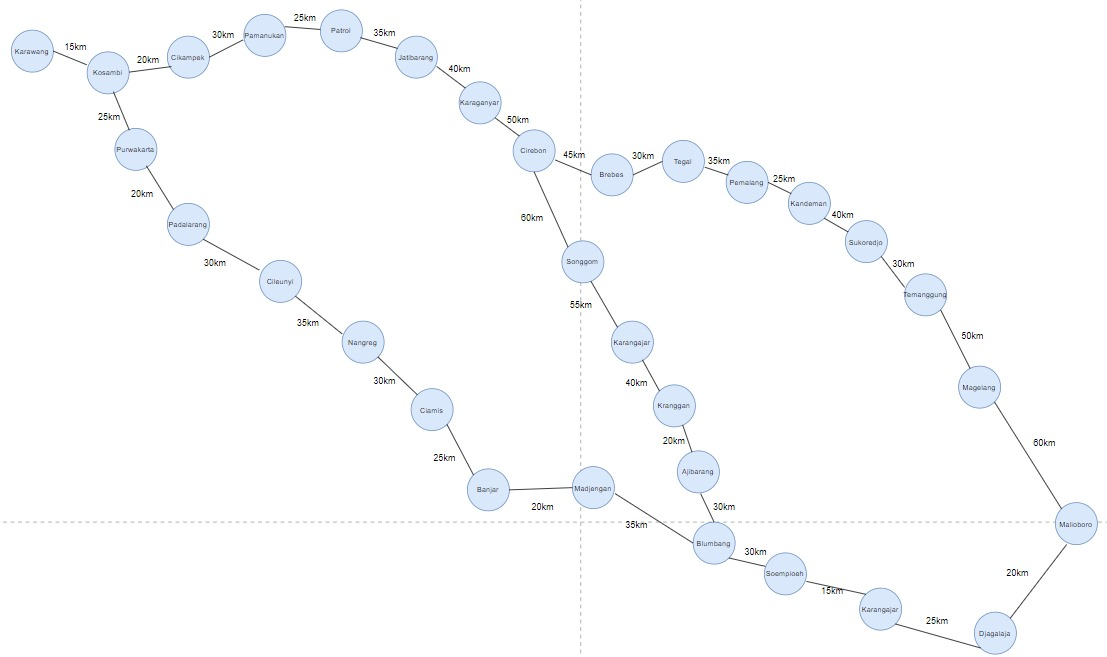
Jadi rute yang optimal dengan jarak terpendek menggunakan Greedy adalah Karawang -> Kosambi -> Cikampek -> Pamanukan -> Patrol -> Jatibarang -> Karanganyar -> Cirebon -> Brebes -> Tegal -> Pemalang -> Kandeman -> Sukoredjo -> Temanggung -> Magelang -> Malioboro dengan total jarak tempuhnya adalah 530km.



4.1. 16 Rute Karawang ke Malioboro

## **4.2 Hasil Pengujian Jarak**

Untuk algoritma Greedy, langkah pertama adalah menentukan titik awal. Setelah memilih titik awal, misalnya Karawang, klik tombol "Cari jalur Terdekat", yang akan menampilkan hasil jalur terdekat dan terpendek dari titik awal. Hasil pencarian akan menampilkan rute dalam bentuk peta, serta jarak, waktu tempuh, dan jumlah titik yang dilalui yang dibutuhkan algoritma untuk menyelesaikan proses pencarian.

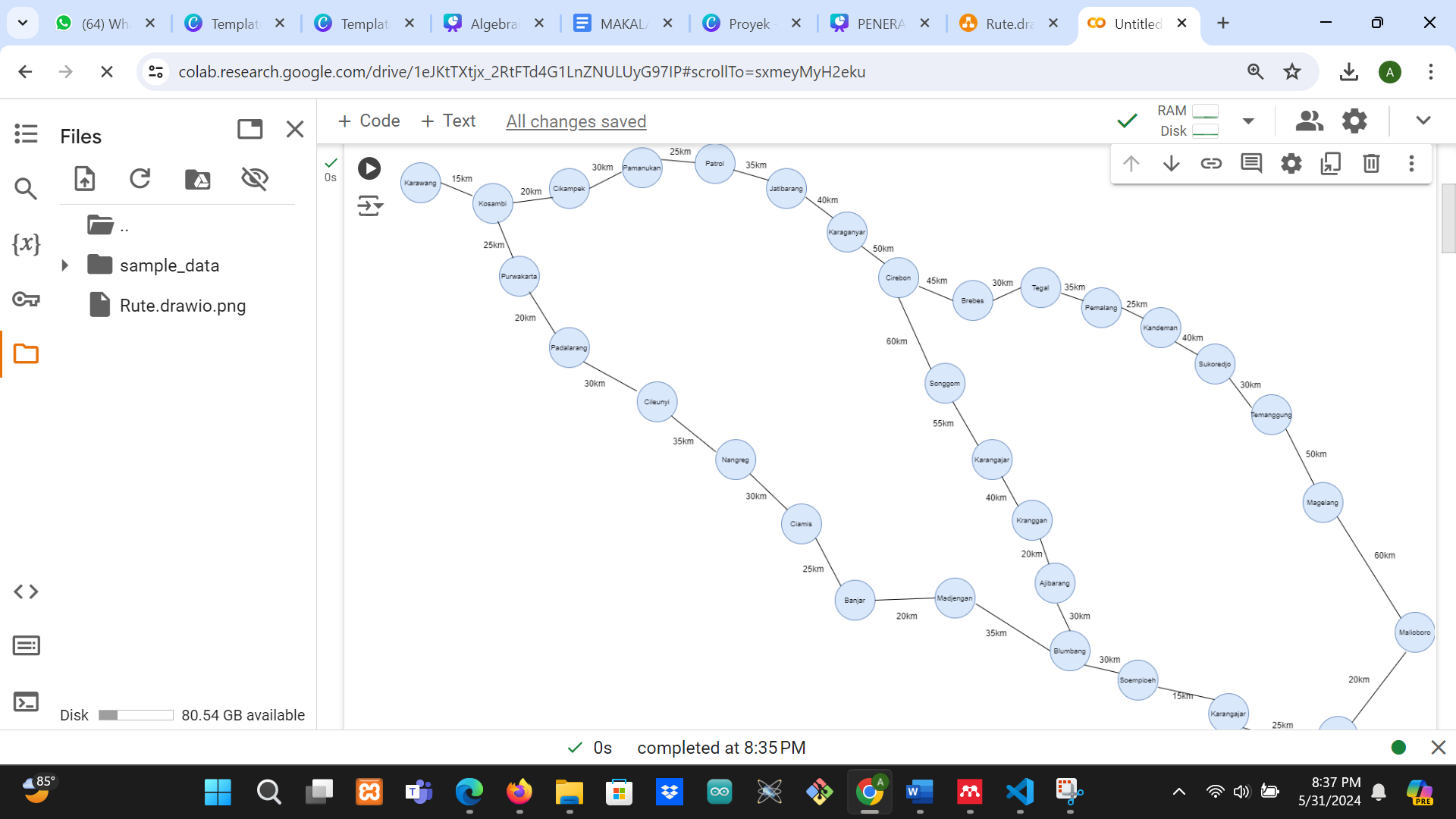


4.2. 1 Graf Rute Karawang hingga ke Malioboro

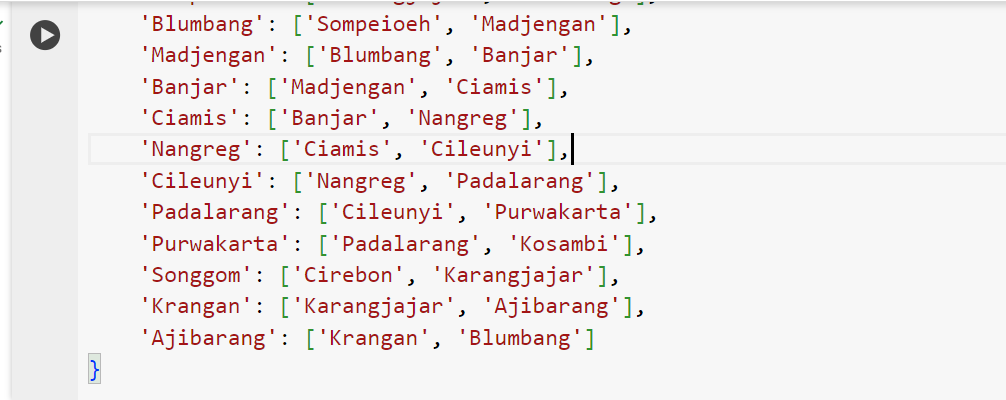
## **Implementasi Menggunakan Pemrograman python**

Berikut adalah hasil implementasi menggunakan pemograman python untuk mencari rute terdekat yang telah dibuat, kita memiliki sejumlah rute perjalanan yang menghubungkan beberapa kota, dan kita ingin menemukan rute tercepat berdasarkan jarak total yang ditempuh. Langkah-langkah yang dilakukan adalah sebagai berikut:





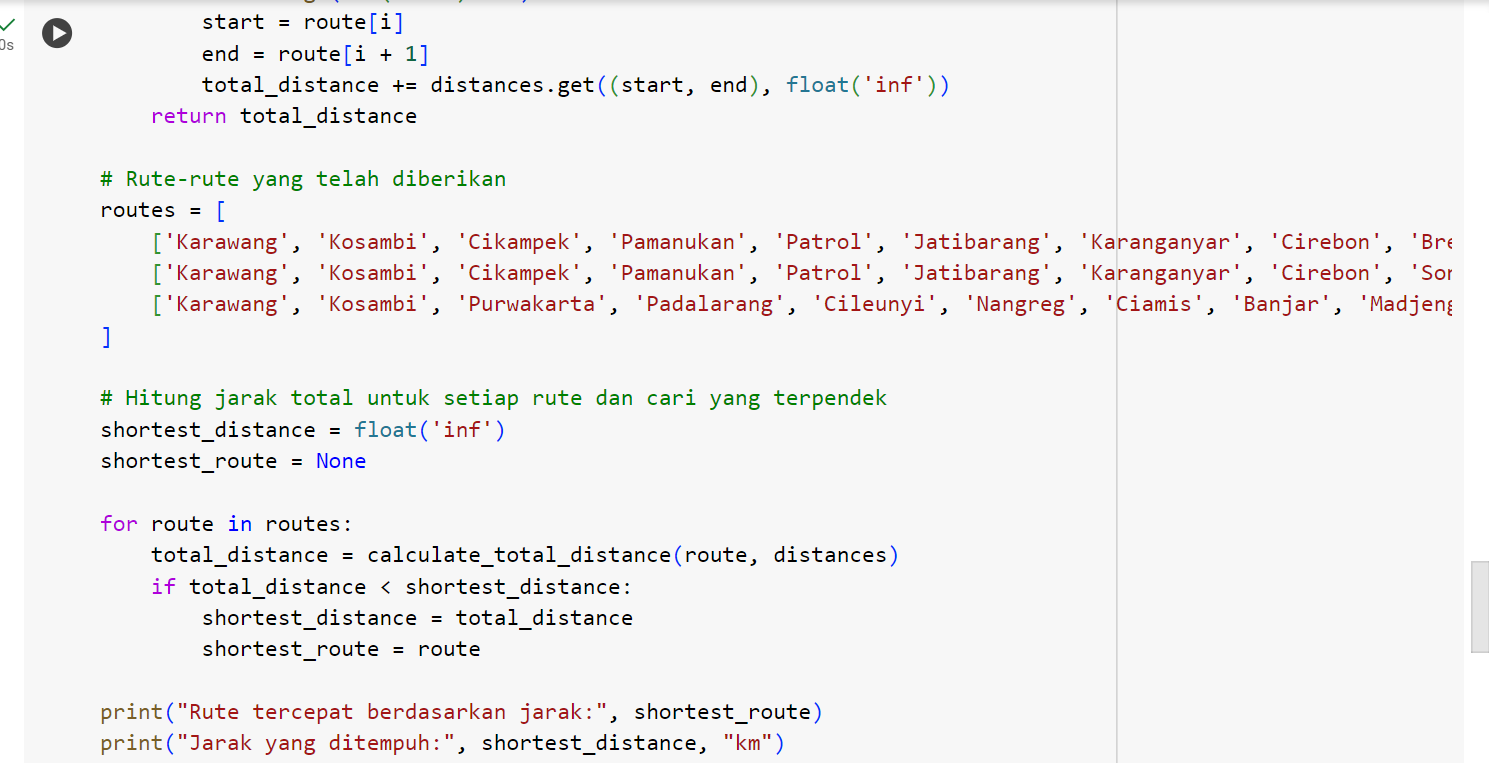






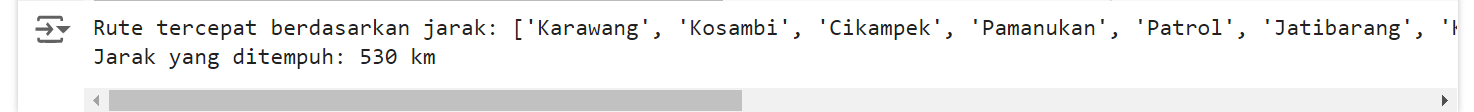






4.3. 2Implementasi dalam bentuk codingan python

Jadi rute yang optimal dengan jarak terpendek menggunakan Greedy adalah Karawang -> Kosambi -> Cikampek -> Pamanukan -> Patrol -> Jatibarang -> Karanganyar -> Cirebon -> Brebes -> Tegal -> Pemalang -> Kandeman -> Sukoredjo -> Temanggung -> Magelang -> Malioboro dengan total jarak tempuhnya adalah 530km.



4.3. 3 Output dari implemntasi dalm bentuk codingan python

# BAB V

# PENUTUP

## **5.1 Kesimpulan**

Penerapan algoritma greedy dalam menentukan rute terpendek dari Kota Karawang ke Kota Malioboro memberikan solusi yang cepat dan cukup efisien untuk masalah ini. Algoritma greedy memilih jalan dengan jarak terpendek pada setiap langkahnya tanpa memperhitungkan keseluruhan rute, yang membuatnya sangat cepat dalam pengambilan keputusan. Namun, penting untuk dicatat bahwa meskipun algoritma greedy dapat menemukan rute yang cukup baik, ia tidak selalu menjamin solusi optimal dalam setiap kasus. Dalam konteks rute dari Karawang ke Bandung, algoritma ini berhasil memberikan rute yang lebih pendek dengan beberapa asumsi dan kondisi tertentu, seperti kondisi jalan dan ketersediaan informasi jarak yang akurat.

## **5.2 Saran**

Saran yang perlu dikembangkan dalam penelitian tersebut adalah sebagai berikut:

1. Algoritma ini mempertimbangkan keseluruhan rute, bukan hanya keputusan lokal.
2. Pastikan informasi jarak dan kondisi jalan selalu akurat dan perhatikan faktor-faktor seperti lalu lintas,kondisi jalan dll.
3. Lakukan pengujian dan validasi rute yang dihasilkan pada berbagai waktu dan kondisi serta Pastikan konsistensi dan keandalan rute dalam berbagai situasi.
4. Model ini memanfaatkan kecepatan greedy dan keakuratan algoritma lain untuk menghasilkan rute yang lebih optimal dan efisien.

# DAFTAR PUSTAKA

Darnita, Y., & Toyib, R. (2019). Penerapan Algoritma Greedy Dalam Pencarian Jalur Terpendek… Penerapan Algoritma Greedy Dalam Pencarian Jalur Terpendek Pada Instansi-Instasi Penting Di Kota Argamakmur Kabupaten Bengkulu Utara. In *Jurnal Media Infotama* (Vol. 15, Issue 2).

Hayati, E. N., & Yohanes, A. (2014). *Seminar Nasional IENACO-2014*.

Juniar, A., Juniar Jurusan Sistem Informasi, A., & Tinggi Manajemen Industri -Kementerian Perindustrian Jl, S. (2015). Penerapan Algoritma Greedy pada Penjadwalan Produksi Single-Stage dengan Parallel Machine di Industri Konveksi. *OKTOBER 2015 IJCCS*, *16*, 1–5.

Nova Arviantino, F., Gata, W., Kurniawati, L., Setiawan, Y. A., & Priansyah, D. (2021). Penerapan Algoritma Greedy Dalam Pencarian Jalur Terpendek Pada Masjid–Masjid Di Kota Samarinda. *METIK JURNAL*, *5*(1), 8–11. <https://doi.org/10.47002/metik.v5i1.188>

Oktaviana, S., & Naufal, A. (2017). Algoritma Greedy untuk Optimalisasi Ruangan dalam Penyusunan Jadwal Perkuliahan. In *Dipublikasikan Mei*.