

Perbandingan Algoritma UCS, GBFS, dan A* dalam menentukan Rute Terpendek menuju Labtek V Ganesha

Rafiki Prawhira Harianto - 13522065
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): rafikiprawhira@gmail.com

Abstract—Makalah ini membandingkan implementasi Algoritma route planning—UCS, GBFS, dan A*—dalam menentukan rute terpendek menuju Labtek V dalam kampus Institut Teknologi Bandung (ITB) Ganesha, yang merupakan fasilitas penting bagi civitas ITB.

Keywords—Algoritma Route Planning, UCS, GBFS, A*, Labtek V ITB

I. PENDAHULUAN

Dalam era modern yang didominasi oleh perkembangan teknologi, kebutuhan akan algoritma yang efisien untuk menentukan rute terpendek menjadi sangat vital. Penentuan rute terpendek ini tidak hanya diterapkan dalam konteks jaringan komputer, tetapi juga dalam navigasi dunia nyata, seperti dalam perjalanan menuju lokasi tertentu. Salah satu lokasi penting di Institut Teknologi Bandung adalah Labtek V, yang sering menjadi tujuan mahasiswa dan civitas ITB, terutama bagi fakultas STEI (Sekolah Teknik Elektro dan Informatika). Menemukan rute tercepat menuju Labtek V dapat menghemat waktu dan sumber daya, terutama bagi mahasiswa dan dosen yang ingin menggunakan waktunya secara efektif.

Algoritma-algoritma seperti Uniform Cost Search (UCS), Greedy Best-First Search (GBFS), dan A* adalah beberapa pendekatan yang digunakan untuk memecahkan masalah rute terpendek. Ketiga algoritma ini menawarkan metode yang berbeda dalam menentukan jalur terbaik, masing-masing dengan kelebihan dan kekurangan tersendiri. UCS, misalnya, menjamin solusi optimal dengan mencari jalur berdasarkan biaya yang seragam, namun sering kali membutuhkan waktu komputasi yang besar. GBFS lebih cepat dalam menemukan solusi, tetapi tidak selalu memberikan rute optimal karena hanya mempertimbangkan keuntungan jangka pendek. A* menggabungkan kelebihan kedua pendekatan tersebut dengan menggunakan fungsi heuristik yang mengarahkan pencarian menuju solusi optimal lebih efisien.

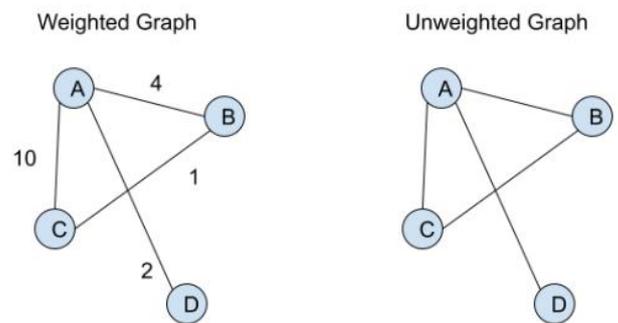
Pemilihan algoritma yang tepat untuk menentukan rute terpendek menuju Labtek V Ganesha dapat memberikan wawasan yang berharga tidak hanya bagi pengembangan sistem navigasi, tetapi juga bagi implementasi teknologi yang

lebih luas dalam berbagai bidang, seperti transportasi dan jaringan komputer. Melalui perbandingan performa UCS, GBFS, dan A*, penelitian ini bertujuan untuk mengidentifikasi algoritma yang paling efisien dan efektif dalam mencari rute di lingkungan kampus ITB, serta sebagai contoh kasus penggunaan UCS, GBFS, dan A* dalam konteks yang sederhana.

II. LANDASAN TEORI

A. Graf Berbobot

Graf adalah struktur data yang digunakan untuk menggambarkan hubungan antara objek-objek diskrit. Dalam graf, terdapat simpul (node) yang mewakili objek dan tepi (edge) yang menghubungkan simpul-simpul tersebut. Graf berbobot adalah graf di mana setiap tepinya memiliki nilai tertentu yang menunjukkan harga atau bobot.



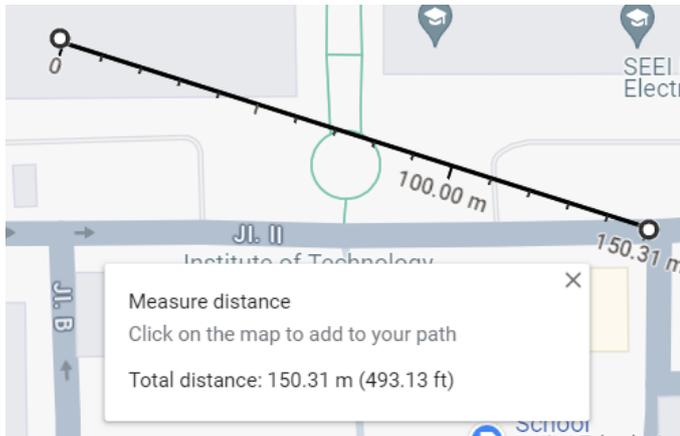
Gambar 1: Contoh Graf berbobot dan Graf tidak berbobot
Sumber: Homepage Rinaldi Muni

Contoh graf berbobot dapat dilihat pada Gambar 1, yang menunjukkan perbandingan antara graf berbobot dan graf tanpa bobot. Graf berbobot ini penting dalam konteks jaringan lintasan antara lokasi-lokasi di sekitar kampus ITB, di mana bobot pada tepi-tepi graf tersebut mencerminkan jarak atau biaya perpindahan dari satu lokasi ke lokasi lainnya.

Simpul-simpul dalam graf ini merepresentasikan jalan-jalan strategis di kampus yang strategis dekat Labtek V.

Tepi-tepi graf menghubungkan simpul-simpul ini dan diberi bobot berupa jarak atau waktu tempuh antar lokasi. Algoritma route planning seperti UCS, GBFS, dan A* dapat menangani graf yang memiliki bobot pada tepi-tepi, karena bobot tersebut memungkinkan algoritma untuk mempertimbangkan variabel seperti jarak dan biaya. Hasilnya adalah lintasan terpendek berdasarkan bobot yang ada.

Dalam pembuatan graf berbobot seperti pada graf Labtek V, Google Maps adalah alat yang bermanfaat. Google Maps menyediakan peta rinci dengan citra satelit untuk berbagai lokasi di dunia, dan fitur *Measure Distance* yang memungkinkan pengguna untuk mengukur jarak antara dua titik. Fitur ini dapat digunakan untuk menentukan bobot pada tepi-tepi graf berbobot.



Gambar 2: Contoh Penggunaan Measure Distance
Sumber: Google Maps

Selain tepi, graf berbobot juga memiliki simpul. Dalam konteks ini, simpul merupakan titik-titik strategis pada peta yang penting untuk mencapai Labtek V. Simpul utama adalah Labtek V itu sendiri, diberi nomor 0. Simpul penting lainnya termasuk Parkir Sipil (1), Gerbang Depan (2), Parkir SR (3), dan Gerbang Tamansari (4), yang merupakan pintu-pintu masuk ITB. Terdapat juga 10 simpul lainnya (nomor 5-15) yang merupakan titik-titik strategis lainnya di sekitar Labtek V.

Dengan informasi ini, dibuatlah graf berbobot yang menggambarkan peta menuju Labtek V, seperti yang terlampir. Selain itu, juga terdapat graf heuristik yang akan dijelaskan lebih lanjut pada bagian berikutnya.

B. Algoritma Route Planning

Terdapat dua kategori algoritma route planning: Uninformed search, yaitu pencarian tanpa informasi tambahan graf, serta Informed search, pencarian dengan informasi tambahan graf. Algoritma yang dipakai dalam makalah ini adalah UCS (Uniform Cost Search) yang termasuk Uninformed search, serta GBFS (Greedy Best First Search) dan A* Search yang termasuk Informed search.

UCS, GBFS, dan A* Search memiliki algoritma umum yang serupa. Perbedaan ketiga algoritma tersebut berupa nilai cost atau $f(n)$. Terdapat 2 variabel utama dalam cost,

yaitu $g(n)$ dan $h(n)$. $g(n)$ ditentukan berdasarkan jumlah perubahan dari awal hingga node, sedangkan $h(n)$ berupa fungsi heuristik yang *admissible*. Dalam makalah ini, kedua variabel tersebut berupa:

- $g(n)$: Jumlah perubahan dari node awal hingga node n
- $h(n)$: Jarak lurus dari node n menuju Labtek V

Hal yang perlu diperhatikan dalam menentukan $h(n)$ berupa syaratnya, yaitu fungsi heuristik $h(n)$ harus *admissible*. Dalam kasus ini, $h(n)$ termasuk *admissible* karena $h(n)$ selalu *overestimate* jarak asli node n dengan node akhir (Labtek V), yaitu $h(n)$ selalu memiliki nilai yang lebih kecil dari jarak aslinya. Hal ini dapat dilihat karena jarak lurus dari poin A ke poin B selalu lebih dekat dari jarak dari poin A ke poin B yang melalui poin lain terlebih dahulu, sehingga jarak asli pasti lebih besar dari $h(n)$. Sesuai teorema, karena $h(n)$ *admissible*, pencarian algoritma A* terjamin optimal. Heuristik atau $h(n)$ yang digunakan digambarkan dalam bentuk graf, yang dapat dilihat pada Gambar 5 pada lampiran.

Node diproses berdasarkan prioritas *cost* yang paling kecil. Nilai cost dari masing-masing algoritma adalah sebagai berikut:

- UCS: $f(n) = g(n)$
- GBFS: $f(n) = h(n)$
- A*: $f(n) = g(n) + h(n)$

Secara teoritis, algoritma A* lebih efisien daripada UCS. Karena $h(n)$ *admissible*, A* menjamin keoptimalan solusi dan karena ada cost heuristik $h(n)$ membuat pencarian A* lebih terarah menuju rute akhir. Namun, GBFS tidak menjamin keoptimalan solusi. GBFS tidak ada cost $g(n)$, membuat lingkup pencarian sangat terfokus terhadap rute terpendek berikutnya dan membuat pencarian yang cepat, tetapi tidak adanya cost $g(n)$ membuat GBFS pencarian yang tidak sistematis dan mudah terperangkap dalam minimum lokal.

III. PEMROGRAMAN RUTE TERPENDEK

Program yang dibuat merupakan program berbasis java. Program dibagi menjadi 3 file, Main.java, Node.java, dan RouteQueue.java. Main berupa titik masuk program dan berisi pula algoritma yang dipakai. Algoritma dalam program hanya membandingkan $f(n)$, dengan kelas turunan dari algoritma, yaitu UCS, GBFS, dan Astar, mendefinisikan $f(n)$ sesuai algoritmanya masing-masing.

Node berupa struktur data utama dari program, yang berisi info, nilai $g(n)$ dan $h(n)$, dan thread yang menyimpan nilai node-node sebelumnya yang telah ditelusuri. Terakhir, RouteQueue berupa kelas yang mengambil kelas Algoritma sebagai parameter dan menjalankan route planning dengan konsep struktur data PriorityQueue. RouteQueue menyimpan pula graf ITB sebagai matriks graf integer berukuran $N \times N$, dengan setiap cell merepresentasikan jarak, dan array heuristik sebagai array of integer berukuran N .

Kode program dapat diakses melalui tautan repository *github* pada lampiran. Cara penggunaan program dapat dilihat melalui file *readme* yang tertera. Pengguna dapat mengubah variabel *graph*, *heuristik*, dan *MAX_NODE* jika ingin mengubah program untuk keperluan graf dan peta yang berbeda.

	visited: 6 Jumlah jarak: 370m Waktu eksekusi: 11ms	visited: 6 Jumlah jarak: 370m Waktu eksekusi: 12ms	visited: 6 Jumlah jarak: 370m Waktu eksekusi: 17ms
--	---	---	---

Tabel 1: Hasil Pengujian

```
Selamat datang ke program rute terpendek menuju labtek V!
Mau mulai dari node berapa? [0-15]: 1
===== List Algoritma =====
1: UCS (Uniform Cost Search)
2: GBFS (Greedy Best First Search)
3: A* Search
=====
Algorithm yang dipakai: 3
Memakai A* Search...
1 -> 9 -> 0
Jumlah visited: 7
Jumlah jarak: 368m
Waktu eksekusi: 15ms
PS D:\Rafiki-ITB\Kuliah\Semester 4\Stima\Makalah-Stima >
```

Gambar 3: Contoh Penggunaan Program

IV. HASIL PROGRAM

Menggunakan program, dicari semua rute menuju labtek V dari gerbang-gerbang utama di ITB: Parkir Sipil, Gerbang Depan, Parkir SR, dan Gerbang Tamansari. Hasil dari pengujian berada di Tabel 1.

Hasil pengujian melihat bahwa hasil rute yang dihasilkan algoritma UCS, GBFS, maupun A* sama. Secara teori, GBFS tidak menjamin menghasilkan rute yang optimal dengan imbalan jumlah node yang dihasilkan lebih sedikit. Namun, pengujian menunjukkan bahwa GBFS berhasil menemukan rute yang optimal dalam graf ITB Labtek V. Hal yang mungkin menyebabkan GBFS pada makalah ini optimal adalah heuristik yang tepat, dan graf yang hanya mengambil sampel jalan-jalan strategis yang menunjuk ke Labtek V. Algoritma A* juga berhasil menghasilkan rute yang optimal, dengan jumlah visited yang lebih efisien dibandingkan UCS.

Untuk rute terpendek menuju Labtek V, jarak terpendek untuk semua algoritma sama. Jarak terpendek melalui Parkir Sipil adalah 368m, untuk Gerbang Depan 341m, untuk Parkir SR 411m, dan untuk Gerbang Tamansari 370m. Rute yang dihasilkan pula sama, dan dapat dilihat melalui mencocokkan rute pada Tabel 1 dengan graf Labtek V pada lampiran. Untuk waktu eksekusi, tidak ada perbedaan yang signifikan karena kompleksitas waktu algoritma yang murah dan graf yang relatif sederhana.

V. KESIMPULAN

Makalah ini telah membandingkan tiga algoritma route planning—Uniform Cost Search (UCS), Greedy Best-First Search (GBFS), dan A*—dalam menentukan rute terpendek menuju Labtek V di Institut Teknologi Bandung. Hasil pengujian menunjukkan bahwa ketiga algoritma tersebut berhasil menemukan rute yang optimal, dengan jarak dan rute yang sama. Meski secara teoritis GBFS tidak selalu menjamin solusi optimal, pada kasus ini GBFS memberikan hasil yang optimal, kemungkinan disebabkan oleh penggunaan heuristik yang tepat dan kompleksitas graf yang relatif sederhana. Algoritma A* terbukti efisien dengan menggabungkan keuntungan UCS dan GBFS, Sementara itu, UCS, meskipun menjamin solusi optimal, membutuhkan waktu komputasi yang lebih besar dibandingkan A* dan GBFS. Hasil pengujian menunjukkan GBFS dapat digunakan sebagai algoritma yang paling efisien dan efektif untuk implementasi route planning dalam konteks lingkungan kampus ITB.

VI. UCAPAN TERIMA KASIH

Puji syukur penulis sampaikan kepada Allah swt, karena berkat Rahmat-Nya, penulis mampu menyelesaikan makalah ini dengan baik dan tepat waktu.

Penulis makalah juga mengucapkan terima kasih kepada Bapak Rinaldi Munir sebagai dosen K01 untuk mata kuliah Strategi Algoritma IF2211, dan dalam websitenya yang lengkap terkait berbagai mata kuliah dalam Informatika yang sangat membantu dalam pembelajaran mandiri.

TestCase	UCS	GBFS	A*
Parkir Sipil (1)	1 -> 9 -> 0 Jumlah visited: 12 Jumlah jarak: 368m Waktu eksekusi: 11ms	1 -> 9 -> 0 Jumlah visited: 6 Jumlah jarak: 368m Waktu eksekusi: 15ms	1 -> 9 -> 0 Jumlah visited: 7 Jumlah jarak: 368m Waktu eksekusi: 14ms
Gerbang Depan (2)	2 -> 11 -> 10 -> 9 -> 0 Jumlah visited: 10 Jumlah jarak: 341m Waktu eksekusi: 11ms	2 -> 11 -> 10 -> 9 -> 0 Jumlah visited: 9 Jumlah jarak: 341m Waktu eksekusi: 11ms	2 -> 11 -> 10 -> 9 -> 0 Jumlah visited: 9 Jumlah jarak: 341m Waktu eksekusi: 12ms
Parkir SR (3)	3 -> 12 -> 13 -> 9 -> 0 Jumlah visited: 10 Jumlah jarak: 411m Waktu eksekusi: 12ms	3 -> 12 -> 13 -> 9 -> 0 Jumlah visited: 9 Jumlah jarak: 411m Waktu eksekusi: 13ms	3 -> 12 -> 13 -> 9 -> 0 Jumlah visited: 10 Jumlah jarak: 411m Waktu eksekusi: 10ms
Gerbang Tamansari (4)	4 -> 14 -> 15 -> 0 Jumlah	4 -> 14 -> 15 -> 0 Jumlah	4 -> 14 -> 15 -> 0 Jumlah

Terakhir, penulis berterima kasih yang sebesar-besarnya kepada kedua orang tua yang telah membesarkan saya dengan kasih sayang dan memberi nafkah untuk kuliah. Penulis berterima kasih kepada semua pihak yang terlibat dalam penulisan makalah ini atas segala bantuan dan dukungannya.

REFERENSI

- [1] Munir, Rinaldi. Homepage Rinaldi Munir. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/19-Graf-Bagian1-2023.pdf>. Diakses pada 12 Juni 2024.
- [2] Munir, Rinaldi. Homepage Rinaldi Munir. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Route-Planning-Bagian1-2021.pdf>. Diakses pada 12 Juni 2024.
- [3] Munir, Rinaldi. Homepage Rinaldi Munir. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Route-Planning-Bagian2-2021.pdf>. Diakses pada 12 Juni 2024.

- [4] Google Maps. <https://www.google.com/maps/>. Diakses pada 11 Juni 2024.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

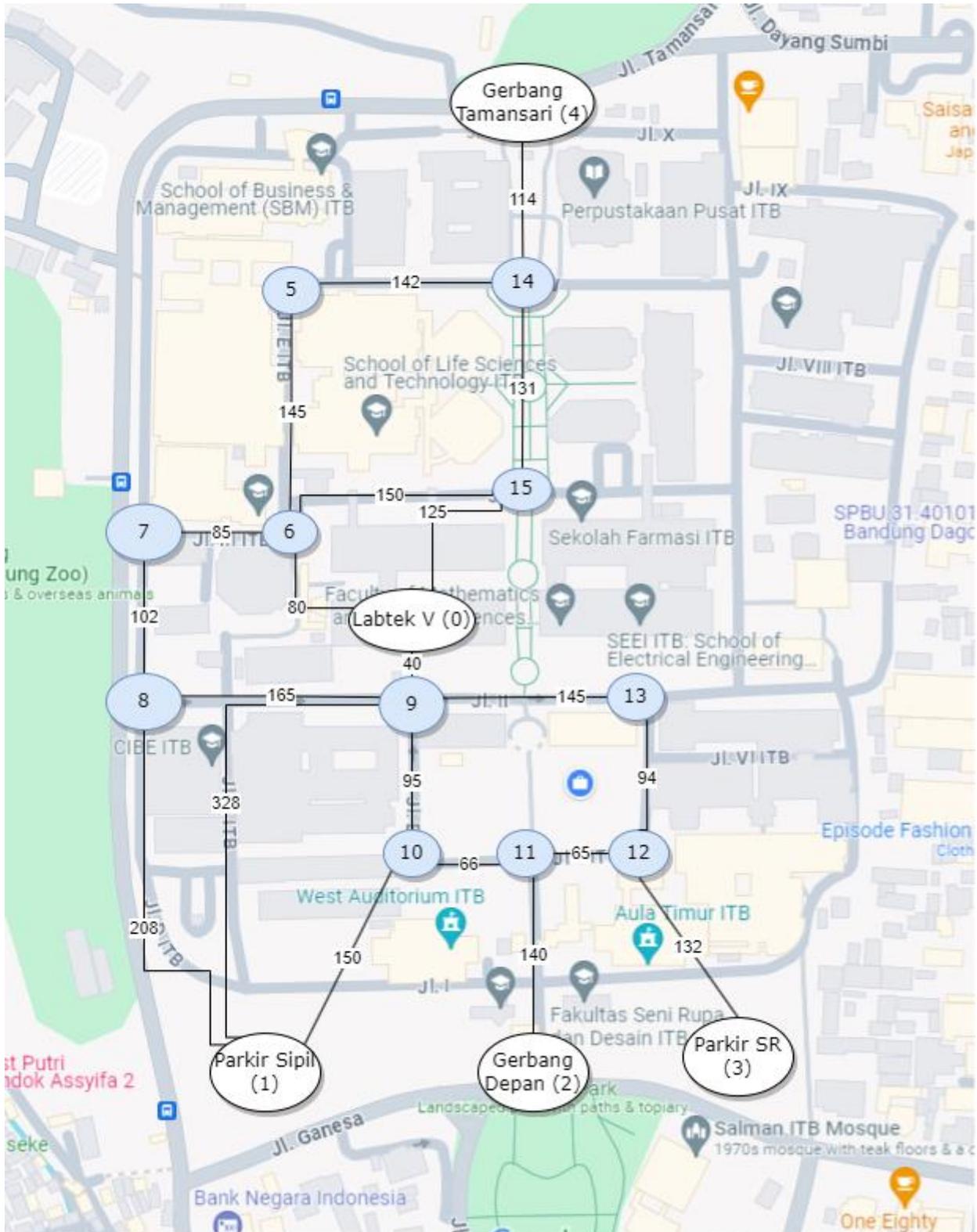
Bandung, 12 Juni 2024



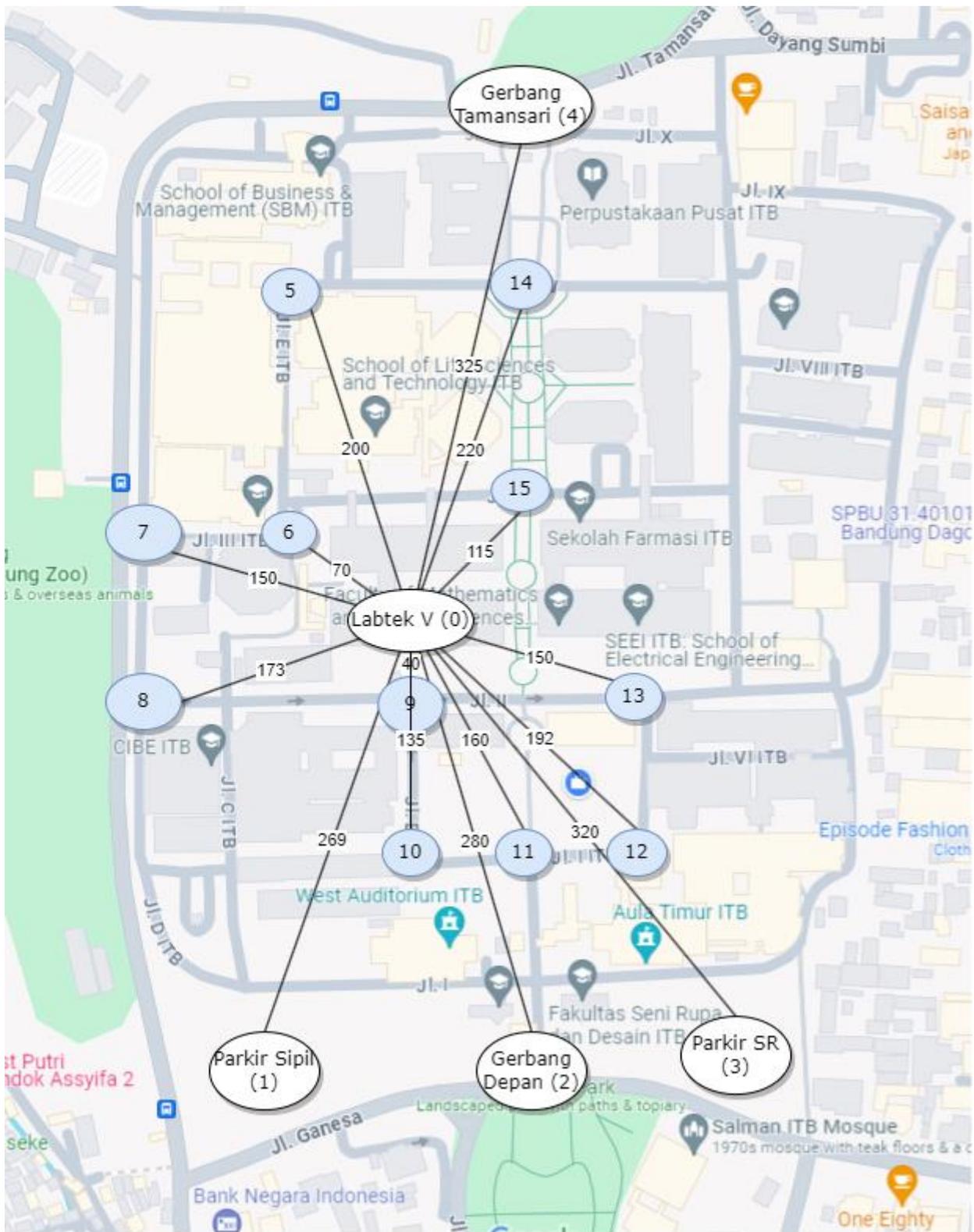
Rafiki Prawhira Harianto
13522065

VII. LAMPIRAN

- Tautan Repository Program: <https://github.com/Intermaze/Makalah-Stima>



Gambar 4: Graf Labtek V



Gambar 5: Graf Heuristik Labtek V