

Aplikasi Algoritma Dijkstra dalam menentukan Rute Terpendek Menuju Labtek V ITB Ganesha

Rafiki Prawhira Harianto - 13522065

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13522065@itb.ac.id

Abstract—Makalah ini mengeksplorasi implementasi Algoritma Dijkstra dalam menentukan rute terpendek menuju Labtek V dalam kampus Institut Teknologi Bandung (ITB) Ganesha, yang merupakan fasilitas penting bagi civitas ITB, terutama prodi STEI, yang mengambil manfaat dari navigasi yang efisien menuju Labtek V dalam aktivitas akademiknya sehari-hari.

Keywords—Rute Terpendek, Algoritma Dijkstra, Labtek V ITB

I. PENDAHULUAN

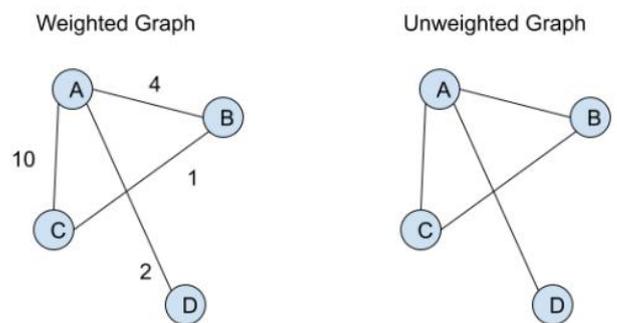
Teknologi informasi memiliki peran penting dalam mendukung efisiensi dan produktivitas di berbagai bidang, termasuk transportasi dan logistik. Seiring dengan bertambah pentingnya waktu, penentuan rute terpendek menjadi suatu kebutuhan esensial. Makalah ini bertujuan untuk mengeksplorasi penerapan algoritma Dijkstra dalam konteks spesifik, yaitu menentukan rute terpendek menuju Labtek V di Institut Teknologi Bandung (ITB) Ganesha.

Labtek V, sebagai salah satu tempat belajar dan riset utama di ITB, menjadi pusat aktivitas mahasiswa dan dosen, terutama dalam program studi STEI (Sekolah Teknik Elektro dan Informatika). Menemukan rute terpendek menuju lab ini dapat mempermudah civitas akademik dalam melaksanakan kegiatannya tepat waktu. Algoritma Dijkstra, yang dikenal efektif dalam menentukan lintasan terpendek dalam graf berbobot, menjadi pilihan yang tepat untuk diaplikasikan dalam konteks ini.

Melalui makalah ini, akan dijelaskan konsep dasar algoritma Dijkstra dan bagaimana penerapannya dapat memberikan solusi untuk menemukan rute terpendek ke Labtek V ITB. Diharapkan makalah ini dapat memberikan wawasan tentang potensi dan keterbatasan algoritma Dijkstra dalam konteks penentuan rute di lingkungan kampus, dengan fokus pada kepentingan praktis dalam mendukung mobilitas di dalam kampus ITB.

II. GRAF BERBOBOT

Graf adalah struktur data yang digunakan untuk menyatakan hubungan antara objek-objek diskrit. Graf terdiri dari berbagai simpul (node) dan tepi (edge). Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot).



Gambar 1: Contoh Weighted graph (Graf berbobot) dan Unweighted graf (Graf tidak berbobot).

Sumber:

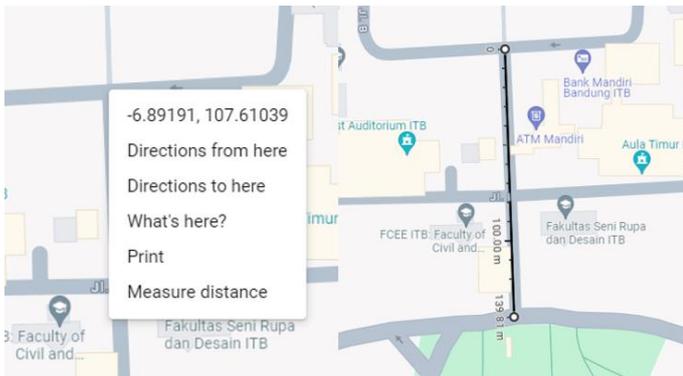
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/19-Graf-Bagian1-2023.pdf>

Dalam konteks ini, graf berbobot merepresentasikan jaringan lintasan antara lokasi-lokasi di sekitar kampus ITB, dan bobot pada tepi-tepi graf tersebut mencerminkan jarak atau “biaya” perpindahan dari satu lokasi ke lokasi lainnya.

Simpul-simpul dalam graf ini dapat dianggap sebagai lokasi-lokasi strategis di kampus, seperti gedung perkuliahan, pusat kegiatan mahasiswa, dan Labtek V. Tepi-tepi graf menghubungkan simpul-simpul ini dan diberi bobot berupa jarak atau waktu tempuh antarlokasi.

Graf berbobot menjadi penting dalam konteks ini karena algoritma Dijkstra dirancang khusus untuk menangani graf yang memiliki bobot pada tepi-tepi graf. Pemberian bobot memungkinkan algoritma untuk mempertimbangkan variabel-variabel seperti jarak atau biaya, sehingga hasil akhirnya adalah lintasan terpendek berdasarkan bobot tersebut.

Dalam pembuatan Graf berbobot dalam suatu peta, bermanfaat untuk menggunakan Google Maps. Google Maps adalah alat berbasis web yang dikembangkan oleh Google, yang menyediakan peta rinci dengan citra satelit untuk berbagai lokasi di dunia. Selain akses mudah terhadap peta rinci ITB Ganesha, Google Map menyediakan fitur yang bermanfaat dalam konteks ini, yaitu *Measure Distance*.



Gambar 2: Contoh Penggunaan Measure Distance dalam Google Maps

Sumber: <https://www.google.com/maps>

Dengan fitur ini, dapat dilakukan pencarian ukuran antara suatu poin pada peta dengan poin lain, memberikan opsi mangkus dan sangkil dalam mencari bobot dalam tepi graf berbobot ini.

Selain tepi, pada graf berbobot terdapat pula simpul. Simpul dalam konteks ini merupakan poin-poin pada peta yang strategis dalam mencapai Labtek V. Simpul utama merupakan Labtek V sendiri, dan diberi nomor 0. Simpul penting lainnya terdapat lima, yaitu Parkir Sipil (1), Gerbang Depan (2), Parkir SR (3), Gerbang Tamansari (4), Gerbang SBM (5), dan Gerbang Batan (6). Kelima simpul ini yang akan dibandingkan dan dianalisis, karena merupakan pintu-pintu masuk ITB. Selain itu, terdapat 9 simpul lainnya bernomor 7-15, yang merupakan poin-poin strategis dekat Labtek V yang dapat dijalaninya oleh manusia.

Dilengkapi dengan pengetahuan tersebut, dihasilkan graf berbobot dengan peta ITB yang terdapat pada lampiran. Selain itu, terdapat pula graf berbobot tanpa peta, yang juga disertakan dalam lampiran.

III. ALGORITMA DIJKSTRA

Algoritma Dijkstra merupakan algoritma yang sangat efektif dalam mencari rute terpendek antar simpul dalam graf berbobot. Algoritma tersebut ditemukan oleh Edsger W. Dijkstra pada tahun 1956 dan diterbitkan 3 tahun kemudian.

Dalam rangka peresmian proyek pembangunan komputer ARMAC di Mathematical Centre, Amsterdam, Dijkstra merancang sebuah program untuk memecahkan masalah yang menarik: Diberikan jaringan jalan yang menghubungkan kota-kota, apa rute terpendek antara dua kota? Algoritma-algoritma terkenal pada waktu itu memiliki waktu eksekusi $O(n^3)$; waktu eksekusi algoritma Dijkstra secepat $O(n^2)$. Dikembangkan dalam waktu 20 menit ketika Dijkstra bersantai di teras kafe bersama tunangannya, Maria (Ria) C. Debets, algoritma Rute Terpendek-nya masih digunakan dalam berbagai aplikasi modern dan dikembangkan menjadi berbagai algoritma ekstensi seperti pencarian A*.

Algoritma Dijkstra bekerja mirip dengan cara *brute-force*, tetapi dalam urutan jalur yang paling cepat. Menggunakan graf berbobot, caranya yaitu:

- 1) Inisialisasi array jarak semua simpul pada graf dengan tak hingga, kecuali simpul 0.
- 2) Kemudian, lihat tetangga simpul 0, jumlahkan jarak pada

simpul dengan tepi 0 dengan masing-masing tetangga.

- 3) Pada setiap tetangga, jika hasilnya lebih kecil dari array jarak[0], masukkan pada array jarak[0] dan urutkan berdasarkan array jaraknya.
- 4) Ketika sudah memproses semua tetangga simpul 0, ulangi dari langkah 2 dengan simpul 1, dan seterusnya sampai semua tetangga simpul telah diproses.
- 5) Keluarkan nilai-nilai pada array jarak.

Dalam implementasi Algoritma Dijkstra dalam kode, digunakan representasi graf dalam matriks. Oleh karena itu, graf berbobot yang dibuat perlu diubah dalam bentuk matriks 16×16 . Hasil graf matriks terdapat pada Gambar 3.

Simpul	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	112	0	108	0	192	0	130	174	120
1	0	0	0	0	0	0	0	0	208	328	150	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	140	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	132	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	114	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200
6	0	0	0	0	0	0	0	159	0	0	0	0	0	0	0	140
7	112	0	0	0	0	0	159	0	102	0	0	0	0	0	0	0
8	0	208	0	0	0	0	0	102	0	0	0	0	0	0	0	0
9	108	328	0	0	0	0	0	0	0	0	95	0	0	145	0	0
10	0	150	0	0	0	0	0	0	0	95	0	66	0	0	0	0
11	192	0	140	0	0	0	0	0	0	0	66	0	65	0	0	0
12	0	0	0	132	0	0	0	0	0	0	0	65	0	94	0	0
13	130	0	0	0	0	0	0	0	0	145	0	0	94	0	0	0
14	174	0	0	0	114	0	0	0	0	0	0	0	0	0	0	0
15	120	0	0	0	0	200	140	0	0	0	0	0	0	0	0	0

Gambar 3: Graf berbobot Rute Labtek V ITB dalam bentuk Matrix 16×16

Kode Algoritma Dijkstra dalam makalah ini digunakan kode C yang disediakan oleh geeksforgeeks.com. Berikut merupakan programnya dalam pseudokode:

```

{Fungsi Pembantu}
{Fungsi pembantu untuk mencari simpul dengan jarak
terpendek dari simpul-simpul yang belum termasuk pohon
rute terpendek}
function minDistance(array of integer: dist, array of
boolean: sptSet, integer : jumlahSimpul) -> integer
{Kamus Lokal}
min, i           : integer
min_index       : integer
{Algoritma}
min <- ∞
min_index <- ∞
i traversal [0..jumlahSimpul-1]
  if (sptSet[i] = false) and (dist[i] <= min) then
    min <- dist[i]
    min_index <- i
-> min_index

{Program Utama}
{Kamus}
graph           : matrix of integer
dist            : array of integer
sptSet          : array of boolean
i, j, u, jumlahSimpul : integer

{Inisialisasi}
graph <- {Masukkan graf berbobot disini}
jumlahSimpul <- 16
dist[0] <- 0
i traversal [1..jumlahSimpul-1]
  dist[i] <- ∞

```

```

{Algoritma}
i traversal [1..jumlahSimpul-2]
  u <- minDistance(dist, sptSet, jumlahSimpul)
  sptSet[u] <- true
  j traversal [0..jumlahSimpul-1]
    if ((sptSet[j] = true) and (graph[u][j] ≠ 0) and (dist[u]
    ≠ ∞) and (dist[u] + graph[u][j] < dist[j])) then
      dist[j] <- dist[u] + graph[u][j]
output(dist)

```

Dari pseudokode, terlihat bahwa jumlahSimpul dapat diatur sesuai dengan spesifikasi graf berbobot. Pada makalah ini, digunakan graf dengan simpul 0-15, sehingga jumlahSimpul adalah 16 simpul.

Selain itu, pseudokode tidak memberikan inisialisasi matriks graf. Hal ini disebabkan graf terlalu besar, dan diinisialisasi dengan cara mengisi setiap nilai matriks dalam bahasa C, yaitu array berisi 16 array baris yang berukuran 16. Untuk itu, disediakan graf matriks pada gambar 3.

Terakhir, terdapat sptSet pada program, yang berfungsi untuk menyimpan informasi simpul-simpul yang telah ditemukan jarak terpendeknya. sptSet kosong saat algoritma mulai.

IV. HASIL PROGRAM

Menggunakan program kode C geeksforgeeks.com dan dengan menggunakan matriks graf berbobot 16x16, dihasilkan luaran program sebagai berikut:

Simpul	Jarak dari Sumber (Labtek V)
0 (Labtek V)	0
1 (Parkir Sipil)	353
2 (Gerbang Depan)	332
3 (Parkir SR)	356
4 (Gerbang Tamansari)	228
5 (Gerbang SBM)	320
6 (Gerbang Batan)	260
7	112
8	214
9	108
10	203
11	192
12	224
13	130
14	174
15	120

Tabel 1: Hasil Program Algoritma Dijkstra

Bedasarkan hasil program, didapatkan bahwa jarak terpendek menuju Labtek V adalah Gerbang Tamansari, dengan jarak 228m dan rute 4-14-0. Tetapi, dalam perjalanan aslinya rute ini seharusnya tidak terlalu efisien. Ketidakpasan ini mungkin terjadi karena graf tidak memperhatikan ketinggian dalam rute antara Gerbang Tamansari dengan poin 14.

Selain itu, jarak terpendek kedua merupakan Gerbang Batan, dengan panjang 260m dan rute 6-14-0. Jarak terpendek ketiga merupakan gerbang SBM dengan panjang 320m dengan rute 5-15-0. Jarak terpendek keempat, kelima, keenam tidak beda jauh

antar sesama, yaitu Gerbang Depan, Gerbang Sipil, dan Gerbang SR dengan panjang 332m, 353m, dan 356m.

V. KESIMPULAN

Algoritma Dijkstra efektif dalam mencari rute terpendek dalam graf berbobot, tetapi faktor lain seperti ketinggian perlu ditangani dalam pembuatan graf berbobot. Hasil yang didapatkan, yaitu Rute terpendek menuju Labtek V adalah menggunakan Gerbang Tamansari dengan jarak 228m, diikuti oleh rute terpendek kedua yaitu menggunakan Gerbang Batan dengan jarak 260m.

VI. UCAPAN TERIMA KASIH

Saya mengucapkan syukur kepada Allah swt, yang telah memberikan saya hidup, ilmu, dan panduan dalam menulis makalah ini. Saya mengucapkan terima kasih yang sebesar-besarnya kepada kedua orang tua yang telah membesarkan saya dengan kasih sayang dan memberi nafkah untuk kuliah. Saya berterima kasih pula kepada:

- 1) Bu Fariska Zakhralativa Ruskanda, sebagai pengajar mata kuliah Matematika Diskrit K2 yang telah mengajar dengan sangat jelas
- 2) Pak Rinaldi Munir, sebagai pembuat website lengkap terkait berbagai mata kuliah dalam Informatika yang sangat membantu dalam pembelajaran mandiri
- 3) David Benson, sebagai pembuat draw.io, aplikasi gambar diagram yang dipakai pada pembuatan graf berbobot dalam makalah ini
- 4) Geeksforgeeks.com, yang menyediakan kode Algoritma Dijkstra yang dipakai dalam makalah ini

Terakhir, saya memberikan apresiasi dan kehormatan kepada Edsger W. Dijkstra sebagai pembuat Algoritma Dijkstra, yang sangat berperan dalam perkembangan algoritma pencarian rute hingga masa kini.

REFERENSI

- [1] Riley, Sean. "Dijkstra's Algorithm - Computerphile". <https://www.youtube.com/watch?v=GazC3A4OQTE>. Diakses 9 Desember.
- [2] "How to find Shortest Paths from Source to all Vertices using Dijkstra's Algorithm". <https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/>. Diakses 10 Desember.
- [3] Munir, Rinaldi. "Graf (Bagian 1) (Versi update 2023)". <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/19-Graf-Bagian1-2023.pdf>. Diakses 10 Desember.
- [4] Richards, Hamilton. "Edsger Wybe Dijkstra". A.M. Turing Award. Association for Computing Machinery. https://amturing.acm.org/award_winners/dijkstra_1053701.cfm. Diakses 11 Desember.
- [5] Frana, Phil. "An Interview with Edsger W. Dijkstra". Communications of the ACM. 53 (8): 41–47. doi:10.1145/1787234.1787249. Agustus 2010.
- [6] Dijkstra, E. W. "A note on two problems in connexion with graphs". Numerische Mathematik. 1: 269–271. doi:10.1007/BF01386390. 1959.

PERNYATAAN

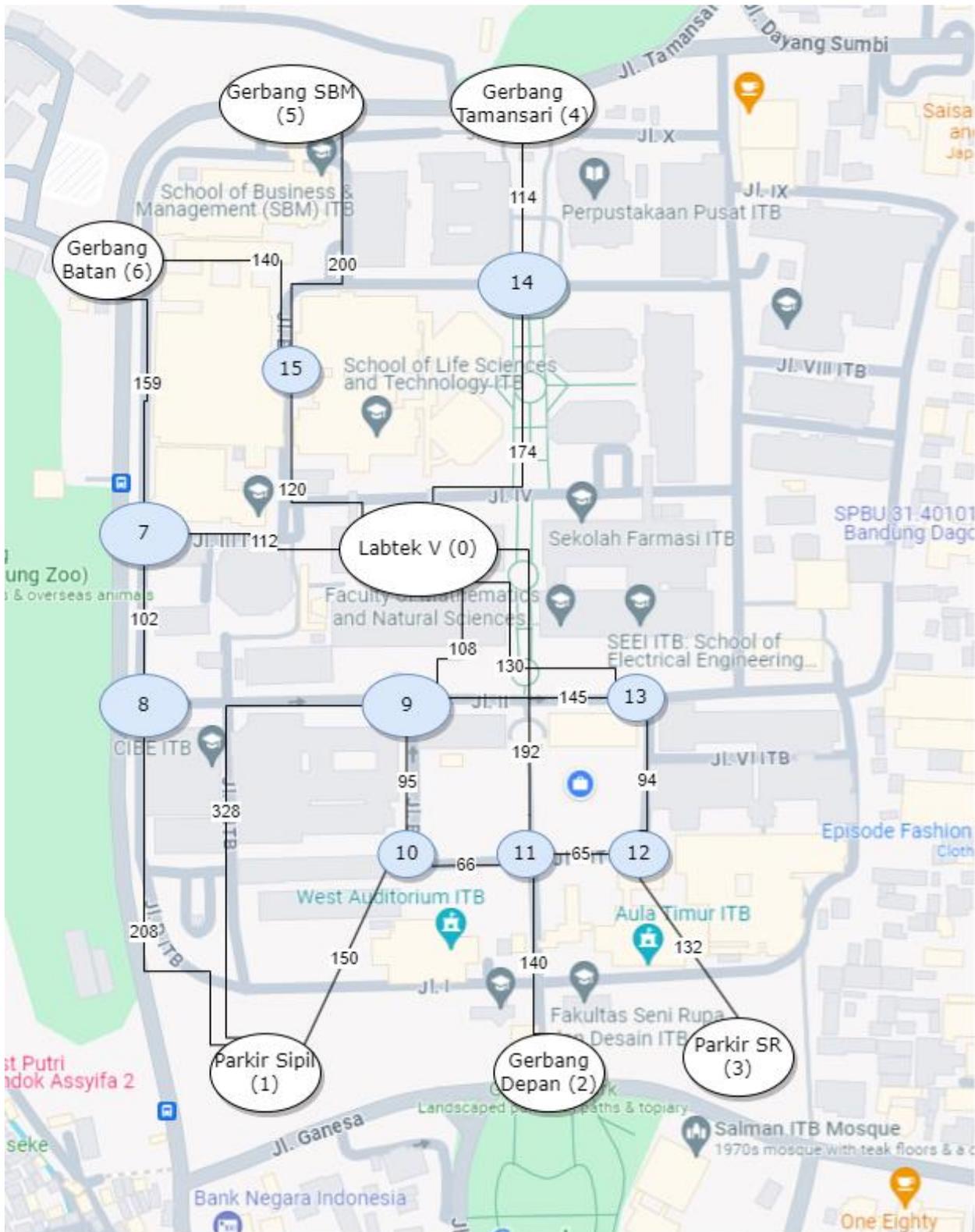
Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Desember 2023

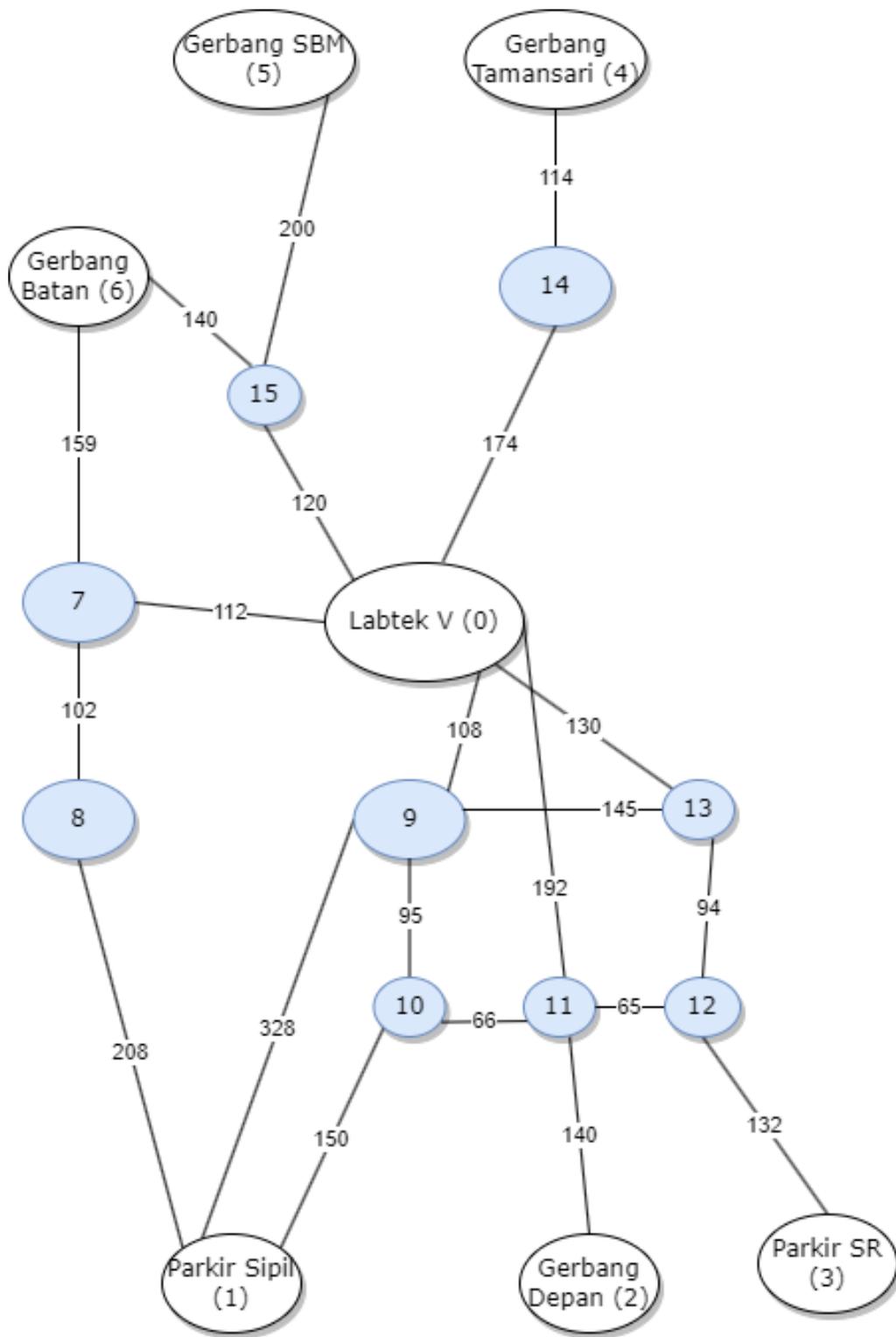


Rafiki Prawhira Harianto
13522065

VII. LAMPIRAN



Gambar 4: Graf berbobot Rute Labtek V ITB (jarak dalam meter)



Gambar 5: Graf berbobot Rute Labtek V ITB tanpa peta (jarak dalam meter)