Penerapan Algoritma Dijkstra dalam Penentuan Rute Transportasi Umum Terintegrasi di Jakarta

Fayza Nadia 13520001[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13520001@std.stei.itb.ac.id

***Abstrak*—Seiring berkembangnya zaman, transpotasi sudah menjadi bagian penting di kehidupan kita sehari-hari. Transportasi memungkinkan penggunanya untuk melakukan mobilisasi antartempat. Pada kota-kota besar, moda transportasi yang dapat dipilihpun semakin beragam. Kota Jakarta merupakan salah satu kota dengan banyaknya moda transportasi yang dapat dipilih, mulai dari LRT, MRT, KRL, Transjakarta, dan masih banyak lagi. Banyaknya moda transportasi diikuti pula dengan semakin banyaknya tujuan yang dapat dituju. Tidak jarang pengguna menemukan lebih dari satu rute yang dapat ditempuh untuk mencapai tujuan yang sama. Tentunya, pengguna ingin menempuh rute terpendek untuk menghemat waktu sebanyak mungkin. Penentuan rute terpendek dapat dilakukan dengan berbagai cara, salah satunya adalah dengan menggunakan algoritma Dijkstra.**

***Kata Kunci*—Algoritma Dijkstra, Graf, Rute, Transportasi.**

I. Pendahuluan

Dalam kehidupan sehari-hari, kita, sebagai makhluk sosial, tentunya seringkali perlu melakukan mobilisasi antartempat. Seiring berkembangnya zaman, mobilisasi dapat dilakukan dengan mudah dan tidak terbatas untuk kalangan tertentu. Perkembangan transportasi yang semakin pesat memungkinkan kita sebagi pengguna untuk memilih satu dari banyaknya moda transportasi yang tersedia sesuai kebutuhan masing-masing.

Jakarta, ibu kota negara kita, adalah salah satu kota besar di Indonesia. Dikenal sebagai pusat metropolitan, Jakarta menjadi titik berkumpulnya segala jenis tempat, mulai dari perkantoran, pusat perbelanjaan, sekolah, tempat rekreasi, tempat tinggal, dan masih banyak lagi. Jarak antarsatu tempat ke tempat lainnya yang cukup berjauhan seringkali mengharuskan kita untuk menggunakan transportasi, baik pribadi ataupun umum, untuk berpergian.

Sebagian besar penduduk Kota Jakarta memilih transportasi umum sebagai moda transportasinya sehari-hari. Alasan yang mendasari hal tersebut di antaranya adalah murahnya biaya yang dikeluarkan per harinya, keberangkatannya yang terjadwal, serta waktu tempuh yang jauh lebih cepat dibandingkan dengan menggunakan transportasi pribadi dikarenakan rawannya kemacetan di sebagian besar ruas jalan di Kota Jakarta. Salah satu keunggulan yang dimiliki Kota Jakarta adalah banyaknya moda transportasi umum yang dapat dipilih, seperti MRT, LRT, Transjakarta, dan KRL *Commuter Line*. Moda transportasi tersebut memiliki kekurangan serta kelebihannya masing-masing. Keempatnya memiliki tujuan-tujuan akhir yang beragam, begitu pula dengan rute yang dilewatinya.

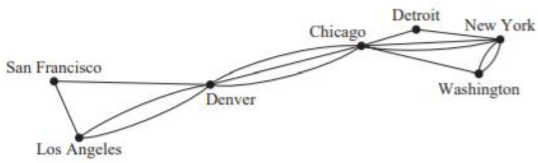
Banyaknya rute yang dapat dilewati untuk menuju ke stasiun pemberhentian yang sama membuat kita, sebagai pengguna, haruslah memiliki perencanaan yang baik agar dapat sampai dengan waktu sesingkat mungkin. Dalam menentukan rute terpendek yang dapat ditempuh, kita dapat memodelkan rute transportasi umum terintegrasi di Jakarta sebagai graf berbobot dengan panjang lintasan sebagai bobot antar simpul—dalam hal ini adalah stasiun pemberhentian. Pemodelan ini kemudian dapat diproses dengan algoritma Dijkstra untuk mendapatkan rute terpendek yang dapat ditempuh dari stasiun awal menuju stasiun tujuan.

II. Landasan Teori

A. Graf

1) Definisi Graf

Graf adalah sebuah struktur diskrit yang terdiri dari simpul (*vertex*) dan sisi (*edge*) yang menjadi penghubung antarsimpul. Graf dapat digunakan untuk memodelkan banyak permasalahan, beberapa di antaranya adalah pemodelan tautan antarsitus di internet, jaringan komputer, rangkaian listrik, serta peta atau rute perjalanan.



Gambar 1.
Pemodelan Jaringan Komputer antarkota sebagai Graf [1]

Secara definisi, graf G dituliskan sebagai:

$$G = (V, E)$$

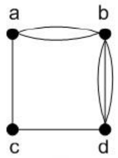
di mana V merepresentasikan himpunan simpul-simpul yang tidak kosong dan E merepresentasikan himpunan dari sisi yang memiliki dua simpul terhubung.

2) Jenis Graf

Berdasarkan arah sisi, graf terbagi menjadi dua jenis:

a) Graf tak berarah (undirected graph)

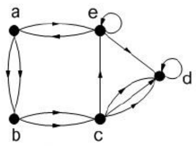
Graf tak-berarah adalah graf yang sisinya tidak memiliki orientasi arah yang digambarkan dengan mata panah.



Gambar 2.
Graf Tak-Berarah [2]

b) Graf berarah (directed graph)

Graf berarah adalah graf yang sisinya memiliki orientasi arah menuju salah satu simpul yang digambarkan dengan mata panah.



Gambar 3.
Graf Berarah [2]

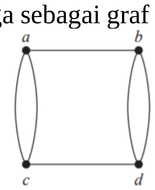
Sedangkan, berdasarkan ada tidaknya sisi ganda dan gelang, graf terbagi menjadi dua jenis:

a) Graf sederhana (simple graph)

Graf sederhana adalah di mana setiap sisi terhubung dengan dua simpul berbeda dan setiap simpul terhubung oleh sisi yang berbeda pula.

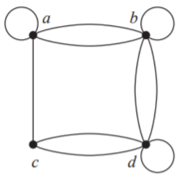
b) Graf tak sederhana (unsimple graph)

Graf tak sederhana adalah graf yang memiliki sisi ganda ataupun sisi gelang. Graf yang memiliki sisi ganda disebut juga sebagai graf ganda (*multigraph)*.



Gambar 4.
Graf Ganda [1]

Sedangkan, graf yang memiliki sisi ganda dan sisi gelang disebut sebagai graf semu (*pseudograph)*.



Gambar 5.
Graf Semu [1]

Berikut adalah tabel kesimpulan dari jenis-jenis graf serta kemungkinan sisinya.

TABEL I. TABEL JENIS GRAF

Tipe	Sisi	Ganda?	Gelang?
Graf sederhana	Tak berarah	Tidak	Tidak
Graf tak sederhana	Tak berarah	Iya	Tidak
Graf semu	Tak berarah	Iya	Iya
Graf berarah sederhana	Berarah	Tidak	Tidak
Graf berarah tak sederhana	Berarah	Iya	Iya
Graf campuran	Berarah dan tak berarah	Iya	Iya

3) Terminologi Graf

a) Ketetanggaan (Adjacent)

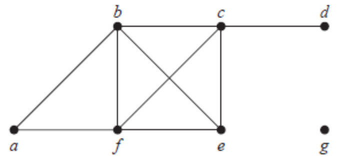
Ketetanggaan adalah istilah di mana dua simpul u dan v pada sebuah graf tak berarah G merupakan ujung dari sebuah sisi e.

b) Bersisian (Incidency)

Bersisian adalah istilah di mana suatu sisi sembarang $e = (v_j, v_k)$ memiliki ujung akhir simpul v_j dan v_k.

c) Simpul Terpencil (Isolated Vertex)

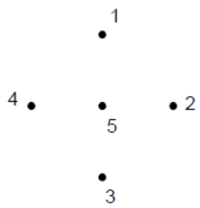
Simpul terpencil adalah sebuah simpul yang memiliki derajat nol. Oleh karena itu, simpul terpencil tidak bertetangga dengan simpul lainnya. Pada gambar 6, dapat dilihat bahwa simpul g merupakan contoh simpul terpencil.



Gambar 6.
Graf dengan Simpul Terpencil [1]

d) Graf Kosong (Null *atau* Empty Graph)

Graf kosong adalah graf yang sisinya memiliki himpuan kosong seperti pada gambar 7.



Gambar 7.
Graf Kosong [2]

e) Derajat (Degree)

Derajat adalah suatu istilah yang menyatakan jumlah sisi yang bersisian dengan sebuah simpul. Derajat dapat dinyatakan dengan notasi $d(v)$ dengan v merupakan sebuah simpul. Pada contoh gambar 6, dapat dilihat bahwa $d(b)$ bernilai 4 dan $d(g)$ bernilai 0 atau disebut juga simpul terpencil.

f) Lintasan (Path)

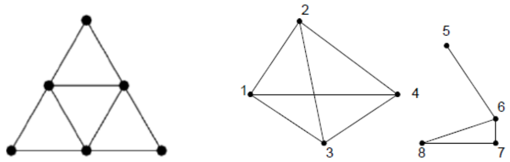
Lintasan dengan panjang n yang membentang dari simpul awal v_0 hingga simpul tujuan v_n adalah barisan simpul dan sisi yang berselang-seling dan membentuk v_0, e_1, v_1, e_2, v_2, …, v_{n-1}, e_n, v_n sehingga $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, …, $e_n = (v_{n-1}, v_n)$ merupakan sisi dari graf. Seperti digambarkan pada gambar 6, dapat dilihat bahwa lintasan *a, b, f, c* adalah lintasan dengan barisan sisi (*a*, *b*), (*b*, *f*), (*f*, *c*) dengan panjang lintasan adalah tiga.

g) Siklus atau Sirkuit (Cycle *atau* Circuit)

Sirkuit adalah lintasan yang memiliki simpul awal sama dengan simpul akhir. Pada gambar 6, sirkuit ditunjukkan pada *a, b, f, a* dengan panjang sirkuit adalah tiga.

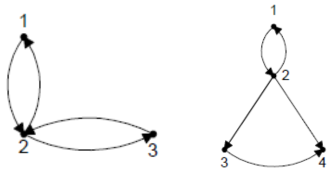
h) Keterhubungan (Connected)

Keterhubungan adalah suatu istilah jika terdapat lintasan yang menghubungkan kedua simpul. Pada graf, dikenal sebuah istilah graf terhubung yang berarti setiap pasang simpulnya terdapat lintasan antara keduanya.



Gambar 8.
Graf Terhubung (Kiri) dan Graf Tak Terhubung (Kanan) [2]

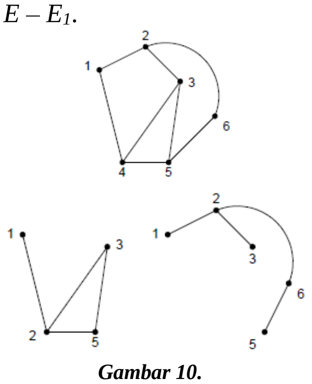
Dalam menyatakan keterhubungan suatu graf berarah, keterhubungan dibagi menjadi terhubung secara kuat (*strongly connected*) dan terhubung lemah (*weakly connected*). Sebuah graf berarah dikatakan terhubung kuat jika, untuk setiap pasang simpul, terdapat lintasan dari simpul *u* menuju *v* dan sebaliknya. Sedangkan, graf dikatakan terhubung lemah jika simpul *u* dan *v* tidak terhubung oleh lintasan dengan arah sebaliknya.



Gambar 9.
Graf Berarah Terhubung Kuat (Kiri) dan
Graf Berarah Terhubung Lemah (Kanan) [2]

i) Upagraf dan Komplemen Upagraf (Subgraph)

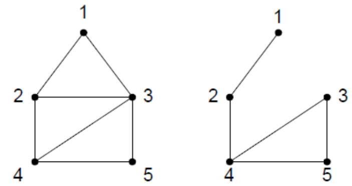
Upagraf adalah graf $G_1 = (V_1, E_1)$ di mana simpul V_1 dan sisi E_1 merupakan himpunan bagian dari $G = (V, E)$. Sedangkan, komplemen dari upagraf G_1 digambarkan dengan sebuah graf $G_2 = (V_2, E_2)$ di mana $E_2 = E - E_1$.



Gambar 10.
Graf G (Atas), Upagraf G_1 (Kiri),
dan Komplemen Upagraf G_2 (Kanan) [2]

j) Upagraf Merentang (Spanning Subgraph)

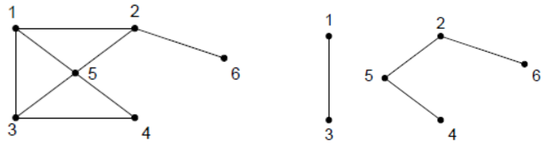
Upagraf merentang adalah upagraf di mana semua simpul yang terdapat pada graf *G* dapat ditemukan pada upagraf tersebut.



Gambar 11.
Graf (Kiri) dan Upagraf Merentang (Kanan) [2]

k) Cut-Set

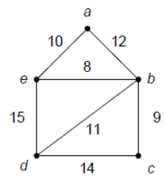
Cut-set adalah himpunan sisi yang dapat menyebabkan graf menjadi tak terhubung jika salah satu sisinya dihilangkan. Dalam satu graf terhubung, terdapat banyak *cut-set* yang membagi graf menjadi bentuk-bentuk berbeda.



Gambar 12.
Graf (Kiri) dan Cut-Set *(Kanan)* [2]

l) Graf Berbobot (Weighted Graph)

Graf berbobot merupakan graf yang memiliki angka sebagai penanda nilai atau besaran dari suatu sisi.

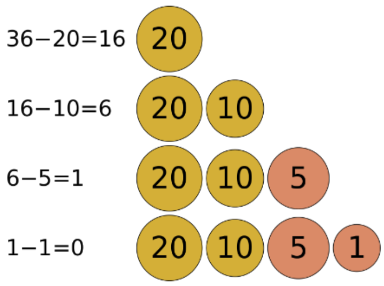


Gambar 13.
Graf Berbobot [2]

B. *Algoritma Dijkstra*

Algoritma Dijkstra adalah algoritma yang digunakan untuk mencari rute terpendek di antara dua simpul pada sebuah graf. Algoritma ini diaplikasikan pada graf berbobot untuk mencari lintasan dengan total bobot terkecil, umumnya pada graf sebagai pemodelan dari rute atau jalur.

Algoritma Dijkstra merupakan salah satu contoh dari algoritma *greedy* yang memiliki prinsip mengambil solusi yang terlihat terbaik di antara pilihan lainnya pada saat itu. Walaupun algoritma *greedy* tidak selalu berhasil, saat algoritma *greedy* terbukti benar, algoritma ini cenderung memiliki tingkat efisiensi yang jauh lebih tinggi dibanding dengan menggunakan algoritma lainnya.



Gambar 14.
Ilustrasi Algoritma Greedy

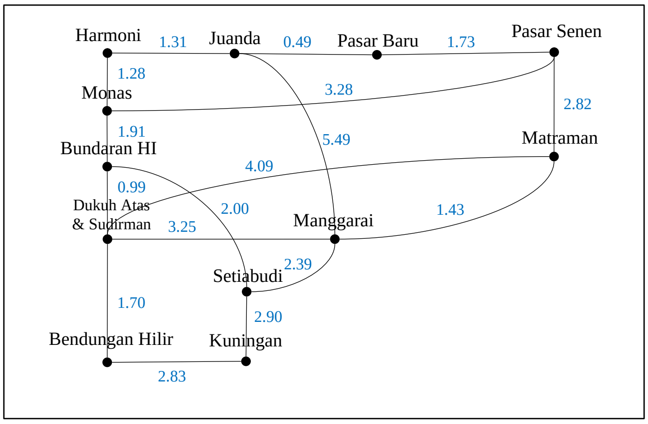
Cara menghitung lintasan terpendek yang dapat ditempuh dengan menggunakan Algoritma Dijkstra adalah dengan menentukan simpul awal dan simpul akhir pada sebuah graf berbobot yang ingin dicari rute terpendeknya. Misalkan, simpul *u* adalah titik awal dan simpul *v* adalah titik akhir yang ingin dituju. Perhitungan dilakukan dengan meninjau simpul dengan sisi bobot terkecil dari *u* yang belum pernah dikunjungi sebelumnya dan menandai bobotnya jika lebih kecil dari nilai semula. Hal ini dilakukan terus-menerus hingga sampai ke simpul akhir yaitu simpul *v*. Setelah sampai di simpul tujuan akhir, lintasan terpendek pun berhasil didapatkan.

III. PEMBAHASAN

Pertama-tama, peta rute transportasi umum terintegrasi di Jakarta perlu dimodelkan dalam bentuk graf berbobot agar dapat dihitung dengan menggunakan algoritma Dijkstra. Tinjau gambar 15, dapat dilihat bahwa peta rute transportasi umum terintegrasi di Jakarta dimodelkan sebagaimana stasiun digambarkan sebagai simpul atau *vertices* dan lintasan antarstasiun digambarkan sebagai sisi atau *edges*.

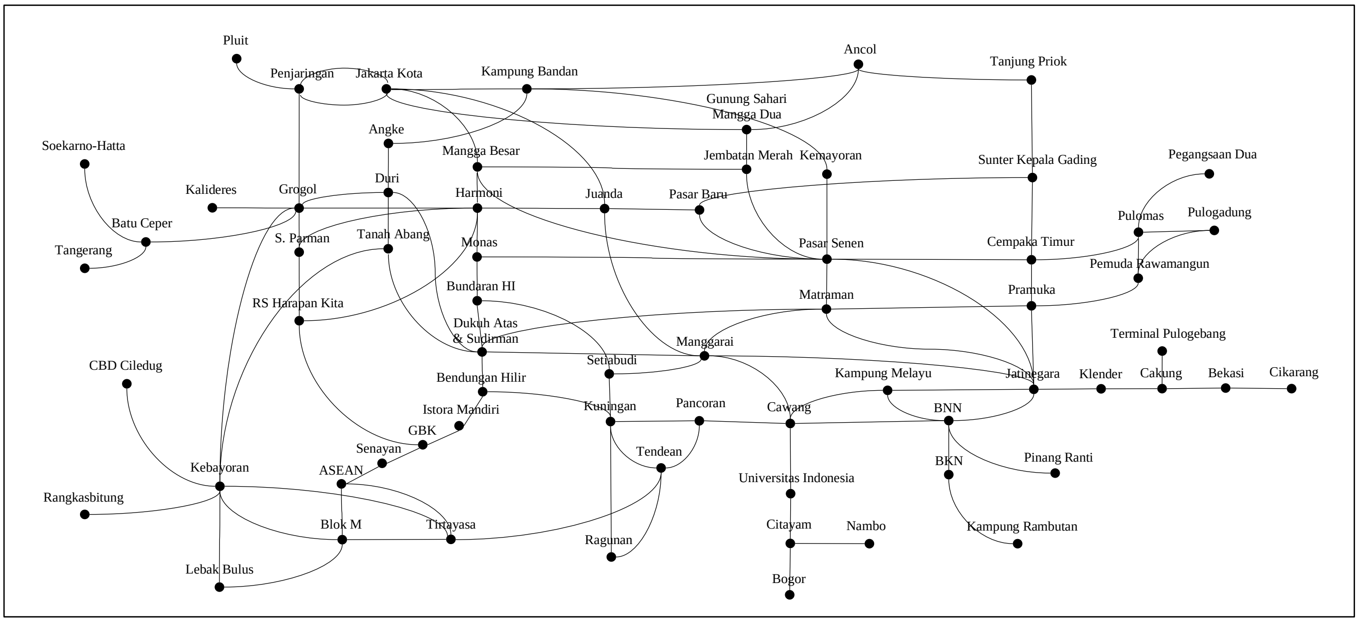
Untuk kemudahan dalam perhitungan yang akan dilakukan di tahap selanjutnya, graf yang dibuat disimplifikasi terlebih dahulu. Beberapa simplifikasi yang dilakukan di antaranya adalah penghapusan beberapa titik stasiun pada peta yang bukan merupakan stasiun pertemuan antara dua jalur atau lebih. Selain itu, graf juga disimplifikasi dengan adanya penggabungan dari stasiun moda transportasi berbeda (KRL, MRT, Transjakarta, dan LRT) pada daerah yang sama.

Perhitungan algoritma Dijkstra dapat dilakukan dari seluruh simpul awal dan simpul akhir yang terdapat pada graf. Namun, pada percobaan kali ini, graf akan dibatasi untuk kemudahan perhitungan. Tinjau gambar 16, sebagai sebuah upagraf dari graf 15 yang menggambarkan rute pada sebagian kecil daerah di Kota Jakarta, tepatnya dalam lingkup daerah simpul asal dan tujuan yang dituju. Dapat dilihat pula bahwa gambar 16 menunjukkan *weighted graph* atau graf berbobot dengan angka pada sisi menunjukkan aproksimasi jarak antar dua stasiun dalam kilometer.



Gambar 16.
Upagraf dari Graf Rute Transportasi Terintegrasi di Jakarta

Jika seseorang ingin pergi menuju Stasiun Pasar Baru dari Stasiun Bendungan Hilir, maka orang tersebut dapat melintasi banyak rute untuk mencapai tujuan tersebut. Algoritma Dijkstra akan memeriksa satu per satu simpul atau stasiun hingga didapatkan rute dengan total jarak terpendek.



Gambar 15.
Pemodelan Graf pada Rute Transportasi Terintegrasi di Jakarta

Langkah-langkah perhitungan dengan algoritma Dijkstra dijabarkan sebagai berikut:

TABEL II. LANGKAH KE-1

Stasiun	Jarak	Cek
Bendungan Hilir	0	-
Bundaran HI	∞	-
Dukuh Atas & Sudirman	∞	-
Harmoni	∞	-
Juanda	∞	-
Kuningan	∞	-
Manggarai	∞	-
Matraman	∞	-
Monas	∞	-
Pasar Baru	∞	-
Pasar Senen	∞	-
Setiabudi	∞	-

Pada langkah ini, seluruh simpul stasiun, kecuali simpul stasiun awal, ditetapkan nilai awal tak hingga karena belum dikunjungi.

TABEL III. LANGKAH KE-2

Stasiun	Jarak	Cek
Bendungan Hilir	0	✓
Bundaran HI	∞	-
Dukuh Atas & Sudirman	1.70	-
Harmoni	∞	-
Juanda	∞	-
Kuningan	2.83	-
Manggarai	∞	-
Matraman	∞	-
Monas	∞	-
Pasar Baru	∞	-
Pasar Senen	∞	-
Setiabudi	∞	-

Pada langkah ini, dilakukan pemeriksaan pada simpul Bendungan Hilir. Pada graf, dapat dilihat bahwa simpul Bendungan Hilir bertetangga dengan simpul Kuningan dan simpul Dukuh Atas & Sudirman sehingga bobot antara simpul tersebut dicatat.

TABEL IV. LANGKAH KE-3

Stasiun	Jarak	Cek
Bendungan Hilir	0	✓
Bundaran HI	2.69	-
Dukuh Atas & Sudirman	1.70	✓
Harmoni	∞	-
Juanda	∞	-
Kuningan	2.83	-
Manggarai	4.95	-
Matraman	5.79	-
Monas	∞	-
Pasar Baru	∞	-
Pasar Senen	∞	-
Setiabudi	∞	-

Pada langkah ini, jarak terdekat yang dapat ditempuh adalah simpul Dukuh Atas & Sudirman dengan jarak 1.70 km. Oleh karena itu, tinjau simpul tersebut dan simpul yang bertetangga dengan simpul Dukuh Atas & Sudirman untuk dicatat jaraknya.

TABEL V. LANGKAH KE-4

Stasiun	Jarak	Cek
Bendungan Hilir	0	✓
Bundaran HI	2.69	✓
Dukuh Atas & Sudirman	1.70	✓
Harmoni	∞	-
Juanda	∞	-
Kuningan	2.83	-
Manggarai	4.95	-
Matraman	5.79	-
Monas	4.60	-
Pasar Baru	∞	-
Pasar Senen	∞	-
Setiabudi	4.69	-

Pada langkah ini, simpul Bundaran HI dipilih karena memiliki jarak terkecil di antara simpul-simpul lain yang belum diperiksa.

TABEL VI. LANGKAH KE-5

Stasiun	Jarak	Cek
Bendungan Hilir	0	✓
Bundaran HI	2.69	✓
Dukuh Atas & Sudirman	1.70	✓
Harmoni	∞	-
Juanda	∞	-
Kuningan	2.83	✓
Manggarai	4.95	-
Matraman	5.79	-
Monas	4.60	-
Pasar Baru	∞	-
Pasar Senen	∞	-
Setiabudi	4.69	-

Pada langkah ini, simpul Kuningan dipilih karena memiliki jarak terkecil di antara simpul-simpul lain yang belum diperiksa.

TABEL VII. LANGKAH KE-6

Stasiun	Jarak	Cek
Bendungan Hilir	0	✓
Bundaran HI	2.69	✓
Dukuh Atas & Sudirman	1.70	✓
Harmoni	5.88	-
Juanda	∞	-
Kuningan	2.83	✓
Manggarai	4.95	-

Matraman	5.79	-
Monas	4.60	✓
Pasar Baru	∞	-
Pasar Senen	7.88	-
Setiabudi	4.69	-

Pada langkah ini, simpul Monas dipilih karena memiliki jarak terkecil di antara simpul-simpul lain yang belum diperiksa.

TABEL VIII. LANGKAH KE-7

Stasiun	Jarak	Cek
Bendungan Hilir	0	✓
Bundaran HI	2.69	✓
Dukuh Atas & Sudirman	1.70	✓
Harmoni	5.88	-
Juanda	∞	-
Kuningan	2.83	✓
Manggarai	4.95	-
Matraman	5.79	-
Monas	4.60	✓
Pasar Baru	∞	-
Pasar Senen	7.88	-
Setiabudi	4.69	✓

Pada langkah ini, simpul Setiabudi dipilih karena memiliki jarak terkecil di antara simpul-simpul lain yang belum diperiksa.

TABEL IX. LANGKAH KE-8

Stasiun	Jarak	Cek
Bendungan Hilir	0	✓
Bundaran HI	2.69	✓
Dukuh Atas & Sudirman	1.70	✓
Harmoni	5.88	-
Juanda	10.44	-
Kuningan	2.83	✓
Manggarai	4.95	✓
Matraman	5.79	-
Monas	4.60	✓
Pasar Baru	∞	-
Pasar Senen	7.88	-
Setiabudi	4.69	✓

Pada langkah ini, simpul Manggarai dipilih karena memiliki jarak terkecil di antara simpul-simpul lain yang belum diperiksa.

TABEL X. LANGKAH KE-9

Stasiun	Jarak	Cek
Bendungan Hilir	0	✓
Bundaran HI	2.69	✓
Dukuh Atas & Sudirman	1.70	✓
Harmoni	5.88	-
Juanda	10.44	-
Kuningan	2.83	✓
Manggarai	4.95	✓
Matraman	5.79	✓
Monas	4.60	✓
Pasar Baru	∞	-
Pasar Senen	7.88	-
Setiabudi	4.69	✓

Pada langkah ini, simpul Matraman dipilih karena memiliki jarak terkecil di antara simpul-simpul lain yang belum diperiksa.

TABEL XI. LANGKAH KE-10

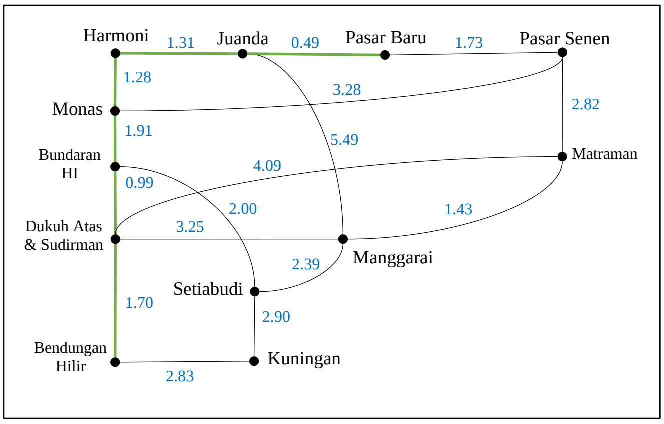
Stasiun	Jarak	Cek
Bendungan Hilir	0	✓
Bundaran HI	2.69	✓
Dukuh Atas & Sudirman	1.70	✓
Harmoni	5.88	✓
Juanda	7.19	-
Kuningan	2.83	✓
Manggarai	4.95	✓
Matraman	5.79	✓
Monas	4.60	✓
Pasar Baru	∞	-
Pasar Senen	7.88	-
Setiabudi	4.69	✓

Pada langkah ini, simpul Harmoni dipilih karena memiliki jarak terkecil di antara simpul-simpul lain yang belum diperiksa. Dapat dilihat pula jarak terkecil simpul Juanda diperbaharui dari 10.44 km menjadi 7.19 km jika melewati simpul Harmoni.

TABEL XII. LANGKAH KE-11

Stasiun	Jarak	Cek
Bendungan Hilir	0	✓
Bundaran HI	2.69	✓
Dukuh Atas & Sudirman	1.70	✓
Harmoni	5.88	✓
Juanda	7.19	✓
Kuningan	2.83	✓
Manggarai	4.95	✓
Matraman	5.79	✓
Monas	4.60	✓
Pasar Baru	7.68	✓
Pasar Senen	7.88	✓
Setiabudi	4.69	✓

Pada langkah ini, simpul Juanda dipilih karena memiliki jarak yang lebih kecil dibandingkan dengan simpul Pasar Senen. Setelah simpul Juanda diperiksa, otomatis semua simpul sudah diperiksa karena hanya tersisa dua simpul (Pasar Baru dan Pasar Senen) sehingga algoritma Dijkstra dinyatakan selesai.



Gambar 17.
Rute Terpendek dari Stasiun Bendungan Hilir ke Pasar Baru

Oleh karena itu, jika seseorang ingin pergi menuju Stasiun Pasar Baru dari Stasiun Bendungan Hilir, rute terpendek yang dapat dilalui adalah melalui Bendungan Hilir – Dukuh Atas & Sudirman – Bundaran HI – Monas – Harmoni – Juanda – Pasar Baru. Total jarak tempuh jika melalui rute tersebut adalah 7.68 km, yang merupakan jarak tempuh terpendek yang dapat dilalui. Tinjau gambar 17, rute tersebut ditandai dengan lintasan yang diberi warna hijau.

Perhitungan yang sama tentunya juga dapat dilakukan pada graf yang berbeda sesuai dengan kebutuhan masing-masing. Pemotongan graf menjadi upagraf yang lebih sederhana dapat dilakukan pula untuk meningkatkan efisiensi algoritma Dijkstra.

IV. KESIMPULAN

Graf merupakan salah satu struktur dalam matematika diskrit yang memiliki banyak kegunaan, di antaranya adalah pemodelan sebuah peta rute transportasi. Pemodelan yang dapat dilakukan adalah dengan menggambarkan titik stasiun sebagai simpul dan jarak antarstasiun sebagai sisi. Tentunya terdapat banyak cara untuk mencari lintasan terpendek antarstasiun, salah satunya adalah dengan menggunakan algoritma Dijkstra. Pada kasus ini, didapatkan hasil perhitungan lintasan terpendek yang dapat ditempuh dari Stasiun Bendungan Hilir menuju Stasiun Pasar Baru adalah sepanjang 7.68 km.

V. UCAPAN TERIMA KASIH

Puji dan syukur penulis panjatkan kepada Tuhan yang Maha Esa atas rahmatnya sehingga makalah ini, yang berjudul "Penerapan Algoritma Dijkstra dalam Penentuan Rute Transportasi Umum Terintegrasi di Jakarta", dapat diselesaikan dengan baik dan tepat pada waktunya. Tentunya, penulis tidaklah sendiri dalam proses penyusunan makalah ini. Penulis mengucapkan terima kasih untuk seluruh dosen mata kuliah IF2120 Matematika Diskrit 2021/2022 yang telah membimbing dan membagikan ilmunya kepada seluruh mahasiswa Teknik Informatika ITB angkatan 2020. Kemudian, terima kasih juga penulis ucapkan untuk teman-teman yang selalu menemani dan memberikan dukungan di setiap saat. Terakhir, tidak lupa penulis mengucapkan terima kasih sebesar-besarnya untuk kedua orang tua yang selalu bersama penulis.

DAFTAR PUSTAKA

[1] Rosen, K., 2012. Discrete mathematics & its applications. 7th ed. New Delhi: McGraw-Hill.

[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/matdis21-22.htm, diakses pada 2 Desember 2021.

[3] https://www.programiz.com/dsa/dijkstra-algorithm, diakses pada 8 Desember 2021.

[4] https://www.geeksforgeeks.org/greedy-algorithms/, diakses pada 8 Desember 2021.

[5] https://transjakarta.co.id/peta-rute/, diakses pada 2 Desember 2021.

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.

Tangerang, 14 Desember 2021

Fayza Nadia
13520001