

Penerapan Algoritma Hybrid Dalam Menentukan Rute Terpendek Antara Cabang Kampus

Waeisul Bismi¹, Windu Gata², Anton³, Taufik Asra⁴

^{1, 2, 3, 4} Ilmu Komputer, Universitas Nusa Mandiri, Jakarta, Indonesia

¹14002464@nusamandiri.ac.id, ²Windu@nusamandiri.ac.id, ³Anton@nusamandiri.ac.id,

⁴Taufik.tas@nusamandiri.ac.id

Diterima 03 Desember 2020

Disetujui 15 Juni 2020

Abstract—Traffic congestion in the Capital City is a familiar sight for residents of the Capital City, because two-wheeled vehicle users dominate the streets of the Capital City as much as 72.8 percent while four-wheeled vehicle users are 26.5 percent. With traffic congestion due to the increasingly uncontrollable number of vehicles in the capital, finding the closest route is the right solution for a worker, student or anyone to reach their destination on time using the Hybrid method. The Hybrid method of the Dijkstra Algorithm and the A* (A-star) Algorithm will be carried out on the same object, namely determining the shortest route between the campus branches, namely the Nusa Mandiri Kramat University campus as the starting point and the Nusa Mandiri Jatiwaringin University as the destination point, this Hybrid method aims to improve the performance and accuracy of the two shortest route search algorithms. Based on the step of applying the Hybrid method in finding the shortest route, it was found with a total distance of 12.4 kilometers as a better and more effective opportunity compared to the results of applying the Dijkstra algorithm with a total distance of 14 kilometers and the A * (A-star) algorithm only with a total distance of 13.1 Kilometers.

Index Terms—Astar's Algorithm, Dijkstra's Algorithm, Hybrid Method.

I. PENDAHULUAN

Kemacetan jalanan ibu kota adalah pemandangan yang sudah lumrah bagi warga ibu kota, yang disebabkan karena pengguna kendaraan roda dua mendominasi jalanan ibu kota sebanyak 72,8 persen sementara itu pengguna kendaraan roda empat 26,5 persen dan kepadatan penduduk di Jakarta serta banyaknya aktifitas yang beragam, Menurut (Suyuti, 2012) ketidak seimbangannya antara kebutuhan dan permintaan yakni meningkatnya alat transportasi dengan kapabilitas prasana yang ada, mengakibatkan sering terjadinya kemacetan pada lalu lintas. Selain itu, Menurut (Salean et al., 2019) tingginya muatan yang terbentuk secara terus menerus menyebabkan lalu lintas (*through traffic*) dan berbaurnya perlintas lokal maupun dari daerah menjadikan masalah pokok

kemacetan di ibu kota.

Dikarenakan tingginya kepadatan lalu lintas serta semakin tidak terkendalinya dan banyaknya jumlah kendaraan di jalanan ibukota, membuat kota Jakarta seolah-olah tidak tidur dari lalu lintas dan menjadikannya pemandangan sehari-hari, agar mencapai tujuan dengan cepat dan tepat waktu pencarian rute yang terdekat merupakan solusi yang sempurna dan terbaik bagi setiap pekerja di ibukota, mahasiswa atau siapapun pengguna jalan di Jakarta, menurut (Kiki Setiawan, Supriyadin, Imam Santoso, 2018). Pada saat seseorang ingin menuju ke suatu tempat tentu biasanya menggunakan pencarian dalam sebuah *maps* yang dalam hal tersebut sangat berakaitan erat dengan pencarian rute seperti ke tempat SPBU, tempat pariwisata, perjalanan rumah ke kantor ataupun perjalanan antara cabang kampus, seseorang pengguna jalan tentu akan memilih rute yang dianggapnya sempurna dengan estimasi waktu lebih cepat dan rute yang *relative* singkat dalam menganalisa rute menuju tujuan.

Maka dari itu solusi yang efektif bagi pencarian rute terpendek antar cabang kampus yang disebabkan oleh kemacetan di ibukota khususnya bagi para mahasiswa yang ada kegiatan di cabang kampus berbeda adalah dengan menggunakan metode *hybrid*.

Metode *Hybrid* adalah penggabungan dari Algoritma Dijkstra dan Algoritma A* (*A-star*) akan diterapkan pada satu objek yang sama yaitu menentukan rute terpendek antara cabang kampus yakni kampus Universitas Nusa Mandiri Kramat sebagai titik awal dan Universitas Nusa Mandiri Jatiwaringin sebagai titik tujuan.

Pada penerapan metode *hybrid* ini digunakan untuk meningkatkan akurasi serta *performance* dari kedua algoritma pencarian rute terpendek dan menurut (Noviriandini & Safitri, 2017) rute terpendek pada suatu lintasan dapat dimaknai menjadi nilai terkecil dari sebuah lintasan, yakni dengan total nilai dari

segala lengkungan yang membentuk lintasan. beberapa algoritma matematika dapat memecahkan permasalahannya dan Menurut (Cantona et al., 2020) dalam menyelesaikan masalah rute terpendek pada *graf* berarah, ada banyak metode algoritma yang dapat digunakan.

Algoritma dalam menetapkan rute terdekat ataupun pencarian lintasan terpendek terdapat beraneka ragam namanya dan juga metode penyelesaiannya, diantara lain ialah Algoritma A* (*A-star*) yang menggunakan metode *heuristic* dalam melakukan perhitungan pencarian lintasan terpendek yang juga menganalisa nilai input serta mengevaluasi beberapa jalur yang mungkin dilalui sehingga membentuk hasil. Hart, Nilsson, dan Raphael merupakan orang-orang yang mengemukakan Algoritma A* (*A-star*) sehingga dipergunakan secara luas pada *graph traversal* yakni penemuan dan memproses dalam menentukan jalur yang mampu dilewati secara efisien dan efektif pada sekitar titik yang diklaim sebagai *node*. Walaupun tidak selalu benar kelebihan yang dimiliki dari Algoritma A* (*A-star*) yakni dalam waktu menemukan rute perlintasan dianggap lebih baik dan cepat dibandingkan dengan algoritma - algoritma pencarian rute lainnya kemudian jumlah *Loop A star* lebih sedikit dan rute yang ditemukan dapat berbeda tapi mempunyai biaya yang sama.

Lalu terdapat Algoritma Dijkstra yang menggunakan metode *blind method* dalam pencarian rute terpendek, Algoritma Dijkstra dianggap dapat dijadikan solusi sebagai prosedur pemecahan masalah dalam memastikan jalur terpendek dengan memperhitungkan kalkulasi terhadap nilai satu titik dengan titik lainnya dalam sebuah jalur lintasan beserta semua kemungkinannya sehingga dapat menghasilkan rute terpendek dengan menghitung lintasan berdasarkan rute yang ditempuh pada tiap wilayah (Noviriandini & Safitri, 2017). Edsger Dijkstra merupakan nama seseorang penemu yang namanya dinamai sesuai dengan hasil temuannya yakni Algoritma Dijkstra. Keunggulan mekanisme penyelesaian pada Algoritma Dijkstra terletak pada prosesnya yakni awal proses pemilihan titik mana yg akan menjadi bobot jarak di *node* yang pada tahap pengembangan selanjutnya pencarian origin berasal satu titik ke titik lainnya serta ke poin selanjutnya secara bertahap sampai menemukan akhir *node* yang dituju sebagai titik yang dimaksudkan dan Adapun kekurangan dalam Algoritma Dijkstra yaitu hanya berdasarkan pada informasi yang dependen sehingga didapati nilai optimum hanya pada saat itu. Jadi dalam Algoritma Dijkstra, kita tidak perlu memikirkan konsekuensi yang akan terjadi jika kita memilih suatu keputusan pada tahapan - tahapan tertentu.

Dengan melihat kelebihan dan kekurangan dari Algoritma Dijkstra dan Algoritma A* (*A-star*), penulis membandingkan keduanya untuk menunjukkan hasil dari kedua algoritma berdasarkan deterministik yang kemudian dilakukan metode *Hybrid* dengan kedua algoritma tersebut dalam menentukan rute terpendek antar cabang kampus Universitas Nusa Mandiri Kramat dan Universitas Nusa Mandiri Jatiwaringin yang menghasilkan bentuk Algoritma *Best First Search* agar efisiensi dalam hal jarak serta waktu tempuh.

II. LANDASAN TEORI

A. Teori Graf

Pada tahun 1736 merupakan tahun sejarah teori graph bermula pada saat seseorang berkebangsaan Swiss berhasil dalam memecahkan masalah teka-teki jembatan Konigberg, dia adalah seorang matematikawan bernama Leonhard Euler. (Andriansyah et al., 2010). Menurut (Fitria & Triansyah, 2013) mengungkapkan bahwa sebuah *graf* bisa dimaknai menjadi pasangan himpunan (V, E) , lalu berikutnya ditulis menggunakan notasi $G = (V, E)$, pada perihal ini V merupakan kumpulan simpul yang memiliki isi (*Vertices or Node*) dan E merupakan bagian sisi tepi yang ditetapkan (*edge or arc*) yang menghubungkan beberapa pasang sebuah simpul. Simpul pada sebuah *graf* dapat diberi lambang dengan huruf abjad semacam A, I, U, ... dsb. ataupun penomoran dengan angka, seperti nomor 2, 4, 6, ... dsb. atau bisa juga dengan mengkombinasi dari keduanya antar angka dan huruf sesuai kebutuhan.

Apabila simbol (e) merupakan bagian sisi yang menghubungkan titik u dan titik v maka simbol (e) dapat juga diinisalkan rumus $e = (u, v)$ karena bagian sisi yang menghubungkan kedua titik tersebut diberi inisial rumus (u, v) dan bisa juga dilambangkan dengan e^1, e^2, e^3, \dots en. bila dipahami secara geometris sebuah *graf* dideskripsikan sebagai himpunan titik (simpul) pada aspek 2 bidang perspektif yang dihubungkan dari sekumpulan garis (sisi).

Permasalahan dalam menemukan jalur terpendek di *graf* artinya tergolong dalam salah satu *problem optimize*. *Graf* yg dipergunakan dalam pencarian jalur terpendek adalah grafik berbobot (*weighted graph*), yaitu *graf* yg masing-masing mengilustrasikan seperti jeda jarak antar wilayah, proses ketika pengiriman pesan, porto pembangunan serta lain sebagainya. seluruh perkiraan yang digunakan ialah bobot bernilai positif.

Secara global, istilah terpendek memiliki arti yakni meminimalkan nilai bobot pada sebuah jalur *graf* akan tetapi hal tersebut tergantung dari typical permasalahan yang dibahas dan dipecahkan (Noviriandini & Safitri, 2017).

Algoritma Dijkstra bisa diterapkan pada suatu persoalan pencarian lintasan terpendek dengan cara

membuat jalur ke satu simpul optimal pada setiap Langkah (Hariyadi et al., 2020). Jadi di langkah ke (n) setidaknya terdapat (n) *node* yang telah kita ketahui jalur terpendek. Langkah-langkah menerapkan algoritma Dijkstra pada suatu persoalan dapat dilakukan dengan langkah-langkah berikut :

- a) Pada langkah pertama pilih titik mana yang akan menjadi simpul awal, kemudian berikan bobot nilai jarak pada simpul pertama ke simpul terdekat satu per satu, Dijkstra akan mengembangkan pencarian dari satu titik ke titik lainnya dan ke titik berikutnya selangkah demi selangkah.
- b) Beri bobot nilai (jarak) untuk setiap titik ke titik lain, kemudian tetapkan nilai 0 pada simpul awal dan nilai tak hingga untuk simpul lainnya (tidak terisi)
- c) Atur seluruh titik *node* yang belum dilalui dan kemudian aturlah titik *node* awal menjadi "Embarcation node"
- d) Berasal dari *node embarkasi*, melakukan peninjauan padatitik *node* tetangga yang belum dilalui kemudian hitung jarak asal dari titik embarkasi. apabila jarak yang didapati lebih kecil dari jarak sebelumnya (yang telah terekam pada sebelumnya) maka hapus data lama, kemudian laukan update pada data jarak dengan jarak yang baru
- e) Pada saat terselesaikannya perhitungan pada setiap jarak terhadap titik *node* tetangga, kemudian beri tanda pada titik *node* yang dilalui menjadi "*node* yang dilintasi". Dan titik *node* yang telah dilewati tidak akan pernah di kroscek kembali, jarak yang telah dihitung serta yang telah disimpan merupakan jarak terakhir serta yang memiliki bobot paling sedikit.
- f) Setel "simpul tak terlewati" menggunakan jarak terkecil (dari simpul embarkasi) ke "simpul embarkasi" berikutnya dan ulangi langkah e.

Menentukan titik waktu intervensi yang terkait dengan simpul yang telah ditentukan sebelumnya, yaitu bobot terkecil yang terlihat dari tabel dan garis besar sebagai simpul yang ditentukan selanjutnya.

B. Matriks Ketetanggaan

Matriks ketetanggaan atau istilah lain matriks tetangga merupakan adalah *graf* simetris yang sederhanayaitu $a_{ij} = a_{ji}$. Perihal ini bermula oleh kedua-duanya yakni bernilai 1 ketika v_i dan v_j memiliki sisi, dan bernilai 0 apabila tidak ada sisi diantara kedua nya, maka hal ini dinamakan juga matriks bernilai nol-satu, sedangkan buat *graf* berarah, matriks tetangga belum tentu simetris (akan bergejala Bila itu merupakan grafik arah lengkap). Selain itu, *graf* sederhana tidak memiliki cincin, sebagai akibatnya diagonal utama selalu 0 karena a_{ii} , $i = 1, 2, 3, \dots, n$ memiliki nilai tetap 0.

Dengan merepresentasikan matriks salah satunya dengan cara nilai matrik ketetanggaan maka dapat mempermudah dalam perhitungan pada *graf*. Matriks yang berdekatan pada titik asal suatu grafik mengacu pada urutan simpul, Jadi sebesar nilai n urutan berbeda dibuat dari titik asal simpul keuntungan representasi menggunakan matriks ialah elemen matriksnya bisa dilakukan akses secara eksklusif melalui nilai pada indeks, Misalkan terdapat sebuah *graf* $G = (V, E)$ dengan menggunakan jumlah simpul n . Matriks ketetanggaan G adalah matriks bujur sangkar dengan menghitung ukuran $n \times n$. atau $M = [m_{ij}]$, dengan $m_{ij} = 1$ jika simpul i dan j bertetangga, kebaliknnya $m_{ij} = 0$ jika simpul i dan j tidak bertetangga.

C. Metode Heuristik

Heuristik dalam bahasa inggris menjadi *Heuristic* serta istilah tersebut berasal dari kata kerja bahasa Yunani yakni *heuriskein*, yang memiliki arti yaitu "mencari atau menemukan". Pada era dunia pemograman saat ini, dimana istilah *heuristic* diartikan menjadi suatu proses yang dapat merampungkan pada suatu persoalan namun tidak terdapat agunan padanya bahwa solusi yang dicarinya tersebut bisa dapat dipastikan.

Dalam mengkaji dan menelaah metode ini, istilah *heuristic* dapat juga diartikan menjadi suatu fungsi yang dapat menyampaikan sebuah nilai asumsi dan nilai estimasi yakni dari suatu perkiraan solusi. Teknik dalam pencarian nilai dalam metode *heuristic* adalah salah satu bagian dalam melakukan suatu proses pencarian secara hati-hati serta ketat dan juga dapat mengarahkan kepada suatu proses pencarian yang mendapati kemungkinan peluang lebih besar dan memiliki prbabilitas keberhasilan terbesar, namun menggunakan kemungkinan pada kelengkapan (complete). Fungsi heuristik ialah aturan yang digunakan buat menerima solusi yg diinginkan sebagai implementasi dari pencarian metode heuristik

D. Algoritma Dijkstra

Menurut (Masri et al., 2019) Algoritma Dijkstra adalah algoritma yang digunakan untuk menemukan jalur (*path*) teringkas dari simpul awal ke simpul akhir (*the minimum path of the weighted graph*). Untuk membandingkan nilai bobot terkecil dari titik sumber menuju titik tujuan agar mendapatkan hasil yang efisien metode algoritma Dijkstra lah yang umumnya digunakan utnk memecahkan permasalahan tersebut (Masri et al., 2019), karena pada dasarnya Algoritma Dijkstra termasuk golongan teknik *greedy* yakni menggunakan sekumpulan (*vertex*) nilai titik S yang pertamanya diisikan dengan nilai awal pada setiap langkah (*vertex*) nilai titik v lalu berikutnya ditambahkan yang mempunyai bobot terkecil dan dalam merekam jalur terpendek dari titik awal hingga titik akhir biasanya menerapkan penggunaan *Array D*.

Lalu Menurut (Yunata & Widayati, 2020) Algoritma *Greedy* biasa dipergunakan dalam memecahkan permasalahan *optimize* yakni dengan penyelesaian secara selangkah demi selangkah (*step by step*). Dan juga menurut (Harsadi & Nugroho, 2020) Algoritma tersebut merupakan algoritma yang umum digunakan dalam menyelesaikan permasalahan terkait pencarian *graf* dari suatu lintasan terpendek bersamaan dengan sebuah *graf* yang berbeda pohon lintasan terpendek..

Untuk itu Algoritma Dijkstra menerapkan prinsip dan teknik *greedy* dalam pemilihan sisi yg memiliki bobot terkecil pada setiap langkah demi langkahnya dan lalu dimasukkan kedalam himpunan solusinya, hal tersebut mengemukakan bahwa algoritma Dijkstra mendapati jalur terpendek pada *graf* berarah dengan panjang sisi *non-negatif*., sehingga sangat cocok dengan karakteristik perkotaan yang mempunyai jalan satu arah (Nggufon et al., 2019).

E. Algoritma A* (A-star)

Pada tahun 1968 Algoritma A* (*A-star*) pertama kali diperkenalkan dengan menggunakan metode *heuristic* yang merupakan bagian dari format pencarian *heuristic* dalam menghitung efektifitas dan efisiensi sebuah solusi yang optimal oleh peter hart, Nils Nilson, dan Bertram Raphael

A* diperkenalkan oleh Peter Hart, Nils Nilsson dan Bertram Raphael pertama kali pada tahun 1968 dengan menggunakan *heuristic*. Algoritma A* (*A-star*) merupakan format pencarian heuristik untuk menghitung efisiensi solusi optimal.

Algoritma A* (*A-star*) adalah bagian dari penyempurnaan algoritma Dijkstra serta menghasilkan sebuah algoritma pencarian pertama yang terbaik atau BFS (Best First Search). Algoritma A* (*A-star*) biasanya akan bekerja dengan menjumlahkan nilai $g(n)$, yakni menghitung banyaknya jumlah pergerakan awal dari satu *node* ke *node* lain dan lalu $h(n)$ yaitu estimasi dari sebuah biaya *node* n ke tempat titik pada tujuan akhir. Sehingga di dapatkan rumus persamaan seperti berikut ini :

$$f(n)=g(n)+h(n) \quad (1)$$

$f(n)$ = jumlah total biaya (jarak) yang dibutuhkan dalam perjalanan dari satu *node* ke tujuan.

$h(n)$ = estimasi biaya (jarak) dari titik *node* n ke titik tujuan akhir (heuristik).

$g(n)$ = total jumlah pada pergerakan dari satu *node* menuju *node* lain. (graph)

Dengan menggunakan istilah lain maka biaya atau *cost* merupakan jarak lintas yang sudah dilalui serta Panjang garis lurus yang dihasilkan antara *node* n (titik awal) dengan *node* akhir (tujuan) merupakan asumsi dari nilai *heuristic* nya dan juga semakin kecil nilai pada nilai $f(n)$, maka semakin tinggi dan meningkat nilai bentuk prioritasnya.

F. Google Maps

Google Maps merupakan peta yang berbasis online dan dapat diakses melalui perangkat *mobile*, *dekstop*, ataupun *web* yang dapat dilihat melalui situs <https://maps.google.com>. karena aplikasi ini disediakan oleh *Google* secara gratis. *Google maps* memudahkan kita dalam mencari informasi seputar geografis karena pada layanan ini kita dapat melihat seluruh wilayah dibumi hanya dengan menggeser, *zoom in* dan *zoom out* sesuai kebutuhan pengguna (Ismayanto, 2018). Selama tersedianya akses internet berjalan lancar dengan menggunakan aplikasi *Google maps* ini memudahkan kita dalam mencari rute termudah, terpendek dan tercepat agar pengguna tidak tersesat dan terjebak dalam kemacetan.

Selain *Google Maps*, aplikasi *Waze* juga dapat digunakan untuk peta jalan berbasis online. Aplikasi *Waze* hampir sama dengan *Google Maps*, yang membedakan adalah pencarian rutanya saja yang mengarahkan pengguna ke rute yang lebih jauh serta merepotkan. Dan fitur yang ada di dalam aplikasi *Waze* tidak sebanyak dengan fitur yang ada di dalam aplikasi *Google Maps*. Menurut (Setya et al., 2020) mengungkapkan bahwa *Google Maps* memiliki tingkat presentase sebesar 54% sedangkan penggunaan *Waze* di Indonesia menempati peringkat ke – 34 dengan indeks rating sebesar 3.54.

Pendekatan kualitatif merupakan pendekatan yg diterapkan dalam penelitian ini, karena teknik yang digunakan merupakan metode observasi lapangan dan juga berfokus pada eksplorasi agar mendapati pemahaman tentang alasan yang mendasar dan juga guna menghasilkan uraian - uraian penelitian yang pragmatis serta sesuai fakta kebenaran hasil lapangan secara detail dan terperinci. Menurut (Noviriandini & Safitri, 2017) menggunakan pendekatan kualitatif pada suatu penelitian sangat sesuai dalam menyamakan antara metode deskriptif dengan teori yang berlaku.

III. METODE PENELITIAN

Adapun metode penelitian yang dipakai sebagai proses pengumpulan data dalam penelitian mencari rute terpendek antar cabang kampus menggunakan algoritma djisktra sebagai berikut :

A. Pengamatan Lapangan (Observation)

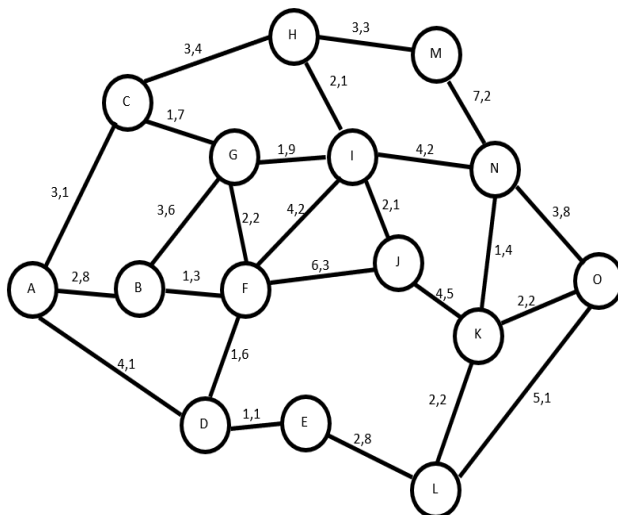
Dalam proses pengumpulan data peneliti melakukan eksplorasi lapangan pada *object* guna mendapatkan kebenaran hasil lapangan dengan melakukan perjalanan secara akurat dan langsung antar cabang dari kampus Universitas Nusa Mandiri Kramat sebagai titik sumber dan Universitas Nusa Mandiri Jatiwaringin sebagai titik tujuan untuk menghitung jarak antar cabang kampus melalui rute yang berbeda-beda dan untuk memastikan rute yang ditempuh kemudian dilakukan pengecekan kembali melalui *google maps* untuk total rutanya.

B. Literatur Studi (Library Research)

Untuk memperkuat landasan teori dalam penelitian ini peneliti melakukan pengumpulan data rute-rute yang digunakan menuju kampus dan juga artikel dari paper, buku, internet atau referensi lainnya agar memperoleh informasi yang terarah sesuai fokus penelitian.

IV. HASIL DAN PEMBAHASAN

Peneliti meneliti mengenai jarak rute tempuh antar cabang kampus dari Jalan Kramat Raya yakni kampus Universitas Nusa Mandiri Kramat menuju ke Jalan Raya Jatiwaringin yakni Universitas Nusa Mandiri Jatiwaringin dengan menerapkan Metode *Hybrid*, pada dasarnya banyak rute yang dapat dilalui oleh peneliti pada saat menuju ketempat tujuan tersebut dan untuk pembuatan titik *graf* pada saat observasi object dilakukan, setiap perlintas yang dilalui oleh peneliti dibuatlah tanda yang kemudian dijadikan beberapa titik *graf*. Hasil pemberian beberapa tanda pada tiap perlintas, menghasilkan gambar titik - titik *graf* yang bisa dilihat pada gambar 1 dibawah. Titik - titik *graf* yang dibuat oleh peneliti berlandaskan titik sumber, titik tujuan dan juga panjang jarak (kilometer) yang ditempuh.



Gambar 1. Titik *graf* rute perjalanan antar cabang kampus

Adapun keterangan tiap titik pada perlintasan rute perjalanan yang telah ditempuh antar cabang kampus dapat terlihat pada Tabel 1.

Tabel 1. Keterangan Titik *Grif* Rute Perjalanan

No	Titik	Keterangan
1	A	Jl. Kramat Raya
2	B	Jl. Salemba Raya
3	C	Jl. Lendjten Suprpto
4	D	Jl. Jatinegara Timur
5	E	Terminal Kp. Melayu
6	F	JL. Matraman
7	G	Jl. Pramuka
8	H	Jl. Perintis Kemerdekan
9	I	Jl. Pemuda Rawamangun
10	J	Jl. Klender
11	K	Jl. Raya Bakasi
12	L	Jl. Jendral Basuki
13	M	Jl. Pantura
14	N	JL. Inpeksi Kalimalang
15	O	Jl. Raya Jatiwaringin

Setelah mengetahui rute-rute perlintasan wilayah menuju wilayah yang lain, dalam hal penelitian ini peneliti akan menerapkan Metode *Hybrid* yakni dengan Algoritma Dijkstra dan Algoritma A* (*A-star*) sebagai metode yang dianggap ampuh memecahkan permasalahan pencarian rute, lalu menjadikan tiap - tiap persimpangan dengan persimpangan lainnya sebagai *input* atau nilai masukan pada sebuah *graf* yang mana persimpangan.

A. Penerapan Algoritma Dijkstra

Algoritma Dijkstra merupakan bentuk dari Algoritma Greedy yang bersifat rakus yang pada dasarnya algoritma ini akan memeriksa simpul ke segala arah yang dimungkinkan hingga menghasilkan beberapa rute perjalanan yang dapat dilalui yang dapat dilihat pada table 2 berikut.

Tabel 2. Daftar Rute Perlintasan

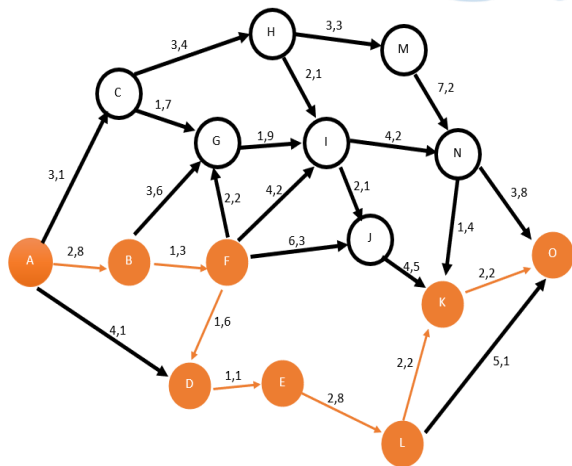
No	Lintasan	Keterangan
1	A-C-H-M-N-O	Rute 1
2	A-C-H-M-N-K-O	Rute 2
3	A-C-H-I-N-O	Rute 3
4	A-C-H-I-N-K-O	Rute 4
5	A-C-H-I-J-K-O	Rute 5
6	A-C-G-I-N-O	Rute 6
7	A-C-G-I-N-K-O	Rute 7
8	A-C-G-I-J-K-O	Rute 8
9	A-B-F-G-I-J-K-O	Rute 9
10	A-B-F-I-J-K-O	Rute 10
11	A-B-F-I-N-K-O	Rute 11
12	A-B-F-I-N-O	Rute 12
13	A-B-F-J-K-O	Rute 13
14	A-B-F-D-E-L-K-O	Rute 14
15	A-B-F-D-E-L-O	Rute 15

Dalam penerapan Algoritma Dijkstra pemilihan rute dijalankan dengan menggunakan cara pencarian pertama terbaik atau dengan istilah lain yakni *Best First Search (BFS)*, Proses ini akan diterapkan secara berulang dengan cara setiap lintasan rute yang diperhitungkan jaraknya dari titik *node* awal perjalanan hingga menuju titik *node* lainnya, kemudian lintasan rute tersebut akan dibandingkan dan rute yang memakai interval terpendek akan terpilih sebagai rute terpendek, yang kemudian digambarkan melalui matriks yang dapat dilihat pada gambar 2 dibawah ini,

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
A	0	2,8	3,1	4,1	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
B	2,8	0	∞	∞	∞	1,3	3,6	∞	∞	∞	∞	∞	∞	∞	∞
C	3,1	∞	0	∞	∞	∞	1,7	3,4	∞	∞	∞	∞	∞	∞	∞
D	4,1	∞	∞	0	1,1	1,6	∞	∞	∞	∞	∞	∞	∞	∞	∞
E	∞	∞	∞	1,1	0	∞	∞	∞	∞	∞	∞	2,8	∞	∞	∞
F	∞	1,3	∞	1,6	∞	0	2,2	∞	4,2	6,3	∞	∞	∞	∞	∞
G	∞	3,6	1,7	∞	∞	2,2	0	∞	1,9	∞	∞	∞	∞	∞	∞
H	∞	∞	3,4	∞	∞	∞	∞	0	2,1	∞	∞	∞	3,3	∞	∞
I	∞	∞	∞	∞	∞	4,2	1,9	2,1	0	2,1	∞	∞	∞	4,2	∞
J	∞	∞	∞	∞	∞	6,3	∞	∞	2,1	0	4,5	∞	∞	∞	∞
K	∞	∞	∞	∞	∞	∞	∞	∞	∞	4,5	0	2,2	∞	1,4	2,2
L	∞	∞	∞	∞	2,8	∞	∞	∞	∞	∞	2,2	0	∞	∞	5,1
M	∞	∞	∞	∞	∞	∞	∞	3,3	∞	∞	∞	∞	0	7,2	∞
N	∞	∞	∞	∞	∞	∞	∞	∞	4,2	∞	1,4	∞	7,2	0	3,8
O	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	2,2	5,1	∞	3,8	0

Gambar 2. Tabel Matriks

Dari gambar 2 diketahui baris yang terdapat pada matriks menentukan lokasi titik awal dan pada bagian kolom menentukan lokasi tujuan. Lalu penerapan Algoritma Dijkstra dalam pencarian rute antar cabang kampus menghasilkan lintasan rute terpendek yakni rute 14 dengan total jarak tempuhnya 14 Km yang bisa dilihat pada gambar 3 dibawah ini.



Gambar 3. Rute Penerapan algoritma Dijkstra

Dan berikut data keterangan jalur – jalur yang dipilih, didapat dari google maps :

1. Jl.Kramat raya
2. Jl.Kramat raya – Jl. Salemba Raya
3. Jl.Salemba raya – Jl. Matraman
4. Jl. Matraman – Jl.Jatinegara timur
5. Jl.Jatinegara timur – Terminal Kp.melayu
6. Terminal Kp.melayu – Jl.Jendral Basuki
7. Jl.Jendral Basuki – Jl. Raya Bekasi
8. Jl. Raya Bekasi – Jl. Raya Jatiwaringin

B. Penerapan Algoritma A*(A-star)

Pada Tahap penerapan algoritma A* (Astar) dalam menentukan rute terpendek dibutuhkan nilai heuristik, untuk menghitung jarak *heuristic* menggunakan metode *Euclidean distance* pada persamaan. Menurut (Umar et al., 2021) *Euclidean distance* adalah perhitungan 2 buah titik mengenai jarak dalam cakupan lingkup *Euclidean space*. Pada tahun 300 SM dipublikasikanlah istilah *Euclidean space* oleh Euclid, yakni seorang matematikawan yang berasal dari Yunanidalam mengkaji suatu hubungan antara sudut simpul dan juga jarak serta *Euclidean* ini berkaitan juga dengan *Teorema Phytagoras* yang pada umumnya dilakukan pada setidaknya satu serta dua atau tiga dimensi dan Berikut formula *Euclidean*.

$$d = \sqrt{(x1 - x2) + (y1 - y2)2} \quad (2)$$

Maka hasil penerapan dari rumus formula diatas akan menjadi seperti berikut :

$$d = \sqrt{(lat1 - lat2) + (long1 - long2)2} \quad (3)$$

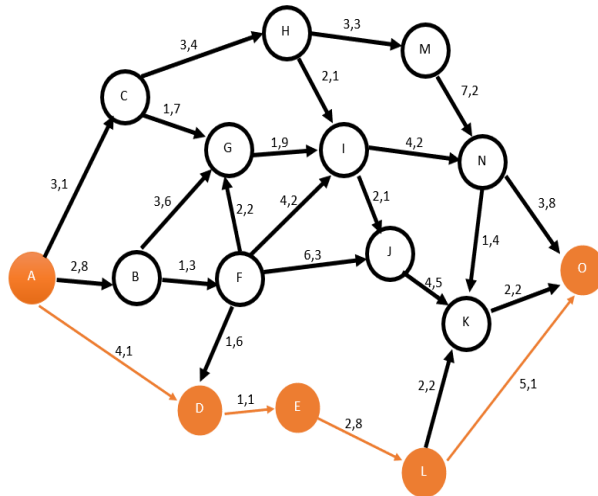
D adalah Jarak dengan satuan kilometer (Km)
 Lat^1 adalah istilah *Latitude* titik *node* awal
 Lat^2 adalah istilah *Latitude* titik tujuan
 $Long^1$ adalah *Longitude* titik *node* awal
 $Long^2$ adalah *Longitude* titik *node* tujuan

Maka dapat dilihat pada Tabel 3 berikut dibawah ini,merupakan hasil perhitungan dari titik-titik *node* dengan menggunakan rumus diatas :

Tabel 3. Nilai Titik Heuristik

No	Titik Indeks	Hasil
1	A (0) ke B (2,8)	2,8
2	A (0) ke C (3,1)	3,1
3	A (0) ke D (4,1)	4,1
4	B (2,8) ke F (1,3)	4,1
5	B (2,8) ke G (3,6)	6,4
6	C (3,1) ke G (3,6)	6,7
7	C (3,1) ke H (3,4)	6,5
8	D (4,1) ke F (1,6)	5,7
9	D (4,1) ke E (1,1)	5,5
11	F (1,3) ke D (1,6)	2,9
12	F (1,3) ke G (2,2)	3,5
13	F (1,3) ke I (4,2)	5,3
14	F (1,3) ke J (6,3)	7,4
15	J (6,3) ke K (4,3)	10,6
16	E (1,1) ke L (2,8)	3,9
17	L (2,8) ke K (2,2)	5
18	L (2,8) ke O (5,1)	7,9
19	K (2,2) ke O (2,2)	4,4
20	E (1,1) ke L (2,8)	3,9

Setelah mendapati nilai *heuristic* pada setiap masing-masing titik *node*, maka pada tahapan selanjutnya yakni dilakukan perhitungan algoritma A* menggunakan rumus $f(n)=g(n)+h(n)$ yang menghasilkan lintasan rute terpendek antara cabang kampus yang dapat dilihat pada gambar 4 berikut.

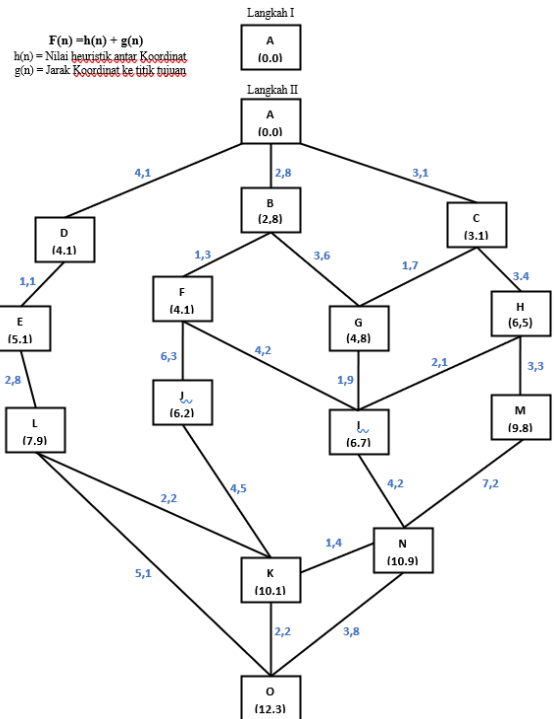


Gambar 4. Rute Penerapan Algoritma A-star

Karena antara satu *graf* mewakili perhitungan jarak Kilometer (KM), maka jalur lintasan yang didapati dari algoritma A* (*A-star*) adalah A-D-E-L-O dengan total jarak 13,1 Km.

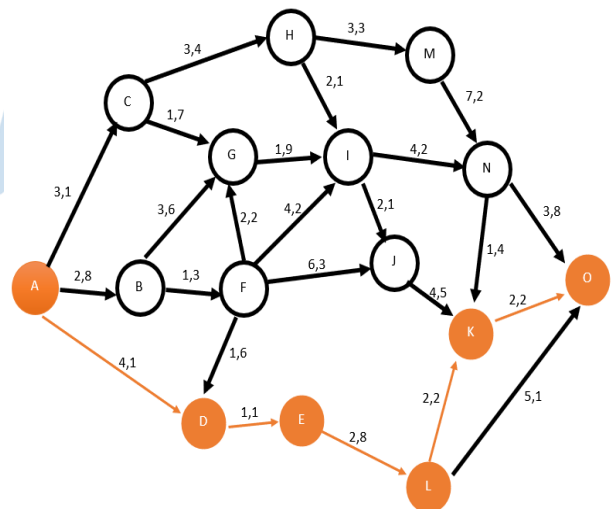
C. Penerapan Metode *Hybrid*

Pada Metode *Hybrid* ini dilakukan penggabungan kedua algoritma yakni menerapkan Algoritma Dijkstra yang bersifat greedy untuk menghasilkan nilai matriks dan menerapkan Algoritma A* (*A-star*) untuk penyempurnaan Algoritma Dijkstra dengan representasi graph pada matriks Algoritma Dijkstra kemudian Langkah selanjutnya menghitung jarak dari dua titik untuk dijadikan nilai heuristik. Maka bisa dijelaskan dengan istilah lain, bahwa heuristik memiliki peranan fungsi optimasi yang menjadikan algoritma A* lebih baik dari pada algoritma lainnya dan pada gambar 5 berikut adalah langkah pencarian rute terpendek dengan menerapkan metode *Hybrid*.



Gambar 5. Langkah Pencarian rute terpendek

Berdasarkan langkah pencarian dengan metode *Hybrid*, peneliti telah berhasil mendapati nilai serta menemukan rute terpendek dalam mencari lokasi antar titik dengan cara menghitungkan semua jarak keseluruhan lintasan yang dilewatinya yang dapat dilihat pada gambar 6 berikut.



Gambar 6. Rute Penerapan Metode *Hybrid*

Karena antara satu *graf* mewakili perhitungan jarak Kilometer (KM), maka jalur lintasan yang didapat dari penerapan metode *Hybrid* adalah A-D-E-L-K-O dengan total jarak 12,4 Km. yang mana bisa dianggap sebagai peluang yang lebih baik dan efektif untuk menghindari kemacetan lalu lintas dari Universitas Nusa Mandiri kampus Kramat sebagai titik awal menuju Universitas Nusa Mandiri kampus Jatiwaringin sebagai titik tujuan.

V. KESIMPULAN

Berdasarkan serangkaian penelitian yang telah peneliti lakukan dalam mencari rute terpendek untuk studi kasus antara kampus, maka dapat disimpulkan beberapa hal:

1. Penerapan Algoritma Dijkstra dengan perhitungan matriks nya didapat rute perlintasan A-B-F-D-E-L-K-O dengan total jarak 14 Km. lalu Penerapan Algoritma A* (*A-star*) dengan nilai titik heuristiknya mendapati rute perlintasan A-D-E-L-O dengan total jarak 13,1 Km. dan dengan penerapan metode *Hybrid* dari algoritma Dijkstra serta algoritma A* star (*A-star*) menghasilkan perlintasan rute A-D-E-L-K-O yang mana bisa dijadikan peluang terbaik dengan total jarak tempuh 12,4 Km.
2. Penelitian ini menerapkan metode *Hybrid* yaitu kombinasi dari Algoritma Dijkstra dan Algoritma A* (*A-star*), dalam menentukan rute terpendek antar cabang kampus yang menghasilkan bentuk Algoritma *best first search* agar efisiensi dalam hal jarak serta waktu tempuh lalu menjadikan tiap - tiap persimpangan satu dengan persimpangan lainnya sebagai *input* atau nilai masukan pada sebuah *graf* yang mana persimpangan ini juga bisa diartikan sebagai titik simpul (*node*) dan menjadikan jalur yang menghubungkan persimpangan sebagai simpul (*path*).
3. Metode *Hybrid* digunakan untuk meningkatkan akurasi serta *performance* dari kedua algoritma dan menjadi solusi yang efektif bagi pencarian rute terpendek bagi mahasiswa yang akan melakukan kegiatan antar Universitas Nusa Mandiri kampus Kramat sebagai titik awal dan Universitas Nusa Mandiri kampus Jatiwaringin sebagai titik tujuan.

UCAPAN TERIMA KASIH

Ucapan terima kasih peneliti sampaikan kepada Program Studi Ilmu Komputer Universitas Nusa Mandiri yang telah mendukung dalam proses penelitian yang telah dilakukan.

DAFTAR PUSTAKA

- [1] Andriansyah, Sofian Efendi, F., Syaofullah, Pinto, M., Pujianto, & Stepven Tempake, H. (2010). Implementasi Algoritma Greedy Untuk Melakukan Graph Coloring: Studi Kasus Peta Propinsi Jawa Timur. *Jurnal Informatika Ahmad Dahlan*, 4(2), 103610. <https://doi.org/10.26555/jifo.v4i2.a5275>
- [2] Cantona, A., Fauziah, F., & Winarsih, W. (2020). Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta. *Jurnal Teknologi Dan Manajemen Informatika*, 6(1), 27–34. <https://doi.org/10.26905/jtmi.v6i1.3837>
- [3] Fitria, & Triansyah, A. (2013). Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan. *Jurnal Sistem Informasi (JIS)*, 5(2), 611–621. <http://ejournal.unsri.ac.id/index.php/jsi/article/download/840/430>
- [4] Hariyadi, D., Nakulo, B., Sari, I. D., & Aini, F. N. (2020). IMPLEMENTASI ALGORITMA DIJKSTRA DALAM PENENTUAN JALUR TERPENDEK STUDI KASUS JARAK TEMPAT KULIAH TERDEKAT. *Indonesian Journal of Business Intelligence*, 3(1), 14–19.
- [5] Harsadi, P., & Nugroho, D. (2020). Implementasi Algoritma Dijkstra Dan Metode Haversine Pada Penentuan Jalur Terpendek Pendakian Gunung Merapi Jalur Selo Berbasis Android. *Jurnal Teknologi Informasi Dan Komunikasi (TIKOMSiN)*, 8(1), 61–67. <https://doi.org/10.30646/tikomsin.v8i1.483>
- [6] Ismayanto, G. (2018). *Sistem Informasi Geografis Pariwisata Kabupaten Sragen Berbasis Android Dengan Memanfaatkan Google Maps Api*. 3(1), 224–234.
- [7] Kiki Setiawan, Supriyadin, Imam Santoso, R. B. (2018). Menghitung Rute Terpendek Menggunakan Algoritma a * Dengan Fungsi Euclidean Distance. *Seminar Nasional Teknologi Informasi Dan Komunikasi, 2018*(Sentika), 70–79.
- [8] Masri, M., Kiswanto, A. P., & Kusuma, B. S. (2019). *Implementasi Algoritma Dijkstra Dalam Perancangan Pariwisata Danau Toba Dan Sekitarnya*. 221–225.
- [9] Nggufuron, N., Rochmad, & Mashuri. (2019). Pencarian Rute Terbaik Pemadam Kebakaran Kota Semarang Menggunakan Algoritma Dijkstra dengan Logika Fuzzy sebagai Penentu Bobot pada Graf. *Unnes Journal of Mathematics*, 8(1), 40–49. <https://doi.org/10.15294/ujm.v8i1.19461>
- [10] Noviriandini, A., & Safitri, M. (2017). *Implementasi Algoritma Dijkstra Untuk Menentukan Jalur Terpendek Wilayah Pisangan Dan Kampus Nusa Mandiri Tangerang*. 13(2), 181–186.
- [11] Salean, S. T., Si, M., & Hadyan, M. H. (2019). *Analisis Kemacetan Lalu-Lintas Di Jalan Matraman Raya-Jalan Bekasi Barat , Jakarta Timur*. 13(1).
- [12] Setya, B. F., Handoko, E., & Puspita, D. (2020). “ *Smart Traffic Controller* ” Aplikasi Pencarian Rute Optimal Berdasarkan Waktu , Biaya , Kondisi Cuaca , Kejadian Lokal , dan Jumlah Kendaraan dengan Menggunakan Algoritma Fuzzy Mamdani dan Modified Markov Model Berbasis Android Mobile dan Google Map API. 7–24.
- [13] Suyuti, R. (2012). Implementasi ” Intelligent Transportation System (Its) ” Untuk Mengatasi. *KONSTRUKSIA*, 3, 13–21.
- [14] Umar, R., Yudhana, A., & Prayudi, A. (2021). *WISATA KABUPATEN DOMPU COMPARATIVE ANALYSIS OF DIJKSTRA , A-STAR , AND FLOYD WARSHALL ALGORITHM IN SEARCHING THE NEAREST ROUTE SEARCH TO TOURISM*. 8(2), 227–234.

<https://doi.org/10.25126/jtiik.202182866>

- [15] Yunata, A., & Widayati, Q. (2020). Penjadwalan Pembuatan Container Portacamp Menggunakan Algoritma

Greedy. *Bina Darma Conference on ...*, 183–195.
<http://conference.binadarma.ac.id/index.php/BDCCS/article/view/1015>

