

Penerapan Route planning dalam Menentukan Jalur Mudik pada Aplikasi Google Maps

Faris Fadhilah 13518026
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13518026@std.stei.itb.ac.id

Abstract—Saat ini, semua orang melakukan kegiatan sehari-hari mereka dengan teknologi, mulai dari aplikasi transportasi *online*, aplikasi belanja *online*, dan masih banyak lagi. Kemajuan teknologi yang pesat ini tidak lepas dari peran raksasa teknologi Google yang menjadi pelopor aplikasi-aplikasi yang dikembangkan untuk memudahkan pekerjaan manusia sehingga manusia menjadi lebih produktif, efisien, dan mangkus dalam berkegiatan sehari-hari. Google Maps merupakan salah satu aplikasi yang sangat membantu manusia dalam menentukan rute terpendek dan tercepat dari satu tempat ke tempat lainnya. Pengguna yang menggunakan aplikasi Google Maps dapat menghemat waktu dan biaya. Saat ini, Google Maps telah banyak digunakan oleh aplikasi lainnya.

Keywords—*Google Maps, Algoritma A star, Algoritma Uniform Cost Search, Algoritma Greedy Best first search*

I. PENDAHULUAN

Saat ini, hampir semua orang tidak bisa lepas dari pesatnya kemajuan teknologi dalam berbagai bidang, mulai dari hadirnya berbagai macam aplikasi-aplikasi *online* hingga otomatisasi berbagai hal yang memudahkan kehidupan manusia pada saat ini. Salah satunya adalah aplikasi yang dikembangkan oleh perusahaan teknologi paling terkenal saat ini, yaitu Google. Google telah menciptakan berbagai macam yang sangat berguna mulai dari Google Email (Gmail), Google Maps, Youtube, Google Drive, Google Play Music, Google Play Movies & TV, Google, Duo, Google Photos, dan masih banyak lagi.

Google Maps adalah layanan pemetaan web yang dikembangkan oleh Google. Layanan Google Maps memberikan citra satelit, peta jalan, hingga panorama dengan kamera 360°, kondisi lalu lintas, dan perencanaan rute untuk bepergian dari satu tempat ke tempat lainnya. Google Maps dapat digunakan untuk merencanakan rute bepergian baik untuk pengguna yang berjalan kaki, mengendarai mobil, mengendarai sepeda, atau transportasi umum. Google Maps merupakan program *desktop* dengan bahasa pemrograman C++ yang pertama kali dirancang oleh Lars dan Jens Eilstrup Rasmussen. Pada tahun 2004 Google Maps diakuisi dari perusahaan milik Lars dan Jens oleh Google. Google Maps resmi diluncurkan ke publik pada tahun 2005 dengan menggunakan JavaScript, XML, dan AJAX. Tampilan Satelit dari Google Maps adalah “*top-down*” dengan citra resolusi yang tinggi karena diambil

dari udara. Google Maps untuk ponsel cerdas dirilis pada September 2008 dan pada Agustus 2013, Google Maps memiliki target untuk menjadi aplikasi yang paling populer di dunia untuk ponsel cerdas, dengan lebih dari 54% pemilik ponsel cerdas telah menggunakan Google Maps setidaknya sekali. Fitur-fitur baru yang telah dikembangkan adalah Maps GL. Google Maps telah disempurnakan dengan memberikan kinerja grafis lebih baik, grafis 3D lebih kaya, transisi yang halus antar citra, rotasi tampilan dengan sudut 45° dan akses yang lebih nyata melalui Google Street View.

Google Maps merupakan aplikasi penunjuk arah atau navigasi yang memberikan pengguna panduan arah untuk dapat mencapai lokasi B dari lokasi A. Google Maps menampilkan panduan arah dengan mencari rute terpendek dengan estimasi waktu tercepat yang dapat ditempuh oleh pengguna. Google Maps dapat menampilkan rute yang mengalami kemacetan dan akan mengalihkan pengguna ke rute lain yang memiliki estimasi sampai ditujuan lebih cepat. Google Maps dapat mendeteksi kemacetan dengan mengukur seberapa banyak orang yang menggunakan Google Maps pada suatu jalan. Jika pada suatu jalan banyak yang menggunakan Google Maps, maka jalan tersebut dikategorikan jalan macet.

Fitur-fitur terbaru dari Google Maps adalah dapat diakses tanpa kuota, dengan syarat perlu mengaktifkan fitur “*offline maps*” saat membuka. Pengguna harus menyiapkan ruang penyimpanan pada ponsel pintar karena fitur “*offline maps*” ini akan mengunduh peta dan pengguna dapat mengatur seberapa jauh peta yang ingin diunduh, Google Maps juga dapat berbagi lokasi pengguna secara *real time*. Google Maps dapat digunakan untuk memesan transportasi *online* karena aplikasi akan memunculkan pilihan harga dan *provider* transportasi *online* yang ingin digunakan. Google Maps juga dapat menunjukkan riwayat lokasi yang dikunjungi oleh pengguna dengan memilih “*your timeline*” dan pilih tanggal yang sesuai untuk menunjukkan riwayat lokasi. Pengguna Google Maps juga dapat memberikan ulasan pada lokasi-lokasi yang terdapat pada aplikasi, ulasan berupa rating, komentar, hingga menambahkan foto atau video. Google Maps memiliki fitur untuk memperbesar ukuran layar navigasi hanya dengan mengetuk layar sebanyak dua kali menggunakan satu jari. Fitur selanjutnya adalah pengguna Google Maps dapat memesan secara langsung kursi di restoran ketika sedang mencari lokasi restoran.

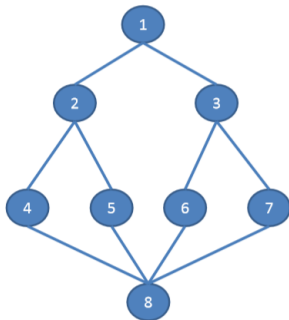
II. LANDASAN TEORI

Penentuan rute terpendek dan tercepat pada aplikasi Google Maps erat kaitannya dengan penerapan *route planning* dengan 3 algoritma, yaitu algoritma *Uniform Cost Search*, algoritma *Greedy Best first search*, dan algoritma *A star*. Untuk persoalan menentukan rute tercepat dengan factor estimasi waktu tidak akan dibahas pada makalah ini, karena masalah yang lebih cukup kompleks dan lebih membutuhkan pemahaman mendalam.

1. Algoritma Uniform Cost Search

Algoritma *Uniform Cost Search* atau biasa disingkat UCS adalah perbandingan antara algoritma *Breadth First Search* dan algoritma *Iterative Deepening Search*.

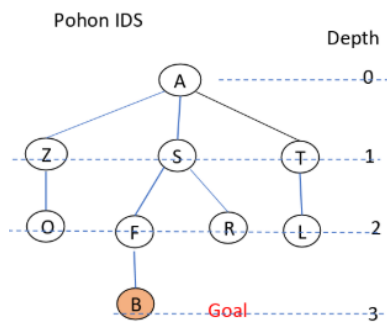
Algoritma *Breadth First Search* adalah pencarian melebar. Urutan algoritmanya adalah mengunjungi dahulu simpul v kemudian kunjungi semua simpul yang bertetangga dengan simpul v terlebih dahulu atau yang masih berada pada satu level, lalu kunjungi simpul yang belum dikunjungi dan bertetangga dengan simpul yang telah dikunjungi sebelumnya. Untuk lebih jelasnya dapat melihat gambar 1. Dari gambar 1 didapat urutan simpul yang dibangkitkan adalah 1,2,3,4,5,6,7,8.



Gambar 1. Ilustrasi BFS

(Sumber: BFS-dan-DFS(2020).pdf)

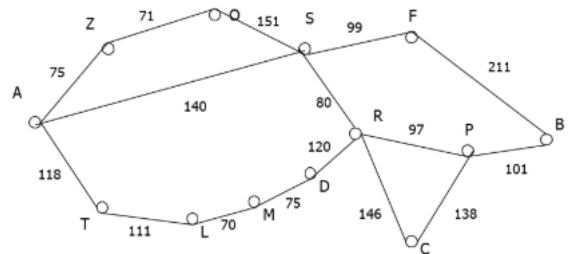
Algoritma *Iterative Deepening Search* (IDS) adalah algoritma lanjutan dari algoritma *Deep Limited Search* (DLS). Algoritma DLS sendiri membantasi pencarian solusi sampai level n , sedangkan algoritma IDS akan dilakukan penambahan level sebanyak 1 apabila pada level ke n solusi belum ditemukan. Untuk lebih jelasnya dapat dilihat pada gambar 2. Dari gambar 2 didapat urutan simpul yang dibangkitkan adalah A,Z,S,T,O,F,R,L,B.



Gambar 2. Ilustrasi IDS

(Sumber: Route-Planning-Bagian1.pptx)

Jadi, algoritma *Uniform Cost Search* adalah algoritma yang mengitung *cost* dari simpul akar menuju simpul ke n . *Cost* yang dicari biasanya adalah jarak dari satu kota ke kota lainnya biasa dilambangkan dengan $g(n)$ dengan n adalah nama simpul. Terdapat banyak pilihan rute untuk menuju satu kota ke kota lain sehingga dihitung *cost* semua kemungkinan ada dan kemudian dipilih rute yang memiliki *cost* paling sedikit, dimana *cost* yang paling sedikit merepresentasikan rute terpendek untuk mencapai tujuan. Untuk lebih jelasnya dapat melihat gambar 3. Dari gambar 3 didapat rute terpendek yang didapat melalui jalur $A \rightarrow S \rightarrow R \rightarrow P \rightarrow B$ dengan nilai *cost* 418.



Gambar 3. Contoh UCS

(Sumber: Route-Planning-Bagian1.pptx)

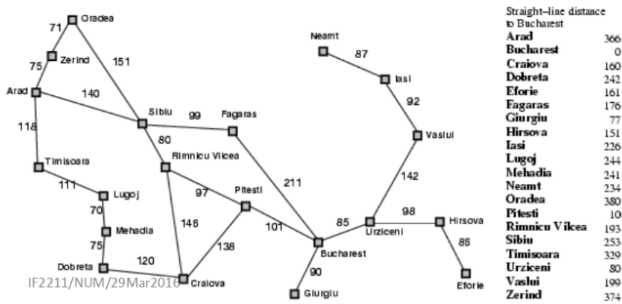
Simpul awal yang dibangkitkan adalah simpul A lalu memiliki anak simpul Z,T,S. Lalu dibangkitkan lagi simpul Z karena memiliki *cost* yang paling kecil diantara T dan S, lalu dibangkitkan simpul T karena memiliki *cost* simpul lebih kecil bila dibandingkan dengan simpul S,O,L. Kemudian dibangkitkan lagi simpul S karena memiliki *cost* yang lebih kecil dibandingkan dengan *cost* simpul O,R,L,F,O. dan seterusnya sampai didapat simpul goal.

2. Algoritma Greedy Best first search

Algoritma *Greedy Best first search* adalah menggunakan sebuah evaluasi atau bisa dibilang taksiran pada suatu simpul. Jika pada UCS nilai *cost* didasarkan pada jarak sebenarnya dari satu kota ke kota lainnya, maka pada algoritma *Greedy Best first search* nilai *cost* merupakan jarak garis lurus dari satu ke kota lainnya yang biasa dilambangkan dengan $h(n)$ dengan n adalah nama simpul. Algoritma *Greedy Best first search* merupakan pemanfaatan dari heuristik. Heuristik adalah merupakan nilai estimasi yang cenderung konsisten. Pada kasus *route planning* ini digunakan nilai heuristik yaitu jarak garis lurus dari satu kota ke kota lainnya. Kelemahan dari algoritma *Greedy Best first search* adalah *not complete*, maksudnya adalah setiap kota satu dengan kota lainnya pasti terdapat jarak lurus diantara keduanya, namun belum tentu kedua kota tersebut berhubungan atau terdapat sisi diantara kedua simpul. Selanjutnya adalah *get stuck with local plateau*, maksudnya adalah pada saat dilakukan pencarian dengan algoritma *greedy best first search* maka pencarian akan berputar membentuk suatu *loop* pada suatu simpul yang memiliki *cost* terkecil sehingga tidak mengarah kepada solusi. Lalu *irrevocable* atau tidak dapat berbalik atau diganti, maksudnya apabila suatu simpul telah dipilih kita tidak dapat melakukan pembatalan untuk memilih simpul tersebut. Pada

gambar 4, dengan menggunakan algoritma *Greedy Best first search* didapat rute terpendek adalah Arad → Sibiu → Fagaras → Bucharest dengan nilai *cost* 450.

Heuristik adalah nilai estimasi atau perkiraan dari suatu simpul. Penentuan nilai heuristik dapat berbeda-beda pada setiap kasusnya. Nilai heuristik dapat berupa jarak *Euclidean*, jarak garis lurus antar 2 simpul, jarak *Hamilton* dan masih banyak lagi. Nilai heuristik sendiri tidak menentu dan berubah.



Gambar 4. Contoh *Greedy Best first search*

(Sumber: Route-Planning-Bagian1.pptx)

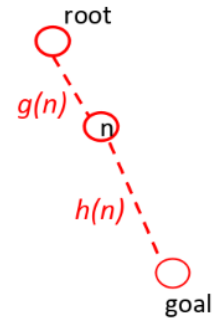
Simpul pertama yang dibangkitkan adalah simpul Arad, lalu dibangkitkan simpul Sibiu, Timisoara, dan Zerind. Lalu dipilih Sibiu karena memiliki nilai *cost* paling kecil. Simpul selanjutnya yang dibangkitkan adalah Arad, Fagaras, Oradea, Rimicu Vicea. Lalu dipilih Fagaras karena memiliki simpul terkecil dibandingkan yang lain. Lalu dibangkitkan lagi simpul Sibiu dan Bucharest, dipilih Bucharest sehingga sampai pada simpul *goal* atau tujuan.

3. Algoritma A star

Algoritma *A star* merupakan gabungan dari algoritma *Uniform Cost Search* dan algoritma *Greedy Best first search*. Fungsi evaluasi dari algoritma *A star* yang biasa dilambangkan dengan $f(n)$ dengan n adalah nama simpul adalah penjumlahan antara *cost* dari simpul awal menuju simpul ke n dengan nilai heuristik atau estimasi *cost* dari simpul n menuju ke *goal*. Dapat disimpulkan bahwa $f(n) = g(n) + h(n)$. Untuk lebih jelasnya dapat dilihat pada gambar 5. Apabila $f(n) = g(n)$ merupakan algoritma *Uniform Cost Search* sedangkan apabila $f(n) = h(n)$ merupakan algoritma *Greedy Best first search*. Menggunakan gambar 4, apabila dilakukan dengan algoritma *A star* maka akan didapat rute terpendek adalah Arad → Sibiu → Rimicu Vicea → Pitesti → Bucharest dengan nilai *cost* adalah 418. Nilai *cost* yang dihasilkan lebih sedikit apabila dibandingkan dengan algoritma *Greedy Best first search*.

Simpul pertama yang dibangkitkan adalah simpul Arad, lalu dibangkitkan simpul Sibiu, Timisoara, dan Zerind. Lalu dipilih Sibiu karena memiliki nilai *cost* paling kecil. Simpul selanjutnya yang dibangkitkan adalah Arad, Fagaras, Oradea, Rimicu Vicea. Lalu dipilih Rimicu Vicea karena memiliki simpul terkecil dibandingkan yang lain. Lalu dibangkitkan lagi simpul Craiova, Pitesti, Sibiu. Lalu pilih Pitesti karena memiliki nilai *cost* terkecil bila dibandingkan dengan yang lain. Lalu dibangkitkan simpul Bucharest, Craiova, dan

Rimicu Vicea. Terakhir pilih Bucharest sehingga sampai pada simpul *goal* atau tujuan.



Gambar 5. Ilustrasi Algoritma *A star*
(Sumber: Route-Planning-Bagian2.pptx)

Algoritma *A star* ini memiliki tujuan untuk mencari *cost* terendah untuk mencapai simpul tujuan. Algoritma *A star* sendiri merupakan algoritma yang *complete* karena semua simpulnya akan dipastikan memiliki sisi untuk menghubungkan antara kedua simpul, lalu Algoritma *A star* memiliki kompleksitas waktu dan kompleksitas ruang yang sama yaitu $O(b^m)$, dan tentu saja algoritma *A star* ini merupakan algoritma yang optimal.

Jika dibandingkan dengan algoritma *branch and bound*, batas pada algoritma *A star* akan dijamin memiliki batas yang lebih kecil dibandingkan dengan nilai *cost* sebenarnya, pencarian pohon juga dapat dilakukan sesuai dengan keinginan pengguna, terdapat pemangkasan atau pemotongan ruang solusi apabila hasil penjumlahan dari *cost* ditambah dengan *bound* lebih besar dari solusi terbaik sehingga jumlah simpul yang diperiksa tidak banyak, dan *bound* biasanya lebih canggih.

Heuristik yang digunakan pada algoritma *A star* harus heuristik yang dapat diterima. Maksud dari nilai heuristik yang dapat diterima adalah nilai heuristik tidak boleh lebih dari besar dari nilai *cost* yang sebenarnya. Karena jika itu terjadi akan mempengaruhi hasil perhitungan akhir dan data menjadi tidak valid. Setiap *cost* yang sebenarnya dapat mencapai status *goal* maka harus terdapat pula nilai heuristik yang diterima.

Algoritma *A star* menjadi algoritma yang paling optimal bila dibandingkan dengan algoritma *Uniform Cost Search* dan algoritma *Greedy Best first search*. Karena algoritma *A star* dan algoritma *Greedy Best first search* merupakan algoritma *informed search* sedangkan algoritma *Uniform Cost Search* merupakan algoritma *blind search*. Algoritma *informed search* merupakan algoritma yang di dalam pencarian pada pohon terdapat informasi misalnya pada simpul atau sisinya. Untuk kasus *route planning* ini informasi yang diberikan adalah berupa *straight line distance* dari satu simpul ke simpul tujuan. Sedangkan *blind search* adalah algoritma pencarian tanpa adanya suatu informasi yang terdapat pada suatu sisi. Dengan tidak adanya informasi ini menyebabkan ruang pohon status yang dibangkitkan menjadi lebih banyak bila dibandingkan dengan algoritma *informed search*. Meskipun banyak ruang status belum tentu solusi yang dihasilkan tidak optimal.

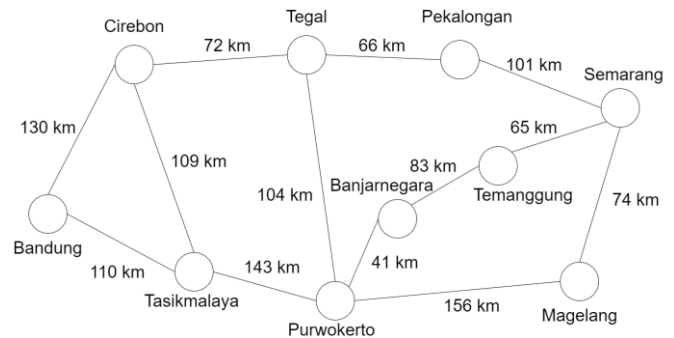
III. PEMBAHASAN

Mudik merupakan tradisi yang terjadi pada saat lebaran setelah bulan Ramadan. Orang ramai-ramai pulang ke kampung halaman yang biasanya jauh dari kota tempat bekerja. Masyarakat umumnya menggunakan aplikasi Google Maps untuk mencapai tujuan mereka dengan jarak yang terdekat sehingga memungkinkan untuk sampai dengan waktu yang realtif cepat. Pada aplikasi Google Maps pada umumnya terdapat jarak antara kota asal dengan kota tujuan serta waktu estimasi pengguna sampai tujuan. Di aplikasi Google Maps yang menjadi prioritas adalah waktu estimasi dibandingkan dengan jarak tempuh, maksudnya aplikasi akan memilih rute yang jaraknya lebih jauh bila tetapi waktu estimasinya lebih cepat dibandingkan dengan yang jaraknya dekat namun waktu estimasi lebih lama. Untuk pembahasan kali ini, faktor waktu estimasi akan diabaikan akan lebih mudah dalam membahasnya sehingga yang menjadi prioritas adalah rute terpendek.

Pada kasus ini, keluarga dari kota Bandung ingin melakukan mudik lebaran menuju kota Yogyakarta. Rute dari Bandung menuju Yogyakarta sangatlah beragam dan bervariasi. Oleh karena itu, pembahasan ini akan disederhanakan akan tidak terlalu panjang dan bertele-tele. Saat ini, kota-kota kecil akan diabaikan sehingga jumlah simpul yang akan ditangani menjadi lebih sedikit. Variasi jalan juga tidak akan terlalu banyak agar memudahkan dalam penyelesaian masalah.

Pada saat arus mudik terdapat dua jalan utama yaitu jalur Pantai Utara (Pantura) atau jalur Selatan. Dari Bandung menuju Semarang jika melewati jalur Pantura maka akan melewati Cirebon, Tegal, Pemalang, Pekalongan, Semarang. Sedangkan jika pemudik melewati jalur Timur dari kota Bandung akan melewati kota Tasikmalaya, Purwokerto dan pada Purwokerto terdapat 2 kemungkinan lajur yaitu melewati Kebumen, Magelang, Semarang atau Banjarnegara, Temanggung, Semarang. Diketahui bahwa jarak dari Bandung menuju Cirebon adalah 130 km, jarak dari Cirebon menuju Tegal adalah 72 km, jarak dari Tegal menuju Pekalongan adalah 66 km, jarak dari Pekalongan menuju Semarang adalah 101 km, jarak dari Bandung menuju Tasikmalaya adalah 110 km, jarak dari Tasikmalaya menuju Purwokerto adalah 143 km jarak dari Purwokerto menuju Magelang adalah 156 km, jarak dari Magelang menuju Semarang adalah 74 km, jarak dari Purwokerto ke Banjarnegara adalah 41 km, jarak dari Banjarnegara menuju Temanggung adalah 65 km, jarak dari Temanggung menuju Semarang adalah 45 km, Jarak dari Cirebon menuju Tasikmalaya adalah 109 km, dan jarak dari Tegal menuju Purwokerto adalah 104 km . Penyelesaian masalah ini akan menggunakan ketiga algoritma *route planning* untuk mendapatkan rute terpendek dari Bandung menuju Semarang. Untuk lebih jelasnya dapat melihat gambar peta rute perjalanan yang mungkin pada gambar 6. Jarak-jarak antar kota yang terdapat pada gambar 6 merupakan jarak sebenarnya atau *true cost* dari satu simpul ke simpul lainnya. Jarak tersebut didapat dari pengukuran menggunakan aplikasi Google Maps. Sebenarnya terdapat banyak sekali simpul dan sisi yang terdapat antara simpul Bandung dan simpul Semarang, namun penyelesaian dibuat lebih sederhana agar

lebih mudah dipahami dan tidak terlalu redundan atau berulang.



Gambar 6. Peta Jarak Sebenarnya Antar Kota

(Sumber: Penulis)

Pada metode algoritma *A star* dan algoritma *Greedy Best first search* diperlukan suatu nilai yang dapat menjadi nilai heuristik dalam penentuan jarak menggunakan algoritma tersebut. Nilai heuristik yang mungkin dapat ditentukan adalah jarak lurus antar kota. Perhitungan jarak lurus antara kota satu dengan kota lain menggunakan aplikasi Google Maps. Jarak lurus yang digunakan adalah jarak lurus dari suatu kota menuju ke kota tujuan. Didapat jarak dari Bandung menuju Semarang adalah 311 km, jarak Cirebon menuju Semarang adalah 211 km, jarak Tegal menuju Semarang adalah 145 km, jarak Pekalongan menuju Semarang adalah 86 km, jarak Tasikmalaya menuju Semarang adalah 244 km, jarak Purwokerto menuju Semarang adalah 141 km, jarak Banjarnegara menuju Semarang adalah 111 km, jarak Temanggung menuju Semarang adalah 45 km, jarak dari Magelang menuju Semarang adalah 57 km, dan jarak Semarang menuju Semarang adalah 0 km. Nilai-nilai heuristik yang digunakan harus dapat diterima dengan melanggar aturan salah satu nilai heuristik harus lebih kecil bila dibandingkan dengan nilai jarak sebenarnya. Contohnya dapat dilihat bahwa jarak sebenarnya Magelang ke Semarang adalah 74 km sedangkan jarak pada nilai heuristik adalah 57 km sehingga tidak melanggar aturan dan dapat diterima. Untuk lebih mempermudah, nilai heuristik disimpan dalam bentuk tabel 1.

No	Lintasan	Jarak Garis Lurus (km)
1	Bandung - Semarang	311
2	Cirebon – Semarang	211
3	Tegal – Semarang	145
4	Pekalongan – Semarang	86
5	Tasikmalaya – Semarang	244
6	Purwokerto – Semarang	141
7	Banjarnegara – Semarang	111
8	Temanggung – Semarang	45
9	Magelang – Semarang	57
10	Semarang - Semarang	0

Tabel 1. Jarak Garis Lurus

(Sumber: Penulis)

Pencarian rute terpendek menggunakan 3 algoritma yang telah disebutkan di landasan teori

A. Algoritma Uniform Cost Search (UCS)

Algoritma *Uniform Cost* adalah algoritma yang mencari rute terpendek berdasarkan nilai *cost* terkecil. Simpul yang dibangkitkan selanjutnya merupakan simpul dengan nilai *cost* terkecil.

Simpul Ekspan	Simpul Hidup
Bandung	Tasikmalaya (110), Cirebon (130)
Tasikmalaya (110)	Cirebon (130), Cirebon (219), Purwokerto (253)
Cirebon (130)	Tegal (202), Cirebon (219), Purwokerto (253)
Tegal (202)	Pekalongan (268), Purwokerto (306)
Pekalongan (268)	Purwokerto (306), Semarang (369)
Purwokerto (306)	Banjarnegara (347), Semarang (369), Magelang (462)
Banjarnegara (347)	Semarang (369), Temanggung (430), Magelang (462)
Semarang (369)	Solusi ditemukan

Tabel 2. Pohon Ruang Status UCS

(Sumber: Penulis)

Dengan menggunakan algoritma *Uniform Cost Search* tidak didapat rute terpendek dari Bandung – Semarang adalah didapat rute terpendek dari Bandung – Semarang adalah Bandung → Cirebon → Tegal → Pekalongan → Semarang. Nilai *cost* yang dibutuhkan untuk mencapai Semarang dari Bandung adalah 369 km

B. Algoritma Greedy Best first search

Algoritma *Greedy Best first search* adalah algoritma penentuan jarak dengan mencari *cost* terkecil berdasarkan nilai heuristik yang telah diketahui sebelumnya.

Simpul Ekspan	Simpul Hidup
Bandung	Cirebon (211), Tasikmalaya (244)
Cirebon (211)	Tegal (145), Tasikmalaya (244)
Tegal (145)	Pekalongan (86), Tasikmalaya (244).

Pekalongan (86)	Semarang (0), Tasikmalaya (244)
Semarang (0)	Solusi ditemukan

Tabel 3. Pohon Ruang Status Greedy Best first search

(Sumber: Penulis)

Dengan menggunakan algoritma *Greedy Best first search* didapat rute terpendek dari Bandung – Semarang adalah Bandung → Cirebon → Tegal → Pekalongan → Semarang. Nilai *cost* yang dibutuhkan untuk mencapai Semarang dari Bandung adalah 369 km.

C. Algoritma A star

Algoritma *A star* adalah algoritma yang menggabungkan antara *cost* jarak sebenarnya ditambah dengan nilai heuristik. Algoritma *A star* ini merupakan algoritma yang paling optimal bila dibandingkan dengan algoritma *Uniform Cost Search* dan algoritma *Greedy Best first search*.

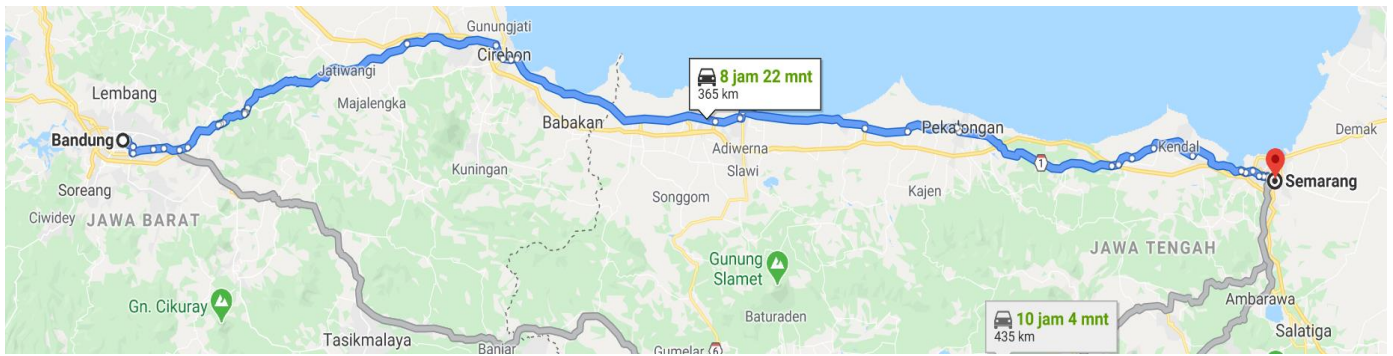
Simpul Ekspan	Simpul Hidup
Bandung	Cirebon (341), Tasikmalaya (354)
Cirebon (341)	Tegal (347), Tasikmalaya (354), Tasikmalaya (483)
Tegal (347)	Pekalongan (354), Tasikmalaya (354), Purwokerto (447)
Pekalongan (354)	Tasikmalaya (354), Semarang (369), Purwokerto (447)
Tasikmalaya (354)	Semarang (369), Purwokerto (394), Purwokerto (447)
Semarang (369)	Solusi ditemukan

Tabel 4. Pohon Ruang Status A star

(Sumber: Penulis)

Dengan menggunakan algoritma *Greedy Best first search* didapat rute terpendek dari Bandung – Semarang adalah Bandung → Cirebon → Tegal → Pekalongan → Semarang. Nilai *cost* yang dibutuhkan untuk mencapai Semarang dari Bandung adalah 369 km.

Dari ketiga algoritma diatas didapat hasil yang identik baik untuk penentuan rute mudik maupun *cost* yang dihasilkan sehingga diketahui pada dasarnya penentuan rute pada aplikasi Google Maps memanfaatkan penerapan algoritma *route planning* seperti algoritma *Uniform Cost Search*, algoritma *Greedy Best first search*, dan algoritma *A star*. Pada gambar 7 merupakan tampilan dari aplikasi Google Maps sebagai pembandingan bahwa penerapan yang diimplementasikan pada makalah ini telah sesuai dengan aplikasi tersebut. Penerapan ini merupakan dasar dari kekompleksitasan aplikasi Google Maps. Penerapan ini tidak memperhatikan estimasi waktu karena bukan merupakan bagian topik dari algoritma-algoritma *route planning* saat ini.



Gambar 7. Pencarian Rute dengan Google Maps

IV. KESIMPULAN

Dari Bab III yang merupakan pembahasan mengenai penerapan algoritma *route planning* yaitu algoritma *Uniform Cost Search*, algoritma *Greedy Best first search*, dan algoritma *A star* dapat disimpulkan bahwa aplikasi Google Maps Ketika pengguna melakukan pencarian rute terdekat untuk rute mudik merupakan pemanfaatan ketiga algoritma diatas. Pada dasarnya penerapan yang digunakan adalah ketiga algoritma dan untuk pembahasan lebih mendalam akan dibahas di makalah lainnya.

V. UCAPAN TERIMA KASIH

Pertama-tama, penulis mengucapkan puji syukur kepada Allah swt. Tuhan yang Maha Esa atas segala nikmat kesehatan maupun kekuatan sehingga penulis dapat menyelesaikan tugas makalah IF2211 Strategi Algoritma. Terima kasih kepada dosen Strategi Algoritma, Ibu Ulfa, Pak Rinaldi, dan Ibu Masayu yang telah memberikan ilmu yang bermanfaat untuk menyokong pembuatan makalah Strategi Algoritma ini. Semoga Allah swt. membalas semua kebaikan dengan kebaikan yang berlipat ganda. Semoga pembahasan pada makalah ini tidak berhenti sampai disini dan terus dikembangkan lebih lanjut lagi. Makalah ini bukan makalah sempurna, masih kekurangan didalamnya. Penulis memohon maaf yang sebesar-besarnya.

VIDEO LINK AT YOUTUBE

Berikut adalah laman video penjelasan yang diupload di media berbagi video Youtube dengan link:

<https://youtu.be/wuoDNjODaQI>

REFERENCES

- [1] Munir,Rinaldi."Route planning Bagian 1".2020.Route-Planning-Bagian1.pptx.Diakses pada 4 Mei 2020
- [2] Munir,Rinaldi."Route planning Bagian 2".2020.Route-Planning-Bagian2.pptx.Diakses pada 4 Mei 2020
- [3] Anonim."Google Maps".2020.
https://id.wikipedia.org/wiki/Google_Maps.Diakses pada 4 Mei 2020
- [4] Yasmin, Puti."7 Fitur Baru Canggih Google Maps yang Wajib Dicoba".
<https://inet.detik.com/cyberlife/d-4898711/7-fitur-canggih-google-maps-terbaru-yang-wajib-dicoba>.Diakses pada 4 Mei 2020
- [5] Munir, Rinaldi."BFS dan DFS".2020.
[http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/BFS-dan-DFS-\(2020\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/BFS-dan-DFS-(2020).pdf).Diakses pada 4 Mei 2020.

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, 4 Mei 2020

Ttd
Faris Fadhilah 13518026