

Implementasi Algoritma Greedy pada Lampu Lalu Lintas untuk Mengurangi Kemacetan

Alfian Kafilah Ba'its 120140109 (*Author*)

Program Studi Teknik Informatika

Jurusan Teknologi Produksi dan Industri

Institut Teknologi Sumatera, Jalan Terusan Ryacudu, Lampung Selatan

E-mail (gmail): alfian.120140109@student.itera.ac.id

Abstract— Algoritma Greedy adalah salah satu cara yang digunakan untuk mendapatkan hasil yang optimal tanpa memperhatikan keadaan sekitar. Penerapan algoritma greedy ini digunakan untuk mengatur warna lampu lalu lintas yang dapat bekerja sesuai dengan kondisi di persimpangan jalan. Implementasi graf juga bertujuan untuk menentukan kendaraan yang masuk dan kendaraan yang keluar. Dengan demikian, pengimplementasian Algoritma Greedy tersebut mampu mendapatkan solusi yang mendekati hasil optimal.

Keywords—*algoritma greedy; graf; optimum; lampu lalu lintas.*

I. PENDAHULUAN

Kemacetan merupakan kondisi terhambatnya arus lalu lintas sehingga mobilitas kendaraan sangat terbatas. Jumlah kendaraan yang selalu meningkat menjadi salah satu faktor kemacetan di jalan. Permasalahan ini tentu saja berdampak besar pada berbagai kehidupan, seperti ekonomi, pendidikan, kesehatan, produktivitas, dan lain sebagainya. Oleh karena itu, dibutuhkan salah satu cara untuk meningkatkan arus lalu lintas menjadi lebih teratur dan kondusif.

Sarana yang digunakan untuk mengatur arus lalu lintas adalah lampu lalu lintas. Hal itu dapat mengatur arah jalan kendaraan yang sedang melintas di persimpangan jalan. Lampu lalu lintas ini terdiri dari 3 kondisi warna, yaitu lampu berwarna merah penanda untuk berhenti, lampu berwarna kuning penanda untuk waspada, dan lampu berwarna hijau penanda untuk jalan.

Saat ini, lampu lalu lintas yang digunakan pada setiap persimpangan jalan masih belum terkomputasi dengan baik dan masih diatur dengan konstan maupun manual. Durasi perubahan kondisi warna pada setiap lampu lalu lintas nilainya selalu tetap dan tidak memperhatikan kondisi persimpangan jalan tersebut. Lampu lalu lintas tersebut menunjukkan kondisi lampu berwarna merah namun persimpangan jalan lain dipenuhi kendaraan. Kemudian, lampu lalu lintas berwarna hijau namun persimpangan jalan lain sepi kendaraan.

Lampu lalu lintas yang terletak di persimpangan jalan seharusnya mempunyai cara kerja yang ideal. Ketika lampu lalu lintas berwarna hijau dan arus kendaraan yang melewati persimpangan jalan tersebut sangat padat, maka durasi waktu lampu berwarna hijau tersebut harus lebih lama dibandingkan sisi jalan yang sepi.

Maka dari itu, diperlukan suatu cara untuk mencegah kemacetan lalu lintas di persimpangan jalan dengan memperbaiki kinerja dari lampu lalu lintas tersebut. Pendekatan yang paling tepat untuk digunakan pada kasus ini ialah dengan menggunakan algoritma greedy yang diterapkan pada sensor tiap lampu lalu lintas. Algoritma greedy ini sangat cocok untuk mengatur kendaraan yang melintas karena dibantu dengan teknik graf. Kemudian, Algoritma tersebut dapat mengambil keputusan yang tepat dan paling menguntungkan dari banyak kemungkinan.

II. TEORI DASAR

A. Algoritma Greedy

Algoritma Greedy adalah salah satu cara yang digunakan untuk mendapatkan hasil yang optimal tanpa memperhatikan keadaan sekitar. Algoritma ini mempunyai cara kerja secara langkah per langkah. Pada setiap langkahnya harus dieksplorasi kembali untuk mendapatkan keputusan yang terbaik ketika menentukan sebuah pilihan. Akan tetapi, keputusan yang telah diambil pada masing-masing langkah tersebut tidak dapat diubah lagi pada langkah selanjutnya. Hal itu disebut dengan solusi dengan cara optimum lokal

Algoritma Greedy ini

Adapun elemen-elemen algoritma greedy, yaitu :

1. Himpunan Kandidat, S
Berisi kandidat yang akan dipilih dan sebagai elemen pembentuk solusi. Pada setiap langkah hanya diambil satu buah kandidat dari himpunannya
2. Himpunan Solusi, C
Berisi kandidat yang sudah dipilih dan sebagai solusi persoalan.
3. Fungsi Solusi
Berisi penentuan kandidat yang sudah dipilih tersebut dapat menghasilkan sebuah solusi persoalan.
4. Fungsi Seleksi
Berisi pemilihan kandidat yang didasarkan pada sifat heuristik untuk mencapai solusi optimal.
5. Fungsi Kelayakan

Berisi pemeriksaan untuk menentukan kandidat yang layak dimasukkan ke dalam himpunan solusi. Kandidat tersebut harus terbentuk bersama dengan himpunan solusi yang sudah ada dan tidak melanggar berbagai kendala.

6. Fungsi Objektif

Berisi pengoptimalan sebuah solusi dengan menggunakan 2 jenis optimasi, yaitu maksimum dan minimum.

Dengan demikian, algoritma greedy ini tentu saja melibatkan banyak himpunan bagian, S , dari himpunan kandidat C . Kemudian, kandidat S harus memenuhi berbagai syarat untuk menjadi solusi sehingga S dapat dioptimisasi dengan fungsi objektif..

Berikut skema umum algoritma greedy:

```
function greedy(input C: himpunan_kandidat)→
himpunan_kandidat { Mengembalikan solusi dari
persoalan optimasi dengan algoritma greedy Masukan:
himpunan kandidat C Keluaran: himpunan solusi yang
bertipe himpunan_kandidat }
```

Deklarasi x : kandidat

S : himpunan_kandidat

Algoritma:

```
S ← {} { inisialisasi S dengan kosong }
while (not SOLUSI(S)) and (C ≠ {} )do
x ← SELEKSI(C) { pilih sebuah kandidat dari C}
C ← C - {x} { elemen himpunan kandidat berkurang
satu }
if LAYAK(S ∪ {x}) then
    S ← S ∪ {x}
endif
Endwhile
{ SOLUSI(S) or C = {} }
if SOLUSI(S) then
    return S
else
    write('tidak ada solusi')
endif
```

Dengan demikian, solusi yang terbentuk adalah optimum lokal dan setiap akhir perulangan while-do diperoleh optimum global. Akan tetapi, tiap optimum global belum tentu mendapatkan hasil yang terbaik, namun *sub-optimum* atau *pseudo-optimum*

Algoritma greedy dapat memperoleh solusi optimum secara global. Akan tetapi solusi yang didapat belum tentu mendapatkan solusi yang optimum (terbaik). Hal itu dikarenakan algoritma greedy bekerja tidak secara menyeluruh terhadap berbagai alternatif solusi yang ada dan juga terdapat berbagai fungsi seleksi yang berbeda-beda. Untuk mendapatkan hasil yang optimum, maka hal itu harus membuat fungsi seleksi identik dengan fungsi objektif. Dengan demikian, algoritma greedy ini hanya dapat memberikan solusi yang mendekati nilai optimum.

B. Graf

Graf merupakan salah satu ilmu matematika yang berkembang sangat pesat karena berhubungan dengan operasi, himpunan, dan sifat-sifat struktur. Graf adalah suatu himpunan pada objek dan tiap sisi dihubungkan dengan sebuah node sehingga masing-masing objek atau verteks saling terhubung satu sama lain.

Teori grafi sendiri disajikan dengan titik (verteks) dan garis (edge). Hal itu digunakan untuk merepresentasikan berbagai objek diskrit dan hubungannya satu sama lain. Struktur pada graf itu dibuat dan dikembangkan dengan cara membuat bobot pada setiap edge. Graf yang sudah dibuat itu menjelaskan tentang berbagai konsep yang berbeda. Misalnya, apabila suatu graf menjelaskan atau melambangkan suatu jaringan jalan maka bobotnya adalah panjang jalan. Selanjutnya, terdapat graf berarah yang dibuat dengan cara membuat edgenya berarah.

Suatu graf G terdiri atas dua himpunan yang berhingga, di antaranya adalah himpunan yang berisi banyak titik yang kosong itu disimbolkan dengan $V(G)$ dan himpunan yang berisi banyak garis itu disimbolkan dengan $E(G)$. Kemudian, setiap garis yang dibuat itu saling berhubungan pada sebuah satu atau dua titik. Apabila garis hanya dihubungkan pada satu titik, maka graf itu disebut Loop. Kemudian, apabila dua garis yang berbeda itu dihubungkan pada titik yang sama, maka hal itu disebut garis paralel.

Graf berarah adalah semua garis yang menghubungkannya itu berisi arah. Apabila semua garisnya tidak berarah, maka graf itu disebut dengan graf tak berarah. Biasanya, graf itu dinyatakan dengan gambar. Misalnya, gambar pada suatu graf G itu terdiri dari himpunan titik-titik dan himpunan garis-garis yang saling terhubung dengan jelas (berarah atau tidak).

C. Dynamic Programming

Dynamic Programming adalah suatu cara kuantitatif yang digunakan untuk menghubungkan satu sama lain dalam membuat suatu rangkaian. Hal tersebut termasuk prosedur matematis yang digunakan dalam merancang dan memperbaiki perhitungan permasalahan secara matematis dengan menguraikan atau membaginya menjadi masalah-masalah yang kecil sehingga kedepannya mudah untuk diselesaikan.

Tujuan dari *Dynamic Programming* itu sendiri adalah untuk mengoptimalkan berbagai permasalahan dengan melalui pendekatan matematika dan memperhatikan berbagai batasan yang ada. Kemudian, permasalahan tersebut diubah menjadi sub-sub terkecil dari masalah.

Berikut kelebihan dan kekurangan *Dynamic Programming* di antaranya adalah sebagai berikut:

Kelebihan

- Berbagai pendekatan yang dilakukan oleh *dynamic programming* mampu memecahkan permasalahan secara fleksibel

- *Dynamic programming* itu dapat menyesuaikan sistematika perhitungan menurut ukuran permasalahan yang selalu berubah-ubah dengan cara yang tetap dan melakukan langkah demi langkah hingga permasalahan terselesaikan.
- Proses pemecahan permasalahan pada *Dynamic Programming* dapat membuat suatu permasalahan diselesaikan dengan runtut dan lebih jelas untuk diketahui akar masalahnya.
- Terdapat Sensitivitas pada setiap variabel pada proses perhitungan.

Kekurangan

- *Dynamic Programming* tidak efisien apabila hal itu dilakukan secara tidak tepat. Maka dari itu dibutuhkan pengetahuan dan keahlian yang berkaitan dengan merumuskan suatu masalah yang kompleks
- Penambahan perhitungan pada variabel matematika akan menambah beban memori komputer sehingga perhitungan menjadi lebih lambat.
- *Dynamic Programming* tidak mempunyai bentuk formulasi matematika yang tetap sehingga ketika melakukan perhitungan maka solusi optimal menjadi lebih terbatas.

III. ANALISIS DAN PEMBAHASAN

Pada kasus permasalahan ini, dapat diterapkan dan dianalogikan sebagai perancangan suatu sistem yang menggunakan input dan juga output yang terdefinisi. Sistem tersebut ialah pembuatan sensor pengaturan durasi waktu pada lampu lalu lintas. Sistem tersebut meliputi :

1. Sensor pendeteksi pada sistem

Sensor yang dibuat itu memiliki konsep bahwa pada sistem pengaturan lalu lintas itu menerapkan operasi dinamis algoritma greedy. Algoritma tersebut didasarkan pada panjang antrian di setiap sisi persimpangan jalan. Apabila persimpangan sedang sepi atau kendaraan tidak ada yang mengantri, maka lampu lalu lintas di simpang jalan tersebut beroperasi secara normal. Sebaliknya, apabila persimpangan sedang ramai atau kendaraan sedang mengantri panjang, maka durasi lampu hijau akan lebih lama. Untuk membaca dan mengetahui kasus ini, maka dibutuhkan sensor inframerah. Tiap persimpangan jalan terdapat lima sensor yang terpasang. Pada jarak 0 cm – 10 cm, dan 10 cm – 25 cm untuk setiap simpang dan dua sensor pada jalur khusus seperti Bus Way dan sebagainya.

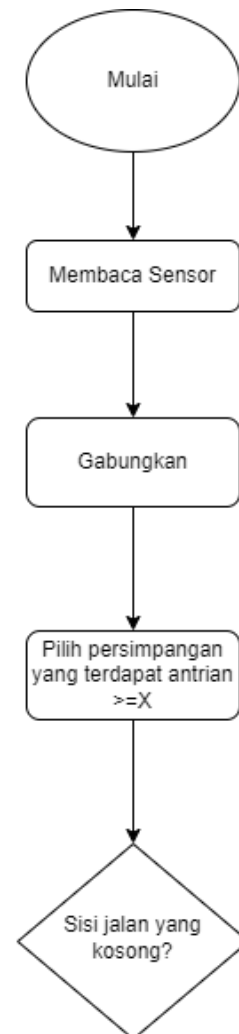
Penempatan sensor pada tiap simpang jalan tentu saja mampu mendeteksi antrian kendaraan yang terdapat pada sisi jalan persimpangan. Penentuan durasi lampu hijau ini menerapkan algoritma greedy yang mampu meningkatkan kinerja hal tersebut. Penggunaan sensor

tersebut diterapkan pada Arduino yang telah dilengkapi algoritma greedy.

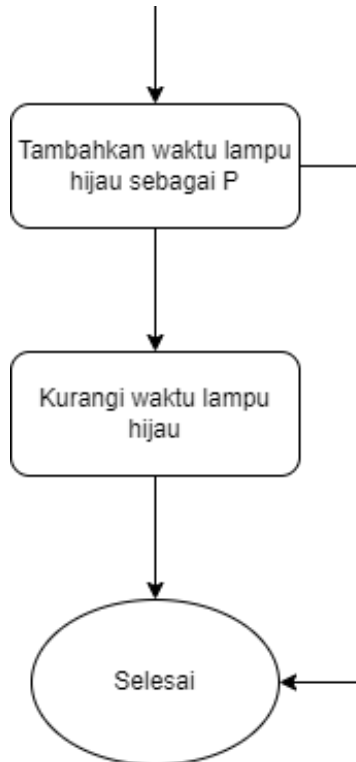
2. Output Sistem

Sensor yang digunakan tersebut mempunyai persyaratan sebagai berikut : Mobil adalah objek yang akan dideteksi oleh sensor infra merah. Selanjutnya, inframerah terdiri dari penerima dan pemancar memancarkan cahaya inframerah. Jika cahaya terhalang oleh suatu objek, maka infra merah sebagai penerima akan mendeteksi objek tersebut dan sinyal akan menyala. Arduino sebagai pengontrol akan memproses input yang diterima dari infra merah sensor, yang kemudian akan diproses dan diterapkan pada algoritma greedy untuk diteruskan ke Komponen rangkaian lampu LED. Pada set komponen LED, akan menerima proses Arduino dan menambahkan penerapan algoritma greedy yang digunakan sebagai output. Output ini akan mengatur lamanya waktu yang menyatakan kondisi lampu hijau menyala.

A. Flow Chart



Flowchart diatas merupakan langkah-langkah dari algoritma greedy. Sensor yang dibuat akan membaca kepadatan lalu lintas dan data akan diikirmkan ke Arduino. Sensor mampu membaca panjang antrian kendaraan lebih besar atau sama dengan X. X itu sendiri merupakan sensor yang lebih besar dan dapat memasukkan nilai ke dalam sirkuit resolusi.



Selanjutnya algoritma greedy ini mampu menghasilkan optimal yang dapat dilihat pada gambar sebagai berikut :

$S^* \cap \{a_1, \dots, a_j\} = S_i$
 $j = 1$:
 Let $S^* \subseteq \{a_1, \dots, a_m\}$ be some optimal solution, $S^* = \{a_{j1}, \dots, a_{jk}\}$
 $S^* = a_{j1}, a_{j2}, a_{j3}, \dots, a_{jk}$
 $j - 1 \rightarrow j$:
 Let $S^* \subseteq \{a_1, \dots, a_m\}$ be some optimal solution with $S^* \cap \{a_1, \dots, a_{j-1}\} = S_{j-1}$
 Consider $R = S^* \setminus S_{j-1}$

Selanjutnya, solusi optimal untuk menemukan himpunan ukuran maksimum dari kasus tersebut dapat dilihat pada gambar sebagai berikut:

Case 1: $s_j < f_{last}$

$$S_{j-1} = s_j, f_1, \dots, a_{last}, f_{last}$$

S_j is not compatible with S_{j-1}

S_j is not contained in S^*

$$S^* \cap \{a_1, \dots, a_j\} = S_{j-1} = S_i$$

Case 2: $s_j \geq f_{last}$

$$S_{j-1} = s_j, f_1, \dots, a_{last}, f_{last}$$

S_j is compatible with S_{j-1}

There is: $R \subseteq \{a_j, \dots, a_n\}$

$$R = b_1, b_2, b_3, \dots, b_j a_j$$

$B^* = S_{j-1} \cup (R \setminus \{b_1\}) \cup \{a_j\}$ is optimal

$$B^* \cap \{a_1, \dots, a_j\} = S_{j-1} \cup \{a_j\} = S_j$$

B. Graf

• Implementasi Node

Node merupakan simbol dari lampu lalu lintas. Node ini terdiri atas durasi waktu, status waktu, jumlah kepadatan kendaraan, dan nomor node. Pada permasalahan di atas, terdapat 18 node yang mampu menggambarkan lampu lalu lintas.

• Implementasi Busur

Busur pada graf ini menggambarkan hubungan antara dua buah node. Sehingga, kedua node tersebut dapat terhubung satu sama lain dan membentuk representasi dari lalu lintas.

Dari permasalahan lalu lintas tersebut terdapat tabel yang menyatakan daftar busur. Hal itu dapat dilihat sebagai berikut:

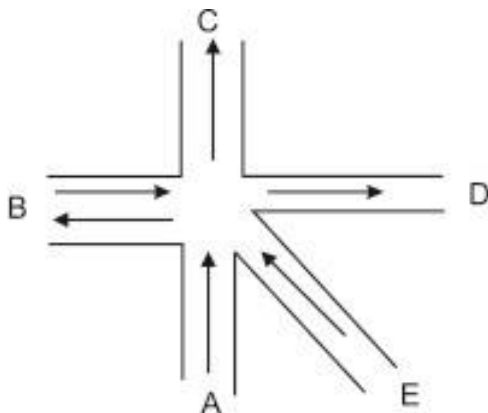
Daftar Busur	
BUSUR(START1,1)	BUSUR(5,9)
BUSUR(START2,2)	BUSUR(6,17)
BUSUR(START2,10)	BUSUR(7,4)
BUSUR(START3,11)	BUSUR(8,9)
BUSUR(START2,12)	BUSUR(9,13)
BUSUR(START4,15)	BUSUR(10,7)
BUSUR(START4,16)	BUSUR(11,7)
BUSUR(START5,18)	BUSUR(15,4)
BUSUR(1,5)	BUSUR(18,10)

BUSUR(1,6)	BUSUR(18,12)
------------	--------------

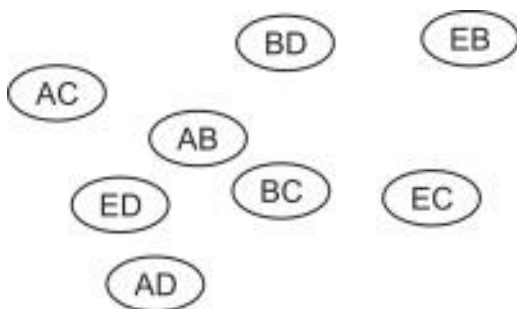
BUSUR(1,END1)
BUSUR(2,END2)
BUSUR(3,END2)
BUSUR(9,END3)
BUSUR(13,END4)
BUSUR(12,END3)
BUSUR(14,END5)
BUSUR(16,END6)
BUSUR(17,END6)

Dapat disimpulkan bahwa tabel di atas merupakan implementasi untuk menentukan arah lalu lintas sehingga lalu lintas tidak terjadi kemacetan. Tabel tersebut berisi 8 cara untuk masuk ke daerah problem. Selanjutnya terdapat 10 cara untuk keluar yang terdiri atas 6 titik keluar dan 5 titik masuk.

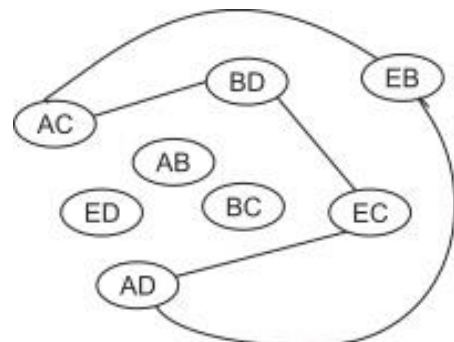
Berikut contoh persimpangan jalan dengan memanfaatkan teori graf.



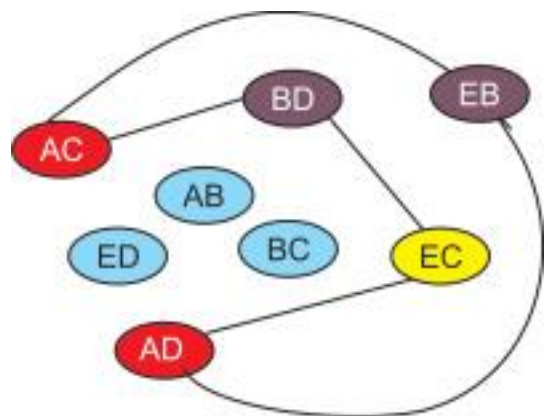
Pada gambar di atas, terdapat lima simpang yang terdiri atas sisi A, B, C, D, dan E. Sisi A hanya dapat melewati sisi B. Sisi A hanya dapat melewati sisi C, dan sisi A hanya dapat melewati sisi D. Sehingga graf tersebut dapat digambarkan menjadi sebuah simpul



Langkah selanjutnya ialah dengan 2 simpul yang akan melewati node tersebut. Pada gambar tersebut, dapat dilihat bahwa jalur AB dan BD bersebrangan sehingga dapat dihubungkan simpulnya satu sama lain. Hal itu dapat dilihat pada gambar sebagai berikut.



Selanjutnya, beri warna pada masing-masing node dengan menyesuaikan warna lampu lalu lintas. Node-node yang diberi warna hanyalah node-node yang saling terhubung satu sama lain. Sehingga, dapat dilihat pada gambar sebagai berikut.



C. Langkah-langkah penyelesaian dengan menggunakan graf

- Inisialisasi dengan menginputkan node, busur, dan constraint.
- Memulai Perulangan sebagai penanda bahwa sistem telah berjalan sebagaimana mestinya
- Menerima Node yang berisi informasi kepadatan kendaraan. Proses ini terus berjalan dan menerima informasi melalui sensor yang telah dibuat secara real time. Kemudian, secara otomatis dapat melakukan pencarian node yang paling panjang.
- Mendapatkan graf yang optimal, graf ini saling terhubung satu sama lain yang menghubungkan node pada flowchart mulai sampai selesai.

D. Langkah-langkah algoritma greedy dengan skema umum

1. Membangun dan memilih kandidat yang dapat dijadikan sebuah solusi pada permasalahan lalu lintas ini
2. Menginisialisasi himpunan solusi S menjadi himpunan yang kosong
3. Pilih satu langkah yang akan digunakan sebagai pengujian oleh fungsi objektif
4. Mencari selisih rata-rata solusi tersebut kemudian jadikan sebagai fungsi objektif
5. Melakukan fungsi kelayakan dan juga fungsi seleksi terhadap solusi sementara. Hal itu dilakukan dengan membandingkan nilai SR yang ada dengan nilai SR sementara sehingga akan didapati sebuah hasil yang optimal
6. Mengulangi langkah-langkah tersebut hingga seluruh anggota pada himpunan sudah dihitung dan prosesnya selesai

IV. KESIMPULAN

Permasalahan arus lalu lintas seperti kemacetan yang terjadi persimpangan jalan merupakan masalah yang sering dihadapi oleh berbagai daerah. Lampu lalu lintas yang masih diatur secara konstan dan manual ternyata juga masih menimbulkan kemacetan. Dengan mengintegrasikan suatu sistem dengan penerapan Algoritma Greedy membuat lampu lalu lintas bekerja secara optimal. Sehingga, pengguna jalan dapat meminimalkan waktu dan terhindar dari macet di persimpangan jalan.

Algoritma greedy tentu saja tidak dapat menghasilkan solusi yang optimal. Algoritma ini hanya dapat menghasilkan solusi yang mendekati optimal sehingga masih banyak faktor-faktor yang harus diperhatikan kembali, seperti faktor lebar jalan, jumlah jalur, dan lain sebagainya.

VIDEO LINK AT YOUTUBE

https://youtu.be/I_w004HmpEg

UCAPAN TERIMA KASIH

Alhamdulillah, puji syukur atas kehadiran Allah SWT. karena atas kehendak dan ridho-Nya jurnal Implementasi Algoritma Greedy pada Lampu Lalu Lintas untuk Mengurangi Kemacetan telah diselesaikan. Saya berharap, jurnal yang saya buat dapat bermanfaat bagi siapapun yang membacanya. Kemudian, saya ucapkan terima kasih kepada Pak Imam selaku dosen dari mata kuliah Strategi Algoritma yang telah memberikan saya banyak pengetahuan dari hal-hal yang belums saya ketahui. Kemudian, jenis-jenis algoritma yang telah diajarkan ternyata sangat berdampak bagi kehidupan sehari-hari terutama dalam menyelesaikan suatu permasalahan.

REFERENCES

- [1] Munir, Rinaldi, "Diktat Kuliah IF3051 Strategi Algoritma", Bandung : Institut Teknologi Bandung, 2009
- [2] Edwin Romelta, Metode Pencarian Lintasan Terpendek Dalam Graf, JURNAL ILMU KOMPUTER DAN TEKNOLOGI INFORMASI, VOL III NO.2, OKTOBER 2003

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.

Lampung, 29 Maret 2022



Alfian Kafilah Ba'its 120140109