Aplikasi Graf dalam Pengaturan Lampu Lalu Lintas

Patrick Nugroho Hadiwinoto, 13515040
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515040@std.stei.itb.ac.id

***Abstrak*—Lampu lalu lintas merupakan salah satu komponen penting dalam menunjang keselamatan di jalan raya. Lampu lalu lintas mempunyai tiga warna yang selalu berurutan dalam menyala. Urutan antara warna satu dengan yang lain sangatlah penting untuk diperhatikan, karena ini berhubungan dengan keselamatan orang banyak, apalagi bila terdapat di perempatan. Urutan penyalaan warna antara beberapa lampu lalu lintas di perempatan jalan ini menggunakan prinsip graf. Penerapan graf dalam hal ini yaitu untuk mengatur sisi jalan mana yang dapat jalan terlebih dahuhlu dan selang waktu di antaranya. Prinsip ini terbilang sederhana, namun memiliki peran vital dalam mencegah kecelakaan di jalan raya.**

***Kata Kunci*— Graf, lampu lalu lintas, pengurutan**

I. Pendahuluan

Dewasa ini, jumlah kendaraan yang ada di jalan raya semakin banyak dan beragam. Jalan raya yang dahulu sepi sekarang menjadi lebih ramai dan padat. Semua orang tentu ingin sampai tujuan dengan cepat, sehingga semua ingin lebih dahulu dari yang lain. Hal ini bisa menimbulkan bahaya di jalan raya. Terlebih dengan jumlah kendaran yang begitu banyak dan pengendara yang memiliki karakter berkendara yang beragam.

Maka dari itu, harus ada solusi untuk mengatur kendaraan-kendaraan tersebut di jalan raya. Salah satu solusinya adalah dengan lampu lalu lintas. Lampu ini memiliki tiga warna: merah, kuning dan hijau. Merah yang berarti berhenti. Kuning berarti bersiap-siap dan warna hijau berarti jalan.

Di setiap persimpangan jalan pastilah ada lampu lalu lintas. Hal ini untuk mencegah terjadinya tabrakan antar kendaraan yang melintas. Marilah ambil contoh di perempatan jalan. Ada 4 ruas jalan yang bertemu menjadi satu dengan masing-masing satu lampu lalu lintas dan masing-masing ruas terdapat banyak kendaraan yang akan melintas.

Dengan adanya lampu lalu lintas, urutan melintasnya kendaraan dapat diatur, sehingga tidak saling bertabrakan. Urutan menyala dari keempat lampu lalu lintas di perempatan inilah yang diatur dengan teori graf.

Teori ini mempelajari bagaimana mengatur lampu A akan memberikan warna apa, sehingga tidak bertabrakan dengan lampu B, C dan D. Durasi penyalaan warna juga diperhitungkan. Begitu pula dengan jeda antara lampu yang satu dengan lampu yang lain dalam memberikan warna.

II. Dasar Teori

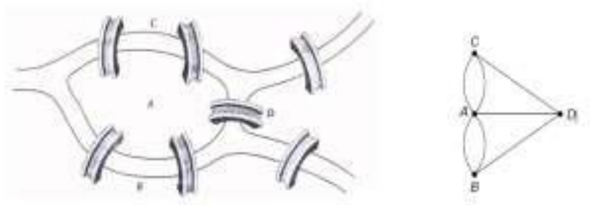
Graf G didefinisikan sebagai pasangan himpunan (V, E), yang dalam hal ini:

V = himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*) = $\{v_1, v_2, \ldots, v_n\}$

dan

E = himpunan sisi (*edge* atau *arcs*) yang menghubungkan sepasang simpul = $\{e_1, e_2, \ldots, e_n\}$

atau dapat ditulis singkat notasi $G=(V,E)$.



(a) (b)

Gambar 2.1

(a) Jembatan Konigsberg [ROS99], dan (b) graf yang merepresentasikan jembatan Konigsberg

Sumber : Rinaldi Munir, *Diktat Kuliah IF2120 Matematika Diskrit*, Teknik Informatika ITB, 2006

Diakses: 9 Desember 2016

2.1. Jenis Graf

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, maka graf digolongkan menjadi dua:

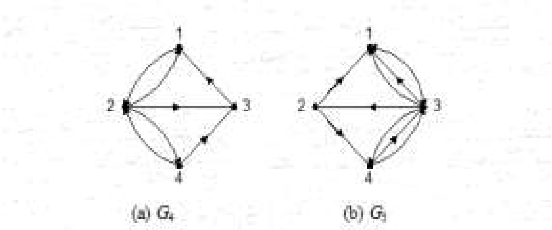
1. Graf sederhana (*simple graph*)
   Adalah graf yang tidak mengandung gelang (*loop*) maupun sisi ganda.
2. Graf tak-sederhana (*unsimple graph*)
   Adalah graf yang mengandung sisi ganda atau gelang. Ada dua jenis graf tak-sederhaa, yaitu graf ganda dan graf semu. Graf ganda adalah graf yang mengandung sisi ganda. Sedangkan graf semu adalah graf yang mengandung gelang.

Berdasarkan jumlah simpul pada suatu graf, maka graf dapat dibedakan menjadi dua:

1. Graf berhingga (*limited graph*)

Yaitu graf yang jumlah simpulnya berhingga.
2. Graf tak-berhingga (*unlimited graph*)
Yaitu graf yang jumpah simpulnya tidak berhingga banyaknya.

Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan menjadi dua:
1. Graf tak-berarah (*undirected graph*)
Yaitu graf yang sisinya tidak mempunya orientasi arah.
2. Graf berarah (*directed graph* atau *digraph*)
Adalah graf yang setiap sisinya diberikan orientasi arah. Sehingga ada istilah simpul asal dan simpul terminal.



Gambar 2.2 (a) Graf berarah (b) Graf ganda berarah
Sumber : Rinaldi Munir, *Diktat Kuliah IF2120 Matematika Diskrit*, Teknik Informatika ITB, 2006
Diakses: 9 Desember 2016

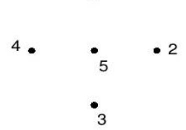
2.2. Terminologi Dasar dalam Graf

Beberapa terminologi yang terkait dengan graf:
1. Bertetangga (*Adjacent*)
Dua simpul pada graf dikatakan bertetangga jika kedua simpul tersebut terhubung langsung oleh suatu sisi.

2. Bersisian (*Incident*)
Untuk sembarang sisi $e = (v_j,v_k)$, sisi e dikatakan bersisian dengan simpul v_j dan simpul v_k

3. Simpul terpencil (*Isolated simpul*)
Adalah simpul yang tidak memiliki sisi yang bersisian dengannya (tidak bertetangga dengan simpul manapun)

4. Graf kosong (*Null graph* atau *empty graph*)
Adalah graf yang himpunan sisinya merupakan himpunan kosong.



Gambar 2.3 Graf Kosong (N_5)
Sumber : Rinaldi Munir, *Diktat Kuliah IF2120 Matematika Diskrit*, Teknik Informatika ITB, 2006
Diakses: 9 Desember 2016

5. Derajat (*Degree*)
Derajat suatu simpul pada graf tak berarah adalah jumlah sisi yang bersisian dengan simpul tersebut.

6. Lintasan (*Path*)
Adalah sekumpulan sisi yang menghubungkan simpul-simpul yang ada digraf. Sisi sisi tersebut terhubung dengan tidak terputus.

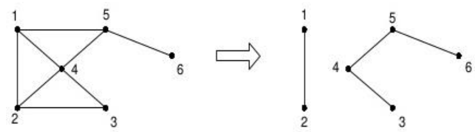
7. Siklus (*Cycle*) atau Sirkuit (*Circuit*)
Adalah lintasan yang berawal dan berakhir pada simpul yang sama.

8. Terhubung (*Connected*)
Graf tak berarah G disebut graf terhubung jika untuk setiap pasang simpul paadd graf tersebut ada lintasan yang melaluinya. Jika tidak, graf G disebut graf tak-terhubung.

9. Upagraf (*Subgraf*) dan Komplemen Upagraf
Graf G disebut sebagai upagraf dari graf H jika himpunan sisi-sisi dan simpul-simpul graf G merupakan himpunan bagian dari sisi-sisi dan simpul-simpul graf H .
Komplemen dari upagraf G terhadap graf H adalah himpunan simpul dan sisi graf H yang bukan termasuk graf G.

10. Upagraf Merentang (*Spanning subgraf*)
Graf G1 dikatakan upagraf rentang dari G jika G1 mengandung semua simpul dari G.

11. Cut-Set
Cut set dari graf terhubung G adalah himpunan sisi yang bila dibuang dari G menyebabkan G tidak terhubung. Jadi, cut-set selalu menghasilkan dua buah komponen.

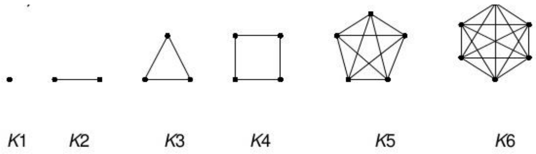


Gambar 2.4 Contoh cut set
Sumber : Rinaldi Munir, *Diktat Kuliah IF2120 Matematika Diskrit*, Teknik Informatika ITB, 2006
Diakses: 9 Desember 2016

12. Graf Berbobot (*Weighted graph*)
Adalah graf yang setiap sisinya diberi sebuah harga. Salah satu kegunaannya adalah menentukkan lintasan minimum dari suatu rute.

2.3. Beberapa Graf Sederhana Khusus
1. Graf Lengkap (*Complete graph*)
Ialah graf sederhana yang setiap simpulnya mempunyai sisi ke semua simpul lainnya. Graf

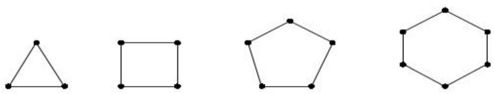
lengkap dengan *n* buah simpul dilambangkan dengan K_n . Jumlah sisi pada Graf lengkap dengan *n* buah simpul adalah n(n-1)/2.



Gambar 2.5 Graf Lengkap bersisi satu sampai 5

Sumber : Rinaldi Munir, *Diktat Kuliah IF2120 Matematika Diskrit*, Teknik Informatika ITB, 2006
Diakses: 8 Desember 2016

2. Graf Lingkaran
   Adalah graf sederhana yang setiap simpulnya berderajat dua.
   Graf lingkaran dengan n simpul dilambangkan dengan C_n.



(a) C3 (b) C4 (c) C5 (d) C6
Gambar 2.6 Graf Lingkaran

Sumber : Rinaldi Munir, *Diktat Kuliah IF2120 Matematika Diskrit*, Teknik Informatika ITB, 2006
Diakses: 8 Desember 2016

3. Graf Teratur (*Regular graph*)
   Adalah graf sederhana yang setiap simpulnya memiliki derajat yang sama. Jumlah sisi pada graf teratur derajat r dengan n buah simpul adalah nr/2.

4. Graf Bipartit (*Bipartite graph*)
   Adalah graf yang himpunan simpulnya dapat dipisah menjadi dua himpunan bagian V_1 dan V_2, sedemikian sehingga setiap sisi pada graf menghubungkan sebuah simpul di V_1 ke sebuah simpul di V_2 dan dinyatakan sebagai $G(V_1,V_2)$.

III. Pembahasan

Lampu kuning digunakan untuk mengumumkan perubahan tanda. Stoffers [1968, 201]
menunjukkan bahwa panjang interval dari state kuning ini ditetapkan bernilai tertentu di beberapa negara. Panjang ini mungkin tergantung pada ukuran atau geometri persimpangan.
Bagaimana kita harus menggabungkan lampu kuning ke model kita? Lampu kuning
pada akhir periode hijau merupakan indikasi bahwa para pengendara harus pergi dari persimpangan.
Kendaraan yang terkena lampu kuning yang berada di atau dekat persimpangan
diizinkan untuk beredar, sementara aliran kendaraan yang tidak berbarengan dengan mereka haruslah masih dalam state merah. Namun, kita dapat mempertimbangkan bahwa aliran memiliki lampu hanya merah atau hijau: Kami menerjemahkan solusi ke dunia nyata dengan mengubah ke kuning yang sesuai interval pada akhir setiap periode hijau. Sebagai akibatnya, penjadwalan seperti
satu digambarkan dalam Gambar 4 untuk persimpangan Gambar 1 akan terlihat berbeda pada
jalan setelah periode kuning ditambahkan. Sebagai contoh, jika kita menggunakan *shading* (arsir) padat
untuk periode kuning, kita mungkin memiliki hasil pada Gambar 5.
Mulai sekarang, model kami mengabaikan lampu kuning, sehingga diagram kami akan
terlihat seperti orang-orang di Angka 3 dan 4.
Waktu yang dibutuhkan kendaraan berhenti untuk memasuki persimpangan-yang
tergantung, antara lain, pada posisi kendaraan di garis

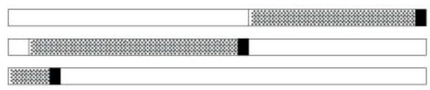


Figure 5. The scheduling of red/green lights from **Figure 4** with yellow periods added (solid shading).

Gambar 3.1 Penjadwalan lampu merah/hijau dari Figure 4 dengan sudah dilakukan penambahan area kuning (diarsir)
Sumber: www.cengage.com/math/book_content/0495011592_giordano/.../traffic_lights.pdf
Diakses: 8 Desember 2016

tunggu dan jumlah kendaraan tiba di persimpangan per unit waktu-adalah im-dalah penting untuk menentukan sebuah panjang hijau minimum untuk setiap aliran. Bahkan,sejak periode kuning (yang dianggap bagian dari periode hijau) yang "aman kesenjangan "dan dengan demikian memiliki panjang aminimum, periode hijau juga memiliki sebuah panjang minimum. Para ahli lalu lintas menentukan batas ini dengan mengambil pengukuran pada persimpangan. Van Vuren dan Van Vliet [1992, 751], misalnya, memberikan 2,5 detik sebagai perkiraan kasar untuk penundaan mulai, yaitu, waktu yang dibutuhkan untuk kendaraan untuk dimulai setelah mobil di depan telah dimulai. Cara lain untuk menjelaskan keterlambatan ini dengan mengukur kecepatan gelombang themotion. Para peneliti memberikan nilai 5 meter/detik untuk kecepatan gelombang bergerak kembali dari garis berhenti.

Jadi model kami akan membutuhkan jumlah minimum waktu tertentu hijau per arus per siklus. Namun, jika kita membolehkan lebih dari satu interval hijau per siklus,kita bisa mendapatkan sebuah tugas dengan setidaknya jumlah minimum hijau tetapi dengan beberapa interval hijau tidak cukup lama untuk mengandung waktu keselamatan kuning. Masalah lain dari beberapa hijau dicatat oleh Stoffers [1968, 205]. Di setiap awal periode hijau, "waktu terbuang" (*lost times*) terjadi, sampai kendaraan berjalan. Jumlah total aliran per hijau maka akan terpengaruh oleh jumlah yang tidak diketahui ini kali memimpin. Oleh karena itu, di sini yang dibicarakan hanya satu hijau interval per siklus per arus (*stream*).

Selanjutnya, beberapa refleksi pada panjang siklus. siklus panjang, memungkinkan untuk lebih lama interval hijau, dapat menampung lebih banyak kendaraan; tapi siklus lebih lama dari diperlukan untuk menangani lalu lintas di tangan mungkin dapat menimbulkan kemacetan yang cukup parah. Van Vuren dan Van Vliet menyarankan bahwa yang terbaik adalah menggunakan panjang siklus praktis terpendek yang diperlukan untuk mengakomodasi volume lalu lintas [1992].

Dari pembahasan di atas, kita memiliki:

• graf yang menunjukkan pasangan arus (*stream*) mana yang kompatibel, dan mana yang tidak

• untuk setiap aliran v,

- Sebuah bilangan positif r_v, yaitu panjang minimal selang hijau untuk aliran itu dan

- Bilangan positif N lain, panjang maksimum siklus (tentu saja N setidaknya sama dengan r_v terbesar.)

"*Maximal Clique*"

Himpunan dari arus yang saling kompatibel berkorespondensi dengan simpul-simpul yang saling bertetangga dalam graf ini. Kami mendefinisikan sebuah konsep yang sangat penting dalam penjelasan berikut:

Definisi. "*Maximal Clique*" dari graf G adalah sebuah himpunan K yang terdiri dari simpul-simpul yang saling bertetangga dalam graf G yang tidak mengandung himpunan lain yang lebih besar dengan sifat ini. Dengan kata lain:

• dua simpul sembarang dari K bertetangga, dan

• simpul yang bukan anggota K (jika ada) tidak bisa bertetangga dengan setiap simpul di K.

Misalnya, graf pada Gambar 2 memiliki dua "*maximal clique*": $K_1 = \{x, z, w\}$

dan $K_2 = \{x, y\}$.

"***Phasing and Intersection Assignments***"

Kami melihat *assignment* S sehingga $Sv \cap Sw = \emptyset$ untuk v dan w apapun adalah simpul nonadjacent yang berbeda, tapi kami mendapat lebih dari itu. Bahkan, solusi optimal kami memenuhi sebaliknya; yaitu, jika dua simpul yang bertetangga, maka interval yang bersesuaian akan saling berpotongan. Untuk memasukkannya ke dalam istilah formal, kita membedakan berikut kondisi:

$v \neq w, \{v, w\}$ not element in $E \Rightarrow Sv \cap Sw = \emptyset$…(1),dan

$\{v, w\} \in E = \Rightarrow Sv \cap Sw \neq \emptyset$……… (2)

Interval assignment yang memenuhi (1) disebut *phasing*s, dan yang memenuhi kedua persamaan (1) dan (2) disebut *intersection assignment*.

Sekarang adalah saat yang tepat untuk rekapitulasi apa yang telah kita lakukan sejauh ini dan untuk mengambil melihat jalan di depan. Kami memiliki amodel dari persimpangan lalu lintas, yang terdiri dari graf dan beberapa data numerik. Masalah yang kita ditetapkan untuk memecahkan adalah cara mendapatkan baik pentahapan atau *intersection assignment* yang memenuhi kondisi model dan juga optimal dalam memberikan waktu hijau maksimum.

Kami menggunakan program linier dengan data dan struktur graf untuk mendapatkan pentahapan atau *intersection assignment*. Namun, sama sekali tidak jelas bahwa kita tidak bisa mendapatkan penjadwalan yang lebih baik dengan menggunakan teknik lain, mungkin tidak terkait dengan pemrograman linear, dan independen dari *maximal clique*.

Pertanyaan lain muncul secara alami dalam konteks ini. Karena kondisi untuk *phasing* kurang ketat daripada yang digunakan untuk untuk *intersection assignment*, dapat dibayangkan bahwa kita bisa berbuat lebih baik dengan *phasings* daripada dengan *intersection assignment*.

Dengan kata lain, mungkin kita bisa menemukan *phasing* dengan total waktu hijau yang lebih besar dari yang *intersection assignment* peroleh.

Algoritma

Kami meringkas ide-ide yang dibahas sejauh ini yang dapat digunakan untuk menemukan *phasing*s, *intersection assignment*, dan parameter yang terkait dari lalu lintas yang diberikan graf. Dalam beberapa kasus, algoritma ini tidak akan memberikan jawaban yang pasti; kami arahkan yang sesuai.

Perhatikan graf lalu lintas G, r, N.

1. Mendapatkan *maximal clique* G dan menentukan apakah mereka mengakui *phasing* berturut-turut. (Untuk contoh kecil yang kita bahas di sini, penentuan ini dapat dilakukan dengan mempertimbangkan kemungkinan sederhana. Jika *maximal clique* mengakui *phasing* berturut-turut (yaitu, G adalah graph interval), pergi ke langkah 2, jika tidak (G bukan graf interval) pergi ke langkah 5.

2. Mengatur L dengan cara berikut.

(A) Variabel yang, katakanlah, d_1,. . . , d_s (satu per klik maksimal.)

(B) Untuk setiap simpul v jumlah ds sesuai dengan semua geng yang berisi v harus $\geq r_v$.

(C) Jumlah semua ds harus $\leq$ N.

(D) Semua ds adalah negatif.

(E) Fungsi Tujuan (dioptimasi) adalah jumlah dari semua *term* $P_i d_i$, di mana p_i adalah jumlah simpul dari *maximal clique* ke i.

Pergi ke 3).

3. Jika L tidak layak, graf lalu lintas tidak mengakui pentahapan atau persimpangan. Keluar. Jika tidak, menghitung solusi optimal d_1,. . . , d_s dan nilai L. Tentukan sebanyak interval sebagai *maximal clique* sebagai berikut.

Buatlah $J_1 = [0, d_1)$, $J_2 = [d_1, d_1 + d_2)$, $J_3 = [d_1 + d_2, d_1 +$

$d_2 + d_3$). . . ; kemudian, untuk setiap simpul v, mengatur S_v sama dengan serikat interval sesuai dengan *maximal clique* yang v miliki yaitu, S_v =untuk semua i: v∈K_i J_i.) Ini mendefinisikan pentahapan yang optimal dari graf lalu lintas, dan *phasing number* nya adalah nilai dari L. Pergi ke 4.

4. Jika d_i> 0 untuk semua i, prosedur yang diuraikan dalam langkah 3 memberikan sebuah *intersection assignment* yang optimal. Jumlah persimpangan, jumlah pentahapan, dan nilai L semua sama. Keluar. Jika beberapa d_i = 0, pecahkan kasus LP berikut (yang dalam kasus ini memiliki solusi yang optimal.)

$$\text{minimize} \sum d_i \text{ subject to}$$
$$\sum_{i:v\in K_i} d_i \geq r_v \text{ for all vertices } v$$
$$d_i \geq 0 \text{ for all } i.$$

Gambar 3.2 Rumus Minimisai d_i

Sumber:
www.cengage.com/math/book_content/0495011592_giordano/.../traffic_lights.pdf

Diakses: 8 Desember 2016

Dalam sekuel ini, kami lihat LP ini sebagai L'. Solusi optimal, jika ada, dilambangkan dengan N_0.

Kami selalu membutuhkan $N_0 \leq N$. Jika N_0 <N, maka graf lalu lintas mengakui sebuah *intersection assignment*, dan nomor pentahapan dan persimpangan adalah sama dengan nilai L. Keluar. Jika N_0 = N, algoritma adalah inkonklusif.

5. Tentukan subgraf H dari G dengan simpul yang sama ditetapkan sebagai G sebagai berikut. Mulailah dengan graf kosong (tanpa sisi) dan terus menambahkan sisi dari G selama subgraf yang dihasilkan graf selang (perhatikan bahwa graf kosong adalah graf interval.) Berhenti jika sudah tidak memungkinkan. Pertimbangkan setiap graf yang dapat diperoleh dengan cara ini, dan menetapkan ke simpul nilai-nilai r_v yang sama dan N sebagai yang dipunyai graf. Pergi ke 6).

6. Untuk setiap subgraf ditemukan menerapkan prosedur yang baru saja dijelaskan, menentukan N_0 sebagaimana didefinisikan dalam 4) di atas.

(A) Jika untuk semua ini terjadi bahwa N_0 = N, algoritma ini inkonklusif.

Keluar.

(B) Jika semua subgrafs memiliki N_0> N, maka G, r, N tidak mengakui *phasing*. Keluar.

(C) Sebaliknya, untuk subgraf dengan N0<N, hitung jumlah perpotongannya sesuai langkah 3 dan 4. Maka, maksimum dari bilangan- bilangan itu adalah *phasing number* dari graf lalu lintas. *Phasing* dari subgraf yang mengandung bilangan ini bisa didapat dari langkah 3. Ini adalah *phasing* dari graf lalu lintas yang sebenarnya yang mengandung *phasing number*. Keluar.

(dikutip dari www.cengage.com/math/book_content/0495011592_giordano/.../traffic_lights.pdf)

IV. KESIMPULAN

Lampu lalu lintas memegang peranan penting dalam mengatur jalannya lalu lintas di jalan raya. Dengan adanya lampu lalu lintas, kecelakaan lalu lintas dapat dihindari. Lampu lalu lintas menggunakan prinsip graf dalam menjadwalkan akan menyalakan warna yang mana, karena ada beberapa lampu lalu lintas di suatu persimpangan, sehingga jeda dan durasi nyala lampu harus diperhatikan agar kendaraan tidak saling bertumbukkan.

V. UCAPAN TERIMA KASIH

Puji dan syukur ke hadirat Tuhan Yang Maha Esa karena hanya dengan berkat dan rahmat-Nya lah penulis dapat menyelesaikan makalah ini dengan baik. Tidak lupa penulis juga berterima kasih kepada Ibu Dra. Harlili, M.Sc dan Bapak Ir. Rinaldi Munir selaku dosen mata kuliah Matematika Diskrit yang telah membimbing penulis dalam menyelesaikan makalah ini. Serta kepada teman-teman yang memberikan koreksi dan motivasi dalam penulisan.

REFERENSI

[1] Munir, Rinaldi. *Diktat Kuliah IF2120 Matematika Diskrit* Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung. 2006.

[2] www.cengage.com/math/book_content/0495011592_giordano/.../traffic_lights.pdf

[3] http://www.slideshare.net/nidashafiyanti/penggunaan-teori-graf-pada-pengaturan-lampu-lalu-lintas

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, 9 Desember 2016

Patrick Nugroho Hadiwinoto, 13515040