

# Aplikasi Shortest Path dengan Menggunakan Graf dalam Kehidupan Sehari-hari

Andika Mediputra – NIM : 13509057

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

andika.mediputra@students.itb.ac.id

**Abstrak** — Makalah ini membahas tentang penggunaan teori dalam struktur diskrit yaitu mencari shortest path (lintasan terpendek) dengan memakai graf dan penerapannya dalam kehidupan sehari-hari. Lintasan terpendek adalah lintasan yang paling minimal jaraknya dengan tujuan tempat yang sama. Ada berbagai cara dan langkah untuk menentukan suatu lintasan terpendek, hasilnya mungkin bervariasi namun bertujuan sama yaitu mencari lintasan yang seefektif mungkin. Jika menggunakan kendaraan maka aplikasi shortest path ini sangat berguna untuk mengefisienkan bahan bakar kendaraan tersebut dan lebih hemat waktu.

**Kata Kunci** — Graf, Jarak, Jalur, Terpendek.

## I. PENDAHULUAN

Dalam kehidupan sehari-hari, masyarakat selalu bergerak ke suatu tempat untuk mencapai tujuannya masing-masing. Di jaman globalisasi ini, seringkali waktu menjadi sangat berharga bagi setiap manusia, sehingga ada perumpamaan “Waktu adalah uang”. Oleh karena itu dalam bepergian ke suatu tempat, tidak sedikit masyarakat mencari dan menggunakan jalur terpendek dan secepat mungkin untuk mencapai tujuan tempat tersebut.

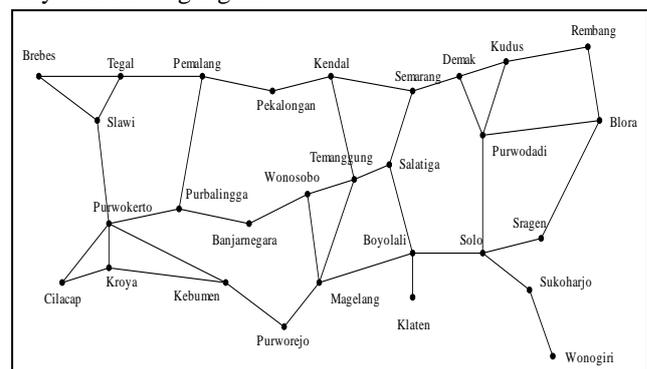
Dalam bepergian terkadang kita membutuhkan peta atau untuk jaman sekarang lebih modern dengan menggunakan GPS (Global Positioning System). Dari gambar peta tersebut bisa dilihat banyak variasi jalur yang bisa digunakan untuk mencapai tempat yang dituju. Bisa dilihat dan dicari manakah jalur terpendek yang bisa digunakan untuk bepergian. Shortest path ini sangat berguna bagi pemilik kendaraan pribadi, contohnya ialah dengan memakai jalur terpendek dibandingkan dengan jalur yang lebih panjang, konsumsi BBM kendaraan tersebut dengan asumsi kecepatan kendaraan tersebut sama kecepatannya antara jalur yang pendek dengan jalur yang panjang, akan terlihat bahwa dengan memakai jalur terpendek konsumsi BBM tersebut akan lebih irit. Waktu yang digunakan selama perjalananpun menjadi lebih singkat, dengan asumsi tidak ada hambatan atau kemacetan di jalur pendek maupun jalur yang lebih panjang. Shortest path itu lebih menguntungkan.

## II. METODE

Dalam makalah ini penulis akan membahas teori struktur diskrit yang berhubungan dengan judul makalah ini yaitu tentang graf. Bagaimana cara melihat dan menggunakan graf dalam kehidupan sehari-hari untuk memakai shortest path ini. Tidak hanya graf saja, namun ada bab lain yang relevan dengan bahasan ini yaitu bab pohon dan sedikit dibahas dengan algoritma.

### 2.1 Graf

Teori graf merupakan pokok bahasan yang sudah tua usianya namun memiliki banyak terapan sampai saat ini. Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antar objek-objek tersebut. Representasi visual dari graf adalah dengan menguyatakan objek dinyatakan sebagai noktah, bulatan, atau titik, sedangkan hubungan antara objek dinyatakan dengan garis. Sesungguhnya peta adalah sebuah graf, yang dalam hal ini kota dinyatakan sebagai bulatan sedangkan jalan dinyatakan sebagai garis.



**Gambar 1.** Peta jaringan jalan raya di Provinsi Jawa Tengah yang merupakan suatu graf

Secara matematis, graf didefinisikan sebagai berikut:

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

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

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

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

### 2.1.1 Jenis-jenis Graf

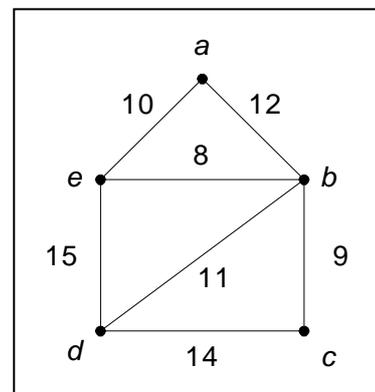
- Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf:
  - a. Graf sederhana  
Graf yang tidak mengandung gelang maupun sisi-ganda dinamakan graf sederhana.
  - b. Graf tak-sederhana  
Graf yang mengandung sisi ganda atau gelang dinamakan graf tak-sederhana (*unsimple graph*).
- Berdasarkan orientasi arah pada sisi:
  - a. Graf tak-berarah  
Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah.
  - b. Graf berarah  
Graf yang setiap sisinya diberikan orientasi arah disebut graf berarah.

### 2.1.2 Terminologi Dasar

- Bertetangga (*Adjacent*)  
Dua buah simpul pada graf tak-berarah  $G$  dikatakan bertetangga bila keduanya terhubung langsung dengan sebuah sisi.
- Bersisian (*Incident*)  
Untuk sembarang sisi  $e = (v_j, v_k)$ , sisi  $e$  dikatakan bersisian dengan simpul  $v_j$  dan simpul  $v_k$ .
- Simpul Terpencil (*Isolated Vertex*)  
Simpul terpencil ialah simpul yang tidak mempunyai sisi yang bersisian dengannya.
- Graf Kosong (*Null graph* atau *Empty Graph*)  
Graf yang himpunan sisinya merupakan himpunan kosong disebut sebagai graf kosong dan ditulis sebagai  $N_n$ , yang dalam hal ini  $n$  adalah jumlah simpul.
- Derajat (*Degree*)  
Derajat suatu simpul pada graf tak-berarah adalah jumlah sisi yang bersisian dengan simpul tersebut. Notasi:  $d(v)$ .
- Lintasan (*Path*)  
Lintasan yang panjangnya  $n$  dari simpul awal  $v_0$  ke simpul tujuan  $v_n$  di dalam graf  $G$  ialah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk  $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$  sedemikian sehingga  $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \dots, e_n = (v_{n-1}, v_n)$  adalah sisi-sisi dari graf  $G$ .
- Siklus (*Cycle*) atau Sirkuit (*Circuit*)  
Lintasan yang berawal dan berakhir pada simpul yang sama disebut sirkuit atau siklus.
- Terhubung (*Connected*)  
Dua buah simpul  $v_1$  dan simpul  $v_2$  disebut terhubung jika terdapat lintasan dari  $v_1$  ke  $v_2$ .  $G$  disebut graf

terhubung (*connected graph*) jika untuk setiap pasang simpul  $v_i$  dan  $v_j$  dalam himpunan  $V$  terdapat lintasan dari  $v_i$  ke  $v_j$ . Jika tidak, maka  $G$  disebut graf tak-terhubung (*disconnected graph*).

- Upagraf Merentang (*Spanning Subgraph*)  
Upagraf  $G_1 = (V_1, E_1)$  dari  $G = (V, E)$  dikatakan upagraf merentang jika  $V_1 = V$  (yaitu  $G_1$  mengandung semua simpul dari  $G$ ).
- 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.
- Graf Berbobot  
Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot). Bobot pada tiap sisi dapat menyatakan jarak antara dua buah kota, biaya perjalanan antara dua buah kota, waktu tempuh pesan (*message*) dari sebuah simpul komunikasi ke simpul komunikasi lain (dalam jaringan komputer), ongkos produksi, dan sebagainya. Terminologi graf berbobot inilah yang menjadi pembahasan dalam makalah ini



Gambar 2. Graf berbobot

### 2.1.3 Beberapa Graf Khusus

- Graf Lengkap  
Graf lengkap ialah graf sederhana yang setiap simpulnya mempunyai sisi ke semua simpul lainnya. Graf lengkap dengan  $n$  buah simpul dilambangkan dengan  $K_n$ .
- Graf Lingkaran  
Graf lingkaran adalah graf sederhana yang setiap simpulnya berderajat dua. Graf lingkaran dengan  $n$  simpul dilambangkan dengan  $C_n$ .
- Graf Teratur  
Graf yang setiap simpulnya mempunyai derajat yang sama disebut graf teratur. Apabila derajat setiap simpul adalah  $r$ , maka graf tersebut disebut sebagai graf teratur derajat  $r$ .

### 2.1.4 Lintasan dan Sirkuit

- a. Lintasan dan Sirkuit Euler:
- Lintasan Euler ialah lintasan yang melalui masing-masing sisi di dalam graf tepat satu kali.
  - Sirkuit Euler ialah sirkuit yang melewati masing-masing sisi tepat satu kali.
  - Graf yang mempunyai sirkuit Euler disebut graf Euler (*Eulerian graph*). Graf yang mempunyai lintasan Euler dinamakan juga graf semi-Euler.
- b. Lintasan dan Sirkuit Hamilton:
- Lintasan Hamilton ialah lintasan yang melalui tiap simpul di dalam graf tepat satu kali.
  - Sirkuit Hamilton ialah sirkuit yang melalui tiap simpul di dalam graf tepat satu kali, kecuali simpul asal (sekaligus simpul akhir) yang dilalui dua kali.
  - Graf yang memiliki sirkuit Hamilton dinamakan graf Hamilton, sedangkan graf yang hanya memiliki lintasan Hamilton disebut graf semi-Hamilton.

### 2.1.5 Beberapa Aplikasi Graf

Terdapat banyak aplikasi yang berkaitan dengan graf. Di dalam aplikasi itu, graf digunakan sebagai alat untuk merepresentasikan atau memodelkan persoalan. Berdasarkan graf yang dibentuk, barulah persoalan tersebut diselesaikan. Ada beberapa aplikasi yang berkaitan dengan lintasan/sirkuit di dalam graf, diantaranya adalah:

- Lintasan terpendek
- Persoalan pedagang keliling
- Persoalan tukang pos Cina
- Pewarnaan graf

## III. APLIKASI SHORTEST PATH DENGAN MENGGUNAKAN GRAF DALAM KEHIDUPAN SEHARI-HARI

Persoalan mencari lintasan terpendek di dalam graf merupakan persoalan salah satu optimasi. Graf yang digunakan dalam pencarian lintasan terpendek ialah graf berbobot (*weighted graph*), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf dapat menyatakan jarak antar kota, waktu pengiriman pesan, ongkos pembangunan, dan sebagainya. Asumsi yang kita gunakan di sini adalah bahwa semua bobot bernilai positif. Kata “terpendek” jangan selalu diartikan secara fisik sebagai jarak minimum, sebab kata “terpendek” berbeda-beda maknanya bergantung pada tipikal persoalan yang diselesaikan. Namun, secara umum “terpendek” berarti meminimasi bobot pada suatu lintasan di dalam graf.

Contoh terapan pencarian lintasan terpendek misalnya simpul pada graf dapat merupakan kota, sedangkan sisi menyatakan jalan yang menghubungkan dua buah kota. Bobot sisi graf dapat menyatakan jarak antara dua buah

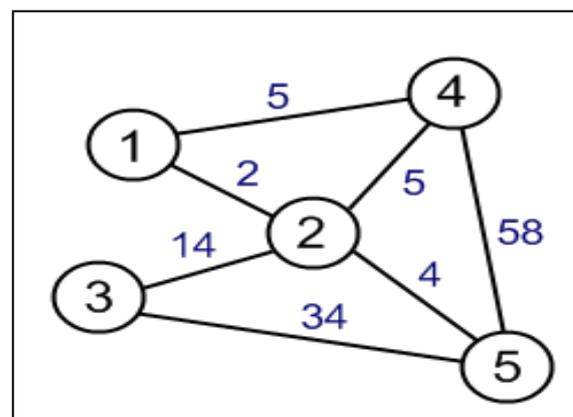
kota atau rata-rata waktu tempuh antara dua buah kota. Apabila terdapat lebih dari satu lintasan dari kota A ke kota B, maka persoalan lintasan terpendek disini adalah menentukan jarak terpendek atau waktu tersingkat dari kota A ke kota B.

Ada beberapa macam persoalan lintasan terpendek, antara lain:

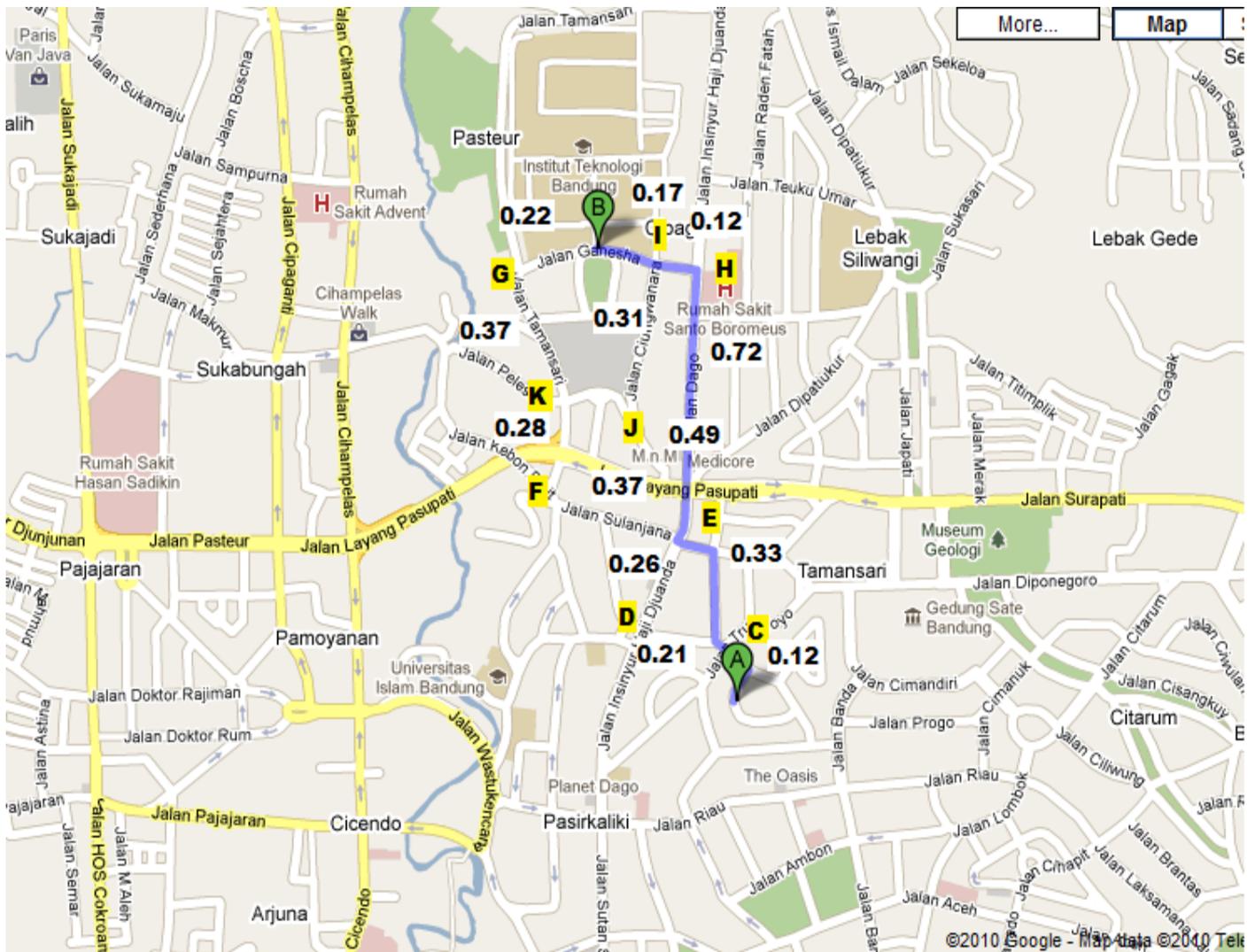
1. Lintasan terpendek antara dua buah simpul tertentu.
2. Lintasan terpendek antara semua pasangan simpul.
3. Lintasan terpendek dari simpul tertentu ke semua simpul yang lain.
4. Lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu.

Sampai saat ini, sudah banyak algoritma mencari lintasan terpendek yang pernah ditulis orang. Algoritma lintasan terpendek yang paling terkenal adalah algoritma Dijkstra. Aslinya, algoritma Dijkstra diterapkan untuk mencari lintasan pada graf berarah. Namun, algoritma ini juga benar untuk graf tak-berarah. Algoritma Dijkstra mencari lintasan terpendek dalam sejumlah langkah. Algoritma ini menggunakan prinsip *greedy*. Prinsip *greedy* pada algoritma Dijkstra menyatakan bahwa pada setiap langkah kita memilih sisi yang berbobot minimum dan memasukkannya ke dalam himpunan solusi.

Dalam mencari lintasan terpendek, bisa juga dengan menggunakan teori pohon di struktur diskrit. Dalam hal pencarian lintasan terpendek salah satu yang relevan adalah dengan algoritma Prim dan algoritma Kruskal. Algoritma Prim membentuk pohon merentang minimum langkah per langkah. Pada setiap langkah kita mengambil sisi dari graf  $G$  yang mempunyai bobot minimum namun terhubung dengan pohon merentang  $T$  yang telah terbentuk. Pada algoritma Kruskal, sisi-sisi graf diurut terlebih dahulu berdasarkan bobotnya dari kecil ke besar. Sisi yang dimasukkan ke dalam himpunan  $T$  adalah sisi graf  $G$  sedemikian sehingga  $T$  adalah pohon. Ada perbedaan diantara kedua algoritma ini namun hasil akhir yang didapat ialah sama, yaitu jarak yang dilalui ialah minimum.



Gambar 3. Contoh lintasan dengan graf berbobot



**Gambar 4. Peta dari rumah penulis ke ITB**

Saatnya menerapkan teori graf tersebut untuk mencari lintasan terpendek dalam kehidupan sehari-hari. Gambar di atas menunjukkan peta sebagian kecil kota Bandung, yang penulis teliti di sini adalah peta dari rumah penulis menuju kampus Institut Teknologi Bandung.

Dari peta tersebut terlihat bahwa banyak jalan yang bisa ditempuh, dalam hal ini posisi rumah penulis ada di A, menuju ke ITB, dalam peta ini dilambangkan dengan kode B. Tentunya dalam menuju ke tempat tujuan selalu ingin dicari manakah jalur yang paling cepat dan pendek untuk ditempuh. Asumsi bahwa tidak ada kemacetan yang terjadi dan dengan menggunakan kecepatan yang sama pada tiap jalur yang dilalui.

Peta di atas bisa dilihat sebagai graf berbobot dimana simpul-simpul dilambangkan dengan kode A, B, C, D, E, F, G, H, I, J, K, dan bobot yang tertulis di peta adalah jarak yang penulis ukur dengan menggunakan bantuan *Google Earth* dan *Google Maps* dalam satuan kilometer (km). Setiap simpul ke simpul yang lainnya sudah diukur jaraknya oleh penulis, tinggal dipergunakan bobot tersebut sesuai keperluan.

Ada banyak variasi jalan yang bisa dipakai untuk menuju tempat tujuan, tentunya setiap jalur mempunyai kelebihan dan kekurangan masing-masing. Semakin

pendek jalur yang digunakan maka akan semakin cepat sampai di tempat tujuan dan lebih menghemat waktu, dan juga jika menggunakan kendaraan pribadi akan semakin menghemat konsumsi bahan bakar yang digunakan. Penjelasan lebih detail akan dipaparkan lewat tabel berikut ini.

**Tabel 1. Jarak diantara 2 simpul**

Simpul	Jarak dalam km
A, C	0,12
C, D	0,21
C, E	0,33
E, F	0,37
E, H	0,72
E, J	0,49
F, K	0,28
K, G	0,37
J, I	0,31
H, I	0,12
I, B	0,17
G, B	0,22
D, E	0,26

**Tabel 2. Perbandingan jarak dari masing-masing rute yang digunakan**

<b>Rute yang digunakan</b>	<b>Jarak total dalam km</b>
A-C-D-E-H-I-B	1,60
A-C-E-H-I-B	1,46
A-C-E-F-K-G-B	1,69
A-C-E-J-I-B	1,56
A-C-D-E-F-K-G-B	1,83

Berdasarkan hasil penelitian terhadap rute yang digunakan penulis dengan total jaraknya, terlihat bahwa rute yang menunjukkan shortest path ialah rute A-C-E-H-I-B dengan total jarak sebesar 1,46 km, tergambar di peta dengan jalur yang berwarna biru. Rute yang lain juga bisa digunakan, namun terlihat bahwa rute lainnya itu lebih panjang dibanding rute terpendek ini. Terbukti dengan menggunakan algoritma, bahwa shortest path itu bisa dicari. Shortest path ini sangat berguna untuk mencapai tujuan dengan cepat karena menggunakan jarak yang lebih pendek dibanding lainnya dan menghemat waktu juga pengeluaran bensin bagi yang memakai kendaraan pribadi, karena waktu sangatlah berarti.

## V. KESIMPULAN

Persoalan mencari lintasan terpendek di dalam graf merupakan persoalan salah satu optimasi. Mencari shortest path (jalur terpendek) untuk mencapai suatu tempat merupakan cara efektif bagi siapa saja agar lebih menghemat waktu, serta biaya dalam mencapai suatu tujuan. Ada berbagai cara untuk mencari jalur terpendek melalui berbagai algoritma, tiap algoritma tersebut memiliki kelebihan dan kekurangannya masing-masing, tinggal bagaimana menggunakannya sesuai dengan kebutuhan pribadi.

## REFERENSI

- [1] <http://maps.google.com/> (tanggal akses 16 Desember 2010 pukul 21.00)
- [2] [http://www.algolist.net/Data\\_structures/Graph](http://www.algolist.net/Data_structures/Graph) (tanggal akses 16 Desember 2010 pukul 19.00)
- [3] Munir, Rinaldi. 2008. *Struktur Diskrit*. Bandung: Penerbit Institut Teknologi Bandung.

## 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, 17 Desember 2010

ttd

Andika Mediputra  
13509057