

Penerapan Algoritma *Sortest Path* pada Aplikasi Google Maps

Ahmad Fajar Prasetyo (13514053)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
ahmad.fajar@students.itb.ac.id

Abstrak—Google Maps merupakan aplikasi yang paling banyak dipakai masyarakat didunia. Google Maps memungkinkan kita untuk mencari jalan dari suatu tempat ke tempat yang lain. Ilmu dasar yang digunakan dalam aplikasi Google Maps adalah Ilmu Graf. Graf yang digunakan adalah graf berarah. Graf juga memiliki bobot(harga) yang merepresentasikan berapa waktu yang dibutuhkan. Selain itu jalan yang dicari berdasarkan waktu tercepat untuk mencapai suatu tempat. Algoritma yang digunakan dalam aplikasi Google Maps adalah algoritma *Sourtest Path* yaitu algoritma yang digunakan untuk mencari jalan dengan bobot paling kecil dalam graf berbobot.

Keywords—Google Maps, *Shortest Path*, Graf, Weight Graf.

I. PENDAHULUAN

Google Maps adalah aplikasi yang berbasis Web yang menyediakan berbagai informasi mengenai keadaan geografi yang ada di dunia ini. Informasi yang disediakan oleh Google Maps antara lain: jalan raya, jalan kereta api, gunung, dan tempat-tempat penting lainnya. Pada beberapa tempat Google Maps juga menyediakan foto tentang keadaan jalan tersebut yang mana foto itu diambil oleh Google dengan berkeliling dunia menggunakan mobil^[1].

Google Maps mengalami perubahan dari tahun-tahun. Setiap tahun fitur dari Google Maps pun bertambah. Sekarang Google Maps juga bisa mencari petunjuk jalan dari suatu tempat ketempat lain. Selain menunjukkan jalan Google Maps juga menghitung jarak yang ditempuh dan juga prediksi waktu tempuh dengan memanfaatkan *Global Positioning System* (GPS) yang dapat menentukan kecepatan kita bergerak.

Selain menambah berbagai fitur Google Maps juga menambah kompatibilitas dengan berbagai perangkat lunak. Sehingga para pengembang perangkat lunak dapat memanfaatkan aplikasi Google Maps pada perangkat lunak yang dibuatnya.

Sehingga banyak sekali aplikasi yang digunakan oleh masyarakat yang basis dari aplikasi itu adalah Google Maps. Contoh aplikasi yang sering digunakan adalah

aplikasi yang mengetahui angkot agar sampai ketujuan, mengetahui posisi teman, mengetahui kecepatan kita bergerak, kemana saja kita pada hari.

Yang menarik disini adalah ketika kita menggunakan aplikasi yang berbasis pada Google Maps, kita bisa mendapatkan jalan terpendek yang bisa ditempuh tanpa harus berfikir jalan mana yang harus kita tempuh. Dari sini dapat dilihat bahwa Google Maps menggunakan Algoritma *sourtest path* yang dapat menentukan jalan mana yang harus ditempuh agar mendapatkan jalan yang terpendek.

II. TEORI GRAF

Graf adalah struktur diskrit yang terdiri dari simpul dan sisi yang menghubungkan setiap simpul. Graf terdiri dari beberapa macam, tergantung sisinya. Banyak sekali masalah yang dapat dimodelkan dengan Graf, contohnya jaringan internet, jalan di kota, struktur kimia. Bentuk graf sering digunakan karena lebih mudah dimengerti dan diolah^[2].

A. Definisi Graf

Sebuah graf $G = (V, E)$ terdiri atas V , yang merupakan himpunan simpul (vertices) tidak kosong, dan E , yang merupakan himpunan sisi (edges). Setiap sisi dalam graf menghubungkan satu atau dua simpul^[2].

B. Jenis-jenis Graf

Berdasarkan banyaknya simpul, graf dapat digolongkan menjadi dua yaitu:

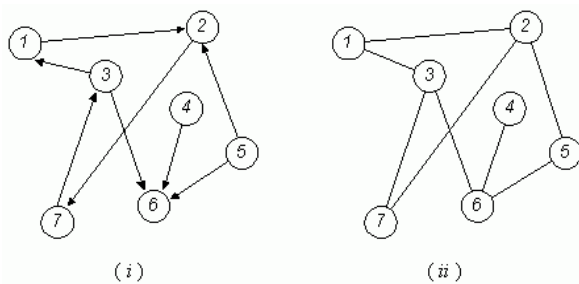
1. Graf berhingga: graf yang memiliki banyak simpul terhingga.
2. Graf tak-berhingga: graf yang memiliki banyak simpul tak terhingga.

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

1. Graf sederhana: graf yang tidak memiliki sisi-gelang maupun sisi-ganda.
2. Graf tak-sederhana: graf yang memiliki sisi ganda atau gelang

Berdasarkan ada tidaknya arah pada sisi graf, maka graf dapat digolongkan menjadi dua yaitu:

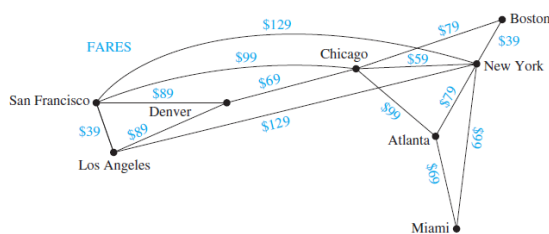
1. Graf tak-berarah: graf tanpa sisi berarah.
2. Graf berarah: graf yang semua sisinya memiliki arah.



Gambar 2.1^[3] (i) merupakan contoh graf berarah (ii) merupakan contoh graf tak-berarah.

C. Graf Berbobot

Graf yang memiliki bobot(harga) pada setiap sisinya. Misalnya pada graf yang simpulnya merepresentasikan kota, maka bobot pada graf tersebut dapat berupa jarak, waktu tempuh, harga perjalanan atau yang lainnya.



Gambar 2.2^[2] Contoh graf berbobot yang simpulnya merupakan kota, sedangkan bobotnya merepresentasikan biaya penerbangan.

III. PEMBAHASAN

A. Google Maps

Google Maps adalah sebuah aplikasi pemetaan yang berbasis *web-service* yang dikembangkan oleh Google. Sebenarnya Google Maps tidak murni ciptaan Google. Pada tahun 2004, dua bersaudara asal Denmark, Lars dan Jens Eilstrup Rasmussen menyampaikan pada Google sebuah ide peta yang tidak statis, tapi bisa digunakan untuk mencari lokasi dan dapat diperbesar^[4]. Setelah itu dibentuklah sebuah tim yang beranggotakan 50 orang yang membuat Google Maps. Pada awalnya awalnya Google Maps dan Google Earth adalah dua aplikasi yang

berbeda tapi karena kegunaan dari kedua aplikasi ini Google memutuskan untuk menggabungkan keduanya.

B. Fitur-Fitur dari Google Maps

1) Earth

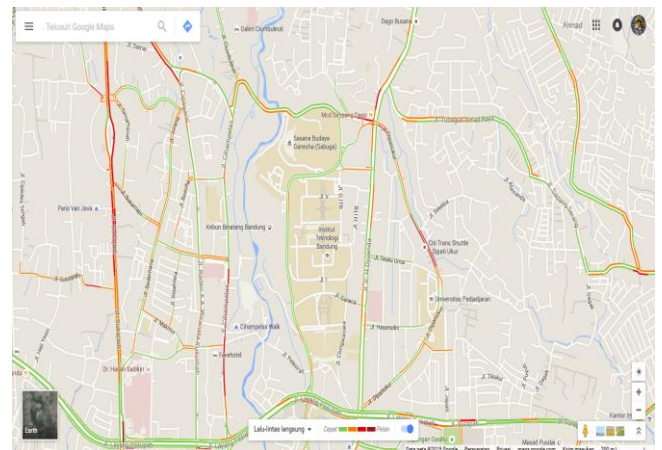
Sebenarnya earth adalah tampilan dari Google Earth. Ini adalah salah satu fitur yang terjadi karena adanya penggabungan antara Google Maps dan Google Earth. Fitur ini memungkinkan kita untuk melihat bentuk bumi dari satelit.



Gambar 3.1 merupakan contoh tampilan dari fitur Earth dalam Google Maps.

2) Lalu lintas

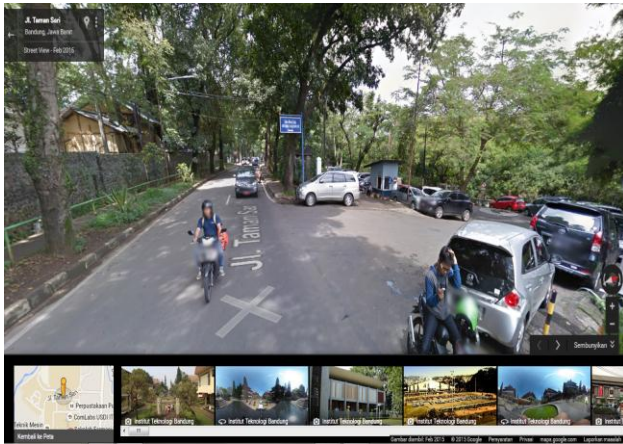
Lalu lintas adalah salah satu fitur yang dimiliki oleh Google Maps. Fitur ini memberitahu kita bagaimana keadaan lalu lintas di jalan. Kita bisa melihat keadaan lalu lintas pada waktu biasanya atau pada waktu sekarang. Warna merah menunjukkan bahwa lalu lintas sangat padat. Sedangkan warna kuning menunjukkan bahwa lalu lintas lumayan padat. Sedangkan warna hijau lalu lintas lumayan sepi. Google memanfaatkan teknologi GPS untuk mengetahui kepadatan lalu lintas. Dengan menghitung banyaknya orang yang ada di jalan. Tetapi fitur ini tidak selalu benar karena tidak semua pengguna jalan menggunakan smartphone sehingga lalu lintas pasti lebih padat dari pada yang di dapat dari fitur Google Maps ini.



Gambar 3.2 merupakan contoh tampilan dari fitur lalu lintas yang ada pada Google Maps

3) Street View

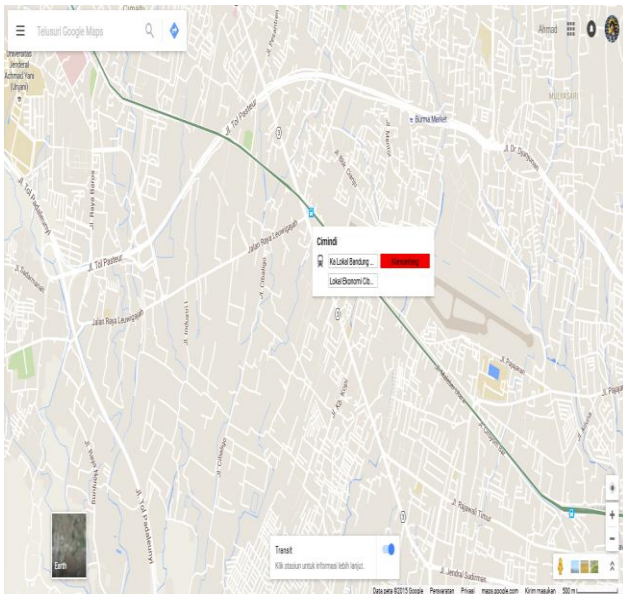
Street View merupakan salah satu fitur unggulan yang ada dalam Google Maps. Fitur ini merupakan fitur yang memungkinkan pengguna Google Maps melihat kondisi jalan yang ada dan dapat di putar sampai 360° . Fitur ini dibangun berdasarkan kontribusi pengguna Google Maps yang rela memfoto jalan dan pihak Google yang berkeliling dunia untuk memfoto jalan.



Gambar 3.3 merupakan contoh tampilan dari fitur Street View yang ada dalam Google Maps

4) Transit

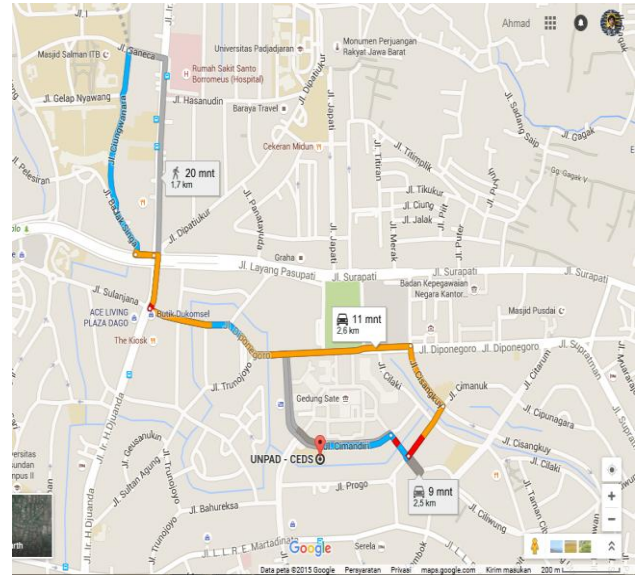
Tidak hanya jalan raya, Google Maps juga membuat graf antara stasiun. Di fitur ini kita dapat mengetahui kereta apa saja yang transit di stasiun tersebut. Sehingga Google Maps mampu membuat rute gabungan antara jalan raya ataupun transportasi yang lainnya.



Gambar 3.4 merupakan contoh tampilan dari fitur Transit yang ada dalam Google Maps

C. Implementasi Graf dalam Google Maps

Saat menggunakan aplikasi Google Maps kita akan melihat bahwa simpul adalah suatu tempat yang dapat kita tujuan. Sedangkan sisinya merupakan jalan menuju suatu tempat tersebut.



Gambar 3.5 merupakan contoh dari aplikasi Google Maps.

Dari sini dapat kita lihat bahwa bobot dari graf yang ada di dalam Google Maps adalah waktu tempuh dan jarak. Karena disana dapat dilihat waktu yang digunakan untuk menempuh dan jarak yang memisahkan kedua tempat atau simpul dalam graf.

Tetapi disini kita melihat bahwa yang dipakai dalam algoritma *Sortest Path* adalah waktu tempuh bukan jarak. Hal itu terjadi karena kita melihat bahwa jarak tempuh minimum dari ITB ke UPAD-CEDS adalah 1,7 KM dengan jalan kaki yang dapat ditempuh dengan waktu 20 menit. Sedangkan jika ditempuh dengan menggunakan mobil jarak tempuhnya menjadi 2,6 KM dengan waktu tempuh sekitar 11 menit. Karena bobot yang digunakan dalam algoritma *Sortest Path* adalah waktu maka hasil yang pertama kali muncul adalah waktu tercepat.

D. Algoritma Sortest Path

Banyak sekali algoritma yang digunakan untuk mencari jalan terpendek antara suatu simpul ke simpul yang lain. Persoalan mencari jalan terpendek dari suatu graf merupakan persoalan yang umum dijumpai dalam graf. Banyak sekali algoritma untuk menyelesaikan persoalan ini.

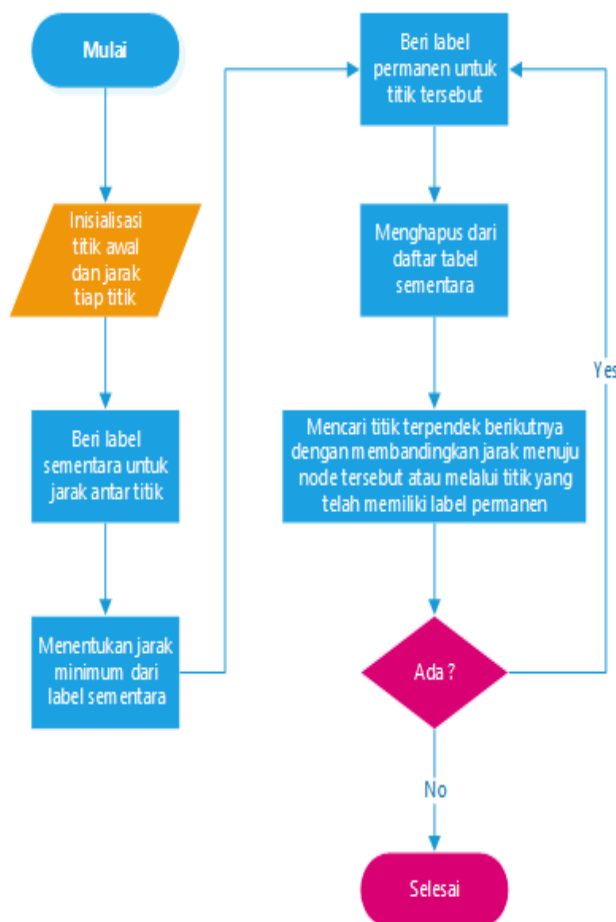
Dalam permasalahan terdapat beberapa ciri-ciri^[6]:

- 1) Merupakan graf yang berarah.

- 2) Bobot tidak selalu menyatakan jarak.
- 3) Tidak mengharuskan semua simpul dapat dicapai.
- 4) Biasanya graf berjenis graf sederhana.
- 5) Solusi tidak selalu unik (pemecahan solusi bisa lebih dari satu solusi).
- 6) Mungkin untuk bobot yang bernilai negatif

Setiap algoritma memiliki kelemahan dan kelebihan masing-masing. Mungkin algoritma ini cocok untuk menyelesaikan masalah ini tapi tidak cocok untuk menyelesaikan masalah yang lain. Terkadang algoritma baru bisa dipakai jika telah memenuhi syarat-syarat tertentu. Maka dari itu salah satu algoritma yang paling bagus dalam menyelesaikan masalah ini adalah Dijkstra. Karena pada algoritma digunakan untuk mencari jalan dengan bobot terpendek dari suatu simpul ke simpul yang lain. Graf yang digunakan adalah graf berarah dan bobot dari suatu graf bernilai positif. Algoritma ini ditemukan oleh matematikawan Belanda bernama Edsger Wybe Dijkstra.

Berikut ini flowcart dan pseudocode dari algoritma Dijkstra:



Gambar 3.6^[5] merupakan flowcart dari Algoritma Dijkstra

```

dijkstra.sh
1 function Dijkstra(Graph, source):
2   for each vertex v in Graph:           // Initializations
3     dist[v] := infinity;               // Unknown distance function from
4                                       // source to v
5     previous[v] := undefined;         // Previous node in optimal path
6   end for                               // from source
7
8   dist[source] := 0;                   // Distance from source to source
9   Q := the set of all nodes in Graph; // All nodes in the graph are
10                                       // unoptimized - thus are in Q
11   while Q is not empty:               // The main loop
12     u := vertex in Q with smallest distance in dist[]; // Start node in first case
13     remove u from Q;
14     if dist[u] = infinity:
15       break;                          // all remaining vertices are
16     end if                              // inaccessible from source
17
18     for each neighbor v of u:         // where v has not yet been
19                                       // removed from Q.
20       alt := dist[u] + dist_between(u, v);
21       if alt < dist[v]:                // Relax (u,v,a)
22         dist[v] := alt;
23         previous[v] := u;
24         decrease-key v in Q;          // Reorder v in the Queue
25       end if
26     end for
27   end while
28   return dist;
  
```

Gambar 3.7^[5] merupakan pseudocode dari Algoritma Dijkstra

Disini penulis akan menjelaskan secara detail cara kerja Algoritma Dijkstra. Algoritma Dijkstra akan menghasilkan sebuah pohon merentang yang mengandung rute minimum dari simpul asal ke seluruh simpul lainnya.

Berikut ini adalah tahapan dari Algoritma Dijkstra:

a) Inisialisasi titik awal dan jarak tempuh. Pada tahap ini kita harus memilih titik awal. Dan kita harus menentukan bobot setiap sisi yang ada pada graf.

b) Beri label sementara untuk jarak antar titik. Untuk label titik sementara kita bisa menggunakan sebuah himpunan(dalam makalah ini kami menggunakan priority queue) yang mana anggotanya adalah titik yang diberi label sementara. Kita disini memberi label yang merepresentasikan bahwa simpul ini belum ditentukan jarak terpendeknya. Dalam makalah ini penulis mengambil infinity sebagai jarak sementara karena jika nanti dibandingkan dengan jarak yang sebenarnya pasti lebih besar, karena lebih besar pasti ada penggantian jarak.

c) Menentukan jarak minimum dari label sementara. Dari titik start kita memberi label ke titik yang masih

berada dalam himpunan dan terhubung dengan simpul start. Disetiap titik itu ada jarak dari simpul start. Karena ini pertama jadi semua jaraknya adalah infinity sehingga semua jarak akan diganti dengan jarak ke simpul start.

d) Beri label permanen untuk titik tersebut. Maksud dari beri label permanen disini adalah menghapus dari himpunan karena himpunan hanya berisi simpul yang diberi label sementara. Dengan menghapus simpul dari himpunan sudah tidak akan berganti lagi jarak antara simpul tersebut dengan simpul sumber.

e) Mencari titik terpendek berikutnya dengan membandingkan jarak menuju node tersebut atau melalui titik yang telah memiliki label permanen. Disini kita juga bisa mencari jalan dari simpul yang telah ditandai permanen.

f) Mengecek apakah dalam simpul yang terahir yang terhubung masih ada tidak di dalam himpunan. Jika ada maka dicari yang paling kecil nilainya, jika tidak ada berarti kita sudah mendapatkan yang kita cari.

Algoritma ini terbukti benar karena kita hanya mengganti jika kita mendapatkan jalan dengan bobot yang lebih kecil. Sehingga tidak ada bobot yang terahir didapat adalah bobot yang paling kecil.

Cara implementasi himpunannya dengan menggunakan priority queue didapat kompleksitas algoritma Dijkstra ini adalah $O(E \log V)$ dengan E sebagai simpul sedangkan V sebagai sisi^[6]. Priority Queue adalah sebuah urutan yang berdasarkan prioritas. Dalam algoritma Dijkstra yang di prioritaskan adalah simpul yang memiliki jarak yang paling pendek dari sumber.

IV. KESIMPULAN

Ilmu Graf dapat menyelesaikan berbagai masalah yang ada didunia ini. Salah satu contoh masalah yang dapat diselesaikan oleh ilmu graf adalah mencari jarak terpendek.

Cara mencari jarak terpendek dalam suatu graf yang berbobot dengan menggunakan algoritma shortest path. Salah satu algoritma Shortest Path yang paling terkenal adalah algoritma Dijkstra.

Algoritma Dijkstra memiliki beberapa syarat agar algoritma ini dapat berjalan dengan baik dan benar. Syaratnya graf merupakan graf yang memiliki bobot positif. Graf juga harus memiliki arah.

Google Maps menerapkan algoritma shortest path dan konsep graf untuk mencari rute tercepat. Graf pada Google Maps adalah graf berbobot yang bobotnya merepresentasikan jarak dan waktu tempuh.

Google Maps memiliki banyak fitur yang disediakan agar pengguna Google Maps dapat mencari rute tak

terbatas dari jalan raya saja, tetapi juga memungkinkan dari rute pesawat maupun rute kereta api.

V. UCAPAN TERIMA KASIH

Pertama penulis ingin mengucapkan terima kasih kepada Tuhan yang Maha Esa atas hikmat dan waktu yang telah diberikan kepada penulis agar dapat menyelesaikan makalah ini. Tak lupa penulis juga mengucapkan terima kasih kepada kedua orang tua penulis karena tanpa jasa dan bimbingannya penulis tidak dapat menuntut ilmu di Institut Teknologi Bandung dan menyelesaikan makalah ini. Penulis juga ingin mengucapkan terima kasih kepada Bapak Rinaldi Munir dan Ibu Harlili karena melalui pengajarannya, saya dapat mengerti konsep Matematika Diskrit dan teori graf yang menjadi dasar makalah ini.

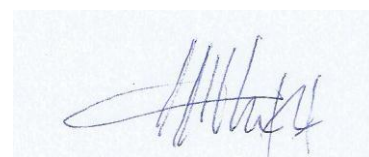
REFERENSI

- [1] <http://whatis.techtarget.com/definition/Google-Maps>, diakses pada tanggal 5 Desember 2015 pukul 19:42
- [2] K.H. Rosen, Discrete Mathematics and Its Applications 7th Edition. New York: McGraw-Hill, 2012 .
- [3] <http://aimyaya.com/images/contohgraf.gif>, diakses pada tanggal 9 Desember 2015 pukul 12:39
- [4] http://imma.mobi/index.php?option=com_content&view=article&id=379:sejarah-google-maps&catid=7:news&Itemid=46, diakses pada tanggal 9 Desember 2015 pukul 12:59
- [5] <https://wirasetiawan29.wordpress.com/2015/04/02/tentang-algoritma-dijkstra/>, diakses pada tanggal 9 Desember 2015 pukul 13:23
- [6] Sedgewick, Robert dan Kevin Wane. Algorithm 4th Edition. Boston: Pearson Education Inc, 2011 .

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 2015



Ahmad Fajar Prasetyo (13514053)