

Aplikasi Teori Graf dalam Pencarian Jalur Terpendek di *Global Positioning System*

Sekar Anglila Hapsari / 13514069
Program Studi Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514069@std.stei.itb.ac.id

Abstrak — *Global Positioning System* telah menjadi sistem navigasi manusia selama bertahun-tahun. Pertama kali dikemukakan pada tahun 60-an, GPS kini telah dapat diakses oleh semua orang yang mempunyai teknologi dengan GPS Receiver. Dalam menavigasi, GPS memperhitungkan jalur terpendek yang bisa dilewati pengguna dari suatu tempat ke tempat lainnya. Untuk melakukan hal itu, *Global Positioning System* mengaplikasikan teori graf. Dalam makalah ini, akan dijelaskan pengaplikasian teori graf dalam pencarian jalur terpendek dalam *Global Positioning System*.

Kata Kunci — *Global Positioning System*, Lintasan Terpendek, Graf Berbot, Matriks Ketetangaan

I. PENDAHULUAN

Peta merupakan benda esensial yang pasti dibutuhkan semua orang untuk menunjukkan arah. Zaman dahulu, manusia harus melihat peta besar atau buku peta tebal untuk mencari jalur dari suatu daerah ke daerah lain. Namun hal tersebut tidak mudah dan sangat tidak efisien. Tapi, kini telah ada *Global Positioning System*, atau lebih sering disebut GPS, yang merupakan inovasi dari peta sederhana.

Global Positioning System merupakan sistem navigasi yang telah dikembangkan oleh manusia. *Global Positioning System* merupakan sistem yang menggunakan satelit untuk menampilkan peta daerah manapun dan kapanpun untuk siapapun yang mempunyai *GPS receiver*. Walaupun ketika pertama kali dibuat hanya untuk digunakan pemerintah Amerika Serikat, di abad ke-21 ini semua teknologi yang dimiliki masyarakat umum (contoh: telepon seluler), telah dilengkapi *GPS Receiver*. Sehingga, GPS dapat diakses hampir semua orang di dunia ini.



Gambar 1 : Contoh Aplikasi GPS di telepon seluler, Google Maps ^[1]

Global Positioning System sangat bermanfaat, selain sebagai media untuk membantu observasi, atau mendeteksi lokasi, salah satu manfaat utama yang secara langsung dirasakan masyarakat adalah membantu untuk menavigasi jalan tercepat dari suatu tempat ke tempat lainnya.

Untuk menavigasi jalan, *Global Positioning System* menggunakan teori graf untuk membantu memperhitungkan jalan tercepat menuju suatu tempat, serta alternatifnya. *GPS* akan memperhitungkan jarak dan aspek-aspek lainnya dari setiap alternatif jalan hingga menemukan jalan tercepat untuk membantu pengguna. Perhitungan yang dilakukan dengan pengaplikasian teori graf yang akan dijelaskan dalam makalah ini.

II. DASAR TEORI

2.1. Definisi Graf

Graf merupakan struktur diskrit yang merepresentasikan hubungan antar objek-objek. Graf secara matematis didefinisikan sebagai pasangan himpunan. Graf dapat ditulis dengan notasi $G = (V, E)$. Dari notasi tersebut, dapat dilihat bahwa graf terdiri dari

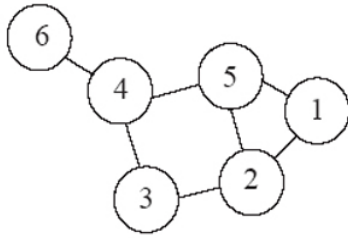
- V : Himpunan tidak kosong dari simpul – simpul (*Vertices* atau *Nodes*).

$$V = \{v_1, v_2, v_3, \dots, v_n\}$$

- E : Himpunan sisi (*Edges* atau *Arcs*) yang menghubungkan sepasang simpul.

$$E = \{e_1, e_2, e_3, \dots, e_n\}$$

(Rinaldi Munir, 2006 : VIII – 2).



$$V = \{1,2,3,4,5,6\}$$

$$E = \{\{1,2\},\{1,5\},\{2,3\},\{2,5\},\{3,4\},\{4,5\},\{4,6\}\}$$

Gambar 2 : Contoh graf dan himpunan simpul dan sisinya^[3]

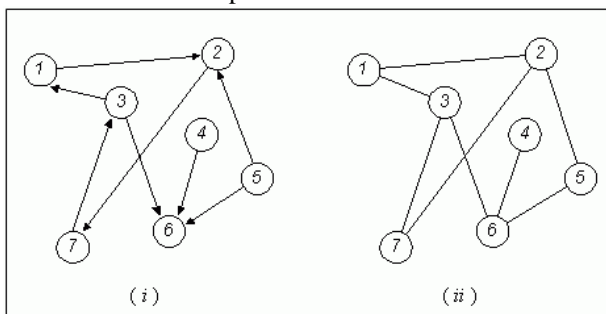
Dapat dilihat dari definisi diatas bahwa sebuah graf tidak boleh mempunyai himpunan simpul (V) yang kosong, namun boleh mempunyai himpunan sisi (E) yang kosong. Sebuah graf dengan himpunan sisi yang kosong dan satu simpul disebut Graf Trivial.

Sisi yang berbeda namu menghubungkan pasangan simpul yang sama disebut sisi ganda (*multiple edges* atau *parallel edges*). Sedangkan simpul yang menghubungkan simpul yang sama (membentuk sebuah lingkaran karena kembali ke simpul asal) disebut gelang atau kalang (loop).

2.2. Jenis Graf

Ada beberapa jenis graf yang dikelompokkan berdasarkan keberadaan sisi ganda atau sisi gelang, berdasarkan jumlah simpul, dan berdasarkan orientasi arah sisi. Berikut adalah penjabarannya:

- a) Berdasarkan Sisi Ganda dan Sisi Gelang
 1. Graf sederhana : Tidak mengandung sisi ganda atau sisi gelang.
 2. Graf tak – sederhana : Mengandung sisi ganda atau sisi gelang.
- b) Berdasarkan Jumlah Simpul
 1. Graf berhingga : Mengandung jumlah simpul yang berhingga.
 2. Graf tak – berhingga : Mengandung jumlah simpul yang tak dapat dihitung.
- c) Berdasarkan Orientasi Arah Sisi
 1. Graf tak – berarah : Mengandung sisi yang tidak mempunyai orientasi arah.
 2. Graf berarah : Mengandung sisi yang mempunyai orientasi arah. Setiap sisi mempunyai simpul asal dan simpul terminal.



Gambar 3: Graf berarah (i) dan graf tak – berarah (ii) ^[4]

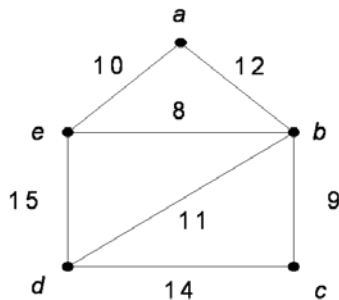
Dapat dilihat dari ketiga kelompok jenis graf, kelompok graf yang berdasarkan orientasi arah sisi adalah graf yang digunakan dalam *Global Positioning System*.

2.3. Terminologi Dasar Graf

Tentu saja graf mempunyai terminologi atau istilah dasar yang akan sering terpakai. Berikut adalah penjabarannya.

- a) Bertetangga
Sepasang simpul dikatakan bertetangga atau *Adjacent* ketika kedua simpul terhubung langsung dengan sisi yang sama.
- b) Bersisian
Sebuah sisi bersisian atau *Incident* dengan simpul-simpul yang sisi tersebut hubungkan langsung.
- c) Simpul Terpencil
Simpul terpencil adalah simpul yang tidak mempunyai tetangga.
- d) Graf Kosong
Graf kosong adalah graf yang himpunan sisinya kosong dengan jumlah simpul sebanyak n .
- e) Derajat
Derajat atau *degree* sebuah simpul pada graf tidak berarah adalah jumlah sisi yang bersisian dengan simpul tersebut. Sedangkan untuk graf berarah, derajat simpul terbagi menjadi dua,
 1. Derajat Masuk ($d_{in}(v)$) : jumlah sisi atau busur yang masuk ke dalam simpul v .
 2. Derajat Keluar ($d_{out}(v)$) : jumlah sisi atau busur yang keluar dari simpul v .
 Derajat total simpul tersebut adalah derajat masuk ditambah derajat keluarnya.
- f) Lintasan
Lintasan atau *path* adalah panjangnya jarak dari simpul awal ke simpul tujuan dalam graf G .
- g) Siklus atau Sirkuit
Siklus adalah lintasan yang berawal dan berakhir pada simpul yang sama.
- h) Terhubung
Sebuah graf disebut terhubung jika setiap pasang sisi v_i dan v_j mempunyai lintasan yang menuju satu sama lain (terdapat lintasan dari v_i ke v_j dan sebaliknya).
- i) Upagraf dan Komplemen Upagraf
Upagraf atau *subgraph* G_1 adalah graf yang seluruh anggota simpul dan sisinya juga merupakan anggota dari himpunan simpul dan sisi graf G (graf utama). Sedangkan Komplemen Upagraf adalah graf dengan anggota himpunan simpul dan sisinya terdiri dari sisi dan simpul yang ada dalam graf G namun tidak dalam graf G_1 (upagraf G).
- j) Upagraf Merentang
Upagraf Merentang adalah upagraf yang elemen himpunan simpulnya terdiri dari seluruh elemen himpunan simpul graf G (walaupun elemen himpunan sisinya tidak).

- k) *Cut-set*
Cut-set merupakan himpunan sisi dari graf terhubung G yang jika dihilangkan akan menghasilkan dua komponen terhubung.
- l) Graf Berbobot
 Graf berbobot adalah graf yang sisinya memiliki sebuah bobot tertentu.



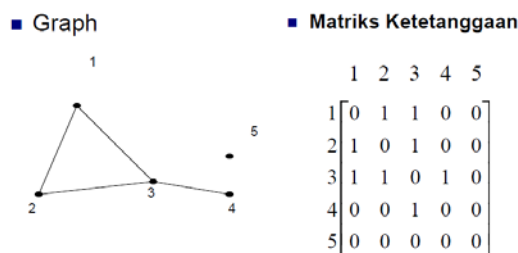
Gambar 4 : Graf Berbobot ^[5]

2.4. Representasi Graf

Terdapat beberapa representasi graf yang dipakai untuk mempermudah pemrosesan graf dalam program komputer. Berikut adalah penjabarannya.

- a) Matriks Ketetanggaan
 Matriks Ketetanggaan atau *Adjacency Matrix* adalah matriks yang merepresentasikan graf dengan ketetanggaan sisi-sisinya. Matriks Ketetanggaan merupakan matriks yang berisi angka 0 (*false*) dan 1 (*true*).
 A_{ij} berisi 1 jika simpul i dan j bertetangga, 0 jika tidak.

Matriks Ketetanggaan (*adjacency matrix*)



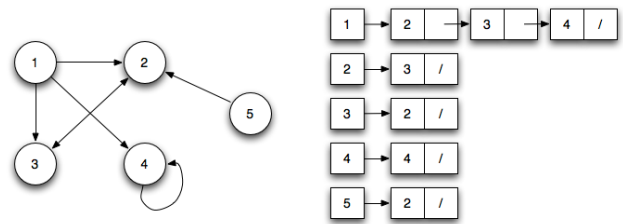
Gambar 5 : Matriks Ketetanggaan ^[6]

Untuk graf berbobot, matriks ketetanggaan berisi bobot dari sisi yang menghubungkan simpul-simpul yang bertetangga. Jika tidak bertetangga, akan diisi nilai tak berhingga(∞).

- b) Matriks Bersisian
 Matriks bersisian merupakan representasi graf dengan menyatakan *incidency* antara sisi dan simpul dalam graf tersebut. Seperti matriks ketetanggaan, matriks bersisian diisi dengan nilai 0 (*false*) dan 1 (*true*).
 A_{ij} berisi 1 jika simpul i dan sisi j bersisian, 0 jika

tidak.

- c) Senarai Ketetanggaan
 Senarai Ketetanggaan merepresentasikan graf seperti matriks ketetanggaan, namun dalam bentuk daftar. Daftar tersebut hanya diisi dengan sisi yang bertetangga dengan daftar simpul tersebut.



Gambar 6 : Senarai Ketetanggaan ^[7]

2.5. Global Positioning System (GPS)

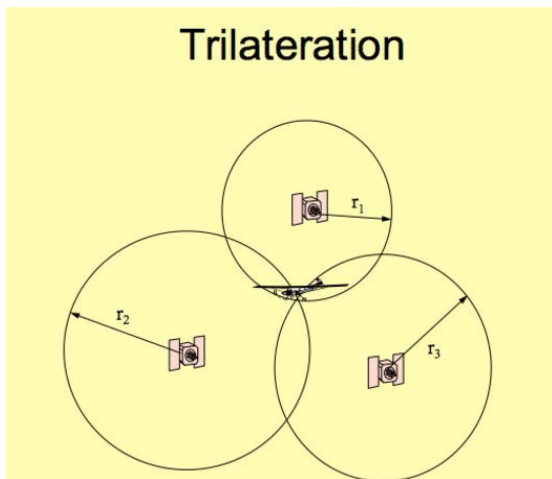
Seperti yang telah dijelaskan secara singkat dalam pendahuluan, *Global Positioning System* merupakan sistem navigasi yang telah dikembangkan oleh manusia (pertama dikembangkan di Amerika Serikat). Konsep yang pertama kali dikemukakan pada tahun 60-an akhirnya terealisasi pada Februari 1978 saat satelit pertamanya diluncurkan. Sejak itu, teknologi GPS terus berkembang hingga dapat digunakan seperti sekarang ini.

Global Positioning System bisa dibagi menjadi tiga bagian: luar angkasa, kontrol, dan pengguna. Berikut adalah penjabarannya.

- a) Luar Angkasa
 Terdapat sekitar 30 satelit di luar angkasa yang mengorbit Bumi dengan ketinggian sekitar 20.000 km. Setiap satelit dilengkapi dengan komputer, radio, dan jam atomik. Satelit akan secara terus-menerus mengirimkan posisi dan waktunya ke bumi. Satelit mengorbit bumi dua kali dalam sehari dan menyamakan waktunya dengan waktu di daratan sekali sehari.
- b) Kontrol
 Kontrol akan terus memonitor keberadaan dan performa satelit dan memastikan bahwa satelit bekerja dengan baik, memonitor transmisi dari satelit, dan mengirimkan perintah.
- c) Pengguna
 Terdiri dari *GPS Receivers* yang telah dimiliki sebagian besar orang di dunia ini. *GPS Receiver* akan memroses transmisi yang diterimanya menjadi informasi yang dibutuhkan oleh pengguna.
Global Positioning System bekerja dengan cara memperhitungkan rentang waktu yang diperlukan *GPS Receiver* untuk menerima sinyal dari satelit. Pertama, sebuah *GPS Receiver* harus dapat menerima sinyal dari tiga satelit berbeda. Ketiga satelit itu mengirimkan posisi dan waktunya ke *GPS Receiver*. *GPS Receiver* lalu akan memperhitungkan rentang waktu yang dibutuhkan sinyal untuk sampai. Informasi tersebut menghasilkan jarak *GPS Receiver* dari ketiga satelit yang terdeteksi oleh *GPS Receiver*. Dari seluruh informasi yang telah diproses, akan

dihasilkan koordinat bujur dan lintang (2-D *Trilateration*) *GPS Receiver* di bumi.

Lalu, akan dilakukan proses 3-D *Trilateration* yang akan menghasilkan posisi akurat *GPS Receiver* di bumi, yaitu koordinat bujur, lintang, dan ketinggiannya. Karena jam atomik di satelit sangat akurat, sedangkan jam yang berada di *GPS Receiver* tidak, dibutuhkan empat satelit untuk terdeteksi oleh *GPS Receiver*. Satelit keempat akan mengirimkan waktu penerimaan yang lebih akurat sehingga menghasilkan posisi yang lebih akurat. Ketika *GPS Receiver* sudah dapat menentukan posisinya, *GPS Receiver* baru akan bisa memroses informasi lainnya seperti keadaan jalan di sekitar, jarak, dan lainnya.



Gambar 7 : Proses Trilatertasi^[8]

GPS sekarang sudah sangat akurat karena pemrosesan sinyal oleh *GPS Receiver* dikelola dengan sangat baik. Namun proses diatas hanya memperhitungkan posisi *GPS Receiver* di bumi, belum memperhitungkan cara untuk mencapai suatu tempat dengan jalur tercepat. Disinilah teori graf akan dipakai. Aplikasi teori graf dalam GPS akan dijelaskan pada bab selanjutnya.

III. PEMBAHASAN

3.1. Lintasan Terpendek

Dalam GPS, pengguna akan ditunjukkan posisinya dalam peta, lalu pengguna dapat menentukan tempat tujuannya. GPS lalu akan menunjukkan jalur terpendek yang dapat dilewati pengguna, juga alternatifnya. Jalur yang direkomendasikan oleh GPS adalah lintasan dengan tingkat kemacetan paling rendah dan waktu yang dibutuhkan untuk melewatinya paling cepat. Untuk mendapatkan jalur ini, GPS menggunakan teori graf untuk implementasinya.

Lintasan terpendek atau *Shortest Path* merupakan salah satu implementasi teori graf yang paling umum. "Tempat" dalam *Global Positioning System* merupakan simpul-simpul dari graf, dan jalur dalam peta merupakan sisi-sisi graf yang berbobot. Maka, pencarian lintasan terpendek merupakan aplikasi dari graf berbobot.

Terdapat tiga variasi pencarian lintasan terpendek, yaitu:

- Pencarian lintasan terpendek dari satu sumber (*Single source shortest path problem*). Harus dilakukan pencarian lintasan terpendek dari simpul sumber tersebut ke simpul-simpul lainnya dalam graf tersebut.
- Pencarian lintasan terpendek ke satu tujuan (*single destination shortest path problem*). Kebalikan dari poin a, harus dilakukan pencarian lintasan terpendek ke suatu simpul tujuan tertentu dari simpul-simpul lainnya dalam graf.
- Pencarian lintasan terpendek dari setiap pasangan simpul (*all pairs shortest path problem*). Harus dilakukan pencarian lintasan terpendek dari setiap pasangan simpul dalam graf tersebut (v dengan v' , v dengan v'' , v' dengan v'' , dst).

Terdapat beberapa algoritma pencarian lintasan terpendek. Beberapa diantaranya meliputi Algoritma Dijkstra untuk mencari lintasan terpendek dari satu sumber dan Algoritma Floyd-Warshall untuk mencari lintasan terpendek dari semua pasangan simpul. Kedua algoritma tersebut akan dijelaskan dalam makalah ini.

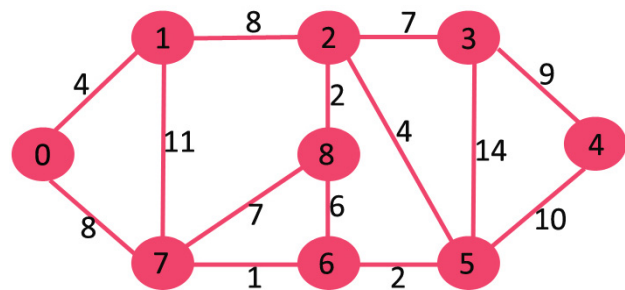
3.2. Algoritma Dijkstra

Algoritma Dijkstra dinamakan berdasarkan penemunya, Edger Wybe Dijkstra. Algoritma Dijkstra diterapkan untuk mencari lintasan terpendek dari satu sumber dalam graf berarah (walaupun pada graf tidak berarahpun algoritma ini benar) berbobot tidak negatif. Algoritma Dijkstra memakai konsep *greedy algorithm*. Pada setiap langkah yang dilakukan, Algoritma Dijkstra akan memilih suatu sisi dengan bobot minimum dan memasukkannya dalam himpunan solusi.

Bobot dari semua fungsi akan dihitung dengan rumus

$$W = E \rightarrow [0, \infty)$$

Dimana W adalah *Weight* atau bobot dan E adalah *Edges* atau sisi. Algoritma sangat mirip dengan Algoritma Prim pada pohon merentang minimum. Matriks ketetanggan akan sangat berguna dalam algoritma ini.



Gambar 8 : Contoh Graf Berbobot^[9]

Algoritma Dijkstra akan dari sumbernya dan menyatakan bobot awal bernilai 0 ($W = 0$). Lalu algoritma tersebut akan menghitung nilai bobot sementara dari semua simpul yang bertetangga (yang belum dikunjungi) dengan simpul aktif (simpul yang sudah dikunjungi, untuk sekarang hanya simpul awal). Misalnya pada gambar 8,

simpul sumber adalah 0 dan simpul yang bertetangga adalah 1 (bobot = 4) dan 7 (bobot = 8). Maka bobot 0 ke 1 adalah $0 + 4 = 4$; dan 0 ke 7 adalah $0 + 8 = 8$. Dapat dilihat bahwa $4 < 8$ maka akan dipilih lintasan ke 1 dengan lintasan berbobot 4.

Proses tersebut terus dilakukan kepada simpul-simpul lainnya. Diperhitungkan bobot sementara dari setiap simpul dari semua simpul aktif yang bertetangga. Misalnya, dari graf yang sama seperti contoh diatas, simpul 1 dan 7 bertetangga dengan sisi berbobot 11. Maka bobot 1 ke 7 adalah $4 + 11 = 15$; dan 0 ke 7 adalah $0 + 8 = 8$. Maka dipilih lintasan 0 ke 7. Dilakukan langkah ini hingga mencapai simpul tujuan atau setiap simpul dalam graf. Maka dari Algoritma Djikstra akan tercapai sebuah lintasan minimum dari sumber tertentu ke simpul tujuan.

```

1 function Dijkstra(Graph, source):
2
3   create vertex set Q
4
5   for each vertex v in Graph:
6     dist[v] ← INFINITY
7     prev[v] ← UNDEFINED
8     add v to Q
9
10  dist[source] ← 0
11
12  while Q is not empty:
13    u ← vertex in Q with min dist[u]
14    remove u from Q
15
16    for each neighbor v of u:
17      alt ← dist[u] + length(u, v)
18      if alt < dist[v]:
19        dist[v] ← alt
20        prev[v] ← u
21
22  return dist[], prev[]

```

Gambar 9 : Pseudocode Algoritma Djikstra^[10]

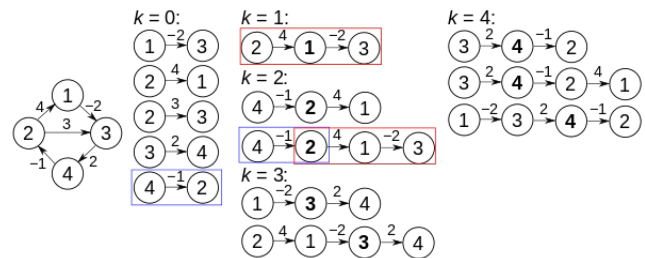
3.2. Algoritma Floyd-Warshall

Algoritma Floyd-Warshall merupakan algoritma pencarian lintasan terpendek dari semua pasangan simpul dalam graf berbobot. Bobot boleh positif ataupun negatif, namun tidak boleh ada siklus dengan bobot negatif. Algoritma ini merupakan *dynamic programming*. Algoritma ini berjalan dengan waktu $\Theta = (|V|^3)$.

Berbeda dengan Algoritma Djikstra, Algoritma Floyd-Warshall memeriksa lintasan terpendek dari semua pasangan simpul v_i ke v_j dan memeriksa apakah jika melewati simpul lain (lintasan menjadi $v_i - v_k - v_j$) akan memiliki bobot yang lebih kecil. Seperti pada Algoritma Djikstra, matriks ketetangaan akan sangat berguna dalam algoritma ini.

Algoritma ini memperbolehkan lintasan untuk melewati simpul-simpul lain antara v_i dan v_j asalkan menghasilkan lintasan dengan bobot minimum. Algoritma ini sangat berguna dalam GPS. Tempat awal adalah v_i dan tempat tujuan adalah v_j . Lalu diperhitungkan jalur minimum

antara pasangan tempat tersebut sehingga ditemukan jalur minimum yang dapat digunakan pengguna dan beberapa alternatif jalur lainnya.



Gambar 10 : Aplikasi Algoritma Floyd-Warshall^[11]

Walaupun algoritma-algoritma yang telah dibahas sudah sangat efektif, GPS masih mempunyai beberapa kekurangan. Contohnya, jika pengemudi berganti arah yang tidak sesuai dengan lintasan terpendek yang sudah dihitung, GPS harus menghitung ulang semuanya sehingga dibutuhkan beberapa waktu menunggu pemrosesan selesai. Namun, seiring berjalannya waktu hal ini pasti akan dapat terselesaikan.

IV. KESIMPULAN

Global Positioning System merupakan suatu inovasi luar biasa yang telah membantu penggunanya untuk menavigasi dengan baik dan efektif. Untuk mencari jalur terpendek dari suatu tempat ke tempat lainnya, GPS menggunakan teori graf dalam pengaplikasiannya. Terdapat dua algoritma yang dipakai untuk menyelesaikan masalah tersebut, yaitu :

1. Algoritma Djikstra :
Sebuah algoritma berprinsip *greedy algorithm* yang mencari jalur terpendek dari satu sumber tertentu ke simpul lainnya dalam graf.
2. Algoritma Floyd – Warshall :
Sebuah algoritma *dynamic programming* yang mencari jalur terpendek dari semua pasangan simpul dalam graf.

Tentu saja, pengaplikasian graf dalam *Global Positioning System* bukan hanya ini. Teori graf yang luas sangat berguna untuk berbagai hal dalam kehidupan nyata dan sangat mudah untuk dikembangkan. Namun untuk penentuan jalur terpendek, graf berbobot dan matriks ketetangaan adalah yang diaplikasikan.

V. UCAPAN TERIMA KASIH

Pada kesempatan ini, Penulis mengucapkan terima kasih kepada Tuhan YME yang telah memberi kesempatan dan kemampuan agar Penulis dapat belajar dan menuliskan makalah ini. Penulis juga ingin berterima kasih kepada kedua orangtua Penulis serta teman-teman

Penulis. Selain itu, Penulis juga mengucapkan terima kasih sebesar-besarnya untuk Ibu Dra. Harlili S., M.Sc. dan Bapak Dr.Ir. Rinaldi Munir, MT. selaku dosen pengajar mata kuliah Matematika Diskrit yang telah mengajar dan membimbing Penulis semester ini. Terakhir, Penulis ingin berterima kasih kepada siapapun yang telah membantu dalam penulisan makalah ini.

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, 8 Desember 2015



Sekar Anglila Hapsari / 13514069

DAFTAR PUSTAKA

- [1] <https://www.seeklogo.net/wp-content/uploads/2015/09/new-google-maps-logo.png>, diakses pada 5 Desember 2015.
- [2] Munir, Rinaldi, 2006. “*Matematika Diskrit*”, Bandung: Penerbit Informatika Bandung.
- [3] <https://pustakailmuperpustakaan.files.wordpress.com/2010/08/graf.jpg>, diakses pada 6 Desember 2015.
- [4] <http://aimyaya.com/images/contohgraf.gif>, diakses pada 7 Desember 2015.
- [5] http://2.bp.blogspot.com/-VS5VihnC2nk/Tvscrq3qj_I/AAAAAAAAAJE/cdSG0L_gq0M/s1600/Picture1.8.png, diakses pada 7 Desember 2015.
- [6] http://2.bp.blogspot.com/-OMBBMq1PmgM/TuRz4jrNfNI/AAAAAAAAAHw/FiLkr_TfKOw/s1600/43.png, diakses pada 7 Desember 2015.
- [7] <http://faculty.ycp.edu/~dbabcock/PastCourses/cs360/lectures/images/lecture15/adjlistexample.png>, diakses pada 7 Desember 2015.
- [8] <https://62e528761d0685343e1c-f3d1b99a743ffa4142d9d7f1978d9686.ssl.cf2.rackcdn.com/files/20819/width668/nsfc25rf-1362119268.jpg>, diakses pada 8 Desember 2015.
- [9] <http://d1g1xt8vb0knt.cloudfront.net/wp-content/uploads/Fig-11.jpg>, diakses pada 8 Desember 2015.
- [10] https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm, diakses pada 8 Desember 2015.
- [11] https://upload.wikimedia.org/wikipedia/commons/thumb/2/2e/Floyd-Warshall_example.svg/600px-Floyd-Warshall_example.svg.png, diakses pada 8 Desember 2015.
- [12] Bovy, P. and E. Stern .1990. “*Route Choice: Wayfinding in Transportation Networks*”. Dordrecht: Kluwer Academic Publishers.
- [13] Anonim. 2011. “*What is GPS? How does it work?*”. <https://www.loc.gov/rr/scitech/mysteries/global.html>, diakses pada 5 Desember 2015.
- [14] Anonim. 2013. “*Explainer : What is GPS?*”. <http://theconversation.com/explainer-what-is-gps-12248>, diakses pada 8 Desember 2015.
- [15] Anonim. “*What is GPS?*”. <http://www.gps.gov/applications/>, diakses pada 7 Desember 2015.
- [16] Anonim. “*All pairs shortest path*”. <http://www.wolframalpha.com/input/?i=all-pairs+shortest+path>, diakses pada 7 Desember 2015.
- [17] Anonim. “*Floyd-Warshall Algorithm*”. <http://mathworld.wolfram.com/Floyd-WarshallAlgorithm.html>, diakses pada 7 Desember 2015.
- [18] Anonim. “*Dijkstra Algorithm*”. <http://mathworld.wolfram.com/DijkstrasAlgorithm.html>, diakses pada 7 Desember 2015.
- [19] Anonim. “*Greedy Algorithms | Set 7 (Dijkstra’s shortest path algorithm)*”. <http://www.geeksforgeeks.org/greedy-algorithms-set-6-dijkstras-shortest-path-algorithm/>, diakses pada 8 Desember 2015.