Penggunaan Teori Graf Dalam Sistem Navigasi GPS

Marchy Tio Pandapotan
13509026
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13509026@std.stei.itb.ac.id

***Abstrak* — Kendala yang diakibatkan oleh meningkatnya kebutuhan transportasi adalah penentuan jalur perjalanan yang magkus. Salah satu solusi dari masalah ini adalah dengan membuat sebuah sistem yang bisa membantu para pengendara dalam menentukan rute perjalanan yang efisien. Teori Graf dapat digunakan untuk melakukan pencarian lintasan terpendek. Dengan membandingkan waktu tempuh dari setiap jalan maka akan didapatkan suatu jalur yang efisien. GPS(*Global Positioning System*) *receiver* adalah realisasi dari sistem tersebut. Alat ini berfungsi untuk menunjukkan posisi dari kendaraan yang memakai sistem navigasi.**

Kata Kunci: Teori Graf, sistem navigasi, lintasan terpendek.

1. PENDAHULUAN

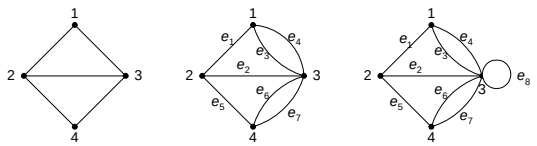
Global Positioning System adalah sistem navigasi satelit yang dikembangkan oleh Departemen Pertahanan Amerika (US DoD = *United States Department of Defense*) dengan nama lengkapnya adalah NAVSTAR GPS. NAVSTAR bukanlah sebuah singkatan. NAVSTAR adalah nama yang diberikan John Walsh, seorang penentu kebijakan penting dalam program GPS. Sistem ini mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini diterima oleh alat penerima di permukaan dan digunakan untuk menentukan arah, posisi, kecepatan, dan waktu..

2. LANDASAN TEORI

2.1 Definisi Graf

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek – objek tersebut. Graf G = (V,E), yang berarti :

- V = himpunan tidak-kosong dari simpul-simpul (vertices) = $\{v_1, v_2,...,v_n\}$
- E = himpunan sisi (edges) yang menghubungkan sepasang simpul = $\{e_1, e_2, ..., e_n\}$



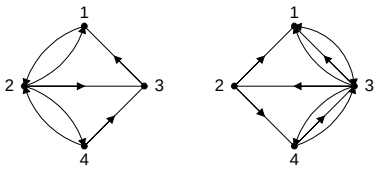
G_1 G_2 G_3

Gambar 2.1 (a) graf sederhana, (b) graf ganda, dan (c) graf semu

2.2 Jenis - jenis Graf

Berdasarkan ada tidaknya sisi ganda pada suatu graf, maka graf digolongkan menjadi dua jenis yaitu **graf sederhana** dan **graf tak-sederhana.** Grad sederhana adalah graf yang tidak mengandung gelang maupun sisi ganda. Sedangkan graf tak-sederhana adalah graf yang mengandung sisi ganda atau gelang.

Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas dua jenis yaitu **graf tak-berarah** dan **graf berarah**. Graf tak-berarah dalah graf yang sisinya tidak mempunyai orientasi arah. Sedangkan graf berarah adalah graf yang setiap sisinya diberikan orientasi arah.



Gambar 2.2 (a) graf berarah, (b) graf-ganda berarah

2.3 Terminologi Graf

- Ketetanggan, dua buah simpul dikatakan bertetangga bila keduanya terhubung langsung.
- Bersisian, sebuah sisi yang menghubungkan dua buah simpul dikatakan bersisian dengan simpul tersebut.
- Simpul terpencil, simpul yang tidak mempunyai sisi yang bersisian dengannya.
- Graf kosong, gram yang himpunan sisinya merupakan himpunan kosong.
- Derajat, jumlah sisi yang bersisian dengan simpul tersebut.
- Lintasan, 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 $v_0, e_1, v_1, e_2, v_2,... , v_{n-1}, e_n, v_n$ sedemikian sehingga $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, ... , $e_n = (v_{n-1}, v_n)$ 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.

- Upagraf dan Komplemen Upagraf. Misalkan $G = (V, E)$ adalah sebuah graf. $G_1 = (V_1, E_1)$ adalah **upagraf** (*subgraph*) dari G jika $V_1 \subseteq V$ dan $E_1 \subseteq E$. **Komplemen** dari upagraf G_1 terhadap graf G adalah graf $G_2 = (V_2, E_2)$ sedemikian sehingga $E_2 = E - E_1$ dan V_2 adalah himpunan simpul yang anggota-anggota E_2 bersisian dengannya
- Upagraf Rentang, Upagraf $G_1 = (V_1, E_1)$ dari $G = (V, E)$ dikatakan **upagraf rentang** jika $V_1 = V$ (yaitu G_1 mengandung semua simpul dari G).
- *Cut-Set,* adalah himpunan sisi yang bila dibuang dari G menyebabkan G tidak terhubung. Jadi, *cut-set* selalu menghasilkan dua buah komponen.
- Graf Berbobot, adalah graf yang setiap sisinya diberi sebuah harga (bobot).

3. PEMBAHASAN

Dalam makalah ini , sistem navigasi GPS akan dijelaskan mulai dari definisi GPS, implementasi teori graf dalam GPS, hingga algoritma yang digunakan dalam penentuan jalur terpendek.

3.1 Definisi GPS

GPS terdiri dari 3 segmen : segmen angkasa, segmen kontrol/pengendali, dan segmen pengguna. Segmen angkasa terdiri dari 24 satelit yang beroperasi dalam 6 orbit pada ketinggian 20.200 km dan inklinasi 55 derajat dengan periode 12 jam. Satelit tersebut memutari orbitnya sehingga minimal ada 6 satelit yang dapat dipantau di titik manapun di Bumi. Satelit tersebut mengirimkan posisi dan waktu kepada seluruh pengguna.

Segmen kedua adalah segmen kontrol. Pusat kendali utama terdapat di Colorado Spring, dan 5 stasiun pemantau lainnya dan 3 antena yang tersebar di Bumi. Stasiun pemantau memantau semua satelit dan mengumpulkan data lalu mengirimkan informasi tersebut kepada pusat pengendali utama. Pusat pengendali utama akan melakukan perhitungan dan pengecekan orbit satelit. Informasi akan dikoreksi dan dikirim kembali ke satelit GPS.

Segmen terakhir adalah segmen pengguna. Pengguna membutuhkan penerima GPS, selanjutnya kita sebut **perangkat GPS**, yang terdiri dari penerima, prosesor, dan antena, sehingga memungkinkan kita untuk menerima sinyal dari satelit GPS dan kemudian menghitung posisi, kecepatan, dan waktu.

3.2 Cara Kerja GPS

GPS menggunakan 24 satelit yang mengorbit Bumi untuk memancarkan sinyal dan ditangkap oleh sebuah alat penerima. Cara kerja GPS dibagi menjadi tiga bagian, yaitu bagian kontrol, bagian angkasa, dan bagian pengguna.

Bagian kontrol berfungsi untuk mengontrol. Satelit yang mengorbit di Bumi bisa saja sedikit keluar dari orbit, sehingga bagian ini melacak orbit satelit, lokasi, ketinggian, dan kecepatan. Sinyal – sinyal dari satelit diterima oleh bagian kontrol, lalu dikoreksi, dan dikirimkan kembali ke satelit. Data yang sudah dikoreksi ini disebut dengan data ephemeris.

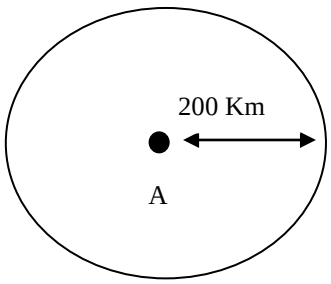
Bagian angkasa terdiri dari kumpulan satelit yang mengorbit Bumi dan berada di ketinggian 12.000 mil di atas permukaan bumi. Letak dari satelit diatur sehingga alat navigasi dapat menerima sinyal setiap saat paling sedikit dari empat buah satelit yang mengorbit. Sinyal satelit ini memiliki kekurangan yaitu tidak dapat menembus gedung atau gunung. Satelit dilengkapi dengan jam atom, dan juga memancarkan informasi dari "jam" ini. Data ini dipancarkan dengan kode 'pseudo-random'. Setiap satelit memiliki kode identitasnya masing – masing. Nomor kode ini akan ditampilkan pada alat navigasi penerima, sehingga kita bisa melakukan identifikasi sinyal satelit yang sedang mengirimkan sinyal kepada kita. Data ini digunakan oleh alat navigasi untuk mengukur jaraknya dengan dengan satelit, yang kemudian digunakan untuk mengukur koordinat lokasi. Hal lain yang mempengaruhi perhitungan adalah kekuatan sinyal. Sebuah alat akan menerima sinyal lebih kuat apabila berada tepat dibawah satelit.

Gelombang yang dipakai untuk alat navigasi berbasis satelit pada umumnya memiliki dua jenis. Jenis pertama biasa dikema; dengan sebutan L1 pada 1575,42 MHz dan diterima oleh alat navigasi. Satelit jenis kedua, yang lebih dikenal dengan gelombang L2 berada pada frekuensi 1227,6 Mhz dan digunakan untuk tujuan militer dan bukan untuk umum.

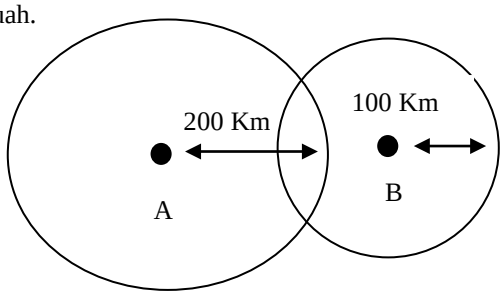
Bagian pengguna terdiri dari alat navigasi yang digunakan. Satelit akan mengirimkan dua jenis data. Data yang pertama adalah data almanak. Data ini merupakan perkiraan lokasi satelit yang dipancarkan secara terus menerus. Jenis yang kedua adalah data ephemeris. Data ephemeris valid untuk jangka waktu 4 – 6 jam. Untuk menunjukkan koordinat sebuah titik (tiga dimensi), diperlukan paling sedikit sinyal dari tiga buah satelit, dikenal dengan ***2-D Trilateration***. Untuk menunjukkan data ketinggian sebuah titik (tiga dimensi), diperlukan tambahan sinyal dari 1 buah satelit lagi, dikenal dengan ***3-D Trilateration***.

3.3 2-D Trilateration

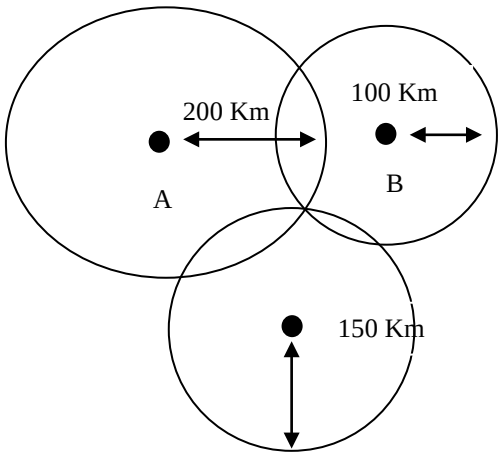
Jika kita sedang tersesat dan kita bertanya pada orang setempat dan mendapatkan informasi bahwa kita berada 200 km dari kota A, mungkin ini menjadi fakta yang bagus. Tetapi, hal ini belum cukup, kita bisa berada di setiap tempat di radius 200 km dari kota A, seperti gambar di bawah ini.



Tidak lama kemudian, ada petunjuk yang mengatakan bahwa kita sedang berada 100 Km dari kota B. Dengan menggabungkan dua buah informasi yang didapat, kemungkinan tempat kita berada dipersempit menjadi dua buah.



Jika di petunjuk jalan tersebut juga terdapat informasi bahwa kita berada 150 Km dari kota C, maka kita bisa menghilangkan salah satu dari kemungkinan yang ada, dan mendapatkan posisi kita sesungguhnya.



Konsep seperti inilah yang dipakai oleh alat navigasi untuk menentukan lokasi kita. Untuk itu diperlukan minimal 3 buah satelit, karena kita membutuhkan minimal 3 buah data untuk menentukan lokasi pasti dari tempat kita berada.

3.4 3-D Trilateration

Pada dasarnya, 3-D Trilateration tidak terlalu berbeda dengan 2-D Trilateration. Pada 3-D Trilateration, kita menggunakan bola sebagai ganti lingkaran. Jika kita tahu, bahwa kita berada 10 Km dari satelit A yang sedang mengorbit, kita bisa berada di manapun di permukaan bola dalam radius 10 Km. Jika kita juga tahu, bahwa kita berada 15 Km dari satelit B, maka kita bisa menggabungkan dengan informasi sebelumnya. Bola ini bersinggungan dalam lingkaran sempurna. Jika kita tahu jarak ke satelit ketiga, maka kita mendapatkan bola ketiga yang bersinggungan dengan lingkaran tadi di dua titik.

Bumi tempat kita berada bisa dijadikan sebagai bola keempat. Hanya satu dari dua titik yang mungkin berada di permukaan planet, sehingga kita bisa mengabaikan titik lainnya yang berada di angkasa. Pada umumnya, alat penerima mengacu pada empat atau lebih satelit, dengan tujuan meningkatkan ketepatan informasi.

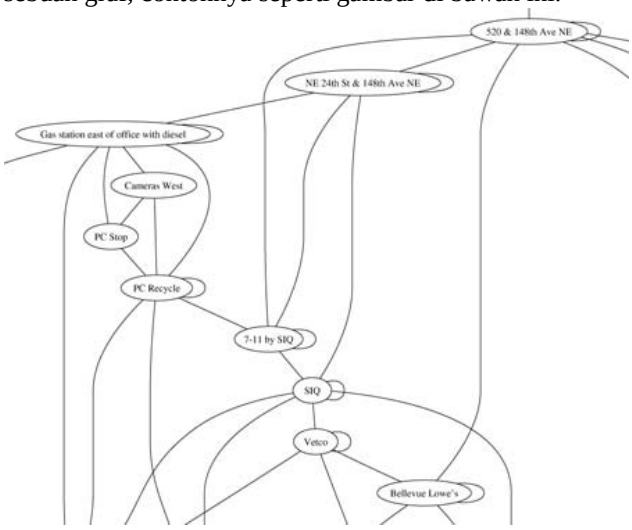
Untuk mempermudah perhitungan, maka alat penerima GPS harus mengetahui dua hal :

- Lokasi minimal 3 buah satelit yang mengorbit
- Jarak antara alat penerima dengan tiap dari satelit tersebut

Alat penerima menentukan dua hal diatas dengan menganalisi frekuensi tinggi, sinyal radio berdaya rendah dari satelit GPS. Perangkat yang lebih baik memiliki beberapa penerima, sehingga bisa menangkap sinyal dari beberapa satelit secara bersamaan.

3.5 Implementasi Teori Graf dalam GPS

Ketika menggunakan GPS, kita akan melihat titik – titik tempat yang dapat kita tuju. Jika, Tempat – tempat tersebut dapat kita gambarkan sebagai simpul dan jalan yang menghubungkan tempat – tempat tersebut digambarkan sebagai sisi, maka kita akan mendapatkan sebuah graf, contohnya seperti gambar di bawah ini.



Gambar 2.1 Representasi GPS dalam graf

Setiap jalan yang direpresentasikan oleh sisi mempunyai bobot. Ketika kita menentukan tempat yang akan kita tuju, GPS akan melakukan perhitungan terhadap bobot tiap sisi dan mencari sisi yang mempunyai bobot paling kecil untuk mencapai tempat yang kita tuju tersebut. Dalam menentukan jalan terpendek yang dipilih, GPS dapat menggunakan beberapa algoritma, antara lain adalah algoritma Dijkstra, algoritma Floyd – Warshall, dan algoritma Johnson. Algoritma – algoritma tersebut

mempunyai fungsi yang berbeda satu sama lainnya yang akan dijelaskan pada upabab selanjutnya.

3.6 Algoritma Dijkstra

Algoritma ini ditemukan oleh Edsger Dijkstra. Algoritma ini adalah sebuah *greedy algorithm* yang dipakai untuk memeachkan permasalah jarak terpendek (*shortest path problem)* untuk sebuah graf berarah dengan bobot – bobot sisi yang bernilai tak-negatif.

Misalnya, simpul dari sebuah graf melambangkan kota – kota dan bobot sisi melambangkan jarak antara kota – kota tersebut, maka algoritma Dijkstra dapat digunakan untuk menemukan jarak terpendek antara dua kota.

Masukan dari algoritma ini adalah sebuah graf berarah yang mempunyai bobot G dan sebuah sumber simpul S dalam G dan V adalah himpunan semua simpul dalam graf G. Setiap sisi dari graf ini adalah pasangan simpul (u,v) yang melambangkan hubungan dari simpul u ke simpul v. Himpunan semua sisi disebut E.

Bobot dari semua sisi dihitung dengan fungsi

$$w: E \rightarrow [0, \infty)$$

Jadi, w(u,v) adalah jarak tak-negatif dari vertex u ke vertex v. Biaya dari sebuah sisi dapat dianggap sebagai jarak antara dua simpul, yaitu jumlah jarak semua sisi dalam jalur tersebut. Untuk sepasang simpul s dan t dalam V, algoritma ini menghitung jarak terpendek dari s ke t.

Kita misalkan simpul tempat kita akan memulai adalah simpul awal. Lalu, jarak dari simpul Y menjadi jarak antara simpul awal ke Y. Algoritma Dijkstra akan menentukan beberapa nilai jarak awal dan mengembangkan melalu beberapa langkah.

- Tentukan bobot jarak pada setiap simpul. Untuk simpul awal berbobot 0 dan untuk simpul lainnya berbobot tak hingga.
- Tandai semua simpul sebagai simpul yang belum dikunjungi. Untuk simpul awal ditandai sebagai aktif.
- Untuk simpul aktif, lihat semua tetangganya yang belum dikunjungi dan hitung jarak sementara dari simpul awal. Contoh, jika simpul aktif (A) mempunyai jarak 6, dan sebuah sisi yang menghubungkan simpul itu dengan simpul lainnya (B) adalah 2, maka jarak ke B dari A menjadi 6 + 2 = 8. Jika, jarak ini, lebih kecil dari jarak yang terekam sebelumnya(tak hingga untuk simpul lainnya, nol untuk simpul awal), maka tulis ulang nilai dari jarak tersebut.
- Ketika telah membandingkan semua tetangga dengan simpul yang aktif, maka tandai simpul – simpul tetangga tadi sebagai sudah dikunjungi. Simpul yang telah dikunjungi tidak akan di cek lagi; jaraknya yang terekam adalah jarak minimal.
- Algoritma ini selesai jika semua simpul sudah dikunjungi. Dengan kata lain, berikan simpul yang belum dikunjungi jarak terkecil (dari simpul awal) sebagai simpul aktif dan lanjutkan dari langkah ketiga.

3.7 Algoritma Floyd – Warshall

Algoritma Floyd-Warshall menghitung jarak terpendek untuk semua pasangan titik pada sebuah graf, dan melakukannya dalam waktu berorde kubik.

Algoritma ini memiliki input graf berarah dan berbobot (V,E), yang berupa daftar titik (node/vertex V) dan daftar sisi (edge E). Jumlah bobot sisi-sisi pada sebuah jalur adalah bobot jalur tersebut. Sisi pada E diperbolehkan memiliki bobot negatif, akan tetapi tidak diperboleh bagi graf ini mempunyai siklus yang berbobot negatif. Algoritma ini menghitung bobot terkecil dari semua jalur yang menghubungkan sebuah pasangan titik, dan melakukannya sekaligus untuk semua pasangan titik. Algoritma ini berjalan dengan waktu $\Theta(|V|^3)$. Hal ini luar biasa dengan mengingat bahwa mungkin ada sampai dengan Ω (| V | 2) sisi dalam graf dan setiap kombinasi sisi akan diuji. Algoritma ini bekerja dengan meningkatkan secara bertahap perkiraan di jalan terpendek antara dua simpul, sampai perkiraan mencapai optimal.

Pertimbangkan graf G bersimpul V, masing – masing diberi nomor 1 sampai N. Selanjutnya, pertimbangkan fungsi shortestPath(i,j,k) yang mengembalikan jalur terpendek yang mungkin dari i ke j hanya menggunakan simpul dari set {1,2,...,k} sebagai titik tengah sepanjang jalan. Sekarang, mengingat fungsi ini, tujuan kami adalah untuk menemukan lintasan terpendek dari setiap i ke setiap j hanya menggunakan simpul 1 sampa k + 1.

Ada dua kemungkinan untuk masing-masing jalur : baik yang benar-benar jalur terpendek yang hanya menggunakan simpul di set {1,...,k}, atau terdapat bebrapa jalur yang langsung dari i ke k + 1, kemudian dari k + 1 sampai j yang lebih baik. Kita tahu bahwa lintasan terbaik dari i ke j yang hanya menggunakan simpul 1 sampai k didefinisikan oleh shortestPath(i,j,k) dan jelas bahwa jika ada lintasan yang lebih baik dari i ke + 1 sampai j, maka panjang lintasan ini akan menjadi gabungan dari lintasan terpendek dari i ke k + 1(menggunakan simpul dalam {1,...,k}) dan lintasan terpendek dari k + 1 sampai j (juga menggunakan simpul di {1,..,k}).

3.8 Algoritma Johnson

Algoritma ini digunakan untuk memecahkan masalah *All-pairs Shortest Path* pada sparse graph. Algoritma ini memungkinkan beberapa bobot sisi bernilai negatif, tetapi tidak boleh ada siklus yang berbobot negatif. Algoritma ini bekerja dengan menggunakan algoritma Bellman-Ford untuk menghitung transformasi dari grafik input yang menghilangkan semua bobot negatif, sehingga memungkinkan algoritma Dijkstra untuk digunakan pada graf yang bertransformasi. Algoritma ini dinamakan berdasarkan Donald B. Johnson, orang pertama yang mengenalkannya pada tahun 1977.

Algoritma Johnson terdiri dari langkah – langkah berikut :

- Pertama, simpul baru yang dinamakan q ditambahkan ke graf, dihubungkan oleh sisi yang berbobot nol ke simpul yang lain.
- Kedua, kita mulai menggunakan algoritma Bellman-Ford, mulai dari simpul q untuk menemukan setiap simpul v yang melalui lintasan dengan bobot paling kecil dari q ke v. Jika langkah ini menemukan siklus negatif, maka algoritma dihentikan.
- Selanjutnya, sisi dari graf yang asli diberikan bobot baru dengan menggunakan nilai yang dihitung menggunakan algoritma Bellman-Ford: sebuah sisi dari u ke v, yang mempunyai panjang w(u,v) diberikan panjang baru yaitu w(u,v) + h(u) – h (v).
- Terakhir, hilangkan q, dan algoritma Dijkstra digunakan untuk menemukan lintasan terpendek dari setiap simpul s ke setiap simpul lainnya dari graf yang telah mempunyai bobot yang baru.

Pada graf yang sudah mempunyai bobot baru, semua lintasan dari dari pasangan simpul s dan t ditambahkan h(s) – h(t), jadi lintasan terpendek dari graf yang asli tetap menjadi lintasan terpendek dalam graf yang sudah berubah dan sebaliknya. Namun, dikarenakan cara h(v) dihitung, semua panjang sisi yang dimodifikasi bernilai non-negatif, memastikan keoptimalan dari lintasan yang ditemukan oleh algoritma Dijkstra. Jarak pada graf yang asli bisa dihitung dari jarak yang dihitung menggunakan algoritma Dijkstra pada graf yang mempunyai bobot baru dengan membalik transformasi bobot.

3.9 Kelemahan GPS

Karena alat navigasi ini bergantung seluruhnya pada sinyal satelit, maka alat ini tidak dapat bekerja maksimal ketika ada gangguan pada sinyal satelit. Beberapa hal yang dapat mengurangi kekuatan sinyal satelit :

- Kondisi geografis. Selama masih dapat melihat langit yang cukup luas, alat ini dapat berfungsi
- Hutan. Semakin lebat hutan, maka keberterimaan sinyal makin berkurang.
- Air
- Kaca film mobil, terutama yang mengandung metal.
- Alat – alat elektronik yang dapat mengeluarkan gelombang elektromagnetik
- Gedung-gedung. Tidak hanya di dalam gedung, berada diantara 2 buah gedung tinggi juga akan menyebabkan efek seperti berada di dalam lembah.
- Sinyal yang memantul, misal bila berada diantara gedung – gedung tinggi, dapat mengacaukan perhitungan alat navigasi sehingga alat navigasi dapat menunjukkan posisi yang salah atau tidak akurat.

4. PSEUDOCODE

Pseducode dari beberapa algoritma diatas :

4.1 Algoritma Dijkstra

```
 1  function Dijkstra(G, w, s)
 2     for each vertex v in V[G]                        // Initializations
 3           d[v] := infinity
 4           previous[v] := undefined
 5     d[s] := 0                                        // Distance from s to s
 6     S := empty set
 7     Q := V[G]                                        // Set of all vertices
 8     while Q is not an empty set                      // The algorithm itself
 9           u := Extract_Min(Q)
10           S := S union {u}
11           for each edge (u,v) outgoing from u
12                 if d[u] + w(u,v) < d[v]              // Relax (u,v)
13                       d[v] := d[u] + w(u,v)
14                       previous[v] := u
```

4.2 Algoritma Floyd - Warshall

```
function fw(int[1..n,1..n] graph) {
    // Inisialisasi
    var int[1..n,1..n] jarak := graph
    var int[1..n,1..n] sebelum
    for i from 1 to n
        for j from 1 to n
            if jarak[i,j] < Tak-hingga
                sebelum[i,j] := i
    // Perulangan utama pada algoritma
    for k from 1 to n
        for i from 1 to n
            for j from 1 to n
                if jarak[i,j] > jarak[i,k] + jarak[k,j]
                    jarak[i,j] = jarak[i,k] + jarak[k,j]
                    sebelum[i,j] = sebelum[k,j]
    return jarak
}
```

5. KESIMPULAN

Dari penjelasan pada bab-bab sebelumnya, dapat kita tarik beberapa kesimpulan tentang implementasi graf dalam sistem navigasi GPS :

- Pada sistem navigasi GPS, tempat – tempat yang akan dituju dapat direpresentasikan sebagai simbul. Sedangkan, jalan yang menghubungkan tempat – tepat tersebut dapat dianggap sebagai sisi.
- Dalam menentukan lintasan terpendek, dapat digunakan beberapa algoritma seperti algoritma Dijkstra, algoritma Folyd – Warshall, dan algoritma Johnson.
- Beberapa hal yang perlu dipertimbangkan sebelum memilih sistem navigasi GPS adalah alasan kita membeli, harga, kapasitas penyimpanan, daya tahan baterai, tahan air, akurasi, program, dan yang terakhir adalah peta.

6. DAFTAR REFERENSI

[1] Munir, Rinaldi. 2008. Diktak Kuliah IF2091 Struktur Diskrit. Edisi keempat, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

[2] http://id.wikipedia.org/wiki/Global_Positioning_System diakses 10 Desember 2010 pukul 12.46.

[3] http://navigasi.net/gofaq.php diakses 10 Desember 2010 pukul 12.51.

[4] http://id.wikipedia.org/wiki/Algoritma_Dijkstra diakses 14 Desember 2010 pukul 14.03.

[5] http://id.wikipedia.org/wiki/Algoritma_Floyd-Warshall diakses 14 Desember 2010 pukul 14.03.

[6] http://en.wikipedia.org/wiki/Johnson%27s_algorithm diakses 14 Desember 2010 pukul 14.05.

7. 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, 16 Desember 2010

Marchy Tio Pandapotan (13509026)