

Penggunaan Algoritma *Dynamic Programming* pada Aplikasi *GPS Car Navigation System*

Muhammad Anis 1350868
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
if18068@students.if.itb.ac.id

ABSTRAK

GPS merupakan singkatan dari *Global Positioning System* atau sistem penentuan posisi menggunakan satelit. Sistem ini sudah banyak digunakan mulai dari militer hingga masyarakat sipil. Salah satu aplikasi yang menggunakan sistem ini adalah sistem *GPS* ini adalah sistem navigasi penunjuk jalan yang ditanam pada mobil. Nama sistem ini adalah *GPS Car Navigation System*. *GPS Car Navigation Sistem* atau sistem navigasi pada mobil menggunakan *GPS* adalah sistem navigasi satelit yang didesain untuk digunakan pada kendaraan bermotor. Sistem ini menggunakan alat navigasi *GPS* untuk menentukan data posisi pengguna saat ini secara *realtime*. Sistem ini juga mampu menunjukkan jalan terbaik ke suatu tujuan dengan memperhatikan *traffic* di jalanan, jarak tempuh terpendek yang dapat dilalui. Data lainnya didapatkan melalui database yang memiliki data jalur-jalur yang bertentangan dengan jalur yang sedang dijalan oleh user tersebut saat ini. Penunjukan jalur terbaik ini dapat menggunakan salah algoritma yaitu *Dynamic programming*.

Kata Kunci: *GPS, Dynamic Programmin, Navigation System.*

1. PENDAHULUAN

Pengguna kendaraan di dunia saat ini sudah sangat banyak dan terus berkembang hingga saat ini. Pengguna kendaraan biasanya ingin menuju ke suatu tujuan namun tidak tahu harus melewati jalur yang mana atau bahkan tidak tahu dimana tujuan yang dimaksud tersebut. Pada cara yang manual yang masih sering digunakan adalah bertanya langsung pada orang yang kebetulan sedang berada di dekat daerah tersebut. Namun masalah terjadi jika di daerah sekitar juga sedang tidak ada orang yang bisa untuk ditanyai. Untuk mereka yang mengetahui daerah tujuan pun terkadang terdapat beberapa alternative jalur yang terkadang pengendara sendiri bingung untuk menggunakan jalur mana yang dapat mengantarkan pengendara ke suatu tujuan dalam waktu sesingkat mungkin. Oleh karena itu diperlukan sebuah sistem yang mampu menunjukkan jalur terbaik tersebut kepada pengguna yang mampu membangun jalur terbaik secara cepat, tepat dan real time.

Pembangunan sistem semacam ini kian pesat dan menjadi suatu nilai tambah tersendiri pada kendaraan

bermotor terutama mobil. Sistem ini disebut dengan *GPS Car Navigation System*. Sistem ini mampu menunjukkan jalur terbaik pada pengguna kendaraan dari tempat dia berada sekarang ke suatu tujuan yang telah ditentukan dengan memperhatikan beberapa variable lain seperti kemacetan dan hal lainnya.

2. DASAR TEORI

2.1 GPS (Global Positioning System)

GPS (*Global Positioning System*) adalah sistem satelit navigasi penentuan posisi dengan mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini lalu diterima oleh alat penerima di permukaan dan digunakan untuk menentukan posisi dan kecepatan tiga-dimensi serta informasi mengenai waktu secara kontinu di seluruh dunia tanpa tergantung pada waktu dan cuaca.

Ketelitian dari *GPS* dapat mencapai beberapa mm untuk ketelitian posisinya, beberapa cm/s untuk ketelitian kecepatannya dan beberapa nano detik untuk ketelitian waktunya. Prinsip penentuan posisi dengan *GPS* adalah dengan menggunakan metode reseksi jarak, dimana pengukuran jarak dilakukan secara simultan ke beberapa satelit yang telah diketahui koordinatnya. Pada pengukuran *GPS*, sedikitnya ada empat parameter yang harus ditentukan: yaitu 3 parameter koordinat X,Y,Z dan satu parameter kesalahan akibat ketidaksinkronan jam osilator antar satelit dengan jam di receiver *GPS*.

2.2 *Dynamic programming*

Dynamic programming awalnya digunakan pada tahun 1940-an oleh Richard Bellman untuk menjelaskan pemecahan suatu masalah dimana pemilihan solusi dilakukan satu demi satu. *Dynamic programming* adalah teknik optimasi masalah matematika dan computer programming. Pada kedua konteks, *dynamic programming* merujuk kepada pemecahan suatu

masalah besar dengan cara memecah masalah besar tersebut ke dalam sub-masalah yang lebih sederhana menggunakan rekursif. Dynamic dapat dibagi menjadi dua bagian yaitu *Top Down Approach* dan *Bottom Up Approach*.

- *Top Down Approach*

Pendekatan *Top Down Approach* adalah dasar dari formulasi rekursif dari semua masalah. Jika solusi dari sebuah masalah dapat diformulasikan secara rekursif menggunakan solusi dari upa-masalah dan upa-masalah itu berulang maka kita dapat dengan mudah meletakkan setiap solusi kedalam sebuah table. Ketika kita mencoba menyelesaikan sebuah masalah baru, kita mencari ke table apakah solusi masalah tersebut sudah ada dalam table atau tidak. Jika belum ada, maka kita menyelesaikan upa-masalah tersebut dan meletakkan solusinya ke dalam table tersebut.

- *Bottom Up Approach*

Bottom Up Approach adalah teknik pendekatan penyelesaian rekursif yang lebih menarik. Teknik ini mencoba menyelesaikan sebuah masalah dengan formulasi upa-masalahnya. Teknik ini memiliki konsep bahwa kita menyelesaikan upa-masalah terlebih dahulu dan menggunakan solusinya untuk menyelesaikan masalah yang lebih besar. ini juga digunakan untuk membangkitkan solusi masalah yang lebih besar lagi. Sebagai contoh, jika kita telah mengetahui solusi dari F_{41} dan F_{40} kita dapat menemukan dari nilai F_{42} .

2.3 GPS Navigation System

Sistem navigasi berbasis GPS pada mobil adalah barang yang sudah tidak langka saat ini. Dengan mengkombinasikan antara sinyal GPS dari satelit dan sebuah map di mobil yang interaktif, sistem ini mampu menentukan rute terbaik ke sebuah tujuan tertentu berdasarkan pada beberapa variable. Beberapa sistem navigasi GPS terhubung pada sumber dari informasi lalu-lintas dan secara otomatis membangun sistem penentuan rute terbaik. Jika pengendara salah mengambil jalur maka sistem ini akan memberikan peringatan dan membangun jalur terbaik yang baru.

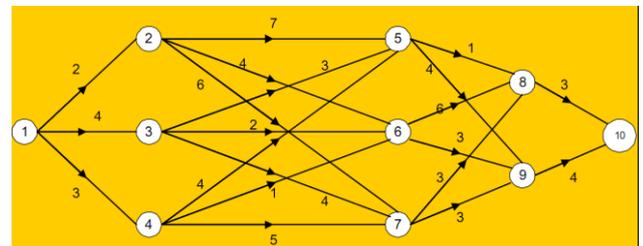
3. METODE PENYELESAIAN MASALAH

3.1 Analisis Masalah

Masalah penunjukan rute terbaik pada sistem navigasi berbasis GPS pada mobil ini dapat kita analogikan menggunakan graf berarah yang memiliki bobot. Graf berarah

disini dimaksudkan bahwa ada jalur yang memiliki lajur dua arah dan ada jalur yang memiliki satu arah. Bobot disini adalah jarak tempuh antar simpul serta beberapa variable tambahan seperti tingkat kemacetan dan variable-variable lain yang mempengaruhi kecepatan perjalanan. Bobot-bobot ini dapat berubah secara realtime tergantung pada keadaan jalanan saat itu. Misal pada suatu saat terjadi kemacetan panjang di suatu jalan maka bobot jalan tersebut akan bertambah. Atau misalkan terdapat kecelakaan sehingga jalan tersebut ditutup akan menyebabkan hilangnya jalur dari satu simpul ke simpul lainnya.

Sangat dinamisnya perubahan bobot dan rangkaian antar node menyebabkan perubahan hubungan ketetanggaan dan penentuan rute terbaik akan sangat sering berubah tergantung waktu dan keadaan yang ada di jalanan.



Gambar1 Representasi Graph Menuju Suatu Tujuan

3.2 Prinsip Pemecahan Masalah Menggunakan *Dynamic programming*

Seperti telah disebutkan di atas bahwa perubahan table ketetanggaan serta perubahan bobot antara satu simpul dan simpul lainnya akan sangat sering berubah. Penentuan jalur terbaik dengan membangun matriks ketetanggaan kembali lalu mencari jalur terbaik seperti pada algoritma Dijkstra akan sangat memakan waktu. Oleh karena itu penggunaan algoritma *Dynamic programming* akan sangat membantu penyelesaian masalah ini.

Prinsip penyelesaian masalah dengan menggunakan *dynamic programming* adalah dengan menyelesaikan suatu masalah yang besar menjadi masalah-masalah yang kecil. Salah satu prinsip yang dapat kita gunakan adalah prinsip optimalitas yang menyatakan jika solusi total optimal maka bagian solusi sampai tahap ke-k juga optimal. Dengan prinsip ini dapat dijamin bahwa pengambilan keputusan pada suatu tahap adalah keputusan yang benar untuk tahap-tahap selanjutnya. Pada kasus kali ini kita asumsikan bahwa pada awalnya perusahaan penyedia jasa sistem ini telah membuat table ketetanggaan serta bobot secara lengkap, misal seperti ini:

a. Table Menuju

	1	2	3	4	5	6	7	8	9	10
1	-	2	3	5	7	3	5	1	-	4
2	2	-	3	5	4	-	2	5	7	6
3	7	8	-	9	6	4	6	-	4	5
4	5	2	12	-	3	4	5	7	7	6
5	13	4	5	21	-	14	6	5	8	9
6	23	4	2	3	4	-	5	7	7	-
7	6	5	3	2	8	5	-	3	7	5
8	-	4	5	6	3	9	9	-	-	6
9	14	21	12	8	3	3	3	6	-	4
10	12	23	24	1	2	8	5	9	6	-

b. Tabel Dari

	1	2	3	4	5	6	7	8	9	10
1	-	4	3	4	7	3	6	7	-	4
2	2	-	3	5	34	5	2	-	7	6
3	7	5	-	6	6	4	6	6	4	5
4	5	2	12	-	3	24	15	7	7	6
5	13	4	5	21	-	14	6	7	8	9
6	23	4	2	3	4	-	5	7	7	-
7	6	5	3	2	8	5	-	5	7	5
8	-	4	5	6	3	9	9	-	-	6
9	14	21	12	8	3	3	3	23	-	4
10	2	3	3	1	2	8	5	10	6	-

Penjelasan table

a. Table menuju

Tabel menuju adalah table dari suatu kota menuju kota lain. Misal: 1 merepresentasikan ITB dan 2 merepresentasikan Pasteur. Jalur menuju ITB ke Pasteur dapat melalui bermacam-macam jalur. Table menuju disini adalah table yang merepresentasikan jalur dari ITB ke Pasteur. ITB ke Pasteur dapat melalui Cihampelas namun tidak sebaliknya. Oleh karena itu tanda - menandakan bahwa jalur tersebut tidak ada.

b. Table Dari

Tabel dari merepresentasikan jalur dari suatu tempat ke tempat lain. Menggunakan contoh di atas bahwa dari Pasteur Menuju ITB tidak bisa melalui Cihampelas. Table dari ini artinya bahwa dari Pasteur ke ITB dapat melalui jalur mana saja.

Dengan adanya table seperti ini pada awalnya, jika terdapat suatu perubahan keadaan pada jalan sehingga terjadi perubahan ketetangaan dan perubahan jarak maka tidak semua table perlu diupdate. Hanya simpul-simpul yang terkena dampak saja yang akan

diupdate table nya. Hal ini sangat penting karena akan mempercepat performansi pemecahan masalah menggunakan *dynamic programming*. Secara dasar cara penyelesaian masalah *dynamic programming* adalah sebagai berikut: Misal $f(a,b)$ adalah jarak terdekat yang menyambungkan antara kota a dan kota b. $f(a,b)$ ternyata adalah sebuah fungsi rekursif sebagai berikut:

- $F(a,b) = f(a,c)+f(c,d)+f(d,e)+f(e,a)$
- $F(a,b) = f(a,c) + f(c,r) + f(r,b)$
- $F(a,b) = f(a,g) + f(g,h) + f(h,b)$

Melalui fungsi diatas kita dapat merumuskan memilih jalur mana yang terdekat dan melalui rute yang mana.

4. IMPLEMENTASI ALGORITMA

Seperti yang telah disebutkan di atas bahwa penulis mencoba menyelesaikan masalah penentuan jalur terbaik menggunakan *dynamic programming*. Hal pertama yang dilakukan adalah menentukan basis

$$\text{Dist}(u,v) = \begin{cases} 0 & \text{if } u=v \\ \text{Min}(\text{dist}(u,x) + w(x \rightarrow v)) & \text{otherwise} \end{cases}$$

```

BentukTabel (V,E,w)
For all vertices u memberof V
  For all vertices v memberof V
    If u = v
      Dist[u,v] ← 0
For k ← 1 to V-1
  For all vertices u memberof V
    Dist[u,v] ← min {
      dist[u, v], dist[u, x]+ w(x,v)
    }
  
```

```

CariJalurTerbaik(a,b,V)
If a=b
    Return 0;
Else
    Return
    min(CariJalurTerbaik(a,x,V-x) + w(x,b))

Where x is member of V and a,x has a
line that connect both of them

```

Algoritma di atas adalah dua algoritma yang sederhana yang merepresentasikan teknik pembuatan table awal dan penentuan jalur terbaik. Ketika terjadi perubahan pengupdatean akan dilakukan terhadap pihak-pihak yang mengalami perubahan termasuk simpul-simpul yang kehilangan tetangganya atau yang berubah bobotnya menuju tempat lain.

5. ANALISIS ALGORITMA

Algoritma di atas terbagi atas dua bagian yaitu algoritma pembentuk table dan algoritma penentuan jalur terbaik menuju suatu tujuan. Bobot yang terdapat pada table tidak hanya berupa jarak dari satu simpul ke simpul lain tapi juga keadaan serta kondisi lalu lintas serta jalanan yang ada di lapangan secara real time. Data tersebut dapat diperoleh dari dinas atau provider terkait.

Algoritma pembentuk table hanya berlangsung pada saat pertama kali peluncuran alat dan ketika perubahan langsung di lapangan terjadi. Pada saat awal peluncuran alat, data awal digunakan sebagai acuan untuk penentuan data selanjutnya. Jika terjadi perubahan data seperti penambahan simpul, pengurangan bobot karena jalanan sedang lengang dan lain sebagainya akan dilakukan update secara partial dan berlaku hanya pada simpul-simpul yang terkena dampak secara langsung pada perubahan tersebut.

Hal ini dapat dilakukan karena *dynamic programming* memecah suatu masalah menjadi upa-masalah. Sehingga jika upa-masalah telah berhasil diupdate maka otomatis masalah besar yang sebenarnya pun dapat diselesaikan dengan baik. Hal ini yang membedakan antara algoritma Dijkstra dan *dynamic programming* dimana jika kita menggunakan dijkstra kita harus membuat kembali matriks ketetanggaan secara penuh jika terjadi suatu update.

Algoritma penentuan jalur terpendek digunakan pada saat seorang user meminta sistem ini untuk

menunjukkan jalur untuk menuju suatu rute tertentu. Dengan menggunakan data table yang selalu terupdate dengan kondisi sebenarnya di lapangan maka penunjukan jalur terbaik akan mendapatkan hasil yang maksimal dengan performansi yang baik.

Teknik untuk mendapatkan jalur terbaik adalah dengan memecah jalur antara sumber dan tujuan menjadi beberapa bagian lagi. Misal kita sekarang sedang berada di ITB dan ingin berjalan menuju buah batu. Maka algoritma ini akan memecah masalah ini menjadi jarak dari ITB ke jalan Riau, dari ITB ke Cihampelas dan seterusnya. Dari opsi-opsi jalur yang ada, algoritma ini lalu memilih nilai yang paling minimum yang dapat diperoleh dari kumpulan opsi-opsi yang ada secara rekursif. Dengan kata lain algoritma ini memilih nilai minimum dari nilai-nilai minimum yang ada berdasarkan variable-variabel yang telah disebutkan sebelumnya. Oleh karena itu, dengan menggunakan *dynamic programming* diharapkan penunjukan jalur terbaik dapat diperoleh dengan waktu yang singkat dan ketepatan yang maksimal.

6. PENGIMPLEMENTASIAN DI LAPANGAN

Untuk pengimplementasian di lapangan, algoritma pembentuk table dan algoritma pencarian jalur terbaik sebaiknya dilakukan oleh dua computer yang berbeda dan memiliki spesifikasi yang berbeda pula tentunya. Hal ini perlu dilakukan agar perhitungan pencarian jalur terbaik tidak akan dihambat oleh pembentukan table yang terupdate tiap waktu. Oleh karena itu diperlukan sinkronisasi yang baik antara computer yang mengerjakan algoritma pembentukan table dan algoritma penentuan jalur terbaik.

Pada alat-alat yang menerapkan sistem navigasi ini, terkadang alat yang terpasang pada kendaraan bermotor ini, juga sebagai alat untuk menjalankan algoritma penentuan jalur terbaik ini. Selain menjalankan algoritma, alat ini juga berfungsi menjalankan antar muka yang baik, menampilkan peta serta jalur-jalur yang harus diambil jika ingin menuju ke suatu tujuan.

7. KESIMPULAN

Penggunaan algoritma *dynamic programming* pada pengimplementasian GPS *Car Navigation Sistem* diharapkan dapat memberikan hasil yang cukup baik pada penunjukan jalur terbaik sesuai dengan harapan pengguna kendaraan bermotor. Dengan adanya algoritma ini, diharapkan para pengguna kendaraan bermotor dapat menghindari

kemacetan serta tidak menghabiskan waktu untuk berlama-lama di jalan baik karena macet ataupun karena tersesat. Penggunaan algoritma ini harus didukung dengan sistem pengambilan data yang baik sehingga mampu memberikan informasi kepada alat pengambil keputusan penentuan jalur terbaik yang terbaru secara real time langsung dari lapangan.

8. DAFTAR REFERENSI

- [1]<http://compgeom.cs.uiuc.edu/~jeffe/teaching/algorithms/notes/14-aps.pdf> diakses pada tanggal 8 Desember 2010
- [2] Munir, Rinaldi, "Diktat Kuliah IF3051 Strategi Algoritma" Program Studi Teknik Informatika, Sekolah Tinggi Teknik Informatika, Institut Teknologi Bandung, 2010
- [3] [Dynamic programming - Wikipedia, the free encyclopedia](#) diakses pada 8 Desember 2010
- [4] http://www.ehow.com/about_4728266_mapbased-gps-car-navigation-systems.html diakses pada 6 Desember 2010

9. 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 2010



Muhammad Anis (13508068)