



PENERAPAN ALGORITMA DIJKSTRA UNTUK MENEMUKAN RUTE TERPENDEK DAERAH WISATA DI KABUPATEN BANYUWANGI PADA *LOCATION BASED SERVICE* DI PLATFORM ANDROID

*I Wayan Gede Suma Wijaya^{1, *)}, Eko Heri Susanto²⁾*

Teknik Informatika, STIKOM PGRI Banyuwangi

Jl. Jend. A Yani 82, Banyuwangi, Jawa Timur, 68416

Telp : (0333) 7700669, Fax : (0333) 7700669

E-mail : gedesumawijaya@gmail.com¹⁾

ABSTRAK

Banyuwangi mempunyai beragam tempat wisata yang unik dan menarik. Sebagian besar berada di daerah-daerah pelosok yang dapat ditempuh melalui beberapa rute perjalanan dari pusat kota dengan jarak tempuh yang berbeda-beda. Karena keberadaannya yang jauh dari pusat kota, maka navigasi atau pemanduan menjadi hal penting untuk wisatawan yang belum pernah mengunjungi lokasi-lokasi wisata tersebut.

Selain itu, jarak untuk menuju daerah wisata juga menjadi pertimbangan tersendiri dari sekian banyak jalan yang dapat dilintasi. Dimana wisatawan membutuhkan rute terpendek untuk mencapai tempat wisata yang dituju guna mempersingkat waktu perjalanan. Sehingga dalam 1 hari, wisatawan dapat mengunjungi beberapa daerah wisata sekaligus dengan waktu yang seoptimal mungkin.

Untuk mencari rute yang terpendek ke beberapa daerah wisata, maka bisa diselesaikan dengan permodelan graf menggunakan algoritma dijkstra. Algoritma dijkstra menggunakan prinsip Greedy, yaitu mencari jalur terpendek dari satu titik (vertex) ke titik lainnya yang terhubung. Prinsip ini digunakan untuk memecahkan solusi optimum dalam konteks yang baik, dengan cara mengambil apa saja yang diperoleh sekarang. Algoritma dijkstra ini diterapkan pada sebuah aplikasi *location based service* dengan platform Android yang memanfaatkan Google Map sebagai petanya, sehingga diharapkan para wisatawan dapat dengan cepat untuk mencapai tempat wisata melalui rute yang terpendek dengan sebuah aplikasi *location based service* berbasis platform Android.

Kata kunci: *graf, rute terpendek, location based service, algoritma dijkstra*

PENDAHULUAN

Kebutuhan untuk memperoleh dan mengirim informasi dengan cepat dan mudah telah memacu perkembangan teknologi telekomunikasi. Berbagai macam jaringan telekomunikasi yang menghubungkan berbagai perangkat komunikasi telah dirancang dan diimplementasikan untuk memenuhi kebutuhan akan komunikasi yang cepat dan efisien. Tidak hanya di kota besar, di kota kecil kebutuhan akan teknologi telekomunikasi dewasa ini juga sudah menjadi hal yang mutlak dibutuhkan.

Banyuwangi adalah sebuah kabupaten yang terletak di ujung timur pulau Jawa sehingga sering dijuluki Sunrise of Java dan masuk ke dalam daerah Jawa Timur. Banyuwangi banyak mempunyai potensi-potensi wisata yang unik dan menarik. Mulai dari wisata kuliner, wisata kerajinan, wisata budaya sampai wisata alam yang tidak kalah indahnya dengan daerah yang lain di Indonesia. Dari sekian potensi wisata yang ada, wisata alam menjadi tujuan favorit dari para wisatawan asing maupun lokal. Sebagian besar lokasi wisata yang ada di kabupaten Banyuwangi berada di daerah pelosok dan jauh dari pusat kota. Maka dari itu



navigasi atau pemanduan menjadi hal penting bagi wisatawan yang belum pernah mengunjungi lokasi-lokasi wisata tersebut. Bisa menggunakan bantuan alat khusus yang sudah dibekali dengan teknologi GPS (*Global Positioning System*) maupun *smartphone* yang sudah dilengkapi dengan fitur GPS. Akan ditampilkan sebuah peta lengkap dengan kompas dan koordinat longitude maupun koordinat latitide dimana wisatawan sedang berada (Prahasta, 2001).

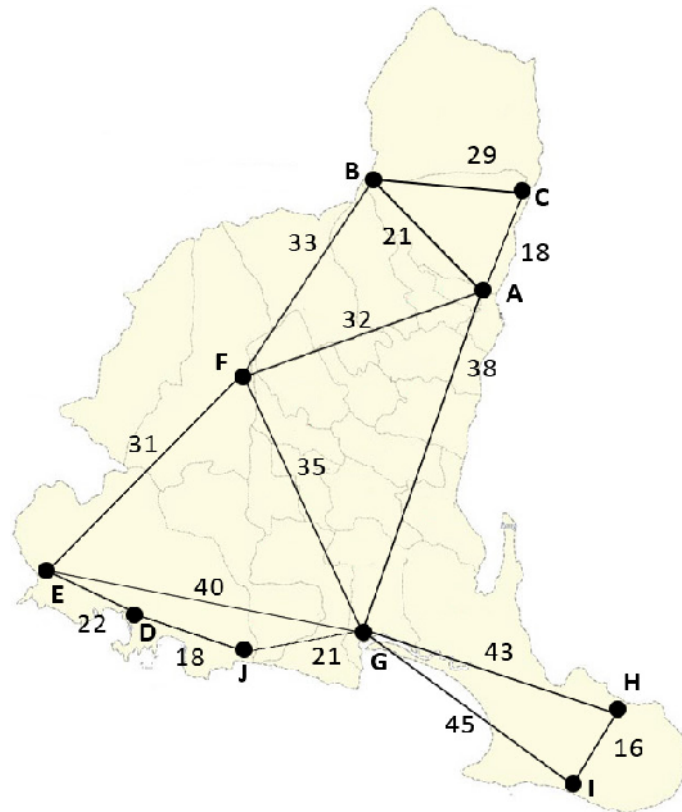
Dengan kemampuan GPS dewasa ini, jarak menuju suatu tempat juga dapat ditampilkan beserta rutenya. Namun rute yang disajikan belum optimal dan belum tentu merupakan rute terpendek. Jarak untuk menuju daerah wisata menjadi pertimbangan tersendiri dari sekian banyak jalan yang tersedia dan dapat dilintasi. Wisatawan membutuhkan rute terpendek untuk mencapai tempat wisata yang dituju guna mempersingkat waktu perjalanan. Sehingga wisatawan dapat mengunjungi beberapa daerah wisata sekaligus dengan waktu yang seoptimal mungkin dalam 1 hari.

Masalah pencarian rute terpendek di atas, bisa diselesaikan dengan permodelan graf menggunakan algoritma dijkstra. Algoritma dijkstra menggunakan prinsip *greedy*, yaitu mencari jalur terpendek dari satu titik (*vertex*) ke titik lainnya yang terhubung. Prinsip ini digunakan untuk memecahkan solusi optimum dalam konteks yang baik, dengan cara mengambil apa saja yang diperoleh sekarang (Alfred. V. Aho, 1995). Vertek-vertik diambil dari beberapa tempat strategis yang dapat dijadikan acuan dan mudah untuk diingat oleh para wisatawan. Algoritma dijkstra ini diterapkan pada sebuah aplikasi *location based service* dengan platform Android yang memanfaatkan Google Map sebagai peta navigasinya.

Pada penelitian ini dihasilkan sebuah aplikasi *location based service* untuk mencari rute terpendek ke beberapa tempat wisata di kabupaten Banyuwangi dengan menggunakan algoritma dijkstra. Sehingga diharapkan para wisatawan dapat dengan mudah dan cepat untuk mencapai tempat wisata yang ingin dituju melalui rute yang terpendek menggunakan *smartphone* Android, yang bisa digunakan sebagai alternatif alat GPS yang mahal dan susah untuk dioperasikan.

METODE

Penelitian ini bersifat *simulatif* dan *deksriptif*. Dengan mengambil beberapa *sample* daerah wisata yang sering menjadi tujuan para wisatawan. Daerah-daerah wisata yang dijadikan *sample* diasumsikan sebagai titik (*vertex*), dimana titik-titik tersebut kemudian dihubungkan oleh garis-garis yang disimulasikan sebagai jalan. Pada tahap awal, titik-titik tersebar pada sebuah peta buta kabupaten Banyuwangi, terhubung oleh garis-garis yang masing-masing diberikan nilai.



Gambar 1. Permodelan Graf Berbobot

Pada permodelan graf berbobot di atas, diambil 10 *sample* tempat wisata. Pada graf berbobot di atas, lintasan merupakan barisan berselangseling dari simpul-simpul dan sisi-sisi. Graf dapat direpresentasikan dengan menggunakan sebuah matriks ketetanggaan (adjacency matrix). Matriks ketetanggaan merepresentasikan hubungan tiap simpul dan bobot yang ada dalam suatu graf. Graf berbobot di atas dengan n buah simpul dinyatakan dengan matriks $M=[m_{xy}]$, dimana :

m_{xy} = bobot sisi (x,y)

$m_{xx} = 0$

$m_{xy} = \infty$, jika tidak adanya sisi dari simpul X ke simpul Y

		\xrightarrow{y}										
			A	B	C	D	E	F	G	H	I	J
$\downarrow x$	(A) Alun-alun kota	A	∞	42	∞	85	63	32	38	81	83	103
	(B) Kawah Ijen	B	21	∞	∞	86	64	33	59	102	104	104
	(C) Pantai Watu Dodol	C	18	29	∞	103	81	50	56	99	101	121
	(D) Teluk Hijau	D	∞	∞	∞	∞	∞	∞	39	82	84	18
	(E) Pantai Sukamade	E	∞	∞	∞	22	∞	∞	40	83	85	40
	(F) Air Terjun Lider	F	∞	∞	∞	53	31	∞	35	78	80	71
	(G) Pantai Bedul	G	∞	∞	∞	∞	∞	∞	∞	43	45	∞
	(H) Alas Purwa	H	∞	∞	∞	∞	∞	∞	∞	∞	16	∞
	(I) Pantai Plengkung	I	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	(J) Pulau Merah	J	∞	∞	∞	∞	∞	∞	∞	21	64	66

Gambar 2. Matriks Ketetanggaan (adjacency matrix)



Selain dengan matriks, perhitungan lintasan terpendek juga dapat ditunjukkan dengan tabel. Dimana penjelasannya lebih lengkap dengan adanya titik awal, titik tujuan, lintasan yang dilalui hingga total bobot nilai minimum yang didapat. Pada tabel di bawah ini diambil satu *sample*, yaitu titik C yang digunakan sebagai titik awal.

Tabel 1. Perhitungan Lintasan Terpendek

Langkah	Awal	Tujuan	Lintasan	Simpul (S)	Panjang Lintasan (D)									
				A B C D E F G H I J	A	B	C	D	E	F	G	H	I	J
0	C	-	-	0 0 0 0 0 0 0 0 0 0	18	29	∞	∞	∞	∞	∞	∞	∞	∞
1	C	C	C	0 0 1 0 0 0 0 0 0 0	18	29	∞	∞	∞	∞	∞	∞	∞	∞
2	C	B	C,B	0 1 1 0 0 0 0 0 0 0	18	29	∞	∞	∞	62	∞	∞	∞	∞
3	C	A	C,A	1 1 1 0 0 0 0 0 0 0	18	29	∞	∞	∞	50	56	∞	∞	∞
4	C	F	C,A,F	1 1 1 0 0 1 0 0 0 0	18	29	∞	∞	81	50	56	∞	∞	∞
5	C	E	C,A,F,E	1 1 1 0 1 1 0 0 0 0	18	29	∞	103	81	50	56	∞	∞	∞
6	C	D	C,A,F,E,D	1 1 1 1 1 1 0 0 0 0	18	29	∞	103	81	50	56	∞	∞	121
7	C	J	C,A,F,E,D,J	1 1 1 1 1 1 0 0 0 1	18	29	∞	103	81	50	56	∞	∞	121
8	C	G	C,A,G	1 1 1 1 1 1 1 0 0 1	18	29	∞	∞	∞	50	56	99	101	∞
9	C	H	C,A,G	1 1 1 1 1 1 1 1 0 1	18	29	∞	∞	∞	50	56	99	101	∞

Pada tabel di atas, untuk simpul (x) yang termasuk ke dalam lintasan yang terpendek diberikan nilai 1. Sedangkan simpul (x) yang tidak termasuk ke dalam lintasan terpendek diberikan nilai 0. Dari hasil perhitungan tabel di atas, didapatkan lintasan terpendek dari :

- C ke B melalui lintasan C,B dengan jarak = 29
- C ke A melalui lintasan C,A dengan jarak = 18
- C ke F melalui lintasan C,A,F dengan jarak = 50
- C ke E melalui lintasan C,A,F,E dengan jarak = 81
- C ke D melalui lintasan C,A,F,E,D dengan jarak = 103
- C ke J melalui lintasan C,A,F,E,D,J dengan jarak = 121
- C ke G melalui lintasan C,A dengan jarak = 56
- C ke H melalui lintasan C,A,G dengan jarak = 99
- C ke I melalui lintasan C,A,G dengan jarak = 101

Permasalahan graf berbobot untuk menentukan lintasan terpendek di atas dapat diselesaikan dengan algoritma dijkstra. Algoritma dijkstra menggunakan prinsip *greedy*, yaitu mencari jalur terpendek dari satu titik (vertex) ke titik lainnya yang terhubung. Prinsip ini digunakan untuk memecahkan solusi optimum dalam konteks yang baik, dengan cara mengambil apa saja yang diperoleh sekarang (Alfred. V. Aho, 1995). Pseudocode untuk aplikasinya seperti gambar di bawah ini :

procedure Dijkstra(input m: matriks, a: simpul awal)

{mencari lintasan terpendek dari simpul awal a ke semua simpul lainnya. Masukan: matriks (m) dari graf berbobot G dan simpul awal a. Keluaran: lintasan terpendek dari a ke semua simpul lainnya }

Deklarasi

s1, s2, ..., sn : int

d1, d2, ..., dn : int



x, y, k, waktu : int

Algoritma

{langkah 0 inialisasi: }

for x ← 1 to n do

sx ← 0

dx ← m a x

endfor

{langkah1:}

sa ← 1 {karena simpul a adalah simpul awal, maka simpul a sudah pasti terpilih dalam lintasan terpendek}

da ← ∞ {tidak ada lintasan terpendek dari simpul a ke a}

{langkah 2, 3, ..., n-1: }

for k ← 2 to n-1 do

y ← simpul dengan sy = 0 dan dy minimal

syj ← 1b{simpul y sudah terpilih ke dalam lintasan terpendek}

{perbarui table d}

for semua simpul x dengan sx = 0 do

if dy + myx < dx then

dx ← dy + myx

endif

endfor

endfor

Gambar 3. Pseudocode Algoritma Dijkstra

HASIL DAN PEMBAHASAN

Algoritma dijkstra diterapkan pada sebuah aplikasi yang dapat berjalan pada platform Android. Android merupakan subset perangkat lunak untuk handphone, smartphone, dan tablet yang meliputi sistem operasi, middleware dan aplikasi kunci yang di release oleh Google (Android Development Guide, 2011). Saat ini disediakan Android SDK (software Development kit) sebagai alat bantu dan API yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Pada uji coba ini, titik-titik wisata dimasukkan ke dalam sebuah array sesuai dengan jarak antara titik yang terhubung.

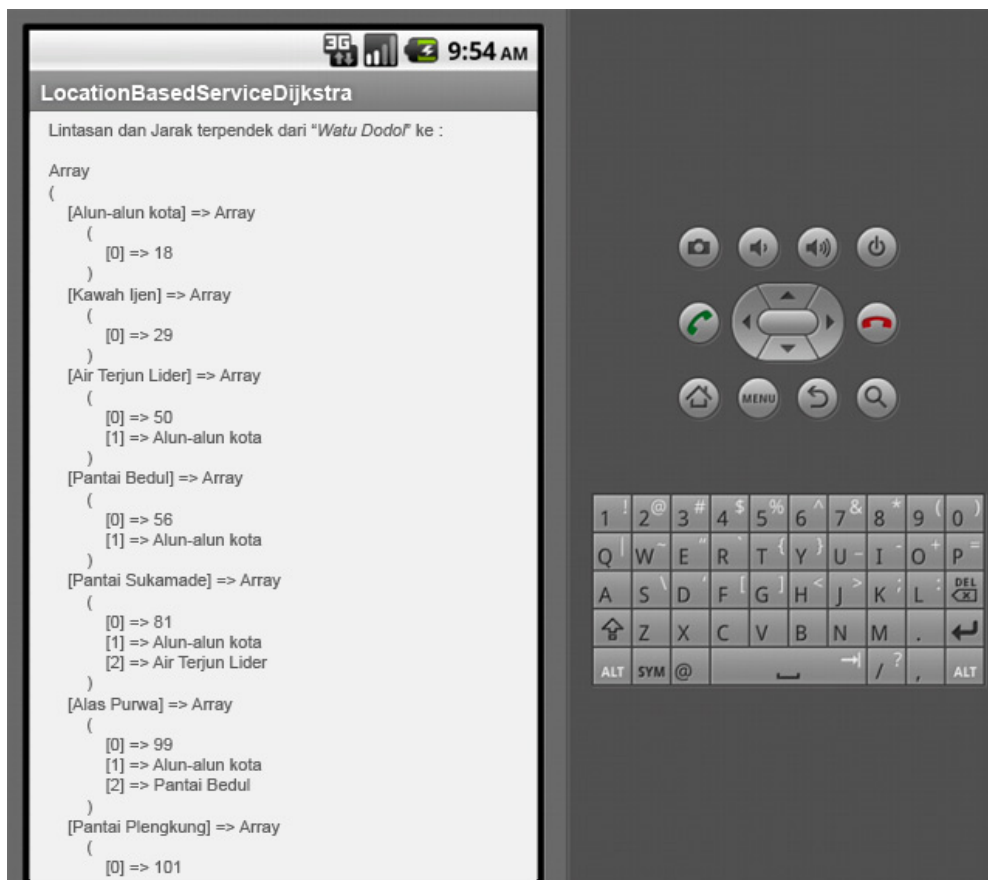
```
neighbors['Pantai Watu Dodol'] = array('Alun-alun kota' => 18, 'Kawah Ijen' => 29);  
neighbors['Kawah Ijen'] = array('Alun-alun kota' => 21, 'Pantai Watu Dodol' => 29);  
neighbors['Alun-alun kota'] = array('Air Terjun Lider' => 32, 'Pantai Bedul' => 38);  
neighbors['Air Terjun Lider'] = array('Pantai Sukamade' => 31, 'Pantai Bedul' => 35);  
neighbors['Pantai Sukamade'] = array('Teluk Hijau' => 22, 'Pantai Bedul' => 40);  
neighbors['Teluk Hijau'] = array('Pulau Merah' => 18);  
neighbors['Pulau Merah'] = array('Pantai Bedul' => 21);  
neighbors['Pantai Bedul'] = array('Alas Purwa' => 43, 'Pantai Plengkung' => 45);  
neighbors['Alas Purwa'] = array('Pantai Plengkung' => 16);  
neighbors['Pantai Plengkung'] = array('Pantai Plengkung' => 0);
```

Gambar 4. Pseudocode Array Titik-Titik Wisata



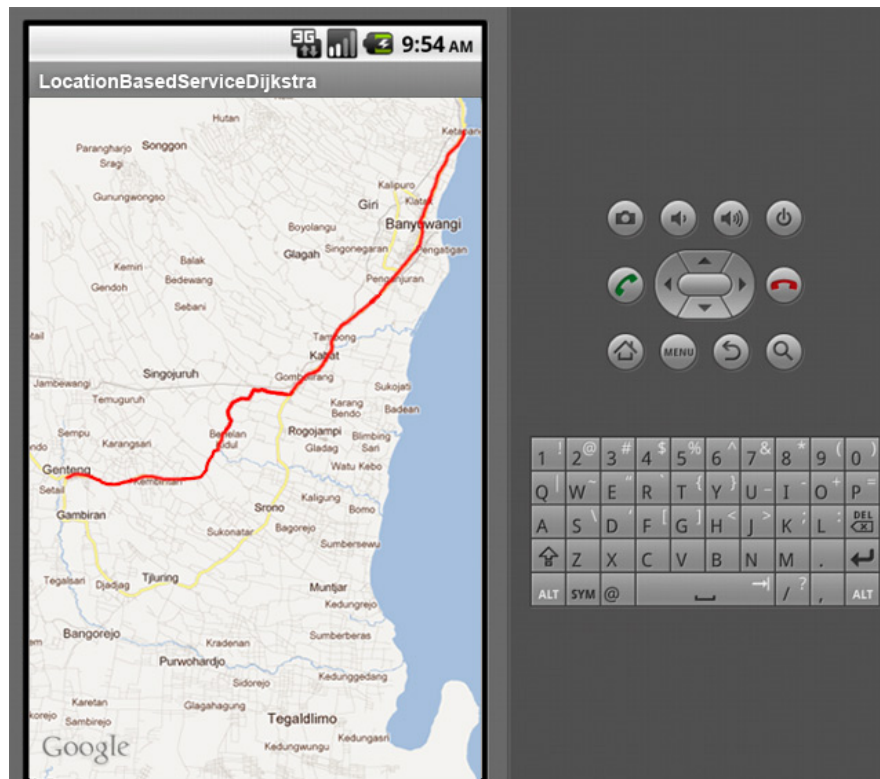
Aplikasi *location based service* ditampilkan dengan menggunakan sebuah *emulator* yang merupakan salah satu fitur yang dipaketkan dalam Android SDK dan mempunyai tampilan yang sama dengan sistem operasi Android yang terdapat di *handphone/smartphone*. Guna mendukung uji coba ini digunakan hardware dan software seperti berikut ini : hardware laptop Intel Core i3, memory RAM 4 GB, sistem operasi Windows 7 Profesional, Eclipse Helios, Android SDK GoogleAPIS 2.2 level 8, serta ADT plugin.

Skenario uji coba yang pertama adalah aplikasi dijalankan dengan memasukkan array titik-titik wisata. Hasil uji coba yang pertama menghasilkan lintasan terpendek ke beberapa titik tujuan dengan menggunakan perhitungan bobot nilai yang minimum. Hasil uji coba ditunjukkan pada gambar 5 dan gambar 6.



Gambar 5. Jarak dan Lintasan Terpendek Untuk Menuju ke Beberapa Daerah Wisata

Selain itu, pada uji coba yang pertama ini berhasil ditampilkan fitur navigasi dengan melalui jalur terpendek yang telah didapatkan sebelumnya. Ditunjukkan dengan memberi warna merah pada lintasan yang terpendek.



Gambar 6. Aplikasi Location Based Service Pada Emulator Berhasil Digunakan Menentukan Lintasan Terpendek

Pada uji coba yang kedua, dihasilkan waktu komputasi yang optimal dengan melakukan uji coba secara berulang-ulang selama 10 kali. Dengan rata-rata waktu yang didapatkan ialah 0,06501. Hasil uji coba yang kedua ditunjukkan pada tabel 2.

Tabel 2. Perhitungan Lintasan Terpendek

Nomor	Awal	Nilai Lintasan Terpendek										Waktu
		A	B	C	D	E	F	G	H	I	J	
1	C	18	29	0	103	81	50	56	99	101	121	0,0621
2	C	18	29	0	103	81	50	56	99	101	121	0,0659
3	C	18	29	0	103	81	50	56	99	101	121	0,0646
4	C	18	29	0	103	81	50	56	99	101	121	0,0666
5	C	18	29	0	103	81	50	56	99	101	121	0,0672
6	C	18	29	0	103	81	50	56	99	101	121	0,0628
7	C	18	29	0	103	81	50	56	99	101	121	0,0655
8	C	18	29	0	103	81	50	56	99	101	121	0,0662
9	C	18	29	0	103	81	50	56	99	101	121	0,0657
10	C	18	29	0	103	81	50	56	99	101	121	0,0635

KESIMPULAN DAN SARAN

Dari hasil uji coba di atas dapat ditarik kesimpulan yaitu aplikasi *location based service* yang menggunakan algoritma dijkstra berhasil digunakan untuk menentukan jalur terpendek dengan waktu komputasi yang optimal. Dengan prinsip *greedy* yang rakus, dimana mengambil apa saja yang diperoleh sekarang, mengakibatkan algoritma ini gagal memberikan



solusi optimal. Hal ini terutama dikarenakan sulit melakukan penentuan urutan lintasan yang akan diperiksa. Maka dari itu, untuk pengembangan dan penelitian ke depannya bisa menggunakan algoritma *ant colony*, dimana proses penentuan lintasan terpendek di dapatkan dari perbandingan lintasan yang sudah pernah dilewati sebelumnya.

DAFTAR PUSTAKA

Alfred. V. Aho, John E. Hopcroft, Jeffrey D. Ullman. , 1995, *Data Structures and Alghorithms*, Addison-Wesley Publishing Company.

Rosen, Kenneth H. 1995, *Discrete Mathematics and Its Applications*, 3rd edition, New York, Mc Graw Hill, Inc..

Prahasta, Eddy., 2001, *Konsep-konsep Dasar Sistem Informasi Geografis*, Informatika, Bandung.

Munir, Rinaldi. 2005. *Diktat Kuliah IF 2251 Strategi Algoritmik*. Laboraorium Ilmu dan Rekayasa Komputasi Departemen Teknik Informatika ITB. Bandung.

Android Development Guide, 2011. Android Development Guide Version 2.3.3 Platform, Available at: <http://www.developer.android.com/guide> [Accessed 10 September 2011].