

SISTEM PENDUKUNG KEPUTUSAN OTOMATISASI *PLANNING* WISATA BERBASIS *WEBSITE* DENGAN MENGGUNAKAN METODE DIJKSTRA

Eko Suryanto, Anton Setiawan Honggowibowo, Nurcahyani Dewi Retnowati

Jurusan Teknik Informatika

Sekolah Tinggi Teknologi Adisutjipto Yogyakarta

informatika@stta.ac.id

Abstract

The application development of website that have a good progress from Website is very important in tourism field where website can promote the interesting and unique of tourism place for tourist. In tourism field website can be important if completed with superiority is tourism planning where tourist can plan their holidays effective and efficient. Dijkstra method is a graph search algorithm is used to solve the problem with a single source shortest path on a graph that produces a shortest path tree. The usage of dijkstra method in decision the tourism planning base website is appropriate because used to plan the holidays based on the route or the sorthest way that has been found by latitude and longitude counting which is very important to decide range from each place, so it can help the tourist.

Keywords: Tourism Planning Automatitaton, Website, Dijkstra Method

1. Pendahuluan

Perkembangan aplikasi *website* yang semakin pesat sejak munculnya teknologi *internet* sangat membantu dalam kemudahan serta kecepatan pengiriman, penyampaian dan penerimaan informasi. Mulai dari perusahaan-perusahaan, sekolah-sekolah, perguruan tinggi, dan lembaga atau organisasi lainnya telah banyak memanfaatkan aplikasi *website* dalam kegiatan penjualan, promosi, belajar dan kegiatan lainnya dimana dibutuhkan pengiriman, penyebaran dan penerimaan informasi sehingga memberikan kemudahan bagi pengguna (*user*) yang membutuhkan.

Website juga sangat berperan pada dunia pariwisata dimana sebuah *website* bisa memasarkan sebuah tempat wisata yang unik dan menawan yang mana akan menarik para wisatawan untuk mengunjunginya. *Website* akan sangat berperan lagi jika didalam sebuah *website* tersebut dilengkapi juga dengan sebuah keunggulan yaitu dimana sebuah wisatawan bisa merencanakan jadwal kunjungan yang harus dikunjunginya. Sehingga wisatawan bisa menggunakan waktu kunjungannya dengan baik. Maka perlu dibuat sebuah aplikasi untuk melengkapi *website* tersebut agar *website* lebih memiliki kegunaan yaitu dengan menambahkan aplikasi perencanaan kunjungan wisata yang menggunakan metode dijkstra.

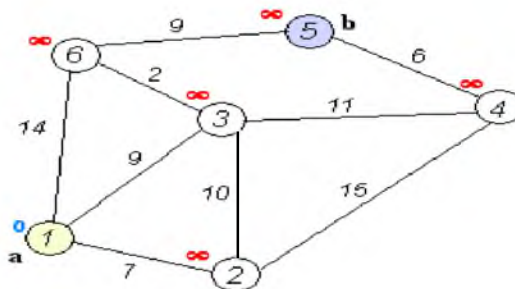
2. Landasan Teori

2.1 Sistem Pendukung Keputusan

Konsep Sistem Pendukung Keputusan (SPK) / *Decision Support System* (DSS) pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah *Management Decision Sistem*. Sistem tersebut adalah suatu sistem yang berbasis komputer yang ditujukan untuk membantu pengambil keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur. Istilah SPK mengacu pada suatu sistem yang memanfaatkan dukungan komputer dalam proses pengambilan keputusan.

2.2 Metode Dijkstra

Algoritma Dijkstra (dinamai menurut penemunya, seorang ilmuwan komputer, Edsger Dijkstra), Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik menggambarkan gedung dan garis menggambarkan jalan, maka algoritma Dijkstra melakukan kalkulasi terhadap semua bobot terkecil dari setiap titik.



Gambar 1 Contoh Keterhubungan AntarTitik Dalam Algoritma Dijkstra

Pada dasarnya, algoritma ini merupakan salah satu bentuk algoritma *greedy*. Algoritma ini termasuk algoritma pencarian graf yang digunakan untuk menyelesaikan masalah lintasan terpendek dengan satu sumber pada sebuah graf yang tidak memiliki cost sisi negatif, dan menghasilkan sebuah pohon lintasan terpendek.

3. Analisa dan Perancangan Sistem

Sistem yang dibuat ini dianalisa agar penerapan teori ke dalam program dapat berjalan sesuai dengan yang diharapkan, yaitu sesuai antara perhitungan yang dilakukan secara manual dengan perhitungan yang di masukkan ke dalam program sehingga apabila dilakukan tes uji coba perhitungan manual maupun dengan perhitungan menggunakan programnya menghasilkan nilai yang tidak jauh berbeda.

3.1 Perancangan Diagram Alir Data

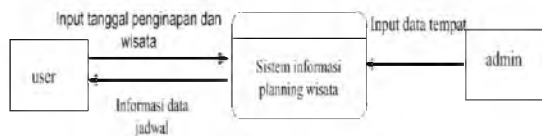
Untuk mengetahui alur data yang ada di dalam suatu sistem, digunakan DAD untuk membuat suatu alur data program sehingga diketahui entitas-entitas yang terlibat, proses yang dibuat untuk menggambarkan dari mana asal data dan ke mana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang

menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. Sehingga dengan DAD ini bisa diketahui dimana data disimpan dan bagaimana transformasi datanya.

3.1.1 Diagram Konteks

Diagram konteks adalah arus data yang berfungsi untuk menggambarkan keterkaitan aliran-aliran data antara sistem dengan bagian-bagian luar (kesatuan luar). Kesatuan luar ini merupakan sumber arus data atau tujuan data yang berhubungan dengan sistem informasi tersebut. Diagram konteks memberikan batasan yang jelas mengenai besaran-besaran entitas yang berada diluar sistem yang sedang dibuat, artinya diagram ini menggambarkan secara jelas batasan-batasan dari sebuah sistem yang sedang dibuat.

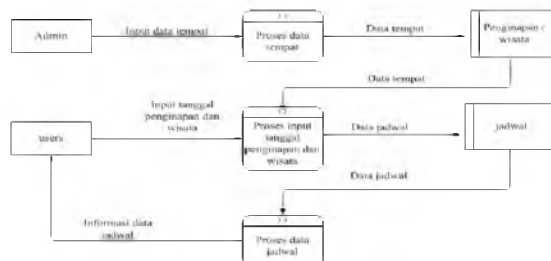
Dibawah ini merupakan diagram konteks untuk sistem pendukung keputusan *planning* otomatisasi wisata berbasis *website* dengan menggunakan metode dijkstra. Terdapat 2 entitas yang terdiri dari admin dan *user*. Admin bertugas menginput data tempat yang di masukan ke sistem *planning* otomatisasi wisata kemudian *user* mengaksesnya dengan cara menginputkan tanggal wisata dan penginapan kemudian masuk ke sistem yang mana setelah itu *user* akan mendapatkan informasi data jadwal wisatanya. Berikut diagram konteks ditunjukan pada gambar 2.



Gambar 2 Diagram konteks

3.1.2 DAD Level 1

Diagram level 1 adalah diagram yang menggambarkan proses dari data flow diagram. Diagram level 1 memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukkan tentang fungsi-fungsi utama atau proses yang ada, aliran data, dan *eksternal entity*.



Gambar 3 DAD Level 1

3.1.3 Flowchart Sistem

Dalam membuat suatu sistem yang sistematis, diperlukannya suatu *Flowchart* sistem untuk melihat proses aliran data yang ada dari awal sampai selesai.



Gambar 4 Flowchart system

4. Implementasi Sistem

Implementasi sistem merupakan tahap pendiskripsian suatu sistem agar sistem tersebut siap untuk dioperasikan sesuai dengan yang diharapkan. Tahap implementasi ini akan membahas mengenai desain *input* data, desain proses data, desain *output* sistem serta analisis hasil yang berupa pengujian sistem tersebut secara manualnya. Dalam implementasi sistem ini dapat diketahui cara kerja suatu sistem yang dijalankan, apakah telah berjalan dengan baik atau tidak.

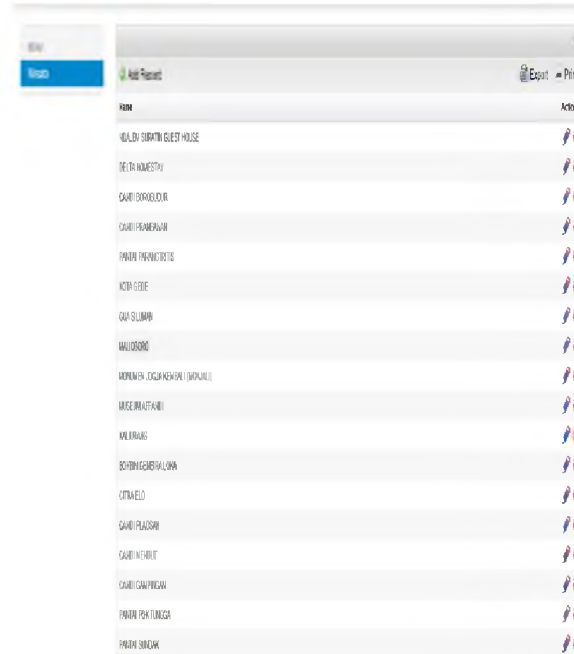
4.1 Implementasi Menu Login





































Tampilan pertama dari form ini menu *login*. Merupakan input untuk mengisi *username* dan *password*. Jika benar dalam mengisi *username* dan *password*, maka klik tombol *login* untuk masuk dalam aplikasi sedangkan tombol kembali ke menu akan menuju ke menu *user*. Tampilan pertama dari form ini hanya berbentuk *text*. Menu *login* merupakan *input* untuk mengisi *username* dan *password*. Jika benar dalam mengisi *username* dan *password*, maka klik tombol *login* untuk masuk dalam aplikasi dan tombol kembali ke menu untuk kembali ke menu utama yaitu menu *user*. Tampilan form *login*, dapat dilihat pada Gambar 5.

Gambar 5 Menu Login

4.1.1 Implementasi Halaman Admin

Didalam halaman admin ini merupakan sebuah tampilan utama dari halaman admin dimana didalam halaman admin ini terdapat postingan yang telah admin *input*. Didalam halaman admin ini juga seorang admin bisa mengedit ataupun menghapus hasil postingannya tidak hanya itu seorang admin juga bisa menambah postingannya melalui *add record*. Tampilan form halaman admin, dapat dilihat pada Gambar 6.

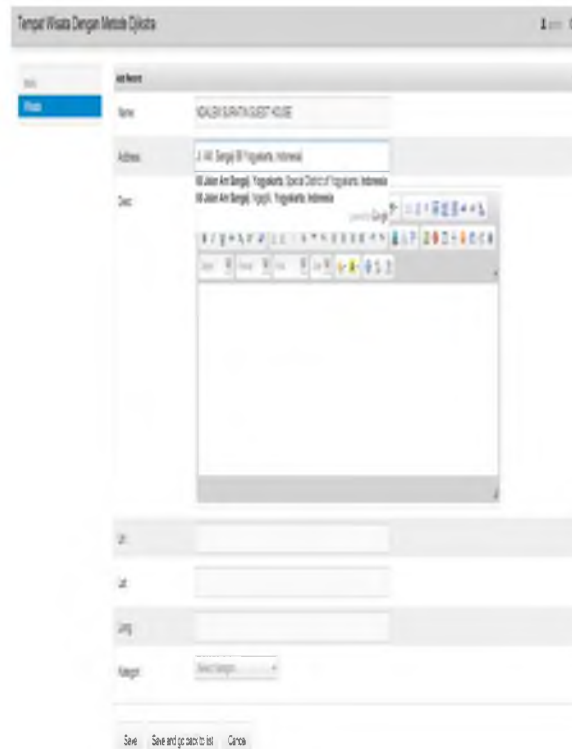


NAME	Actions
HEALIN SURATHI GUEST HOUSE	 
DE TR WIDESTY	 
CAUTI DORODEUR	 
CAUTI PERANGKAW	 
PAWANI PERANGKAW	 
KORA GEDE	 
GAU SILIMAN	 
MAULUSONE	 
MORIMAN LOGAN KAPPA (BANGKALU)	 
MAREMAUTRANI	 
KALIRAGAS	 
BOHINGSERBALAWA	 
CITRA ELIO	 
CAUTI PLASSON	 
CAUTI INTARIT	 
CAUTI CAWIRAW	 
PAWANI PERANGKAW	 
PAWANI PERANGKAW	 

Gambar 6 Halaman Admin

4.1.2 Implementasi Add Record

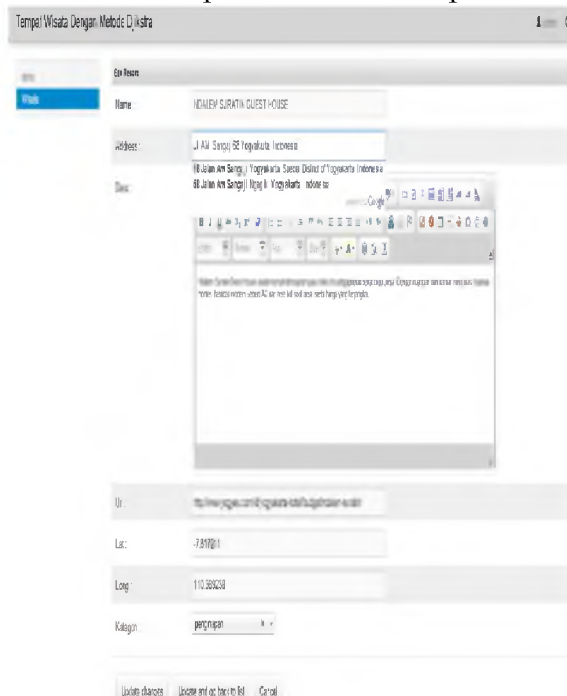
Didalam *add record* ini seorang admin bisa menambah postingannya sehingga ada tempat atau penginapan baru yang ingin ditambah seorang admin tinggal mudah menambahkannya. Didalam *add record* ini terdapat beberapa inputan seperti nama, *address*, *desc*, *url*, *lat*, *long* dan kategori. Untuk *lat* dan *long* akan secara otomatis muncul ketika memasukkan *address* yang mana untuk menentukan *lat* dan *long* langsung terkoneksi dengan Google Maps. Setelah seorang admin menginputkan semua admin tinggal *button save*, *save and back to list* atau *cancel*. Tampilan *add record* dapat dilihat pada Gambar 7.



Gambar 7 Add Record

4.1.3 Implementasi Menu Edit

Didalam menu edit ini seorang admin bisa mengubah info tentang pariwisata yang telah di simpan. Sehingga jika ada kesalahan dalam memosting info pariwisata tersebut admin tidak kebingungan. Admin tinggal mengubah dimana letak kesalahannya. Untuk tampilan menu edit ini sama dengan tampilan *add record*. Tampilan menu edit dapat dilihat pada Gambar 8.



Gambar 8 Menu Edit

4.2 Implementasi Menu Utama

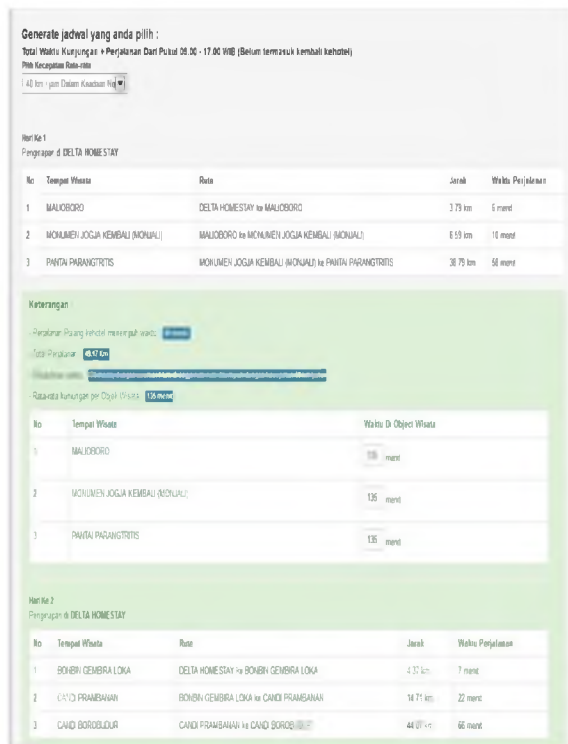
Menu utama akan muncul ketika aplikasi mulai dijalankan. Pada menu utama ini terdapat beberapa menu diantaranya yaitu tanggal digunakan untuk mengisi kapan mulai liburannya dan berakhir sampai kapan. Menu hotel digunakan untuk memilih penginapan yang akan ditempati di menu hotel ini disediakan beberapa hotel yang ada di Yogyakarta sedangkan untuk menu wisata sama dengan menu hotel dimana *user* tinggal memilih tempat - tempat wisata yang ada di Yogyakarta setelah itu *user* tinggal menekan *button submit*. Untuk menu utama bisa dilihat Gambar 9.



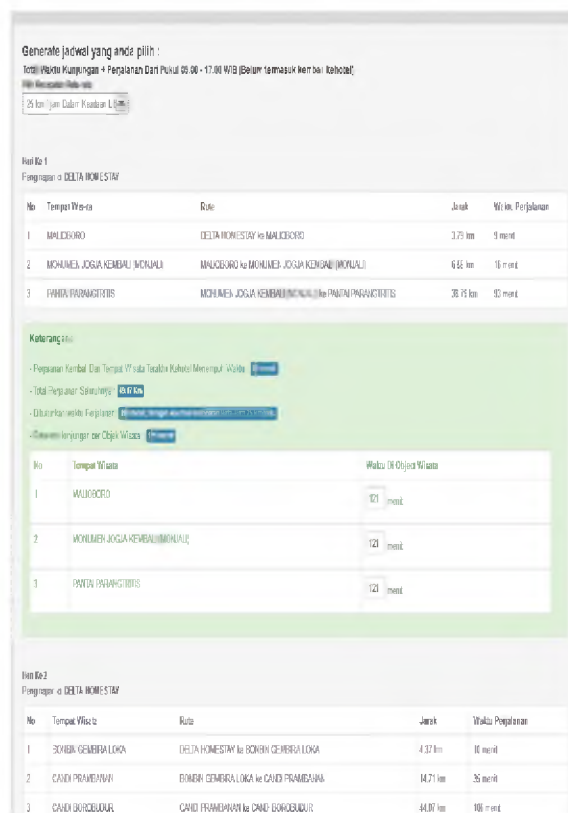
Gambar 9 Menu Utama

4.2.1 Implementasi Menu Hasil Rencana Wisata

Menu rencana wisata merupakan menu hasil dari inputan yang telah di inputkan oleh *user* pada menu utama. Menu rencana wisata ini berisikan rencana kunjungan wisata yang akan dikunjungi. Selain itu juga terdapat waktu kunjungan berapa jam di tempat wisata tersebut. Didalam menu wisata ini ada dalam keadaan perjalanan lancar dan ada yang waktu perjalanan dalam keadaan liburan. Dalam keadaan lancar rata-rata jarak tempuhnya yaitu sekitar 40 km/jam sedangkan dalam keadaan liburan yaitu berkisar 25 km/jam. Total waktu kunjungan dan perjalanannya belum termasuk kembali ke hotelnya. Untuk menu hasil rencana wisata dapat dilihat Gambar 10.



Gambar 10 Menu Hasil Rencana Wisata Dalam Keadaan Lancar



Gambar 11 Menu Hasil Rencana Wisata Dalam Keadaan Liburan

4.3 Analisa Hasil

Hasil sistem yang didapatkan harus diuji nilainya dengan membandingkan hasil pengujian sistem secara manual, sehingga dengan perbandingan ini akan didapatkan hasil sistem yang benar-benar diakui karena sesuai dengan hitungan manualnya dan dapat dibandingkan dengan hasil dari perhitungan aplikasi.

Tahap perhitungan yang diperlukan adalah memilih beberapa tempat wisata yang akan dikunjungi sekaligus tempat penginapannya. Setelah memilih tempat wisata dan penginapannya maka dilakukan pencarian jarak dari masing-masing pariwisata ke hotel dan jarak dari pariwisata ke pariwisata. Setelah mengetahui jaraknya maka langkah selanjutnya yaitu menghitung dengan metode Dijkstra yang mana metode ini menentukan rute terpendek setelah dilakukan perhitungan dengan metode dijkstra maka akan memperoleh hasil pariwisata mana saja yang harus dikunjungi pertama kali dan kunjungan berikutnya menurut rute terpendek dari penginapan ke wisata dan wisata ke wisata selanjutnya.

Sebagai contoh analisa untuk melakukan perhitungan secara manual, diambil contoh seorang wisatawan berlibur selama 2 hari. Tempat wisata yang harus dikunjunginya Candi Borobudur, Monumen Jogja Kembali, Bonbin Gembiraloka, Candi Prambanan, Pantai Parangtritis, Malioboro sedangkan untuk penginapan yang dipilihnya yaitu Delta Homestay. Setelah mengetahui tempat wisata mana saja yang dikunjungi dan penginapannya maka selanjutnya dilakukan pencarian jarak dari masing-masing pariwisata ke penginapan. Untuk mencari jarak dari masing masing pariwisata ke hotel dan pariwisata ke pariwisata dengan cara mengetahui *latitude* dan *longitude*. Cara mengubah *latitude* dan *longitude* menjadi Kilometer seperti persamaan 3.1 :

$$d = \text{acos}(\sin(\text{latitude1}) \times \sin(\text{latitude2}) + \cos(\text{latitude1}) \times \cos(\text{latitude2}) \times \cos(\text{longitude1} - \text{longitude2})); \text{Rad}$$

Keterangan :

latitude1 = *latitude* awal

latitude2 = *latitude* tujuan

longitude1 = *longitude* awal

longitude2 = *longitude* tujuan

Tabel 1 Latitude longitude pariwisata dan Penginapan

Nama wisata	latitude	longitude
1. Candi Borobudur	-7.792253	110.365709
2. Kebun binatang gembiraloka	-7.805519	110.398196
3. Munumen jogya kembali	-7.744672	110.368627
4. Candi Prambanan	-7.750923	110.491341
5. Pantai Parangtritis	-8.017484	110.315231
6. Malioboro	-7.792253	110.365709
7. Delta Homestay	-7.819422	110.369554

Setelah melakukan perhitungan merubah Latitude dan *Longitude* menjadi kilometer maka diperoleh jarak Jarak masing-masing pariwisata ke hotel sebagai berikut :

1. Candi Borobudur - Delta Homestay = 42,3 km
2. Kebun binatang gembiraloka - Delta Homestay = 4,37 km
3. Munumen jogya kembali - DeltaHomestay = 7,4 km
4. Candi Prambanan - Delta Homestay = 17,5 km
5. Pantai Parangtritis - Delta Homestay = 27,3 km
6. Malioboro - Delta Homestay = 3,79 km

Dengan ditemukan jarak antara pariwisata ke hotel maka metode dijkstra berjalan untuk memecahkan masalah tentang jarak terpendek yang akan digunakan untuk rencana kunjungan yang akan di kunjunginya.

Misalkan diambil dari data yang di atas di peroleh untuk 2 hari kunjungan yang mana kunjungan pertama hari pertama dan kunjungan pertama hari kedua diperoleh dapat dilihat pada table 2.

Tabel 2 Kunjungan hari pertama dan kedua

Hari pertama	Hari kedua
Delta Homestay - Malioboro	Delta Homestay - Kebun Binatang Gembiraloka

Setelah mengetahui kunjungan dari penginapan ke tempat wisata yang harus di kunjungi pertama dalam waktu kunjungan 2 hari maka langkah selanjutnya yaitu mencari jarak dari kunjungan pertama yaitu Malioboro dan Kebun Binatang Gembiraloka ke tempat wisata selanjutnya yang terdekat.

Pencarian kunjungan kedua untuk hari pertama

1. Malioboro - Candi Borobudur = 36,03 km
2. Malioboro - Candi Prambanan = 14,71 km
3. Malioboro - Pantai Parangtritis = 27,05 km
4. Malioboro - Monumen Jogja Kembali = 6,59 km

Pencarian kunjungan kedua untuk hari kedua

1. Kebun Binatang Gembiraloka - Candi Borobudur = 43,6 km
2. Kebun Binatang Gembiraloka - Candi Prambanan = 14,71 km
3. Kebun Binatang Gembiraloka - Pantai Parangtritis = 31,7 km
4. Kebun Binatang Gembiraloka - Monumen Jogja Kembali = 16,4 km

Pencarian kunjungan ketiga untuk hari pertama

1. Monumen Jogja Kembali - Candi Borobudur = 39,1 km
2. Monumen Jogja Kembali - Pantai Parangtritis = 38,79 km

Pencarian kunjungan ketiga untuk hari kedua

1. Candi Prambanan - Candi Borobudur = 44,7 km

Tabel 3 Kunjungan wisata selama 2 hari

Hari Pertama	Hari Kedua
Delta Homestay ke Malioboro	Delta Homestay ke Kebun Binatang Gembiraloka
Malioboro ke monument Jogja Kembali	Kebun Binatang Gembiraloka ke Candi Prambanan
Monument Jogja Kembali ke Pantai Parangtritis	Candi Prambanan ke Candi Borobudur

Untuk kunjungan wisata selama 2 hari yang harus dikunjungi wisatawan menurut hasil analisa yang telah dibuat maka kunjungan untuk hari pertama yaitu penginapan Delta homestay ke Malioboro lanjut Malioboro ke Monumen Jogja Kembali dan diteruskan ke Pantai Parangtritis sedangkan untuk hari keduanya dari penginapan menuju Kebun Binatang Gembiraloka lanjut ke Candi Prambanan dan di teruskan ke Candi Borobudur. Hasil tersebut diperoleh dari jarak terpendek dari tempat asal. Setelah mengetahui tujuan wisata yang akan dikunjunginya dan jarak tempuh berapa kilometernya saatnya menghitung waktu tempuh perjalanannya tersebut. Untuk mengetahui waktu perjalanannya sebagaimana dicantumkan pada persamaan 3.2 :

$$t = \frac{s}{v}$$

Dengan ketentuan:

s = Jarak yang ditempuh (m atau km)

v = Kecepatan (km/jam atau m/s)

t = Waktu tempuh (jam atau sekon)

Setelah mengetahui rencana kunjungan wisata kemana saja dan berapa jarak tempuh perjalanannya sekarang menghitung untuk waktu tempuh perjalanan setiap kilometernya. Untuk asumsi kecepatannya yaitu 25 km/jam dalam keadaan liburan dan 40 km/jam dalam keadaan lancar.

Contoh perhitungannya sebagai berikut :

Delta Homestay ke Malioboro = 3.79 km

$$t = \frac{3.79}{25 \text{ km / jam}} = 0.1516 \text{ jam} \times 60 = 9 \text{ menit}$$

$$t = \frac{3.79}{40 \text{ km / jam}} = 0.09475 \text{ jam} \times 60 = 6 \text{ menit}$$

Malioboro ke Monumen Jogja Kembali = 6.59 km

$$t = \frac{6.59}{25 \text{ km / jam}} = 0.2636 \text{ jam} \times 60 = 16 \text{ menit}$$

$$t = \frac{6.59}{40\text{km} / \text{jam}} = 0.16475 \text{ jam} \times 60 = 10 \text{ menit}$$

Monumen Jogja Kembali ke Pantai Parangtritis = 38.79 km

$$t = \frac{38.79}{25\text{km} / \text{jam}} = 1.5516 \text{ jam} \times 60 = 93 \text{ menit}$$

$$t = \frac{38.79}{40\text{km} / \text{jam}} = 0.96975 \text{ jam} \times 60 = 58. \text{menit}$$

Delta Homestay ke Kebun Binatang Gembiraloka = 4.37 km

$$t = \frac{4.37}{25\text{km} / \text{jam}} = 0.1748 \text{ jam} \times 60 = 10 \text{ menit}$$

$$t = \frac{4.37}{40\text{km} / \text{jam}} = 0.10625 \text{ jam} \times 60 = 7 \text{ menit}$$

Kebun Binatang Gembiraloka ke Candi Prambanan = 14.71 km

$$t = \frac{14.71}{25\text{km} / \text{jam}} = 0.5884 \text{ jam} \times 60 = 35. \text{menit}$$

$$t = \frac{14.71}{40\text{km} / \text{jam}} = 0.36775 \text{ jam} \times 60 = 26. \text{menit}$$

Candi Prambanan ke Candi Borobudur = 44.07 km

$$t = \frac{44.07}{25\text{km} / \text{jam}} = 1.7625 \text{ jam} \times 60 = 106. \text{menit}$$

$$t = \frac{44.07}{40\text{km} / \text{jam}} = 1.10175 \text{ jam} \times 60 = 66. \text{menit}$$

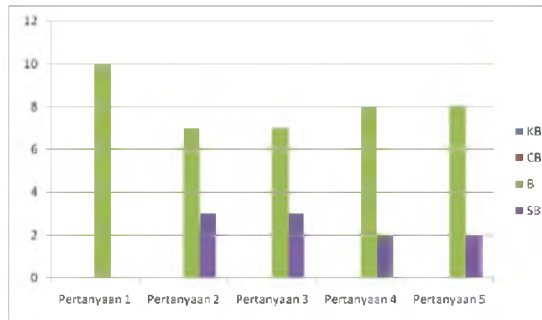
4.4 Uji Coba Pengguna

Uji coba pengguna terhadap program aplikasi digunakan untuk mengetahui kenyamanan *user* dalam menggunakan program yang telah dibuat. Untuk itu dilakukan uji coba dengan menggunakan kuesioner yang ditujukan langsung kepada *user*.

User bisa mengoprasikan secara langsung program aplikasi yang telah dibuat sehingga *user* bisa memberikan penilaian sesuai dengan pertanyaan yang ada pada keusioner yang telah disediakan.

Tabel 4 Hasil kuisisioner

No	Kuisisioner	Jumlah Nilai			
		KB	CB	B	SB
1.	Apakah tampilan aplikasi tersebut menarik menurut Anda?	0	0	10	0
2.	Apakah menurut Anda aplikasi tersebut mudah untuk dipergunakan?	0	0	7	3
3.	Apakah menurut Anda aplikasi tersebut membantu dalam liburan Anda?	0	0	7	3
4.	Apakah menurut Anda aplikasi tersebut efisien bisa menghemat waktu dan biaya berwisata Anda?	0	0	8	2
5.	Apakah menurut Anda hasil dari penggunaan aplikasi tersebut dapat menjadi acuan untuk berwisata Anda?	0	0	8	2
Jumlah		0	0	40	10
Persentase		0%	0%	80%	20%



Gambar 12 Grafik hasil kuisisioner

Melihat data hasil kuesioner yang telah diperoleh setelah dihitung mendapatkan hasil untuk nilai KB = 0%, CB = 0%, B = 80% dan SB = 20%. Hasil tersebut diperoleh dari penjumlahan data dari 10 *user* yang telah mengisi kuesioner, kemudian dibagi dengan 50 dan dikalikan 100%. Nilai 50 diperoleh dari penjumlahan *user* yang mengisi kuesioner yaitu 10 *user* dikalikan dengan jumlah pertanyaan yaitu 5 sehingga diperoleh nilai 50.

5 Kesimpulan

Kesimpulan yang diperoleh dari penelitian ini adalah :

1. Metode Dijkstra dapat digunakan untuk pengambilan keputusan otomatisasi perencanaan wisata berbasis *website* dengan bantuan *latitude* dan *longitude* yang diambil melalui Google Maps.
2. Dari hasil survey 80% pengguna menyatakan aplikasi dinilai baik, sedangkan 20% pengguna lainnya menyatakan aplikasi sangat baik.

Program aplikasi ini masih dapat dikembangkan lagi menjadi sebuah aplikasi yang lebih kompleks lagi, antara lain:

1. Aplikasi sistem pendukung keputusan planning wisata untuk kedepannya bukan hanya untuk satu daerah Yogyakarta saja tetapi untuk beberapa daerah yang ada di Indonesia.
2. Aplikasi sistem perencanaan wisata ini bisa dikembangkan dengan menambahkan fitur GPS (*Global Positioning System*).

6 Daftar Pustaka

C.Giancoli, Douglas, 2001, *Fisika Edisi Kelima Jilid 1*. CV. Erlangga, Jakarta.

Nugroho, Adi, 2011, *Perancangan dan implementasi sistem Basis Data*. CV. Andi, Yogyakarta.

Siang, Jong Jek, Drs., Msc., 2009, *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. CV. Andi, Yogyakarta.

Syafii.M, 2005, *Panduan Membuat Aplikasi Database dengan PHP 5 Mysql Postgresql Oracle*. CV. Andi, Yogyakarta.

Turban, Efram, Aronson, Jay E, dan Peng-Liang, Ting, 2003, *Decision Support Systems and Intelligent Systems (Sistem Pendukung Keputusan dan Sistem Cerdas)*. CV. Andi Offset, Yogyakarta.

Widodo, Aris Puji., 2007, *Simulasi Lintasan Jalur Terpendek Algoritma Dijkstra Berbasis Extensible Markup Language (XML)*, Jurnal Matematika Vol 10 No 1 April 2007, Universitas Diponegoro, ISSN 1410-8518.