

PENENTUAN RUTE TERPENDEK JALAN DAN LOKASI PARIWISATA DI KOTA SURAKARTA MENGGUNAKAN ALGORITMA DIJKSTRA DAN WAP PADA *HANDPHONE*

Siti Nandiroh¹, Haryanto².

^{1,2}Universitas Muhammadiyah Surakarta
Jl. Ahmad Yani Tromol Pos I pabelan Surakarta 57102
e-mail: iroh@rocketmail.com, haryanto_solo@yahoo.com

Abstrak

Perkembangan mobile communication (handphone) sejak tahun 1995 berkembang dengan sangat pesat. Salah satu fasilitas yang menunjang perkembangannya adalah penggunaan WAP (Wireless Application Protocol), yaitu suatu protocol untuk terhubung ke internet melalui handphone. WAP menyediakan dukungan bagi pengembang aplikasi agar bisa memanfaatkan koneksi dengan jaringan pada sistem komunikasi yang real time.

Pada penelitian ini dikembangkan sistem baru yang memanfaatkan teknologi komputer dan teknologi komunikasi, khususnya yang berbasis WAP serta algoritma Dijkstra, sebagai navigasi bagi pengguna jalan yang berkunjung serta ingin berwisata di kota Surakarta. Dalam sistem navigasi perjalanan tersebut terdapat beberapa informasi berkaitan dengan rute perjalanannya antara lain informasi kondisi jalan dan mengetahui rute terpendek dari jalan ke jalan atau dari lokasi ke lokasi, demikian juga bisa diketahui jalur alternatif, apabila terdapat hambatan pada jalan yang hendak dituju.

Kata kunci : *algoritma dijkstra; handphone; jalan; teknologi; WAP.*

Pendahuluan

Salah satu perkembangan teknologi yang cepat adalah di bidang telepon seluler (*handphone*). Fasilitas yang sekarang sedang populer adalah WAP (*Wireless Application Protocol*), merupakan suatu protocol untuk terhubung ke dunia internet melalui *handphone* (Djojo, M.,2000).

Berdasarkan perkembangan teknologi komputer dan jaringan, maka pada tulisan ini di kembangkan suatu sistem dengan mengkombinasikan kedua teknologi tersebut dengan teknologi WAP dan menggunakan algoritma *Dijkstra*, yang berkaitan dengan rute jalan dan lokasi pariwisata di kota Surakarta, karena banyak jalur alternatif dan jalan searah. Sehingga rute jalan merupakan alat penunjuk yang sangat penting untuk mencari lokasi, dan saat ini banyak *software-software* untuk pencarian rute jalan tetapi *software* tersebut masih beroperasi di PC (*Personal Computer*), sedangkan pencarian rute jalan yang dilakukan pengguna jalan tidak berada di depan komputer melainkan dalam perjalanan (Arunanto, dkk, 2006).

Dengan dasar tersebut, WML dijadikan sebagai media untuk memberikan informasi dalam pencarian rute jalan, karena menggunakan koneksi WAP atau GPRS (*General Packet Radio System*) dengan biaya yang relatif murah. Aplikasi dapat selalu terhubung dengan server namun biaya hanya dikenakan apabila terjadi pengiriman atau penerimaan data yang menjadikan biaya menjadi relatif murah. Sehingga ini bisa dijadikan alternatif daripada harus membuka berlembar-lembar peta yang digunakan dalam mencari rute jalan. Diharapkan dengan digunakannya fasilitas WML ini user bisa mendapatkan segala macam dan bentuk informasi yang dibutuhkan dengan mudah dan cepat, termasuk informasi tentang kondisi lalu lintas.

Algoritma Dijkstra

Algoritma mencari lintasan terpendek yang pernah ditulis orang. Algoritma yang paling terkenal adalah Algoritma *Dijkstra* (sesuai dengan nama penemunya). Aslinya, algoritma *Dijkstra* diterapkan untuk mencari lintasan terpendek pada graf berarah. Namun, algoritma ini juga benar untuk graf tak berarah (Narayanan, S, 2003). Contoh penerapan Algoritma *Dijkstra* adalah sebagai berikut, misalkan sebuah graf berbobot n buah simpul dinyatakan dengan matriks ketetanggaan $M=[m_{ij}]$, yang dalam hal ini,

m_{ij} = bobot sisi(i,j) (pada graaf berarah $m_{ij}=m_{ji}$)

$m_{ij} = 0$

$m_{ij} = \infty$, jika tidak ada sisi dari simpul i ke simpul j

Selain matrik M , kita juga menggunakan larik $S = [s_i]$ yang dalam hal ini,

$s_i = 1$, Jika simpul i termasuk ke dalam lintasan terpendek.

$s_i = 0$, Jika simpul i tidak termasuk ke dalam lintasan terpendek.

dan larik/ tabel $D = [d_i]$ yang dalam hal ini,

d_i = panjang lintasan dari simpul awal ke simpul i

Algoritma Lintasan terpendek Dijkstra (*mencari lintasan terpendek dari simpul a ke semua simpul lain*)

Langkah 0 (inisialisasi)

- inisialisasi $s_i = 0$ dan $d_i = m_{ai}$ untuk $i = 1, 2, \dots, n$

Langkah 1:

- isi s_a dengan 1 (karena simpul a adalah simpul asal lintasan terpendek, jadi sudah pasti terpilih)
- isi d_a dengan ∞ (tidak ada lintasan terpendek dari simpul a ke a)

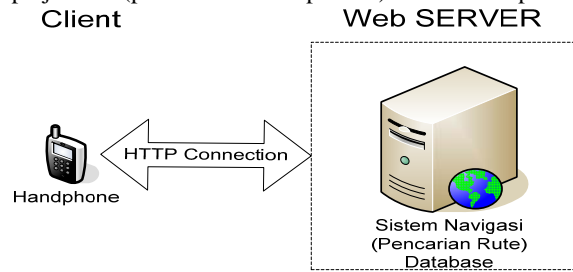
Langkah 2,3,...,n-1:

- cari j sedemikian sehingga $s_j = 0$ dan $d_j = \min \{d_1, d_2, \dots, d_n\}$
- isi s_j dengan 1
- perbarui d_i , untuk $i = 1, 2, 3, \dots, n$ dengan: d_i (baru)
 $= \min \{d_i \text{ (lama)}, d_j + m_{ij} \}$

Metode Penelitian

Arsitektur Umum Perancangan Sistem

Sistem ini mempunyai dua sisi, yaitu sisi *client* dan sisi *server*. Sisi *client* merupakan user yang menggunakan *handphone* yang sudah ada aplikasi WML untuk navigasi perjalanan (pencarian rute). Sisi *server* terdiri dari web server yang menggunakan bahasa pemrograman PHP dan server database yang menggunakan MySQL. Ketika user memilih rute yang dikehendaki maka *handphone* akan mengirimkan *request* ke web server melalui *http connection*, Seperti halnya mengetikkan *URL* untuk mengakses *website*. Maka web server akan mencari rute yang terpendek dengan mengambil data yang tersedia di database server dan hasilnya dikirim kembali ke *handphone*. Aplikasi WML bertugas membaca hasil yang diterima dan menampilkan ke user. Adapun gambaran dari arsitektur sistem navigasi perjalanan (pencarian rute terpendek) ini adalah seperti Gambar 1



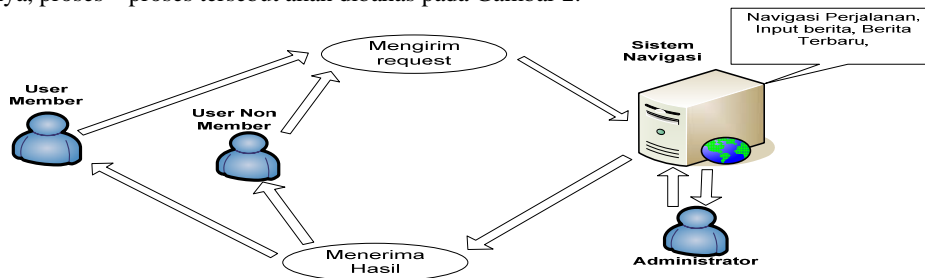
Gambar 1. Arsitektur sistem

Perancangan Proses

Dalam perangkat lunak ini, terdapat beberapa proses yang akan dilakukan. Diantaranya adalah:

1. Navigasi perjalanan (Pencarian rute terpendek):
 - a. Dari lokasi ke lokasi
 - b. Dari jalan ke jalan
2. Input Berita
3. Berita Terbaru
4. Pengiriman dan Penerimaan data pada aplikasi WML.

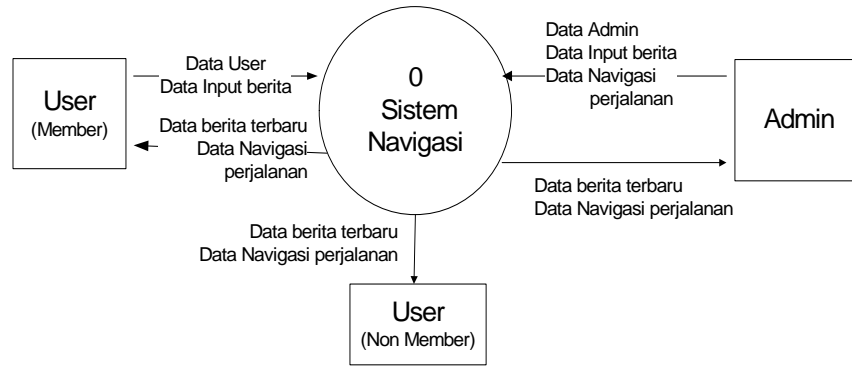
Untuk jelasnya, proses – proses tersebut akan dibahas pada Gambar 2.



Gambar 2 Proses yang dibutuhkan

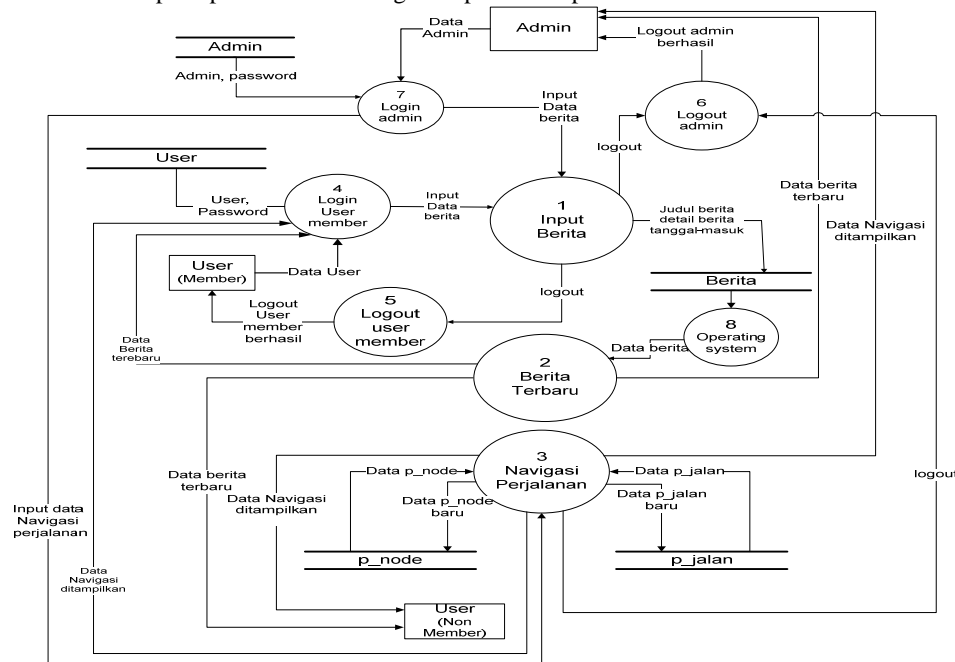
Data Flow Diagram (DFD)

Diagram arus data (*data flow diagram*), atau DFD, adalah suatu gambaran grafis dari suatu sistem yang menggunakan sejumlah bentuk-bentuk simbol untuk menggambarkan bagaimana data mengalir melalui suatu proses yang berkaitan (McLeod., 2001). DFD sistem navigasi kota Surakarta disajikan seperti Gambar 3.



Gambar 3. DFD Level 0 Sistem Navigasi

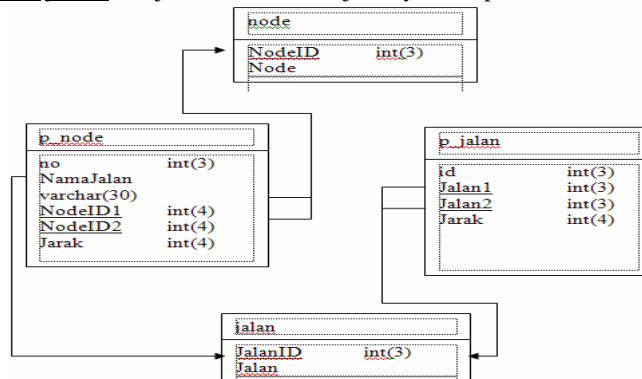
Sedangkan DFD Level 1 pada proses sistem navigasi dapat dilihat pada Gambar 4.



Gambar 4. DFD Level 1 Sistem Navigasi

Perancangan Perangkat Lunak

Dalam perancangan tabel lokasi dan jalan, data lokasi menggunakan node dan p_node. node berisi NodeID dan node, p_node terdiri no, NamaJalan, NodeID1, NodeID2 dan jarak, p_node di berikan atribut jarak karena untuk mendapatkan jarak harus diketahui dua buah node. Pada tabel jalan terdiri atas JalanID dan Jalan, Kemudian untuk p_jalan terdiri atas id, Jalan1, Jalan2 dan jarak. Untuk lebih jelasnya lihat pada Gambar 5.



Gambar 5. Perancangan relasi antar tabel Lokasi dan Jalan

Hasil dan Pembahasan

Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam sitem navigasi dapat dilihat pada Tabel 1, Tabel 1 Kebutuhan Pembuatan Aplikasi

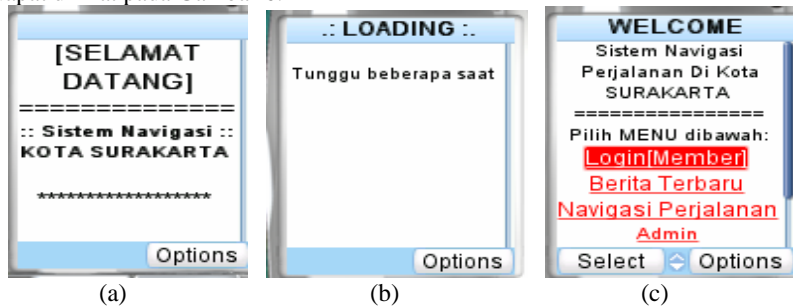
No	Jenis Lingkungan Sistem	Perangkat Lunak
1	Sistem Operasi	Microsoft XP Professional
2	Web Server	Apache(1.3.23)
3	DBMS Server	MySQL 3.23.51-nt
4	Web Scripting	PHP v4.1.1
5	Web Browser	Web Browser Internet Explorer
6	WML emulator/editor	Openwave V7 Simulator

Implementasi Halaman Utama

Halaman ini berisi menu yang menampilkan beberapa informasi seperti pencarian rute yang diinginkan di menu navigasi perjalanan, berita terbaru. Di menu navigasi perjalanan User non member dapat memilih jenis sumber pencarian, apakah jalan atau lokasi. Ada 2 tipe pencarian rute, yaitu :

- Pencarian rute dari jalan ke jalan
- Pencarian rute dari lokasi ke lokasi

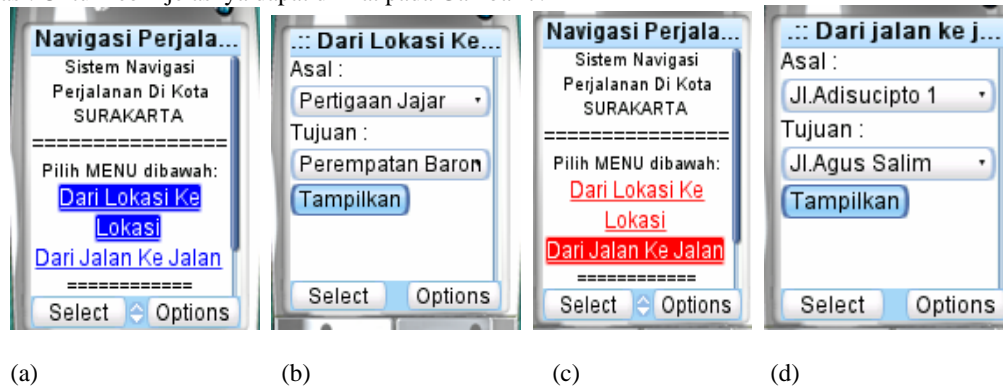
Halaman utama dapat dilihat pada Gambar 6.



(a) (b) (c)
Gambar 6 (a) dan (b) Proses Loading (c) Menu Halaman utama

Implementasi Halaman Input Navigasi Perjalanan

Halaman input adalah halaman yang menyediakan fasilitas agar user member atau non member dapat memasukkan data-data yang dibutuhkan untuk pencarian rute. Setelah memilih tipe pencarian rute, user non member harus mengisi data asal dan tujuan. Asal adalah posisi awal dimana pencarian rute akan dilakukan, Asal dapat berupa jalan atau lokasi. Tujuan adalah posisi akhir dimana pencarian rute akan dilakukan, tujuan dapat berupa jalan atau lokasi. Untuk lebih jelasnya dapat dilihat pada Gambar 7.



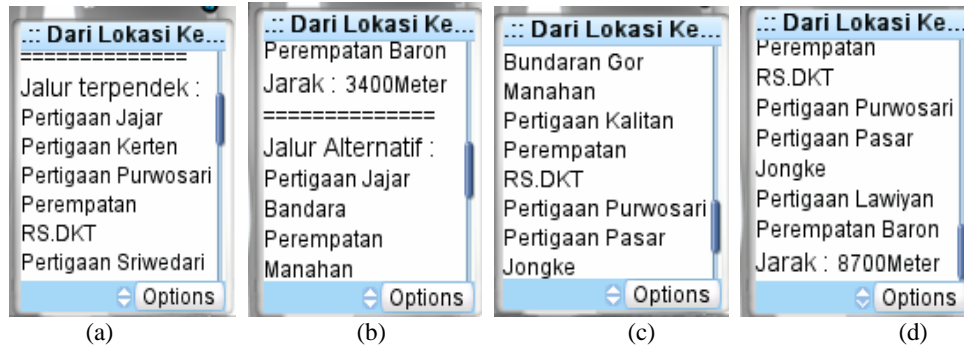
(a) (b) (c) (d)
Gambar 7 (a) Menu navigasi perjalanan, dan memilih dari lokasi ke lokasi
(b) Tampilan menu dari lokasi ke lokasi
(c) Menu navigasi perjalanan dan menyorot/memilih dari jalan ke jalan,
(d) Tampilan menu dari jalan ke jalan.

Penjelasan dari Gambar 7 adalah user non member bisa memilih menu dari lokasi ke lokasi atau dari jalan ke jalan, setelah memilih salah satu berikutnya tampak Gambar7 (b), yaitu menu yang harus ditentukan asal dan

tujuannya, sehingga bisa ditampilkan hasilnya, Untuk Gambar 7 (c) dan 7 (d) sama prinsip kerjanya dengan memilih lokasi, hanya saja yang dicari adalah jalan.

Implementasi Halaman Hasil Teks navigasi perjalanan

Hasil pencarian rute terpendek ditampilkan pada halaman ini dalam format teks. Dalam penampilan ke layar, hasil pencarian ini disusun secara menurun dan dapat disorot pada masing-masing baris sehingga *user non member* dapat melihat hasil dengan baik walaupun dengan keterbatasan kemampuan layar. Dapat dilihat jelas pada Gambar 8, untuk menu dari lokasi ke lokasi.



Gambar 8 (a) Halaman hasil text, karena layar terbatas maka di scrol tampak seperti Gambar (b), kemudian dengan layar berikutnya tampak juga seperti Gambar (c) dan (d)

Penjelasannya adalah setelah *user non member* menentukan asal dan tujuan, dalam hal ini adalah asal pertigaan jajar dan tujuan perempatan baron, maka setelah di eksekusi dengan cara melakukan klik pada tombol Tampilkan, maka hasilnya adalah jalur terpendek kemudian muncul pertigaan jajar, selanjutnya pertigaan kerten, pertigaan purwosari, perempatan RS.DKT, Pertigaan Sriwedari dan terakhir perempatan baron, serta diketahui jaraknya 3400 meter. Jika *user non member* tidak menginginkan jalur terpendek tersebut, *user non member* juga bisa memilih jalur alternatif, hasil dari jalur alternatif terpendeknya adalah: Pertigaan jajar, selanjutnya Bandara, perempatan manahan, Bundaran gor manahan, pertigaan kalitan, perempatan RS DKT, pertigaan purwosari, pertigaan pasar jongke, pertigaan Lawiyan, baru dengan tujuan terakhir yaitu perempatan Baron dan diketahui jarak rute alternatifnya yaitu 8700 meter. Sehingga *user non member* bisa mengetahui selisih dari jarak rute terpendek dengan rute alternatif tersebut yaitu:

$$\begin{aligned} \text{Selisih jarak} &= (\text{Jarak terpendek alternatif}) - (\text{jarak rute terpendek}) \\ &= 8700 \text{ m} - 3400 \text{ m} \\ &= 5300 \text{ m} \end{aligned}$$

Untuk menu dari jalan ke jalan tampak seperti Gambar 9.



Gambar 10 (a) Halaman hasil teks untuk pencarian dari jalan ke jalan
(b) Lanjutan hasil pencarian dari jalan ke jalan

Untuk pilihan pencarian Jalan prinsip kerjanya sama dengan mencari lokasi, yaitu jalan yang dilalui dan jarak serta rute alternatif beserta jaraknya. Keterangan berdasarkan Gambar 10 (a), (b) adalah asal yang dipilih adalah Jl.Adisucipto 1, Tujuan yang dipilih adalah Jl.Agus Salim, maka Jalur terpendeknya adalah Jl.Adisucipto 1, Jl.Muwardi 2, Jl.Muwardi 1, Jl.Slamet Riyadi 2 dan yang dituju adalah Jl.Agus Salim jaraknya 5400 Meter. Kemudian untuk rute alternatifnya adalah Jl. Adisucipto 1, Jl.Ayani 6, Jl.Slamet Riyadi 2 dan terakhir Jl.Agus Salim dengan jarak 5600 Meter.

Simpulan

Setelah dilakukan serangkaian uji coba dan analisa terhadap perangkat lunak yang dibuat, maka dapat diambil kesimpulan sebagai berikut:

1. Terciptanya sistem layanan informasi yang *real time*, dan sistem navigasi yang bisa diakses dengan telepon seluler yang mampu menunjukkan rute yang paling pendek, serta dapat menunjukkan rute alternatif jika terjadi kemacetan atau terjadi kecelakaan di salah satu ruas jalan atau di suatu lokasi.
2. Algoritma Dijkstra dapat diimplementasikan untuk pencarian rute, dalam proses pencarian rute tersebut ada faktor yang berpengaruh, di antaranya adalah jarak.
3. Hasil dari pencarian rute dengan memanfaatkan aplikasi WML akan mempermudah user dengan menunjukkan pedoman sesuai dengan hasil pencarian rute yang ada pada data.
4. Biaya yang dikeluarkan dalam pencarian rute menggunakan aplikasi WML lebih murah karena memanfaatkan koneksi *GPRS* daripada pencarian rute menggunakan teknologi SMS.

Persantunan

Terima kasih disampaikan kepada DP2M Diknas, Rektor ,LPPM, dan jurusan Teknik Industri Universitas Muhammadiyah Surakarta, serta pihak DLLAJ kota Surakarta atas dukungan secara moral dan finansial sehingga penelitian ini dapat terlaksana.

Daftar Pustaka

- Arunanto, F.X., Ulum, A., Johan, A .F., (2006), "Perancangan dan Pembuatan Perangkat Lunak Pencarian Rute Tercepat menggunakan SMS", *Prosiding Seminar nasional Pascasarjana VI FTIF ITS*, Surabaya, ISBN-979-545-027-1
- Djojo, M., 2000, "Perkembangan Internet Pada Mobile Device", *Article Technologies*, <http://www.aracle.net>, diakses 1-9-2007.
- Mcleod, R. Jr., (2001), "*Management Information System: eight edition*", Prentice-Hall, Inc., New Jersey
- Narayanan, S, (2003), "*Simulation of Dijkstra Routing Algorithm*", BE- Electronics