SISTEM PENCARIAN JALUR TERPENDEK UNTUK JASA PESAN ANTAR PADA PT.FASTFOOD INDONESIA (KFC) CABANG MALANG TOWN SQUARE

SKRIPSI

Disusun oleh :

TITO TRI WAHYU ABADI
NIM : 0612659

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER & INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

2011

LEMBAR PERSETUJUAN

SISTEM PENCARIAN JALUR TERPENDEK UNTUK JASA PESAN ANTAR PADA PT.FASTFOOD INDONESIA (KFC) CABANG MALANG TOWN SQUARE

SKRIPSI

Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik

Disusun Oleh :

TITO TRI WAHYU ABADI
NIM : 06.12.659

Mengetahui,
Ketua Jurusan Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda,MT
NIP.Y.1018800189

Diperiksa dan Disetujui

Dosen Pembimbing I	Dosen Pembimbing II
Irmalia Suryani Faradisa,ST.MT	**Sandy Nataly Mantja,S.Kom**
NIP.P. 103 0000 365	**NIP.P.103 0800 418**

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2011

ABSTRAKSI

SISTEM PENCARIAN JALUR TERPENDEK UNTUK JASA PESAN ANTAR PADA PT.FASTFOOD INDONESIA (KFC) CABANG MALANG TOWN SQUARE

Tito Tri Wahyu Abadi
Email **:** titoabadi@ymail.com
Jurusan Teknik Elektro
Konsentrasi Teknik Komputer & Informatika S-1
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jl. Raya Karanglo Km 2 Telp (0341) 417636 Fax (0341) 417634 Malang
Email**:** itn@itn.ac.id

Dosen Pembimbing I : Irmalia Suryani Faradisa, ST. MT
Dosen Pembimbing II : Sandy Nataly Mantja, S.Kom

Kebutuhan akan informasi sangat penting dalam kehidupan sehari-hari, setiap orang membutuhkan informasi yang akurat, tepat waktu, dan ter up to date.Bahkan saat ini kebutuhan atas informasi jalan-jalan di suatu daerah perkotaan baik itu kondisi rute-rute jalan maupun sarana transportasi yang ada banyak dibutuhkan dalam kehidupan sehari-hari bagi mereka para pengguna jalan. Dengan memanfaatkan website pada jaringan internet, informasi yang diinginkan akan sampai ke tujuan dengan waktu yang relative lebih singkat dan akurat.

Jasa pesan antar merupakan pelayanan antar makanan dari satu lokasi ke lokasi lain. Akibat tuntutan pelayanan yang semakin meningkat, penyedia jasa dihadapkan dengan perancanaan rute pelayanan dan penanggulangan permintaan antar makanan. Salah satu permasalahan mendasar yang dihadapi jasa pesan antar adalah merencanakan rute secara tepat sehingga mendapatkan waktu tempuh yang relatif singkat dan biaya perjalanan yang lebih murah.

Oleh karena itulah pada tulisan ini, memuat konsep implementasi GIS (*Geographic Information System*) yang dapat menyajikan informasi-informasi data rute jalan terdekat, dengan harapan para pekerja pesan antar yang ingin mencari lokasi tujuan pengiriman makanan tidak mengalami kesulitan dalam menentukan rute yang akan dilewatinya.

Kata Kunci : *sistem informasi geografis, rute terpendek, algoritma djikstra*

KATA PENGANTAR

Dengan mengucap syukur kehadirat Tuhan YME yang dengan segala Kasih dan Anugerah – Nya, telah memberikan kekuatan, kesabaran, bimbingan dan perlindungan sehingga penulis dapat menyelesaikan laporan skripsi dengan judul : **" SISTEM PENCARIAN JALUR TERPENDEK UNTUK JASA PESAN ANTAR PADA PT.FASTFOOD INDONESIA (KFC) CABANG MALANG TOWN SQUARE"**

Pembuatan skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata I di Institut Teknologi Nasional Malang. Dalam penyusunan skripsi ini penulis banyak mendapat bantuan baik moril maupun materiil, saran dan dorongan semangat dari berbagai pihak, untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Ir. Soeparno Djiwo, MT selaku rektor ITN Malang
2. Bapak Ir. Sidik Noertjahjono, MT selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir.Yusuf Ismail Nakhoda,MT selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
4. Ibu Irmalia Suryani Faradisa,ST.MT selaku Dosem Pembimbing I.
5. Ibu Sandy Nataly Mantja,S.Kom selaku Dosem Pembimbing II.
6. Bapak Trijono dan Ibu Aminah selaku orang tua penulis yang selalu memberi motivasi serta do'a yang mereka panjatkan untuk penulis sehingga dapat menyelesaikan skripsi.
7. Fera Andriani dan teman-teman Garda12 yang selalu membantu dan memberikan motivasi bagi penulis untuk menyelesaikan skripsi ini.
8. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan.

Akhir kata, penulis mohon maaf kepada semua pihak bilamana selama penyusunan skripsi ini penyusun membuat kesalahan secara tidak sengaja dan semoga skripsi ini dapat bermanfaat bagi kita semua.

Malang, September 2011

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kebutuhan akan informasi sangat penting dalam kehidupan sehari-hari, setiap orang membutuhkan informasi yang akurat, tepat waktu, dan ter up to date.Bahkan saat ini kebutuhan atas informasi jalan-jalan di suatu daerah perkotaan baik itu kondisi rute-rute jalan maupun sarana transportasi yang ada banyak dibutuhkan dalam kehidupan sehari-hari bagi mereka para pengguna jalan. Dengan memanfaatkan website pada jaringan internet, informasi yang diinginkan akan sampai ke tujuan dengan waktu yang relative lebih singkat dan akurat.

Jasa pesan antar merupakan pelayanan antar makanan dari satu lokasi ke lokasi lain. Akibat tuntutan pelayanan yang semakin meningkat, penyedia jasa dihadapkan dengan perancanaan rute pelayanan dan penanggulangan permintaan antar makanan. Salah satu permasalahan mendasar yang dihadapi jasa pesan antar adalah merencanakan rute secara tepat sehingga mendapatkan waktu tempuh yang relatif singkat dan biaya perjalanan yang lebih murah.

Oleh karena itulah pada tulisan ini, memuat konsep implementasi GIS (*Geographic Information System*) yang dapat menyajikan informasi-informasi data rute jalan terdekat, dengan harapan para pekerja pesan antar yang ingin mencari lokasi tujuan pengiriman makanan tidak mengalami kesulitan dalam menentukan rute yang akan dilewatinya.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas maka timbul suatu permasalahan bagaimana membuat suatu sistem yang bisa menentukan rute jalan terdekat untuk menuju ke lokasi.

1.3 Tujuan

Tujuan dari skripsi ini adalah mengimplementasikan sistem yang dapat membantu para pekerja jasa pesan antar mencari jalur terpendek yang bisa dilalui oleh pekerja untuk menuju lokasi pengiriman makanan.

1.4 Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan maka pembahasan dibatasi pada hal-hal sebagai berikut :

1. Perancangan system ini hanya untuk pecarian jalan.
2. Penanganan pencarian rute jalan hanya berlaku untuk jalan yang bisa dilewati menggunakan kendaraan bermotor.
3. Keadaan jalan tidak diperhitungkan.

4. Daerah pembahasan hanya radius ±5 km dari KFC Malang Town Square
5. Parameter yang digunakan untuk menentukan jalur terpendek adalah jarak tempuh (meter), bukan waktu untuk mencapai lokasi yang dituju.

1.5 Metodologi Penelitian

Dalam menyelesaikan permasalahan yang dihadapi, dilakukan penelitian dengan menggunakan metode pengumpulan data dan metode implementasi.

1.5.1 Metode Pengumpulan Data

Data merupakan sumber atau bahan mentah yang sangat berharga bagi proses menghasilkan informasi. Oleh sebab itu dalam pengambilan data perlu dilakukan penanganan secara cermat dan hati-hati, sehingga data yang diperoleh dapat bermanfaat dan berkualitas.

Dalam pengumpulan data penyusun mengunakan metode sebagai berikut :

1. Studi Lapangan

   Dengan metode ini data-data diperoleh langsung dari sumber yang bersangkutan, dimana peneliti berhadapan langsung dengan obyek yang diteliti, yang dilakukan dengan cara :

   a. Survey

      Teknik pengumpulan data dengan cara terjun secara langsung dan mencatat secara sistematis terhadap obyek masalah.

   b. Wawancara / Interview

      Teknik pengumpulan data dengan jalan mengadakan komunikasi langsung dengan pihak yang bersangkutan.
      (ex : KFC Malang Town Square).

2. Studi Pustaka / Literatur

   Pengumpulan data ini dilakukan dengan cara mencari bahan-bahan kepustakaan sebagai landasan teori yang ada hubunganya dengan permasalahan yang dijadikan obyek penelitian.

1.5.2 Metode Implementasi

Dalam implementasi sistem untuk menentukan rute jalan terpendek ini ada beberapa langkah yang harus dilakaukan antara lain :

1. Analisis

   Tahap ini melakukan pengumpulan elemen-elemen yang berkaitan dengan proses implementasi dan data yang diperlukan, meliputi: data, metode, fungsi, proses atau procedure. Hasil akhir tahap ini adalah spesifikasi kebutuhan yang diperlukan, khususnya dalam hal perangkat lunak.

2. Design

   Spesifikasi perangkat lunak yang dihasilkan dari tahap analisa ditransformasikan kedalam bentuk arsitektur perangkat lunak yang memiliki karakteristik mudah dimengerti dan tidak sulit untuk diimplementasikan.

3. Coding / Pemrograman

   Tahap ini dilakukan implementasi hasil rancangan kedalam baris-baris kode program yang dapat dimengerti oleh mesin.

4. Testing

   Pengujian dilakukan untuk setiap modul. Jika hasil pengujian tidak menemukan adanya masalah, modul-modul yang terpisah tersebut diintegrasikan untuk mendapatkan perangkat lunak yang utuh. Kemudian, dilakukan pengujian ditingkat perangkat lunak yang memfokuskan pada masalah-masalah logika internal, fungsi eksternal, potensi masalah yang mungkin terjadi dan pemeriksaan hasil.

1.6 Sistematika Penulisan

Sistematika penulisan yang diuraikan dalam penyusunan skripsi ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Bab ini berisi tentang latar belakang, tujuan, permasalahan, batasan masalah, dan sistematika pembahasan dari skripsi ini.

BAB II : TINJAUAN PUSTAKA

Bab ini berisi penjelasan tentang teori – teori yang mendukung dalam implementasi system yang meliputi teori GIS, PHP, MySQL, Metode yang digunakan dalam pencarian rute terpendek/terdekat.

BAB III : PERANCANGAN SISTEM

Bab ini berisi tentang serangkaian langkah yang ditempuh dalam pelaksanaan kegiatan penelitian ini serta langkah perancangan model sistem yang diusulkan dari awal sampai akhir, rancangan struktur database yang dipergunakan dalam aplikasi serta desain masukan dan keluaran aplikasi (*design user interface*).

BAB IV : HASIL DAN ANALISA

Bab ini berisi pembahasan hasil pengujian dan analisa mengenai cara kerja dari aplikasi.

BAB V : PENUTUP

Bab ini berisi kesimpulan dan saran dari hasil pembahasan pada skripsi ini.

BAB II
LANDASAN TEORI

2.1 GIS (Geographic Information System)[1]

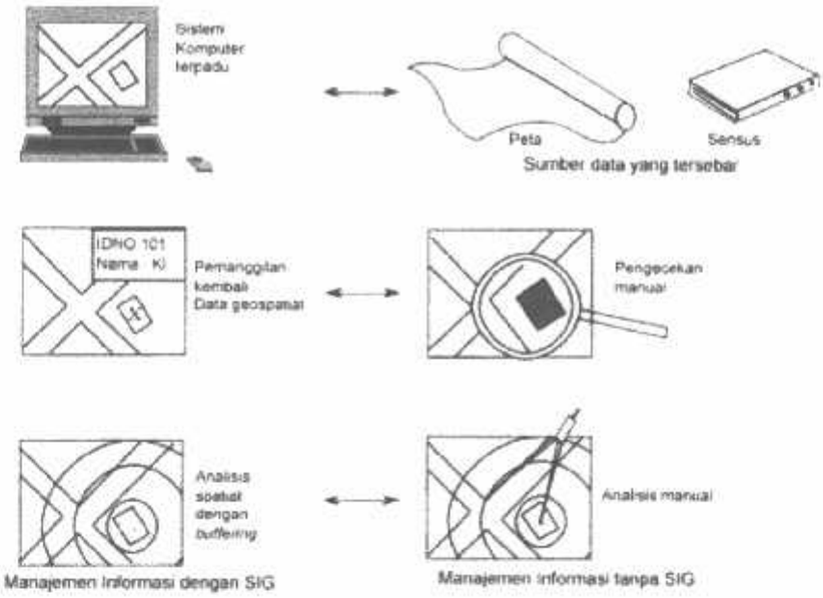
GIS adalah suatu sistem yang berbasis komputer yang digunakan untuk menyimpan dan mengintegrasikan data spasial (peta vektor dan citra digital) dan atribut (tabel sistem basis data) sehingga menghasilkan data bereferensi geografi atau data geospatial. Fungsionalitas perangkat lunak GIS adalah kemampuan dalam menjawab hal – hal terkait analisis (query). GIS dapat memecahkan masalah – masalah analisis spasial, atribut, dan kombinasinya.

GIS merupakan suatu sistem yang mengorganisir perangkat keras (Hardwere), perangkat lunak (Software) dan data, serta dapat mendayagunakan sistem penyimpanan, pengolahan maupun analisis data secara simultan, sehingga dapat diperoleh informasi yang berkaitan dengan aspek keruangan. GIS yang juga merupakan manajemen data spasial (peta) dan non-spatial (tabular/tekstual) yang berbasis komputer dan menyajikan secara bersamaan

Pada tabel 2.1 dan gambar 2.1 ini memperlihatkan kelebihan GIS dibanding dengan pengerjaan secara manual.

Tabel 2.1 Keunggulan GIS

Peta	**GIS**	**Pekerjaan Manual**
Penyimpanan	Database digital baku dan terpadu	Skala dan standart beda
Pemanggilan kembali	Pencarian dengan komputer	Cek manual
Pemutakhiran	Sistematis	Mahal dan makan waktu
Analisa Overlay	Sangat cepat	Memakan waktu dan tenaga
Analisa Spatial	Mudah	Rumit
Penayangan	Murah dan cepat	Mahal



Gambar 2.1 Perbandingan GIS dengan manual

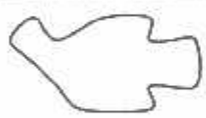
2.1.1 Data dalam GIS

Ada dua macam data dalam GIS (Geographic Information System), yaitu *data spatial* dan *data non-spatial (atribut).*

2.1.1.1 Data Spatial

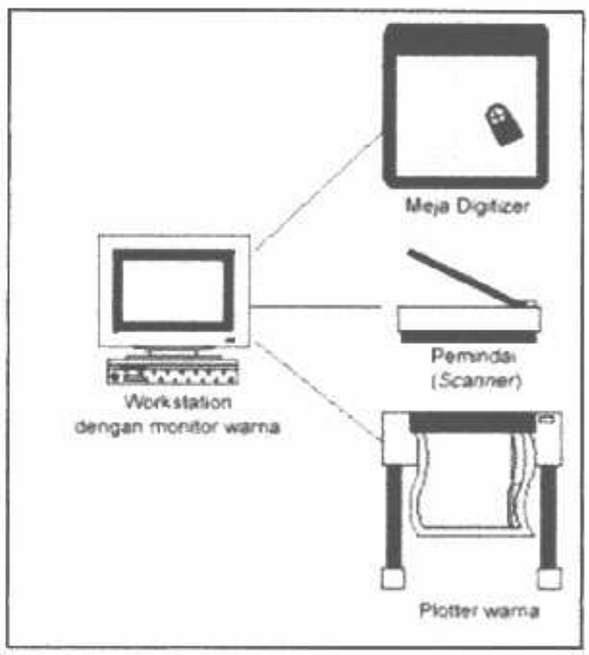
Data spatial merupakan data yang menggambarkan informasi dalam bentuk titik, garis, luasan (area). Oleh karena itu setiap fenomena geografis pada dasarnya dapat dinyatakan atau diwakili dalam bentuk titik (contoh: posisi pos pengamatan, posisi stasiun kereta, posisi terminal bus), garis (contoh: jalan, sungai, jembatan) dan polygon / luasan / area (contoh: batas administrasi, batas wilayah). Secara visual fenomena tersebut disajikan secara digital oleh teknologi komputer. Hal ini dilakukan untuk memudahkan pengguna jasa dalam melakukan analisis keruangan secara tepat guna.

Tabel 2.2 Ilustrasi Data Spasial

			
TITIK	GARIS	AREA /POLIGON	PERMUKAAN
Format titk :	Format garis :	Format area :	Format permukaan :
- Koordinat tunggal	- Koordinat titik awal dan titik akhir	- Koordinat titik awal dan titik akhir sama	- Area dengan koordinat vertikal
- Tanpa panjang	- Mempunyai	- Mempunyai	- Area dengan ketinggian

Contoh: - Letak pohon - Titik tinggi	panjang dan luasan Contoh - Jalan - Sungai	panjang dan luasan Contoh - Tanah - Bangunan	Contoh - Peta slope - Bangunan bertinggak

Dengan spatial dari beberapa sumber dan dalam bermacam-macam format data, maka perlu ada proses pengolahaan data agar dapat diterima oleh GIS. Data yang dimasukkan dalam GIS merupakan data yang sudah berupa digital. Sehingga diperlukan langkah-langkah untuk merubah data dari analog ke digital. Salah satu cara memudahkan data spasial yang berupa analog menjadi data digital adalah dengan cara digitasi menggunakan *digitizer* seperti pada gambar 2.2.



Gambar 2.2 Proses Pengolahan Data Analog Menjadi Data Digital

2.1.1.2 Data Non-Spatial

Data non-spatial bersumber dari data sekunder dan catatan statistic atau sumber lainnya seperti hasil survey dan eksplorasi. Data non-spatial sifatnya sebagai data attribute atau data pelengkap bagi data spatial, yaitu sebagai deskripsi tambahan pada titik, garis, polygon. Data non-spatial dapat berupa table-tabel statistik, kependudukan,

iklim, sumberdaya lahan, social ekonomi, kawasan politik yang dapat dikaitkan dengan luasan administrasi.

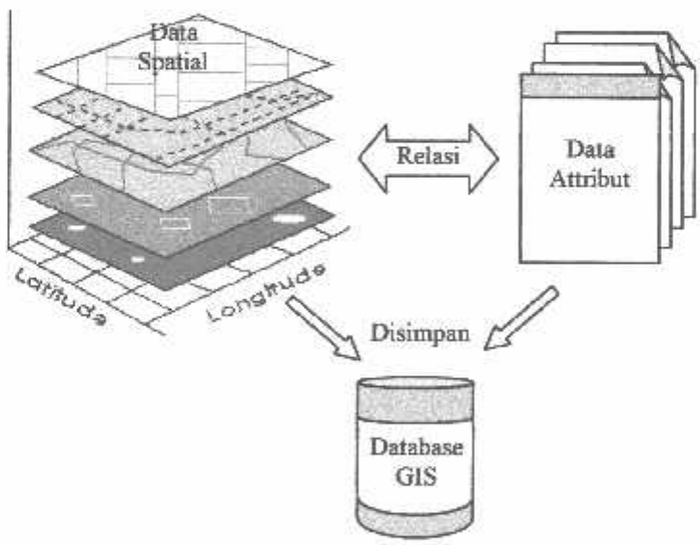
Agar data non-spatial dapat digunakan pada GIS, maka perlu diubah terlebih dahulu dengan suatu software yang mendukung Data Base Management System (DBMS). DBMS merupakan sistem yang digunakan untuk memudahkan pembuatan dan pemeliharaan basis data yang terkomputasi. Sistem ini bertujuan untuk mengolah data yang digunakan secara bersamaan dengan satu tujuan, dan terintegrasi kedalam basis data.

DBMS merupakan "*interface*" yang mengatur:

a. Bagaimana setruktur data tersebut akan disimpan dan dapat dipergunakan kembali dengan mudah, misalnya mencari kembali data (retrival data).
b. Prosedur untuk mengakses data.
c. Pembentukan file, modifikasi, penyimpanan, up-dating dan proteksi file.

Contoh perngkat lunak yang mendukung DBMS adalah: Ms Access, Dbase, SQL Server, My SQL dan lain-lain.

Untuk membuat peta digital diperlukan penggabungan data-data schingga membentuk suatu relasi seperti pada gambar 2.3.



Gambar 2.3 Penggabungan Data Spatial dan Data Non-Spatial pada Aplikasi GIS

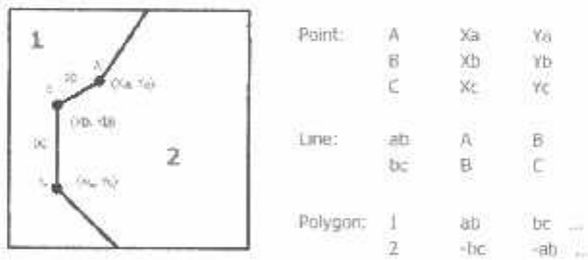
2.1.2 Komponen GIS

GIS terdiri dari empat komponen dasar, yaitu data, perangkat lunak, perangkat keras dan sumber daya manusia. Komponen tersebut saling berhubungan seperti gambar 2.4:

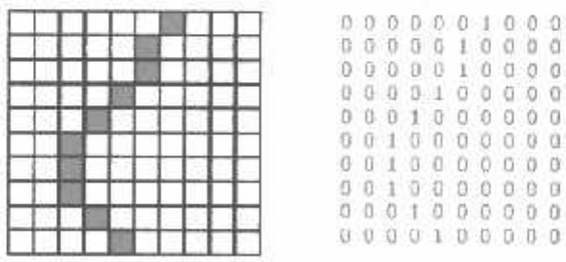
Gambar 2.4 Hubungan antar komponen GIS

2.1.2.1 Data

Data merupakan sebuah gambaran dari fakta-fakta, konsep –konsep atau instruksi-intruksi di dalam sebuah perumusan yang sesuai untuk komunikasi, interpretasi atau processing oleh manusia atau mesin. Data masukan GIS terdiri dari data spatial dan non-spatial (seperti yang telah dijelaskan pada poin 2.1.1), yang berupa raster, vektor, dan data tabular alfanumerik seperti pada gambar 2.5 dan gambar 2.6.



Gambar 2.5 Model data spatial vektor



Gambar 2.6 Model data spatial rester

2.1.2.2 Perangkat Keras

Perangkat keras GIS memiliki pengertian perangkat-perangkat fisik yang digunakan oleh sistem komputer. Komponen dasar perangkat keras GIS dapat dikelompokkan sesuai dengan fungsinya, antara lain:

a. Peralatan pemasukan data, antara lain: papan Digital (*digitizer*),penyiam (*scanner*).
b. Peralatan penyimpan dan pengolahan data, antara lain: komputer dan perlengkapannya: monitor, papan ketik (*Keyboard*), unit pusat pengolahan (*Central Processing Unit – CPU*), penyimpanan (HDD,CD).
c. Perangkat untuk mencetak hasil, seperti: printer dan plotter
d. Susunan keperluan: Perangkat keras ini bervariasi dari bentuk yang paling sederhana seperti komputer pribadi dengan hanya printer dan plotter sampai ke yang lebih kompleks dengan *workstation* atau *main frame* dengan berbagai komponen yang lengkap.

2.1.2.3 Perangkat Lunak

Perangkat lunak merupakan komponen untuk mengintegrasikan berbagi macam data masukkan. Perangkat lunak GIS didesain untuk melakukan analisis geografis dan sebagian besar perangkat lunak tersebut dapat digunakan untuk manipulasi data spatial dan data non-spatial. Contoh perngkat lunak yang sering digunakan adalah ArcInfo, MapInfo, ArcView, Ilwis, Span dan sebagainya.

Perangkat lunak khususnya GIS digunakan untuk menjalankan tugas GIS. Perangkat lunak ini tersedia dalam bentuk paket prerangkat lunak yang terdiri dari multi program dalam mendukung kemampuan khusus untuk pemetaan, manajemen, dan analisa data geografis. Perangkat lunak dikembangkan untuk GIS secara konseptual terdiri dari dua bagian yaitu: Paket Inti (core) diguanakn untuk pemetaan dasar dan management data. Aplikasi-aplikasi yang terintegrasi dengan paket inti untuk menjalankan pemetaan khusus dan aplikasi analisis GIS.

2.1.2.4 Sumber Daya Manusia

Sumber daya manusia merupakan pengguna system dan yang mengoperasikan perangkat lunak mapun perangkat keras.

2.1.3 Perkembangan GIS

Seiring dengan perkembangan jaringan internet saat ini, maka aplikasi GIS kini juga dapat diterapkan pada jaringan internet yang lebih dikenal dengan istilah *Web-Based GIS*. Sehingga muncul sebuah ide bagaiman mengimplementasi *Web-Based GIS* untuk menentukan jalur/rute jalan terpendek di Kota Malang yang dapat diakses oleh

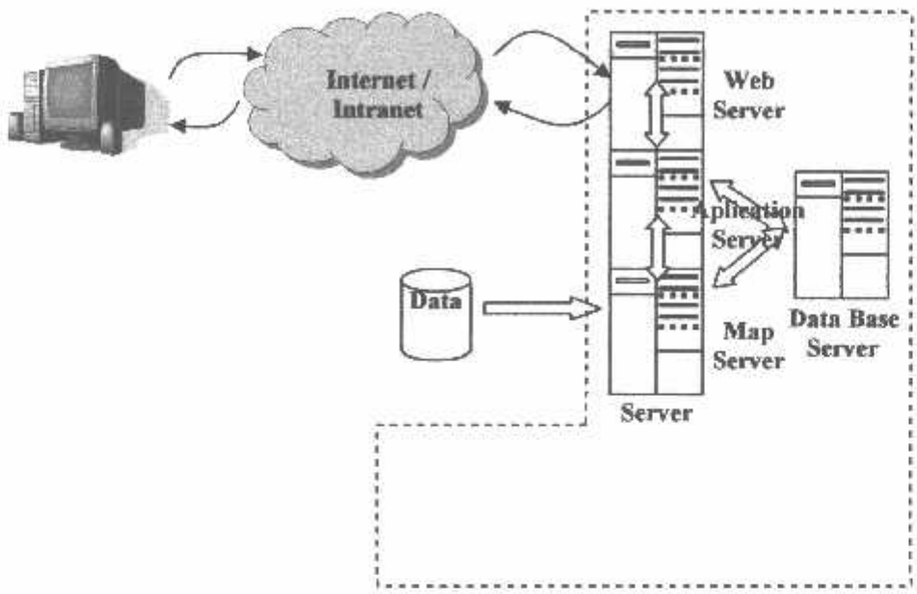
masyrakat luas memalui jaringan internet dengan cepat, tepat dan akurat, di banding aplikasi GIS saat ini yang sebagian besar masih Stand Alone (Desktop).

Ada beberapa perbedaan fenomena yang terdapat pada aplikasi GIS yang berjalan di sistem komputer PC (*desktop*) dengan yang berjalan pada platform jaringan internet/intranet, antara lain:

a. Tujuan / terget pengembangan aplikasi GIS berbasis desktop memang berbeda dengan aplikasi GIS berbasis internet (*webbased*).
b. Pengembangan aplikasi *webbased* GIS yang didasarkan pada konsep arsitektur *web client-server* menjadikannya tidak mudah untuk dibandingkan secara sederhana dengan *desktop based.*
c. Kecepatan akses ke jaringan internet, kondisi existing volume lalulintas di jaringan internet terkait, dan unjuk kerja server yang bersangkutan selalu menjadi faktor kendala bagi aplikasi *webbased* GIS. Sementara *desktop based* GIS tidak mengalaminya.
d. Pengguna bebas menjalankan *query* dan analisis spasialnya (*geoprocessing*) di aplikasi *desktop based* GIS. Ia bebas menjalankan fungsi-fungsi terkait selama perangkat lunak yang bersangkutan menyediakannya. Tetapi di aplikasi *webbased* GIS, fungsionalitasnya yang sama akan sangat bergantung pada komponen *map server aplication server* [akan di bahas pada poin berikutnya]. Biasanya ukuran komponen *map server* lebih kecil dari pada *desktop based* GIS. Selain itu, tidak semua fungsionalitas yang terdapat di dalamnya selalu digunakan oleh pengembang dalam mengimplementasikan *application server*-nya. Atau dengan kata lain, fungsionalitas *webbased* GIS sangat terbatas di banding *desktop based* GIS.
e. Pada *desktop* GIS, pengguna berinteraksi langsung dengan *user interface & engine*-nya (tanpa sekat). Sementara pada *webbased* GIS, pengguna (client) tidak dapat berhubungan langsung dengan *engine*-nya. Ia terlebih dahulu harus berhubungan dengan *web server* dan *server* aplikasinya.
f. Aplikasi GIS yang berjalan di internet cenderung hanya menampilkan pete-peta digital dengan simbol-simbol & legenda berwarna, dan tabel attribut juga menyediakan beberapa fungsi manipulasi tampilan: *zoom in, zoom out, pan*, menjalankan fungsi pencarian dan *quey* interaktif sederhana, dan menyediakan beberapa file untuk di *download.* Sementara *desktop based* GIS menyediakan

semua fungsi-fungsi diatas secara penuh plus beberapa tambahan lainnya minus fasilitas *download*.

Pada gambar 2.7 adalah rancangan sistem peta yang berjalan pada platform jaringan internet.



Gambar 2.7 Arsitektur aplikasi *Web Based GIS*

2.2 Aplication Server (PHP)[9]

2.2.1 PHP *Hypertext Prepocessor*

PHP (*Hypertext Prepocessor*) merupakan suatu bahasa pemrograman untuk *website* yang dinamis. Dikatakan dinamis, karena PHP digunakan untuk menampilkan data terbaru yang berhubungan dengan *database* sesuai dengan perintah yang diberikan. Selain itu PHP juga dapat digabungkan dengan berbagai macam tipe bahasa pemrograman *web* lainya.

PHP dikatakan sebagai *surver-side embedded script language* artinya *sintaks-sintaks* dan perintah yang diberikan akan sepenuhnya dijalankan oleh *server* tetapi disertakan pada halaman HTML biasa. Ketika menggunakan PHP sebagai server-side embedded script language maka server akan melakukan hal-hal sebagai berikut :

1) Membaca permintaan dari client/browser
2) Mencari halaman/page di server
3) Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/page.
4) Mengirim kembali halaman tersebut kepada client melalui internet atau intranet.

2.2.2 Keunggulan php

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaanya.
2. Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahamanan, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system.

2.2.3 Type Data PHP

PHP memiliki beberapa tipe data yaitu *integer, double, boolean, string, object, array, null, dan Resource*. Berikut ini penjelasan dari ke tujuh tipe data tersebut:

2.2.3.1 *Integer*

Integer digunakan untuk merujuk kepada tipe data apapun yang merepresentasikan bilangan bulat, atau beberapa bagian dari bilangan bulat. Disebut juga sebagai *Integral Data Type.*

Nilai sebuah data dari sebuah tipe data integer adalah nilai bilangan bulat tersebut dalam matematika. *Representasi* data ini merupakan cara bagaimana nilainya disimpan di dalam memori komputer. Tipe data integral terbagi menjadi dua buah kategori, baik itu bertanda (signed) ataupun tidak bertanda (unsigned). Bilangan bulat bertanda mampu merepresentasikan nilai bilangan bulat negatif, sementara bilangan bulat tak bertanda hanya mampu merepresentasikan bilangan bulat positif. *Representasi integer* positif di dalam komputer sebenarnya adalah untaian bit, dengan menggunakan sistem bilangan *biner*. Urutan dari bit-bit tersebut pun bervariasi, bisa berupa Little Endian ataupun Big Endian. Selain ukuran, lebar atau ketelitian (presisi) bilangan bulat juga bervariasi, tergantung jumlah bit yang direpresentasikanya. Bilangan bulat yang memiliki n bit dapat mengodekan 2^n. Jika tipe bilangan bulat tersebut adalah bilangan bulat tak bertanda, maka jangkauannya adalah dari 0 hingga 2^n-1.

2.2.3.2 *Double*

Double/Floating point adalah tipe data yang berisi bilangan real atau pecahan. Jangkauan/range dari tipe data ini adalah antara 1.7e-308 sampai 1.7e+308. Data tersebut berbentuk desimal ataupun berbentuk pangkat.

Contoh :

$c = 4.352;

$b = 1.2e3;

2.2.3.3 *Boolean*

Dalam matematika dan ilmu komputer, *Aljabar Boolean* adalah struktur aljabar yang "mencakup intisari" operasi logika AND, OR dan NOR dan juga teori himpunan untuk operasi union, interseksi dan komplemen.

Penamaan *Aljabar Boolean* sendiri berasal dari nama seorang matematikawan asal Inggris, bernama *George Boole*. Dialah yang pertama kali mendefinisikan istilah itu sebagai bagian dari sistem logika pada pertengahan abad ke-19. *Boolean* adalah suatu tipe data yang hanya mempunyai dua nilai. Yaitu true atau false (benar atau salah). Pada beberapa bahasa pemograman nilai true bisa digantikan 1 dan nilai false digantikan 0.

2.2.3.4 *String*

String dalam pemrograman komputer adalah sebuah deret simbol. Tipe data *string* adalah tipe data yang digunakan untuk menyimpan barisan karakter. Dalam penulisannya, tipe data *string* menggunakan tanda kutip tunggal (' ') atau bisa juga menggunakan tanda kutip ganda (" "). Dari kedua cara penulisan dengan tanda kutip tersebut, ada perbedaan antara keduanya yaitu pada saat penggunaan *variabel*. Jika menggunakan tanda kutip tunggal, maka apabila sebuah *variabel* berisi tipe data *string* dan berisi tipe data yang lain, yang terjadi adalah nilai dari variabel tersebut akan dibaca atau tetap dicetak dengan nama *variable* itu sendiri.

2.2.3.5 *Object*

Tipe data *Object* bisa berupa bilangan, variabel, ataupun fungsi. Tipe data tersebut dapat membantu programmer untuk membuat sebuah program. Data itu dapat disertakan dalam program sehingga meringkas beberapa fungsi dan dapat memperkecil ukuran file. Semakin kecil ukuran file semakin singkat waktu yang dibutuhkan untuk mengakses file tersebut.

2.2.3.6 *Array*

Tipe data *array* ini mampu untuk menyimpan lebih dari satu data akan tetapi tiap element data dalam *array* dibedakan menurut nomor indeksnya. Selain itu juga, *array* merupakan tipe data terstruktur yang berguna untuk menyimpan sejumlah data yang

bertipe sama. Bagian bagian yang menyusun array disebut juga dengan *element array*, yang masing masing dari element dapat diakses secara tersendiri melalui *indeks array*. *Array* terdiri dari *Array* berdimensi satu dan *Array Multidimensi.*

2.2.3.7 *Null*

NULL adalah spesial type yang tidak memiliki nilai yang biasa digunakan untuk mereset nilai dari sebuah variabel.

2.2.3.8 *Resource*

Sebuah *resource* adalah sebuah spesial variabel. Resource terbuat dan digunakan oleh beberapa fungsi spesial.

2.2.4 Variable PHP

Variabel digunakan untuk menyimpan suatu nilai, seperti text, angka atau array. Ketika sebuah variabel dibuat, variabel tersebut dapat dipakai berulang-ulang.

Pada PHP semua variabel harus dimulai dengan karakter '$'. Variabel PHP tidak perlu dideklarasikan dan ditetapkan jenis datanya sebelum kita menggunakan variabel tersebut. Hal itu berarti pula bahwa tipe data dari variabel dapat berubah sesuai dengan perubahan konteks yang dilakukan oleh user. Secara tipikal, variabel PHP cukup diinisialisasikan dengan memberikan nilai kepada variabel tersebut.

Contoh berikut akan mencetak "PHP" :

```
$text = "PHP";
print "$text";
```

Identifier dalam PHP adalah case-sensitive, sehingga $text dengan $Text merupakan variabel yang berbeda. Built-in function dan structure tidak case-sensitive, sehingga echo dengan ECHO akan mengerjakan perintah yang sama. Identifier dapat berupa sejumlah huruf, digit/angka, underscore, atau tanda dollar tetapi identifier tidak dapat dimulai dengan digit/angka.

Aturan Penamaan Variabel

a. Nama variabel harus diawali dengan sebuah huruf atau garis bawah (underscore) " _ " .

b. Nama variabel hanya boleh mengandung karakter alpha-numeric dan underscore (a-Z, 0-9, dan _) .

c. Nama variabel tidak boleh mengandung spasi.

2.3 Web Server (Apache)

Web serever adalah *software server* yang menjadi tulang belakang dari *word wide web* (www). *Web server* menuggu perintah dari client yang menggunakan browser, seperti *opera, navigation, Internet Explorel, Modzila* dan program browser

lainya. Jika ada permintaan dari browser, maka web server akan memproses permintan itu dan kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke browser. *Server Web Apache* adalah *server web* yang dapat dijalankan di banyak sistem operasi (*Unix, BSD, Linux, Microsoft Windows* dan *Novell Netware* serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs *web*. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP. *Apache* memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat *dikonfigur, autentikasi* berbasis basis data dan lain-lain. *Apache* juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan *server* menjadi mudah. *Apache* merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation.*

1) Web server Apache mempunyai beberapa kelebihan yaitu:
    a. Apache termasuk dalam kategori freeware.
    b. Apache mudah sekali proses instalasinya jika dibanding web server lainnya seperti NCSA, IIS, dan lain-lain.
    c. Mampu beroperasi pada berbagai platform sistem operasi.
    d. Mudah mengatur konfigurasinya. Apache mempunyai hanya empat file konfigurasi.
    e. Mudah dalam menambahkan peripheral lainnya ke dalam platform web servernya.

2) Fasilitas atau ciri khas dari web server Apache adalah :
    a. Dapat dijadikan pengganti bagi NCSA web server.
    b. Perbaikan terhadap kerusakan dan error pada NCSA 1.3 dan 1.4.
    c. Apache merespon web client sangat cepat jauh melebihi NCSA.
    d. Mampu di kompilasi sesuai dengan spesifikasi HTTP yang sekarang.
    e. Apache menyediakan feature untuk *multihomed* dan *virtual server.*
    f. Kita dapat menetapkan respon error yang akan dikirim web server dengan menggunakan file atau skrip.
    g. Server apache dapat otomatis berkomunikasi dengan *client browser*nya untuk menampilkan tampilan terbaik pada *client browser*nya. Web server Apache.

2.4 Graf[4]

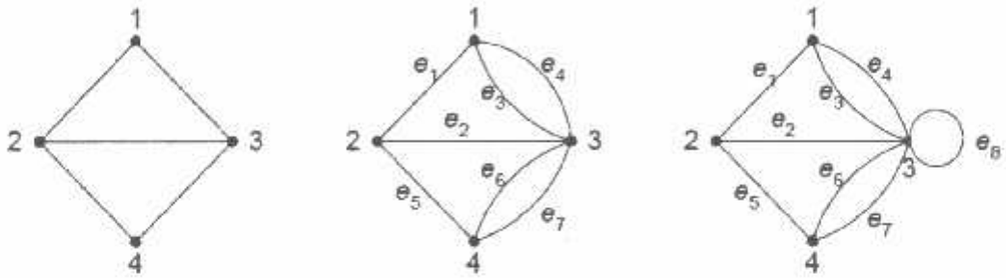
Graf $G\,(V, E)$, adalah koleksi atau pasangan dua himpunan .

a. Himpunan V yang elemennya disebut *simpul* atau *titik*, atau *vertex*, atau *point*, atau *node*.

b. Himpunan E yang merupakan pasangan tak terurut dari simpul, disebut *ruas* atau *rusuk*, atau *sisi*, atau *edge*, atau *line*.

c. Banyaknya simpul (anggota V) disebut *order* Graf G, sedangkan banyaknya ruas (anggota E) disebut *ukuran (size)* Graf G

Graf terdiri dari berbagai jenis, yaitu garf sederhana, multigraf seperti pada gambar 2.8.



G_1 G_2 G_3

Gambar 2.8 (G_1) graf sederhana, (G_2) multigraf, dan (G_3) multigraf

Pada Gambar 2.8 G_1 adalah graf dengan

$V = \{ 1, 2, 3, 4 \}$

$E = \{ (1, 2), (1, 3), (2, 3), (2, 4), (3, 4) \}$

G_2 adalah graf dengan

$V = \{ 1, 2, 3, 4 \}$

$E = \{ (1, 2), (2, 3), (1, 3), (1, 3), (2, 4), (3, 4), (3, 4) \}$

$= \{ e_1, e_2, e_3, e_4, e_5, e_6, e_7 \}$

G_3 adalah graf dengan

$V = \{ 1, 2, 3, 4 \}$

$E = \{ (1, 2), (2, 3), (1, 3), (1, 3), (2, 4), (3, 4), (3, 4), (3, 3) \}$

$= \{ e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8 \}$

a. Pada G_2, sisi $e_3 = (1, 3)$ dan sisi $e_4 = (1, 3)$ dinamakan **ruas berganda** atau **ruas sejajar** (*multiple edges* atau *paralel edges*), karena kedua sisi ini menghubungi dua buah simpul yang sama, yaitu simpul 1 dan simpul 3.

b. Pada G_3, sisi $e_8 = (3, 3)$ dinamakan *gelung* atau *self-loop* karena ia berawal dan berakhir pada simpul yang sama.

2.5 Pencarian Rute Terpendek

Secara umum penyelesaian masalah pencarian jalur terpendek dapat dilakukan menggunakan dengan dua buah metode, yaitu metode algoritma konvensional dan metode heuristik. Metode algoritma konvensional diterapkan dengan cara perhitungan matematis seperti biasa, sedangkan metode heuristik diterapkan dengan perhitungan kecerdasan buatan, dengan menentukan basis pengetahuan dan perhitungannya.

a. Metode konvensional

Metode konvensional berupa algoritma yang menggunakan perhitungan matematis biasa. Metode konvensional yang biasa digunakan untuk melakukan pencarian jalur terpendek, diantaranya algoritma Djikstraa, algoritma Floyd-Warshall, dan algoritma Bellman-Ford

b. Metode heuristik

Adalah sub bidang dari kecerdasan buatan yang digunakan untuk melakukan pencarian dan penentuan jalur terpendek. Ada beberapa algoritma pada metode heuristik yang biasa digunakan dalam pencarian jalur terpendek.

2.5.1 Algortima Dijkstra[2]

Algoritma Dijkstra adalah suatu algoritma rakus dimana algoritma ini digunakan untuk mencari rute permasalahan terpendek antara simpul sumber dan simpul tujuan untuk sebuah graf berarah berdasarkan bobot pada sisi yang bernilai tidak negatif.

Algoritma Dijkstra bekerja dengan cara mengunjungi simpul-simpul yang ada, dimulai dari simpul sumber. Kemudian algoritma ini memilih simpul-simpul yang lokasi nya terdekat dan dilakukan secara berulang lalu kemudian menghitung total bobot semua sisi yang dilewati untuk mencapai simpul tujuan

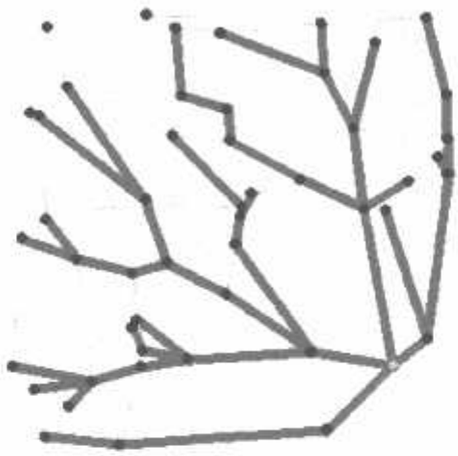
Algoritma Dijkstra dijamin dapat menemukan rute terpendek asalkan tidak terdapat bobot negatif pada setiap sisi dalam graph.

Input untuk algoritma ini adalah sebuah graph yang berarah dan berbobot G. Dan sebuah sumber *vertex* s dalam G. Dan V merupakan himpunan semua *vertice* dalam

grapf G. Setiap sisi dari graph ini adalah pasangan *vertice* (u,v) yang melambangkan hubungan *vertex u* dengan *vertex v* dan himpunan semua tepi E.

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

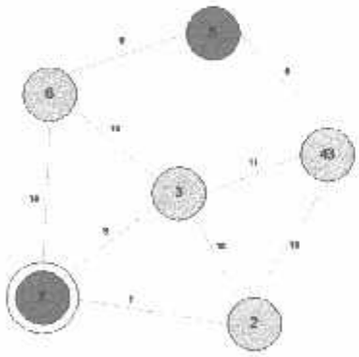
jadi $w(u, v)$ merupakan jarak non-negatif dari *vertex u* ke *vertex v*. Biaya sebuah sisi dapat dianggap sebagai jarak antara dua buah vertex dan merupakan jumplah jarak tiap sisi dalam jalur tersebut.



Gambar 2.9 Ilustrasi Algoritma Dijkstra

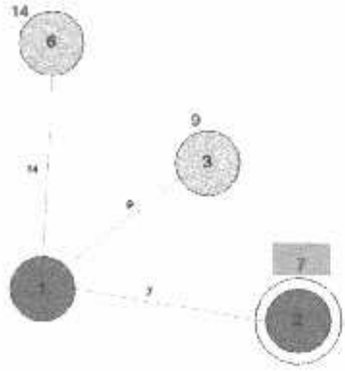
Pada gambar 2.10 sampai gambar2.14 adalah penjelasan langkah per langkah pencarian jalur terpendek secara rinci dimulai dari node awal sampai node tujuan dengan nilai jarak terkecil.

1. Node awal 1, Node tujuan 5. Setiap edge yang terhubung antar node telah diberi nilai



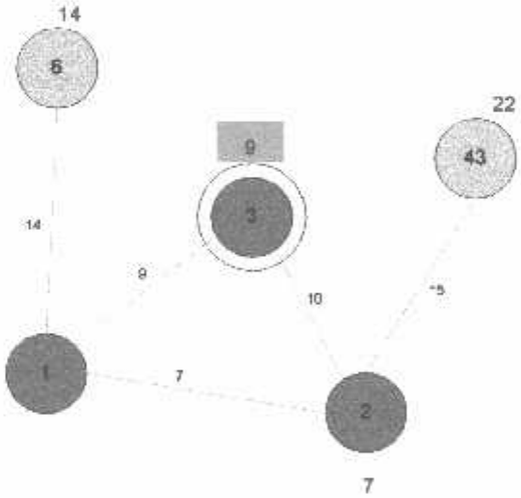
Gambar 2.10 Contoh kasus Djikstra - Langkah 1

2. Dijkstra melakukan kalkulasi terhadap node tetangga yang terhubung langsung dengan node keberangkatan (node 1), dan hasil yang didapat adalah node 2 karena bobot nilai node 2 paling kecil dibandingkan nilai pada node lain, nilai = 7 (0+7).



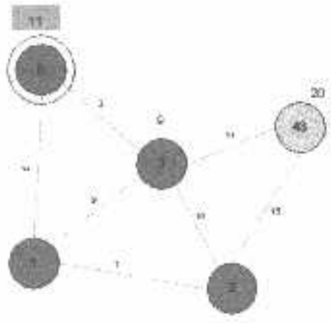
Gambar 2.11Contoh kasus Djikstra - Langkah 2

3. Node 2 diset menjadi node keberangkatan dan ditandai sebagi node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node yang telah terjamah. Dan kalkulasi dijkstra menunjukan bahwa node 3 yang menjadi node keberangkatan selanjutnya karena bobotnya yang paling kecil dari hasil kalkulasi terakhir, nilai 9 (0+9).



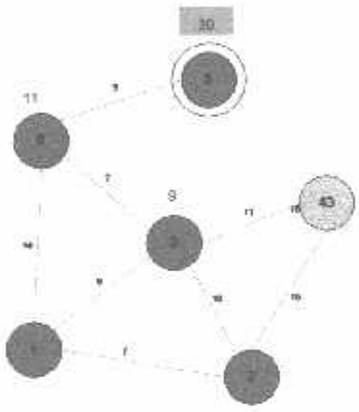
Gambar 2.12 Contoh kasus Djikstra - Langkah 3

4. Perhitungan berlanjut dengan node 3 ditandai menjadi node yang telah terjamah. Dari semua node tetangga belum terjamah yang terhubung langsung dengan node terjamah, node selanjutnya yang ditandai menjadi node terjamah adalah node 6 karena nilai bobot yang terkecil, nilai 11 (9+2).



Gambar 2.13 Contoh kasus Djikstra - Langkah 4

5. Node 6 menjadi node terjamah, dijkstra melakukan kalkulasi kembali, dan menemukan bahwa node 5 (node tujuan) telah tercapai lewat node 6. Jalur terpendeknya adalah 1-3-6-5, dan niilai bobot yang didapat adalah 20 (11+9). Bila node tujuan telah tercapai maka kalkulasi dijkstra dinyatakan selesai.



Gambar 2.14 Contoh kasus Djikstra - Langkah 5

BAB III
PERANCANGAN SISTEM

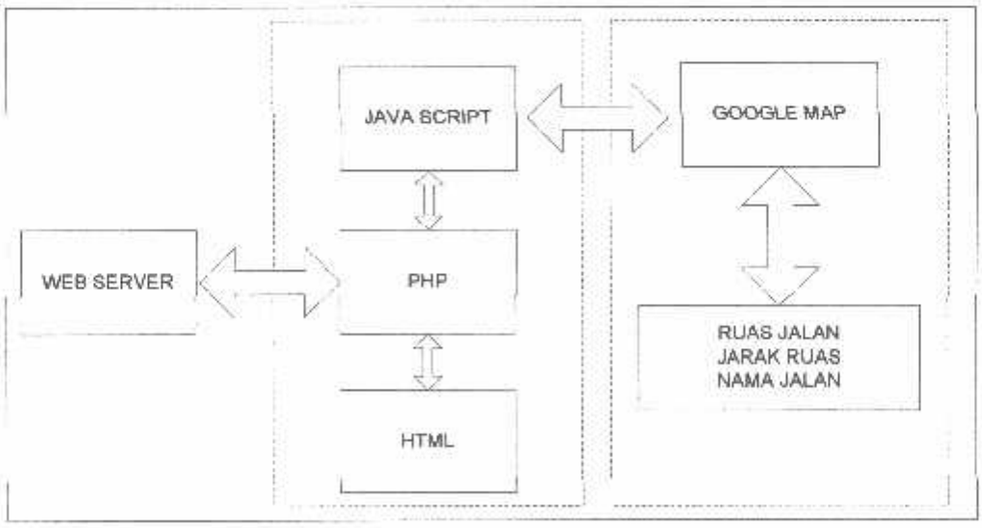
Pada bab ini dijelaskan mengenai analisis dan perancangan sistem aplikasi. Analisis ditujukan untuk memberikan gambaran secara umum terhadap aplikasi dan memberikan solusi terhadap permasalahan yang dihadapi. Dalam menyelesaikan masalah diatas digunakan metode dalam meneliti masalah yang dihadapi. Adapun metode penelitian yang digunakan adalah :

3.1 Metode Pengembangan

Model yang digunakan dalam pembuatan sistem pencarian rute terdekat ini adalah perancangan dan pembuatan. Dengan cara mengumpulkan informasi, pencarian data, informasi yang dikumpulkan berupa kondisi jalan di Kota Malang. Data yang dicari adalah data geografis jalan.

Metode yang digunakan dalam pencarian rute terdekat adalah *Algoritma Dijkstra* pencarian lintasan terpendek dari simpul tertentu ke semua simpul yang lain (*single-source shortest path*). Dengan metode *single-source shortest path,* maka akan didapat jarak terdekat ke semua node / simpul. Sehingga dapat dibandingkan untuk mendapatkan jarak yang paling dekat / terdekat. Algoritma ini lebih efisiensi waktu di banding algortima konvnesional lainnya (Floyd-Warshall, Bellman-Ford).

Aplikasi ini berbasis web (*web based*), sehinga para pekerja dapat mengakses melalui computer client yang berada di kantor. Dengan menentukan jalan atau lokasi tujuan maka pekerja dapat mengetahui rute terdekat yang bisa dilewati untuk menuju tujan pengiriman makanan. Output dari sistem, berupa informasi visual peta rute jalan dan jarak terdekat



Gambar 3.1 Desain implementasi sistem.

Pada gambar 3.1 merupakan desain arsitektur dari implementasi sistem sesungguhnya. Didalam implementasi sistem sesungguhnya terdapat tiga komponen utama yaitu: *server* yang memproses request dari client dan mengirimkan kembali hasil request (respon) ke client, *internet* sebagai jaringan yang menghubungkan client ke server dan *client* yang melakukan request ke server. Aplikasi yang di bangun akan di pasang pada Server, sehingga user / client hanya tinggal mengakses ke server untuk menjalankannya.

Permintaan user atas informasi rute terdekat pertama kali akan di terima oleh HTTP Server. Kemudian akan dilakukan pengecekan apakah ada script PHP yang perlu diproses atau dieksekusi. Script PHP akan diproses dan dieksekusi oleh java script dengan dukungan Google Map dan HTML Template sebagai data external melalui API yang disediakan, dan hasilnya akan dikirim kembali ke client oleh HTTP Server yang sudah berupa HTTP respon.

Di dalam server terdapat bagian utama yang berperan penting dalam kinerja aplikasi yang dibangun. Bagian-bagian tersebut saling terkait, yaitu:

a. Web Server : Yang menunggu request dari client dan mengirimkan respon ke admin setelah request di proses.
b. Map File : Merupakan file GIS yang digunakan untuk menajamen peta.(Google Map)
c. PHP : Sebagai aplikasi script yang mengeksekusi PHP Script.

3.2 Prosedur Pengembangan

Pada bagian ini prosedur yang dilakukan yaitu menganalisis dan merancang aplikasi.

3.3 Identifikasi Permasalahan

Tahap *identifikasi* masalah merupakan tahapan paling awal untuk melakukan perancangan dan pembuatan aplikasi. Tahapan ini digunakan untuk melakukan *observasi* atau penelusuran permasalahan untuk mendapatkan permasalahan umum dari permasalahan yang dihadapi. Didalam tahapan ini juga dilakukan perumusan permasalahan yaitu merumuskan atau menetapkan permasalahan yang dihadapi, sehingga penelitian dapat lebih fokus untuk mencari dan memecahkan permasalahan yang ada.

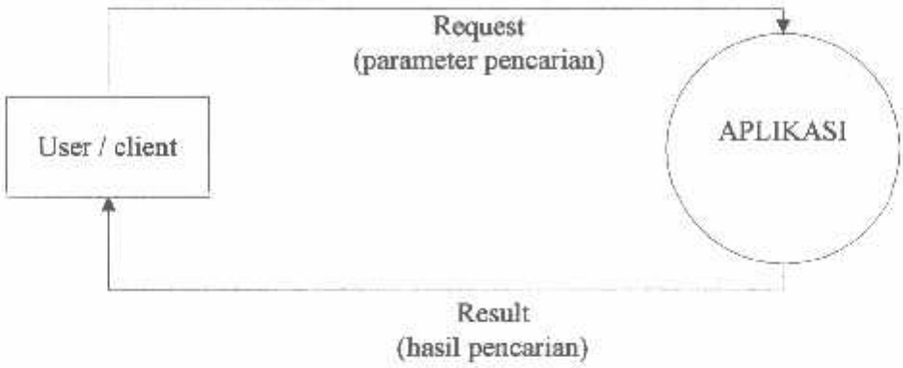
Pada tugas akhir ini, fokus usaha penyelesaian masalah terdapat pada perancangan dan pembuatan aplikasi yang mengimplementasikan GIS dalam pencarian rute terdekat dengan menggunakan algoritma Dijkstra.

3.4 Perancangan dan Pemodelan Sistem

Ada beberapa bagian penting dalam melakukan perancangan dan pemodelan suatu sistem, yaitu melakukan perancangan dan pembuatan alur sistem.

3.5 Context Diagram

Context diagram merupakan diagram pertama dalam rangkaian suatu DFD yang menggambarkan entitas-entitas yang berhubungan dengan suatu sistem.



Gambar 3.2 Context Diagram

Pada context diagram (gambar 3.2) terdapat hanya dua buah entitas yang berhubungan dengan sistem yaitu user atau client yang akan berinteraksi dan menjalankan sistem ini dengan memberikan paramater-parameter input yang digunakan dalam proses pencarian. Entitas kedua adalah *google map*, *google map* merupakan peta digital gratis yang dapat dimodifikasi.

Di dalam Web *based* GIS sangat sulit sekali untuk melakukan pengeditan object secara visual seperti pada *desktop based* GIS, hal ini-lah yang menjadi kelemahan *web based* GIS. Sehingga secara tidak langsung diperlukannya *desktop based* GIS untuk melakukan hal tersebut.

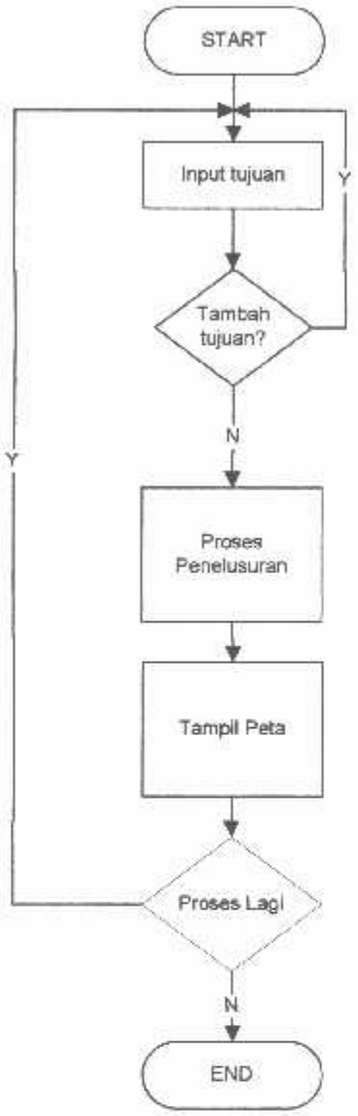
3.6 Alur Proses Aplikasi

Untuk lebih memperjelas alur dari sistem, maka alur proses aplikasi akan digambarkan pada *flowchart* dibawah. *Flowchart* alur proses tersebut menjelaskan rancangan urutan proses yang terjadi pada aplikasi. Dimulai dari proses pemasukan *input*, pencarian rute terdekat dengan algortima Dijkstra maupun proses lain, hingga output. Dimana inputan pertama berupa ruas jalan asal dan yang kedua adalah ruas jalan atau lokasi tujuan, tergantung keiinginan user.

Pada sistem ini untuk lokasi awal sudah ditentukan, sehingga user hanya memasukkan lokasi atau jalan tujuan saja untuk mencari rute terdekat. Untuk lokasi tujuan user bisa menginputkan lebih dari satu tujuan dan setelah proses penelusuran

system akan memberikan urutan rute dari lokasi tujuan yang terdekat sampai yang terjauh.

3.6.1 Flowchart Alur Aplikasi



Gambar 3.3 Flowchart Alur Aplikasi

Penjelasan flowchart alur aplikasi (gambar 3.3) adalah sebagai berikut:

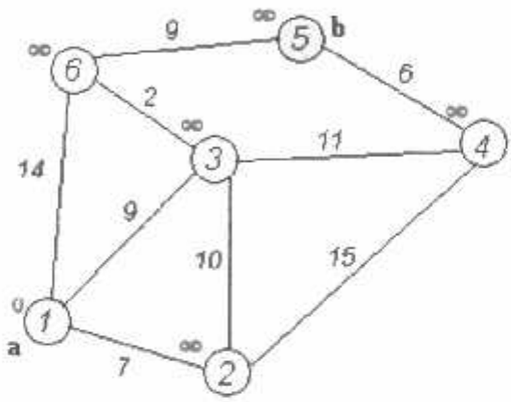
Dimana pada tahapan akan dimunculkan tampilan menu utama yang terdiri dari Tampilan Awal, serta terdapat tombol **tambahkan**, **proses**, **baru** serta **input teks** untuk inputan lokasi tujuan. Semua tampilan awal ini ditampilkan ke dalam bentuk web berbasis *local web*.

Pada tampilan ini terdapat sebuah proses dimana *user* atau memasukan nama jalan yang akan di tuju. Jika user memilih Tambahkan maka lokasi tujuan akan muncul

dipeta, dan *user* masih bisa menambahkan lokasi tujuan kembali. Setelah user selesai menentukan lokasi tujuan proses akan dilanjutkan ke tahapan selanjutnya (pencarian rute). Jika *user* tidak memilih tambahkan tujuan maka *user* dapat mencari rute dengan cara memilih proses.

3.7 Metode Pencarian Jalur Terpendek *(Dijkstra Algorithm)*

Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya seperti pada gambar 3.4. Misalkan titik mengambarkan gedung dan garis menggambarkan jalan, maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.



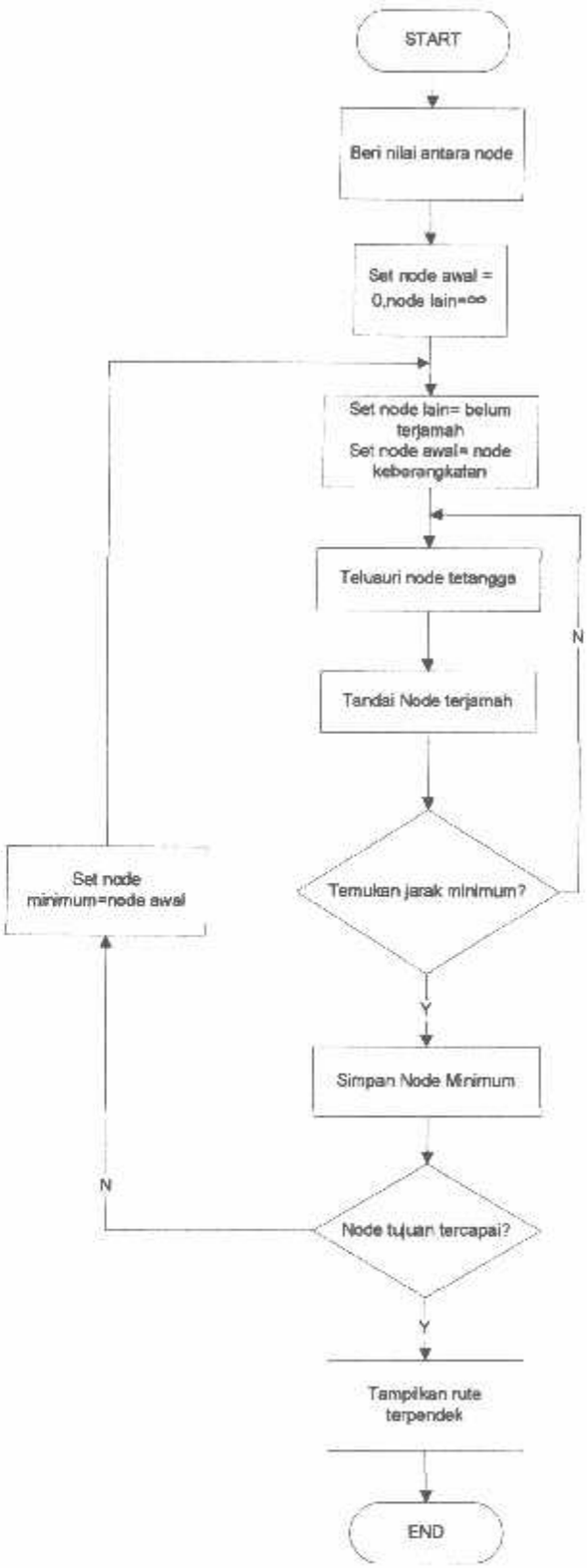
Gambar 3.4 Keterhubungan antar titik dalam algoritma Dijkstra

Pertama-tama tentukan titik mana yang akan menjadi node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu per satu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap. Inilah urutan logika dari algoritma Dijkstra:

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi)
2. Set semua node "Belum terjamah" dan set node awal sebagai "Node keberangkatan"
3. Dari node keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke node C berjarak 2, maka jarak ke C melewati B menjadi 6+2=8. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.

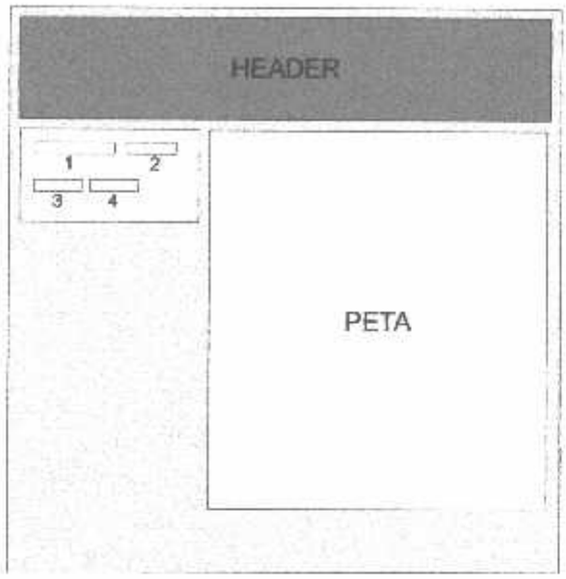
4. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai "Node terjamah". Node terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. Set "Node belum terjamah" dengan jarak terkecil (dari node keberangkatan) sebagai "Node Keberangkatan" selanjutnya dan lanjutkan dengan kembali ke step 3.

3.7.1 Flowchart Algoritma Dijkstra



Gambar 3.5 Flowchart algortima dijkstra

3.8 Desain *Interface* Aplikasi



Gambar 3.6 Rencana Tampilan Aplikasi

Pada desain tampilan aplikasi (Gambar 3.6) terdapat header yang terdapat pada bagian atas, yang berisi selamat datang. Setelah header di sebelah kiri terdapat 4 menu di simbolkan dengan angka 1,2,3, dan 4. Dimana yang pertama adalah input teks (untuk mengisikan alamat tujuan), kedua adalah TAMBAHKAN (Menambahkan peta kedalam pencarian). Ketiga adalah PROSES (Menampilkan rute terpendek), dan keempat adalah BARU (Melakukan pencarian rute baru). Ditengah halaman utama terdapat sebuah *frame*, dimana *frame* inilah yang akan digunakan untuk menampilkan sebuah data, peta digital.

BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini akan dijelaskan mengenai implementasi dan uji coba dari sistem pencarian rute terdekat.

4.1 Kebutuhan Sistem

Sebelum menjalankan program atau aplikasi secara standalone, ada beberapa hal yang perlu diperhatikan, antara lain kebutuhan sistem akan perangkat keras (*hardware*) dan perangkat lunak (*software*), serta langkah-langkah yang harus dilakukan untuk dapat melakukan instalasi aplikasi agar dapat berfungsi sebagaimana mestinya. Dalam perancangan dan pembuatan aplikasi ini ada beberapa perangkat keras dan lunak komputer yang dibutuhkan antara lain :

4.1.1 Perangkat Keras

Perangkat keras komputer adalah komponen-komponen fisik peralatan yang membentuk suatu sistem komputer, serta peralatan-peralatan lain yang mendukung komputer dalam menjalankan tugasnya. Adapun perangkat keras yang diperlukan dalam pengujian aplikasi ini adalah :

1. CPU dengan processor 1200 Mhz atau lebih
2. Monitor XGA
3. Memory 1 GB atau lebih.
4. VGA Card dengan memory 8 MB atau lebih.
5. Printer
6. Mouse, Keyboard dan CDROM.

4.1.2 Perangkat Lunak

Perangkat lunak yang diperlukan adalah program komputer yang diperlukan untuk mengoperasikan fungsi dari perangkat keras. Adapun perangkat lunak yang diperlukan dalam perancangan dan pembuatan aplikasi ini adalah :

1. Sistem Operasi Windows XP
2. Macromedia Dreamweaver 8
3. Macromedia Flash MX
4. Apache 2.0.55
5. PHP 4.4.0

6. Adobe Photoshop
7. Corel Draw

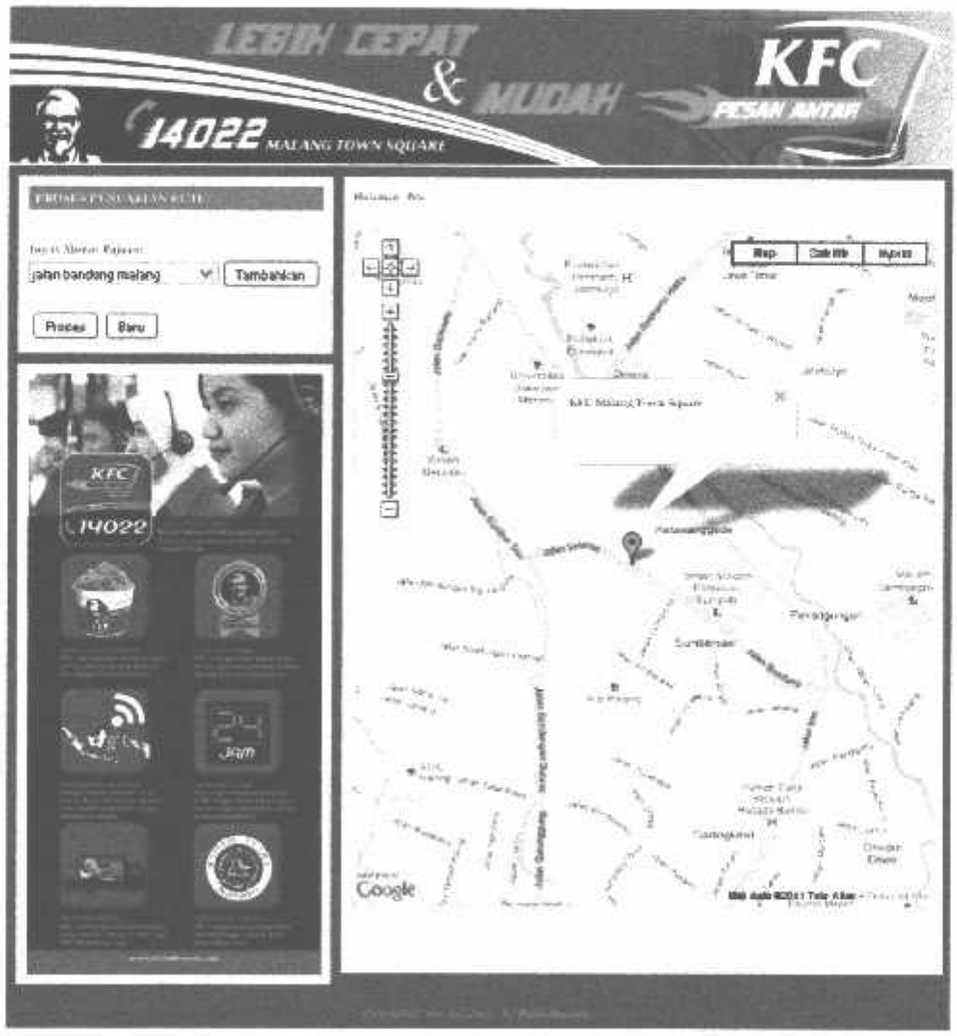
4.2 Penggunaan Aplikasi

Pada sub bab akan dijelaskan tentang penggunaan aplikasi per sistem menu, mulai dari tampilan aplikasi, fungsi dan cara penggunaannya.

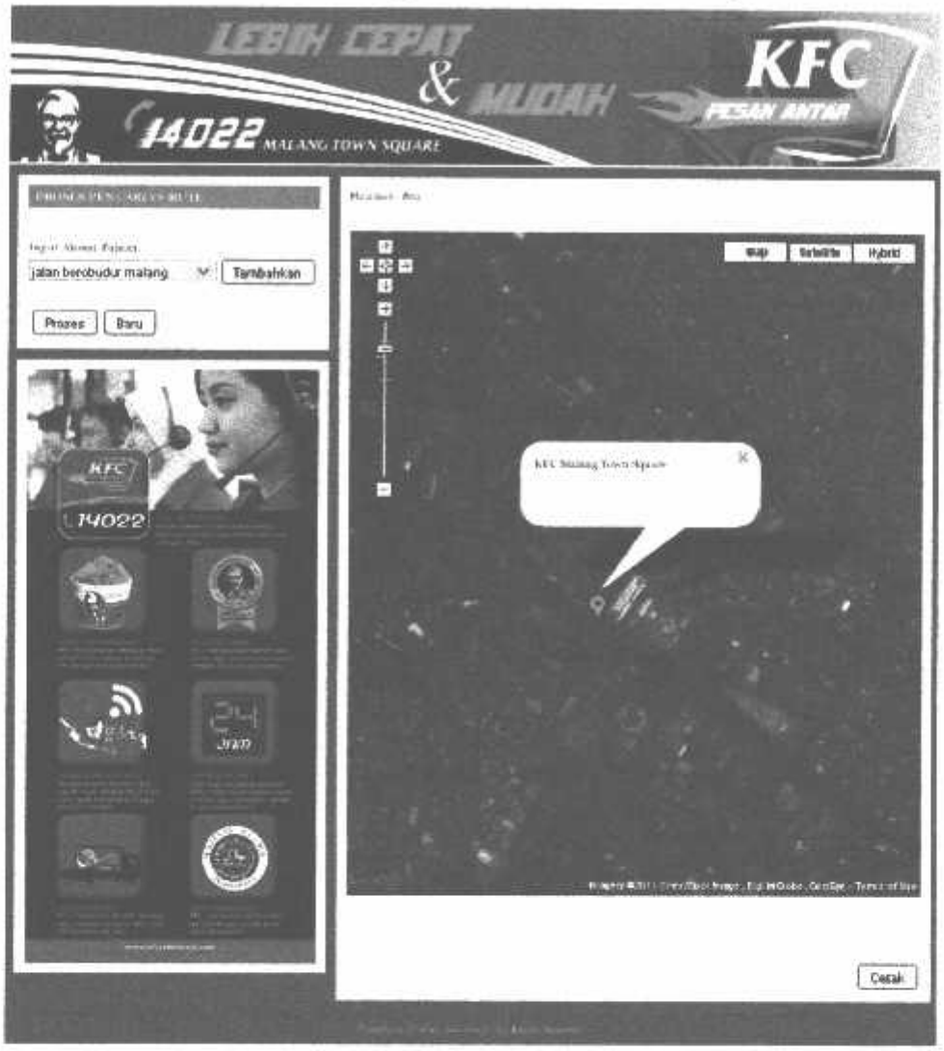
4.2.1 Tampilan Menu Utama

Tampilan menu utama merupakan tampilan pembuka pada saat aplikasi pertama kali dijalankan. Di dalamnya terdapat menu utama dan sub menu dari masing-masing menu utama.

Pada gambar 4.1 tampilan peta yang dipilih adalah mode map, sehingga yang ditampilkan hanya jalan dan namanya dan pada gambar 4.2 mode yang dipilih adalah mode satelite sehingga keadaan geografisnya terlihat dengan jelas.



Gambar 4.1 Tampilan Menu Utama Mode Map



Gambar 4.2 Tampilan Menu Utama Mode Satelite

Dalam pengujian program aplikasi pada Tugas Akhir ini dilakukan dengan tujuan untuk mengetahui kinerja program, berupa ketepatan analisa antara perencanaan dan hasil akhir yang diperoleh. Seperti pada gambar (Gambar 4.1dan 4.2) yang merupakan hasil dari tampilan pada aplikasi ini.

4.2.2 Tampilan Pencarian Rute Terdekat

4.2.2.1 Satu Tujuan Pengiriman

Di dalam halaman ini kita dapat melakukan pencarian rute jalan terdekat. Parameter pencarian adalah titik jalan asal dan titik jalan tujuan. Titik tujuan dapat dipih pada input teks.

Hasil pencarian adalah jarak terdekat, jalan yang akan dilewati pada pencarian rute terdekat akan ditampilkan ke dalam peta dengan blok berwarna "biru" pada ruas jalan.

Di halam ini juga terdapat tools untuk melakukan menejemen peta seperti halnya standart aplikasi GIS yang lain, diantaranya adalah :

a. Pan : Untuk menggeser tampilan peta.

b. Zoom in : Untuk perbesaran tampilan peta.

c. Zoom out : Untuk perkecilan tampilan peta.

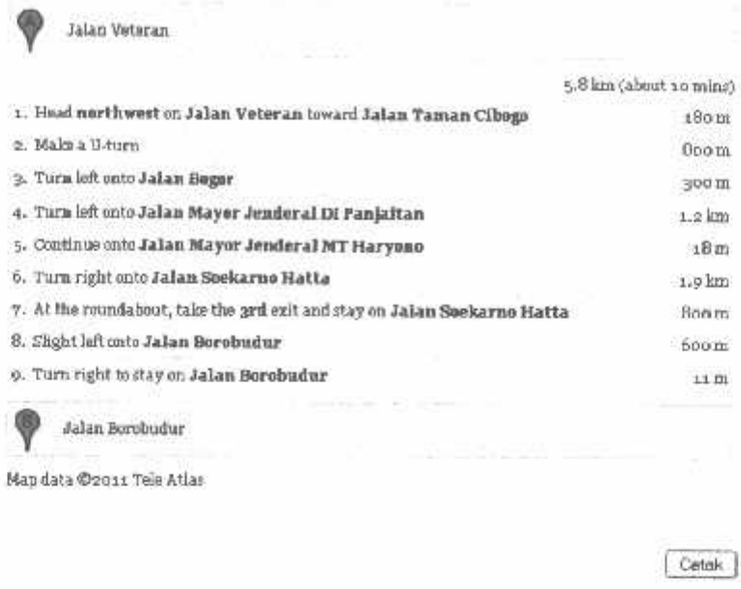
d. Perbesaran : Untuk berapa kali peta akan di perbesar atau diperkecil.

Untuk input daerah tujuan sudah dibatasi,sehingga daerah tujuan yang bisa dicari hanya yang sudah ditentukan seperti pada gambar 4.3.

Gambar 4.3 Tampilan input tujuan

Gambar 4.4 Tampilan peta setelah proses pencarian rute terpendek

Pada gambar 4.4 adalah tampilan rute terpendek yang ditemukan sistem untuk daerah tujuan jalan Borobudur.

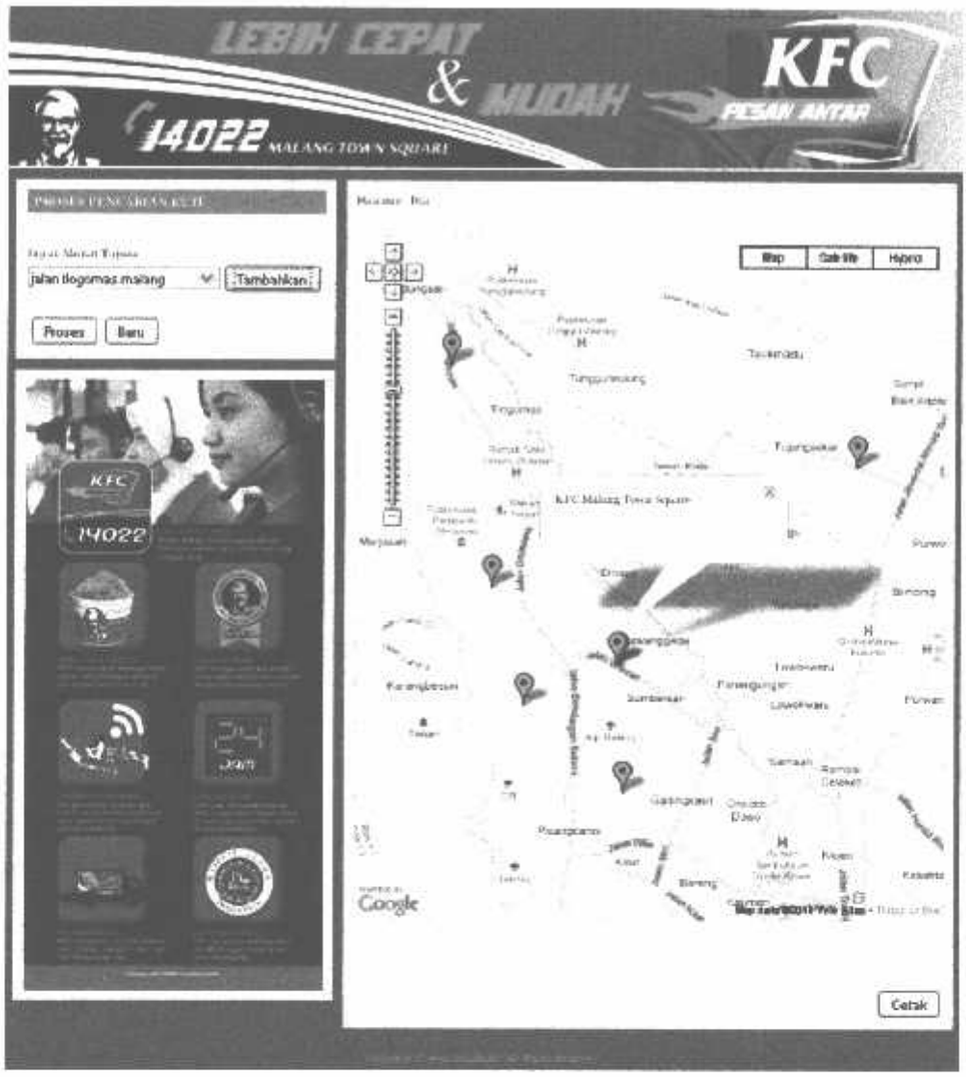


Gambar 4.5 Tampilan rute jalan yang di lewati

Pada gambar 4 5 adalah urutan jalaur terpendek yang bisa dilalui oleh pekerja untuk menuju jalan Borobudur.

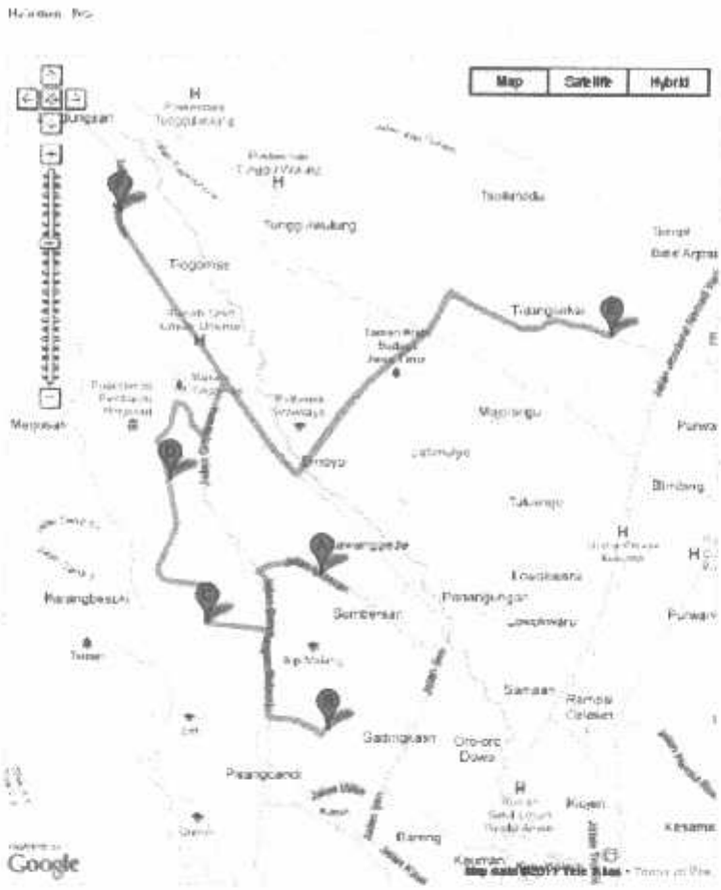
4.2.2.2 Dua Atau Lebih Tujuan Pengiriman

Banyaknya tuntutan pelayanan yang semakin meningkat, penyedia jasa diperhadapkan dengan perencanaan rute pelayanan dan penanggulangan permintaan antar makanan,oleh karena itu untuk sekali pengiriman,pekerja jasa pesan antar tidak hanya membawa pesanan untuk satu tujuan. Sehingga pada sistem ini bisa melakukan pencarian rute terpendek untuk beberapa tujuan.



Gambar 4.6 Tampilan Input Tujuan

Pada gambar 4.6 ini adalah tampilan ketika masih terjadi proses penambahan lokasi pengiriman,jadi yang terlihat hanya titik-titik lokasi tujuan yang terlihat



Gambar 4.7 Tampilan Setelah Proses Pencarian Rute Terdekat

Pada gambar 4 4 adalah tampilan rute terpendek yang ditemukan sistem untuk daerah tujuan jalan Borobudur.

Jalan Veteran

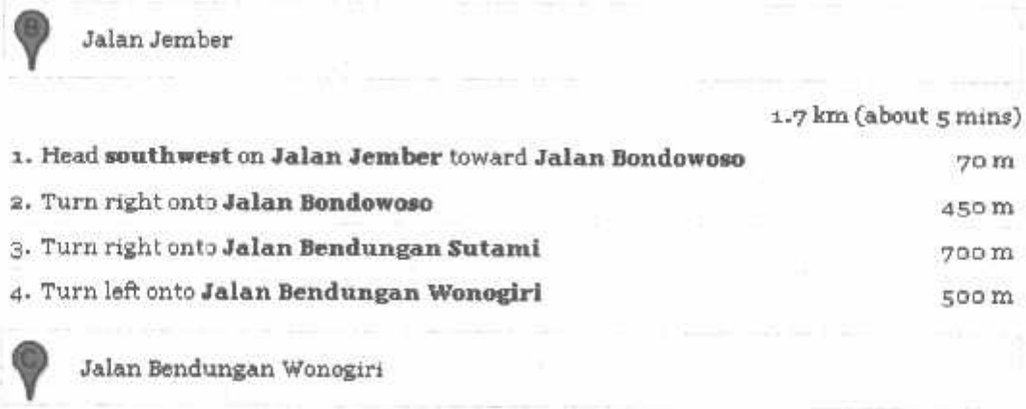
2.3 km (about 4 mins)

1. Head **northwest** on **Jalan Veteran** toward **Jalan Taman Cibogo** 550 m
   Go through 1 roundabout
2. Turn left onto **Jalan Bendungan Sutami** 1.2 km
3. Turn left onto **Jalan Bondowoso** 450 m
4. Turn left onto **Jalan Jember** 70 m

Jalan Jember

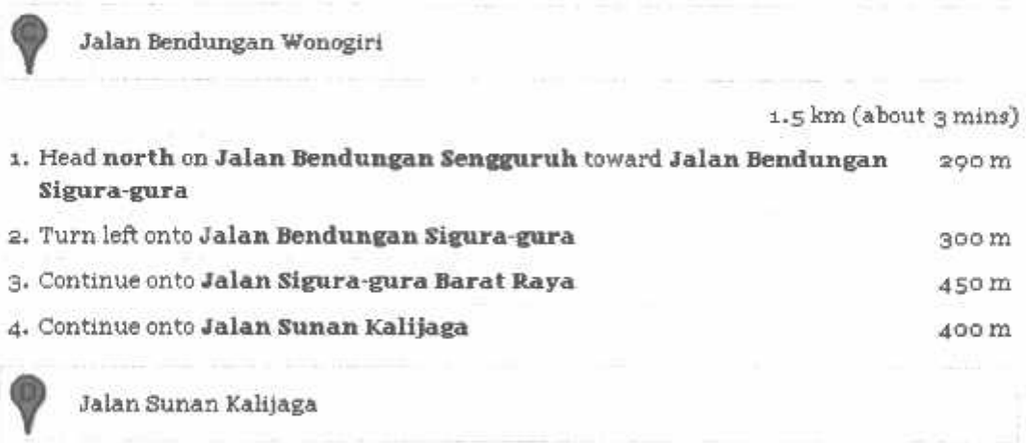
Gambar 4.8 Rute Tujuan Pertama

Pada gambar 4.8 adalah urutan jalaur terpendek yang bisa dilalui oleh pekerja untuk menuju jalan Jember yang merupakan tujuan terdekat.



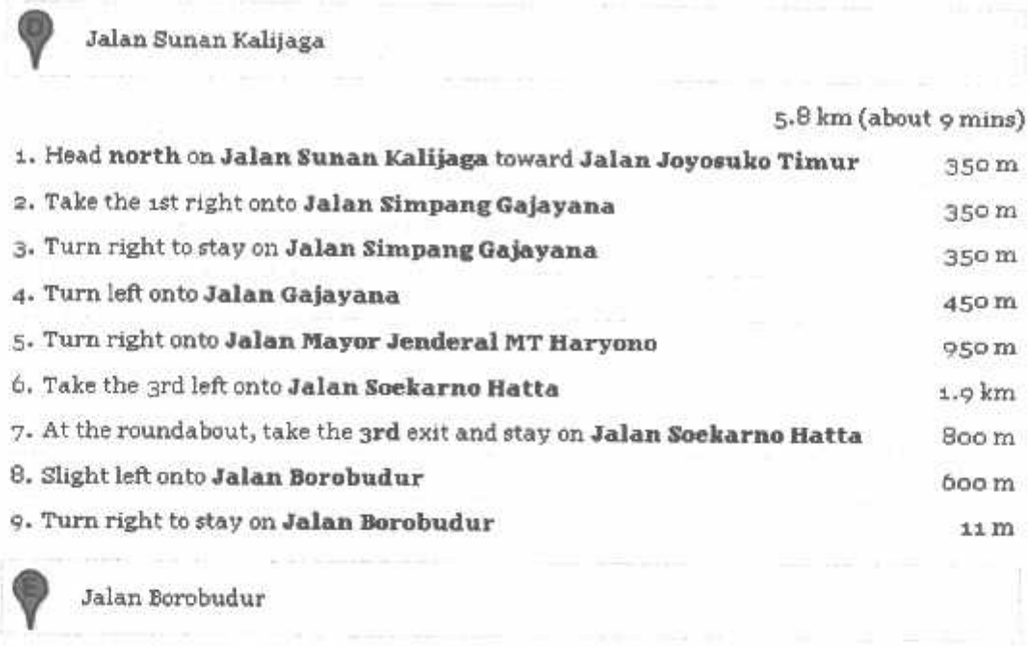
Gambar 4.9 Rute Tujuan Kedua

Pada gambar 4 9 adalah urutan jalaur terpendek yang bisa dilalui oleh pekerja dari jalan Jember menuju jalan Bendungan Wonogiri.



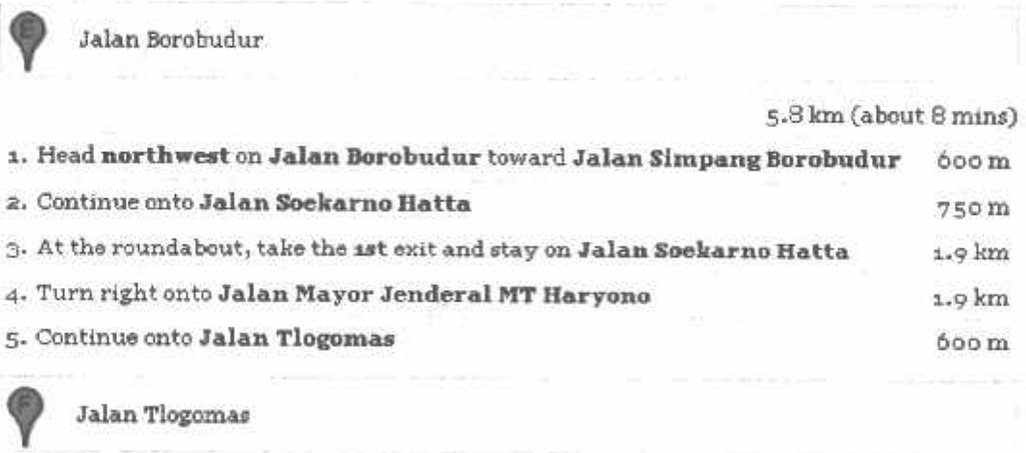
Gambar 4.10 Rute Tujuan Ketiga

Pada gambar 4.10 adalah urutan jalaur terpendek yang bisa dilalui oleh pekerja dari jalan Bendungan Wonogiri menuju jalan Sunan Kalijaga.



Gambar 4.11 Rute Tujuan Keempat

Pada gambar 4.10 adalah urutan jalaur terpendek yang bisa dilalui oleh pekerja dari jalan Sunan Kalijaga menuju jalan Borobudur.



Gambar 4.12 Rute Tujuan Kelima

Pada gambar 4.10 adalah urutan jalaur terpendek yang bisa dilalui oleh pekerja dari jalan Borobudur menuju jalan Tlogomas yang merupakan tujuan terakhir.

4.3 Pengujian Sistem

Pengujian dilakukan bertujuan untuk mengetahui apakah aplikasi yang telah dibangun telah berjalan dengan baik dan memenuhi spesifikasi yang telah ditentukan. Pengujian pada aplikasi *web based* GIS penentuan rute terdekat berfokus pada kemampuan pencarian rute, kinerja algoritma *Dijkstra* yang digunakan dan kemampuan dari GIS pada aplikasi.

4.3.1 Pengujian Perbandingan Jarak Hasil Pada Peta Dengan Pengukuran

Di dalam pengujian kali ini akan dilakukan pencarian rute terdekat dari KFC Malang Town Square (JL.Veteran) menuju beberapa lokasi tujuan. Yang dijadikan perbandingan adalah jarak hasil pencarian (Jarak Hasil) pada sistem dengan jarak sesungguhnya hasil pengukuran Spedometer (Jarak Real).

Tabel 4.1 Perbandingan Jarak Peta Dengan Jarak Pengukuran

No	Asal	Tujuan	Jalan yang dilewati	Jarak pada peta (m)	Jarak pengukuran (m)
1	KFC Malang Town Square (JL.Veteran)	Jl.Borobudur	Jalan Taman Cibogo 800m Jalan Bogor 300 m Jalan Mayor Jenderal DI Panjaitan 1.2 km Jalan Mayor Jenderal MT Haryono 18 m Jalan Soekarno Hatta 1.9 km Jalan Borobudur	4218	4500
2		Jl.Kawi	Jalan Bandung 550 m jalan Ijen 1.2 km Jalan Semeru 650 m Jalan Tangkuban Perahu 100m Jalan Kawi	2500	2700

3		Jl.Mertojoyo	Jalan Veteran 550 m Jalan Sumber Sari 750 m Jalan Gajayana 500 m Jalan Simpang Gajayana 350 m Jalan Joyo Tambak Sari 350 m Jalan Mertojoyo	2500	2300
4		Jl.Raya Dieng	Jalan Veteran 550 m Jalan Bendungan Sutami1.2 km Jalan Galunggung500 m Jalan Bukit Barisan350 m Jalan Halimun 110 m Jalan Taman Agung 84 m Jalan Simpang Dieng Utara 140 m Jalan raya dieng	2934	3200
5		Jl.L.Jenderal Sutoyo	Jalan Veteran 180 m Jalan Bandung 550 m Jalan Brigadir Jenderal Slamet Riyadi 1.5 km Jalan Jenderal Basuki Rahmat 160 m Jalan Jaksa Agung Suprapto 950 m Jalan Monginsidi 120 m Jalan Letnan Jenderal Sutoyo	3460	3100
6		Jl.Raya Tidar	Jalan Veteran 550 m Jalan Bendungan Sutami 1.2 km Jalan Tidar Raya	1750	1600

7		Jl.Sudimoro	Jalan Veteran 180 m Jalan Bogor 300 m Jalan Mayor Jenderal DI Panjaitan1.2 km Jalan Soekarno Hatta1.9 km Jalan Sudimoro	3580	3400
8		Jl.Tlogomas	Jalan Veteran 550 m Jalan Sumber Sari750 m Jalan Gajayana1.0 km Jalan Mayor Jenderal MT Haryono1.0 km Jalan Tlogomas	3300	3100
9		Jl.Bareng	Jalan Veteran 180 m Jalan Bandung 550 m Jalan Ijen1.5 km Jalan Terusan Ijen180 m Jalan Bareng	2410	2200
10		Jl.Anjasmoro	Jalan Veteran 180 m Jalan Bandung550 m Jalan Brigadir Jenderal Slamet Riyadi400 m Jalan Panggung38 m Jalan Anjasmoro	1168	1100

Setelah dilakukan pengujian sebanyak 10 kali seperti pada tabel 4.1, maka didapat perbandingan jarak hasil pencarian dari sistem dengan jarak sesungguhnya hasil pengukuran *Spedometer*. Hasilnya adalah sebagai berikut

Selisih jarak peta dengan Jarak Pengukuran : [Jarak Peta-Jarak Pengukuran]

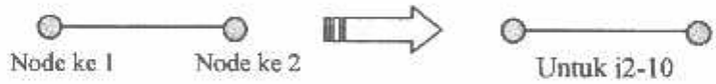
1. Percobaan No 1 : [4218-4500] = 282
2. Percobaan No 2 : [2500-2700] = 200
3. Percobaan No 3 : [2500-2300] = 200
4. Percobaan No 4 : [2934-3200] = 266

5. Percobaan No 5 : [3460-3100] = 360
6. Percobaan No 6 : [1750-1600] = 150
7. Percobaan No 7 : [3580-3400] = 180
8. Percobaan No 8 : [3300-3100] = 200
9. Percobaan No 9 : [2410-2200] = 210
10. Percobaan No10 : [1168-1100] = 68

Berdasarkan hasil perhitungan diatas maka selisih rata-rata jarak pada peta dengan jarak pengukuran adalah ± 211,6 meter.

4.3.2 Pengujian Algoritma Dijkstra (*single-source shortest path*)

Untuk pengujian kali ini dilakukan dua pencarian dari Jl.Veteran ke Jl.Ikan Nila dan sebaliknya, dengan ketentuan rute hasil pencarian nantinya tidak ada ruas jalan tipe 1 arah. Hasil uji berupa tree jarak terdekat dari node asal ke semua node yang ada. Tree tersebut merupkan outputan dari Dijkstra (*Single-source shortest path*). Seperti dalam pembahasan sebelumnya bahwa inputan dari Algoritma Diskstra adalah graph berbobot, node asal dan node tujuan. Graph berbobot dapat diperoleh (diload) dari data atribut GIS (tabel node). Sedangkan node asal dan tujuan didapat dari ruas asal dan tujuan. Dalam presentasinya satu ruas terdiri dari dua node yang dihubungkan oleh garis penghubung, sebagai gambaran akan di jelaskan pada gambar sebagai berikut.



Gambar 4.13 Pemisahan node pada ruas

Di pengujian kali ini diambil node ke satu dari ruas asal atau tujuan sebagai node asal dan tujuan.

4.3.2.1 Pencarian dari Jl.Veteran ke Jl.Ikan Nila

Seperti pada penjelasan di atas maka di dapat node asal Jl.Veteran dan node tujuan Jl.Ikan Nila.

Dari hasil pencarian dapat diketahui jarak dan rute terdekat dari Jl.Veteran ke Jl.Ikan Nila. (yang terblok warna biru pada tabel), adalah sebagai berikut:

a. Jarak terdekat : 7 233 m

b. Rute terdekat : Jalan Veteran- Jalan Bogor- Jalan Mayor Jenderal DI Panjaitan- Jalan Mayor Jenderal MT Haryono- Jalan Soekarno Hatta- Jalan Sudimoro- Jalan Ikan

Kakap- Jalan Ikan Piranha Atas- Perumahan Tunjung Sekar- Jalan Ikan Belinda 1-Jalan Ikan Nila.

4.3.2.2 Pencarian dari Jl.Ikan Nila ke Jl.Veteran

Seperti pada penjelasan di atas maka di dapat titik asal Jl.Ikan Nila dan titik tujuan. Jl.Veteran,pengujian ini dilakukan sebelum posisi awal di KFC Malang Town Square (Jl.Veteran)

Dari hasil pencarian dapat diketahui jarak dan rute terdekat dari Jl.Veteran ke Jl.Ikan Nila. (yang terblok warna biru pada tabel), adalah sebagai berikut:

a. Jarak terdekat : 7 233 m
b. Rute terdekat : Jalan Ikan Nila - Jalan Ikan Belinda 1- Perumahan Tunjung Sekar - Jalan Ikan Piranha Atas - Jalan Ikan Kakap - Jalan Sudimoro - Jalan Soekarno Hatta - Jalan Mayor Jenderal MT Haryono - Jalan Mayor Jenderal DI Panjaitan - Jalan Bogor - Jalan Veteran

4.3.2.3 Perbandingan Pencarian (Jl.Veteran ke Jl.Ikan Nila) dan (Jl.Ikan Nila ke Jl.Veteran)

Dari hasil pencarian di atas (poin 4.3.2.1 dan 4.3.2.2) di dapat jarak dan rute terdekat yang sama. Hal ini menunjukkan bahwa pengujian algortima Dijkstra berjalan dengan baik. Dapat dikatakan *dijkstra* bekerja sesuai dengan hasil yang diharapkan dan sesuai dengan spesifikasi yang telah ditentukan.

BAB V
PENUTUP

5.1 Kesimpulan

Setelah melakukan perancangan dan pembuatan aplikasi *web based* GIS pencarian rute terdekat serta pengujian aplikasi, maka kesimpulan yang dapat diambil adalah sebagai berikut :

1. Dengan adanya sistem ini dapat mempermudah pekerja pesan antar untuk mengetahui lokasi pemesan serta mengetahui jalan-jalan terdekat untuk menuju lokasi pengiriman.
2. Daerah pencarian rute hanya bisa dilakukan untuk daerah yang berjarak atau berada pada radius ±5 kilo meter.
3. Pada perbandingan hasil pengukuran jalan menggunakan *spedometer* dengan pengukuran pada sistem terdapat selisih rata-rata ± 211.6 meter.

5.2 Saran

Aplikasi pencarian rute terpendek ini masih dapat dikembangkan lebih jauh lagi karena dalam pembuatannya masih jauh dari sempurna dan masih banyak kekurangan. Adapun saran yang dapat dikemukakan agar aplikasi ini bisa berfungsi dengan lebih optimal adalah:

1. Aplikasi ini diletakan hanya pada satu cabang, oleh karena itu tidak menutup kemungkinan ada penambahan sistem untuk cabang-cabang lain agar para pekerja pesan antar bisa terbantu.
2. Aplikasi ini hanya sebatas atau terfokus membahas tentang pencarian jalan atau jalur terpendek. Mengingat luasnya sistem sebagai pertimbangan dapat dipertimbangkan dan dikembangkan menjadi sistem informasi manajemen yang juga mencangkup pemesanan makanan secara online dan lain-lain.

DAFTAR PUSTAKA

[1] (http://id.wikipedia.org/wiki/Sistem_informasi_geografis, diakses tanggal 21 Februari 2011)

[2] Dijkstra's Algorithm". Rajagopalan, Sriram. http://www.eas.asu.edu/trace/eee459/sriram.html. diakses tanggal 3 Juli 2011

[3] Hernita P. 2009. Panduan Praktis Adobe Dreamweaver cs4. Yogyakarta: Andi.

[4] (*http://en.giswiki.net/wiki/Dijkstra%27s_algorithm*, diakses tanggal 24 Februari 2011)

[5] (*http://www.artfulsoftware.com/infotree/queries.php#766*, diakses tanggal 20 April 2011)

[6] (*http://id.wikipedia.org/wiki/Algoritma_Dijkstra*, diakses tanggal18 Mei 2011)

[7] (*http://www.4shared.com/file/a_ZVqAkB/Dijkstra.html*, diakses tanggal 18 Mei 2011)

[8] (*http://www.kfcindonesia.com/index.php?option=com_content&task=view&id=128&Itemid=231&limit=1&limitstart=1*, diakses tanggal 15 Juni 2011)

[9] Kadir, Abdul, 2003.Dasar Pemrograman Web Dinamis Menggunakan PHP, Penerbit ANDI, Yogyakarta.

[10] Prahasta, E, 2001. *Konsep-konsep Dasar Sistem Informasi Geografis*. Bandung. CV. Informatika

LAMPIRAN

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I Jl. Bandungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

NAMA : TITO TRI WAHYU ABADI
NIM : 06.12.659
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer dan Informatika
MASA BIMBINGAN : 4 Juli 2011 s/d 4 Januari 2012
JUDUL : **SISTEM PENCARIAN JALUR TERPENDEK UNTUK JASA PESAN ANTAR PADA PT.FASTFOOD INDONESIA (KFC) CABANG MALANG TOWN SQUARE**

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Sabtu
Tanggal : 13 Agustus 2011
Dengan Nilai : 82,75 (A)

PANITIA UJIAN SKRIPSI

KETUA,

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

SEKRETARIS,

Dr. Eng. Aryuanto S, ST, MT
NIP.P.1030800417

ANGGOTA PENGUJI

Dosen Penguji I

M. Ibrahim Ashari, ST. MT
NIP.P.1030100358

Dosen Penguji II

Ahmad Faisol, ST
NIP.P.1031000431

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Karanglo KM.2 Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi Teknik Komputer & Inforamtika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama	:	TITO TRI WAHYU ABADI
Nim	:	06.12.659
Jurusan	:	Teknik Elektro S-1
Konsentrasi	:	Teknik Komputer & Informatika
Judul	:	**SISTEM PENCARIAN JALUR TERPENDEK UNTUK JASA PESAN ANTAR PADA PT.FASTFOOD INDONESIA (KFC) CABANG MALANG TOWN SQUARE**

No.	Penguji	Tanggal	Uraian	Paraf
1	Penguji I	13/08/2011	Tambahkan keterangan pada gambar dan tabel	
2	Penguji II	13/08/2011	Tentukan wilayah antar KFC cabang MATOS agar tidak lebih dari 5km	
			Tambahkan kesimpulan	
			Tambahkan ciri khas KFC cabang MATOS	

Disetujui :

Penguji I

M. Ibrahim Ashari, ST. MT
NIP.P.1030100358

Penguji II

Ahmad Faisol, ST
NIP.P.1031000431

Mengetahui :

Dosen Pembimbing I

Irmalia Suryani Faradisa, ST. MT
NIP.P.1030000365

Dosen Pembimbing II

Sandy Nataly Mantja, S.Kom
NIP.P.1030800418

INSTITUT TEKNOLOGI NASIONAL
JL. RAYA KARANGLO KM.2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : TITO TRI WAHYU ABADI
Nim : 06.12.659
Masa Bimbingan : 4 Juli 2011 s/d 4 Januari 2012
Judul Skripsi : SISTEM PENCARIAN JALUR TERPENDEK UNTUK JASA PESAN ANTAR PADA PT.FASTFOOD INDONESIA (KFC) CABANG MALANG TOWN SQUARE

NO.	TANGGAL	URAIAN	PARAF PEMBIMBING
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			

Malang,
Dosen Pembimbing I

Irmalia Suryani Faradisa,ST,MT
NIP: P.1030000365

Form S-4B

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, Km. 2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : **TITO TRI WAHYU ABADI**
Nim : **06.12.659**
Masa Bimbingan : **4 Juli 2011 s/d 4 Januari 2012**
Judul Skripsi : **SISTEM PENCARIAN JALUR TERPENDEK UNTUK JASA PESAN ANTAR PADA PT.FASTFOOD INDONESIA (KFC) CABANG MALANG TOWN SQUARE**

NO.	TANGGAL	URAIAN	PARAF PEMBIMBING
1	14 Juli 2011	Perbaiki BAB I dan BAB II	
2	18 Juli 2011	Perbaiki BAB I dan BAB II	
3	22 Juli 2011	BAB I dan BAB II ok,Perbaiki BAB III	
4	23 Juli 2011	Demo program ok, lanjut BAB IV & V	
5	25 Juli 2011	Perbaiki BAB IV & V	
6	26 Juli 2011	BAB IV dan BAB V ok , lanjut makalah	
7	27 Juli 2011	Makalah ok	

Malang, September 2011

Dosen Pembimbing II,

Sandy Nataly Mantja,S.Kom
NIP. P.1030800418

Form S-4b

Institut Teknologi Nasional Malang
Fakultas Teknologi Industri
Jurusan Teknik Elektro S-1

BERITA ACARA SEMINAR PROPOSAL SKRIPSI
JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika/ Teknik Komputer & Informatika*)

1	Nama Mahasiswa: TITO TRI WAHYU A?ADI			Nim
2	Keterangan	Tanggal	Waktu	Tempat
	Pelaksanaan	04 Juli 2017		Ruang:
3	Spesifikasi Judul (berilah tanda silang)**)			
	a Sistem Teraga Elektrik b Energi & Konversi Energi c Tegangan Tinggi & Pengukuran d Sistem Kendali Industri		e. Elektronika & Komponen f Elektronika Digital & Komputer g Elektronika Komunikasi h lainnya	
4	Judul Proposal yang diseminarkan Mahasiswa	SISTEM [illegible] JALUR TERCEPAT UNTUK JASA PESAN ANTAR PADA PT. FASTEVER INDONESIA (FTC) CABANG MALANG [illegible]		
5	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian	"Jalur tercepat" diganti "Jalur terdekat"		
6.	Catatan:			
	Catatan:			

7	Disetujui, Dosen Keahlian I	Disetujui Dosen Keahlian II	Disetujui, Dosen Keahlian III Soetyohadi, ST
	Mengetahui, Ketua Jurusan, Ir. Yusuf Ismail Nakhoda,MT NIP. Y 1018800189	Disetujui, Calon Dosen Pembimbing ybs Pembimbing I	Pembimbing II

Perhatian.
1. Keterangan *) Coret yang tidak perlu

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

INSTITUT TEKNOLOGI NASIONAL MALANG
PASCASARJANA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting) Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 08 Juli 2011

Nomor : ITN-359/I.TA/2/11
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr./i. **IRMALIA S. FARADISA, ST, MT**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama	: TITO TRI WAHYU. \
Nim	: 0612650
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai tanggal :

04 Juli 2011 s/d 04 Januari 2012

Sebagai satu syarat untuk ...

...nya kami sampaikan terima

Ketua Jurusan
Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT
Nip. Y 1018800189

Tembusan Kepada Yth :
1. Mahasiswa Yang Bersangkutan
2. Arsip
3. Coret yang tdak perlu

Form S 4a

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
IT BNI (PERSERO) MALANG
BANK NIAGA MALANG
kampus I : Jl. Bendungan Sigura-gura No.2 Telp. (0341) 551431 (Hunting) Fax. (0341) 553015 Malang 65145
kampus II : Jl. Raya Karanglo Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 08 Juli 2011

Nomor : ITN-360/I.TA/2/11
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr./i. **SANDY NATALY M, S.KOM**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama	: TITO TRI WAHYU. A
Nim	: 0612659
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai tanggal :

04 Juli 2011 s/d 04 Januari 2012

Sebagai salu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih

Ketua Jurusan
Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT
Nip. Y.1018800189

Tembusan Kepada Yth :
1. Mahasiswa Yang Bersangkutan
2. Arsip
3. Coret yang tidak perlu

Form S 4a

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, Km. 2
MALANG

Lampiran : 1 (satu) berkas
Pembimbing Skripsi

Kepada : Yth. Ibu Irmalia Suryani Faradisa, ST.MT
Dosen Institut Teknologi Nasional
MALANG

Yang bertanda tangan di bawah ini :

Nama : Tito Tri Wahyu Abadi
Nim : 06.12.659
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing Utama / Pendamping *), untuk penyusunan Skripsi dengan judul (proposal terlampir) :

SISTEM PENCARIAN JALUR TERCEPAT UNTUK JASA PESAN ANTAR PADA PT.FASTFOOD INDONESIA (KFC) CABANG MALANG TOWN SQUARE

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Tugas Akhir Sarjana Teknik.
Demikian permohonan kami dan atas kesediaan Bapak kami Ucapkan terima kasih.

Malang, juni 2011

Mengetahui
Ketua Jurusan Teknik Elektro

Ir Ir. Yusuf Ismail Nakhoda,MT
NIP. Y. 1018800189

Hormat Kami,

Tito Tri Wahyu A

*) coret yang tidak perlu

Form S-3a

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, Km. 2
MALANG

Lampiran : 1 (satu) berkas
Pembimbing Skripsi

Kepada : Yth. Ibu Sandy Nataly Mantja,S.kom
Dosen Institut Teknologi Nasional
MALANG

Yang bertanda tangan di bawah ini :

Nama	: Tito Tri Wahyu Abadi
Nim	: 06.12.659
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Komputer & Informatika

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing Utama / Pendamping *), untuk penyusunan Skripsi dengan judul (proposal terlampir) :

SISTEM PENCARIAN JALUR TERCEPAT UNTUK JASA PESAN ANTAR PADA PT.FASTFOOD INDONESIA (KFC) CABANG MALANG TOWN SQUARE

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Tugas Akhir Sarjana Teknik.
Demikian permohonan kami dan atas kesediaan Bapak kami Ucapkan terima kasih.

Malang, juni 2011

Mengetahui
Ketua Jurusan Teknik Elektro

Ir Ir. Yusuf Ismail Nakhoda,MT
NIP. Y. 1018800189

Hormat Kami,

Tito Tri Wahyu A

*) coret yang tidak perlu

Form S-3a

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, Km. 2
MALANG

PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama	: TITO TRI WAHYU ABADI
NIM	: 0612659
Semester	:
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: ~~TEKNIK ELEKTRONIKA~~ ~~TEKNIK ENERGI LISTRIK~~ TEKNIK KOMPUTER DAN INFORMATIKA ~~TEKNIK KOMPUTER~~ ~~TEKNIK TELEKOMUNIKASI~~
Alamat	:

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat ***SKRIPSI Tingkat Sarjana.*** Untuk melengkapi permohonan tersebut, bersama ini kami lampirkan persyaratan-persyaratan yang harus dipenuhi.
Adapun persyaratan-persyaratan pengambilan ***SKRIPSI*** adalah sebagai berikut :

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya ( )
2. Telah lulus dan menyerahkan Laporan Praktek Kerja ( )
3. Telah lulus seluruh mata kuliah kearifan (MKDU) sesuai konsentrasinya ( )
4. Telah menempuh mata kuliah ≥ 134 sks dengan IPK ≥ 2 dan tidak ada nilai E ( )
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan ( )
6. Memenuhi persyaratan administrasi ( )

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dari atas perhatiannya kami ucapkan terima kasih

Telah diteliti kebenaran data tersebut diatas
Recording Teknik Elektro

( )

Malang, 6 APRIL 2011
Pemohon

(TITO TRI W. A)

Disetujui
Ketua Jurusan Teknik Elektro

Ir. Yusuf Ismail Nachoda, MT
NIP. Y. 1018800189

Mengetahui
Dosen Wali

( )

Catatan :
Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Jurusan/Sekretaris Jurusan T. Elektro S-1

Source Code Tampilan

```
<title>KFC-Map</title>
        <link rel="stylesheet" href="js/style.css"  />
 <link rel='stylesheet' id='sociable-front-css-css'  href="js/sociable.css-ver=3.0.4.css"  type='text/css' media='all' />

<style type="text/css">
  v\:* {
   behavior:url(#default#VML);
  }
.style2 {color: #ECE9D8}
.style3 {color: #339900}
</style>
<script
src="http://maps.google.com/maps?file=api&amp;v=2&amp;key=ABQIAAAAvygGeDLnH_oMEXYhZUG2jBSRvfnVL9
Z7D89GKRL1uQiEuOhTJxTYb5idNXuzfbvu5EJcVlEbesMtmQ"
type="text/javascript"></script>
<script type="text/javascript" src="js/Route.js"></script>
<script type="text/javascript">

var tsp;
var isDone = false;
var reasons = new Array();
reasons[G_GEO_SUCCESS]        = "Sukses";
reasons[G_GEO_MISSING_ADDRESS]  = "Alamat Salah: Lokasi Alamat Tidak Ditemukan.";
reasons[G_GEO_UNKNOWN_ADDRESS]  = "Alamat Tidak Terdaftar:  Alamat Tidak Ada Pada Peta.";
reasons[G_GEO_UNAVAILABLE_ADDRESS]= "Alamat Tidak Tersedia:  Proses Geocode Alamat Tidak Tersedia";
reasons[G_GEO_BAD_KEY]        = "Kesalahan Kunci API: Kunci Api Tidak valid";
reasons[G_GEO_TOO_MANY_QUERIES]  = "Limit Peta: Situs Sudah Melebihi Limit untuk View Peta";
reasons[G_GEO_SERVER_ERROR]     = "Server error: Proses Geocode Error.";



function formatLength(meters) {
 var km = parseInt(meters / 1000);
 meters -= km * 1000;
 var ret = "";
 if (km > 0)
   ret += km + " km ";
 if (km < 10)
   ret += meters + " m";
 return(ret);
}

function drawMarker(latlng, num) {
 var icon;
```

```
 var letter = num == 1 ? 'r' : 'b';
 icon = new GIcon(G_DEFAULT_ICON, "icons/icon" + letter + num + ".png");
 // icon.printImage = "icons/icon" + letter + num + ".png";
 //icon.mozPrintImage = "icons/icon" + letter + num + ".gif";

// gebMap.addOverlay(new GMarker(latlng, {icon: icon }));
        gebMap.addOverlay(new GMarker(latlng));
}

function setViewportToCover(waypoints) {
 var bounds = new GLatLngBounds();
 for (var i = 0; i < waypoints.length; ++i) {
  bounds.extend(waypoints[i]);
 }
 gebMap.setCenter(bounds.getCenter());
 gebMap.setZoom(gebMap.getBoundsZoomLevel(bounds));
}

function loadAtStart(lat, lng, zoom) {
 if (GBrowserIsCompatible()) {
  gebMap = new GMap2(document.getElementById("map"));
  directionsPanel = document.getElementById("my_textual_div");

  gebMap.setCenter(new GLatLng(lat, lng), zoom);
  gebMap.addControl(new GLargeMapControl());
  gebMap.addControl(new GMapTypeControl());

  tsp = new Djikstra(gebMap, directionsPanel);
  gebDirections = tsp.getGDirections();
  GEvent.addListener(gebDirections, "error", function() {
   alert("Permintaan Gagal: " + reasons[gebDirections.getStatus().code]);
  });

 }
 else {
  alert('Browser Anda Tidak Support Google MAP..');
 }
}


function callMarker(){
  function createMarker(point,html) {
    var marker = new GMarker(point);
    GEvent.addListener(marker, "click", function() {
     marker.openInfoWindowHtml(html);
    });
```

```
    return marker;
  }
  var point = new GLatLng( -7.95696, 112.61736);
  var marker = createMarker(point,'KFC Malang Town Square');
  gebMap.addOverlay(marker);

}


function addWaypoint(latLng) {
 tsp.addWaypoint(latLng);
}

function addAddress(addr) {
 tsp.addAddress(addr, addAddressSuccessCallback);
}

function clickedAddAddress() {
 if (document.address.addressStr2.value!='') {
        tsp.addAddress(document.address.addressStr2.value, addAddressSuccessCallback2);
 document.address.addressStr2.value='';
 }
 tsp.addAddress(document.address.addressStr.value, addAddressSuccessCallback2);
 document.address.addressStr.value='';


}

function addAddressSuccessCallback(address, latlng) {
 if (latlng)
  drawMarker(latlng, tsp.getWaypoints().length);
 else
  alert('Proses Geocode Gagal: ' + address);
}

function addAddressSuccessCallback2(address, latlng) {
 if (latlng) {
  drawMarker(latlng, tsp.getWaypoints().length);
  setViewportToCover(tsp.getWaypoints());
 }
 else
  alert('Proses Geocode Gagal: ' + address);
}

function startOver() {
 document.getElementById("my_textual_div").innerHTML = "";
```

```
 document.getElementById("path").innerHTML = "";

 gebMap.clearOverlays();
 tsp.startOver();
 document.address.addressStr2.value='Jalan Veteran, Malang';
 callMarker();
}

function directions(m, walking, avoid) {
 gebMap.clearOverlays();
 tsp.setAvoidHighways(avoid);
 if (walking)
  tsp.setTravelMode(G_TRAVEL_MODE_WALKING);
 else
  tsp.setTravelMode(G_TRAVEL_MODE_DRIVING);
 gebMap.clearOverlays();
 if (m == 0)
  tsp.solveRoundTrip(onSolveCallback);
 else
  tsp.solveAtoZ(onSolveCallback);
}

function onSolveCallback(myTsp) {
 var dir = tsp.getGDirections();

 var pathStr = "<fieldset style='background-color:#f0f0f0;border: 1px solid #CCCCCC'><legend><strong>Hasil
Pencarian dengan Djikstra</strong></legend><div><p><strong>Waktu (1 Km/Menit):</strong> " +
formatTime(dir.getDuration().seconds) + "<br>";
 pathStr += "<strong>Jarak Tempuh: </strong>" + formatLength(dir.getDistance().meters) + "</p></div>
</fieldset>";
 document.getElementById("path").innerHTML = pathStr;

 var bestPathLatLngStr = "";
 for (var i = 0; i < gebDirections.getNumGeocodes(); ++i) {
  bestPathLatLngStr += "(" + gebDirections.getGeocode(i).Point.coordinates[1]
   + ", " + gebDirections.getGeocode(i).Point.coordinates[0] + ")\n";
 }

 var durationsMatrixStr = "";
 var dur = tsp.getDurations();
 for (var i = 0; i < dur.length; ++i) {
  for (var j = 0; j < dur[i].length; ++j) {
   durationsMatrixStr += dur[i][j];
   if (j == dur[i].length - 1) {
    durationsMatrixStr += "\n";
   } else {
```

```
      durationsMatrixStr += ", ";
     }
    }
   }

   isDone = true;
}

function isDone() {
 return isDone;
}

function init() {
 isDone = false;

 loadAtStart(-7.95714, 112.61799, 15);
 callMarker();
}
</script>

</head>

<body onload='init()' onunload='GUnload()'>

<div id="wrap" style="border:#f1f1f1 1px solid; background-color:#990000">
 <div id="header">
  <div class="headerleft" id="imageheader"> </div>
  </div>
 <div class="clear"></div>
 <div id="subnavbar" style="height:3px">
  <div style="color:#FFFFFF;text-shadow:#CCCCCC;margin-top:0px ;" > </div>
 </div>
 <div class="clear"> </div>
 <div id="content">
  <div id="homepage">
   <center>
   </center>
   <div class="postarea" >
    <div class="breadcrumb">Halaman : Peta </div>
    <div id="map"  style="width: 590px; height: 480px;border: 1px solid #CCCCCC" >
     <span style="color:#676767;font-size:11px;margin:10px;padding:4px;">Loading...</span>
    </div>
    <br>
    <div id="path"></div>
    <div id="exportData" ></div>
    <div id="durationsData"></div>
```

```
      <div id="ordering"></div>
      <div id="my_textual_div"></div>
      <br>
     </div>
    </div>
    <div id="sidebar">
     <ul id="sidebarwidgeted">
      <li id="text-6" class="widget widget_text">
       <h4>Proses Pencarian Rute </h4>
       <div class="textwidget"><span class="style2"></span> <span class="style2"></span>
        <hr size="1" />
        <div>
         <form name="address" onSubmit="clickedAddAddress(); return false;">
          Input Alamat Tujuan:<br />
          <input name="addressStr2" type="text" style="width:65%;display:none" value="Jalan Veteran, Malang"
/>
          <input name="addressStr" type="text" style="width:65%" value="jalan ijen malang" />
          <input type="button" value="Tambahkan" onClick="clickedAddAddress(); ">
         </form>
        </div>
        <hr />
        <form name="travelOpts" style="display:none">
         <table width="331">
          <tr>
           <td><input id="walking" type="checkbox">
            Walking</input><br></td>
           <td><input id="avoidHighways" type="checkbox">
            Avoid highways</input><br></td>
          </tr>
         </table>
        </form>
        <table class="buttonTable">
         <tr>
          <td><input id="button1" type="button" value="Proses" onClick="directions(1,
document.forms['travelOpts'].walking.checked, document.forms['travelOpts'].avoidHighways.checked)"></td>
          <td><input id='button3' class="calcButton" type='button' value='Baru' onClick='startOver()'></td>
         </tr>
        </table>
       </div>
      </li>
                  <li id="text-6" class="widget widget_text">

       <div class="textwidget"><span class="style2"></span> <span class="style2"></span>
        <hr size="1" />
        <div>
        <img src="images/kfc.gif"/> </div>
```

```
                                    <div>
            <img src="images/kfc.gif"/></div>
        </div>
       </li>
      </ul>
      <div id="l_sidebar"> </div>
      <div id="r_sidebar"> </div>
    </div>
  </div>
  <div class="clear" bgcolor="#CCFFCC"></div>
  <div id="footer" style="margin-top:5px">
    <p>Copyright &copy; 2011.tito|abadi &middot; All Rights Reserved</p>
  </div>
</div>
</div>

</body>
</html>
```

Source Code Algoritma Dijkstra

```
(function() {
var tsp;
var gebMap;
var directionsPanel;
var gebDirections;
var gebGeocoder;
var maxTspSize = 24;
var maxTspBF = 0;
var maxTspDjikstra = 15;
var maxSize = 24;
var maxTripSentry = 2000000000;
var avoidHighways = false;
var travelMode = G_TRAVEL_MODE_DRIVING;
var distIndex;
var reasons = new Array();
reasons[G_GEO_SUCCESS]         = "Sukses";
reasons[G_GEO_MISSING_ADDRESS]   = "Alamat Salah: Lokasi Alamat Tidak Ditemukan.";
reasons[G_GEO_UNKNOWN_ADDRESS]   = "Alamat Tidak Terdaftar:  Alamat Tidak Ada Pada Peta.";
reasons[G_GEO_UNAVAILABLE_ADDRESS]= "Alamat Tidak Tersedia:  Proses Geocode Alamat Tidak
Tersedia";
reasons[G_GEO_BAD_KEY]         = "Kesalahan Kunci API: Kunci Api Tidak valid";
reasons[G_GEO_TOO_MANY_QUERIES]  = "Limit Peta: Situs Sudah Melebihi Limit untuk View Peta";
reasons[G_GEO_SERVER_ERROR]     = "Server error: Proses Geocode Error.";
reasons[G_GEO_BAD_REQUEST]      = "Kesalahan Permintaan: Kesalahan Parsing Arah.";
reasons[G_GEO_MISSING_QUERY]    = "Kesalahan Query: Kesalahan Pada Parameter HTML.";
reasons[G_GEO_UNKNOWN_DIRECTIONS] = "Arah Tidak Diketahui: Tidak Bisa Menuju Titik Tujuan Yang
Dituju.";
var waypoints = new Array();
var addresses = new Array();
var addr = new Array();
var wpActive = new Array();
var addressRequests = 0;
var wayStr;
var distances;
var durations;
var dist;
var dur;
var visited;
var currPath;
var bestPath;
var bestTrip;
var nextSet;
var numActive;
var chunkNode;
var cachedDirections = false;

var onSolveCallback = function(){};
var originalOnFatalErrorCallback = function(tsp, errMsg) { alert("Request failed: " + errMsg); }
var onFatalErrorCallback = originalOnFatalErrorCallback;
var doNotContinue = false;
var onLoadListener = null;
```

```
var onFatalErrorListener = null;


function nextSetOf(num) {
 var count = 0;
 var ret = 0;
 for (var i = 0; i < numActive; ++i) {
  count += nextSet[i];
 }
 if (count < num) {
  for (var i = 0; i < num; ++i) {
   nextSet[i] = 1;
  }
  for (var i = num; i < numActive; ++i) {
   nextSet[i] = 0;
  }
 } else {

  var firstOne = -1;
  for (var i = 0; i < numActive; ++i) {
   if (nextSet[i]) {
          firstOne = i;
          break;
   }
  }

  var firstZero = -1;
  for (var i = firstOne + 1; i < numActive; ++i) {
   if (InextSet[i]) {
          firstZero = i;
          break;
   }
  }
  if (firstZero < 0) {
   return -1;
  }

  nextSet[firstZero] = 1;

  for (var i = 0; i < firstZero - firstOne - 1; ++i) {
   nextSet[i] = 1;
  }
  for (var i = firstZero - firstOne - 1; i < firstZero; ++i) {
   nextSet[i] = 0;
  }
 }

 for (var i = 0; i < numActive; ++i) {
  ret += (nextSet[i]<<i);
 }
 return ret;
}
```

```
/*
 *
source : http://id.wikipedia.org/wiki/Algoritma_Dijkstra#Pseudocode
function Dijkstra(G, w, s)
  for each vertex v in V[G]                  // Initializations
      d[v] := infinity
      previous[v] := undefined
  d[s] := 0                              // Distance from s to s
  S := empty set
  Q := V[G]                              // Set of all vertices
  while Q is not an empty set                  // The algorithm itself
      u := Extract_Min(Q)
      S := S union {u}
      for each edge (u,v) outgoing from u
          if d[u] + w(u,v) < d[v]          // Relax (u,v)
              d[v] := d[u] + w(u,v)
              previous[v] := u


*/



function tspDjikstra(mode) {
 var numCombos = 1<<numActive;
 var C = new Array();
 var parent = new Array();
 for (var i = 0; i < numCombos; ++i) {
  C[i] = new Array();
  parent[i] = new Array();
  for (var j = 0; j < numActive; ++j) {
   C[i][j] = 0.0;
   parent[i][j] = 0;
  }
 }
 for (var k = 1; k < numActive; ++k) {
  var index = 1 + (1<<k);
  C[index][k] = dur[0][k];
 }

 for (var s = 3; s <= numActive; ++s) {
  for (var i = 0; i < numActive; ++i) {
   nextSet[i] = 0;
  }
  var index = nextSetOf(s);
  while (index >= 0) {

   for (var k = 1; k < numActive; ++k) {
                    if (nextSet[k]) {
                     var prevIndex = index - (1<<k);
                     C[index][k] = maxTripSentry;

                     for (var m = 1; m < numActive; ++m) {
```

```
                        if (nextSet[m] && m != k) {

                         if (C[prevIndex][m] + dur[m][k] < C[index][k]) {
                                C[index][k] = C[prevIndex][m] + dur[m][k];
                                parent[index][k] = m;
                         }

                        }
                    }

          }
    }
    index = nextSetOf(s);
  }
 }

 for (var i = 0; i < numActive; ++i) {
  bestPath[i] = 0;
 }

 var index = (1<<numActive)-1;
 if (mode == 0) {
  var currNode = -1;
  bestPath[numActive] = 0;
  for (var i = 1; i < numActive; ++i) {
   if (C[index][i] + dur[i][0] < bestTrip) {
         bestTrip = C[index][i] + dur[i][0];
         currNode = i;
   }
  }
  bestPath[numActive-1] = currNode;
 } else {
  var currNode = numActive - 1;
  bestPath[numActive-1] = numActive - 1;
  bestTrip = C[index][numActive-1];
 }
 for (var i = numActive - 1; i > 0; --i) {
  currNode = parent[index][currNode];
  index -= (1<<bestPath[i]);
  bestPath[i-1] = currNode;
 }

}

function makeLatLng(latLng) {
 return(latLng.toString().substr(1,latLng.toString().length-2));
}

function getWayStr(curr) {

 var nextAbove = -1;
 for (var i = curr + 1; i < waypoints.length; ++i) {
  if (wpActive[i]) {
```

```
    if (nextAbove == -1) {
      nextAbove = i;
    } else {
      wayStr.push(makeLatLng(waypoints[i]));
      wayStr.push(makeLatLng(waypoints[curr]));
    }
   }
  }
  if (nextAbove != -1) {
   wayStr.push(makeLatLng(waypoints[nextAbove]));
   getWayStr(nextAbove);
   wayStr.push(makeLatLng(waypoints[curr]));
  }
}

function getDistTable(curr, currInd) {
  var nextAbove = -1;
  var index = currInd;
  for (var i = curr + 1; i < waypoints.length; ++i) {
   if (wpActive[i]) {
    index++;
    if (nextAbove == -1) {
      nextAbove = i;
    } else {
      dist[currInd][index] = distances[distIndex];
      dur[currInd][index] = durations[distIndex++];
      dist[index][currInd] = distances[distIndex];
      dur[index][currInd] = durations[distIndex++];
    }
   }
  }
  if (nextAbove != -1) {
   dist[currInd][currInd+1] = distances[distIndex];
   dur[currInd][currInd+1] = durations[distIndex++];
   getDistTable(nextAbove, currInd+1);
   dist[currInd+1][currInd] = distances[distIndex];
   dur[currInd+1][currInd] = durations[distIndex++];
  }
}

function directions(mode) {
  if (cachedDirections) {

   doTsp(mode);
  }
  wayStr = new Array();
  numActive = 0;
  for (var i = 0; i < waypoints.length; ++i) {
   if (wpActive[i]) ++numActive;
  }

  for (var i = 0; i < waypoints.length; ++i) {
   if (wpActive[i]) {
```

```
    wayStr.push(makeLatLng(waypoints[i]));
    getWayStr(i);
    break;
   }
  }

  if (numActive > maxTspSize) {
   alert("Too many locations! You have " + numActive + ", but max limit is " + maxTspSize);
  } else {
   distances = new Array();
   durations = new Array();
   chunkNode = 0;
   nextChunk(mode);
  }
}

function nextChunk(mode) {
 if (chunkNode < wayStr.length) {
  var wayStrChunk = new Array();
  for (var i = 0; i < maxSize && i + chunkNode < wayStr.length; ++i) {
   wayStrChunk.push(wayStr[chunkNode+i]);
  }
  chunkNode += maxSize;
  if (chunkNode < wayStr.length-1) {
   chunkNode--;
  }

  var myGebDirections = new GDirections(gebMap, directionsPanel);

  GEvent.addListener(myGebDirections, "error", function() {
   alert(myGebDirections.getStatus().code);
   var errorMsg = reasons[myGebDirections.getStatus().code];
   if (typeof errorMsg == 'undefined') errorMsg = 'Unknown error.';
   var doNotContinue = true;
   onFatalErrorCallback(tsp, errorMsg);
  });

  GEvent.addListener(myGebDirections, "load", function() {

        for (var i = 0; i < myGebDirections.getNumRoutes(); ++i) {
         durations.push(myGebDirections.getRoute(i).getDuration().seconds);
         distances.push(myGebDirections.getRoute(i).getDistance().meters);
        }
        nextChunk(mode);
  });

  GEvent.addListener(myGebDirections, "addoverlay", function() {

        gebMap.clearOverlays();
        directionsPanel.innerHTML = "";
   });
```

```
    myGebDirections.loadFromWaypoints(wayStrChunk, { getSteps:false, getPolyline:false,
  preserveViewport:true, avoidHighways:avoidHighways, travelMode:travelMode });
   } else {
    readyTsp(mode);
   }
  }

  function readyTsp(mode) {

   distIndex = 0;
   dist = new Array();
   dur = new Array();
   numActive = 0;
   for (var i = 0; i < waypoints.length; ++i) {
    if (wpActive[i]) {
     dist.push(new Array());
     dur.push(new Array());
     addr[numActive] = addresses[i];
     numActive++;
    }
   }
   for (var i = 0; i < numActive; ++i) {
    dist[i][i] = 0;
    dur[i][i] = 0;
   }
   for (var i = 0; i < waypoints.length; ++i) {
    if (wpActive[i]) {
     getDistTable(i, 0);
     break;
    }
   }

   doTsp(mode);
  }

  function doTsp(mode) {

   visited = new Array();
   for (var i = 0; i < numActive; ++i) {
    visited[i] = false;
   }
   currPath = new Array();
   bestPath = new Array();
   nextSet = new Array();
   bestTrip = maxTripSentry;
   visited[0] = true;
   currPath[0] = 0;
   cachedDirections = true;
   if (numActive <= maxTspDjikstra + mode) {
    tspDjikstra(mode);
   }

   wayStrChunk = new Array();
```

```
  var wpIndices = new Array();
  for (var i = 0; i < waypoints.length; ++i) {
   if (wpActive[i]) {
    wpIndices.push(i);
   }
  }
  var bestPathLatLngStr = "";
  for (var i = 0; i < bestPath.length; ++i) {

   wayStrChunk.push(makeLatLng(waypoints[wpIndices[bestPath[i]]]));
   bestPathLatLngStr += makeLatLng(waypoints[wpIndices[bestPath[i]]]) + "\n";
  }
  if (typeof onSolveCallback == 'function') {
   if (onLoadListener)
    GEvent.removeListener(onLoadListener);

   onLoadListener = GEvent.addListener(gebDirections, 'addoverlay', function(){
    onSolveCallback(tsp);
   });
  }

  if (onFatalErrorListener)
   GEvent.removeListener(onFatalErrorListener);
  onFatalErrorListener = GEvent.addListener(gebDirections, 'error', onFatalErrorCallback)

  gebDirections.loadFromWaypoints(wayStrChunk, {getSteps:true, getPolyline:true, preserveViewport:false,
avoidHighways:avoidHighways, travelMode:travelMode});
}

function addWaypoint(latLng) {
  var freeInd = -1;
  for (var i = 0; i < waypoints.length; ++i) {
   if (!wpActive[i]) {
    freeInd = i;
    break;
   }
  }
  if (freeInd == -1) {
   if (waypoints.length < maxTspSize) {
    waypoints.push(latLng);
    wpActive.push(true);
    freeInd = waypoints.length-1;
   } else {
    return(-1);
   }
  } else {
   waypoints[freeInd] = latLng;
   wpActive[freeInd] = true;
  }
  return(freeInd);
}

function addAddress(address, callback) {
```

```
  ++addressRequests;
  gebGeocoder.getLatLng(address, function(latLng) {
   addressRequests--;
   if (!latLng) {
    if (typeof(callback) == 'function')
     callback(address);
   } else {
          var freeInd = addWaypoint(latLng);
          addresses[freeInd] = address;
    if (typeof callback == 'function')
     callback(address, latLng);
   }
  });
}

Djikstra.prototype.startOver = function() {
  waypoints = new Array();
  addresses = new Array();
  addr = new Array();
  wpActive = new Array();
  wayStr = "";
  distances = new Array();
  durations = new Array();
  dist = new Array();
  dur = new Array();
  visited = new Array();
  currPath = new Array();
  bestPath = new Array();
  bestTrip = new Array();
  nextSet = new Array();
  numActive = 0;
  chunkNode = 0;
  addressRequests = 0;
  cachedDirections = false;
  onSolveCallback = function(){};
  doNotContinue = false;
}


function Djikstra(map, panel, onFatalError) {
  if (tsp) {
   alert('You can only create one Djikstra at a time.');
   return;
  }

  gebMap            = map;
  directionsPanel   = panel;
  gebGeocoder       = new GClientGeocoder();
  gebDirections     = new GDirections(gebMap, directionsPanel);
  onFatalErrorCallback = onFatalError;
  tsp               = this;
}
```

```
Djikstra.prototype.addAddress = function(address, callback) {
 if (!this.isReady()) {
  setTimeout(function(){ tsp.addAddress(address, callback) }, 20);
  return;
 }
 addAddress(address, callback);
};

Djikstra.prototype.addWaypoint = function(latLng) {
 addWaypoint(latLng);
};

Djikstra.prototype.getWaypoints = function() {
 var wp = [];
 for (var i = 0; i < waypoints.length; i++) {
  if (wpActive[i])
   wp.push(waypoints[i]);
 }
 return wp;
};

Djikstra.prototype.getAddresses = function() {
 var addrs = [];
 for (var i = 0; i < addresses.length; i++) {
  if (wpActive[i])
   addrs.push(addresses[i]);
 }
 return addrs;
};

Djikstra.prototype.removeWaypoint = function(latLng) {
 for (var i = 0; i < waypoints.length; ++i) {
  if (wpActive[i] && waypoints[i].equals(latLng)) {
   wpActive[i] = false;
   return true;
  }
 }
 return false;
};

Djikstra.prototype.removeAddress = function(addr) {
 for (var i = 0; i < addresses.length; ++i) {
  if (wpActive[i] && addresses[i] == addr) {
   wpActive[i] = false;
   return true;
  }
 }
 return false;
};

Djikstra.prototype.getGDirections = function() {
 return gebDirections;
};
```

```
Djikstra.prototype.getOrder = function() {
 return bestPath;
}

Djikstra.prototype.getAvoidHighways = function() {
 return avoidHighways;
}

Djikstra.prototype.setAvoidHighways = function(avoid) {
 avoidHighways = avoid;
}

Djikstra.prototype.getTravelMode = function() {
 return travelMode;
}

Djikstra.prototype.setTravelMode = function(travelM) {
 travelMode = travelM;
}

Djikstra.prototype.getDurations = function() {
 return dur;
}


Djikstra.prototype.getTotalDuration = function() {
 return gebDirections.getDuration().seconds;
}


Djikstra.prototype.isReady = function() {
 return addressRequests == 0;
};

Djikstra.prototype.solveRoundTrip = function(callback) {
 if (doNotContinue) {
  alert('Cannot continue after fatal errors.');
  return;
 }

 if (!this.isReady()) {
  setTimeout(function(){ tsp.solveRoundTrip(callback) }, 20);
  return;
 }
 if (typeof callback == 'function')
  onSolveCallback = callback;

 directions(0);
};

Djikstra.prototype.solveAtoZ = function(callback) {
```

```
  if (doNotContinue) {
    alert('Cannot continue after fatal errors.');
    return;
  }

  if (!this.isReady()) {
    setTimeout(function(){ tsp.solveAtoZ(callback) }, 20);
    return;
  }

  if (typeof callback == 'function')
    onSolveCallback = callback;

  directions(1);
};

window.Djikstra = Djikstra;

})();
```