

**PENENTUAN JARAK TERPENDEK DAN WAKTU TEMPUH
MENGUNAKAN ALGORITMA DIJKSTRA
DENGAN PEMROGRAMAN BERBASIS OBJEK**

Skripsi
untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-1

Program Studi Teknik Informatika



disusun oleh
Farida Ardiani
06650018

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA
2011



Universitas Islam Negeri Sunan Kalijaga

FM-UINSK-BM-05-07/R0

PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : UIN.02/D.ST/PP.01.1/1071/2011

Skripsi/Tugas Akhir dengan judul : Penentuan Jarak Terpendek dan Waktu Tempuh
Menggunakan Algoritma Dijkstra dengan Pemrograman Berbasis Objek

Yang dipersiapkan dan disusun oleh :
Nama : Farida Ardiani
NIM : 06650018
Telah dimunaqasyahkan pada : 7 Juni 2011
Nilai Munaqasyah : A -

Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

Ketua Sidang

Landung Sudarmo, M.Kom
NIY. 0527027001

Penguji I

Maria Ulfah Siregar, M.IT
NIP.19780106 200212 2 001

Penguji II

Ade Ratnasari, M.T
NIP. 19801217 200604 2 002

Yogyakarta, 16 Juni 2011
UIN Sunan Kalijaga
Fakultas Sains dan Teknologi
Dekan



H. Akh. Minhaji, M.A, Ph.D
NIP.19580919 198603 1 002



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Surat Persetujuan Skripsi/Tugas Akhir
Lamp : -

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta
di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Farida Ardiani
NIM : 06650018
Judul Skripsi : PENENTUAN JARAK TERPENDEK DAN WAKTU TEMPUH MENGGUNAKAN
ALGORITMA DIJKSTRA BERBASIS OBJEK

sudah dapat diajukan kembali kepada Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Teknik Informatika.

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Yogyakarta, 15 mei 2011

Pembimbing

Landung Sudarmana, M.Kom

NIP. 0527027001



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Surat Persetujuan Skripsi/Tugas Akhir

Lamp : -

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Farida Ardiani
NIM : 06650018
Judul Skripsi : PENENTUAN JARAK TERPENDEK DAN WAKTU TEMPUH MENGGUNAKAN ALGORITMA DIJKSTRA BERBASIS OBJEK

sudah dapat diajukan kembali kepada Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Teknik Informatika.

Dengan ini kami berharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Yogyakarta, 15 mei 2011

Pembimbing

Nurochman, M.Kom

NIP. 19801223 200901 1 007

PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini:

Nama : Farida Ardiani
NIM : 06650018
Program Studi : Teknik Informatika
Fakultas : Sains Dan Teknologi

Menyatakan bahwa skripsi dengan judul **“PENENTUAN JARAK TERPENDEK DAN WAKTU TEMPUH MENGGUNAKAN ALGORITMA DIJKSTRA BERBASIS OBJEK”** tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 15 Mei 2011

Mahasiswa



Farida Ardiani
NIM. 06650018

MOTTO

**SETIAP ADA USAHA
MEMBUAT ADA HARAPAN
KERJA KERAS PASTI DIHARGAI..
SELEBIHNYA PERCAYA PADA DIRI SENDIRI**

J

HALAMAN PERSEMBAHAN

*Dengan rasa hormat dan baktiku,
Ananda persembahkan karya ini kepada Ibunda tercinta,
Ibunda Chomisyah, yang telah melimpahkan kasih sayang
untukku, selalu memberiku kekuatan, dan aliran doa yang
tiada henti mengalir untukku sepanjang waktu. Karena kasih
sayang dan doa beliau saya mendapat kesempatan mengenyam
pendidikan yang lebih tinggi dan dapat menyelesaikan
pendidikan jenjang (S1). I love you Mom..*

*Kedua untuk kakak dan adikku tersayang, Ardian Syuhada'
dan Faizah Rahmawati, terima kasih karena kalian selalu
memberiku dukungan dan semangat untuk menyelesaikan
karya ini.*

KATA PENGANTAR

Alhamdulillah robbil ‘alamin, puji syukur selalu kami panjatkan kepada maha segala maha dalam kehidupan ini, Allah SWT. Karena dengan kebesaran serta keagunganNya telah melimpahkan keislaman, rizki, kesehatan dan ilmu yang melimpah ruah. Kedua kalinya shalawat serta salam kami haturkan kepada baginda besar Nabi Muhammad SAW yang dengan kasih sayang dan kesabaran beliau selalu membimbing dan menunjukkan jalan kebenaran untuk ummatnya.

Penulisan skripsi ini tidak akan terselesaikan tanpa izin Allah dan dukungan serta bantuan berbagai pihak, oleh karena itu dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Ibundaku tercinta, ibunda Chomisyah.
2. Bapak Prof. Drs. H. Akh Minhaji, M.A, Ph.D, selaku Dekan Fakultas Sains & Teknologi UIN Sunan Kalijaga.
3. Bapak Agus Mulyanto, M.Kom selaku Ketua Program Studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta.
4. Bapak Landung Sudarmana, M.Kom selaku dosen pembimbing I yang telah memberikan bimbingan, dukungan dan arahan dalam menyelesaikan skripsi ini.
5. Bapak Nurochman, M.Kom selaku dosen pembimbing II yang memberikan masukan, arahan dan bimbingan selama proses pelaksanaan dan penyelesaian skripsi.

6. Segenap Dosen Program Studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta.
7. Bapak Umar Affandhi selaku tentor privat Delphi. Terima kasih telah membantu saya menyelesaikan program dalam skripsi ini.
8. Kakak dan adikku, Ardian Syuhada' dan Faizah Rahmawati, terima kasih atas dukungan dan do'a kalian.
9. Untuk Key Appa, Onew Appa, Taemin Appa, Minho Appa dan Jonghyun Appa, kamsahamnida kalian selalu memberi saya semangat selama pembuatan skripsi ini. Untuk Eva (mts) terima kasih sudah memberikan pinjaman alat transportasi selama pembuatan skripsi ini. Dan untuk semua teman-teman kost Bougenville yang tidak bisa saya sebutkan satu persatu, terima kasih untuk semuanya.

Semoga berkat bantuan, dukungan serta semangat seluruh pihak yang telah membantu penulis untuk menyelesaikan skripsi ini diberikan pahala yang melimpah. Amin Ya Robbal 'Alamin.

Akhir kata, semoga penelitian ini dapat bermanfaat bagi penulis dan pembaca dengan sebaik-baiknya.

Yogyakarta, 23 April 2011

Penulis

Farida Ardiani
NIM.06650018

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN	ii
SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR	ii
PERNYATAAN KEASLIAN SKRIPSI.....	iv
MOTTO.....	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xvi
ABSTRAK	xvii
ABSTRACT	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Keaslian Penelitian	5

BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI	6
2.1 Tinjauan Pustaka	6
2.2 Landasan Teori	9
2.2.1 Rekayasa Perangkat Lunak.....	9
2.2.2 Jarak Terpendek.....	11
2.2.3 WaktuTempuh	11
2.2.4 Algoritma Dijkstra	12
2.2.5 Pemrograman Berorientasi Objek.....	19
2.2.6 Delphi	20
2.2.7 Konsep PBO dalam Delphi.....	23
2.2.8 MySQL.....	26
2.2.9 Pemodelan Data	29
BAB III METODE PENELITIAN	36
3.1 Subjek Penelitian.....	36
3.2 Alat Penelitian.....	36
3.3 Metodologi Penelitian.....	37
3.4 Perancangan Sistem.....	38
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	40
4.1 Perancangan Sistem.....	40
4.1.1 Perancangan Proses Data.....	40
4.1.1.1 Desain DFD	40
4.1.1.1.1 Diagram Konteks.....	40
4.1.1.1.2 Diagram Level 1(Proses Sistem Penentuan jarak	

Terpendek dan Waktu Tempuh).....	41
4.1.1.1.3 Diagram Level 2 (Pendataan Jarak Terpendek) .	42
4.1.1.1.4 Diagram Level 2 (Pendataan Waktu Tempuh)...	43
4.1.1.2 Desain ERD	44
4.1.2 Perancangan Database.....	44
4.1.3 Perancangan Tabel	45
4.1.4 Perancangan <i>User Interface</i>	47
4.1.5 Perancangan Kendali Sistem	49
4.2 Implementasi	50
4.2.1 Implementasi Basis Data	50
4.2.2 Implementasi Membuat <i>Database</i> Baru.....	51
4.2.3 Implementasi Program	52
4.2.3.1 Halaman Utama.....	52
4.2.3.2 Tampilan Help.....	54
4.2.3.3 Tampilan Pembuatan Titik.....	55
4.2.3.4 Tampilan Pembuatan Garis.....	56
4.2.3.5 Tampilan Masukan Titik Awal Pencarian	61
4.2.3.6 Tampilan Masukan Titik Akhir Pencarian.....	62
4.2.3.7 Tampilan Pencarian Jarak Terpendek	64
4.2.3.8 Tampilan Blokir Jalan	65
4.2.3.9 Tampilan Pencarian Jarak Terpendek Alternatif.....	66
4.2.3.10 Tampilan Masukan Kecepatan.....	67
4.2.3.11 Tampilan Pencarian Waktu Tempuh	69

4.2.3.12 Tampilan Penyimpanan Graf	69
4.2.3.13 Tampilan Pemanggilan Graf	70
4.2.4 Implementasi Kendali Sistem	72
4.2.4.1 Tampilan Pesan Kesalahan Masukan Kosong	72
4.2.4.2 Tampilan Pesan Kesalahan Masukan Nomor Garis	72
4.2.4.3 Tampilan Pesan Kesalahan Masukan Huruf	73
4.2.4.4 Tampilan Pesan Kesalahan Batas Kecepatan	74
4.3 Pengujian Sistem	74
BAB V KESIMPULAN DAN SARAN	81
5.1 Kesimpulan	81
5.2 Saran	82
DAFTAR PUSTAKA	83
LAMPIRAN	85

DAFTAR GAMBAR

Gambar 2.1 Contoh graf tak berarah.....	16
Gambar 2.2 Simbol entity luar pada teknik Yourdon and De Marco	30
Gambar 2.3 Simbol aliran data pada teknik Yourdon and De Marco.....	31
Gambar 2.4 Simbol proses pada teknik Yourdon and De Marco	31
Gambar 2.5 Simbol berkas pada teknik Yourdon and De Marco.....	31
Gambar 2.6 Simbol himpunan entitas.....	32
Gambar 2.7 Simbol himpunan relasi	32
Gambar 2.8 Simbol penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dan atributnya	32
Gambar 2.9 Cardinality <i>one to one relationship</i>	34
Gambar 2.10 Cardinality <i>one to many relationship</i>	35
Gambar 2.11 Cardinality <i>many to many relationship</i>	35
Gambar 4.1 DFD level 0 (Diagram Konteks)	41
Gambar 4.2 DFD level 1	42
Gambar 4.3 DFD Level 2 (Proses Pendataan Jarak Terpendek)	43
Gambar 4.4 DFD Level 2 (Proses Pendataan Waktu Tempuh).....	43
Gambar 4.5 ERD.....	44
Gambar 4.6 Relasi antar tabel.....	45
Gambar 4.7 Design <i>User Interface</i>	47
Gambar 4.8 Pesan kesalahan masukan kosong	49
Gambar 4.9 Pesan kesalahan masukan nomor garis	49

Gambar 4.10 Pesan kesalahan masukan huruf	50
Gambar 4.11 Pesan kesalahan batas kecepatan	50
Gambar 4.12 MySQL dalam keadaan aktif.....	51
Gambar 4.13 Membuat <i>database</i> baru.....	52
Gambar 4.14 Tampilan halaman utama	54
Gambar 4.15 Tampilan help.....	54
Gambar 4.16 Tampilan memberi masukan nama titik.....	55
Gambar 4.17 Tampilan hasil pembuatan 1 titik	55
Gambar 4.18 Tampilan pilihan satuan	56
Gambar 4.19 Tampilan memberi masukan titik pangkal.....	57
Gambar 4.20 Tampilan memberi masukan titik ujung	57
Gambar 4.21 Tampilan hasil jarak otomatis	58
Gambar 4.22 Jarak ditampilkan dalam listbox	58
Gambar 4.22 Tampilan memasukan titik awal pencarian.....	54
Gambar 4.23 Tampilan memberi masukan titik pangkal.....	59
Gambar 4.24 Tampilan memberi masukan titik ujung	59
Gambar 4.25 Tampilan memberi masukan jarak.....	60
Gambar 4.26 Jarak ditampilakn dalam listbox	60
Gambar 4.27 Tampilan memasukan titik awal pencarian.....	61
Gambar 4.28 Tampilan titik awal pencarian muncul di <i>listbox</i>	62
Gambar 4.29 Tampilan masukan titik akhir pencarian	63
Gambar 4.30 Tampilan titik akhir pencarian muncul di <i>listbox</i>	63
Gambar 4.31 Tampilan pencarian jarak terpendek.....	64

Gambar 4.32 Tampilan masukan nomor blokir jalan	65
Gambar 4.33 Tampilan salah satu jalur sudah terblokir	66
Gambar 4.34 Tampilan pencarian jarak terpendek alternatif	67
Gambar 4.35 Tampilan masukan kecepatan.....	68
Gambar 4.36 Tampilan kecepatan muncul di <i>listbox</i>	68
Gambar 4.37 Tampilan pencarian waktu tempuh.....	69
Gambar 4.38 Tampilan simpan graf	70
Gambar 4.39 Tampilan sukses menyimpan graf	70
Gambar 4.40 Tampilan pemanggilan graf.....	71
Gambar 4.41 Tampilan graf yang telah dipanggil	71
Gambar 4.42 Pesan kesalahan masukan kosong	72
Gambar 4.43 Pesan kesalahan masukan nomor garis	73
Gambar 4.44 Pesan kesalahan masukan huruf	73
Gambar 4.45 Pesan kesalahan batas kecepatan	74
Gambar 4.46 Contoh hasil jarak terpendek pengujian responden titik 1 ke 6....	76
Gambar 4.47 Contoh hasil jarak terpendek alternatif pengujian responden titik 1 ke 6.....	78
Gambar 4.48 Contoh hasil waktu tempuh pengujian responden titik 1 ke 6	79

DAFTAR TABEL

Tabel 2.1 Tabel tinjauan pustaka	8
Tabel 2.2 Tabel penjelasan graf menggunakan dijkstra	18
Tabel 4.1 Tabel Sec.....	46
Tabel 4.2 Tabel Titik.....	46
Tabel 4.3 Tabel Garis.....	46
Tabel 4.4 Pembuatan 9 titik.....	75
Tabel 4.5 Pembuatan 20 garis.....	75
Tabel 4.6 Pengujian <i>Interface</i>	79
Tabel 4.7 Pengujian fungsional	79

ABSTRAK

Setiap orang dalam melakukan perjalanan pasti memilih jarak terpendek untuk mencapai tujuannya, karena dapat menghemat waktu, tenaga serta bahan bakar tentunya. Kesulitan menentukan jarak terpendek timbul karena terdapat banyak jalur alternatif yang ada dari suatu daerah ke daerah lain dan juga memungkinkan memilih jalur alternatif apabila terdapat suatu hambatan pada jalur terpendek utama.

Penelitian ini menggunakan algoritma dijkstra untuk menemukan jarak terpendek dengan menghitung jarak terpendek dari titik awal ke masing-masing titik yang ditemui. Dijkstra merupakan algoritma dengan prinsip greedy yang memecahkan masalah lintasan terpendek untuk sebuah graf dengan bobot sisi yang tidak negatif. Greedy adalah strategi yang memecahkan masalah langkah demi langkah pada setiap langkah dan hanya memikirkan solusi terbaik yang akan diambil pada setiap langkah tanpa memikirkan konsekuensi ke depan. Implementasi pada sistem ini menggunakan bahasa pemrograman *Delphi7*, *database MySQL* dengan koneksi ODBC.

Sistem penentuan jarak terpendek dan waktu tempuh ini memberikan kemudahan bagi *user* untuk menemukan jarak terpendek dan jarak terpendek alternatif apabila terjadi hambatan serta mengetahui waktu tempuh. Sistem ini menampilkan sebuah hasil jarak terpendek dan jarak terpendek alternatif baik melalui pemanggilan graf yang telah tersimpan dalam database atau membuat graf sendiri, kemudian memasukkan titik awal pencarian dan titik akhir pencarian maka sistem ini akan melakukan perhitungan dengan algoritma dijkstra sesuai dengan titik awal pencarian dan titik akhir pencarian yang telah dimasukkan sebelumnya. Untuk mengetahui waktu tempuh dari jarak terpendek yang telah didapat hanya dengan memasukan kecepatan maka sistem akan melakukan perhitungan dari jarak terpendek yang telah didapat dibagi dengan kecepatan yang telah di masukan.

Kata Kunci : Algoritma dijkstra, jarak terpendek, jarak terpendek alternatif dan waktu tempuh

ABSTRACT

Everyone in journey surely chosen short distance to reach the target, because can economize the time, energy and also fuel perhaps. Difficulty determine the short distance come up because there are a lot of alternative strip from an area to other, and also enabling to chosen the alternative strip if there are resistance at especial short strip.

This research use the dijkstra algorithm to find the short distance by calculating short distance from starting point to every points which met. Dijkstra represent the algorithm with greedy princip is solving problem short trajectory to a graph with the side wight which is not negative. Greedy is strategy to solving problem step by step in every step and only think the best solution will be taken in every step without thinking consequence forwards. Implementation of this system use Delphi7 language, MySQL database by ODBC connection.

This short distance and time go through determination system give the amenity for user to find the short distance and alternative short distance if there are resistance and also know the time went through. This system present a short distance result and alternative short distance passing by denominating graph which have on file in database or make the graph by self, then input the starting point and final point seeking so this system will doing the calculation with the dijkstra algorithm as according with starting point of final point seeking which have been entered previously. To know the time go through from short distance which have been got only with input the speed then system will doing the calculation from short distance which have been got to be divided with the speed which have been entered previously.

Keyword : Dijkstra algorithm, short distance, alternative short distance and time went through.

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Dalam kehidupan sehari-hari sering kali seseorang melakukan perjalanan dari suatu daerah ke daerah lain. Dan sangat lazim sekali apabila memilih jarak terdekat dari kedua daerah tersebut untuk dilalui, karena dapat menghemat waktu, tenaga serta bahan bakar tentunya. Untuk suatu industri tertentu jarak terdekat dan waktu tempuh sangat diperhitungkan sekali untuk menghindari kerugian, seperti contoh pengiriman industri bahan baku susu segar. Penentuan jarak terpendek dan mengetahui waktu tempuh sangat dibutuhkan sekali agar susu yang dibawa untuk mencapai daerah tujuan tetap segar dan tidak basi.

Kesulitan menentukan jarak terpendek timbul karena terdapat banyak jalur yang ada pada tiap daerah karena dalam kenyataannya dari daerah A ke daerah Z tidak hanya memiliki satu jalur saja, banyak sekali jalur yang dapat dilalui sehingga terbentuk suatu jaringan. Suatu jaringan memiliki banyak jalur alternatif dari kedudukan semula ke kedudukan yang dikehendaki.

Untuk membantu dalam menentukan jarak terpendek dapat digunakan peta konvensional dan memilih mana jalur yang dianggap terpendek dari daerah asal ke daerah tujuan. Namun hal ini dirasa kurang maksimal dan memperlambat waktu karena harus memilih sendiri dari banyak jalur yang ada dan melakukan

perhitungan sendiri mana kira-kira jarak terpendek dari daerah asal menuju daerah tujuan yang dihendaki.

Untuk itu diperlukan suatu sistem yang dapat membantu menemukan jarak terpendek dari daerah asal ke daerah tujuan yang disajikan secara sederhana serta terkomputerisasi sehingga memudahkan dalam menentukan jarak terpendek dan menemukan waktu tempuh dari jarak terpendek yang didapat.

Permasalahan yang lain timbul apabila terjadi pemutusan jalur terpendek semula. Hal ini bisa saja terjadi pada kehidupan sehari-hari. Adanya perbaikan jalan, keramaian kampanye partai politik dapat menyebabkan satu ruas jalan tidak dapat dilalui dan berakibat putusnya jalur yang melewati ruas jalan tersebut.

Untuk itu selain diperlukan suatu cara untuk menentukan jalur terpendek dari suatu jaringan, maka diperlukan juga mencari jalur alternatif yang lain apabila terjadi pemutusan jalur.

Ada banyak algoritma yang dapat digunakan untuk menemukan jarak terpendek pada sebuah graf diantaranya Algoritma Dijkstra, Algoritma Ford dan Algoritma Floyd. Dalam penelitian ini peneliti menggunakan Algoritma Dijkstra untuk pencarian jarak terpendek.

Selain lebih menguntungkan dari sisi running time, kelebihan algoritma dijkstra dapat menyelesaikan beberapa kasus pencarian lintasan terpendek, yaitu:

1. pencarian lintasan terpendek antara dua buah simpul tertentu (*a pair shortest path*)
2. pencarian lintasan terpendek antara semua pasangan simpul (*all pairs shortest path*)

3. pencarian lintasan terpendek dari simpul tertentu ke semua simpul yang lain (*single-source shortest path*)
4. pencarian lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu (*intermediate shortest path*).

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Apakah dapat dibuat perangkat lunak menggunakan konsep pemrograman berorientasi objek untuk menentukan jarak terpendek dan jarak terpendek alternatif serta waktu tempuh dengan algoritma dijkstra?
2. Apakah dapat dicari jarak terpendek alternatif jika terdapat hambatan pada jarak terpendek semula?

1.3 Batasan Masalah

Penelitian ini akan membatasi cakupan permasalahan yang terkait dengan penentuan jarak terpendek dan waktu tempuh menggunakan algoritma dijkstra berbasis objek. Masalah dalam penelitian ini dibatasi dalam hal:

1. Penentuan jarak terpendek, dimana jarak terpendek berarti jalur terpendek.
2. Waktu tempuh adalah waktu tempuh terpendek dari jarak terpendek yang telah didapat.

3. Dalam penelitian ini hambatan yang dimaksud adalah hambatan yang menyebabkan jalan benar-benar tidak dapat dilalui sehingga harus dialihkan pada jalur lain, misalnya demo besar-besaran, perbaikan jalan, pasar tumpah dll.
3. Pemrograman yang digunakan adalah pemrograman berorientasi objek.
4. Penentuan jalur terpendek alternatif, bila terjadi hambatan yang menyebabkan pemutusan jalur terpendek semula.
5. Graf yang dipakai adalah graf tidak berarah, artinya tidak memperhatikan titik pangkal dan titik ujung suatu garis.
6. Jarak langsung dari 2 simpul adalah unik (hanya ada satu nilai).

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk membuat sistem informasi penentuan jarak terpendek yang dilengkapi dengan pencarian jalur alternatif jika terjadi hambatan serta waktu tempuh untuk sampai ke kota tujuan.

1.5 Manfaat Penelitian

Hasil penelitian ini diharapkan dapat memberi manfaat sebagai berikut:

- a. Dapat digunakan sebagai langkah awal dalam pemanfaatan teknologi informasi, sebagai media informasi penentuan jarak terpendek dan waktu tempuh.
- b. Memberi kemudahan pengguna untuk mengetahui jarak terpendek yang harus dilalui untuk mencapai suatu daerah.

- c. Memberi kemudahan bagi pengguna untuk menemukan jarak terdekat alternatif bila terjadi hambatan.

1.6 Keaslian Penelitian

Penelitian yang berhubungan dengan masalah pemanfaatan teknologi informasi dalam pengembangan pencarian jarak terdekat sudah pernah dilakukan oleh beberapa peneliti sebelumnya, sedangkan untuk pencarian jarak terdekat dan waktu tempuh dengan mempertimbangkan segala kemungkinan hambatan yang terjadi di jalanan belum pernah dilakukan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melalui beberapa proses dalam perancangan dan implementasi sistem penentuan jarak terpendek dan waktu tempuh menggunakan algoritma dijkstra maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Telah dapat dibuat suatu sistem yang dapat membantu menemukan jarak terpendek disertai waktu tempuh menggunakan algoritma dijkstra dengan konsep pemrograman berorientasi objek.
2. Sistem mampu menemukan jalur alternatif apabila terjadi pemutusan jalan pada jalur semula.

Berdasarkan pengujian 10 responden, sistem penentuan jarak terpendek dan waktu tempuh menggunakan algoritma dijkstra dengan pemrograman berbasis objek dapat membantu mempermudah pengguna untuk menemukan jarak terpendek antar daerah dan jarak terpendek alternatif apabila terdapat hambatan pada jarak terpendek semula dengan persentase untuk pengujian *interface* menunjukkan bahwa 65% responden setuju dan 35% kurang setuju. Sedangkan untuk pengujian fungsional sistem menunjukkan bahwa 93% menyatakan setuju dan 7% menyatakan kurang setuju.

5.2 Saran

Adapun beberapa saran yang penulis berikan sebagai acuan untuk pengembangan sistem selanjutnya adalah :

1. Desain pada sistem ini masih sederhana sehingga diperlukan modifikasi *user interface* agar lebih menarik, seperti ditambahkan visualisasi pada sistem.
2. Menerapkan algoritma lain selain dijkstra, yaitu dengan algoritma Floyd atau algoritma Ford untuk mencari jarak terpendek dalam suatu lintasan.

DAFTAR PUSTAKA

- Anonim., 2002, *Pemrograman Borland Delphi 7*, Andi, Yogyakarta.
- Beta ALGOLIST Algorithms Visualizers Source Code. “*Dijkstra’s Algorithm*”. http://www.algolist.com/Dijkstra%27s_algorithm diunduh 26 Februari 2011.
- Ekaputra, Adriansyah., 2006, *Aplikasi Graf Pada Persoalan Lintasan Terpendek Dengan Algoritma Dijkstra*, Jurnal program studi teknik informatika Institut Teknologi Bandung (ITB), Bandung.
- Eryanta, Agus., 2010, *Pencarian Rute Terpendek Wisata di Bali Menggunakan Algoritma A* (A-Star)*, skripsi S1 Teknik Informatika, Universitas Ahmad Dahlan, Yogyakarta.
- Google Map of Yogyakarta. http://maps.google.co.id/maps?q=google+map+of+yogyakarta&oe=utf-8&rls=org.mozilla:en-US:official&client=firefox-a&um=1&ie=UTF-8&hq=&hnear=0x2e7a5787bd5b6bc5:0x21723fd4d3684f71,Yogyakarta&gl=id&ei=oYL5Tdm3G4-rrAeXrrD1Dw&sa=X&oi=geocode_result&ct=image&resnum=1&ved=0CBYQ8gEwAA diunduh 20 April 2011.
- Kristanto, Andri., 2003, *Perancangan Sistem Informasi dan Aplikasinya*, Gava Media, Yogyakarta.
- McHugh, J., 1990, *Algorithmic Graph Theory*, Prentice Hall International Inc, New York.

- Rahmanto, Dedy., 2004, *Aplikasi Sig Untuk Informasi jalur Tranportasi Angkutan Kota Di Wilayah Kotamadya Yogyakarta*, skripsi S1 Teknik Informatika Universitas Ahmad Dahlan, Yogyakarta.
- Reita, Ruka., 2008, *Sistem Basis Data*, <http://one.indoskripsi.com/judul-skripsi-tugas-makalah/tugas-kuliah-lainnya/mysql>. Diunduh 06 Desember 2010.
- Sidi, Erick Purnomo., 2007, *Simulasi Pemilihan Rute Terpendek Dengan Algoritma Dijkstra Menggunakan Macromedia Flash*, skripsi S1 Teknik Informatika, Universitas Ahmad Dahlan, Yogyakarta.
- Stroustrup, B., 1991, *The C++ Programming Language, 2nded.*, Addison Weasley Publishing Company, Massachusett.
- Susanto, K., 1995, *Pemrograman Berorientasi pada Objek dengan Borland C++*, Andi Offset, Yogyakarta.
- Sutiasih., 2002, *Rekayasa Perangkat Lunak Penentuan Jalur Terpendek dengan Algoritma Dijkstra Menggunakan Konsep Pemrograman Berorientasi Obyek*, skripsi S1 Teknik Informatika, Universitas Ahmad Dahlan, Yogyakarta.
- Swanawati, R., 1998, *Visualisasi Pohon Jarak Terpendek dengan algoritma dijkstra*, Universitas Kristen Duta Wacana, Yogyakarta.

LAMPIRAN

```
unit UMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, jpeg;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Panel1: TPanel;
    Start: TButton;
    Finish: TButton;
    Blokir_Jalan: TButton;
    Cari: TButton;
    Kecepatan: TButton;
    btnWaktu: TButton;
    Help: TButton;
    ListBox1: TListBox;
    Panel2: TPanel;
    Hapus_Layar: TButton;
    Buat_Garis: TButton;
    Matikan: TCheckBox;
    Keluar: TButton;
    ScrollBox1: TScrollBox;
    Image1: TImage;
    conn: TADOConnection;
    q1: TADOQuery;
    Panggil_Graf: TButton;
    Simpan_Graf: TButton;
    Alternatif: TButton;

    procedure Image1MouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure FormCreate(Sender: TObject);
    procedure StartClick(Sender: TObject);
    procedure FinishClick(Sender: TObject);
    procedure Blokir_JalanClick(Sender: TObject);
    procedure Simpan_GrafClick(Sender: TObject);
    procedure Panggil_GrafClick(Sender: TObject);
    procedure CariClick(Sender: TObject);
```

```

procedure KeluarClick(Sender: TObject);
procedure Hapus_LayarClick(Sender: TObject);
procedure MatikanClick(Sender: TObject);
procedure Buat_GarisClick(Sender: TObject);
procedure KecepatanClick(Sender: TObject);
procedure btnWaktuClick(Sender: TObject);
procedure HelpClick(Sender: TObject);
procedure AlternatifClick(Sender: TObject);

private
  Mati: boolean;
end;

var
  Form1: TForm1;
  jumlah: integer;
  Titik: array [1..50] of TTitik;
  JmlGaris: integer;
  Garis: array [1..50] of TGaris;
  Tt1,Tt2: String;
  Ttk1, Ttk2: array[1..50] of string;
  AkhirCari: integer;
  AwalCari: integer;

implementation

uses TataCara;

{$R *.dfm}

procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
var
  namaTitikBaru: String;

begin
  if Button= MbRight then
    if mati= false then
      if (jumlah>50) then
        begin
          showMessage('Titik Maksimal');
          jumlah:=jumlah-1;
        end else

```

```

begin
    inc(jumlah);
    Image1.Canvas.Ellipse(x-7, y-7, x+7, y+7);
    Image1.Canvas.TextOut(x-5, y-5,intToStr (jumlah));

repeat
    namaTitikBaru := InputBox('Masukkan nama titik', 'Nama Titik: ', '');
    until Length(namaTitikBaru) > 0;
Image1.Canvas.TextOut(x, y-20, namaTitikBaru);
Titik[jumlah]:=TTitik.Create(self);
Titik[jumlah].SetNoTitik(jumlah);
Titik[jumlah].SetNameTitik(namaTitikBaru);
end;

Hapus_Layar.Enabled:=True;
Simpan_Graf.Enabled:=True;
Matikan.Enabled:=True;
Buat_Garis.Enabled:=True;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    mati:= false;
    jumlah:= 0;
end;

function minimum(a,b: real): real;
begin
    if a<b then
        minimum:=x
    else
        minimum:=y;
end;

procedure TForm1.CariClick(Sender: TObject);
var D: array [1..50] of real;
    M: array [1..50,1..50] of real;
    Permanen: array [1..50] of boolean;
    i,j,k: byte;
    Ds: array [1..50,1..50] of real;
    Dp: array [1..50] of byte;
    x,F: integer;
    Iterasi: integer;

```

```

begin

// mengosongkan matriks M
for i:=1 to 50 do
begin
  for j:=1 to 50 do
  end;

// mengisi matriks M
for i:=1 to jmlGaris do
begin
  Titik[jumlah]:=TTitik.Create(self);
  Titik[jumlah].SetNoTitik(jumlah);
  Titik[jumlah].SetNameTitik(namaTitikBaru);

  permanen[i]:=false;
  M[Garis[i].GetNoTitikAwal,Garis[i].GetNoTitikAkhir]:=Garis[i].GetBobot;

  if garisBlokir[i] = 1 then
    M[Garis[i].GetNoTitikAwal,Garis[i].GetNoTitikAkhir]:=Infinity;
end;

// memberi label pada titik awal
D[AwalCari]:=0;

// memberi label sementara pada semua titik
for i:=1 to jumlah do
  if i<>AwalCari then D[i]:=1;

k:=AwalCari;
Permanen[AwalCari]:=true;

for i:=1 to jumlah do
begin
  Ds[iterasi,i]:=D[i];
end;

// selama terminal belum permanen
while not permanen[AkhirCari] do
begin
  inc(Iterasi);

// menentukan label sementara untuk
// titik yang tidak berlabel permanen
for i:=1 to jumlah do
  if not permanen[i] then

```



```

        D[i]:=minimum(D[i],D[k]+M[k,i]);

// menentukan nilai minimum sementara
for i:=1 to jumlah do
begin
    if not permanen[i] then

        begin
            if d[i] < mins then

                begin
                    mins:=D[i];
                end;
            end;
        end;

k:=AwalCari;
Permanen[AwalCari]:=true;

for i:=1 to jumlah do
begin
    Ds[iterasi,i]:=D[i];
end;

    for i:=1 to jumlah do
    begin
        Ds[Iterasi,i]:=D[i];
    end;
    Dp[Iterasi]:=k;
end;

image1.Canvas.MoveTo(Titik[AkhirCari].GetX,Titik[AkhirCari].GetY);
image1.Canvas.Pen.Color:=ClRed;
image1.Canvas.Pen.Width:=3;

    F:=AkhirCari;
    i:=iterasi;
    while F<>awalCari do

begin
    Dec(i);
    If ((Ds[i,F]>Ds[i+1,F]) or (i=awalCari)) then
    begin
        x:=Titik[f].GetNoTitik;
    end;
end;

```

```

Image1.Canvas.LineTo(Titik[x].GetX,Titik[x].GetY);
Image1.Canvas.MoveTo(Titik[x].GetX,Titik[x].GetY);
end;
end;

Image1.Canvas.LineTo(Titik[AwalCari].GetX,Titik[AwalCari].GetY);

ListBox1.Items.add("");
ListBox1.Items.add('Jarak Terpendek dari '+IntToStr(AwalCari)+' ke
'+IntToStr(AkhirCari)+' = '+FloatToStr(mins)+' Km');
//ListBox1.Items.add('Jml iterasi: '+IntToStr(iterasi));

Blokir_Jalan.Enabled:=True;
Kecepatan.Enabled:=True;
end;

procedure TForm1.Buat_GarisClick(Sender: TObject);
var Tt1,Tt2: String;
    Ttk1, Ttk2: array[1..50] of string;
    x1, x2, y1, y2: integer;
    jarak: string;
    i: integer;
    Ok: boolean;

begin
    inc(JmlGaris);
    if JmlGaris>50 then
        showMessage('Garis Maksimal..')
    else begin

//memberi masukan titik pangkal
repeat
//Ok:=True;
    repeat
        Ok:=True;
        Tt1:=Inputbox('Masukkan titik pangkal:',' ');

        if Length(Tt1) = 0 then
            begin MessageDlg('Masukan belum terisi. Silahkan ulangi kembali',
                mtWarning, [mbOK], 0);

                begin
                    Dec(JmlGaris);
                end;
            end;
        end;
    end;
end;

```

```

    Exit;
end;

    for i:=1 to length(Tt1) do
        if not (Tt1[i] in ['0'..'9'])
            then Ok:=false;
        if not Ok then MessageDlg('Isi Angka', mtWarning, [mbOK], 0);
        until Ok;

if StrToInt (Tt1) > jumlah then
begin
    MessageDlg('Nomor Titik tidak sesuai. Silahkan ulangi kembali',
    mtWarning, [mbOK], 0);

    begin
        Dec(JmlGaris);
    end;
    exit;
end
until Ok;

if Length(Tt1) = 0 then
begin
    Dec(JmlGaris);
    Exit;
end
else
    Ttk1[JmlGaris]:=Tt1;

//memberi masukan titik ujung
repeat
//Ok:=true;
    repeat
        Ok:=true;
        Tt2:=Inputbox('Masukan titik ujung:', '');
    if Length(Tt2) = 0 then
        begin MessageDlg('Masukan belum terisi. Silahkan ulangi kembali',
        mtWarning, [mbOK], 0);

        begin
            Dec(JmlGaris);
        end;
        Exit;
    end;
    for i:=1 to length(Tt2) do

```

```

        if not (Tt2[i] in ['0'..'9','.'])
            then Ok:=false;
        if not Ok then MessageDlg('Isi Angka', mtWarning, [mbOK], 0);
    until Ok;

if (StrToInt (Tt2) > jumlah) or (StrToInt (Tt2) = StrToInt (Tt1)) then
begin
    MessageDlg('Nomor Titik tidak sesuai. Silahkan ulangi kembali',
    mtWarning, [mbOK], 0);

    begin
        Dec(JmlGaris);
    end;
    exit;
end
until Ok;

if Length(Tt2) = 0 then
begin
    Dec(JmlGaris);
    Exit;
end
else
    Ttk2[JmlGaris]:=Tt2;

if JmlGaris > 1 then
begin
    for i:=1 to (JmlGaris-1) do
        if ((StrToInt(Ttk1[JmlGaris])=Garis[i].GetNoTitikAwal)
        and
        (StrToInt(Ttk2[JmlGaris])=Garis[i].GetNoTitikAkhir)
        or (StrToInt(Ttk2[JmlGaris])=Garis[i].GetNoTitikAKhir)
        and (StrToInt(Ttk1[JmlGaris])=Garis[i].GetNoTitikAwal))
        then begin
            showMessage('Garis sudah ada..');
            JmlGaris:=JmlGaris-1;
        End;
    end;
end;

//memberi masukan titik jarak
repeat
    Ok:=true;
    Jarak:=inputbox('Masukkan jarak:', 'jarak (Km)', '');

```

```

for i:=1 to length (jarak) do
    if not (jarak[i] in ['0'..'9','.'])
        then Ok:=false;
    If not Ok then MessageDlg('Isi Angka', mtWarning, [mbOK], 0);
until Ok;

```

```

x1:=Titik[StrToInt(Ttk1[JmlGaris])].GetX;
y1:=Titik[StrToInt(Ttk1[JmlGaris])].GetY;
x2:=Titik[StrToInt(Ttk2[JmlGaris])].GetX;
y2:=Titik[StrToInt(Ttk2[JmlGaris])].GetY;

```

```

Form1.image1.Canvas.Pen.width:=1;
Form1.image1.Canvas.Pen.color:=ClBlack;
Form1.image1.Canvas.MoveTo(x1,y1);
Form1.image1.Canvas.LineTo(x2,y2);

```

```

Form1.ListBox1.Items.add(IntToStr(GetNoGaris)+ ' ' +
IntToStr(GetNoTitikAwal)+ ' ke ' +IntToStr(GetNoTitikAkhir)+ ' jarak:
'+FloatToStr(GetBobot)+' Km');
end;

```

```

Start.Enabled:=True;
end;

```

```

procedure TForm1.MatikanClick(Sender: TObject);
begin
    if Matikan.Checked=true then mati:=true;
end;

```

```

procedure TForm1.KeluarClick(Sender: TObject);
begin
    if MessageDlg('apakah anda yakin akan
keluar?',mtConfirmation,[mbYes,mbNo],0)=mrYes
then
    Application.Terminate;
end;

```

```

procedure TForm1.Hapus_LayarClick(Sender: TObject);
var i: byte;

begin
    ListBox1.Items.Clear;

```

```

        Image1.Picture.Bitmap:=nil;
        jumlah:=0;
    JmlGaris := 0;

    for i := 1 to 50 do
        garisBlokir[i] := 0;
    end;

procedure TForm1.StartClick(Sender: TObject);
var Ttk3: String;
    Ok: boolean;
    i: byte;

begin
    repeat
        //Ok:=True;
        repeat
            Ok:=True;
            Ttk3:=Inputbox('Masukkan awal pencarian:', '');
            if Length(Ttk3) = 0 then

                begin MessageDlg('Masukan belum terisi. Silahkan ulangi kembali',
                    mtWarning, [mbOK], 0);
                    Exit;
                end;

        until Ok;

        if (Length(Ttk3) = 0) or (StrToInt(Ttk3) > jumlah) then
            begin
                MessageDlg('Nomor garis tidak sesuai. Silahkan ulangi kembali', mtWarning,
                    [mbOK], 0);
                ok:=false;
            end
        until Ok;

        conn.Open();
        q1.SQL.Text := 'SELECT * FROM titik WHERE sec = ' + pilihSec + ' AND
            NoTitik = ' + Ttk3;
        q1.Open;

        AwalCari:=StrToInt(Ttk3);
        ListBox1.Items.add("");
        ListBox1.Items.add('Titik Awal pencarian: '+IntToStr(AwalCari));

```

```

ListBox1.Items.add('Titik Awal pencarian: '+IntToStr(AwalCari) + ' -
q1.FieldValues['NamaTitik']);
conn.Close;}

```

```

Finish.Enabled:=True;
end;

```

```

procedure TForm1.FinishClick(Sender: TObject);
var Ttk4: String;
    Ok: boolean;
    i: byte;

begin
  repeat
    //Ok:=True;
    repeat
      Ok:=True;
      Ttk4:=Inputbox('Masukkan akhir pencarian:','');
      begin MessageDlg('Masukan belum terisi. Silahkan ulangi kembali',
mtWarning, [mbOK], 0);
        Exit;
      end;

      for i:=1 to length(Ttk4) do
        if not (Ttk4[i] in ['0'..'9','.']) then Ok:=false;
        If not Ok then MessageDlg('Isi Angka', mtWarning, [mbOK], 0);
      until Ok;

      if (Length(Ttk4) = 0) or (StrToInt(Ttk4) > jumlah) then
        begin
          MessageDlg('Nomor garis tidak sesuai. Silahkan ulangi kembali', mtWarning,
[mbOK], 0);
          ok:=false;
        end
      until Ok;

      conn.Open();
      q1.SQL.Text := 'SELECT * FROM titik WHERE sec = ' + pilihSec + ' AND
NoTitik = ' + Ttk4;
      q1.Open;

      AkhirCari:=StrToInt(Ttk4);
      ListBox1.Items.add('Titik Akhir pencarian: '+IntToStr(AkhirCari));

```

```
ListBox1.Items.add('Titik Akhir pencarian: '+IntToStr(AkhirCari) + ' - ' +
q1.FieldValues['NamaTitik']);
conn.Close;
```

```
Cari.Enabled:=True;
end;
```

```
procedure TForm1.Blokir_JalanClick(Sender: TObject);
```

```
var x: string;
    i,x1,x2,y1,y2: integer;
    Ok: boolean;
```

```
begin
```

```
  Repeat
```

```
  //Ok:=true;
```

```
  Repeat
```

```
    Ok:=true;
```

```
    x:=inputBox('Nomor garis yang di blokir: ','');
```

```
    if Length(x) = 0 then
```

```
      begin MessageDlg('Garis blokir belum terisi. Silahkan ulangi kembali',
```

```
        mtWarning, [mbOK], 0);
```

```
        Exit;
```

```
      end;
```

```
    for i:=1 to length(x) do
```

```
      if not (x[i] in ['0'..'9'])
```

```
        then Ok:=false;
```

```
      if not Ok then MessageDlg('Isi Angka', mtWarning, [mbOK], 0);
```

```
    until Ok;
```

```
  if (Length(x) = 0) or (StrToInt(x) > JmlGaris) then
```

```
  begin
```

```
    MessageDlg('Nomor garis tidak sesuai. Silahkan ulangi kembali', mtWarning,
```

```
    [mbOK], 0);
```

```
    ok:=false;
```

```
  end
```

```
  else
```

```
  begin
```

```
    garisBlokir[StrToInt(x)] := 1;
```

```
    Garis[StrToInt(x)].SetBobot(0);
```

```
    i := StrToInt(x);
```

```
    x1:=Titik[Garis[i].GetNoTitikAwal].GetX;
```

```
    y2:=Titik[Garis[i].GetNoTitikAkhir].GetY;
```

```
    Form1.image1.Canvas.Pen.width:=3;
```



```

    Form1.image1.Canvas.Pen.color:=clBlue;
    Form1.Image1.Canvas.MoveTo(x1,y1);
    Form1.Image1.Canvas.LineTo(x2,y2);
end;
until ok;
ListBox1.Items.add('Nomor garis yang diblokir: '+x);

Alternatif.Enabled:=True;
end;

procedure TForm1.Simpan_GrafClick(Sender: TObject);
var i: byte;
    secTerakhir: Byte;
    namaSec : String;

begin
try
try
conn.Open;
q1.SQL.Text := 'START TRANSACTION';
q1.ExecSQL;

if q1.RecordCount > 0 then
begin
secTerakhir := q1.FieldValues['sec'];
secTerakhir := secTerakhir + 1;
end
else
secTerakhir := 1;

repeat
namaSec := InputBox('Nama Scenario', 'Masukkan Nama Scenario: ', '');
until Length(namaSec) > 0;

q1.SQL.Text := 'INSERT INTO sec (sec, namasec) VALUES (:a, :b)';
q1.Parameters[0].Value := secTerakhir;
q1.Parameters[1].Value := namaSec;
q1.ExecSQL;

for i:=1 to jumlah do
begin
//q1.Parameters.Clear;
q1.Parameters[0].value := secTerakhir;
q1.Parameters[1].value := Titik [i].GetX;

```

```

    q1.Parameters[2].value := Titik [i].GetY;
    q1.Parameters[3].value := Titik [i].GetNoTitik;
    q1.Parameters[4].value := Titik [i].GetNamaTitik;
    q1.ExecSQL;
end;

for i:=1 to JmlGaris do
begin
    q1.Parameters[0].value := secTerakhir;
    q1.Parameters[1].value := Garis [i].GetNoGaris;
    q1.Parameters[2].value := Garis [i].GetNoTitikAwal;
    q1.Parameters[3].value := Garis [i].GetNoTitikAkhir;
    q1.Parameters[4].value := Garis [i].GetBobot;
    q1.ExecSQL;
end;

q1.SQL.Text := 'COMMIT';
q1.ExecSQL;

    MessageDlg('sukses simpan data', mtInformation, [mbOk], 0) ;
except
    q1.SQL.Text := 'ROLLBACK';
    q1.ExecSQL;

    MessageDlg('gagal simpan data', mtInformation, [mbOk], 0) ;
end;
finally
    conn.Close;
end;
end;

procedure TForm1.Panggil_GrafClick(Sender: TObject);
var i: byte;
    x1, y1, x2, y2: integer;
    NoGaris, NoTitikAwal, NoTitikAkhir: integer;
    maxSec: string;
    Ok : Boolean;

begin
    if MessageDlg('Yakin untuk reset data?', mtConfirmation, [mbYes, mbNo], 0) =
mrYes then
    begin
        Hapus_LayarClick(Sender);
        jumlah := 0;
        JmlGaris := 0;

```

```

//pembersihan garis dan titik
for i := 1 to 50 do
begin
  Titik[i] := nil;
  Garis[i] := nil;
  Ttk1[i] := "";
  Ttk2[i] := "";

end;

//mengisi titik, diambil dari database
try
  try

    repeat
      //Ok:=True;
      repeat
        begin
          Ok:=True;
          pilihSec := "";

          q1.SQL.Text := 'SELECT * FROM sec';
          q1.Open;

          for i := 1 to StrToInt(maxSec) do
            begin
              pilihSec := pilihSec + q1.Fields[0].AsString + '.' + q1.Fields[1].AsString
                + chr(13);
              q1.Next;
            end;

          pilihSec := InputBox('Pilih Secenario', 'Masukkan nomor scenario yang
            dipilih: ' + chr(13) + pilihSec, "");
          if Length(pilihSec) = 0 then
            begin
              MessageDlg('Masukan belum terisi. Silahkan ulangi kembali',
                mtWarning, [mbOK], 0);
              Exit;
            end;

          for i:=1 to length(pilihSec) do
            if not (pilihSec[i] in ['0'..'9','.']) then Ok:=false;
            if not Ok then MessageDlg('Isi Angka', mtWarning, [mbOK], 0);

```

```

        begin
        if (Length(pilihSec) = 0) then
            exit;
        end;
        end;
until Ok;

if (StrToInt(pilihSec) > StrToInt(maxSec)) or (StrToInt(pilihSec) <= 0) then
begin
    MessageDlg('Nomor Graf tidak sesuai. Silahkan ulangi kembali', mtWarning,
    [mbOK], 0);
    ok:=false;
    end;
until ok ;

    q1.SQL.Text := 'SELECT * FROM titik WHERE (sec = ' + pilihSec + ')
ORDER BY NoTitik';
    q1.Open;

    begin
    x1 := q1.FieldValues['x'];
    y1 := q1.FieldValues['y'];
    Titik[i]:=TTitik.Create(self);
    Ttk1[Titik[i].GetNoTitik] := IntToStr(x1);
    Ttk2[Titik[i].GetNoTitik] := IntToStr(y1);
    Image1.Canvas.Ellipse(x1-7, y1-7, x1+7, y1+7);
    Image1.Canvas.TextOut(x1-5, y1-5,IntToStr(i));
    Image1.Canvas.TextOut(x1, y1-20, Titik[i].GetNamaTitik);
    jumlah := jumlah + 1;
    q1.Next;
    end;
except
    MessageDlg('Gagal koneksi', mtError, [mbOK], 0);
end;

finally
    q1.Close;
    conn.Close;
end;

//-----

```

```

//mengisi garis, diambil dari database
try
try
    conn.Open;
    q1.SQL.Text := 'SELECT * FROM garis WHERE sec = ' + pilihSec;
    q1.Open;

    begin
        Garis[i]:=TGaris.Create;
        Garis[i].SetNoGaris(NoGaris);
        Garis[i].SetNoTitikAwal(NoTitikAwal);
        Garis[i].SetNoTitikAkhir(NoTitikAkhir);
        Garis[i].SetBobot(q1.FieldValues['Bobot']);

        x1 := StrToInt(ttk1[Garis[i].getNoTitikAwal]);
        y1 := StrToInt(ttk2[Garis[i].getNoTitikAwal]);
        x2 := StrToInt(ttk1[Garis[i].getNoTitikAkhir]);
        y2 := StrToInt(ttk2[Garis[i].getNoTitikAkhir]);

        Form1.image1.Canvas.Pen.width:=1;
        Form1.image1.Canvas.Pen.color:=ClBlack;
        Form1.image1.Canvas.MoveTo(x1,y1);
        Form1.image1.Canvas.LineTo(x2,y2);

        Form1.ListBox1.Items.add(IntToStr(Garis[i].GetNoGaris)+ ' ' +
        IntToStr(Garis[i].GetNoTitikAwal)+
        ' ke '+IntToStr(Garis[i].GetNoTitikAkhir)+' jarak:
        '+FloatToStr(Garis[i].GetBobot)+' Km');
    end;
except
    MessageDlg('Gagal koneksi, silahkan membuat graf manual', mtError,
    [mbOK], 0);
end;

finally
    q1.Close;
    conn.Close;
end;

ListBox1.Items.Add(Format('Jumlah Nomor Garis Jalan: %d, Jumlah Titik:
%d', [JmlGaris, jumlah]))
end;

Start.Enabled:=True;
Simpan_Graf.Enabled:=True;
Hapus_Layar.Enabled:=True;

```

```

    Buat_Garis.Enabled:=True;
    Matikan.Enabled:=True;
end;

procedure TForm1.KecepatanClick(Sender: TObject);
var Kec : string;
    Ok : boolean;
    i : byte;

begin
    repeat
        //Ok:=True;
        repeat
            Ok:=True;
            Kec:=Inputbox('Masukkan kecepatan:', '(Km/jam)', '');
            if Length(Kec) = 0 then
                begin
                    MessageDlg('Kecepatan belum terisi. Silahkan ulangi kembali',
                        mtWarning, [mbOK], 0);
                    Exit;
                end;

            for i:=1 to length(Kec) do
                if not (Kec[i] in ['0'..'9', '.']) then Ok:=false;
                if not Ok then MessageDlg('Isi Angka', mtWarning, [mbOK], 0);
            until Ok;

            if StrToInt (Kec) > 180 then
                begin
                    MessageDlg('Kecepatan Maksimal 180 Km/jam. Silahkan ulangi kembali',
                        mtWarning, [mbOK], 0);
                    ok:=false;
                end;

            Titik[jumlah]:=TTitik.Create(self);
            Titik[jumlah].SetNoTitik(jumlah);
            Titik[jumlah].SetNamaTitik(namaTitikBaru);

        end
    until ok;
    Kc:= StrToFloat(Kec);

    btnWaktu.Enabled:=True;
end;

```

```

procedure TForm1.btnWaktuClick(Sender: TObject);
var Waktu : real;
    jam : Integer;
    menit : Real;

begin
Waktu:= mins/Kc;
//ListBox1.Items.add('Kecepatan: '+FloatToStr(Kc)+' Km/jam');
ListBox1.Items.Add(Format('Waktu Tempuh: %d Jam, %0.0f menit', [jam,
menit]));
end;

procedure TForm1.HelpClick(Sender: TObject);
begin
    Form2.Show;
end;

procedure TForm1.AlternatifClick(Sender: TObject);
var D: array [1..50] of real;
    M: array [1..50,1..50] of real;
    Permanen: array [1..50] of boolean;
    i,j,k: byte;
    Ds: array [1..50,1..50] of real;
    Dp: array [1..50] of byte;
    x,F: integer;
    Iterasi: integer;

begin

    // mengosongkan matriks M
    for i:=1 to 50 do
    begin
        for j:=1 to 50 do
        end;

    // mengisi matriks M
    for i:=1 to jmlGaris do
    begin
        M[Garis[i].GetNoTitikAwal,Garis[i].GetNoTitikAkhir]:=Garis[i].GetBobot;
        if garisBlokir[i] = 1 then
            M[Garis[i].GetNoTitikAwal,Garis[i].GetNoTitikAkhir]:=Infinity;
            D[i]:=minimum(D[i],D[k]+M[k,i]);
        end;

    // memberi label pada titik awal
    D[AwalCari]:=0;

```

```

// memberi label sementara pada semua titik
for i:=1 to jumlah do
    if i<>AwalCari then D[i]:=1;

for i:=1 to jumlah do
begin
    Ds[iterasi,i]:=D[i];
    if not permanen[i] then
        D[i]:=minimum(D[i],D[k]+M[k,i]);

end;

// selama terminal belum permanen
while not permanen[AkhirCari] do
begin
    inc(Iterasi);

// menentukan label sementara untuk
// titik yang tidak berlabel permanen
for i:=1 to jumlah do
    if not permanen[i] then
        D[i]:=minimum(D[i],D[k]+M[k,i]);

// menentukan nilai minimum sementara
for i:=1 to jumlah do
begin
    if not permanen[i] then
begin
    if d[i] < mins then
begin
    k:=i;
end;
end;
end;

end;

for i:=1 to jumlah do
begin
    Ds[Iterasi,i]:=D[i];
end;
    Dp[Iterasi]:=k;
end;

```



```

        image1.Canvas.MoveTo(Titik[AkhirCari].GetX,Titik[AkhirCari].GetY);
        image1.Canvas.Pen.Color:=ClYellow;
        image1.Canvas.Pen.Width:=3;
        F:=AkhirCari;
        i:=iterasi;

        while F<>awalCari do
begin
    Dec(i);
    If ((Ds[i,F]>Ds[i+1,F]) or (i=awalCari)) then
Begin
    F:=Dp[i];
    x:=Titik[f].GetNoTitik;

    Image1.Canvas.LineTo(Titik[x].GetX,Titik[x].GetY);
    Image1.Canvas.MoveTo(Titik[x].GetX,Titik[x].GetY);
    Image1.Canvas.MoveTo(Titik[AkhirCari].GetX,Titik[AkhirCari].GetY);
    Image1.Canvas.Pen.Color:=ClYellow;
    Image1.Canvas.Pen.Width:=3;

    F:=AkhirCari;
    i:=iterasi;
    while F<>awalCari do
Titik[jumlah]:=TTitik.Create(self);
Titik[jumlah].SetNoTitik(jumlah);
Titik[jumlah].SetNamaTitik(namaTitikBaru);

end;
end;

Image1.Canvas.LineTo(Titik[AwalCari].GetX,Titik[AwalCari].GetY);
ListBox1.Items.add("");
ListBox1.Items.add('Jarak Terpendek Alternatif dari '+IntToStr(AwalCari)+' ke
'+IntToStr(AkhirCari)+' = '+FloatToStr(mins)+' Km');
//ListBox1.Items.add('Jml iterasi: '+IntToStr(iterasi));
end;

end.

```

```
unit UGaris;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, jpeg;

type
  TGaris = class(TImage)

  Private
    NoGaris: integer;
    NoTitikAwal: integer;
    NoTitikAkhir: integer;
    Bobot: real;

  Public
    procedure SetNoGaris(NoGarisBaru: integer);
    procedure SetNoTitikAwal(NoTitikAwalBaru: integer);
    procedure SetNoTitikAkhir(NoTitikAkhirBaru: integer);
    procedure SetBobot(BobotBaru: real);
    function GetNoGaris(): integer;
    function GetNoTitikAwal(): integer;
    function GetNoTitikAkhir(): integer;
    function GetBobot(): real;
  end;

implementation

Procedure TGaris.SetNoGaris(NoGarisBaru: integer);
Begin
  NoGaris:=NoBaru;
End;

Procedure TGaris.SetNoTitikAwal(NoTitikAwalBaru: integer);
Begin
  NoTitikAwal:=TitikAwalBaru;
End;

Procedure TGaris.SetNoTitikAkhir(NoTitikAkhirBaru: integer);
Begin
  NoTitikAkhir:=NoTitikAkhirBaru;
End;
```

```
Procedure TGaris.SetBobot(BobotBaru: real);
Begin
    Bobot:=BobotBaru;
End;

Function TGaris.GetNoGaris: integer;
Begin
    Result:=X;
End;

Function TGaris.GetNoTitikAwal: integer;
Begin
    Result:=Y;
End;

Function TGaris.GetNoTitikAkhir: integer;
Begin
    Result:=NoTitikAkhir;
End;

Function TGaris.GetBobot: real;
Begin
    Result:=Bobot;
End;

End.
```

```
unit UTitik;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls, jpeg;
```

```
type
```

```
TTitik = class(TImage)
```

```
Private
```

```
    x,y: integer;
    NoTitik: integer;
    NamaTitik: String;
```

```
Public
  procedure SetX(Xbaru: integer);
  procedure SetY(Ybaru: integer);
  procedure SetNoTitik(NoBaru: integer);
  procedure SetNamaTitik>NamaBaru: String);
  function GetX(): integer;
  function GetY(): integer;
  function GetNoTitik(): integer;
  function GetNamaTitik(): string;
end;

implementation

procedure TTitik.SetNamaTitik>NamaBaru: string);
begin
  NamaTitik :=>Nama;
end;

Procedure TTitik.SetX(XBaru: integer);
Begin
  X:=X;
End;

Procedure TTitik.SetY(YBaru: integer);
Begin
  Y:=Y;
End;

Procedure TTitik.SetNoTitik>NoBaru: integer);
Begin
  NoTitik:=NoBaru;
End;

Function TTitik.GetNamaTitik: String;
begin
  Result :=>NamaTitik;
end;

Function TTitik.GetX: integer;
Begin
  Result:=TitikX;
End;
```

```
Function TTitik.GetY: integer;  
Begin  
    Result:=TitikY;  
End;  
  
Function TTitik.GetNoTitik: integer;  
Begin  
    Result:=NoTitik;  
End;  
  
End.
```

KUESIONER

Dengan ini peneliti memohon kesediaan saudara/i untuk berpartisipasi dalam pengujian sistem **Penentuan Jarak Terpendek Dan Waktu Tempuh Menggunakan Algoritma Dijkstra Dengan Pemrograman Berbasis Objek** dengan mengisi kuisisioner ini.

Data Responden

Jenis Kelamin : laki-laki.....
Semester/jurusan : 2 / Informatika

Petunjuk pengisian

Jawablah dengan memberi tanda check list (✓) pada salahsatu alternatif jawaban yang telah tersedia.

Keterangan

S : Setuju
KS : Kurang Setuju
TS : Tidak Setuju

1. Pengujian Interface

No	Pernyataan	S	KS	TS
1	Program yang dihasilkan menarik		✓	
2	Program mudah dioperasikan	✓		

2. Pengujian Fungsional

No	Pernyataan	S	KS	TS
1	Program dapat menemukan jarak terpendek	✓		
2	Program dapat menemukan jarak terpendek alternatif setelah terdapat jalur yang diblokir	✓		
3	Hasil perhitungan jarak terpendek sesuai dengan rumus algoritma dijkstra	✓		

KUESIONER

Dengan ini peneliti memohon kesediaan saudara/i untuk berpartisipasi dalam pengujian sistem **Penentuan Jarak Terpendek Dan Waktu Tempuh Menggunakan Algoritma Dijkstra Dengan Pemrograman Berbasis Objek** dengan mengisi kuisisioner ini.

Data Responden

Jenis Kelamin : Perempuan
Semester/jurusan : 8 / Informatika

Petunjuk pengisian

Jawablah dengan memberi tanda check list (✓) pada salahsatu alternatif jawaban yang telah tersedia.

Keterangan

S : Setuju
KS : Kurang Setuju
TS : Tidak Setuju

1. Pengujian Interface

No	Pernyataan	S	KS	TS
1	Program yang dihasilkan menarik	✓		
2	Program mudah dioperasikan	✓		

2. Pengujian Fungsional

No	Pernyataan	S	KS	TS
1	Program dapat menemukan jarak terpendek	✓		
2	Program dapat menemukan jarak terpendek alternatif setelah terdapat jalur yang diblokir	✓		
3	Hasil perhitungan jarak terpendek sesuai dengan rumus algoritma dijkstra	✓		

KUESIONER

Dengan ini peneliti memohon kesediaan saudara/i untuk berpartisipasi dalam pengujian sistem **Penentuan Jarak Terpendek Dan Waktu Tempuh Menggunakan Algoritma Dijkstra Dengan Pemrograman Berbasis Objek** dengan mengisi kuisisioner ini.

Data Responden

Jenis Kelamin : Laki - laki
Semester/jurusan : V / P. Fisika

Petunjuk pengisian

Jawablah dengan memberi tanda check list (✓) pada salahsatu alternatif jawaban yang telah tersedia.

Keterangan

S : Setuju
KS : Kurang Setuju
TS : Tidak Setuju

1. Pengujian Interface

No	Pernyataan	S	KS	TS
1	Program yang dihasilkan menarik	✓		
2	Program mudah dioperasikan	✓		

2. Pengujian Fungsional

No	Pernyataan	S	KS	TS
1	Program dapat menemukan jarak terpendek	✓		
2	Program dapat menemukan jarak terpendek alternatif setelah terdapat jalur yang diblokir		✓	
3	Hasil perhitungan jarak terpendek sesuai dengan rumus algoritma dijkstra	✓		

KUESIONER

Dengan ini peneliti memohon kesediaan saudara/i untuk berpartisipasi dalam pengujian sistem **Penentuan Jarak Terpendek Dan Waktu Tempuh Menggunakan Algoritma Dijkstra Dengan Pemrograman Berbasis Objek** dengan mengisi kuisisioner ini.

Data Responden

Jenis Kelamin : laki-laki
Semester/jurusan : 6 / Inf.

Petunjuk pengisian

Jawablah dengan memberi tanda check list (✓) pada salahsatu alternatif jawaban yang telah tersedia.

Keterangan

S : Setuju
KS : Kurang Setuju
TS : Tidak Setuju

1. Pengujian Interface

No	Pernyataan	S	KS	TS
1	Program yang dihasilkan menarik		✓	
2	Program mudah dioperasikan	✓		

2. Pengujian Fungsional

No	Pernyataan	S	KS	TS
1	Program dapat menemukan jarak terpendek	✓		
2	Program dapat menemukan jarak terpendek alternatif setelah terdapat jalur yang diblokir	✓		
3	Hasil perhitungan jarak terpendek sesuai dengan rumus algoritma dijkstra	✓		

KUESIONER

Dengan ini peneliti memohon kesediaan saudara/i untuk berpartisipasi dalam pengujian sistem **Penentuan Jarak Terpendek Dan Waktu Tempuh Menggunakan Algoritma Dijkstra Dengan Pemrograman Berbasis Objek** dengan mengisi kuisisioner ini.

Data Responden

Jenis Kelamin : Perempuan
Semester/jurusan : VIII / Ilmu Perpustakaan

Petunjuk pengisian

Jawablah dengan memberi tanda check list (✓) pada salahsatu alternatif jawaban yang telah tersedia.

Keterangan

S : Setuju
KS : Kurang Setuju
TS : Tidak Setuju

1. Pengujian Interface

No	Pernyataan	S	KS	TS
1	Program yang dihasilkan menarik	✓		
2	Program mudah dioperasikan	✓		

2. Pengujian Fungsional

No	Pernyataan	S	KS	TS
1	Program dapat menemukan jarak terpendek	✓		
2	Program dapat menemukan jarak terpendek alternatif setelah terdapat jalur yang diblokir	✓		
3	Hasil perhitungan jarak terpendek sesuai dengan rumus algoritma dijkstra	✓		

KUESIONER

Dengan ini peneliti memohon kesediaan saudara/i untuk berpartisipasi dalam pengujian sistem **Penentuan Jarak Terpendek Dan Waktu Tempuh Menggunakan Algoritma Dijkstra Dengan Pemrograman Berbasis Objek** dengan mengisi kuisioner ini.

Data Responden

Jenis Kelamin

Semester/jurusan

Perempuan
:VI/ Pendidikan Biologi

Petunjuk pengisian

Jawablah dengan memberi tanda check list (✓) pada salahsatu alternatif jawaban yang telah tersedia.

Keterangan

S : Setuju

KS : Kurang Setuju

TS : Tidak Setuju

1. Pengujian Interface

No	Pernyataan	S	KS	TS
1	Program yang dihasilkan menarik	✓		
2	Program mudah dioperasikan	✓		

2. Pengujian Fungsional

No	Pernyataan	S	KS	TS
1	Program dapat menemukan jarak terpendek	✓		
2	Program dapat menemukan jarak terpendek alternatif setelah terdapat jalur yang diblokir	✓		
3	Hasil perhitungan jarak terpendek sesuai dengan rumus algoritma dijkstra	✓		

KUESIONER

Dengan ini peneliti memohon kesediaan saudara/i untuk berpartisipasi dalam pengujian sistem **Penentuan Jarak Terpendek Dan Waktu Tempuh Menggunakan Algoritma Dijkstra Dengan Pemrograman Berbasis Objek** dengan mengisi kuisisioner ini.

Data Responden

Jenis Kelamin : Perempuan
Semester/jurusan : 2 / Pend. Kimia

Petunjuk pengisian

Jawablah dengan memberi tanda check list (✓) pada salahsatu alternatif jawaban yang telah tersedia.

Keterangan

S : Setuju
KS : Kurang Setuju
TS : Tidak Setuju

1. Pengujian Interface

No	Pernyataan	S	KS	TS
1	Program yang dihasilkan menarik		✓	
2	Program mudah dioperasikan	✓		

2. Pengujian Fungsional

No	Pernyataan	S	KS	TS
1	Program dapat menemukan jarak terpendek	✓		
2	Program dapat menemukan jarak terpendek alternatif setelah terdapat jalur yang diblokir	✓		
3	Hasil perhitungan jarak terpendek sesuai dengan rumus algoritma dijkstra	✓		

KUESIONER

Dengan ini peneliti memohon kesediaan saudara/i untuk berpartisipasi dalam pengujian sistem **Penentuan Jarak Terpendek Dan Waktu Tempuh Menggunakan Algoritma Dijkstra Dengan Pemrograman Berbasis Objek** dengan mengisi kuisioner ini.

Data Responden

Jenis Kelamin : Perempuan
Semester/jurusan : VI / Pend Kimia

Petunjuk pengisian

Jawablah dengan memberi tanda check list (✓) pada salahsatu alternatif jawaban yang telah tersedia.

Keterangan

S : Setuju
KS : Kurang Setuju
TS : Tidak Setuju

1. Pengujian Interface

No	Pernyataan	S	KS	TS
1	Program yang dihasilkan menarik		✓	
2	Program mudah dioperasikan	✓		

2. Pengujian Fungsional

No	Pernyataan	S	KS	TS
1	Program dapat menemukan jarak terpendek	✓		
2	Program dapat menemukan jarak terpendek alternatif setelah terdapat jalur yang diblokir		✓	
3	Hasil perhitungan jarak terpendek sesuai dengan rumus algoritma dijkstra	✓		

KUESIONER

Dengan ini peneliti memohon kesediaan saudara/i untuk berpartisipasi dalam pengujian sistem **Penentuan Jarak Terpendek Dan Waktu Tempuh Menggunakan Algoritma Dijkstra Dengan Pemrograman Berbasis Objek** dengan mengisi kuisisioner ini.

Data Responden

Jenis Kelamin : Perempuan...
Semester/jurusan : 2 / Informatika

Petunjuk pengisian

Jawablah dengan memberi tanda check list (✓) pada salahsatu alternatif jawaban yang telah tersedia.

Keterangan

S : Setuju
KS : Kurang Setuju
TS : Tidak Setuju

1. Pengujian Interface

No	Pernyataan	S	KS	TS
1	Program yang dihasilkan menarik		✓	
2	Program mudah dioperasikan	✓		

2. Pengujian Fungsional

No	Pernyataan	S	KS	TS
1	Program dapat menemukan jarak terpendek	✓		
2	Program dapat menemukan jarak terpendek alternatif setelah terdapat jalur yang diblokir	✓		
3	Hasil perhitungan jarak terpendek sesuai dengan rumus algoritma dijkstra	✓		

KUESIONER

Dengan ini peneliti memohon kesediaan saudara/i untuk berpartisipasi dalam pengujian sistem **Penentuan Jarak Terpendek Dan Waktu Tempuh Menggunakan Algoritma Dijkstra Dengan Pemrograman Berbasis Objek** dengan mengisi kuisioner ini.

Data Responden

Jenis Kelamin : laki laki.....
Semester/jurusan : 8.../...Matematika

Petunjuk pengisian

Jawablah dengan memberi tanda check list (✓) pada salahsatu alternatif jawaban yang telah tersedia.

Keterangan

S : Setuju
KS : Kurang Setuju
TS : Tidak Setuju

1. Pengujian Interface

No	Pernyataan	S	KS	TS
1	Program yang dihasilkan menarik		✓	
2	Program mudah dioperasikan		✓	

2. Pengujian Fungsional

No	Pernyataan	S	KS	TS
1	Program dapat menemukan jarak terpendek	✓		
2	Program dapat menemukan jarak terpendek alternatif setelah terdapat jalur yang diblokir	✓		
3	Hasil perhitungan jarak terpendek sesuai dengan rumus algoritma dijkstra	✓		

CURRICULUM VITAE



Nama : Farida Ardiani
Tempat, Tanggal Lahir : Surabaya, 2 Desember 1987
Jenis Kelamin : Perempuan
Agama : Islam
Nama Ayah/Pekerjaan : Abdul Hadi/Wiraswasta
Nama Ibu/Pekerjaan : Chomisyah/Wiraswasta
Alamat Rumah : Jl. Raya Mulyosari no 05 Surabaya
Alamat Kost : Jl. Timoho GK IV/982 Gendeng Yogyakarta
No. HP : +6285747708862
E-mail : ardianie_12@yahoo.co.id

Riwayat Pendidikan

1994-2000 : SDN Kalisari I 242 Surabaya
2000-2003 : MTsN Tambak Beras Jombang
2003-2006 : MAN Tambak Beras Jombang
2006-2011 : Program Studi Teknik Informatika Fakultas Sains dan
Teknologi Universitas Islam Negeri Sunan Kalijaga
Yogyakarta