

**PENCARIAN RUTE TERPENDEK MENGGUNAKAN  
ALGORITMA NEAREST NEIGHBOR DAN SORTED EDGES  
DALAM SISTEM PENJEMPUTAN JEMAAT**

Skripsi



oleh

**NICO ADHITYA KURNIAWAN**

**22104891**

PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA

2018

## PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

### **PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA NEAREST NEIGHBOR DAN SORTED EDGES DALAM SISTEM PENJEMPUTAN JEMAAT**

yang saya kerjakan untuk melengkapi sebagian persyaratan menjadi Sarjana Komputer pada pendidikan Sarjana Program Studi Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, bukan merupakan tiruan atau duplikasi dari skripsi kesarjanaaan di lingkungan Universitas Kristen Duta Wacana maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Jika dikemudian hari didapati bahwa hasil skripsi ini adalah hasil plagiasi atau tiruan dari skripsi lain, saya bersedia dikenai sanksi yakni pencabutan gelar kesarjanaaan saya.

Yogyakarta, 8 Januari 2018



**NICO ADHITYA KURNIAWAN**  
22104891

## HALAMAN PERSETUJUAN

Judul Skripsi : PENCARIAN RUTE TERPENDEK  
MENGUNAKAN ALGORITMA NEAREST  
NEIGHBOR DAN SORTED EDGES DALAM  
SISTEM PENJEMPUTAN JEMAAT

Nama Mahasiswa : NICO ADHITYA KURNIAWAN

N I M : 22104891

Matakuliah : Skripsi (Tugas Akhir)


Kode : TIW276

Semester : Gasal

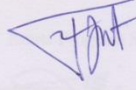
Tahun Akademik : 2017/2018

Telah diperiksa dan disetujui di  
Yogyakarta,  
Pada tanggal 8 Januari 2018

Dosen Pembimbing I

  
R. Gunawan Santosa, Drs. M.Si.

Dosen Pembimbing II

  
Antonius Rachmat C., S.Kom.,M.Cs.

**HALAMAN PENGESAHAN**

**PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA  
NEAREST NEIGHBOR DAN SORTED EDGES DALAM SISTEM  
PENJEMPUTAN JEMAAT**

Oleh: NICO ADHITYA KURNIAWAN / 22104891

Dipertahankan di depan Dewan Penguji Skripsi  
Program Studi Informatika Fakultas Teknologi Informasi  
Universitas Kristen Duta Wacana - Yogyakarta  
Dan dinyatakan diterima untuk memenuhi salah satu syarat memperoleh gelar  
Sarjana Komputer  
pada tanggal 19 Desember 2017

Yogyakarta, 8 Januari 2018  
Mengesahkan,

Dewan Penguji:

1. R. Gunawan Santosa, Drs. M.Si.
2. Antonius Rachmat C., S.Kom., M.Cs.
3. Danny Sebastian, S.Kom., M.M., M.T.
4. Gani Indriyanta, Ir. M.T.

Dekan

(Budi Susanto, S.Kom., M.T.)

Ketua Program Studi

(Gloria Virginia, Ph.D.)

## UCAPAN TERIMA KASIH

Dalam penelitian tugas akhir ini, penulis mendapatkan bantuan, saran, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Budi Susanto, S.Kom., M.T. selaku Dekan Fakultas Teknologi Informasi Universitas Kristen Duta Wacana.
2. Ibu Gloria Virginia, S.Kom., MAI., Ph.D. selaku Ketua Program Studi Informatika Universitas Kristen Duta Wacana.
3. Bapak Drs. R. Gunawan S., M. Si, selaku dosen pembimbing I yang telah memberikan waktu secara rutin untuk melakukan konsultasi dan memberikan saran dan masukan mengenai pemrograman sistem, dan penyelesaian masalah dengan cara yang lebih sederhana.
4. Bapak Antonius Rachmat C., S.Kom., M.Cs. selaku dosen pembimbing II yang telah meluangkan waktu untuk memberikan bimbingan dan memberikan saran serta masukan mengenai penulisan laporan juga analisis sistem.
5. Orangtua, saudara-saudara dan teman-teman terdekat yang selalu memberikan dukungan motivasi kepada penulis untuk menyelesaikan tugas akhir ini.
6. Pihak – pihak lain yang tidak dapat penulis sebutkan satu per satu yang berperan secara langsung maupun tidak langsung selama pengerjaan tugas akhir.

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat dan kasih karunia-Nya sehingga penulis dapat menyelesaikan pembuatan sistem dan laporan tugas akhir dengan judul “PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA *NEAREST NEIGHBOR* DAN *SORTED EDGES* DALAM SISTEM PENJEMPUTAN JEMAAT” dengan baik.

Penulisan laporan tugas akhir ini diajukan sebagai salah satu syarat guna mencapai gelar Sarjana Strata Satu (S1) di Fakultas Teknologi Informasi Program Studi Informatika Universitas Kristen Duta Wacana Yogyakarta.

Dalam pembuatan laporan ini, penulis menyadari bahwa masih ada kekurangan, baik dari materi maupun teknik penyajiannya. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran dari pembaca. Akhir kata penulis memohon maaf apabila dalam penulisan laporan ini, ada kalimat yang kurang berkenan. Semoga hasil dari pengerjaan tugas akhir ini dapat berguna dan bermanfaat bagi banyak pihak.

Yogyakarta, 29 November 2017

Penulis

## INTISARI

### PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA *NEAREST NEIGHBOR* DAN *SORTED EDGES* DALAM SISTEM PENJEMPUTAN JEMAAT

Penjemputan jemaat merupakan sebuah masalah yang mirip dengan *Travelling Salesman Problem*. Tujuan dari penelitian ini adalah mengembangkan sistem yang dapat menemukan rute penjemputan jemaat dengan jarak tempuh optimal, dengan menerapkan sekaligus menganalisa algoritma *Nearest Neighbor*, *Repetitive Nearest Neighbor*, dan *Sorted Edges*.

Pada penelitian ini, penulis menggunakan data berupa koordinat lokasi dalam kota Magelang beserta jalan yang menghubungkan lokasi-lokasi tersebut. Hasil dari penerapan algoritma untuk mencari rute terpendek akan dibandingkan dengan hasil dari algoritma *Brute Force*.

Hasil menunjukkan rata-rata *relative error* yang didapatkan melalui algoritma *Repetitive Nearest Neighbor* sebesar 7.56%, *Sorted Edges* 11.31%, dan *Nearest Neighbor* 20.99%. Kelemahan sistem adalah sistem terbatasnya data lokasi penjemputan yang dapat diuji sehingga hasil analisa hanya mencerminkan implementasi algoritma pada data kecil. Sistem juga tidak mengenali parameter-parameter yang mempengaruhi pemilihan rute seperti: tingkat kemacetan jalan, ada tidaknya lampu lalu lintas, dan sebagainya, sehingga rute yang dihasilkan tidak menggambarkan rute optimal secara nyata.

**Kata Kunci:** *Travelling Salesman Problem*, *Nearest Neighbor*, *Sorted Edge*, Rute Penjemputan, *Greedy*

## DAFTAR ISI

HALAMAN JUDUL .....	iii
PERNYATAAN KEASLIAN SKRIPSI.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PENGESAHAN.....	v
UCAPAN TERIMA KASIH.....	vi
KATA PENGANTAR .....	vii
INTISARI.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR .....	xiii
DAFTAR LAMPIRAN.....	xv
BAB 1    PENDAHULUAN .....	1
1.1.    Latar Belakang .....	1
1.2.    Rumusan Masalah .....	2
1.3.    Batasan Masalah.....	2
1.4.    Tujuan Penelitian.....	3
1.5.    Metodologi Penelitian .....	3
1.6.    Sistematika Penulisan.....	4
BAB 2    TINJAUAN PUSTAKA .....	6
2.1.    Tinjauan Pustaka .....	6
2.2.    Landasan Teori .....	7
2.2.1.    Struktur Data Graf.....	7
2.2.2.    Traveling Salesman Problem .....	9
2.2.3.    Algoritma <i>Greedy</i> .....	11
2.2.4.    Algoritma Floyd-Warshall .....	12
2.2.5.    Algoritma <i>Nearest Neighbor</i> .....	13
2.2.6.    Algoritma <i>Sorted Edges</i> .....	14
2.2.7.    Algoritma <i>Brute Force</i> .....	15
2.2.8. <i>Relative Error</i> .....	16
BAB 3    PERANCANGAN SISTEM.....	17



3.1.	Analisis Kebutuhan Sistem .....	17
3.1.1.	Kebutuhan Fungsional .....	17
3.1.2.	Kebutuhan Non – Fungsional.....	17
3.1.3.	Kebutuhan Hardware dan Software .....	17
3.2.	Rancangan Sistem .....	18
3.2.1.	Rancangan Penyimpanan Data.....	19
3.2.2.	Alur Kerja Tahap <i>Preprocessing</i> .....	20
3.2.3.	<i>Perancangan All Pairs Shortest Path</i> .....	21
3.2.4.	Rancangan Implementasi Algoritma <i>Nearest Neighbor</i> .....	22
3.2.5.	Rancangan Implementasi Algoritma <i>Sorted Edges</i> .....	23
3.2.6.	<i>Use Case Diagram</i> .....	25
3.2.7.	<i>User Interface</i> .....	27
3.2.8.	Rancangan Database .....	28
3.2.9.	Rancangan Pengujian .....	29
BAB 4	IMPLEMENTASI DAN ANALISIS SISTEM.....	31
4.1.	Implementasi Sistem .....	31
4.1.1.	Implementasi Penyimpanan Data.....	31
4.1.2.	Implementasi Pemodelan Graf .....	33
4.1.3.	Implementasi Pencarian <i>All Pairs Shortest Path</i> .....	33
4.1.4.	Implementasi Algoritma <i>Nearest Neighbor</i> .....	35
4.1.5.	Implementasi Algoritma <i>Repetitive Nearest Neighbor</i> .....	36
4.1.6.	Implementasi Algoritma <i>Sorted Edges</i> .....	37
4.1.7.	Implementasi <i>User Interface</i> .....	38
4.2.	Implementasi Program .....	39
4.2.1.	<i>Preprocessing</i> .....	39
4.2.2.	Proses Pencarian <i>All Pairs Shortest Path</i> .....	40
4.2.3.	Proses <i>Nearest Neighbor</i> .....	41
4.2.4.	Proses <i>Repetitive Nearest Neighbor</i> .....	42
4.2.5.	Proses <i>Sorted Edges</i> .....	42
4.2.6.	Proses <i>Brute Force</i> .....	43
4.3.	Hasil Pengujian Sistem.....	43
4.3.1.	Hasil Pengujian Terhadap 12 Lokasi Penjemputan .....	44

4.3.2.	Hasil Pengujian Terhadap 2 - 11 Lokasi Penjemputan .....	44
4.3.2.1.	Hasil Pengujian 11 Lokasi Penjemputan .....	45
4.3.2.2.	Hasil Pengujian 10 Lokasi Penjemputan .....	45
4.3.2.3.	Hasil Pengujian 9 Lokasi Penjemputan .....	46
4.3.2.4.	Hasil Pengujian 8 Lokasi Penjemputan .....	47
4.3.2.5.	Hasil Pengujian 7 Lokasi Penjemputan .....	47
4.3.2.6.	Hasil Pengujian 6 Lokasi Penjemputan .....	48
4.3.2.7.	Hasil Pengujian 5 Lokasi Penjemputan .....	49
4.3.2.8.	Hasil Pengujian 4 Lokasi Penjemputan .....	49
4.3.2.9.	Hasil Pengujian 3 Lokasi Penjemputan .....	50
4.3.2.10.	Hasil Pengujian 2 Lokasi Penjemputan .....	51
4.3.2.11.	Hasil Pengujian <i>Relative Error</i> .....	51
4.3.2.12.	Hasil Pengujian Waktu .....	52
4.4.	Analisis Pengujian Sistem .....	53
4.4.1.	Analisis Hasil Pengujian Jarak Tempuh .....	53
4.4.2.	Analisis Hasil Pengujian Waktu Pengerjaan Algoritma .....	57
4.4.3.	Pembahasan Analisis .....	58
BAB 5	KESIMPULAN DAN SARAN .....	59
5.1.	Kesimpulan .....	59
5.2.	Saran .....	59
DAFTAR PUSTAKA	.....	61
LAMPIRAN A	.....	A-1
LAMPIRAN B	.....	B-1
LAMPIRAN C	.....	C-1

## DAFTAR TABEL

Tabel 3.1 Tabel Use Case Memilih Lokasi Penjemputan .....	26
Tabel 3.2 Tabel Use Case Memilih Algoritma yang Dijalankan .....	26
Tabel 3.3 Tabel Use Case Simpan Hasil Kerja Sistem .....	27
Tabel 4.1 Hasil Pengujian terhadap 12 Lokasi Penjemputan.....	44
Tabel 4.2 Hasil Pengujian terhadap 11 Lokasi Penjemputan.....	45
Tabel 4.3 Hasil Pengujian terhadap 10 Lokasi Penjemputan.....	45
Tabel 4.4 Hasil Pengujian terhadap 9 Lokasi Penjemputan.....	46
Tabel 4.5 Hasil Pengujian terhadap 8 Lokasi Penjemputan.....	47
Tabel 4.6 Hasil Pengujian terhadap 7 Lokasi Penjemputan.....	47
Tabel 4.7 Hasil Pengujian terhadap 6 Lokasi Penjemputan.....	48
Tabel 4.8 Hasil Pengujian terhadap 9 Lokasi Penjemputan.....	49
Tabel 4.9 Hasil Pengujian terhadap 4 Lokasi Penjemputan.....	49
Tabel 4.10 Hasil Pengujian terhadap 3 Lokasi Penjemputan.....	50
Tabel 4.11 Hasil Pengujian terhadap 2 Lokasi Penjemputan.....	51
Tabel 4.12 Hasil Pengujian Pengujian Relative Error .....	52
Tabel 4.13 Hasil Pengujian Pengujian Waktu .....	52

## DAFTAR GAMBAR

Gambar 2.1. Contoh Penggambaran Graf .....	8
Gambar 2.2. Penggambaran path atau Rute P .....	8
Gambar 2.3. Jumlah Kota Beserta Kombinasinya pada TSP .....	9
Gambar 2.4. Hasil penelitian Dr. Iris van Rooij terhadap TSP .....	10
Gambar 2.5. <i>Klasifikasi TSP</i> .....	11
Gambar 2.6. <i>Pseudocode</i> Algoritma <i>Floyd-Warshall</i> .....	13
Gambar 2.7. <i>Pseudocode</i> Algoritma <i>Brute Force</i> .....	15
Gambar 3.1. Alur Kerja Sistem .....	19
Gambar 3.2. Alur Kerja Tahapan <i>Preprocessing</i> .....	21
Gambar 3.3. Alur Kerja Perancangan <i>All Pairs Shortest Path</i> .....	22
Gambar 3.4. Alur Kerja Rancangan Algoritma <i>Nearest Neighbor</i> .....	23
Gambar 3.5. Alur Kerja Rancangan Algoritma <i>Sorted Edges</i> .....	24
Gambar 3.6. <i>Use Case</i> Diagram Sistem .....	25
Gambar 3.7. Rancangan Tampilan Utama Sistem .....	27
Gambar 3.8. Rancangan Tampilan Sistem menjalankan <i>Sorted Edges</i> .....	28
Gambar 3.9. Rancangan Database .....	29
Gambar 4.1. Contoh Isi Data <i>db_vertex.txt</i> .....	31
Gambar 4.2. Contoh Isi Data <i>db_edge.txt</i> .....	32
Gambar 4.3. Contoh Isi Data <i>db_destination.txt</i> .....	32
Gambar 4.4. Alur Kerja Implementasi Pencarian <i>All Pairs Shortest Path</i> .....	34
Gambar 4.5. Alur Kerja Implementasi Algoritma <i>Nearest Neighbor</i> .....	35
Gambar 4.6. Alur Kerja Implementasi Algoritma <i>Repetitive Nearest Neighbor</i> .....	36
Gambar 4.7. Alur Kerja Implementasi Algoritma <i>Sorted Edges</i> .....	37
Gambar 4.8. Tampilan <i>User Interface</i> .....	38
Gambar 4.9. Contoh Tampilan Pengerjaan Algoritma <i>Nearest Neighbor</i> .....	39
Gambar 4.10. <i>Pseudocode</i> Tahapan <i>Preprocessing</i> .....	40
Gambar 4.11. <i>Pseudocode</i> Pencarian <i>All Pairs Shortest Path</i> .....	41
Gambar 4.12. <i>Pseudocode</i> Implementasi Algoritma <i>Nearest Neighbor</i> .....	41
Gambar 4.13. <i>Pseudocode</i> Implementasi Algoritma <i>Repetitive Nearest Neighbor</i> .....	42
Gambar 4.14. <i>Pseudocode</i> Implementasi Algoritma <i>Sorted Edges</i> .....	42
Gambar 4.15. <i>Pseudocode</i> Implementasi Algoritma <i>Brute Force</i> .....	43

Gambar 4.16. Grafik Hasil Pengujian Jarak Tempuh pada 11 Lokasi Penjemputan .....	54
Gambar 4.17. Grafik Hasil Pengujian Jarak Tempuh pada 2 Lokasi Penjemputan .....	55
Gambar 4.18. Grafik Hasil Pengujian <i>Relative Error</i> .....	55
Gambar 4.19. Grafik Perbandingan Rata-Rata <i>Relative Error</i> Antar Algoritma.....	56
Gambar 4.20. Grafik Hasil Pengujian Waktu .....	57

©UKDW

## DAFTAR LAMPIRAN

Lampiran A .....	A-1
Lampiran B.....	B-1
Lampiran C.....	C-1

©UKDW

## INTISARI

### PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA *NEAREST NEIGHBOR* DAN *SORTED EDGES* DALAM SISTEM PENJEMPUTAN JEMAAT

Penjemputan jemaat merupakan sebuah masalah yang mirip dengan *Travelling Salesman Problem*. Tujuan dari penelitian ini adalah mengembangkan sistem yang dapat menemukan rute penjemputan jemaat dengan jarak tempuh optimal, dengan menerapkan sekaligus menganalisa algoritma *Nearest Neighbor*, *Repetitive Nearest Neighbor*, dan *Sorted Edges*.

Pada penelitian ini, penulis menggunakan data berupa koordinat lokasi dalam kota Magelang beserta jalan yang menghubungkan lokasi-lokasi tersebut. Hasil dari penerapan algoritma untuk mencari rute terpendek akan dibandingkan dengan hasil dari algoritma *Brute Force*.

Hasil menunjukkan rata-rata *relative error* yang didapatkan melalui algoritma *Repetitive Nearest Neighbor* sebesar 7.56%, *Sorted Edges* 11.31%, dan *Nearest Neighbor* 20.99%. Kelemahan sistem adalah sistem terbatasnya data lokasi penjemputan yang dapat diuji sehingga hasil analisa hanya mencerminkan implementasi algoritma pada data kecil. Sistem juga tidak mengenali parameter-parameter yang mempengaruhi pemilihan rute seperti: tingkat kemacetan jalan, ada tidaknya lampu lalu lintas, dan sebagainya, sehingga rute yang dihasilkan tidak menggambarkan rute optimal secara nyata.

**Kata Kunci:** *Travelling Salesman Problem*, *Nearest Neighbor*, *Sorted Edge*, Rute Penjemputan, *Greedy*

# BAB 1

## PENDAHULUAN

### 1.1. Latar Belakang

Penjemputan jemaat menjadi kegiatan rutin yang dilakukan oleh Gereja Kristen Indonesia (GKI) Pajajaran Magelang. Kegiatan penjemputan diperuntukan bagi jemaat yang hendak mengikuti ibadah atau kegiatan gereja namun terkendala dengan masalah transportasi. Jemaat akan dijemput di rumah masing-masing menggunakan fasilitas mobil gereja yang telah disediakan.

Masalah yang ada adalah rute perjalanan yang dilalui seringkali tidak efektif dalam hal pemanfaatan waktu dan sumber daya bensin. Dikarenakan perlunya menjemput beberapa jemaat yang tinggal dengan alamat berbeda-beda, sekaligus menghemat penggunaan bensin dan waktu, maka diperlukan sistem yang dapat mencari rute penjemputan jemaat terpendek tersebut.

*Travelling Salesman Problem* merupakan masalah dalam ilmu komputer dimana seorang pedagang diberikan daftar kota untuk dilewati, dan ia perlu menemukan cara terefisien untuk mengunjungi semua kota tersebut dan kembali lagi ke tempat ia berasal (Pacha-Sucharzewski, 2011). Masalah ini mirip dengan kondisi yang dipaparkan di atas, dimana mobil perlu menjemput beberapa jemaat, dengan meminimalis jarak perjalanan.

Mohammad & Khidhir (2016) menyebutkan bahwa algoritma *Nearest Neighbor* dan *Sorted Edges* dapat dengan cepat menemukan rute terpendek pada kasus *Travelling Salesman Problem*, namun hasil yang diperoleh tidak selalu optimal. Pengembangan dari *Nearest Neighbor* yang disebut *Repetitive Nearest Neighbor* juga dapat digunakan untuk menyelesaikan kasus ini dengan cepat.

Dalam penelitian ini, penulis akan menggunakan algoritma *Nearest Neighbor*, *Repetitive Nearest Neighbor* dan *Sorted Edges* dalam menyelesaikan masalah pencarian rute penjemputan jemaat. Ketiga algoritma akan dibandingkan dan diuji dalam hal kecepatan serta ketepatan hasil yang dikeluarkan.



## 1.2. Rumusan Masalah

Berdasarkan masalah yang dijabarkan pada latar belakang, maka perumusan masalah yang ada antara lain yaitu:

- a) Apakah algoritma *Nearest Neighbor*, *Repetitive Nearest Neighbor* dan *Sorted Edges* dapat menemukan rute penjemputan dengan jarak tempuh mendekati jarak tempuh yang dihasilkan algoritma *Brute Force*?
- b) Bagaimana perbandingan hasil dari algoritma *Nearest Neighbor*, *Repetitive Nearest Neighbor* dan *Sorted Edges* berdasarkan jarak tempuh dan kecepatan dari algoritma?

## 1.3. Batasan Masalah

Dalam penelitian ini terdapat beberapa batasan masalah yang digunakan, yaitu :

- a) Sistem dimodelkan dalam struktur data graf, dengan lokasi-lokasi dalam peta merupakan *vertices* dan jalan yang menghubungkan dua lokasi merupakan *edges*.
- b) Data-data yang dipakai oleh sistem, akan disimpan dalam file teks secara lokal.
- c) Data lokasi yang digunakan diambil dari data <http://www.map.google.com/>.
- d) Data lokasi yang diambil, merupakan lokasi yang terdapat di sebagian kota Magelang, lebih tepatnya lokasi-lokasi yang terdapat pada Kelurahan Panjang, Kelurahan Rejowinangun Utara, Kelurahan Rejowinangun Selatan, Kelurahan Tidar Utara, Kelurahan Selatan, Kelurahan Wates, Kelurahan Gelangan, Kelurahan Panjang, Kelurahan Magersari, Kelurahan Cacaban, dan Kelurahan Magelang.
- e) Lokasi jemaat yang dijemput diambil dari data anggota jemaat GKI Pajajaran Magelang, tanpa menyertakan alamat lengkap, dan berjumlah total 12 lokasi.

- f) *Input* sistem berupa pemilihan lokasi jemaat yang perlu dijemput dan pemilihan algoritma yang akan dijalankan.

#### 1.4. Tujuan Penelitian

Melalui penelitian ini, tujuan yang ingin dicapai adalah:

- a) Mengembangkan sistem dengan pemodelan graf, yang mampu mencari rute penjemputan jemaat GKI Pajajaran Magelang terpendek.
- b) Mengetahui tingkat ketepatan dan kecepatan algoritma *Nearest Neighbor*, *Repetitive Nearest Neighbor* dan *Sorted Edges* dalam mencari rute penjemputan jemaat terpendek.

#### 1.5. Metodologi Penelitian

Proses penelitian pencarian rute terpendek dengan algoritma *Nearest Neighbor* dan *Sorted Edges* memiliki beberapa langkah. Langkah – langkah yang akan dilakukan adalah sebagai berikut:

- a) Tahap Pertama (Studi Pustaka)

Studi pustaka dilakukan dengan cara mempelajari teori - teori yang berkaitan dengan *Traveling Salesman Problem*, algoritma *Nearest Neighbor*, dan *Sorted Edges* melalui beberapa buku, jurnal, artikel, dan bahan lain yang mendukung untuk penelitian ini.

- b) Tahap Kedua (Pengumpulan data dan pemodelan graf)

Pengumpulan data diambil dari data anggota jemaat GKI Pajajaran Magelang beserta titik-titik lokasi dan persimpangan yang terdaftar dalam peta kota Magelang. Lokasi tersebut akan disimpan oleh sistem sebagai *vertex* dalam graf dan jalan-jalan yang menghubungkan lokasi-lokasi tersebut disimpan sebagai *edge*.

- c) Tahap Ketiga (Implementasi atau *coding*)

Tahap ini merupakan tahap pengerjaan sistem untuk mendapatkan rute penjemputan terpendek dengan menggunakan algoritma *Nearest Neighbor* dan

*Sorted Edges*. Sistem akan menghasilkan *output* berupa rute perjalanan, jarak tempuh, serta kecepatan sistem dalam menjalankan algoritma tersebut dalam satuan waktu.

d) Tahap Keempat (Evaluasi sistem)

Tahap evaluasi akan dilakukan dengan cara membandingkan output yang dihasilkan oleh algoritma *Nearest Neighbor* dan *Sorted Edges*. Sistem akan mengevaluasi jarak tempuh yang dihasilkan kedua algoritma serta kecepatan algoritma tersebut.

e) Tahap Kelima (Kesimpulan)

Pada tahap ini akan ditarik kesimpulan berdasarkan hasil dari tingkat kecepatan dan efisiensi dari hasil evaluasi yang telah dilakukan.

## **1.6. Sistematika Penulisan**

Sistematika penulisan yang digunakan penulis untuk mengerjakan laporan tugas akhir ini sesuai dengan ketentuan dari buku panduan penulisan tugas akhir. Penulisan akan dibagi menjadi 5 bab, yaitu :

Bab I : Pendahuluan

Pada bab pertama akan dibahas mengenai latar belakang, rumusan masalah, batasan dari sistem yang akan dibangun, tujuan yang ingin dicapai oleh penulis, dan metodologi penelitian.

Bab II : Tinjauan Pustaka

Pada bab kedua akan menjelaskan mengenai tinjauan pustaka dan landasan teori. Pada tinjauan pustaka akan berisi penelitian-penelitian yang sudah pernah dilakukan oleh orang lain berkaitan dengan penelitian yang akan dilakukan oleh penulis. Sedangkan landasan teori akan berisi tentang dasar-dasar teori yang akan digunakan pada penelitian ini, termasuk metode-metode yang akan digunakan.

Bab III : Perancangan Sistem

Pada bab ketiga penulis membahas mengenai perancangan sistem yang akan dibuat. Bab ini menjelaskan bagaimana tahapan – tahapan pembangunan

sistem, analisis kebutuhan sistem, rancangan sistem, serta rancangan pengujian dari sistem.

#### Bab IV : Implementasi dan Analisis Sistem

Pada bab keempat penulis akan membahas mengenai hasil implementasi yang telah dilakukan. Hasil yang akan didapatkan dapat berupa hasil riset atau penelitian yang telah dilakukan oleh penulis.

#### Bab V : Kesimpulan

Pada bab kelima penulis akan membahas mengenai kesimpulan dari hasil penelitian dan saran. Kesimpulan berisi tentang hasil yang telah didapatkan selama penelitian berlangsung. Sedangkan saran dapat berupa sesuatu hal yang dapat dilakukan berkaitan dengan pengembangan topik penelitian tersebut di masa mendatang serta metode yang dapat digunakan.

©UKDW

## BAB 5

### KESIMPULAN DAN SARAN

#### 5.1. Kesimpulan

Berdasarkan hasil analisis penelitian yang telah dibahas pada Bab 4, maka dapat ditarik kesimpulan sebagai berikut:

1. Algoritma *Repetitive Nearest Neighbor* dapat menemukan rute penjemputan dengan jarak tempuh yang mendekati jarak tempuh yang dihasilkan algoritma *Brute Force*. Algoritma *Nearest Neighbor* dan *Sorted Edges* juga dapat menemukan rute penjemputan, namun dengan perbedaan jarak tempuh yang jauh lebih besar.
2. Algoritma *Nearest Neighbor* berjalan paling cepat dan konsisten, namun tidak menghasilkan rute dengan jarak optimal. Algoritma *Sorted Edges* berjalan cukup cepat, terutama pada data uji dengan lokasi penjemputan yang sedikit, dan menghasilkan rute dengan perbedaan jarak yang lebih baik dari algoritma *Nearest Neighbor*. Algoritma *Repetitive Nearest Neighbor* berjalan secepat algoritma *Sorted Edges*, dan menghasilkan rute yang paling optimal jika dibandingkan dengan algoritma *Nearest Neighbor* dan *Sorted Edges*.

#### 5.2. Saran

Sistem ini sangat memungkinkan untuk dilakukan pengembangan lebih lanjut sesuai kebutuhan yang terus bertambah, sehingga dapat meningkatkan akurasi sistem. Saran yang diajukan penulis dalam pengembangan sistem kedepannya adalah sebagai berikut:

1. Memperbaiki dan memperkaya daftar lokasi dan jalan yang digunakan oleh sistem, juga menyertakan kondisi jalan seperti ada tidaknya lampu lalu lintas, tingkat kemacetan, dan lain-lain. Sehingga proses pencarian rute dapat menghasilkan hasil yang lebih akurat dan nyata.

2. Memperkaya data uji yaitu lokasi penjemputan, sehingga analisa algoritma *Nearest Neighbor* dan *Sorted Edges* dapat menghasilkan perbedaan algoritma yang lebih signifikan dan akurat.

©UKDW

## DAFTAR PUSTAKA

- Abdulkarim, H. A., & Alshammari, I. F. (2015). Comparison of Algorithms for Solving Traveling Salesman Problem. *International Journal of Engineering and Advanced Technology (IJEAT)*, 4(6), 76-79.
- Abid, M. M., & Muhammad, I. (2015). Heuristic Approaches to Solve Traveling Salesman Problem. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 15(2), 390-396.
- Adewole, A., Agwuegbo, S., & Otubamowo, K. (2011). Simulated Annealing For Solving The Traveling Salesman Problem. *Journal of Natural Sciences, Engineering and Technology*, 10(2), 1-9.
- AlSalibi, B. A., Jelodar, M. B., & Venkat, I. (2013). A Comparative Study between the Nearest Neighbor and Genetic Algorithms: A revisit to the Traveling Salesman Problem. *International Journal of Computer Science and Electronics Engineering (IJCSEE)*, 3(1), 34-38.
- Arora, K., Agarwal, S., & Tanwar, R. (2016). Solving TSP using Genetic Algorithm and Nearest Neighbour Algorithm and their Comparison. *International Journal of Scientific & Engineering Research*, 7(1), 1014-1048.
- Biron, D. (2006). *The Traveling Salesman Problem: Deceptively Easy to State; Notoriously Hard to Solve*. Liberty University.
- Clark, D. M. (2007). Graph and Optimization. *Journal of Inquiry-Based Learning in Mathematics*, 2.
- Diestel, R. (2000). *Graph Theory*. New York: Springer.
- Garg, H., & Rawat, P. (2012). An Improved Algorithm for Finding All Pair Shortest Path. *International Journal of Computer Applications*, 47(25), 35-37.
- Gutin, G., Yeo, A., & Zverovich, A. (2001). Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. *Discrete Applied Mathematics*(117), 81-68.
- Li, X. R., & Zhao, Z. (2005). *Relative Error Measures for Evaluation of Estimation Algorithms*. Institute of Electrical and Electronics Engineers.
- Matai, R., Singh, S., & Mittal, M. L. (2010). *Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches*. (P. D. Davendra, Ed.) InTech.

Mohammad, A. K., & Khidhir, G. I. (2016). A Study Of The Efficiency Of Sequencing Within Plants Using Computational Algorithm. *ZANCO Journal of Pure and Applied Sciences (ZJPAS)*, 28(3), 86-96.

Pacha-Sucharzewski, M. (2011). Analysis of the “Travelling Salesman Problem” and an Application of Heuristic Techniques for Finding a New Solution. *Undergraduate Review*, 7, pp. 81-86.

Stencek, J. (2013). *Traveling Salesman Problem*. JAMK University of Applied science.

Subhadra, A. (2016). Greedy Algorithms: Analysis, Design & Applications. *International Journal of Informative & Futuristic Research*, 3(5), 1749-1764.

©UKPDW