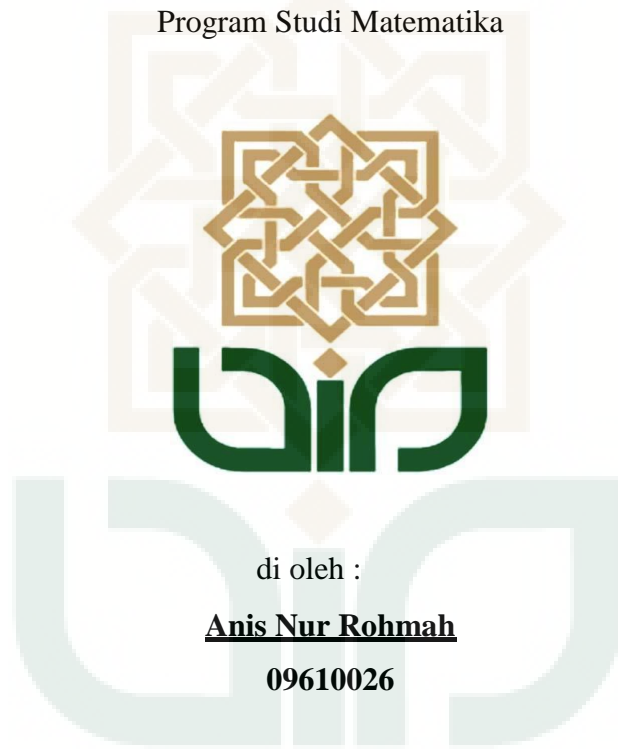


**APLIKASI ALGORITMA *CHEAPEST INSERTION HEURISTICS* (CIH)
PADA PENDISTRIBUSIAN SURAT SUARA PEMILIHAN UMUM DI DESA
MLIWIS, CEPOGO, BOYOLALI**

SKRIPSI

untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-1
Program Studi Matematika



di oleh :

Anis Nur Rohmah

09610026

Kepada

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA**

2013



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi

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 : Anis Nur Rohmah

NIM : 09610026

Judul Skripsi : Aplikasi Algoritma *Cheapest Insertion Heuristics (CIH)* pada Pendistribusian Surat Suara Pemilihan Umum di Desa Mliwis, Cepogo, Boyolali

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

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

Wassalamu'alaikum wr. wb.

Yogyakarta, 30 Agustus 2013

Pembimbing

Noor Saif Muhammad Musafi, S.Si, M.Sc.
NIP. 19820617 200912 1 005



PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : UIN.02/D.ST/PP.01.1/2784/2013

Skripsi/Tugas Akhir dengan judul : Aplikasi Algoritma *Cheapest Insertion Heuristics* (CIH) pada Pendistribusian Surat Suara Pemilihan Umum di Desa Mliwis, Cepogo, Boyolali

Yang dipersiapkan dan disusun oleh :
Nama : Anis Nur Rohmah
NIM : 09610026
Telah dimunaqasyahkan pada : 12 September 2013
Nilai Munaqasyah : A
Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

Ketua Sidang

Noor Saif Muh. Mussafi, M.Sc
NIP. 19820617 200912 1 005

Penguji I

Mochammad Abrori, S.Si, M.Kom
NIP.19720423 199903 1 003

Penguji II

Pipit Pratiwi Rahayu, M.Sc

Yogyakarta, 16 September 2013

UIN Sunan Kalijaga

Fakultas Sains dan Teknologi

Dekan



Prof. Drs. H. Akh. Mijhaji, M.A, Ph.D
NIP. 19580919 198603 1 002

SURAT PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini :

Nama : Anis Nur Rohmah
NIM : 09610026
Prodi / Smt : Matematika / VIII
Fakultas : Sains dan Teknologi

Dengan ini saya menyatakan bahwa skripsi ini 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 orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.



Yogyakarta, 30 Agustus 2013

Yang menyatakan



Anis Nur Rohmah

NIM: 09610026

SURAT PERNYATAAN BERJILBAB

Yang bertanda tangan di bawah ini:

Nama : Anis Nur Rohmah
NIM : 09610026
Prodi : Matematika
Fakultas : Sains dan Teknologi

Dengan ini menyatakan bahwa pas foto yang disertakan dalam daftar munaqosah itu adalah pas foto berjilbab, dan saya menanggung resiko dari pas foto tersebut. Jika di suatu hari ada hal-hal yang tidak diinginkan berkenaan dengan jilbab yang saya kenakan, maka hal tersebut tidak ada kaitannya dengan pihak universitas.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.
Diharapkan maklum adanya.

Yogyakarta, 30 Agustus 2013

Yang bersangkutan

METERAI
TEMPEL
PAJAK MEMBANGUN BANGSA
TGL. 20
AF097AAF403884225

ENAM RIBU RUPIAH
6000 DJP


Anis Nur Rohmah

NIM. 09610026

MOTTO

∞ *Hidup takkan berhenti selama hati dan pikiran tidak mati* ∞

∞ *Sesuatu yang belum dikerjakan, seringkali tampak mustahil; kita baru yakin kalau kita telah berhasil melakukannya dengan baik* ∞


- Evelyn Underhill-


∞ *Keutamaan seorang 'alim dibandingkan dengan seorang 'abid ibarat bulan purnama dan semua bintang* ∞


-HR. Abu Dawud-

HALAMAN PERSEMBAHAN

Skripsi ini penulis persembahkan kepada:

-  *Kedua Orang tua, Bapak Sofwan S.Pd.I dan Ibu Nuryani.*

-  *Almamater UIN Sunan Kalijaga Yogyakarta.*

-  *PPS Desa Mliwis, Cepogo, Boyolali.*

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan segala rahmat, nikmat dan karunia-Nya, sehingga penulis mampu menyelesaikan skripsi ini dalam rangka mengabdikan kepada-Nya. Sholawat beserta salam tak lupa penulis panjatkan kepada suri tauladan umat manusia sepanjang masa, Rasulullah SAW sang revolusioner sejati yang menjadi inspirasi setiap saat dalam memperbaiki umat manusia menuju masyarakat islami.

Alhamdulillah, akhirnya penulis dapat menyelesaikan penulisan skripsi yang berjudul **"Aplikasi Algoritma *Cheapest Insertion Heuristics (CIH)* pada Pendistribusian Surat Suara Pemilihan Umum di Desa Mliwis, Cepogo, Boyolali"** ini. Namun, tidak dapat dipungkiri bahwa skripsi ini dapat penulis buat dengan bantuan dan dukungan dari banyak pihak. Oleh karena itu, atas dukungan dan bantuan tersebut maka penulis mengucapkan banyak terima kasih kepada:

1. Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
2. Ketua Program Studi Matematika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
3. Bapak Noor Saif Muhammad Mussafi, S.Si, M.Sc., selaku dosen pembimbing yang telah meluangkan waktu dan tenaga untuk memberikan bimbingan dan arahan kepada penulis dalam menyelesaikan skripsi ini.
4. Segenap staf dosen dan karyawan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
5. Kedua Bapak Sofwan, S.Pd.I dan Ibu Nuryani, Dek Afik dan Dek Dewi serta seluruh keluarga yang telah memberikan dorongan baik moril maupun materiil selama penulis menimba ilmu di FST UIN Sunan Kalijaga Yogyakarta.

6. PPS Desa Mliwis, Cepogo, Boyolali atas semua bantuan dalam penelitian.
7. Sahabat-sahabat atas keceriaan, dukungan, tempat curhat dan semangat yang kalian berikan.
8. Moch. Shidqul Ahdi atas waktu dan tutorial singkat yang sangat bermanfaat.
9. Kos Perancis 3, terimakasih atas suka dan duka selama menimba ilmu serta motivasi dan dukungan dalam penyusunan skripsi ini.
10. Seluruh teman-teman Matematika 2009, untuk kebersamaan kita selama menimba ilmu di FST UIN Sunan Kalijaga selama ini.
11. Semua pihak yang tidak dapat penulis sebutkan satu per-satu yang telah membantu dalam penulisan skripsi ini.

Dengan penuh kesadaran bahwa skripsi ini masih jauh dari kesempurnaan. Oleh karena itu, saran dan kritik yang membangun dan mengarahkan untuk lebih baik, penulis terima dengan tangan terbuka. Walaupun demikian, penulis berharap skripsi ini dapat bermanfaat untuk kemaslahatan umat. Amin.

Yogyakarta, 30 Agustus 2013

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN SKRIPSI	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN KEASLIAN SKRIPSI	iv
HALAMAN PERNYATAAN BERJILBAB	v
HALAMAN MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR LAMBANG	xii
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xiv
ABSTRAKSI	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	3
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
1.6 Tinjauan Pustaka	5
1.7 Sistematika Penulisan	7
1.8 Metode Penelitian	8
BAB II DASAR TEORI	9
2.1. Teori Graf	9
2.1.1. Pengertian Graf	9
2.1.2. Ketetangaan dan Bersisian (<i>Adjacency and Incidency</i>).....	10
2.1.3. Derajat Simpul	10

2.1.4. Jalan, Lintasan, Jalur, Sirkuit	10
2.1.5. Graf Terhubung	12
2.1.6. Jenis-Jenis Graf	12
2.1.7. Graf Berbobot	14
2.1.8. Graf Lengkap	14
2.1.9. Graf Hamilton	14
2.2. Traveling Salesman Problem	15
2.3. Algoritma <i>Cheapest Insertion Heuristics</i>	17
2.3.1. Metode Heuristik	17
2.3.2. Algoritma <i>Cheapest Insertion Heuristics</i>	17
2.3.3. Contoh Kasus	19
2.4. Java	22
BAB III PEMBAHASAN	29
3.1 Pengumpulan Data dan Skema Penelitian	29
3.1.1 Profil Desa Mliwis	29
3.1.2 Pengumpulan Data	30
3.1.3 Evaluasi Pemilihan Umum Gubernur Tahun 2008.....	33
3.1.4 Skema Penelitian	34
3.1.5 Asumsi Penelitian	34
3.2 Pencarian Rute Optimal Menggunakan Algoritma <i>Cheapest Insertion Heuristics (CIH)</i>	35
3.3 Rancang Bangun Algoritma <i>Cheapest Insertion Heuristics (CIH)</i> dalam Pencarian Rute Optimal pada Pendistribusian Surat Suara di Desa Mliwis, Cepogo, Boyolali	56
3.3.1 Spesifikasi Kebutuhan	57
3.3.2 Pembuatan Rancang Bangun	57
3.3.3 Uji Coba Program	69
BAB IV PENUTUP	74
4.1. Kesimpulan	74
4.2. Saran.....	75
DAFTAR PUSTAKA	77
LAMPIRAN – LAMPIRAN	79

DAFTAR LAMBANG

G = Graf G

v = Simpul

e = Sisi

$V(G)$ = Himpunan simpul pada graf G

$E(G)$ = Himpunan sisi pada graf G

Arc / c_{ij} = Jarak antar simpul i dan simpul j

DAFTAR GAMBAR

Gambar 2.1	Graf G	9
	Gambar 2.2 Jalur Terpanjang x_0, \dots, x_k	11
	Gambar 2.3 Jalur Terpanjang x_i, \dots, x_k dan simpul yang bertetangga dengan x_k	12
Gambar 2.4	Sikel dengan panjang $\delta(G)+1$	12
Gambar 2.5	<i>Multiple Graph</i>	13
Gambar 2.6	<i>Pseudograph</i>	13
Gambar 2.7	Graf Berarah	14
Gambar 2.8	Graf Lengkap K_n	14
Gambar 2.9	Contoh <i>Subtour</i>	18
Gambar 2.10	Ilustrasi Jarak Antar Kota.....	19
Gambar 2.11	Solusi Optimal	21
Gambar 2.12	Tampilan Awal <i>NetBeans 6.9.1</i>	23
Gambar 3.1	Graf Lengkap K_{13}	36
Gambar 3.2	<i>New Project</i>	61
Gambar 3.3	<i>Design Tampilan Prototype</i>	65
Gambar 3.4	<i>Running Program</i>	69
Gambar 3.5	<i>Form Awal Program</i>	70
Gambar 3.6	Proses <i>Input Data</i>	71
Gambar 3.7	<i>Form tampilan Output</i>	72

DAFTAR TABEL

Tabel 2.1	Jarak Antar Kota	19
Tabel 2.2	Arc Penambah <i>subtour</i> ke-1.....	20
Tabel 2.3	Ukuran dan rentang tipe <i>integer</i>	24
Tabel 2.4	Ukuran dan rentang tipe <i>floating-point</i>	24
Tabel 3.1	Daftar TPS dan jumlah hak pilih di wilayah Desa Mliwis, Cepogo, Boyolali	31
Tabel 3.2	Jarak PPS ke setiap TPS dan antar TPS	32
Tabel 3.3	Asumsi simpul pada graf	36
Tabel 3.4	Arc Penambah <i>subtour</i> ke-1	38
Tabel 3.5	Arc Penambah <i>subtour</i> ke-2	39
Tabel 3.6	Arc Penambah <i>subtour</i> ke-3	41
Tabel 3.7	Arc Penambah <i>subtour</i> ke-4	42
Tabel 3.8	Arc Penambah <i>subtour</i> ke-5	44
Tabel 3.9	Arc Penambah <i>subtour</i> ke-6	46
Tabel 3.10	Arc Penambah <i>subtour</i> ke-7	48
Tabel 3.11	Arc Penambah <i>subtour</i> ke-8	50
Tabel 3.12	Arc Penambah <i>subtour</i> ke-9	52
Tabel 3.13	Arc Penambah <i>subtour</i> ke-10	54
Tabel 3.14	Arc Penambah <i>subtour</i> ke-11.....	55
Tabel 3.15	Spesifikasi Perangkat Keras.....	57
Tabel 3.16	Spesifikasi Perangkat Lunak.....	57
Tabel 3.17	Hasil Perhitungan	73
Tabel 3.18	Kelebihan dan Kekurangan	73

**APLIKASI ALGORITMA *CHEAPEST INSERTION HEURISTICS* (CIH)
PADA PENDISTRIBUSIAN SURAT SUARA PEMILIHAN UMUM
DI DESA MLIWIS, CEPOGO, BOYOLALI**

Anis Nur Rohmah
(09610026)

ABSTRAK

Surat suara merupakan elemen penting dalam sebuah pemilihan umum (Pemilu). Agar pendistribusian surat suara dari Panitia Pemungutan Suara (PPS) ke Tempat Pemungutan Suara (TPS) efektif dan efisien maka dibutuhkan jalur atau rute perjalanan yang pada umumnya menghasilkan waktu lebih singkat. Oleh karena itu, diperlukan cara untuk menentukan rute terpendek agar pendistribusian surat suara lebih efektif dan efisien. Masalah ini dapat dikategorikan sebagai masalah *Traveling Salesman problem* (TSP) yaitu permasalahan seorang *salesman* untuk mengunjungi n kota dengan masing-masing kota terhubung satu sama lain dan terdapat bobot perjalanan antarkota tersebut sehingga membentuk suatu graf lengkap berbobot. Berangkat dari suatu kota awal tertentu, seorang *salesman* harus mengunjungi $(n-1)$ kota lainnya tepat satu kali dan kembali pada kota awal keberangkatan.

Banyak algoritma yang telah ditemukan untuk menyelesaikan TSP, salah satunya algoritma *Cheapest Insertion Heuristic* (CIH). Algoritma CIH adalah Algoritma *Insertion* yang pada setiap penambahan kota baru yang akan disisipkan ke dalam *subtour* mempunyai bobot penyisipan paling minimal. Bobot penyisipan diperoleh dari persamaan $c(i,k,j) = c(i,k) + c(k,j) - c(i,j)$. Algoritma ini memberikan rute perjalanan yang berbeda tergantung dari urutan penyisipan kota-kota pada *subtour* yang bersangkutan.

Pada penelitian ini Algoritma CIH diterapkan pada pendistribusian surat suara di Desa Mliwis, Cepogo, Boyolali. Algoritma CIH berhasil dibuat menjadi perangkat lunak dengan memanfaatkan *IDE NetBeans 6.9.1* dan Basis Data, sehingga proses perhitungan dan penentuan rute terdekat akan jauh lebih cepat dibandingkan dengan perhitungan secara manual. Berdasarkan perhitungan program Algoritma CIH maupun perhitungan manual diperoleh jarak tempuh minimal untuk pendistribusian surat suara di Desa Mliwis, Cepogo Boyolali adalah 9,55 Km. Dengan penerapan rute optimal, diharapkan PPS dapat lebih maksimal dalam mempersiapkan pemilu dan pemilu dapat berjalan sesuai dengan ketentuan yang berlaku.

Kata kunci : *Traveling Salesman Problem*, Algoritma *Cheapest Insertion Heuristik*, *IDE NetBeans 6.9.1*, Desa Mliwis, Cepogo, Boyolali.

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pemilihan umum adalah sarana untuk melaksanakan kedaulatan rakyat berdasarkan asas langsung, umum, bebas, dan rahasia (LUBER), serta jujur dan adil (JURDIL). Salah satu hal yang sangat penting dalam pemilihan umum adalah surat suara. Agar surat suara dapat terdistribusi secara cepat maka diperlukan jalur distribusi yang efisien dari Panitia Pemungutan Suara (PPS) ke Tempat Pemungutan Suara (TPS). Oleh karena itu rute atau jalur terpendek dipilih untuk mengoptimalkan pendistribusian surat suara tersebut.

Penentuan rute merupakan aktifitas penting dalam proses pendistribusian, proses penentuan rute yaitu pergerakan antara dua zona (yang didapat dari tahap sebaran pergerakan) untuk moda tertentu (yang didapat dari pemilihan moda) dibebankan ke rute tertentu yang terdiri dari ruas jaringan jalan tertentu. Penentuan rute menjadi salah satu dari banyak permasalahan yang ada dalam aktivitas pendistribusian. Penentuan rute terpendek ini diharapkan dapat meminimalisir permasalahan dalam pemilihan umum. Misalnya keterlambatan pendistribusian surat suara sehingga menghambat pelaksanaan pemungutan suara.

Rute pendistribusian surat suara dari PPS ke beberapa TPS secara abstrak dapat digambarkan dengan suatu graf, dimana PPS dan TPS digambarkan sebagai simpul (*vertex*). Sedangkan jalan yang menghubungkan antara keduanya digambarkan sebagai sisi (*edge*). Masalah optimasi yang sering dijumpai dalam

pencarian rute terpendek yaitu *Travelling Salesman Problem (TSP)*. Masalah TSP dapat diselesaikan dengan menggunakan beberapa algoritma, antara lain yaitu *Algoritma Brute Force*, *Algoritma Branch and Bound*, *Algoritma Dijkstra*, *Heuristics* dan lain-lain. Salah satu algoritma dalam heuristik yang cukup efektif digunakan untuk menyelesaikan masalah TSP adalah Algoritma penyisipan (*Insertion*). Dalam Algoritma Penyisipan terdapat beberapa metode, salah satunya adalah Algoritma *Cheapest Insertion Heuristics*. Konsep dari Algoritma CIH yaitu menyisipkan suatu kota yang belum terlewati dengan jarak tambahan terpendek untuk menemukan solusi dari permasalahan. Algoritma ini memberikan hasil yang lebih baik jika dibandingkan dengan metode *insertion* yang lain karena untuk proses seleksi simpul yang akan disisipkan dilakukan pada setiap simpul di luar *tour* dan setiap sisi di dalam *tour*.

Kendala di lapangan menunjukkan perhitungan dalam pendistribusian serta sarana-prasarana pendistribusian yang belum memadai, letak beberapa TPS yang sulit dijangkau, dan kondisi beberapa jalan yang kurang baik menjadi penyebab lambatnya pendistribusian surat suara. Permasalahan ini menjadi sangat menarik dan penting untuk dilakukan kajian lebih lanjut dalam rangka meningkatkan efisiensi jarak pendistribusian, sehingga diperlukan penelitian dan perhitungan pencarian rute terpendek dengan metode *Algoritma Cheapest Insertion Heuristics*.

1.2. Rumusan Masalah

Rumusan masalah penelitian ini adalah

1. Bagaimana penerapan *Algoritma Cheapest Insertion Heuristics (CIH)* untuk optimasi rute pendistribusian surat suara di Desa Mliwis, Cepogo, Boyolali ?
2. Bagaimana rancang bangun program aplikasi optimasi rute pendistribusian surat suara di Desa Mliwis, Cepogo, Boyolali dengan menggunakan *Algoritma Cheapest Insertion Heuristics (CIH)*?

1.3. Batasan Masalah.

Batasan masalah dalam penelitian ini yaitu:

- a. Permasalahan yang dapat diselesaikan yaitu *Symetric Traveling Salesman Problem* di Desa Mliwis Cepogo Boyolali dengan menggunakan *Algoritma Cheapest Insertion Heuristics (CIH)*.
- b. Data yang digunakan yaitu data sekunder yang diperoleh dari PPS Desa Mliwis Cepogo Boyolali.
- c. Jalan yang digunakan dalam penelitian ini adalah jalan yang dapat dilalui oleh kendaraan dengan kondisi jalan sesuai dengan kondisi jalan yang dilewati kendaraan PPS.
- d. Analisis data menggunakan *IDE NetBeans 6.9.1*.
- e. Program aplikasi dapat menyelesaikan permasalahan dengan jumlah simpul maksimal adalah 20 simpul.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai dalam skripsi ini adalah

1. Menerapkan *Algoritma Cheapest Insertion Heuristics (CIH)* untuk optimasi rute pendistribusian surat suara di Desa Mliwis, Cepogo, Boyolali.
2. Mendesain rancang bangun program aplikasi optimasi rute pendistribusian surat suara di Desa Mliwis, Cepogo, Boyolali dengan menggunakan *Algoritma Cheapest Insertion Heuristics (CIH)*.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat bermanfaat bagi berbagai pihak, antara lain:

1. Manfaat bagi akademisi yaitu menambah pemahaman dari aplikasi teori graf terhadap penerapan *Algoritma Cheapest Insertion Heuristics (CIH)* untuk optimasi rute pendistribusian surat suara di Desa Mliwis, Cepogo, Boyolali.
2. Manfaat bagi PPS yaitu memberikan rekomendasi rute yang optimum tentang pendistribusian surat suara di Desa Mliwis, Cepogo, Boyolali, sehingga diharapkan pada masa mendatang pendistribusian surat suara lebih efisien.

1.6. Tinjauan Pustaka

Tinjauan pustaka skripsi ini terdiri dari beberapa jurnal dan skripsi sebagai referensi pelengkap guna menunjang kelengkapan penelitian.

Adapun penelitian sebelumnya yang menjadi acuan penulis antara lain jurnal yang menjadi rujukan utama adalah jurnal yang ditulis oleh Kusri, Jazi Eko Istiyanto (2007) yang berjudul "*Penyelesaian Travelling Salesman Problem Dengan Algoritma Cheapest Insertion Heuristics Dan Basis Data*". Penelitian ini menjelaskan pembuatan rancang bangun program aplikasi untuk menyelesaikan *Traveling Salesman Problem (TSP)* dengan menggunakan Algoritma *CIH* dan memanfaatkan *Borland Delphi 6* dan *Interbase 6* mulai dari perancangan algoritma, implementasi rancangan dan uji coba program.

Skripsi Ahmad Badru Tamam (2012) yang berjudul "*Aplikasi Algoritma Christofides' dan Software Logware pada Pendistribusian Produk PT. HEINZ ABC Indonesia Cabang Solo*". Penelitian ini menjelaskan tentang penyelesaian pendistribusian produk PT. HEINZ ABC Indonesia Cabang Solo dengan menggunakan Algoritma *Christofides'* secara manual dan *Software Logware* kemudian membandingkan hasil perhitungan yang dilakukan. Dalam penelitian ini diperoleh bahwa jarak tempuh kendaraan yang digunakan dalam pendistribusian menggunakan rute optimal yang direncanakan menggunakan algoritma *Christofides'* jarak tempuh lebih efisien 3 km dan waktu tempuh 28 menit lebih lama dibandingkan dengan rute pendistribusian yang ada di PT. HEINZ ABC Indonesia Cabang Solo. Sedangkan menggunakan software *Logware* lebih efisien 35 km dan waktu tempuh 5 menit lebih cepat dibandingkan dengan rute pendistribusian yang ada di PT. HEINZ

ABC Indonesia Cabang Solo dan lebih efisien 32 km dan 33 menit lebih cepat dibandingkan dengan algoritma Christofides’.

Jurnal M. Pasca Nugroho (2010) yang berjudul “*Penerapan Algoritma Branch and Bound dalam Menentukan Rute Terpendek untuk Perjalanan Antarkota di Jawa Barat*”. Penelitian ini menjelaskan cara menentukan rute perjalanan antarkota di Jawa Barat dengan memanfaatkan Algoritma *Branch and Bound*. Hasil dari penelitian ini, rute terpendek untuk perjalanan antarkota di Jawa Barat yang melibatkan 5 kota (Bandung, Jakarta, Sukabumi, Indramayu, dan Tasikmalaya) dengan total jarak tempuh 718 km yakni Bandung → Sukabumi → Jakarta → Indramayu → Tasikmalaya → Bandung.

Perbedaan penelitian ini dengan penelitian sebelumnya yaitu pada penelitian sebelumnya algoritma yang digunakan untuk menentukan rute optimal adalah Algoritma *Christofides’* dan Algoritma *Branch and Bound*, sedangkan pada penelitian ini menggunakan Algoritma *Cheapest Insertion Heuristics*, selain itu pada penelitian sebelumnya rancang bangun program aplikasi Algoritma CIH dirancang dengan menggunakan memanfaatkan *Borland Delphi 6*. Sementara itu, penelitian ini menggunakan *IDE NetBeans 6.9.1* untuk membuat rancang bangun program aplikasi Algoritma CIH untuk menyelesaikan masalah pendistribusian surat suara di Desa Mliwis Cepogo Boyolali.

1.7. Sistematika Penulisan

Tugas akhir ini disusun dengan sistematika sebagai berikut :

BAB I PENDAHULUAN

Bab ini membahas latar belakang penulisan skripsi, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan metode penelitian.

BAB II LANDASAN TEORI

Bab ini memuat landasan teori yang berfungsi sebagai sumber dalam memahami permasalahan yang berkaitan dengan teori graf, *Travelling Salesman Problem* , dan *Algoritma Cheapest Insertion Heuristics (CIH)*.

BAB III PEMBAHASAN

Bab ini merupakan pembahasan dari hasil penelitian yang berupa implementasi *Algoritma Cheapest Insertion Heuristics (CIH)* untuk optimasi rute pendistribusian surat suara di Desa Mliwis, Cepogo, Boyolali dengan perhitungan secara manual serta membuat rancang bangun *Algoritma Cheapest Insertion Heuristics (CIH)* untuk optimasi rute pendistribusian surat suara di Desa Mliwis, Cepogo, Boyolali.

BAB IV KESIMPULAN DAN SARAN

Bab ini merupakan bab penutup yang berisikan kesimpulan dan saran dari hasil penulisan tugas akhir ini.

1.8. Metode Penelitian

Tahap-tahap yang dilakukan dalam tugas akhir ini adalah identifikasi masalah, studi literatur, pengumpulan data, representasi dalam graf, dan penentuan rute optimal. Pada tahap Identifikasi masalah, permasalahan yang dibahas dalam skripsi ini adalah menentukan rute optimal pendistribusian surat suara dengan menggunakan Algoritma CIH.

Pada tahap studi literatur dan pengumpulan data dilakukan pengumpulan data sekunder dari PPS Desa Mliwis Cepogo Boyolali dan studi literatur. Studi ini meliputi hal-hal yang berkaitan dengan Algoritma CIH. Pembelajaran ini didapat baik dari buku-buku literatur, jurnal, paper, maupun beberapa artikel di internet.

Pada tahap selanjutnya adalah merepresentasikan masalah ke suatu graf lengkap berbobot sesuai dengan data sekunder yang telah diperoleh dari PPS Desa Mliwis Cepogo Boyolali.

Tahap terakhir adalah penentuan rute optimal yang merupakan proses perhitungan dengan menggunakan perhitungan manual maupun menggunakan rancang bangun program aplikasi optimasi rute pendistribusian surat suara di Desa Mliwis Cepogo Boyolali menggunakan Algoritma CIH.

BAB IV

PENUTUP

4.1 Kesimpulan

Berdasarkan pembahasan yang ada pada bab sebelumnya dalam merencanakan rute distribusi surat suara dari PPS ke seluruh TPS di Desa Mliwis, Cepogo, Boyolali dengan jarak tempuh minimal, maka dapat disimpulkan bahwa :

1. Langkah untuk mencari rute pendistribusian menggunakan Algoritma *CIH* secara manual dimulai dengan merepresentasikan permasalahan ke suatu graf lengkap, kemudian dibuat sebuah *subtour* yang menghubungkan PPS dengan TPS 12. Selanjutnya perjalanan dilanjutkan ke TPS lain hingga seluruh TPS terlewati dan kembali ke PPS dengan memperhitungkan tambahan jarak minimal berdasarkan rumus $c(i,k,j) = c(i,k) + c(k,j) - c(i,j)$ untuk menentukan TPS yang akan disisipkan ke dalam *subtour*. Menggunakan algoritma yang memiliki kompleksitas waktu $O(n^3)$ ini diperoleh jarak tempuh minimal pendistribusian surat suara di Desa Mliwis, Cepogo, Boyolali dengan menggunakan Algoritma *Cheapest Insertion Heuristics (CIH)* adalah 9550 m atau 9,55 km. Rute tersebut yaitu dari PPS → TPS 9 → TPS 3 → TPS 2 → TPS 12 → TPS 5 → TPS 7 → TPS 8 → TPS 4 → TPS 6 → TPS 10 → TPS 1 → TPS 11 → PPS.
2. Pembuatan rancang bangun Algoritma *CIH* ini terdiri dari dua tahap, yaitu tahap input data dan tahap olah data. pada tahap input data dibutuhkan informasi mengenai jumlah kota dan jarak antar kota. Banyaknya kombinasi

jarak antar kota diperoleh dengan memanfaatkan rumus kombinasi. Tahap selanjutnya adalah olah data. Olah data merupakan proses inti yang terdiri dari penentuan nilai kota sisipan untuk mengetahui kota yang menjadi kota sisipan, penentuan kota sisipan yaitu proses menentukan kota yang dapat disisipkan ke dalam *subtour*, dan menampilkan hasil akhir merupakan proses yang dilakukan setelah semua kota masuk ke dalam *subtour*. Selanjutnya dilakukan uji coba program untuk menentukan rute terpendek pendistribusian surat suara di Desa Mliwis, Cepogo, Boyolali sehingga diperoleh rute sebagai berikut: PPS → TPS 9 → TPS 3 → TPS 2 → TPS 12 → TPS 5 → TPS 7 → TPS 8 → TPS 4 → TPS 6 → TPS 10 → TPS 1 → TPS 11 → PPS dengan total jarak tempuh 9550 m atau 9,55 Km.

4.2 Saran-saran

Adapun saran-saran yang perlu diperhatikan untuk penerapan dan pengembangan metode ini adalah :

1. Algoritma *Cheapest Insertion Heuristics (CIH)* dapat diaplikasikan pada masalah lain, seperti : *Knapsack Problem*, *Shortest Path Problem*, *Vehicle Routing Problem*, dan Optimasi Penjadwalan.
2. Program untuk Algoritma *Cheapest Insertion Heuristics (CIH)* pada penelitian ini hanya terbatas pada *Symetric Traveling Salesman Problem (STSP)*, sehingga peneliti selanjutnya dapat mengembangkan program untuk menyelesaikan *Asymmetric Traveling Salesman Problem (ATSP)*.

3. Program untuk Algoritma *Cheapest Insertion Heuristics (CIH)* pada penelitian ini hanya terbatas untuk menangani permasalahan TSP dengan jumlah simpul maksimal adalah 20 simpul sehingga peneliti selanjutnya dapat mengembangkan program untuk menyelesaikan permasalahan TSP dengan jumlah simpul yang lebih besar dari 20 simpul.
4. Bagi peneliti selanjutnya dapat membandingkan Algoritma *Cheapest Insertion Heuristics (CIH)* dengan algoritma-algoritma lain untuk mengetahui algoritma mana yang lebih efektif untuk menyelesaikan masalah *Traveling Salesman Problem*.
5. Rancang bangun Algoritma *Cheapest Insertion Heuristics (CIH)* ini menggunakan bahasa pemrograman Java. Untuk itu disarankan bagi peneliti selanjutnya menggunakan bahasa pemrograman yang lain, seperti PHP, C++, Matlab dan lain-lain.

DAFTAR PUSTAKA

- Davendra, Donald. 2010. *Traveling Salesman Problem, Theory and Applications*. Kroasia : InTech.
- Desiyanti, Nur Eka. 2012. *Penggunaan Program Dinamik Dalam Penyelesaian Traveling Salesman Problem*. Yogyakarta: Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
- Diestel, Reinhard. 2005. *Graph Theory*. New York: Springer.
- Efendi, Rusdi dan Siti Maulinda. 2010. *Studi Perbandingan Algoritma Cheapest Insertion Heuristics dan Ant Colony System dalam Pemecahan Traveling Salesman Problem*. Yogyakarta : SNATI.
- Goetschalckx, Marc. 2011. *Supply chain engineering*. London : Springer.
- Gutin, Greogy dan Abraham P. Punnen. 2002. *Traveling Salesman Problem and Its Variations*. Dordrecht : Kluwer Academic Publisher.
- Jogiyanto. 2006. *Konsep Dasar Pemrograman Bahasa C*. Yogyakarta : Penerbit ANDI.
- Jungnickel, Dieter. 2000. *Graphs, Networks and Algorithms*. Second Edition. Jerman : Springer.
- Kang, Seungmo. 2011. *Transportation and Logistics Optimization – Complexity*. Korea.
- Kusrini dan Eko Jazi Istiyanto. 2007. *Penyelesaian Traveling Salesman Problem dengan Algoritma Cheapest Insertion Heuristics dan Basis Data*. diakses pada tanggal 9 Oktober 2012 dari <http://puslit.petra.ac.id/journals/informatics>.
- Mutakhirroh, Iing, dkk. 2007. *Pemanfaatan Metode Heuristik dalam Pencarian Jalur Terpendek dengan Algoritma Semut dan Algoritma Genetika*. Yogyakarta : SNATI 2007.
- Nugroho, M. Pasca. 2010. *Penerapan Algoritma Branch and Bound dalam Menentukan Rute Terpendek untuk Perjalanan Antarkota di Jawa Barat*. Bandung.
- Raharjo, Budi, Imam Heryanto dan Arif Haryono. 2012. *Mudah Belajar JAVA Revisi Kedua*. Bandung : Informatika Bandung.
- Rothlauf, Franz. 2011. *Design of Modern Heuristics*. London : Springer.

Tamam, M. Badru. 2012. *Aplikasi Algoritma Christofides' dan Software Logware pada Pendistribusian Produk PT. HEINZ ABC INDONESIA Cabang Solo*. Yogyakarta : UIN Sunan Kalijaga Yogyakarta.

Updy, Fery. *Modul Praktikum Pemrograman Berorientasi Objek (Menggunakan Java with Netbeans)*. 2011. Curug : SMKS YUPPEN TEK 2.

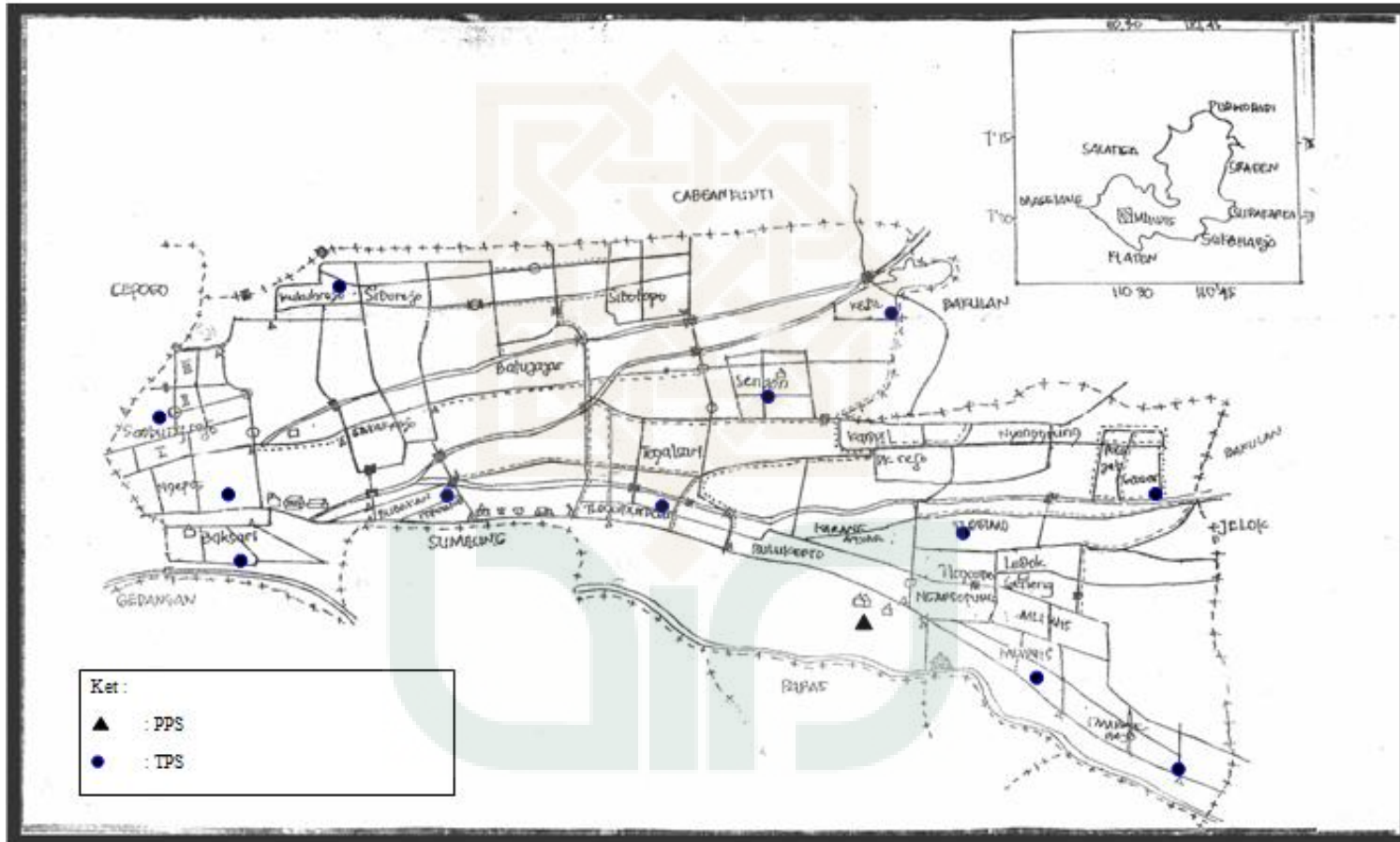
Wilson, Robin J. 1996. *Introduction to Graph Theory*. Fourth Edition. Malaysia : Longman.



a. Peta Kecamatan Cepogo



b. Ilustrasi lokasi PPS dan TPS di wilayah Desa Mliwis, Cepogo, Boyolali



Lampiran 2

SOURCE CODE UNTUK PROGRAM TSP DENGAN CIH

```
package TSPwithCIH;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

public class JFrameTSP extends javax.swing.JFrame {

    int inputX,hkom,nomor; //hkom = hasil kombinasi

    int Cij,Cik,Ckj,jmljarak=0;

    int as,si,tu,assip,tusip,sisip;

    /** Creates new form JFrameTSP */

    public JFrameTSP() {

        initComponents();

        setExtendedState(JFrameTSP.MAXIMIZED_BOTH);

        this.DeleteData();

        this.DeleteSisipan();

        CoPasTable copas = new CoPasTable(jTable1);

    }
```

```
*/  
  
@SuppressWarnings("unchecked")  
  
// <editor-fold defaultstate="collapsed" desc="Generated Code">  
private void initComponents() {  
  
    jLabel1 = new javax.swing.JLabel();  
    jTextField1 = new javax.swing.JTextField();  
    jButton1 = new javax.swing.JButton();  
    jScrollPane1 = new javax.swing.JScrollPane();  
    jTable1 = new javax.swing.JTable();  
    jScrollPane2 = new javax.swing.JScrollPane();  
    jTextArea1 = new javax.swing.JTextArea();  
    jLabel2 = new javax.swing.JLabel();  
    jLabel3 = new javax.swing.JLabel();  
    jScrollPane3 = new javax.swing.JScrollPane();  
    jTextArea2 = new javax.swing.JTextArea();  
    jLabel4 = new javax.swing.JLabel();  
  
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOS  
E);  
  
    setTitle("TSP dengan Algoritma CIH (Prototype)");  
  
    jLabel1.setText("Jumlah Kota yang dikunjungi");
```

```
        jTextField1.setDocument(new
FilterOnlyNumber().getFilterOnlyNumber());

        jTextField1.setDocument(new
DocumentLimit(2,DocumentLimit.ONLY_DIGIT));

        jTextField1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField1ActionPerformed(evt);
            }
        });

        jButton1.setText("Proses");

        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jTable1.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
                "Asal", "Tujuan", "Jarak (m)"
```

```
    }  
  ){  
    Class[] types = new Class [] {  
      java.lang.Integer.class, java.lang.Integer.class,  
      java.lang.Integer.class  
    };  
    boolean[] canEdit = new boolean [] {  
      false, false, true  
    };  
    public Class getColumnClass(int columnIndex) {  
      return types [columnIndex];  
    }  
    public boolean isCellEditable(int rowIndex, int columnIndex) {  
      return canEdit [columnIndex];  
    }  
  });  
  jTable1.setColumnSelectionAllowed(true);  
  jTable1.setName("jTable1"); // NOI18N  
  jTable1.getTableHeader().setReorderingAllowed(false);  
  jScrollPane1.setViewportViewView(jTable1);
```



```
jTable1.getColumnModel().getSelectionModel().setSelectionMode(javax.s  
wing.ListSelectionModel.MULTIPLE_INTERVAL_SELECTION);  
  
jTable1.getColumnModel().getColumn(0).setResizable(false);  
jTable1.getColumnModel().getColumn(1).setResizable(false);  
jTable1.getColumnModel().getColumn(2).setResizable(false);  
  
jTextArea1.setColumns(20);  
jTextArea1.setEditable(false);  
jTextArea1.setRows(5);  
jScrollPane2.setViewportView(jTextArea1);  
  
jLabel2.setText("Proses TSP dengan Algoritma CIH");  
  
jLabel3.setText("Hasil Akhir TSP");  
  
jScrollPane3.setVerticalScrollBarPolicy(javax.swing.ScrollPaneConstants.  
VERTICAL_SCROLLBAR_NEVER);
```

```
jTextArea2.setColumns(20);  
jTextArea2.setEditable(false);  
jTextArea2.setRows(5);  
jScrollPane3.setViewportView(jTextArea2);
```

```

jLabel4.setText("Jumlah Kombinasi : ");

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());

    getContentPane().setLayout(layout);

    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel1)

                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

```

```
.addComponent(jButton1)

.addComponent(jLabel4)

.addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 203,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.TRAILING)

.addComponent(jScrollPane3,
javax.swing.GroupLayout.DEFAULT_SIZE, 653, Short.MAX_VALUE)

.addComponent(jScrollPane2,
javax.swing.GroupLayout.DEFAULT_SIZE, 653, Short.MAX_VALUE)

.addComponent(jLabel3,
javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 260,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addContainerGap()

);
```

```
layout.setVerticalGroup(
```

```
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
```

```
G)
```

```
    .addGroup(layout.createSequentialGroup()
```

```
        .addContainerGap()
```

```
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASILINE)
```

```
        .addComponent(jLabel1)
```

```
        .addComponent(jTextField1,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(jLabel2))
```

```
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
```

```
D)
```

```
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.TRAILING)
```

```
        .addComponent(jScrollPane1,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE, 403, Short.MAX_VALUE)
```

```
.addComponent(jScrollPane2,  
javax.swing.GroupLayout.DEFAULT_SIZE, 403, Short.MAX_VALUE))  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE  
D)  
  
.addComponent(jLabel3)  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment  
.LEADING)  
    .addGroup(layout.createSequentialGroup()  
        .addGap(17, 17, 17)  
        .addComponent(jLabel4)  
        .addGap(10, 10, 10)  
        .addComponent(jButton1))  
    .addGroup(layout.createSequentialGroup()  
        .addGap(11, 11, 11)  
        .addComponent(jScrollPane3,  
javax.swing.GroupLayout.PREFERRED_SIZE, 77,  
javax.swing.GroupLayout.PREFERRED_SIZE)))  
    .addContainerGap()  
);  
  
pack();
```

```
}// </editor-fold>
```

```
private void jTextField1ActionPerformed(java.awt.event.ActionEvent  
evt) {  
    // TODO add your handling code here:  
  
    this.DeleteData();  
  
    jTextArea1.setText("");  
    jTextArea2.setText("");  
  
    inputX = Integer.valueOf(jTextField1.getText());  
  
    //Kombinasi  
    int n=inputX;  
    int k=2;  
  
    if (n<k){  
        JOptionPane.showMessageDialog(null,"Data  
Salah","Error",JOptionPane.ERROR_MESSAGE);  
    } else if (n==k){  
        hkom=1;  
        jLabel4.setText("Jumlah Kombinasi : "+hkom);  
    } else if (n>k) {  
  
        long fakn=1;
```

```

long fakk=1;

long faknmink=1;

for (int N=n;N>=1;N--){

    fakn=fakn*N;

}

for (int K=k;K>=1;K--){

    fakk*=K;

}

for (int NminK=n-k;NminK>=1;NminK--){

    faknmink*=NminK;

}

hkom=(int) (fakn/(faknmink*fakk));

jLabel4.setText("Jumlah Kombinasi : "+hkom);

}

//

if (hkom<=0){

    JOptionPane.showMessageDialog(null,"Hasil Kombinasi Kurang
dari atau sama dengan 0","Error",JOptionPane.ERROR_MESSAGE);

}else{

    int i=2;

    for (int x=1;x<=inputX-1;x++){

        for (int y=i;y<=inputX;y++){

            this.InsertData(x,y,0);

```

```
        }i+=1;
    }
}
this.SelectData();
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    for (int row=0;row<=hkom-1;row++){
        int asal=Integer.parseInt(jTable1.getValueAt(row, 0).toString());
        int tujuan=Integer.parseInt(jTable1.getValueAt(row, 1).toString());
        int jarak=Integer.parseInt(jTable1.getValueAt(row, 2).toString());
        this.UpdateData(asal,tujuan,jarak);
    }

    this.InsertHasil(1, inputX,1);
    this.InsertHasil(inputX, 1,2);
    jTextArea1.setText("");

    for (int loop=1;loop<=inputX-2;loop++){
        this.DeleteSisipan();
        this.ProsesCIH();
    }
}
```



```
        this.JalurTerpilih();
    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new JFrameTSP().setVisible(true);
            }
        });
    }

    private void ProsesCIH(){
        int jarak=0;
        String str="";
        int numrow = 0, numrow2 = 0;

        Koneksi conn = new Koneksi();

        String query = "SELECT f_asal_h,f_tujuan_h FROM hasil ORDER
BY f_asal_h ASC";

        String query2 = "SELECT DISTINCT f_asal FROM data WHERE
f_asal NOT IN (SELECT f_asal_h FROM hasil ORDER BY f_asal_h
ASC)";
```

```
String query3 = "SELECT f_jmljarak,f_asal_s,f_sisipan,f_tujuan_s
FROM sisipan WHERE f_jmljarak = (SELECT MIN(f_jmljarak) FROM
sisipan)";
```

```
ResultSet q1 = null,q2 = null,q3 = null,q4 = null,q5 = null;
```

```
try {
    //Proses mengambil sisipan
    q2 = conn.createStatement().executeQuery(query2);
    if (!q2.next()) {
        jarak = 0;
        str = "";

        //Proses dua kota
        q1 = conn.createStatement().executeQuery(query);
        while (q1.next()) {
            jarak = jarak + C(q1.getInt("f_asal_h"),
q1.getInt("f_tujuan_h"));

            str=q1.getInt("f_asal_h")+ " ~~> "+q1.getInt("f_tujuan_h")+ " =
";

            jTextArea1.append(str+jarak+"\n");
        }
        q1.close();
```

```
} else {  
    do {  
        numrow=q2.getRow();//jumlah baris  
    } while (q2.next());  
  
    int[] data = new int[numrow];  
    int loop=0;  
  
    try{  
        q2 = conn.createStatement().executeQuery(query2);  
        while (q2.next()){  
            data[loop]=q2.getInt("f_asal");  
            loop++;  
        }  
    }catch (SQLException e){}  
  
    q1 = conn.createStatement().executeQuery(query);  
    while (q1.next()) {  
        for (int l=0;l<numrow;l++){  
            //input ke tabel sisipan  
            as = q1.getInt("f_asal_h");  
            si = data[l];  
            tu = q1.getInt("f_tujuan_h");
```

```

        Cij=C(as, tu); Cik=C(as, si); Ckj=C(si, tu);

        if (Cij==0){ Cij=C(tu, as); }

        if (Cik==0){ Cik=C(si, as); }

        if (Ckj==0){ Ckj=C(tu, si); }

        jmljarak=Math.abs(Cik+Ckj-Cij);

        InsertSisipan(as, si, tu, jmljarak);

        JTextArea1.append(as+" ~> "+si+" ~> "+tu+" =
"+jmljarak+" \n");
    }
}

JTextArea1.append("=====\n");

q3 = conn.createStatement().executeQuery(query3);//Mencari jarak
minimum dari tabel sisipan

if (q3.next()) {

    assip=q3.getInt("f_asal_s");
    sisip=q3.getInt("f_sisipan");
    tusip=q3.getInt("f_tujuan_s");

    JTextArea1.append("Jalur dengan jarak terpendek:\n");

    JTextArea1.append(assip+" ~> "+sisip+" ~> "+tusip+" =
"+q3.getInt("f_jmljarak")+"\n");

JTextArea1.append("=====\n");

```

```
q4 = conn.createStatement().executeQuery("SELECT * FROM
hasil");

do {
    numrow=q4.getRow();
} while (q4.next());

int[] asal = new int[numrow], nmr = new int[numrow];
loop=0;

try{
    q4 = conn.createStatement().executeQuery("SELECT * FROM
hasil ORDER BY f_nomor DESC");
    while (q4.next()){
        asal[loop]=q4.getInt("f_asal_h");
        nmr[loop]=q4.getInt("f_nomor");
        loop++;
    }
} catch (SQLException e){ }

q5 = conn.createStatement().executeQuery("SELECT f_nomor
FROM hasil WHERE f_nomor > '1'");

do {
    numrow2=q5.getRow();
```

```

    } while (q5.next());

    int[] nomer = new int[numrow2], assal = new int[numrow2];
    loop=0;
    try {
        q5 = conn.createStatement().executeQuery("SELECT
f_asal_h,f_nomor FROM hasil WHERE f_nomor > '1' ORDER BY f_nomor
DESC");
        while (q5.next()){
            assal[loop]=q5.getInt("f_asal_h");
            nomer[loop]=q5.getInt("f_nomor");
            loop++;
        }
    } catch (SQLException e){ }

    for (int i=0;i<numrow;i++){
        //MULAI INPUTAN KEDALAM TABEL HASIL
        if (assip==asal[i]){
            if (assip==1){
                conn.createStatement().executeUpdate("UPDATE hasil
SET f_tujuan_h='"+sisip+"' WHERE f_asal_h=1");
                for (int j=0;j<numrow2;j++){

```

```

        conn.createStatement().executeUpdate("UPDATE hasil
SET f_nomor=(f_nomor+1) WHERE f_asal_h = '"+assal[j]+" ");

    }

    this.InsertHasil(sisip, tusip, 2);

} else {

    conn.createStatement().executeUpdate("UPDATE hasil
SET f_tujuan_h='"+sisip+"' WHERE f_asal_h='"+assip+"'");

    conn.createStatement().executeUpdate("UPDATE hasil
SET f_nomor=(f_nomor+2) WHERE f_nomor > '"+nmr[i]+'");

    this.InsertHasil(sisip, tusip, (nmr[i]+1));

}

}

ResultSet qhasil =
conn.createStatement().executeQuery("SELECT * FROM hasil ORDER BY
f_nomor ASC");

jTextArea1.append("Jalur yang dilalui : \n");

while (qhasil.next()){

    jTextArea1.append(qhasil.getInt("f_asal_h")+" ~~>
"+qhasil.getInt("f_tujuan_h")+" , ");

}

jTextArea1.append("\n\n");

```

```
    }  
    }  
} catch (SQLException ex){  
    JOptionPane.showMessageDialog(null,""+ex.getMessage());  
}  
}  
  
private void JalurTerpilih(){  
    Koneksi conn = new Koneksi();  
    try {  
        ResultSet qhasil = conn.createStatement().executeQuery("SELECT *  
FROM hasil ORDER BY f_nomor ASC");  
        int tot=0;  
        while (qhasil.next()){  
            as=qhasil.getInt("f_asal_h");  
            tu=qhasil.getInt("f_tujuan_h");  
            Cij= C(as,tu);  
            if (Cij==0){ Cij=C(tu, as); }  
  
            tot += Cij;  
  
            if(qhasil.getInt("f_nomor")==1){
```



```

jTextArea2.append(qhasil.getInt("f_asal_h")+","+qhasil.getInt("f_tujuan_h"
));

        } else {

jTextArea2.append("~~>"+qhasil.getInt("f_asal_h")+","+qhasil.getInt("f_tuj
uan_h"));

        }
    }

    float kilo= (float) tot/1000;

    jTextArea2.append("\nTotal Jarak : "+tot+" meter = "+kilo+"
kilometer" );

    } catch (SQLException ex) {}

}

private void InsertHasil(int asal,int tujuan,int nomor){

    Koneksi conn=new Koneksi();

    String qinsert="INSERT INTO hasil (f_asal_h,f_tujuan_h,f_nomor)
VALUES (""+asal+"",""+tujuan+"",""+nomor+"");

    try{

        conn.createStatement().executeUpdate(qinsert);

    }catch(SQLException ex){

```

```
JOptionPane.showMessageDialog(this, "Error: Query - gagal
menyimpan");
    }
}

private void InsertSisipan(int asal,int sisipan,int tujuan,int jarak){
    Koneksi conn=new Koneksi();
    String query="INSERT INTO sisipan
(f_asal_s,f_sisipan,f_tujuan_s,f_jmljarak) VALUES
(""+asal+"",""+sisipan+"",""+tujuan+"",""+jarak+"");

    try{
        conn.createStatement().executeUpdate(query);
    }catch(SQLException ex){
        JOptionPane.showMessageDialog(this, "Error: Query - tidak dapat
diedit");
    }
}

private void DeleteSisipan(){
    Koneksi conn = new Koneksi();
    String query = "TRUNCATE TABLE sisipan";
```

```
try {  
    conn.createStatement().executeUpdate(query);  
} catch (SQLException ex) {  
    JOptionPane.showMessageDialog(this, "Error: Query - tidak bisa  
menghapus");  
}  
}  
  
//function memanggil jarak dari database data  
public int C(int asal,int tujuan){  
    Koneksi conn=new Koneksi();  
    ResultSet res=null;  
    int ij=0;  
    String query="SELECT f_jarak FROM data WHERE f_asal="+asal+"  
AND f_tujuan="+tujuan;  
  
try{  
    res=conn.createStatement().executeQuery(query);  
    while(res.next()){  
        ij=Integer.parseInt(res.getString("f_jarak"));  
    }  
}catch(SQLException ex){  
    JOptionPane.showMessageDialog(this, "Error: Query Error");  
}
```

```
    }return ij;
}

private void InsertData(int asal,int tujuan,int jarak){
    Koneksi conn=new Koneksi();

    String query="INSERT INTO data (f_asal,f_tujuan,f_jarak) VALUES
("+asal+",","+tujuan+",","+jarak+"");

    try{
        conn.createStatement().executeUpdate(query);
    }catch(SQLException ex){
        JOptionPane.showMessageDialog(this, "Error: Query - data tidak
dapat disimpan");
    }
}

private void UpdateData(int asal,int tujuan,int jarak){
    Koneksi conn=new Koneksi();

    String query="UPDATE data SET f_jarak="+jarak+" WHERE
f_asal="+asal+" AND f_tujuan="+tujuan+"";

    try{
        conn.createStatement().executeUpdate(query);
```

```
    }catch(SQLException ex){  
        JOptionPane.showMessageDialog(this, "Error: Query - data tidak  
dapat diedit");  
    }  
}
```

```
private void DeleteData(){  
    Koneksi conn = new Koneksi();  
    String qdata = "TRUNCATE TABLE data";//menghapus semua isi  
tabel data  
    String qhasil = "TRUNCATE TABLE hasil";  
  
    try {  
        conn.createStatement().executeUpdate(qdata);  
        conn.createStatement().executeUpdate(qhasil);  
    } catch (SQLException ex) {  
        JOptionPane.showMessageDialog(this,"Error: Query - data tidak  
dapat dihapus");  
    }  
}
```

```
private void SelectData(){  
    Koneksi conn=new Koneksi();
```

```

ResultSet data=null;

int JmlBrs=0;

String query="SELECT * FROM data ORDER BY f_asal,f_tujuan
ASC";

try{
    data=conn.createStatement().executeQuery(query);
    while (data.next()){
        JmlBrs++;
    }
}catch(SQLException ex){
    JOptionPane.showMessageDialog(null,"Error pada jumlah baris :
"+ex.getMessage());
}

String isi[][]= new String[JmlBrs][3];

// masukkan data dari ResultStatement kedalam matriks

int i=0;//just for loop

try {
    //eksekusi kembali query

    data = conn.createStatement().executeQuery(query);

    while (data.next()) {

```

```
// ini sintaks untk mengambil isinya, "f_asal,f_tujuan,f_jarak"
```

adalah nama kolom dalam DB

```
isi[i][0] = "" + data.getString("f_asal");
isi[i][1] = "" + data.getString("f_tujuan");
isi[i][2] = "" + data.getString("f_jarak");
i++;
}
} catch (SQLException ex) {
}
}
```

```
String NamaKolom[] ={"Asal","Tujuan","Jarak (m)"};//nama kolom
```

yang nantinya akan ada pada tabel

```
DefaultTableModel model= new DefaultTableModel(isi,NamaKolom)
```

```
{
Class[] types = new Class [] {
    java.lang.Integer.class, java.lang.Integer.class,
    java.lang.Integer.class
```

```
};
```

```
boolean[] canEdit = new boolean [] {
```

```
    false, false, true
```

```
};
```

```
@Override
```

```
public Class getColumnClass(int columnIndex) {  
    return types [columnIndex];  
}  
  
@Override  
public boolean isCellEditable(int rowIndex, int columnIndex) {  
    return canEdit [columnIndex];  
}  
  
};  
  
//selanjutnya adalah melakukan pengesetan pada tabel  
jTable1.setModel(model);  
  
//done!!!  
}  
  
// Variables declaration - do not modify  
private javax.swing.JButton jButton1;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JScrollPane jScrollPane2;
```



```
private javax.swing.JScrollPane jScrollPane3;  
  
private javax.swing.JTable jTable1;  
  
private javax.swing.JTextArea jTextArea1;  
private javax.swing.JTextArea jTextArea2;  
private javax.swing.JTextField jTextField1;  
  
// End of variables declaration  
}
```



CURRICUUM VITAE

A. IDENTITAS DIRI

Nama : Anis Nur Rohmah
 Tempat, Tanggal Lahir : Boyolali, 11 Desember 1991
 Jenis Kelamin : Perempuan
 Agama : Islam
 Kebangsaan : Indonesia
 Alamat : Tlogo Imo 07/01 Mliwis, Cepogo, Boyolali 57362
 Nama Ayah : Sofwan
 Nama Ibu : Nuryani
 Alamat Orang Tua : Tlogo Imo 07/01 Mliwis, Cepogo, Boyolali 57362

B. RIWAYAT PENDIDIKAN

No	JENJANG PENDIDIKAN	NAMA SEKOLAH	TAHUN LULUS
1	SD / MI	MI Gunung Wijil	2003
2	SMP / MTS	MTs N Boyolali	2006
3	SMA / MA	MAPK Surakarta	2009
4	PT / PTAI	UIN Sunan Kalijaga Yogyakarta	2013

Demikian daftar riwayat hidup ini saya buat dengan sebenar-benarnya dan dapat dipertanggung jawabkan.