

**IMPLEMENTASI ALGORITMA *BRANCH AND BOUND*
UNTUK OPTIMASI RUTE PENGANGKUTAN SAMPAH
KOTA YOGYAYAKARTA**

Skripsi
Untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-1

Program Studi Matematika



diajukan oleh

M. Shofi Alkheiroda'i

08610034

Kepada

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UIN SUNAN KALIJAGA
YOGYAKARTA**

2013



Universitas Islam Negeri Sunan Kalijaga

FM-UINSK-BM-05-07/R0

PENGESAHAN SKRIPSI/TUGAS AKHIR


Nomor : UIN.02/D.ST/PP.01.1/1648/2013

Skripsi/Tugas Akhir dengan judul : Implementasi Algoritma *Branch and Bound* untuk Optimasi Rute Pengangkutan Sampah Kota Yogyakarta

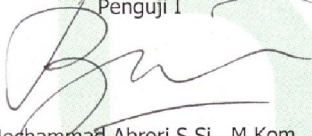
Yang dipersiapkan dan disusun oleh :
Nama : M. Shofi Alkhoiroda'i
NIM : 08610034
Telah dimunaqasyahkan pada : 31 Mei 2013
Nilai Munaqasyah : A
Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

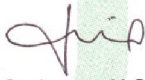
Ketua Sidang


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

Penguji I

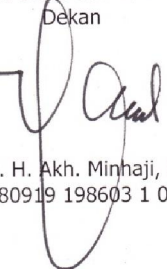

Mochammad Abrori, S.Si., M.Kom
NIP.19720423 199903 1 003

Penguji II


Sugiyanto, M.Si
NIP.19800505 200801 1 028

Yogyakarta, 05 Juni 2013
UIN Sunan Kalijaga
Fakultas Sains dan Teknologi
Dekan




Prof. Drs. 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 : M.Shofi Alkhoiroda'i
NIM : 08610034
Judul Skripsi : Implementasi Algoritma Branch and Bound untuk Optimasi Rute Pengangkutan Sampah Kota Yogyakarta.

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

Wassalamu'alaikum wr. wb.

Yogyakarta, Mei 2013

Pembimbing

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

SURAT PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini :

Nama : M.Shofi Alkhoiroda'i

NIM : 08610034

Program Studi : Matematika

Fakultas : Sains dan Teknologi

Menyatakan dengan sesungguhnya bahwa skripsi ini merupakan hasil pekerjaan penulis sendiri dan sepanjang pengetahuan penulis tidak berisi materi yang dipublikasikan atau ditulis orang lain, dan atau telah digunakan sebagai persyaratan penyelesaian Tugas Akhir di Perguruan Tinggi lain, kecuali bagian tertentu yang penulis ambil sebagai bahan acuan. Apabila terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab penulis.

Yogyakarta, Mei 2013

Yang menyatakan



M.Shofi Alkhoiroda'i

NIM. 08610034

MOTTO

قُلْ إِنَّ صَلَاتِي وَنُسُكِي وَمَحْيَايَ وَمَمَاتِي لِلَّهِ رَبِّ الْعَالَمِينَ ﴿١٦٢﴾

“Katakanlah Sesungguhnya sholatku, ibadahku, hidupku dan matiku hanyalah untuk Allah Tuhan semesta alam”

(QS:Al- an'am:162)

“Manusia dapat dihancurkan, manusia dapat dimatikan tetapi manusia tidak dapat dikalahkan selama manusia itu masih setia kepada dirinya sendiri atau ber Setia-Hati kepada dirinya sendiri”

[Falsafah PSHT]

HALAMAN PERSEMBAHAN

Alhamdulillahillobbi 'alamiin..

Karya kecil ini penulis persembahkan kepada:

Ibu yang tanpa lelah memberikan cahaya cinta tanpa batas,

Bapak yang mendidik dan mendorong kebijaksanaan,

kakak penulis dan semua guru yang memberikan bekal dunia akhirat.

Serta kepada almamater tercinta program studi Matematika

fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta.

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan rahmat, taufiq, dan hidayah-Nya, serta nafas kepada penulis sampai detik ini, sehingga penulis mampu menyelesaikan penulisan skripsi ini. Sholawat dan salam semoga senantiasa terlimpahkan kepada Nabi Muhammad SAW yang telah membawa umat manusia menuju zaman dengan kemajuan ilmu pengetahuan dan Islam.

Penulis menyadari bahwa proses penulisan skripsi ini tidak terlepas dari dukungan, kerjasama dan bimbingan dari berbagai pihak. Oleh karena itu, iringan doa dan terimakasih penulis sampaikan dengan tulus kepada:

1. Prof. Drs. H. Akh. Minhaji, M.A., Ph.D selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Yogyakarta.
2. Muchammad Abrori, M.Kom selaku Ketua Prodi Matematika Fakultas Sains dan Teknologi.
3. Noor Saif Muhammad M., S.Si., M.Sc selaku dosen pembimbing karena atas ilmu, bimbingan, arahan dan motivasinya penulisan skripsi ini dapat terselesaikan.
4. Semua guru, dosen dan ustadz atas arahan dan ilmu yang telah diberikan, serta bimbingan kepada penulis untuk menjadi manusia yang lebih baik.
5. Rasa Setya Iman., S.E selaku Kepala Sub Bidang Pengangkutan BLH kota Yogyakarta yang telah mempermudah serta memandu jalannya penelitian di BLH kota Yogyakarta.

6. Kedua orangtua tercinta, Bapak Mujahidin dan Ibu Umi Hidayatul Mustafit, yang selalu memberikan bimbingan, dukungan baik moral, material dan do'a yang tulus agar selalu diberikan yang terbaik oleh Allah SWT.
7. Kanda M. Farojihut Tawakal, S.Pd.I yang selalu berbagi semangat dan nasihat akan lika-liku perjalanan hidup.
8. Fanny Brilliyanti, S.Si sang motivator pribadi jazakillahu khoiron, semoga Allah SWT selalu memberikan keistiqomahan kepada kita hingga akhir nanti
9. Teman seperjuangan Matematika 08. Terimakasih untuk semua kerja sama dan bantuannya selama kuliah.
10. Seluruh saudara kadang pendekar PSHT (Persaudaraan Setia Hati Terate) Komisariat UIN Sunan Kalijaga Yogyakarta dan Cabang Sleman.
11. Teman-teman pengurus takmir Masjid Uswatun Hasanah, Gus Fauz, Kang Didik, Kang Rudi, Rosyid, Arik, Habibi, Wildan dan Musyadad, yang senantiasa berbagi suka duka sebagai penjaga Masjid.
12. Akhmad Kholis, rekan takmir penjaga Masjid Nurul Istiqomah, terimakasih untuk semua kerjasama kita semoga selalu diberikan yang terbaik oleh Allah SWT.

Semoga skripsi ini dapat bermanfaat bagi penulis dan semua pihak yang membutuhkan.

Yogyakarta, 31 Mei 2013

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN PERSETUJUAN	iii
SURAT PERNYATAAN KEASLIAN SKRIPSI	iv
HALAMAN MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
ABSTRAK	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Tinjauan Pustaka	4
1.7 Sistematika Penelitian	7
1.8 Metode Penelitian	8
BAB II LANDASAN TEORI	9
2.1 Matriks Diagonal	9
2.2 Teori Graf	9
2.3 Terminologi Graf	11
2.4 Keterhubungan (<i>Connectivity</i>)	13
2.5 Graf Lengkap	16
2.6 Graf Berbobot (<i>Weighted Graph</i>)	18

2.7 Pohon (<i>Tree</i>)	18
2.8 Graf Hamilton	21
2.9 Representasi Graf	21
2.10 <i>Travelling Salesman Problem</i>	22
2.11 Algoritma <i>Branch and Bound</i>	24
2.12 MATLAB	29
2.13 GUI Pada MATLAB	30
2.14 <i>Statement Control</i> Pada MATLAB	30
2.15 Reduksi Matriks Pada MATLAB	36
BAB III PEMBAHASAN	39
3.1 Implementasi Algoritma <i>Branch and Bound</i> untuk Optimasi Rute Pengangkutan Sampah	39
3.1.1 Pemetaan Pengangkutan Sampah Kota Yogyakarta.....	39
3.1.2 Implementasi Algoritma <i>Branch and Bound</i> untuk Optimasi Rute Pengangkutan Sampah Kota Yogyakarta	44
3.2 Rancang Bangun Algoritma Branch and Bound dalam Pencarian Rute Terpendek Pengangkutan Sampah Kota Yogyakarta	119
3.2.1 Spesifikasi Kebutuhan	119
3.2.1.1 Perangkat Keras	119
3.2.1.2 Perangkat Lunak	120
3.2.2 Membuat Aplikasi GUI dengan MATLAB	120
3.2.3 Uji Coba Program	131
BAB IV PENUTUP	138
4.1 Kesimpulan	138
4.2 Saran	139
DAFTAR PUSTAKA	140
LAMPIRAN	141

DAFTAR GAMBAR

- Gambar 2.1 Graf G
- Gambar 2.2 (a) Sisi berlipat. (b) *Loop*. (c) Graf sederhana.
- Gambar 2.3 Graf G
- Gambar 2.4 Graf G
- Gambar 2.5 Graf lengkap K_4
- Gambar 2.6 Graf Berbobot
- Gambar 2.7 Pohon
- Gambar 2.8 Graf Hamilton
- Gambar 2.9 Graf ABCDE
- Gambar 2.10 Konversi graf menjadi pohon pencarian untuk masalah TSP
- Gambar 2.11 *Flowchart* algoritma *Branch and Bound*.
- Gambar 2.12 Pembentukan GUI secara sederhana
- Gambar 2.13 Contoh tampilan fig-file
- Gambar 2.14 Contoh tampilan M-file
- Gambar 2.15 Contoh tampilan fungsi plot
- Gambar 3.1 Letak transfer depo dan TPA
- Gambar 3.2 Pohon pencarian iterasi 1
- Gambar 3.3 Pohon pencarian simpul 4 pada iterasi 2
- Gambar 3.4 Pohon pencarian simpul 13 pada iterasi 2
- Gambar 3.5 Pohon pencarian iterasi 2

- Gambar 3.6 Pohon pencarian iterasi 3
- Gambar 3.7 Pohon pencarian iterasi 4
- Gambar 3.8 Pohon pencarian iterasi 5
- Gambar 3.9 Pohon pencarian iterasi 6
- Gambar 3.10 Pohon pencarian iterasi 7
- Gambar 3.11 Pohon pencarian iterasi 8
- Gambar 3.12 Pohon pencarian iterasi 9
- Gambar 3.13 Pohon pencarian iterasi 10
- Gambar 3.14 Pohon pencarian iterasi 11
- Gambar 3.15 Pohon pencarian iterasi 12
- Gambar 3.16 Pohon pencarian iterasi 13
- Gambar 3.17 Pohon pencarian *Branch and Bound*
- Gambar 3.18 Desain *figure* aplikasi
- Gambar 3.19 Proses Running Program GUI
- Gambar 3.20 Form Awal Program
- Gambar 3.21 Form help
- Gambar 3.22 Form Input Data
- Gambar 3.23 Proses Input Data
- Gambar 3.24 Proses Perhitungan Program
- Gambar 3.25 Form Tampilan *Output*.

DAFTAR TABEL

- Tabel 2.1. Tabel nilai IPK rata-rata mahasiswa matematika
- Tabel 3.1. Data transfer depo kota Yogyakarta
- Tabel 3.2. Data TPA
- Tabel 3.3. Jarak beberapa transfer depo dan TPA (km).
- Tabel 3.4. Matriks jarak beberapa transfer depo dan TPA (km)
- Tabel 3.5. Spesifikasi perangkat keras
- Tabel 3.6. Spesifikasi perangkat lunak
- Tabel 3.7. Rekapitulasi Hasil Perhitungan

**IMPLEMENTASI ALGORITMA *BRANCH AND BOUND*
UNTUK OPTIMASI RUTE PENGANGKUTAN SAMPAH
KOTA YOGYAKARTA**

Oleh :
M. Shofi Alkhoiroda'i
08610034

ABSTRAK

Peningkatan jumlah penduduk di Indonesia, khususnya di kota Yogyakarta, diikuti oleh peningkatan jumlah sampah yang dihasilkan. Hal ini harus ditunjang oleh sarana-prasarana pengelolaan sampah yang mendukung supaya tidak terjadi penumpukan sampah. Penyebab kondisi penumpukan sampah disebabkan oleh minimnya jumlah pekerja dan jumlah sarana-prasarana pengangkut (seperti truk dan pick-up) oleh Badan Lingkungan Hidup (BLH) kota Yogyakarta. Tujuan dari penelitian ini adalah mengoptimalkan proses pengangkutan dengan satu kali putaran rute dan membuat rute pengangkutan yang efektif, sehingga mampu meningkatkan efisiensi biaya bahan bakar dan waktu.

Permasalahan rute pengangkutan sampah secara abstrak dapat digambarkan dengan suatu graf yang merupakan masalah optimasi dalam pencarian rute terpendek *Travelling Salesman Problem* (TSP). Pemecahan permasalahan tersebut adalah dengan merepresentasikan peta pengangkutan sampah ke dalam bentuk graf lengkap berbobot, selanjutnya permasalahan diselesaikan menggunakan algoritma *Branch and Bound*. Perhitungan dilakukan baik secara manual maupun dengan program aplikasi sederhana yang dibuat dengan bantuan *software* MATLAB versi 7.9.0 (R2009b).

Berdasarkan perhitungan menggunakan algoritma *Branch and Bound* untuk optimasi rute pengangkutan sampah di kota Yogyakarta menghasilkan solusi rute pengangkutan: TPA - Depo Jl.Kemasan - Depo Pamukti - Depo Nitikan – Depo Wirosaban – Depo Jl.Brigjen Katamso – Depo Dukuh – Depo Pasar Ngasem – Depo TMP Ngabean – Depo Pringgokusuman – Depo Utoroloyo – Depo Sagan – Depo Mandalakrida – Depo Jl.Veteran – TPA dengan total jarak kunjungan 28,65 Km.

Kata kunci : Algoritma *Branch and Bound*, MATLAB, optimasi rute, *Travelling Salesman Problem* (TSP).

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pengelolaan sampah merupakan salah satu permasalahan yang terjadi di kota Yogyakarta. Permasalahan ini terjadi hampir di seluruh wilayah Indonesia baik di kota besar maupun di kota kecil. Jumlah timbunan sampah yang dihasilkan meningkat dari tahun ke tahun sejalan dengan meningkatnya jumlah penduduk dan gaya hidup masyarakat. Semakin banyak jumlah penduduk, sampah buangan dari kegiatan rumah tangga, pasar dan industri semakin banyak. Sedangkan tingginya tingkat gaya hidup masyarakat modern akan berpengaruh terhadap tingkat konsumsi masyarakat dan berdampak pada sampah yang dihasilkan. Jumlah timbunan yang besar dan tanpa penanganan yang baik, sampah akan menimbulkan berbagai masalah sosial dan lingkungan yang sangat berat. Pengangkutan sampah dari TPS ke TPA haruslah optimal, sehingga pemilihan rute terpendek untuk mengoptimalkan pengangkutan sampah sangat diperlukan untuk menangani permasalahan ini.

Pentingnya pemilihan rute pengangkutan sampah yang tepat diharapkan dapat meminimalisir permasalahan yang ada. Misalnya pencemaran air limbah semakin berkurang, prosentase sampah yang dapat terangkut lebih meningkat dari sebelumnya, efisiensi dalam penggunaan BBM dan biaya angkut, serta kecepatan dalam hal pelayanan. Pencemaran air limbah dapat terjadi karena kurang

optimalnya pengangkutan sampah sehingga terjadi penginapan timbunan dalam skala besar yang dapat mempengaruhi air bersih di sekitar TPS.

Rute pengangkutan sampah dari beberapa TPS ke TPA secara abstrak dapat digambarkan dengan suatu graf, dimana TPS dan TPA digambarkan sebagai simpul (*vertex*). Sedangkan jalan yang menghubungkan antara beberapa TPS dan TPA digambarkan sebagai sisi (*edge*). Dalam kajian teori graf salah satu masalah optimasi yang sering dijumpai dalam pencarian rute terpendek adalah *Travelling Salesman Problem* (TSP). Terdapat beberapa Algoritma yang dapat menyelesaikan *Travelling Salesman Problem* seperti Algoritma *Greedy*, *Dijkstra*, *Brute Force* dan lain-lain. Pada dasarnya dalam jumlah simpul yang banyak, belum terbukti masing-masing Algoritma tersebut menghasilkan penyelesaian yang paling optimal. *Algoritma Branch and Bound* merupakan salah satu Algoritma untuk pencarian solusi dalam masalah optimasi. Konsep dari Algoritma B&B yaitu menggunakan taksiran nilai batas (*estimated bound*) dan menggunakan strategi pencarian nilai batas terkecil (*least cost search*) untuk mempercepat menemukan simpul solusi. Algoritma ini diharapkan mampu memberikan solusi yang lebih optimal.

Kendala di lapangan menunjukkan minimnya jumlah pekerja dan jumlah sarana-prasarana pengangkut (seperti truk dan pick-up) menjadi penyebab lambatnya pengangkutan sampah. Permasalahan ini menjadi sangat menarik dan penting untuk dilakukan kajian lebih lanjut dalam rangka meningkatkan efisiensi biaya dan waktu, sehingga diperlukan penelitian dan perhitungan pencarian rute terpendek dengan metode *Algoritma Branch and Bound*.

1.2. Rumusan Masalah

Rumusan masalah penelitian ini adalah

1. Bagaimana penerapan *Algoritma Branch and Bound* untuk optimasi rute pengangkutan sampah kota Yogyakarta ?
2. Bagaimana rancang bangun *Algoritma Branch and Bound* dalam pencarian rute terpendek pengangkutan sampah kota Yogyakarta ?

1.3. Batasan Masalah

Dalam pengerjaan skripsi ini, ditetapkan beberapa batasan masalah sebagai berikut:

1. Jenis TPS yang dipilih adalah jenis transfer depo. Transfer depo adalah tempat pemindahan sampah dari alat pengumpul ke alat pengangkut.
2. Posisi dan jumlah transfer depo dan TPA yang terdapat dalam rute pengangkutan sampah telah diketahui berdasarkan data dari Badan Lingkungan Hidup kota Yogyakarta.
3. Rute terpendek yang dimaksud adalah jarak dari TPA menuju beberapa transfer depo dan kembali lagi ke TPA.
4. Faktor kemacetan jalan diabaikan.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai dalam skripsi ini adalah

1. Menerapkan *Algoritma Branch and Bound* untuk optimasi rute pengangkutan sampah kota Yogyakarta.

2. Membuat rancang bangun *Algoritma Branch and Bound* dalam pencarian rute terpendek pengangkutan sampah kota Yogyakarta.

1.5. Manfaat Penelitian

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

1. Menambah pemahaman aplikasi teori graf yaitu penerapan *Algoritma Branch And Bound* pada optimasi rute pengangkutan sampah di kota Yogyakarta.
2. Memberikan rekomendasi rute yang optimum kepada Badan Lingkungan Hidup seksi pembersihan dan pengangkutan sampah kota Yogyakarta, sehingga diharapkan pada masa mendatang pengangkutan sampah lebih optimal.

1.6. Tinjauan Pustaka

Tinjauan pustaka skripsi ini terdiri dari beberapa buku dan penelitian sebelumnya, antara lain buku karangan Suyanto yang berjudul "*Algoritma Optimasi Deterministik atau Probabilistik*" yang diterbitkan pada tahun 2010 dan buku karangan Eko Budi Purwanto berjudul "*Perancangan dan Analisis Algoritma*" yang diterbitkan pada tahun 2008. Dalam buku tersebut, dijelaskan bahwa algoritma *Branch and Bound* dapat diterapkan dalam berbagai permasalahan antara lain: *N-Queen Problem*, *N-Puzzle*, *Knapsack Problem* dan *Travelling Salesman Problem*.

Selain itu, digunakan juga tinjauan pustaka dari beberapa penelitian sebelumnya sebagai referensi utama yaitu skripsi yang ditulis oleh Eka Riyanti pada tahun 2004 yang berjudul "*Penerapan Algoritma Branch and Bound untuk Penentuan Rute Obyek Wisata*". Dalam skripsi tersebut, dijelaskan mengenai definisi, konsep dasar algoritma *Branch and Bound* dan aplikasinya dalam menyelesaikan permasalahan menentukan rute obyek wisata di provinsi Jawa Barat. Penelitian lain yang digunakan sebagai referensi yaitu skripsi yang ditulis oleh Bowo Kristanto pada tahun 2012 yang berjudul "*Aplikasi Algoritma Bellman-Ford dalam Meminimumkan Biaya Operasional Rute Penerbangan*" dan skripsi yang ditulis oleh Dian Nurdiana pada tahun 2010 yang berjudul "*Implementasi Algoritma Lebah untuk Pencarian Jalur Terpendek dengan Mempertimbangkan Heuristik*". Dalam dua penelitian tersebut, membahas mengenai permasalahan dalam penentuan sebuah rute terpendek, sehingga dapat dijadikan tambahan sumber referensi dalam penulisan skripsi ini.

Skripsi ini akan membahas mengenai implementasi dari algoritma *Branch and Bound* yang dapat diterapkan dalam dunia nyata, khususnya permasalahan penentuan rute pengangkutan sampah di kota Yogyakarta. Selain itu, dalam skripsi ini juga akan dibahas mengenai rancang bangun algoritma *Branch and Bound* dengan bantuan *software* MATLAB versi R2009b.

Adapun penelitian sebelumnya yang berhubungan dengan penelitian skripsi ini dapat diklasifikasikan sebagai berikut:

No	Nama Peneliti	Judul Penelitian	Perbedaan
1	Eka Riyanti (2004)	<i>Penerapan Algoritma Branch and Bound untuk Penentuan Rute Obyek Wisata.</i>	Membahas Konsep dan cara kerja <i>Algoritma Branch and Bound</i> serta aplikasinya terhadap penentuan rute obyek wisata di Jawa Barat.
2	Bowo Kristanto (2012)	<i>Aplikasi Algoritma Bellman-Ford dalam Meminimumkan Biaya Operasional Rute Penerbangan</i>	Membahas konsep dan cara kerja <i>Algoritma Bellman-Ford</i> serta penerapannya dalam menentukan rute penerbangan untuk meminimumkan biaya operasional.
3	Dian Nurdiana (2010)	<i>Implementasi Algoritma Lebah untuk Pencarian Jalur Terpendek dengan Mempertimbangkan Heuristik</i>	Membahas mengenai konsep dan cara kerja <i>Algoritma Lebah</i> serta aplikasinya dalam menentukan jalur mudik terpendek di pulau jawa.

No	Nama Peneliti	Judul Penelitian	Perbedaan
4	M.Shofi Alkhoirodai (2013)	<i>Implementasi Algoritma Branch and Bound untuk Optimasi Rute Pengangkutan Sampah Kota Yogyakarta.</i>	Membahas konsep dan cara kerja <i>Algoritma Branch and Bound</i> untuk optimasi rute pengangkutan sampah kota Yogyakarta yang disertai dengan rancang bangun menggunakan software <i>Matlab</i>

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 Branch and Bound*.

BAB III PEMBAHASAN

Bab ini merupakan pembahasan dari hasil penelitian yang berupa implementasi *Algoritma Branch and Bound* untuk optimasi rute pengangkutan sampah kota Yogyakarta dengan perhitungan secara manual serta membuat rancang bangun *Algoritma Branch and Bound* untuk optimasi rute pengangkutan sampah kota Yogyakarta menggunakan software *Matlab*.

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

Metode penelitian yang digunakan dalam penyusunan skripsi ini adalah:

1. Studi pustaka, yaitu pengumpulan data yang dilakukan dengan cara membaca buku-buku catatan secara literatur, mempelajari, dan meneliti berbagai buku bacaan yang berkaitan dengan masalah yang dibahas.
2. Studi lapangan, yaitu dengan mengamati posisi dan jumlah transfer depo dan TPA berdasarkan data sekunder yang diperoleh dari Badan Lingkungan Hidup kota Yogyakarta. Kemudian melakukan perhitungan jarak dari TPA menuju beberapa transfer depo dan kembali lagi ke TPA.

BAB IV

PENUTUP

4.1. Kesimpulan

Berdasarkan hasil pembahasan tentang implementasi algoritma *Branch and Bound* untuk optimasi rute pengangkutan sampah kota Yogyakarta, maka dapat ditarik kesimpulan sebagai berikut :

1. Penerapan algoritma *Branch and Bound* untuk optimasi rute pengangkutan sampah kota Yogyakarta dilakukan dengan cara merepresentasikan peta pengangkutan sampah ke dalam bentuk graf lengkap berbobot terlebih dahulu. Selanjutnya permasalahan yang berprinsip dengan graf Hamilton diselesaikan menggunakan algoritma *Branch and Bound*. Sehingga dihasilkan solusi rute pengangkutan sampah dengan jarak minimal. Berdasarkan perhitungan secara manual diperoleh solusi rute pengangkutan sampah: 1-13-14-3-5-9-7-8-10-6-11-12-2-4-1 (TPA - Depo Jl.Kemasan - Depo Pamukti - Depo Nitikan – Depo Wirosaban – Depo Jl.Brigjen Katamso – Depo Dukuh – Depo Pasar Ngasem – Depo TMP Ngabean – Depo Pringgokusuman – Depo Utoroloyo – Depo Sagan – Depo Mandalakrida – Depo Jl.Veteran – TPA) dengan total jarak kunjungan 28,65 Km.
2. Rancang bangun algoritma *Branch and Bound* dalam pencarian rute terpendek pengangkutan sampah kota Yogyakarta dilakukan dengan pembuatan program aplikasi menggunakan pemrograman GUI MATLAB 2009b. Terdapat dua hal mendasar untuk melakukan rancang bangun

tersebut yaitu pengaturan layout komponen GUI (Fig-File) dengan *user interface control*, dan pemrograman komponen GUI menggunakan M-File yang disesuaikan dengan algoritma *Branch and Bound*, sehingga dapat bekerja sesuai yang diharapkan. Berdasarkan perhitungan program aplikasi menghasilkan solusi rute pengangkutan yang sama dengan perhitungan manual yaitu: 1-13-14-3-5-9-7-8-10-6-11-12-2-4-1 atau (TPA - Depo Jl.Kemasan - Depo Pamukti - Depo Nitikan – Depo Wirosaban – Depo Jl.Brigjen Katamso – Depo Dukuh – Depo Pasar Ngasem – Depo TMP Ngabean – Depo Pringgokusuman – Depo Utoroloyo – Depo Sagan – Depo Mandalakrida – Depo Jl.Veteran – TPA) dengan total jarak kunjungan 28,65 Km.

4.2 Saran

Berdasarkan pada penelitian yang telah dilakukan, maka terdapat beberapa saran sebagai berikut:

1. Kemampuan program aplikasi yang dibuat masih terbatas pada nilai input data dengan nilai c_{ij} maksimal 3 digit di depan koma. Diharapkan penelitian selanjutnya untuk program aplikasi perlu dikembangkan dan disempurnakan lebih lanjut.
2. Dalam penelitian ini hanya membahas pemecahan masalah TSP dengan algoritma *Branch and Bound*. Diharapkan penelitian selanjutnya dapat mengembangkan algoritma lain seperti algoritma *Tabu Search* dan algoritma *Hill Climbing* yang dapat dijadikan bahan pertimbangan dalam menyelesaikan masalah TSP.

DAFTAR PUSTAKA

- Anton, Howard. 2000. *Dasar-dasar Aljabar*. Edisi 7. Jakarta: Interaksara.
- Abdusakir dkk. 2009. *Teori Graf*. Malang: UIN-Malang Press.
- Aldous, Joan M. and Wilson, Robin J. 2000. *GRAPH AND APPLICATIONS An Introductory Approach*. Great Britain : Springer.
- Bondy, J.A. dan Murty, U.S.R. 1976. *Graph Theory with Applications*. London: The Macmillan Press ltd.
- Diestel, Reinhard. 2005. *Graph Theory*. New York: Springer-Verlag Heidelberg.
- Jong, Jek Siang. 2002. *Matematika Diskrit dan Aplikasinya Pada Ilmu Komputer*. Yogyakarta: Andi.
- Knight, Andrew. 2000. *Basic of MATLAB and Beyond*. Boca Raton: CRC Press LLC.
- Kristanto, Bowo. 2012. Aplikasi Algoritma Bellman-Ford dalam Meminimumkan Biaya Operasional Rute Penerbangan. *Skripsi*. Yogyakarta: Program Studi Matematika Fakultas Sains dan Teknologi UIN Sunan Kalijaga.
- Levitin, Anany. 2010. *Pengantar Desain dan Analisis Algoritma*. Edisi ke 2. Diterjemahkan oleh: Efrizal Zaida. Jakarta : Salemba Infotek.
- Lipschutz, Seymour and Lipson, Marc. 2008. *Schaum's Outlines MATEMATIKA DISKRET*. Edisi ke 3. Diterjemahkan oleh Thombi Layukallo. Jakarta: Erlangga.
- Otto, S.R. and Denir, J.P. 2005. *An introduction to programming and numerical methods in MATLAB*. London: Springer.
- Purwanto, Eko Budi. 2008. *Perancangan dan Analisis Algoritma*. Yogyakarta: Graha Ilmu.
- Riyanti, Eka. 2004. Penerapan Algoritma Branch And Bound Untuk Penentuan Obyek Wisata. *Skripsi*. Bandung: Jurusan Teknik Informatika Fakultas Teknik Dan Ilmu Komputer Universitas Komputer Indonesia.
- Rossen, Kenneth H. 2012. *DISCRETE MATHEMATICS AND ITS APPLICATIONS*. 7th ed. New York: The McGraw-Hill Companies, Inc.
- Sugiharto, Aris. 2006. *Pemrograman GUI dengan MATLAB*. Yogyakarta: Penerbit Andi.
- Suryahadi. 1990. *Pengantar Teori dan Algoritma GRAPH*. Jakarta: Penerbit Gunadarma.
- Suyanto. 2010. *ALGORITMA OPTIMASI (Deterministik atau Probabilistik)*. Yogyakarta: Graha Ilmu.
- Wibisono, Samuel. 2004. *Matematika Diskrit*. Yogyakarta: Graha Ilmu.
- Wijaya, Willy Yandi. 2011. Petersen Graph And Some Related Properties. *Skripsi*. Yogyakarta: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada

LAMPIRAN 1:

Kode Program

```
function varargout = GUI_Aplikasi(varargin)
% GUI_APLIKASI M-file for GUI_Aplikasi.fig
%   GUI_APLIKASI, by itself, creates a new GUI_APLIKASI or raises the
existing
%   singleton*.
%
%   H = GUI_APLIKASI returns the handle to a new GUI_APLIKASI or the
handle to
%   the existing singleton*.
%
%   GUI_APLIKASI('CALLBACK',hObject,eventData,handles,...) calls the
local
%   function named CALLBACK in GUI_APLIKASI.M with the given input
arguments.
%
%   GUI_APLIKASI('Property','Value',...) creates a new GUI_APLIKASI or
raises the
%   existing singleton*. Starting from the left, property value pairs
are
%   applied to the GUI before GUI_Aplikasi_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to GUI_Aplikasi_OpeningFcn via
varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_Aplikasi

% Last Modified by GUIDE v2.5 04-Jun-2013 10:08:04

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @GUI_Aplikasi_OpeningFcn, ...
                  'gui_OutputFcn',  @GUI_Aplikasi_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```

% --- Executes just before GUI_Aplikasi is made visible.
function GUI_Aplikasi_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_Aplikasi (see VARARGIN)

% Choose default command line output for GUI_Aplikasi
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_Aplikasi wait for user response (see UIRESUME)
% uiwait(handles.figure1);
set(handles.axes1, 'visible', 'off');
set(handles.axes2, 'visible', 'on');
set(handles.text3, 'visible', 'on');
set(handles.text8, 'visible', 'on');
set(handles.figure1, 'name', 'TSP');
axes(handles.axes2);
gmb=imread('logo kampus.jpg');
imshow(gmb);
TSP= ' ';
save('temp.mat', 'TSP');

% --- Outputs from this function are returned to the command line.
function varargout = GUI_Aplikasi_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;
set(handles.axes1, 'visible', 'on');
set(handles.axes2, 'visible', 'off');
set(handles.text3, 'visible', 'off');
set(handles.text8, 'visible', 'off');
axes(handles.axes1);
set(handles.pushbutton1, 'enable', 'off');
set(handles.text1, 'string', 'Tunggu sedang analisa ...');
set(handles.edit1, 'string', ' ');
load('opsi.mat');
axes(handles.axes1);
if opsi==1
    gmb=imread('peta.jpg');
else
    gmb=imread('blank.jpg');
end
imshow(gmb);
hold off;
pause(0.1);
TSP= ' ';

```

```

save('temp.mat','TSP');

matriks_jarak=textread('matriks jarak.txt');
disp('Matriks jarak :');
disp(matriks_jarak);
[nbaris nkolom]=size(matriks_jarak);
if nkolom>nbaris
    nkolom=nkolom-1;
end
%mengubah harga nol menjadi harga yang cukup besar
for baris=1:nbaris
    matriks_jarak(baris,baris)=999;
end
%reduksi baris
matriks_A=matriks_jarak;
disp('Matriks mula-mula :');
disp(matriks_A);
% Total reduksi mula-mula dibuat 0
total_pereduksi=0;
% Scanning semua baris
for baris=1:nbaris
    % Untuk mencari nilai minimum,maka nilai minimum awaldibuat sbesar mungkin
    minim=999;
    % Scanning untuk semua kolom
    for kolom=1:nkolom
        % Bila ada nilai yang lebih kecil dari nilai minimum,
        % maka nilai tersebut diambil (dianggap nilai minimum)
        if matriks_A(baris,kolom)<minim
            minim=matriks_A(baris,kolom);
        end
    end
    % jika ditemukan nilai reduksi (tidak infiniti),
    % maka reduksi total ditambahkan dengan nilai minimumtersenut
    if minim<999
        total_pereduksi=total_pereduksi+minim;
    end
    % Matriks direduksi semua kolomnya,jika bukan nilai infiniti
    for kolom=1:nkolom
        if matriks_A(baris,kolom)>500
        else
            matriks_A(baris,kolom)=matriks_A(baris,kolom)-minim;
        end
    end
    strx=['Batis ke ' num2str(baris) ' , direduksi dengan '
num2str(minim)];
    disp(strx);
end
%return
disp(' ');
disp('Matriks setelah reduksi baris :');
disp(matriks_A);
disp(' ');
%reduksi kolom
% Scanning semua kolom
for kolom=1:nkolom
    % Untuk mencari nilai minimum,maka nilai minimum awaldibuat sbesar mungkin
    minim=999;
    % Scanning semua baris
    for baris=1:nbaris
        % Bila ada nilai yang lebih kecil dari nilai minimum,
        % maka nilai tersebut diambil (dianggap nilai minimum)
        if matriks_A(baris,kolom)<minim
            minim=matriks_A(baris,kolom);
        end
    end
end

```

```

        end
    end
    if minim<999
        if minim>0
            total_pereduksi=total_pereduksi+minim;
        end
    end
    % Matriks direduksi semua barisnya,jika bukan nilai infiniti, dan nilai
    minimum >0
    for baris=1:nbaris
        if matriks_A(baris,kolom)>500
        else
            matriks_A(baris,kolom)=matriks_A(baris,kolom)-minim;
        end
    end
    strx=['Kolom ke ' num2str(kolom) ' , direduksi dengan '
num2str(minim)];
    disp(strx);
end
disp(' ');
disp('Matriks setelah reduksi kolom :');
disp(matriks_A);
matriks_B=matriks_A;
disp('Total reduksi :');
C_root=total_pereduksi;
disp(C_root);
CS(1)=C_root;
CS1=C_root;
%return
simpulB=1;
str='matriks_B0.mat';
save(str,'matriks_B','simpulB','CS1');
%return
ulang=1;
xx=1;
indeks=0;
cs_min=999;
jml_data=1;
lanjut=1;
iter=0;
jml_data=1;
counter=1;
for iterasi=1:1
    iter=iter+1;
    str1=['Iterasi ke : ' num2str(iter)];
    disp(str1);
    str=['matriks_B' num2str(indeks)];
    %return
    load(str);
    for ibaris=1:nbaris
        for ikolom=1:nkolom
            B(ibaris,ikolom)=matriks_B(ibaris,ikolom);
        end
    end
    [nbaris nkolom]=size(B);
    %Pembacaan file teks oleh matlab kelebihan 1 kolom,maka jumlah kolom
    dikoreksi
    if nkolom>nbaris
        nkolom=nkolom-1;
    end
    % Mangambil data simpul sebelumnya sebagai simpul awal
    simpulA=simpulB;
    nsimpul=nbaris;

```

```

for simpul=2:nsimpul
    if simpul==simpulA
    else
        str=['Simpul ke ' num2str(simpulA) ' - ' num2str(simpul)];
        disp(str);
        clear reduksi_baris;
        clear reduksi_kolom;
        simpulB=simpul;
        B1=B;
%
        for kolom=1:nkolom
            B1(simpulA,kolom)=999;
            B1(1,kolom)=999;
        end
        for baris=1:nbaris
            B1(baris,simpulB)=999;
        end
        B1(simpulB,1)=999;
        disp('B1 awal');
        disp(B1);
        cnt1=0;
%reduksi baris
        for i=1:nbaris
            ada=1;
            for j=1:nkolom
                if B1(i,j)==0
                    ada=0;
                end
            end
            cnt2=0;
            for j=1:nkolom
                if B1(i,j)>500
                    cnt2=cnt2+1;
                end
            end
            if cnt2==nkolom
                ada=0;
            end
            if ada==1
                cnt1=cnt1+1;
                reduksi_baris(cnt1)=i;
            end
        end
        if cnt1>0
            %reduksi_baris
        end
        r=0;
        for i=1:cnt1
            minimum=999;
            baris=reduksi_baris(i);
            for kolom=1:nkolom
                if B1(baris,kolom)<minimum
                    minimum=B1(baris,kolom);
                end
            end
            %minimum
            strx=['Batis ke ' num2str(baris) ' , direduksi dengan '
num2str(minimum)];
            disp(strx);
            r=r+minimum;
            for kolom=1:nkolom
                if B1(baris,kolom)<999

```

```

        B1 (baris, kolom)=B1 (baris, kolom)-minimum;
    end
end
end
%reduksi kolom
cnt1a=0;
for i=1:ncolom
    ada=1;
    for j=1:nbaris
        if B1(j,i)==0
            ada=0;
        end
    end
    cnt2=0;
    for j=1:nbaris
        if B1(j,i)>100
            cnt2=cnt2+1;
        end
    end
    if cnt2==nbaris
        ada=0;
    end
    if ada==1
        cnt1a=cnt1a+1;
        reduksi_kolom(cnt1a)=i;
    end
end
if cnt1a>0
    %reduksi_kolom
end
for i=1:cnt1a
    minimum=999;
    kolom=reduksi_kolom(i);
    for baris=1:nbaris
        if B1 (baris, kolom)<minimum
            minimum=B1 (baris, kolom);
        end
    end
    %minimum
    strx=['Kolom ke ' num2str(kolom) ' , direduksi dengan '
num2str(minimum)];
    disp(strx);
    r=r+minimum;

    %B1
    for baris=1:nbaris
        if B1 (baris, kolom)<999
            B1 (baris, kolom)=B1 (baris, kolom)-minimum;
        end
    end
end
disp(' ');
disp('B1 akhir :');
disp(B1);
strx1=['Total reduksi (r) = ' num2str(r)];
disp(strx1);
strx1=['CS (' num2str(simpulA) ') = ' num2str(CS(simpulA))];
disp(strx1);
strx1=['Matriks (' num2str(simpulA) ', ' num2str(simpul) ') =
' num2str(B(simpulA, simpul))];
disp(strx1);

```

```

CS(simpul)=CS(simpulA)+matriks_B(simpulA,simpul)+r;

CS1=CS(simpul);
disp('Nilai CS');
disp(CS(simpul));
if CS(simpul)<cs_min
    cs_min=CS(simpul);
    cmin=cs_min;
    titik=simpulB;
    indeks=xx;
    strx=['matriks_B' num2str(indeks) '.mat'];
    save(strx,'B1','simpulB','cmin');
    jml_data=1;
    tabu(counter)=simpulA;
else
    if CS(simpul)==cs_min
        indeks=indeks+1;
        strx=['matriks_B' num2str(indeks) '.mat'];
        save(strx,'B1','simpulB','cmin');
        jml_data=jml_data+1;
    end
end
end
end
%cs_min
%return
%clc
xx=indeks+1;
%return
cs_min=999;
CS2=0;
for ulang=1:nbaris-2
    counter=counter+1;
    cs_min=999;
if jml_data>0
    %jml_data=1;
    indeks1=xx-jml_data;
    %return
    batas=jml_data;
    for datax=1:batas
        str=['matriks_B' num2str(indeks1+datax-1)]
        load(str);
        CS1=cmin;
        if CS2>0
            %CS1=CS2;
        end
        disp('Matriks awal :');
        B=B1;
        disp(B);
        disp(' ')
        [nbaris nkolom]=size(B);
        if nkolom>nbaris
            nkolom=nkolom-1;
        end
        simpulA=simpulB;
        if ulang==2
            %    xx
            %    indeks1
            %pause
        end
        nsimpul=nbaris;
        for simpul=2:nsimpul

```



```

terus=1;
if simpul==simpulA
    terus=0;
end
[m n]=size(tabu);
%simpul
for icari=1:n

    %tabu(icari)
    if simpul==tabu(icari)
        terus=0;
    end
end
if ulang==2
    if simpul==13
        %terus
        %return
    end
end
if terus==1
    %terus
    str=['Simpul ke ' num2str(simpulA) ' - '
num2str(simpul)];
    disp(str);
    clear reduksi_baris;
    clear reduksi_kolom;
    simpulB=simpul;
    B1=B;

%
    for kolom=1:nkolom
        B1(simpulA,kolom)=999;
        B1(1,kolom)=999;
    end
    for baris=1:nbaris
        B1(baris,simpulB)=999;
    end
    B1(simpulB,1)=999;
    disp('Matriks sebelum reduksi :');
    disp(B1);
    disp(' ');
    cnt1=0;

%reduksi baris
    for i=1:nbaris
        ada=1;
        for j=1:nkolom
            if B1(i,j)==0
                ada=0;
            end
        end
        cnt2=0;
        for j=1:nkolom
            if B1(i,j)>100
                cnt2=cnt2+1;
            end
        end
        if cnt2==nkolom
            ada=0;
        end
        if ada==1
            cnt1=cnt1+1;
            reduksi_baris(cnt1)=i;
        end
    end
end

```

```

        if cnt1>0
            %reduksi_baris
        end
        r=0;
        for i=1:cnt1
            minimum=999;
            baris=reduksi_baris(i);
            for kolom=1:nkolom
                if B1(baris,kolom)<minimum
                    minimum=B1(baris,kolom);
                end
            end
            %minimum
            strx=['Batis ke ' num2str(baris) ' , direduksi
dengan ' num2str(minimum)];
            disp(strx);
            r=r+minimum;
            for kolom=1:nkolom
                B1(baris,kolom)=B1(baris,kolom)-minimum;
            end
        end
    end
    %reduksi kolom
    cnt1a=0;
    for i=1:nkolom
        ada=1;
        for j=1:nbaris
            if B1(j,i)==0
                ada=0;
            end
        end
        cnt2=0;
        for j=1:nbaris
            if B1(j,i)>100
                cnt2=cnt2+1;
            end
        end
        if cnt2==nbaris
            ada=0;
        end
        if ada==1
            cnt1a=cnt1a+1;
            reduksi_kolom(cnt1a)=i;
        end
    end
    if cnt1a>0
        %reduksi_kolom
    end
    for i=1:cnt1a
        minimum=999;
        kolom=reduksi_kolom(i);
        %B1(:,kolom)
        for baris=1:nbaris
            if B1(baris,kolom)<minimum
                minimum=B1(baris,kolom);
            end
        end
        %minimum
        strx=['Kolom ke ' num2str(kolom) ' , direduksi dengan '
num2str(minimum)];
        disp(strx);
        r=r+minimum;
        for baris=1:nbaris
            B1(baris,kolom)=B1(baris,kolom)-minimum;
        end
    end
end

```

```

        end
    end
    disp(' ');
    disp('Matriks setelah reduksi :');
    disp(B1);
    disp(' ');
    strx1=['Total reduksi (r) = ' num2str(r)];
    disp(strx1);
    strx1=['CS (' num2str(simpulA) ') = ' num2str(CS(simpulA))];
    disp(strx1);
    strx1=['Matriks (' num2str(simpulA) ',' num2str(simpul) ') = ' num2str(B(simpulA,simpul))];
    disp(strx1);

    if batas>1
        CS_old=CS(simpul);
    else
        CS_old=0;
    end
    CS(simpul)=CS1+B(simpulA,simpul)+r;
    disp('Nilai CS');
    disp(CS(simpul));
        if CS(simpul)<cs_min
            cs_min=CS(simpul);
            cmin=cs_min;
            indeks=xx;
            strx=['matriks_B' num2str(indeks) '.mat'];
            save(strx,'B1','simpulB','cmin');
            jml_data=1;
            tabu(counter)=simpulA;
        else
            if CS(simpul)==cs_min
                indeks=indeks+1;
                strx=['matriks_B' num2str(indeks) '.mat'];
                save(strx,'B1','simpulB','cmin');
                jml_data=jml_data+1;
            end
        end
        if batas>1
            if CS(simpul)>CS_old
                if CS_old>0
                    CS(simpul)=CS_old;
                end
            end
        end
    end
    end
    end
    %xx=indeks+1;
    %datax
    %cs_min
    cs_min=999;
end
xx=indeks+1;
end
end
%pause
end
[m n]=size(tabu);
for i=1:nbaris
    ada=0;
    for j=1:n
        if tabu(j)==i
            ada=1;

```

```

        break;
    end
end
if ada==0
    counter=n+1;
    tabu(counter)=i;
end
end
TSP=tabu;
[m n]=size(TSP);
clear CS1;
for i=1:n
    j=TSP(i);
    CS1(i)=CS(j);
end
for i=1:n
    if i==1
        str=num2str(TSP(i));
    else
        str=[str num2str(TSP(i))];
    end
    str=[str '(' num2str(CS1(i)) ')'];
    if i<n
        str=[str ' - '];
    end
end
disp('TSP :');
disp(str);
set(handles.text1,'string','Ready ...');
set(handles.edit1,'string',str);
set(handles.edit2,'string',CS1(i));
pause(0.1);

load('opsi.mat');
axes(handles.axes1);
if opsi==1
    gmb=imread('peta.jpg');
else
    gmb=imread('blank.jpg');
    plot off;
end
imshow(gmb);
hold on;
xy=textread('koordinat.txt');
titik1=TSP(1);
for i=2:n
    titik2=TSP(i);

x(1)=xy(titik1,1);
y(1)=xy(titik1,2);
x(2)=xy(titik2,1);
y(2)=xy(titik2,2);
plot(x,y);
titik1=titik2;
end
titik2=TSP(1);

x(1)=xy(titik1,1);
y(1)=xy(titik1,2);
x(2)=xy(titik2,1);
y(2)=xy(titik2,2);

```

```

plot(x,y,'r');
hold off
save('temp.mat','TSP');
set(handles.pushbutton1,'enable','on');

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
!notepad matriks jarak.txt

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a
double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function Option_Callback(hObject, eventdata, handles)
% hObject    handle to Option (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function help_Callback(hObject, eventdata, handles)
% hObject    handle to help (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.uipanel2,'visible','off');
set(handles.close,'visible','on');
set(handles.uipanel6,'visible','on');
set(handles.text6,'visible','on');
set(handles.edit1,'visible','off');
set(handles.pushbutton1,'visible','off');
set(handles.pushbutton2,'visible','off');
set(handles.text2,'visible','off');
set(handles.axes1,'visible','off');
set(handles.axes2,'visible','off');
set(handles.listbox2,'visible','on');
txt1='help.txt';

```

```

fr2=fopen(txt1,'r');
lanjut=1;
baris=0;
while(lanjut==1)
    str2=fgets(fr2)
    if strcmp(str2,'end')==1
        lanjut=0;
    else
        baris=baris+1
        list1(baris,:)=cellstr(str2);
    end
end
fclose(fr2);
list1
set(handles.listbox2,'string',list1);
disp('ok');

% -----
function Option1_Callback(hObject, eventdata, handles)
% hObject    handle to Option1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
opsi=1;
save('opsi.mat','opsi');

% -----
function Option2_Callback(hObject, eventdata, handles)
% hObject    handle to Option2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
opsi=2;
save('opsi.mat','opsi');

% --- Executes on button press in close.
function close_Callback(hObject, eventdata, handles)
% hObject    handle to close (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.close,'visible','off');
set(handles.uipanel6,'visible','off');

set(handles.edit1,'visible','on');
set(handles.pushbutton1,'visible','on');
set(handles.pushbutton2,'visible','on');

load('temp.mat');
load('opsi.mat');
[m n]=size(TSP);
clc;
TSP
set(handles.uipanel1,'visible','on');
set(handles.listbox2,'visible','off');
if TSP==' '
set(handles.uipanel2,'visible','on');
set(handles.text2,'visible','on');
else
set(handles.axes1,'visible','on');
set(handles.axes2,'visible','off');
set(handles.text3,'visible','off');
axes(handles.axes1);
if opsi==1

```

```

        gmb=imread('peta.jpg');
    else
        gmb=imread('blank.jpg');
    end
    imshow(gmb);
    hold on;
    xy=textread('koordinat.txt');
    titik1=TSP(1);
    for i=2:n
        titik2=TSP(i);

    x(1)=xy(titik1,1);
    y(1)=xy(titik1,2);
    x(2)=xy(titik2,1);
    y(2)=xy(titik2,2);
    plot(x,y);
    titik1=titik2;
    end
    titik2=TSP(1);

    x(1)=xy(titik1,1);
    y(1)=xy(titik1,2);
    x(2)=xy(titik2,1);
    y(2)=xy(titik2,2);
    plot(x,y,'r');
    hold off
    end

    pause(0.1);

    % --- Executes on selection change in listbox1.
    function listbox1_Callback(hObject, eventdata, handles)
    % hObject    handle to listbox1 (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)

    % Hints: contents = cellstr(get(hObject,'String')) returns listbox1
    contents as cell array
    %         contents{get(hObject,'Value')} returns selected item from
    listbox1

    % --- Executes during object creation, after setting all properties.
    function listbox1_CreateFcn(hObject, eventdata, handles)
    % hObject    handle to listbox1 (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    empty - handles not created until after all CreateFcns
    called

    % Hint: listbox controls usually have a white background on Windows.
    %         See ISPC and COMPUTER.
    if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

    % --- Executes on selection change in listbox2.
    function listbox2_Callback(hObject, eventdata, handles)
    % hObject    handle to listbox2 (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = cellstr(get(hObject,'String')) returns listBox2
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
listBox2

% --- Executes during object creation, after setting all properties.
function listBox2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listBox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: listBox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of edit2 as a
double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----

```


LAMPIRAN 2:

Hasil Output Program

