



UNIVERSITAS INDONESIA

**OPTIMASI RUTE DISTRIBUSI PRODUK MENGGUNAKAN
ALGORITMA *DIFFERENTIAL EVOLUTION*
UNTUK *VRP* DENGAN *TIME WINDOWS*
(STUDI KASUS DI PT.X)**

SKRIPSI

ANNISA ZAHARA

0706166516

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2011**



UNIVERSITAS INDONESIA

**OPTIMASI RUTE DISTRIBUSI PRODUK MENGGUNAKAN
ALGORITMA *DIFFERENTIAL EVOLUTION*
UNTUK *VRP* DENGAN *TIME WINDOWS*
(STUDI KASUS DI PT.X)**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana teknik

ANNISA ZAHARA


0706166516

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2011**

ii

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Annsia Zahara
NPM : 070616615
Tanda Tangan : 
Tanggal : 20 Juni 2011

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Annisa Zahara
NPM : 0706166516
Program Studi : Teknik Industri
Judul Skripsi : Optimasi Rute Distribusi Produk Menggunakan Algoritma *Differential Evolution* untuk *VRP* dengan *Time Windows* (Studi Kasus di PT. X).

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana S1 pada Program Studi Teknik Industri, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Ir. Amar Rachman, MEIM

Penguji : Prof. Dr. Ir. T. Yuri M. Zagloel, M.EngSc

Penguji : Ir. Sri Bintang Pamungkas, MSISE, Ph.D

Penguji : Ir. Djoko Sihono Gabriel, M.T

Ditetapkan di : Depok

Tanggal : 20 Juni 2011

KATA PENGANTAR

Puji syukur penulis ucapkan ke hadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini tepat pada waktunya. Penyusunan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Departemen Teknik Industri pada Fakultas Teknik Universitas Indonesia. Penyelesaian tugas akhir ini tidak lepas dari bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Kedua orang tua penulis, Ibu dan Papa, terima kasih tak terhingga penulis ucapkan untuk kasih sayang, perhatian, dukungan dan doa yang tiada habis-habisnya selama ini dan selama proses penyelesaian tugas akhir ini.
2. Bapak Ir. Amar Rachman, MEIM selaku dosen pembimbing skripsi yang sangat baik dan sabar dalam membimbing penulis dalam menyelesaikan tugas akhir ini. Terima kasih atas waktu yang telah Bapak luangkan untuk memberikan bimbingan, arahan, koreksi, dan diskusi selama pengerjaan tugas akhir ini.
3. Ibu Ir. Erlinda Muslim, MEE selaku dosen pembimbing akademis. Terima kasih atas arahan dan bimbingan selama penulis menjalankan studi.
4. Dr. Ir. Teuku Yuri M. Zagloel, MengSC, sebagai ketua Departemen Teknik Industri yang telah memberikan kesempatan pada penyusunan penelitian ini
5. Pak Komarudin yang telah memberikan pengarahan dan masukan mengenai algoritma *Differential Evolution* dan *Vehicle Routing Problem*.
6. Seluruh dosen Departemen Teknik Industri: Ibu Isti, Ibu Ana, Ibu Arian Dhini, Pak Ahmad, Pak Omar, Pak Boy, Pak Yadrifil, Pak Farizal, Pak Bintang, Pak Dachyar, Pak Djoko, Pak Rahmat, Ibu Amalia, Pak Dendi dan (Alm) Ibu Betrianis yang telah memberikan ilmu, pengalaman, dan pelajaran hidup kepada penulis.
7. Uda Irwan, Bapak Agus, Mas Ferdi, Bapak Hasan, dan karyawan lainnya atas semua bantuan, informasi dan bimbingannya selama berada di perusahaan
8. Adik-adik penulis tersayang, Yusuf dan Yasin, serta anggota keluarga lainnya yang selalu menghibur dan menyemangati penulis.

9. Mutia Pratiwi dan Vinda, teman penulis saat harus magang di PT.X untuk melakukan penelitian skripsi.
10. Teman senasip sepenanggungan, Zakiah Sungkar dan Bonbin yang sama-sama berjuang menghadapi algoritma *Differential Evolution* dan bahasa pemrograman Matlab. Makasih ya atas bantuan dan dukungannya.
11. Paul, Daril, Gersen, dan Oskar yang bersedia mengajarkan bahasa pemrograman Matlab di lab SEMS.
12. Ayuning dan Sarah, yang bersama-sama menempuh jalaran Fasilkom-FT untuk mencari ilmu mengenai bahasa pemrograman Matlab.
13. Aulya Nuraini, Reni Afriani, Khairiyah, dan Astriana Gita yang selalu menghibur dan menyemangati penulis saat penulis menghadapi kejenuhan dalam mengerjakan skripsi.
14. Seluruh teman-teman TI 2007 atas dukungan dan kebersamaannya.
15. Seluruh warga Teknik Industri UI yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyusunan penelitian ini dan prosedur-prosedur lainnya.
16. Pihak-pihak lain yang tidak bisa penulis sebutkan satu persatu.

Penulis sangat menyadari sepenuhnya bahwa skripsi ini masih jauh dari sempurna dan mempunyai banyak keterbatasan. Oleh karena itu, penulis menerima segala saran dan kritik yang membangun demi kesempurnaan skripsi ini. Akhir kata, penulis berharap skripsi ini tidaklah sia-sia dan dapat bermanfaat bagi siapapun yang berkepentingan.

Depok, Juni 2011

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini :

Nama : Annisa Zahara
NPM : 0706166516
Program Studi : Teknik Industri
Departemen : Teknik Industri
Fakultas : Teknik
Jenis karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

“Optimasi Rute Distribusi Produk Menggunakan Algoritma *Differential Evolution* Untuk *VRP* Dengan *Time Windows*. (Studi Kasus di *PT.X*).”

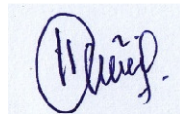
beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis / pencipta dan sebagai pemilih Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : Juni 2011

Yang menyatakan



(Annisa Zahara)

ABSTRAK

Nama : Annisa Zahara
Program Studi: Teknik Industri
Judul : Optimasi Rute Distribusi Produk Menggunakan Algoritma
Differential Evolution Untuk VRP Dengan *Time Windows*.
(Studi Kasus di PT.X)

Biaya transportasi merupakan salah satu komponen utama dalam struktur biaya logistik total. Transportasi yang efisien dapat memberikan kontribusi pada penurunan biaya logistik total. Salah satu komponen utama transportasi yang harus diperbaiki untuk mengurangi biaya transportasi adalah penentuan rute kendaraan yang dikenal dengan istilah *vehicle routing problem*. Tujuan dari VRP adalah mengatur urutan rute pengiriman sehingga menghasilkan jarak tempuh total seminimal mungkin. Untuk menyelesaikan permasalahan pengiriman produk di perusahaan pembuatan komponen otomotif ini, maka dikembangkan model penyelesaian VRP menggunakan algoritma *Differential Evolution*. Keunggulan DE adalah strukturnya yang sederhana, mudah diimplementasikan, cepat dalam mencapai tujuan, dan tangguh. Rute usulan hasil penelitian menghasilkan perbaikan, yaitu pengurangan jarak tempuh sebesar 23,87% dan biaya transportasi sebesar 17%.

Kata Kunci:

Vehicle routing problem, differential evolution, efisiensi biaya, optimasi

ABSTRACT

Name : Annisa Zahara
Study Program : Industrial Engineering
Title : Distribution Route Optimization Using Differential Evolution
Algorithm for Vehicle Routing Problem With Time Windows
(Case Study at PT.X)

Transportation cost constitute a main component in total logistic cost. Efficient transportation can contribute in saving total logistic cost. One of transportation component that should be corrected to reduce cost of transportation is vehicle routing problem. The objective of VRP is to set delivering routes order so that its total distance can be minimize. In order to solve delivery problem in this Component Automotif Manufacturing Industry, VRP model was developed using Differential Evolution algorithm. DE is a population based and direct stochastic search algorithm which simple, yet powerful and straightforward The result of this research shows the improvement that reduce total distance traveled by and transportation cost by 11,13%.

Key words:

Vehicle routing problem, differential evolution, cost efficiency, optimization

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN ORISINALITAS	iii
HALAMAN PENGESAHAN	iv
KATA PENGANTAR	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI	vi
ABSTRAK	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Diagram Keterkaitan Masalah.....	3
1.3 Rumusan Masalah.....	3
1.4 Tujuan Penelitian.....	5
1.5 Pembatasan Masalah.....	5
1.6 Metodologi Penelitian.....	5
1.7 Sistematika Penulisan.....	8
BAB 2 LANDASAN TEORI	9
2.1 <i>Vehicle Routing Problem</i>	9
2.1.1 Definisi <i>Vehicle Routing Problem</i>	9
2.1.2 <i>Vehicle Routing and Scheduling</i>	12
2.2 Metode Penyelesaian <i>Vehicle Routing Problem</i>	14
2.3 Algoritma <i>Differential Evolution</i>	15
2.3.1 Sejarah <i>Differential Evolution</i>	15
2.3.2 Konsep Dasar.....	17
2.3.3 Tahapan Pengerjaan DE.....	18
2.3.3.1 Inisialisasi.....	18
2.3.3.2 Mutasi.....	18

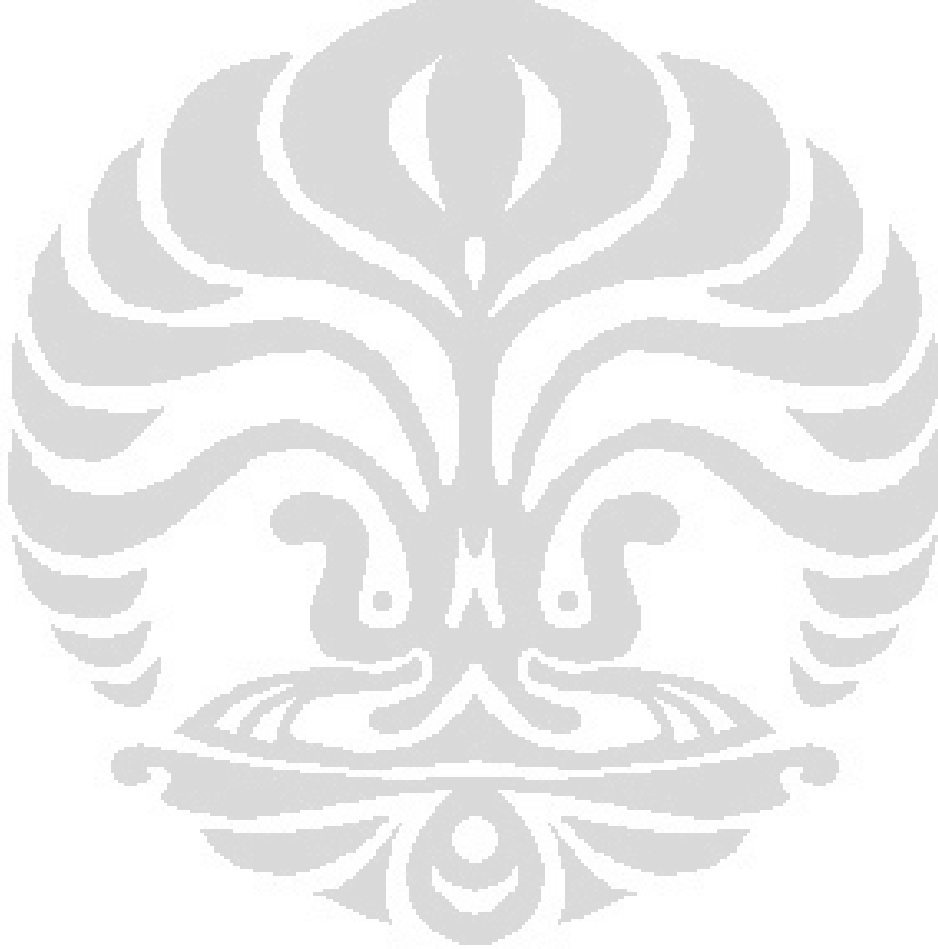
2.3.3.3	Pindah silang.....	19
2.3.3.4	Seleksi.....	19
2.3.3.5	Terminasi.....	20
BAB 3	PENGUMPULAN DATA.....	21
3.1	Profil Perusahaan.....	21
3.2	Pengumpulan Data.....	22
3.2.1	Armada Pengiriman.....	22
3.2.2	Lokasi Pelanggan.....	22
3.2.3	Jumlah Permintaan Pelanggan.....	23
3.2.4	Waktu.....	24
3.2.5	Jarak.....	25
3.2.6	Biaya Kendaraan.....	25
BAB 4	PENGOLAHAN DATA DAN ANALISIS.....	26
4.1	Penyusunan Algoritma.....	26
4.2	Verifikasi dan Validasi Program.....	34
4.3	Input Data.....	40
4.4	Hasil Pengolahan Data.....	42
4.5	Analisis.....	44
4.5.1	Analisis Metode.....	44
4.5.2	Analisis Program.....	45
4.5.3	Analisis Hasil Optimasi.....	46
4.5.3.1	Analisis Usulan Rute Distribusi.....	47
4.5.3.2	Analisis Utilitas Truk.....	50
4.5.3.3	Analisis Biaya.....	51
BAB 5	KESIMPULAN DAN SARAN.....	52
5.1	Kesimpulan.....	52
5.2	Saran.....	52
DAFTAR REFERENSI.....		54

DAFTAR TABEL

Tabel 3.1 Alamat Pelanggan.....	22
Tabel 3.2 Permintaan Pelanggan.....	24
Tabel 3.3 Biaya Sewa Kendaraan.....	25
Tabel 4.1 Hasil Percobaan Parameter Kontrol.....	27
Tabel 4.2 Hasil Percobaan Penentuan Iterasi.....	32
Tabel 4.3 Data Matriks Jarak <i>Dummy</i>	34
Tabel 4.4 Data Permintaan Pelanggan <i>Dummy</i>	34
Tabel 4.5 Parameter yang Digunakan Dalam Proses Validasi.....	35
Tabel 4.6 Matriks Populasi Target.....	35
Tabel 4.7 Permutasi Populasi Target.....	36
Tabel 4.8 Populasi Mutan.....	36
Tabel 4.9 Populasi Trial.....	37
Tabel 4.10 Permutasi Populasi Trial.....	37
Tabel 4.11 Populasi Solusi TSP Target.....	37
Tabel 4.12 Populasi Solusi TSP Trial.....	38
Tabel 4.13 Populasi Solusi VRP Awal.....	38
Tabel 4.14 Populasi Solusi VRP Trial.....	38
Tabel 4.15 Perbandingan Total Jarak Solusi Target dan Solusi Trial.....	39
Tabel 4.16 Matriks Jarak Pelanggan.....	40
Tabel 4.17 Daftar Permintaan Pelanggan.....	41
Tabel 4.18 Daftar Permintaan Pelanggan yang Melebihi Kapasitas Kendaraan.....	41
Tabel 4.19 Hasil <i>Run 1</i>	42
Tabel 4.20 Hasil <i>Run 2</i>	43
Tabel 4.21 Hasil <i>Run 3</i>	43
Tabel 4.22 Hasil <i>Run 4</i>	43
Tabel 4.23 Hasil <i>Run 5</i>	44
Tabel 4.24 Rute, Jarak, dan Waktu Tempuh Perusahaan Saat Ini.....	47
Tabel 4.25 Rute, Jarak, dan Waktu Tempuh Hasil Optimasi.....	48
Tabel 4.26 Perbandingan Total Jarak dan Waktu Tempuh.....	49
Tabel 4.27 Perbandingan Utilitas Truk Rute Saat Ini dan Rute Usulan.....	50
Tabel 4.28 Perbandingan Biaya Sewa Truk Rute Saat Ini dan Rute Usulan.....	51

DAFTAR GAMBAR

Gambar 1.1 Diagram Keterkaitan Masalah.....	4
Gambar 1.2 Metodologi Penelitian.....	7
Gambar 2.1 Permasalahan Dasar Kelas-Kelas VRP dan Keterkaitannya.....	11
Gambar 2.2 Tahap Pengerjaan <i>Differential Evolution</i>	20
Gambar 3.1 Lokasi Depot dan Pelanggan.....	23
Gambar 4.1 Diagram Alir Pengerjaan Algoritma <i>Differential Evolution</i>	33

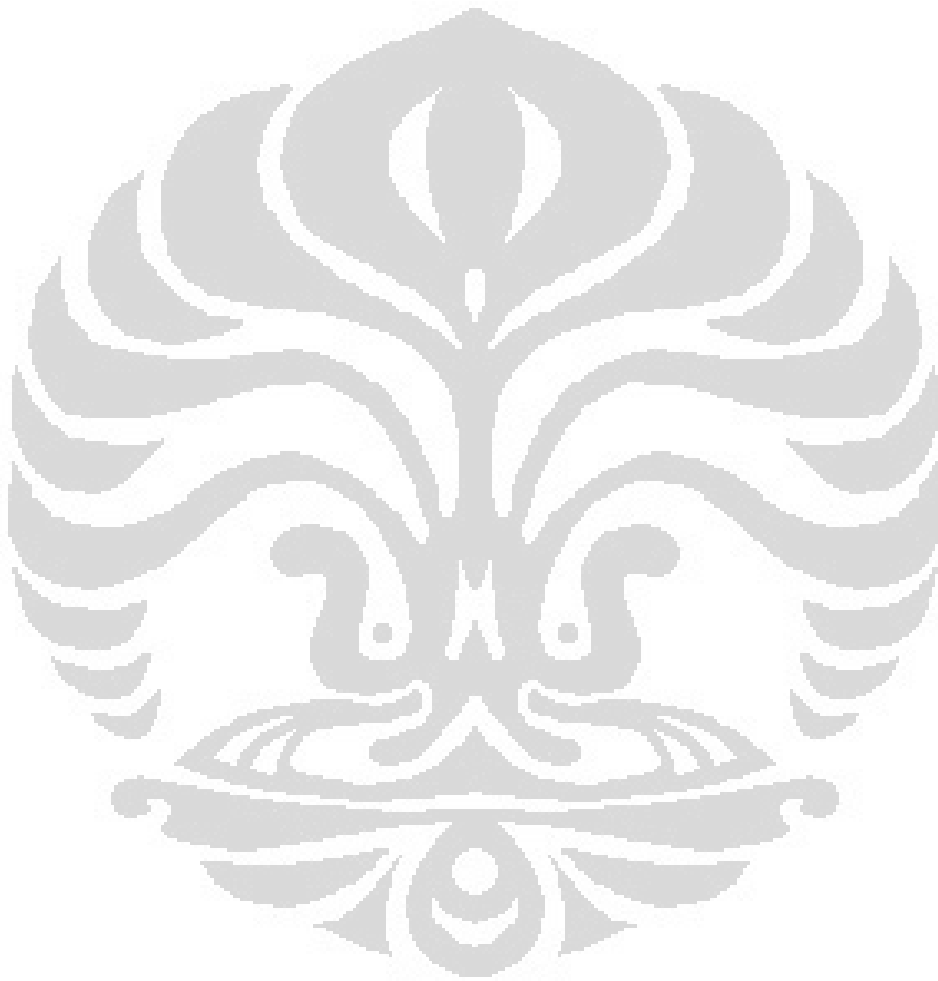


DAFTAR LAMPIRAN

Lampiran 1 : Matriks Jarak Antar Pelanggan

Matriks Waktu Tempuh Kendaraan

Lampiran 2 : *Script M-File* Program



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia industri, kemampuan untuk mengelola jaringan distribusi merupakan salah satu komponen keunggulan kompetitif yang memiliki peranan penting dalam suatu perusahaan. Banyaknya persaingan serta kebutuhan pelanggan yang tinggi mendorong perusahaan untuk melakukan berbagai perbaikan dalam kegiatan distribusi dan transportasi. Permasalahan distribusi dan transportasi menjadi penting karena besarnya proporsi biaya yang harus dikeluarkan untuk aktivitas-aktivitas distribusi dalam manajemen logistik. Biaya transportasi merupakan salah satu komponen utama dalam struktur biaya logistik total karena 1/3 sampai 2/3 dari total biaya logistik adalah total biaya transportasi¹. Suatu perusahaan harus dapat mengoptimalkan sistem distribusinya agar dapat bersaing dengan perusahaan sejenis lainnya. Salah satu cara yang dapat dilakukan adalah dengan mengoptimalkan transportasi.

Secara umum, masalah yang dapat menyebabkan kurang optimalnya transportasi di suatu perusahaan adalah jumlah permintaan yang berbeda-beda untuk tiap pelanggan, keterbatasan kapasitas kendaraan, batas waktu pengiriman untuk suatu titik, lokasi pelanggan, permintaan yang berfluktuatif, kapan sebaiknya pengiriman itu dilakukan, dan bagaimana membuat suatu rute dengan tujuan untuk meminimasi jumlah dari total jarak dan total waktu perjalanan agar dapat mengurangi biaya transportasi. Masalah ini juga dihadapi oleh PT. X, sebuah perusahaan manufaktur yang memproduksi komponen-komponen otomotif.

Pertimbangan perusahaan dalam mengatur rute distribusi hanyalah keterbatasan kapasitas, volume barang yang diangkut dan lokasi pelanggan, tanpa mengetahui apakah jarak tempuh rute yang dipilih sudah minimal atau belum sehingga biaya bahan bakar yang dikeluarkan belum tentu minimal. Penjadwalan

¹ Ronald H. Ballou, 2004

pengiriman dilakukan berdasarkan kapan pemesanan dari pelanggan datang dan besar permintaan komponen otomotif tersebut. Karena jumlah permintaan dari pelanggan tidak selalu sama, ada kalanya kapasitas kendaraan tidak mencukupi sehingga diperlukan biaya tambahan untuk menyewa kendaraan tambahan dan ada kalanya kendaraan tersebut dalam keadaan tidak terutilisasi secara penuh. Selain itu, jumlah permintaan yang tidak selalu sama juga berpengaruh terhadap waktu untuk memuat barang. Waktu yang dibutuhkan relatif lama jika permintaan pelanggan sedikit karena barang harus dikumpulkan terlebih dahulu hingga jumlah yang mencukupi untuk dikirim. Akibatnya perusahaan masih menemui kesulitan dalam mengirimkan barang ke pelanggan tepat waktu.

Untuk menyelesaikan permasalahan transportasi yang dihadapi perusahaan, salah satu komponen utama transportasi yang harus diperbaiki adalah penentuan rute kendaraan untuk melayani pelanggan. Masalah penentuan rute kendaraan ini dikenal dengan istilah *vehicle routing problem* (VRP). Permasalahan *Vehicle Routing Problem* (VRP) didefinisikan sebagai permasalahan mencari rute dengan ongkos minimal dari suatu depot yang melayani berbagai pelanggan yang letaknya tersebar dengan jumlah permintaan masing-masing. Tujuan VRP adalah mendapatkan suatu rencana rute yang meminimumkan total jarak, waktu atau biaya dengan tidak melebihi kapasitas kendaraan yang digunakan pada setiap rute. Rute yang ada dibuat sedemikian rupa sehingga setiap pelanggan pasti dikunjungi hanya sekali oleh satu kendaraan.

Permasalahan VRP yang akan dibahas dalam penelitian ini, salah satunya adalah *VRP Time Windows* (VRPTW). *Vehicle Routing Problem with Time Window* (VRPTW) hampir sama dengan VRP, namun memiliki batas tambahan yaitu sebuah jangka waktu yang berhubungan dengan setiap pelanggan, pelanggan memiliki interval waktu kapan pelanggan harus disuplai.

Beberapa metode yang digunakan untuk menyelesaikan VRP antara lain adalah dengan pendekatan eksak, heuristik, dan metaheuristik. Dibandingkan dengan heuristik klasik, metaheuristik menunjukkan pencarian solusi yang lebih teliti. Penelitian metaheuristik ini lebih menunjukkan perkembangan yang hebat dalam dekade terakhir dan telah menghasilkan heuristik VRP yang lebih efektif

dan fleksible². *Differential Evolution* (DE) merupakan metode terbaru yang dapat diimplementasikan untuk permasalahan VRP ini. Prinsip dasar DE didasarkan pada konsep evolusi biologi yang terdiri dari proses populasi, proses mutasi, proses pindah silang, dan proses penyeleksian. DE menggunakan acak sampling sehingga akan menyelesaikan penyelesaian yang berbeda meskipun model awalnya tidak berubah³.

Keunggulan DE adalah struktur yang sederhana, mudah diimplementasikan, cepat dalam mencapai tujuan, dan sifat yang tangguh³. DE akan menggabungkan elemen-elemen dari solusi-solusi yang telah ada untuk menciptakan solusi baru dengan mewarisi ciri-ciri yang dimiliki oleh tiap orang tua. Pencarian rute distribusi baru dengan algoritma *Differential Evolution* ini akan menghasilkan rute distribusi usulan yang terbaik dengan jarak terpendek dan biaya paling minimal yang pada akhirnya dapat meningkatkan sistem transportasi PT. X.

1.2 Diagram Keterkaitan Masalah

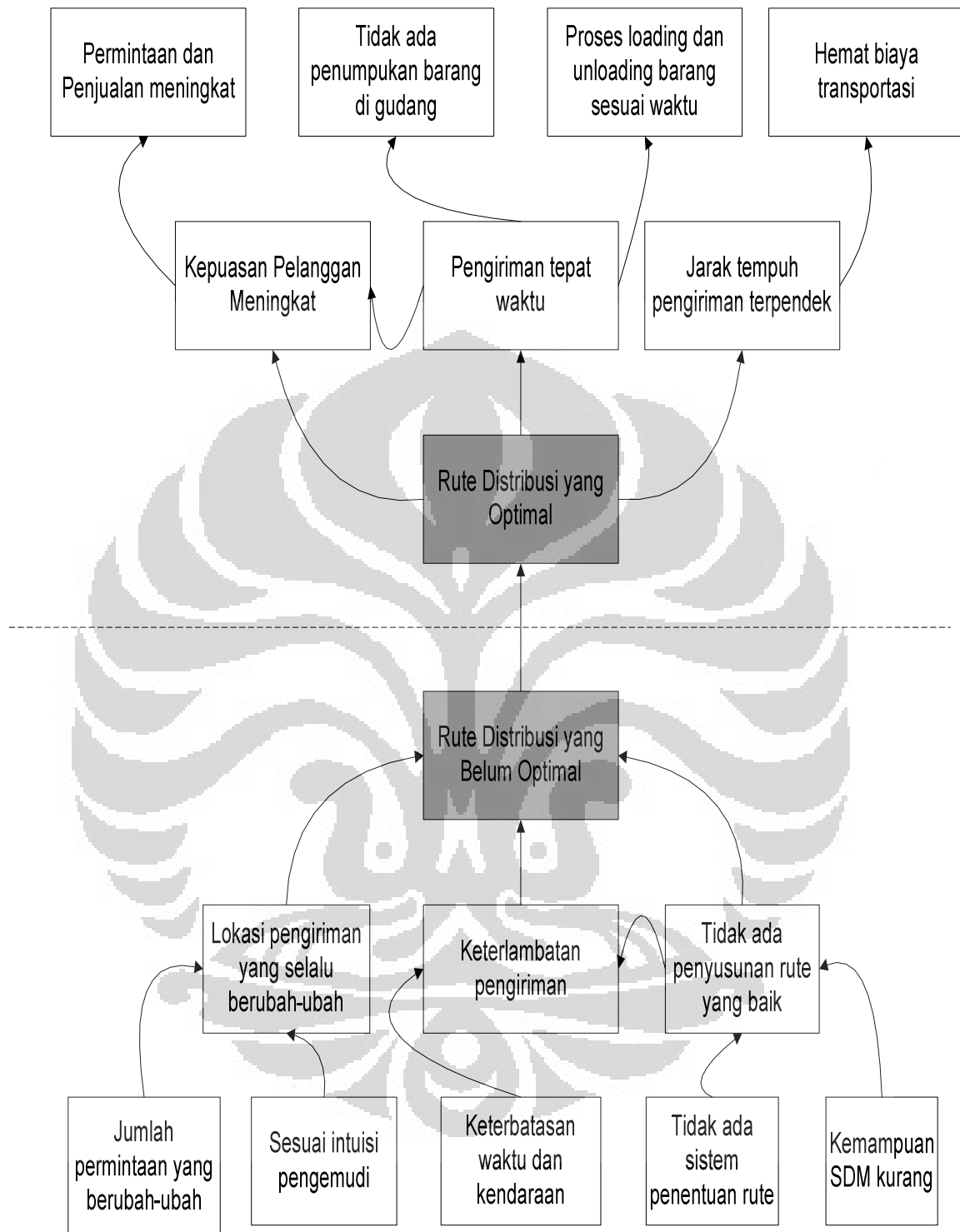
Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka dapat dibuat diagram keterkaitan masalah yang memberikan gambaran dari permasalahan secara sistematis. Diagram keterkaitan masalah dari penelitian ini dapat dilihat pada gambar 1.1.

1.3 Rumusan Masalah

Permasalahan yang dijadikan fokus dalam penelitian ini adalah bagaimana membuat rute distribusi yang optimal untuk meningkatkan efisiensi pengiriman barang di PT. X.

² K. V Price, "An Introduction to Differential Evolution". Editor D. Corne, M. Dorigo, dan F. Glover, *New Ideas in Optimization*, pages 79-108. Mc Graw-hill, UK, 1999

³ Ibid



Gambar 1.1 Diagram Keterkaitan Masalah

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini adalah memperoleh rute distribusi baru yang optimal untuk mengurangi biaya transportasi serta meningkatkan efisiensi pengiriman barang di PT. X dengan algoritma *Differential Evolution*.

1.5 Pembatasan Masalah

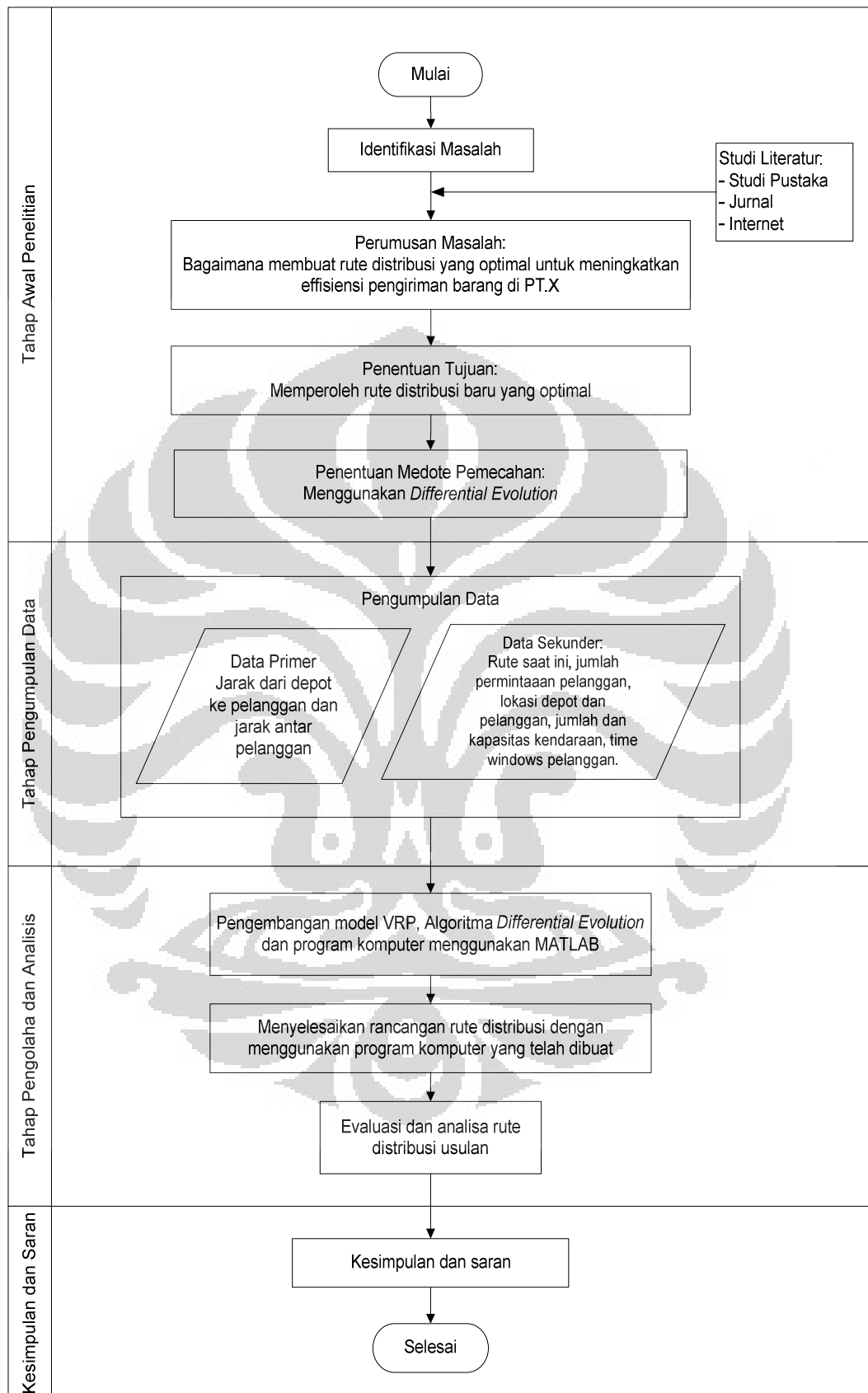
Batasan masalah dari penelitian ini digunakan agar masalah yang diteliti lebih terarah dan terfokus sehingga penelitian dapat dilakukan sesuai perencanaan awal dan memberikan hasil yang optimal. Batasan yang digunakan dalam penelitian ini adalah sebagai berikut:

- Penelitian dilakukan di salah satu pabrik milik PT. X (plant 1)
- Kondisi truk dianggap sama untuk tiap armada sejenis (kecepatan truk dan pemakaian bahan bakar)
- Rute yang akan dioptimasi adalah rute untuk suatu hari tertentu.
- Pemecahan permintaan pelanggan yang melebihi kapasitas kendaraan dibagi dengan cara mengisi penuh kendaraan yang memiliki kapasitas terbesar terlebih dahulu.
- VRP yang akan diselesaikan adalah VRP dengan *time windows* atau batasan waktu pelayanan.
- Pengolahan data dilakukan dengan pengembangan program komputer khusus dengan menggunakan bahasa pemrograman MATLAB.
- Penyimpangan yang terjadi di lapangan dianggap tidak ada, sehingga ketidakteraturan yang terjadi merupakan toleransi dari perusahaan dalam pelaksanaan di lapangan.

1.6 Metodologi Penelitian

Pada bagian ini, akan dijelaskan mengenai metodologi penelitian atau langkah-langkah yang dilakukan dalam penelitian ini. Metodologi penelitian dapat dilihat dari gambar 1.2 dan berikut penjelasan mengenai langkah-langkah yang dilakukan dalam penelitian ini:

1. Identifikasi masalah. Dalam penelitian ini, masalah yang diidentifikasi adalah rute distribusi yang saat ini digunakan oleh perusahaan pembuatan komponen otomotif tempat penelitian ini dilakukan belum optimal dan efisien.
2. Penentuan landasan teori melalui studi literatur. Setelah mengetahui permasalahan yang ada, maka tahap selanjutnya adalah menentukan landasan teori yang berhubungan dengan topik penelitian ini sebagai dasar dalam pelaksanaan penelitian. Studi literatur dilakukan dari berbagai sumber seperti buku, jurnal, dan artikel. Literatur utama yang digunakan dalam penelitian ini adalah *Vehicle Routing Problem* dan *Algoritma Differential Evolution*.
3. Menentukan tujuan penelitian. Penetapan tujuan penelitian ini disesuaikan dengan permasalahan yang ada. Oleh karena itu, tujuan dari penelitian ini adalah mendapatkan rute distribusi yang optimal untuk mengurangi biaya transportasi serta meningkatkan efisiensi pengiriman barang di PT. X.
4. Menetapkan batasan masalah. Batasan masalah perlu ditetapkan agar penelitian lebih terarah, terfokus dan berjalan sesuai dengan rencana.
5. Identifikasi dan pengumpulan data yang dibutuhkan. Pada tahap ini dilakukan identifikasi data yang dibutuhkan untuk menyelesaikan permasalahan dan kemudian dilakukan pengumpulan data-data tersebut. Data-data yang dibutuhkan yaitu data pelanggan, data armada yang digunakan, dan data rute distribusi yang diterapkan perusahaan saat ini.
6. Pengolahan data. Pada tahap ini data yang telah didapatkan kemudian diolah dan digunakan untuk membuat suatu rancangan rute distribusi baru dengan menggunakan metode *Differential Evolution*. Perancangan rute distribusi dengan VRP dilakukan dengan menggunakan bahasa pemrograman MATLAB.
7. Analisis hasil rancangan rute distribusi. Pada tahap ini dilakukan analisis terhadap rute distribusi yang telah dibuat dengan cara membandingkannya dengan rute yang selama ini digunakan.
8. Kesimpulan. Tahap terakhir adalah membuat kesimpulan dari penelitian berdasarkan hasil analisis yang telah dilakukan.



Gambar 1.2 Metodologi Penelitian

1.7. Sistematika Penulisan

Uraian tentang keseluruhan proses penelitian disusun dengan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab ini memuat latar belakang masalah, diagram keterkaitan masalah, perumusan masalah, tujuan penelitian, batasan masalah, metodologi penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini memuat teori-teori yang berhubungan dengan penelitian. Teori-teori tersebut diantaranya tentang *vehicle routing problem* dan metode *differential evolution* yang digunakan untuk menentukan rute distribusi yang optimal untuk transportasi perusahaan.

BAB III PENGUMPULAN DATA

Bab ini memuat uraian tentang profil perusahaan dan data-data yang digunakan dalam penelitian ini.

BAB IV PENGOLAHAN DATA DAN ANALISIS

Bab ini memuat tentang pengolahan data, data-data hasil penelitian, dan analisisnya.

BAB V KESIMPULAN DAN SARAN

Bab ini memuat kesimpulan dan saran. Kesimpulan tentang hasil pengumpulan dan pengolahan data serta pembahasan dan saran-saran bagi perusahaan.

BAB 2

LANDASAN TEORI

Secara umum, bab ini membahas mengenai landasan teori yang digunakan dalam penelitian. Pembahasan difokuskan pada VRP mulai dari definisi umum, modelnya, serta teknik pencarian solusi melalui algoritma eksak, heuristik, dan metaheuristik. Kemudian dijelaskan pembahasan yang lebih mendalam mengenai algoritma *Differential Evolution* sebagai salah satu pendekatan metaheuristik yang dapat digunakan untuk menyelesaikan permasalahan VRP pada penelitian ini.

2.1 *Vehicle Routing Problem*

2.1.1 Definisi *Vehicle Routing Problem*

Vehicle Routing Problem (VRP) adalah permasalahan mencari rute dengan biaya minimal dari suatu depot yang melayani berbagai pelanggan yang letaknya tersebar dengan jumlah permintaan masing-masing. Tujuan VRP adalah mendapatkan suatu rencana rute yang meminimumkan total jarak, waktu atau biaya dengan tidak melebihi kapasitas kendaraan yang digunakan pada setiap rute. Rute yang ada dibuat sedemikian rupa sehingga setiap pelanggan pasti dikunjungi hanya sekali oleh satu kendaraan.

Dalam penerapannya, para ahli menggunakan beberapa nama berbeda dengan permasalahan sama. Berikut adalah istilah VRP yang lain yaitu *vehicle scheduling problems* dan *vehicle dispatching*. *Vehicle routing problem* menekankan pada bagaimana membuat urutan mengunjungi pelanggan dengan kendaraan yang berangkat dan berakhir di depot. Bila diberikan tambahan keterangan waktu seperti waktu keberangkatan dan waktu kedatangan maka permasalahan menjadi *Vehicle scheduling problems*⁴. VRP pertama kali dipelajari oleh Dantzig dan Ramser (1959) dalam bentuk rute dan penjadwalan truk. Clarke dan Wright (1964) kemudian melanjutkan penelitian ini dengan memperkenalkan istilah depot sebagai tempat keberangkatan dan kembalinya

⁴ Bodin and Golden (1981).

kendaraan. Sejak saat itu penelitian VRP terus berkembang karena peran VRP yang penting dalam distribusi dunia industri. Tujuan utama dari VRP adalah menemukan solusi yang optimal. Untuk sebuah permasalahan VRP berskala besar di mana terdapat banyak pelanggan yang harus dilayani, upaya pencarian solusi yang optimal akan memakan waktu yang sangat lama. Oleh karena itu, berkembanglah berbagai macam pendekatan heuristik yang mampu menghasilkan solusi yang mendekati optimal dan dalam waktu yang lebih cepat. Pendekatan heuristik ini terus berkembang dengan dibuatnya pendekatan ke dalam prosedur yang terkomputerisasi sehingga menjadi sangat aplikatif dan mudah digunakan. Penggunaan prosedur yang terkomputerisasi di dalam perencanaan proses pendistribusian telah mampu menghasilkan penghematan yang cukup signifikan, yaitu berkisar antara 5-20% biaya transportasi.⁵

Karakteristik pelanggan dalam VRP:

- Menempatkan *road graph* di mana pelanggan berada
- Adanya *demand* dalam berbagai tipe dan harus diantarkan ke tempat pelanggan
- Waktu yang dibutuhkan untuk mengantarkan barang ke lokasi pelanggan, hal tersebut dapat berhubungan dengan jenis kendaraan
- Sekelompok kendaraan tersedia digunakan untuk melayani pelanggan
- Terdapat periode waktu (*time window*) dimana pelanggan dapat dilayani

Terdapat empat tujuan umum VRP, yaitu :

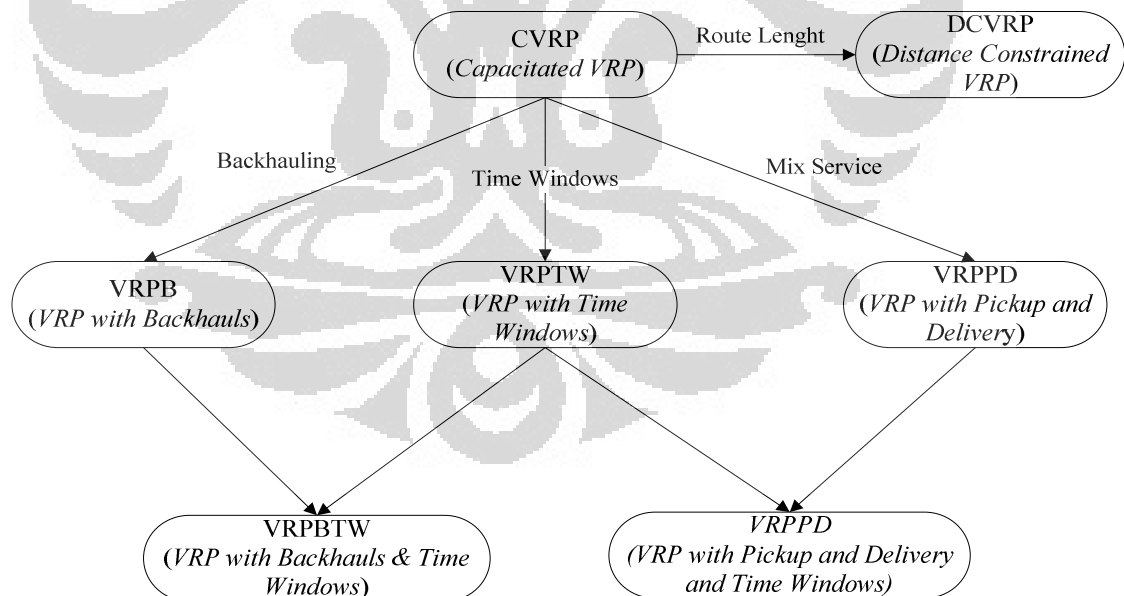
- Meminimalkan biaya transportasi global, terkait dengan jarak dan biaya tetap yang berhubungan dengan kendaraan
- Meminimalkan jumlah kendaraan yang dibutuhkan untuk melayani semua pelanggan
- Menyeimbangkan rute, untuk waktu perjalanan dan muatan kendaraan
- Meminimalkan penalti akibat *service* yang kurang memuaskan dari pelanggan.

⁵ Paolo Toth dan Daniele Vigo, *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, Philadelphia, 2001, h.1

Menurut Toth dan Vigo (2002) ditemukan beberapa variasi permasalahan utama VRP yaitu⁶:

- *Capacitated VRP (CVRP)*, merupakan kelas VRP yang paling sederhana dan yang paling banyak dipelajari dimana kendala yang ada hanya berupa kapasitas kendaraan yang terbatas.
- *Distance Constrained VRP (DCVRP)*, merupakan VRP dengan kendala batasan panjang rute.
- *VRP with time windows (VRPTW)*, yaitu kasus VRP dimana setiap konsumen memiliki batasan rentang waktu pelayanan.
- *VRP with Pick up and Delivery (VRPPD)*, merupakan VRP dengan pelayanan campuran, yaitu pengiriman dan pengambilan barang dalam satu rute.
- *VRP with Backhauls (VRPB)*, dimana pengambilan baru dapat dilakukan setelah semua pengiriman selesai.

Gambar berikut menunjukkan hubungan antara kelas-kelas VRP tersebut.



Gambar 2.1 Permasalahan Dasar Kelas-Kelas VRP dan Keterkaitannya

(Sumber: Toth & Vigo, 2002)

⁶ *Ibid.* hal.1

2.1.2 Vehicle Routing and Scheduling

Vehicle routing and scheduling merupakan perluasan dari *vehicle routing problem*. Beberapa batasan yang realistis yang termasuk di dalamnya adalah sebagai berikut⁷ :

1. Dalam setiap titik pemberhentian, ada sejumlah volume yang diambil dan dikirim.
2. Kendaraan yang digunakan mempunyai keterbatasan kapasitas yang berbeda baik kapasitas berat maupun volume.
3. Pelaksanaan pengambilan maupun pengiriman hanya dapat dilakukan pada waktu tertentu (*time windows*).
4. Operator kendaraan diperbolehkan istirahat atau makan siang pada waktu tertentu.
5. Setiap pelanggan memiliki type produk yang berbeda.

Beberapa batasan di atas menambah kompleksitas permasalahan sehingga sulit untuk menemukan solusi yang optimal. Untuk menemukan solusi yang paling optimal dapat diperoleh dengan cara menerapkan beberapa panduan untuk menghasilkan *routing* yang baik atau beberapa prosedur *logical heuristic* dengan pertimbangan kendaraan memulai perjalanan dari depot menuju ke beberapa titik pemberhentian untuk melakukan pengiriman dan kembali ke depot pada hari yang sama.

Pengembangan rute kendaraan yang bagus dapat dilakukan dengan mengaplikasikan delapan prinsip dasar berikut⁸:

1. Mengisi truk sebanyak volume pemberhentian yang akan didatangi dimana titik-titik pemberhentian tersebut letaknya berdekatan satu sama lain. Setelah itu titik-titik pemberhentian yang berdekatan perlu dibuat kelompok rute utk meminimumkan jarak antar pemberhentian, sehingga total waktu dalam satu rute menjadi minimum dengan demikian total waktu perjalanan dalam rute tersebut juga diminimumkan.

⁷ Ballou, Ronald H., *Bussiness Logistic Management* 5th edition, 2004,hal 235

⁸ Ibid h.235

2. Dalam pembuatan rute dimulai dari titik pemberhentian terjauh dari depot agar mendapatkan rute yg efisien. Rute yang efisien dapat dikembangkan dengan dimulai dari titik pemberhentian paling jauh dari depot ke titik yg paling dekat. Saat titik pemberhentian terjauh dari depot teridentifikasi, kapasitas yang tersisa dari kendaraan yang ditugaskan sebaiknya diisi dengan memilih sekelompok yang berdekatan dengan titik pemberhentian tersebut. Setelah kendaraan ditugaskan untuk volume titik-titik pemberhentian tersebut, mulailah membuat rute dengan kendaraan lain dan identifikasi titik-titik pemberhentian terjauh dari sisa titik-titik pemberhentian yg belum ditugaskan pada kendaraan. Terus lakukan prosedur ini sampai seluruh titik pemberhentian telah ditugaskan pada kendaraan.
3. Urutan pemberhentian pada sebuah rute sebaiknya membentuk pola air mata (*tear drop pattern*). Hal ini ditujukan agar tidak ada jalur yang bersilangan.
4. Rute yang paling efisien dibangun dengan menggunakan kendaraan dengan kapasitas terbesar. Idealnya, penggunaan truk berkapasitas besar untuk melayani banyak titik pemberhentian dalam satu rute akan meminimalkan jarak tempuh kendaraan. Sehingga, truk dengan kapasitas terbesar harus dialokasikan terlebih dahulu.
5. Pengambilan barang (*pick up*) sebaiknya digabungkan dengan rute pengiriman barang (*delivery*), daripada pengambilan barang baru dilakukan setelah semua pengiriman dilakukan. Hal ini guna meminimalkan jalur yg bersilangan yang dapat terjadi bila pengambilan dilakukan setelah seluruh pengiriman dilakukan.
6. Titik pemberhentian yang terpisah dari pengelompokan rute adalah kandidat terbaik untuk penggunaan alat transportasi lain. Titik pemberhentian yang terpisah dari pengelompokan, terutama titik pemberhentian dengan volume yang kecil, dilayani dengan waktu dan biaya yang relatif besar. Menggunakan kendaraan berkapasitas kecil untuk melayani titik pemberhentian tersebut dapat lebih ekonomis.
7. Batasan *time windows* titik pemberhentian yang berdekatan harus dihindari. Batasan *time windows* yang sangat dekat di antara pemberhentian dapat memaksa pembentukan urutan pemberhentian jauh dari pola ideal. Oleh karena *time windows* tidak bersifat mutlak maka sebaiknya dilakukan negosiasi

terhadap titik pemberhentian yang dipaksa untuk dilayani sesuai pola routing yg diinginkan.

2.2 Metode Penyelesaian Vehicle Routing Problem

Permasalahan untuk mendapatkan hasil solusi yang optimal dari pemecahan VRP (*Vehicle Routing Problems*) menjadi bertambah jika terdapat penambahan kendala (*constraint*) pada kasus yang harus diselesaikan. Kendala-kendala tersebut antara lain batasan waktu (*time window*), jenis kendaraan angkut yang berbeda-beda kapasitas angkutnya, total waktu maksimum operator kendaraan melakukan pengiriman, hambatan-hambatan yang di perjalanan, waktu istirahat operator kendaraan ketika melakukan pengiriman dan lain sebagainya. Pada dasarnya terdapat 3 macam penyelesaian VRP:

a. Solusi eksak

Pada solusi eksak dilakukan pendekatan dengan menghitung setiap solusi yang mungkin sampai satu solusi terbaik dapat diperoleh. *Branch and bound* dan *branch and cut* merupakan contoh dari penyelesaian eksak.

b. Heuristik

Metode Heuristik memberikan suatu cara untuk menyelesaikan permasalahan optimasi yang lebih sulit dan dengan kualitas dan waktu penyelesaian yang lebih cepat daripada solusi eksak. Contoh metode heuristik antara lain: *Saving Based*, *Matching Based*, *Multiroute improvement heuristic*, dll. Dari banyak pendekatan untuk memecahkan masalah VRP terdapat dua metode yang paling umum digunakan yaitu *sweep method* dan *savings method*.

c. Metaheuristik

Algoritma heuristik modern atau yang lebih dikenal dengan metaheuristik memecahkan masalah dengan melakukan perbaikan mulai dengan satu atau lebih solusi awal. Solusi awal ini bisa dihasilkan secara acak, bisa pula dihasilkan berdasarkan heuristik. Kualitas solusi yang dihasilkan dari metode ini jauh lebih baik daripada solusi yang didapat heuristik klasik. Contoh metaheuristik adalah *genetic algorithm*, *simulated annealing*, *tabu search*, *ant colony system*, *differential evolution* dsb.

2.3 Algoritma *Differential Evolution*

2.3.1 Sejarah *Differential Evolution*

Algoritma merupakan kumpulan perintah yang dapat digunakan untuk menyelesaikan suatu masalah dalam bidang matematika dan komputasi. Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat berupa apa saja, dengan catatan untuk setiap masalah, ada kriteria kondisi awal yang harus dipenuhi sebelum menjalankan algoritma. Algoritma sering mempunyai langkah pengulangan (iterasi) atau memerlukan keputusan (logika boolean dan perbandingan) sampai tugasnya selesai⁹. Algoritma memiliki banyak kegunaan di mana salah satu kegunaannya adalah untuk permasalahan optimasi, sehingga algoritma jenis ini biasa disebut dengan algoritma optimasi. Algoritma banyak digunakan dalam pemecahan masalah optimasi karena pada umumnya di banyak kasus di dunia nyata, permasalahan yang timbul memiliki permasalahan yang sulit atau tidak mungkin dikerjakan dengan menggunakan teknik-teknik optimasi konvensional yang dikerjakan secara manual. Misalnya, dalam kebanyakan kasus berskala besar, permasalahan optimasi yang ada memiliki jumlah variabel yang sangat besar hingga mencapai ratusan, memiliki fungsi-fungsi, baik kendala maupun tujuan, yang bersifat non-linier sehingga memiliki banyak optima lokal, atau fungsi yang non-kontinu.

Algoritma optimasi memiliki jenis yang sangat banyak, namun menurut Wolpert dan Macready hingga saat ini belum ada suatu algoritma superior yang dapat menyelesaikan permasalahan. Selama empat dekade, belum ada penelitian yang dapat memberikan solusi algoritma yang terbaik. Hingga pada akhir tahun 1995, Storn dan Price menawarkan suatu terobosan baru, yaitu algoritma *Differential Evolution* (DE). DE pertama kali mulai dikembangkan ketika Price mencoba memecahkan permasalahan *polynomial Chebyshev* yang diajukan oleh Storn. Dalam mencoba memecahkan permasalahan tersebut, Price terinspirasi untuk menggunakan selisih dari vector dalam mencari suatu solusi penyelesaian, hingga setelah melalui diskusi yang panjang dan simulasi dengan menggunakan

⁹ Microsoft Encarta Premium 2006, Algorithm. (2005).Microsoft

program komputer yang dilakukan oleh Storn dan Price, dikembangkanlah *Genetic Annealing* yang merupakan cikal bakal dari DE itu sendiri.

DE merupakan algoritma optimasi yang masuk ke dalam sub-kelompok algoritma evolusioner (EA). Sama seperti EA yang lainnya seperti *Genetic Algorithm (GA)*, *Evolution Strategy*, *Learning Classifier System*, dan lain-lain, DE memiliki konsep yang terinspirasi dari teori evolusi biologi, dimana di dalamnya terdapat reproduksi, mutasi, rekombinasi, dan seleksi. DE pertama kali dijelaskan oleh Price dan Storn di ICSI *technical report* pada tahun 1995. Satu tahun kemudian, DE sukses didemonstrasikan di *First International Contest on Evolutionary Optimization* yang diadakan bersamaan dengan *International Conference on Evolutionary Computation* yang diadakan oleh IEEE (*Institute of Electrical and Electronics Engineers*) dan berhasil memenangkan tempat ketiga. Terinspirasi dari hasil tersebut, Price dan Storn menulis sebuah artikel untuk jurnal Dr. Dobbs ("*Differential Evolution: A Simple Evolution Strategy for Fast Optimization*") yang diterbitkan pada April 1997 dan selanjutnya mereka menerbitkan artikel lagi untuk *Journal of Global Optimization* ("*Differential Evolution: A Simple and Efficient Heuristic for Global Optimization over Continuous Space*"). Artikel-artikel ini memperkenalkan DE ke komunitas internasional dan mendemonstrasikan keunggulan yang dimiliki oleh DE dibandingkan dengan metode heuristik yang lainnya. Pada tahun 1999, Price membuat suatu ringkasan yang berisi penjelasan mengenai algoritma DE ini dalam bentuk buku yang berjudul "*New Ideas in Optimization*". DE telah sukses diterapkan di berbagai bidang, baik teknik maupun sains, beberapa contoh diantaranya adalah:

- Desain filter digital (Storn, 1996)
- Pengambilan keputusan untuk produksi bahan bakar alkohol (Wang et al., 1998)
- Proses fermentasi untuk lot size tertentu (Chiou dan Wang, 1999)
- Perpaduan multi sensor (Joshi dan Sanderson, 1999)
- Optimasi dinamis untuk reaksi polimer yang terus-menerus (Lee et al., 1999)
- Optimasi pertukaran panas (Babu dan Munawar, 2000)

Universitas Indonesia

- Perencanaan persediaan (Srikanta dan Rambabu, 2003), dan lain-lain

2.3.2 Konsep Dasar

Seperti yang dijelaskan di atas, DE bekerja dengan meniru teori evolusi biologi. DE memiliki beberapa keunggulan dibandingkan dengan metode optimasi klasik, diantaranya adalah:

- Memiliki populasi yang berisikan calon-calon penyelesaian.
- Menghasilkan solusi-solusi yang berbeda meskipun model awalnya tidak dirubah, karena bekerja dengan menggunakan *random sampling*.
- Menggunakan elemen-elemen dari solusi-solusi yang telah ada untuk menciptakan solusi baru dengan ciri-ciri yang diwariskan dari elemen-elemen induknya.

Algoritma *Differential Evolution* tidak jauh berbeda dengan Algoritma Evolusioner lainnya, DE menggunakan vektor-vektor yang merepresentasikan kandidat-kandidat penyelesaian dimana teknik pencariannya dilakukan sekaligus atas sejumlah solusi yang disebut dengan populasi. Populasi awal (generasi ke nol) dibentuk dengan membangkitkan bilangan acak, sedangkan populasi berikutnya merupakan hasil evolusi dari vektor-vektor yang telah melalui tahap reproduksi, mutasi, rekombinasi, dan seleksi. Setiap individu didefinisikan sebagai vektor berdimensi-D dimana vektor-vektor tersebut dilambangkan sebagai $x_{i,g}$ yang merupakan anggota populasi pada generasi ke-g. Populasi dinotasikan sebagai P_x yang terdiri atas vektor-vektor tersebut yang berdimensi N_p dimana N_p merupakan ukuran populasi. Oleh karena itu, Populasi dan vektor yang menjadi calon-calon penyelesaian dapat dilambangkan ke dalam bentuk umum seperti berikut:

$$P_{x,g} = (x_{i,g}), i = 0,1,\dots,N_p-1, g = 0,1,\dots,g_{\max} \quad (2.1)$$

$$x_{i,g} = (x_{j,i,g}), j = 0,1,\dots,D-1 \quad (2.2)$$

Pada setiap generasi, tiap individu calon penyelesaian akan melewati proses evaluasi dimana individu-individu tersebut akan membentuk vektor target

dan dihitung fungsi objektifnya. Selain itu, pada individu-individu tersebut akan dilakukan proses mutasi dan pindah silang (*crossover*) agar dapat membentuk vektor trial yang digunakan untuk membentuk populasi anak (Populasi generasi selanjutnya). Populasi generasi selanjutnya akan dibentuk dengan cara membandingkan fungsi objektif dari vektor induk dan vektor trial dimana individu dengan nilai fungsi objektif yang terbaik akan lolos ke generasi selanjutnya. Proses tersebut akan terus diulang hingga kriteria terminasi terpenuhi.

2.3.3 Tahapan Pengerjaan DE

Penggunaan DE dalam proses pencarian solusi akan melalui tahapan-tahapan berupa inialisasi, mutasi, pindah silang, seleksi, dan terminasi. Tahapan ini dapat dilihat pada gambar 2.1.

2.3.3.1 Inialisasi

Pada tahap ini terjadi penetapan populasi awal (generasi ke-0) dan parameter kontrol. Populasi awal adalah seluruh individu dalam populasi sebelum iterasi DE dimulai. Individu dalam populasi itu merupakan sejumlah solusi awal yang dapat diperoleh dari metode heuristik maupun dari metode secara acak. Tiga parameter DE yaitu ukuran populasi (N_p), koefisien mutasi (F), dan koefisien pindah-silang (Cr). N_p merupakan jumlah individu dalam satu generasi, dan nilainya tidak akan berubah selama proses pencarian. Namun, jika pencarian mengalami hambatan, maka N_p dapat dinaikkan. Pada umumnya, nilai dari $N_p = 10 \times d$, di mana d merupakan ukuran dimensi (berupa baris matriks).

2.3.3.2 Mutasi

Proses selanjutnya setelah inialisasi adalah proses mutasi. Mutasi merupakan proses untuk membentuk vektor mutasi ($v_{i,g}$) yang diperoleh dari mengalihkan selisih dari dua vektor pada generasi sekarang yang dipilih secara acak dengan parameter kontrol mutasi (F) lalu dijumlahkan dengan vektor yang ketiga yang juga dipilih secara acak. Proses ini diformulasikan dengan rumus:

$$V_{i,g} = X_{r0,g} + F \cdot (X_{r1,g} - X_{r2,g}) \quad (2.3)$$

Keterangan :

$V_{i,g}$ = vektor mutasi ke i pada generasi ke g
 $X_{r0,g}, X_{r1,g}, X_{r2,g}$ = vektor yang dipilih secara acak pada generasi g
 F = koefisien mutasi

2.3.3.3 Pindah silang

Pindah silang merupakan proses yang bertujuan untuk menambah keanekaragaman gen dalam populasi yang akan memasuki generasi berikutnya dengan menyilangkan gen yang dimiliki oleh populasi vektor target dan populasi vektor mutan. Proses pindah silang ini melibatkan parameter kontrol pindah silang (Cr). Parameter pindah silang ini merupakan elemen yang menentukan gen-gen mana saja yang diperoleh dari vektor target dan mutan untuk diwariskan kepada vektor trial. Penentuan ini dilakukan dengan cara membandingkan nilai Cr tersebut dengan bilangan yang dibangkitkan secara acak.

Jika nilai Cr lebih besar dari bilangan acak, maka gen dari vektor mutasi akan lolos untuk memasuki vektor trial, sedangkan jika nilai Cr lebih kecil atau sama dengan bilangan acak, maka gen dari vektor target yang akan lolos memasuki vektor trial. Setelah diperoleh populasi dari vektor trial, maka vektor trial itu akan dievaluasi nilai objektifnya sebagaimana evaluasi yang dilakukan terhadap vektor target di mana nilai ini digunakan pada proses selanjutnya, yaitu proses seleksi.

2.3.3.4 Seleksi

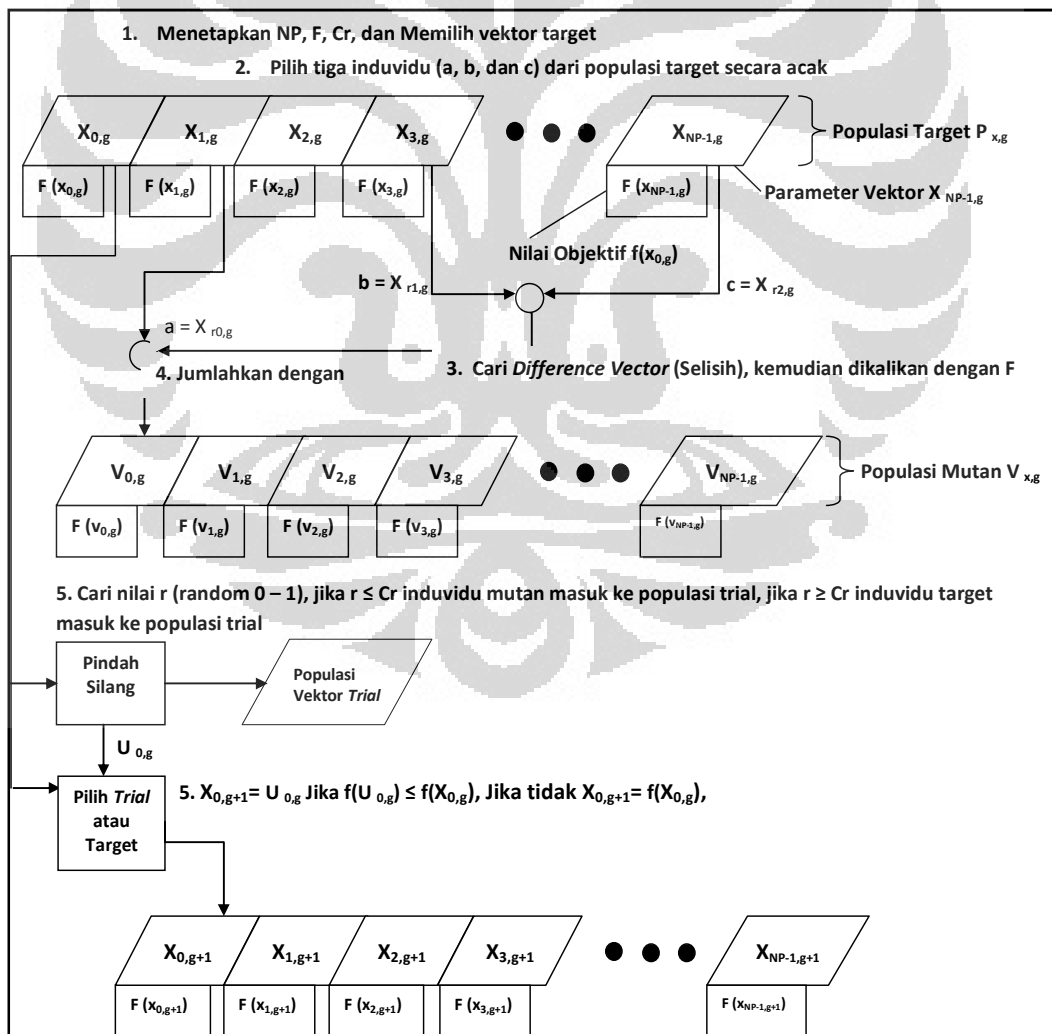
Tahapan ini merupakan tahapan di mana terjadi pemilihan antara vektor target dan vektor trial yang akan lolos untuk masuk ke generasi yang selanjutnya. Penyeleksian dilakukan dengan cara membandingkan nilai yang merupakan hasil dari evaluasi nilai objektif pada vektor target dan vektor trial. Vektor yang akan lolos ke generasi selanjutnya adalah vektor yang memiliki nilai evaluasi yang terbaik.

2.3.3.5 Terminasi

Terminasi merupakan keadaan dimana proses pencarian solusi optimal berhenti. Terminasi terjadi ketika proses pencarian solusi optimal telah mencapai kriteria terminasi. Namun, bila kriteria terminasi belum terpenuhi, maka akan dibentuk lagi generasi baru dengan mengulangi langkah-langkah sebelumnya dari awal. Umumnya kriteria terminasi adalah sebagai berikut

- Jumlah iterasi maksimum
- Waktu komputasi maksimum
- Mencapai keadaan konvergen (nilai dari fungsi objektif yang optimal tidak berubah)

Berikut gambaran proses pengerjaan DE secara keseluruhan:



Gambar 2.2 Tahap Pengerjaan Differential Evolution

BAB 3

PENGUMPULAN DATA

Pada bab ini akan dijelaskan mengenai pengumpulan data yang dibutuhkan untuk melaksanakan penelitian. Data yang digunakan dalam penyusunan penelitian tugas akhir ini merupakan data historis yang diperoleh langsung dari bagian *delivery* perusahaan tempat dilaksanakannya penelitian. Semua data yang diperlukan merupakan data-data yang berkaitan dengan aktivitas pengiriman barang ke pelanggan.

3.1 Profil Perusahaan

PT. X adalah sebuah perusahaan yang bergerak dalam bidang produksi suku cadang motor dan mobil yang terbuat dari material plastik. PT. X didirikan pada tahun 1991 sebagai anak perusahaan PT. FM dengan nama PT. FAS yang berlokasi di Ciluar Bogor. PT. FAS pada awal didirikannya hanya memproduksi beberapa produk yaitu suku cadang plastik dari kendaraan bermotor dan *sun-roof*. Pada bulan Juni tahun 1992, PT. FAS mulai menajaki proses diversifikasi dengan memproduksi *compact camera* bekerja sama dengan Jepang.

Seiring dengan berjalannya waktu, permintaan untuk suku cadang kendaraan terus mengalami peningkatan. Suku cadang plastik untuk kendaraan bermotor yang awalnya hanya diproduksi menggunakan 9 mesin injeksi hingga saat ini telah berkembang menjadi lebih dari 200 mesin injeksi. PT. FAS kemudian bergabung dalam CGA serta berganti nama menjadi PT. ADL-UAP yang kemudian berganti nama lagi menjadi PT. AO-X, karena merupakan divisi yang memproduksi suku cadang yang terbuat dari plastik yang menginduk pada PT. AO. PT.X saat ini terdiri atas *plant* I yang dibangun di atas lahan seluas 21,095m² / 8,436 m². Sedangkan *plant* II yang memproduksi *mirror* dibangun di atas lahan seluas 20.000 m² / 5,537 m².

3.2 Pengumpulan Data

3.2.1 Armada Pengiriman

Dalam melakukan pengiriman produk ke pelanggan, perusahaan pembuatan komponen otomotif ini menggunakan kendaraan berupa truk. Jumlah kendaraan yang tersedia adalah 29 buah dan ada tiga jenis truk yang digunakan dengan data-data sebagai berikut:

- Jumlah Truk : 29 Truk
- Kapasitas Muatan Truk:
 - 10 Truk : Kapasitas 4 ton atau 60 box 7008
 - 15 Truk : Kapasitas 9 ton atau 110 box 7008
 - 4 Truk : Kapasitas 12 ton atau 230 box 7008

Box 7008 merupakan box standar yang digunakan perusahaan untuk mengirim barang ke pelanggan. Ukuran box 7008 : (81,5 x 62 x 42)cm. Jumlah maksimum box 7008 yang dapat dimuat pada setiap truk telah ditetapkan oleh perusahaan berdasarkan penyusunan box yang paling efisien pada setiap jenis truk.

3.2.2 Lokasi Pelanggan

Terdapat beberapa lokasi pengiriman material yang dikunjungi oleh truk tiap harinya, yaitu:

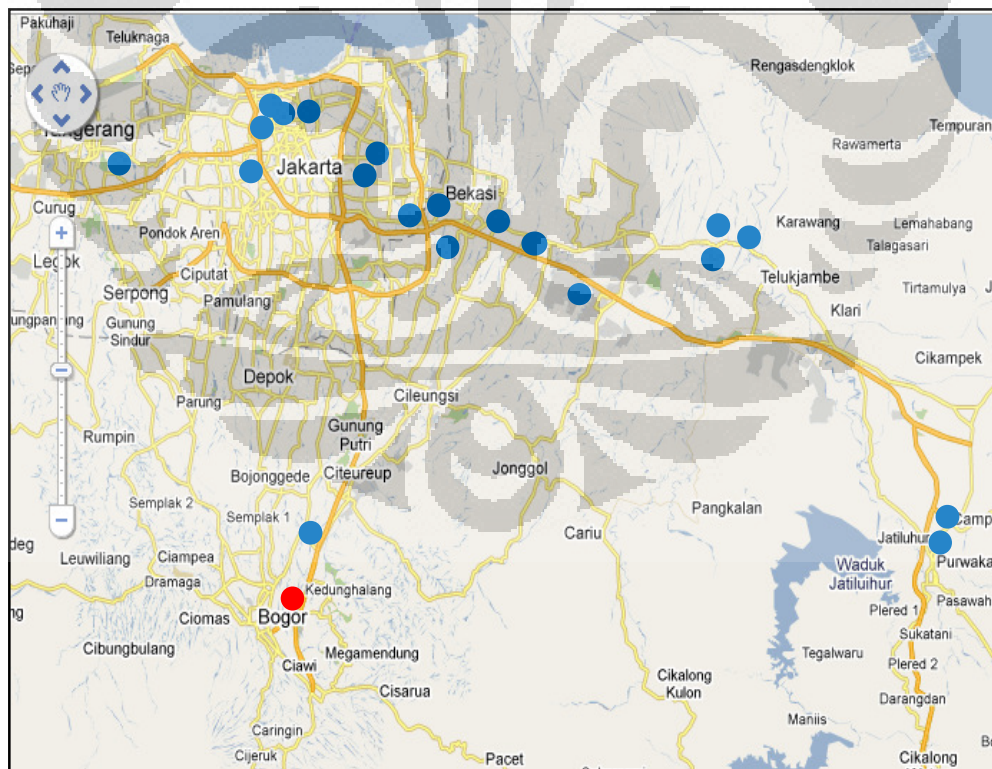
Tabel 3.1 Alamat Pelanggan

Kode Pelanggan	Alamat Pelanggan
ADMSPD	Jl. TOL. Jakarta - Cikampek KM. 47,5 41361 Karawang Barat
ADM-E	Jl. Selayar Kawasan Industri MM2100 Bekasi 17310
AHM1	Jl. Laksda Yos Sudarso Sunter I, Jakarta 14350-Indonesia
AHM2	Jl. Pegangsaan Dua No.66 3009 Pulogadung - Jakarta Utara
AHM3	Jl. Raya Kalimantan Blok AA Kawasan Industri MM2100 Cikarang Barat
HSO	Jl. Tipar Inspeksi Cakung Drain Jakarta Timur 13910
NMT	Jl. Pegangsaan Dua 3009 Pulogadung Jakarta Utara
CBI	Jl. Bekasi Raya Km. 25 Cakung Jakarta Timur 13960
DS	Jl. Gaya Motor I No. 6 Sunter II 14330 Jakarta Utara
DC	Jl. Kalimantan Blok E 1-2 MM 2100 Industrial Estate 17520 - Cibitung Bekasi
PPL	Jl. Gaya Motor III No. 5 Sunter II 14330 Jakarta Utara

Tabel 3.1 Alamat Pelanggan (Lanjutan)

Kode Pelanggan	Alamat Pelanggan
GMT	Jl. Narogong KM.12 3009 Pangkalan II, Bekasi
GS STR	Jl. Laksda Yos Sudarso Sunter I, Jakarta 14350-Indonesia
GS KRW	Jl. Surya Utama Kav. 13-14 Kawasan Industri Surya Cipta BTB 9, Karawang
HN	Kawasan Industri No.1, Purwakarta 41181
IZS	Jl. Gaya Motor II No. 1 Sunter Jakarta 14330
PD UNGU	Jl. Kaliabang Tengah No. 38 Pondok Ungu Bekasi 17125
KWK	Jl. Perintis Kemerdekaan, Kelapa Gading, Jakarta Utara 14250
NSN	Kawasan Industri Kota Bukit Indah Blok AIII Lot. 1-14, Purwakarta
SNG	Delta Silikon Industrial Blok L5-02 17550 Lipo Cikarang - Bekasi
SMU	Jl. Industri Raya 4 Blok AF No. 910 Tangerang
TGGS	Jl. Raya Jakarta Bogor KM 47,5 RT.03/03 16912 Nanggewer, Cibinong
TAM PIO STR	Jl. Gaya Motor III No. 4 Sunter III Jakarta Utara 14330
TAM KRW	Jl. Permata Raya Lot DD-1KWS Industri KIIC Karawang 41361
TAM SPD	Jl. Selayar 17310 Bekasi, Kawasan Industri MM 2100
TMMI	Jl. Laksda Yos Sudarso, Sunter II Jakarta Utara

(Sumber: Perusahaan)



● = Depot

● = Pelanggan

Gambar 3.1 Lokasi Depot dan Pelanggan

Universitas Indonesia

3.2.3 Jumlah Permintaan Pelanggan

Jumlah permintaan setiap pelanggan berbeda-beda dan satuan yang digunakan untuk jumlah permintaan ini adalah box khusus yang digunakan untuk melakukan pengiriman komponen otomotif ke pelanggan. Berikut jumlah permintaan tiap pelanggan:

Tabel 3.2 Permintaan Pelanggan



3.2.4 Waktu

Data waktu yang dibutuhkan dalam penelitian ini adalah waktu kerja perusahaan, waktu pengangkutan barang ke dalam truk (*loading time*), *time windows* setiap titik, dan *service time* (*unloading time*). Waktu kerja perusahaan untuk melakukan pengiriman barang dibagi menjadi beberapa cycle pengiriman dengan total waktu untuk tiap cycle selama 4 jam dan total waktu kerja perusahaan yaitu 8 jam. Untuk *time windows* pelanggan, tiap pelanggan memiliki *time windows* yang sama dengan waktu total pengantaran selama 8 jam dan batas awal pengantaran material dimulai pada pukul 08:00 WIB. Untuk waktu *loading* dan *unloading* ditetapkan dengan menggunakan perhitungan lamanya waktu penggunaan forklift untuk mengambil dan meletakkan box yang berisi kotak komponen dikalikan dengan jumlah box yang harus dipindahkan kemudian

dijumlahkan dengan 15 menit sebagai waktu *allowance*. Persamaan untuk menghitung waktu *loading* dan *unloading* adalah sebagai berikut:

$$t = (0,4 \times n) + 15 \quad (3.1)$$

keterangan:

t = waktu *loading/unloading* yang diperlukan

n = jumlah box yang dipindahkan

3.2.5 Jarak

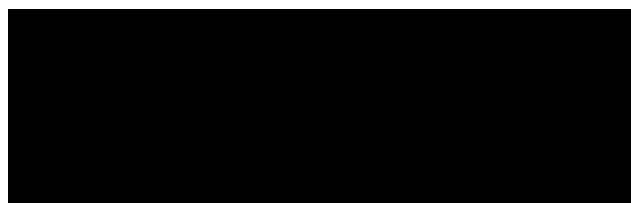
Data jarak yang dikumpulkan adalah jarak antara depot dengan pelanggan dan jarak antar pelanggan. Pengambilan data jarak ini dilakukan dengan menggunakan bantuan *Google Maps*. Aplikasi *Google Maps* tersebut memiliki *tool* atau alat bantu yang bernama *distance measurement tool*. *Tool* ini dapat digunakan untuk mengukur jarak antara dua titik yang berada di peta.

Pengukuran jarak antara dua titik dilakukan dengan mengikuti alur jalan yang ada di peta sehingga data jarak yang diperoleh dapat mendekati jarak aktual yang ditempuh oleh kendaraan. Pemilihan jalan yang mengabungkan dua titik tertentu dilakukan dengan pertimbangan jarak terdekat dan juga kondisi atau karakteristik jalan (tingkat kemacetan). Data jarak dari depot menuju pelanggan dan jarak antara pelanggan ditampilkan dalam bentuk matriks jarak.

3.2.6 Biaya Kendaraan

Biaya sewa untuk berbagai jenis truk dibagi menjadi dua bagian yaitu biaya tetap dan biaya tidak tetap. Biaya tetap yaitu biaya yang harus dikeluarkan setiap kali truk keluar dari depot dan melayani satu rute pengiriman barang. Biaya tidak tetap yaitu biaya yang harus ditambahkan sesuai dengan total jarak yang ditempuh dalam satu rute perjalanan truk. Berikut rincian biaya sewa truk yang telah ditetapkan antara perusahaan dan logistik patnernya:

Tabel 3.3 Biaya Sewa Kendaraan



BAB 4

PENGOLAHAN DATA DAN ANALISIS

Bab ini berisi pengolahan data dan analisisnya. Data yang telah diperoleh dan dijelaskan di bab sebelumnya akan diolah untuk membuat rancangan rute distribusi baru. Pengolahan data ini menerapkan algoritma *differential evolution* untuk memperoleh rute distribusi baru yang optimal. Setelah itu, dilakukan analisis terhadap rute distribusi usulan dengan membandingkan rute distribusi usulan tersebut dengan rute distribusi yang diterapkan perusahaan saat ini.

4.1 Penyusunan Algoritma

Dalam melakukan optimasi rute distribusi di perusahaan pembuatan komponen otomotif ini, salah satu algoritma yang dapat diterapkan adalah algoritma *differential evolution*. Untuk penerapan algoritma tersebut dilakukan dengan membuat program komputer menggunakan Matlab R2009a. (*Script M-file* program terlampir). Tahap-tahap penyusunan algoritma dalam penelitian ini adalah sebagai berikut:

1. Menentukan parameter kontrol

Tahap awal dalam penyusunan algoritma *differential evolution* ini adalah menentukan parameter kontrol yang akan digunakan. Parameter kontrol yang perlu ditetapkan adalah parameter ukuran populasi (NP), parameter kontrol mutasi (F), dan parameter kontrol pindah silang (Cr). Parameter ukuran populasi merupakan parameter untuk menentukan jumlah solusi awal yang ada pada populasi awal.

Berdasarkan studi literatur yang telah dilakukan, parameter kontrol untuk ukuran populasi yang sering digunakan adalah $(NP) = 10 \times d$ (jumlah dimensi/pelanggan) sehingga ukuran populasi untuk permasalahan kali ini adalah $(NP) = 10 \times 21$, yaitu 210 solusi awal pada populasi awal. Parameter mutasi berkisar antara 0-2 dengan nilai efektif antara 0,4-1, dan parameter pindah silang berkisar antara 0-1. Untuk menentukan nilai parameter mutasi dan parameter pindah silang yang terbaik dalam kasus

ini, maka akan dilakukan percobaan dengan menggunakan nilai efektif F dan Cr.

Tabel 4.1 Hasil Percobaan Parameter Kontrol

Parameter		Computation	Hasil	
F	Cr	Time	Total Jarak	Jumlah Perjalanan
0,4	0,5	80,00	1217	10
	0,6	78,73	1218	9
	0,7	78,19	1202	9
	0,8	80,44	1909	9
	0,9	79,00	1156	9
	1	77,13	1169	9
F	Cr	Time	Total Jarak	Jumlah Perjalanan
0,5	0,5	78,52	1225	10
	0,6	76,25	1200	9
	0,7	76,63	1195	9
	0,8	76,46	1213	10
	0,9	76,02	1133	10
	1	76,25	1192	9
F	Cr	Time	Total Jarak	Jumlah Perjalanan
0,6	0,5	76,26	1224	9
	0,6	79,27	1234	9
	0,7	76,67	1240	10
	0,8	81,26	1186	9
	0,9	78,67	1128	9
	1	80,44	1200	9
F	Cr	Time	Total Jarak	Jumlah Perjalanan
0,7	0,5	77,30	1234	10
	0,6	78,11	1214	9
	0,7	78,80	1220	9
	0,8	78,39	1146	9
	0,9	76,86	1146	9
	1	77,49	1226	10
F	Cr	Time	Total Jarak	Jumlah Perjalanan
0,8	0,5	79,45	1202	9
	0,6	78,36	1161	9
	0,7	77,91	1211	9
	0,8	76,99	1166	9
	0,9	76,58	1151	9
	1	76,5	1194	9
F	Cr	Time	Total Jarak	Jumlah Perjalanan
0,9	0,5	77,45	1233	9
	0,6	80,4	1226	9
	0,7	90,01	1195	9
	0,8	76,44	1180	9
	0,9	77,23	1176	9
	1	76,78	1226	9
F	Cr	Time	Total Jarak	Jumlah Perjalanan
1	0,5	78,81	1197	9
	0,6	76,71	1227	9
	0,7	76,46	1214	9
	0,8	76,95	1180	9
	0,9	76,98	1164	9
	1	80,01	1226	9

Dari hasil percobaan terlihat bahwa parameter F (0,6) dan Cr (0,9) menunjukkan hasil yang paling baik sehingga parameter tersebut akan digunakan seterusnya untuk penelitian ini.

2. Menentukan populasi awal

Tahap selanjutnya yaitu membentuk populasi awal. Populasi awal ini merupakan sebuah matrik yang berisi individu awal dengan jumlah (NP). Individu awal ini akan merepresentasikan jumlah rute untuk melayani seluruh pelanggan dan tiap rute memiliki urutan pengantaran masing-masing. Populasi awal dicari dengan melakukan *random sampling* atau secara acak. Karena populasi awal terdiri dari 210 individu dan setiap individu terdiri dari 21 dimensi, maka populasi awal merupakan matriks berukuran 21 x 210.

Pada algoritma DE ini, definisi individu dan vektor adalah serupa, dan dapat pula diartikan sebagai kromosom. Individu awal tidak merepresentasikan urutan rute, dengan demikian perlu dikonversikan ke dalam vektor operasi permutasi. Nilai dimensi ke-1 hingga ke-21 individu awal masih acak susunannya. Karena itu untuk setiap individu awal perlu dilakukan pengurutan nilai dimensi dari yang terkecil hingga yang terbesar. Pengurutan tersebut akan menghasilkan vektor berdimensi 21 yang nilai setiap dimensinya berupa indeks hasil pengurutan.

3. Menentukan fungsi objektif

Fungsi objektif atau fungsi tujuan ditetapkan agar tujuan dari penelitian ini yaitu mendapatkan rute distribusi yang optimal dapat tercapai. Fungsi objektif yang digunakan adalah fungsi yang meminimalkan total jarak tempuh seluruh kendaraan seperti berikut ini:

Fungsi Tujuan

$$\text{Min} \sum_{i=0}^N \sum_{j=0}^N \sum_{k=1}^K c_{ij} x_{ijk} \quad (4.1)$$

Kendala:

$$\sum_{i=1}^N \sum_{k=1}^K x_{ijk} = 1 \text{ for } j = 1, 2, \dots, N \quad (4.2)$$

$$\sum_{j=1}^N \sum_{k=0}^K x_{ijk} = 1 \text{ for } i = 1, 2, \dots, N \quad (4.3)$$

Untuk kendala (4.2) dan (4.3) memastikan bahwa dalam sebuah rute, pelanggan hanya dilayani oleh sebuah kendaraan.

$$\sum_{i=0}^N x_{ijk} - \sum_{j=0}^N x_{jik} = 0 \text{ } j = 0, 1, \dots, N \quad \forall k \in [0, K - 1] \quad (4.4)$$

Kendala (4.4) menunjukkan bahwa jika kendaraan memasuki suatu titik (pelanggan), maka kendaraan tersebut harus keluar dari titik tersebut.

$$\sum_{i=0}^N d_i \sum_{j=0, j \neq i}^N x_{ijk} \leq q_k \quad \forall k \in [0, K - 1] \quad (4.5)$$

Kendala (4.5) menunjukkan bahwa jumlah *demand* tiap titik yang dikunjungi tidak boleh melebihi kapasitas kendaraan yang mengunjungi titik tersebut.

$$\sum_{i=0}^N \sum_{j=0, j \neq i}^N x_{ijk} (t_{ij} + s_i + w_i) \leq w_k \quad \forall k \in [0, K - 1] \quad (4.6)$$

$$t_0 = 0 \quad (4.7)$$

$$t_i + x_{ijk}(t_{ij} + s_i + w_i) \leq t_j \text{ } i \neq j; k \in [0, K - 1] \quad (4.8)$$

Kendala (4.6) merupakan kendala batasan travel time kendaraan (*time windows depot*). Kendala (4.7) dan kendala (4.8) memastikan bahwa pengantaran dilakukan dalam *time windows* pelanggan.

Keterangan:

K = Jumlah kendaraan
 N = Jumlah pelanggan

q_k = kapasitas kendaraan k
 s_j = *service time* di titik j

C_{ij} = jarak dari titik i ke titik j	C_i = pelanggan ke-i
t_i = waktu sampai di titik i	C_0 = depot
t_{ij} = waktu tempuh titik i ke titik j	d_i = permintaan di titik i
w_i = waktu tunggu di titik i	w_k = waktu kerja kendaraan
r_k = waktu kerja maksimum kendaraan k	

4. Evaluasi

Pada tahap ini, individu awal akan dievaluasi berdasarkan fungsi objektif tiap individu. Proses evaluasi ini bertujuan untuk mengetahui individu manakah yang memiliki total jarak tempuh terkecil pada generasi awal.

5. Proses mutasi

Proses mutasi ini merupakan proses untuk menghasilkan individu-individu baru hasil mutasi atau individu-individu mutan yang akan mengisi populasi vektor mutan. Tahapan dari proses ini adalah memunculkan tiga individu dari populasi awal secara acak (individu a, b, dan c). Setelah itu, mencari selisih atau different vector antara individu b dan c dan mengalikan selisih kedua individu tersebut dengan parameter kontrol mutasi (F) yang telah ditetapkan sebelumnya. Hasil pengalian tersebut selanjutnya dijumlahkan dengan individu a. Persamaan untuk proses mutasi ini adalah sebagai berikut:

$$\text{Individu mutan} = a + F(b - c) \quad (4.9)$$

Pada proses ini, penetapan nilai dari parameter kontrol mutasi (F) sangat penting karena akan mempengaruhi nilai dari individu-individu mutan yang terbentuk. Nilai F berkisar antara 0 sampai 2, dengan nilai efektif 0,4 sampai 1. Berdasarkan studi literatur yang dilakukan, nilai F yang sangat kecil akan membuat perpindahan vektor target atau individu awal yang sangat kecil bahkan terkesan seperti tidak terjadi perpindahan. Untuk nilai F yang lebih dari 1, perpindahan vektor target yang terjadi terlalu besar hingga melebihi batas fungsi objektif. Dengan demikian, akan

dilakukan percobaan untuk beberapa nilai F yang berada pada kisaran nilai efektifnya.

6. Proses pindah silang

Proses ini bertujuan untuk menghasilkan individu trial yang berasal dari rekombinasi dari individu mutan dan individu target. Pada proses ini, nilai parameter pindah silang (C_r) dan nilai bilangan acak yang dimunculkan (r) antara 0 sampai 1 sangat mempengaruhi pembentukan populasi individu trial. Nilai C_r yang telah ditetapkan sebelumnya akan dibandingkan dengan nilai r , jika nilai r lebih kecil atau sama dengan nilai C_r , maka nilai individu vektor mutan dimensi ke- i akan menjadi nilai individu vektor trial generasi ke- i . Sebaliknya, jika nilai r lebih besar dari pada nilai C_r maka nilai individu target dimensi ke- i akan menjadi nilai individu trial dimensi ke- i . Proses ini akan berlangsung hingga seluruh populasi individu trial terisi.

7. Seleksi

Pada tahap ini, individu target dan individu trial akan diseleksi untuk memasuki generasi selanjutnya. Proses penyeleksian ini dilakukan dengan membandingkan nilai fungsi objektif tiap individu. Fungsi objektif dalam penelitian ini adalah fungsi yang meminimumkan total jarak tempuh kendaraan. Oleh karena itu, individu target dan individu trial akan diseleksi dengan membandingkan total jarak tempuh kendaraan yang dimiliki tiap individu. Individu yang memiliki total jarak tempuh kendaraan yang paling kecil akan memasuki generasi selanjutnya. Dengan proses ini, tiap generasi baru akan memiliki nilai fungsi objektif lebih baik daripada generasi sebelumnya.

7. Terminasi

Terminasi merupakan tahapan dimana pencarian solusi optimum dihentikan. Kriteria terminasi yaitu jumlah iterasi maksimum dan waktu komputasi maksimum. Pada penelitian ini, kriteria terminasi yang

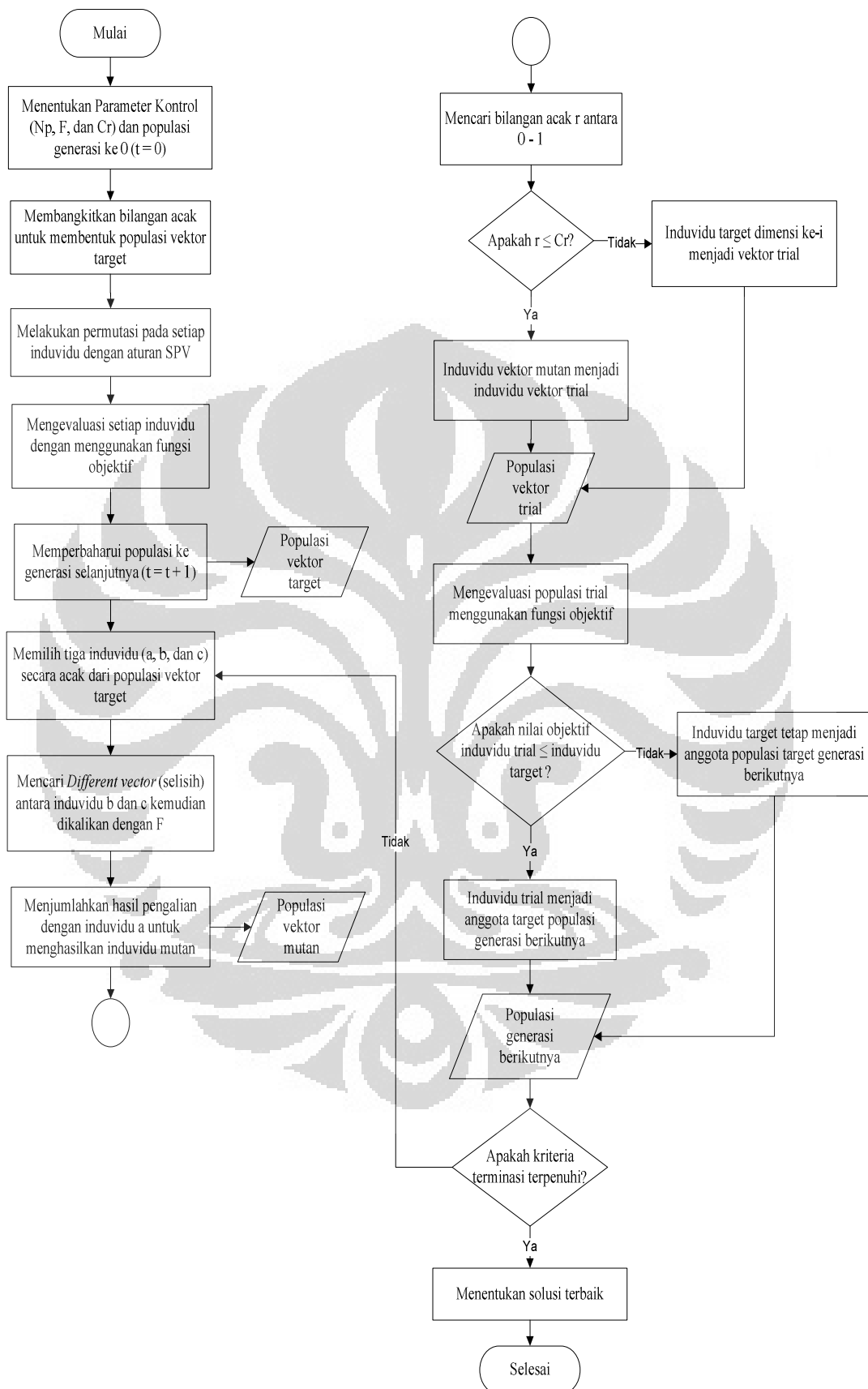
digunakan adalah jumlah iterasi maksimum sehingga proses pencarian solusi optimal akan dihentikan ketika jumlah iterasi maksimum telah dilakukan.

Tabel 4.2 Hasil Percobaan Penentuan Iterasi

Percobaan Ke-	Iterasi	Total Jarak	Total Waktu	Computation Time (detik)
1	100	1247	3035	21,77
2	200	1237	3022	27,48
3	300	1232	3015	34,65
4	400	1206	2981	38,11
5	500	1148	2903	46,21
6	600	1163	2923	53,30
7	700	1169	2931	66,59
8	800	1160	2919	71,83
9	900	1183	2950	74,80
10	1000	1129	2878	82,72
11	1500	1153	2910	117,19

Percobaan dilakukan sebanyak sebelas kali dengan parameter F dan CR yang tetap yaitu masing masing 0.6 dan 0.9 untuk mengetahui pada iterasi berapakah hasil yang terbaik dihasilkan. Berdasarkan tabel di atas angka yang bercetak tebal menunjukkan nilai jarak dan waktu yang paling baik, dimana nilai terbaik tersebut diperoleh pada percobaan 1000 iterasi. Asumsinya adalah dengan melakukan iterasi sebanyak 1000 kali, program akan dapat mencari solusi yang benar-benar paling baik. Namun, pemilihan kriteria dengan 1000 terminasi tentu saja membuat waktu running program menjadi lebih lama.

Berikut adalah digram alir tahap-tahap pengerjaan algoritma *Differential Evolution* secara keseluruhan mulai dari pembentukan solusi awal, mutasi, pindah silang, dan seleksi yang telah dijelaskan sebelumnya:



Gambar 4.1 Diagram Alir Pengerjaan Algoritma *Differential Evolution*
Universitas Indonesia

4.2 Verifikasi dan Validasi Program

Sebelum melakukan pengolahan data untuk mencari rute distribusi baru yang optimal dengan algoritma differential evolution, terlebih dahulu harus dilakukan verifikasi dan validasi program. Verifikasi merupakan tahapan untuk memeriksa apakah program yang dirancang untuk menyelesaikan permasalahan ini telah bekerja sesuai konsep yang dimaksud. Jika program dapat berkerja sesuai dengan konsep awal pembuatannya dan hasil yang diperoleh akan berbeda untuk setiap kali program dijalankan dan parameter kontrol dirubah, maka program tersebut telah terverifikasi. Setelah dilakukan verifikasi, tahap selanjutnya yaitu melakukan validasi terhadap program tersebut. Validasi bertujuan untuk mengetahui apakah program bekerja sesuai dengan fungsinya. Untuk melakukan validasi, dibutuhkan data-data dummy berupa jumlah pelanggan, permintaan pelanggan, dan matriks jarak pelanggan. Untuk data kecepatan kendaraan, kapasitas kendaraan, service time dan time windows yang digunakan sama untuk kasus pada penelitian ini.

Tabel 4.3 Data Matriks Jarak *Dummy*

Matriks Jarak <i>Dummy</i>							
Dari i ke j	1	2	3	4	5	6	
Depot	1	0	50	55	43	115	115
KWK	2	50	0	8	22	92	91
IAMI	3	54	7	0	26	95	95
ADM	4	72	41	26	0	77	76
HN	5	107	75	98	66	0	4
NSN	6	116	84	95	75	4	0

Tabel 4.4 Data Permintaan Pelanggan *Dummy*

Permintaan Pelanggan <i>Dummy</i>		
Nama Pelanggan	Permintaan	
	Volume (m ³)	Jumlah Box
KWK	16,98	80
IAMI	15,28	72
ADM	8,06	38
HN	6,37	30
NSN	16,55	78

Tabel 4.5 Parameter yang Digunakan Dalam Proses Validasi

Penetapan Parameter Kontrol	
NP	1 x Jumlah Pelanggan
F	0,6
Cr	0,9
Jumlah Iterasi	1

Dari data-data di atas, dilakukan run program dan hasil run program tersebut adalah rute distribusi dengan urutan 1-5-6-1-2-1-4-3-1 dengan total jarak 458,3 km dan menggunakan 2 truk dengan waktu tempuh masing-masing 397 menit dan 459 menit. Selanjutnya dilakukan perhitungan manual untuk validasi program. Langkah-langkah yang dilakukan yaitu:

1. Memeriksa populasi target.

Populasi target merupakan populasi awal yang berisi bilangan acak antara -1 sampai 1. Matriks Populasi target ini berukuran 5 x 5 (ukuran populasi x jumlah pelanggan) sehingga dalam populasi target ini terdapat 5 individu awal.

Tabel 4.6 Matriks Populasi Target

Matriks Populasi Target				
0,18749	0,14146	0,86171	0,48021	0,34633
0,92417	0,97764	0,09654	0,49965	0,30959
0,28892	0,24059	0,55024	0,59946	0,73129
0,77024	0,71233	0,98203	0,08401	0,87947
0,92683	0,16121	0,95846	0,42471	0,48413

2. Melakukan pengurutan pada setiap individu dari populasi target.

Urutan individu diperoleh dengan mengurutkan bilangan acak setiap baris individu pada tabel di atas dari nilai terkecil hingga nilai terbesar. Contoh pengurutan untuk individu pertama yang memiliki nilai bilangan acak 0,1874; 0,1414; 0,8617; 0,4802; 0,3463 sehingga jika dilakukan pengurutan, maka urutan pengiriman barang untuk individu pertama adalah 3-2-6-5-4. Pengurutan ini dilakukan untuk setiap individu sehingga terbentuk matriks populasi target yang telah dilakukan pengurutan nilai seperti berikut ini:

Tabel 4.7 Permutasi Populasi Target

Permutasi Populasi Target					
Individu 1	3	2	6	5	4
Individu 2	5	6	2	4	3
Individu 3	3	2	4	5	6
Individu 4	4	3	6	2	5
Individu 5	5	2	6	3	4

3. Pembentukan populasi mutan

Pembentukan populasi mutan ini diawali dengan mencari individu terbaik (a) dari populasi target dan 2 individu lain (b dan c) dari populasi target secara acak. Rumus yang digunakan untuk membentuk individu mutan ini adalah:

$$\text{Individu mutan} = a + (b-c)*F \quad (4.10)$$

Rumus di atas berlaku untuk tiap kolom dari individu mutan. Contoh, kolom pertama untuk individu mutan adalah 0,3812 diperoleh dari $0,2889 + (0,9241 - 0,7702) \times 0,6$ di mana nilai 0,2889; 0,9241; 0,7702 adalah nilai untuk kolom pertama dari individu a, b, dan c. Nilai 0,6 adalah nilai parameter kontrol untuk mutasi yang digunakan.

Tabel 4.8 Populasi Mutan

Populasi Mutan				
0,70938	0,65286	1,16891	0,01246	0,64849
0,86331	0,91816	0,28343	0,4281	0,07862
0,19656	0,0814	1,08153	0,35007	1,07322
0,19497	0,57126	0,56438	0,39503	0,9685
0,38128	0,39977	0,01895	0,84884	0,38937

4. Pembentukan populasi trial

Setelah populasi mutan diperoleh, maka pembentukan populasi trial dapat dilakukan. Populasi trial ini berasal dari individu dari populasi target atau individu dari populasi trial. Untuk memilih individu mana yang akan mengisi populasi trial, digunakan bilangan acak (r) antara 0 sampai 1 yang kemudian akan dibandingkan dengan parameter kontrol untuk pindah silang (Cr) sebesar 0,9. Jika nilai c lebih kecil atau sama

dengan 0,9 maka individu mutan akan memasuki populasi trial, jika r lebih besar dari 0,9 maka individu target yang memasuki populasi trial. Berikut matriks populasi trial:

Tabel 4.9 Populasi Trial

Populasi Trial				
0,70938	0,65286	1,16891	0,01246	0,64849
0,86331	0,91816	0,28343	0,4281	0,07862
0,19656	0,0814	1,08153	0,35007	0,73129
0,19497	0,57126	0,56438	0,39503	0,9685
0,38128	0,39977	0,01895	0,42471	0,38937

5. Melakukan pengurutan untuk tiap individu populasi trial
Cara yang digunakan sama dengan pengurutan untuk individu populasi target.

Tabel 4.10 Permutasi Populasi Trial

Permutasi Populasi Trial					
Individu 1	5	4	6	2	3
Individu 2	5	6	2	4	3
Individu 3	3	2	6	4	5
Individu 4	2	5	4	3	6
Individu 5	3	5	2	6	4

6. Mengubah Individu populasi target dan trial ke solusi TSP
Solusi TSP merupakan solusi yang mengurutkan pengiriman barang ke pelanggan mulai dari depot dan berakhir di depot. Urutan tiap solusi TSP berasal dari urutan pengantaran yang ada di setiap individu populasi trial. Contoh, solusi TSP untuk populasi trial individu pertama adalah 1-5-4-6-2-3-1.

Tabel 4.11 Populasi Solusi TSP Target

Populas Solusi TSP Target							
Individu 1	1	3	2	6	5	4	1
Individu 2	1	5	6	2	4	3	1
Individu 3	1	3	2	4	5	6	1
Individu 4	1	4	3	6	2	5	1
Individu 5	1	5	2	6	3	4	1

Tabel 4.12 Populasi Solusi TSP Trial

Populas Solusi TSP Trial							
Individu 1	1	5	4	6	2	3	1
Individu 2	1	5	6	3	4	2	1
Individu 3	1	3	2	6	4	5	1
Individu 4	1	2	5	4	3	6	1
Individu 5	1	3	5	2	6	4	1

7. Mengubah solusi TSP menjadi solusi VRP

Solusi VRP berasal dari Solusi TSP yang telah disisipkan depot atau angka 1 jika permintaan yang harus diangkut melebihi kapasitas truk. Contoh untuk Individu pertama Solusi TSP 1-5-4-6-2-3-1 dengan permintaan masing-masing titik 0, 30, 38, 78, 80 72, dan 0. Kapasitas kendaraan 110. Sehingga Solusi VRP untuk Individu Solusi TSP pertama adalah 1-5-4-1-6-1-2-1-3-1.

Tabel 4.13 Populasi Solusi VRP Awal

Populasi Solusi VRP Awal										
Individu 1	1	3	1	2	1	6	5	1	4	1
Individu 2	1	5	6	1	2	1	4	3	1	1
Individu 3	1	3	1	2	1	4	5	1	6	1
Individu 4	1	4	3	1	6	1	2	5	1	1
Individu 5	1	5	2	1	6	1	3	4	1	1

Tabel 4.14 Populasi Solusi VRP Trial

Populasi Solusi VRP Trial										
Individu 1	1	5	4	1	6	1	2	1	3	1
Individu 2	1	5	6	1	3	4	1	2	1	1
Individu 3	1	3	1	2	1	6	1	4	5	1
Individu 4	1	2	5	1	4	3	1	6	1	1
Individu 5	1	3	5	1	2	1	6	1	4	1

8. Menghitung jarak tempuh seluruh solusi VRP populasi target dan populasi trial. Untuk menghitung total jarak tiap individu solusi VRP populasi target maupun populasi trial, dibutuhkan matriks jarak. Penghitungan total jarak, untuk solusi VRP populasi target pertama

dengan urutan pengiriman barang 1-3-1-2-1-6-5-1-4-1 adalah 550 km. Setelah dilakukan perhitungan, jarak untuk tiap solusi VRP populasi target dan populasi trial adalah sebagai berikut:

Tabel 4.15 Perbandingan Total Jarak Solusi Target dan Solusi Trial

Solusi VRP Populasi Awal	Total Jarak	Solusi VRP Populasi Trial	Total Jarak
Individu 1	550	Individu 1	693
Individu 2	458	Individu 2	488
Individu 3	666	Individu 3	666
Individu 4	603	Individu 4	603
Individu 5	624	Individu 5	703

9. Proses seleksi

Proses seleksi ini dilakukan dengan membandingkan total jarak tiap individu populasi target dan trial. Individu solusi VRP pertama di populasi target akan dibandingkan dengan individu pertama dari solusi trial. Individu yang memiliki total jarak tempuh paling kecil akan memasuki generasi selanjutnya dan memasuki populasi target pada generasi selanjutnya. Pada kasus ini, individu dari populasi target memiliki total jarak yang lebih kecil daripada individu di populasi trial, sehingga individu pada populasi target akan menjadi individu populasi target generasi selanjutnya.

10. Proses terminasi

Kriteria terminasi pada kasus ini adalah setelah iterasi maksimal dikerjakan, dan jumlah iterasi maksimal adalah 1. Oleh karena itu, proses selanjutnya adalah menentukan solusi VRP yang memiliki total jarak terkecil dari populasi individu target generasi terakhir. Total jarak terkecil adalah 458 km yang dimiliki oleh individu ke 2 dari solusi VRP populasi target. Dengan urutan pengantaran 1-5-6-1-2-1-4-3-1. Karena penelitian ini memiliki time windows, maka untuk melakukan pengiriman barang ke seluruh pelanggan tidak boleh lebih dari 8 jam atau 480 menit. Sehingga total truk yang digunakan adalah 2 dengan waktu

kerja 397 menit untuk truk 1 dengan rute 1-5-6-1 dan 459 menit untuk truk 2 dengan rute 1-2-1-4-3-1.

- Setelah dilakukan pengujian, ternyata nilai pada perhitungan manual sama dengan perhitungan program. Dengan demikian, program ini telah tervalidasi.

4.3 Input Data

Data yang digunakan untuk menyelesaikan masalah dalam penelitian ini adalah data jarak dari depot ke pelanggan dan jarak antar pelanggan, data waktu tempuhnya, dan permintaan tiap pelanggan. Data matriks jarak dan permintaan pelanggan dapat dilihat pada tabel 4. 11 dan 4.12.

Untuk pelanggan yang memiliki permintaan lebih dari kapasitas maksimal truk, maka permintaan pelanggan akan dibagi menjadi beberapa kali pengiriman hingga kapasitas maksimal truk terbesar terisi penuh. Pemecahan permintaan pelanggan (*split delivery*) ini dilakukan dengan mengisi penuh kapasitas maksimal truk paling besar karena biaya yang dikeluarkan akan lebih kecil daripada menggunakan truk dengan kapasitas yang lebih kecil.

Tabel 4.16 Matriks Jarak Pelanggan

Dari ke j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
Depot	1	0	50	55	54	54	55	55	55	56	63	43	43	68	94	88	82	6	24	115	115	6	
KWK	2	50	0	8	8	7	8	8	8	6	5	39	22	22	44	70	64	45	50	45	92	91	49
IAMI	3	54	7	0	3	3	1	0,4	1	10	8	43	26	26	48	74	68	42	53	49	95	95	53
AHM1	4	52	6	2	0	3	1	2	1	8	6	41	24	24	46	72	66	45	52	47	93	93	51
GS STR	5	52	6	2	3	0	1	2	1	8	6	41	24	24	46	72	66	46	52	47	94	93	51
GM	6	54	7	2	2	2	0	1	1	10	8	43	26	26	48	74	68	42	53	49	95	95	53
TAM PIO STR	7	54	7	0,4	2	1	1	0	1	10	7	42	25	25	47	73	67	42	53	48	95	95	52
DENSO STR	8	54	7	1	2	1	1	0	10	8	43	26	26	48	74	68	42	53	49	95	95	53	
NUSMET	9	53	4	11	8	11	12	11	0	2	31	15	15	36	62	56	48	51	47	84	83	51	
AHM2	10	53	6	8	8	8	9	8	6	2	0	33	17	17	38	64	58	50	53	49	86	86	53
AHM3	11	62	33	43	43	43	44	44	44	30	32	0	23	23	12	38	32	71	64	32	59	59	63
ADM SPD	12	72	41	26	26	26	27	27	27	15	17	24	0	0	29	55	49	53	45	40	77	76	45
TAM SPD	13	72	40	26	26	26	26	26	26	14	17	24	0	0	29	55	49	53	45	40	76	76	44
SANENG	14	73	41	52	51	51	52	52	52	39	41	16	32	32	0	34	28	79	72	42	56	56	72
GS KRW	15	96	64	75	74	74	75	75	75	62	64	39	55	55	32	0	19	102	95	68	31	31	95
TAM PIO KRW	16	89	57	68	68	68	68	68	68	55	57	33	48	48	25	19	0	95	89	61	40	40	88
SMU	17	85	57	39	39	39	40	39	40	56	45	69	52	52	74	100	94	0	79	75	121	121	79
TGSSI	18	6	47	51	50	50	51	51	51	51	53	59	40	40	64	91	85	78	0	21	112	112	0,3
GEMITA	19	30	36	40	40	40	41	40	41	33	35	20	25	25	30	56	50	68	31	0	77	77	30
HINO	20	107	75	98	86	86	86	86	86	72	75	51	66	66	43	34	42	125	118	91	0	4	118
NISSAN	21	116	84	95	94	95	95	95	95	82	84	60	75	75	52	31	39	122	115	88	4	0	115
AWP2	22	6	47	51	51	51	53	52	53	51	53	63	45	44	72	95	88	79	0,3	24	118	115	0

Tabel 4.17 Daftar Permintaan Pelanggan

No	Kode Pelanggan	Permintaan	
		Volume (m ³)	Jumlah Box
1	KWK	15,28	72
2	IZS	2,55	12
3	AHM 1	68,97	325
4	GS STR	2,76	13
5	PPL	2,55	12
6	TAM PIO STR	11,88	56
7	DS	4,88	23
8	NMT	5,73	27
9	AHM 2	177,85	838
10	AHM 3	266,56	1.256
11	ADM	2,12	10
12	TAM SPD	3,40	16
13	SNG	67,91	320
14	GS KRW	21,22	100
15	TAM KRW	3,61	17
16	SMU	3,40	16
17	TGGSI	16,98	80
18	GMT	7,22	34
19	HN	4,24	20
20	NSN	3,82	18
21	AWP 2	19,52	92

Berikut permintaan pelanggan melebihi kapasitas maksimal truk (230 box):

Tabel 4.18 Daftar Permintaan Pelanggan yang Melebihi Kapasitas Kendaraan

No	Kode Pelanggan	Permintaan	
		Volume (m ³)	Jumlah Box
3	AHM 1	68,97	325
9	AHM 2	177,85	838
10	AHM 3	266,56	1.256
13	SANENG	67,91	320

Untuk pelanggan AHM 1, Permintaan dipecah menjadi 230 dan 95. Untuk pelanggan AHM 2, permintaan dipecah menjadi 230, 110, 110, 110, 110, 110, dan 58 box. Pelanggan AHM 3 dipecah menjadi 110 sebanyak 9 kali pengiriman 230, dan 36 box. Selanjutnya permintaan SANENG akan dipecah menjadi 230 dan 90. Setelah permintaan pelanggan yang melebihi kapasitas maksimal truk, maka permintaan pelanggan yang akan dimasukkan ke dalam program adalah permintaan yang lebih kecil dari kapasitas truk karena permintaan yang jumlahnya sama

dengan kapasitas maksimal truk hanya dapat melakukan pengiriman ke satu pelanggan dan kembali ke depot lagi.

Time windows untuk melakukan pengiriman barang ke pelanggan ini adalah 8 jam atau 480 menit. Dengan waktu loading dan unloading yang telah dijabarkan di bab sebelumnya. Kecepatan kendaraan diasumsikan konstan yaitu 45 km/jam sehingga waktu tempuh kendaraan dari satu titik ke titik lainnya dapat diketahui.

4.4 Hasil Pengolahan Data

Setelah program tervalidasi, maka data yang telah dijelaskan sebelumnya akan dimasukkan dan kemudian program di-run untuk menghasilkan rute distribusi baru. Hasil dalam pengolahan data ini adalah rute distribusi untuk melayani seluruh pelanggan dengan urutan pengiriman pada rute tersebut, total jarak tempuh kendaraan, jumlah kendaraan yang digunakan, dan rute untuk tiap kendaraan. Program yang ini akan di-run sebanyak 5 kali dan hasil pengolahan tersebut akan dihitung biaya transportasi yang harus dikeluarkan. Rute distribusi yang memiliki total biaya transportasi terkecil akan diambil menjadi solusi terbaik dalam penelitian ini. Berikut ini menunjukkan hasil run program sebanyak 5 kali:

Tabel 4.19 Hasil *Run* 1

Hasil Run 1				
Jumlah Truk yang digunakan = 6		Jumlah Perjalanan= 9		
No Truk	Rute tiap truk	Waktu	Jarak	Biaya Variable
1	1 11 20 21 16 1	405	254	Rp165.165
2	1 19 5 7 1 22 1	374	177	Rp114.920
3	1 6 4 1 8 3 2 1	441	222	Rp143.975
4	1 17 14 1	377	228	Rp148.395
5	1 12 15 1	333	194	Rp126.100
6	1 13 9 10 1 18 1	299	125	Rp81.185
Total		2228	1200	Rp779.740
Biaya Tetap untuk 9 kali perjalanan				Rp2.025.000
Total Biaya				Rp2.804.740

Tabel 4.20 Hasil Run 2

Hasil Run 2												
Jumlah Truk yang digunakan = 5			Jumlah Perjalanan = 9									
No Truk	Rute tiap truk				Waktu	Jarak	Biaya Variable					
1	1	12	15	1	18	1	371	177	Rp114.790			
2	1	17	10	11	1	22	1	453	234	Rp152.100		
3	1	8	7	6	5	1	2	9	1	435	220	Rp143.065
4	1	4	3	1	13	14	1	480	255	Rp165.425		
5	1	19	21	20	16	1	381	236	Rp153.595			
Total					2120	1122	Rp728.975					
Biaya Tetap untuk 9 kali perjalanan							Rp2.025.000					
Total Biaya							Rp2.753.975					

Tabel 4.21 Hasil Run 3

Hasil Run 3												
Jumlah Truk yang digunakan = 5			Jumlah Perjalanan = 9									
No Truk	Rute tiap truk				Waktu	Jarak	Biaya Variable					
1	1	22	1	17	7	3	6	5	1	390	187	Rp121.745
2	1	2	19	1	9	10	8	1	470	244	Rp158.340	
3	1	12	15	1	18	1	411	206	Rp133.900			
4	1	13	14	1	4	1	475	251	Rp163.020			
5	1	11	20	21	16	1	405	254	Rp165.165			
Total					2151	1142	Rp742.170					
Biaya Tetap untuk 9 kali perjalanan							Rp2.025.000					
Total Biaya							Rp2.767.170					

Tabel 4.22 Hasil Run 4

Hasil Run 4												
Jumlah Truk yang digunakan = 5			Jumlah Perjalanan = 9									
No Truk	Rute tiap truk				Waktu	Jarak	Biaya Variable					
1	1	17	10	9	1	18	1	391	194	Rp126.295		
2	1	3	7	8	6	1	19	2	1	438	221	Rp143.455
3	1	5	4	1	12	14	1	480	254	Rp164.905		
4	1	15	1	22	1	405	202	Rp130.975				
5	1	13	20	21	16	11	1	415	257	Rp167.050		
Total					2130	1127	Rp732.680					
Biaya Tetap untuk 9 kali perjalanan							Rp2.025.000					
Total Biaya							Rp2.757.680					

Tabel 4.23 Hasil *Run* 5

Hasil Run 5												
Jumlah Truk yang digunakan = 6			Jumlah Perjalanan = 10									
No Truk	Rute tiap truk				Waktu	Jarak	Biaya Variable					
1	1	12	14	1	4	1	472	251	Rp163.020			
2	1	3	7	6	17	1	18	1	389	194	Rp126.035	
3	1	15	1	22	1				405	202	Rp130.975	
4	1	13	9	10	1	8	5	2	1	444	225	Rp146.250
5	1	19	1						116	54	Rp35.230	
6	1	11	16	21	20	1			395	246	Rp160.160	
Total					2221	1172	Rp761.670					
Biaya Tetap untuk 10 kali perjalanan							Rp2.250.000					
Total Biaya							Rp3.011.670					

Setelah program di-*run* sebanyak lima kali, terlihat bahwa hasil yang terbaik (memiliki total biaya transportasi terkecil) adalah hasil *run* 2. Nilai fungsi objektif, yaitu total jarak tempuh kendaraan terkecil adalah 1122 km, jumlah kendaraan yang digunakan sebanyak 5 buah dengan 9 kali perjalanan, dan waktu tempuh tiap kendaraan tidak lebih dari 8 jam. Dari hasil *run* terbaik yang diperoleh, rute pengiriman barang akan digabungkan dengan rute pengiriman untuk kendaraan yang terisi penuh sesuai dengan kapasitas terbesar truk dimana truk hanya mengunjungi satu pelanggan dalam setiap kali perjalanan.

4.5 Analisis

4.5.1 Analisis Metode

Metode yang digunakan untuk penyelesaian permasalahan optimasi rute distribusi harian pada sebuah perusahaan pembuatan komponen otomotif ini adalah algoritma Differential Evolution. Algoritma Differential Evolution merupakan salah satu algoritma optimasi yang termasuk ke dalam kelompok algoritma Evolusioner (EA). Algoritma Differential Evolution ini memiliki konsep yang terinspirasi dari teori evolusi biologi dengan cara kerja yang terdiri dari tahap reproduksi, mutasi, rekombinasi, dan seleksi.

Populasi awal (generasi ke nol) untuk metode ini dibentuk dengan membangkitkan bilangan acak, sedangkan populasi berikutnya merupakan hasil

evolusi dari vektor-vektor yang telah melalui tahap reproduksi, mutasi, rekombinasi, dan seleksi. Untuk memasuki generasi selanjutnya, individu populasi target akan dibandingkan dengan individu trial hasil rekombinasi berdasarkan fungsi objektifnya. Fungsi objektif dalam penelitian ini yaitu fungsi yang meminimumkan total jarak, sehingga individu yang memiliki total jarak lebih kecil akan memasuki generasi selanjutnya. Dengan demikian, nilai fungsi objektif generasi selanjutnya akan lebih baik dari pada generasi sebelumnya sehingga proses pencarian solusi optimal lebih cepat diperoleh.

Algoritma DE memiliki struktur yang lebih sederhana dari pada algoritma evolusioner lainnya karena parameter kontrol yang digunakan relatif lebih sedikit. Selain itu, DE juga memiliki sifat tangguh yaitu dapat menghasilkan solusi yang tidak jauh berbeda untuk setiap kali dilakukan percobaan. Hal ini terjadi karena kondisi konvergen atau telah dicapai. Keunggulan lainnya yaitu untuk menghasilkan solusi optimal dengan algoritma differential evolution ini, waktu yang dibutuhkan relatif lebih singkat dari pada menggunakan algoritma lainnya.

Dalam penggunaannya, algoritma ini juga memiliki beberapa kelemahan. Salah satunya, yaitu tiap kali dilakukan pengulangan percobaan dengan parameter kontrol yang sama hasil yang diperoleh akan selalu berbeda walaupun tidak signifikan. Hal ini disebabkan karena dalam proses pembentukan populasi awal, populasi mutan, dan populasi trial menggunakan bilangan acak yang akan berbeda tiap kali program dijalankan. Sulitnya menentukan parameter kontrol yang akan digunakan dalam penelitian ini juga menjadi kelemahan DE.

4.5.2 Analisis Program

Secara umum, program yang telah dibuat untuk menyelesaikan permasalahan VRP dengan algoritma *Differential Evolution* ini dapat berjalan dengan baik. Solusi akhir yang dihasilkan selalu lebih baik daripada solusi awal. Selain itu solusi rute optimal yang dihasilkan juga memberikan perbaikan pada rute yang selama ini digunakan oleh pihak perusahaan. Hal tersebut ditandai dengan hasil jarak tempuh perjalanan yang lebih kecil dan juga penurunan jumlah kendaraan yang digunakan.

Kelemahan dari program ini adalah waktu running program yang relatif lama. Waktu running program dipengaruhi oleh kriteria terminasi yang telah ditetapkan. Pada program ini, jumlah iterasi yang harus dikerjakan 100 iterasi. Hal ini akan menyebabkan program membutuhkan waktu running yang lebih lama karena semakin banyak iterasi yang harus dilakukan maka akan semakin lama waktu yang dibutuhkan. Untuk mengurangi waktu running tersebut, beberapa cara yang dapat dilakukan yaitu dengan menurunkan jumlah iterasi dan menetapkan waktu running maksimum. Namun, jika jumlah iterasi diturunkan atau waktu running yang ditetapkan terlalu singkat resiko yang dapat terjadi adalah output dari program ini tidak terlalu baik.

Selain waktu running yang relatif lama, program ini juga memiliki beberapa kekurangan. Salah satunya program tidak dapat menampilkan jumlah box yang diangkut setiap truk. Untuk mengetahui jumlah box yang diangkut tiap kendaraan, harus dilakukan perhitungan secara manual. Selain itu, biaya untuk penggunaan kendaraan juga tidak dapat ditampilkan program sehingga harus dilakukan perhitungan manual juga. Program ini menggunakan asumsi kecepatan kendaraan konstan sehingga jika ingin diterapkan di perusahaan, program harus memiliki data kecepatan yang berbeda-beda sesuai kondisi jalan waktu tempuh kendaraan sesuai dengan kondisi sebenarnya. Untuk jarak antar titik, data diperoleh dari google maps sehingga tidak sama dengan kondisi sebenarnya. Jarak antar titik yang sebenarnya dapat diketahui dengan menghitung odometer truk dari suatu titik ke titik lainnya.

Penulis menyadari bahwa masih ada beberapa kekurangan dari program yang telah dibuat ini. Untuk penggunaannya di perusahaan, masih banyak yang harus diperbaiki, seperti membuat tampilan solusi yang lebih menarik dan hasil solusi yang langsung menunjukkan kode pelanggan pada urutan rute distribusi. Untuk melakukan hal tersebut diperlukan tambahan kode pemrograman yang lebih lanjut pada program ini.

4.5.3 Analisis Hasil Optimasi

Pada bagian ini, hasil optimasi rute distribusi akan dibandingkan dengan rute distribusi yang digunakan perusahaan saat ini. Analisa yang akan dibahas

meliputi analisa rute dan jarak tempuh, analisa utilitas kendaraan, dan analisa biaya pengiriman.

4.5.3.1 Analisis Usulan Rute Distribusi

Secara umum rute distribusi usulan yang diperoleh dari hasil optimasi lebih baik dari pada rute distribusi yang diterapkan saat ini. Tidak optimalnya rute distribusi yang diterapkan perusahaan saat ini disebabkan karena perusahaan masih sering menggunakan truk dengan kapasitas kecil sehingga membutuhkan waktu dan jarak yang lebih besar untuk melakukan pengiriman barang ke pelanggan. Berikut akan ditampilkan rute distribusi yang diterapkan perusahaan dan rute distribusi usulan.

Tabel 4.24 Rute, Jarak, dan Waktu Tempuh Perusahaan Saat Ini

No	Rute	Jarak (km)	Waktu (menit)
1	0-NSN-TAMK-GSK-HN-0	311	480
2	0-GSK-AHM3-0	196	334
3	0-TAMPS-0	108	193
4	0-AWP2-TGGSI-0	12	70
5	0-DS-TAMPS-ISZ-GSS-0	111	201
6	0-AHM1-0	106	211
7	0-AHM1-0	106	209
8	0-AHM2-0	110	207
9	0-AHM3-0	125	233
10	0-AHM3-0	125	241
11	0-AHM3-0	125	232
12	0-GMT-AHM3-0	107	196
13	0-KWK-NMT-0	110	216
14	0-SNG-0	141	289
15	0-AHM3-0	125	283
16	0-AHM2-0	110	260
17	0-AHM2-0	110	248
18	0-TAMSPD-AHM3-ADM-0	163	263
19	0-AWP2-0	12	67
20	0-AHM2-0	110	206
21	0-PPL-AHM1-0	108	218
22	0-TGGSI-0	12	69
23	0-SMU-SNG-0	228	354

Tabel 4.24 Rute, Jarak, dan Waktu Tempuh Perusahaan Saat Ini (Lanjutan)

No	Rute	Jarak (km)	Waktu (menit)
24	0-AHM3-0	125	221
25	0-SNG-0	141	262
26	0-AHM3-0	125	237
27	0-AHM3-0	125	229
28	0-AHM3-0	125	238
29	0-GSK-0	190	297
30	0-AHM3-0	125	240
31	0-AHM3-0	125	240
32	0-AHM3-0	125	237
33	0-AHM2-0	110	216
34	0-AHM2-0	110	214
35	0-AHM2-0	110	216
Total		4307	8130

Tabel 4.25 Rute, Jarak, dan Waktu Tempuh Hasil Optimasi

No	Rute	Jarak (km)	Waktu (menit)
1	0-ADM-GSK-0	165	293
2	0-SNG-AHM2-AHM3-0	222	371
3	0-DS-TAMPS-PPL-GSS-0	110	218
4	0-KWK-NMT-0	110	216
5	0-AHM1-IZS-0	110	219
6	0-SNG-TAMSPD-0	145	265
7	0-TGGSI-0	12	78
8	0-GMT-NSN-HN-TAMK-0	236	381
9	0-AWP2-0	12	82
10	0-AMH1-0	106	263
11	0-AHM2-0	110	268
12	0-AMH3-0	125	289
13	0-SNG-0	141	310
14	0-AHM2-0	110	220
15	0-AHM2-0	110	220
16	0-AHM2-0	110	220
17	0-AHM2-0	110	220
18	0-AHM2-0	110	220
19	0-AMH3-0	125	241
20	0-AMH3-0	125	241
21	0-AMH3-0	125	241

Tabel 4.25 Rute, Jarak, dan Waktu Tempuh Hasil Optimasi (Lanjutan)

No	Rute	Jarak (km)	Waktu (menit)
22	0-AMH3-0	125	241
23	0-AMH3-0	125	241
24	0-AMH3-0	125	241
25	0-AMH3-0	125	241
26	0-AMH3-0	125	241
27	0-AMH3-0	125	241
Total		3279	6525

Dapat dilihat pada tabel rute pengiriman barang di atas, rute pengiriman yang diterapkan perusahaan memiliki jumlah perjalanan yang lebih banyak dibandingkan dengan rute pengiriman hasil optimasi. Hal ini disebabkan karena pada rute pengiriman saat ini perusahaan masih banyak menggunakan kendaraan dengan kapasitas yang lebih kecil sehingga untuk melakukan pengiriman barang ke pelanggan diperlukan beberapa kali perjalanan dan hal tersebut akan mengakibatkan jarak tempuh kendaraan akan bertambah. Dari rute usulan, diperoleh jarak tempuh yang lebih kecil dan hal tersebut akan mengurangi waktu tempuh kendaraan yang digunakan.

Tabel 4.26 Perbandingan Total Jarak dan Waktu Tempuh

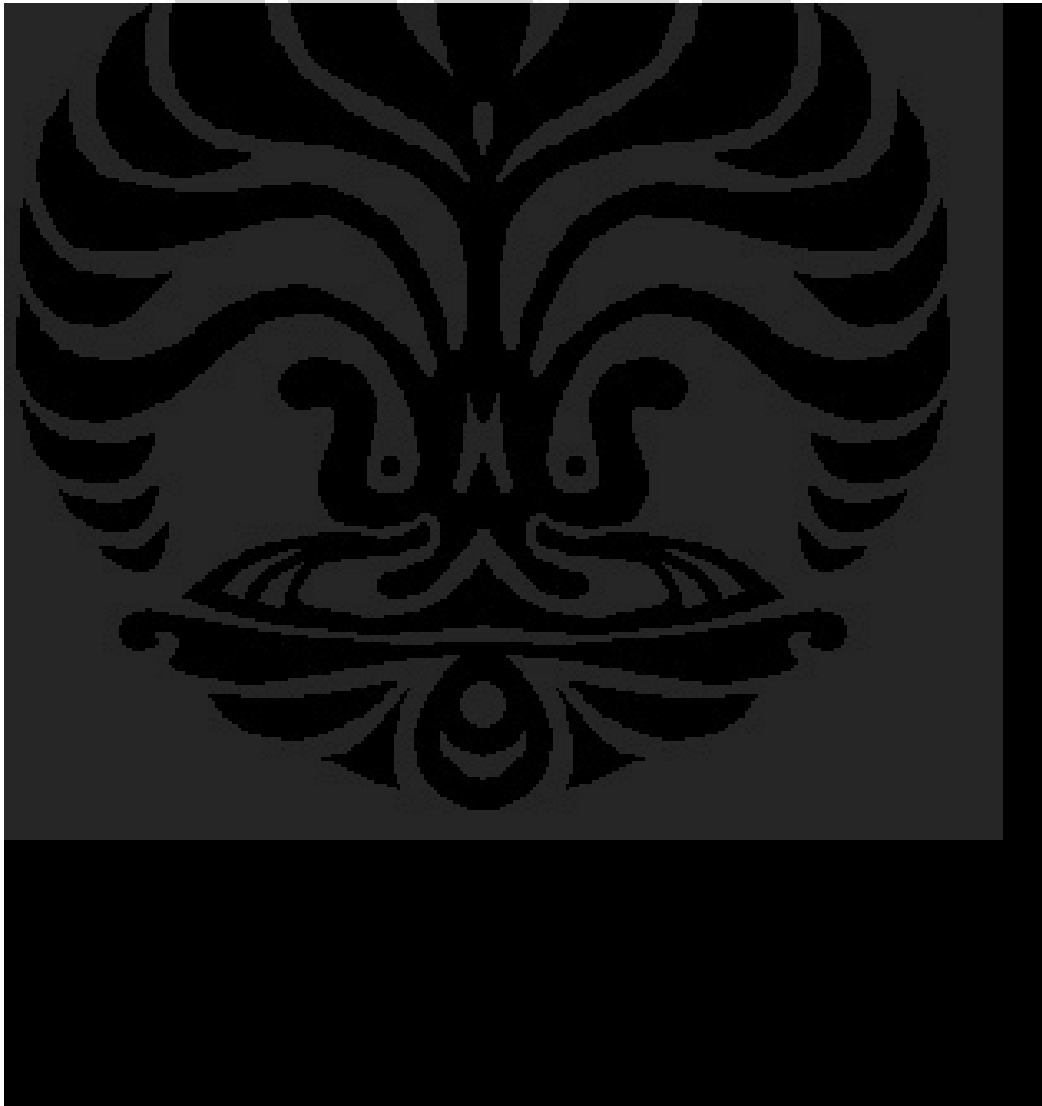
Variabel	Rute Distribusi Saat Ini	Rute Distribusi Usulan
Total Jarak Tempuh	4307 km	3279 km
Total Waktu Tempuh	8130 menit	6525 menit

Jika dilihat secara umum, total jarak tempuh untuk rute distribusi saat ini adalah 4.307 km dan total jarak tempuh untuk rute distribusi usulan adalah 3.279 km. Dengan demikian, telah terjadi penurunan total jarak tempuh kendaraan sebesar 1.028 km atau sebesar 23,87%. Pengurangan jarak tempuh sebanyak ini dapat dikatakan relatif besar. Total kendaraan yang digunakan juga berkurang karena jumlah perjalanan truk juga berkurang.

4.5.3.2 Analisis Utilitas Truk

Salah satu cara yang dapat dilakukan untuk mengurangi biaya transportasi yaitu dengan melakukan optimasi utilitas truk. Optimasi utilitas truk memiliki peranan yang sangat penting karena berdasarkan utilitas truk ini dapat diketahui efisien atau tidaknya penggunaan sebuah truk. Untuk lebih jelasnya, berikut ini merupakan perbandingan utilitas kendaraan antara rute yang digunakan perusahaan saat ini dengan rute usulan yang dihasilkan dari optimasi dengan algoritma *Differential Evolution*.

Tabel 4.27 Perbandingan Utilitas Truk Rute Saat Ini dan Rute Usulan



Berdasarkan perhitungan utilitas truk di atas, dapat dilihat bahwa persentasi utilitas rata-rata truk pada rute distribusi usulan lebih besar dari utilitas truk pada rute distribusi yang diterapkan perusahaan saat ini. Peningkatan utilitas truk rata-rata adalah 8%. Peningkatan utilitas truk tersebut sangat baik karena dengan meningkatnya utilitas truk, maka peluang untuk mengurangi rute kendaraan semakin besar.

4.5.3.3 Analisis Biaya

Dalam melakukan pengiriman barang ke pelanggannya, perusahaan menggunakan jasa logistik yaitu penyewaan truk. Perusahaan harus mengeluarkan biaya untuk penggunaan truk per satu kali perjalanan (mulai dari depot sampai ke depot lagi). Selain biaya tetap untuk satu kali perjalanan tersebut, perusahaan juga harus membayar biaya penyewaan truk berdasarkan jarak tempuh truk dalam satu kali perjalanan. Variabel biaya ini merupakan variabel yang berbanding lurus dengan jarak tempuh kendaraan, dimana semakin besar jarak tempuh kendaraan, maka biayanya juga akan semakin besar. Berikut ini merupakan perbandingan antara biaya pengiriman barang pada rute yang saat ini digunakan oleh pihak perusahaan dan rute usulan hasil optimasi.

Tabel 4.28 Perbandingan Biaya Sewa Truk Rute Saat Ini dan Rute Usulan

Biaya Pengiriman	
Rute Saat Ini	Rute Usulan
Rp 10.532.515	Rp 8.727.035

Dari perbandingan biaya sewa kendaraan di atas, dapat dilihat bahwa penurunan biaya sewa truk dalam satu hari. Biaya sewa truk yang dikeluarkan oleh perusahaan saat ini sebesar Rp. 10.532.515 sedangkan biaya sewa truk yang dikeluarkan setelah dilakukan optimasi sebesar Rp. 8.727.035 Dengan demikian, rute usulan dapat menghemat pengeluaran biaya sewa truk sebesar Rp. 1.805.480 per harinya atau sebesar 17%. Perhitungan optimasi dengan menggunakan algoritma *Differential Evolution* ini ternyata memberikan hasil rute distribusi yang lebih sedikit, dengan total jarak tempuh dan waktu tempuh yang lebih kecil, serta peningkatan utilitas truk.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian penentuan rute distribusi yang optimal pada perusahaan pembuatan komponen otomotif dengan algoritma *Differential Evolution* menggunakan bahasa pemrogramana Matlab, diperoleh beberapa kesimpulan sebagai berikut:

- Penentuan rute distribusi menggunakan algoritma *Differential Evolution* ini menghasilkan rute distribusi usulan yang optimal dan lebih baik dari pada rute distribusi yang diterapkan perusahaan saat ini. Total jarak tempuh pada rute distribusi yang diterapkan perusahaan saat ini adalah 4.307 km, dan penurunan total jarak tempuh pada rute distribusi usulan sebesar 1.028 km atau sebesar 23,87%.
- Karena total jarak tempuh kendaraan berbanding lurus dengan total waktu tempuh kendaraan. Oleh karena itu, total waktu tempuh kendaraan juga mengalami penurunan dari total 8.130 menit menjadi 6.525 menit, atau mengalami penurunan sebesar 19,74%.
- Utilitas rata-rata untuk satu truk pada penggunaan rute usulan adalah sebesar 89%, sedangkan pada rute yang digunakan perusahaan saat ini hanya sebesar 81%. Dengan demikian, rute usulan dapat meningkatkan utilitas kendaraan sebesar 8%.
- Biaya pengiriman pada rute usulan adalah sebesar Rp. 10.532.515,-. Penurunan biaya transportasi pengiriman yang dihasilkan adalah sebesar Rp. 1.805.480 atau sebesar 17%.

5.2 Saran

Berikut adalah beberapa saran untuk penelitian selanjutnya

- Untuk penggunaan algoritma *Differential Evolution*, perlu dilakukan uji coba untuk pembentukan solusi awal dengan cara heuristik lainnya. Selain pembentukan solusi awal, studi parameter juga perlu dilakukan. Sehingga program dapat menghasilkan solusi yang lebih baik.

- Program yang dibuat hendaknya memiliki tampilan solusi yang lebih menarik dan mudah dimengerti. Program juga hendaknya dihubungkan dengan database seperti jumlah dan jenis kendaraan, dan kendaraan yang digunakan memiliki id atau nomor truk sehingga program dapat digunakan untuk membuat jadwal pengiriman barang.

Untuk merencanakan rute distribusi untuk pengiriman barang ke pelanggan, ada beberapa saran yang dapat dilakukan perusahaan, yaitu:

- Dalam merencanakan rute distribusi hendaknya pihak perusahaan memperhatikan utilitas dari tiap kendaraan yang digunakan, sehingga nantinya akan dapat menghasilkan rute distribusi yang lebih efisien.
- Untuk melakukan perencanaan rute distribusi harian, pihak perusahaan dapat menggunakan program yang telah dibuat pada penelitian ini dengan melakukan beberapa perbaikan data jarak, waktu tempuh, dan *service time* yang disesuaikan dengan kondisi sebenarnya.

DAFTAR REFERENSI

- Ballou, R.H. (2004). *Business logistics management* (5th ed). New Jersey: Prentice-Hall Inc.
- Dreo, J., Petworsky, A., & Taillard, E.D. (2006). *Metaheuristics for hard optimization*. Berlin: Springer-Verlag Berlin Heidelberg.
- Homberger, J., & Gehring, H. (2004). A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operation Research*, 220 - 238.
- Karaboga, D. & Okdem, S. (2004). A simple and global optimization algorithm for engineering problems: Differential Evolution algorithm. *Turkey Journal Engineering*, 12, 1-8.
- Pedersen, E.H, Magnus (2010). *Good Parameters for differential evolution*. Hvas Laboratories. Technical Report no. HL1002
- Poot, A., Kant, G., Wagelmans, A.P.M. (2002). A saving based method for real life vehicle routing problem. *Journal of The Operational Research Society*, hal. 57 – 68.
- Price, K.V, Storn, M.R., & Lampinen, J.A. (2005) *Differential evolution: a practical approach to global optimization*. California: Springer.
- Storn, R. and Price, K. (1997), 'Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces', *Journal of Global Optimization*, 11, pp. 341–359.
- Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. Philadelphia: Society for Industrial and Applied Mathematics.
- Microsoft Encarta Premium 2006, Algorithm. (2005).Microsoft

Lampiran 1 Matriks Jarak dan Matriks Waktu

Matriks Jarak Antar Pelanggan

Dari i ke j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
Depot	1	0	50	55	54	54	55	55	55	56	56	63	43	43	68	94	88	82	6	24	115	115	6
KWK	2	50	0	8	8	7	8	8	8	6	5	39	22	22	44	70	64	45	50	45	92	91	49
IAMI	3	54	7	0	3	3	1	0,4	1	10	8	43	26	26	48	74	68	42	53	49	95	95	53
AHM1	4	52	6	2	0	3	1	2	1	8	6	41	24	24	46	72	66	45	52	47	93	93	51
GS STR	5	52	6	2	3	0	1	2	1	8	6	41	24	24	46	72	66	46	52	47	94	93	51
GM	6	54	7	2	2	2	0	1	1	10	8	43	26	26	48	74	68	42	53	49	95	95	53
TAM PIO STR	7	54	7	0,4	2	1	1	0	1	10	7	42	25	25	47	73	67	42	53	48	95	95	52
DENSO STR	8	54	7	1	2	1	1	1	0	10	8	43	26	26	48	74	68	42	53	49	95	95	53
NUSMET	9	53	4	11	8	11	12	11	12	0	2	31	15	15	36	62	56	48	51	47	84	83	51
AHM2	10	53	6	8	8	8	9	8	6	2	0	33	17	17	38	64	58	50	53	49	86	86	53
AHM3	11	62	33	43	43	43	44	44	44	30	32	0	23	23	12	38	32	71	64	32	59	59	63
ADM SPD	12	72	41	26	26	26	27	27	27	15	17	24	0	0	29	55	49	53	45	40	77	76	45
TAM SPD	13	72	40	26	26	26	26	26	14	17	24	0	0	0	29	55	49	53	45	40	76	76	44
SANENG	14	73	41	52	51	51	52	52	39	41	16	32	32	0	34	28	79	72	42	56	56	72	
GS KRW	15	96	64	75	74	74	75	75	75	62	64	39	55	55	32	0	19	102	95	68	31	31	95
TAM PIO KRW	16	89	57	68	68	68	68	68	55	57	33	48	48	25	19	0	95	89	61	40	40	88	
SMU	17	85	57	39	39	39	40	39	40	56	45	69	52	52	74	100	94	0	79	75	121	121	79
TGSSI	18	6	47	51	50	50	51	51	51	51	53	59	40	40	64	91	85	78	0	21	112	112	0,3
GEMITA	19	30	36	40	40	40	41	40	41	33	35	20	25	25	30	56	50	68	31	0	77	77	30
HINO	20	107	75	98	86	86	86	86	86	72	75	51	66	66	43	34	42	125	118	91	0	4	118
NISSAN	21	116	84	95	94	95	95	95	95	82	84	60	75	75	52	31	39	122	115	88	4	0	115
AWP2	22	6	47	51	51	51	53	52	53	51	53	63	45	44	72	95	88	79	0,3	24	118	115	0

Matriks Waktu Tempuh Kendaraan

Dari i Ke j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
AWP1	1	0	67	73	72	72	73	73	73	75	75	84	57	57	91	125	118	109	8	32	153	153	8
KWK	2	67	0	11	10	10	11	11	11	8	6	52	30	30	59	94	86	59	66	60	122	122	66
IAMI	3	71	9	0	4	4	1	1	2	13	10	57	34	34	63	98	90	56	71	65	127	127	70
AHM1	4	69	8	3	0	4	2	2	1	11	8	55	32	32	61	96	88	61	69	63	125	124	68
GS STR	5	69	8	3	4	0	2	3	1	11	9	55	32	32	61	96	88	61	69	63	125	125	69
GM	6	71	10	2	2	2	0	1	2	13	11	57	34	34	63	98	90	55	71	65	127	127	71
TAM PIO STR	7	71	10	1	2	1	1	0	1	13	10	57	34	34	63	98	90	55	70	64	126	126	70
DENSO STR	8	71	10	2	2	2	1	1	0	13	10	57	34	34	63	98	90	56	71	65	127	127	70
NUSMET	9	71	5	15	11	15	16	15	16	0	3	42	20	20	48	83	75	64	68	62	111	111	67
AHM2	10	71	7	11	11	11	12	11	8	3	0	45	23	23	51	86	78	67	71	65	114	114	70
AHM3	11	83	43	58	57	57	58	58	58	40	43	0	31	31	16	51	43	94	85	43	79	79	85
ADM SPD	12	96	54	35	34	34	35	35	35	19	22	32	0	0	39	73	65	71	60	54	102	102	59
TAM SPD	13	96	54	35	34	34	35	35	35	19	22	32	0	0	39	73	66	71	60	54	102	102	59
SANENG	14	97	55	69	68	68	69	69	70	51	54	22	42	42	0	46	38	105	96	56	74	74	96
GS KRW	15	128	86	100	99	99	100	99	100	82	85	53	73	73	42	0	25	136	127	90	42	42	127
TAM PIO KRW	16	119	77	91	90	90	91	90	91	73	76	43	64	64	33	25	0	127	118	81	54	54	118
SMU	17	113	76	52	52	52	53	52	53	75	60	92	69	69	98	133	125	0	106	100	161	161	105
TGSSI	18	8	62	68	67	67	68	67	68	68	70	79	53	53	86	121	113	104	0	27	149	149	0,4
GEMITA	19	40	48	54	53	53	54	53	54	44	47	27	34	34	40	75	67	90	41	0	103	103	40
HINO	20	143	101	131	114	114	115	114	115	96	100	68	88	88	57	45	56	167	157	121	0	5	157
NISSAN	21	155	112	127	126	127	127	126	127	109	112	79	100	100	69	41	52	163	153	117	5	0	153
AWP2	22	8	62	68	68	69	71	70	70	67	70	85	59	59	96	127	118	105	0,4	33	157	153	0

Lampiran 2 Script M-File Program

```
tic;

% Input Data
JumlahTitik = 21;
MatriksJarak = xlsread('Matriks Jarak.xlsx');
Waktu = xlsread('Matriks Waktu.xlsx');
Demand = xlsread('Permintaan.xlsx');
KapasitasKendaraan = 110;

% Parameter DE
JumlahPopulasi = 210; % NP = 10 x Jumlah Pelanggan
F = 0.6; % antara 0 - 2
Cr = 0.9; % antara 0 - 1

JumlahIterasiMaksimum = 1000;
NomorKendaraan = zeros(JumlahPopulasi,11);
WaktuTotal = [];

% Tahap Mutasi dan Rekombinasi
IndividuPopulasiAwal = rand(JumlahPopulasi, JumlahTitik);
IndividuPopulasiTSPAwal = ones(JumlahPopulasi, JumlahTitik + 2);
IndividuPopulasiVRPAwal = ones(JumlahPopulasi, JumlahTitik * 2 + 1);
JarakPopulasiAwal = zeros(1, JumlahPopulasi);

for i = 1 : JumlahPopulasi
    IndividuPopulasiTSPAwal(i, 2 : JumlahTitik + 1) =
PerformSPV(IndividuPopulasiAwal(i, :)) + 1;
    [IndividuPopulasiVRPAwal(i, :)] =
WaktuTotal(1,i),NomorKendaraan(i,:)] =
ConvertToVRPSolution2(IndividuPopulasiTSPAwal(i, :), Demand,
KapasitasKendaraan, Waktu);
    JarakPopulasiAwal(i) = CalculateTotalDistance(
IndividuPopulasiVRPAwal(i, :), MatriksJarak);
end

IndividuPopulasiIterasi = IndividuPopulasiAwal;
IndividuPopulasiTSPIterasi = IndividuPopulasiTSPAwal;
IndividuPopulasiVRPIterasi = IndividuPopulasiVRPAwal;
JarakPopulasiIterasi = JarakPopulasiAwal;

IndividuPopulasiIterasiMutan = IndividuPopulasiAwal;
IndividuPopulasiIterasiTrial = IndividuPopulasiAwal;
IndividuPopulasiTSPIterasiTrial = IndividuPopulasiTSPAwal;
IndividuPopulasiVRPIterasiTrial = IndividuPopulasiVRPAwal;
```

Lampiran 2 Script M-File Program (Lanjutan)

```
JarakPopulasiIterasiTrial = JarakPopulasiAwal;

for i = 1 : JumlahIterasiMaksimum
    % men-generate populasi mutan dan trial
    for j = 1 : JumlahPopulasi
        Index1 = randi(JumlahPopulasi);
        Index2 = Index1;
        while Index1 == Index2
            Index2 = randi(JumlahPopulasi);
        end
        Index3 = Index1;
        while Index3 == Index1 || Index3 == Index2
            Index3 = randi(JumlahPopulasi);
        end

        IndividuPopulasiIterasiMutan(j, :) =
PerformMutation(IndividuPopulasiIterasi(Index1, :),
IndividuPopulasiIterasi(Index2, :),
IndividuPopulasiIterasi(Index3, :), F);
        IndividuPopulasiIterasiTrial(j, :) =
PerformRecombination(IndividuPopulasiIterasi(j, :),
IndividuPopulasiIterasiMutan(j, :), Cr);
    end

    % Ubah Solusi Menjadi Solusi VRP dan Hitung Total Jarak Tempuh
    % Kendaraan
    for j = 1 : JumlahPopulasi
        IndividuPopulasiTSPIterasiTrial(j, 2 : JumlahTitik + 1) =
PerformSPV(IndividuPopulasiIterasiTrial(j, :)) + 1;
        [IndividuPopulasiVRPIterasiTrial(j, :)] =
WaktuTotal(i), NomorKendaraan(i, :)] =
ConvertToVRPSolution2(IndividuPopulasiTSPIterasiTrial(j, :),
Demand, KapasitasKendaraan, Waktu);
        JarakPopulasiIterasiTrial(j) =
CalculateTotalDistance(IndividuPopulasiVRPIterasiTrial(j, :),
MatriksJarak);
    end

    % Tahap Seleksi
    for j = 1 : JumlahPopulasi
        if JarakPopulasiIterasiTrial(j) < JarakPopulasiIterasi(j)
            IndividuPopulasiIterasi(j, :) =
IndividuPopulasiIterasiTrial(j, :);
            IndividuPopulasiTSPIterasi(j, :) =
IndividuPopulasiTSPIterasiTrial(j, :);
            IndividuPopulasiVRPIterasi(j, :) =
IndividuPopulasiVRPIterasiTrial(j, :);
        end
    end
end
```

Lampiran 2 Script M-File Program (Lanjutan)

```
JarakPopulasiIterasi(j) =
JarakPopulasiIterasiTrial(j);

    end
end

end

[jarakTerbaik, indexTerbaik] = min(JarakPopulasiIterasi);
index=(JumlahTitik * 2) + 1;
while (IndividuPopulasiVRPIterasi(indexTerbaik, index))==1
    index=index-1;
end
lastindex=index + 1;
optimal=[];
optimal=IndividuPopulasiVRPIterasi(indexTerbaik, 1:lastindex);

index=11;
while NomorKendaraan(indexTerbaik, index)==0
    index=index-1;
end
if WaktuTotal(indexTerbaik)>480
    WaktuTotal(indexTerbaik)=480;
end
Kendaraan=[];
Kendaraan = NomorKendaraan(indexTerbaik,1:index);
a = size(Kendaraan);

disp('Solusi Terbaik');
disp(optimal);
disp(' ');
disp(' ');
disp('=====');
disp('=====');
disp('=====');
disp('=====');
disp('Jarak Terbaik');
disp(jarakTerbaik);

% OUTPUT Per Hari
RuteKendaraan=[];
perhari=[];
[r c]=find(optimal==24);
c(2:length(c)+1)=c(1:length(c));
c(length(c)+1)=length(optimal(1,:));
c(1)=0;
JarakTempuh=0;
```

Lampiran 2 Script M-File Program (Lanjutan)

```
for ii=2:length(c(1,:))
    RuteKendaraan(1,1:c(ii)-c(ii-1))=optimal(1,c(ii-1)+1:c(ii));
    [rhari chari]=find(RuteKendaraan==23);
    chari(2:length(chari)+1)=chari(1:length(chari));

    if ii==length(c(1,:))
        chari(length(chari)+1)=length(RuteKendaraan(1,:));
    end
    chari(1)=0;

    % Menghitung jarak tempuh total tiap kendaraan

    JarakTempuh=CalculateTotalDistance(RuteKendaraan,MatriksJarak);

    disp('=====');
    disp('=====');

    disp('=====');
    disp('=====');

    disp(['Kendaraan ke ', num2str(ii-1)]);
    disp(['Jarak Tempuh Total : ', num2str(JarakTempuh)]);
    disp(' ');

    for ihari=2:length(chari(1,:))
        perhari(1,1:chari(ihari)-chari(ihari-
1))=RuteKendaraan(1,chari(ihari-1)+1:chari(ihari));
    % perhari
        disp(['Hari ke ', num2str(ihari-1)]);
        if perhari(1,chari(ihari)-chari(ihari-1))==23
            disp(['Rute ', num2str(perhari(1,1:chari(ihari)-
chari(ihari-1)-1))] );
            disp(' ');
        else
            disp(['Rute ', num2str(perhari)]);
            disp(' ');
        end
        perhari=[];
    end
    chari=[];
    RuteKendaraan=[];
end

toc;

% Untuk Melakukan Permutasi atau pengurutan nilai bilangan acak
pada setiap induvidu populasi awal
```


Lampiran 2 Script M-File Program (Lanjutan)

```
function UrutanSPV = PerformSPV(UrutanAwal)
UrutanSPV = zeros(size(UrutanAwal));
[B, index] = sort(UrutanAwal);
for i = 1 : numel(UrutanAwal)
    UrutanSPV(index(i)) = i;
end

% Untuk mengubah solusi TSP menjadi Solusi VRP, digunakan fungsi
berikut:
function [SolusiVRP WaktuTotal NoKendaraan] =
ConvertToVRPSolution2(SolusiTSP, DemandTitik, KapasitasKendaraan,
Waktu)
JumlahTitik = numel(SolusiTSP) - 2;
SolusiVRP = ones(1, JumlahTitik * 2 + 1);
DemandSekarang = 0;
IndeksVRPSekarang = 1;
TotalWaktu = 0;
JamKerja = 480;
HariPengiriman = 1;
NoKendaraan = zeros(1,11);
Kapasitas = 110;
NoKendaraan(1,1)=Kapasitas;
idxkendaraan = 2;
hari = 1;

for i = 2 : JumlahTitik + 1
    if hari <= HariPengiriman
        if DemandSekarang + DemandTitik(SolusiTSP(i)) <=
KapasitasKendaraan % tidak perlu buat rute baru
            DemandSekarang = DemandSekarang +
DemandTitik(SolusiTSP(i));
            IndeksVRPSekarang = IndeksVRPSekarang + 1;
            SolusiVRP(IndeksVRPSekarang) = SolusiTSP(i);
            ServiceTime=(0.4*DemandSekarang)+ 30;
            TotalWaktu = TotalWaktu +
Waktu(SolusiVRP(IndeksVRPSekarang-
1),SolusiTSP(i))+Waktu(SolusiVRP(IndeksVRPSekarang-1),1);
        else % perlu buat rute baru
            % sisipkan depot (kota 1)
            %DemandSekarang = 0;
            IndeksVRPSekarang = IndeksVRPSekarang + 1;
            SolusiVRP(IndeksVRPSekarang) = 1;
            % sisipkan kota berikutnya di rute baru
            DemandSekarang = DemandTitik(SolusiTSP(i));
            IndeksVRPSekarang = IndeksVRPSekarang + 1;
            SolusiVRP(IndeksVRPSekarang) = SolusiTSP(i);
            if TotalWaktu + ServiceTime <= JamKerja
```

Lampiran 2 Script M-File Program (Lanjutan)

```
SolusiVRP(IndeksVRPSekarang) = SolusiTSP(i);
else
    %SolusiVRP(IndeksVRPSekarang) = 1;
    IndeksVRPSekarang = IndeksVRPSekarang + 1;
    SolusiVRP(IndeksVRPSekarang) = 23;
    hari = hari+1;
    if hari <= HariPengiriman
        IndeksVRPSekarang = IndeksVRPSekarang + 1;
        SolusiVRP(IndeksVRPSekarang) = 1;
        IndeksVRPSekarang = IndeksVRPSekarang + 1;
        TotalWaktu =
Waktu(SolusiVRP(IndeksVRPSekarang-1), SolusiTSP(i));
        SolusiVRP(IndeksVRPSekarang) = SolusiTSP(i);
    else
        hari = 1;
        IndeksVRPSekarang = IndeksVRPSekarang + 1;
        SolusiVRP(IndeksVRPSekarang) = 24;
        NoKendaraan(1, idxkendaraan)=Kapasitas;
        idxkendaraan = idxkendaraan + 1;
        IndeksVRPSekarang = IndeksVRPSekarang + 1;
        SolusiVRP(IndeksVRPSekarang) = 1;
        DemandSekarang=0;
        TotalWaktu=0;
    end
end
end
else
    hari = 1;
    IndeksVRPSekarang = IndeksVRPSekarang + 1;
    SolusiVRP(IndeksVRPSekarang) = 24;
    NoKendaraan(1, idxkendaraan) = Kapasitas;
    idxkendaraan = idxkendaraan + 1;
    IndeksVRPSekarang = IndeksVRPSekarang + 1;
    SolusiVRP(IndeksVRPSekarang) = 1;
    DemandSekarang = 0;
end
    if SolusiVRP(IndeksVRPSekarang)==1 &&
DemandTitik(SolusiTSP(i))== 0
        SolusiVRP(IndeksVRPSekarang + 1) = 23;
    end
end

WaktuTotal = TotalWaktu;

% Untuk menghitung total jarak tempuh kendaraan tiap solusi vrp
generasi terakhir digunakan fungsi berikut:

function TotalDistance = CalculateTotalDistance(Solusi,
DistanceMatrix)
```

Lampiran 2 Script M-File Program (Lanjutan)

```
NumberOfJourneys = numel(Solusi) - 1;
TotalDistance = 0;
for i = 1 : NumberOfJourneys
    TotalDistance = TotalDistance + DistanceMatrix(Solusi(i),
Solusi(i + 1));
end

% Fungsi yang digunakan untuk melakukan tahapan mutasi

function SolusiMutasi = PerformMutation(Solusi1, Solusi2, Solusi3,
F)
SolusiMutasi = Solusi1 + F * (Solusi2 - Solusi3);

% Fungsi yang digunakan untuk melakukan tahapan rekombinasi atau
pindah silang:

function SolusiRekombinasi = PerformRecombination(SolusiAwal,
SolusiTrial, Cr)
SolusiRekombinasi = zeros(size(SolusiAwal));
for i = 1 : numel(SolusiAwal)
    if rand < Cr
        SolusiRekombinasi(i) = SolusiTrial(i);
    else
        SolusiRekombinasi(i) = SolusiAwal(i);
    end
end
end
```