



**PEMBENTUKAN RUTE DISTRIBUSI UNTUK
MEMINIMALKAN BIAYA DISTRIBUSI DENGAN
ALGORITMA *NEAREST NEIGHBOUR* DALAM
PENDISTRIBUSIAN TABUNG GAS LPG 3-KG DI PT.
GARUDA MANDIRI**

SKRIPSI

oleh:

**ZIQRA DIKA PUTRI
17509050111027**



**PROGRAM STUDI SARJANA STATISTIKA
JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2021**



**PEMBENTUKAN RUTE DISTRIBUSI UNTUK
MEMINIMALKAN BIAYA DISTRIBUSI DENGAN
ALGORITMA *NEAREST NEIGHBOUR* DALAM
PENDISTRIBUSIAN TABUNG GAS LPG 3-KG DI PT.
GARUDA MANDIRI**

SKRIPSI

Sebagai salah satu syarat memperoleh gelar
Sarjana Statistika

oleh:

ZIQRA DIKA PUTRI

175090501111027



**PROGRAM STUDI SARJANA STATISTIKA
JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2021**

**LEMBAR PENGESAHAN SKRIPSI****PEMBENTUKAN RUTE DISTRIBUSI UNTUK
MEMINIMALKAN BIAYA DISTRIBUSI DENGAN
ALGORITMA NEAREST NEIGHBOUR DALAM
PENDISTRIBUSIAN TABUNG GAS LPG 3 KG DI
PT. GARUDA MANDIRI**

oleh:
ZIQRA DIKA PUTRI

175090501111027

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 30 November 2021

dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Statistika

Dosen Pembimbing

Rahma Fitriani, S.Si., M.Sc., Ph.D

NIP. 197603281999032001

Mengetahui,
Ketua Jurusan Statistika
Fakultas MIPA
Universitas Brawijaya



Rahma Fitriani, S.Si., M.Sc., Ph.D

NIP. 197603281999032001

**LEMBAR PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Ziqra Dika Putri

NIM : 175090501111027

Jurusan : Statistika

Judul Skripsi :

**PEMBENTUKAN RUTE DISTRIBUSI UNTUK
MEMINIMALKAN BIAYA DISTRIBUSI DENGAN
ALGORITMA NEAREST NEIGHBOUR DALAM
PENDISTRIBUSIAN TABUNG GAS LPG 3 KG DI
PT. GARUDA MANDIRI**

Dengan ini menyatakan bahwa:

1. Isi dari Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Skripsi ini.
2. Apabila di kemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung risiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran:

Pekanbaru, 25 Oktober 2021

Yang menyatakan,

Ziqra Dika Putri

NIM: 175090501111027



PEMBENTUKAN RUTE DISTRIBUSI UNTUK MEMINIMALKAN BIAYA DISTRIBUSI DENGAN ALGORITMA *NEAREST NEIGHBOUR* DALAM PENDISTRIBUSIAN TABUNG GAS LPG 3 KG DI PT. GARUDA MANDIRI

ABSTRAK

Dalam pelaksanaan suatu sistem distribusi, sering timbul permasalahan mengenai pemilihan rute distribusi yang tidak optimal. Permasalahan ini dapat menyebabkan biaya distribusi yang harus dikeluarkan perusahaan menjadi meningkat dari yang seharusnya. Tujuan yang ingin dicapai dalam penelitian ini adalah menciptakan rute rekomendasi bagi PT. Garuda Mandiri dalam proses distribusi Gas LPG 3 Kg agar biaya distribusi yang perlu dikeluarkan dapat diminimalkan dengan melakukan penerapan algoritma *nearest neighbour*. Penerapan algoritma *nearest neighbour* dilakukan berdasarkan data jarak yang diperoleh dari lokasi masing-masing pangkalan yang harus dilayani oleh perusahaan. Algoritma *nearest neighbour* berdasarkan data jarak menghasilkan 127 rute dengan total jarak tempuh sebesar 7599,32 Km. Rute yang dihasilkan menggunakan algoritma *nearest neighbour* menghasilkan penghematan jarak tempuh sebesar 1000,68 Km dan biaya bahan bakar sebesar Rp 1.000.680,- untuk bulan Februari 2021.

Kata Kunci: Algoritma *Nearest Neighbor*, Distribusi Gas LPG 3 Kg, *Vehicle Routing Problem*



ESTABLISHMENT OF DISTRIBUTION ROUTES TO MINIMIZE DISTRIBUTION COSTS WITH NEAREST NEIGHBOR ALGORITHM IN DISTRIBUTION OF 3 KG LPG GAS IN PT. GARUDA MANDIRI

ABSTRACT

In the implementation of a distribution system, problems often arise regarding the selection of distribution routes that are not optimal. This problem can cause distribution costs to be incurred by the company to increase than they should. The purpose of this research is to create a recommendation route for PT. Garuda Mandiri is in the process of distributing 3 Kg LPG Gas so that the distribution costs that need to be incurred can be minimized by implementing the nearest neighbour algorithm. The application of the nearest neighbour algorithm is carried out based on distance data obtained from the location of each base that must be served by the company. The nearest neighbour algorithm based on distance data produces 127 routes with a total distance of 7599.32 Km. The route generated using the nearest neighbour algorithm results in a distance savings of 1000.68 Km and fuel costs of Rp 1,000,680,- for February 2021.

Keyword: Nearest Neighbor Algorithm, 3 Kg LPG Gas Distribution, Vehicle Routing Problem



KATA PENGANTAR

Puji syukur sedalam-dalamnya kepada Allah SWT atas rahmat dan hidayah-Nya sehingga skripsi yang berjudul “Pembentukan Rute Distribusi Untuk Meminimalkan Biaya Distribusi dengan Algoritma *Nearest Neighbour* Dalam Pendistribusian Tabung Gas LPG 3 Kg di PT. Garuda Mandiri” dapat terselesaikan. Proses penyusunan skripsi ini tidak lepas dari berbagai bantuan, dukungan, dan doa berbagai pihak. Oleh karena itu, dalam kesempatan ini disampaikan terima kasih kepada:

1. PT. Garuda Mandiri selaku perusahaan yang memberikan izin sehingga penelitian ini dapat berjalan.
2. Rahma Fitriani, S.Si., M.Si., Ph.D. selaku dosen pembimbing skripsi dan ketua Jurusan Statistika FMIPA Universitas Brawijaya atas waktu, bimbingan, dan motivasi yang telah diberikan dalam penyusunan skripsi.
3. Dr. Eni Sumarminingsih, S.Si., MM. selaku dosen penguji I atas waktu, saran, dan bimbingan dalam penyusunan skripsi.
4. Dr. Adji Achmad Rinaldo F., S.Si., M.Sc. selaku dosen penguji II atas waktu, saran, dan bimbingan dalam penyusunan skripsi.
5. Dr. Ir. Solimun, MS. selaku dosen pembimbing akademik atas bimbingan dan motivasi sepanjang perkuliahan.
6. Achmad Efendi, S.Si., M.Sc., Ph.D. selaku ketua program studi Sarjana Statistika FMIPA Universitas Brawijaya.
7. Seluruh dosen, staf pengajar dan karyawan Jurusan Statistika FMIPA Universitas Brawijaya.
8. Mama, Papa, Aqsha, dan seluruh keluarga besar atas doa, dukungan, kasih sayang, dan semangat yang tidak pernah putus.

9. Dinda, Sekar, Adel, Femy, keluarga besar RBC, teman-teman Statistika 2017 dan seluruh sahabat serta teman-teman atas dukungan, semangat, dan bantuan.
10. Semua pihak yang telah membantu penyelesaian penelitian ini.

Penulis menyadari bahwa skripsi ini masih belum sempurna, oleh karena itu diharapkan kritik dan saran yang membangun dari pembaca. Semoga skripsi ini dapat memberikan manfaat dan menambah pengetahuan bagi semua pihak.

Pekanbaru, September 2021

Penulis





DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN SKRIPSI	iii
LEMBAR PERNYATAAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
DAFTAR LAMPIRAN	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	5
1.3 Tujuan Penelitian.....	5
1.4 Manfaat Penelitian.....	6
1.5 Batasan Masalah.....	6
BAB II TINJAUAN PUSTAKA	7
2.1 Distribusi.....	7
2.2 Efektivitas.....	8
2.3 Graf.....	8
2.3.1 Jenis-Jenis Graf.....	9
2.3.2 Representasi Graf.....	12
2.4 <i>Vehicle Routing Problem</i> (VRP).....	14
2.5 <i>Algoritma Nearest Neighbour</i>	22



BAB III METODE PENELITIAN	23
3.1 Sumber Data	23
3.2 Penyelesaian Algoritma <i>Nearest Neighbour</i>	24
3.3 Metode Analisis Data	25
3.4 Diagram Alir Penelitian	27
BAB IV HASIL DAN PEMBAHASAN	29
4.1 Gambaran Umum Perusahaan	29
4.1.1 Profil Perusahaan	29
4.1.2 Rute Distribusi Perusahaan	29
4.1.3 Perhitungan Biaya Distribusi Perusahaan	31
4.2 Pengolahan Data	32
4.2.1 Algoritma <i>Nearest Neighbour</i>	33
4.2.2 Alokasi Pangkalan ke dalam Rute	35
4.2.3 Biaya Distribusi Rute <i>Nearest Neighbour</i>	38
4.3 Perbandingan Biaya Distribusi Rute Perusahaan dan Rute Rekomendasi	39
BAB V PENUTUP	41
5.1 Kesimpulan	41
5.2 Saran	41
DAFTAR PUSTAKA	43
LAMPIRAN	45



DAFTAR GAMBAR

Gambar 2.1	Graf Sederhana	9
Gambar 2.2	Graf Tak-Sederhana	10
Gambar 2.3	Graf Tak-Berarah	10
Gambar 2.4	Graf Berarah	11
Gambar 2.5	Graf dengan matriks ketetangaan masing-masing	13
Gambar 2.6	Graf berbobot dan matriks ketetangannya	13
Gambar 2.7	Variasi Permasalahan VRP dan Hubungannya	15
Gambar 2.8	Contoh solusi CVRP yang terdiri dari 2 rute	18
Gambar 3.1	Diagram Alir Penelitian	27
Gambar 4.1	Peta lokasi depot, SPPBE dan pangkalan	33



DAFTAR TABEL

Tabel 2.1	Definisi Graf.....	11
Tabel 4.1	Rute Distribusi Perusahaan.....	30
Tabel 4.2	Biaya Rute Distribusi Perusahaan.....	32
Tabel 4.3	Rute Distribusi <i>Nearest Neighbour</i>	35
Tabel 4.4	Visualisasi Rute <i>Nearest Neighbour</i>	37
Tabel 4.4	Biaya Distribusi Rute Rekomendasi.....	39



DAFTAR LAMPIRAN

Lampiran 1	Tabel Jarak.....	45
Lampiran 2	Data Permintaan	53
Lampiran 3	Data Biaya Perusahaan 2021	56
Lampiran 4	Rute Distribusi Algoritma <i>Nearest Neighbour</i> ..	57
Lampiran 5	Visualisasi Rute <i>Nearest Neighbour</i>	63
Lampiran 6	<i>Source Code</i> Algoritma <i>Nearest Neighbour</i> menggunakan <i>software Python</i>	71



BAB I PENDAHULUAN

1.1 Latar Belakang

Statistika menurut Yitnosumarto (1990) merupakan metode, ilmu dan seni yang dipergunakan untuk pengumpulan data, analisis data, dan interpretasi hasil analisis serta mempergunakannya untuk maksud-maksud peramalan. Montgomery (1985) menjelaskan bahwa statistika merupakan seni pengambilan keputusan tentang suatu proses atau populasi berdasarkan pada suatu analisis informasi yang terkandung dalam suatu sampel dari populasi. Dengan demikian, dapat dikatakan bahwa statistika merupakan suatu alat yang dapat digunakan untuk memperoleh suatu keputusan.

Riset operasi merujuk pada sebuah riset atau penelitian pada operasi. Operasi yang dimaksud dapat berupa operasi dalam bidang bisnis, industri, jasa, pemerintahan, militer, dsb. Menurut Taha (1993), riset operasi merupakan pendekatan dalam pengambilan keputusan yang bertujuan untuk menentukan arah tindakan terbaik (optimal) dengan sumber daya yang terbatas. Lebih lanjut, Siang (2011) menjelaskan bahwa riset operasi berhubungan dengan prinsip optimisasi yakni tentang bagaimana cara menggunakan sumber daya (biaya, waktu, tenaga, dll.) untuk mengoptimalkan hasil. Mengoptimalkan hasil dapat berupa memaksimalkan sesuatu yang menguntungkan/didapatkan atau meminimalkan sesuatu yang merugikan/dikeluarkan. Pendekatan dalam riset operasi ini bertujuan untuk membantu proses pengambilan keputusan untuk menentukan kebijakan dan tindakan secara ilmiah.

Melalui penjelasan tersebut dapat dilihat bahwa statistika dan riset operasi memiliki tujuan yang sejalan. Terdapat peran statistika dalam penerapan riset operasi. Kedua bidang ini berkaitan antara satu sama lain dan memiliki tujuan yang sama yaitu memberikan informasi untuk mencapai suatu keputusan.



Dalam menjalankan operasional bisnis, sebuah perusahaan tentu mengeluarkan bermacam jenis biaya. Bagi sebuah perusahaan, jumlah pengeluaran tentu akan berpengaruh pada keuntungan yang diperoleh. Semakin besar biaya bisnis yang dikeluarkan perusahaan, maka akan semakin kecil keuntungan yang didapat. Salah satu proses yang dapat di optimalisasi yaitu proses distribusi. Dalam suatu sistem distribusi, rute yang dipilih merupakan elemen penting dalam menentukan jarak yang harus ditempuh dan biaya yang harus dikeluarkan. Jika rute yang dipilih optimal, maka sistem distribusi yang dipilih lebih efektif dan efisien. Semakin optimal jarak yang ditempuh, akan semakin rendah biaya yang harus dikeluarkan perusahaan. Pemilihan strategi distribusi yang tepat dapat memberikan keuntungan bagi semua pihak.

Salah satu contoh permasalahan distribusi yang sangat umum terjadi dalam masyarakat adalah pendistribusian tabung gas LPG (*Liquefied Petroleum Gas*). Sampai saat ini, terhitung sudah lebih dari 10 tahun sejak program konversi minyak tanah ke LPG digencarkan oleh Kementerian Energi dan Sumber Daya Mineral. Hampir seluruh rumah tangga dan industri kecil di Indonesia telah menggunakan tabung gas LPG dan hal ini menunjukkan bahwa kebutuhan gas LPG terus mengalami peningkatan, terutama tabung gas LPG 3 Kg. Hal ini dikarenakan Gas LPG 3 Kg merupakan produk yang praktis dan paling murah dari PT. Pertamina. Dalam prosesnya, distribusi tabung gas LPG sampai ke tangan konsumen melalui serangkaian proses yang kompleks. Proses distribusi berawal dari pengadaan LPG yang diproduksi dari kilang di dalam negeri maupun hasil impor. LPG ini selanjutnya didistribusikan ke depot-depot LPG yang selanjutnya akan didistribusikan ke Stasiun Pengisian dan Pengangkutan Bulk Elpiji (SPPBE). Di SPPBE ini dilakukan pengisian LPG ke tabung gas LPG 3 Kg, 12 Kg, dan 50 Kg yang selanjutnya akan disalurkan ke agen-agen LPG. Agen-agen LPG kemudian akan menyalurkan tabung gas LPG ke subagen/pangkalan LPG yang telah terdaftar.

PT. Garuda Mandiri merupakan salah satu agen tabung gas LPG 3 Kg bersubsidi untuk wilayah penyaluran Kota Pekanbaru. PT. Garuda Mandiri berlokasi di Jl. Satria No. 09, Kelurahan Rejosari, Kecamatan Tenayan Raya, Kota Pekanbaru. Perusahaan ini



menyalurkan tabung gas LPG 3 Kg dari SPPBE kepada subagen/pangkalan yang telah terdaftar di PT. Pertamina. Proses penyaluran ini tentunya membutuhkan biaya distribusi yang harus dikeluarkan oleh perusahaan.

Dalam proses distribusi yang digunakan perusahaan, metode yang digunakan untuk penentuan rute distribusi diserahkan sepenuhnya kepada supir. Metode ini selanjutnya akan disebut sebagai metode klasik. Perusahaan akan memberikan daftar pangkalan yang harus dikunjungi/hari/minggu kepada masing-masing supir. Selanjutnya, pemilihan rute, lokasi pangkalan yang harus dikunjungi terlebih dahulu, jalan mana yang harus ditempuh, dll. dilakukan berdasar penilaian supir sendiri. Metode seperti ini berisiko menyebabkan pemilihan rute yang tidak optimal. Dengan demikian, hambatan pada sistem distribusi sangat mungkin terjadi, sehingga perlu dilakukan optimalisasi agar biaya dapat dikeluarkan secara efisien. Salah satu cara untuk mengoptimalkan biaya distribusi adalah dengan mengoptimalkan jarak yang harus ditempuh dalam proses pendistribusian barang. Oleh karena itu, pemilihan rute distribusi yang tepat dalam hal ini dapat menjadi solusi untuk mengoptimalkan biaya distribusi yang perlu dikeluarkan perusahaan.

Permasalahan distribusi, pembentukan rute, dan optimalisasi merupakan salah satu bagian dari riset operasi. Dengan data yang baik dan metode yang sesuai, perusahaan dapat mencapai keputusan terbaik dari informasi yang diperoleh dengan analisis yang tepat. Dalam hal ini, diperlukan peran besar ilmu statistika dan riset operasi.

Dalam kasus serupa dengan penelitian ini, permasalahan yang sering muncul adalah ketika simpul atau lokasi yang harus dikunjungi dalam suatu sistem distribusi itu banyak dan harus dilakukan tepat satu kali tanpa pengulangan, kemudian akan berakhir kembali pada titik semula. Dengan begitu, rute yang harus ditempuh akan memiliki banyak kemungkinan pemilihan. Permasalahan seperti ini umumnya dikenal sebagai model *vehicle routing problem* (VRP). VRP adalah masalah penentuan rute dan penjadwalan dimana terdapat beberapa kendala dengan tujuan tertentu. Menurut Toth & Vigo (2002), VRP merupakan masalah penentuan rute kendaraan dalam



mendistribusikan barang dari suatu depot ke konsumen dengan tujuan meminimumkan total jarak tempuh kendaraan. Dalam hal ini, depot dapat digambarkan sebagai gudang atau tempat keluar dan kembalinya kendaraan yang digunakan untuk mendistribusikan barang/produk kepada konsumen.

Dalam kasus ini, model yang lebih tepat digunakan adalah menggunakan model *capacitated vehicle routing problem* (CVRP). CVRP adalah bentuk paling dasar dari VRP. CVRP merupakan masalah optimasi untuk menemukan rute dengan biaya minimal (*minimum cost*) untuk sejumlah kendaraan (*vehicles*) dengan kapasitas tertentu yang homogen (*homogeneous fleet*), yang melayani permintaan pelanggan yang kuantitas permintaannya telah diketahui sebelum proses pengiriman berlangsung (Rupiah dkk., 2017).

Permasalahan VRP dapat diselesaikan dengan menggunakan algoritma *Clarke-Wright savings*, algoritma genetika, algoritma *nearest neighbour*, algoritma *ant colony optimization* (ACO), algoritma *sweep*, dan algoritma *sequential insertion*. Algoritma *nearest neighbour* adalah metode yang paling sederhana dibandingkan metode yang lain. Algoritma *nearest neighbour* pada prinsipnya selalu menambahkan tujuan yang jaraknya paling dekat dengan tujuan terakhir yang dilalui kendaraan. Diharapkan dari pencarian jarak terdekat dapat meminimalisasi biaya distribusi yang perlu dikeluarkan perusahaan.

Beberapa penelitian tentang algoritma *nearest neighbour* antara lain, Kurniawati dkk. (2017) melakukan penelitian di PT. Graha Gas Niaga menggunakan algoritma *nearest neighbour* mengenai pendistribusian gas LPG dan memperoleh hasil bahwa penggunaan algoritma *nearest neighbour* mampu memberikan hasil dari rute, jarak, dan waktu tempuh yang minimal dibandingkan rute distribusi yang digunakan perusahaan. Yulianingrum (2019) melakukan penelitian di PT. Harum Ossamac Grobogan mengenai distribusi gas LPG, yang memperoleh hasil bahwa algoritma *nearest neighbour* memberikan solusi yang lebih efektif dibandingkan algoritma *Clarke-Wright savings*.



Berdasarkan hasil dari penelitian terdahulu bahwa algoritma *nearest neighbour* mampu memberikan solusi yang lebih efektif dalam penghematan biaya distribusi dibandingkan algoritma *Clarke-Wright savings*, peneliti tertarik untuk membahas penyelesaian masalah rute pendistribusian tabung gas LPG dengan menggunakan algoritma *nearest neighbour*. Algoritma *nearest neighbour* diterapkan pada model *Capacitated Vehicle Routing Problem* (CVRP). Selanjutnya, akan dibandingkan hasil yang diberikan algoritma *nearest neighbour* dan metode klasik untuk melihat seberapa efektif penggunaan algoritma *nearest neighbour* dapat memberikan solusi penghematan jarak dan biaya dalam proses distribusi untuk diterapkan dalam kasus ini.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah pada penelitian ini adalah sebagai berikut.

1. Bagaimana penyelesaian masalah rute pendistribusian tabung gas LPG 3 Kg dari depot ke subagen/pangkalan LPG dan kembali ke depot serta jarak dan biaya distribusi dengan algoritma *nearest neighbour*?
2. Bagaimana efektivitas penghematan jarak dan biaya distribusi tabung gas LPG 3 Kg dari depot ke subagen/pangkalan LPG dan kembali ke depot menggunakan algoritma *nearest neighbour*?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan penelitian ini adalah sebagai berikut.

1. Menentukan solusi masalah rute pendistribusian tabung gas LPG 3 Kg dari depot ke subagen/pangkalan LPG dan kembali ke depot serta jarak dan biaya distribusi dengan algoritma *nearest neighbour*.
2. Menentukan seberapa efektif algoritma yang digunakan dalam memberikan penghematan jarak dan biaya distribusi tabung gas



LPG 3 Kg dari depot ke subagen/pangkalan LPG dan kembali ke depot menggunakan algoritma *nearest neighbour*.

1.4

Manfaat Penelitian

Manfaat yang diharapkan dari penelitian adalah sebagai berikut.

Memberi wawasan ilmu tentang penerapan algoritma *nearest neighbour* untuk menyelesaikan permasalahan rute terpendek.

Mengevaluasi proses distribusi untuk meningkatkan dan memperbaiki agar dapat meminimalkan biaya distribusi yang perlu dikeluarkan.

Memberikan alternatif solusi yang efektif dan efisien dalam permasalahan pemilihan rute kepada perusahaan.

1.5

Batasan Masalah

Batasan masalah yang digunakan pada penelitian ini adalah sebagai berikut.

1.

Tabung gas LPG 3 Kg yang digunakan layak pakai.

2.

Jumlah kebutuhan/permintaan dari setiap subagen/pangkalan LPG sudah ditentukan.

3.

Pengiriman tabung gas LPG 3 Kg menggunakan armada yang sudah disediakan oleh perusahaan dan dalam kondisi yang baik.

4.

Tidak mempertimbangkan adanya faktor acak, seperti bencana alam, perang, dan sebagainya.

6



BAB II TINJAUAN PUSTAKA

2.1 Distribusi

Proses distribusi merupakan salah satu aspek penting dari kegiatan pemasaran. Menurut Budiarto (1993) distribusi dapat diartikan sebagai kegiatan pemasaran yang berusaha memperlancar serta mempermudah penyampaian produk dan jasa dari produsen kepada konsumen sehingga penggunaannya sesuai (jenis, jumlah, harga, tempat, dan saat) yang dibutuhkan. Tandjung (2004) berpendapat, distribusi merupakan jalan atau rute yang dilalui produk mulai dari produsen sampai ke tangan konsumen. Distribusi yang efektif akan memperlancar arus atau akses barang oleh konsumen.

Sebuah perusahaan distributor adalah perantara yang menghubungkan produsen dan konsumen. Dalam hal ini produsen yang dimaksud adalah PT. Pertamina, dan konsumen mengacu pada subagen/pangkalan yang telah terdaftar. Setelah suatu produk dihasilkan oleh pabrik, selanjutnya dikirimkan/dijual kepada distributor (agen) dan selanjutnya didistribusikan ke *retailer* (subagen).

Proses distribusi dipengaruhi faktor-faktor tertentu agar dapat berjalan lancar, seperti sistem distribusi, rute distribusi, dan transportasi distribusi. Sistem distribusi yang sesuai dapat digunakan untuk mencapai berbagai tujuan rantai pasokan mulai dari biaya rendah hingga daya tanggap tinggi. Sistem yang tidak tepat dapat memiliki efek negatif yang signifikan terhadap profitabilitas perusahaan. Hambatan yang paling sering menyebabkan suatu sistem distribusi menjadi tidak efektif adalah pemilihan rute distribusi yang dilalui. Akibat dari pemilihan rute yang tidak optimal, jarak dan waktu yang perlu di tempuh untuk mendistribusikan barang menjadi lebih besar dan menyebabkan biaya yang perlu dikeluarkan menjadi lebih besar pula.

Menurut KBBI, rute dapat diartikan sebagai jarak (jalan, arah, jalur) yang harus diturut (ditempuh, dilalui) yang menghubungkan dua



tempat berbeda. Pada kasus ini, rute dapat didefinisikan sebagai siklus sederhana yang perlu ditempuh dari titik sumber yang melewati depot awal ke beberapa tujuan hingga total permintaan terpenuhi dan tidak lebih dari kapasitas yang telah ditetapkan. Maka, dengan pemilihan rute yang tepat dapat menghemat biaya transportasi yang perlu dikeluarkan perusahaan.

2.2 Efektivitas

Menurut Siagian (2008) Efektivitas adalah pemanfaatan sumber daya, sarana, dan prasarana dalam jumlah tertentu yang secara sadar ditetapkan sebelumnya untuk menghasilkan sejumlah barang atas jasa kegiatan yang dijalankannya. Efektivitas mengacu pada ketepatangunaan suatu program/metode untuk mencapai tujuan yang diinginkan. Dengan demikian, efektivitas juga dapat diartikan bahwa sasaran yang direncanakan telah tercapai karena adanya suatu proses kegiatan tertentu.

Pada penelitian ini, dilakukan perbandingan efektivitas hasil menggunakan algoritma *nearest neighbour* dan metode klasik oleh perusahaan. Algoritma dikatakan efektif apabila algoritma yang dibandingkan menghasilkan rute yang efektif. Rute yang efektif dapat didefinisikan sebagai rute yang menghasilkan jarak dan biaya distribusi paling minimal dalam menyelesaikan proses distribusi.

2.3 Graf

Graf umumnya digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Menurut Munir (2010) Graf G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$, yang dalam hal ini V adalah himpunan tidak kosong dari simpul dan E adalah himpunan sisi yang menghubungkan sepasang simpul. Dari pengertian diatas jelas bahwa himpunan V tidak boleh kosong, sementara himpunan E boleh kosong. Sebuah graf dimungkinkan tidak memiliki satu pun sisi, akan tetapi minimal harus memiliki satu simpul. Graf jenis ini disebut graf trivial.

Simpul pada graf dapat dinomori dengan huruf (a, b, c, \dots) dengan bilangan asli $(1, 2, 3, \dots)$, atau gabungan keduanya. Sedangkan sisi yang menghubungkan simpul u dengan simpul



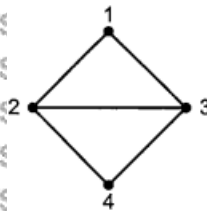
v dinyatakan dengan pasangan (u, v) atau dinyatakan dengan lambang e_1, e_2, \dots . Dengan kata lain, jika e adalah sisi yang menghubungkan simpul u dan v , maka e dapat ditulis $e = (u, v)$.

2.3.1 Jenis-Jenis Graf

Berdasarkan ada tidaknya gelang (*loop*) maupun sisi-ganda (*multiple edges* atau *parallel edges*), maka secara umum graf dapat digolongkan menjadi dua jenis.

a. Graf sederhana (*simple graph*)

Graf sederhana adalah graf yang tidak mengandung gelang maupun sisi-ganda. Pada graf jenis ini, sisi adalah pasangan tak-terurut (*unordered pairs*). Jadi, ketika menulis sisi (u, v) sama saja dengan menulis (v, u) . Dalam definisi umum graf $G = (V, E)$, E adalah himpunan pasangan tak-terurut yang berbeda yang disebut sisi. Contoh graf sederhana dapat dilihat seperti pada gambar 2.1.



Gambar 2.1 Graf Sederhana

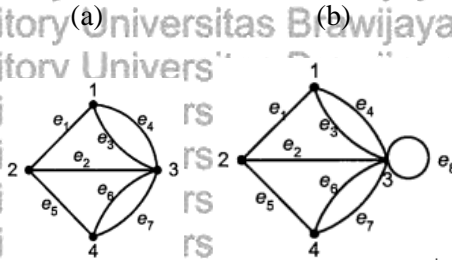
Sumber: Munir, 2010

b. Graf tak-sederhana (*unsimple graph*)

Kebalikan dari graf sederhana, graf yang terdapat sisi-ganda atau gelang dinamakan graf tak-sederhana. Graf yang mengandung sisi ganda disebut graf ganda (*multigraph*). Sisi-ganda dapat diasosiasikan sebagai pasangan tak-terurut yang sama. Dalam definisi umum graf $G = (V, E)$, E adalah himpunan ganda (*multiset*) yang mengandung sisi-ganda. Sedangkan graf yang mengandung gelang disebut graf semu. Graf semu lebih umum daripada graf ganda, karena



sisi pada graf semu dapat terhubung ke dirinya sendiri. Contoh graf tak-sederhana dapat dilihat seperti pada gambar 2.2.



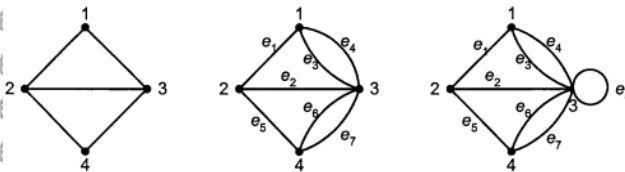
Gambar 2.2 Graf Tak-Sederhana; (a) Graf Ganda, (b) Graf Semu

Sumber: Munir, 2010

Sisi yang terdapat pada graf dapat mempunyai orientasi arah. Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas 2 jenis.

a. Graf tak-berarah (*undirected graph*)

Graf tak-berarah mengacu pada graf yang sisinya tidak mempunyai orientasi atau arah. Pada jenis ini, urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan. Jadi, $(u, v) = (v, u)$ adalah sisi yang sama. Contoh graf tak-berarah dapat dilihat seperti pada gambar 2.3.



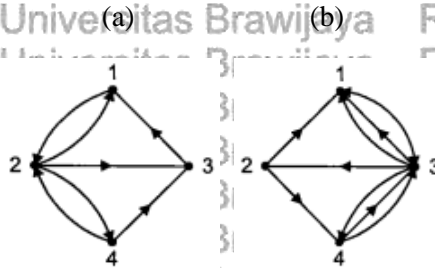
Gambar 2.3 Graf Tak-Berarah

Sumber: Munir, 2010



b. Graf berarah (*directed graph* atau *digraph*)

Kebalikan dari jenis pertama, graf berarah mengacu pada graf yang setiap sisinya diberikan orientasi arah. Umumnya, sisi berarah sering disebut busur (*arc*). Pada graf jenis ini, (u, v) dan (v, u) menyatakan dua buah busur yang berbeda, dengan kata lain $(u, v) \neq (v, u)$. Untuk busur (u, v) , simpul u dinamakan simpul asal (*initial vertex*) dan simpul v dinamakan simpul terminal (*terminal vertex*). Graf berarah sering digunakan untuk menggambarkan aliran proses, peta lalu lintas, dsb. Pada graf berarah, gelang diperbolehkan, tetapi sisi-ganda tidak. Apabila definisi graf diperluas sehingga mencakup graf-ganda berarah (*directed multigraph*), gelang dan sisi-ganda diperbolehkan ada. Contoh graf berarah dan graf-ganda berarah dapat dilihat seperti pada gambar 2.4.



Gambar 2.4 (a) Graf Berarah, (b) Graf Ganda Berarah

Sumber: Mumir, 2010

Untuk meringkas perluasan definisi graf, jenis-jenis graf dan karakteristiknya disajikan pada tabel 2.1.

Tabel 2.1. Definisi Graf

Jenis	Sisi	Sisi ganda dibolehkan?	Sisi gelang dibolehkan?
Graf sederhana	Tak-berarah	Tidak	Tidak
Graf ganda	Tak-berarah	Ya	Tidak
Graf semu	Tak-berarah	Ya	Ya

Tabel 2.1. (lanjutan)

Jenis	Sisi	Sisi ganda dibolehkan?	Sisi gelang dibolehkan?
Graf berarah	Berarah	Tidak	Ya
Graf-ganda berarah	Berarah	Ya	Ya

Sumber: Munir, 2010

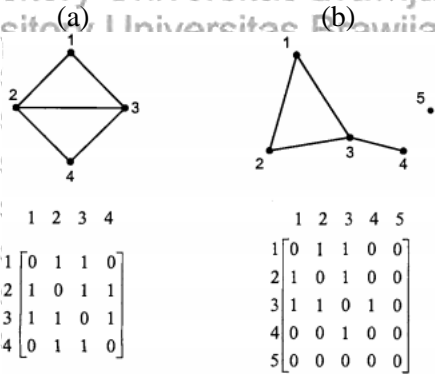
2.3.2 Representasi Graf

Apabila graf akan diproses dengan program komputer, maka graf harus direpresentasikan agar dapat dipahami. Salah satu representasi yang paling sering digunakan untuk memproses graf yaitu matriks ketetanggaan.

Matriks ketetanggaan adalah representasi graf yang paling umum. Misalkan $G = (V, E)$ adalah graf dengan n simpul, $n \geq 1$. Matriks ketetanggaan G adalah matriks berukuran $n \times n$. Bila matriks tersebut dinamakan $A = [a_{ij}]$, maka $a_{ij} = 1$ jika i dan j bertetangga. Sebaliknya, $a_{ij} = 0$ jika simpul i dan j tidak bertetangga.

Karena matriks ketetanggaan hanya berisi 0 dan 1, maka matriks tersebut dinamakan juga matriks nol-satu (*zero-one*). Selain dengan angka 0 dan 1, elemen matriks dapat juga dinyatakan dengan nilai *false* (menyatakan 0) dan *true* (menyatakan 1). Perlu diperhatikan bahwa matriks ketetanggaan didasarkan pada pengurutan nomor simpul. Dengan demikian, terdapat $n!$ cara pengurutan nomor simpul, yang berarti ada $n!$ matriks ketetanggaan berbeda untuk graf dengan n simpul. Untuk menambah pemahaman dalam menyusun matriks ketetanggaan, dapat dilihat pada gambar 2.5.

Matriks ketetanggaan untuk graf sederhana dan tidak berarah selalu simetri, sedangkan matriks ketetanggaan untuk matriks berarah belum tentu simetri (akan simetri jika berupa graf berarah lengkap). Selain itu, diagonal utamanya selalu nol karena tidak ada sisi gelang.

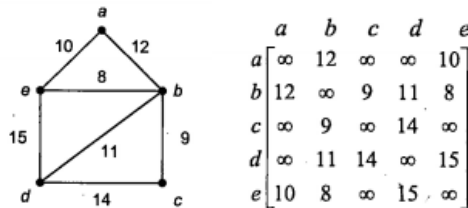


Gambar 2.5. Graf dengan matriks ketetanggaan masing-masing; (a) graf terhubung, (b) graf tak-terhubung, dan (c) graf berarah

Sumber: Munir, 2010

Namun, matriks ketetanggaan nol-satu tidak dapat digunakan untuk merepresentasikan graf yang mempunyai sisi-ganda (graf ganda). Untuk menyiasatinya, maka elemen a_{ij} pada matriks ketetanggaan sama dengan jumlah sisi yang berasosiasi dengan (v_i, v_j) . Matriks ketetanggaannya tentu bukan lagi matriks nol-satu. Untuk graf semu, gelang pada simpul v_i dinyatakan dengan nilai 1 pada posisi (i, i) di matriks ketetanggaannya.

Untuk graf berbobot, a_{ij} menyatakan bobot tiap sisi yang menghubungkan simpul i dengan simpul j . Contoh penulisan matriks ketetanggaan untuk graf berbobot dapat dilihat pada gambar 2.6.



Gambar 2.6. Graf berbobot dan matriks ketetanggaannya

Sumber: Munir, 2010



Tanda “ ∞ ” menyatakan bahwa tidak ada sisi dari simpul i ke simpul j atau dari simpul i ke simpul i itu sendiri, sehingga a_{ij} dapat diberi nilai tak berhingga.

Pada penggunaannya, matriks ketetanggaan akan digunakan untuk merepresentasikan jarak antar lokasi yang terdapat dalam penelitian ini, baik dari depot ke pangkalan, maupun dari pangkalan ke pangkalan lainnya. Matriks ini disebut juga sebagai matriks jarak.

2.4 *Vehicle Routing Problem (VRP)*

Masalah klasik dalam model *vehicle routing problem* umumnya membahas mengenai pencarian rute suatu kendaraan dengan tujuan tertentu. Rute dimulai dan diakhiri di depot (gudang), dengan syarat permintaan dari semua simpul (pelanggan) terpenuhi. Menurut Toth & Vigo (2002), VRP adalah permasalahan yang berkaitan dengan distribusi barang antara depot dan pengguna akhir (pelanggan) yang bertujuan meminimumkan total jarak tempuh kendaraan (biaya transportasi, jarak tempuh kendaraan, dll.).

VRP pertama kali muncul dalam makalah George Dantzig dan John Ramser pada tahun 1959, dimana pendekatan algoritmik pertama ditulis dan diterapkan pada pengiriman bensin. Seringkali, konteks yang digunakan dalam permasalahan VRP adalah pengiriman barang yang berlokasi di depot kepada pelanggan yang telah memesan barang tersebut. Selanjutnya, Clarke dan Wright menyempurnakan pendekatan Dantzig dan Ramser menggunakan pendekatan yang efektif yang disebut algoritma *savings* pada tahun 1964.

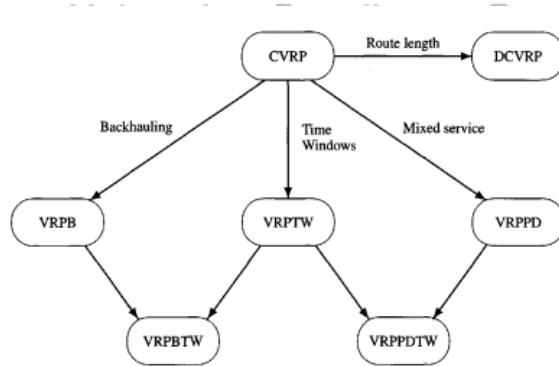
Menurut Toth & Vigo (2002), VRP telah memiliki beberapa variasi permasalahan dengan menyesuaikan kondisi atau batasan yang ada dalam permasalahan penentuan rute yang optimal. Beberapa contoh varian VRP yaitu:

1. Capacitated VRP (CVRP), setiap kendaraan memiliki kapasitas yang terbatas.
2. VRP Time Windows (VRPTW), setiap pelanggan harus disuplai dalam jangka waktu tertentu.



3. VRP Backhauls (VRPB), pelanggan dibagi kedalam dua kelompok.
4. VRP Pick-Up and Delivery (VRPPD), pelanggan mungkin mengembalikan barang ke depot asal.

Hubungan masing-masing variasi permasalahan VRP dapat dilihat pada gambar 2.7.



Gambar 2.7 Variasi Permasalahan VRP dan Hubungannya

Sumber: Toth & Vigo, 2002

Toth & Vigo (2002) menjelaskan bahwa terdapat beberapa karakteristik dalam VRP yang perlu diperhatikan dalam setiap kasus. Berikut merupakan komponen-komponen yang terdapat pada VRP:

1. Jaringan Kerja/Jaringan Jalan (*link*)

Dalam transportasi pada suatu rute, setiap jalan yang tersedia direpresentasikan sebagai sisi pada sebuah graf dan merupakan jaringan kerja/jaringan jalan (*link*). Setiap lokasi merupakan simpul yang merepresentasikan pelanggan dan depot. Setiap sisi dikaitkan dengan biaya, yang umumnya mewakili panjangnya dan waktu tempuh.



2. Pelanggan

Dalam menyelesaikan permasalahan VRP, terlebih dahulu harus menetapkan lokasi pelanggan-pelanggan yang ada. Perlu diperhatikan pula permintaan yang dibutuhkan oleh pelanggan tersebut. Beberapa karakteristik khusus pelanggan yaitu:

- a. Lokasi pelanggan diketahui secara pasti
- b. Jumlah permintaan (*demand*) dari pelanggan berbeda-beda, ada pelanggan yang jumlah permintaannya diketahui secara pasti (kasus deterministik) dan ada pula yang jumlah permintaannya tidak pasti (kasus stokastik).
- c. Ada pelanggan yang mempunyai *time windows* yaitu periode waktu yang menunjukkan jangka waktu pelanggan dapat dilayani.

3. Depot

Depot merupakan titik awal dan akhir dari suatu rute yang akan ditempuh dalam melakukan distribusi. Setiap depot dicirikan berdasarkan tipe dan banyak kendaraan yang berkaitan dengan depot tersebut serta banyaknya barang yang tersedia disana. Perlu diketahui jam operasional yang ditentukan pada depot. Tujuannya untuk membatasi waktu kinerja kendaraan dalam proses distribusi.

4. Kendaraan (*Vehicle*)

Beberapa karakteristik khusus kendaraan antara lain:

- a. Mempunyai kapasitas kendaraan maksimum (berat dan volume maksimum) dalam mengangkut barang. Kapasitas kendaraan ini membatasi permintaan konsumen, artinya jumlah permintaan konsumen tidak boleh melebihi kapasitas kendaraan yang digunakan.
- b. Mempunyai total waktu kerja dari awal keberangkatan dari depot sampai kedatangan kembali ke depot, sesuai peraturan yang diberlakukan oleh perusahaan.



c. Terdapat alat untuk melayani konsumen dan biaya-biaya yang diperlukan untuk melakukan pengiriman (per unit jarak, per unit waktu, per rute, dll.)

5. Pengemudi (*Driver*)

Pengemudi yang mengoperasikan kendaraan harus memenuhi semua kendala yang ditetapkan dalam kontrak kerja dan aturan dari perusahaan. Pengemudi memiliki kendala seperti jam kerja harian, durasi maksimum perjalanan, dan tambahan jam lembur jika diperlukan.

6. Rute

Rute merepresentasikan urutan kunjungan sebuah kendaraan ke pelanggan-pelanggan, dimulai dari depot dan kembali ke depot. Sebuah rute terdiri dari satu depot dan setidaknya satu pelanggan.

Lebih sederhana, berikut adalah karakteristik dari permasalahan VRP secara ringkas.

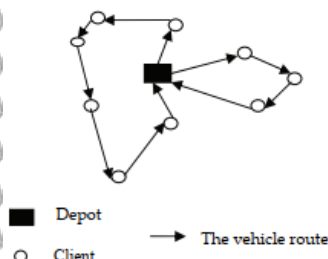
1. Perjalanan kendaraan berawal dan berakhir dari dan ke depot (gudang) awal.
2. Ada sejumlah tempat yang telah terdaftar sebelumnya yang seluruhnya harus dikunjungi dan dipenuhi permintaannya tepat satu kali.
3. Apabila kapasitas kendaraan sudah terpakai dan tidak dapat melayani tempat berikutnya, kendaraan dapat kembali ke depot untuk memenuhi kapasitas kendaraan dan melayani tempat berikutnya.
4. Tujuan dari permasalahan ini adalah untuk meminimumkan total jarak tempuh dengan mengatur urutan tempat yang harus dikunjungi dan kapan kembalinya kendaraan untuk mengisi kapasitas lagi.

Capacitated Vehicle Routing Problem (CVRP) merupakan permasalahan yang paling dasar dalam model VRP. Toth & Vigo



(2002) menjelaskan bahwa dalam CVRP, semua pelanggan sesuai dengan pengiriman dan permintaan telah ditentukan, telah diketahui sebelumnya, dan tidak boleh dibagi. Kendaraan yang digunakan dalam distribusi identik (homogen) dan berbasis di satu depot pusat, dan hanya pembatasan untuk kendaraan yang diberlakukan. Tujuan CVRP adalah untuk meminimalkan biaya total (yaitu, fungsi bobot dari jumlah rute dan panjang atau waktu tempuhnya) untuk melayani semua pelanggan. Dengan demikian, kendala yang perlu diperhatikan hanya pada batasan kapasitas kendaraan yang perlu dipertimbangkan. Contoh solusi permasalahan CVRP dapat dilihat pada gambar 2.8.

Menurut Caric dan Gold (2008), CVRP dapat direpresentasikan sebagai graf berarah berbobot $G = (V, A)$ dimana $V = \{v_0, v_1, v_2, \dots, v_n\}$ merepresentasikan set dari simpul dan $A = \{(v_i, v_j)\}$ merepresentasikan set dari busur atau sisi. Titik v_0 merepresentasikan depot dan simpul lainnya merepresentasikan konsumen. Untuk setiap sisi (v_i, v_j) diberikan nilai non-negatif c_{ij} . Nilai ini dapat berupa biaya perjalanan atau jarak tempuh atau waktu tempuh yang dibutuhkan dari simpul i ke simpul j , setiap titik $v_i \in V$ memiliki permintaan (*demand*) sebagai d_i .



Gambar 2.8 Contoh solusi CVRP yang terdiri dari 2 rute

Sumber: Toth & Vigo, 2002

Secara umum, penggunaan busur gelang (*loop*) tidak diperbolehkan dalam CVRP, hal ini ditunjukkan dengan mendefinisikan $c_{ii} = +\infty, i \in V$. Grafik G harus terhubung kuat dan umumnya dianggap lengkap. Himpunan $K = (k_1, k_2, \dots, k_m)$



merupakan himpunan kendaraan yang homogen dengan kapasitas yang identik yaitu Q , sehingga panjang setiap rute dibatasi oleh kapasitas kendaraan. Menurut Toth & Vigo (2002), permasalahan CVRP adalah menentukan himpunan dari K rute yang memiliki kondisi berikut.

1. Setiap rute berawal dan berakhir di depot.
2. Setiap konsumen harus dilayani tepat satu kali oleh satu kendaraan.
3. Total permintaan konsumen dari setiap rute tidak melebihi kapasitas kendaraan.
4. Total jarak dari setiap rute diminimalkan.

Selanjutnya, permasalahan tersebut kemudian diformulasikan ke dalam suatu model matematika dengan tujuan untuk meminimalkan biaya distribusi yang perlu dikeluarkan.

Didefinisikan variabel keputusannya adalah:

$$x_{ij}^k = \begin{cases} 1, & \text{jika ada perjalanan dari konsumen } i \\ & \text{ke konsumen } j \text{ oleh kendaraan } k \\ 0, & \text{jika tidak ada perjalanan dari konsumen } i \\ & \text{ke konsumen } j \text{ oleh kendaraan } k \end{cases}$$

Keterangan:

$K = (k_1, k_2, \dots, k_m)$ kendaraan yang digunakan

$V =$ himpunan titik/konsumen

$c_{ij} =$ biaya perjalanan yang diperlukan dari konsumen i ke konsumen j

$d_i =$ jumlah permintaan konsumen i

$Q =$ kapasitas masing-masing kendaraan



x_{ij}^k = kendaraan k melakukan perjalanan dari konsumen i ke konsumen j .

u_i^k = sisa kapasitas kendaraan k setelah melayani pelanggan i .

Selanjutnya, fungsi tujuannya adalah meminimumkan biaya distribusi dari perjalanan kendaraan. Jika z adalah fungsi tujuan, maka fungsi tujuan dapat ditulis dalam bentuk model matematis pada persamaan 2.1.

$$\text{Meminimumkan } z = \sum_{k \in K}^m \sum_{i \in V}^n \sum_{j \in V}^n (c_{ij} x_{ij}^k) \quad (2.1)$$

dengan kendala-kendala:

1. Variabel keputusan x_{ij}^k merupakan integer biner. Fungsi kendala ini dapat ditulis dalam bentuk model matematis pada persamaan 2.2.

$$x_{ij}^k \in \{0,1\}, \forall i, j \in V; i \neq j, K = \{k_1, k_2, \dots, k_m\} \quad (2.2)$$

2. Setiap rute hanya dikunjungi tepat satu kali oleh satu kendaraan. Jika x_{ij}^k bernilai 1, maka terdapat perjalanan dari konsumen i ke konsumen j . Dengan demikian, $u_i^k = 1$. Sebaliknya, apabila x_{ij}^k bernilai 0, maka tidak terdapat perjalanan dari konsumen i ke konsumen j , $u_i^k = 0$. Dengan demikian, dapat dikatakan bahwa variabel x_{ij}^k dan variabel u_i^k saling berhubungan. Fungsi kendala ini dapat ditulis dalam bentuk model matematis pada persamaan 2.3.

$$\sum_{k \in K} \sum_{j \in V, i \neq j} x_{ij}^k = 1, \forall i \in V \quad (2.3)$$

3. Total jumlah permintaan konsumen dalam satu rute tidak melebihi kapasitas kendaraan yang melayani rute tersebut. Kapasitas kendaraan untuk memenuhi permintaan konsumen harus dimaksimalkan namun tidak lebih dari kapasitas



kendaraan tersebut. Fungsi kendala ini dapat ditulis dalam bentuk model matematis pada persamaan 2.4.

$$\sum_{i \in V} \sum_{j \in V, j \neq i} d_i x_{ij}^k \leq Q, \forall k \in K \quad (2.4)$$

4. Setiap rute perjalanan kendaraan berawal dari depot. Fungsi kendala ini dapat ditulis dalam bentuk model matematis pada persamaan 2.5.

$$\sum_{k=1}^m \sum_{i=0}^n x_{0i}^k = \sum_{k \in K} \sum_{i \in V} x_{0i}^k = 1 \quad (2.5)$$

5. Setiap rute perjalanan kendaraan berakhir di depot. Fungsi kendala ini dapat ditulis dalam bentuk model matematis pada persamaan 2.6.

$$\sum_{k=1}^m \sum_{i=0}^n x_{in+1}^k = \sum_{k \in K} \sum_{i \in V} x_{in+1}^k = 1 \quad (2.6)$$

6. Kendaraan yang mengunjungi satu titik, setelah selesai melayani akan meninggalkan titik tersebut. Fungsi kendala ini dapat ditulis dalam bentuk model matematis pada persamaan 2.7.

$$\sum_{i \in V} x_{ij}^k - \sum_{j \in V} x_{ji}^k = 0, \forall i, j \in V, \forall k \in K \quad (2.7)$$

7. Batasan ini memastikan bahwa tidak terdapat subrute pada setiap rute yang terbentuk. Fungsi kendala ini dapat ditulis dalam bentuk model matematis pada persamaan 2.8 dan 2.9.

$$x_{ij}^k = 1 \Rightarrow u_i^k - d_j = u_j^k, \forall i, j \in V, \forall k \in K \quad (2.8)$$

$$u_0 = Q, 0 \leq u_i, \forall i \in V \quad (2.9)$$

8. Biaya yang dikeluarkan untuk mengunjungi dirinya sendiri adalah tak hingga. Fungsi kendala ini dapat ditulis dalam bentuk model matematis pada persamaan 2.10.

$$c_{ii} = +\infty, i \in V \quad (2.10)$$



Menggunakan formulasi model matematis CVRP tidak terdapat subrute pada rute-rute yang terbentuk yang dikaitkan dengan batasan kapasitas kendaraan. Variabel keputusan hanya akan terdefinisi jika jumlah permintaan konsumen i dan konsumen j tidak melebihi kapasitas kendaraan.

2.5 Algoritma *Nearest Neighbour*

Algoritma *nearest neighbour* yang pertama kali diperkenalkan pada tahun 1983 merupakan algoritma yang sederhana. Pada setiap iterasinya, dilakukan pencarian konsumen terdekat dengan konsumen terakhir untuk ditambahkan pada akhir rute tersebut. Algoritma *nearest neighbour* selalu menambahkan lokasi yang jaraknya paling dekat dengan lokasi terakhir yang telah dikunjungi.

Langkah awal dalam melakukan algoritma ini adalah keberangkatan kendaraan pengangkut yang berangkat dari depot (gudang), langkah selanjutnya adalah mencari lokasi konsumen dengan jarak paling dekat dengan titik awal/depot. Setelah suatu rute selesai terkait kendala kapasitas, rute yang baru akan dimulai kembali dengan cara yang sama.

Menurut Pop (2011), langkah-langkah dalam algoritma *nearest neighbour* adalah sebagai berikut.

1. Menentukan data pelanggan, jumlah permintaan, dan kapasitas kendaraan sebagai input yang dibutuhkan.
2. Pembentukan rute pertama ($t = 1$).
3. Tentukan pelanggan pertama yang ditugaskan pada rute dengan cara memilih pelanggan yang memiliki jarak terpendek dari depot.
4. Hitung banyaknya jumlah permintaan dari konsumen yang telah terpilih. Apabila jumlah permintaan masih memenuhi kapasitas kendaraan maka lanjut ke langkah 5. Apabila jumlah permintaan melebihi kapasitas kendaraan maka dilanjutkan ke langkah 6.
5. Lanjutkan ke pelanggan berikutnya yang memiliki jarak terpendek dari pelanggan terpilih sebelumnya.
6. Pembentukan rute baru ($t = t + 1$), lanjut ke langkah 3.



BAB III METODE PENELITIAN

3.1 Sumber Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari PT. Garuda Mandiri dan hasil wawancara dengan pihak perusahaan. Data yang digunakan adalah milik perusahaan pada bulan Februari 2021. Data yang diperoleh adalah sebagai berikut.

1. Lokasi subagen (pangkalan) yang telah terdaftar

Data lokasi subagen (pangkalan) didapat dari perusahaan dan akan dipergunakan untuk menentukan jarak tempuh yang perlu dilalui kendaraan dari satu lokasi ke lokasi lainnya. Subagen dinotasikan sebagai i dan j , serta merupakan anggota dari himpunan V .

2. Jarak tempuh

Jarak tempuh merupakan jarak dari setiap lokasi ke lokasi yang dibutuhkan selama proses pendistribusian dilakukan. Jarak tempuh tersebut diperoleh dengan menggunakan bantuan *website google maps* dengan satuan kilometer. Dalam penelitian ini, jarak yang perlu ditempuh kendaraan dalam melakukan distribusi akan dihitung berdampingan dengan biaya distribusi. Selain itu, jarak tempuh juga digunakan sebagai salah satu aspek untuk menghitung perkiraan biaya distribusi yang diperlukan. Data jarak tempuh yang telah diperoleh dapat dilihat pada Lampiran 1.

3. Bahan bakar

Berdasarkan hasil wawancara dengan perusahaan, diperoleh bahwa penggunaan bahan bakar secara rata-rata yaitu Rp 1.000,-/kilometer. Biaya bahan bakar akan digunakan sebagai salah satu aspek untuk menghitung perkiraan biaya distribusi yang diperlukan.

4. Uang jalan

Berdasarkan hasil wawancara dengan perusahaan, diperoleh bahwa uang jalan setiap mobil yang beroperasi dalam satu hari adalah sebesar Rp 80.000,-. Setiap supir dan kenek yang beroperasi



memperoleh uang jalan per kendaraan yang beroperasi. Masing-masing kendaraan yang beroperasi mendapat satu kali uang jalan dalam satu hari. Biaya uang jalan akan digunakan sebagai salah satu aspek untuk menghitung perkiraan biaya distribusi yang diperlukan.

5. Biaya distribusi

Biaya distribusi setiap lokasi dihitung dengan mempertimbangkan jarak tempuh, biaya bahan bakar, dan uang jalan. Biaya distribusi dari lokasi i ke lokasi j dilambangkan dengan notasi c_{ij} .

6. Jumlah permintaan setiap subagen (pangkalan)

Data permintaan diperoleh dari perusahaan. Data ini terdiri dari jadwal dan jumlah permintaan setiap subagen di PT. Garuda Mandiri. Pengiriman tabung gas ini dilakukan setiap hari Senin sampai dengan Sabtu dalam sebulan dengan jumlah permintaan yang sudah ditetapkan. Jumlah permintaan masing-masing pangkalan diberi notasi d_i . Data jumlah permintaan setiap subagen pada bulan Februari 2021 dapat dilihat pada Lampiran 2.

7. Kapasitas kendaraan

PT. Garuda Mandiri memiliki 5 buah kendaraan berupa truk angkut dengan kapasitas angkut sebesar 560 tabung gas LPG 3 Kg setiap pengangkutan. Setiap kendaraan diberi notasi k . Sedangkan untuk kapasitas setiap kendaraan diberikan notasi Q_k .

8. Laporan pengeluaran perusahaan bulan Februari 2021

Laporan pengeluaran perusahaan diperoleh dari PT. Garuda Mandiri guna membandingkan seberapa besar selisih pengeluaran antara metode klasik yang digunakan perusahaan dan metode *nearest neighbour* yang digunakan dalam penelitian. Laporan pengeluaran perusahaan dapat dilihat pada Lampiran 3.

3.2 Penyelesaian Algoritma *Nearest Neighbour*

Berdasarkan langkah-langkah penyelesaian yang telah dijelaskan pada subbab 2.5, maka penyelesaian algoritma *nearest neighbour* dalam penelitian ini adalah sebagai berikut.



1. Menentukan data lokasi pangkalan, jumlah permintaan, setiap pangkalan, kapasitas kendaraan, serta jarak antar lokasi terlebih dahulu sebagai input yang dibutuhkan.
2. Memilih depot/gudang sebagai titik pusat untuk titik awal pengiriman.
3. Berawal dari depot, selanjutnya dilakukan pencarian lokasi pangkalan yang belum dikunjungi yang memiliki jarak terpendek dari depot sebagai lokasi pertama.
4. Lanjutkan ke pangkalan lain yang memiliki jarak terdekat dari pangkalan yang terpilih sebelumnya dan jumlah permintaan pangkalan tidak melebihi kapasitas kendaraan.
 - a. Apabila ada pangkalan yang terpilih sebagai kunjungan berikutnya dan terdapat sisa kapasitas kendaraan, lakukan kembali langkah 4.
 - b. Bila kendaraan tidak memiliki sisa kapasitas, kembali ke langkah 2.
 - c. Bila tidak ada lokasi yang terpilih karena jumlah pengiriman melebihi kapasitas kendaraan, maka kembali ke langkah 2. Dimulai kembali dari gudang dan mengunjungi pangkalan yang belum dikunjungi yang memiliki jarak terdekat.
5. Apabila semua pangkalan telah dikunjungi tepat satu kali, maka algoritma berakhir.

3.3 Metode Analisis Data

Langkah-langkah analisis data yang dilakukan dalam penelitian ini adalah sebagai berikut.

1. Menentukan data lokasi pangkalan, jumlah permintaan setiap pangkalan, kapasitas kendaraan, serta jarak antar lokasi terlebih dahulu sebagai input yang dibutuhkan.



2. Memperoleh data jarak menggunakan lokasi pangkalan dengan bantuan *software google maps* dan *Ms. Excel*. Dilanjutkan dengan merepresentasikan data yang diperoleh dalam bentuk matriks ketetanggaan yang disebut juga sebagai matriks jarak.
3. Membentuk model matematika fungsi tujuan dan fungsi kendala yang digunakan untuk menemukan solusi pembentukan rute yang memberikan hasil yang meminimalkan biaya distribusi. Sesuai dengan persamaan 2.1, fungsi tujuan dalam penelitian ini secara umum yaitu:

$$\text{Meminimumkan } z = \sum_{k=1}^5 \sum_{i=1}^{61} \sum_{j=0}^{61} (c_{ij} x_{ij}^k)$$

Sesuai dengan persamaan 2.2 sampai dengan 2.10, fungsi kendala dalam penelitian ini secara umum yaitu:

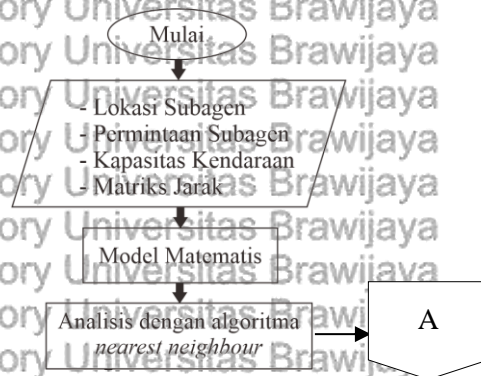
- a. $x_{ij}^k \in \{0,1\}, \forall i, j \in V = \{1,2, \dots, 61\}; i \neq j, K = \{1,2,3,4,5\}$.
 - b. $\sum_{k=1}^5 \sum_{i,j=0, i \neq j}^{61} x_{ij}^k = 1$
 - c. $\sum_{i=0}^{61} \sum_{j=0, j \neq i}^{61} d_i x_{ij}^k \leq Q$
 - d. $\sum_{k=1}^5 \sum_{i=0}^{61} x_{0i}^k = 1$
 - e. $\sum_{k=1}^5 \sum_{i=0}^{61} x_{in+1}^k = 1$
 - f. $\sum_{i=0}^{61} x_{ij}^k = \sum_{j=0}^{61} x_{ji}^k$
 - g. $x_{ij}^k = 1 \Rightarrow u_i^k - d_j = u_j^k, \forall i, j \in V = \{1,2, \dots, 61\}, \forall k \in K = \{1,2,3,4,5\}$
 - h. $u_0 = Q, 0 \leq u_i, \forall i \in V = \{1,2, \dots, 61\}$
 - i. $c_{ii} = +\infty, i \in V = \{1,2, \dots, 61\}$
4. Penyelesaian analisis dan pembentukan rute dilakukan dengan menggunakan algoritma *nearest neighbour* dari penelitian yang dilakukan oleh Rasku pada tahun 2019. (Rasku, J., dkk., 2019) yang diimplementasikan dalam *software python*. Selanjutnya, hasil yang diperoleh dilakukan interpretasi. Langkah-langkah penyelesaian menggunakan algoritma *nearest neighbour* dapat dilihat pada subbab 3.2.



5. Dari hasil yang telah diperoleh sebelumnya, didapat solusi rute distribusi yang dapat meminimalkan biaya distribusi yang perlu dikeluarkan.
6. Menggambarkan rute yang telah dihasilkan dalam bentuk urutan kunjungan pada peta agar lebih mudah dipahami.
7. Langkah terakhir, dilakukan perbandingan hasil jarak dan biaya yang perlu dikeluarkan yang dihitung menggunakan algoritma *nearest neighbour* dengan jarak dan biaya yang dikeluarkan perusahaan pada bulan Februari 2021 untuk mengetahui rute mana yang memberikan hasil lebih baik.

3.4 Diagram Alir Penelitian

Langkah-langkah penelitian ini dapat dilihat lebih ringkas pada gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian



BAB IV HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai data yang telah dikumpulkan dan akan diolah sesuai dengan metodologi penelitian yang telah disusun pada bab sebelumnya. Setelah dilakukan pengolahan data, selanjutnya akan dilakukan interpretasi dan analisis hasil penelitian.

4.1 Gambaran Umum Perusahaan

4.1.1 Profil Perusahaan

PT. Garuda Mandiri merupakan salah satu Agen Gas LPG 3 Kg yang beroperasi di Kota Pekanbaru. PT. Garuda Mandiri bertugas untuk mendistribusikan gas LPG 3 Kg dari SPPBE menuju konsumen yang merupakan Pangkalan Gas LPG 3 Kg yang sudah terdaftar sebelumnya di PT. Pertamina. Perusahaan tersebut membawahi 61 pangkalan yang telah terdaftar dan beroperasi di Kota Pekanbaru serta bertanggungjawab melakukan distribusi dari SPPBE ke 61 pangkalan tersebut.

Dalam pelaksanaan distribusinya, perusahaan melakukan pengisian ulang tabung gas LPG 3 Kg di SPPBE PT. Awal Bros Bumi Pusaka untuk didistribusikan ke konsumen. Setiap bulannya, perusahaan terlebih dahulu melakukan pemesanan pada bulan sebelumnya ke SPPBE agar diberi persediaan gas LPG 3 Kg selama satu bulan berjalan. Pada penyediaan tabung gas LPG 3 Kg untuk bulan Februari 2021, perusahaan harus melakukan pemesanan dengan jumlah yang telah ditentukan pada bulan sebelumnya. Perusahaan melakukan pemesanan sebanyak 59.360 tabung gas LPG 3 Kg atau sebanyak 106 LO untuk didistribusikan selama bulan Februari.

4.1.2 Rute Distribusi Perusahaan

Penentuan rute pendistribusian produk menuju pangkalan selama ini dilakukan oleh masing-masing pengemudi. Setiap pengemudi bebas untuk menentukan rute yang dilalui berdasar pengalaman dan perkiraan pengemudi tersebut. Hal ini dapat



menimbulkan resiko total jarak yang ditempuh akan menjadi lebih besar dibandingkan total jarak yang seharusnya.

Setiap rute yang dilalui kendaraan dimulai dari depot/gudang dengan membawa 560 tabung kosong per kendaraan yang beroperasi menuju SPPBE untuk dilakukan pengisian ulang tabung gas LPG 3 Kg dan selanjutnya akan didistribusikan ke pangkalan tujuan. Apabila sisa kapasitas kendaraan tidak mencukupi untuk melayani pangkalan selanjutnya, kendaraan kembali ke depot, dan rute berakhir. Dalam kondisi lain, apabila seluruh pangkalan telah dilayani, kendaraan kembali ke depot dan rute tersebut berakhir. Sisa kapasitas kendaraan akan disimpan di gudang terlebih dahulu sebelum didistribusikan di hari selanjutnya.

Setiap kendaraan yang beroperasi pada hari tersebut, pengemudi dan kenek akan diberi uang jalan sebesar Rp 80.000,- per kendaraan per hari. Apabila pada suatu hari terdapat 3 rute yang perlu dilalui, maka hanya diperlukan tiga kendaraan yang beroperasi dan uang jalan akan diberikan pada sopir dan kenek yang bertugas pada saat itu masing-masing sebesar Rp 80.000,- per kendaraan. Apabila rute yang perlu dilalui lebih dari 5 rute, kendaraan yang sampai ke gudang lebih dulu akan menjalankan rute yang belum dilalui, dan seterusnya. Tabel 4.1 merupakan daftar rute distribusi eksisting perusahaan untuk pendistribusian pada tanggal 1 Februari 2021.

Tabel 4.1. Rute Distribusi Perusahaan

Kendaraan	Rute	Jarak Tempuh (Km)	Kapasitas Terpakai
1	Depot – SPPBE – 2 – 18 – 44 – 51 – 54 – 9 – Depot	104,6	92,8%
2	Depot – SPPBE – 4 – 31 – 41 – 55 – Depot	114,8	100%



Tabel 4.1. (lanjutan)

Kenda- raan	Rute	Jarak Tempuh (Km)	Kapasitas Terpakai
3	Depot – SPPBE – 12 40 – 57 – Depot	56,6	100%
4	Depot – SPPBE – 13 23 – 28 – 37 – Depot	106,2	100%
	Total	382,2	

Perhitungan jarak tempuh didasarkan pada jarak yang terdapat dalam matriks jarak pada Lampiran 1. Contoh perhitungan total jarak tempuh rute 3 (Depot – SPPBE – 12 – 40 – 57 – Depot) adalah sebagai berikut.

1. Jarak depot menuju SPPBE = 18,3 Km
2. Jarak SPPBE menuju pangkalan 12 = 22,5 Km
3. Jarak pangkalan 12 menuju pangkalan 40 = 10,2 Km
4. Jarak pangkalan 40 menuju pangkalan 57 = 0,2 Km
5. Jarak pangkalan 57 menuju depot = 5,4 Km
6. Total jarak tempuh rute 3 = $18,3 + 22,5 + 10,2 + 0,2 + 5,4 = 56,6$ Km

4.1.3 Perhitungan Biaya Distribusi Perusahaan

Berikut disajikan perhitungan biaya yang harus dikeluarkan perusahaan untuk menjalankan aktivitas distribusi berdasarkan rute perusahaan yang tersedia pada Tabel 4.1. Perhitungan biaya distribusi dilakukan berdasarkan pada besaran biaya dari aspek-aspek yang telah dijelaskan pada bab 3.1 yaitu total jarak tempuh, biaya bahan bakar, dan uang jalan. Tabel 4.2 merupakan perhitungan biaya distribusi untuk rute perusahaan



Tabel 4.2. Biaya Rute Distribusi Perusahaan

Kendaraan	Rute		Jarak Tempuh (A) (Km)	Biaya Bahan Bakar (A x Rp 1000,-)	Uang Jalan (Rp 80.000,- /kendaraan)	Total
	1	Depot – SPPBE – 2 – 18 – 44 – 51 – 54 – 9		104,6	Rp 104.600,-	Rp 80.000,-
2	Depot – SPPBE – 4 – 31 – 41 – 55 – Depot		114,8	Rp 114.800,-	Rp 80.000,-	Rp 194.800,-
3	Depot – SPPBE – 12 – 40 – 57 – Depot		56,6	Rp 56.600,-	Rp 80.000,-	Rp 136.600,-
4	Depot – SPPBE – 13 – 23 – 28 – 37 – Depot		106,2	Rp 106.200,-	Rp 80.000,-	Rp 186.200,-
Total						Rp 702.200,-

Pada bulan Februari 2021, perusahaan mengeluarkan dana untuk biaya distribusi sebesar Rp 17.080.000,-. Biaya distribusi yang dimaksud sesuai yang telah dijelaskan pada bab III terdiri dari biaya bahan bakar sebesar Rp 8.600.000,- dan uang jalan sebesar Rp 8.480.000,-. Data pengeluaran perusahaan dapat dilihat pada Lampiran 3. Perhitungan penggunaan bahan bakar sepanjang bulan Februari 2021 adalah sebagai berikut.

1. Biaya bahan bakar = Rp 1000,-/Km
2. Penggunaan bahan bakar = $\frac{Rp\ 8.600.000,-}{Rp\ 1.000,-} = 8.600\ Km$

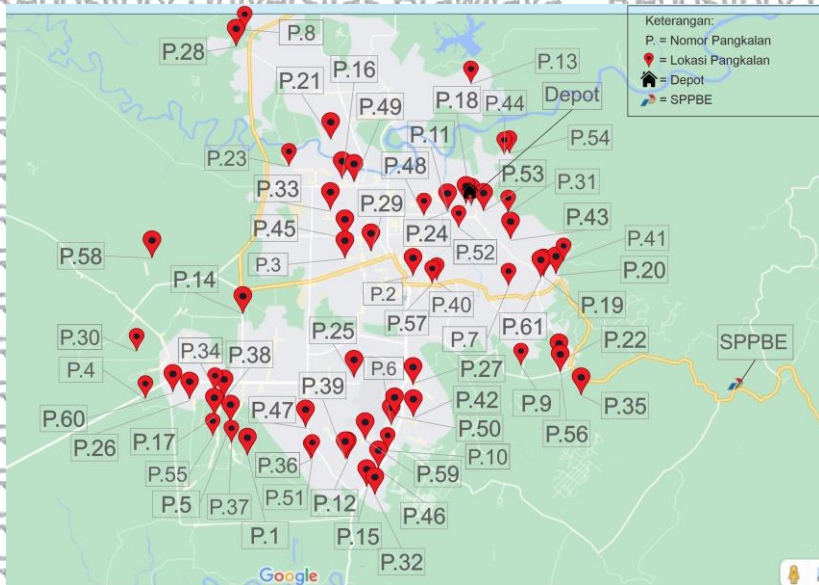
4.2. Pengolahan Data

Pada tahap ini dilakukan pengolahan terhadap data yang telah dikumpulkan pada subbab 3.1. Pengolahan data dilakukan untuk



menentukan rute distribusi yang dapat meminimalkan biaya distribusi sesuai dengan metodologi yang dijelaskan pada bab III.

Terdapat 61 pangkalan yang perlu dilayani dalam pendistribusian tabung gas LPG 3 Kg. Seluruh pangkalan itu tersebar disekitar kota Pekabaru. Peta lokasi depot, SPPBE, dan pangkalan yang perlu dikunjungi dapat dilihat pada gambar 4.1.



Gambar 4.1 Peta lokasi depot, SPPBE, dan pangkalan

Sumber: Google Maps

4.2.1 Algoritma *Nearest Neighbour*

Algoritma yang digunakan untuk menyelesaikan permasalahan distribusi tabung gas LPG 3 Kg pada PT. Garuda Mandiri dalam penelitian ini adalah algoritma *nearest neighbour*. Dalam mengaplikasikan algoritma *nearest neighbour* untuk memperoleh rute distribusi dengan jarak terpendek, digunakan bantuan dari *software python* dengan *input* algoritma *nearest neighbour* seperti yang tersedia



pada Lampiran 6. Model matematis permasalahan dalam penelitian ini dapat dilihat pada subbab 3.2.

Untuk menjalankan algoritma, setiap pangkalan yang akan dikunjungi diberi kode angka $\{0, 1, 2, \dots, n\}$ dengan n merupakan jumlah pangkalan yang perlu dilayani pada hari tersebut. Kode 0 diberikan pada SPPBE atau depot, sesuai dengan kondisi yang berlaku. Dengan kondisi kendaraan harus mengunjungi SPPBE untuk melakukan isi ulang, maka kode 0 diberikan untuk mendefinisikan SPPBE. Dalam kondisi lainnya, kode 0 digunakan untuk mendefinisikan depot.

Sesuai dengan informasi yang diberikan pada subbab 4.1.1, perusahaan telah memesan persediaan gas LPG 3 Kg sebanyak 106 LO untuk bulan Februari 2021. Dengan demikian, rute yang menggunakan SPPBE sebagai titik 0 dibatasi sampai pada rute ke-106 yang terbentuk. Selanjutnya, titik 0 akan digunakan untuk mendefinisikan depot.

Untuk membentuk rute distribusi menggunakan perhitungan algoritma *nearest neighbour*, diperlukan data koordinat lokasi menggunakan format *decimal degree*. Data koordinat ini diperoleh bersama dengan perhitungan jarak melalui bantuan *software google maps*. Data koordinat dimasukkan ke dalam input pada *software python* untuk memperoleh rute distribusi sesuai dengan perhitungan tetangga terdekat.

Pada perhitungan jarak, koordinat dalam *decimal degree* antar lokasi diubah terlebih dahulu menjadi radian. Selanjutnya, dilakukan perhitungan jarak dengan menggunakan formula *haversine*. Dalam rumus *haversine*, terlebih dahulu perlu dicari nilai d dan r . Nilai d merupakan jarak antara dua titik di sepanjang lingkaran besar bola dengan mencari nilai selisih antara koordinat lintang titik A dan titik B, serta selisih koordinat bujur titik A dan titik B dalam radian. Nilai r merupakan jari-jari bola yang dalam kasus ini merupakan jari-jari bumi yaitu sebesar 6.371 Km. Dengan memasukkan nilai-nilai tersebut ke dalam formula *haversine*, akan diperoleh jarak dalam satuan Km.



Output yang diperoleh adalah rute distribusi menggunakan algoritma *nearest neighbour* dengan kode angka. Dengan demikian, rute dengan kode angka perlu disesuaikan terlebih dahulu dengan masing-masing lokasi yang sebenarnya. Total jarak tempuh setiap rute kemudian dihitung menggunakan data jarak yang tersedia pada Lampiran 1.

4.2.2 Alokasi Pangkalan ke dalam Rute

Melalui koordinat lokasi yang diinput, *software python* akan secara otomatis memetakan jarak antar lokasi. Selanjutnya, algoritma akan berjalan dan mulai membentuk satu per satu rute dengan menambahkan lokasi yang memiliki jarak terpendek dari lokasi yang terakhir dikunjungi. Rute akan terus dibentuk sampai mencapai batas kapasitas kendaraan sebanyak 560 tabung. Apabila total permintaan belum mencapai 560 tabung, maka rute akan terus dibentuk. Sebaliknya, apabila total permintaan telah mencapai 560 tabung, maka pembentukan rute dihentikan. Sisa persediaan kendaraan setelah rute berakhir di simpan di gudang untuk didistribusikan di lain hari.

Rute distribusi yang telah disesuaikan dengan lokasi tujuan disajikan pada tabel 4.3.

Tabel 4.3. Rute Distribusi *Nearest Neighbour*

Tanggal	Pangkalan yang Akan Dikunjungi	Ken- da- raan	Rute	Jarak Tempuh Rute (Km)	
1 Februari 2021	2, 4, 9, 12, 13, 18, 23, 28, 31, 37, 40, 41, 44, 51, 54, 55, 57	1	1	Depot - SPPBE - 41 - 9 - 31 - 54 - 44 - depot	52,6
		2	2	Depot - SPPBE - 40 - 18 - 13 - depot	65,7
		3	3	Depot - SPPBE - 57 - 2 - 12 - depot	61,4
		4	4	Depot - SPPBE - 51 - 23 - 28 - 37 - depot	97,9
		5	5	Depot - SPPBE - 55 - 4 - depot	72,3



Tabel 4.3. (lanjutan)

Tanggal	Pangkalan yang Akan Dikunjungi	Kendaraan	Rute	Jarak Tempuh Rute (Km)
2 Februari 2021	1, 5, 17, 19, 20, 25, 26, 35, 38, 42, 43, 45, 46, 47, 50, 60	1	Depot - SPPBE - 35 - 20 - 19 - 43 - depot	40
		2	Depot - SPPBE - 42 - 50 - 46 - 25 - depot	67,7
		3	Depot - SPPBE - 5 - 1 - 47 - 45 - depot	70,5
		4	Depot - SPPBE - 38 - 17 - 26 - depot	67,1
		5	Depot - SPPBE - 60 - depot	66,6
3 Februari 2021	2, 3, 6, 11, 16, 18, 22, 24, 28, 29, 33, 43, 48, 49, 56, 61	1	Depot - SPPBE - 56 - 22 - 61 - 43 - depot	38,8
		2	Depot - SPPBE - 18 - 24 - 11 - 48 - depot	41,6
		3	Depot - SPPBE - 2 - 6 - 3 - depot	57,4
		4	Depot - SPPBE - 28 - 33 - 16 - 49 - 29 - depot	75,9
27 Februari 2021	6, 9, 11, 12, 14, 15, 16, 17, 21, 24, 26, 28, 29, 32, 35, 38, 43, 45, 46, 47, 56, 59, 60	1	Depot - 24 - 11 - 43 - 9 - depot	21,6
		2	Depot - 6 - 46 - 32 - depot	34,3
		6	Depot - 28 - 38 - 17 - 26 - depot	51,4
		3	Depot - 29 - 45 - 16 - 21 - depot	22,7
		4	Depot - 56 - 35 - 15 - 59 - depot	45,9
		5	Depot - 12 - 47 - 14 - depot	41,3
		7	Depot - 60 - depot	37,5
Total kendaraan beroperasi		113	Total Jarak tempuh rute (Km)	7599,32

Tabel lengkap hasil pembentukan rute distribusi menggunakan algoritma *nearest neighbour* dapat dilihat pada Lampiran 4. Dari tabel 4.3, diperoleh total rute terbentuk melalui algoritma *nearest neighbour* untuk bulan Februari 2021 adalah sebanyak 127 rute. Terdapat

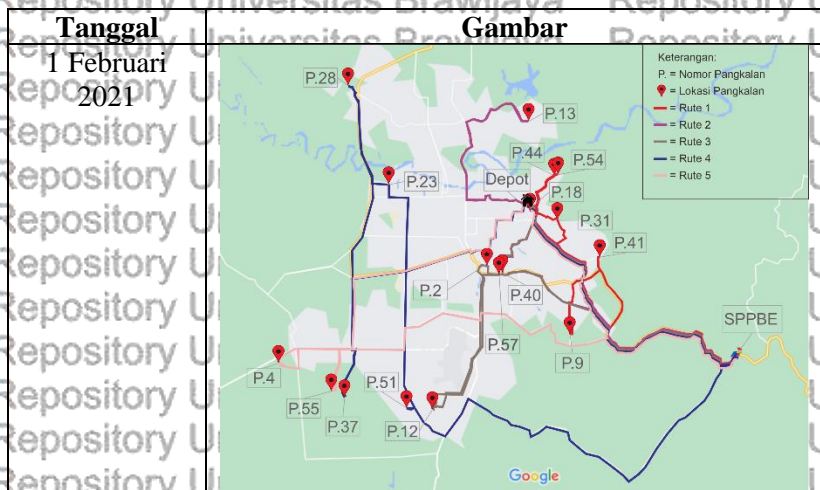


beberapa kendaraan yang melayani dua rute dalam satu hari. Pemilihan kendaraan yang melayani dua rute dalam tulisan ini didasarkan pada rute dengan jumlah pangkalan yang perlu dilayani paling sedikit. Pada pelaksanaannya, pemilihan kendaraan yang melalui dua rute dapat dilakukan oleh kendaraan yang lebih dulu kembali ke gudang.

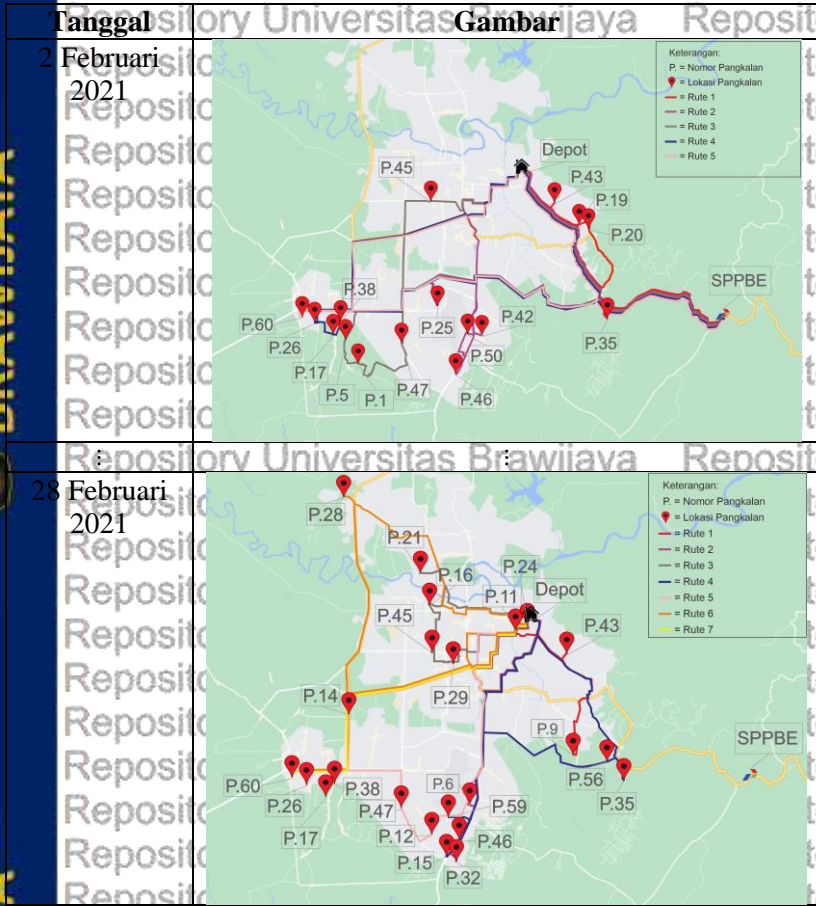
Karena pemesanan yang dilakukan perusahaan terbatas sebanyak 106 LO, maka pada rute ke-107 yaitu rute 2 yang terbentuk pada tanggal 24 Februari, titik SPPBE tidak lagi dikunjungi. Total kendaraan yang beroperasi sepanjang bulan Februari sebanyak 113 kendaraan. Total jarak tempuh sepanjang bulan Februari adalah sebesar 7.599,32 Km.

Visualisasi rute distribusi menggunakan algoritma *nearest neighbour* dalam lintasan disajikan dalam Tabel 4.4 berikut.

Tabel 4.4. Visualisasi Rute *Nearest Neighbour*



Tabel 4.4. (lanjutan)



Sumber: *Google Maps*

Tabel lengkap visualisasi rute distribusi menggunakan algoritma nearest neighbour dapat dilihat pada Lampiran 6.

4.2.3 Biaya Distribusi Rute *Nearest Neighbour*

Berikut disajikan perhitungan biaya yang harus dikeluarkan perusahaan untuk menjalankan aktivitas distribusi berdasarkan rute



rekomendasi menggunakan algoritma *nearest neighbour* pada Tabel 4.4. Perhitungan biaya distribusi dilakukan berdasarkan pada besaran biaya dari aspek-aspek yang telah dijelaskan pada bab 3.1 yaitu total jarak tempuh, biaya bahan bakar, dan uang jalan. Tabel 4.4 merupakan perhitungan biaya distribusi untuk rute menggunakan algoritma *nearest neighbour*.

Tabel 4.5. Biaya Distribusi Rute Rekomendasi

Variabel	Jumlah	Biaya	Total
Jarak Tempuh	7599,32	Rp 1000,- /Km	Rp 7.599.320,-
Kendaraan Beroperasi	113	Rp 80.000,-/ kendaraan	Rp 9.040.000,-
Total Biaya Distribusi			Rp 16.639.320,-

4.3 Perbandingan Biaya Distribusi Rute Perusahaan dan Rute Rekomendasi

Biaya distribusi yang dikeluarkan perusahaan untuk pendistribusian gas LPG 3 Kg pada bulan Februari 2021 adalah sebesar Rp 17.080.000,-. Sebesar Rp 8.600.000,- dari biaya distribusi dialokasikan untuk biaya bahan bakar dan Rp 8.480.000,- dialokasikan untuk uang jalan. Dari jumlah biaya bahan bakar diperoleh jarak tempuh rute yang dilalui perusahaan adalah sebesar 8.600 Km.

Biaya distribusi yang perlu dikeluarkan perusahaan menggunakan rute rekomendasi adalah sebesar Rp 16.639.320,-. Sebesar Rp 7.599.320,- dialokasikan untuk biaya bahan bakar dan Rp 9.040.000,- dialokasikan untuk uang jalan. Total jarak tempuh yang dilalui oleh kendaraan sebesar 7.599,32 Km.

Dari kedua besaran biaya distribusi diatas, dapat dilihat bahwa menggunakan rute rekomendasi dapat meminimalkan total jarak tempuh yang perlu dilalui kendaraan dan biaya bahan bakar yang perlu dikeluarkan. Dengan menggunakan algoritma *nearest neighbour*



dapat menghemat jarak tempuh kendaraan sebesar 1000,68 Km dan biaya bahan bakar sebesar Rp 1.000.680,-.

Pada rute perusahaan, uang jalan yang perlu dikeluarkan oleh perusahaan adalah sebesar Rp 8.480.000,-. Sedangkan pada rute rekomendasi, biaya yang perlu dikeluarkan lebih besar yaitu sebesar Rp 9.040.000,-. Terjadi peningkatan biaya uang jalan sebesar Rp 560.000,-. Hal ini terjadi karena rute yang terbentuk menjadi lebih banyak daripada rute distribusi perusahaan dan menyebabkan penggunaan kendaraan menjadi lebih sering. Dengan demikian, uang jalan yang perlu dikeluarkan juga menjadi lebih besar.

Meskipun terjadi peningkatan pada biaya yang perlu dikeluarkan untuk uang jalan, algoritma ini masih dapat memberikan penghematan biaya distribusi sebesar Rp 440.680,-. Dengan demikian, penggunaan algoritma *nearest neighbour* dapat dikatakan efektif untuk meminimalkan biaya distribusi pada penelitian ini.

Perlu diperhatikan bahwa apabila perusahaan ingin menerapkan solusi ini, ada beberapa kendala yang mungkin dihadapi. Perubahan metode transportasi dari metode klasik yang selama ini digunakan perusahaan ke metode *nearest neighbour* mungkin mendapat tentangan dari para supir karena adanya faktor kebiasaan dimana biasanya supir dapat dengan bebas mengatur kunjungan dan rute perjalanan, namun dengan penerapan solusi ini para supir dituntut untuk mengikuti rute yang telah ditetapkan perusahaan. Agar dapat menerapkan solusi ini secara optimal, perusahaan harus mempunyai cara untuk memastikan bahwa para supir memang mengikuti urutan kunjungan dan rute yang telah diatur oleh perusahaan. Di sisi lain, penelitian ini belum mempertimbangkan adanya kendala dalam perjalanan seperti kemacetan parah, penutupan jalan (dikarenakan adanya perbaikan, unjuk rasa, dll.) dan kondisi yang memerlukan proses stokastik digunakan. Sehingga, membuka peluang untuk penelitian lebih lanjut.



BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan hasil yang telah diperoleh, kesimpulan yang dapat diambil dari penelitian ini adalah:

1. Diperoleh solusi permasalahan pembentukan rute pendistribusian tabung gas LPG 3 Kg yang dapat meminimalkan total jarak tempuh kendaraan dan total biaya bahan bakar menggunakan algoritma *nearest neighbour*. Meskipun terdapat sedikit peningkatan biaya pada uang jalan, penggunaan algoritma ini masih mampu memberikan total jarak dan biaya distribusi yang lebih rendah dibandingkan rute distribusi perusahaan.
2. Berdasarkan hasil yang telah diperoleh, penggunaan algoritma *nearest neighbour* pada distribusi tabung gas LPG 3 Kg bulan Februari 2021 terbukti efektif menghasilkan biaya distribusi yang lebih kecil dibandingkan biaya aktual yang dikeluarkan oleh perusahaan.

5.2 Saran

Beberapa saran yang dapat diberikan berdasarkan hasil penelitian ini adalah sebagai berikut.

1. Rute baru yang telah disusun dengan menggunakan algoritma *nearest neighbour* dapat dijadikan acuan dalam pengalokasian pangkalan pada kendaraan serta penentuan urutan kunjungan pangkalan saat melakukan aktivitas distribusi.
2. Pengalokasian pangkalan maupun tiap rute pada kendaraan sebaiknya dilakukan oleh satu orang agar pembagian kendaraan serta pemanfaatan kapasitas kendaraan dapat terkoordinasi dengan lebih efektif.



3. Sebagai pengembangan lebih lanjut dari penelitian ini, diperlukan penelitian yang lebih mendalam terkait pengembangan aplikasi *Vehicle Routing Problem* untuk menyelesaikan permasalahan yang lebih luas.



DAFTAR PUSTAKA

- Kurniawati, D., Riyanto, A., Hidayati, N., & Magfirona, A. (2017). Penentuan Rute Pendistribusian Gas Lpg dengan Metode Algoritma Nearest Neighbour. *Jurnal Transportasi*, 17(1), 59-70.
- Montgomery, D.C. (1985). *Introduction to Statistical Quality Control 6th Edition*. USA: John Wiley & Sons Inc.
- Pop, P. C., Sitar, C. P., Zelina, I., Lupse, V., & Chira, C. (2011). Heuristic Algorithms for Solving the Generalized Vehicle Routing Problem. *International Journal Computers, Communication & Control*, 11 (1), 158-165.
- Rasku, J., Kärkkäinen, T., & Musliu, N. (2019). *Meta-Survey and Implementations of Classical Capacitated Vehicle Routing Heuristics with Reproduced Results*. In J. Rasku, *Toward Automatic Customization of Vehicle Routing Systems*. *JYU Dissertations* 113, University of Jyväskylä, Finland, (pp. 133-260).
- Rupiah, S, Mulyono, & Sugiharti, E. (2017). Efektivitas Algoritma Clarke-Wright dan Sequential Insertion Dalam Penentuan Rute Pendistribusian Tabung Gas LPG. *Journal of Mathematics*, 6(2), 198-210.
- Siagian, S.P. (2008). *Manajemen Sumber Daya Manusia*. Jakarta: Bumi Aksara.
- Siang, J.J. (2011). *Riset Operasi dalam Pendekatan Algoritmis*. Yogyakarta: Andi.
- Taha, H. A. (1993). *Riset Operasi*. Jakarta: Erlangga.
- Tandjung, J.W. (2004). *Marketing Management: Pendekatan Pada Nilai-Nilai Pelanggan, Edisi Kedua*. Malang: Penerbit Bayu Media.



Toth, P. & Vigo, D. (2002). *An Overview of Vehicle Routing Problem*.
1-26

Yitnosumarto, S. (1990). *Dasar-Dasar Statistika Dengan Penekanan Terapan dalam Bidang Agrokomplks, Teknologi dan Sosial*.
Jakarta: Rajawali Press.

Yulianingrum, W. (2019). *Penentuan Rute Pendistribusian Tabung Gas LPG dan Biaya Transportasi dengan Menggunakan Algoritma Nearest Neighbour Dan Clarke & Wright Savings (Studi Kasus: PT. Harum Ossamac | Grobogan)*. Skripsi, Universitas Negeri Semarang.



LAMPIRAN

Lampiran 1. Tabel Jarak

Jarak (Km)	DEPOT	SPPBE	1	2	3	4	5	6	7	8	9
DEPOT	0.0	18.3	18.7	6.5	6.6	19.9	17.5	12.7	5.2	16.9	9.4
SPPBE	18.3	0.0	28.5	17.8	20.7	30.6	27.9	20.5	14.1	36.9	11.7
1	19.0	28.9	0.0	14.0	12.5	8.2	2.2	8.8	18.1	20.1	18.6
2	6.5	18.4	15.1	0.0	3.3	16.5	13.9	6.8	5.1	18.3	7.3
3	6.1	21.3	13.8	2.1	0.0	14.7	12.6	8.2	7.2	15.2	9.5
4	20.6	31.3	8.3	16.4	14.7	0.0	7.5	13.9	20.5	21.3	21.0
5	17.5	28.1	2.2	13.2	11.5	6.3	0.0	10.7	18.5	18.1	17.7
6	12.3	20.4	9.0	6.7	8.4	13.2	10.5	0.0	10.9	23.9	10.1
7	5.2	14.1	19.2	5.1	7.5	20.7	18.0	10.9	0.0	22.5	4.2
8	16.4	37.9	20.3	18.3	15.0	19.4	18.3	23.4	23.4	0.0	27.5
9	9.4	11.8	18.8	7.3	9.8	20.3	17.6	10.5	4.2	24.8	0.0
10	13.3	19.7	8.8	7.8	9.4	13.0	10.3	1.4	11.9	24.5	11.1
11	1.2	19.7	17.5	5.3	5.2	18.8	16.3	11.4	5.6	17.0	9.8
12	15.5	23.6	5.8	9.9	11.6	11.9	7.4	3.4	14.0	22.6	13.3
13	12.4	30.4	24.5	12.7	12.2	25.8	23.2	18.8	17.3	14.6	20.1
14	13.1	28.3	7.2	9.0	7.1	7.2	5.1	11.5	14.1	13.7	16.4
15	16.5	22.8	6.5	10.9	12.6	12.7	8.1	4.5	15.1	23.3	14.3
16	5.9	25.9	15.0	6.7	4.9	14.9	13.8	12.8	11.2	10.3	14.1
17	17.9	28.6	4.6	13.6	11.9	4.5	4.7	11.1	17.8	18.5	18.2
18	0.9	19.2	18.9	6.7	6.8	20.2	17.6	12.7	5.1	17.3	9.3
19	4.9	14.5	21.4	7.2	9.6	22.8	20.1	13.0	2.4	22.2	4.9
20	5.9	13.5	21.5	7.3	9.8	23.0	20.2	13.1	3.1	23.2	5.0
21	7.9	27.9	18.6	8.6	6.9	17.7	16.6	14.7	13.2	8.9	16.0
22	9.5	9.8	20.4	9.0	11.4	21.8	19.1	12.0	5.3	26.5	2.9
23	8.1	29.6	14.1	8.6	7.2	13.1	12.0	15.5	13.9	8.0	17.7
24	0.6	19.3	18.5	6.4	6.5	19.9	17.3	12.5	5.2	16.9	9.4
25	13.5	22.8	9.9	7.9	7.4	11.4	8.6	8.7	12.0	19.7	12.5
26	18.5	29.2	6.2	14.3	12.5	2.9	5.4	11.7	18.4	19.1	18.9
27	10.6	17.9	12.1	5.0	6.7	14.0	11.3	2.3	8.9	21.8	7.6
28	13.8	35.1	18.8	15.0	13.4	17.8	16.7	21.9	20.5	1.3	24.8
29	6.6	21.9	13.7	2.7	0.6	14.6	12.4	8.8	7.8	14.6	10.1
30	17.7	32.1	10.3	13.7	11.8	4.6	8.3	14.6	18.8	17.5	21.8
31	2.1	17.3	20.2	7.0	8.2	21.6	19.0	13.1	3.9	19.5	8.1
32	17.3	23.6	7.3	11.7	13.4	13.5	8.9	5.3	15.9	24.1	15.1
33	7.0	24.5	13.3	5.2	3.7	14.5	12.1	11.3	10.3	11.3	12.6
34	17.5	28.1	5.8	13.2	11.5	3.7	4.3	10.7	17.4	18.1	17.8
35	10.6	9.1	21.5	10.1	12.5	22.9	20.2	13.1	6.4	27.5	4.0
36	16.1	24.9	4.0	11.3	10.2	10.1	5.6	4.8	15.4	20.7	16.2
37	18.3	29.0	1.2	14.1	12.4	7.2	1.0	9.5	18.2	19.0	18.7
38	16.9	27.6	5.2	12.7	11.0	4.2	3.8	10.1	16.8	17.5	17.3
39	15.5	23.6	5.8	9.9	11.6	11.9	7.4	3.4	14.1	22.6	13.3
40	5.3	17.2	15.4	1.2	3.7	16.8	14.1	7.0	3.9	18.7	6.2
41	5.5	14.0	21.9	7.8	10.2	23.4	20.7	13.6	3.0	22.8	5.5
42	12.2	19.9	10.9	7.3	9.0	15.1	12.0	2.2	9.7	24.0	9.6
43	3.3	15.8	20.5	7.0	8.5	21.8	19.3	13.4	2.7	20.0	6.9
44	2.6	21.0	21.2	8.2	8.8	22.5	20.0	14.0	6.9	18.8	11.1
45	7.7	22.9	12.8	3.7	2.1	13.2	11.6	9.8	8.8	13.1	11.1
46	14.2	20.5	7.6	8.6	10.3	13.8	9.8	2.2	12.7	24.4	12.0



Lampiran 1. (Lanjutan)

Jarak (Km)	DEPOT	SPPBE	1	2	3	4	5	6	7	8	9
47	14.8	25.2	6.8	10.3	8.8	8.8	6.0	5.2	14.5	19.4	14.9
48	2.5	20.3	16.2	4.1	3.9	17.5	15.0	10.2	6.2	15.7	9.2
49	5.6	25.6	15.3	6.4	4.6	15.3	14.1	12.5	11.0	10.8	13.8
50	12.0	20.1	9.3	6.4	8.1	13.6	10.9	0.6	10.5	23.1	9.8
51	16.2	24.9	4.0	11.3	10.2	10.1	5.6	4.8	15.4	20.8	14.7
52	2.3	19.0	17.8	5.4	5.5	19.0	16.6	11.6	4.9	17.3	9.1
53	11.1	18.8	19.1	6.0	7.0	20.3	17.8	12.2	4.7	18.5	8.9
54	2.8	21.2	21.5	8.4	9.0	22.7	20.2	14.2	7.1	19.0	11.3
55	18.7	29.4	3.8	14.5	12.8	4.0	4.4	11.9	18.6	19.3	19.1
56	9.9	9.5	19.8	9.4	11.9	21.3	18.6	11.5	5.7	27.2	2.4
57	5.4	17.3	15.2	1.0	3.5	16.7	13.9	6.8	4.0	18.5	6.3
58	17.0	32.2	13.4	13.0	11.1	14.4	11.4	17.7	18.1	16.3	21.9
59	13.6	20.7	7.5	8.0	9.7	11.7	9.0	1.6	12.2	22.6	11.4
60	18.8	29.5	7.7	14.6	12.8	1.9	5.7	12.0	18.7	19.4	19.1
61	5.0	13.4	20.9	6.8	9.2	22.4	19.7	12.6	2.3	22.2	4.5
62	12.2	19.9	10.9	7.3	9.0	15.1	12.0	2.2	9.7	24.0	9.6
63	3.3	15.8	20.5	7.0	8.5	21.8	19.3	13.4	2.7	20.0	6.9
64	2.6	21.0	21.2	8.2	8.8	22.5	20.0	14.0	6.9	18.8	11.1
65	7.7	22.9	12.8	3.7	2.1	13.2	11.6	9.8	8.8	13.1	11.1
66	14.2	20.5	7.6	8.6	10.3	13.8	9.8	2.2	12.7	24.4	12.0
67	14.8	25.2	6.8	10.3	8.8	8.8	6.0	5.2	14.5	19.4	14.9
68	2.5	20.3	16.2	4.1	3.9	17.5	15.0	10.2	6.2	15.7	9.2

Jarak (Km)	10	11	12	13	14	15	16	17	18	19	20	21
DEPOT	13.7	11.2	15.8	12.8	12.8	15.3	6.3	17.8	0.9	4.9	5.9	8.3
SPPBE	19.7	18.7	22.5	32.2	26.8	21.4	25.4	28.5	18.2	14.5	13.5	27.8
1	8.6	17.2	6.2	24.4	8.8	8.1	15.0	4.6	18.6	20.3	20.4	18.3
2	7.9	5.6	9.9	13.6	9.4	9.5	7.1	14.4	6.4	7.2	7.3	9.2
3	9.3	4.9	11.3	12.1	7.9	10.9	4.6	12.9	6.3	8.9	9.4	7.7
4	15.5	19.4	12.4	26.6	9.9	14.3	16.9	5.3	20.8	22.7	22.8	19.6
5	10.4	16.2	7.6	23.6	6.7	9.8	13.7	4.2	17.6	20.3	20.8	16.4
6	1.5	11.4	3.2	19.2	13.6	3.5	12.7	11.1	12.2	12.7	12.8	14.8
7	12.0	5.6	14.0	17.8	13.6	13.6	11.3	18.6	5.1	2.4	3.1	13.4
8	25.1	16.3	21.9	14.8	13.2	23.9	10.3	18.9	16.8	23.2	24.2	7.9
9	11.7	9.8	13.7	20.1	15.9	13.3	13.6	18.2	9.4	4.9	5.0	15.7
10	0.0	13.0	3.0	19.8	13.4	2.4	13.3	10.9	13.3	13.7	13.8	15.4
11	12.5	0.0	14.6	12.9	11.5	14.2	6.2	16.6	1.3	5.3	6.3	8.2
12	2.8	15.3	0.0	22.0	12.3	3.6	14.5	9.8	15.4	15.9	16.0	17.5
13	19.9	11.8	21.9	0.0	18.5	21.2	11.3	23.6	12.3	17.0	17.5	8.3
14	11.2	11.8	10.0	19.1	0.0	11.9	9.3	5.0	13.2	15.9	16.4	12.0
15	3.3	16.3	3.3	23.0	13.1	0.0	15.2	10.6	16.4	16.9	17.0	18.6
16	13.9	5.8	13.9	10.9	8.7	15.8	0.0	13.8	6.3	11.0	12.0	3.7
17	10.9	16.6	9.6	23.9	7.1	11.5	14.2	0.0	18.0	20.0	20.0	16.8
18	13.3	1.4	16.0	13.2	12.9	14.9	6.7	18.0	0.0	4.8	5.8	8.7
19	14.2	5.3	16.2	17.5	15.8	15.8	11.0	20.7	4.8	0.0	1.1	13.1
20	14.3	6.3	16.3	18.5	15.9	15.9	12.0	20.8	5.8	1.1	0.0	14.1
21	15.9	7.8	17.9	8.5	11.5	17.5	4.5	16.5	8.3	12.9	13.9	0.0
22	13.2	9.9	15.2	22.1	17.6	14.8	16.6	19.7	9.4	5.0	4.7	17.4
23	15.3	7.9	14.0	13.0	6.9	16.0	2.3	12.0	8.5	13.1	14.1	5.8
24	13.4	11.0	15.6	12.8	12.6	15.0	6.3	17.6	10.4	4.9	5.9	8.4



Lampiran 1. (Lanjutan)

Jarak (Km)	10	11	12	13	14	15	16	17	18	19	20	21
25	8.5	12.2	8.0	20.0	9.5	10.4	11.6	9.2	13.4	14.2	14.3	14.6
26	11.5	17.3	10.2	24.5	7.8	12.2	14.8	1.7	18.7	20.6	20.7	17.4
27	3.4	10.3	5.5	17.1	12.4	5.0	14.3	11.9	10.6	10.2	10.3	12.7
28	23.1	13.7	20.4	11.8	11.6	22.3	8.7	16.7	14.2	20.0	21.0	6.4
29	9.9	5.4	11.9	12.1	7.6	11.5	4.1	12.8	6.8	9.6	10.1	6.4
30	14.4	16.5	13.1	23.7	7.0	15.1	13.1	6.5	17.9	20.6	21.1	15.8
31	14.3	3.1	16.3	14.9	14.3	15.9	8.4	19.3	2.0	3.1	3.9	10.4
32	4.1	17.1	4.1	23.8	13.9	0.8	16.0	11.4	17.2	17.7	17.8	19.3
33	12.5	6.6	12.2	11.9	7.4	14.1	1.8	12.4	7.4	11.2	12.2	5.1
34	10.5	16.2	9.2	23.5	6.7	11.1	13.7	1.5	17.6	19.5	19.6	16.4
35	14.3	11.0	16.3	22.9	18.6	15.9	17.7	20.8	10.5	6.1	5.8	18.4
36	4.5	14.9	2.2	22.1	10.5	4.1	12.6	7.7	16.3	18.0	17.7	16.0
37	9.3	17.1	6.6	24.3	7.6	8.6	14.6	4.4	18.5	20.4	20.5	17.3
38	9.9	15.7	8.6	22.9	6.2	10.6	13.2	1.0	17.1	19.0	19.1	15.8
39	2.8	15.3	0.0	22.0	12.3	3.6	14.5	9.8	15.4	15.9	16.0	17.5
40	8.2	4.8	10.2	14.0	9.8	9.8	8.4	14.7	5.2	6.1	6.2	9.6
41	14.7	5.9	16.7	18.1	16.3	16.3	11.6	21.3	5.4	0.7	0.6	13.7
42	2.2	12.0	4.7	19.4	14.7	3.8	16.5	13.0	12.2	11.9	12.0	14.9
43	14.6	3.7	16.6	15.9	14.6	16.2	9.4	19.6	3.3	2.4	3.4	11.5
44	15.1	3.3	18.3	14.7	15.3	16.7	8.2	20.4	2.7	6.6	7.6	10.3
45	10.9	6.4	11.7	12.4	6.9	12.5	3.3	11.9	7.8	10.6	11.1	5.6
46	1.0	14.0	2.5	20.7	14.2	1.1	16.3	11.7	14.1	14.6	14.7	16.2
47	4.9	13.6	3.1	20.8	9.1	5.1	11.3	6.6	15.0	16.6	16.7	14.7
48	11.3	1.2	13.3	11.6	10.3	12.9	5.1	15.3	2.6	5.9	6.9	7.2
49	13.6	5.5	14.2	10.3	9.1	16.2	0.7	14.1	6.0	10.7	11.7	3.1
50	1.9	11.0	3.5	18.4	13.8	3.8	15.6	11.5	11.9	12.3	12.4	14.0
51	4.6	14.9	2.2	22.2	10.5	4.2	12.7	8.0	16.3	18.0	17.7	16.0
52	12.7	1.0	14.9	13.2	11.9	13.9	6.7	16.9	2.3	4.6	5.6	8.8
53	13.4	2.1	16.1	13.8	13.1	14.5	7.3	18.2	1.0	4.4	5.4	9.4
54	15.3	3.5	18.5	14.9	15.5	16.9	8.4	20.6	2.9	6.8	7.8	10.5
55	11.7	17.5	8.7	24.7	8.0	12.4	15.0	1.2	18.9	20.8	20.9	17.6
56	12.6	10.3	14.6	22.3	18.0	14.2	17.0	19.2	9.8	5.4	5.1	17.8
57	8.0	4.7	10.0	13.8	9.6	9.6	8.2	14.5	5.3	6.2	6.3	9.4
58	19.4	15.8	16.2	22.5	6.3	18.2	11.9	11.3	17.2	20.4	20.4	14.6
59	1.3	13.3	1.7	20.1	12.1	3.3	14.5	9.6	13.6	14.0	14.1	15.7
60	11.8	17.6	10.5	24.8	8.0	12.5	15.1	2.9	19.0	20.9	21.0	17.7
61	13.7	5.4	15.7	17.6	15.3	15.3	11.1	20.3	4.9	0.5	0.7	13.1

Jarak (Km)	22	23	24	25	26	27	28	29	30	31	32	33
DEPOT	9.5	8.4	0.6	12.8	18.4	10.9	14.5	6.3	18.0	2.1	15.7	7.4
SPPBE	9.8	28.8	18.3	22.6	29.1	19.1	34.4	20.8	31.5	17.3	21.8	25.1
1	20.1	13.9	18.3	10.3	6.2	10.8	18.6	13.1	10.0	19.8	8.5	13.3
2	9.0	9.3	6.5	8.5	15.0	5.0	15.4	3.5	14.6	6.7	9.9	6.2
3	11.7	7.2	5.9	7.9	13.4	6.5	13.7	0.6	13.0	7.1	11.3	3.7
4	22.5	15.0	20.5	12.8	3.4	15.9	19.7	15.3	5.3	21.6	14.7	15.5
5	19.2	11.8	17.3	9.6	4.2	13.2	16.5	12.1	7.9	18.4	10.2	12.3
6	11.6	16.0	12.3	7.7	11.7	2.3	22.4	8.6	14.8	12.5	3.9	11.4
7	5.3	13.4	5.2	11.4	19.1	9.1	19.6	7.7	18.8	3.9	14.0	10.4
8	27.8	8.0	16.5	19.3	18.8	23.0	1.3	14.5	16.9	20.0	24.3	11.2
9	2.9	15.8	9.4	12.3	18.8	8.8	21.9	10.0	21.1	8.1	13.7	12.7

Lampiran 1. (Lanjutan)

Jarak (Km)	22	23	24	25	26	27	28	29	30	31	32	33
10	12.7	16.0	13.3	8.7	11.5	3.3	21.6	9.7	14.6	13.6	2.8	12.4
11	9.9	8.3	1.0	11.5	17.1	9.7	14.6	5.1	16.3	3.1	14.6	6.1
12	14.8	14.6	16.3	9.9	10.4	5.5	21.0	11.8	13.5	15.7	4.0	12.8
13	21.6	13.4	12.5	20.8	24.1	17.3	11.7	12.3	23.3	15.1	22.2	12.4
14	17.9	7.7	12.9	8.2	3.6	11.8	12.1	7.7	15.2	14.0	12.3	7.9
15	15.8	15.3	18.5	10.6	11.1	6.5	21.8	13.4	14.3	16.7	1.0	13.5
16	15.6	2.3	6.0	11.3	14.3	11.1	8.7	4.8	12.5	9.1	16.2	1.8
17	19.7	12.5	17.7	10.0	1.7	13.1	16.9	12.5	6.0	18.8	11.9	12.7
18	9.4	8.8	0.4	12.9	18.5	10.9	14.9	6.5	17.7	2.0	15.3	7.5
19	5.0	13.1	4.9	13.6	21.3	11.3	19.3	9.7	21.0	3.1	16.2	10.8
20	4.7	14.1	5.9	13.7	21.4	11.4	20.3	10.0	21.1	3.9	16.3	11.8
21	17.5	6.2	8.0	15.2	17.0	13.0	6.4	6.8	15.2	11.0	17.9	5.7
22	0.0	17.4	9.5	13.8	20.3	9.1	23.6	11.7	22.8	8.2	15.2	14.4
23	17.7	0.0	8.1	11.4	12.5	13.5	6.4	6.6	10.7	10.2	16.4	3.3
24	9.5	8.5	0.0	12.6	18.2	10.3	14.6	6.2	17.8	2.1	15.6	7.1
25	14.0	11.7	13.5	0.0	9.8	7.9	18.1	9.7	12.9	13.7	10.8	8.6
26	20.4	13.1	18.3	10.6	0.0	13.8	17.6	13.2	4.5	19.5	12.6	13.3
27	9.1	12.7	10.3	6.0	12.5	0.0	18.8	7.0	15.6	10.8	5.4	9.7
28	24.8	6.4	13.9	17.7	17.2	20.2	0.0	12.7	15.4	18.0	22.7	9.6
29	12.3	6.3	6.4	7.7	13.3	7.1	12.6	0.0	12.1	7.7	11.9	3.1
30	23.3	11.5	17.6	13.5	4.8	17.2	15.9	12.4	0.0	19.3	15.5	11.8
31	8.2	10.5	2.1	14.3	19.9	11.0	16.6	7.8	19.1	0.0	15.9	8.9
32	16.6	16.1	17.3	11.4	11.9	7.3	22.6	13.6	15.1	17.5	0.0	14.3
33	14.8	3.3	7.1	8.8	13.0	9.6	9.7	3.2	11.2	9.4	14.5	0.0
34	19.3	12.1	17.3	9.6	2.1	12.7	16.5	12.1	5.3	18.4	11.5	12.3
35	2.1	18.5	10.6	14.9	21.4	11.4	24.9	12.7	23.8	9.3	16.3	15.4
36	16.2	12.8	16.0	8.0	8.5	6.8	19.2	10.8	11.7	17.1	4.5	10.9
37	20.2	13.0	18.2	10.5	5.6	11.5	17.4	13.0	8.8	19.3	9.2	13.2
38	18.8	11.5	16.7	9.0	2.6	12.2	16.0	11.6	5.8	17.9	11.0	11.7
39	14.8	14.6	15.5	9.9	10.3	5.5	21.0	11.8	13.5	15.7	4.0	12.8
40	8.4	9.6	5.3	8.8	15.3	5.3	15.8	3.9	15.0	5.6	10.2	6.5
41	5.2	13.7	5.5	15.3	21.8	11.9	19.9	10.3	21.5	3.5	16.7	11.4
42	11.1	15.0	12.2	8.2	13.1	2.3	21.1	9.2	17.9	12.5	4.2	11.9
43	7.0	11.6	3.3	15.2	20.2	11.7	17.7	8.2	19.4	1.2	16.6	9.2
44	11.2	10.3	2.7	15.3	20.9	12.9	16.5	8.9	20.0	3.8	17.1	9.3
45	13.3	5.1	7.5	7.9	12.5	8.9	11.5	1.5	10.7	8.7	12.9	1.8
46	13.5	16.3	14.2	9.6	12.2	4.2	22.4	10.5	15.3	14.4	1.5	13.3
47	16.4	11.4	14.6	6.7	7.2	7.2	17.8	9.4	10.3	15.7	5.5	9.6
48	10.5	7.3	2.3	10.3	15.9	8.4	13.4	3.9	15.1	4.0	13.3	4.9
49	15.3	2.8	5.7	11.6	14.6	10.8	9.2	4.5	12.8	8.8	15.6	2.1
50	11.3	14.0	12.4	7.3	12.0	1.9	20.2	8.3	15.2	12.2	4.2	11.0
51	16.2	12.8	16.0	8.1	8.6	6.8	19.2	10.8	11.7	17.1	4.6	11.0
52	9.2	8.9	2.1	11.9	17.5	9.7	15.0	5.5	16.7	2.7	14.6	6.4
53	9.0	9.4	1.0	13.1	18.7	10.3	15.6	6.7	17.9	1.1	14.9	7.7
54	11.4	10.5	3.0	15.5	21.1	12.7	16.7	9.1	20.2	4.0	17.3	9.5
55	20.6	13.3	18.5	10.8	2.5	14.0	17.8	13.4	6.8	19.7	12.8	13.5
56	0.6	17.9	9.9	13.2	19.7	8.5	24.3	12.1	23.0	8.6	14.6	14.8
57	8.5	9.5	5.4	8.6	15.1	5.1	15.6	3.7	14.8	5.6	10.0	6.3
58	23.4	10.3	16.8	14.0	11.6	17.4	14.7	11.7	7.6	18.0	18.6	10.6
59	12.9	14.7	14.1	8.2	10.1	3.6	20.8	10.0	13.3	13.8	3.7	12.5
60	20.6	13.4	18.6	10.9	1.0	14.1	17.5	13.4	3.5	19.8	12.9	13.6



Lampiran 1. (Lanjutan)

Jarak (Km)	22	23	24	25	26	27	28	29	30	31	32	33
61	4.6	13.2	4.9	14.3	20.8	10.9	19.3	9.5	20.5	3.6	15.7	10.8
Jarak (Km)	34	35	36	37	38	39	40	41	42	43	44	45
DEPOT	16.8	10.6	17.3	18.4	16.9	15.8	5.2	5.5	12.5	3.3	2.6	6.7
SPPBE	27.5	9.1	24.4	28.8	27.6	22.5	16.6	13.9	20.6	15.8	20.0	24.5
1	5.3	21.2	4.0	1.2	5.2	6.2	14.3	20.8	11.5	20.1	21.0	12.8
2	13.5	10.1	11.5	14.7	13.5	9.9	1.2	7.8	6.6	7.0	8.2	4.6
3	11.9	12.7	12.3	13.4	11.9	11.3	3.4	9.9	8.1	7.4	8.6	2.1
4	4.9	23.6	12.0	8.0	6.5	12.3	16.7	23.3	16.5	21.9	23.2	14.9
5	2.1	20.3	5.6	1.0	1.6	7.6	14.7	21.2	13.3	18.7	20.0	11.7
6	10.1	12.7	5.0	9.7	10.2	3.2	7.0	13.2	2.9	12.8	14.0	9.8
7	17.6	6.4	16.7	18.8	17.6	14.0	3.9	3.0	10.7	2.7	6.9	8.8
8	17.9	30.1	21.6	19.1	17.9	21.9	18.9	23.8	24.6	20.3	18.4	13.0
9	17.3	4.0	16.3	18.5	17.3	13.7	6.2	5.4	10.4	6.9	11.1	11.1
10	10.0	13.8	4.8	9.5	10.0	3.0	8.1	14.3	3.4	13.9	15.0	10.8
11	15.6	11.0	16.0	17.1	15.6	14.6	4.6	5.9	11.3	3.7	3.3	5.5
12	8.9	15.9	2.3	6.4	8.9	0.0	10.2	16.4	6.1	16.0	17.2	12.2
13	22.6	23.3	23.0	24.1	22.6	21.9	14.2	17.6	18.9	15.4	14.3	12.4
14	4.1	19.0	9.6	6.0	4.1	10.0	10.3	16.8	13.4	14.3	15.6	7.3
15	9.6	16.9	3.1	7.2	9.6	3.3	11.2	17.4	6.5	17.0	18.2	13.0
16	12.8	16.7	13.5	14.6	12.8	13.9	8.0	11.6	12.7	9.5	7.9	4.8
17	1.5	20.8	7.7	4.4	1.0	9.6	14.0	20.5	13.8	19.1	20.4	12.1
18	17.0	10.5	17.4	18.5	17.0	16.0	5.2	5.4	12.5	3.2	2.7	6.9
19	19.8	6.1	18.8	21.0	19.8	16.2	6.1	0.7	12.9	2.4	6.6	10.3
20	19.9	5.8	19.0	21.2	19.9	16.3	6.2	0.6	13.0	3.4	7.6	11.1
21	15.5	18.6	17.5	17.4	15.5	17.8	9.9	13.5	14.6	11.4	9.8	6.7
22	18.8	2.1	16.8	20.0	18.8	15.2	7.9	5.2	11.9	7.0	11.2	12.8
23	11.0	21.0	13.7	12.9	11.0	14.0	10.4	13.7	15.1	11.5	10.0	5.1
24	16.7	10.6	17.1	18.2	16.7	15.6	5.2	5.5	12.3	3.3	2.8	6.5
25	8.3	15.1	8.4	9.5	8.3	8.8	8.2	14.8	9.5	14.0	15.2	7.2
26	1.7	21.5	9.2	5.9	3.8	10.2	14.6	21.1	14.4	19.8	21.2	12.8
27	11.0	10.2	7.0	12.2	11.0	5.5	5.4	10.8	2.2	11.2	12.0	8.1
28	15.7	25.8	20.0	17.6	15.7	20.4	17.2	19.4	21.8	17.3	15.8	11.4
29	11.8	13.3	12.2	13.3	11.8	11.9	4.0	10.2	8.7	8.0	9.1	1.5
30	5.6	24.4	12.8	9.2	7.3	13.1	15.0	21.2	17.3	19.0	20.3	11.3
31	18.4	9.2	17.9	19.9	18.4	16.3	5.5	3.5	12.4	1.2	3.8	8.3
32	10.4	17.7	3.9	8.6	10.4	4.1	12.0	18.0	8.4	17.8	19.0	13.8
33	11.5	15.9	11.9	13.0	11.5	12.2	6.5	11.8	11.2	9.7	9.0	1.7
34	0.0	20.4	8.8	5.2	0.6	9.2	13.5	20.1	13.4	18.7	20.0	11.7
35	19.9	0.0	17.9	21.1	19.9	16.3	8.9	6.2	13.0	8.1	12.3	14.1
36	7.0	17.3	0.0	4.7	7.0	2.2	11.6	18.1	7.4	17.4	18.5	10.4
37	3.0	21.3	4.7	0.0	2.4	6.6	14.4	21.0	12.2	19.6	20.9	12.6
38	0.6	19.9	8.3	4.6	0.0	8.6	13.0	19.5	12.8	18.2	19.5	11.2
39	8.8	15.9	2.3	7.0	8.9	0.0	10.2	16.2	6.1	16.0	17.2	12.2
40	13.8	9.5	11.8	15.0	13.8	10.2	0.0	6.6	6.9	5.7	7.0	5.0
41	20.3	6.2	18.3	21.6	20.3	16.7	6.6	0.0	13.5	3.0	7.2	10.8
42	11.6	12.2	6.5	11.2	11.6	4.9	7.5	12.5	0.0	14.1	13.9	12.5
43	18.7	8.1	18.2	20.2	18.7	16.6	5.8	2.8	13.3	0.0	5.0	8.6
44	19.4	12.3	19.8	20.9	19.4	17.1	6.9	7.2	14.5	5.0	0.0	9.3
45	11.0	14.3	11.4	12.5	11.0	11.7	5.8	12.4	10.5	9.8	10.1	0.0

Lampiran 1. (Lanjutan)

Jarak (Km)	34	35	36	37	38	39	40	41	42	43	44	45
46	10.7	14.6	4.2	8.9	10.7	2.5	8.9	14.6	4.4	15.6	16.8	14.0
47	5.7	17.5	2.8	6.9	5.7	3.1	10.6	17.2	7.8	16.1	17.3	9.0
48	14.4	11.5	14.8	15.9	14.4	13.3	3.6	6.5	10.1	4.3	4.6	4.3
49	13.1	16.3	13.9	15.0	13.1	14.2	7.7	11.7	12.6	9.4	7.6	4.7
50	10.5	12.4	5.3	10.0	10.5	3.7	6.7	12.6	2.6	13.6	14.6	11.6
51	7.1	17.3	0.0	4.7	7.1	2.2	11.6	18.6	7.5	17.4	18.7	10.4
52	16.0	10.3	16.4	17.5	16.0	14.3	4.1	5.2	11.5	3.0	4.0	5.8
53	17.2	10.1	17.6	18.7	17.2	16.2	4.7	5.0	12.1	2.8	2.7	7.1
54	19.6	12.5	20.0	21.1	19.6	17.3	7.1	7.4	14.7	5.2	0.2	9.5
55	2.3	21.7	6.8	3.5	1.8	9.0	14.8	21.3	14.6	20.0	21.3	13.0
56	18.2	1.8	16.2	19.4	18.2	14.6	8.3	5.6	11.3	7.4	11.6	15.6
57	13.6	9.0	11.6	14.8	13.6	10.0	0.2	6.7	6.7	5.8	7.1	6.2
58	10.3	24.5	15.9	12.2	10.3	16.2	14.3	20.5	19.0	18.3	19.1	10.1
59	8.6	14.0	3.4	8.2	8.6	1.7	8.4	14.6	4.2	15.4	16.1	12.0
60	3.0	21.7	10.2	6.5	4.6	10.5	14.9	21.4	14.7	20.1	21.3	13.0
61	19.3	5.6	17.3	20.6	19.3	15.7	5.6	1.0	12.5	2.5	6.7	10.2

Jarak (Km)	46	47	48	49	50	51	52	53	54	55	56	57
DEPOT	14.6	15.5	2.5	5.8	12.2	17.3	2.2	1.1	2.8	18.7	9.9	5.4
SPPBE	20.7	25.3	19.2	25.0	20.4	24.5	18.0	17.8	20.2	29.4	9.5	16.7
1	8.4	5.4	16.1	15.3	9.1	4.0	17.6	18.8	21.2	3.8	19.5	14.1
2	8.8	11.9	4.5	6.7	6.4	12.6	5.4	6.0	8.4	15.3	9.5	1.0
3	10.3	10.2	3.8	4.1	7.8	12.4	5.2	6.4	8.8	13.7	11.6	3.2
4	14.6	10.3	18.3	17.3	14.2	12.0	19.7	21.0	23.4	4.0	21.9	16.5
5	9.8	7.1	15.1	14.1	11.0	5.6	16.6	17.8	20.2	4.4	18.7	14.5
6	2.8	5.6	11.0	12.3	0.6	5.0	11.2	11.8	14.2	12.0	11.0	6.8
7	12.9	16.0	6.2	10.9	10.5	16.7	4.9	4.7	7.1	19.4	5.7	4.0
8	24.2	19.9	16.1	10.8	23.7	21.6	17.6	17.5	18.6	19.7	28.4	18.7
9	12.6	15.7	9.2	13.2	10.2	16.4	9.1	8.9	11.3	19.1	2.4	6.3
10	1.7	5.3	12.0	12.9	1.8	4.8	12.2	12.8	15.2	11.8	12.1	7.9
11	13.5	14.3	1.2	5.7	11.0	16.1	1.0	2.1	3.5	17.4	10.3	4.5
12	2.9	3.3	14.1	14.8	3.7	2.4	15.6	15.0	17.4	9.2	14.2	10.0
13	21.1	21.3	11.2	10.9	18.6	23.1	12.8	13.5	14.5	24.4	22.0	14.0
14	12.2	7.9	10.7	9.7	11.8	9.6	12.2	13.3	15.8	5.9	17.3	10.1
15	2.4	4.0	16.4	15.6	5.0	3.1	17.9	16.0	18.4	11.4	15.3	11.1
16	16.1	11.8	5.2	0.7	12.4	13.6	6.9	7.0	8.1	14.6	16.6	7.8
17	11.8	7.5	15.5	14.5	11.4	7.6	17.0	18.2	20.6	1.2	19.1	13.8
18	14.3	15.7	2.6	6.2	12.3	17.5	2.3	0.9	2.9	18.8	9.8	5.3
19	15.1	18.2	5.9	10.6	12.7	18.9	4.6	4.4	6.8	21.6	5.4	6.2
20	15.2	18.3	6.9	11.6	12.8	19.0	5.6	5.4	7.8	21.7	5.1	6.3
21	16.8	15.8	7.1	4.3	14.4	17.5	8.6	8.9	10.0	17.3	17.9	9.7
22	14.1	17.2	10.5	14.9	11.7	16.8	9.2	9.0	11.4	20.6	0.6	8.0
23	16.2	11.9	7.3	2.8	15.8	13.7	8.8	9.1	10.2	12.8	19.9	10.2
24	14.5	15.4	2.3	5.9	12.1	17.1	2.1	1.0	3.0	18.5	9.9	5.4
25	9.7	6.7	12.2	11.9	9.3	8.5	12.4	13.0	15.4	10.1	13.4	8.0
26	12.5	8.2	16.2	15.1	12.0	9.2	17.6	18.9	21.3	2.5	19.8	14.4
27	4.4	7.7	9.1	10.2	1.9	7.1	9.2	9.8	12.2	12.8	8.5	5.2
28	22.6	18.3	13.0	9.2	21.6	20.1	14.6	14.9	16.0	17.5	25.7	17.8
29	10.9	10.5	4.3	3.7	8.4	12.3	5.7	7.0	9.3	13.6	12.2	3.8
30	15.4	11.1	15.4	13.5	14.9	12.8	16.9	18.1	20.5	7.0	22.7	14.8



Lampiran 1. (Lanjutan)

Jarak (Km)	46	47	48	49	50	51	52	53	54	55	56	57
31	14.8	17.0	4.0	7.9	12.1	17.9	2.7	1.1	4.0	20.2	8.6	5.6
32	3.2	4.8	16.0	16.4	5.8	3.9	16.2	16.8	19.2	12.2	16.1	11.9
33	13.4	10.1	5.4	2.1	11.0	11.9	7.0	8.1	9.2	13.3	14.7	6.3
34	11.4	7.1	15.1	14.1	11.0	8.8	16.6	17.8	20.2	2.3	18.7	13.4
35	15.2	18.3	11.5	15.9	12.8	17.9	10.3	10.1	12.5	21.7	1.8	9.0
36	4.4	1.4	13.8	13.0	5.1	0.0	15.2	16.3	18.7	6.8	15.6	11.4
37	9.1	6.1	16.0	14.9	9.8	4.7	17.4	18.7	21.1	3.5	19.6	14.2
38	10.8	6.5	14.6	13.5	10.4	8.3	16.0	17.3	19.7	1.8	18.2	12.8
39	2.9	3.2	14.2	14.8	3.6	2.4	14.4	15.0	17.4	10.7	14.2	10.0
40	9.1	12.2	3.7	7.0	6.7	11.8	4.1	4.8	7.2	15.6	8.3	0.2
41	15.7	18.7	6.5	11.2	13.2	19.4	5.2	5.0	7.4	22.1	5.6	6.7
42	3.1	7.1	11.7	14.5	2.6	6.5	12.6	11.8	14.1	13.5	10.5	8.3
43	15.5	17.4	4.3	9.0	13.1	19.1	3.0	2.8	5.2	20.5	7.4	5.8
44	16.0	18.1	4.6	7.8	14.2	19.8	4.0	2.7	0.2	21.2	11.6	7.1
45	14.0	9.7	5.6	2.9	10.3	11.4	7.2	8.3	10.5	12.8	14.4	5.6
46	0.0	5.1	13.0	15.8	2.7	4.2	13.8	14.5	16.8	12.5	13.5	8.9
47	5.3	0.0	12.4	11.6	5.5	2.8	13.9	15.1	17.5	7.5	15.8	10.3
48	12.2	13.1	0.0	4.7	9.8	14.8	1.6	2.8	4.8	16.2	10.9	3.5
49	16.4	12.1	5.3	0.0	12.1	13.9	6.4	6.7	7.8	14.9	15.7	7.5
50	3.1	5.9	10.8	13.8	0.0	5.4	12.0	12.6	14.8	12.3	10.7	7.6
51	4.4	1.5	13.8	13.0	5.1	0.0	15.3	16.5	18.9	6.8	17.2	11.8
52	13.8	14.7	1.6	6.3	11.2	16.4	0.0	1.9	4.2	17.8	9.6	4.3
53	14.5	15.9	2.8	6.9	11.9	17.6	1.8	0.0	2.9	19.0	9.4	4.9
54	16.2	18.3	4.8	8.0	14.4	20.0	4.2	2.9	0.0	21.4	11.8	7.3
55	11.2	8.2	16.4	15.3	12.2	6.8	17.8	19.1	21.5	0.0	20.0	14.6
56	12.9	16.6	10.9	15.5	11.1	16.2	9.6	9.4	11.8	20.0	0.0	8.4
57	9.6	10.8	3.6	7.8	6.5	12.7	4.3	4.9	7.3	15.4	8.4	0.0
58	18.8	14.1	14.7	12.3	18.0	15.9	16.1	17.3	19.3	12.2	22.5	14.1
59	2.3	4.0	12.9	14.7	1.9	3.5	13.8	14.4	16.8	10.3	12.4	9.0
60	12.6	8.4	16.5	15.4	12.3	10.2	17.9	19.2	21.5	3.5	20.1	14.7
61	15.3	17.7	5.9	10.6	12.2	18.4	4.6	4.5	6.9	21.1	5.0	5.7

Jarak (Km)	58	59	60	61	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34																																																																	
DEPOT	47.6	13.9	18.7	5.0	12.2	13.8	14.7	11.3	17.8	14.1	18.9	4.9	20.7	14.4	21.8	0.7	14.9	16.0	17.4	13.0	22.4	13.3	20.7	4.6	17.4	13.8	18.6	4.9	14.6	7.2	10.2	13.8	25	7.9	10.2	10.0	20.2	17.3	3.6	12.9	9.7	15.1	20.3	16.9	18.8	29	12.6	10.1	13.7	9.5	30	5.4	13.1	3.7	20.5	31	19.1	14.1	20.3	3.6	32	18.5	5.5	12.3	17.0	33	11.0	12.2	13.4	11.3	34	8.7	9.1	2.5	19.1													
SPPBE	31.7	20.6	29.5	13.3	17	9.5	9.5	3.3	19	21.0	14.3	21.7	0.5	20	20.7	14.4	21.8	0.7	21	14.9	16.0	17.4	13.0	22	22.4	13.3	20.7	4.6	23	10.4	14.0	12.9	13.2	24	17.4	13.8	18.6	4.9	25	14.6	7.2	10.2	13.8	26	14.6	7.2	10.2	13.8	27	7.9	10.2	10.0	20.2	28	17.3	3.6	12.9	9.7	29	15.1	20.3	16.9	18.8	30	12.6	10.1	13.7	9.5	31	5.4	13.1	3.7	20.5	32	19.1	14.1	20.3	3.6	33	18.5	5.5	12.3	17.0	34	11.0	12.2	13.4	11.3



Lampiran 1. (Lanjutan)

Jarak (Km)	58	59	60	61
35	23,5	14,4	21,8	5,6
36	15,1	3,2	8,9	17,1
37	12,2	7,9	6,0	20,0
38	9,2	8,6	3,0	18,6
39	16,9	1,7	10,7	15,2
40	14,6	8,3	15,7	5,6
41	21,2	14,9	22,2	11,0
42	19,6	3,0	13,5	11,5
43	19,1	14,7	20,6	2,5
44	19,7	15,2	21,3	6,7
45	10,4	11,7	12,9	11,5
46	18,4	2,4	12,7	14,7
47	13,7	3,6	7,6	16,2
48	14,8	11,5	16,3	5,9
49	12,5	13,8	15,0	10,7
50	18,6	1,9	12,4	13,3
51	15,1	3,2	9,0	17,6
52	16,2	13,1	17,9	4,7
53	17,5	14,3	19,1	4,5
54	19,9	15,5	21,5	6,9
55	10,4	10,0	3,4	20,4
56	22,8	12,8	20,1	5,0
57	14,4	8,1	15,5	5,7
58	0,0	16,1	10,4	19,8
59	16,8	0,0	10,5	14,8
60	6,9	10,4	0,0	20,4
61	20,2	13,9	21,2	0,0

Lampiran 2. Data Permintaan

No	Pangkalan	Jumlah Permintaan/tarunggal																												
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
1	Abdul Gapur	120	0	0	120	0	0	0	120	0	0	120	0	0	0	120	0	0	120	0	0	120	0	0	120	0	0	120	0	0
2	Adil Jaya/Ahmad Z	100	80	0	100	0	0	100	0	80	0	0	0	0	100	0	80	0	100	0	100	0	0	0	80	0	100	0	0	
3	Afrizal	0	280	0	0	0	0	0	280	0	0	0	0	0	160	280	0	160	280	0	0	0	0	0	0	0	280	0	0	
4	An-Nisa	160	0	0	160	0	0	0	160	0	0	0	0	0	160	0	0	160	0	0	0	160	0	0	0	160	0	0	0	
5	Arizon/Manti	120	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	120	0	0	0	120	0	0	0	120	0	0	0	
6	Bahari/Edy	0	140	0	0	140	0	0	0	0	140	0	0	0	0	140	0	0	140	0	0	0	0	0	140	0	0	140	0	
7	Budi Ari-Setiawati	0	100	0	0	100	0	0	0	0	100	0	0	0	0	100	0	0	100	0	0	0	0	0	100	0	0	100	0	
8	Daniel Syahputra	0	110	0	0	110	0	110	0	110	0	110	0	0	0	110	0	110	0	110	0	110	0	110	0	110	0	110	0	
9	Daru-Daru	100	0	0	100	0	0	100	0	100	0	100	0	0	50	0	0	100	0	100	0	100	0	50	0	100	0	100	0	
10	Dema	0	0	200	0	0	0	0	0	0	200	0	0	0	0	0	0	200	0	0	0	0	0	0	0	200	0	0	0	
11	Dua Purri	0	100	0	0	100	0	0	0	0	100	0	100	0	0	0	100	0	100	0	100	0	0	0	100	0	100	0	100	
12	Eka Amurti	200	0	0	0	50	0	0	200	0	0	50	0	0	200	0	0	200	0	0	0	200	0	0	0	200	0	200	0	
13	Erniida Suri Bakti	150	0	0	150	0	0	150	0	0	150	0	0	0	150	0	0	150	0	150	0	150	0	0	0	150	0	150	0	
14	Halil	0	0	0	160	0	200	0	0	0	200	0	120	0	0	0	0	200	0	200	0	140	0	0	0	200	0	180	0	
15	Hanny Ponsel	0	0	0	0	360	0	0	0	0	0	0	360	0	0	0	0	0	0	0	180	0	0	0	0	0	180	0	180	
16	Haryani Iftiri	0	0	100	0	100	0	100	0	100	0	100	0	0	0	0	0	100	0	100	0	100	0	0	0	100	0	100	0	
17	Hendra	0	180	0	0	180	0	0	0	180	0	160	0	0	0	180	0	0	180	0	180	0	180	0	0	0	160	0	160	
18	Hijrah	100	0	80	0	50	0	0	100	0	80	0	0	0	100	0	80	0	100	0	80	0	100	0	80	0	80	0	80	
19	Ibu Susi/Abdul Gar	110	0	110	0	110	0	110	0	110	0	110	0	0	110	0	110	0	110	0	110	0	110	0	110	0	110	0	110	
20	Isnawan	0	150	0	0	160	0	0	0	150	0	160	0	0	0	150	0	160	0	160	0	160	0	150	0	160	0	160	0	
21	Iwan/Nirwana	0	0	0	0	220	0	0	0	0	0	220	0	0	0	0	0	220	0	220	0	220	0	0	0	220	0	220	0	
22	Jaimin	0	140	0	0	0	0	0	0	140	0	0	0	0	0	140	0	0	140	0	0	0	0	0	140	0	0	0	0	
23	Jony	150	0	150	0	0	0	150	0	150	0	150	0	0	150	0	150	0	150	0	150	0	150	100	150	0	150	0	150	
24	Karin Mandiri	0	200	0	150	0	0	0	0	200	0	200	0	0	0	200	0	200	0	200	0	200	0	200	0	200	0	200	0	
25	Kasih Ibu	0	100	0	0	0	0	0	100	0	100	0	100	0	0	100	0	100	0	100	0	100	0	100	0	100	0	100	0	



Lampiran 2. (Lanjutan)

No	Pangalan	Jumlah Permintaan/Kandang																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
26	Kusno	0	160	0	0	170	0	0	0	160	0	0	170	0	0	160	0	0	0	170	0	0	0	160	0	0	170	0	0
27	Langgeng	0	0	0	0	0	140	0	0	0	0	140	0	0	0	0	0	0	120	0	110	0	0	0	0	0	0	0	0
28	Lucia Ernauli	110	0	120	0	0	140	0	0	120	0	140	0	0	110	0	120	0	140	0	0	0	0	0	120	0	140	0	0
29	Lusi Amrina	0	0	100	0	100	0	100	0	100	0	100	0	0	100	0	100	0	100	0	0	0	0	0	100	0	100	0	0
30	Mas Rasyid	0	0	0	0	140	0	0	0	0	0	0	140	0	0	140	0	0	0	0	0	0	0	0	0	140	0	0	0
31	Nelson Tua Marto	140	0	0	0	0	0	0	0	0	0	0	0	0	140	0	0	0	0	0	180	0	0	0	0	0	140	0	0
32	Nurul	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180	0
33	Pak Aceh	0	0	100	0	140	0	0	0	0	100	0	140	0	0	0	0	0	0	140	0	0	0	100	0	0	100	0	0
34	Prasetyo	0	0	0	0	200	0	0	0	0	0	200	0	0	0	0	0	0	0	200	0	0	0	0	0	200	0	0	0
35	Rano	0	150	0	0	0	160	0	0	0	100	0	140	0	0	0	140	0	0	0	100	0	150	0	0	100	60	0	0
36	Ranto Roni Sianipa	0	0	0	0	360	0	0	0	0	360	0	0	0	0	0	0	0	0	360	0	0	0	0	0	280	0	0	0
37	Rempatih Putra	150	0	0	0	150	0	0	0	150	0	150	0	0	150	0	150	0	150	0	0	0	150	0	0	150	0	0	0
38	Romi Ramzi	0	120	0	0	150	0	0	0	120	0	150	0	0	120	0	120	0	150	0	0	0	120	0	0	150	0	0	0
39	Rudi Susilo	0	0	0	0	210	80	0	0	0	0	210	80	0	0	0	0	0	210	80	0	0	0	0	0	210	80	0	0
40	Sajuti	180	0	0	0	170	0	0	0	180	0	170	0	0	180	0	170	0	180	0	160	0	100	0	130	0	0	0	0
41	Seno Semosa/Tirt	120	0	0	0	120	0	0	0	120	0	120	0	0	120	0	120	0	120	0	0	0	120	0	0	120	0	0	0
42	Shaza	0	140	0	0	0	0	0	0	140	0	100	0	0	120	0	120	0	140	0	0	0	140	0	0	140	0	0	0
43	Siregar	0	140	140	0	0	140	0	0	0	100	140	0	0	110	110	140	0	140	0	0	0	100	140	0	140	0	0	0
44	Siti Fatimah	40	0	0	0	40	0	0	0	40	0	40	0	0	40	0	40	0	40	0	40	0	40	0	40	0	40	0	0
45	SPBU 142 82 629/1	0	100	0	100	0	100	0	100	0	100	0	100	0	100	0	100	0	100	0	100	0	100	0	100	0	100	0	0
46	SPBU 142 82 650/1	0	140	0	0	180	0	0	0	140	0	180	0	0	160	0	160	0	140	0	180	0	140	0	140	0	180	0	0
47	SPBU 142 82 663/1	0	120	0	0	120	0	0	0	120	0	120	0	0	120	0	120	0	120	0	120	0	120	0	120	0	120	0	0
48	Stefina	0	0	140	0	100	0	0	0	0	140	0	0	0	140	0	140	0	140	0	140	0	140	0	140	0	140	0	0
49	Sudarsono	0	0	140	0	0	0	0	0	140	0	140	0	0	140	0	140	0	140	0	140	0	140	0	140	0	140	0	0
50	Sulisman Aneer	0	180	0	0	0	0	0	0	180	0	0	0	0	180	0	180	0	180	0	0	0	180	0	180	0	180	0	0



Lampiran 2. (Lanjutan)

No	Pangalalan	Jumlah Permintaan/hanggal																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
51	Sulistiyawati/ABS	100	0	0	120	0	0	0	0	0	100	0	0	0	0	100	0	0	100	0	0	0	100	0	0	0	0	0	0
52	Surya Najwa	0	0	0	0	0	0	140	0	0	0	0	0	0	0	0	0	140	0	0	0	0	0	0	0	0	0	0	
53	Syafiq/Mohd. Syafiq	0	0	0	0	0	150	0	0	0	100	0	0	0	0	0	0	200	0	0	0	0	0	0	0	0	0	0	
54	Syamsuardi	80	0	0	80	0	0	80	0	0	90	0	0	0	0	100	0	0	0	0	0	120	0	0	0	90	0	0	
55	Tania	140	0	0	0	0	0	0	0	0	140	0	0	0	0	0	0	0	0	0	0	140	0	0	0	0	0	0	
56	Toko Makmur Jaya	0	0	180	0	0	180	0	0	180	0	180	0	180	0	180	0	180	0	180	0	180	0	180	0	180	0	180	
57	Toko Sumber Jaya	180	0	0	180	0	0	180	0	0	180	0	0	0	0	180	0	180	0	180	0	180	0	180	0	180	0	0	
58	Yuli Alfian	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	360	0	0	0	
59	Yuliami	0	0	0	0	0	0	140	0	0	0	0	140	0	0	0	0	0	0	0	140	0	0	0	0	0	140	0	
60	Yulianti Arimbi	0	200	0	0	0	200	0	0	200	0	200	0	200	0	200	0	200	0	200	0	200	0	200	0	200	0	0	
61	Zakun	0	0	100	0	100	0	100	0	100	0	80	0	0	0	100	0	0	0	0	0	100	0	100	0	100	0	0	



Lampiran 3. Data Biaya Perusahaan 2021

B. Biaya 2021

Keterangan	Total	Januari	Februari
SOLAR	18,000,000	9,400,000	8,600,000
BM8394TO	4,200,000	2,000,000	2,200,000
BM9338TO	3,400,000	2,000,000	1,400,000
BM9224TS	2,400,000	1,200,000	1,200,000
BM9225TS	4,000,000	2,000,000	2,000,000
BM9929TS	4,000,000	2,200,000	1,800,000
STNK & KIR	280,000	280,000	-
BM8394TO	70,000	70,000	-
BM9338TO	-	-	-
BM9224TS	70,000	70,000	-
BM9225TS	70,000	70,000	-
BM9929TS	70,000	70,000	-
REP & MAINT	16,430,000	7,454,000	8,976,000
BM8394TO	2,283,000	307,000	1,976,000
BM9338TO	4,550,000	675,000	3,875,000
BM9224TS	3,000,000	3,000,000	-
BM9225TS	6,102,000	3,472,000	2,630,000
BM9929TS	495,000	-	495,000
UANG JALAN	17,280,000	8,800,000	8,480,000
Plastik Wrap	3,024,000	1,540,000	1,484,000
Jasa Pemasangan Plastik Wrap	6,480,000	3,300,000	3,180,000
By Bongkar	36,288,000	18,480,000	17,808,000
TOTAL	97,782,000	49,254,000	48,528,000



Lampiran 4. Rute Distribusi Algoritma Nearest Neighbour

Tanggal	Pangkalan yang Akan Dikunjungi	Kendaraan	Rute	Jarak Tempuh Rute (Km)
1 Februari 2021	2, 4, 9, 12, 13, 18, 23, 28, 31, 37, 40, 41, 44, 51, 54, 55, 57	1	Depot - SPPBE - 41 - 9 - 31 - 54 - 44 - depot	52,6
		2	Depot - SPPBE - 40 - 18 - 13 - depot	65,7
		3	Depot - SPPBE - 57 - 2 - 12 - depot	61,4
		4	Depot - SPPBE - 51 - 23 - 28 - 37 - depot	97,9
		5	Depot - SPPBE - 55 - 4 - depot	72,3
2 Februari 2021	1, 5, 17, 19, 20, 25, 26, 35, 38, 42, 43, 45, 46, 47, 50, 60	1	Depot - SPPBE - 35 - 20 - 19 - 43 - depot	40
		2	Depot - SPPBE - 42 - 50 - 46 - 25 - depot	67,7
		3	Depot - SPPBE - 5 - 1 - 47 - 45 - depot	70,5
		4	Depot - SPPBE - 38 - 17 - 26 - depot	67,1
		5	Depot - SPPBE - 60 - depot	66,6
3 Februari 2021	2, 3, 6, 11, 16, 18, 22, 24, 28, 29, 33, 43, 48, 49, 56, 61	1	Depot - SPPBE - 56 - 22 - 61 - 43 - depot	38,8
		2	Depot - SPPBE - 18 - 24 - 11 - 48 - depot	41,6
		3	Depot - SPPBE - 2 - 6 - 3 - depot	57,4
		4	Depot - SPPBE - 28 - 33 - 16 - 49 - 29 - depot	75,9
4 Februari 2021	4, 7, 8, 10, 13, 14, 23, 30, 31, 34, 36, 37, 39, 40, 41, 44, 45, 51, 54, 57	1	Depot - SPPBE - 41 - 7 - 31 - 54 - 44 - depot	45,9
		2	Depot - SPPBE - 13 - 40 - 57 - depot	70,3
		3	Depot - SPPBE - 10 - 39 - depot	56,5
		6	Depot - SPPBE - 37 - 34 - 4 - depot	74,4
		4	Depot - SPPBE - 45 - 36 - depot	70,3
		7	Depot - SPPBE - 30 - depot	67,5
		5	Depot - SPPBE - 51 - 23 - 8 - 14 - depot	89,9



Lampiran 4. (Lanjutan)

Tanggal	Pangkalan yang Akan Dikunjungi	Kendaraan	Route	Jarak Tempuh Rute (Km)	
5 Februari 2021	1, 2, 9, 12, 15, 18, 19, 20, 24, 26, 33, 38, 39, 47, 61	1	1	Depot - SPPBE - 20 - 19 - 61 - 9 - 18 - depot	48,2
		2	2	Depot - SPPBE - 24 - 2 - depot	49,5
		3	3	Depot - SPPBE - 15 - 12 - 39 - depot	58,51
		4	4	Depot - SPPBE - 33 - 47 - 1 - 38 - depot	82,4
		5	5	Depot - SPPBE - 26 - depot	65,9
6 Februari 2021	6, 11, 14, 16, 17, 21, 27, 28, 29, 35, 43, 45, 46, 48, 53, 56, 58, 59, 60	1	1	Depot - SPPBE - 35 - 56 - 43 - depot	39,9
		7	7	Depot - SPPBE - 58 - depot	67
		2	2	Depot - SPPBE - 53 - 11 - 48 - 27 - depot	58,4
		3	3	Depot - SPPBE - 6 - 46 - 59 - 29 - depot	60,6
		4	4	Depot - SPPBE - 45 - 16 - 21 - depot	57,7
		5	5	Depot - SPPBE - 14 - 17 - depot	68
8 Februari 2021	2, 4, 8, 9, 12, 13, 18, 23, 31, 37, 40, 41, 44, 52, 54, 57, 61	1	1	Depot - SPPBE - 41 - 61 - 9 - 31 - 54 - depot	52,6
		2	2	Depot - SPPBE - 44 - 18 - 52 - 40 - depot	52,7
		3	3	Depot - SPPBE - 13 - 57 - 2 - depot	72
		4	4	Depot - SPPBE - 12 - 23 - 8 - depot	79,8
		5	5	Depot - SPPBE - 37 - 4 - depot	74,9
9 Februari 2021	1, 17, 19, 20, 25, 26, 38, 42, 43, 45, 46, 47, 50, 60	1	1	Depot - SPPBE - 20 - 19 - 43 - 42 - depot	60,8
		2	2	Depot - SPPBE - 50 - 46 - 25 - 45 - depot	66,3
		3	3	Depot - SPPBE - 47 - 1 - 38 - 17 - depot	74,5
		4	4	Depot - SPPBE - 26 - 60 - depot	67,2



Lampiran 4. (Lanjutan)

Tanggal	Pangkalan yang Akan Dikunjungi	Kendaraan	Route	Jarak Tempuh Route (Km)	
10 Februari 2021	2, 3, 6, 11, 14, 16, 18, 22, 23, 24, 28, 29, 33, 35, 43, 48, 49, 56, 61	1	Depot - SPPBE - 35 - 56 - 22 - 61 - depot	39,4	
		6	Depot - SPPBE - 28 - depot	66,5	
		2	2	Depot - SPPBE - 43 - 18 - 24 - 11 - depot	40
		3	3	Depot - SPPBE - 48 - 2 - 6 - depot	60,7
		4	4	Depot - SPPBE - 3 - 29 - 49 - depot	48,9
11 Februari 2021	7, 8, 9, 10, 13, 31, 34, 36, 37, 39, 40, 41, 44, 45, 51, 53, 54, 55, 57	5	5	Depot - SPPBE - 16 - 33 - 23 - 14 - depot	68,8
		1	1	Depot - SPPBE - 41 - 9 - 7 - 31 - 54 - depot	52,6
		2	2	Depot - SPPBE - 44 - 53 - 13 - 40 - depot	74,3
		3	3	Depot - SPPBE - 57 - 10 - depot	56,3
		6	6	Depot - SPPBE - 8 - 34 - 55 - depot	94,1
12 Februari 2021	1, 12, 17, 19, 20, 25, 26, 33, 38, 39, 42, 47, 61	4	4	Depot - SPPBE - 45 - 39 - depot	70
		5	5	Depot - SPPBE - 36 - 51 - depot	58,9
		7	7	Depot - SPPBE - 37 - depot	65,4
		1	1	Depot - SPPBE - 20 - 19 - 61 - 42 - 25 - depot	67,6
		2	2	Depot - SPPBE - 12 - 39 - 47 - 33 - 1 - depot	85,9
13 Februari 2021	6, 9, 11, 14, 15, 21, 24, 27, 28, 29, 35, 43, 45, 46, 56, 59, 60	3	3	Depot - SPPBE - 38 - 17 - 26 - depot	67,1
		1	1	Depot - SPPBE - 35 - 56 - 9 - 43 - depot	41,8
		2	2	Depot - SPPBE - 24 - 11 - 27 - depot	57,9
		3	3	Depot - SPPBE - 6 - 46 - depot	55,8
		6	6	Depot - SPPBE - 28 - 60 - depot	88,4
	60	4	4	Depot - SPPBE - 59 - 21 - 45 - 29 - depot	69,4
		5	5	Depot - SPPBE - 15 - 14 - depot	65,9



Lampiran 4. (Lanjutan)

Tanggal	Pangkalan yang Akan Dikunjungi	Kendaraan		Route	Jarak Tempuh Rute (Km)
15 Februari 2021	2, 4, 12, 13, 18, 23, 28, 30, 31, 37, 40, 41, 43, 44, 51, 54, 57, 61	1	1	Depot - SPPBE - 41 - 61 43 - 31 - depot	39
		2	2	Depot - SPPBE - 54 - 44 - 18 - 13 - 40 - depot	74,1
		3	3	Depot - SPPBE - 57 - 2 - 12 - depot	61,4
		4	4	Depot - SPPBE - 51 - 23 - 28 - 37 - depot	97,9
		5	5	Depot - SPPBE - 4 - 30 - depot	71,9
16 Februari 2021	1, 5, 9, 17, 19, 20, 25, 26, 38, 42, 43, 45, 46, 47, 48, 50, 60	1	1	Depot - SPPBE - 20 - 19 - 9 - 43 - 48 - depot	51,5
		2	2	Depot - SPPBE - 42 - 50 46 - 25 - depot	67,7
		3	3	Depot - SPPBE - 5 - 1 - 47 - 45 - depot	70,5
		4	4	Depot - SPPBE - 38 - 17 - 26 - depot	67,1
		5	5	Depot - SPPBE - 60 - depot	66,6
17 Februari 2021	2, 3, 6, 11, 14, 16, 22, 28, 29, 35, 43, 49, 53, 56	1	1	Depot - SPPBE - 35 - 56 22 - depot	43,1
		2	2	Depot - SPPBE - 43 - 53 - 11 - 2 - depot	50,8
		3	3	Depot - SPPBE - 6 - 3 - 29 - 49 - depot	57,1
		4	4	Depot - SPPBE - 16 - 14 - 28 - depot	60
18 Februari 2021	3, 4, 7, 8, 10, 13, 18, 23, 24, 27, 37, 39, 41, 44, 45, 51, 52, 57	1	1	Depot - SPPBE - 41 - 7 - 44 - 18 - 24 - depot	45,8
		2	2	Depot - SPPBE - 13 - 52 - 57 - depot	73
		3	3	Depot - SPPBE - 27 - 10 - depot	54,1
		4	4	Depot - SPPBE - 3 - 45 - depot	48,8
		6	6	Depot - SPPBE - 8 - 37 - 4 - depot	102,1
5	5	Depot - SPPBE - 39 - 51 - 23 - depot	64,1		



Lampiran 4. (Lanjutan)

Tanggal	Pangkalan yang Akan Dikunjungi	Kendaraan	Route	Jarak Tempuh Route (Km)	
19 Februari 2021	1, 2, 5, 9, 12, 19, 20, 26, 33, 34, 36, 38, 39, 40, 47, 61	1	1	Depot - SPPBE - 20 - 19 - 61 - 9 - depot	47,3
		2	2	Depot - SPPBE - 40 - 2 - 12 - 39 - depot	61,51
		3	3	Depot - SPPBE - 33 - 47 - depot	68,3
		4	4	Depot - SPPBE - 36 - 1 - depot	65,7
		5	6	Depot - SPPBE - 26 - depot	65,9
20 Februari 2021	6, 11, 14, 15, 16, 17, 18, 21, 24, 27, 28, 32, 35, 43, 45, 46, 48, 56, 59, 60	5	5	Depot - SPPBE - 5 - 38 - 34 - depot	65,9
		1	1	Depot - SPPBE - 35 - 56 - 43 - 18 - depot	40,8
		2	2	Depot - SPPBE - 24 - 11 - 48 - 27 - depot	57,8
		3	3	Depot - SPPBE - 6 - 46 - 32 - depot	60,4
		6	6	Depot - SPPBE - 28 - 60 - depot	88,4
22 Februari 2021	2, 4, 8, 9, 12, 13, 18, 23, 31, 37, 41, 44, 51, 54, 55, 57, 61	4	4	Depot - SPPBE - 15 - 59 - 45 - 16 - depot	65,6
		5	5	Depot - SPPBE - 21 - 14 - 17 - depot	80,5
		1	1	Depot - SPPBE - 41 - 61 - 9 - 31 - 54 - depot	52,6
		2	2	Depot - SPPBE - 44 - 18 - 13 - 57 - depot	73,6
		3	3	Depot - SPPBE - 2 - 12 - 51 - 23 - depot	69,3
23 Februari 2021	1, 5, 17, 19, 20, 23, 25, 26, 33, 35, 38, 40, 42, 43, 45, 46, 50, 60	4	4	Depot - SPPBE - 8 - 37 - 55 - 4 - depot	102,4
		1	1	Depot - SPPBE - 35 - 20 - 19 - 43 - depot	40
		2	2	Depot - SPPBE - 40 - 42 - 50 - 46 - depot	61,7
		3	3	Depot - SPPBE - 25 - 45 - 33 - 23 - 1 - depot	86,3
		4	4	Depot - SPPBE - 5 - 38 - 17 - depot	66,7
		5	5	Depot - SPPBE - 26 - 60 - depot	67,2



Lampiran 4. (Lanjutan)

Tanggal	Pangkalan yang Akan Dikunjungi	Kendaraan	Rute	Jarak Tempuh Rute (Km)	
24 Februari 2021	2, 6, 9, 11, 14, 22, 28, 29, 43, 47, 48, 49, 56, 58, 61	1	1	Depot - SPPBE - 56 - 22 - 61 - 9 - depot	46,9
		2	2	Depot - 11 - 48 - 2 - 6 - 29 - depot	28,5
		3	3	Depot - 43 - 49 - 47 - depot	36,8
		4	4	Depot - 14 - 28 - depot	38,7
		5	5	Depot - 58 - depot	34,6
25 Februari 2021	4, 7, 8, 10, 13, 18, 23, 24, 30, 31, 36, 37, 39, 40, 41, 44, 45, 51, 54, 57	1	1	Depot - 54 - 44 - 13 - 18 - 24 - depot	31
		2	2	Depot - 40 - 57 - 10 - depot	26,7
		3	3	Depot - 31 - 7 - 41 - 45 - depot	27,5
		4	4	Depot - 39 - 36 - depot	34,2
		5	5	Depot - 4 - 30 - depot	42,9
26 Februari 2021	1, 2, 3, 5, 12, 18, 19, 20, 25, 26, 33, 34, 35, 39, 61	1	1	Depot - 18 - 2 - 3 - 25 - depot	32,3
		2	2	Depot - 61 - 19 - 20 - 35 - 12 - depot	44,2
		3	3	Depot - 39 - 33 - 1 - 5 - depot	61,6
		4	4	Depot - 34 - 26 - depot	37,4
27 Februari 2021	6, 9, 11, 12, 14, 15, 16, 17, 21, 24, 26, 28, 29, 32, 35, 38, 43, 45, 46, 47, 56, 59, 60	1	1	Depot - 24 - 11 - 43 - 9 - depot	21,6
		2	2	Depot - 6 - 46 - 32 - depot	34,3
		6	6	Depot - 28 - 38 - 17 - 26 - depot	51,4
		3	3	Depot - 29 - 45 - 16 - 21 - depot	22,7
		4	4	Depot - 56 - 35 - 15 - 59 - depot	45,9
		5	5	Depot - 12 - 47 - 14 - depot	41,3
		7	7	Depot - 60 - depot	37,5
Total kendaraan beroperasi		113	Total Jarak tempuh rute (Km)	7599,32	



Lampiran 5. Visualisasi Rute *Nearest Neighbour*

Tanggal	Gambar
<p>1 Februari 2021</p>	
<p>2 Februari 2021</p>	
<p>3 Februari 2021</p>	



Lampiran 5. (lanjutan)

Tanggal	Gambar
4 Februari 2021	
5 Februari 2021	
6 Februari 2021	



Lampiran 5. (lanjutan)

Tanggal	Gambar
8 Februari 2021	
9 Februari 2021	
10 Februari 2021	



Lampiran 5. (lanjutan)

Tanggal	Gambar
11 Februari 2021	
12 Februari 2021	
13 Februari 2021	



Lampiran 5. (lanjutan)

Tanggal	Gambar
<p>15 Februari 2021</p>	
<p>16 Februari 2021</p>	
<p>17 Februari 2021</p>	



Lampiran 5. (lanjutan)

Tanggal	Gambar
18 Februari 2021	<p>Keterangan: P = Nomor Pangkalan ● = Lokasi Pangkalan - = Route 1 - = Route 2 - = Route 3 - = Route 4 - = Route 5 - = Route 6</p>
19 Februari 2021	<p>Keterangan: P = Nomor Pangkalan ● = Lokasi Pangkalan - = Route 1 - = Route 2 - = Route 3 - = Route 4 - = Route 5 - = Route 6</p>
20 Februari 2021	<p>Keterangan: P = Nomor Pangkalan ● = Lokasi Pangkalan - = Route 1 - = Route 2 - = Route 3 - = Route 4 - = Route 5 - = Route 6</p>



Lampiran 5. (lanjutan)

Tanggal	Gambar
<p>22 Februari 2021</p>	<p>Keterangan: P = Noncor Pangkalan ● = Lokasi Pangkalan — = Route 1 — = Route 2 — = Route 3 — = Route 4</p>
<p>23 Februari 2021</p>	<p>Keterangan: P = Noncor Pangkalan ● = Lokasi Pangkalan — = Route 1 — = Route 2 — = Route 3 — = Route 4 — = Route 5</p>
<p>24 Februari 2021</p>	<p>Keterangan: P = Noncor Pangkalan ● = Lokasi Pangkalan — = Route 1 — = Route 2 — = Route 3 — = Route 4 — = Route 5</p>



Lampiran 5. (lanjutan)

Tanggal	Gambar
25 Februari 2021	
26 Februari 2021	
27 Februari 2021	



Lampiran 6. *Source Code* Algoritma Nearest Neighbour menggunakan *software Python*

```

from __future__ import print_function
from __future__ import division

from logging import log, DEBUG
from operator import itemgetter
from collections import deque
from math import radians, sin, cos, asin, sqrt
from itertools import groupby

import sys
import logging
import numpy as np

def objf(sol, D):
    return sum(( D[sol[i]-
1],sol[i]] for i in range(1,len(sol))))

def without_empty_routes(sol):
    return [n[0] for n in groupby(sol)]

def is_better_sol(best_f, best_K, sol_f, sol_K, minimize_K):
    if sol_f is None or sol_K is None:
        return False
    if best_f is None or best_K is None:
        return True
    elif minimize_K:
        return (sol_K<best_K) or (sol_K==best_K and sol_f<best_
f)
    else:
        return sol_f<best_f

def sol2routes(sol):
    if not sol or len(sol)<=2: return []
    return [[0]+list(r)+[0] for x, r in groupby(sol, lambda z:
z == 0) if not x]

def routes2sol(routes):
    if not routes:
        return None

    sol = [0]
    for r in routes:
        if r:
            if r[0]==0:
                sol += r[1:]
            else:
                sol += r
            if sol[-1]!=0:

```



Lampiran 6. (Lanjutan)

```

        sol += [0]

    return sol

DEBUG_VERBOSITY = 3
COST_EPSILON = 1e-10
CAPACITY_EPSILON = 1e-10
S_EPS = COST_EPSILON
C_EPS = CAPACITY_EPSILON
DEFAULT_DEBUG_VERBOSITY = DEBUG_VERBOSITY

def set_logger_level(level, logfile=None):
    if level>=0:
        logging.basicConfig(format="%(levelname)s: %(message)s",
                            level=logging.DEBUG-level,
                            stream=sys.stdout)
        for lvl in range(1,10):
            logging.addLevelName(lvl, "DEBUG")

        if logfile is not None:
            fileloghandler = logging.FileHandler(logfile)
            fileloghandler.setLevel(logging.DEBUG-level)
            fileloghandler.setFormatter(logging.Formatter("%(l
evelname)s: %(message)s" ))
            logging.getLogger('').addHandler(fileloghandler)

def cli(init_name, init_desc, init_f):
    single = False
    measure_time = False
    verbosity = DEFAULT_DEBUG_VERBOSITY
    minimize_K = False
    output_logfilepath = None
    best_of_n = 1
    interrupted = False

    for i in range(0, len(sys.argv)-1):
        if sys.argv[i]=="-v" and sys.argv[i+1].isdigit():
            verbosity = int(sys.argv[i+1])
        if sys.argv[i]=="-n" and sys.argv[i+1].isdigit():
            best_of_n = int(sys.argv[i+1])
        if sys.argv[i]=="-1":
            single = True
        if sys.argv[i]=="-t":
            measure_time = True
        if sys.argv[i]=="-l":
            output_logfilepath = sys.argv[i+1]
        if sys.argv[i]=="-b":
            otarget = sys.argv[i+1].lower()

```



Lampiran 6. (Lanjutan)

```

if otarget=="cost" or otarget=="c":
    minimize_K = False
elif otarget=="vehicles" or otarget=="k":

    minimize_K = True
else:
    print("WARNING: Ignoring unknown optimization t
target %s"%otarget)

if verbosity>=0:
    set logger level(verbosity, logfile=output logfilepath)

if sys.argv[-1].isdigit():
    N = int(sys.argv[-1])
    problem_name = "random "+str(N)+" point problem"
    N, points, _, d, D, C, _ = cvrp_io.generate_CVRP(N, 100,
20, 5)

    d = [int(de) for de in d]
    D_c = D
    L,st = None, None
    wtt = "EXACT_2D"

    best_sol = None
    best_f = float('inf')
    best_K = len(D)
    for i in range(best_of_n):
        sol, sol_f, sol_K = None, float('inf'), float('inf'
)

        try:
            sol = init_f(points, D_c, d, C, L, st, wtt, sin
gle, minimize_K)
        except KeyboardInterrupt as e:
            print ("WARNING: Solving was interrupted, retur
ning "+
                    "intermediate solution", file=sys.stderr
)

            interrupted = True

            if len(e.args)>0 and type(e.args[0]) is list:
                sol = e.args[0]

    if sol:
        sol = cvrp_ops.normalize_solution(sol)
        sol_f = objf(sol, D_c)
        sol_K = sol.count(0)-1

        if is_better_sol(best_f, best_K, sol_f, sol_K,
minimize_K):
            best_sol = sol

```



Lampiran 6. (Lanjutan)

```

        best_f = sol_f
        best_K = sol_K

    if interrupted:
        break

    print_solution_statistics(best_sol, D, D_c, d, C, L, st
, verbosity=verbosity)

def _haversine(pt1, pt2):
    # ubah koordinat derajat desimal ke radian
    lon1, lat1, lon2, lat2 = map(radians, [pt1[0], pt1[1], pt2[
0], pt2[1]])

    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
    c = 2 * asin(sqrt(a))
    Km = 6371 * c
    return Km

class _PeekQueue:

    def __init__(self, l):
        self.posleft = -1
        self.posright = 0
        self.l = list(l)

    def __len__(self):
        return len(self.l)-(self.posleft+1)+(self.posright)

    def __getitem__(self, idx):
        if idx>=len(self):
            raise IndexError
        return self.l[self.posleft+1+idx]

    def peekleft(self):
        if len(self)==0:
            raise IndexError
        return self.l[self.posleft+1]

    def popleft(self):
        if len(self)==0:
            raise IndexError
        self.posleft+=1
        return self.l[self.posleft]

```




Lampiran 6. (Lanjutan)

```

def peekright(self):
    if len(self)==0:
        raise IndexError
    return self.l[self.posright-1]

def popright(self):
    if len(self)==0:
        raise IndexError
    self.posright-=1
    return self.l[self.posright]

def get_seed_node(seed_mode,D, node_nearest_neighbors, served)
:
    N = len(node_nearest_neighbors)
    seed_node = None
    while True:
        if seed_mode=='farthest':
            seed_node = node_nearest_neighbors[0].popleft()[0]
        elif seed_mode=='closest':
            seed_node = node_nearest_neighbors[0].popright()[0]
        elif seed_mode=='nearest':
            nearest_distance = float('inf')
            for i in range(1,N):
                if served[i]:
                    continue
                elif seed_node==None:
                    seed_node = i
                    while not served[i] and len(node_nearest_neighb
ors[i])>0:
                        nearest = node_nearest_neighbors[i].peeklef
t()[0]
                        if served[nearest]:
                            node_nearest_neighbors[i].popleft()
                        else:
                            break
                        if D[i,nearest] < nearest_distance:
                            seed_node = i
                            nearest_distance = D[i,nearest]
            else:
                raise ValueError("Only 'farthest', 'closest', and "
                                "'nearest' route initialization "+
                                "methods are supported")
        if not served[seed_node]:
            break

```



Lampiran 6. (Lanjutan)

```

return seed_node

def nearest_neighbor_init(D, d, C, L=None,
                        emerging_route_count=1,
                        initialize_routes_with="farthest",
                        add_only_to_end=False,
                        forbidden_nodes=None,
                        route_improvement_callback=None):

    # Buat daftar tetangga terdekat
    N = len(D)
    node_nearest_neighbors = [None]*N
    for i in range(N):
        node_nearest_neighbors[i] = _PeekQueue( sorted(enumerat
e(D[i][:]),
                                                    key=ite
mgetter(1)) )
        node_nearest_neighbors[i].popleft() # pop reference to
self

    # Pembukuan pada node yang telah dilayani
    served = [False]*N
    served[0]=True

    if forbidden_nodes:
        for fn in forbidden_nodes:
            served[fn] = True

    sol = [0]
    route_nodes = [None]*emerging_route_count
    prev_added = [None]*emerging_route_count
    route_demands = [0.0]*emerging_route_count
    route_costs = [0]*emerging route count

    try:
        k = 0
        while True:

            if not route_nodes[k]:
                seed_node = get_seed_node(initialize_routes_wi
th, D,
                                        node_nearest_neighb
ors, served)

                route_nodes[k] = deque()
                route_nodes[k].append(seed_node)
                prev_added[k] = seed_node

```



Lampiran 6. (Lanjutan)

```

if C: route_demands[k] = d[seed_node]
if L: route_costs[k] = 0.0
served[seed_node]=True

if __debug__:
    route = [0]+list(route_nodes[k])+[0]
    log(DEBUG-
1, "Initialize route #d with n%d, resulting to %s (%.2f)%"
    (k,seed_node, str(route), objf(route, D
)))

else:
    first_node = route_nodes[k][0]
    last_node = route_nodes[k][-1]
    prev_node = prev_added[k]

    nn_node = None
    while True:
        nn_node = node_nearest_neighbors[prev_node]
        .peekleft()[0]
        if served[nn_node]:
            #print("pop", nn_node)
            node_nearest_neighbors[prev_node].pople
ft()
        else:
            break

    if __debug__:
        log(DEBUG-
2, "Consider to extend route #d with n%d"%(k,nn_node))

    delta = D[0,first_node]+D[last_node,nn_node]+D[
nn node,0]
    add_to_end = True
    if not add_only_to_end:
        delta_start = D[0,nn_node]+D[nn_node,first
_node]+D[last_node,0]
        if delta_start<delta:
            delta = delta_start
            add_to_end = False

    # Memeriksa kendala
    constraints_violated = \
        (C and route_demands[k]+d[nn_node]-
C_EPS>C) or \
        (L and route_costs[k]+delta-S_EPS>L)

```



Lampiran 6. (Lanjutan)

```

        if constraints_violated:
            current_route = [0]+list(route_nodes[k])+[0]
        ]
        if route_improvement_callback:
            current_cost = D[0,first_node]+route_costs[k]+D[last_node,0]
            improved_route = route_improvement_callback(
                current_route, route_demands[k], current_cost,
                D, d, C, L, served, node_nearest_neighbors)
            if __debug__:
                if improved route != current route:
                    log(DEBUG-
2, "Route improved from %s (%.2f) to %s (%.2f)"%
                    (str(current_route), objf(current_route, D),
                    str(improved_route), objf(improved_route, D)))
                    current_route = improved_route
            if __debug__:
                log(DEBUG-
1, "Constraint violated, storing route #%d as %s (%.2f)"%
                    (k,str(current_route), objf(current_route,D)))
            sol.extend( current_route[1:] )
            route_nodes[k] = None
        else:
            if C:
                route_demands[k] += d[nn_node]
            if add_to_end:
                route_nodes[k].append(nn_node)
            if L:
                route_costs[k] += D[last_node,nn_node]
            else:
                route_nodes[k].appendleft(nn_node)
            if L:
                route_costs[k] += D[nn_node,first_node]

```



Lampiran 6. (Lanjutan)

```

prev_added[k] = nn_node
served[nn_node]=True

if __debug__:
    route = [0]+list(route_nodes[k])+[0]
    log(DEBUG-
1, "Extending route #%d with %d, resulting to %s (%.2f)"%
        (k, nn_node, str(route), objf(route
,D)))

    node nearest neighbors[prev node].popleft()
    k+=1

# Versi paralel membangun rute K secara paralel
if k==emerging_route_count:

    k = 0

except (IndexError, KeyboardInterrupt) as e:
    interrupted = type(e) is KeyboardInterrupt

for rcustomers in route_nodes:
    if rcustomers:
        sol.extend(rcustomers)
        sol.append(0)

    if __debug__:
        if not interrupted:
            current_route = [0]+list(rcustomers)+[0
]

            croute_f = objf(current_route,D)
            log(DEBUG-
1, "Constraint violated, storing route "
                "%d as %s (%.2f)"%(k,str(current r
oute),croute_f))

            if interrupted:
                if sol:
                    unroutedr = [[n] for n in range(N)
                        if (n not in sol)]
                    if sol and len(sol)>1:
                        interrupted_sol = sol[:-
1]+routes2sol( unroutedr )
                    else:
                        interrupted_sol = routes2sol( unroutedr )
                        raise KeyboardInterrupt(interrupted_sol)

                elif len(sol)==1:

```



Lampiran 6. (Lanjutan)

```

        sol = None

    return sol

def get_snn_algorithm():
    algo_name = "vB95-SNN"
    algo_desc = "van Breedam (1994) Sequential Nearest Neighbor
construction "+\
        "heuristic"
    def call_init(points, D, d, C, L, st, wtt, single, minimize
K):
        if minimize_K:
            raise NotImplementedError("Nearest neighbor algorit
hm does "+
                " not support minimiz
ing the number"+
                " of vehicles")

        return nearest_neighbor_init(D,d,C,L,emerging_route_co
unt=1)
    return (algo_name, algo_desc, call_init)

def get_pnn_algorithm(emerging_route_count="auto"):
    algo_name = "vB95-PNN"
    algo_desc = "Parallel Nearest Neighbor construction heurist
ic"
    if emerging_route_count=="auto":
        def call_init(points, D, d, C, L, st, wtt, single, mini
mize_K):
            if minimize_K:
                raise NotImplementedError("Nearest neighbor alg
orithm does "+
                    " not support minimiz
ing the number"+
                    " of vehicles")

            sol_snn = nearest_neighbor_init(D, d, C, L, emergin
g_route_count=1)
            if single:
                return sol_snn

            auto_route_count = sol_snn.count(0)-1

            best_sol = sol_snn
            best_f = objf(sol_snn,D)
            best_K = auto_route_count
            for k in range(2,auto_route_count+1):

```



Lampiran 6. (Lanjutan)

```

sol = nearest_neighbor_init(D, d, C, L, emergin
g_route_count=k)
sol = without_empty_routes(sol)
sol_f = objf(sol,D)
sol_K = sol.count(0)-1

if is_better_sol(best_f, best_K, sol_f, sol_K,
minimize_K):
    best_sol = sol
    best_f = sol_f
    best_K = sol_K

    return best_sol
elif emerging_route_count>1:
def call_init(points, D, d, C, L, st, wtt, single, mini
mize_K):
    if minimize K:
        raise NotImplementedError("Nearest neighbor alg
orithm does "+
                                " not support minimiz
ing the number"+
                                " of vehicles")
    return nearest_neighbor_init(D, d, C, L, emerging_r
oute_count)
else:
    raise ValueError("Not a valid emerging route count valu
e "+
                    "(%s) for parallel algorithm"%
                    str(emerging_route_count))
return (algo_name, algo_desc, call_init)

if name == " main ":
    cli(*get_pnn_algorithm())
#Menghitung Rute Perjalanan pada Tanggal 1
points = [#Masukkan daftar koordinat pangkalan yang akan
dikunjungi]

n_incl_depot = len(points)
gedesic D = np.zeros( (n_incl_depot, n_incl_depot) )

```



Lampiran 6. (Lanjutan)

```
for i, p1 in enumerate(points):
    for j, p2 in enumerate(points):
        geodesic_D[i,j] = 0.0 if p1==p2 else _haversine(p1,p2)

d={#Masukkan daftar permintaan masing-masing pangkalan}
C=#Masukkan batasan kapasitas kendaraan

solution = nearest_neighbor_init(
    D=geodesic_D,
    d=d,
    C=C)

for route_idx, route in enumerate(sol2routes(solution)):

    print("Route #%d : %s"%(route_idx+1, route))
```