

**IMPLEMENTASI *LINEAR PROGRAMMING* PADA MODEL  
CVRPP UNTUK PENGELOLAAN OPERASIONAL LOGISTIK**

**TUGAS AKHIR**

**Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana  
Terapan Komputer pada Program Studi D-IV Teknik Informatika**



**Oleh:**

**NUR TRI RAMADHANTI ADININGRUM**

**1.20.4.061**

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA  
FAKULTAS SEKOLAH VOKASI  
UNIVERSITAS LOGISTIK DAN BISNIS INTERNASIONAL  
BANDUNG  
2024**

***IMPLEMENTATION OF LINEAR PROGRAMMING IN CVRPP  
MODEL FOR LOGISTICS OPERATIONAL MANAGEMENT***

***FINAL REPORT***

***Submitted as One of the Requirements to Obtain the Degree of Applied  
Bachelor of Informatics Engineering in the Applied Bachelor of Informatics  
Engineering Program***



***Created by:***

**NUR TRI RAMADHANTI ADININGRUM**

**1.20.4.061**

***APPLIED BACHELOR OF INFORMATICS ENGINEERING PROGRAM  
FACULTY OF VOCATIONAL SCHOOL  
UNIVERSITY OF LOGISTICS AND INTERNATIONAL BUSINESS  
BANDUNG  
2024***

## HALAMAN PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

Nama : Nur Tri Ramadhanti Adiningrum  
NPM : 1.20.4.061  
Program Studi : D4 Teknik Informatika  
Perguruan Tinggi : Universitas Logistik dan Bisnis Internasional

Dengan ini menyatakan bahwa tugas akhir yang telah saya buat dengan judul: **"IMPLEMENTASI *LINEAR PROGRAMMING* PADA MODEL CVRPP UNTUK PENGELOLAAN OPERASIONAL LOGISTIK"** adalah asli (orisinil) dan belum pernah diterbitkan/dipublikasikan dimanapun dan dalam bentuk apapun. Demikianlah surat pernyataan ini saya buat dengan sebenar-benarnya tanpa ada paksaan dari pihak manapun juga. Apabila dikemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa tugas akhir yang telah saya buat adalah hasil karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari Universitas Logistik dan Bisnis Internasional dicabutkan/dibatalkan.

Dibuat di : Bandung

Pada tanggal: 29 Juli 2024

Yang menyatakan,



Nur Tri Ramadhanti Adiningrum

## HALAMAN PENGESAHAN

# IMPLEMENTASI *LINEAR PROGRAMMING* PADA MODEL CVRPP UNTUK PENGELOLAAN OPERASIONAL LOGISTIK

1.20.4.061

NUR TRI RAMADHANTI ADININGRUM

Laporan Program Tugas Akhir ini telah diperiksa, disetujui dan disidangkan di  
Bandung, 29 Juli 2024

Oleh :

Penguji Utama



Cahyo Prianto, S.Pd., M.T., CDSP, SFPC  
NIK: 117.84.222

Penguji Pendamping



Nisa Hanum Harani, S.Kom., M.T., CDSP, SFPC.  
NIK: 117.89.223

Menyetujui,  
Koordinator Tugas Akhir



M. Yusril Helmi Setyawan, S.Kom., M.Kom., SFPC.  
NIK: 113.74.163

## HALAMAN PENGESAHAN

# IMPLEMENTASI *LINEAR PROGRAMMING* PADA MODEL CVRPP UNTUK PENGELOLAAN OPERASIONAL LOGISTIK

1.20.4.061

NUR TRI RAMADHANTI ADININGRUM

Laporan Program Tugas Akhir ini telah diperiksa, disetujui dan disidangkan di  
Bandung, 29 Juli 2024

Oleh :

Pembimbing Utama



Cahyo Prianto, S.Pd., M.T., CDSP, SFPC  
NIK: 117.84.222

Pembimbing Pendamping



M. Yusril Helmi Setyawan, S.Kom., M.Kom., SFPC  
NIK: 113.74.163

Mengetahui,  
Ketua Program Studi D-IV Teknik Informatika



Roni Andarsyah, S.T., M.Kom., SFPC  
NIK: 115.88.193

## **KATA PENGANTAR**

Puji serta syukur Alhamdulillah saya panjatkan pada Allah SWT. Karena atas berkat serta rahmat-Nya, saya bisa selesai menyelesaikan Laporan Tugas Akhir. Penulisan Laporan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Terapan pada Program Studi Diploma 4 Teknik Informatika Universitas Logistik dan Bisnis Internasional. Saya menyadari bahwasannya tanpa bantuan serta bimbingan dari berbagai pihak, selama masa perkuliahan sampai pada penyusunan laporan Tugas Akhir ini, sangat sulit bagi saya untuk menyelesaikannya. Karenanya, saya ucapkan banyak terima kasih kepada:

1. Bapak Roni Andarsyah, S.T., M.Kom., SFPC selaku ketua prodi D4 Teknik Informatika Universitas Logistik dan Bisnis Internasional;
2. Bapak M. Yusril Helmi Setyawan, S.Kom., M.Kom., SFPC. selaku koordinator Tugas Akhir tahun 2023/2024 sekaligus pembimbing kedua Tugas Akhir, yang telah yang telah menyediakan waktu, tenaga, dan pikiran untuk membimbing saya dalam penyusunan penulisan laporan Tugas Akhir hingga selesai dituntaskan;
3. Bapak Cahyo Prianto, S.Pd., M.T., CDSP, SFPC. selaku dosen pembimbing pertama Tugas Akhir, yang telah yang telah menyediakan waktu, tenaga, dan pikiran dengan sabar untuk membimbing saya dalam penyusunan penulisan laporan Tugas Akhir hingga selesai dituntaskan;
4. Ibu Nisa Hanum Harani, S.Kom., M.T., CDSP, SFPC selaku dosen penguji Tugas Akhir sekaligus Dosen Wali yang telah yang telah menyediakan waktu, tenaga, dan pikiran untuk menguji dan mengevaluasi saya pada saat sidang berlangsung;
5. PT Pos Indonesia yang telah banyak membantu dalam usaha memperoleh data yang saya perlukan;
6. Orang tua yaitu Ayah dan almh. Ibu yang selalu menjadi pendukung saya baik secara moral, doa, fisik, material dan pendukung lainnya dalam menyelesaikan seluruh pendidikan hingga sampai pada titik akhir pada kuliah sarjana, yang mana Tugas Akhir ini saya persembahkan untuk Ayah dan almh. Ibu yang telah mengisi dunia saya dengan begitu banyak kebahagiaan dan kasih sayang yang begitu dalam. Terimakasih telah menjadi orang tua saya;
7. Kiki dan Elin sebagai adik kandung yang selalu semangat menyambut ketika saya pulang ke rumah dan menjadi hiburan hangat untuk saya;
8. Serta seluruh keluarga saya yang telah memberikan bantuan dukungan material dan moral;

9. Marsello yang selalu menemani serta menyemangati saya dan melakukan banyak hal luar biasa bagi saya dalam segi dukungan dengan tulus selama masa perkuliahan hingga akhir perkuliahan;
10. Algies, Hasna, Ara yang telah banyak menemani, memberikan semangat, memberikan hiburan, dan membantu saya dalam menyelesaikan seluruh tugas di masa perkuliahan ini, tanpa mereka juga saya mungkin tidak akan bisa sampai ditahap ini. Banyak kenangan kami selama kuliah yang tidak akan dapat dilupakan sampai saatnya kami nanti akan mempunyai kehidupan masing-masing.;
11. Shela sebagai sahabat yang telah ikut berjuang bersama serta sebagai partner cerita selama masa SMA hingga masa perkuliahan ini dan telah memberikan bantuan doa dan dukungannya;
12. Teman-teman lainnya Anita, Resa, Mayke yang sering belajar dan diskusi bersama dengan saya;
13. Kelas 4C prodi D4TI Angkatan 2020, yang mau menerima saya dalam bentuk pertemanan di kelas;
14. Serta pihak-pihak lainnya yang tidak bisa saya sebutkan satu persatu;
15. Ucapan terimakasih pada diri saya sendiri yang telah berjuang dengan maksimal selama masa perkuliahan ini, yang telah belajar untuk lebih mandiri, selalu bertahan disaat banyak cobaan dan rintangan yang dihadapi, dan tentunya selalu memaksimalkan seluruh tugas sebagai mahasiswa ULBI.

Akhir kata, semoga Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu, sehingga penelitian ini membawa manfaat bagi pengembangan ilmu.

Bandung, 29 Juli 2024

Nur Tri Ramadhanti Adiningrum

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

---

Sebagai sivitas akademik Universitas Logistik dan Bisnis Internasional (ULBI), saya yang bertanda tangan di bawah ini:

Nama : Nur Tri Ramadhanti Adiningrum  
NPM : 1.20.4.061  
Program Studi : D4 Teknik Informatika  
Jenis karya : Tugas Akhir

Demi pengembangan ilmu pengetahuan, memberikan kepada ULBI **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty- Free Right*) atas karya ilmiah saya yang berjudul:

**“IMPLEMENTASI *LINEAR PROGRAMMING* PADA MODEL CVRPP UNTUK PENGELOLAAN OPERASIONAL LOGISTIK”**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non eksklusif ini Universitas Logistik dan Bisnis Internasional berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Bandung

Pada tanggal : 29 Juli 2024

Yang menyatakan



(Nur Tri Ramadhanti Adiningrum)



Nama Mahasiswa : Nur Tri Ramadhanti Adiningrum  
Program Studi : D-IV Teknik Informatika  
Judul Tugas Akhir : Implementasi *Linear Programming* Pada Model  
CVRPP Untuk Pengelolaan Operasional Logistik

## ABSTRAK

Perusahaan logistik merupakan perusahaan yang memiliki kekhususan dalam penyediaan layanan logistik, yang membantu dalam mengelola fungsi rantai pasokan termasuk pergudangan, distribusi, dan transportasi. Salah satu perusahaan logistik di kota Bandung, yang bergerak di bidang layanan jasa logistik dan memiliki layanan *pickup* yang bertugas untuk memasarkan produk serta melakukan layanan penjemputan barang. Pada layanan ini, tahap perencanaan aktivitas seperti rute dan kapasitas kendaraan merupakan tahapan yang penting. Namun, pada penerapannya perusahaan ini belum menerapkan aktifitas perjalanan dengan rute terbaik atau hanya berdasar pengalaman *driver*, serta kurang memaksimalkan kapasitas angkut kendaraan. *Capacitated Vehicle Routing Problem with Pickup* (CVRPP) adalah metode yang digunakan dalam penanganan masalah ini. Penelitian ini bertujuan pada pembuatan model pencarian jarak dan pemaksimalan kapasitas kendaraan yang dapat meningkatkan efisiensi operasional dalam pengelolaan kapasitas dan rute logistik. Untuk mencapai tujuan penelitian, *Linear Programming* dengan bahasa pemrograman Python digunakan sebagai proses perhitungan yang digunakan dan menghasilkan solusi terbaik. Hasil penelitian menunjukkan bahwa rute yang terbentuk menggunakan *Linear Programming* menghasilkan jarak paling pendek diantara rute lainnya dengan penghematan jarak sebesar 19.99% pada analisis dan 31.92% pada aplikasi. Hal itu juga didukung dengan evaluasi dengan *Optimality Gap* yang bernilai 0% atau solusi yang ditemukan adalah optimal atau sangat baik.

Kata kunci : Logistik, Rute terbaik, Kapasitas kendaraan, Pemrograman Linier, Penjemputan

Nama Mahasiswa : Nur Tri Ramadhanti Adiningrum  
Program Studi : D-IV Teknik Informatika  
Judul Tugas Akhir : *Implementation of Linear Programming in  
CVRPP Model for Logistics Operational  
Management*

### ***ABSTRACT***

*A logistics company is a company that specializes in providing logistics services, which helps in managing supply chain functions including warehousing, distribution, and transportation. One of the logistics companies in the city of Bandung, which is engaged in logistics services and has a pickup service that is tasked with marketing products and providing goods pick-up services. In this service, the stages of activity planning such as routes and vehicle capacity are important stages. However, in its implementation, this company has not implemented travel activities with the best routes or only based on driver experience, and has not maximized the vehicle's transportation capacity. Capacitated Vehicle Routing Problem with Pickup (CVRPP) is the method used in handling this problem. This study aims to create a distance search model and maximize vehicle capacity that can improve operational efficiency in managing logistics capacity and routes. To achieve the research objectives, Linear Programming with the Python programming language is used as the calculation process used and produces the best solution. The results of the study show that the route formed using Linear Programming produces the shortest distance among other routes with a distance savings of 19.99% in the analysis and 31.92% in the application. This is also supported by an evaluation with an Optimality Gap of 0% or the solution found is optimal or very good.*

*Keywords: Logistics, Best Route, Vehicle Capacity, Linear Programming, Pickup*

## DAFTAR ISI

<b>HALAMAN PERNYATAAN ORISINALITAS.....</b>	<b>i</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>iii</b>
<b>KATA PENGANTAR.....</b>	<b>iv</b>
<b>HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS.....</b>	<b>vi</b>
<b>ABSTRAK .....</b>	<b>vii</b>
<b><i>ABSTRACT</i>.....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>ix</b>
<b>DAFTAR TABEL .....</b>	<b>xi</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xiii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah.....	3
1.3    Tujuan dan Manfaat .....	4
1.4    Ruang Lingkup.....	4
1.5    Sistematika Penulisan .....	5
<b>BAB II LANDASAN TEORI .....</b>	<b>7</b>
2.1    Tinjauan Studi .....	7
2.1.1    Penelitian Terkait .....	7
2.2    Tinjauan Teori .....	13
2.2.1    PT. Pos Indonesia.....	13
2.2.2    O-Ranger.....	14
2.2.3    Optimasi.....	16
2.2.4 <i>Vehicle Routing Problem (VRP)</i> .....	16
2.2.5 <i>Vehicle Routing Problem with Pickup and Delivery</i> .....	17
2.2.6 <i>Capacited Vehicle Routing Problem</i> .....	17
2.2.7    Rute Kendaraan.....	17
2.2.8    Kapasitas Kendaraan.....	17
2.2.9    Metode <i>Heuristic</i> .....	18
2.2.10 <i>Linear Programming</i> .....	18
2.2.11    Python .....	18
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>20</b>
3.1    Metodologi Penelitian.....	20

3.2	Tahapan Diagram Metodologi Penelitian.....	21
3.3	Algoritma <i>Linear Programming</i> .....	26
3.4	Alur Kerja <i>Linear Programming</i> .....	27
3.5	Contoh Kerja <i>Linear Programming</i> .....	31
3.6	Metodologi <i>User Centered Design</i> (UCD) .....	37
<b>BAB IV EKSPERIMEN DAN HASIL .....</b>		<b>39</b>
4.1	Data Yang Digunakan .....	39
4.2	Analisa Data.....	43
4.3	Variabel Keputusan, Fungsi Tujuan, <i>Constraint</i> (Batasan).....	51
4.3.1	Variabel Keputusan .....	51
4.3.2	Fungsi Tujuan.....	52
4.3.3	<i>Constraint</i> (Batasan) .....	52
4.4	Implementasi <i>Linear Programming</i> .....	54
4.4.1	Grafik Analisis .....	54
4.4.2	Perhitungan <i>Linear Programming</i> .....	55
4.5	Evaluasi Model Pada Grafik Analisis .....	61
4.5.1	Perbandingan Jarak .....	61
4.5.2	<i>Optimality Gap</i> .....	61
4.6	Implementasi Aplikasi.....	62
4.6.1	Penggunaan Google API Key .....	63
4.6.2	<i>Prototipe</i> .....	65
4.6.3	Hasil Antar Muka .....	68
4.7	Evaluasi Model Pada Aplikasi .....	70
4.7.1	Perbandingan Jarak Pada Aplikasi .....	70
4.7.2	<i>Optimality Gap</i> Pada Aplikasi.....	72
<b>BAB V KESIMPULAN .....</b>		<b>73</b>
<b>BAB VI SARAN .....</b>		<b>74</b>
<b>DAFTAR PUSTAKA .....</b>		<b>75</b>
<b>LAMPIRAN.....</b>		<b>80</b>

## DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait.....	7
Tabel 3. 1 Bentuk tabel simpleks .....	28
Tabel 3. 2 Contoh <i>tabel simpleks</i> .....	31
Tabel 3. 3 Elemen yang terpilih .....	32
Tabel 3. 4 Tabel simpleks setelah baris 1 dinormalisasi .....	33
Tabel 3. 5 Koefisien kolom pivot pada baris 2.....	33
Tabel 3. 6 Tabel simpleks setelah baris 2 diperbarui.....	34
Tabel 3. 7 Koefisien kolom pivot pada baris 3.....	34
Tabel 3. 8 Tabel simplex iterasi 1.....	34
Tabel 4. 1 Kolom untuk data historis <i>pickup per-customer</i> .....	39
Tabel 4. 2 Data Lokasi Dan Label Lokasi.....	40
Tabel 4. 3 Data Lokasi dan <i>Demand</i> .....	43
Tabel 4. 4 Jarak Total Rute Random .....	55
Tabel 4. 5 Lokasi Dan Permintaan Konsumen.....	55
Tabel 4. 6 Hasil Rute Variabel Keputusan .....	56
Tabel 4. 7 Peluang Rute .....	58
Tabel 4. 8 Rute Yang Dihasilkan <i>Linear Programming</i> .....	60
Tabel 4. 9 Perbandingan Hasil Analisis .....	60
Tabel 4. 10 Jarak Yang Dihasilkan Berbagai Rute .....	61
Tabel 4. 11 Hasil <i>Optimality Gap</i> .....	62
Tabel 4. 12 Data Lokasi dan Demand Untuk Aplikasi.....	69
Tabel 4. 13 Jarak Rute Aplikasi .....	71
Tabel 4. 14 <i>Optimality Gap</i> Rute Aplikasi .....	72

## DAFTAR GAMBAR

Gambar 2. 1 Logo PT Pos Indonesia .....	13
Gambar 2. 2 Alur Operasi Pickup .....	15
Gambar 3. 1 Diagram Alur Metodologi Penelitian .....	20
Gambar 3. 2. Langkah kerja Linear Programming .....	26
Gambar 3. 3 Tahap User - Centered Design(Interaction Design Foundation, n.d.) .....	37
Gambar 4. 1 Rute Graf Dengan Label A-Z.....	42
Gambar 4. 2 Motor O-Ranger .....	42
Gambar 4. 3 Rute Yang Ditentukan Google Maps.....	44
Gambar 4. 4 Rute Yang Diberi Label.....	44
Gambar 4. 5 Dataframe <i>latitude</i> dan <i>longitude</i> titik lokasi.....	45
Gambar 4. 6 Radius lokasi yang digunakan.....	49
Gambar 4. 7 Lokasi tambahan berwarna hijau .....	50
Gambar 4. 8 Garis Hubung Satu Titik ke Titik Lain.....	50
Gambar 4. 9 Bentuk Rute Yang Dihasilkan .....	54
Gambar 4. 10 Hasil rute yang didapatkan.....	59
Gambar 4. 11 Prototype halaman index .....	65
Gambar 4. 12 Prototype halaman hasil .....	67
Gambar 4. 13 Halaman Index Aplikasi .....	68
Gambar 4. 14 Halaman Hasil Aplikasi.....	70

## **DAFTAR LAMPIRAN**

Lampiran 1. Kartu Bimbingan .....	80
Lampiran 2 Surat Persetujuan Pelaksanaan Wawancara & Observasi.....	83
Lampiran 3 Surat Permohonan Wawancara Observasi Ke KCU Bandung.....	84

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Logistik adalah aliran barang atau jasa dari sumber ke tujuan untuk memenuhi kebutuhan pelanggan. Fokus logistik adalah pergerakan barang atau jasa (Ridwan et al., 2020). Tujuan logistik adalah menyediakan barang dalam jumlah yang tepat, pada waktu yang tepat, di tempat yang tepat, dan dengan biaya yang tepat. Perusahaan logistik memiliki kekhususan dalam penyediaan layanan logistik, yang membantu dalam mengelola fungsi rantai pasokan termasuk pergudangan, distribusi, dan transportasi (Ngo, 2023). Pendekatan terbaik untuk logistik adalah dengan memprioritaskan efektivitas untuk mengelola ketepatan ketika memenuhi permintaan pelanggan (Bantacut & Fadhil, 2018). Salah satu perusahaan logistik di Indonesia adalah PT Pos Indonesia. PT Pos Indonesia adalah suatu Badan Usaha Milik Negara (BUMN) yang bergerak di bidang layanan jasa logistik, kurir, serta transaksi keuangan (Pos Indonesia, 2023a). Perusahaan ini memiliki layanan bernama O-Ranger yang bertugas untuk memasarkan produk Pos Indonesia serta melakukan layanan penjemputan barang (*pick up service*) (Pos Indonesia, 2023b).

Beberapa dekade terakhir terjadi peningkatan pemanfaatan optimasi, berdasarkan teknik Riset Operasi dan Pemrograman Matematika, untuk manajemen penyediaan barang dan jasa yang efektif dalam sistem distribusi dan transportasi (Moghdani et al., 2021). Bidang tersebut dapat dioptimalkan dengan menerapkan *Vehicle Routing Problem* (VRP), yang digambarkan sebagai masalah perancangan rute pengiriman atau pengumpulan yang optimal dari satu depot ke sejumlah kota atau pelanggan yang tersebar secara geografis, dengan adanya kendala samping (Sitompul & Manasye Horas, 2021). Masalah perutean kendaraan dapat didefinisikan sebagai masalah pencarian rute optimal dengan tujuan meminimalkan jarak tempuh, waktu, dan biaya yang digunakan dalam suatu proses pada bidang distribusi dan transportasi (Liu et al., 2023). Banyak algoritma telah diusulkan untuk mengatasi VRP seperti *Heuristic Algorithm* (Claudiu Pop et al., 2011) sesuai dengan kebutuhan proyek bisnis seperti optimasi rute dan kapasitas untuk menemukan jarak optimal berdasarkan permintaan pelanggan dan kapasitas



kendaraan(Moudya et al., 2023)(Chi & He, 2023). Adapun contoh lain penerapannya adalah masalah perutean kendaraan berkapasitas *pickup* (Koc et al., 2020).

Pada layanan *pickup*, tahap perencanaan aktivitas dan manajemen merupakan tahapan yang penting. Perencanaan rute yang efektif dapat secara signifikan mengurangi biaya transportasi keseluruhan(Chi & He, 2023). Selain dari rute, tentunya perusahaan logistik memiliki moda transportasi yang tentunya isi barang harus sesuai dengan kapasitasnya(Limei et al., 2018). Atribut berat barang banyak digunakan untuk mencerminkan pembatasan kapasitas kendaraan pada barang yang dimuat(Chi & He, 2023). Meskipun terdapat berbagai algoritma pencarian rute dan optimasi kapasitas yang tersedia, namun penerapannya dalam konteks layanan *pickup* di PT. Pos Indonesia ini masih belum menerapkan perjalanan rute terbaik. Dimana perusahaan ini belum menerapkan aktifitas perjalanan dengan rute terpendek atau hanya berdasar pengalaman *driver* dan belum memaksimalkan barang pada kapasitas kendaraan. Baik rute yang ditempuh tersebut jauh atau tidak, *pickup* tetap dilakukan untuk perjalanan tersebut.

Dalam konteks tersebut, optimasi rute logistik dan kapasitas kendaraan menjadi krusial pada pengaturan *pickup*. Dengan mempertimbangkan variabel-variabel seperti jarak dan kapasitas kendaraan, hasil penelitian ini ditujukan untuk memberikan solusi yang optimal dalam pelayanan *pickup*. Sehingga, variabel yang dipertimbangkan dalam penelitian ini adalah hanya jarak dan kapasitas kendaraan dengan faktor-faktor lain seperti waktu, biaya, atau kondisi lalu lintas tidak dipertimbangkan, dengan asumsi bahwa faktor-faktor lain tidak signifikan atau dapat diabaikan. Maka, penelitian ini bertujuan pada pembuatan model pencarian jarak dan kapasitas kendaraan yang dapat meningkatkan efisiensi operasional dalam pengelolaan rute logistik. Oleh karena itu, penelitian ini direkomendasikan untuk mengisi gap pengetahuan ini dengan fokus pada optimasi jarak dan kapasitas kendaraan untuk layanan *pickup* dalam konteks spesifik di PT. Pos Indonesia.

Pada penelitian ini pencarian yang dilakukan adalah meminimalkan jumlah kendaraan yang digunakan dan meminimalkan total jarak tempuh rute perjalanan, maka jenis VRP yang dapat digunakan adalah *Capacitated VRP* (CVRP) (Jiang et al., 2022) dan *Vehicle Routing Problem Pickup Delivery* (VRPPD) (Mazzuco D. et

al., 2017). Namun, dalam kasus ini, penelitian dispesifikan menjadi *pickup*. Sedangkan *delivery* diabaikan sesuai dengan kegiatan resmi O-Ranger yaitu *pickup*. Maka dalam penelitian ini model yang digunakan adalah *Capacitated Vehicle Routing Problem with Pickup* (CVRPP).

Untuk mencapai tujuan penelitian, penggunaan algoritma *Linear Programming* (LP) yang menggunakan paket PuLP dengan bahasa pemrograman Python yang dilakukan sebagai proses perhitungan yang digunakan dan menghasilkan solusi terbaik. Untuk memaksimalkan hasil jarak pada rute yang dilalui, digunakan media Google API Key agar jarak dihitung sesuai dengan jalan sebenarnya atau bukan dalam bentuk garis lurus antar titik lokasi. Pembuatan antar muka sederhana juga dapat digunakan sebagai visualisasi dari model yang telah dibuat dalam pengelolaan rute logistik dengan menggunakan *framework* Flask.

Kondisi O-Ranger sebagai objek yang diteliti saat ini adalah, pada layanan *pickup*, proses yang dilakukan adalah kurir yang berada di depot (perusahaan logistik) menerima permintaan *pickup* dari konsumen. Kemudian kurir akan *mengambil* barang ke lokasi konsumen tersebut. Setelah kurir menerima barang dari konsumen, maka kurir akan kembali ke depot untuk meletakkan barang yang diterima oleh kurir.

Penelitian ini menerapkan proses perhitungan dengan memanfaatkan variabel yang didapat langsung dari PT. Pos Indonesia, baik data lokasi kantor depot, lokasi konsumen, dan ketentuan kapasitas transportasi. Dengan demikian, penelitian ini diharapkan dapat memberikan manfaat untuk perusahaan bidang logistik, khususnya pada jasa layanan *pickup* serta diharapkan dapat membantu dalam peningkatan kinerja layanan *pickup* pada O-Ranger.

## 1.2 Rumusan Masalah

Adapun identifikasi masalah pada penelitian ini sebagai berikut,

- a) Bagaimana cara mencari jarak terbaik dan memaksimalkan kapasitas kendaraan dalam kegiatan *pickup*?
- b) Bagaimana cara melakukan optimasi rute logistik dan kapasitas kendaraan dalam layanan *pickup*?

### 1.3 Tujuan dan Manfaat

Adapun tujuan pada penelitian ini sebagai berikut,

- a) Melakukan pembuatan model optimasi jarak dan kapasitas kendaraan yang dapat meningkatkan efisiensi operasional dalam pengelolaan rute logistik menggunakan model *Capacitated Vehicle Routing Problem with Pickup*.
- b) Menggunakan algoritma *Linear Programming* yang menggunakan paket PuLP Python dengan bahasa pemrograman Python yang dilakukan sebagai proses perhitungan optimasi pada layanan *pickup*. Variabel yang digunakan adalah variabel jarak antar lokasi dan informasi kapasitas kendaraan yang memengaruhi rute jarak dan kapasitas kendaraan dari pelaksanaan *pickup*.

Adapun manfaat pada kegiatan ini sebagai berikut,

- a) Pengoptimalan rute logistik dan kapasitas kendaraan, perusahaan dapat meningkatkan efisiensi operasional dalam proses *pickup*.
- b) Optimasi rute logistik dan kapasitas kendaraan dapat membantu perusahaan mengurangi biaya pengiriman dengan meminimalkan jarak tempuh, menghindari perjalanan yang tidak perlu, dan mengoptimalkan penggunaan kapasitas kendaraan.
- c) Perusahaan dapat meningkatkan produktivitas dan kinerja keseluruhan dalam pengaturan *pickup*.

### 1.4 Ruang Lingkup

Berikut ini menjelaskan ruang lingkup penelitian.

- a) Penelitian hanya mencakup tentang optimasi rute jarak kendaraan dengan kapasitas kendaraan pada kegiatan *pickup*.
- b) Variabel yang digunakan adalah jarak antar lokasi dan kapasitas kendaraan.
- c) Titik lokasi yang digunakan adalah KCU Bandung sebagai depot dan 25 lokasi lainnya merupakan lokasi konsumen.
- d) Radius lokasi yang digunakan adalah sepanjang 1.017 km dari pusat lokasi dengan koordinat (-6.920806, 107.604099). Diluar radius tersebut, lokasi diabaikan.

- e) Model yang digunakan untuk optimasi adalah Linear Programming dengan media Pustaka PuLP.
- f) Bahasa pemrograman yang digunakan adalah Python.

Adapun Batasan masalah dalam penelitian ini adalah sebagai berikut.

- a) Penelitian ini tidak mempertimbangkan kondisi jalan raya seperti kemacetan atau kondisi jalan yang rusak.
- b) Waktu perjalanan tidak diperhitungkan dalam model optimasi.
- c) Konsumsi bahan bakar kendaraan tidak dimasukkan sebagai variabel dalam penelitian ini.
- d) Faktor eksternal seperti cuaca dan kejadian tak terduga lainnya tidak dipertimbangkan.
- e) Penelitian ini tidak mengintegrasikan data real-time untuk kondisi lalu lintas dan cuaca.
- f) Sistem tracking GPS untuk pemantauan rute secara real-time tidak digunakan dalam penelitian ini.
- g) Radius lokasi dibatasi hanya pada 1.017 km dari pusat lokasi, sehingga lokasi di luar radius tersebut diabaikan.
- h) Penelitian ini hanya menggunakan satu depot (KCU Bandung) dan 25 lokasi konsumen yang telah ditentukan.

## **1.5 Sistematika Penulisan**

Penyusunan laporan yang cermat menjadi enam bagian ini didasarkan pada sejarah dan rumusan masalah yang diangkat di atas, khususnya :

### **BAB I PENDAHULUAN**

Bab ini berisi penjelasan terkait dengan pemaparan teori umum dengan topik yang dibahas secara khusus dan mengaitkan dengan referensi yang ada. Identifikasi masalah menjelaskan mengenai masalah dalam optimasi rute logistik dengan kapasitas kendaraan serta memberikan solusi atas masalah tersebut. Ruang lingkup menjelaskan mengenai batasan dalam pemodelan dan aplikasi tersebut. Serta sistematika penulisan menjelaskan tentang isi dari aplikasi tersebut.

## **BAB II LANDASAN TEORI**

Bab ini berisi penjelasan mengenai konsep dasar dan pendukung dari penelitian yang akan dilakukan dengan menggunakan metode tertentu, antara lain teori-teori dasar yang mendukung dan relevan dengan penelitian serta penelitian sebelumnya yang berhubungan dengan tema yang di ambil.

## **BAB III METODOLOGI PENELITIAN**

Bab ini memberikan gambaran tentang diagram metodologi penelitian serta tahapan-tahapan diagram metodologi penelitian yang harus diselesaikan agar penelitian dapat diselesaikan dan menghasilkan hasil yang diinginkan.

## **BAB IV EKSPERIMEN DAN HASIL**

Bab ini akan memperkenalkan eksperimen yang dilakukan untuk menguji teori dan metodologi yang telah dijelaskan sebelumnya. Pada bagian ini dijelaskan secara rinci tentang desain eksperimen, variabel yang digunakan, proses pengumpulan data, serta analisis data yang dilakukan. Setelah itu, sampaikan hasil dari eksperimen tersebut dengan menggunakan perbandingan dengan grafik, tabel, atau visualisasi lainnya yang sesuai.

## **BAB V KESIMPULAN**

Rangkuman utama dari seluruh penelitian, termasuk temuan, kontribusi, jawaban terhadap pertanyaan penelitian, implikasi, dan saran untuk penelitian masa depan atau aplikasi praktis.

## **BAB VI SARAN**

Rekomendasi bagi pembaca yang ingin melanjutkan penelitian atau menerapkan temuan dalam konteks praktis, termasuk saran untuk penelitian lanjutan dan implementasi praktis.

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Studi

Pada sebuah upaya dalam melakukan suatu analisis, maka dibutuhkan suatu panduan ataupun rujukan serta dukungan untuk setiap hasil analisis yang sudah ada sebelumnya. Yang tentunya panduan atau rujukan tersebut akan berkaitan dengan suatu analisis yang sedang dilakukan. Hasil dari penelitian-penelitian terdahulu tersebut terdiri dari topik dan pembahasan hasil penelitian yang telah dilakukan sebelumnya.

##### 2.1.1 Penelitian Terkait

Tabel 2. 1 Penelitian Terkait

No	Area Penelitian	Karakteristik Data	Metode	Hasil Penelitian	Objektif
1.	Masalah rute kendaraan berkapasitas penjemputan dengan kendala pemuatan tiga dimensi: Model dan algoritma(2023)(Chi & He, 2023)	Data sample kapasitas, bentuk volumetri 3D	Tree Search Algorithm (TRSA) dan Greedy Heuristic Algorithm (GHA)	Model menghasilkan penghematan biaya yang signifikan: rata-rata 9,37% dan maksimum 26,38%. Selain itu, waktu tambahan untuk memperoleh solusi berkualitas tinggi juga dapat diterima: rata-rata 73,05 detik.	Dasar untuk pengaplikasian teori dalam CVRP <i>pickup</i> .

2.	Optimasi Kapasitas Armada Dengan Menggunakan Metode <i>Linear Programming</i> (2020)(Mardiko & Sulistyowati, 2020)	Data pengiriman dan pengambilan barang dan kapasitas armada	<i>Linear Programming</i>	Hasil uji yang telah dilakukan menunjukkan penghematan biaya bahan bakar armada, yaitu rata-rata 49% untuk pengiriman barang dan 38% untuk pengambilan barang, dengan pengujian sistem sebanyak 36 kali pengujian secara acak. Dengan metode <i>Linear Programming</i> , tentunya dapat membantu operasional perusahaan dalam mengoptimalkan kapasitas armada.	Penerapan model CVRP <i>pickup</i> .
3.	Masalah perutean kendaraan penjemputan dan pengiriman multi-depot kolaboratif dengan beban terpisah dan jendela waktu(2021)(Y. Wang et al., 2021)	Multi-depot, pickup, and delivery	<i>Mixed Integer Linear Programming</i> (MILP)	Efisiensi peningkatan operasi pickup dan delivery melalui kolaborasi antar depot.	Peningkatan efisiensi dalam CVRP <i>pickup</i> .

4.	Optimasi linier bilangan bulat campuran untuk pengambilan dan pengiriman truk penuh (2021)(A. Wang et al., 2021)	<i>Full truckload, pickup, dan delivery</i>	<i>Mixed Integer Linear Programming (MILP)</i>	Pengurangan biaya transportasi dan peningkatan efisiensi rute	Penerapan untuk mengurangi biaya transportasi dan meningkatkan efisiensi operasional.
5.	Implementasi Model <i>Capacited Vehicle Routing Problem with Time Windows</i> dengan Pendekatan <i>Goal Programming</i> pada Penentuan Rute Terbaik Distribusi Barang(2021)(Irawan et al., 2021)	Data simulasi model diambil 25 data pelanggan	Goal Programming	Keempat rute perjalanan terbaik menghasilkan total biaya perjalanan Rp. 233.000,00, total waktu distribusi 17 jam 57 menit, serta total kapasitas barang yang didistribusikan sebesar 6.150 kg.	Menemukan rute perjalanan terbaik yang meminimalkan biaya.
6.	Studi Kasus Optimasi Rute Kendaraan(2021)(Lesch et al., 2021)	Rute kendaraan	<i>Mixed Integer Linear Programming (MILP)</i>	Studi kasus ini menunjukkan bagaimana model MILP dapat mengoptimalkan rute kendaraan dengan	Mengoptimalkan rute kendaraan dengan mempertimbangkan parameter nyata, seperti kapasitas



				mempertimbangkan berbagai parameter nyata.	kendaraan, waktu perjalanan, dan biaya operasional.
7.	Optimalisasi masalah perutean kendaraan berkapasitas dengan pengiriman alternatif, penjemputan, dan jangka waktu(2021)(Paweł Sitek et al., 2021)	12 data kendala dan masalah yang terjadi pada Masalah perutean kendaraan berkapasitas dengan pengiriman alternatif, penjemputan, dan jangka waktu.	BIP ( <i>Binary Integer Programming</i> ) dan integrasi CP ( <i>Constraint Programming</i> ), GA ( <i>Genetic Algorithm</i> ) dan MP ( <i>Mathematical Programming</i> )	Waktu pengoptimalan untuk masing-masing instance data menggunakan versi modifikasi dari pendekatan hybrid (CLP&GA), dapat dikatakan bahwa ini adalah versi yang sangat menjanjikan dari implementasi model jenis ini.	Mengoptimalkan masalah perutean kendaraan berkapasitas untuk meningkatkan efisiensi dan efektivitas operasi logistik.
8.	Optimasi Capacitated Vehicle Routing Problem with Time Windows dengan Menggunakan Ant Colony Optimization(2019)(Soenadi et al., 2019)	Data perusahaan otomotif di daerah Cibitung, Jawa Barat	Ant Colony Optimization	Dapat menghasilkan lima rute optimal dengan pengurangan jarak tempuh kendaraan sebanyak 36 km/hari dibandingkan rute yang sedang berjalan.	Membuat rute optimal yang dapat mengurangi jarak tempuh kendaraan dan meningkatkan efisiensi operasional.

9.	Masalah perutean kendaraan berkapasitas dengan penjemputan dan pengiriman alternatif (CVRPPAD)(2019)(Sitek & Wikarek, 2019)	Informasi latar belakang oleh permasalahan distribusi kiriman pos	MP, hybrid CLP /MP, dan lingkungan CLP/heuristik hybrid	Penggunaan pendekatan terintegrasi ini telah mengurangi ukuran model sebanyak hampir 10 kali lipat, sehingga mengurangi waktu optimasi secara signifikan dan, untuk contoh yang lebih besar, menemukan solusi dalam rentang waktu yang dapat diterima, dibandingkan dengan implementasi MP.	Mengoptimalkan masalah perutean kendaraan berkapasitas dengan penjemputan dan pengiriman alternatif dalam konteks distribusi kiriman pos.
10.	Model VRPPD baru dan pendekatan solusi heuristik hibrid untuk e-tailing(2014)(Yanik et al., 2014)	Kumpulan data nyata di Istanbul, Turki di mana penggunaan <i>e-grocery</i> mengalami percepatan yang nyata baik dalam cakupan maupun skala	Algoritma Genetika	Algoritma Genetika yang diusulkan memandu pencarian keputusan lokasi penjemputan vendor yang optimal, dan untuk setiap solusi yang dihasilkan dalam populasi genetik, CVRPMPDTW yang sesuai diselesaikan	Penerapan VRPPD untuk mengoptimalkan e-tailing, dengan fokus pada penjemputan dan pengiriman.

				menggunakan algoritma penghematan.	
11.	Implementasi Algoritma A-Star Dalam Mencari Jalur Terpendek Distribusi Minyak Goreng di Kabupaten Karo Menggunakan Grafik(2023)(Ropiqoh & Syafitri Lubis, 2023)	Data lokasi titik lokasi distribusi barang	Algoritma A*	Algoritma A-Star dapat meminimalkan jarak dan juga menghemat biaya perusahaan. Dimana persentase penghematan jarak tempuh sebesar 62,16%.	Untuk mencari jalur terpendek dalam distribusi dengan tujuan untuk meminimalkan jarak tempuh dan menghemat biaya operasional perusahaan

## 2.2 Tinjauan Teori

### 2.2.1 PT. Pos Indonesia

Adapun perusahaan yang dipilih untuk menjadi objek penelitian adalah sebagai berikut.

Nama Perusahaan : PT Pos Indonesia (Persero)  
 Kota : Bandung  
 Alamat : Graha Pos Indonesia Jalan Banda No.30, Bandung  
 40115

PT Pos Indonesia adalah suatu Badan Usaha Milik Negara (BUMN) yang bergerak di bidang layanan jasa logistik, kurir, serta transaksi keuangan (Pos Indonesia, 2023a).



Gambar 2. 1 Logo PT Pos Indonesia

PT Pos Indonesia merupakan sebuah Badan Usaha Milik Negara Indonesia yang dimana perusahaan ini menyediakan layanan pos. Untuk mendukung dari kegiatan bisnisnya, hingga diakhir tahun 2021, perusahaan ini telah memiliki 6 kantor regional, 42 kantor cabang utama, serta 168 kantor cabang yang tersebar di Indonesia (Pos Indonesia, 2021). Tujuan perusahaan adalah membangun bangsa yang lebih berdaya saing dan sejahtera (Pos Indonesia, 2023c).

Adapun visi dan misi PT Pos Indonesia (Pos Indonesia, 2023c) adalah sebagai berikut.

**Visi** : “Menjadi *Postal Operator*, Penyedia Jasa Kurir, Logistik dan Keuangan Paling Kompetitif.”

**Misi** : Bertindak Efektif Untuk Mencapai *Performance* Terbaik

### 2.2.2 O-Ranger

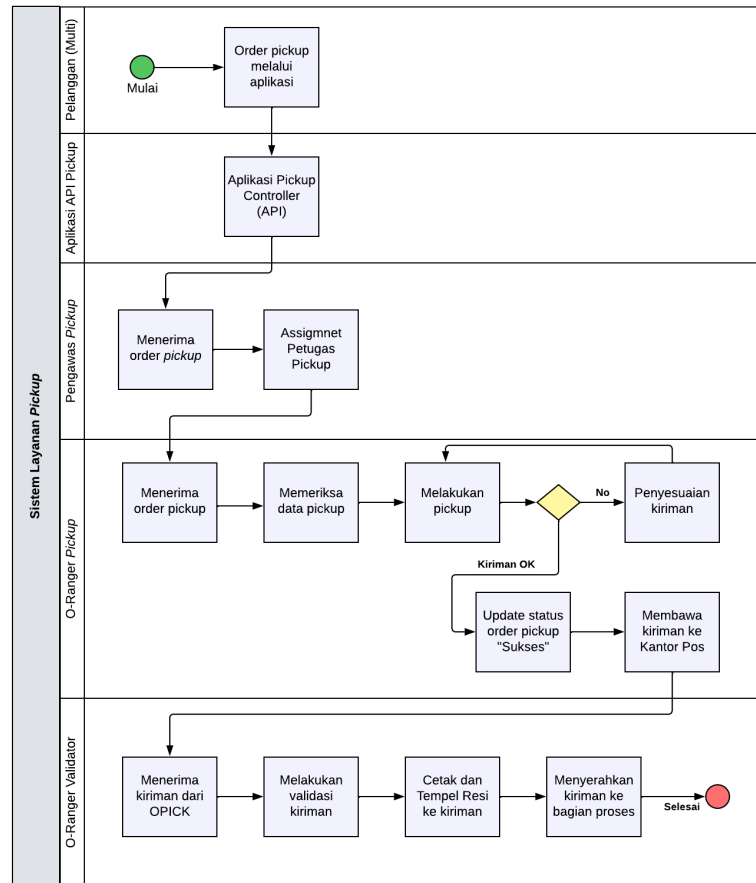
Pada PT Pos Indonesia, terdapat salah satu jasa layanan *pickup* bernama Optimis-Ranger. Optimis-Ranger selanjutnya disingkat O-Ranger adalah sebutan bagi mitra perusahaan pada bidang penjualan, pelayanan, dan operasi (Hariadi (Direktur Operasi Dan Digital Service), 2023). Optimis-Ranger atau O-Ranger adalah perusahaan dari Mitra bersama Pos Indonesia yang berdedikasi untuk menjual produk dari Pos Indonesia dan menyediakan layanan berupa penjemputan di setiap wilayah (PT. Pos Indonesia, 2023).

Pelaksana *pickup* terdiri dari O-Ranger Pickup dan O-Ranger Mobile. Pelaksanaan *pickup* diatur sebagai berikut:

1. Layanan *pickup* ritel, layanan pickup marketplace dan layanan *pickup* agenpos
2. dilakukan oleh O-Ranger Pickup.
3. Layanan *pickup drop point* dilakukan oleh O-Ranger Mobile.

Area *pickup* ditetapkan oleh masing-masing unit komersial, berupa nama wilayah beserta kodepos wilayah tersebut dan dipetakan kepada petugas *pickup* melalui aplikasi *web* SIM Kemitraan. Area *pickup* berfungsi untuk menentukan wilayah *pickup* untuk setiap petugas *pickup*. Dalam sistem, area *pickup* disebut sebagai zona *pickup*.

Gambar 2.2 menunjukkan alur dari proses layanan *pickup* O-Ranger.



Gambar 2. 2 Alur Operasi *Pickup*

Berikut adalah penjelasan alur dari sistem layanan *pickup* :

1. Order permintaan *pickup* berasal dari pelanggan baik melalui aplikasi PosAja, SuperApp, Agenpos, *Marketplace* dan *Drop Point*.
2. Selanjutnya order permintaan *pickup* (*pickup request*) akan masuk ke sebuah sistem layanan *pickup* berupa *API Pickup Controller* berupa aplikasi atau *service*.
3. Dari *API Pickup Controller* akan mengirimkan pesan atau notifikasi permintaan *pickup* ke *Pickup Partner* yaitu pertama ke Pengawas lalu Pengawas akan melakukan assignment ke O-Ranger *Pickup*.
4. O-Ranger *Pickup* menerima notifikasi dan memeriksa notifikasi permintaan *pickup* (lokasi, berat dan dimensi kiriman) lalu O-Ranger *Pickup* melakukan *pickup* ke lokasi, memeriksa kiriman dan melakukan validasi 'Sukses *Pickup*' pada aplikasi di lokasi pelanggan. Selanjutnya

O-Ranger *Pickup* akan membawa kiriman ke kantorpos dan melakukan serah terima kepada O-Ranger Validator.

5. O-Ranger Validator menerima kiriman lalu melakukan validasi kiriman dengan melakukan pemeriksaan berat kiriman, dimensi kiriman, isi kiriman, kemasan kiriman, data pengirim dan penerima. Bila kiriman sudah valid (memenuhi syarat pengiriman) maka O-Ranger Validator mengunduh dan mencetak resi serta menempelkannya pada kiriman. Kiriman yang sudah ditempel resi diserahkan ke Bagian Pemrosesan Kiriman (Puri).

### 2.2.3 Optimasi

Optimasi, disebut juga optimalisasi, adalah proses menghasilkan suatu hasil atau keluaran yang sesuai dengan hasil yang diinginkan, sehingga menjadi strategi yang layak dan efisien. Selain itu, optimasi merupakan strategi untuk memaksimalkan suatu proyek sehingga dapat memberikan hasil yang diinginkan atau tidak diharapkan. Tujuan dari pekerjaan ini adalah untuk menghasilkan hasil yang seefektif dan seefisien mungkin (Rattu et al., 2022).

### 2.2.4 *Vehicle Routing Problem (VRP)*

*Vehicle Routing Problem (VRP)* adalah masalah optimasi kombinatorial penentuan rute yang dapat diterapkan pada pengaturan pendistribusian barang. Penelitian ini menguji salah satu algoritma metaheuristik yaitu algoritma semut pada VRP untuk menentukan rute yang minimal dalam pendistribusian barang (Hadhiatma & Purbo, 2017).

Masalah perutean kendaraan atau yang biasa disebut *Vehicle Routing Problem (VRP)* dapat didefinisikan sebagai masalah pencarian rute optimal dengan tujuan meminimalkan jarak tempuh, waktu, dan biaya yang digunakan dalam suatu proses pada bidang distribusi dan transportasi. Saat ini, berbagai bidang pada manajemen transportasi dan perpindahan barang serta perpindahan manusia banyak diterapkan di lingkungan bisnis dan VRP telah mendapatkan perhatian yang signifikan dari komunitas riset karena banyaknya penerapan di dunia nyata (Liu et al., 2023).

### **2.2.5 Vehicle Routing Problem with Pickup and Delivery**

*Vehicle Routing Problem with Pickup and Delivery* (VRPPD) merupakan perluasan dari *Vehicle Routing Problem* (VRP), dimana kendaraan tidak hanya diharuskan mengantarkan barang ke pelanggan tetapi juga mengambil beberapa barang di lokasi pelanggan. Biasanya diasumsikan bahwa barang yang disimpan di suatu lokasi pelanggan tidak dapat langsung diangkut ke pelanggan lain. Dengan kata lain, semua barang harus berasal dari, atau berakhir, di depot (Nagy & Salhi, 2005).

Pelanggan menginginkan dikirimkan barang atau dokumen dari node 0 yang disebut depot menuju depot yang lain. Rute pengiriman yang dimulai dan diakhiri di depot diperlukan untuk kendaraan agar semua permintaan pelanggan terpenuhi dan setiap pelanggan hanya dikunjungi oleh satu kendaraan (Ary, 2022).

### **2.2.6 Capacitated Vehicle Routing Problem**

*Capacitated Vehicle Routing Problem* (CVRP) merupakan untuk menentukan rute armada kendaraan untuk melayani sekelompok pelanggan dengan biaya perjalanan minimum. Tujuannya adalah untuk merancang serangkaian rute dengan biaya perjalanan paling sedikit, semuanya dimulai dan diakhiri di depo. Total permintaan pelanggan pada setiap rute tidak boleh melebihi kapasitas kendaraan, dan setiap pelanggan hanya dikunjungi satu kali (Jiang et al., 2022).

### **2.2.7 Rute Kendaraan**

Rute kendaraan dalam konteks optimasi logistik biasanya mengacu pada jalur yang diambil oleh kendaraan untuk mengirim barang atau penumpang dari titik asal ke berbagai tujuan dengan cara yang paling efisien. Rute kendaraan harus direncanakan sedemikian rupa sehingga setiap pelanggan dikunjungi tepat satu kali oleh satu kendaraan, dan setiap kendaraan memulai dan mengakhiri rutenya di depot yang sama (Tan & Yeh, 2021).

### **2.2.8 Kapasitas Kendaraan**

Kapasitas kendaraan mengacu pada batas maksimum barang atau penumpang yang dapat diangkut oleh kendaraan dalam satu perjalanan. Dalam konteks VRP, kapasitas kendaraan merupakan salah satu parameter penting yang harus dipertimbangkan untuk memastikan bahwa tidak ada kendaraan yang melebihi



batas kapasitasnya saat mengunjungi berbagai titik pengiriman atau pengambilan. Kapasitas ini dapat dinyatakan dalam berbagai satuan tergantung pada jenis barang atau penumpang yang diangkut, seperti volume, berat, atau jumlah unit. Optimasi kapasitas kendaraan biasanya melibatkan alokasi yang efisien dari muatan ke dalam kendaraan agar memaksimalkan penggunaan kapasitas sambil meminimalkan jumlah perjalanan yang diperlukan (Tan & Yeh, 2021).

### **2.2.9 Metode *Heuristic***

Metode *Heuristic* adalah algoritma dirancang untuk menemukan solusi yang baik dengan cepat, meskipun tidak menjamin solusi optimal. Metode *heuristic* memberikan suatu cara untuk menyelesaikan permasalahan optimasi yang lebih sulit dan dengan kualitas dan waktu penyelesaian yang lebih cepat. Teknik ini mencakup seperti *heuristic* konstruktif dan Algoritma Clarke dan Wright.

### **2.2.10 *Linear Programming***

*Linear Programming* (LP) adalah teknik matematis yang digunakan untuk menemukan solusi optimal dari masalah yang melibatkan sejumlah batasan linear. *Linear programming* merupakan metode yang sangat efektif dalam bidang optimasi dan banyak digunakan dalam berbagai aplikasi seperti manajemen operasi, logistik, ekonomi, dan teknik.

Terdapat dua metode penyelesaian dalam *Linear Programming* yang meliputi metode grafik dan metode simpleks. Metode grafik menggunakan pendekatan grafik dalam menentukan nilai variabel keputusan optimum namun hanya terbatas pada dua variabel keputusan. Sedangkan untuk dua atau lebih variabel keputusan dapat menggunakan metode simpleks dimana metode ini menggunakan pendekatan tabel simpleks sebagai penentu nilai variabel keputusan optimum yang didapat dengan melakukan iterasi secara berulang terhadap tabel simpleks sampai ditemukan nilai yang optimum (Nurmayanti & Sudrajat, 2021).

### **2.2.11 Python**

Python adalah bahasa pemrograman serbaguna untuk keperluan umum digunakan di banyak bidang seperti web, data, statistik, seluler. Ketika Bahasa R sering terbatas untuk bekerja di bidang analisis statistik. Dengan Python, kita bisa belajar dan membangun kepercayaan diri untuk menggunakannya konsep

pemrograman umum dalam bahasa pemrograman populer lainnya seperti Java, C, atau C++.

Python juga mengkompilasi program konsep umum dengan penekanan pada persiapan, operasi, dan pengiriman catatan. Selain itu, Python adalah alat pembelajaran yang kuat, seperti pembelajaran mesin atau pembelajaran mesin. Python telah menjadi alat untuk informasi ilmu pengetahuan atau ilmu informasi.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Metodologi Penelitian

Metodologi penelitian adalah suatu cara untuk memecahkan masalah penelitian secara sistematis. Ini dapat dipahami sebagai ilmu yang mempelajari bagaimana penelitian dilakukan secara ilmiah. Di dalamnya dipelajari berbagai langkah yang umumnya dilakukan seorang peneliti dalam mempelajari masalah penelitiannya beserta logika yang melatarbelakanginya. Peneliti perlu mengetahui tidak hanya metode/teknik penelitian tetapi juga metodologinya (Patel & Patel, 2019). Adapun dalam penelitian ini terdiri dari metode pengumpulan data dan diagram alur metodologi yang digunakan untuk memecahkan masalah.

Adapun langkah-langkah dalam pengerjaan penelitian dipaparkan dalam bentuk diagram alur seperti pada Gambar 3.1.



Gambar 3. 1 Diagram Alur Metodologi Penelitian

### 3.2 Tahapan Diagram Metodologi Penelitian

Berdasarkan diagram alur metodologi penelitian diatas, terdapat indikator capaian sebagai berikut.

#### 3.2.1 Tahap Perencanaan

Tahap perencanaan ini menggunakan metode observasi dan wawancara untuk mengetahui proses bisnis awal yang bertujuan untuk menemukan masalah. Pada tahap ini terdapat tiga kegiatan yaitu dipaparkan sebagai berikut.

##### 1. Mendeskripsikan proses bisnis

Penelitian ini berfokus pada pengoptimalkan rute dan kapasitas kendaraan pada masalah kendaraan berkapasitas *pickup* dan rute perjalanan *pickup*. Metodologi yang digunakan mencakup analisis algoritma optimasi yaitu *Linear Programming* untuk meminimalkan jarak tempuh dan penggunaan kendaraan. Proses ini melibatkan identifikasi titik-titik penjemputan, serta penentuan rute optimal yang dapat meminimalisasi total jarak perjalanan. Selain itu, penelitian ini juga mempertimbangkan kapasitas kendaraan agar jumlah kendaraan yang digunakan seminimal mungkin tanpa melebihi kapasitas yang tersedia. Hasil yang diharapkan dari penelitian ini adalah tercapainya efisiensi dalam penggunaan waktu dan sumber daya kendaraan, yang pada akhirnya akan mendukung optimalisasi operasional dan biaya dalam logistik.

##### 2. Menentukan topik

Menentukan topik penelitian adalah proses memilih subjek atau masalah tertentu yang akan diselidiki lebih lanjut dengan tujuan mengidentifikasi, menganalisis, dan memahami aspek tertentu dari topik tersebut sesuai dengan tujuan penelitian yang telah ditetapkan. Dimana topik yang dipilih dalam penelitian ini adalah mengoptimalkan rute dan kapasitas kendaraan layanan *pickup* PT Pos Indonesia.

##### 3. Menentukan perumusan masalah dan tujuan

Menentukan perumusan masalah dan tujuan adalah langkah awal dalam proses penelitian yang melibatkan identifikasi secara jelas dan spesifik mengenai pertanyaan atau permasalahan yang ingin dipecahkan, serta tujuan yang ingin dicapai melalui kegiatan penelitian tersebut. Adapun

masalah yang diangkat adalah adanya kebutuhan tentang rute kendaraan yang optimal dan optimasi kapasitas kendaraan layanan *pickup* (Pane et al., 2019). Berdasarkan hal tersebut tujuan penelitian adalah untuk mengetahui rute kendaraan yang paling optimal dan melakukan optimasi kapasitas kendaraan, dalam hal ini adalah rute terpendek berdasarkan faktor jarak dan kapasitas kendaraan untuk melakukan layanan *pickup* O-Ranger. Titik lokasi yang digunakan adalah KCU Bandung sebagai depot dan 25 lokasi lainnya merupakan lokasi konsumen. Radius lokasi yang digunakan adalah sepanjang 1.017 km dari pusat lokasi dengan koordinat (-6.920806, 107.604099). Diluar radius tersebut, lokasi diabaikan. Bahasa pemrograman yang digunakan adalah Python.

### 3.2.2 Tahap Pengumpulan Data

Tahap pengumpulan data ini menggunakan metode pengumpulan data dari hasil tahap perencanaan dan menggunakan metode *literature review*. Pada tahap ini terdapat dua kegiatan yaitu dipaparkan sebagai berikut.

#### 1. Pengumpulan data lokasi dan kapasitas kendaraan

Pengumpulan data historis merujuk pada proses menghimpun informasi, fakta, dan catatan yang terkait dengan peristiwa atau keadaan masa lampau untuk digunakan dalam analisis, penelitian, atau pemahaman lebih lanjut mengenai perkembangan suatu topik atau konteks tertentu dari sudut waktu yang telah lampau. Adapun data yang didapatkan adalah data historis layanan *pickup* yang mencakup wilayah Kota Bandung sebagai sampel data dari bulan Januari sampai dengan Juni 2023.

Selain itu, pengumpulan informasi data lokasi merujuk pada proses menghimpun informasi, fakta, dan catatan yang terkait dengan peristiwa atau keadaan masa lampau untuk digunakan dalam analisis, penelitian, atau pemahaman lebih lanjut mengenai perkembangan suatu topik atau konteks tertentu dari sudut waktu yang telah lampau. Dalam hal ini, informasi yang didapatkan adalah lokasi kantor cabang utama beserta dengan lokasi alamat konsumen dan lokasi percabangan rute. Data yang didapat adalah titik koordinat lokasi. Titik lokasi ini disebut sebagai *node*.

Adapun wilayah yang dipilih untuk digunakan dalam penelitian ini adalah wilayah Kota Bandung dengan kode KPRK nomor 40000.

## **2. Studi Pustaka**

Dilaksanakan dengan cara mempelajari beberapa jurnal, penelitian maupun dokumen yang terkait atau memiliki hubungan dengan penelitian yang sedang dilaksanakan (Trivaika & Senubekti, 2022).

### **3.2.3 Tahap Analisa**

Tahap analisa ini digunakan dua pendekatan formula yaitu Haversine untuk mengukur jarak terpendek dan Euclidean untuk mengukur tingkat kemiripan jarak. Pada tahap ini terdapat dua kegiatan yaitu dipaparkan sebagai berikut.

#### **1. Mengkaji data dan informasi**

Mengumpulkan data mengenai titik-titik penjemputan dan pengantaran, serta *demand* barang. Lalu dilakukan analisis dari data yang telah dikumpulkan untuk mengidentifikasi pola dan karakteristik yang dapat mempengaruhi proses optimasi. Analisis ini mencakup perhitungan jarak antar titik, dan kapasitas kendaraan yang tersedia. Hasil analisis kemudian dipetakan ke dalam model optimasi. Informasi ini digunakan sebagai input dalam algoritma untuk menentukan rute dan kapasitas kendaraan yang optimal.

#### **2. Menghitung jarak antar lokasi**

Untuk mencari jarak terpendek, diperlukan dua formula perhitungan. Rumus Haversine merupakan salah satu cara mencari jarak antara dua titik dengan memperhatikan bahwa bumi bukanlah bidang datar melainkan merupakan bidang yang mempunyai derajat kelengkungan (Al Amin & Wahyudiyono, 2021). Rumus Haversine akan menghasilkan jarak terpendek antara dua titik, misalnya pada bola yang diambil dari garis bujur (*longitude*) dan garis lintang (*latitude*). Nilai Euclidean diterapkan sebagai fungsi heuristik yang diperoleh dari jarak langsung yang ada pada jarak koordinat (Eviana et al., 2022). Euclidean Distance digunakan untuk mengukur tingkat kemiripan jarak antara data dengan rumus Euclidean.

Jarak Euclidean, yaitu perbedaan antara dua titik dalam ruang yang digunakan sebagai landasan dalam rotasi pendulum.

### 3.2.4 Tahap Implementasi

Tahap implementasi ini digunakan dua pendekatan formula yaitu kendala atau *constraint* untuk membatasi apa saja yang harus dilakukan dengan aturan *constraint* dan penerapan algoritma *Linear Programming* untuk implementasi optimasi arak dan kapasitas kendaraan. Pada tahap ini terdapat dua kegiatan yaitu dipaparkan sebagai berikut.

#### 1. Variabel Keputusan, Fungsi Tujuan dan Kendala (*Constraint*)

Tahap menentukan batasan dalam penelitian ini mencakup beberapa aspek penting untuk memastikan fokus dan keberhasilan optimasi. Pertama, penelitian ini hanya mencakup optimasi rute jarak kendaraan dengan kapasitas kendaraan pada kegiatan *pickup*. Variabel utama yang digunakan adalah jarak antar lokasi dan kapasitas kendaraan.

*Constraint* :

- Setiap kendaraan memiliki kapasitas maksimum yang tidak boleh dilampaui oleh total volume barang yang diangkut.
- Setiap Lokasi dikunjungi tepat 1 kali.
- Setiap kendaraan harus memulai rutenya dari depot dan kembali ke depot setelah menyelesaikan rutenya.

#### 2. Implementasi algoritma untuk optimasi jarak dan kapasitas kendaraan

Implementasi algoritma *Linear Programming* (LP) dalam penelitian ini dilakukan menggunakan *library* PuLP di Python. Algoritma LP dipilih karena keunggulannya dalam menangani masalah optimasi dengan banyak variabel dan kendala, serta kemampuannya untuk menemukan solusi optimal dengan efisiensi tinggi. Library PuLP mendukung formulasi model LP yang mudah dan intuitif, memungkinkan peneliti untuk mendefinisikan fungsi tujuan dan kendala dengan jelas.

### 3.2.5 Evaluasi

Tahap evaluasi ini digunakan pendekatan analisis komparatif hasil model dengan kemungkinan hasil lainnya. Pada tahap ini terdapat satu kegiatan yaitu dipaparkan sebagai berikut.

#### 1. *Optimality Gap*

Untuk mengukur seberapa baik atau seberapa dekat sebuah solusi yang ada dengan solusi terbaik, digunakan evaluasi dengan *Optimality Gap*.

#### 2. Melakukan perbandingan untuk pengujian

Analisis komparatif dilakukan untuk mengetahui apakah hasil dari model yang dibuat terbukti lebih baik daripada hasil kemungkinan lainnya yang dicari tanpa menggunakan model. Penggunaan perbandingan dengan kemungkinan rute-rute lain selain yang dipilih oleh algoritma *Linear Programming* dilakukan dengan perbandingan total penjumlahan jarak berangkat dengan jarak kembali. Selain itu perbandingan kapasitas dengan kemungkinan kapasitas lain juga dilakukan.

### 3.2.6 Visualisasi Antar Muka

Tahap visualisasi antar muka ini digunakan metode *User - Centered Design* (UCD) dengan *framework* Flask. Pada tahap ini terdapat satu kegiatan yaitu dipaparkan sebagai berikut.

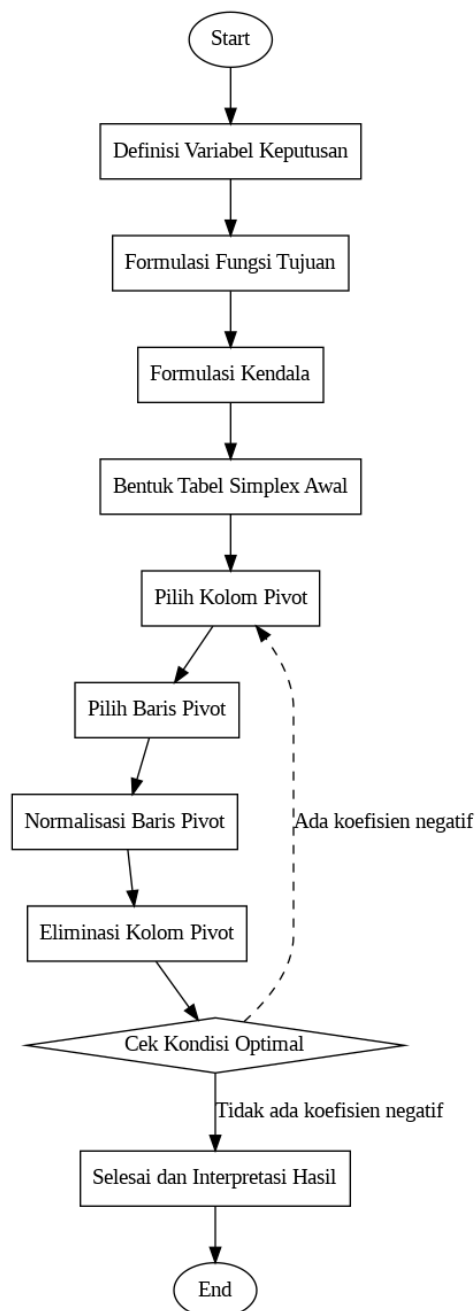
#### 1. Membuat visualisasi inputan sederhana dari hasil pemodelan

*User Centered Design* (UCD) merupakan suatu metode perancangan yang berfokus pada kebutuhan pengguna (Binus University, 2019). Pembuatan GUI dilakukan dengan tujuan untuk penggunaan interaktif untuk perangkat lunak komputer. Maka dari itu, Python dapat digunakan untuk membuat antarmuka yang dapat dijalankan pada kode editor. Penggunaan API Key digunakan untuk mengakses data eksternal yang diperlukan, seperti rute dan jarak yang sebenarnya. Antarmuka diimplementasikan dengan *framework* Flask untuk membuat aplikasi web interaktif. Flask merupakan *micro web framework* yang menggunakan bahasa Python (Wijayanto & Susetyo, 2022). Flask dipilih karena kemampuannya dalam membangun aplikasi web yang ringan.



### 3.3 Algoritma *Linear Programming*

*Linear Programming* (LP) ialah teknik matematis yang digunakan untuk menemukan solusi optimal dari masalah yang melibatkan sejumlah batasan linier. *Linear programming* merupakan metode yang sangat efektif dalam bidang optimasi dan banyak digunakan dalam berbagai aplikasi seperti manajemen operasi, logistik, ekonomi, dan teknik (Nurmayanti & Sudrajat, 2021). Adapun pada Gambar 3.2 adalah cara kerja dari *Linear Programming*.



Gambar 3. 2. Langkah kerja *Linear Programming*

### 3.4 Alur Kerja *Linear Programming*

Alur kerja *Linear Programming* mencakup serangkaian langkah yang sistematis untuk menyelesaikan masalah optimasi menggunakan metode Simplex. Berdasarkan Gambar 3.3, terdapat langkah kerja yang dipaparkan sebagai berikut.

#### 3.4.1 Definisi Variabel Keputusan

Pada tahap ini, ditentukan variabel keputusan yang akan dioptimalkan. Variabel keputusan mewakili keputusan yang harus diambil untuk memecahkan masalah. Misalnya, dalam masalah optimasi rute kendaraan, variabel keputusan dapat berupa jumlah unit yang diangkut oleh kendaraan tertentu ke lokasi tertentu.

Contoh:

$x_{ij}$  : Jumlah unit yang diangkut oleh kendaraan  $i$  ke lokasi  $j$

#### 3.4.2 Formulasi Fungsi Tujuan

Pada tahap ini merumuskan fungsi tujuan yang akan dioptimalkan, baik untuk diminimalkan maupun dimaksimalkan. Fungsi tujuan biasanya mencerminkan tujuan utama dari masalah, seperti meminimalkan biaya atau memaksimalkan keuntungan.

Formula umum:

$$\text{Minimalkan atau Maksimalkan } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

Dimana:

$Z$ : Fungsi tujuan

$c_i$ : Koefisien dari variabel keputusan  $x_i$

$x_i$ : Variabel keputusan

#### 3.4.3 Formulasi Kendala

Pada tahap ini, dirumuskan kendala-kendala yang harus dipenuhi dalam masalah. Kendala-kendala ini mencerminkan batasan-batasan yang ada dalam dunia nyata, seperti kapasitas kendaraan, permintaan di setiap lokasi, dan batasan non-negatif untuk variabel keputusan.

Formula Umum:

$$\begin{aligned}
 a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\leq b_1 \\
 a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\leq b_2 \\
 &\vdots \\
 a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\leq b_m \\
 x_i &\leq 0 \text{ untuk semua } i
 \end{aligned}$$

Dimana:

$a_{ij}$ : Koefisien dari variabel keputusan  $x_i$  dalam kendala ke- $j$

$b_j$ : Batasan kendala ke- $j$

### 3.4.4 Bentuk Tabel Simplex Awal

Pada tahap ini, dibentuk tabel Simplex awal dengan memasukkan fungsi tujuan dan kendala ke dalam bentuk standar. Tambahkan variabel slack (variabel tambahan) untuk mengubah kendala menjadi persamaan.

Struktur Umum Tabel Simplex:

1. Basis: Variabel basis yang saat ini berada dalam basis.
2. Variabel Keputusan: Kolom yang mewakili setiap variabel keputusan.
3. Variabel *Slack*: Kolom yang mewakili setiap variabel *slack*.
4. RHS (*Right Hand Side*): Kolom yang menunjukkan nilai di sisi kanan persamaan kendala.
5. Fungsi Tujuan: Baris yang mewakili koefisien dari fungsi tujuan.

Adapun bentuk tabel simplex secara umum adalah pada Tabel 3.1.

Tabel 3. 1 Bentuk tabel simpleks

Basis	$x_1$	$x_2$	...	$x_n$	$s_1$	$s_2$	...	$s_m$	RHS
$s_1$	$a_{11}$	$a_{12}$	...	$a_{1n}$	1	0	...	0	$b_1$
$s_2$	$a_{21}$	$a_{22}$	...	$a_{2n}$	0	1	...	0	$b_2$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$s_m$	$a_{m1}$	$a_{m2}$	...	$a_{mn}$	0	0	...	1	$b_m$
Z			...		0	0	...	0	Z

#### Penjelasan Kolom dan Baris

1. Basis: Menunjukkan variabel yang saat ini berada dalam basis. Biasanya dimulai dengan variabel slack.
2. Variabel Keputusan ( $x_1, x_2, \dots, x_m$ ): Kolom untuk setiap variabel keputusan yang dioptimalkan.
3. Variabel Slack ( $s_1, s_2, \dots, s_m$ ): Kolom untuk setiap variabel slack yang ditambahkan untuk mengubah kendala menjadi persamaan.
4. RHS (Right Hand Side): Nilai di sisi kanan persamaan kendala.
5. Baris Fungsi Tujuan (Z): Koefisien dari fungsi tujuan yang mencerminkan biaya atau keuntungan yang ingin diminimalkan atau dimaksimalkan.

#### 3.4.5 Pilih Kolom Pivot

Pada tahap ini, dipilih kolom pivot berdasarkan koefisien di baris fungsi tujuan. Untuk masalah minimisasi, kita memilih kolom dengan koefisien negatif terbesar (nilai negatif dengan angka absolut terbesar).

Formula:

Pilih kolom  $j$  dengan  $\min(c_j)$

#### 3.4.6 Pilih Baris Pivot

Pada tahap ini, dipilih baris pivot dengan menghitung rasio antara nilai RHS (*Right Hand Side*) dan koefisien di kolom pivot untuk setiap baris yang memiliki koefisien positif di kolom pivot. Baris dengan rasio terkecil dipilih sebagai baris pivot.

Formula:

Pilih baris  $i$  dengan untuk  $\left(\frac{b_i}{a_{ij}}\right)$  untuk  $a_{ij} > 0$

Atau

$$\text{Hitung rasio} = \frac{RHS}{\text{Koefisien di kolom pivot}}$$

### 3.4.7 Normalisasi Baris Pivot

Pada tahap ini, kita menormalisasi baris pivot dengan membagi seluruh elemen dalam baris pivot dengan elemen pivot. Tujuannya adalah untuk membuat elemen pivot menjadi 1.

Formula:

$$\text{Baris Pivot Baru} = \frac{\text{Baris Pivot Lama}}{\text{Elemen Pivot}}$$

### 3.4.8 Eliminasi Kolom Pivot

Pada tahap ini, dilakukan operasi baris untuk memastikan elemen lainnya di kolom pivot menjadi 0. Ini dilakukan dengan mengurangi kelipatan dari baris pivot yang dinormalisasi dari baris lainnya.

Formula:

$$\text{Baris Baru} = \text{Baris Lama} - (\text{Koefisien Kolom Pivot} \times \text{Baris Pivot yang Dinormalisasi})$$

### 3.4.9 Cek Kondisi Optimal

Pada tahap ini, diperiksa apakah solusi yang ada sudah optimal. Untuk masalah minimisasi, kita cek apakah semua koefisien di baris fungsi tujuan non-negatif. Jika ya, maka solusi optimal telah tercapai. Jika masih ada koefisien negatif, kita kembali ke langkah 3.4.5 dan mengulangi proses hingga mencapai kondisi optimal.

Kondisi yang dikatakan optimal adalah jika semua  $c_j \geq 0$ , Solusi optimal tercapai. Apabila  $c_j$  masih memiliki nilai negatif atau  $c_j \leq 0$ , maka tabel simpleks terakhir dijadikan sebagai iterasi selanjutnya. Begitu juga seterusnya (melakukan iterasi) hingga kondisi optimal sudah tercapai.

### 3.4.10 Selesai dan Interpretasi Hasil

Pada tahap ini, diselesaikan iterasi dan menginterpretasikan hasil dari tabel Simplex terakhir. Hasil ini mencakup nilai optimal dari fungsi tujuan dan nilai variabel keputusan yang memberikan solusi optimal. Interpretasi hasil sangat penting untuk memahami bagaimana solusi ini dapat diterapkan dalam konteks masalah yang dihadapi.

Hasil akhir yang didapat adalah nilai optimal fungsi tujuan ( $Z$ ) dan nilai variabel keputusan ( $x_1, x_2, \dots, x_m$ ).

Untuk membaca nilai variabel keputusan adalah sebagai berikut.

Untuk setiap variabel keputusan  $x_i$ :

1. Jika variabel tersebut ada dalam basis, nilai dari kolom RHS di baris tersebut adalah nilai variabel keputusan  $x_i$ .
2. Jika variabel tersebut tidak ada dalam basis, nilai variabel keputusan  $x_i$  adalah 0.

### 3.5 Contoh Kerja *Linear Programming*

Diketahui fungsi tujuan untuk diminimalkan yaitu:

$$\text{Minimalkan } Z = 3x_1 + 5x_2$$

Dengan kendala yaitu:

$$2x_1 + 3x_2 \leq 8$$

$$4x_1 + 1x_2 \leq 6$$

$$x_1, x_2 \geq 0$$

1. Menambahkan variabel slack.

Langkah awal adalah menambahkan variabel slack ( $s_1$  dan  $s_2$ ) untuk mengubah kendala menjadi persamaan, yaitu menjadi:

$$2x_1 + 3x_2 + s_1 = 8$$

$$4x_1 + 1x_2 + s_2 = 6$$

2. Membuat tabel simpleks.

Selanjutnya masukan semua persamaan yang ada ke dalam tabel simpleks.

Tabel 3. 2 Contoh *tabel simpleks*

Basis	$x_1$	$x_2$	$s_1$	$s_2$	RHS
$s_1$	2	3	1	0	8
$s_2$	4	1	0	1	6
Z	-3	-5	0	0	0

Penjelasan tabel simpleks:

- a. Basis: Awalnya, variabel slack  $s_1$  dan  $s_2$  berada dalam basis.

- b. Variabel Keputusan ( $x_1, x_2$ ): Kolom untuk variabel keputusan  $x_1$  dan  $x_2$ .
- c. Variabel Slack ( $s_1, s_2$ ): Kolom untuk variabel slack  $s_1$  dan  $s_2$ .
- d. RHS (*Right Hand Side*): Nilai di sisi kanan persamaan kendala, yaitu 8 dan 6.
- e. Baris Fungsi Tujuan (Z): Koefisien dari fungsi tujuan Z, dengan koefisien negatif dari variabel keputusan menunjukkan bahwa sedang mencari nilai minimum.

### 3. Memilih kolom pivot.

Pilih kolom dengan koefisien negatif terbesar di baris Z, yaitu  $x_2$  dengan koefisien -5.

### 4. Memilih baris pivot

$$\text{Hitung rasio} = \frac{RHS}{\text{Koefisien di kolom pivot}}$$

$$\text{Untuk baris 1 } (s_1) = \frac{8}{3} = 2,67$$

$$\text{Untuk baris 2 } (s_2) = \frac{6}{1} = 6$$

Maka, rasio terkecil yang digunakan adalah baris 1 dengan hasil rasio 2,67.

### 5. Elemen yang dipilih.

Kolom pivot yang dipilih adalah  $x_2$  dan baris pivot yang dipilih adalah  $s_1$ . Maka elemen yang terpilih adalah 3.

Tabel 3. 3 Elemen yang terpilih

Basis	$x_1$	$x_2$	$s_1$	$s_2$	RHS
$s_1$	2	3	1	0	8
$s_2$	4	1	0	1	6
Z	-3	-5	0	0	0

### 6. Normalisasi baris pivot

Normalisasi baris 1 untuk membuat elemen pivot (koefisien  $x_2$  di baris 1) menjadi 1:

$$\text{Baris Pivot Baru} = \frac{\text{Baris Pivot Lama}}{\text{Elemen Pivot}}$$

$$\text{Baris 1 Baru} = \frac{2, 3, 1, 0, 8}{3}$$

$$\text{Baris 1 Baru} = \left[ \frac{2}{3}, 1, \frac{1}{3}, 0, \frac{8}{3} \right]$$

Tabel simpleks yang baru:

Tabel 3. 4 Tabel simpleks setelah baris 1 dinormalisasi

Basis	$x_1$	$x_2$	$s_1$	$s_2$	RHS
$s_1$	$2/3$	1	$1/3$	0	$8/3$
$s_2$	4	1	0	1	6
Z	-3	-5	0	0	0

#### 7. Eliminasi kolom pivot

Lakukan operasi baris untuk memastikan elemen lainnya di kolom pivot menjadi 0:

$$\text{Baris Baru} =$$

$$\text{Baris Lama} - (\text{Koefisien Kolom Pivot} \times \text{Baris Pivot yang Dinormalisasi})$$

Eliminasi baris 2:

Tabel 3. 5 Koefisien kolom pivot pada baris 2

Basis	$x_1$	$x_2$	$s_1$	$s_2$	RHS
$s_1$	$2/3$	1	$1/3$	0	$8/3$
$s_2$	4	1	0	1	6
Z	-3	-5	0	0	0

$$\text{Baris 2 Baru} =$$

$$\text{Baris 2 Lama} - (\text{Koefisien Kolom Pivot} \times \text{Baris 1 Baru})$$

$$\text{Baris 2 Baru} =$$

$$[4, 1, 0, 1, 6] - \left( 1 \times \left[ \frac{2}{3}, 1, \frac{1}{3}, 0, \frac{8}{3} \right] \right)$$



$$\text{Baris 2 Baru} = \left[ \frac{10}{3}, 0, -\frac{1}{3}, 1, \frac{10}{3} \right]$$

Tabel simplex yang baru:

Tabel 3. 6 Tabel simpleks setelah baris 2 diperbarui

Basis	$x_1$	$x_2$	$s_1$	$s_2$	RHS
$x_2$	$2/3$	1	$1/3$	0	$8/3$
$s_2$	$10/3$	0	$-1/3$	1	$10/3$
Z	-3	-5	0	0	0

Eliminasi baris 3 (Z):

Tabel 3. 7 Koefisien kolom pivot pada baris 3

Basis	$x_1$	$x_2$	$s_1$	$s_2$	RHS
$x_2$	$2/3$	1	$1/3$	0	$8/3$
$s_2$	$10/3$	0	$-1/3$	1	$10/3$
Z	-3	-5	0	0	0

Baris Z Baru =

Baris Z Lama – (Koefisien Kolom Pivot  $\times$  Baris 1 Baru)

Baris Z Baru =

$$[-3, -5, 0, 0, 0] - \left( -5 \times \left[ \frac{2}{3}, 1, \frac{1}{3}, 0, \frac{8}{3} \right] \right)$$

$$\text{Baris 2 Baru} = \left[ \frac{5}{3}, 0, \frac{5}{3}, 0, \frac{40}{3} \right]$$

Sehingga diperoleh tabel simpleks yang baru.yaitu:

Tabel 3. 8 Tabel simplex iterasi 1

Basis	$x_1$	$x_2$	$s_1$	$s_2$	RHS
$x_2$	$2/3$	1	$1/3$	0	$8/3$
$s_2$	$10/3$	0	$-1/3$	1	$10/3$
Z	$5/3$	0	$5/3$	0	$40/3$

8. Cek kondisi optimal

Tidak ada lagi koefisien negatif di baris fungsi tujuan (Z), jadi solusi optimal tercapai.

9. Penjelasan detail hasil tabel simpleks terakhir

Basis: Menunjukkan variabel apa saja yang ada dalam basis pada iterasi terakhir. Dalam hal ini,  $x_2$  dan  $s_2$  adalah variabel basis.

Kolom RHS: Menunjukkan nilai-nilai yang terkait dengan variabel dalam basis pada iterasi terakhir. Dimana:

$x_2$  memiliki nilai  $8/3$ .

$x_2$  memiliki nilai  $10/3$ .

10. Hasil akhir

a. Nilai optimal fungsi tujuan (RHS).

$$Z = \frac{40}{3} = 13,33$$

b. Nilai variabel keputusan.

Nilai variabel keputusan  $x_2$ :

Basis menunjukkan bahwa  $x_2$  adalah variabel yang masuk dalam basis di baris pertama. Nilai  $x_2$  dapat dilihat dari kolom RHS (*Right Hand Side*) dari baris yang memiliki  $x_2$  dalam basis, yaitu:

$$x_2 = \frac{8}{3} = 2,67$$

Nilai variabel keputusan  $x_1$ :

Basis tidak menunjukkan  $x_1$  sebagai variabel dalam basis (variabel slack), yang berarti  $x_1$  adalah variabel non-basis. Variabel non-basis di luar basis memiliki nilai 0 karena tidak berkontribusi dalam kombinasi linear pada solusi optimal. Artinya variabel slack tidak relevan untuk nilai  $x_1$ . Sehingga dihasilkan variabel keputusan berupa:

$$x_1 = 0$$

11. Interpretasi nilai

Untuk mencapai biaya total minimal sebesar 13.33, kendaraan harus mengangkut sekitar 2.67 unit ke lokasi tertentu.

Untuk menggambarkan hasil akhir dalam bentuk formula, dapat dilakukan dengan menyatakan solusi optimal sebagai kombinasi dari variabel keputusan yang memenuhi kendala dan meminimalkan fungsi tujuan.

Fungsi tujuan:

$$Z = 3x_1 + 5x_2$$

Dengan substitusi nilai optimal dari variabel keputusan  $x_1$  dan  $x_2$ , kita dapat menyatakan nilai optimal dari fungsi tujuan sebagai:

$$x_1 = 0$$

$$x_2 = 2,67$$

Maka,

$$Z = 3(0) + 5\left(\frac{8}{3}\right)$$

$$Z = \frac{40}{3} = 13,33$$

Solusi optimal juga harus memenuhi semua kendala yang diberikan.

Dalam kasus ini, nilai  $x_1$  dan  $x_2$  yang optimal adalah:

$$2x_1 + 3x_2 + s_1 = 8$$

$$2(0) + 3\left(\frac{8}{3}\right) + s_1 = 8$$

$$s_1 = 8 - 8$$

$$s_1 = 0$$

$$4x_1 + 1x_2 + s_2 = 6$$

$$4(0) + 1\left(\frac{8}{3}\right) + s_2 = 6$$

$$s_2 = 6 - \frac{8}{3}$$

$$s_2 = \frac{10}{3} = 3,33$$

Maka solusi optimal adalah:

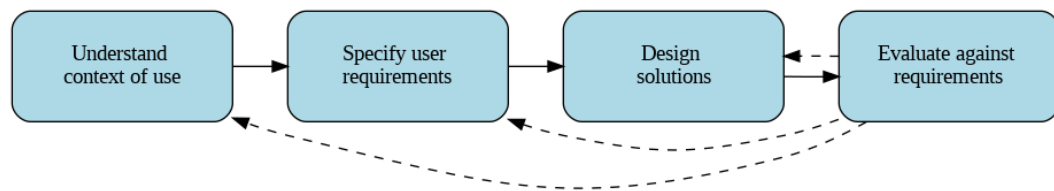
- $x_1 = 0$
- $x_2 = 2,67$

- $s_1 = 0$
- $s_2 = 3,33$

Dengan demikian, masalah optimasi ini telah diselesaikan menggunakan metode Simplex, dan hasil optimal yang diperoleh sesuai dengan kendala yang diberikan.

### 3.6 Metodologi *User Centered Design* (UCD)

*User Centered Design* (UCD) merupakan suatu metode perancangan yang berfokus pada kebutuhan pengguna. Dalam kaitannya dengan Sistem Informasi, *User Centered Design* merupakan bagian dari SDLC (*System Development Life Cycle*), sehingga perancangan aplikasi yang dikembangkan melalui UCD akan dioptimalkan dan berfokus pada kebutuhan *end-user* sehingga aplikasi yang dibuat akan mengikuti kebutuhan pengguna dan pengguna tidak perlu mengubah perilaku untuk menggunakan aplikasi tersebut (Agarina, 2019).



Gambar 3. 3 Tahap *User - Centered Design* (Interaction Design Foundation, n.d.)

Adapun tahapan *User – Centered Design* adalah sebagai berikut (Agarina, 2019).

#### 1. *Understand context of use* (memahami konteks penggunaan)

Perancang sistem perlu memahami konteks di mana sistem akan digunakan, termasuk siapa yang akan menggunakan aplikasi tersebut, tujuan penggunaannya, dan situasi di mana aplikasi akan digunakan.

#### 2. *Specify user requirements* (menentukan kebutuhan pengguna)

Setelah memahami konteks penggunaan, perancang dapat melanjutkan ke tahap berikutnya yaitu menentukan kebutuhan pengguna (*user requirements*). Pada tahap ini, perancang harus dapat mengidentifikasi kebutuhan pengguna dalam konteks bisnis dan tujuan yang ingin dicapai.

**3. *Design solutions* (merancang solusi)**

Tahap berikutnya adalah merancang solusi berdasarkan kebutuhan pengguna yang telah diidentifikasi. Proses ini mencakup beberapa langkah mulai dari konsep awal, pembuatan prototipe, hingga desain akhir.

**4. *Evaluate against requirement* (evaluasi terhadap kebutuhan)**

Evaluasi dilakukan dengan melibatkan pengguna yang akan menggunakan aplikasi. Proses evaluasi ini dilakukan secara bertahap, mulai dari satu proses ke proses berikutnya, untuk memastikan solusi memenuhi kebutuhan pengguna.

## BAB IV

### EKSPERIMEN DAN HASIL

Berdasarkan metodologi yang telah dibuat, adapun pengerjaan dan hasil yang dibahas.

#### 4.1 Data Yang Digunakan

Salah satu tahapan utama dalam penelitian adalah pengumpulan data yang memungkinkan peneliti menemukan jawaban atas pertanyaan penelitian. Pengumpulan data merupakan proses pengumpulan data yang bertujuan untuk memperoleh wawasan mengenai topik penelitian (Taherdoost, 2021). Dari variabel yang digunakan, jarak tempuh yang minimum dipengaruhi oleh bagaimana rute perjalanan yang dilalui (Basriati & Aziza, 2017). Selain itu, penggunaan kapasitas kendaraan distribusi yang tidak maksimal mengakibatkan kebutuhan kendaraan menjadi tinggi. Hal ini tentunya berpengaruh terhadap jauhnya jarak tempuh kendaraan yang dilalui dan biaya yang dikeluarkan membesar (Efendi, 2022).

Sumber yang digunakan untuk penelitian ini yaitu hasil dari wawancara untuk mengetahui proses bisnis layanan O-Ranger. Selain wawancara, adapun data internal perusahaan adalah aktivitas historis transaksi dari O-Ranger. Data yang digunakan dalam penelitian ini adalah data historis *pickup* O-Ranger dengan periode Januari sampai dengan Juni 2023. Data historis layanan O-Ranger digunakan sebagai data internal untuk mempelajari pola atau tren bisnis pada layanan. Data historis O-Ranger mencakup wilayah Kota Bandung sebagai sampel data.

Adapun data historis yang digunakan memiliki total *record* sebanyak 3077 *record*, dan memiliki jumlah kolom sebanyak 11 kolom. Adapun kolom dan penjelasan dari kolom tersebut ialah sebagai Tabel 4.1.

Tabel 4. 1 Kolom untuk data historis *pickup* per-customer

No	Nama Kolom	Penjelasan Kolom
1	OrderID	Id pemesanan atau 1 kali transaksi

2	Tanggal	Waktu <i>customer</i> melakukan order. Waktu pemesanan atau permintaan <i>pickup</i> dapat memengaruhi ketersediaan pengemudi atau kurir. Misalnya, jika banyak pelanggan memesan <i>pickup</i> pada saat yang sama, waktu tunggu mungkin lebih lama.
3	Tanggal_Order	Tanggal pemesanan/order <i>pickup</i> oleh <i>customer</i>
4	Jam_Order	Waktu <i>customer</i> melakukan order (jam)
5	Jenis layanan	Jenis layanan <i>pickup</i> (Pos Regules/Pos Nextday/Pos Sameday/PAKETPOS DANGEROUS GOODS/PKH Marketplace)
6	Lokasi_Penjemputan	Lokasi wilayah <i>pickup customer</i> (dapat berupa daerah kecamatan)
7	Lokasi_Tujuan	Lokasi wilayah tujuan <i>customer</i> (dapat berupa daerah kecamatan)
8	Jadwal_pickup	Jadwal layanan <i>pickup</i> beroperasi
9	Waktu_pickup	Waktu <i>pickup</i> dilaksanakan
10	Pickup_Time	Waktu <i>pickup</i> dilaksanakan
11	Kprk_asal	Kantor pos pengawas cabang

Adapun alamat dan label *node* yang digunakan adalah sebagaimana pada Tabel 4.2.

Tabel 4. 2 Data Lokasi Dan Label Lokasi

No	Nama Lokasi	Label	Koordinat
1	KCU	A	(-6.919792, 107.606014)
2	Lokasi 1	B	(-6.915593, 107.601029)
3	Lokasi 2	C	(-6.927422, 107.610256)
4	Lokasi 3	D	(-6.924132, 107.606715)
5	Lokasi 4	E	(-6.916932, 107.604728)
6	Lokasi 5	F	(-6.927167, 107.606442)
7	Lokasi 6	G	(-6.920149, 107.606520)

8	Lokasi 7	H	(-6.921020, 107.606448)
9	Lokasi 8	I	(-6.921194, 107.607428)
10	Lokasi 9	J	(-6.922484, 107.607500)
11	Lokasi 10	K	(-6.922474, 107.607067)
12	Lokasi 11	L	(-6.922382, 107.606355)
13	Lokasi 12	M	(-6.925660, 107.606076)
14	Lokasi 13	N	(-6.925884, 107.607131)
15	Lokasi 14	O	(-6.927419, 107.607004)
16	Lokasi 15	P	(-6.928340, 107.609310)
17	Lokasi 16	Q	(-6.927377, 107.605908)
18	Lokasi 17	R	(-6.927251, 107.603658)
19	Lokasi 18	S	(-6.920806, 107.604099)
20	Lokasi 19	T	(-6.920128, 107.598304)
21	Lokasi 20	U	(-6.916292, 107.598179)
22	Lokasi 21	V	(-6.916175, 107.600292)
23	Lokasi 22	W	(-6.914935, 107.600452)
24	Lokasi 23	X	(-6.916161, 107.600585)
25	Lokasi 24	Y	(-6.915815, 107.604469)
26	Lokasi 25	Z	(-6.915489, 107.606512)

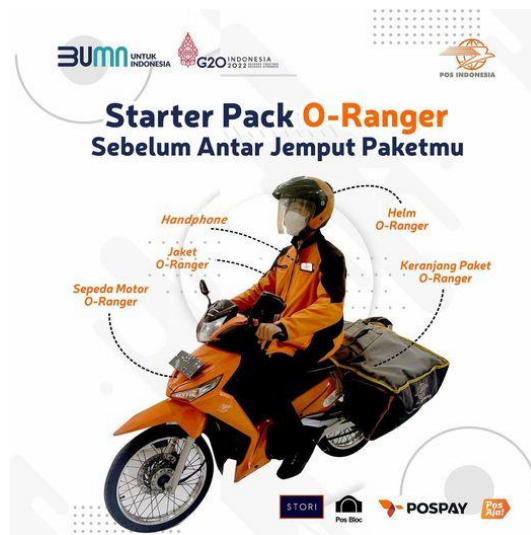
Sementara itu, *node* G-Z merupakan *node* lainnya yang terdiri dari informasi percabangan rute berupa *longitude* dan *latitudenya* masing-masing. *Node* A ditentukan sebagai titik awal, sehingga rute dimulai dari *node* A dan berakhir di *node* A. Informasi dicari menggunakan Google Maps untuk menyajikan rute yang terbentuk seperti pada Gambar 4.1.





Gambar 4. 1 Rute Graf Dengan Label A-Z

Data lokasi depot yang digunakan adalah terletak pada koordinat *latitude* - 6.919792 dan koordinat *latitude* 107.606014. Untuk lokasi konsumen dapat menggunakan lokasi koordinat konsumen.



Gambar 4. 2 Motor O-Ranger

Sesuai Gambar 4.2, pada KCU PT Pos Indonesia Bandung, O-Ranger menggunakan kendaraan bermotor untuk melakukan tugas *pickup*. Terdapat aturan hukum yang sudah tercantum dalam Peraturan Pemerintah No. 74 tahun 2014 tentang barang bawaan atau tepatnya merujuk ke pasal 10 Ayat 4 dan pasal 11. Dalam peraturan itu disebutkan muatan memiliki lebar tidak melebihi setang kemudi. Lalu barang muatan ditempatkan di belakang pengemudi. Selain itu, soal tinggi barang disebutkan bahwa barang bawaan tidak melebihi 900 milimeter atau

kurang dari satu meter dari atas tempat duduk pengemudi. Dijelaskan juga pengendara harus mengutamakan faktor keselamatan (Kompas.com, 2019). Oleh karena itu, tas obrok umumnya digunakan untuk jasa pengiriman dan didesain dengan standar keselamatan. Karena sepeda motor memiliki muatan yang kecil, keberadaan tas yang obrok sangat membantu memudahkan kurir untuk membawa dan mengantarkan paket. Tas obrok adalah tas yang memiliki bentuk persegi panjang besar dan juga memiliki dua kantong besar di kiri dan kanan (Supriani et al., 2023). Total maksimal beban kapasitas yang dapat diangkut oleh tas obrok adalah 60 kilogram (Tokopedia, n.d.), dengan total kendaraan yang tersedia diasumsikan terdapat empat kendaraan yang tersedia pada depot.

Adapun lokasi dan *demand* permintaan yang dibuat secara sintesis dengan menggunakan bahasa pemrograman Python dengan data *dummy*. Untuk lokasi depot berada ditengah – tengah dan dikelilingi oleh lokasi konsumen sehingga membentuk seperti *hexagon*. *Demand* pada depot adalah 0 karena tidak terjadi kegiatan *pickup*. Adapun data yang dibuat adalah sebagai Tabel 4.3.

Tabel 4. 3 Data Lokasi dan *Demand*

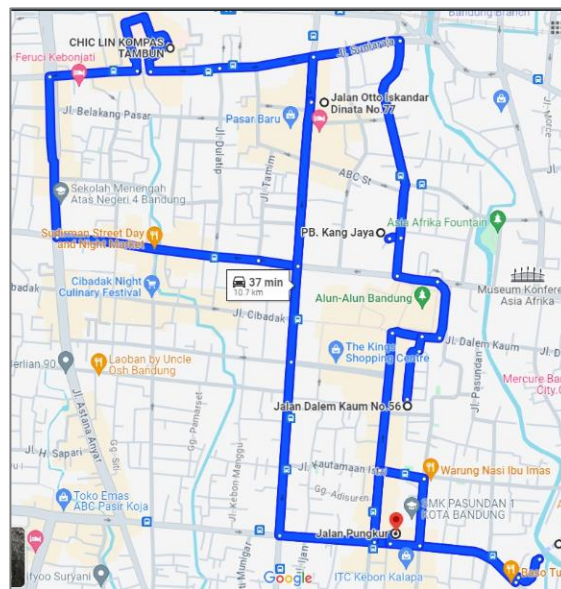
Lokasi	<i>Latitude</i>	<i>Longitude</i>	<i>Demand</i>
Depot (0)	-6.919792	107.606014	0
1	-6.918875	107.606014	28
2	-6.918600	107.608079	24
3	-6.921718	107.609349	16
4	-6.925110	107.606014	10
5	-6.923185	107.600138	10
6	-6.915666	107.598867	12

## 4.2 Analisa Data

Pemilihan lokasi yang dilakukan ialah memilih Kantor Cabang Utama Bandung sesuai dengan data historis yang ada. KCU Bandung memiliki kode 40000. Dimana berdasarkan data tersebut menunjukkan bahwa KCU Bandung terletak di Jl. Asia Afrika No. 49 dengan posisi letak *longitude longitude* adalah -6.919792, 107.606014. Pada hal ini KCU Bandung dijadikan sebagai *node A. A.*

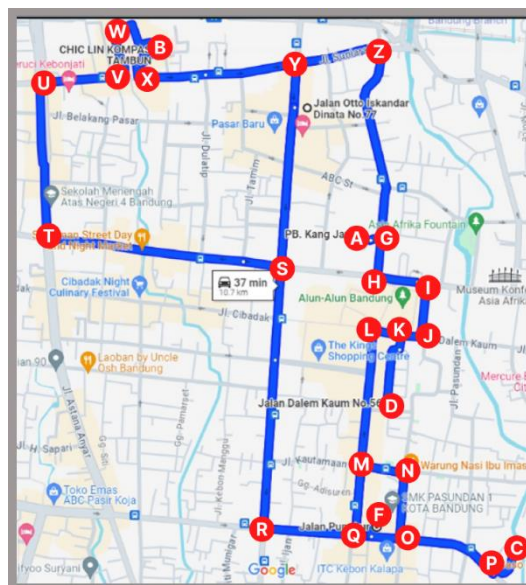
Sementara itu, *node* B-Z merupakan *node* lainnya yang terdiri dari informasi *longitude*, *latitude*, dan alamatnya. Data ini diperoleh secara sintetis untuk mewakili lokasi dan peristiwa historis terkait layanan O-Ranger pada rentang waktu Januari hingga Juni 2023.

Adapun pemilihan *node* berdasarkan gambar visualisasi peta pada Google Maps yang setiap *latitude* dan *longitudenya* dikumpulkan secara manual. Hal itu terjabarkan pada Gambar 4.3.



Gambar 4. 3 Rute Yang Ditentukan Google Maps

Sehingga menjadi seperti pada Gambar 4.4.



Gambar 4. 4 Rute Yang Diberi Label

Kumpulan titik latitude dan longitude akan dibuat menjadi bentuk dataframe. Sehingga *dataframe* node adalah sebagai Gambar 4.5.

	Nodes	Latitude_Node	Longitude_Node
0	A	-6.919792	107.606014
1	B	-6.915593	107.601029
2	C	-6.927422	107.610256
3	D	-6.924132	107.606715
4	E	-6.916932	107.604728
5	F	-6.927167	107.606442
6	G	-6.920150	107.606521
7	H	-6.921020	107.606449
8	I	-6.921194	107.607429
9	J	-6.922485	107.607501
10	K	-6.922475	107.607068
11	L	-6.922382	107.606356
12	M	-6.925661	107.606077
13	N	-6.925884	107.607131
14	O	-6.927419	107.607005
15	P	-6.928341	107.609311
16	Q	-6.927378	107.605908
17	R	-6.927252	107.603658
18	S	-6.920807	107.604100
19	T	-6.920128	107.598305
20	U	-6.916293	107.598179
21	V	-6.916175	107.600292
22	W	-6.914936	107.600453
23	X	-6.916161	107.600585
24	Y	-6.915815	107.604469
25	Z	-6.915490	107.606513

Gambar 4. 5 Dataframe *latitude* dan *longitude* titik lokasi

Berdasarkan hasil pemilihan lokasi yang digunakan, lokasi tersebut dibatasi menggunakan radius dari titik pusat. Adapun langkah-langkah pengerjaan penentuan batas lokasi adalah sebagai berikut.

1. Menentukan titik pusat

Titik pusat yang digunakan adalah KCU Bandung atau depot yang berkoordinat *latitude* -6.919792 dan koordinat *longitude* 107.606014. Dengan demikian, Node A dapat digunakan sebagai titik pusat lokasi untuk batasan penelitian.

2. Menghitung jarak

Untuk menentukan jarak antara dua titik, digunakan rumus Haversine. Rumus Haversine merupakan salah satu cara mencari jarak antara dua titik dengan memperhatikan bahwa bumi bukanlah bidang datar melainkan

merupakan bidang yang mempunyai derajat kelengkungan (Al Amin & Wahyudiyono, 2021).

Adapun formula haversine adalah sebagai berikut.

$$\begin{aligned}\Delta lat &= y = lat_2 - lat_1 \\ \Delta long &= long_2 - long_1 \\ x &= \Delta long \cdot \cos\left(\frac{(lat_1 + lat_2)}{2}\right) \\ d &= \sqrt{(x \cdot x + y \cdot y)} \cdot R\end{aligned}$$

Dimana :

R	= jari-jari bumi sebesar 6371(km)
$\Delta lat$	= besaran perubahan <i>latitude</i>
$\Delta long$	= besaran perubahan <i>longitude</i>
x	= <i>Longitude</i> (Bujur)
y	= besaran perubahan <i>latitude</i>
d	= jarak (km)
1 derajat	= 0.0174532925 radian

Berikut adalah contoh perhitungan untuk menemukan jarak pencarian sebenarnya menggunakan formula haversine. Berikut adalah perhitungan pencarian jarak sebenarnya dengan menggunakan contoh node antara R ke Q dengan formula haversine. *Latitude* 1 dan *longitude* 1 adalah koordinat R. Sedangkan *Latitude* 2 dan *longitude* 2 adalah koordinat Q pada Tabel 4.2.

**a. Radian**

$$\begin{aligned}\text{Latitude 1} &= - 0.927251 \times 0.0174532925 \\ &= - 0.12090333806206971 \text{ Radian} \\ \text{Longitude 1} &= 107.603658 \times 0.0174532925 \\ &= 1.8780381192899365 \text{ Radian} \\ \text{Latitude 2} &= - 6.927377 \times 0.0174532925 \\ &= - 0.12090553717692723 \text{ Radian} \\ \text{Longitude 2} &= 107.605908 \times 0.0174532925 \\ &= 1.8780773891981064 \text{ Radian}\end{aligned}$$

**b. Menyelisihkan *Latitude* 1 dengan *Latitude* 2 yang telah diubah menjadi radian**

$$\begin{aligned} y &= \text{Latitude } 2 - \text{Latitude } 1 \\ &= -0.12090553717692723 - (-0.12090333806206971) \\ &= -2.1991148575206676 \end{aligned}$$

**c. Menyelisihkan *Longitude* 1 dengan *Longitude* 2 yang telah diubah menjadi radian**

$$\begin{aligned} \Delta \text{long} &= \text{Longitude } 2 - \text{Longitude } 1 \\ &= 1.8780773891981064 - 1.8780381192899365 \\ &= 3.9269908169936585 \end{aligned}$$

**d. Mencari nilai x**

$$\begin{aligned} x &= (\Delta \text{long}) \cdot \cos((\text{lat1} + \text{lat2})/2) \\ &= 3.9269908169936585 \cdot \cos \\ &\quad (-0.12090333806206971 + (-0.12090553717692723))/2) \\ &= 3.9269908169936585 \cdot 0.9926999575824607 \\ &= 3.898323617456317 \end{aligned}$$

**e. Mencari nilai d**

$$\begin{aligned} d &= \sqrt{(x \cdot x + y \cdot y)} \cdot R \\ &= \sqrt{(x^2 + y^2)} \cdot R \\ &= \sqrt{(3.898323617456317^2 + (-2.1991148575206676)^2)} \cdot 6371 \\ &= \sqrt{1.5245288087983388} \cdot 6371 \\ &= 0.248757064308297 \text{ km} \end{aligned}$$

Perhitungan mencari jarak lainnya juga dilakukan dengan hal perhitungan yang sama.

### 3. Mengitung Radius dengan Euclidean Distance

Euclidean Distance digunakan untuk mengukur tingkat kemiripan jarak antara data dengan rumus Euclidean. Jarak Euclidean, yaitu perbedaan antara dua titik dalam ruang yang digunakan sebagai landasan dalam rotasi pendulum, memberikan informasi kepada pengguna tentang lokasi terdekat

dengan membandingkan jarak lokasi (Marcelina & Yulianti, 2020). Euclidean ini berkaitan dengan Teorema Pythagoras. Adapun penerapan rumus Euclidean sebagai berikut.

$$dist(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Atau

$$h(n) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$x_i$  (*latitude*) = garis lintang mengarah dari khatulistiwa (0) ke kutub selatan, atau khatulistiwa ke kutub utara (sudut 0-90 dan 0 -90).

$y_i$  (*longitude*) = garis bujur adalah garis horizontal seperti dari khatulistiwa. Sudut 0 (Greenwich) ke arah Hawaii adalah 0-180, sedangkan kebalikannya dari 0 ke -180.

$i$  = index dari atribut

$n$  = jumlah data

Berikut adalah contoh cara menghitung nilai  $h(n)$ . Diketahui goal *node* atau tujuan akhir yang dituju adalah *node* pusat (Pusat) yaitu latitude - 6.9214801293128385 dan longitude 107.60494200542986.

$$\begin{aligned} h(R, \text{Pusat}) &= \sqrt{(x_P - x_R)^2 + (y_{\text{Pusat}} - y_{\text{pusat}})^2} \\ &= \sqrt{(-6.928341 - (-6.921480))^2 + (107.609311 - 107.604942)^2} \\ &= \sqrt{(-0.006861)^2 + (0.004369)^2} \\ &= \sqrt{0.0000470733 + 0.0000190882} \\ &= \sqrt{0.0000661615} \\ &= 0.008133972 \\ &= 0.00813 \end{aligned}$$

Perhitungan mencari nilai  $h(n)$  lainnya juga dilakukan dengan hal perhitungan yang sama.



#### 4. Menentukan Radius

Selanjutnya adalah penentuan radius paling jauh untuk mencakup seluruh node lainnya sebagai lokasi *pickup*. Radius adalah jarak maksimum dari titik pusat ke node mana pun.

Formula radius adalah sebagai berikut.

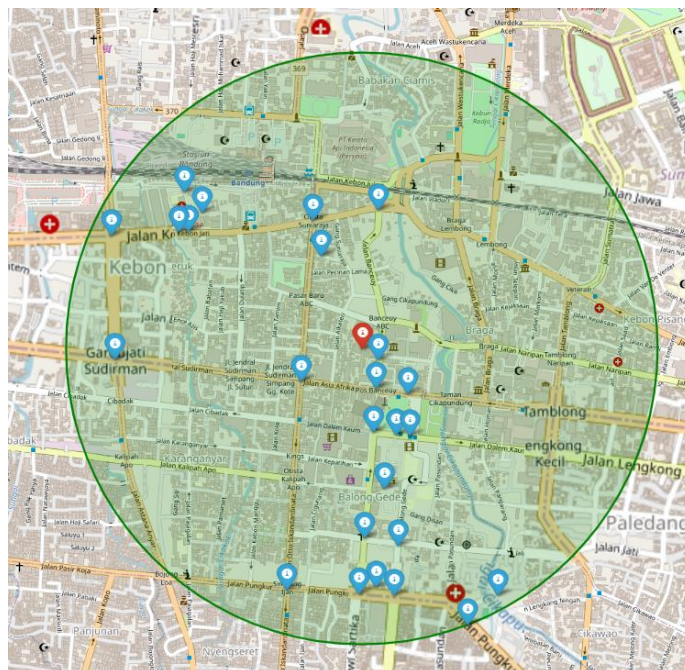
$$radius = \max (d_1, d_2, d_3, \dots, d_n)$$

Dimana:

$d_i$  adalah jarak dari titik pusat ke titik ke-i (menggunakan rumus haversine)

Dengan demikian, radius dihitung sebagai jarak maksimum dari titik median ke semua titik lainnya menggunakan rumus Haversine.

Adapun rentang terjauh yang sudah terhitung untuk menemukan radius adalah sejauh 1.0178281801843392 kilometer.



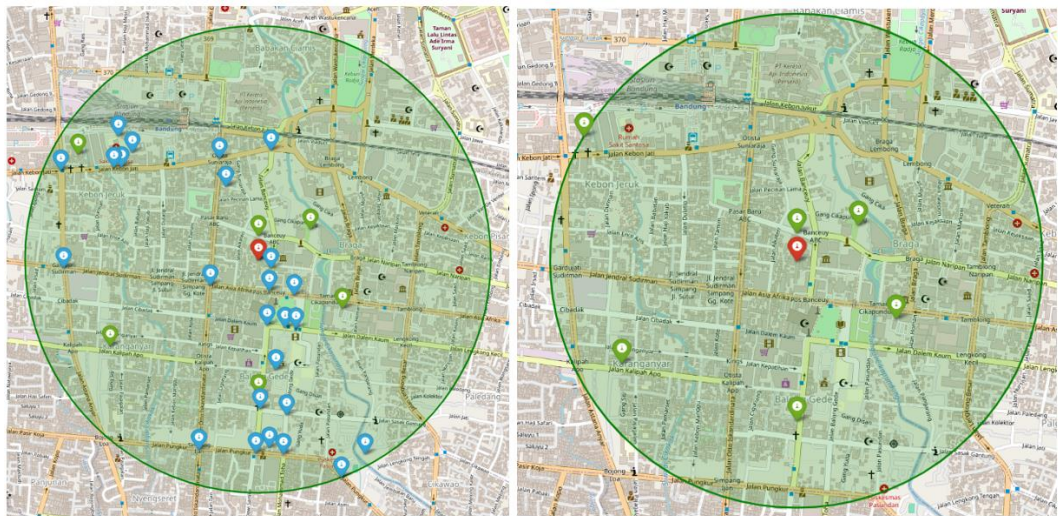
Gambar 4. 6 Radius lokasi yang digunakan

Dari Gambar 4.6, berdasarkan hasil keseluruhan, maka titik lokasi yang digunakan adalah KCU Bandung sebagai depot dan 25 lokasi lainnya merupakan lokasi konsumen. Radius lokasi yang digunakan adalah sepanjang 1.017 km dari pusat lokasi dengan koordinat latitude -6.920806, dan longitude 107.604099). Diluar radius tersebut, lokasi diabaikan.

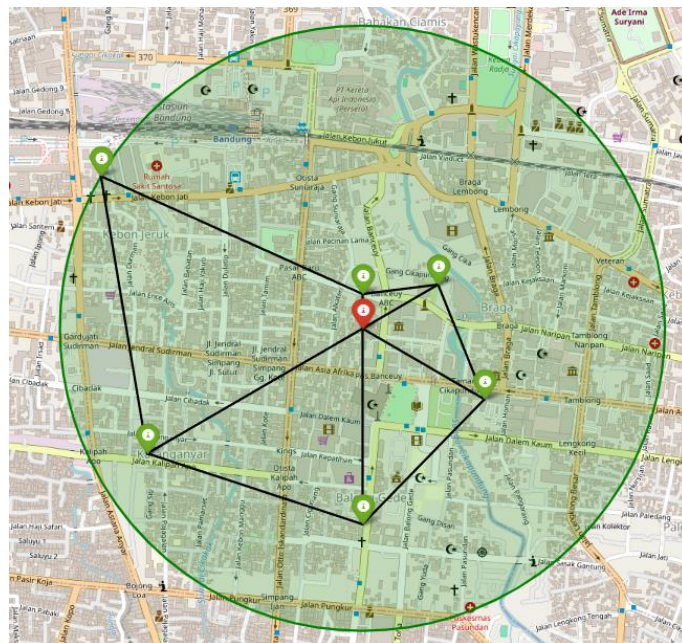


Selain daripada data yang sudah ditentukan, Adapun data lokasi lainnya yang digunakan sebagai untuk mempermudah analisis dalam mengetahui kinerja dari *Linear Programming*. Data yang digunakan adalah data buatan yang wilayahnya mencakup paling maksimal adalah sepanjang 1.017 km dari pusat lokasi.

Titik akan dibentuk hexagon dengan 7 titik. Satu titik merupakan pusat dan 6 lainnya adalah titik sekeliling pusat. Dengan membuat kode acak untuk menambahkan titik baru adalah dengan menggunakan pustaka numpy pada Tabel 4.3.



Gambar 4. 7 Lokasi tambahan berwarna hijau



Gambar 4. 8 Garis Hubung Satu Titik ke Titik Lain

### 4.3 Variabel Keputusan, Fungsi Tujuan, *Constraint* (Batasan)

Pada setiap kendaraan, proses *pickup* dilakukan sebanyak satu kali. Yaitu dari depot ke setiap konsumen, kemudian kembali lagi ke depot (Kristina et al., 2020). Maka dalam hal ini, rute yang digunakan untuk pemodelan adalah depot  $\rightarrow$  konsumen  $\rightarrow$  depot. Kemudian untuk kapasitas digunakan total beban maksimal kapasitas kendaraan dalam satuan kilogram. Maka variabel yang digunakan untuk penelitian ini adalah titik lokasi untuk menentukan jarak, dan jumlah beban total yang dapat diangkut kendaraan untuk memaksimalkan kapasitas kendaraan dalam membawa barang.

Batasan yang digunakan dalam kasus untuk optimasi kapasitas kendaraan dan optimasi jarak pada *pickup* adalah sebagai berikut.

#### 4.3.1 Variabel Keputusan

Variabel keputusan dalam masalah *Capacitated Vehicle Routing Problem with Pickup* (CVRP) dengan *Linear Programming* yang dijelaskan di atas adalah variabel biner  $x[i][j][k]$ .

Variabel biner  $x[i][j][k]$  menunjukkan apakah kendaraan  $k$  melakukan perjalanan dari titik  $i$  ke titik  $j$ . Variabel ini bernilai 1 jika kendaraan  $k$  melakukan perjalanan tersebut, dan bernilai 0 jika tidak. Berikut adalah detailnya:

- a.  $i$  dan  $j$  adalah indeks yang mewakili titik-titik (pelanggan dan depot).
- b.  $k$  adalah indeks yang mewakili kendaraan.

Contoh:

- a.  $x[0][1][0]=1$  berarti kendaraan 0 melakukan perjalanan dari depot (titik 0) ke pelanggan 1.
- b.  $x[2][3][1]=0$  berarti kendaraan 1 tidak melakukan perjalanan dari pelanggan 2 ke pelanggan 3.

```
x = [[pulp.LpVariable(f"x_{i}_{j},{k}", cat="Binary") if i
!= j else None for k in range(vehicle_count)]
      for j in range(customer_count)] for i in
range(customer_count)]
```

Pada kode diatas, `pulp.LpVariable(f"x_{i}_{j},{k}", cat="Binary")` mendefinisikan variabel keputusan  $x[i][j][k]$  sebagai variabel biner. Pada `if i != j else None` memastikan bahwa tidak ada variabel yang didefinisikan

untuk perjalanan dari suatu titik ke dirinya sendiri (tidak relevan dalam konteks VRP).

#### 4.3.2 Fungsi Tujuan

Fungsi tujuan dalam konteks optimasi adalah sebuah persamaan matematis yang digunakan untuk mengukur kinerja dari suatu solusi yang diusulkan. Tujuan utama dari fungsi ini adalah untuk mengarahkan proses pencarian solusi menuju hasil yang optimal sesuai dengan kriteria yang telah ditentukan.

Dalam masalah *Capacitated Vehicle Routing Problem* (CVRP), fungsi tujuan umumnya difokuskan pada minimisasi total jarak yang ditempuh oleh semua kendaraan, yang secara langsung berhubungan dengan pengurangan biaya dan peningkatan efisiensi operasional. Berikut adalah kode yang digunakan untuk mendefinisikan fungsi tujuan untuk meminimasi total jarak yang ditempuh.

```
# Definisikan fungsi tujuan untuk meminimalkan total jarak yang
ditempuh
problem += pulp.lpSum(distance_matrix[i][j] * x[i][j][k] if i
!= j else 0
                        for k in range(vc)
                        for j in range(len(nodes_coordinates))
                        for i in range(len(nodes_coordinates)))
```

Variabel  $x_{ij}$  mengartikan jarak antara titik  $i$  dan titik  $j$ . Dimana  $x_{ij} = 1$  menunjukkan bahwa rute dari  $i$  ke  $j$  diambil. Di sini,  $distance[i][j]$  adalah jarak antara titik  $i$  dan  $j$ , dan  $x[i][j][k]$  adalah variabel biner yang bernilai 1 jika kendaraan  $k$  melakukan perjalanan dari titik  $i$  ke titik  $j$ , dan 0 jika tidak. Fungsi tujuan ini menjumlahkan total jarak yang ditempuh oleh semua kendaraan dengan mempertimbangkan semua rute yang mungkin.

#### 4.3.3 Constraint (Batasan)

Batasan dalam masalah ini adalah sebagai berikut:

1. Setiap pelanggan harus dikunjungi sekali oleh satu kendaraan.

```
for j in range(1, customer_count):
    problem += pulp.lpSum(x[i][j][k] if i != j else 0
                          for i in range(customer_count)
                          for k in range(vehicle_count)) == 1
```

Batasan ini memastikan bahwa setiap pelanggan (titik 1 hingga  $\text{customer\_count} - 1$ ) hanya dikunjungi sekali oleh satu kendaraan. Jumlah variabel  $x[i][j][k]$  untuk setiap pelanggan  $j$  harus sama dengan 1.

2. Kendaraan harus berangkat dan depot sekali dan kembali ke depot sekali.

```
for k in range(vehicle_count):
    problem += pulp.lpSum(x[0][j][k] for j in range(1,
customer_count)) == 1
    problem += pulp.lpSum(x[i][0][k] for i in range(1,
customer_count)) == 1
```

Batasan ini memastikan bahwa setiap kendaraan berangkat dari depot (titik 0) sekali dan kembali ke depot sekali. Jumlah variabel  $x[0][j][k]$  dan  $x[i][0][k]$  untuk setiap kendaraan  $k$  harus sama dengan 1.

3. Keseimbangan aliran kendaraan.

```
for k in range(vehicle_count):
    for j in range(customer_count):
        problem += pulp.lpSum(x[i][j][k] if i != j else 0
for i in range(customer_count)) -
pulp.lpSum(x[j][i][k] if i != j else 0
for i in range(customer_count)) == 0
```

Batasan ini memastikan keseimbangan aliran kendaraan, artinya jumlah kendaraan yang masuk ke titik  $j$  harus sama dengan jumlah kendaraan yang keluar dari titik  $j$ .

4. Total permintaan pelanggan yang dilayani oleh kendaraan tidak boleh melebihi kapasitas kendaraan.

```
for k in range(vehicle_count):
    problem += pulp.lpSum(df.demand[j] * x[i][j][k]
if i != j else 0 for i in range(customer_count) for j
in range(1, customer_count)) <= vehicle_capacity
```

Batasan ini memastikan bahwa total permintaan pelanggan yang dilayani oleh kendaraan  $k$  tidak melebihi kapasitas kendaraan tersebut ( $\text{vehicle\_capacity}$ ).

5. Penghilangan subtour.

```
subtours = []
```

```

for i in range(2, customer_count):
    subtours += itertools.combinations(range(1,
customer_count), i)
for s in subtours:
    problem += pulp.lpSum(x[i][j][k] if i != j else 0
for i, j in itertools.permutations(s, 2) for k in
range(vehicle_count)) <= len(s) - 1

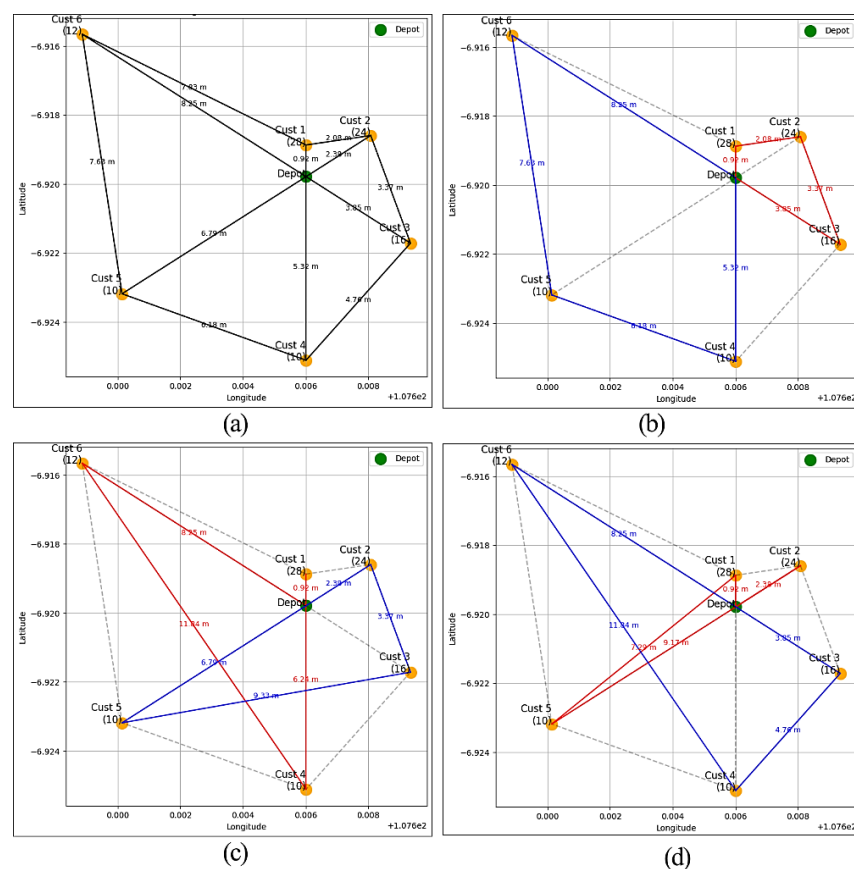
```

Batasan ini memastikan bahwa tidak ada subtour yang terbentuk. Subtour adalah rute yang tidak mencakup depot, yang mengakibatkan kendaraan berputar-putar di antara beberapa pelanggan tanpa kembali ke depot. Batasan ini menggunakan kombinasi dari semua titik pelanggan untuk memastikan bahwa subtour tidak terjadi.

## 4.4 Implementasi *Linear Programming*

### 4.4.1 Grafik Analisis

Berdasarkan Tabel 4.3, dibuatkan rute lain (rute random) yang kemungkinan berpeluang untuk melakukan *pickup* dengan 3 sampel pada Gambar 4.9(b), 4.9(c), dan 4.9(d).



Gambar 4. 9 Bentuk Rute Yang Dihasilkan

Hasil rute-rute tersebut adalah rute yang dibuat tanpa menggunakan *Linear Programming*. Adapun hasil jarak tempuh total yang dihasilkan rute-rute tersebut ada pada Tabel 4.4 berikut.

Tabel 4. 4 Jarak Total Rute Random

Rute Random	Rute	Permintaan	Jarak Total (meter)
1 (4.9(b))	0-1-2-3-0	68	37.60
	0-4-5-6-0	32	
2 (4.9 (c))	0-1-4-6-0	50	49.11
	0-2-3-5-0	50	
3 (4.9 (d))	0-1-5-2-0	62	48.46
	0-3-4-6-0	38	

#### 4.4.2 Perhitungan *Linear Programming*

Pada kasus ini, perhitungan *Linear Programming* menggunakan metode simpleks karena berfungsi untuk mencari nilai optimum (Fikri et al., 2021). Untuk menggunakan metode simpleks sebagai penentu nilai variabel keputusan optimum yang didapat dengan melakukan iterasi secara berulang terhadap tabel simpleks sampai ditemukan nilai yang optimum (Nurmayanti & Sudrajat, 2021). Pada tujuan minimisasi, tabel sudah optimal jika semua nilai pada baris Z sudah negatif atau 0. Jika belum, kembali ke langkah kedua; jika sudah optimal, baca solusi optimal (Suroso & Nugroho, 2023).

Proses perhitungan dilakukan dengan cara mengambil data dari koordinat lokasi dan data permintaan *demand*. Data lokasi dan *demand* tertera pada Tabel 4.3 yang akan kembali dipaparkan pada Tabel 4.5 dibawah.

Tabel 4. 5 Lokasi Dan Permintaan Konsumen

Lokasi	<i>Latitude</i>	<i>Longitude</i>	<i>Demand</i>
Depot (0)	-6.919792	107.606014	0
1	-6.918875	107.606014	28
2	-6.918600	107.608079	24



3	-6.921718	107.609349	16
4	-6.925110	107.606014	10
5	-6.923185	107.600138	10
6	-6.915666	107.598867	12

Untuk mencapai tujuan penelitian, penggunaan algoritma *Linear Programming* yang menggunakan paket PuLP Python dilakukan sebagai proses perhitungan untuk menghasilkan solusi terbaik.

Hasil perhitungan yang didapatkan, didapatkan nilai jarak pada Tabel 4.6 dengan hasil minimasi rute yang terbentuk. Variabel  $x_{ij}$  mengartikan jarak antara titik  $i$  dan titik  $j$ . Dimana  $x_{ij} = 1$  menunjukkan bahwa rute dari  $i$  ke  $j$  diambil.

Tabel 4. 6 Hasil Rute Variabel Keputusan

Variabel	Jarak(m)
$x_{03}$	3.85
$x_{34}$	0.18
$x_{45}$	4.76
$x_{56}$	7.68
$x_{60}$	14.20
$x_{01}$	0.92
$x_{12}$	2.08
$x_{20}$	2.38

Dengan menggabungkan jarak-jarak rute tersebut, persamaan  $Z$  minimasi akhirnya adalah:

$$Z \text{ Minimalkan} = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

$$Z = (3.85x_{03} + 0.18x_{34} + 4.76x_{45} + 7.68x_{56} + 14.20x_{60} + 0.92x_{01} + 2.08x_{12} + 2.38x_{20})$$

Jika di substitusi nilai jaraknya maka perhitungannya adalah:

$$Z = (3.85*1) + (0.18*1) + (4.76*1) + (7.68*1) + (14.20*1) + (0.92*1) + (2.08*1) + (2.38*1)$$

$$Z = 36.05 \text{ meter}$$

Selanjutnya adalah tahap mengenai permintaan (*demand*) konsumen dengan kapasitas kendaraan yang disesuaikan. Dengan formula kendala kapasitas berupa bahwa total permintaan dari lokasi yang dilayani oleh satu kendaraan tidak boleh melebihi kapasitas kendaraan. Dimana maksimal kapasitas akan diterapkan dengan  $Q$ , dengan kendaraan memiliki kapasitas maksimal 60 kg. dengan formula :

$$\sum_j d_j \cdot x_{ij} \leq Q$$

Di mana  $d_j$  adalah permintaan di lokasi  $j$ .

Dari hasil bentuk  $Z$  minimize:

$$Z = (3.85x_{03} + 0.18x_{34} + 4.76x_{45} + 7.68x_{56} + 14.20x_{60} + 0.92x_{01} + 2.08x_{12} + 2.38x_{20})$$

Akan dipastikan bahwa permintaan pada setiap rute tidak melebihi kapasitas kendaraan, maka harus diterapkan kendala kapasitas sesuai dengan jumlah permintaan di setiap lokasi.

Dari data permintaan, total permintaan adalah :

$$28 + 24 + 16 + 10 + 10 + 12 = 100$$

Karena setiap kendaraan memiliki kapasitas maksimum 60 kg, total permintaan (100 kg) tidak dapat diangkut oleh satu kendaraan saja.

$$28x_{01} + 24x_{12} + 16x_{20} + 10x_{34} + 10x_{45} + 12x_{56} \leq 60$$

Untuk menentukan jumlah kendaraan yang diperlukan dalam masalah perutean kendaraan berkapasitas dapat dilakukan dengan perkiraan atau dapat dilakukan dan teknik analisis permintaan dan kapasitas dengan cara menghitung total permintaan.

$$\text{Total Permintaan} = \sum_i Demand_i$$

$$\text{Total Permintaan} = 28 + 24 + 16 + 10 + 10 + 12 = 100 \text{ kg}$$

Kemudian bagi total permintaan dengan kapasitas kendaraan untuk mendapatkan jumlah kendaraan minimal yang diperlukan:



$$\text{Jumlah Kendaraan Minimum} = \left( \frac{\text{Total permintaan}}{\text{Kapasitas kendaraan}} \right)$$

$$\text{Jumlah Kendaraan Minimum} = \left( \frac{100 \text{ kg}}{60 \text{ kg}} \right) = 1,66 \text{ atau dibulatkan menjadi } 2$$

Hasil perhitungan menunjukkan bahwa dengan menggunakan dua kendaraan, permintaan dapat didistribusikan. Untuk memastikan bahwa solusi yang diperoleh adalah yang paling optimal, digunakan pertimbangan peluang berbagai probabilitas rute alternatif dengan memperhatikan kapasitas kedua kendaraan tersebut. Dengan hasil dua kendaraan, dapat dibagi permintaan sebagai Tabel 4.7 berikut.

Tabel 4. 7 Peluang Rute

No Rute	Rute	Permintaan	Jarak	Jarak Total
1	0-2-5-4-6-0	56	37.83	46.98
	0-3-1-0	44	9.15	
2	0-5-3-6-4	48	45.38	50.76
	0-2-1-0	52	5.38	
3	0-1-4-5-6-0	60	29.21	38.82
	0-2-3-0	40	9.60	
4	0-1-6-3-0	56	24.71	52.81
	0-4-2-5-0	44	28.10	
5	0-1-2-0	52	5.38	43.47
	0-6-3-4-5-0	48	38.08	
6	0-6-3-5-4-0	48	41.18	46.57
	0-2-1-0	52	5.38	
7	0-5-6-1-4-0	60	33.80	43.40
	0-3-2-0	40	9.60	
8	0-5-3-4-2-0	60	38.08	47.90
	0-1-6-0	40	17.00	
9	0-5-2-6-4-0	56	42.78	51.93
	0-1-3-0	44	9.15	
10	0-3-6-5-4-0	48	35.08	40.47
	0-1-2-0	52	5.38	

Dari 10 kemungkinan rute *pickup* yang dapat berjalan, setelah dicari dan dianalisis, hasil penggunaan LP dengan mempertimbangkan jarak dan kapasitas kendaraannya, mendapatkan pilihan hasil keputusan berikut.

Kendaraan 1:  $28 + 24 = 52$  kg

Kendaraan 2:  $16 + 10 + 10 + 12 = 48$  kg

Sehingga menghasilkan formula kendala kapasitas :

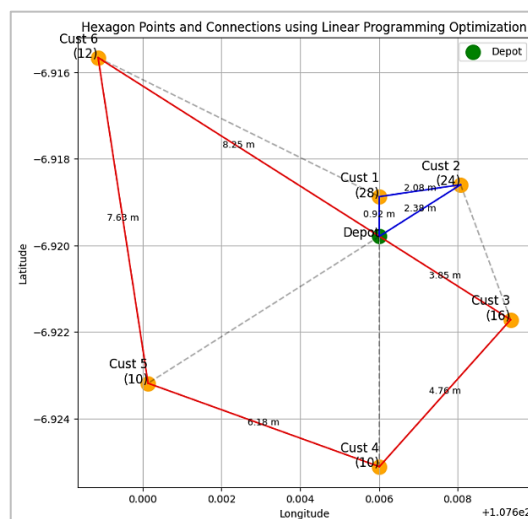
$$28x_{01} + 24x_{12} \leq 60, \text{ untuk kendaraan pertama.}$$

$$16x_{03} + 10x_{34} + 10x_{45} + 12x_{56} \leq 60, \text{ untuk kendaraan kedua.}$$

Total permintaan terbentuk menjadi dua bagian yang masing-masing tidak melebihi kapasitas 60 kg. Oleh karena itu, diperlukan minimal dua kendaraan untuk memenuhi seluruh permintaan tanpa melebihi kapasitas kendaraan.

- Kendaraan 1 dengan menempuh rute  $0 \rightarrow 1 \rightarrow 2 \rightarrow 0$  dengan total jarak 5.38 meter dan membawa permintaan sebesar 52 kg.
- Kendaraan 2 dengan menempuh rute  $0 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 0$  dengan total jarak 30.67 meter dan membawa permintaan sebesar 48 kg.

Hasil Z minimasi dan penyesuaian kapasitas kendaraan yang didukung dengan hasil visualisasi pada Gambar 4.10 dan penjelasan pada Tabel 4.6.



Gambar 4. 10 Hasil rute yang didapatkan

Berdasarkan Gambar 4.10, terdapat hasil pencarian rute yang didapatkan. Dimana total kendaraan yang digunakan sebanyak dua kendaraan dari empat kendaraan yang tersedia. Hal itu juga dibuktikan dengan rute yang terbentuk terbagi menjadi dua rute yang tertera pada warna merah dan warna biru. Hasil rute dipaparkan pada Tabel 4.8.

Tabel 4. 8 Rute Yang Dihasilkan *Linear Programming*

Kendaraan	Warna	Hasil Rute	Permintaan (kg)	Jarak (m)
1	Merah	0→3→4→5→6→0	48	30.67
2	Biru	0→1→2→0	52	5.38
Total			100	36.05

Tabel 4.8 menunjukkan hasil rute yang dihasilkan dari solusi *Linear Programming* (LP). Dalam tabel tersebut, terdapat dua kendaraan yang digunakan untuk melayani seluruh permintaan. Kendaraan pertama (ditandai dengan warna merah) mengikuti rute dari depot ke lokasi 3, kemudian ke lokasi 4, 5, 6, dan kembali ke depot. Kendaraan ini mengangkut total permintaan sebesar 48 kg dan menempuh jarak 30,67 meter. Kendaraan kedua (ditandai dengan warna hijau) mengikuti rute dari depot ke lokasi 1, kemudian ke lokasi 2, dan kembali ke depot. Kendaraan ini mengangkut total permintaan sebesar 52 kg dan menempuh jarak 5,38 meter.

Sehingga jika dibandingkan dari grafik analisis, maka hasil ada pada Tabel 4.9 berikut.

Tabel 4. 9 Perbandingan Hasil Analisis

No Rute	Rute	Permintaan	Jarak Total
LP	0-3-4-5-6-0	48	36.05
	0-1-2-0	52	
1 (4.9(b))	0-1-2-3-0	68	37.60
	0-4-5-6-0	32	
2 (4.9 (c))	0-1-4-6-0	50	49.11
	0-2-3-5-0	50	
3 (4.9 (d))	0-1-5-2-0	62	48.46
	0-3-4-6-0	38	

Dari Tabel 4.9, perhitungan menggunakan LP menunjukkan bahwa solusi yang dihasilkan adalah yang paling optimal dibandingkan dengan rute-rute alternatif lainnya, baik dari segi total jarak tempuh maupun efisiensi distribusi beban.

## 4.5 Evaluasi Model Pada Grafik Analisis

### 4.5.1 Perbandingan Jarak

Analisis komparatif atau perbandingan jarak untuk mengetahui apakah hasil model lebih baik daripada kemungkinan hasil lainnya. Jika dirangkum, jarak rute-rute random dan rute yang dihasilkan oleh *Linear Programming* ada pada Tabel 4.10 berikut.

Tabel 4. 10 Jarak Yang Dihasilkan Berbagai Rute

Rute	Jarak (m)
<b><i>Linear Programming</i></b>	<b>36.05</b>
Random 1	37.60
Random 2	49.11
Random 3	48.46

Berdasarkan hasil jarak total pada Tabel 4.10, dapat diketahui bahwa rute yang terbentuk menggunakan *Linear Programming* menghasilkan jarak yang paling pendek yaitu 36.05 meter atau hemat jarak sekitar 19.99% dari rata-rata jarak rute random. Hal ini menunjukkan bahwa rute yang dihasilkan menggunakan *Linear Programming* menghasilkan jarak paling pendek dibandingkan dengan jarak di rute-rute yang lainnya.

### 4.5.2 Optimality Gap

Untuk mengukur seberapa baik atau seberapa dekat sebuah solusi yang ada dengan solusi terbaik, digunakan evaluasi dengan *Optimality Gap*. Adapun formula *Optimality Gap* adalah sebagai berikut (Bernardino & Paías, 2024).

$$\text{Optimality Gap} = \left( \frac{\text{Jarak rute yang dihitung} - \text{Jarak Optimal}}{\text{Jarak Optimal}} \right) \times 100\%$$

Interpretasi persentase *Optimality Gap*:

1. 0%: *Optimality gap* 0% berarti solusi yang ditemukan adalah optimal atau sangat baik mendekati nilai optimal yang sebenarnya.
2. <1%: Sering dianggap sangat baik dalam banyak konteks optimasi yang membutuhkan presisi tinggi.
3. 1-5%: Umumnya dianggap baik dan dapat diterima terutama untuk masalah yang tidak memerlukan solusi yang presisi.
4. 5-10%: Masih dapat diterima dalam beberapa kasus tergantung pada kompleksitas masalah dan kebutuhan praktisnya.
5. >10%: Menunjukkan bahwa solusi yang ditemukan jauh dari nilai optimal dan memerlukan perbaikan atau evaluasi lebih lanjut.

Setelah ditemukan jarak total rute random dan jarak total yang dibentuk oleh LP, *Optimality Gap* dapat dihitung dan dihasilkan yang tertera pada Tabel 4.11.

Tabel 4. 11 Hasil *Optimality Gap*

Rute	Jarak (m)	<i>Optimality Gap</i>
<b><i>Linear Programming</i></b>	<b>36.05</b>	<b>0%</b>
Random 1	37.60	4.28%
Random 2	49.11	36.22%
Random 3	48.46	34.42%

Berdasarkan Tabel 4.11, rute yang dihasilkan menggunakan LP memiliki *Optimality Gap* sama dengan 0%, atau hasil optimal yang dihasilkan sangat baik. Sedangkan rute lain memiliki *Optimality Gap* yang cukup jauh dari 0%. Maka hasil yang telah didapatkan menggunakan *Linear Programming* mampu menghasilkan solusi optimal yang baik.

#### 4.6 Implementasi Aplikasi

Terdapat perbedaan fungsi antara analisis menggunakan grafik, seperti yang dijelaskan pada bagian 4.4.1, dan analisis pada aplikasi. Pada grafik analisis, perhitungan jarak didasarkan pada garis lurus antara dua titik. Ini berarti jarak dihitung tanpa mempertimbangkan bentuk dan kondisi sebenarnya dari jalan yang ada. Sebaliknya, pada aplikasi, rute yang dibentuk diimplementasikan pada peta

sebenarnya. Di sini, rute yang dihasilkan adalah rute jalan yang disesuaikan berdasarkan arah dan kondisi jalan yang sesungguhnya, bukan dalam bentuk garis lurus. Hal ini memungkinkan rute yang dihasilkan dalam aplikasi lebih realistis dan dapat diaplikasikan langsung di lapangan, memberikan hasil yang lebih akurat dan relevan untuk keperluan operasional sehari-hari.

Untuk menangani perbedaan ini, digunakan API key pada aplikasi. API key memungkinkan aplikasi untuk mengakses layanan peta yang menyediakan data jalan secara real-time. Dengan menggunakan API key, aplikasi dapat mengkalkulasi rute yang akurat berdasarkan kondisi jalan yang sebenarnya, termasuk arah jalan, kemacetan, dan rute alternatif yang tersedia.

#### 4.6.1 Penggunaan Google API Key

Adapun fungsi untuk menangani perbedaan antara analisa menggunakan grafik dan aplikasi adalah dengan menggunakan Google Maps API.

##### 1. Menghitung Jarak Menggunakan Google Maps API.

```
def get_distance_from_google_maps(start_coords,
end_coords, api_key):
    url =
    f"https://maps.googleapis.com/maps/api/directions/json?ori
gin={start_coords[0]},{start_coords[1]}&destination={end_c
oords[0]},{end_coords[1]}&key={api_key}"
    response = requests.get(url)
    if response.status_code == 200:
        directions = response.json()
        if directions['status'] == 'OK':
            return
            directions['routes'][0]['legs'][0]['distance']['value']
        else:
            print(f"Error in directions response:
{directions['status']}")
    else:
        print(f"HTTP Error: {response.status_code}")
    return None
```

Fungsi `get_distance_from_google_maps` menghitung jarak antara dua koordinat berdasarkan data real-time dari Google Maps. Ini memastikan bahwa jarak yang dihitung sesuai dengan kondisi jalan yang sebenarnya.

## 2. Fungsi Menghitung Matriks Jarak Menggunakan Google Maps API

```
def _distance_matrix_from_google_maps(df, api_key):
    distance_result = np.zeros((len(df), len(df)))
    for i in range(len(df)):
        for j in range(len(df)):
            if i != j:
                distance = get_distance_from_google_maps(
                    (df.latitude[i], df.longitude[i]),
                    (df.latitude[j], df.longitude[j]),
                    api_key
                )
                if distance is not None:
                    distance_result[i][j] = distance /
1000.0 # Convert to kilometers
                else:
                    distance_result[i][j] = float('inf')
    return distance_result
```

Fungsi `_distance_matrix_from_google_maps` membangun matriks jarak antar node dengan menggunakan data dari Google Maps. Ini memungkinkan aplikasi untuk menghitung jarak antar semua node berdasarkan kondisi jalan yang sebenarnya.

## 3. Fungsi Mendapatkan Rute Menggunakan Google Maps API

```
def get_route_from_google_maps(start_coords, end_coords,
api_key):
    url =
f"https://maps.googleapis.com/maps/api/directions/json?ori
gin={start_coords[0]},{start_coords[1]}&destination={end_c
oords[0]},{end_coords[1]}&key={api_key}"
    response = requests.get(url)
    if response.status_code == 200:
        directions = response.json()
        if directions['status'] == 'OK':
            return
directions['routes'][0]['overview_polyline']['points']
        else:
            print(f"Error in directions response:
{directions['status']}")
    else:
        print(f"HTTP Error: {response.status_code}")
    return None
```

Fungsi `get_route_from_google_maps` mendapatkan rute (polyline) antara dua koordinat dengan mempertimbangkan kondisi jalan yang sebenarnya, seperti arah jalan dan rute alternatif.

Dengan menggunakan Google Maps API, aplikasi dapat menghitung jarak dan rute yang lebih akurat dan realistis berdasarkan kondisi jalan yang sebenarnya. Hal ini memungkinkan solusi rute yang dihasilkan untuk dapat diterapkan secara langsung di lapangan, meningkatkan efisiensi dan keandalan dalam manajemen rute dan distribusi.

#### 4.6.2 *Prototipe*

Pada tahap perancangan antarmuka, akan digunakan metode *prototyping*. *Prototyping* merupakan suatu proses yang digunakan untuk membantu pengembangan perangkat lunak dalam membentuk model perangkat lunak. *Prototyping* ini merupakan versi awal dari suatu tahap sistem perangkat lunak yang digunakan untuk menyajikan suatu ide, melakukan eksperimen terhadap suatu desain, menemukan sebanyak mungkin masalah, dan menemukan solusi untuk menyelesaikan masalah tersebut (Fridayanthie et al., 2021).

Adapun hasil rancangan antarmuka yang dibuat adalah sebagai berikut.

##### 1. *Prototype halaman index*

Gambar 4. 11 *Prototype halaman index*

Pada Gambar 4.11, halaman index aplikasi CVRPP merupakan antarmuka utama yang digunakan untuk memasukkan koordinat lokasi dan permintaan untuk setiap lokasi. Halaman ini terdiri dari beberapa elemen penting sebagai berikut:



1. Judul Halaman.

Terdapat judul halaman yang jelas "Inputkan Koordinat Lokasi dan Demand" untuk memandu pengguna dalam mengisi data.

2. Form Input Lokasi.

Nama Lokasi: Kolom input untuk memasukkan nama lokasi.

Latitude: Kolom input untuk memasukkan koordinat latitude lokasi.

Longitude: Kolom input untuk memasukkan koordinat longitude lokasi.

Demand (kg): Kolom input untuk memasukkan permintaan dalam kilogram di lokasi tersebut.

3. Tombol Aksi.

Hapus: Tombol untuk menghapus data yang telah dimasukkan di form input.

Reset: Tombol untuk mereset seluruh form input sehingga kosong kembali.

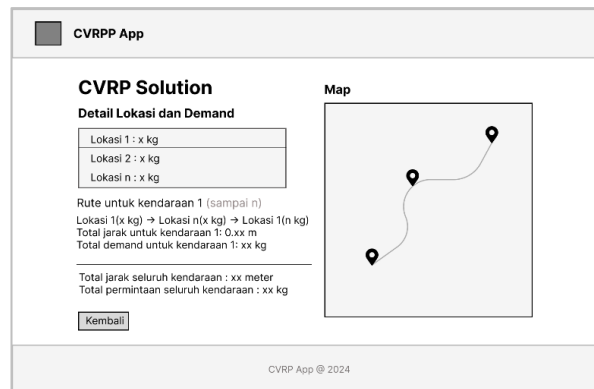
Tambah Lokasi: Tombol untuk menambahkan lokasi baru beserta koordinat dan permintaan.

Selesai: Tombol untuk menyelesaikan input data dan melanjutkan ke halaman hasil.

4. Peta.

Terdapat peta interaktif yang berfungsi untuk memilih lokasi yang akan diinputkan dan form secara otomatis akan langsung terisi. Pada gambar ini, terdapat dua pin yang menunjukkan "Lokasi 1" dan "Depot".

## 2. *Prototype* halaman *hasil*



Gambar 4. 12 *Prototype* halaman hasil

Pada Gambar 4.12, halaman hasil aplikasi CVRPP menampilkan solusi rute optimal berdasarkan input koordinat lokasi dan permintaan. Halaman ini terdiri dari beberapa elemen penting sebagai berikut:

### 1. Judul Halaman.

Terdapat judul "CVRP Solution" untuk menunjukkan bahwa halaman ini berisi hasil solusi dari masalah CVRP.

### 2. Detail Lokasi dan Demand.

Menampilkan daftar lokasi beserta permintaan masing-masing lokasi. Contoh format yang ditampilkan adalah:

Lokasi 1: x kg

Lokasi 2: x kg

Lokasi n: x kg

### 3. Rute untuk Kendaraan.

Menampilkan rute optimal yang harus dilalui oleh setiap kendaraan.

Contoh format yang ditampilkan adalah:

Lokasi 1(x kg) -> Lokasi n(x kg) -> Lokasi 1(n kg)

Total jarak untuk kendaraan tersebut dalam kilometer.

Total permintaan untuk kendaraan tersebut dalam kilogram.

### 4. Total Jarak dan Permintaan Seluruh Kendaraan.

Menampilkan total jarak seluruh kendaraan dalam meter.

Menampilkan total permintaan seluruh kendaraan dalam kilogram.

#### 5. Peta.

Terdapat peta yang menampilkan rute optimal yang harus dilalui oleh kendaraan untuk memenuhi seluruh permintaan di berbagai lokasi.

#### 6. Tombol Kembali.

Tombol untuk kembali ke halaman index guna menginput ulang data jika diperlukan.

### 4.6.3 Hasil Antar Muka

Untuk memberikan hasil visualisasi yang lebih baik, perancangan sistem untuk pencarian rute terbaik dan memaksimalkan kapasitas kendaraan dibuat dengan menggunakan *framework* Flask. Flask adalah kerangka kerja web yang ditulis dalam bahasa Python. Flask juga menyediakan pustaka dan koleksi kode program yang dapat digunakan untuk membangun situs web (Novindri & Saian, 2022).

Berikut adalah deskripsi dari aplikasi CVRPP yang telah dibangun berdasarkan prototipe dan hasil akhir yang diperoleh..

#### 1. Halaman Index

Gambar 4. 13 Halaman Index Aplikasi

Pada Gambar 4.13, halaman index dari aplikasi CVRPP adalah antarmuka utama yang digunakan untuk memasukkan data lokasi dan permintaan (demand) dari setiap lokasi. Halaman index digunakan untuk menginputkan nama lokasi, koordinat lokasi, dan berat barang permintaan konsumen.

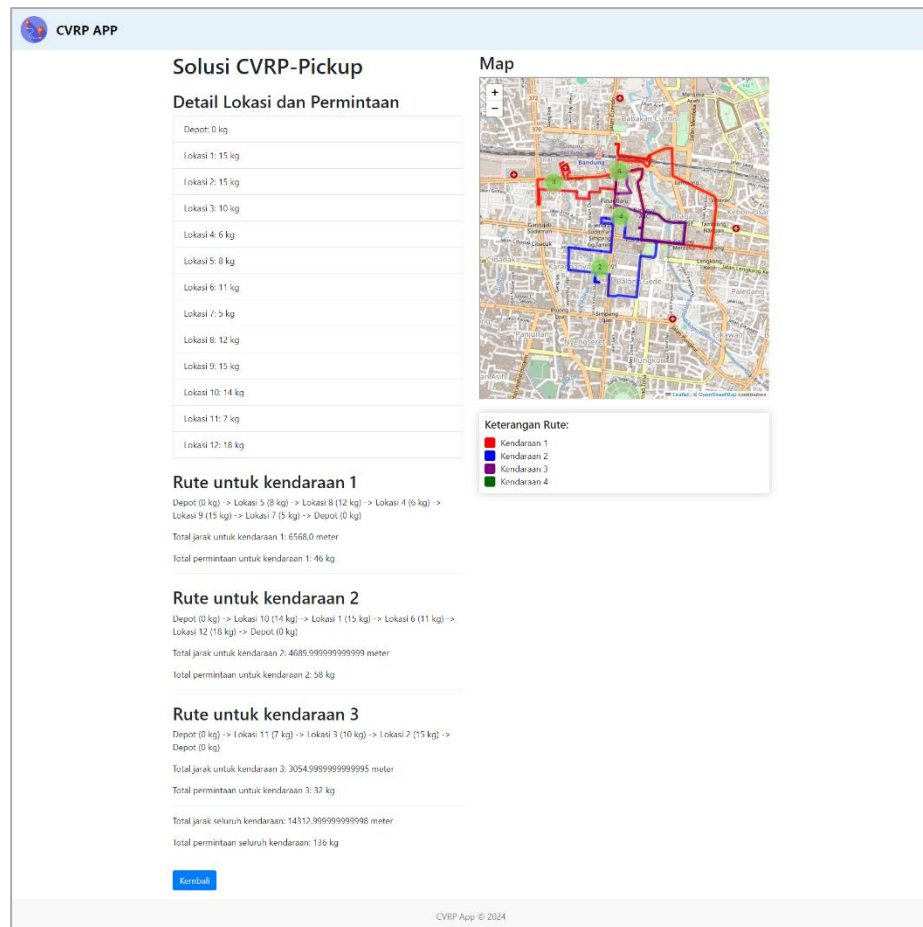
## 2. Halaman mengisi Index

Inputan untuk aplikasi dapat menggunakan data lokasi secara bebas dengan Batasan lokasi sejauh 1 km dari depot sebagai area pickup. Adapun data terbaru yang digunakan sebanyak 12 lokasi pada Tabel 4.12.

Tabel 4. 12 Data Lokasi dan Demand Untuk Aplikasi

Lokasi	<i>Latitude</i>	<i>Longitude</i>	<i>Demand</i>
Depot (0)	-6.919792	107.606014	0
1	-6.915593	107.601029	15
2	-6.927422	107.610256	15
3	-6.915885	107.604966	10
4	-6.916197	107.600073	6
5	-6.916580	107.603893	8
6	-6.922024	107.603206	11
7	-6.913229	107.604022	5
8	-6.918247	107.598056	12
9	-6.915175	107.600288	15
10	-6.920486	107.604966	14
11	-6.916954	107.605867	7
12	-6.924538	107.603077	18

### 3. Halaman Hasil (*result*)



Gambar 4. 14 Halaman Hasil Aplikasi

Pada Gambar 4.14, halaman hasil dari aplikasi CVRPP menampilkan solusi rute optimal berdasarkan data lokasi dan permintaan yang telah diinput pada halaman index. Hasil yang ditampilkan adalah report rute masing-masing kendaraan beserta total permintaan dan jaraknya, dan juga map yang dibentuk menggunakan OpenStreetMap.

## 4.7 Evaluasi Model Pada Aplikasi

Tentunya evaluasi tetap dilakukan pada aplikasi. Evaluasi yang digunakan sama seperti pada evaluasi grafik yaitu perbandingan jarak rute dan *Optimality Gap*.

### 4.7.1 Perbandingan Jarak Pada Aplikasi

Berdasarkan inputan dari Tabel 4.12 pada halaman index aplikasi, terdapat hasil jarak antara rute yang kemungkinan bisa digunakan dengan rute yang dihasilkan ada pada Tabel 4.13.

Tabel 4. 13 Jarak Rute Aplikasi

Rute	Hasil Rute	Permintaan (kg)	Jarak (m)
Random	0→1→4→7→11→12→0	51	9348.0
	0→3→6→8→0	33	5038.0
	0→2→5→9→10→0	52	6641.0
Total Jarak		136	21027.0
<i>Linear Programming</i>	0→5→8→5→9→7→0	22	4137.0
	0→10→1→6→12→0	56	5168.0
	0→11→3→2→0	58	3054.0
Total Jarak		136	<b>14312.9</b>

Tabel 4.13 menunjukkan bahwa hasil evaluasi yang dilakukan, total jarak yang dicapai menggunakan metode LP menunjukkan efisiensi yang signifikan dibandingkan dengan rute random. Metode LP menghasilkan total jarak terpendek sebesar 14,312.9 meter. Ini berarti bahwa dengan menggunakan LP, terdapat penghematan jarak sebesar 31.92% dibandingkan dengan jarak yang dihasilkan oleh rute random. Penghematan ini menunjukkan bahwa metode LP tidak hanya lebih efisien dalam hal jarak tempuh, tetapi juga berpotensi mengurangi biaya operasional dan waktu perjalanan, sehingga meningkatkan keseluruhan efektivitas distribusi dan logistik.

Selain itu, tabel juga memperlihatkan bahwa metode LP mampu memenuhi seluruh permintaan dengan total permintaan sebesar 136 kg. Kendaraan pertama dalam rute LP melayani permintaan sebesar 22 kg, kendaraan kedua melayani permintaan sebesar 56 kg, dan kendaraan ketiga melayani permintaan sebesar 58 kg. Setiap kendaraan beroperasi dengan mempertimbangkan kapasitas maksimal yang dapat diangkut, memastikan bahwa tidak ada kendaraan yang melebihi kapasitasnya dan seluruh permintaan dapat dilayani secara efektif. Hal ini menunjukkan bahwa metode LP tidak hanya efisien dalam mengurangi jarak tempuh, tetapi juga efektif dalam mengelola kapasitas kendaraan untuk memenuhi permintaan distribusi.

#### 4.7.2 *Optimality Gap* Pada Aplikasi

Berdasarkan hasil jarak dari Tabel 4.13 pada halaman index aplikasi, terdapat hasil jarak antara rute yang kemungkinan bisa digunakan dengan rute yang dihasilkan ada pada Tabel 4.14.

Tabel 4. 14 *Optimality Gap* Rute Aplikasi

Rute	Jarak (m)	<i>Optimality Gap</i>
<b><i>Linear Programming</i></b>	<b>14312.9</b>	<b>0%</b>
Random 1	21027.0	46.91%

Pada Tabel 4.14, berdasarkan hasil evaluasi, rute yang dibentuk oleh metode *Linear Programming* (LP) memiliki *Optimality Gap* terbaik, yaitu 0%. Ini menandakan bahwa solusi yang dihasilkan oleh metode LP adalah solusi optimal yang sesungguhnya, tanpa adanya deviasi dari solusi terbaik yang mungkin dicapai.

Dengan demikian, pencarian rute dan pemaksimalan kapasitas kendaraan yang dilakukan menggunakan metode *Linear Programming* yang telah diimplementasikan dalam aplikasi ini terbukti mampu menghasilkan solusi yang optimal.

## BAB V

### KESIMPULAN

Dapat disimpulkan bahwa penelitian ini bertujuan untuk melakukan pembuatan model optimasi jarak dan kapasitas kendaraan yang dapat meningkatkan efisiensi operasional dalam pengelolaan rute logistik menggunakan model *Capacitated Vehicle Routing Problem with Pickup* (CVRPP) yang dioptimalkan dengan *Linear Programming* (LP). Proses perhitungan optimasi dilakukan menggunakan paket PuLP Python dengan bahasa pemrograman Python. Variabel yang digunakan adalah jarak antar lokasi dan kapasitas kendaraan yang mempengaruhi rute dan efisiensi operasional kendaraan dalam pelaksanaan pickup.

Berdasarkan hasil penelitian, didapatkan bahwa dari empat kendaraan yang tersedia, hanya dua kendaraan yang digunakan. Hal ini menunjukkan bahwa penerapan LP pada kapasitas kendaraan dapat memberikan hasil berupa penggunaan kendaraan yang lebih efektif atau diminimalisir. Selain itu, perbandingan jarak rute menunjukkan bahwa rute yang dibentuk menggunakan LP memiliki jarak terpendek di antara rute-rute lainnya. Hal ini didukung dengan hasil *Optimality Gap* untuk rute LP yang sebesar 0%, yang berarti solusi yang ditemukan adalah optimal atau sangat baik.

Penelitian ini menunjukkan bahwa model CVRPP menggunakan LP mampu mendapatkan hasil rute terbaik dan dapat memaksimalkan kapasitas kendaraan. Implementasi model ini dalam bentuk aplikasi menggunakan *framework* Flask juga menunjukkan hasil yang konsisten, di mana rute yang terbentuk oleh *Linear Programming* adalah rute terpendek dan memiliki nilai *Optimality Gap* 0%.

Dengan demikian, metode LP tidak hanya efektif dalam memastikan total jarak tempuh yang minimal, tetapi juga dalam memaksimalkan penggunaan kapasitas kendaraan secara efisien. Implementasi *Linear Programming* dalam aplikasi ini terbukti memberikan hasil yang sangat baik, dengan solusi yang optimal, dan dapat secara signifikan meningkatkan efisiensi operasional serta mengurangi biaya dalam manajemen logistik dan distribusi.



## **BAB VI**

### **SARAN**

Penelitian selanjutnya dapat mempertimbangkan untuk menambahkan variabel tambahan lainnya, seperti kondisi jalan raya, waktu perjalanan, konsumsi bahan bakar, dan faktor eksternal lainnya yang dapat mempengaruhi efisiensi rute. Selain itu, pengembangan aplikasi dapat dilakukan lebih lanjut untuk meningkatkan fungsionalitas dan keandalannya. Misalnya, aplikasi dapat diintegrasikan dengan data *real-time* untuk kondisi lalu lintas dan cuaca, serta fitur-fitur tambahan yang memungkinkan penyesuaian dinamis terhadap perubahan situasi di lapangan. Selain itu, integrasi sistem *tracking* GPS akan memberikan pemantauan rute yang lebih akurat dan memungkinkan optimasi rute secara *real-time*. Dengan demikian, penelitian dan pengembangan yang lebih lanjut akan dapat memberikan solusi yang lebih komprehensif dan praktis dalam manajemen rute dan logistik.

## DAFTAR PUSTAKA

- Agarina. (2019). User-Centered Design Method in the Analysis of User Interface Design of the Department of Informatics System's Website. Proceeding International Conference On Information Technology And Business (ICITB). <https://jurnal.darmajaya.ac.id/index.php/icitb/article/view/2098>
- Al Amin, I. H., & Wahyudiyono, W. (2021). Implementasi Metode Haversine Untuk Pencarian Optical Distribution Point. *Jutnal Ilmiah Teknologi Informasi*, 13(1).
- Ary, M. (2022). Optimasi Vehicle Routing Problem Pada Rute Pendistribusian Menggunakan Metode Ant Colony Optimization. *Jurnal Tekni Insentif*, 16(2).
- Bantacut, T., & Fadhil, R. (2018). Penerapan LOGISTIK 4.0 dalam Manajemen Rantai Pasok Beras Perum BULOG: Sebuah Gagasan Awal. *Jurnal Pangan*, 27(2). <https://doi.org/https://doi.org/10.33964/jp.v27i2.371>
- Basriati, S., & Aziza, D. (2017). Penentuan Rute Distribusi pada Multiple Depot Vehicle Routing Problem (MDVRP) Menggunakan Metode Insertion Heuristic (Studi Kasus : Orange Laundry di Kota Pekanbaru). *Jurnal Sains Matematika Dan Statistika*, 3. <https://doi.org/10.24014/jsms.v3i1.4465>
- Bernardino, R., & Paia, A. (2024). The family capacitated vehicle routing problem. *European Journal of Operational Research*, 314. <https://doi.org/10.1016/j.ejor.2023.10.042>
- Binus University. (2019). User Centered Design. Binus University School of Information System. <https://sis.binus.ac.id/2019/05/31/user-centered-design/>
- Chi, J., & He, S. (2023). Pickup capacitated vehicle routing problem with three-dimensional loading constraints: Model and algorithms. *Transportation Research Part E: Logistics and Transportation Review*, 176. <https://doi.org/10.1016/j.tre.2023.103208>
- Claudiu Pop, P., Zelina, I., Lupăș, V., Pop Sitar, C., & Chira, C. (2011). Heuristic Algorithms for Solving the Generalized Vehicle Routing Problem. *International Journal of Computers Communications & Control*, 6(1). <https://univagora.ro/jour/index.php/ijccc/article/view/2210>
- Efendi, Y. (2022). Perancangan Vehicle Routing Problem Menggunakan Algoritma Nearest Neighbor Dan Local Search Guna Optimasi Biaya Distribusi Pada Pt Madubar. <https://space.uin.ac.id/handle/123456789/40110>
- Eviana, A., Fauzan, Abd. C., Harliana, & Putra, F. N. (2022). Komparasi Jarak Euclidean dan Jarak Manhattan untuk Deteksi Covid-19 Melalui Citra CT-Scan Paru-Paru. *Jurnal Sistem Komputer*, 11(2). <https://doi.org/https://doi.org/10.34010/komputika.v11i2.5380>
- Fikri, A. J., Aini, S., Sukandar, R. S., Sfiyanah, I., & Listiasari, D. (2021). Optimalisasi Keuntungan Produksi Makanan Menggunakan Pemrograman Linear Melalui Metode Simpleks. *Jurnal Bayesian*, 1.

- Fridayanthie, E. W., Haryanto, H., & Tsabitah, T. (2021). Penerapan Metode Prototype Pada Perancangan Sistem Informasi Penggajian Karyawan (Persis Gawan) Berbasis Web. *Jurnal Khatulistiwa Informatika*, 9. <https://doi.org/https://dx.doi.org/10.31294/p.v23i2.10998>
- Hadhiatma, A., & Purbo, A. (2017). Vehicle Routing Problem Untuk Distribusi Barang Menggunakan Algoritma Semut. *Prosiding Snatif*.
- Hariadi (Direktur Operasi Dan Digital Service). (2023). Petunjuk Pelaksanaan Kemitraan Pickup.
- Interaction Design Foundation. (n.d.). What is User Centered Design (UCD). Interaction Design Foundation - IxDF.
- Irawan, W., Manaqib, M., & Fitriyati, N. (2021). Implementation of the Model Capacited Vehicle Routing Problem with Time Windows with a Goal Programming Approach in Determining the Best Route for Goods Distribution. *JMSK*, 17. <https://doi.org/10.20956/jmsk.v17i2.11107>
- Jiang, H., Lu, M., Tian, Y., Qiu, J., & Zhang, X. (2022). An evolutionary algorithm for solving Capacitated Vehicle Routing Problems by using local information. *Applied Soft Computing*, 17. <https://doi.org/https://doi.org/10.1016/j.asoc.2022.108431>
- Koc, C., Laporte, G., & Tukenmez, I. (2020). A review of vehicle routing with simultaneous pickup and delivery. *Computers & Operations Research*, 122. <https://doi.org/https://doi.org/10.1016/j.cor.2020.104987>
- Kompas.com. (2019). Ingat, Bawa Barang di Sepeda Motor Ada Aturannya. Kompas.Com. <https://otomotif.kompas.com/read/2019/07/11/074200115/ingat-bawa-barang-di-sepeda-motor-ada-aturannya>
- Kristina, S., Sianturi, R. D., & Husnadi, R. (2020). Penerapan Model CVRP Menggunakan Google Or-Tools Untuk Penentuan Rute Pengantaran Obat Pada Perusahaan Pedagang Besar Farmasi. *Jurnal Telematika*, 15.
- Lesch, V., Konig, M., Kounev, S., Stein, A., & Krupitzer, C. (2021). A Case Study of Vehicle Route Optimization. *Neural and Evolutionary Computing*. <https://doi.org/https://doi.org/10.48550/arXiv.2111.09087>
- Limei, H., Tannady, H., & Nurprihatin, F. (2018). Meminimumkan Biaya Transportasi pada Capacitated Vehicle Routing Problem dengan Metode Heuristik. *Seminar Rekayasa Teknologi Semrestek* 2018. <https://teknik.univpancasila.ac.id/semrestek/prosiding/index.php/12345/article/view/277>
- Liu, X., Chen, Y.-L., Por, Y. L., & Ku, C. S. (2023). A Systematic Literature Review of Vehicle Routing Problems with Time Windows. *Sustainability*, 15(15).
- Marcelina, D., & Yulianti, E. (2020). Aplikasi Pencarian Rute Terpendek Lokasi Kuliner Khas Palembang Menggunakan Algoritma Euclidean Distance dan A\*(Star). *Jurnal Sistem Informasi Dan Komputer*, 9(2). <https://doi.org/https://doi.org/10.32736/sisfokom.v9i2.827>

- Mardiko, I. A., & Sulistyowati. (2020). Optimasi Kapasitas Armada Dengan Menggunakan Metode Linear Programming. *Journal of Information Technology*, 5. <https://ejournal.itats.ac.id/integer/article/viewFile/908/784>
- Mazzuco D., Oliveira D., & Frazzon E.M. (2017). State of the art in simulation-based optimization approaches for vehicle routing problems along manufacturing supply chains. 24th International Conference on Production Research. [https://doi.org/State of the art in simulation-based optimization approaches for vehicle routing problems along manufacturing supply chains](https://doi.org/State%20of%20the%20art%20in%20simulation-based%20optimization%20approaches%20for%20vehicle%20routing%20problems%20along%20manufacturing%20supply%20chains)
- Moghdani, R., Salimifard, K., Demir, E., & Benyettou, A. B. (2021). The green vehicle routing problem: A systematic literature review. *Journal of Cleaner Production*, 279. <https://doi.org/https://doi.org/10.1016/j.jclepro.2020.123691>
- Moudya, F., Rarasati, N., & Syafmen, W. (2023). Optimisasi Rute Pada Cvrp Dalam Pendistribusian Gas Oksigen Menggunakan Algoritma Clarke And Wright Savings. *Jurnal Pendidikan Matematika Dan Matematika*, 9. <https://doi.org/10.24853/fbc.9.1.105-118>
- Nagy, G., & Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research*, 162. <https://doi.org/https://doi.org/10.1016/j.ejor.2002.11.003>
- Ngo, Q.-H. (2023). The effectiveness of market orientation in the logistic industry: A focus on SMEs in an emerging country. *Heliyon*, 9(7). <https://doi.org/https://doi.org/10.1016/j.heliyon.2023.e17666>
- Novindri, G. F., & Saian, P. O. N. (2022). Implementasi Flask Pada Sistem Penentuan Minimal Order Untuk Tiap Item Barang Di Distribution Center Pada PT XYZ Berbasis Website. *Jurnal MNEMONIC*, 5. <https://doi.org/10.36040/mnemonic.v5i2.4670>
- Nurmayanti, L., & Sudrajat, A. (2021). Implementasi linear programming metode simpleks pada home industry khasanah sari karawang. *Jurnal Manajemen*, 13(3). <https://doi.org/https://doi.org/10.30872/jmmn.v13i3.10085>
- Pane, S. F., Awangga, R. M., Rahmadani, E. V., & Permana, S. (2019). Implementasi Algoritma Genetika Untuk Optimalisasi Pelayanan Kependudukan. *Jurnal Teknik Insentif*, 13(2), 36–43. <https://doi.org/https://doi.org/10.36787/jti.v13i2.130>
- Patel, M., & Patel, N. (2019). Exploring Research Methodology: Review Article. *International Journal of Research and Review*, 6(3). [https://www.ijrrjournal.com/IJRR\\_Vol.6\\_Issue.3\\_March2019/IJRR0011.pdf](https://www.ijrrjournal.com/IJRR_Vol.6_Issue.3_March2019/IJRR0011.pdf)
- Paweł Sitek, Jarosław Wikarek, Katarzyna Rutczyńska-Wdowiak, Grzegorz Bocewicz, & Zbigniew Banaszak. (2021). Optimization of capacitated vehicle routing problem with alternative delivery, pick-up and time windows: A modified hybrid approach. *Neurocomputing*, 423. <https://doi.org/https://doi.org/10.1016/j.neucom.2020.02.126>
- Pos Indonesia. (2021). Pos Indonesia Annual Report. <https://drive.google.com/u/0/uc?id=1rGyhBZ-1ISXJmmJ0AIt-kLCOCu7hZ4fh&export=download>

- Pos Indonesia. (2023a). Sejarah Pos. Pos Indonesia. <https://www.posindonesia.co.id/id/content/sejarah-pos>
- Pos Indonesia. (2023b). Tentang Oranger. Kemitraan Pos Indonesia.
- Pos Indonesia. (2023c). Visi, Misi, Tujuan dan Tata Nilai. Pos Indonesia. <https://www.posindonesia.co.id/id/content/visi-misi-tujuan-dan-tata-nilai>
- PT. Pos Indonesia. (2023). O-Ranger. PT. Pos Indonesia.
- Rattu, P. N., Pioh, N. R., & Sampe, S. (2022). Optimalisasi Kinerja Bidang Sosial Budaya Dan Pemerintahan Dalam Perencanaan Pembangunan (Studi Di Kantor Badan Perencanaan Pembangunan, Penelitian Dan Pengembangan Daerah Kabupaten Minahasa). *Jurnal Governance*, 2(1), 1–9. <https://ejournal.unsrat.ac.id/index.php/governance/article/download/40264/36067>
- Ridwan, Saputra, M. A., & Indriyanti, R. (2020). Penerapan Logistik 4.0 Dalam Pendistribusian Barang Produksi Pt. Solusi Bangun Indonesia Tbk. Cilacap. Prosiding Seminar Sosial, 2(1). <https://e-journal.akpelni.ac.id/index.php/prosiding-nsmis/article/view/154>
- Ropiqoh, R., & Syafitri Lubis, R. (2023). Implementation of A-Star Algorithm In Finding The Shortest Route of Cooking Oil Distribution in Karo Regency Using Graph. *Jurnal Matematika Dan Pendidikan Matematika*, 9(2). <https://mathline.unwir.ac.id/index.php/Mathline/article/view/442>
- Sitek, P., & Wikarek, J. (2019). Capacitated vehicle routing problem with pick-up and alternative delivery (CVRPPAD): model and implementation using hybrid approach. *Annals of Operations Research*, 273. <https://link.springer.com/article/10.1007/s10479-017-2722-x>
- Sitompul, C., & Manasye Horas, O. (2021). A Vehicle Routing Problem with Time Windows Subject to the Constraint of Vehicles and Good's Dimensions. *International Journal of Technology (IJTech)*, 12(4).
- Soenadi, I. A., Joice, & Marpaung, B. (2019). Optimasi Capacitated Vehicle Routing Problem with Time Windows dengan Menggunakan Ant Colony Optimization. *Jurnal Sistem Dan Manajemen Industri*, 3(1). <https://doi.org/http://dx.doi.org/10.30656/jsmi.v3i1.1469>
- Supriani, B., Andrianto, A., & Syarif, E. B. (2023). Perancangan Kompartemen Tas Obrok Untuk Menunjang Keamanan Paket Yang Di Bawa Kurir. *EProceedings of Art & Design*. <https://openlibrarypublications.telkomuniversity.ac.id/index.php/artdesign/article/view/19598>
- Suroso, J. S. D., & Nugroho, P. (2023). Analisis Optimalisasi Produksi dengan Linear Programming Melalui Metode Simpleks (Studi Kasus UMKM Aqisa Rumah Rosella Surabaya). *Jurnal Kajian Ilmu Manajemen*, 3. <https://doi.org/10.21107/jkim.v3i2.18918>
- Taherdoost, H. (2021). Data Collection Methods and Tools for Research; A Step-by-Step Guide to Choose Data Collection Technique for Academic and Business Research

- Projects. *International Journal of Academic Research in Management*, 10(1), 10–38.  
<https://elvedit.com/journals/IJARM/wp-content/uploads>
- Tan, S.-Y., & Yeh, W.-C. (2021). The Vehicle Routing Problem: State-of-the-Art Classification and Review. *Applied Sciences*, 11.  
<https://doi.org/https://doi.org/10.3390/app112110295>
- Tokopedia. (n.d.). Tas Obrok Angkut Jumbo Murah Anti Air Kuat Hingga Beban 60 Kg. Tokopedia. Retrieved July 20, 2024, from <https://www.tokopedia.com/suryaguna/tas-obrok-angkut-jumbo-murah-anti-air-kuat-hingga-beban-60-kg>
- Trivaika, E., & Senubekti, M. A. (2022). Perancangan Aplikasi Pengelola Keuangan Pribadi Berbasis Android . *Nuansa Informatika Technology And Information Journal*, 16(1). <https://doi.org/https://doi.org/10.25134/nuansa>
- Wang, A., Ferro, N., Majewski, R., & E. Gounaris, C. (2021). Mixed-integer linear optimization for full truckload pickup and delivery. *Optim Lett*, 15.  
<https://doi.org/https://doi.org/10.1007/s11590-021-01736-x>
- Wang, Y., Li, Q., Guan, X., Fan, J., Xu, M., & Wang, H. (2021). Collaborative multi-depot pickup and delivery vehicle routing problem with split loads and time windows. *Knowledge-Based System*, 231.  
<https://doi.org/https://doi.org/10.1016/j.knosys.2021.107412>
- Wijayanto, C., & Susetyo, Y. A. (2022). Implementasi Flask Framework Pada Pembangunan Aplikasi Sistem Informasi Helpdesk (SIH). *Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika*, 7.
- Yanik, S., Bozkaya, B., & deKervenoael, R. (2014). A new VRPPD model and a hybrid heuristic solution approach for e-tailing. *European Journal of Operational Research*, 236.

## LAMPIRAN

### Lampiran 1. Kartu Bimbingan

7/24/24, 9:14 AM	Rekap Percakapan Bimbingan
 <div style="margin-top: 10px;"> <b>Universitas Logistik dan Bisnis Internasional</b>  <small>Jl. Sari Asih No.54, Sarijadi, Kec. Sukasari, Kota Bandung, Jawa Barat 40151  Website : <a href="http://www.ulbi.ac.id/">www.ulbi.ac.id/</a> e-Mail : <a href="mailto:info@ulbi.ac.id">info@ulbi.ac.id</a> / Telepon : 081311110194</small> </div>	
<b>REKAP PERCAKAPAN BIMBINGAN</b>	
<b>Judul Tugas Akhir</b> : Optimasi Rute Logistik Dengan Pembauran Kapasitas Kendaraan Pada Penanganan Pickup And Delivery Menggunakan Metode Heuristic	
<b>Sesi / Bahasan</b> : ke-1 / Menjelaskan topik TA dengan judul "OPTIMASI RUTE LOGISTIK DENGAN PEMBAURAN KAPASITAS KENDARAAN PADA PENANGANAN PICKUP AND DELIVERY MENGGUNAKAN METODE HEURISTIC", diperlukan arsitektur pada penjelasan atau narasi penelitian.	
<b>Mahasiswa</b> : 1204061 - Nur Tri Ramadhanti Adiningrum <b>Dosen Pembimbing</b> : 0407117405 - MUHAMMAD YUSRIL HELMI SETYAWAN	
<div style="border: 1px solid black; padding: 2px;"> <b>Dosen Pembimbing</b>  Senin, 10 Juni 2024, 14:43:32  Lanjutkan </div>	
<b>Sesi / Bahasan</b> : ke-1 / Mendeskripsikan topik Tugas Akhir = Capacitated Vehicle Routing Problem with Pickup and Delivery	
<b>Mahasiswa</b> : 1204061 - Nur Tri Ramadhanti Adiningrum <b>Dosen Pembimbing</b> : 0427078404 - CAHYO PRIANTO	
<div style="border: 1px solid black; padding: 2px;">Tidak ada data percakapan</div>	
<b>Sesi / Bahasan</b> : ke-2 / batasan diperlukan dalam penelitian, selain batasannya dari data, tentunya diperlukan juga argumen untuk batasan penelitian, eksplor smoothing eksplor mengenai "incoming dan outgoing" dalam logistik sebagai acuan batasan penelitian penjelasan algoritma ortools dan algoritma a-star masing-masing berperan untuk apa	
<b>Mahasiswa</b> : 1204061 - Nur Tri Ramadhanti Adiningrum <b>Dosen Pembimbing</b> : 0407117405 - MUHAMMAD YUSRIL HELMI SETYAWAN	
<div style="border: 1px solid black; padding: 2px;"> Senin, 10 Juni 2024, 14:44:21  Lanjutkan </div>	
<b>Sesi / Bahasan</b> : ke-2 / Penjelasan variabel yang digunakan, rencana metode yang digunakan	
<b>Mahasiswa</b> : 1204061 - Nur Tri Ramadhanti Adiningrum <b>Dosen Pembimbing</b> : 0427078404 - CAHYO PRIANTO	
<div style="border: 1px solid black; padding: 2px;">Tidak ada data percakapan</div>	
<b>Sesi / Bahasan</b> : ke-3 / Progress CVRP pada google colab. algoritma menggunakan PuLP. menghasilkan rute dan demand secara dasar / belum sempurna.	
<b>Mahasiswa</b> : 1204061 - Nur Tri Ramadhanti Adiningrum <b>Dosen Pembimbing</b> : 0407117405 - MUHAMMAD YUSRIL HELMI SETYAWAN	
<a href="https://siakad.ulbi.ac.id/siakad/fist_bimbingankonsultasi/printall/963">https://siakad.ulbi.ac.id/siakad/fist_bimbingankonsultasi/printall/963</a> <div style="float: right;">1/3</div>	

7/24/24, 9:14 AM

Rekap Percakapan Bimbingan

Senin, 10 Juni 2024, 15:35:38

1. Pertimbangkan penggunaan data yang telah diperoleh dari riset sebelumnya(int-1) node A-Z
2. Pertimbangkan penggunaan radius jarak untuk menotifikasi kedekatan terhadap node tertentu untuk mendapatkan akses rekomendasi
3. gunakan pendekatan linier programming untuk fokus solusi (cari referensi jurnal terkait kontribusi pendekatan ini pada kasus - kasus serupa)
4. perbaiki rumusan masalah
- 4.

**Sesi / Bahasan** : ke-3 / Menggunakab google colab dan coding menggunakan library PuLP

**Mahasiswa** : 1204061 - Nur Tri Ramadhanti Adiningrum **Dosen Pembimbing** : 0427078404 - CAHYO PRIANTO

Tidak ada data percakapan

**Sesi / Bahasan** : ke-4 / Mendeskripsikan rumusan masalah: untuk masalah ke b, dihapus karena tidak ada pemilihan variabel (sudah tetap) penetapan batasan atau ruang lingkup metodologi utama ditambahkan 1 untuk pembuatan UI/aplikasi buat metodologi aplikasi disetiap langkah metodologi ada metodenya

**Mahasiswa** : 1204061 - Nur Tri Ramadhanti Adiningrum **Dosen Pembimbing** : 0407117405 - MUHAMMAD YUSRIL HELMI SETYAWAN

Tidak ada data percakapan

**Sesi / Bahasan** : ke-4 / menjelaskan proses bisnis pickup and delivery. harapannya adalah meminimalisir jumlah kendaraan yang bertugas. mencari penjelasan mengenai algoritma linier programming dan menjelaskan menggunakan pustaka PuLP

**Mahasiswa** : 1204061 - Nur Tri Ramadhanti Adiningrum **Dosen Pembimbing** : 0427078404 - CAHYO PRIANTO

Tidak ada data percakapan

**Sesi / Bahasan** : ke-5 / Belajar mengenai cara perhitungan dan proses menghitung manual linear programming. perdalam perhitungan bagian metode simplex.

**Mahasiswa** : 1204061 - Nur Tri Ramadhanti Adiningrum **Dosen Pembimbing** : 0427078404 - CAHYO PRIANTO

Tidak ada data percakapan

**Sesi / Bahasan** : ke-5 / gunakan data buatan menggunakan data sintesis kapsaitas cukup tas obrok saja evaluasi perbandingan jarak

**Mahasiswa** : 1204061 - Nur Tri Ramadhanti Adiningrum **Dosen Pembimbing** : 0407117405 - MUHAMMAD YUSRIL HELMI SETYAWAN

Tidak ada data percakapan

**Sesi / Bahasan** : ke-6 / Progress hasil cara kerja linear programming hingga menemukan kondisi optimal. memenuhi fungsi tujuan, memenuhi dan variabel keputusan. melanjutkan codingan. variabel yang digunakan : jarak, kapasitas kendaraan, jumlah moda kendaraan

**Mahasiswa** : 1204061 - Nur Tri Ramadhanti Adiningrum **Dosen Pembimbing** : 0427078404 - CAHYO PRIANTO

Tidak ada data percakapan

**Sesi / Bahasan** : ke-6 / Evaluasi ----- Evaluasi model menggunakan jarak dan juga jumlah kendaraan yang digunakan data evaluasi adalah data sintesis untuk memudahkan cara analisis kapasitas menggunakan tas obrok kurir motor (maksimal kapasitas 60kg) Flask ----- Menggunakan Flask, untuk pustaka Tkinter tidak digunakan Jurnal ----- Cari jurnal yang berkontak WA secara langsung. tanya LoA yang bisa didapatkan di bulan Agustus

[https://siakad.ulbi.ac.id/siakad/fist\\_bimbingankonsultasi/printall/963](https://siakad.ulbi.ac.id/siakad/fist_bimbingankonsultasi/printall/963)

2/3



7/24/24, 9:14 AM

Rekap Percakapan Bimbingan

**Mahasiswa** : 1204061 - Nur Tri Ramadhanti Adiningrum **Dosen Pembimbing** : 0407117405 - MUHAMMAD YUSRIL HELMI SETYAWAN

Tidak ada data percakapan

**Sesi / Bahasan** : ke-7 / aplikasi flask, perbandingan jarak catatan: BAB 1 = masalah, impact. Data/area studi, data dan tempat, metodologi hasil = kesimpulan menggambarkan kinerja model (proses evaluasi, cari matrix evaluasi LP) dan aplikasi

**Mahasiswa** : 1204061 - Nur Tri Ramadhanti Adiningrum **Dosen Pembimbing** : 0407117405 - MUHAMMAD YUSRIL HELMI SETYAWAN

Tidak ada data percakapan

**Sesi / Bahasan** : ke-7 / Spesifikan menjadi Pickup saja tambahkan lokasi baru menggunakan latitude dan longitude gunakan lokasi bentuk melingkar perbaiki tampilan GUI patokan maksimal kapasitas liat dari kapasitas motornya

**Mahasiswa** : 1204061 - Nur Tri Ramadhanti Adiningrum **Dosen Pembimbing** : 0427078404 - CAHYO PRIANTO

Tidak ada data percakapan

**Sesi / Bahasan** : ke-8 / Memperbaiki Jurnal ilmiah penelitian untuk publikasi

**Mahasiswa** : 1204061 - Nur Tri Ramadhanti Adiningrum **Dosen Pembimbing** : 0407117405 - MUHAMMAD YUSRIL HELMI SETYAWAN

Tidak ada data percakapan

**Sesi / Bahasan** : ke-8 / Evaluasi ----- buatlah evaluasi secara apple to apple jika solusi hasil 2 kendaraan, buatlah juga evaluasi hanya 2 kendaraan untuk rute random Flask ----- Perbaiki aplikasi agar rute menggunakan API Jurnal ----- Cari jurnal yang berkontak WA secara langsung, tanya LoA yang bisa didapatkan di bulan Agustus

**Mahasiswa** : 1204061 - Nur Tri Ramadhanti Adiningrum **Dosen Pembimbing** : 0427078404 - CAHYO PRIANTO

Tidak ada data percakapan

**Sesi / Bahasan** : ke-9 / Aplikasi final menggunakan flask, dan API Key, sudah berbentuk garis lurus. Evaluasi analisis 2 kendaraan, demand dan lokasi sama, hasil menunjukan dengan LP rutenya sangat pendek. Optimality Gap = 0%

**Mahasiswa** : 1204061 - Nur Tri Ramadhanti Adiningrum **Dosen Pembimbing** : 0427078404 - CAHYO PRIANTO

Tidak ada data percakapan

## Lampiran 2 Surat Persetujuan Pelaksanaan Wawancara &amp; Observasi



Nomor : 093/TKA/1024  
 Lampiran : -  
 Perihal : Persetujuan Pelaksanaan  
 Wawancara & Observasi

Bandung, 15 Maret 2024

Kepada  
 Yth Ketua Program Studi D-IV Teknik  
 Informatika  
 Universitas Logistik & Bisnis  
 Internasional

Dengan hormat,

Menunjuk surat Saudara tanggal 07 Maret 2024, 028-001/PROD4TI-ULBI/SPm/III/2024 perihal Permohonan Melakukan Wawancara & Observasi, kami beritahukan bahwa pada prinsipnya kami menyetujui Pelaksanaan Wawancara & Observasi pada tanggal 18 Maret 2024 di institusi kami bagi mahasiswa Saudara sebagai berikut :

NO	NAMA	NIM	PRODI	UNIVERSITAS
1.	Nur Tri Ramadhanti Adiningrum	1204061	Teknologi Informatika	ULBI
2.	Rifqi Fathurrohman	1204076	Teknik Informatika	ULBI

Untuk selanjutnya, mohon sebelum Pelaksanaan Wawancara & Observasi dimaksud, kepada mahasiswa yang bersangkutan agar terlebih dahulu melapor kepada kami.

Demikian kami sampaikan, atas perhatian dan kerja samanya diucapkan terima kasih.

Hormat kami,

POS INDONESIA



AAN NURHAMDAN

Manajer Tata Kelola Administrasi

## Lampiran 3 Surat Permohonan Wawancara Observasi Ke KCU Bandung

20/03/24, 11.05

Permohonan Wawancara & Observasi

Nomor : C.Pos.16397/ND 100/ND06111/2024

Lampiran : 1 (Satu) Berkas

Perihal : Permohonan Wawancara & Observasi

Kepada

Pj. Manajer Pelayanan  
Outlet dan Operasi  
Cabang KCU Bandung

Menunjuk surat dari Ketua Program Studi D-IV Teknik Informatika Nomor : 028-001/PROD4TI-ULBI/SPm/III/2024 Universitas Logistik & Bisnis Internasional perihal Permohonan Wawancara & Observasi, diberitahukan bahwa mahasiswa tersebut ini akan melaksanakan kegiatan observasi dan wawancara pada Bulan Maret 2024 ini di unit kerja Saudara, yaitu sebagai berikut :

No.	NAMA	NIM	PRODI	Universitas
1.	Nur Tri Ramadhanti Adiningrum	1204061	D-IV Teknik Informatika	ULBI
2.	Rifqi Fathurrohman	1204076	D-IV Teknik Informatika	ULBI

Demikian kami sampaikan, atas perhatian dan kerjasamanya diucapkan terima kasih.

Bandung, 18 Maret 2024

Manajer Tata Kelola  
Administrasi

Aan Nurhamdan

Nippos. 975301305

**PT POS INDONESIA (PERSERO)**  
**Tata Kelola Administrasi**  
Bandung

Dokumen ini dan informasi yang terkandung di dalamnya hanya dipergunakan untuk kepentingan internal PT. Pos Indonesia. Setiap perbuatan atau tindakan, apapun cara dan bentuknya, yang mengakibatkan kandungan informasi tersebut diketahui oleh pihak-pihak yang tidak berhak dapat dikenai sanksi indisipliner dan/atau sanksi hukum.

<https://nde.posindonesia.co.id/nde/print/390112?token=dcb8fd281ab2ceebd9147b2e1d7bcb42>

1/1