MINIMASI JARAK DAN KETERLAMBATAN PADA KASUS *CAPACITATED VRP WITH TIME WINDOW* MENGGUNAKAN *NEAREST INSERTION* DAN *GENETIC ALGORITHM* (STUDI KASUS: PT POS LOGISTIK INDONESIA)

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata-1
Program Studi Teknik Industri - Fakultas Teknologi Industri
Universitas Islam Indonesia

Nama : Nabila Lintang Kirana
No. Mahasiswa : 21522170

PROGRAM STUDI TEKNIK INDUSTRI PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA

2025

PERNYATAAN KEASLIAN

Saya mengakui bahwa tugas akhir ini adalah hasil karya saya sendiri kecuali kutipan dan ringkasan yang seluruhnya sudah saya jelaskan sumbernya. Jika dikemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah maka saya bersedia ijazah yang telah saya terima ditarik kembali oleh Universitas Islam Indonesia.

Yogyakarta, 01 – Agustus - 2025



(Nabila Lintang Kirana)

21522170

SURAT BUKTI PENELITIAN

FAKULTAS
TEKNOLOGI INDUSTRI
Gedung KH. Mas Mansur
Kampus Terpadu Universitas Islam Indonesia
Jl. Kaliurang km 14,5 Yogyakarta 55584
T. (0274) 898444 ext. 4110, 4100
F. (0274) 896007
E. fti@uii.ac.id
W. fti.uii.ac.id

SURAT KETERANGAN PENELITIAN

Nomor : 114/Ka.lab SIMANTI/20/Lab.SIMANTI/VI/2025

Assalamu'alaikum Warohmatullahi Wabarokaatuh

Dengan hormat,

Yang bertanda tangan dibawah ini, menerangkan bahwa:

Nama : Nabila Lintang Kirana

Nim : 21522170

Jurusan : Teknik Industri

Dosen Pembimbing : Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM

Menyatakan bahwa mahasiswa tersebut diatas telah melaksanakan penelitian tugas akhir dengan judul " **MINIMASI JARAK DAN KETERLAMBATAN PADA KASUS *CAPACITATED VRP WITH TIME WINDOW* MENGGUNAKAN *NEAREST INSERTION* DAN *GENETIC ALGORITHM* (STUDI KASUS: PT POS LOGISTIK INDONESIA)".** Mulai pelaksanaan penelitian Mulai Penelitian, 10 sampai 5 Mei 2025

Demikian surat keterangan penelitian ini kami buat. Atas perhatiannya dan kerja samanya kami mengucapkan terima kasih.

Wassalamu'alaikum Warohmatullahi Wabarokaatuh

Yogyakarta, 16 Juni 2025

Kepala Laboratorium

Sistem Manufaktur Terintegrasi

Putri Dwi Annisa, S.T., M.Sc.

LEMBAR PENGESAHAN PEMBIMBING

MINIMASI JARAK DAN KETERLAMBATAN PADA KASUS *CAPACITATED VRP WITH TIME WINDOW* MENGGUNAKAN *NEAREST INSERTION* DAN *GENETIC ALGORITHM* (STUDI KASUS: PT. POS LOGISTIK INDONESIA)

TUGAS AKHIR

Disusun Oleh :

Nama	**: Nabila Lintang Kirana**
No. Mahasiswa	**: 20522170**

Yogyakarta, 01 Agustus 2025

Dosen Pembimbing

Ir. Muhammad Ridwan Andi Purnomo, ST., M.Sc., Ph.D., IPM

NIK. 015220101

LEMBAR PENGESAHAN DOSEN PENGUJI

MINIMASI JARAK DAN KETERLAMBATAN PADA KASUS *CAPACITATED VRP WITH TIME WINDOW* MENGGUNAKAN *NEAREST INSERTION* DAN *GENETIC ALGORITHM* (STUDI KASUS: PT. POS LOGISTIK INDONESIA)

TUGAS AKHIR

Disusun Oleh :

Nama : Nabila Lintang Kirana

No. Mahasiswa : 21522170

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata-1 Teknik Industri Fakultas Tekonologi Industri Universitas Islam Indonesia

Yogyakarta, 04 Agustus 2025

Tim Penguji

Ir. M. Ridwan Andi Purnomo, ST., M.Sc., Ph.D., IPM	
Ketua	
Wahyudhi Sutrisno, S.T., M.M., M.T.	200825
Anggota I	
Putri Dwi Annisa, S.T., M.Sc.	
Anggota II	

Mengetahui,

Ketua Program Studi Teknik Industri Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

UNIVERSITAS ISLAM INDONESIA YOGYAKARTA FAKULTAS TEKNOLOGI INDUSTRI

Ir. Muhammad Ridwan Andi Purnomo, ST., M.Sc., Ph.D., IPM

NIK. 015220101

HALAMAN PERSEMBAHAN

Segala puji bagi Allah Subhanahu Wa Ta'ala, yang dengan izin dan ridha-Nya, akhirnya saya dapat menyelesaikan tugas akhir ini sebagai salah satu syarat dalam menempuh gelar Sarjana Teknik Industri di Universitas Islam Indonesia. Dengan penuh rasa syukur dan bangga, karya ini saya persembahkan pertama-tama untuk diri saya sendiri—atas segala usaha dan semangat yang tidak pernah padam dalam menghadapi setiap tantangan. Ini adalah bukti bahwa perjuangan dan keyakinan mampu membawa langkah kecil menjadi pencapaian yang berarti. Ucapan terima kasih dan persembahan yang tulus saya tujukan kepada kedua orang tua, untuk saudara-saudaraku, untuk sahabat-sahabatku dan teman-teman seperjuanganku, terima kasih atas dukungan, canda tawa, dan kehadiran kalian yang membuat langkah ini terasa lebih ringan. Semoga tulisan sederhana ini menjadi awal dari kontribusi kecil saya bagi ilmu pengetahuan dan kehidupan, serta menjadi sesuatu yang membanggakan bagi orang-orang yang saya cintai.

MOTTO

Fa-idza 'azamta fatawakkal 'alallah

"Maka apabila kamu telah bertekad, bertawakallah kepada Allah."

— QS. Ali 'Imran: 159

"*You have your own lane, own race, and own pace!*"

KATA PENGANTAR

Assalamu'alaikum Warrahmatullahi Wabarakatuh.

Alhamdulillahirabbil 'alamin, puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, taufik, serta hidayah-Nya. Shalawat dan salam tercurah kepada Nabi Muhammad SAW, beserta keluarga, para sahabat, serta seluruh umatnya yang senantiasa bertakwa kepada Allah SWT. Berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan Tugas Akhir yang berjudul: **"Minimasi Jarak dan Keterlambatan Pada Kasus *Capacitated VRP With Time Window* Menggunakan *Nearest Insertion* dan *Genetic Algorithm* (Studi Kasus: Pt. Pos Logistik Indonesia)"**.

Tugas Akhir ini disusun sebagai salah satu syarat untuk menyelesaikan jenjang pendidikan Strata-1 pada Program Studi Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia. Dalam proses penyusunan Tugas Akhir ini, penulis banyak menerima bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, dengan penuh rasa hormat dan penghargaan, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. *Allah SWT,* atas segala limpahan rahmat dan pertolongan-Nya dalam setiap langkah penulis.
2. Bapak Prof. Dr. Ir. Hari Purnomo, MT. IPU. ASEAN. Eng selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Dr. Drs. Imam Djati Widodo, M.Eng. Sc, selaku Ketua Jurusan Teknik Industri Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Bapak, Ir. M. Ridwan Andi Purnomo, ST., M.Sc., Ph.D., IPM. selaku Ketua Program Studi Teknik Industri Program Sarjana Universitas Islam Indonesia.
5. Bapak Ir. M. Ridwan Andi Purnomo, ST., M.Sc., Ph.D., IPM., selaku Dosen Pembimbing yang telah bersedia membimbing penulis dalam pelaksanakan penelitian, bersedia meluangkan waktu juga pikirannya serta memberikan motivasi, semangat dan dukungan.
6. Bapak Wahyu Kurniawan, S.T., M.T., CISCP dan Ibu Putri Dwi Annisa, S.T., M.Sc. yang telah memberikan bimbingan serta wadah untuk peneliti sehingga dapat melaksanakan penelitian tugas akhir di PT. Pos Logistik Indonesia dengan baik.
7. Kedua orang tua tercinta dan kelaurga atas doa, kasih sayang, dan dukungan moril maupun materil yang tiada henti.
8. Kepada Rifki Nurul Mukarim, Mochammad Adib Heramunida, Vira Dwi Apryola, Sastyaviani Irami Putri dan Adhira Yudanto yang telah membersamai dan memberikan dukungan mental selama pengerjain Tugas Akhir ini.
9. Sahabat-sahabat dan orang terkasih disekitar yang senantiasa memberikan semangat dan dorongan penuh atas kerja keras yang saya lakukan.
10. Seluruh keluarga Laboratorium Sistem Manufaktur dan asisten Angkatan 2021, yang telah berjuang bersama sama, saling menguatkan, memberikan dukungan dan doa kepada penulis sehingga dapat menyelesaikan tugas akhir dengan baik.
11. Seluruh pihak yang tidak dapat penulis sebutkan satu per satu, yang turut membantu secara langsung maupun tidak langsung dalam proses penyusunan Tugas Akhir ini.

Penulis menyadari bahwa dalam penyusunan laporan ini masih terdapat banyak kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi perbaikan di masa mendatang. Semoga Tugas Akhir ini dapat memberikan manfaat dan menjadi kontribusi positif bagi pengembangan ilmu pengetahuan, khususnya di bidang Teknik Industri.
Wassalamu'alaikum warrahmatullahi wabarakatuh,

Yogyakarta, 01 Agustus 2025



Nabila Lintang Kirana
21522170

ABSTRAK

Distribusi logistik merupakan salah satu aktivitas krusial dalam memastikan pengiriman barang dari gudang ke pelanggan berjalan tepat waktu, efisien, dan sesuai dengan permintaan. PT. Pos Logistik Indonesia sebagai penyedia jasa logistik nasional menghadapi tantangan dalam pengelolaan rute distribusi akibat keterbatasan metode perencanaan yang digunakan. Permasalahan yang muncul berkaitan dengan Capacitated Vehicle Routing Problem with Time Windows (CVRPTW), di mana kendaraan harus mengantarkan barang ke sejumlah pelanggan dengan mempertimbangkan kapasitas angkut dan batasan waktu layanan (time windows). Penentuan rute kendaraan yang optimal sangat penting untuk mengurangi total jarak tempuh dan meminimalkan waktu keterlambatan, sehingga biaya operasional dapat ditekan dan tingkat kepuasan pelanggan meningkat. Pendekatan konvensional yang selama ini digunakan belum mampu memberikan solusi yang efisien. Oleh karena itu, penelitian ini menerapkan pendekatan *Nearest Insertion* sebagai solusi awal dan Algoritma Genetika (*Genetic Algorithm*) sebagai metode optimasi untuk menyelesaikan permasalahan CVRPTW. Model optimasi ini mempertimbangkan jarak tempuh, waktu pelayanan, serta penalti akibat keterlambatan pengiriman. Melalui usulan rute optimasi yang diperoleh yaitu 4 rute kendaraan dengan total distribusi ke 100 titik pelanggan, hasil penelitian menunjukkan bahwa pendekatan ini mampu menurunkan total jarak tempuh sebesar 26,62% dan menghilangkan keterlambatan hingga 100% dibandingkan rute eksisting perusahaan. Dengan demikian, metode yang diusulkan terbukti efektif dalam meningkatkan efisiensi distribusi logistik serta dapat menjadi kontribusi bagi pengembangan studi optimasi rute berbasis metaheuristik.

Kata kunci: *Vehicle Routing Problem*, *Time Windows, Nearest Insertion*, Algoritma Genetika, Optimasi, Logistik.

DAFTAR ISI

PERNYATAAN KEASLIAN ..... **ii**
SURAT BUKTI PENELITIAN ..... **iii**
LEMBAR PENGESAHAN PEMBIMBING ..... **iv**
LEMBAR PENGESAHAN DOSEN PENGUJI ..... **v**
HALAMAN PERSEMBAHAN ..... **vi**
MOTTO ..... **vii**
KATA PENGANTAR ..... **viii**
ABSTRAK ..... **x**
DAFTAR ISI ..... **xi**
DAFTAR TABEL ..... **xiii**
DAFTAR GAMBAR ..... **xiv**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 5
1.3 Tujuan Penelitian 5
1.4 Manfaat Penelitian 5
1.5 Batasan Penelitian 6
1.6 Sistematika Penulisan 6
BAB II TINJAUAN PUSTAKA **8**
2.1 *Supply Chain & Supply Chain Management* 14
2.2 Logistik 16
2.3 Manajemen Logistik 20
2.4 Transportasi dan Distribusi 23
2.5 *Vehicle Routing Problem* (VRP) 24
2.6 Algoritma Optimasi 26
2.7 *Stochastic Programming* 28
2.7.1 Pemrograman Stokastik linier 29
2.7.2 Pemrograman Stokastik non-linier 29
2.8 *Genetic Algorithm* (Algoritma Genetika) 29
BAB III METODE PENELITIAN **35**
3.1 Objek Penelitian 35
3.2 Instrumen Penelitian 36
3.3 Sumber Data Primer dan Sekunder 36
3.4 Metode Pengumpulan Data 37
3.5 Populasi dan Sampel 37
3.5.1 Populasi 37
3.5.2 Sampel 37
3.6 Metode Pengolahan Data 38
3.6.1 Model Matematis CVRPTW 38
3.6.2 Metode *Nearest Insertion* (NI) 42
3.6.3 Metode *Genetic Algorithm* (GA) 43
3.7 Alur Penelitian 44
BAB IV PENGUMPULAN DAN PENGOLAHAN DATA **50**
4.1 Profil Perusahaan 50
4.1.1 Deskripsi Perusahaan 50

4.1.2 Sejarah Perusahaan 50
4.1.3 Struktur Organisasi 51
4.1.4 Visi dan Misi 52
4.1.5 Produk dan Layanan 52
4.2 Pengumpulan Data 57
4.2.1 Data Kendaraan 57
4.2.2 Data Titik Lokasi 58
4.2.3 Data *Demand* Pengiriman Barang 59
4.2.4 Data *Time Windows* 60
4.2.5 Data Pendukung 61
4.2.6 Rute Eksisting Kebijakan Perusahaan 61
4.3 Pengolahan Data 65
4.3.1 Pengembangan Model Konseptual 65
4.3.2 Pengembangan Program dan Model Matematis 66
4.3.3 Pengembangan Algoritma 76
4.3.4 Verifikasi dan Validasi Model 87
4.3.5 Eksperimen Iterasi dan Implementasi Algoritma 90
4.3.6 Hasil Rute Optimal 93
BAB V HASIL DAN PEMBAHASAN 95
5.1 Analisis Rute Hasil Optimasi 95
5.2 Analisis Penurunan Jarak dan Waktu Tempuh 103
5.3 Analisis Keterlambatan dan Penalti 106
5.4 Analisis Berbasis *Estimated Time Arrival* (ETA) dan Potensi Pengembangan Model 108
5.5 Analisis Dampak Lingkungan (*Sustainability*) 109
5.6 Analisis Faktor Kemacetan (*Traffic)* 109
BAB VI PENUTUP 111
6.1 Kesimpulan 111
6.2 Saran 112
DAFTAR PUSTAKA 114
LAMPIRAN B-1

DAFTAR TABEL

Tabel 2. 1 Posisi Penelitian 12
Tabel 4. 1 Jenis Kendaraan 58
Tabel 4. 2 Data Titik Lokasi 58
Tabel 4. 3 Data *Demand* Pengiriman Barang 59
Tabel 4. 4 Data Waktu Layanan PT Pos Logistik Indonesia 60
Tabel 4. 5 Data *Time Windows* 60
Tabel 4. 6 Data Pendukung Penelitian 61
Tabel 4. 7 Total Jarak dan Waktu Tempuh Kendaraan 1 62
Tabel 4. 8 Total Jarak dan Waktu Tempuh Kendaraan 2 63
Tabel 4. 9 Total Jarak dan Waktu Tempuh Kendaraan 3 64
Tabel 4. 10 *Grand Total* Rute Eksisting Perusahaan 65
Tabel 4. 11 Nilai Faktor Koreksi ETA 70
Tabel 4. 12 *Distance Matrix Node 0-100* 72
Tabel 4. 13 *Distance Matrix* dengan Faktor Koreksi 0-100 73
Tabel 4. 14 *Output* Rute *Nearest Insertion (NI)* 79
Tabel 4. 15 Contoh Hasil Kromosom Terbentuk GA 80
Tabel 4. 16 Ukuran Populasi GA 82
Tabel 4. 17 Contoh Hasil *Fitness* Terbentuk GA 82
Tabel 4. 18 Contoh Hasil Proses *Ordered Crosover (OX)* 200 Gen 84
Tabel 4. 19 Verifikasi Model 87
Tabel 4. 20 Validasi Model 89
Tabel 4. 21 Hasil Rute Optimal 94
Tabel 5. 1 Detail Rute Kendaraan 1 95
Tabel 5. 2 Detail Rute Kendaraan 2 97
Tabel 5. 3 Detail Rute Kendaraan 3 100
Tabel 5. 4 Ringkasan Hasil Rute Optimal 102
Tabel 5. 5 Ringkasan Penurunan Jarak dan Waktu Tempuh 103
Tabel 5. 6 Titik dengan Keterlambatan Pengiriman 106
Tabel 5. 7 Ringkasan Perbandingan 107

DAFTAR GAMBAR

Gambar 2. 1 Skema Alur SCM 15
Gambar 2. 2 *The First Party Logistic* 18
Gambar 2. 3 *The Second Party Logistic* 19
Gambar 2. 4 *The Third Party Logistic* 19
Gambar 2. 5 *The Fourth Party Logistic* 20
Gambar 2. 6 *Logistic Providers* 20
Gambar 2. 7 Peran Manajemen Logistik 21
Gambar 2. 8 Fungsi Manajemen Logistik 22
Gambar 2. 9 Siklus Sistem Manajemen Logistik 23
Gambar 2. 10 *Stakeholder* Transportasi 24
Gambar 2. 11 Contoh Solusi VRP 25
Gambar 2. 12 Pengelompokkan Algoritma Optimasi 27
Gambar 2. 13 Klasifikasi Algoritma Optimasi 28
Gambar 2. 14 Visualisasi Algoritma Genetika 30
Gambar 2. 15 Proses *Order Crossover* 32
Gambar 2. 16 Proses *Swap Mutation* 33
Gambar 2. 17 *Flowchart* Umum Algoritma Genetika 33
Gambar 3. 1 PT Pos Logistik Indonesia 35
Gambar 3. 2 Alur Penelitian 45
Gambar 4. 1 Sejarah PT Pos Logistik Indonesia 51
Gambar 4. 2 Struktur Organisasi PT Pos Logistik Indonesia 51
Gambar 4. 3 Layanan Pergudangan 53
Gambar 4. 4 Layanan Distribusi 54
Gambar 4. 5 Layanan *Heavy Lifting* 55
Gambar 4. 6 Layanan Kargo Domestik 55
Gambar 4. 7 Layanan *International Cargo* 57
Gambar 4. 8 Layanan *Cold Chain Services* 57
Gambar 4. 9 Plot Rute Eksisting Kendaraan 1 62
Gambar 4. 10 Plot Rute Eksisting Kendaraan 2 63
Gambar 4. 11 Plot Rute Eksisting Kendaraan 3 64
Gambar 4. 12 Skema Model Konseptual 66
Gambar 4. 13 *Pseudo Code* Parameter 67
Gambar 4. 14 *Pseudo Code Haversine* 69
Gambar 4. 15 Model Fungsi Koreksi ETA 70
Gambar 4. 16 *Pseudo Code Travel Time (ETA)* 74
Gambar 4. 17 *Pseudo Code Multi-Objective Function* 74
Gambar 4. 18 *Pseudo Code Constraint* 76
Gambar 4. 19 *Pseudo Code* Inisialiasi Rute 77
Gambar 4. 20 *Pseudo Code* Inisialiasi Kendaraan 77
Gambar 4. 21 *Pseudo Code* Pencarian *Customer Feasibility* 78
Gambar 4. 22 *Pseudo Code* Pencarian *Best Customer* 78
Gambar 4. 23 *Pseudo Code* Pengembalian Kendaraan ke Depot 79

Gambar 4. 24 *Pseudo Code* Menyisipkan Sisa *Customer* 79
Gambar 4. 25 *Pseudo Code* GA Representasi Kromosom 80
Gambar 4. 26 *Pseudo Code* GA Inisialisasi Populasi 81
Gambar 4. 27 *Pseudo Code* GA Inisialisasi Populasi 82
Gambar 4. 28 *Pseudo Code* GA Proses Seleksi 83
Gambar 4. 29 *Pseudo Code* GA Proses *Crossover* 84
Gambar 4. 30 *Source Code* GA Proses *Mutation* 85
Gambar 4. 31 Contoh Proses Mutasi *Swap* 86
Gambar 4. 32 *Pseudo Code* GA Proses Pembentukan Generasi Baru 87
Gambar 4. 33 Hasil Eksperimen Populasi 91
Gambar 4. 34 Hasil Eksperimen *Mutation Rate* 92
Gambar 4. 35 Hasil Eksperimen Generasi 93
Gambar 5. 1 Visualisasi Rute Optimal Kendaraan 1 97
Gambar 5. 2 Visualisasi Rute Optimal Kendaraan 2 99
Gambar 5. 3 Visualisasi Rute Optimal Kendaraan 3 102
Gambar 5. 4 Grafik Persentase Penurunan Total Jarak Tempuh 3 Rute (Km) 104
Gambar 5. 5 Grafik Persentase Penurunan Total Waktu Tempuh 3 Rute (Menit) 105
Gambar 5. 6 Grafik Perbandingan Hasil Rute 108

BAB I
PENDAHULUAN

1.1 Latar Belakang

Logistik merupakan suatu kegiatan operasional dalam memberikan layanan kepada konsumen mencakup aktivitas yang melibatkan perencanaan, pelaksanaan, dan pengendalian arus barang serta jasa dari sumber pasokan hingga ke tangan konsumen. Proses ini meliputi pengangkutan, penyimpanan, packing, hingga distribusi (Fahreza, et al., 2023). Dalam logistik, distribusi juga melibatkan pemilihan jalur pengiriman terbaik, pengoptimalan rute, serta koordinasi antara berbagai pihak yang terlibat dalam rantai pasokan, termasuk produsen, distributor, pengangkut, dan pengecer, untuk memastikan produk sampai kepada pelanggan dengan cepat dan dalam kondisi yang baik (Galih & Sukmadewi, 2024). Tujuannya untuk memastikan barang atau jasa didistribusikan tepat waktu, sesuai jenis, dan pada lokasi yang benar, sekaligus mengoptimalkan waktu dan biaya agar operasional perusahaan dapat berjalan lebih efisien (Pratiwi, et al., 2023).

PT. Pos Logistik Indonesia merupakan salah satu perusahaan distribusi pihak ketiga logistik (*third-parties logistic*) di Indonesia yang berkontribusi besar dalam menyediakan layanan distribusi barang ke seluruh wilayah Indonesia. Salah satu strategi penting dalam logistik adalah penentuan rute kendaraan yang efisien, dikenal sebagai *Vehicle Routing Problem* (VRP). Suatu perusahaan perlu memperhatikan strategi penentuan rute yang optimal kendaraan (*Vehicle Routing Problem*), sehingga efisiensi dalam logistik tersebut dapat meningkatkan profitabilitas, daya saing, dan kepuasan pelanggan (Galih & Sukmadewi, 2024). Permasalahan ini semakin menantang dalam konteks geografis Indonesia, yang terdiri dari ribuan pulau dan infrastruktur logistik yang tidak merata. Kondisi ini menimbulkan hambatan tambahan dalam menentukan rute optimal, terutama bagi perusahaan distribusi besar seperti PT. Pos Logistik Indonesia.

Salah satu permasalahan utama yang dihadapi oleh PT. Pos Logistik Indonesia adalah bagaimana cara perusahaan menyelesaikan permasalahan *Capacitated Vehicle Routing Problem with Time Window* (CVRPTW). Pada kasus CVRPTW, kendaraan tidak hanya mengantarkan barang tetapi juga harus mempertimbangkan kapasitas kendaraan, dan batasan waktu (*time windows*) yang telah ditentukan yaitu pada pukul 08.00 hingga pukul 21.00 yang dibagi menjadi 3 periode. Kapasitas kendaraan yang ditentukan perusahaan dalam distribsui

menggunakan kendaraan jenis *blindvan* adalah sebesar 1000 Kg, yang mana kapasitas tersebut harus mampu menampung seluruh *demand* barang pelanggan yang akan diantarkan menuju ke titi kantar. Selain itu, PT. Pos Logistik Indonesia dalam menghadapi permasalahan tersebut dihadapkan pada tantangan untuk meningkatkan efisiensi distribusi, khususnya dalam mengurangi jarak tempuh dan ketepatan waktu pengiriman, yang secara langsung berdampak pada penurunan biaya operasional.

Berdasarkan pengamatan lapangan dan diskusi dengan pihak operasional, diketahui bahwa rute yang terlalu panjang merupakan salah satu faktor utama terbesar yang mengakibatkan pengantaran yang melebihi jadwal *time window* pelanggan, akibatnya supir sering kali kembali ke depot larut malam diluar batas kerja perusahaan hingga pukul 12 malam. Selain itu berpengaruh kepada konsumsi bahan bakar harian kendaraan yang kurang optimal. Ketidakefisienan ini tidak hanya berdampak pada kelelahan pengemudi, tetapi juga membatasi jumlah ritase harian dan menambah beban biaya bahan bakar.

Selain itu, tingginya biaya penalti akibat keterlambatan pengiriman menjadi indikasi kuat bahwa proses distribusi belum berjalan optimal. Berdasarkan data internal, keterlambatan pengiriman yang melampaui batas waktu (*time window*) dikenakan penalti pada tiap titik sebesar Rp. 2000. Berdasarkan data distribusi area Jakarta Raya pada tahun 2024 sebanyak 100 titik, terjadi pengiriman yang jumlah keterlambatannya mencapai 16% dari total pengiriman keseluruhan dengan keterlambatan terbesar selama 704 menit dan biaya penalti sebesar Rp. 18.000. Hal ini menunjukkan adanya masalah sistemik dalam penjadwalan dan penentuan rute yang efektif. Meskipun persentase keterlambatan pengiriman hanya sebesar 16%, angka tersebut tetap signifikan jika dilihat dari volume pengiriman yang besar. Kondisi tersebut berpotensi menurunkan standar SLA (*Service Level Agreement*) ketepatan waktu yang berdampak pada kepuasan pelanggan. Hal ini didukung oleh adanya capaian SLA (*Service Level Agreement*) yang ditetapkan perusahaan yaitu sebesar 95% per bulan Juli 2021.

Fenomena ini sejalan dengan penyelesaian masalah yang masih banyak mengandalkan metode manual atau pendekatan konvensional yang kurang efektif dan memakan waktu. Akibatnya, proses distribusi cenderung tidak optimal, dengan biaya operasional yang tinggi serta risiko keterlambatan pengiriman yang lebih besar. Berdasarkan hasil wawancara dengan narasumber, proses penentuan rute distribusi selama ini masih bergantung pada subjektivitas pengemudi dan evaluasi rutin tahunan yang dilakukan oleh perusahaan. Pendekatan ini belum didasarkan pada metode perhitungan yang sistematis dan terotomatisasi, sehingga sering kali

menghasilkan rute yang tidak optimal. Ketidakefisienan dalam penentuan rute tersebut memberikan dampak signifikan terhadap kinerja distribusi, salah satunya tercermin dari besarnya biaya penalti akibat keterlambatan pengiriman.

Permasalahan *Vehicle Routing Problem* memiliki banyak variasi tergantung faktor pembatas dan tujuan yang akan dicapai, salah satunya adalah *Capacitated Vehicle Routing Problem with Time Windows* (CVRP-TW). CVRP-TW merupakan salah satu varian VRP yang dilakukan dengan mempertimbangkan keterbatasan kapasitas kendaraan serta harus dilakukan dalam rentang waktu tertentu (*time windows*). Pada permasalahan ini, seluruh kendaraan yang digunakan berasal dari satu depot dan memiliki karakteristik yang seragam, baik dari segi jenis, bentuk, maupun kapasitas (Lydia, et al., 2011). Kasus CVRP-TW merupakan variasi VRP yang tergolong ke dalam *Non-Polynomial Hard* (*NP-Hard*) dengan adanya faktor batasan atau *constraint* tambahan yang membuat permasalahan semakin kompleks (Gunawan, 2020). VRP merupakan permasalahan *NP-Hard Combinatorial* yang berarti semakin banyak ukuran permasalahan atau variabelnya, usaha komputasi yang digunakan lebih sulit dan lama seiring meningkatnya ruang lingkup masalah (Rahmatullah, 2019). Pada penelitian ini adalah lingkup titik, *constraint* kapasitas, *time windows* dan lainnya. Permasalahan yang tergolong dalam kategori *Non-Polynomial Hard (NP-Hard*) memerlukan algoritma yang efisien dan efektif dalam menyelesaikan permasalahan kompleks yang terjadi di PT. Pos Logistik Indonesia (Soenandi, et al., 2019).

Dalam penyelesaian masalah kompleks dan multi-objektif seperti CVRP-TW, berbagai metode metaheuristik telah banyak digunakan. *Simulated Annealing* (SA) bekerja dengan pendekatan probabilistik berbasis mekanika statistik dan umum digunakan pada permasalahan penjadwalan (Utomo & Setiafindari, 2021). *Particle Swarm Optimization (PSO)* juga telah banyak dimanfaatkan untuk kasus serupa dalam penelitian terdahulu (Febiani, et al., 2024). Sementara itu, penelitian Putri Anita Utami menggunakan *Ant Colony Optimization* (ACO) dan *Tabu Search* untuk penjadwalan *job shop* dengan tujuan meminimalkan *makespan*. Adapun *Genetic Algorithm (GA)* telah banyak diaplikasikan untuk berbagai persoalan optimasi seperti routing, penjadwalan, dan transportasi (Tanujaya, et al., 2011).

Penelitian sebelumnya oleh Latifah Fitri NW (2025) yang menggunakan metode *Particle Swarm Optimization* (PSO) menghasilkan total jarak tempuh sebesar 1.982,85 km dengan melibatkan 4 kendaraan. Hasil tersebut menunjukkan bahwa meskipun PSO dapat menyelesaikan permasalahan rute distribusi, namun masih menghasilkan jarak tempuh yang

relatif besar dibandingkan rute eksisting saat ini serta penggunaan kendaraan yang lebih banyak dari yang diharapkan. Kondisi ini menandakan bahwa masih terdapat peluang perbaikan dalam hal efisiensi rute dan pemanfaatan jumlah kendaraan.

Berdasarkan permasalahan yang dihadapi PT. Pos Logistik Indonesia serta keterbatasan metode sebelumnya, diperlukan pendekatan yang lebih sistematis untuk menyelesaikan CVRP-TW. Dalam penelitian ini digunakan pendekatan *Nearest Insertion* untuk membentuk solusi awal rute kendaraan secara konstruktif, yang selanjutnya dioptimalkan menggunakan *Genetic Algorithm* (GA). GA dipilih karena kemampuannya dalam menangani masalah multi-objektif, seperti pengurangan jarak tempuh dan minimisasi keterlambatan, serta kesesuaiannya untuk menyelesaikan permasalahan berskala besar yang tergolong *NP-Hard*. Ketidakefisienan dalam pengelolaan rute distribusi dapat menimbulkan dampak serius bagi PT. Pos Logistik Indonesia, mulai dari meningkatnya biaya operasional hingga turunnya kepuasan pelanggan akibat keterlambatan pengiriman. Oleh karena itu, solusi optimasi rute berbasis GA diharapkan mampu meminimalkan jarak tempuh sekaligus mengurangi keterlambatan dengan memperhatikan *Estimated Time of Arrival* (ETA).

Dengan demikian, penelitian ini bertujuan untuk menerapkan model optimasi berbasis *Nearest Insertion* dan *Genetic Algorithm* (GA) guna menjawab kesenjangan dan meningkatkan efisiensi distribusi logistik di PT. Pos Logistik Indonesia. Ketidakefisienan dalam pengelolaan rute distribusi kendaraan dapat memberikan dampak serius pada PT. Pos Logistik Indonesia, mulai dari peningkatan biaya operasional hingga penurunan tingkat kepuasan pelanggan akibat keterlambatan pengiriman. Oleh karena itu, diperlukan solusi yang mampu mengatasi masalah tersebut yaitu dengan cara meminimalkan jarak dan waktu keterlambatan dengan *ETA*. Selain itu, *Genetic Algorithm* (GA) merupakan salah satu pendekatan metaheuristik berbasis komputasi yang memiliki potensi besar untuk menyelesaikan permasalahan CVRP-TW.

Maka dari itu, dilakukannya penelitian ini diharapkan mampu memberikan solusi yang efektif bagi PT. Pos Logistik Indonesia untuk mengatasi tantangan distribusi, sekaligus berkontribusi pada pengembangan studi kasus CVRP-TW dengan pendekatan *Nearest Insertion* dan *Genetic Algorithm* (GA). Dengan fokus pada pengoptimalan rute yang bertujuan untuk menurunkan jarak tempuh dan mengurangi waktu keterlambatan, diharapkan PT. Pos Logistik Indonesia mampu menghindari kerugian yang berkepanjangan dan menjadi perusahaan 3PL (*Third Party Logistic)* terbaik di Indonesia

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan pada bagian sebelumnya, penelitian ini difokuskan untuk mengetahui persentase penurunan total jarak tempuh yang dapat dicapai melalui penerapan optimasi rute distribusi pada kasus *Capacitated Vehicle Routing Problem with Time Windows* (CVRP-TW) di PT Pos Logistik Indonesia, serta mengukur sejauh mana penerapan optimasi tersebut mampu menurunkan jumlah keterlambatan pengiriman dan total biaya penalti. Rumusan ini diharapkan dapat memberikan gambaran yang jelas mengenai efektivitas optimasi rute dalam meningkatkan efisiensi operasional sekaligus ketepatan waktu pengiriman pada perusahaan.

1.3 Tujuan Penelitian

Tujuan penelitian yang akan dicapai pada penelitian ini adalah sebagai berikut:

1. Mengetahui persentase penurunan total jarak tempuh yang dapat dicapai melalui penerapan metode optimasi rute distribusi CVRP-TW di PT Pos Logistik Indonesia.
2. Mengetahui besarnya penurunan jumlah keterlambatan pengiriman dan total biaya penalti setelah dilakukan optimasi rute distribusi menggunakan pendekatan CVRP-TW di PT Pos Logistik Indonesia.

1.4 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat bermanfaat bagi pihak-pihak terkait. Adapun manfaat yang diperoleh dari penelitian ini sebagai berikut:

1. Bagi perusahaan, penelitian ini memberikan solusi praktis untuk perbaikan permasalahan distribusi sehingga mendukung pengurangan biaya operasional dan waktu keterlambatan pengiriman yang dapat meningkatkan kepuasan pelanggan. Serta memberikan landasan pertimbangan bagi keputusan strategis terkait penentuan dan pengelolaan rute distribusi secara efektif melalui model optimasi berbasis Algoritma *Nearest Insetion* dan *Genetic Algorithm* (GA).
2. Bagi mahasiswa, dari penelitian ini memungkinkan untuk mengembangkan keterampilan riset, pemodelan hingga analisis hasil. Mahasiswa juga terlatih dalam melakukan eksperimen dengan studi kasus nyata melalui pendekatan ilmiah dan sistematis seperti *metaheuristic* pada kasus logistik.

1.5 Batasan Penelitian

Untuk menghindari meluasnya masalah dan mempermudah memahami permasalahan yang akan dibahas maka perlu adanya batasan masalah, yaitu:

1. Penelitian hanya dilakukan pada jalur pendistribusian pada depot-depot Kantor Pos Logistik Indonesia di Jakarta Raya meliputi Jakarta Timur, Jakarta Barat, Jakarta Selatan, Jakarta Utara dan Jakarta Pusat.
2. Penelitian ini hanya akan fokus pada *Capacitated Vehicle Routing Problem with Time Window* (CVRPTW) yang terjadi di PT. Pos Logistik Indonesia. Penelitian ini tidak membahas masalah logistik lainnya yang tidak melibatkan CVRP-TW
3. Penelitian yang dilakukan pada PT. Pos Logistik Indonesia hanya berasal dari data yang diberikan perusahaan dan hanya berdasarkan periode selama 2024.
4. Penelitian ini hanya mempertimbangkan variabel dasar terkait distribusi, seperti biaya operasional, kapasitas kendaraan dan *Time windows*. Faktor-faktor lain seperti kebijakan internal perusahaan, kapasitas *main hub*, faktor cuaca, *traffic* atau kemacetan atau keadaan darurat yang mempengaruhi distribusi barang tidak menjadi fokus penelitian ini.
5. Penelitian ini tidak akan memperhitungkan data eksternal seperti perubahan permintaan pasar, pengaruh regulasi pemerintah, atau perubahan signifikan lainnya yang dapat mempengaruhi distribusi barang.
6. Pada tugas akhir ini hanya memperhatikan berat barang (Kg) dan tidak memperhatikan isi dari masing – masing barang yang didistribusikan.
7. Penelitian ini hanya memfokuskan pada optimasi rute distribusi untuk meminimalkan total jarak tempuh dan biaya penalti keterlambatan, tanpa menghitung secara menyeluruh dampak terhadap biaya operasional total perusahaan seperti bahan bakar, gaji supir, atau perawatan kendaraan.

1.6 Sistematika Penulisan

Dalam penelitian ini terdiri dari beberapa bab yang nantinya sebagai acuan supaya sistematis dalam penulisan penelitian ini, adapun sistematika dalam penulisan penelitian ini sebagai berikut:

BAB I PENDAHULUAN

Pada bab I berisi tentang latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan penelitian ini.

BAB II KAJIAN LITERATUR

Bagian ini berisi tentang kajian induktif dan kajian deduktif yang diambil dari jurnal, buku, dan literatur lainnya yang mendukung dalam penyelesaian penelitian ini

BAB III METODE PENELITIAN

Berisi tentang uraian kerangka penelitian, metode yang digunakan, aplikasi, bahan baku, material, alat, tata cara penelitian, dan data yang digunakan.

BAB IV PENGAMBILAN DAN PENGOLAHAN DATA

Dalam bagian ini akan menjelaskan pengumpulan data dengan prosedur yang sudah ditentukan, kemudian diolah sesuai dengan metode yang digunakan dan disertakan gambar dan grafik yang diperoleh dari hasil penelitian

BAB V PEMBAHASAN

Berisi pembahasan kritis mengenai hasil bab sebelumnya dan belum dipaparkan di bab sebelumnya.

BAB VI KESIMPULAN DAN SARAN

Kesimpulan berisi pernyataan singkat yang ditulis dengan menggunakan urutan angka untuk menjabarkan hasil penelitian yang dilakukan. Kesimpulan harus menjawab rumusan permasalahan dan membuktikan hipotesis yang ada. Saran berisi beberapa rekomendasi pengembangan penelitian lanjutan dengan menggunakan cara, alat ataupun metode lain dengan tujuan untuk memperluas pengembangan ilmu. Selain itu, bagian ini juga berisi saran yang diperlukan jika penelitian lanjutan akan dikembangkan berdasarkan keterbatasan/hambatan yang ditemukan selama penelitian dilakukan. Saran dapat dihasilkan dari pembahasan yang telah dilakukan di bab sebelumnya.

DAFTAR PUSTAKA

Bagian ini memuat sumber literatur yang menjadi rujukan dari penelitian ini, baik berupa jurnal, buku, maupun literasi lainnya yang menunjang.

LAMPIRAN

Berisi lampiran daftar tabel, gambar, atau bagian lainnya yang mendukung penelitian.

BAB II
TINJAUAN PUSTAKA

Pada bagian bab ini akan dijelaskan hasil kajian pustaka yang bersumber dari paper, artikel dan sejenisnya tetang penjelasan dari penelitian yang dilakukan sebelumnya, dengan topik atau tema yang berkaitan dengan penelitian yang akan dilakukan oleh penulis. Bertujuan untuk sebuah acuan dalam pengembangan dan perbaikan dari sebuah produk yang sudah dibuat sebelumnya dan sebagai landasan teori terkait dengan ilmu-ilmu yang mendukung dalam penelitian yang akan dilakukan. Selain itu juga dijelaskan *state of the art* penelitian dan perbedaannya dengan penelitian-penelitian sebelumnya.

Beberapa penelitian terdahulu yang relevan dengan penelitian "Minimasi Jarak dan Keterlambatan Pada Kasus *Capacitated* VRP *with Time Window* Menggunakan *Nearest Insertion* dan *Genetic Algorithm* (Studi Kasus: PT. Pos Logistik Indonesia ini adalah sebagai berikut.

Penelitian yang dilakukan oleh Amelia, Mohamad Yasin, dan Mimiep (2022) mengenai implementasi metode *simulated annealing* (SA) pada kasus VRPSDP mengimplementasikan metode SA untuk menemukan rute optimal yang meminimalkan total jarak tempuh, dengan mempertimbangkan kapasitas kendaraan dan permintaan konsumen. Hasil penelitian menunjukkan bahwa metode SA mampu menghasilkan solusi rute yang lebih optimal dibandingkan dengan metode lain seperti *Insertion Heuristic* dan algoritma *Ant Colony System*, terutama dalam uji coba dengan 7 dan 9 titik. Saran pada penelitian ini yaitu untuk menggunakan iterasi yang besar untuk mendapatkan hasil yang lebih optimal dan mengusulkan pengembangan lebih lanjut dengan menambahkan kendala waktu dan kecepatan kendaraan.

Penelitian yang dilakukan oleh Tedi dan Khairunnisa (2020) mengenai optimasi Distribusi VRP mengimplementasikan metode *Nearest Neighbour* untuk menemukan rute distribusi yang efisien, dengan mempertimbangkan kapasitas kendaraan dan kebutuhan distribusi. Hasil penelitian menunjukkan bahwa pendekatan *Nearest Neighbour* dapat memberikan solusi yang mendekati optimal dengan waktu penyelesaian yang relatif cepat dibandingkan metode konvensional. Kekurangan penelitian terdapat pada metode, karena peneliti menyarankan penggunaan metode heuristik lain untuk meningkatkan hasil optimasi distribusi.

Penelitian yang dilakukan oleh Annisa Indah Pratiwi dkk. (2023) mengenai kasus VRP pada proses pendistribusian Produk Wafer mengimplementasikan Algoritma Genetika untuk menemukan rute distribusi yang optimal, dengan mempertimbangkan jarak antar lokasi, waktu, dan biaya bahan bakar untuk setiap truk. Metode distribusi menggunakan teknik *crossover* dan *one crosspoint crossover*, mutasi, dan seleksi elitisme. Hasil penelitian menunjukkan bahwa Algoritma Genetika dapat memberikan solusi rute distribusi terbaik dengan biaya distribusi terendah.

Penelitian yang dilakukan oleh Siti Nur Afifah Rahmania, dkk. (2023) mengenai kasus VRP untuk distribusi barang mengimplementasikan metode *saving matrix, NN* dan *Insertion* untuk membuat rute dengan kendaraan berkapasitas tertentu yang tiap kendaraan berawal dan berakhir di gudang, dengan syarat pelanggan hanya dikunjungi sekali dan total permintaan tidak melebihi kapasitas kendaraan. Hasil penelitian menunjukkan bahwa metode *Insertion* memberikan solusi rute distribusi paling optimal dengan total jarak sebesar 139,10 km, yang menghasilkan penghematan jarak sebesar 38,26% dari rute aktual dan penghematan biaya distribusi sebesar Rp3.314.932 per bulan.

Penelitian yang dilakukan oleh Anugerah Trisatryo dkk. (2021) pada kasus VRPPD di Trayek Sekunder *Pos Logistic Distribution Center* Tambun mengimplementasikan metode algoritma *Nearest Neighbor* (ANN) dan dua fase algoritma *Multi-Start Adaptive Large Neighborhood Search* (MSALNS). Hasil penelitian menunjukkan bahwa pendekatan ini efektif dalam menghasilkan rute yang lebih efisien dan mengurangi biaya operasional di PDC Tambun.

Penelitian yang dilakukan oleh Cahaya Annisaa Fathonah dkk. (2023) pada kasus minimasi jarak pada Penentuan Rute Pengangkutan Limbah Medis Rumah Sakit mengimplementasikan metode *Vehicle Routing Problem with Multiple Trips and Intermediate Facility* (VRPMTIF) dengan menggunakan algoritma *Nearest Neighbor*. Metode ini mempertimbangkan kondisi nyata dalam pengangkutan limbah medis dengan tujuan mencapai efektivitas dan efisiensi distribusi. Hasil penelitian menunjukkan bahwa pendekatan ini dapat mengoptimalkan rute pengangkutan limbah medis, mengurangi ketergantungan pada pihak ketiga, dan meningkatkan efisiensi pengelolaan limbah medis di Kota Yogyakarta.

Penelitian yang dilakukan oleh Erika Fatma dkk. (2022) mengenai kasus VRP Multi Depot pada rute Pengangkut Limbah Medis mengimplementasikan metode *Vehicle Routing Problem with Time Window* (VRPTW) untuk mengatasi tantangan pengelolaan limbah medis yang meningkat selama pandemi COVID-19. Metode ini mempertimbangkan waktu kerja

petugas dan kapasitas kendaraan dalam pengangkutan limbah medis. Hasil penelitian menunjukkan bahwa pendekatan ini dapat meminimalkan jarak dan biaya transportasi, serta meningkatkan efektivitas sistem pengelolaan limbah medis di berbagai fasilitas kesehatan.

Penelitian yang dilakukan oleh Lukman Adhitama dkk. (2023) mengenai kasus VRP pada Rute Distribusi Bantuan Logistik bertujuan untuk mengidentifikasi rute distribusi optimal bantuan logistik kesehatan dalam menghadapi bencana banjir. Penelitian ini menggunakan algoritma *Simulated Annealing* (SA) untuk membandingkan hasil optimasi rute distribusi tanpa klaster dan dengan klaster berdasarkan kecamatan. Hasil penelitian menunjukkan bahwa penggunaan klaster efektif dalam meminimalkan jarak dan waktu tempuh distribusi, dengan skenario klaster mencapai jarak tempuh 12,9 km dalam 37 menit, dibandingkan dengan rute tanpa klaster yang mencapai 23,8 km. Temuan ini memberikan kontribusi praktis untuk perencanaan tanggap darurat bencana banjir, memastikan distribusi bantuan logistik kesehatan yang efisien dan responsif di daerah rawan banjir Kota Yogyakarta.

Penelitian yang dilakukan oleh Siska Amonalisa Silalahi (2021) mengenai penentuan rute distribusi Limbah Medis Covid-19 yang bertujuan untuk merancang rute distribusi limbah medis Covid-19 yang efisien di Surabaya menggunakan metode *Vehicle Routing Problem with Time Window* (VRPTW) dan algoritma *Nearest Neighbor* untuk menentukan rute optimal yang dapat mengakomodasi seluruh limbah medis dari 14 rumah sakit rujukan Covid-19. Hasil penelitian menunjukkan bahwa dengan menyediakan minimal 6 kendaraan homogen, kinerja distribusi limbah medis dapat dinilai baik, dengan semua limbah medis terangkut secara efisien. Temuan ini memberikan kontribusi penting dalam pengelolaan limbah medis, terutama dalam konteks penanganan pandemi, memastikan bahwa limbah medis dikelola dengan baik dan sesuai dengan regulasi yang ada.

Penelitian yang dilakukan oleh Ila Asih Setiani dan Lukmandono (2021) mengenai optimasi rute distribusi obat untuk meminimalkan biaya transportasi bertujuan untuk mengatasi masalah distribusi obat di Clinisindo Putra Perkasa Inc menggunakan metode *Saving Matrix* dan *Nearest Neighbor* untuk menentukan rute distribusi yang efisien, sehingga dapat mempercepat pengiriman produk kepada pelanggan dan meminimalkan biaya transportasi. Hasil penelitian menunjukkan bahwa dengan penerapan kedua metode tersebut, rute distribusi dapat dioptimalkan, yang berkontribusi pada peningkatan kepuasan pelanggan dan efisiensi biaya. Temuan ini memberikan kontribusi penting dalam pengelolaan distribusi obat,

memastikan bahwa produk dapat sampai ke tangan pelanggan dengan cepat dan biaya yang lebih rendah.

Penelitian yang dilakukan oleh Mochammad Chaerul dkk. (2022) mengenai Optimasi Rute Pengangkutan Sampah di Kabupaten Manokwari bertujuan untuk mengatasi masalah pengangkutan sampah yang tidak efisien di Kabupaten Manokwari, Papua menggunakan metode *Nearest Neighbor* untuk menentukan rute pengangkutan yang optimal, sehingga dapat mengurangi waktu pengangkutan dan jumlah truk yang diperlukan. Hasil penelitian menunjukkan bahwa dengan penerapan metode ini, waktu pengangkutan sampah dapat dioptimalkan dari 3.289 menit per hari menjadi 3.038 menit per hari, menghemat 251 menit setiap harinya dan mengurangi jumlah truk dari 15 menjadi 13. Temuan ini memberikan kontribusi penting dalam pengelolaan sampah, membantu pemerintah daerah dalam meningkatkan efisiensi sistem pengangkutan dan mengurangi biaya operasional. Penelitian ini juga menyoroti pentingnya penelitian lebih lanjut dalam pengelolaan rute pengangkutan sampah di Indonesia, yang sering kali bergantung pada pengalaman empiris pengemudi.

Penelitian yang dilakukan oleh Deti Saskia Dela Ferdiani dkk. (2022) ini berokus penentukan rute pengangkutan sampah yang paling optimal di Kecamatan Sungai Kunjang, Kota Samarinda. Penelitian ini menggunakan metode *Vehicle Routing Problem* untuk meminimalkan total jarak tempuh semua kendaraan, sehingga dapat menghemat waktu dan bahan bakar. Hasil penelitian menunjukkan bahwa rute optimasi *Nearest Neighbors* mampu menghemat jarak sepanjang 123,74 km per rit, menghemat waktu tempuh 4,26 jam per rit, dan menghemat kebutuhan bahan bakar sebanyak 24,75 liter per rit. Temuan ini memberikan kontribusi penting dalam pengelolaan sampah, membantu Dinas Lingkungan Hidup Kota Samarinda dalam meningkatkan efisiensi sistem pengangkutan dan mengurangi biaya operasional. Penelitian ini juga menyoroti pentingnya penelitian lebih lanjut dalam optimasi rute pengangkutan sampah di daerah lain di Indonesia.

Penelitian yang dilakukan oleh Ekky Armandi dkk. (2019) bertujuan untuk menentukan rute pengangkutan sampah yang optimal di Kota Yogyakarta. Penelitian ini mengembangkan model *Vehicle Routing Problem* (VRP) dengan mempertimbangkan batasan *heterogeneous fleet, multiple trips, intermediate facility,* dan *split delivery*, serta menggunakan metode *hybrid genetic algorithm* (GA) untuk menyelesaikan masalah tersebut. Hasil penelitian menunjukkan bahwa model VRP yang dikembangkan dapat digunakan untuk menyelesaikan persoalan pengangkutan sampah di Kota Yogyakarta. Temuan ini memberikan kontribusi penting dalam

pengelolaan sampah, membantu Dinas Lingkungan Hidup Kota Yogyakarta dalam meningkatkan efisiensi sistem pengangkutan dan mengurangi biaya operasional. Penelitian ini juga menyoroti perlunya penelitian lebih lanjut dalam optimasi rute pengangkutan sampah di daerah lain.

Penelitian yang dilakukan oleh Rully Rumaida dkk. (2024) ini bertujuan untuk mencari rute pengangkutan sampah yang optimal di Kota Padang Sidempuan. Penelitian ini menerapkan metode Algoritma *Tabu Search* untuk menyelesaikan masalah *Capacitated Vehicle Routing Problem* (CVRP) dengan mempertimbangkan data dari Dinas Lingkungan Hidup dan Kebersihan (DLHK) mengenai jenis kendaraan, lokasi RPS, jenis TPS, dan volume sampah. Hasil penelitian menunjukkan bahwa Algoritma *Tabu Search* dapat digunakan untuk menentukan rute optimal dalam pengangkutan sampah, yang berkontribusi pada efisiensi sistem pengangkutan dan pengelolaan sampah di Kota Padang Sidempuan. Penelitian ini juga menekankan pentingnya pengumpulan data yang akurat dan analisis yang mendalam untuk meningkatkan hasil optimasi rute pengangkutan sampah.

Penelitian yang dilakukan oleh Y Yogaswara dan F F Fatin (2020) ini bertujuan untuk menentukan rute dan jadwal pengangkutan sampah yang efisien di Kota Bandung dan wilayah Bandung Barat. Penelitian ini menerapkan metode Algoritma *Tabu Search* untuk menyelesaikan masalah penjadwalan dan penentuan rute pengangkutan sampah, dengan mempertimbangkan data dari PD. Kebersihan Kota Bandung. Hasil penelitian menunjukkan bahwa Algoritma *Tabu Search* dapat digunakan untuk mengoptimalkan rute dan jadwal pengangkutan sampah, yang berkontribusi pada peningkatan efisiensi dalam sistem pengangkutan dan pengelolaan sampah di daerah tersebut. Penelitian ini juga menekankan pentingnya pengumpulan data yang tepat dan analisis yang komprehensif untuk meningkatkan hasil optimasi rute pengangkutan sampah.

Tabel 2. 1 Posisi Penelitian

Penulis	**Obyek**	**Metode**	**Min. Total Jarak**	**Min. *Lateness***
(Aida, et al., 2022)	Logistik	SA	✓	
(Dahniar & Khairunnisa, 2020)	Logistik	NN	✓	
(Pratiwi, et al., 2023)	Logistik	GA	✓	
(Rahmania, et al., 2023)	Logistik	NN	✓	

Penulis	Obyek	Metode	Min. Total Jarak	Min. *Lateness*
(Trisatryo, et al., 2021)	Logistik	NN, ALNS	✓	
(Annisaa, et al., 2023)	Dist. Medis	NN	✓	
(Fatma, et al., 2022)	Dist. Medis	ALNS	✓	
(Adhitama, et al., 2023)	Dist. Medis	SA	✓	✓
(Silalahi, 2021)	Dist. Medis	NN	✓	
(Setiani & Lukmandono, 2021)	Dist. Medis	NN	✓	
(Chaerul, et al., 2022)	Dist. Sampah	NN	✓	✓
(Ferdiani, et al., 2022)	Dist. Sampah	NN	✓	
(Armandi, et al., 2019)	Dist. Sampah	GA	✓	
(Rumaida, et al., 2024)	Dist. Sampah	Tabu Search	✓	
(Yogaswara & Fatin, 2020)	Dist. Sampah	Tabu Search	✓	
Usulan	**Logistik**	**GA + NI**	✓	✓

Perbedaan antara penelitian ini dengan penelitian terdahulu yaitu penelitian ini memiliki permasalahan spesifik pada metode penyelesaiannya. Permasalahan yang ada menjadikan penelitian ini membutuhkan metode khusus karena permasalahan yang kompleks dan multi-objektif. Sementara itu, PT. Pos Logistik Indonesia belum cukup memahami kondisi seperti ini dan masih menggunakan cara kovensional dalam melakukan perutean kendaraan. Hal tersebut memerlukan tekonologi yang bisa membantu meningkatkan efisiensi perusahaan dalam melakukan layanan distribusi produk. Selain itu permasalahan utama seperti terdapat nya *constraint* tambahin seperti *time windows* yang memberikan urgensi penelitian untuk menghasilkan rute optimal yang bebas keterlambatan juga menjadikan perbedaan penelitian ini dari penelitian terdahulu.

2.1 *Supply Chain & Supply Chain Management*

Supply Chain merupakan sebuah jaringan perusahaan yang bekerja sama dengan tujuan memproduksi dan mendistribusikan produk (hulu) hingga sampai ke tangan konsumen akhir (hilir). Perusahaan-perusahaan yang terlibat dalam rantai pasokan ini meliputi *supplier*, pabrik, distributor, pengecer, serta pihak pendukung seperti layanan logistik (*provider*) (Sucahyowati, 2011). Terdapat tiga hal utama yang perlu dikelola dalam *supply chain* menurut (Sucahyowati, 2011), yaitu:

1. Aliran barang yang bergerak dari hulu ke hilir, seperti pengiriman bahan baku dari pemasok ke pabrik, distribusi produk setelah produksi ke distributor dan pengecer, hingga sampai ke konsumen akhir.
2. Aliran uang atau pembayaran yang mengalir dari hilir ke hulu.
3. Aliran informasi yang dapat bergerak baik dari hulu ke hilir maupun sebaliknya. Rantai pasokan adalah serangkaian elemen, seperti pemasok, titik distribusi, dan penyedia transportasi, yang berfungsi untuk mengubah bahan baku menjadi produk yang siap digunakan oleh konsumen akhir.

Sedangkan, *Supply Chain Management* (SCM) adalah istilah yang digunakan untuk mengendalikan dan mengatur rantai pasokan. Jika *Supply Chain* adalah jaringannya, *Supply Chain Management* SCM) adalah metode atau alat dalam pengelolaannya (Sucahyowati, 2011). *Supply Chain Management* (SCM) mencakup pengelolaan persediaan barang dan pengaturan distribusinya hingga produk selesai dikirimkan ke pengecer dan akhirnya sampai ke konsumen akhir.

2.1.1 Komponen *Supply Chain Management*

SCM melibatkan koordinasi yang erat antar semua pihak dalam rantai pasokan untuk memastikan efisiensi dan efektivitas dalam setiap tahap, mulai dari perencanaan hingga pengiriman produk yang tepat waktu dan dengan kualitas yang sesuai (Yusuf & Soediantono, 2022). Sebuah model rantai pasokan sederhana terdiri dari empat komponen:

1. *Supplier*

   Meliputi proses persediaan bahan baku mulai dari pengadaan dan pengiriman ke perusahaan (hulu).

2. Produsen/manufaktur

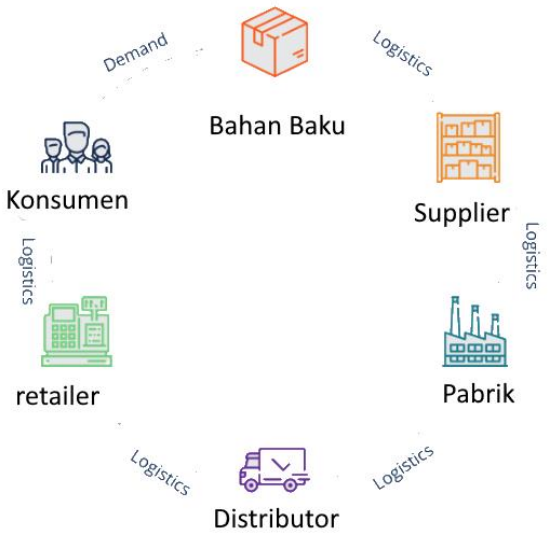
   Merupakan proses menghasilkan produk yaitu merubah bahan baku menjadi produk jadi yang akan di distribusikan baik ke consumer maupun pengecer.

3. Gudang atau Pusat Distribusi (*retailer*)

   Meliputi toko-toko pengecer dan kapal-kapal produk yang membantu proses distribusi produk ke konsumen akhir atau pelanggan.

4. Pengguna Akhir

   Menerima produk (hilir).



Gambar 2. 1 Skema Alur SCM

Sumber: (Sunardi, 2020)

Dari skema alur SCM menurut (Sunardi, 2020) tersebut, setiap elemen dalam proses rantai pasok tidak terlepas dari peran logistik yang berfungsi dalam proses distribusi antar komponen rantai pasok. Dalam *Supply Chain Management* dikenal sebuah kegiatan, yaitu

logistik. Logistik meliputi kegiatan seperti pergudangan (*warehouse*), distribusi barang (*distribution*), transportasi barang (*freight transportation*), dan pengelolaan pesanan (*sales order processing*) (Haryotejo & Kusumawardhani, 2015).

2.1.2 Alat Fungsionalitas *Supply Chain Management*

Pengapliksian *Supply Chain Management* yang baik mampu bersifat menguntungkan bagi perusahaan, terlebih pada perkembangan teknologi saat ini memungkinkan sebuah perusahaan menerapkan SCM untuk menanggapi perpindahan produk dengan cepat. Hal tersebut dapat didukung dengan menerapkan beberapa peralatan fungsional dalam *Supply Chain Management* seperti (Widyarto, 2012):

1. *Demand management / forecasting*
2. *Advanced planning and scheduling*
3. *Transportation management*
4. *Distribution and deployment*
5. *Production planning*
6. *Available to promise*
7. *Supply Chain Modeler*
8. *Optimizer (Linier programming, non linier programming, heuristic, dan genetic algorithm).*

2.2 Logistik

Logistik merupakan suatu kegiatan penting dalam bentuk memanfaatkan sumber daya manusia dan material secara luas. Logistik bertujuan untuk memastikan barang atau jasa didistribusikan tepat waktu, sesuai jenis, dan pada lokasi yang benar, sekaligus mengoptimalkan waktu dan biaya agar operasional perusahaan dapat berjalan lebih efisien (Pratiwi, et al., 2023). Disamping itu, logistik juga berperan dalam mempengaruhi ekonomi nasional yang mencakup berbagai hal seperti pemeliharaan, penyaluran, penyimpanan dan pengiriman (Sutini & Wismana, 2021).

Dalam pelaksanaan logistik, terdapat proses yang sangat penting yaitu perpindahan barang yang perlu dilakukan pada titik-titik simpul, baik antar moda transportasi dengan kegiatan pergudangan (*warehousing*) yang mana perpindahan barang tersebut dari gudang dan atau tempat pengambilan barang diangkut dengan sarana transportasi darat ke pelabuhan/bandara selanjutnya dimuat ke sarana pengangkut utama seperti kapal atau pesawat dan dari pelabuhan/bandara diangkut dengan sarana transportasi darat ke gudang dan atau ke

tempat tujuan akhir barang sesuai dengan perjanjian dan waktu yang telah disepakati (Sutini & Wismana, 2021). Logistik juga merupakan bagian dari proses *supply chain* yang merencanakan, mengimplementasikan dan mengontrol keefisienan dan keefektifan aliran dan penyimpanan dari suatu produk dari *origin point* hingga ke konsumsi terakhir dalam rangka untuk memenuhi permintaan *customer.*

2.2.1 Komponen Sistem Logistik

Kemudian berjalannya sistem logisitik yang baik tentu saja membutuhkan komponen-komponen pendukung. Menurut (Rahmiyati & Irianto, 2021), terdapat 5 komponen dalam sistem logistik, diantaranya adalah:

1. Struktur lokasi fasilitas

   Jaringan fasilitas perusahaan mencakup berbagai lokasi yang digunakan untuk mengangkut dan mendistribusikan material serta produk. Dalam perencanaan, fasilitas struktur lokasi biasanya meliputi pabrik, gudang, dan toko pengecer (*supplier*). Jika perusahaan menggunakan layanan tambahan seperti jasa pengangkutan atau penyimpanan, fasilitas tersebut menjadi bagian penting dari jaringan operasional.

2. Transportasi

   Kecepatan layanan transportasi mengacu pada waktu yang diperlukan untuk menyelesaikan proses pengangkutan. Hal ini berhubungan dengan kemampuan transportasi dalam memberikan layanan yang cepat, meskipun seringkali disertai dengan tarif yang lebih tinggi. Semakin cepat layanan yang diberikan, semakin singkat pula waktu yang dibutuhkan dalam proses produksi barang.

3. Persediaan

   Pengadaan material dalam sistem logistik dilakukan dengan tujuan yang berbeda dari pengadaan produk jadi. Dengan menggunakan penjadwalan *Material Requirement Planning* (MRP), fokus utama pengadaan material dan persediaan adalah memastikan kelancaran jadwal produksi dan diwaktu bersamaan juga meminimalkan kebutuhan pengadaan persediaan.

4. Komunikasi

   Komunikasi memegang peranan penting dalam sistem logistik dan tidak dapat diabaikan. Kecepatan aliran informasi memiliki hubungan langsung dengan integrasi antara fasilitas, transportasi, dan persediaan. Semakin efisien desain sistem logistik

suatu perusahaan, semakin sensitif pula sistem tersebut terhadap gangguan dalam aliran informasi.

5. Penanganan (*handling*) dan Penyimpanan

   Penanganan dan penyimpanan dalam arti luas mencakup pergerakan (*movement*), pengepakan (*packing*), dan pengemasan (*containerization*). Semakin sedikit produk yang ditangani, semakin efisien alur fisiknya secara keseluruhan. Jika proses penanganan ini diintegrasikan dengan baik, dapat membantu mempercepat dan mempermudah alur kerja dalam sistem tersebut.

2.2.2 *Logistic Service Provider*

Dalam dunia logistik terdapat beberapa jenis penyedia (*provider*) sesuai dengan tujuannya yaitu untuk proses pendistribusian kegiatan operasional perusahaan. Beberapa jenis penyedia jasa logistik tersebut diantaranya (Puspitorini & Sabdoadi, 2019):

1. *The First Party Logistic* (1PL)

   The First Party Logistic atau dikenal dengan 1PL merupakan perusahaan dengan aktivitas logistik dan distribusi dikelola secara internal oleh bisnis, tetapi sebagian besar bisnisnya saat ini dialihkan ke penyedia eksternal. Contohnya adalah *supplier* daging merah Australia, Samex mengekspor ke *wholesaler*, distributor dan supermarket ke seluruh dunia dengan operasi logistiknya sendiri.

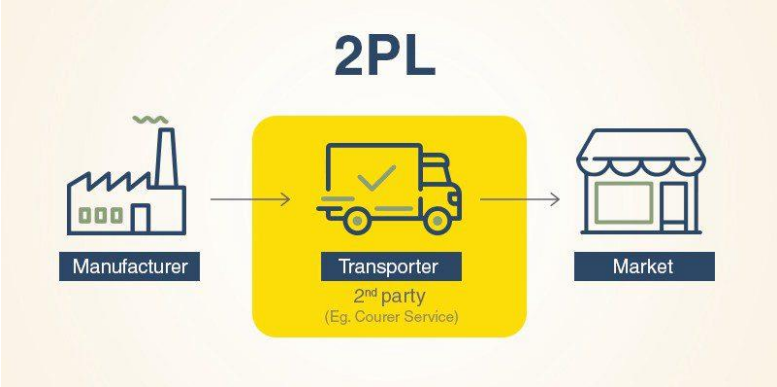


Gambar 2. 2 *The First Party Logistic*

Sumber: navata.com

2. *The Second Party Logistic* (2PL)

   The Second Party Logistic (2PL) merupakan pengelolaan *supply chain* secara eksternal (dari gudang ke dermaga) tetapi lebih menyukai me-*manage* bagian lain *supply chain* secara internal.

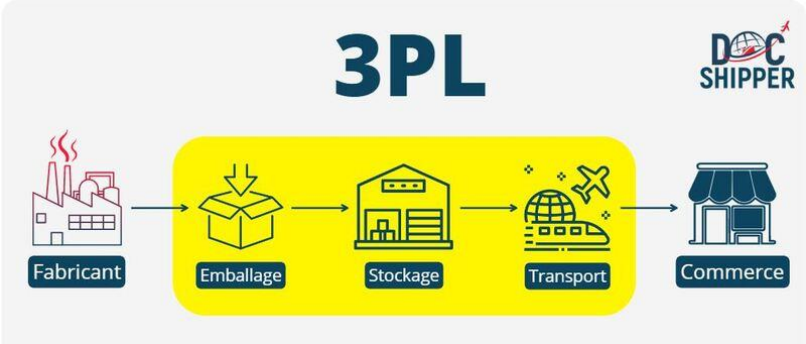


Gambar 2. 3 *The Second Party Logistic*

Sumber: navata.com

3. *The Third Party Logistic* (3PL)

   The Third Party Logistic (3PL) merupakan perusahaan *outsourcing* sebagian atau seluruh fungsi manajemen *supply chain* terdiri dari (i) manajemen dan penyimpanan persediaan, (ii). *Picking dan packing*, (iii) pengiriman barang (*freight forwarding*), (iv) distribusi, (v) kontrak managemen, (vi) *cross docking*, (vii) solusi IT dan (viii) *custom brokerage*.

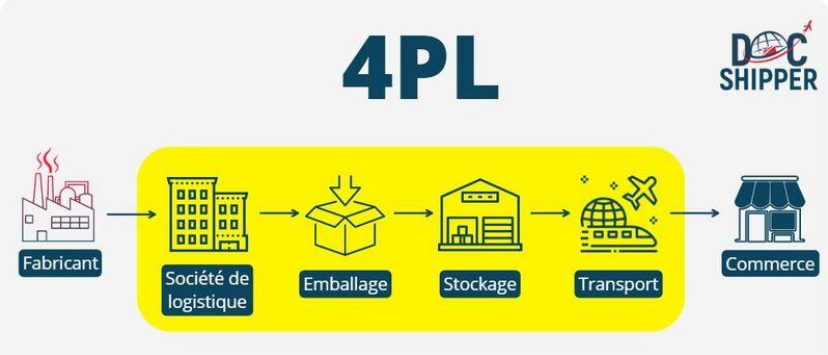


Gambar 2. 4 *The Third Party Logistic*

Sumber: FNM Malaisie

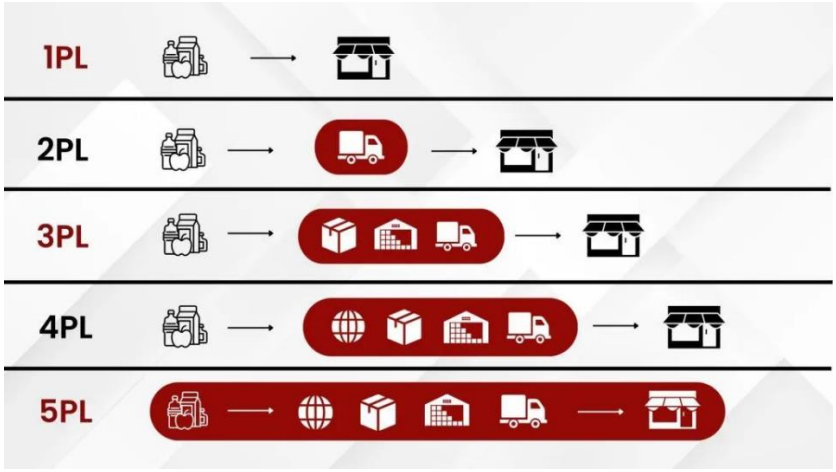
4. *The Fourth Party Logistic* (4PL)

   The Fourth Party Logistic (4PL) merupakan aktivitas 3PL dengan bisnis inti mengelola sumber daya, teknologi dan infrastruktur selain menyediakan solusi *supply chain* disamping merancang dan mendesain *supply chain*. Aktivitas bisnisnya terdiri dari (i) strategi logistik, (ii). *Inbound, outbound* dan *reverse logistic*, (iii) perencanaan bisnis, (iv) *project management*, (v) analisis dan desain jaringan, (vi) *inventory planning* dan *management,* (vii) strategi pengiriman dan (viii) analisis transportasi, penggunaan kapasitas dan kinerja operator.



Gambar 2. 5 *The Fourth Party Logistic*

Sumber: FNM Malaisie

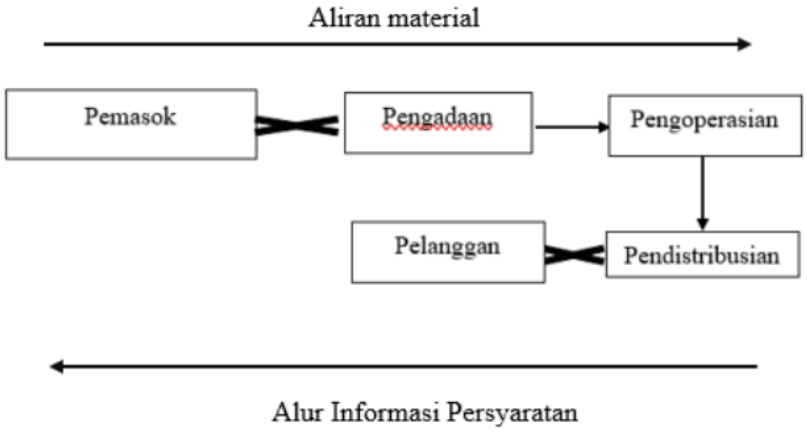


Gambar 2. 6 *Logistic Providers*

Sumber: *godamwale.com*

2.3 Manajemen Logistik

Manajemen logistik adalah bagian dari *supply chain* sebagai upaya pemenuhan kebutuhan dalam proses perencanaan, pelaksanaan, pengendalian dan penyimpanan (Anggaraeni & Zainafree, 2022). Manajemen logistik mencakup serangkaian aktivitas yang meliputi perencanaan, pengorganisasian, dan pengawasan terhadap seluruh proses pengadaan, pencatatan, distribusi, penyimpanan, serta pemeliharaan barang. Tujuan utama dari manajemen logistik adalah memastikan ketersediaan barang yang sesuai dengan kebutuhan secara tepat waktu, tepat tempat, dan tepat jenis, sehingga mendukung tercapainya tujuan organisasi secara efektif dan efisien (Farhan & Hidayat, 2021). Dapat dikatakan manajemen logistik suatu perusahaan akan mencerminkan kinerja perusahaan tersebut. Semakin efektif manajemen logistik suatu perusahaan makan akan semakin baik juga kinerjanya (Sunaryo, et al., 2021).



Gambar 2. 7 Peran Manajemen Logistik

2.3.1 Unsur dan Asas Manajemen Logistik

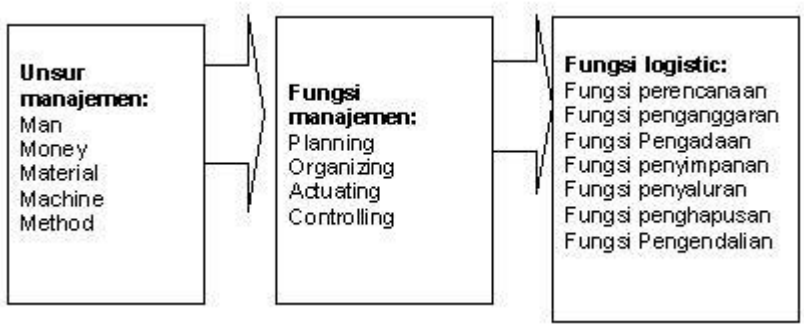
Dalam manajemen logistic terdapat 5 unsur utama yang biasa diketahui dengan 5M. Berikut ini merupakan unsur-unsur yang ada dalam manajemen logistik menurut (Mustafa, 2017):

1. Manusia (*Man*);
2. Uang/dana *(Money*);
3. Bahan-bahan (*Material*);
4. Mesin (*Machine*), dan
5. Cara/metode (*Method*).

Unsur dalam manajemen logistik tersebut merupakan dasar dari proses pelaksaan manajemen logistik itu sendiri yang di representasikan dalam asas-asas atau prinsip dasar yang mendasari manajemen logistik tersebut. Berikut merupakan asas-asas dalam manajemen logistik:

1. Koordinasi
2. Integrasi
3. Sinkronisasi
4. Simplikasi, keempat dari asas-asas manajemen logistik tersebut yang menjadi dasar dari norma-norma logistik yang mengatur pelaksanaan fungsi-fungsi manajemen logistik.

2.3.2 Fungsi-fungsi Manajemen Logistik

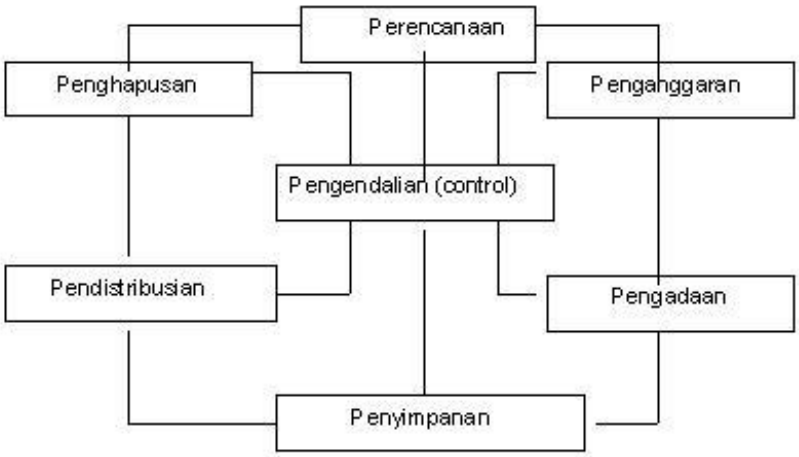


Gambar 2. 8 Fungsi Manajemen Logistik

Sumber: (Urrahman, 2009)

Pelaksanaan manajemen yang baik disusun atas unsur-unsur manajemen yang diproses melalui fungsi-fungsi manajemen dan fungsi tersebut merupakan pegangan umum untuk dapat terselenggaranya fungsi-fungsi logistik. Berikut merupakan fungsi-fungsi logistik dikutip dari (Urrahman, 2009):

1. Perencanaan adalah proses untuk menetapkan tujuan, pedoman, dan kriteria yang menjadi dasar dalam pengelolaan perlengkapan secara efektif.
2. Pengadaan melibatkan pemenuhan kebutuhan barang atau jasa dengan memastikan kualitas yang optimal dan harga yang efisien.
3. Penyimpanan berkaitan dengan pengelolaan barang agar dapat diakses dengan cepat dan tepat saat dibutuhkan.
4. Pendistribusian mencakup pengaturan dan pengelolaan pemindahan barang dari tempat penyimpanan ke lokasi yang membutuhkan, memastikan kelancaran aliran barang.
5. Penghapusan merupakan kegiatan untuk mengevaluasi dan menghapus barang yang sudah tidak diperlukan, sehingga dapat dikeluarkan dari sistem pengelolaan material.
6. Pengendalian berfungsi untuk memastikan bahwa semua kegiatan berjalan sesuai dengan rencana, dengan menggunakan umpan balik untuk mengevaluasi dan memastikan pencapaian tujuan yang telah ditetapkan.



Gambar 2. 9 Siklus Sistem Manajemen Logistik
Sumber: (Urrahman, 2009)

2.4 Transportasi dan Distribusi

Menurut (Pujawan & Mahendrawati, 2010), transportasi dan distribusi melibatkan proses pemindahan produk dari tempat produksinya ke lokasi konsumen atau pengguna, yang seringkali terhalang oleh jarak yang cukup jauh. Distribusi dan transportasi yang baik merupakan suatu hal yang penting dalam suatu perusahaan agar suatu produk dapat dikirim sampai kepada konsumen tepat waktu, tepat pada tempat yang ditentukan, dan barang dalam kondisi baik (Rahmah & Ningrat, 2020). Proses distribusi dan transportasi akan menentukan apakah suatu perusahaan dapat memuaskan setiap pelanggan. Salah satu faktor penting yang dapat mempengaruhi permasalahan ialah rute distribusi yang dilalui alat transportasi dalam mengirim bahan baku. Biaya-biaya pengiriman akan sangat berpengaruh dari keputusan rute pengiriman dilalui oleh setiap kenderaan. Salah satu cara yang dapat dilakukan agar proses pendistribusian produk dapat berlangsung dengan lebih efektif dan juga efisien adalah dengan cara membuat perencanaan rute distribusi secara tepat agar produk tersebut dapat dikirimkan kepada pelanggan dengan waktu yang sudah ditentukan tanpa keterlambatan sedikitpun (Purnomo, 2019).

Menurut (Nasution, 2008), terdapat 6 unsur dalam pengangkutan dalam konteks transportasi, diantaranya sebagai berikut:

1. Muatan yang diangkut
2. Alat angkut

3. Jalan/jalur yang akan dilalui
4. Terminal asal
5. Terminal tujuan
6. Sumber daya manusia dan organisasi atau manajemen yang menggerakkan kegiatan tersebut.

2.4.1 Ruang Lingkup Transportasi

Dalam transportasi terdapat *stakeholder* yang berperan sebagai penggerak aktivitas transportasi dan distribusi. Menurut *Asscociation for Supply Chain Management (*ASCM) pada *Planning Inventory Management* (PIM), menyebutkan terdapat 5 *Stakeholder* dalam transportasi seperti pada gambar berikut:



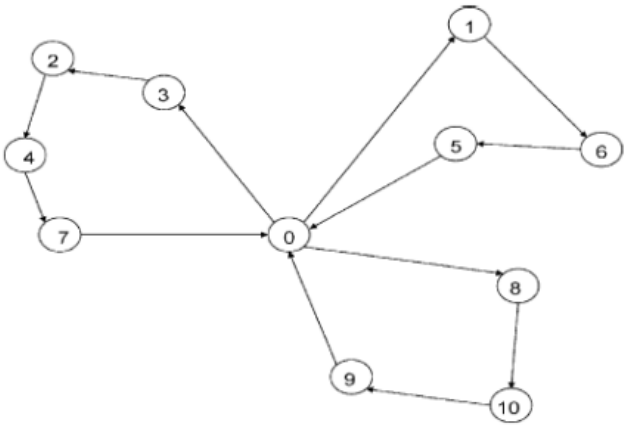
Gambar 2. 10 *Stakeholder* Transportasi
Sumber: (Fauzi & Hartati, 2021)

2.5 *Vehicle Routing Problem* (VRP)

Vehicle Routing Problem (VRP) merupakan suatu metode dalam menentukan rute pengiriman dengan tujuan meminimasi biaya transportasi, total jarak serta waktu tempuh kendaraan (Hajar, et al., 2024). *Vehicle routing problem* (VRP) merupakan perluasan dari *traveling salesman problem* (TSP) sehingga VRP sering disebut sebagai m-TSP (Andani, 2023). VRP mendefinisikan permasalahan penentuan rute distribusi dimana sebuah kendaraan yang mendatangi beberapa titik dan tiap titik tersebut hanya dapat dikunjungi oleh 1 kendaraan saja dimana tiap kendaraan berasal dari depot utama (*main hub*) dan harus kembali ke depot utama (*main hub*) tersebut pada akhir perjalanannya (Anshori, 2019). Dalam VRP, terdapat beberapa asumsi yang sering digunakan yaitu kendaraan diasumsikan memiliki jumlah kapasitas yang

sama, jumlah kendaraan yang tidak terbatas, jumlah permintaan tiap pemberhentian (*node*) diketahui, serta tidak ada jumlah permintaan tunggal yang melebihi kapasitas kendaraan (Sugiono, 2022). Solusi dari sebuah VRP yaitu menentukan sejumlah rute, yang masing-masing dilayani oleh suatu kendaraan yang berasal dan berakhir pada depotnya, sehingga kebutuhan pelanggan terpenuhi, semua permasalahan operasional terselesaikan dan biaya transportasi secara umum diminimalkan (Fitriastutik, 2020). Berikut ini merupakan tujuan umum dari *Vehicle Routing Problems*, diantaranya adalah:

1) Meminimalkan biaya transportasi global, terkait dengan jarak dan *fixed cost* yang berhubungan dengan kendaraan.
2) Meminimalkan jumlah kendaraan (atau pengemudi) yang dibutuhkan untuk melayani semua permintaan
3) Menyeimbangkan rute, termasuk waktu perjalanan dan muatan kendaraan.
4) Meminimalkan penalti akibat servis yang kurang memuaskan dalam upaya memenuhi permintaan.



Gambar 2. 11 Contoh Solusi VRP
Sumber: Bell & McMullen (2004)

Menurut (Kurniawan, et al., 2014), terdapat beberapa variasi VRP yang sangat bergantung pada jumlah faktor pembatas dan tujuan yang akan dicapai. Pembatas yang paling umum digunakan yaitu jarak dan waktu. Tujuan yang ingin dicapai biasanya minimasi jarak tempuh, waktu maupun biaya. Berikut merupakan contoh beberapa klasifikasi dalam kasus *Vehicle Routing Problem* (VRP):

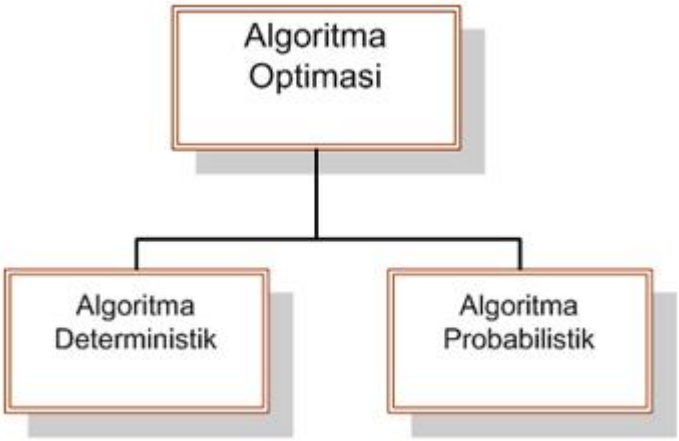
5) VRP *with multiple trips*: setiap kendaraan dapat melakukan lebih dari satu rute untuk memenuhi kebutuhan pelanggan.

6) VRP *with time windows*: setiap pelanggan yang dilayani oleh kendaraan memiliki waktu *service*.
7) VRP *with pickup and delivery*: terdapat sejumlah barang yang perlu dipindahkan dari lokasi penjemputan tertentu ke lokasi pengiriman lainnya.
8) *Capacitated* VRP: kendaraan yang memiliki keterbatasan daya angkut (kapasitas) barang yang harus diantarkan kesuatu tempat.
9) VRP *with Multiple Products*: pelanggan memiliki pesanan lebih dari satu jenis produk yang harus diantarkan.
10) VRP *with Multiple Depots*: depot awal untuk melayani pelanggan lebih dari satu.
11) *Periodic* VRP: adanya horison perencanaan yang berlaku untuk satuan waktu tertentu.
12) VRP *with heterogeneous fleet of vehicles*: kapasitas kendaraan antar kendaraan satu dengan kendaraan lain tidak selalu sama. Jumlah dan tipe kendaraan diketahui.

2.6 Algoritma Optimasi

Optimasi merupakan suatua cabang dalam ilmu matematika yang berkaitan dengan pencarian nilai maksimum atau minimum dari suatu fungsi tujuan, dengan atau tanpa kendala (Area, 2024). Optimasi bertujuan untuk memperoleh hasil optimal dari suatu permasalahan yang diarahkan pada titik maksimum atau minimum suatu fungsi tujuan dengan mempertimbangkan Batasan-batasan tertentu (Chairunisah & Sihotang, 2021). Penemuan solusi terbaik pada proses optimasi dilakukan dari proses pemilihan solusi terbaik dari sejumlah alternatif yang tersedia (Zahara, et al., 2024). Penemuan solusi terbaik tersebut, dibutuhkan suatu teknik pengolahan data masukan supaya memperoleh hasil berupa *fitness*, yang mana hasil nilai *fitness* dijadikan sebagai acuan untuk memperoleh solusi yang maksimum (Dinia, et al., 2021).

Algoritma Optimasi (*Optimization Algorithms)* dapat didefinisikan sebagai algoritma yang digunakan untuk menemukan nilai x sedemikian sehingga menghasilkan f(x) yang bernilai "sekecil" atau "sebesar" mungkin untuk suatu fungsi f yang diberikan, yang mungkin disertai dengan beberapa batasan pada x. Dimana x dapat berupa scalar atau vector dari nilai-nilai kontinu maupun diskrit. Algoritma optimasi sedikit berbeda dengan algoritma pencarian (*search algorithms*). Pada algoritma pencarian terdapat suatu kriteria tertentu yang menyatakan apakah elemen xi merupakan solusi atau bukan. Sebaliknya, pada algoritma optimasi mungkin tidak terdapat kriteria tersebut melainkan hanya fungsi-fungsi objektif yang menggambarkan bagus tidaknya suatu konfigurasi yang diberikan (Hasad, 2011).



Gambar 2. 12 Pengelompokkan Algoritma Optimasi

Sumber: andihasad.com

2.6.1 Klasifikasi AO Berdasarkan Metode Operasi

Menurut (Hasad, 2011), berdasarkan metode operasinya, algoritma optimasi dibagi ke dalam dua kelas besar, yaitu algoritma deterministik (*deterministic*) dan probabilistik (*probabilistic*).

1. Algoritma Deterministik

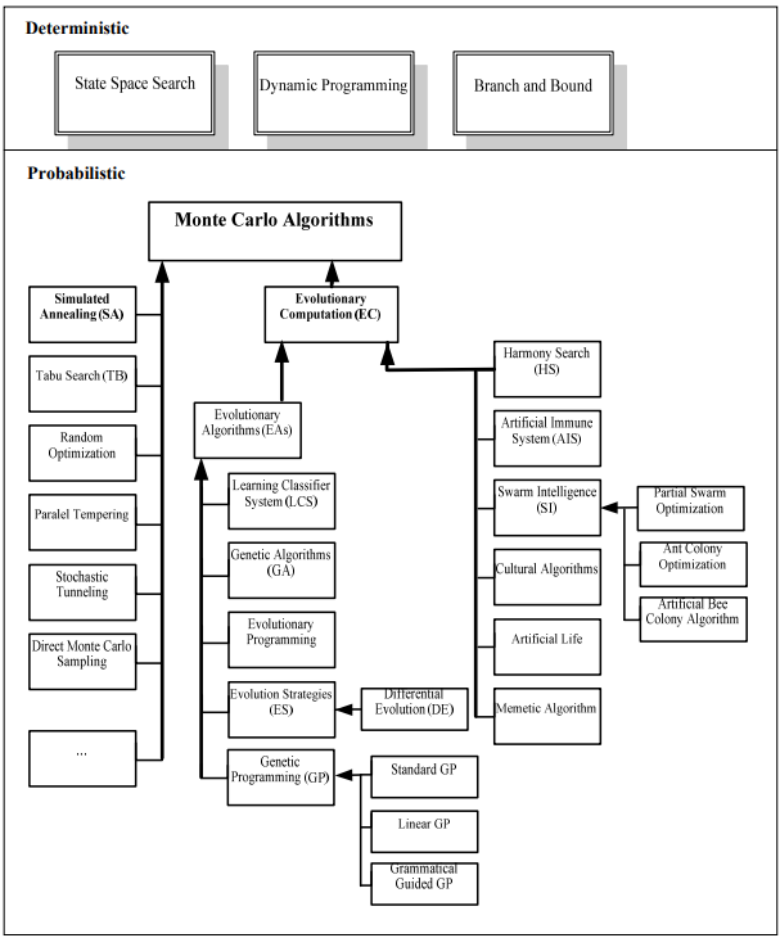
   Pada Algoritma optimasi deterministik hanya terdapat satu jalan untuk diproses, jika tidak ada jalan atau solusi lain yang dianggap terbaik maka algoritma dianggap selesai. Algoritma optimasi deterministik selalu menghasilkan solusi yang tetap untuk suatu input yang diberikan dan digunakan pada permasalahan dengan ruang solusi yang tidak luas.

   Algoritma deterministik adalah algoritma yang, dengan masukan tertentu, akan selalu menghasilkan keluaran yang sama, dengan mesin yang ditatanya selalu melewati urutan status yang sama (Ariansyah, et al., 2024). Pada penelitian yang dilakukan oleh (Nursyanti & Shalsabila, 2020), contoh algoritma pendekatan deterministik diantaranya adalah *Algoritma Wagner-Whitin, Silver Meal, Least Unit Cost, Economic Part Period, Lot for Lot, Least Total Cost, Period Order Quantity* dan *Part Period Balancing.*

2. Algoritma Probabilistik

   Algoritma Probabilistik digunakan untuk menyelesaikan permaalahan dengan ruang solusi yang sangat besar atau *high complexity*, bahkan tidak terbatas. Algoritma Probabilistik memiliki prinsip untuk menemukan solusi yang terbaik tanpa melebihi batasan waktu yang telah ditentukan. Namun, solusi terbaik dalam algoritma probabilistic bukan berarti solusi yang paling optimal namun mampu diterima oleh *user* dan di implementasikan kedalam kasus ril. Contoh dari beberapa metode dengan pendekatan algoritma probabilistik

diantaranya adalah *Stochastic Programming* dan *Change-Constrained Programming* (Marpaung, 2013).



Gambar 2. 13 Klasifikasi Algoritma Optimasi

2.7 *Stochastic Programming*

Pemograman stokastik merupakan suatu metode pemograman matematik dimana semua data yang tergabung ke dalam fungsi tujuan dan fungsi kendalanya berbentuk ketidakpastian. Masalah pemrograman stokastik berbeda dengan masalah optimasi deterministik yang dirumuskan dengan parameter-parameter yang diketahui, dalam masalah di dunia nyata selalu memasukkan beberapa parameter-parameter yang tidak diketahui. (Hasbiyati & Hasriati, 2017). Pemrograman stokastik berkaitan erat dengan optimasi *'real-time*, yang digunakan untuk mencapai hasil perencanaan dan pengambilan keputusan yang optimal dalam menghadapi data yang tidak pasti (Tanjung, et al., 2023). Tujuan dari pemrograman stokastik yaitu untuk mendapatkan solusi optimal yang bersifat acak/random (Marpaung, 2013).

Masalah pemrograman stokastik dapat dibedakan menjadi dua, yaitu masalah pemrograman stokastik linier dan masalah pemrograman stokastik nonlinier (Hasbiyati & Hasriati, 2017):

2.7.1 Pemrograman Stokastik linier

Masalah pemrograman stokastik linier adalah masalah pemrograman stokastik dengan fungsi tujuan dan fungsi kendalanya adalah fungsi-fungsi linier. Untuk masalah pemrograman linier digunakan metode simpleks dan metode dekomposisi, metode dekomposisi dual. Linier, jika: X adalah himpunan konvek dan fungsi gi: $\mathbb{R}n \rightarrow \mathbb{R}$, $i = 0, \cdots, m$ linier.

2.7.2 Pemrograman Stokastik non-linier

Masalah pemrograman stokastik nonlinier adalah masalah pemrograman stokastik dengan salah satu atau keduanya dari fungsi tujuan dan fungsi kendalanya berbentuk fungsi non-linier. Untuk masalah pemrograman non-linier digunakan metode *cutting-plane*, metode *descent*, metode *penalty* dan metode *lagrange*. Non-liner, jika:

paling sedikit satu dari fungsi gi: $\mathbb{R}n \rightarrow \mathbb{R}$, $i = 0, \cdots, m$ nonlinier atau X bukan himpunan konvek, dengan ketentuan sebagai berikut:

a. Konveks, jika $X \cap \{x|gi(x) \leq 0, i = 1, \cdots, m\}$ adalah himpunan konveks dan $g0$ adalah fungsi konveks,
b. Nonkonveks, jika tidak ada dari $X \cap \{x|gi(x) \leq 0, i = 1, \cdots, m\}$ yang bukan konveks atau fungsi objektif $g0$ bukan konveks.

2.8 *Genetic Algorithm* (Algoritma Genetika)

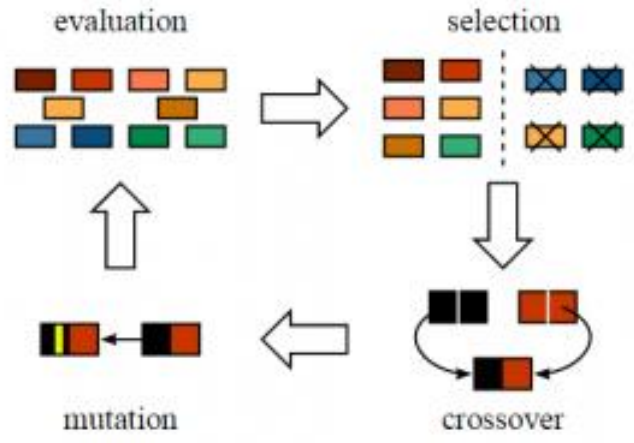
Genetic Algorithm (GA) adalah metode yang digunakan untuk menyelesaikan masalah optimasi, termasuk mencari rute terpendek. Proses komputasi yang terjadi dalam algoritma ini analog dengan proses seleksi makhluk hidup dalam sebuah populasi. Oleh karena itu, proses pencarian dalam algoritma genetika dilakukan sekaligus atas sejumlah penyelesaian masalah yang mungkin (Isai & Nugroho, 2024).

Genetic Algorithm (GA) meniru proses evolusi alami. Dalam evolusi, individu yang lebih kuat dan lebih mampu beradaptasi akan bertahan dan berkembang biak, sementara yang lebih lemah tersingkir. Prinsip ini diterapkan dalam Algoritma Genetika untuk menemukan solusi terbaik dalam suatu populasi. Awalnya, populasi dibentuk secara acak, dengan setiap individu memiliki karakteristik yang ditentukan oleh gen kromosomnya. Seiring berjalannya proses,

seleksi, mutasi, dan kombinasi gen menghasilkan individu yang lebih baik, hingga akhirnya diperoleh solusi yang optimal (Rahimi, et al., 2023).

Genetic Algorithm (GA) merupakan salah satu metode yang banyak digunakan dalam proses optimisasi. Tujuan utama dari optimisasi adalah memperoleh solusi yang benar-benar optimal. Akan tetapi, dalam penerapannya sering dijumpai kondisi *local optimum* atau optimum lokal. Optimum lokal merupakan keadaan ketika algoritma hanya menemukan nilai terbaik pada sebagian kandidat solusi, bukan pada keseluruhan ruang solusi. Kondisi ini berbeda dengan *global optimum* yang menunjukkan pencapaian nilai terbaik dari seluruh kandidat solusi yang ada. Terjadinya optimum lokal biasanya disebabkan oleh populasi yang mengalami konvergensi terlalu cepat (*premature convergence*) sehingga proses pencarian solusi berhenti sebelum mencapai hasil yang benar-benar optimal (Engelbrecht., 2006.).

Kriteria konvergensi dalam algoritma genetika umumnya ditandai ketika sekitar 80% hingga 85% kromosom memiliki nilai gen yang identik. Untuk menghindari terjadinya konvergensi prematur, diperlukan upaya menjaga keberagaman kromosom dalam populasi. Pada GA, keberagaman tersebut dapat dipertahankan melalui penerapan operator *crossover* dan *mutation* (mutasi) (Rajeev & Krishnamoorthy, 1992).



Gambar 2. 14 Visualisasi Algoritma Genetika

Sumber: *Papers with Code*

Menurut Haupt dan Haupt dalam (Isai & Nugroho 2024) struktur dasar algoritma genetika terdiri atas beberapa langkah (Isai & Nugroho, 2024):

1) Inisialisasi Populasi

   Populasi awal dengan ukuran N diinisialisasi secara acak. Populasi awal direpresentasikan sebagai kumpulan kromosom, [K1, K2, ..., KN].

2) Evaluasi Populasi

Setiap individu dalam populasi dievaluasi menggunakan fungsi *fitness*, f(Ki). Hasil evaluasi *fitness* digunakan untuk menentukan seberapa baik kromosom tersebut dalam mencapai solusi yang diinginkan. Persamaan yang digunakan untuk menghitung nilai fitness adalah sebagai berikut, (Sundarningsih, et al., 2017):

$$Fitness = \frac{1}{(fx)}$$

Dimana:

$$fx = \Sigma(\alpha D_{ij}) + \Sigma(\alpha P)$$

Dengan keterangan:

- D_{ij} adalah jarak tempuh dari titik i ke titik j.
- P merupakan Penalty apabila customer dilayani diluar jadwal.

3) Seleksi

Populasi Sebagian kromosom terbaik dipilih dari populasi saat ini untuk melahirkan generasi berikutnya. Seleksi dapat dilakukan berdasarkan nilai *fitness* individu atau probabilitas seleksi.

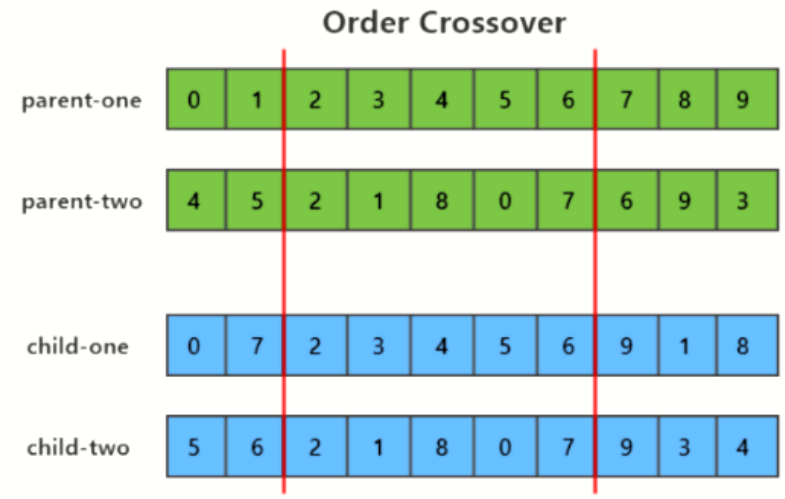
4) Proses Penyilangan

Crossover merupakan operator AG yang mengkombinasikan dua parent untuk menghasilkan chromosome keturunan (Sundarningsih, et al., 2017). Pada langkah ini, pasangan kromosom tertentu dipilih untuk melakukan proses penyilangan (*crossover*). Proses ini melibatkan pertukaran materi genetik antara pasangan kromosom untuk menghasilkan kromosom anak. Secara umum, mekanisme *crossover* adalah sebagai berikut:

a. Memilih dua buah *Chromosome* yang berperan sebagai induk
b. Memilih secara acak posisi dalam *chromosome*, biasa disebut dengan titik *crossover*, sehingga masing-masing chromosome induk terpecah menjadi dua segmen.
c. Melakukan pertukaran antar *segmen chromosome* induk untuk menghasilkan *chromosome* anak.

Crossover yang digunakan untuk penelitian ini yaitu jenis *Order Crossover*, yang mana dalam metode *Order Crossover* (OX), dua titik dipilih secara acak dari kedua induk. Kemudian, gen-gen yang berada di antara kedua titik tersebut dari induk pertama

langsung ditransfer ke anak pertama, dan dari induk kedua langsung ditransfer ke anak kedua. Setelah itu, untuk anak pertama, proses dilanjutkan dari titik kedua induk kedua, dengan menambahkan gen yang belum ada di anak pertama secara berurutan. Hal yang sama dilakukan untuk anak kedua, tetapi menggunakan induk pertama sebagai referensi (Karazmoodeh, 2023).

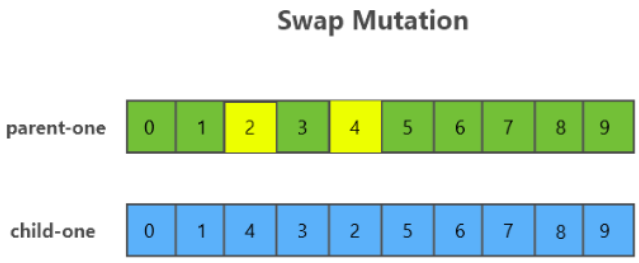


Gambar 2. 15 Proses *Order Crossover*
Sumber: (Karazmoodeh, 2023).

5) Proses Mutasi

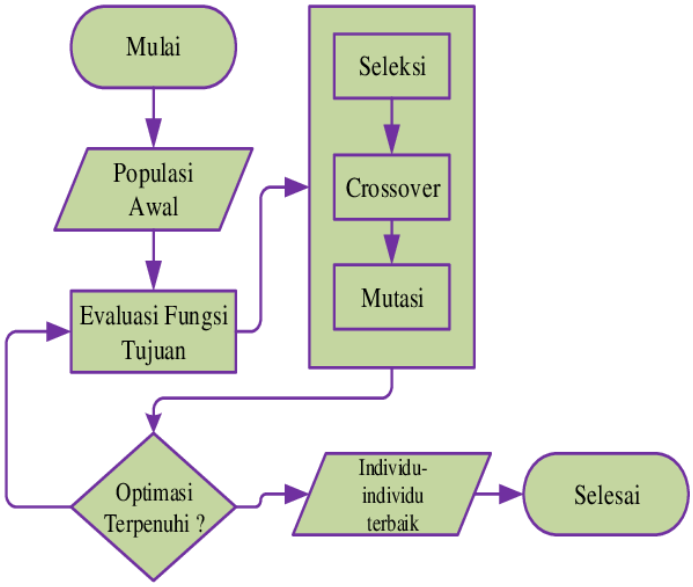
Setelah mengalami proses *crossover*, pada *offspring* dapat dilakukan mutasi. Mutasi dilakukan dengan cara melakukan perubahan pada satu gen atau lebih dari individu. Namun tidak semua keturunan dari crossover mengalami mutasi. Banyaknya parent yang mengalam mutasi akan bergantung pada probabiitas mutase yang ditentukan.

Mutasi berfungsi untuk menggantikan gen yang hilang dari populasi selama seleksi serta menyediakan gen yang tidak ada dalam populasi awal. Sehingga mutasi akan meningkatkan variasi populasi Pada langkah ini, kromosom tertentu dipilih untuk mengalami proses mutasi (Sundarningsih, et al., 2017). Proses mutasi menghasilkan perubahan acak pada gen dalam kromosom. Pada penelitian ini, jenis mutasi yang digunakan adalah *swap mutation*. Dalam *swap mutation*, dua gen dipilih secara acak, lalu nilai dari kedua gen tersebut ditukar.



Gambar 2. 16 Proses *Swap Mutation*
Sumber: (Karazmoodeh, 2023).

6) Evaluasi Populasi Baru. Setelah proses penyilangan dan mutasi selesai, populasi baru terbentuk. Langkah ini melibatkan evaluasi *fitness* terhadap populasi baru yang terbentuk.
7) Ulangi dari Langkah 3. Proses langkah 3 hingga langkah 6 diulang sampai syarat berhenti terpenuhi. Syarat berhenti dapat berupa mencapai solusi optimal, mencapai batas iterasi tertentu, atau kondisi berhenti lainnya.



Gambar 2. 17 *Flowchart* Umum Algoritma Genetika
Sumber: trivusi.web.id

2.9 *Nearest Insertion Algorithm*

Algoritma *Nearest Insertion* merupakan salah satu metode konstruksi dalam pendekatan *greedy* untuk membentuk rute pada permasalahan *Vehicle Routing Problem* (VRP). Berbeda dengan *Nearest Neighbor* yang sangat *greedy* karena selalu memilih titik terdekat berikutnya secara

langsung, algoritma *Nearest Insertion* memberikan fleksibilitas lebih dengan memperbolehkan penyisipan titik (node) ke dalam *sub-tour* yang sedang dibangun yang membuat solusi awal yang dihasilkan cenderung lebih optimal dibandingkan pendekatan *greedy* murni (Eche, et al., 2025).

Algoritma ini bekerja dengan memulai dari satu pelanggan sebagai rute awal, kemudian secara bertahap menyisipkan pelanggan terdekat ke dalam rute tersebut, dengan mempertimbangkan posisi penyisipan yang memberikan peningkatan jarak minimum. Cara kerja metode *Nearest Insertion* adalah pemilihan lokasi pelanggan berdasarkan jarak terdekat dari lokasi terakhir yang kemudian akan dikomputasi menjadi rute dengan jarak yang optimal (Kurniawan, et al., 2014).

Langkah-langkah yang dilakukan pada *Nearest Insertion* menurut Pop dalam (Kurniawan, et al., 2014) adalah sebagai berikut:

1) Proses diawali dari depot, kemudian dilanjutkan dengan mencari lokasi pelanggan yang belum dikunjungi dengan jarak terpendek dari depot.
2) Proses dilanjutkan ke lokasi lain dengan jarak terdekat dari lokasi yang terpilih sebelumnya dan jumlah pengiriman tidak melebihi kapasitas kendaraan.
3) Apabila ada lokasi yang terpilih sebagai lokasi berikutnya dan terdapat sisa kapasitas pada kendaraan pengangkut, maka proses kembali ke langkah (2).
4) Jika kendaraan tidak memiliki sisa kapasitas, maka proses kembali ke langkah (1).
5) Jika tidak ada lokasi yang terpilih karena jumlah pengiriman melebihi kapasitas kendaraan, maka Kembali ke langkah (1).
6) Proses dimulai kembali dari gudang dan mengunjungi pelanggan yang belum dikunjungi dan memiliki jarak terdekat.

   Jika semua pelanggan telah dikunjungi sekali, maka algoritmanya telah berakhir.

BAB III
METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian adalah sesuatu yang akan menjadi fokus penelitian oleh peneliti terhadap proses optimasi rute kendaraan atau *Vehicle Routing Problem* untuk kebutuhan PT Pos Logistik. Fokus penelitian ini yaitu mengoptiptimalkan rute untuk meminimalkan jarak, waktu tempuh, dan keterlambatan pada kasus CVRP-TW. Kemudian hasil yang diharapkan yaitu mampu menghasilkan rute terbaik dengan meminimasi jarak dan keterlambatan. Subjek penelitian ini meliputi operasional PT Pos Logistik Indonesia dan *stakeholder* yang terkait dengan proses bisnis perusahaan yaitu distrubusi barang. *Stakeholder* pada penelitian ini diantaranya pelanggan, pengemudi atau operator kendaraan, dan manajer operasional logistik. Subjek-subjek ini berperan sebagai sumber data dan informasi yang diperlukan untuk menyelesaikan masalah *Capacitated VRP with Time Window.* Berikut merupakan informasi tambahin terkait biodata perusahaan:

Nama Perusahaan	:	PT Pos Logistik Indonesia
Alamat Perusaahaan	:	Gedung Pos Ibukota, Jl. Lap. Banteng Utara No.1, Pasar Baru, Sawah Besar, Central Jakarta City, Jakarta 10710
Nomor Telepon	:	+6281297274831
Bidang Usaha	:	Industri distribusi, pergudangan dan logistik.
Tahun Berdiri	:	2021
Jenis Perusahaan	:	Perseroan terbatas



Gambar 3. 1 PT Pos Logistik Indonesia

3.2 Instrumen Penelitian

Dalam studi ini, terdapat beberapa alat yang diperlukan untuk mendukung pengumpulan dan pengolahan data. Berikut adalah instrumen penelitian yang digunakan:

1. Survei dan data historis perusahaan, dibutuhkan dalam pengumpulan data untuk proses pengolahan kode. Data tersebut dapat berupa data *demand delivery*, penalty, kapasitas kendaraan yang digunakan, Lokasi pengantaran dan penjemputan, dan lain sebagainya.
2. *Google Maps,* diperlukan untuk kebutuhan data titik *longitude* dan *latitude* yang akan di konversikan dari titik pada *Google Maps*.
3. Perangkat lunak *Python* jenis 3.13.1, digunakan untuk menjalankan Algoritma Genetika, melakukan simulasi, menghitung jarak, dan menganalisis hasil optimasi dengan *library* seperti *geopy, numpy,* dan *pandas*.
4. *Software Visual Studio Code*, digunakan dalam memudahkan proses komputasi kode yang akan dijalankan dengan *Python*.

3.3 Sumber Data Primer dan Sekunder

Sumber data berasal dari data primer dan data sekunder, kedua sumber data ini akan saling melengkapi dan menguatkan keilmuan penelitian.

1) Data Primer, merupakan data yang dikumpulkan melalui serangkaian kegiatan langung atau wawancara yang dilakukan di lapangan tanpa melalui perantara, dimana sumber data memberikan langung data kepada pengumpul data. Data ini meliputi:
   a. Data *demand*
   b. Data *Time windows*
   c. Data Kapasitas kendaraan
   d. Data Kecepatan kendaraan
   e. Data lokasi
   f. Data Kebijakan atau Kondisi eksisting perusahaan.
2) Data Sekunder, merupakan data yang diperoleh melalui perantara, maupun melalui kajian yang dilakukan terhadap dokumen organisasi, buku, jurnal, artikel dan sebagainya. Data sekunder juga dapat diperoleh menggunakan alat atau perangkat lunak. Data sekunder digunakan sebagai penunjang penelitian dan penguat deskripsi-deskripsi kualitatif dalam penelitian. Data ini meliputi:
   a. Data *longitude* dan *latitude (Google maps)*

b. Jarak antar Lokasi
c. Data pendukung penelitian (jurnal artikel).

3.4 Metode Pengumpulan Data

Data dikumpulkan untuk mendapatkan informasi yang diperlukan dalam proses pengolahan data demi mencapai tujuan penelitian ini. Penelitian menggunakan metode studi literatur dan pengumpulan data melalui wawancara kepada narasumber terkait secara langsung. Berikut merupakan penjelasan tiap metode penelitian yang digunakan pada penelitian ini:

1. Studi Literatur

   Studi literatur yang digunakan bersumber dari jurnal dan buku sebagai landasan teori dan referensi penelitian digunakan untuk mengumpulkan data/referensi yang berkaitan dengan topik penelitian yang terdapat dalam literatur, buku, jurnal, dan penelitian terkait lainnya. Studi literatur dapat bertujuan untuk memperkuat topik atau teori yang diterapkan oleh peneliti.

2. Wawancara

   Wawancara merupakan suatu metode pengumpulan data yang dilakukan dengan cara melalui sistem tanya jawab yang mana bersifat dua arah. Wawancara dilakukan bersama orang yang memiliki peran penting dalam permasalahan yang akan diatasi yang mana dianggap *expert* di perusahaan tersebut. Peneliti melakukan wawancara untuk mengidentifikasi permasalahan yang dihadapi oleh perusahaan serta untuk mendapatkan data lainnya.

3.5 Populasi dan Sampel

3.5.1 Populasi

Populasi merupakan sekumpulan objek atau subjek yang memiliki kualitas dan karakteristik tertentu yang dijadikan fokus oleh peneliti untuk diteliti dan kemudian diambil kesimpulannya (Sugiyono, 2011). Dalam penelitian ini, populasi merupakan seluruh titik pengantaran dan penjemputan dalam keseluruhan rute distribusi pada PT Pos Logistik Indonesia.

3.5.2 Sampel

sampel adalah bagian dari populasi yang dipilih untuk mewakili karakteristik populasi secara keseluruhan (Sugiono, 2013). Data yang diperoleh dari sampel harus dapat mewakili populasi. Agar data tersebut valid maka sampel harus representative (Suriani, et al., 2023). Penelitian ini

menggunakan metode random sampling, ini melibatkan pembagian populasi menjadi kelompokkelompok yang homogen, masing-masing kelompok berisi subjek dengan karakteristik yang sama, dan kemudian pengambilan sampel secara acak dalam kelompok-kelompok tersebut (Suriani, et al., 2023). Ukuran sampel dalam penelitian ini mencakup titik distribusi produk di area Jakarta Raya, sebanyak 100 titik.

3.6 Metode Pengolahan Data

Penelitian ini menggunakan *Genetic Algorithm* (GA) sebagai metode optimasi. GA termasuk ke dalam algoritma stokastik karena dalam proses pencarian solusi terdapat komponen probabilistik, antara lain pada tahap inisialisasi populasi, pemilihan kromosom, serta penerapan operator crossover dan mutation. Hal tersebut menyebabkan hasil solusi yang diperoleh dapat bervariasi pada setiap kali eksekusi. GA tidak mengikuti aturan deterministik yang selalu memberikan solusi sama, melainkan mengandalkan prinsip probabilitas untuk menemukan solusi yang lebih baik. Sementara itu, data yang digunakan bersifat deterministik dinamis, yaitu demand, jarak antar titik, kapasitas kendaraan, dan *time window* telah diketahui sejak awal dan berbeda untuk setiap titik. Dengan demikian, karakteristik penelitian ini menggabungkan data deterministik dengan metode optimasi stokastik.

Metode pengolahan data yang dilakukan dalam penelitian ini mengikuti kaidah atau *framework* metode Algoritma Konstruktif Heuristik *Nearest Insertion* dan Metaheuristik *Genetic Algorithm* (GA) berupa operator secara umum seperti yang terdapat pada Buku *Genetic Algorithm and Genetic Programming* (Affenzeller, et al., 2009). Pada bagian ini juga akan menjelaskan formuasi matematis yang berkaitan dengan permasalahan yang diangkat yaitu *Capacitated Vehicle Routing Problem with Time Window* (CVRPTW).

3.6.1 Model Matematis CVRPTW

A. Notasi

Berikut ini merupakan set indeks dan juga parameter yang digunakan dalam penelitian VRP ini, seperti notasi *node*, parameter konstanta yang menjadi *constraint* dan lain sebagainya.

Set dan Indeks:

- $N = \{0,1,2, \dots, n\}$: Merupakan himpunan semua lokasi (0 adalah depot, lainnya adalah pelanggan).
- $K = \{1,2, \dots, k\}$: Merupakan himpunan kendaraan.

Parameter:

- Q: Merupakan kapasitas maksimum kendaraan (1000 kg).
- d_i: Merupakan permintaan (*delivery demand*) pelanggan titik - *i*.
- e_i, l_i: Merupakan *time window* pelanggan iii, yaitu waktu layanan harus dimulai di antara e_i dan l_i (dalam menit).
- S: Merupakan waktu layanan (*service time*) pada setiap titik pelanggan (5 menit).
- v : Merupakan kecepatan kendaraan (22 km/jam).
- c_s: Merupakan biaya pengiriman per pelanggan (Rp. 10.000).
- c_p: Merupakan denda keterlambatan (Penalti) per kedatangan (Rp).
- $dist(i,j)$: Merupakan jarak *Euclidean* (*Haversine*) antara titik iii dan jjj (km).

B. Fungsi Perhitungan Jarak dan Waktu tempuh

Pada penelitian VRP ini perhitungan jarak akan menggunakan fungsi *haversine* untuk memperoleh matriks perhitungan jarak.

1) Program Fungsi *Haversine*

Metode *Haversine* merupakan sebuah metode perhitungan yang digunakan dalam sistem navigasi dimana metode ini akan menghasilkan sebuah perhitungan jarak antara dua titik dari garis bujur (*longitude*) dan garis lintang (*latitude*) (Amin & , 2021), dalam hal ini merupakan titik pelanggan atau *customer* yang dilayani. Berikut merupakan rumus fungsi *heversine*:

$$dist(i,j) = 2R \cdot \arcsin\left(\sqrt{sin^2\left(\frac{\Delta\emptyset}{2}\right) + cos(\emptyset_i)cos(\emptyset_j)sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \quad (3.1)$$

Di mana:

- $R = 6371Km$: Merupakan jari-jari bumi.
- ϕ dan λ : Merupakan lintang dan bujur dalam radian.
- $\Delta\phi = \phi_j - \phi_i$
- $\Delta\lambda = \lambda_j - \lambda_i$

2) Program Fungsi Faktor Koreksi *ETA*

Fungsi *Haversine* merupakan fungsi perhitungan yang mempertimbangkan bentuk lengkung bumi dengan cara menarik garis lurus antar titik. Untuk menghitung waktu tempuh (ETA) yang realistis dibutuhkan faktor koreksi. Faktor koreksi adalah angka

pengali untuk mengonversi jarak garis lurus (*haversine*) menjadi estimasi jarak realistik berdasarkan kondisi jalan sesungguhnya.

Penggunaan *Google Maps* API yang lebih akurat tidak dipilih karena bersifat berbayar, memiliki batasan kuota harian, dan kurang efisien untuk eksperimen dalam skala besar. Oleh karena itu, diterapkan faktor koreksi (misalnya 1.6) untuk memperkirakan jarak riil dengan asumsi bahwa rute jalan umumnya lebih panjang 20–40% dibandingkan garis lurus, terutama pada wilayah urban dengan jaringan jalan yang kompleks. Penentuan faktor koreksi dilakukan dengan kalibrasi manual dengan rumus berikut:

$$Fk = \frac{Da_{i,j}}{Db_{i,j}} \tag{3.2}$$

Di mana:

- Fk: Faktor Koreksi,
- $Da_{i,j}$: Jarak antar titik nyata,
- $Db_{i,j}$: Jarak antar titik *Haversine.*

3) Program Fungsi Waktu Tempuh (ETA)

Untuk memperoleh waktu tempuh yang dibutuhkan dari perjalanan dari titik *I* ke *j,* akan digunakan fungsi perhitungan jarak sebagai berikut:

$$\text{Waktu Tempuh } (ETA) = \frac{F\ x\ D}{V}\ x\ 60 + S \tag{3.3}$$

Di mana:

- Waktu Tempuh (ETA): Waktu estimasi perjalanan dalam menit.
- *F* (Faktor Koreksi): Angka pengali untuk mengonversi jarak garis lurus (*haversine*) menjadi estimasi jarak realistik berdasarkan jalan sungguhan,
- *D* (*Distance*)*:* Hasil jarak dari perhitungan fungsi *Haversine* (Km)*,*
- *V (Vehicle Speed)*: Kecepatan rata-rata kendaraan (22 Km/Jam),
- *S (Service Time)*: Waktu layanan/*loading*

C. Program *Multi-Objective Function*

Penelitian ini memiliki fungsi objektif dengan beberapa tujuan minimasi dan optimasi, yaitu minimasi jarak tempuh dan minimasi biaya *penalty* akibat keterlambatan, yang tergabung menjadi satu fungsi matematis sebagai berikut:

$$Min\ Z = \left(\sum_{k=1}^{K} \left(C_S \cdot\ n_k + \sum_{i \in R_k} P_i \right) \right) \tag{3.4}$$

Dengan variabel sebagai berikut:

- C_s = Rp. 10.000: Merupakan biaya ongkos kirim per pelanggan,
- n_k: Merupakan jumlah *customer* yang dilayani oleh kendaraan ke- *k,*
- P_i: Merupakan penalti keterlambatan pada titik *I* dengan,

$$pi = \begin{cases} P\ \cdot\ \left(a_i - tw_i^{end}\right), & jika\ a_i\ lainnya \\ 0, & \\ & >\ tw_i^{end} \end{cases} \tag{3.5}$$

- P: 20% (Rp. 2000): Merupakan Penalti keterlambatan,
- a_i: Merupakan waktu kedatangan kendaraan di titik *i,*
- tw_i^{end}: Merupakan batas akhir *time windows* titik *i,*
- *K:* Jumlah kendaraan
- R_k: Merupakan urutan titik yang dikunjungi oleh kendaraan ke-*k*

D. Program *Constraint*

Penelitian ini memiliki batasan berupa hambatan atau kendala yang menjadi kriteria tambahan dalam proses penelitian. Kondisi ini merupakan kondisi ril yang terjadi di perusahaan yang harus dipertimbangkan dalam proses penelitian, karena termasuk kedalam model konseptual yang telah dibangun. Berikut merupakan kondisi hambatan dalam penelitian ini:

- Setiap *customer* hanya dikunjungi sekali oleh satu kendaraan,

$$\sum_{k=1}^{K} \sum_{i=0}^{n} x_{ij}^{k} = 1, \forall i \in C \tag{3.6}$$

- Kapasitas kendaraan tidak boleh dilampaui:

$$\sum_{i=\in R_k} D_i \leq Q,\ \forall k\ \in K \tag{3.7}$$

- *Time windows* harus dihormati (penalti jika lewat batas akhir):

$$a_i \leq\ tw_i^{end} + penalty \tag{3.8}$$

- Kendaraan harus berangkat dari depot dan kembali lagi ke depot ($0 \rightarrow 0$):

$$R_k = [0, \dots, 0] \; \forall k \tag{3.9}$$

- Konservasi aliran (*Flow Conservation*), memastikan rute tidak putus:

$$\sum_{i \in N} x_{ij}^k = \sum_{i \in N} x_{jl}^k, \forall j \in C, \forall k \in K \tag{3.10}$$

E. Program *Decission Variable*

Dalam model yang digunakan pada penelitian ini, terdapat beberapa variabel keputusan utama yang mendeskripsikan keputusan dalam proses perutean kendaraan. Variabel-variabel tersebut dijelaskan sebagai berikut:

a) Variabel Biner

$$x_{ij}^k \in \{0,1\} \tag{3.11}$$

- $x_{ij}^k = 1$ jika kendaraan *k* melewati rute dari titik *i* ke titik *j,*
- $x_{ij}^k = 0$ jika tidak.

variabel ini menentukan apakah kendaraan ke-k melakukan perjalanan langsung dari titik *i* ke titik *j*. Titik i dan *j* dapat berupa depot atau pelanggan.

b) Variabel Bantu

- $t_i^k \geq 0$ Waktu kendaraan *k* di titik *i,*

  Variabel ini menyatakan waktu kedatangan kendaraan ke-k di titik i. Waktu kedatangan ini digunakan untuk mengevaluasi apakah pengiriman berada dalam rentang *time window* yang ditetapkan. Jika tidak, maka diberlakukan penalti atas keterlambatan pengiriman.

- $t_i^k \geq 0$ Kapasitas kumulatif kendaraam *k* setelah mengunjungi titik *i.*

  Variabel ini menyatakan jumlah total permintaan (*demand*) yang telah dibawa kendaraan ke-k setelah mengunjungi titik i. Variabel ini digunakan untuk memastikan bahwa kapasitas kendaraan tidak terlampaui selama rute berlangsung.

3.6.2 Metode *Nearest Insertion* (NI)

Berbeda dengan *Nearest Neighbor* yang sangat *greedy* karena selalu memilih titik terdekat berikutnya secara langsung, algoritma *Nearest Insertion* memberikan fleksibilitas lebih dengan

memperbolehkan penyisipan titik (*node*) ke dalam *sub-tour* yang sedang dibangun yang membuat solusi awal yang dihasilkan cenderung lebih optimal dibandingkan pendekatan *greedy* murni. Berikut merupakan langkah yang dilakukan dalam metode *Nearest Insertion* dalam penelitian ini menurut (Eche, et al., 2025):

1. Inisialisasi, yaitu memilih satu kota sebagai titik awal (depot atau *node* dengan nilai tertentu).
2. Pemilihan Titik Terdekat, yaitu memukan pelanggan (titik) yang belum dikunjungi yang memiliki jarak terdekat terhadap salah satu titik dalam rute yang sudah terbentuk.
3. Penyisipan ke Rute, yaitu menyelipkan titik tersebut ke dalam rute eksisting di antara dua kota (i dan j) sehingga penambahan jarak akibat penyisipan kota baru (k) minimal.
4. Iterasi yaitu mengulangi proses pemilihan dan penyisipan hingga semua kota telah dimasukkan ke dalam rute.

3.6.3 Metode *Genetic Algorithm* (GA)

Berikut merupakan langkah-langkah penyelesaian algoritma genetika berupa operator secara umum seperti yang terdapat pada Buku *Genetic Algorithm and Genetic Programming* (Affenzeller, et al., 2009).

1. *Parent Selection*

   Dalam algoritma genetika, fungsi *fitness* memberikan skor pada setiap individu dalam populasi; nilai *fitness* ini menunjukkan kualitas solusi yang diwakili oleh individu tersebut. Setelah populasi dihasilkan dan *fitness*-nya diukur, kumpulan solusi yang dipilih untuk "dikawinkan" dalam suatu generasi ditentukan. Dalam algoritma genetika standar (SGA), probabilitas bahwa suatu kromosom dari populasi saat ini akan dipilih untuk reproduksi sebanding dengan nilai *fitness*-nya (Affenzeller, et al., 2009). Ada beberapa metode yang dapat dilakukan pada proses seleksi atau pemilihan ini, namun yang digunakan pada penelitian ini yaitu metode *Proportional selection (roulette wheel selection).*

   Proses seleksi *Roulette Wheel* dilakukan dengan cara memutar roda secara berulang kali, individu dipilih menggunakan metode *random sampling with replacement* (pengambilan sampel acak dengan pengembalian). Untuk membuat seleksi proporsional menjadi independen dari skala nilai *fitness*, teknik yang digunakan biasanya adalah Teknik *windowing*.

2. *Recombination (Cross Over)*

Dalam formulasi paling sederhana, seperti yang disarankan dalam algoritma genetika (GA) kanonik untuk pengkodean biner, *crossover* melibatkan dua individu dan memotong *string* kromosom mereka di posisi yang dipilih secara acak. *Substring* yang dihasilkan kemudian ditukar untuk menghasilkan dua kromosom baru yang lebih Panjang (Affenzeller, et al., 2009). Pada penelitian ini, teknik *crossover* yang digunakan yaitu kombinasi dari penggabungan rute-rute (i) dengan fungsi tambahin untuk menghilangkan duplikasi atau pengulangan pada tiap hasil perkawinan.

3. *Mutation*

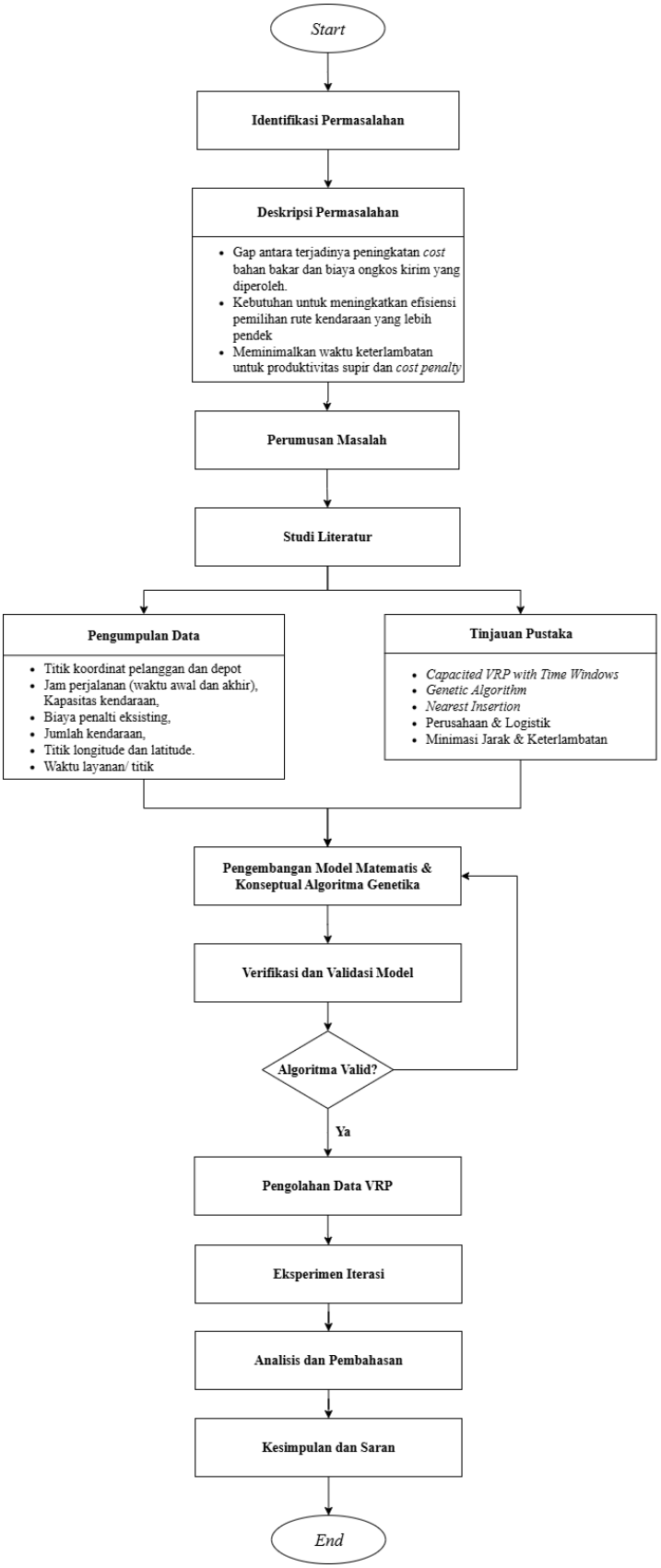
   Pada sistem yang menggunakan angka atau bilangan bulat, mutasi dilakukan dengan mengganti nilai dengan angka acak dalam rentang yang sesuai. Namun, dalam permasalahan yang lebih rumit seperti optimasi kombinatorial (misalnya pada masalah rute), cara ini justru dapat menimbulkan masalah, misalnya terdapat angka yang muncul dua kali padahal seharusnya unik. Untuk mengatasi itu, beberapa metode alternatif disarankan, seperti menggunakan (*pairwise swap*) atau menggeser posisi elemen (*shift operation*). Pada penelitian ini, teknik mutasi yang digunakan adalah *Pairwise Swap Mutation* yaitu Teknik mutasi dengan cara menukar posisi dua elemen.

4. *Replacement Scheme*

   Operator ini biasa dikenal juga dengan evaluasi populasi atau proses *offspring*. Setelah menghasilkan generasi baru keturunan melalui *crossover* dan mutasi, muncul pertanyaan mengenai individu mana yang harus menjadi bagian dari generasi berikutnya. Dalam strategi evolusi, hal ini menentukan umur setiap individu dan sangat mempengaruhi bagaimana algoritma mencapai solusi optimal (*convergence behavior*). Selain itu, ada strategi lain yang sangat mempengaruhi proses pemilihan individu untuk generasi selanjutnya, yaitu *offspring selection* (seleksi keturunan) (Affenzeller, et al., 2009).

3.7 Alur Penelitian

Alur penelitian bertujuan untuk menjabarkan langkah-langkah peneliti dalam mengembangkan dan merancangan penelitian ini dari awal hingga akhir. Berikut adalah *flowchart* dari alur penelitian ini:



Gambar 3. 2 Alur Penelitian

Penjelasan dari alur penelitian di atas adalah sebagai berikut:

1. Mulai
2. Identifikasi Permasalahan

   Topik penelitian datang dari proses awal pengidentifikasian permasalahan. Peneliti melakukan observasi terhadap objek penelitian dan mengamati permasalahan yang dapat diselesaikan. Proses identifikasi melibatkan narasumber dan bagian operasional sehingga memperkuat urgensi dilakukannya penelitian ini.
3. Deskripsi Identifikasi masalah

   Penelitian ini dilakukan untuk menyelesaikan permasalahan-permasalahan yang didapati pada objek penelitian. Sebelum menentukan alat atau metode khusus dalam penelitian, penting untuk mengetahui terlebih dahulu permasalahan yang menjadi dasar dilakukannya suatu penelitian. Urgensi dilakukannya penelitian VRP ini adalah karena terdapat gap antara kondisi eksisting perusahaan yaitu terjadinya peningkatan *cost* operasional, dengan harapan perusahaan dalam menentukan target *cost* yang akan dikeluarkan perusahaan. Perusahaan juga merasa membutuhkan adanya suatu penelitian khusus untuk meningkatkan efisiensi distribusi pada perusahaan.
4. Perumusan Masalah

   Tujuan rumusan masalah adalah untuk memperjelas fokus penelitian dan mengarahkan proses penelitian. Rumusan masalah membantu peneliti untuk menentukan masalah spesifik yang ingin dipecahkan dan membatasi ruang lingkup penelitian.
5. Studi Literatur

   Pada tahapan ini, studi literatur dilakukan dengan tujuan untuk mempelajari dan mencari informasi lebih lanjut terkait permasalahan yang dihadapi perusahaan. Permasalahan yang dihadapi yaitu terkait penentuan rute distribusi barang yang terdapat beberapa Batasan yang dipertimbangkan seperti kapasitas kendaraan yang digunakan dan batasan waktu di setiap pelanggan. Studi literatur ini digunakan untuk mengetahui metode yang dapat digunakan untuk memecahkan permasalahan yang ada. Studi literatur dapat berupa penelitian terdahulu, jurnal, atau buku terkait.
6. Tinjauan Pustaka

   Tinjauan Pustaka dilakukan dengan tujuan untuk mengumpulkan referensi yang relevan dengan penelitian yang dilaksanakan. Pustaka yang dikumpulkan antara lain terkait *capacited VRP with Time Window, Nearest insertion, Genetic Algorithm* yang menjadi

metode komputasi pada penelitian ini. Dengan melakukan tahapan tinjauan Pustaka ini, peneliti mampu memahami dasar teori yang relevan dan mampu mengembangkan model konseptual, matematis, dan algoritma dari masalah yang diangkat.

7. Pengumpulan Data

   Setelah mamahami secara konsep dan juga teori yang relevan dengan penelitian, langkah selanjutnya yaitu melakukan pengumpulan data yang dibutuhkan untuk pengolahan data pada penelitian. Data yang digunakan pada penelitian ini merupakan jenis deterministik dinamis yang mana seluruh parameter telah diketahui sejak awal dan berbeda pada setiap titik. Data-data tersebut diantaranya data terkait titik distribusi, data permintaan, kebijakan dan regulasi seperti waktu layanan dan kapasitas, serta data pendukung latar belakang seperti biaya eksisting dan jumlah keterlambatan yang terjadi pada perusahaan.

8. Pengembangan Model dan Algoritma

   Setelah memperoleh data-data yang dibutuhkan dalam penelitian, langkah selanjutnya adalah melakukan pengembangan *Genetic Algorithm* (GA) untuk menyelesaikan kasus CVRPTW pada penelitian ini. Pengembangan ini terdiri atas proses pembangunan model matematis berisi fungsi-fungsi yang akan menyelesaikan kasus permasalahan. Dengan mempertimbangkan beberapa *constraint* dan *considerity* pada penelitian, model dikembangakan melewati *trial-and-error* secara terus menerus hingga model dapat dikatakan sesuai dan siap digunakan untuk pengolahan data.

9. Validasi dan Verifikasi Algoritma

   Model matematis dan Algoritma yang sudah dikembangkan sebelumnya akan digunakan dalam pengolahan data CVRPTW. Sebelum diterapkan, algoritma tersebut perlu melalui proses verifikasi dan validasi menggunakan data *dummy*. Verifikasi dilakukan untuk memastikan bahwa program berjalan sesuai dengan model yang telah dirancang. Jika program mampu beroperasi sebagaimana mestinya, maka algoritma dinyatakan telah terverifikasi. Sementara itu, validasi bertujuan untuk memastikan bahwa model yang dibuat mencerminkan kondisi nyata dari sistem. Validasi dilakukan dengan cara membandingkan rute yang dihasilkan program dengan perhitungan manual. Melalui proses verifikasi dan validasi ini, hasil dari program dapat dianggap layak untuk digunakan dalam tahap analisis berikutnya.

10. Pengolahan Data VRP

Apabila pengembangan model selesai, data yang sudah diperoleh pada tahap sebelumnya diolah menggunakan *Genetic Algorithm* dengan *tools* Python sebagai alat komputasinya. *Genetic Algorithm* bekerja dengan membentuk populasi inisial, membentuk generasi-generasi yang nantinya generasi tersebut di mutasi dan akan dipilih "*best child"* atau individu terbaik, yang mana pada penelitian ini individu tersebut adalah titik yang membentuk rute terbaik. Penelitian ini juga menggunakan sedikit implementasi algoritma *Nearest Neighbor* untuk membantu algoritma mencari individu dengan jarak terdekat. Proses pengolahan data dilakukan pada *device* HP *Pavilion* 14 *Intel Core* i7 *Generation* 13^{th}.

11. Eksperimen Iterasi

    Eksperimen iterasi dalam penelitian ini bertujuan untuk mengidentifikasi kombinasi parameter terbaik seperti jumlah populasi, generasi, dan tingkat mutase yang menghasilkan solusi paling optimal dan konsisten. Karena sifat GA yang acak (*stochastic*), eksperimen iteratif penting untuk mengukur stabilitas solusi dengan prinsip probabilitas. Selain itu, eksperimen ini membantu membandingkan performa solusi terhadap solusi awal dan mengetahui titik di mana peningkatan parameter tidak lagi memberikan perbaikan signifikan terhadap solusi. Hasil dari iterasi ini dapat dianalisis melalui metrik seperti total jarak, biaya, keterlambatan, dan total penalty untuk memberikan gambaran menyeluruh terhadap efektivitas model yang dikembangkan.

12. Analisis dan Pembahasan

    Pada tahap ini peneliti akan memaparkan hasil yang diperoleh dari pengolahan data. Tujuan dari tahapan ini adalah untuk untuk menginterpretasikan hasil yang diperoleh dan membandingkan hasil yang diperoleh dengan tujuan atau hipotesis penelitian yang telah ditentukan. Pada tahap ini, juga memungkinkan bagi peneliti untuk menjelaskan kelebihan dan perbedaan hasil penelitian ini dari penelitian terhadulu.

13. Kesimpulan dan Saran

    Tujuan dari kesimpulan dalam sebuah penelitian adalah untuk merangkum hasil utama secara singkat dan padat berdasarkan analisis yang telah dilakukan, serta menjawab rumusan masalah atau tujuan penelitian. Kesimpulan memberikan gambaran akhir mengenai temuan utama dan relevansi hasil penelitian, sehingga pembaca dapat memahami inti dari penelitian tanpa harus membaca seluruh isi laporan. Sementara itu, saran bertujuan untuk memberikan rekomendasi praktis atau teoritis berdasarkan hasil

penelitian, baik untuk implementasi di lapangan, perbaikan di masa depan, maupun sebagai masukan untuk penelitian lanjutan. Saran juga menunjukkan bahwa peneliti menyadari batasan studi dan membuka ruang bagi pengembangan lebih lanjut di bidang yang sama.

14. Selesai

BAB IV
PENGUMPULAN DAN PENGOLAHAN DATA

Pada bab ini dipaparkan proses pengolahan data dan data-data yang digunakan pada penelitian CVRP-TWPD menggunakan *Genetic Algorithm* dan *Nearest Insertionr* ini. Pada bab ini juga dijelaskan bagaimana proses Pembangunan model terjadi, proses validasi dan verifikasi model, hingga eksperimen iterasi pada proses pencarian rute terbaik yang akan diterapkan perusahaan.

4.1 Profil Perusahaan

Pada bagian profil perusahaan, penulis tidak hanya menjelaskan Sejarah dan informasi terkait profil PT. Pos Logistik Indonesia, melainkan kebijakan-kebijakan, keberlakuan dan regulasi yang berlaku di perusahaan yang berhubungan secara langsung dengan topik peneltian ini.

4.1.1 Deskripsi Perusahaan

PT Pos Logistik Indonesia (POSLOG) merupakan anak perusahaan dari PT Pos Indonesia (Persero) yang beroperasi secara independen dalam menyediakan layanan logistik. Sebagai pelopor inovasi dalam industri logistik, POSLOG memanfaatkan jaringan luas yang dimiliki oleh Pos Indonesia, dengan lebih dari 4.367 titik jaringan dan 33.000 titik penjualan di seluruh Indonesia. Ditenagai oleh tim profesional yang ahli dalam merumuskan strategi bisnis, PT Pos Logistik Indonesia fokus untuk memaksimalkan peluang dalam industri logistik di Indonesia, dengan tetap menjaga kualitas layanan yang efisien dan inovatif (PT Pos Logistik Indonesia, 2025).

4.1.2 Sejarah Perusahaan

PT Pos Logistik Indonesia berawal dari *Parcel Division/Business Unit Logistics* pada tahun 1996–2004, kemudian berkembang menjadi Proyek Bisnis Total Logistik pada 2004–2007. Pada periode 2007–2011, unit ini bertransformasi menjadi Unit Bisnis Strategis. Akhirnya, pada 17 Februari 2012, PT Pos Logistik Indonesia resmi berdiri sebagai anak perusahaan PT Pos Indonesia (Persero). Dengan dukungan jaringan Pos Indonesia yang luas, perusahaan ini beroperasi secara independen untuk mengoptimalkan peluang bisnis logistik di Indonesia, didukung oleh 4.367 kantor cabang dan 33.000 titik penjualan (PT Pos Logistik Indonesia, 2025).

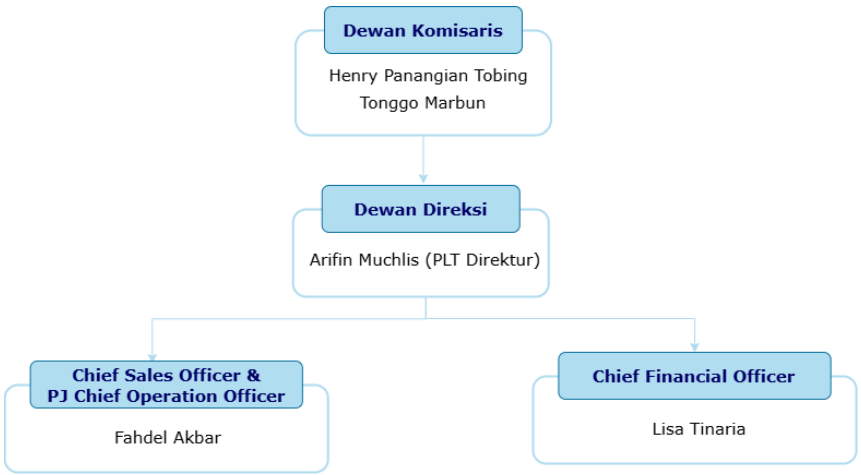


Gambar 4. 1 Sejarah PT Pos Logistik Indonesia

Sumber: *poslogistics.co.id*

4.1.3 Struktur Organisasi

Struktur organisasi PT Pos Logistik Indonesia terdiri atas Dewan Komisaris dan Dewan Direksi. Dewan Komisaris diisi oleh Henry Panangian Tobing, seorang profesional berpengalaman yang meniti karier di berbagai bidang bisnis dan organisasi, serta Tonggo Marbun, yang memiliki pengalaman luas di sektor logistik dan saat ini juga menjabat sebagai Direktur Bisnis Kurir dan Logistik PT Pos Indonesia (Persero). Sementara itu, Dewan Direksi dipimpin oleh Arifin Muchlis sebagai Pelaksana Tugas (PLT) Direktur PT Pos Logistik Indonesia, yang merupakan lulusan Sarjana Ekonomi dan Magister Manajemen dari Universitas Padjadjaran. Jajaran direksi turut diperkuat oleh Fahdel Akbar yang menjabat sebagai *Chief Sales Officer* sekaligus Pj *Chief Operation Officer*, dengan latar belakang kuat di bidang *Sales* dan *Marketing,* serta Lisa Tinaria yang memegang peranan sebagai *Chief Financial Officer* (PT Pos Logistik Indonesia, 2025).



Gambar 4. 2 Struktur Organisasi PT Pos Logistik Indonesia

Sumber: *poslogistics.co.id*

4.1.4 Visi dan Misi

Visi dan misi atau dapat pula diartikan sebagai maksud dan tujuan yang dimiliki sebuah perusahaan, merupakan sebuah pernyataan yang digunakan sebagai cara mengomunikasikan tujuan dari sebuah organisasi. Berikut merupakan visi dan misi PT Pos Logistik Indonesia.

a. Visi

Menjadi penyedia solusi logistik terpadu yang terpercaya, terluas, dan terkemuka di Indonesia.

b. Misi

- Memberikan solusi logistik yang efisien dan terintegrasi bagi pelanggan serta mendukung daya saing logistik nasional.
- Memberikan kontribusi laba yang maksimal dan membangun sinergi usaha dengan PT Pos Indonesia.
- Membangun kemitraan usaha dengan mitra kerja strategis yang saling menguntungkan.
- Terus berupaya mengembangkan kompetensi karyawan dan organisasi agar memiliki daya saing nasional.

4.1.5 Produk dan Layanan

Dalam menghadapi dinamika industri logistik di Indonesia maupun internasional, PT Pos Logistik Indonesia menyediakan berbagai layanan yang disesuaikan dengan kebutuhan konsumen. Berbagai solusi yang ditawarkan mencakup penyimpanan, distribusi, pengangkutan barang berat, hingga pengelolaan pengiriman domestik dan internasional, serta layanan rantai dingin, dengan mengedepankan fasilitas modern, jaringan luas, dan tenaga ahli profesional. Adapun layanan yang disediakan antara lain (PT Pos Logistik Indonesia, 2025):

A. *Warehouse*

PT Pos Logistik menyediakan solusi penyimpanan yang aman dan efisien dengan fasilitas modern serta *Warehouse Management System* (WMS) yang mendukung *stock management* dan distribusi optimal, baik untuk produk umum maupun produk khusus. Layanan *Warehouse* yang disediakan PT Pos Logistik Indonesia diantaranya sebagai berikut:

1) Dedicated Warehouse

Dedicated Warehouse merupakan layanan penyimpanan yang dirancang khusus bagi pelanggan PT Pos Logistik Indonesia untuk dapat menyesuaikan permintaan kustomisasi

sesuai kebutuhan pelanggan jika pelanggan membutuhkan solusi untuk pergudangan yang fleksibel dan aman.

2) *Multi-User Warehouse*

Multi-User Warehouse merupakan layanan penyimpanan yang telah megikuti standard operasional PT Pos Logistik Indonesia, untuk menyediakan gudang, *warehouseman*, dan kelengkapan gudang lainnya. *Multi-User Warehouse* direkomendasikan jika pelanggan membutuhkan ruang penyimpanan sementara untuk kebutuhan musiman dengan volume penyimpanan tidak terlalu tinggi.

3) *Open Yard Warehouse*

Merupakan layanan pada area terbuka dengan penyimpanan barang di dalam container. Layanan ini direkomendasikan bagia pelanggan yang memiliki produk dalam skala besar dengan jenis produk yang tahan cuaca dan tidak sensitif terhadap lingkungan sekitar.



Gambar 4. 3 Layanan Pergudangan

Sumber: poslogistics.co.id

B. *Distribution*

PT Pos Logistik Indonesia menawarkan solusi distribusi yang andal dan tepat waktu yang memastikan produk sampai ke tujuan atau ke pelanggan secara efisien dengan jaringan luas dan dengan dukungan teknologi yang modern.



Gambar 4. 4 Layanan Distribusi

Sumber: *poslogistics.co.id*

C. *Heavy Lifting*

PT Pos Logistik Indonesia juga menawarkan layanan *lifting* yang mampu memberikan jasa pengiriman barang-barang berat denga naman hingga ke tujuan. PT Pos Logistik dalam jasa *Heavy Lifting* memiliki 5 layanan meliputi:

1) Pemetaan dan Survei Rute

Layanan ini mampu membawa barang berat dengan ukuran besar menuju ke lokasi dengan akses terbatas. PT Pos Logistik dapat juga dapat membantu pelanggan untuk melakukan pemetaan rute hingga survei langsung untuk memastikan rute dapat dilewati dengan aman.

2) Rencana Transportasi dan Operasi

Layanan penyedia pilihan-pilihan rencana dengan berbagai jenis transportasi dan pola operasi yang dapat disesuaikan dengan kebutuhan pelanggan.

3) Dedicated Team

Dalam melakukan pelayanan dalam jasa *Heavy Lifting,* PT Pos Logistik mengedepankan kualitas dengan menyediakan tenaga professional dan berpengalaman dalam *heavy lifting.*

4) Transportasi Multi Moda

Layanan penyedia jasa transportasi dengan pilihan yang berbeda (Multi-Moda) sesuai dengan kebutuhan spesifik konsumen demi mengedepankan efisiensi pengiriman.

5) Visibility dan *Monitoring*

Layanan penyimpanan pada area terbuka dengan layanan *open yard warehouse* dengan penyimpanan barang di dalam *container*. Layanan ini di targetkan kepada pelanggan yang memiliki produk dalam skala besar dengan jenis produk yang tahan cuaca dan tidak sensitif terhadap lingkungan sekitar.



Gambar 4. 5 Layanan *Heavy Lifting*

Sumber: *poslogistics.co.id*

D. *Domestic Cargo*

Untuk layanan yang berjalan setiap hari dan menjadi *main-business* PT Pos Logistik Indonesia menyediakan layanan Kargo Domestik, diantaranya adalah karbo domestik darat dan udara. PT Pos Logistik Indonesia melayanai pengiriman kargo mulai dari 30 kg dengan pengiriman meliputi seluruh area Jawa dan Sumatera.



Gambar 4. 6 Layanan Kargo Domestik

Sumber: *poslogistics.co.id*

E. *International Cargo*

Layanan pengiriman kargo internasional yang disediakan mencakup berbagai moda transportasi dan kebutuhan logistik, meliputi:

1) *Sea & Air Freight (Export – Import)*

   Layanan ini menyediakan solusi pengiriman kargo internasional melalui jalur laut dan udara, baik untuk kebutuhan ekspor maupun impor. Pengiriman dilakukan dengan fleksibilitas metode, mulai dari *Full Container Load* (FTL) yang cocok untuk pengiriman dalam jumlah besar, hingga *Less Than Truck Load* (LTL) untuk pengiriman dalam volume yang lebih kecil namun tetap efisien.

2) *Land freight (Export – Import)*

   Layanan ini dirancang untuk memenuhi kebutuhan pengiriman barang yang fleksibel dan dapat disesuaikan dengan kebutuhan logistik pelanggan, termasuk rute, waktu tempuh, dan jenis kendaraan yang digunakan. Layanan darat ini juga sangat berguna untuk mendistribusikan barang dari pelabuhan atau bandara ke lokasi tujuan akhir secara efisien.

3) Integrated Import Services to Last Mile Delivery

   Proses ini mencakup mulai dari penjemputan barang impor di pelabuhan atau bandara, pengurusan dokumen dan bea cukai, hingga distribusi akhir ke penerima barang. Dengan pendekatan terintegrasi ini, pelanggan tidak perlu mengatur sendiri berbagai proses logistik secara terpisah, karena semua sudah ditangani oleh tim yang profesional.

4) Door to Door Freight Services (Retail Cargo)

   Layanan ini memberikan solusi pengangkutan secara *door-to-door*, yaitu penjemputan barang langsung dari alamat pengirim dan pengantaran ke alamat tujuan di negara lain. Dengan layanan ini, efisiensi dan kenyamanan pelanggan menjadi prioritas utama.

5) Custom Clearance & Delivery Service

   Layanan ini bertujuan untuk membantu pelanggan dalam proses kepabeanan dan perizinan yang berkaitan dengan aktivitas ekspor dan impor. *Custom clearance* merupakan tahap penting dalam proses logistik internasional karena menyangkut pemeriksaan dokumen, pembayaran bea masuk, serta pemenuhan ketentuan hukum di negara asal maupun tujuan. Selain itu, layanan ini juga dapat mencakup pengantaran barang setelah proses *clearance* selesai, sehingga memberikan solusi logistik yang menyeluruh.



Gambar 4. 7 Layanan *International Cargo*

Sumber: *poslogistics.co.id*

F. *Cold Chain Service*

PT Pos Logistik Indonesia menyediakan layanan *cold chain* yang dapat memastikan barang pelanggan dalam temperature yang dibutuhkan untuk menjaga kualitas dan keamanan barang. Layanan *Cold Chain Service* ini meliputi *Flexible Storage* (Layanan *Magic Box* kotak pendingin), *Adjustable Temperature, Multi Temperature Storage, End-to-end Cold Chain (Processing and Shipment)* dan *Control Tower Report.*



Gambar 4. 8 Layanan *Cold Chain Services*

Sumber: *poslogistics.co.id*

4.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini diantaranya merupakan data sekunder yang diperoleh dari PT Pos Logistik Indonesia, hasil wawancara, dan konversi titik menjadi *longitude* dan *latitude* berdasarkan *google maps*. Berikut merupakan data-data tersebut:

4.2.1 Data Kendaraan

PT Pos Logistik memiliki berbagai jenis kendaraan yang digunakan untuk mendistribusikan barang tergantung dari jenis layanan dan kebutuhannya. Jumlah kendaraan yang digunakan merupakan jumlah kendaraan yang dioperasikan pada rute dan area distribusi Jakarta Raya dengan *coverage* 100 titik antar. Kendaraan

yang digunakan dalam objek penelitian ini, yaitu layanan distribusi barang adalah sebagai berikut:

Tabel 4. 1 Jenis Kendaraan

Jenis Kendaraan	Kapasitas	Merk	Jumlah
Blind Van	1000 Kg	*Daihatsu Grand Max*	3

4.2.2 Data Titik Lokasi

Berikut ini merupakan titik-titik lokasi yang digunakan sebagai variabel dalam penelitian ini. Titik tersebut meliputi *Main Hub* (depot) dan *Customer*.

Tabel 4. 2 Data Titik Lokasi

No Index	Nama Node	Lokasi	Latitude	Longitude
0	Main Hub	Kawasan Industri JIEP Cakung, Jakarta Timur.	106.916742	-6.202523
1	Customer 1	Jakarta Raya	106,835953	-6,280092
2	Customer 2	Jakarta Raya	106,887305	-6,281572
3	Customer 3	Jakarta Raya	106,819235	-6,279968
4	Customer 4	Jakarta Raya	106,850077	-6,233342
5	Customer 5	Jakarta Raya	106,774135	-6,269443
6	Customer 6	Jakarta Raya	106,809525	-6,212111
7	Customer 7	Jakarta Raya	106,845133	-6,292434
8	Customer 8	Jakarta Raya	106,881958	-6,29109
9	Customer 9	Jakarta Raya	106,793436	-6,267833
10	Customer 10	Jakarta Raya	106,891976	-6,289595
11	Customer 11	Jakarta Raya	106,898122	-6,209064
12	Customer 12	Jakarta Raya	106,881188	-6,225719
13	Customer 13	Jakarta Raya	106,809656	-6,269489
14	Customer 14	Jakarta Raya	106,823482	-6,23099
15	Customer 15	Jakarta Raya	106,888708	-6,218146
16	Customer 16	Jakarta Raya	106,885018	-6,287387

17	Customer 17	Jakarta Raya	106,861826	-6,265529
18	Customer 18	Jakarta Raya	106,844098	-6,240286
19	Customer 19	Jakarta Raya	106,881953	-6,222611
20	Customer 20	Jakarta Raya	106,825788	-6,197719
21	Customer 21	Jakarta Raya	106,836412	-6,251159
22	Customer 22	Jakarta Raya	106,896347	-6,281991
23	Customer 23	Jakarta Raya	106,871795	-6,18419
…	…	…	…	…
100	Customer 100	Jakarta Raya	106,849477	-6,260628

Pemilihan 100 titik *customer* dengan *coverage* sebaran merata di daerah Jakarta Raya didasari oleh beberapa alasan, seperti variasi jarak serta titik yang dinamis dan kompleksitas lalu lintas yang tinggi. Alasan tersebut mampu memperkuat model VRP yang lebih realistis.

4.2.3 Data *Demand* Pengiriman Barang

Dalam konteks *Capacitated* VRP, permintaan *customer* atau *demand* yang berupa berat (Kg) barang menjadi *constraint* utama. Berat barang harus dipertimbangkan dalam tahapan pemodelan supaya dapat memenuhi *constraint* kapasitas kendaraan *Blind Van* yaitu 1000 Kg. Berikut merupakan data permintaan barang 100 pelanggan di area Jakarta:

Tabel 4. 3 Data *Demand* Pengiriman Barang

No Index	Nama Node	Delivery Demand
0	Main Hub	-
1	Customer 1	29
2	Customer 2	22
3	Customer 3	13
4	Customer 4	21
5	Customer 5	23
6	Customer 6	7
7	Customer 7	16

8	Customer 8	28
9	Customer 9	30
10	Customer 10	21
11	Customer 11	24
12	Customer 12	2
13	Customer 13	26
14	Customer 14	12
15	Customer 15	5
16	Customer 16	13
…	…	…
100	Customer 100	3

4.2.4 Data *Time Windows*

Time Windows merupakan *constraint* utama dalam kasus CVRPTW yang merupakan waktu layanan setiap titik yang akan di tuju. Dalam PT Pos Logistik Indonesia, pelanggan memiliki pilihan waktu pengiriman yang sudah disediakan, seperti pada Tabel 4.4.

Tabel 4. 4 Data Waktu Layanan PT Pos Logistik Indonesia

Time	***TW***	***Erl. Minutes***	***Lst. Minutes***
08.00-13.00	1	480	780
13.01-17.00	2	781	1020
17.01-21.00	3	1021	1260

Kemudian untuk Batasan waktu pengiriman yang harus dipenuhi dalam proses pengiriman barang ke pelanggan adalah sebagai berikut:

Tabel 4. 5 Data *Time Windows*

No Index	***Time Window (Min)***	***Time Window (Clock)***	**Kategori**
0	480-1260	08:00-21:00	Depot
1	781-1020	13:01-17:00	TW 2
2	781-1020	13:01-17:00	TW 2
3	1021-1260	17:01-21:00	TW 3

4	781-1020	13:01-17:00	TW 2
5	781-1020	13:01-17:00	TW 2
6	1021-1260	17:01-21:00	TW 3
7	781-1020	13:01-17:00	TW 2
8	781-1020	13:01-17:00	TW 2
9	781-1020	13:01-17:00	TW 2
10	1021-1260	17:01-21:00	TW 3
11	1021-1260	17:01-21:00	TW 3
12	781-1020	13:01-17:00	TW 2
13	781-1020	13:01-17:00	TW 2
...	...	...	...
100	1021-1260	17:01-21:00	TW 3

4.2.5 Data Pendukung

Selain data titik Lokasi, permintaan dan *time windows*, terdapat beberapa data pendukung yang menjaadi konstanta dalam penelitian ini diantaranya sebagai berikut:

Tabel 4. 6 Data Pendukung Penelitian

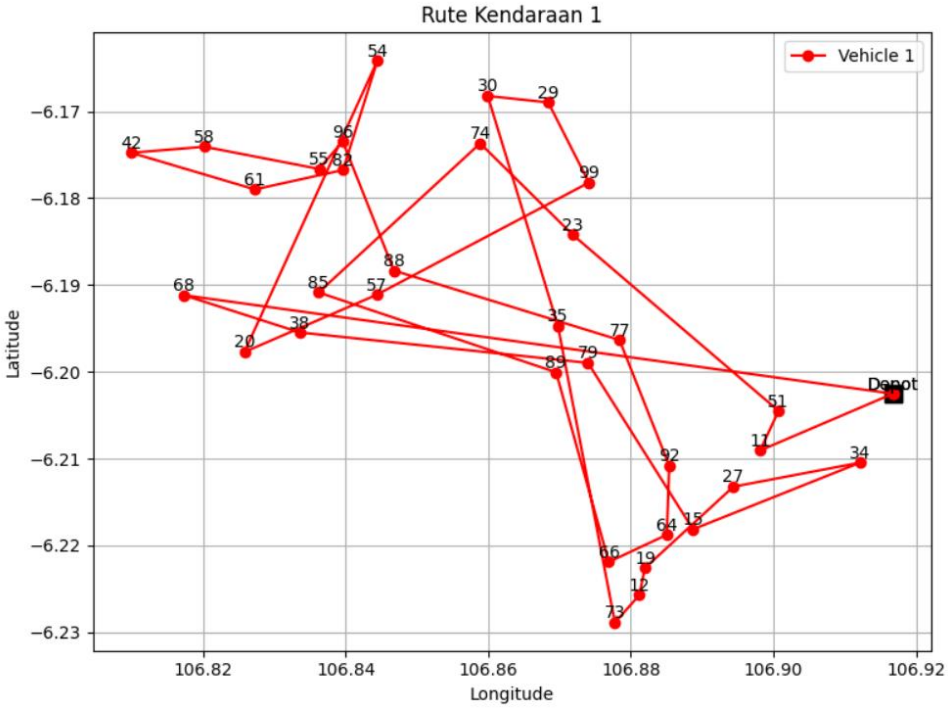
Nama Data	Nilai	Unit
Rata-rata kecepatan kendaraan	22	Km/ Jam
Rata-rata *Loading Time* /Titik	5	Menit
Ongkos Kirim /*Customer*	Rp 10.000,00	Rupiah (*Flat*)
Penalty Keterlambatan	20% / Biaya Ongkir	Rupiah (*Flat)*

4.2.6 Rute Eksisting Kebijakan Perusahaan

Dalam menentukan rute distribusi barang, PT Pos Logistik Indonesia memiliki kebijakan dalam penentuannya menggunakan metode subjektif supir dengan urutan *time windows* yang paling awal. Berdasarkan proses pengumpulan data, diambil rute distribusi yang digunakan PT Pos Logistik Indonesia di area Jakarta. Terdapat 3 kendaraan yang digunakan sesuai titik di Jakarta.

A. Rute Kendaraan 1

Berikut merupakan rute yang terbentuk oleh kendaraan 1 selama melakukan distribusi barang dari *Main Hub* (Depot) ke *customer-customer*:



Gambar 4. 9 Plot Rute Eksisting Kendaraan 1

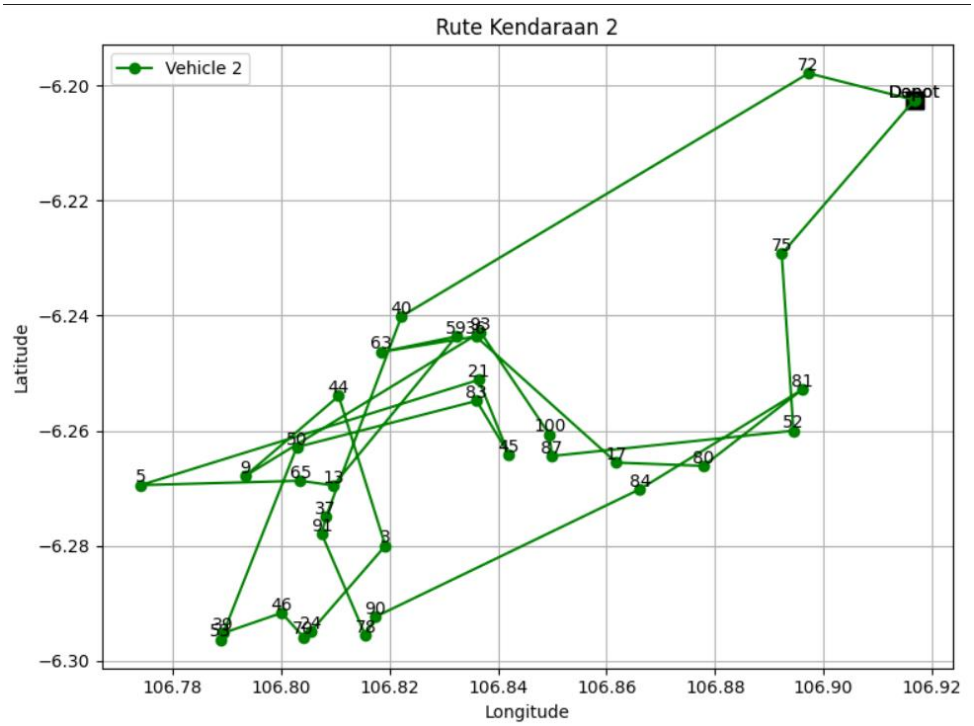
Tabel 4. 7 Total Jarak dan Waktu Tempuh Kendaraan 1

Kendaraan	**Rute Terbentuk**	**Total Jarak (Km)**	**Total Waktu Tempuh (Min)**	**Total Biaya Penalty (Rp)**
Kendaraan 1	0 - 68 - 38 - 79 - 15 - 34 - 27 - 19 - 12 - 73 - 35 - 30 - 29 - 99 - 57 - 20 - 54 - 82 - 61 - 42 - 58 - 55 - 96 - 88 - 77 - 92 - 64 - 66 - 89 - 85 - 74 - 23 - 51 - 11 - 0	131.31	743.15	2 x Rp. 2000 = Rp. 4000

Total titik yang dilewati oleh kendaraan 1 berjumlah 33 *customer* dengan total jarak tempuh sepanjang 131,31 Km dan estimasi total waktu tempuh perjalanan selama 743,15 menit atau 12,3 Jam. Dengan total jarak dan waktu tempuh tersebut dihasilkan biaya *penalty* sebesar Rp. 4000.

B. Rute Kendaraan 2

Berikut merupakan rute yang terbentuk oleh kendaraan 2 selama melakukan distribusi barang dari *Main Hub* (Depot) ke *customer-customer*:



Gambar 4. 10 Plot Rute Eksisting Kendaraan 2

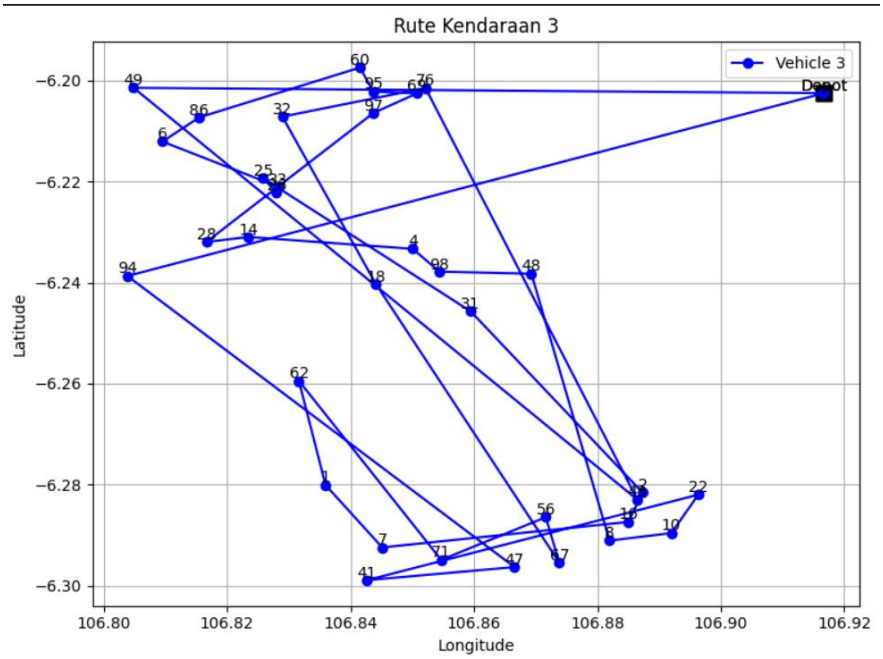
Tabel 4. 8 Total Jarak dan Waktu Tempuh Kendaraan 2

Kendaraan	**Rute Terbentuk**	**Total Jarak (Km)**	**Total Waktu Tempuh (Min)**	**Total Biaya Penalty (Rp)**
Kendaraan 2	0 - 72 - 40 - 37 - 91 - 78 - 90 - 84 - 81 - 80 - 17 - 36 - 63 - 59 - 13 - 65 - 5 - 21 - 45 - 83 - 50 - 53 - 39 - 46 - 70 - 24 - 3 - 44 - 9 - 93 - 100 - 87 - 52 - 75 - 0	154.19	799.51	5 x Rp. 2000 = Rp. 10.000

Total titik yang dilewati oleh kendaraan 1 berjumlah 33 *customer* dengan total jarak tempuh sepanjang 154,19 Km dan estimasi total waktu tempuh perjalanan selama 799,51 menit atau 13, 3 Jam. Dengan total jarak dan waktu tempuh tersebut dihasilkan biaya *penalty* sebesar Rp. 10.000.

C. Rute Kendaraan 3

Berikut merupakan rute yang terbentuk oleh kendaraan 3 selama melakukan distribusi barang dari *Main Hub* (Depot) ke *customer-customer*:



Gambar 4. 11 Plot Rute Eksisting Kendaraan 3

Tabel 4. 9 Total Jarak dan Waktu Tempuh Kendaraan 3

Kendaraan	**Rute Terbentuk**	**Total Jarak (Km)**	**Total Waktu Tempuh (Min)**	**Total Biaya Penalty (Rp)**
Kendaraan 3	0 - 49 - 43 - 76 - 32 - 18 - 67 - 56 - 51 - 72 - 62 - 1 - 7 - 16 - 2 - 25 - 26 - 33 - 6 - 86 - 60 - 95 - 69 - 97 - 28 - 14 - 48 - 48 - 8 - 10 - 22 - 41 - 47 - 94 - 0	214.60	902.32	9 x Rp. 2000 = Rp. 18.000

Total titik yang dilewati oleh kendaraan 1 berjumlah 34 *customer* dengan total jarak tempuh sepanjang 214, 60 Km dan estimasi total waktu tempuh perjalanan selama 902.32 menit atau 15, 04 Jam. Dengan total jarak dan waktu tempuh tersebut dihasilkan biaya *penalty* sebesar Rp. 18.000. Berikut merupakan rangkuman jarak, waktu tempuh serta biaya penalty yang dihasilkan dalam proses distribusi tersebut:

Tabel 4. 10 *Grand Total* Rute Eksisting Perusahaan

Grand Total Distance (Km)	***Grand Total Travel Time (Min)***	***Grand Total Penalty Cost (Rp)***
500,11	2444,99	Rp. 32.000

4.3 Pengolahan Data

4.3.1 Pengembangan Model Konseptual

Proses pengembangan model konseptual bertujuan untuk mendefinisikan setiap langkah dan tahapan penelitian guna memastikan konsep sesuai dan mampu membangun kerangka model yang sesuai dengan permasalahan. Berikut ini merupakan langkah pengembangannya.

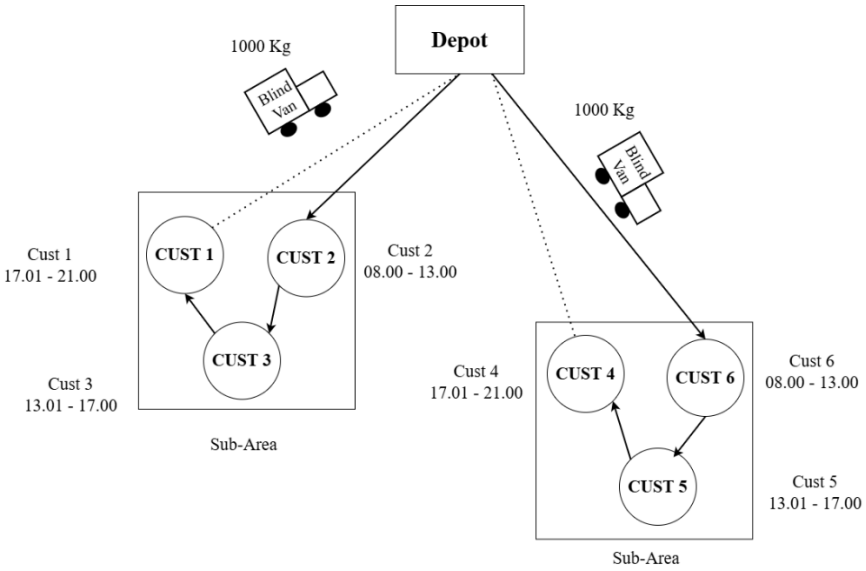
A. Uraian Sistem dan Batasan

Dalam penelitian VRP ini terdapat beberapa pengertian dan pemahaman seperti maksud dari *demand* yang merupakan jumlah barang yang harus diantarkan kepada *customer* dan *service time* yang merupakan waktu *loading/unloading* barang ke kendaraan. Namun terdapat beberapa karakteristik dan kondisi yang berlaku dalam sistem VRP pada penelitian ini, diantaranya:

- Proses distribusi pada area Jakarta mengoperasikan 3 kendaraan dengan asumsi penggunaan Batasan *homogeneous fleet*. Kendaraan yang digunakan memiliki karakteristik dan jenis yang sama yaitu *Blind Van* dengan Batasan kapasitas sebanyak 1000 Kg.
- Titik keberangkatan kendaraan dimulai dari *Main Hub* (Depot) yaitu Gudang utama yang berlokasi di Kawasan Industri JIEP Cakung kemudian menjalankan distribusinya ke beberapa titik di area Jakarta.
- Barang yang diangkut oleh kendaraan tidak boleh melebihi 80% Batasan Kapasitas kendaraan (1000 Kg). Jika kapasitas penuh, maka konsolidasikan barang ke kendaraan lain.
- Proses distribusi harus mematuhi jam Batasan layanan atau *time window*, baik Batasan waktu perusahaan atau *customer*. Kendaraan yang terlambat tiba di titik antar *customer* akan diberlakukan *penalty* sebesar 20% dari total

biaya ongkos kirim yaitu Rp. 10.000 *flat* atau dengan total biaya *penalty* sebesar Rp. 2000 *flat* per keterlambatan.

- 1 Kendaraan wajib melayani 1 area khusus dan tidak boleh mengunjungi titik area lain. Kendaraan merepresentasikan jumlah area, begitu pun sebaliknya. Jumlah rute yang terbentuk dari jumlah kendaraan akan membentuk pembagian area dalam proses pengiriman.



Gambar 4. 12 Skema Model Konseptual

4.3.2 Pengembangan Program dan Model Matematis

Berdasarkan model konseptual yang telah di kembangkan sebelumnya, selanjutnya adalah mendefinisikan kebutuhan perhitungan dengan mengembangkan model matematis. Penelitian ini merupakan klasifikasi VRP Bernama *Capacitated Vehicle Routing Problem with Time Window*. Model yang dikembangkan merupakan kebutuhan Perusahaan yang didasari oleh Permasalahan yang kemudian dikembangkan kembali oleh peneliti dalam proses penyelesaiannya. Berikut ini merupakan model-model matematis yang digunakan:

A. Notasi

Berikut ini merupakan set indeks dan juga parameter-parameter yang digunakan dalam penelitian VRP ini, seperti notasi parameter kapasitas, kecepatan kendaraan, *service time* dan lain sebagainya.

Berikut ini merupakan parameter pada penelitian ini dalam bentuk *pseudo code python:*

```
Deklarasikan parameter utama untuk model:
    kapasitas_kendaraan ← 1000
    kecepatan_kendaraan ← 22  // satuan: km per jam
    waktu_pelayanan ← 5     // satuan: menit
    biaya_per_pelanggan ← 10000
    tingkat_penalti ← 0.2     // 20%
    id_depot ← 0
```

Gambar 4. 13 *Pseudo Code* Parameter

B. Fungsi *Library* Umum

Dalam penelitian ini, digunakan beberapa pustaka (*library*) *Python* yang berperan penting dalam proses perhitungan, optimasi, serta visualisasi hasil. Penjelasan dari masing-masing *library* adalah sebagai berikut:

1) *Math*

Library `(math)` digunakan untuk perhitungan matematis dasar, khususnya dalam perhitungan jarak menggunakan rumus *Haversine*. Fungsi-fungsi seperti *radians, sin, cos, sqrt*, dan *atan2* berasal dari *library* ini.

2) *Random*

Library `(random)` ini digunakan untuk menghasilkan angka acak, seperti dalam proses pembentukan individu awal, seleksi kromosom, mutasi, dan *crossover* dalam *Genetic Algorithm*.

3) *Numpy* (Np)

Library `(numpy)` digunakan untuk pengolahan data numerik secara efisien. Pada penelitian ini, *library numpy* digunakan untuk mengatur *seed* acak (`np.random.seed`) dan juga dalam beberapa perhitungan pada fungsi optimasi.

- Fungsi *Seed* Acak (`np.random.seed`) bertujuan ntuk mengatur nilai awal generator bilangan acak dari *library NumPy*, sehingga setiap kali algoritma dijalankan, hasil eksperimen tetap konsisten dan dapat direproduksi.

4) *Pandas* (Pd)

Library ini digunakan untuk pengolahan data dalam format tabular. Dalam penelitian ini digunakan untuk membantu pencatatan data atau log evolusi, serta dapat digunakan untuk ekspor data ke format seperti *Excel* untuk analisis lanjutan.

5) *Matplotlib.pyplot* (plt)

Digunakan untuk memvisualisasikan hasil rute dari setiap kendaraan. Fungsi *plot, scatter, text*, dan *legend* digunakan untuk menggambarkan jalur rute, lokasi depot, titik pelanggan, serta keterangan visual lainnya.

C. Fungsi Perhitungan Jarak dan Waktu tempuh

Berikut ini merupakan proses perhitungan Jarak dan Waktu tempuh pada penelitian ini. Proses perhitungan meliputi 3 langkah, menghitung jarak *haversine*, faktor koreksi dan waktu tempuh.

1) Program Fungsi *Haversine*

Berikut ini merupakan *code* yang digunakan untuk menghitung jarak dengan metode *haversine*:

```
Fungsi HITUNG_JARAK(a, b, koreksi = True)
    Input:
        a = (longitude1, latitude1)
        b = (longitude2, latitude2)
        koreksi = True (default)
    Langkah:
        Ubah longitude dan latitude dari titik a dan b ke
dalam radian:
            lon1 ← Radian(a.longitude)
            lat1 ← Radian(a.latitude)
            lon2 ← Radian(b.longitude)
            lat2 ← Radian(b.latitude)

        Hitung perbedaan koordinat:
            dlon ← lon2 - lon1
            dlat ← lat2 - lat1
        Hitung nilai A:
            A ← sin(dlat / 2)^2 + cos(lat1) * cos(lat2) *
sin(dlon / 2)^2
```

```
Hitung nilai C:
    C ← 2 * atan2(√A, √(1 - A))
Tetapkan jari-jari bumi:
    R ← 6371 km
Hitung jarak Haversine:
    jarak ← R * C
    Output:
Jarak antar titik dalam satuan kilometer
```

Gambar 4. 14 *Pseudo Code Haversine*

2) Program Fungsi Faktor Koreksi *ETA*

Untuk memperoleh nilai faktor koreksi memerlukan proses komputasi *code* menggunakan *python* dengan rumus fungsi 4.2. Data yang digunakan adalah jarak *haversine* dan 20 sampel pasangan data jarak *real road* yang diperoleh dari *maps*. Faktor koreksi ditentukan dari sebagian kecil sampel, lalu digunakan untuk mengoreksi semua jarak *haversine* untuk mempersingkat waktu.

Proses komputasi dilakukan dengan cara mendaftarkan akun dan menyalin API Key pada *website* https://openrouteservice.org/dev/#/signup. Kemudian memasukkan API Key pada *Python Code,* memasukan input titik pasangan *longitude* dan *latitude* dan rumus lainnya untuk perhitungan memperoleh faktor koreksi.

```
Inisialisasi list haversine_list sebagai list kosong
Inisialisasi list realroad_list sebagai list kosong

Untuk setiap pasangan koordinat dari index 0 hingga n-2:
    Ambil titik awal (start) dan titik akhir (end)

    Hitung jarak haversine antara start dan end, simpan ke
haversine_list

    Coba:
        Ambil rute jalan dari OpenRouteService (ORS) antara
start dan end
```

```
            Ambil jarak jalan dari properti rute (dalam satuan
    km)
            Tambahkan nilai tersebut ke realroad_list
        Jika terjadi error:
            Tampilkan pesan error
            Tambahkan None ke realroad_list

    Inisialisasi list factors sebagai list kosong

    Untuk setiap pasangan nilai (haversine, road) dalam
    haversine_list dan realroad_list:
        Jika nilai haversine > 0 dan road tidak kosong:
            Hitung rasio road / haversine, tambahkan ke factors

    Jika factors tidak kosong:
        Hitung rata-rata dari factors
        Tampilkan rata-rata faktor koreksi
    Else:
        Tampilkan bahwa tidak ada data valid untuk koreksi
```

Gambar 4. 15 Model Fungsi Koreksi ETA

Berdasarkan model tersebut diperoleh nilai faktor koreksi yang akan digunakan dalam perhitungan untuk memperoleh *distance matrix*. Faktor koreksi tersebut merupakan *trade-off* antara akurasi dan efisiensi yang dapat digunakan sebagai alternatif perhitungan jarak secara manual melalui *google maps*. Nilai faktor koreksi diperoleh dari perhitungan selisih atau *gap* antara jarak *haversine* dan *real road* pada 10 sampel titik pelanggan. Rasio tersebut kemudian digunakan untuk memperoleh waktu tempuh untuk 100 titik pelanggan. Berikut merupakan nilai faktor koreksi tersebut:

Tabel 4. 11 Nilai Faktor Koreksi ETA

Sampel Pasangan	***Haversine***	**Sampel Data *Real Road***	***Dengan Fk***
Pasangan 1	12,41 km	19,9 km	19,89 km
Pasangan 2	9,37 km	13,8 km	14 km
Pasangan 3	13,79 km	21,9 km	22 km

Sampel Pasangan	***Haversine***	**Sampel Data *Real Road***	***Dengan Fk***
Pasangan 4	8,12 km	13 km	13 km
Pasangan 5	17,43 km	26,5 km	27,2 km
…	…	…	
Pasangan 10	10 km	15,9 km	15,5 km
Rata-rata faktor koreksi (*Fk*)	**1.600**		

Berdasarkan 2 fungsi perhitungan tersebut, akan diperoleh hasil matriks jarak antar titik dan depot berdasarkan titik geografis yang tersedia seperti pada tabel 4.12 dan 4.13 berikut:

Tabel 4. 12 *Distance Matrix Node 0-100*

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**…**	**100**
0	0	12,42	9,37	13,80	8,13	17,43	11,90	12,75	10,57	15,44	10,06	2,18	4,70	13,98	10,78	3,55	…	9,85
1	12,42	0	5,68	1,85	5,43	6,93	8,10	1,71	5,23	4,89	6,28	10,47	7,85	3,14	5,63	9,02	…	2,63
2	9,37	5,68	0	7,53	6,76	12,58	11,56	4,82	1,21	10,49	1,03	8,15	6,25	8,69	9,02	7,05	…	4,79
3	13,80	1,85	7,53	0	6,20	5,12	7,62	3,18	7,04	3,15	8,11	11,76	9,13	1,57	5,47	10,31	…	3,97
4	8,13	5,43	6,76	6,20	0	9,30	5,07	6,59	7,32	7,34	7,78	5,96	3,54	6,01	2,95	4,59	…	3,03
5	17,43	6,93	12,58	5,12	9,30	0	7,48	8,25	12,16	2,14	13,22	15,26	12,79	3,93	6,93	13,89	…	8,39
6	11,90	8,10	11,56	7,62	5,07	7,48	0	9,76	11,88	6,45	12,54	9,80	8,06	6,38	2,61	8,78	…	6,97
7	12,75	1,71	4,82	3,18	6,59	8,25	9,76	0	4,07	6,33	5,19	10,97	8,42	4,68	7,24	9,56	…	3,57
8	10,57	5,23	1,21	7,04	7,32	12,16	11,88	4,07	0	10,12	1,12	9,29	7,27	8,34	9,30	8,15	…	4,94
9	15,44	4,89	10,49	3,15	7,34	2,14	6,45	6,33	10,12	0	11,16	13,29	10,77	1,80	5,27	11,89	…	6,25
10	10,06	6,28	1,03	8,11	7,78	13,22	12,54	5,19	1,12	11,15	0	8,98	7,20	9,37	9,99	7,95	…	5,70
11	2,18	10,47	8,15	11,76	5,96	15,26	9,80	10,97	9,29	13,29	8,98	0	2,63	11,86	8,60	1,45	…	7,86
12	4,70	7,85	6,25	9,13	3,54	12,79	8,06	8,42	7,27	10,77	7,20	2,63	0	9,28	6,41	1,18	…	5,23
13	13,98	3,14	8,69	1,57	6,01	3,93	6,38	4,68	8,34	1,80	9,37	11,86	9,28	0	4,55	10,44	…	4,51
14	10,78	5,63	9,02	5,47	2,95	6,93	2,61	7,24	9,30	5,27	9,99	8,60	6,41	4,55	0	7,35	…	4,37
15	3,55	9,02	7,05	10,31	4,59	13,89	8,78	9,56	8,15	11,89	7,95	1,45	1,18	10,44	7,35	0	…	6,41
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	4,07
100	9,85	2,63	4,79	3,97	3,03	8,39	6,97	3,57	4,94	6,25	5,70	7,86	5,23	4,51	4,37	6,41	…	0

Tabel 4. 13 *Distance Matrix* dengan Faktor Koreksi 0-100

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**…**	**100**
0	0	19,86	15,00	22,07	13,00	27,89	19,04	20,40	16,92	24,71	16,10	3,49	7,52	22,37	17,25	5,68	**…**	15,76
1	19,86	0	9,09	2,96	8,68	11,10	12,97	2,73	8,37	7,83	10,05	16,75	12,55	5,02	9,01	14,44	**…**	4,21
2	15,00	9,09	0	12,04	10,82	20,13	18,49	7,70	1,94	16,78	1,65	13,04	10,00	13,90	14,44	11,29	**…**	7,66
3	22,07	2,96	12,04	0	9,93	8,19	12,19	5,09	11,27	5,05	12,98	18,81	14,60	2,52	8,75	16,49	**…**	6,36
4	13,00	8,68	10,82	9,93	0	14,89	8,11	10,55	11,72	11,75	12,45	9,53	5,67	9,62	4,72	7,35	**…**	4,86
5	27,89	11,10	20,13	8,19	14,89	0	11,97	13,21	19,45	3,43	21,15	24,42	20,47	6,28	11,09	22,22	**…**	13,42
6	19,04	12,97	18,49	12,19	8,11	11,97	0	15,62	19,01	10,31	20,07	15,68	12,90	10,21	4,17	14,05	**…**	11,15
7	20,40	2,73	7,70	5,09	10,55	13,21	15,62	0	6,52	10,14	8,30	17,54	13,47	7,49	11,58	15,30	**…**	5,71
8	16,92	8,37	1,94	11,27	11,72	19,45	19,01	6,52	0	16,19	1,79	14,87	11,63	13,35	14,88	13,03	**…**	7,90
9	24,71	7,83	16,78	5,05	11,75	3,43	10,31	10,136	16,19	0	17,85	21,26	17,23	2,88	8,44	19,03	**…**	9,99
10	16,10	10,05	1,65	12,98	12,45	21,15	20,07	8,30	1,79	17,85	0	14,37	11,52	14,99	15,98	12,72	**…**	9,11
11	3,49	16,75	13,04	18,81	9,53	24,42	15,68	17,54	14,87	21,26	14,37	0	4,21	18,98	13,77	2,32	**…**	12,58
12	7,52	12,55	10,00	14,60	5,67	20,47	12,90	13,47	11,63	17,23	11,52	4,21	0	14,86	10,25	1,89	**…**	8,37
13	22,37	5,02	13,90	2,52	9,62	6,28	10,21	7,49	13,35	2,88	14,99	18,98	14,86	0	7,27	16,70	**…**	7,22
14	17,25	9,01	14,44	8,75	4,72	11,09	4,17	11,58	14,88	8,44	15,98	13,77	10,25	7,27	0	11,76	**…**	7,00
15	5,68	14,44	11,29	16,49	7,35	22,22	14,05	15,30	13,03	19,03	12,72	2,32	1,89	16,70	11,76	0	**…**	10,26
…	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	7,88
100	15,76	4,21	7,66	6,36	4,86	13,42	11,15	5,71	7,90	9,99	9,11	12,58	8,37	7,22	7,00	10,26	**…**	0

3) Program Fungsi Waktu Tempuh (ETA)

Berikut merupakan proses memperoleh waktu tempun pada penelitian ini:

```
Function TravelTime(distance_in_km)
    Konversi jarak ke mil: jarak_dalam_mil = 1.600 × jarak_dalam_km
    Hitung waktu perjalanan (jam): waktu_dalam_jam = jarak_dalam_mil
/ kecepatan_kendaraan
    Konversi ke menit: waktu_dalam_menit = waktu_dalam_jam × 60
    Tambahkan waktu layanan: total_waktu = waktu_dalam_menit +
waktu_layanan
    Kembalikan total_waktu
Akhiri Fungsi
```

Gambar 4. 16 *Pseudo Code Travel Time (ETA)*

[travel_time] merupakan jumlah seluruh waktu yang ditempuh oleh kendaraan, [distance_km] merupakan akumulasi jarak tempuh kendaraan dalam satuan kilometer, [1.600] merupakan nilai faktor koreksi dalam perhitungan ETA yang didapatkan dari hasil perhitungan rata-rata selisih jarak *haversine* dan jarak aktual via *maps*, [vehicle_speed] merupakan kecepatan rata-rata yang digunakan kendaraan dalam satuan km/jam, [service_time] merupakan waktu layanan/*loading* yang diberikan untuk setiap kendaraan dalam menit dan [60] merupakan pengali untuk konversi km/jam menjadi menit.

D. Program *Multi-Objective Function*

Untuk memenuhi beberapa fungsi objektif dalam penelitian ini yaitu minimasi jarak tempuh dan minimasi penalti akibat keterlambatan, terdapat beberapa fungsi yang digunakan untuk melakukan optimasi tersebut. Berikut ini merupakan *script code python* yang digunakan untuk menghitung *rule* penalti dari rute yang terbentuk:

```
Fungsi HitungBiayaRute(rute)
    total_dist ← HitungTotalJarak(rute)
    total_penalty ← HitungTotalPenalti(rute)
    total_cost ← total_jarak + total_penalti
    Kembalikan total_cost
Akhiri Fungsi
```

Gambar 4. 17 *Pseudo Code Multi-Objective Function*

`[total_dist]` merupakan Jumlah seluruh jarak dari rute yang ditempuh kendaraan, `[total_penalty]` merupakan akumulasi penalti karena keterlambatan pengiriman dan/atau *overload* kapasitas.

E. Program *Constraint*

Untuk memenuhi hambatan dalam penelitian ini, seperti kapasitas, *time windows* dan lainnya, terdapat beberapa model yang dibangun diantaranya sebagai berikut:

- Setiap *customer* hanya dikunjungi sekali oleh satu kendaraan,
- Kapasitas kendaraan tidak boleh dilampaui,
- *Time windows* harus dihormati (penalti jika lewat batas akhir),
- Kendaraan harus berangkat dari depot dan kembali lagi ke depot ($0 \rightarrow 0$),
- Konservasi aliran (*Flow Conservation*), memastikan rute tidak putus.

```
a) Inisialisasi rute dengan [ID_Depot]
Inisialisasi daftar pelanggan_belum_dikunjungi dengan semua
pelanggan selain depot
Inisialisasi total_penalti = 0
Inisialisasi muatan_kendaraan = 0

b) Untuk setiap pelanggan dalam pelanggan_belum_dikunjungi:
    Jika (muatan_kendaraan + permintaan_pelanggan) >
kapasitas_kendaraan:
        Lewati pelanggan ini

    Jika muatan_kendaraan > kapasitas_kendaraan:
        Tambahkan penalti_berat (misal 1.000.000) ke
total_penalti

c)  Hitung waktu_kedatangan ke pelanggan

    Jika waktu_kedatangan < waktu_mulai_TW:
        Tunggu sampai waktu_mulai_TW

    Hitung keterlambatan = maksimum(0, waktu_kedatangan -
waktu_selesai_TW)
```

```
        Tambahkan keterlambatan ke total_penalti jika ada

        Tambahkan pelanggan ke rute
        Kurangi pelanggan dari daftar belum_dikunjungi

    d) Jika titik terakhir dalam rute bukan depot:
        Tambahkan depot ke akhir rute

    e) Untuk setiap pelanggan yang tersisa dalam
       belum_dikunjungi:
        Periksa apakah pelanggan feasible (misal, kapasitas dan
    TW sesuai)
        Jika ya, tambahkan ke daftar_pelanggan_feasible
```

Gambar 4. 18 *Pseudo Code Constraint*

4.3.3 Pengembangan Algoritma

Pengembangan algoritma pada penelitian ini dilakukan dengan pendekatan dua tahap utama, yaitu *heuristic constructive method* menggunakan *Nearest Insertion* (NI) untuk membentuk rute awal, dan *metaheuristic improvement method* menggunakan *Genetic Algorithm* (GA) untuk mengoptimalkan urutan kunjungan dalam setiap rute kendaraan. Pendekatan ini dirancang untuk mengatasi permasalahan *Capacitated Vehicle Routing Problem with Time Windows* (CVRPTW) secara efisien, dengan mempertimbangkan keterbatasan kapasitas kendaraan, batas waktu layanan (*time windows*), serta penalti atas keterlambatan.

A. Implementasi Program *Nearest Insertion* (*NI)*

Tahapan pertama dalam membentuk solusi awal adalah menggunakan metode *Nearest Insertion* (NI), yang bersifat konstruktif dan berbasis *greedy heuristic*. Proses NI bertujuan untuk mengalokasikan pelanggan secara progresif ke dalam rute kendaraan berdasarkan kedekatan waktu dan kelayakan kapasitas. Algoritma NI dimulai dengan memilih depot sebagai titik awal rute. Proses *Nearest Insertion* (NI) dilakukan dengan Langkah-langkah sebagai berikut:

1) Inisialisasi Rute

Algoritma NI dimulai dengan memilih depot sebagai titik awal rute. Mulai dari depot, set semua *customer* sebagai *unassigned*, kemudian batasi jumlah *customer* per kendaraan.

```
Inisialisasi daftar unassigned sebagai semua ID customer
dari data pickup_delivery_demand
Hilangkan ID depot dari daftar unassigned

Hitung maksimum customer per kendaraan:
    Jika jumlah customer > 0:
        max_customer_per_kendaraan = pembagian jumlah
customer dengan jumlah kendaraan
Jika hasilnya 0, gunakan nilai minimum 1
```

Gambar 4. 19 *Pseudo Code* Inisialiasi Rute

Pelanggan yang belum terlayani (*unassigned customers*) akan dipertimbangkan untuk dimasukkan ke dalam rute saat ini apabila kapasitas kendaraan belum terlampaui dan waktu kedatangan masih dalam batas *time windows*.

2) Inisialisasi Kendaraan

Pada tahapan ini, untuk tiap kendaraan mulai rute dari depot, *load* = 0, waktu awal = *time window* depot.

```
Untuk setiap kendaraan dengan ID dari 1 sampai
jumlah_kendaraan:
    Atur waktu saat ini sebagai waktu mulai dari time
window depot
    Atur lokasi saat ini sebagai depot
    Set muatan kendaraan saat ini menjadi 0
Inisialisasi rute dengan titik awal di depot
```

Gambar 4. 20 *Pseudo Code* Inisialiasi Kendaraan

3) Cari *Customer Feasible*

Untuk setiap *customer* yang belum dialokasikan, lakukan pemeriksaan kapasitas muatan dan waktu kedatangan.

```
a) Untuk setiap pelanggan dalam daftar yang belum
   dikunjungi:
    Hitung jarak dan waktu tempuh dari lokasi saat ini ke
pelanggan
```

```
      b) Jika muatan saat ini + permintaan pelanggan melebihi
         kapasitas kendaraan:
              Lewati pelanggan ini (lanjut ke pelanggan
      berikutnya)

          Hitung waktu kedatangan ke pelanggan

      c) Jika waktu kedatangan melebihi batas akhir time window
         pelanggan:
              Hitung penalti keterlambatan

          Tambahkan pelanggan ke daftar pelanggan yang layak
      dikunjungi,
    beserta waktu kedatangan, jarak, dan penalti
```

Gambar 4. 21 *Pseudo Code* Pencarian *Customer Feasibility*

4) Pilih *Customer* Terbaik dan Sisipkan

Pada tahap ini adalah proses pemilihan *customer* dengan waktu tiba tercepat (atau kriteria lain: biaya/*penalty* minimal). Kemudian dari proses pemilihan tersebut, ambahkan ke rute, update waktu, *load*, *distance*, dan *penalty*.

```
      a) Urutkan daftar pelanggan yang layak (feasible_customers)
      berdasarkan waktu kedatangan paling awal

      b) Pilih pelanggan pertama dari daftar (yang paling cepat
      bisa dikunjungi):
          cust ← ID pelanggan
          arrival ← waktu kedatangan
          dist ← jarak tempuh
          penalty ← penalti keterlambatan (jika ada)

      Tambahkan cust ke dalam rute kendaraan
      Tambahkan permintaan pelanggan ke total muatan kendaraan
      Tambahkan dist ke total jarak tempuh kendaraan
      Tambahkan penalty ke total penalti keseluruhan
Perbarui waktu saat ini menjadi arrival + waktu pelayanan
```

Gambar 4. 22 *Pseudo Code* Pencarian *Best Customer*

5) Kembali ke Depot

Tahapan ini menerapkan logika ketika tidak ada *customer feasible*, kendaraan kembali ke depot.

```
      Jika titik terakhir dalam rute bukan depot:
   Tambahkan depot ke akhir rute sebagai titik kembali
```

Gambar 4. 23 *Pseudo Code* Pengembalian Kendaraan ke Depot

6) *Return Output*

Tahap terakhir pada NI adalah proses penyimpanan dan pengembalian semua rute, *load*, jarak, waktu, dan *penalty*.

```
      Tambahkan hasil (rute, jarak, penalti, dll.) ke dalam
      daftar hasil
Kembalikan daftar hasil dan daftar pelanggan yang belum terlayani
```

Gambar 4. 24 *Pseudo Code* Menyisipkan Sisa *Customer*

Berdasarkan proses komputasi *Nearest Insertion (NI)* dilakukan, akan diperoleh hasil rute "awalan" yang akan digunakan kembali sebagai input untuk proses optimasi menggunakan *Genetic Algorithm*. Berikut merupakan *output* rute yang dihasilan *Nearest Insertion* (*NI)*:

Tabel 4. 14 *Output* Rute *Nearest Insertion (NI)*

Kendaraan	***Route***	***Distance (Km)***
Kendaraan 1	[0, 34, 51, 11, 27, 15, 64, 19, 12, 73, 66, 92, 77, 79, 89, 35, 23, 99, 29, 30, 74, 54, 96, 82, 55, 61, 58, 42, 68, 20, 38, 85, 57, 88, 0]	140.72 Km
Kendaraa 2	[0, 72, 75, 81, 52, 80, 84, 17, 87, 100, 45, 83, 21, 36, 93, 59, 40, 63, 44, 50, 65, 13, 37, 91, 3, 90, 78, 24, 70, 46, 39, 53, 9, 5, 0]	182.19 Km
Kendaraa 3	[0, 48, 31, 98, 4, 18, 14, 28, 94, 6, 86, 49, 32, 25, 33, 26, 97, 95,	184.63m

Kendaraan	*Route*	*Distance (Km)*
	60, 69, 76, 62, 1, 7, 41, 71, 47, 67, 56, 8, 16, 43, 2, 22, 10, 0]	

B. Impelementasi Program *Genetic Algorithm* (GA)

Pada tahap ini algoritma *Genetic Algorithm* (GA) digunakan untuk memperbaiki dan mengembangkan rute awal hasil dari metode *heuristic* oleh *Nearest Insertion* (NI). Peran AG adalah sebagai metode *metaheuristic improvement*, yang bertujuan untuk menyempurnakan urutan kunjungan pelanggan dalam masing-masing rute kendaraan agar diperoleh solusi dengan total jarak dan penalti yang minimum dalam skenario *Capacitated Vehicle Routing Problem with Time Windows* (CVRPTW). Implementasi *Genetic Algorithm* (GA) dilakukan melalui 2 bagian utama, yaitu fungsi `[ga_optimize_route]` untuk melakukan optimasi rute Tunggal kendaraan, dan fungsi `[improve_routes_with_ga]` untuk mengaplikasikan GA pada seluruh kendaraan hasil NI.

1) Representasi Kromosom

Setiap solusi atau individu dalam populasi direpresentasikan sebagai kromosom berupa urutan *customer* tanpa depot. Kromosom berisi daftar ID pelanggan [c1, c2, ..., cn], sedangkan depot ditambahkan secara eksplisit di awal dan akhir saat evaluasi *fitness*.

```
      Ambil elemen rute dari indeks ke-1 hingga sebelum terakhir
      sebagai daftar customer
(karena indeks pertama dan terakhir adalah depot, tidak termasuk
kromosom)
```

Gambar 4. 25 *Pseudo Code* GA Representasi Kromosom

Berikut ini merupakan contoh hasil kromosom yang terbentuk dari proses respresentasi kromosom untuk 100 individu yang berperan sebagai *customer*:

Tabel 4. 15 Contoh Hasil Kromosom Terbentuk GA

Individu	Kromosom
1	[67, 62, 32, 49, 41, 95, 69, 71, 26, 76, 8, ..., 94, 18, 16, 7, 10, 25, 56, 43, 97, 22, 14, 86]

Individu	Kromosom
2	[71, 69, 22, 14, 6, 8, 2, 18, 62, 28, 4, 1, ..., 41, 16, 98, 31, 48, 43, 7, 94, 95, 97, 60, 33]
3	[33, 18, 62, 25, 28, 49, 4, 22, 95, 76, …, 43, 7, 31, 10, 86, 69, 60, 48, 32, 94, 26, 98, 71]
…	…
100	[43, 32, 49, 2, 33, 62, 4, 47, 10, 8, 18, ..., 97, 6, 98, 25, 22, 71, 76, 7, 86, 41, 56, 31, 60]

2) Inisialisasi Populasi

Setiap individu dalam populasi merepresentasikan urutan pelanggan dalam satu rute kendaraan. Kromosom disusun sebagai urutan integer (ID *customer*) dari hasil rute awal oleh *Nearest Insertion* (NI). Depot sebagai titik awal dan akhir tidak disertakan dalam kromosom, tetapi ditambahkan saat evaluasi *fitness*.

```
    Untuk i dari 0 hingga (jumlah_populasi - 1), lakukan:
        Buat individu baru dengan memanggil fungsi
    buat_individu()
        Catat log populasi generasi ke-0 dengan informasi:
            - Nomor individu (i + 1)
            - Kromosom (urutan customer dari individu)
            - Nilai fitness individu
        Tambahkan individu ke dalam populasi

    Fungsi buat_individu():
        Salin daftar customer
        Acak urutan customer tersebut secara acak (shuffle)
Kembalikan urutan yang sudah diacak sebagai individu
```

Gambar 4. 26 *Pseudo Code* GA Inisialisasi Populasi

Dari proses tersebut akan terbentuk individu-individu berdasarkan penentuan jumlah populasi. Tahapan inisialisasi populasi akan membuat sejumlah solusi awal (disebut individu atau kromosom) dan individu adalah urutan acak *customer* dalam rute awal. Ukuran populasi awal (`pop_size`) ditetapkan, misalkan 100. Berikut ini merupakan ukuran populasi yang digunakan pada penelitian ini:

Tabel 4. 16 Ukuran Populasi GA

No Iterasi	Populasi	Generasi
1	20	50
2	30	70
3	50	100
4	70	150
5	100	200
6	150	250
7	200	300
8	250	350

3) Fungsi *Fitness*

Fungsi *fitness* (kebugaran) berfungsi untuk mengukur kualitas individu berdasarkan *objective function* nya atau total biaya rute (biaya perjalanan + penalti). Fungsi terbentuk dari proses evaluasi *fitness* melalui jumlah generasi yang terbentuk. Beberapa ukuran fitness digunakan untuk mengevaluasi kromosom selama tiap generasi (Syahputra, 2024). Seperti yang dapat dilihat pada tabel 4.17, *fitness* terbentuk sesuai dengan jumlah generasi yang ditentukan. Nilai *Fitness* membedakan kualitas antar kromosom, semakin besar nilai *fitness* maka semakin baik kromosom untuk dijadikan calon solusi.

```
       Fungsi hitung_fitness(individu):
           Buat rute dengan menyisipkan depot di awal dan akhir
       individu:
               rute = [depot] + urutan_customer_dari_individu +
       [depot]
    Hitung dan kembalikan biaya dari rute tersebut menggunakan
fungsi hitung_biaya_rute(rute)
```

Gambar 4. 27 *Pseudo Code* GA Inisialisasi Populasi

Tabel 4. 17 Contoh Hasil *Fitness* Terbentuk GA

Generasi	*Fitness* Terbaik
1	431516,5445
2	404397,3184

Generasi	*Fitness* Terbaik
3	403547,5302
4	375315,8154
5	337511,9208
6	335455,8735
7	290022,7579
…	…
200	19271,02211

Individu atau kromosom yang berupa kendaraan dengan rute yang berbeda-beda akan terpilih menjadi solusi terbaik (kendaraan dengan jarak terpendek) ketika individu tersebut memiliki nilai *fitness* tertinggi dibandingkan individu yang lain.

4) Proses Seleksi

Pada tahapan ini, Algoritma akan memilih individu terbaik dari populasi saat ini untuk dijadikan *parent* (induk) dalam proses *crossover*. Proses ini menggabungkan pendekatan *elitisme* dan *truncation selection*. *Elitisme* dilakukan dengan mempertahankan 20% individu terbaik dari populasi saat ini ke generasi berikutnya dengan tujuan untuk memastikan bahwa solusi terbaik tidak hilang selama proses evolusi. Kemudian pasangan individu yang akan digunakan untuk proses *crossover* dipilih secara acak dari 50% individu terbaik.

```
a) Urutkan populasi berdasarkan nilai fitness dari yang terbaik
   ke terburuk

b) Ambil 20% individu terbaik sebagai populasi baru (elitisme):
       populasi_baru = 20% individu terbaik dari populasi lama

c) Selama ukuran populasi baru masih kurang dari ukuran populasi
   total:
   Pilih secara acak dua parent dari 50% individu terbaik dalam
populasi lama
```

Gambar 4. 28 *Pseudo Code* GA Proses Seleksi

5) Proses *Crossover*

Proses *crossover* pada penelitian ini menggunakan metode *Ordered Crossover (OX)-like*, yaitu dengan memilih secara acak dua titik potong dari kromosom induk pertama untuk disalin langsung ke kromosom *child*. Kemudian elemen-elemen dari induk kedua tersebut dimasukkan ke posisi kosong pada *child*. Proses ini menjaga keberagaman genetik dan tetap mempertahankan urutan urutan sebagian gen yang mungkin sudah optimal.

```
a) Fungsi crossover(parent1, parent2):
    Tentukan panjang kromosom sebagai size
    Pilih dua indeks acak a dan b, lalu urutkan agar a < b
    Buat child (anak) berisi elemen None sepanjang size
    Salin potongan parent1 dari indeks a sampai b ke posisi
yang sama di child
    Inisialisasi indeks parent2 ke 0

b) Untuk setiap posisi i dalam child:
        Jika child[i] masih kosong (None):
            Cari elemen berikutnya dari parent2 yang belum
ada di child
            Isi child[i] dengan elemen tersebut

Kembalikan child sebagai hasil crossover
```

Gambar 4. 29 *Pseudo Code* GA Proses *Crossover*

Berikut merupakan contoh hasil kromosom berupa *child* yang terbentuk dari proses *ordered crossover*:

Tabel 4. 18 Contoh Hasil Proses *Ordered Crosover (OX)* 200 Gen

Generasi	***Parent1***	***Parent2***	***Child***	***Fitness Child***
1	[32, 8, 26, 67, ..., 60, 94, 33, 7]	[86, 6, 69, ..., 2, 94, 97]	[6, 69, 8, 4, ..., 62, 26, 94]	502001,5134
2	[31, 69, 97, ..., 56, 8, 76, 41]	[2, 98, 16, ..., 25, 47, 97, 1]	[98, 16, 14, ..., 47, 97, 1]	379139,724

Generasi	*Parent1*	*Parent2*	*Child*	*Fitness Child*
3	[8, 32, 16, ..., 56, 31, 2, 25]	[18, 31, 69, ..., 14, 62, 41]	[69, 67, 16, ..., 31, 14, 41]	420903,3131
4	[69, 16, 49, ..., 76, 8, 26]	[31, 95, 43, ..., 67, 8, 71, 6]	[95, 28, 62, 2, ..., 32, 10, 67, 8]	491245,2245
5	[71, 49, 43, ..., 60, 94, 98, 10]	[18, 1, 43, 32, ..., 94, 60, 95, 22]	[18, 1, 43, 7, 6, ..., 60, 8, 10, 94]	315511,2245
…	…	…	…	…
200	[32, 49, 18, 43, ..., 41, 94, 28, 14]	[32, 49, 18, ..., 41, 94, 28, 14]	[32, 49, 18, 43, ..., 94, 28, 14]	5066,383642

Hasil *child* yang terbentuk merupakan hasil proses pertukaran dengan fungsi *crossover* yang merupakan representasi dari urutan rute yang terbentuk untuk setiap kendaraan.

6) Proses Mutasi (*Mutation*)

Setelah melakukan proses pertukaan atau *crossover* selanjutnya adalah proses mutasi yang dilakukan dengan menggunakan pendekatan *Swap Mutation*, yaitu dengan menukar posisi dua gen (konsumen) dalam kromosom secara acak. Mutasi dilakukan dengan probabilitas tertentu *(mutation_rate)* untuk menghindari stagnasi solusi pada *local optimum. Swap Mutation* bekerja dengan menukar posisi dua gen secara acak dalam satu kromosom (rute *customer*).

```
Fungsi mutate(individual):
    Jika bilangan acak antara 0 dan 1 lebih kecil dari
mutation_rate:
        Pilih dua indeks acak i dan j dari kromosom
  Tukar posisi elemen pada indeks i dan j dalam individual
```

Gambar 4. 30 *Source Code* GA Proses *Mutation*

Berikut merupakan Langkah-langkah dalam proses mutase berdasarkan *Source Code* tersebut:

a. Probabilitas Mutasi, kode *(if random.random() < mutation_rate)* artinya, mutasi dilakukan hanya jika nilai acak (0–1) lebih kecil dari *(mutation_rate)* (misalnya 0.1 → 10% kemungkinan).
b. Pemilihan 2 posisi secara acak, kode (`i, j = random.sample(range(len(individual)), 2)`) artinya, dua indeks berbeda dalam kromosom akan dipilih secara acak. Ini menentukan gen (*customer*) mana yang akan ditukar.
c. Pertukaran posisi gen (*swap*), kode (`individual[i], individual[j] = individual[j], individual[i]`) artinya, proses penukaran posisi kedua gen tersebut dalam kromosom.

Kromosom 4	[95, 28, 62, 2, ..., 32, 10, 67, 8]
Hasil Mutasi *Swap*	[62, 28, 95**,** 8, ..., 67, 10, 32, 2]

Gambar 4. 31 Contoh Proses Mutasi *Swap*

7) Pembentukan Generasi Baru

Proses pembentukan generasi baru merupakan tahapan menggabungkan individu terbaik (elit) yang terbentuk dan anak (*child)* hasil *crossover* &mutasi menjadi populasi baru untuk generasi berikutnya sebagai bentuk dari iterasi. Tujuannya adalah untuk melanjutkan proses evolusi dengan kombinasi individu terbaik (*elitisme*) dan solusi baru hasil eksplorasi.

```
a) Urutkan populasi berdasarkan nilai fitness dari terbaik
   ke terburuk
Ambil 20% individu terbaik sebagai elitisme → simpan dalam
new_pop
b) Selama jumlah individu di new_pop masih kurang dari
   ukuran populasi:
    Pilih 2 parent secara acak dari 50% populasi terbaik
    Lakukan crossover antara kedua parent untuk
menghasilkan child
```

```
            Lakukan mutasi pada child (jika terpilih berdasarkan
        mutation_rate)
            Tambahkan child ke dalam new_pop

Ganti populasi lama dengan new_pop sebagai generasi baru
```

Gambar 4. 32 *Pseudo Code* GA Proses Pembentukan Generasi Baru

Proses pembentukan generasi baru dalam *Genetic Algorithm* (GA) ini dilakukan melalui dua pendekatan utama, yaitu *elitisme* dan reproduksi solusi baru.

a. Pertama, sebanyak 20% (*elitism 0.2)* dari individu terbaik berdasarkan nilai *fitness* diseleksi langsung dan diturunkan ke generasi berikutnya tanpa mengalami modifikasi. Pendekatan ini bertujuan untuk menjaga kualitas solusi terbaik agar tidak hilang selama proses evolusi.
b. Sebanyak 80% populasi sisanya dibentuk melalui proses reproduksi, yaitu dengan memilih pasangan *parents* secara acak dari 50% populasi terbaik (`pop_size * 0.5`), kemudian dilakukan proses *crossover* dan mutasi untuk menghasilkan individu baru (*child*). Individu-individu hasil reproduksi ini kemudian mengisi sisa populasi hingga ukuran populasi penuh tercapai.

4.3.4 Verifikasi dan Validasi Model

A. Verifikasi Model

Verifikasi pada penelitian ini bertujuan untuk memastikan bahwa algoritma yang berhasil dikembangkan benar-benar mengimplementasikan logika yang sesuai dengan formulasi masalah yang dibangun. Secara garis besar, proses verifikasi sudah dilakukan secara internal pada program algoritma yang dilakukan pada kode-kode berikut:

Tabel 4. 19 Verifikasi Model

***Pseudo Code* Verifikasi**	**Tujuan dan Fungsi**	**Terverifikasi?**
`Jika permintaan node lebih besar dari sisa kapasitas kendaraan:` `Lewati node ini (continue)`	Bertujuan untuk memastikan bahwa kendaraan tidak mengambil *customer* lebih dari kapasitas maksimum (*vehicle_capacity*).	✓

Pseudo Code Verifikasi	Tujuan dan Fungsi	Terverifikasi?
Jika waktu kedatangan lebih besar dari batas akhir time window: Hitung penalti sebagai (arrival_time - tw_end) × penalty_per_minute Tandai bahwa node ini terlambat (late = True)	Bertujuan untuk memverifikasi apakah waktu kedatangan melewati batas akhir *time window*.	✓
Mulai rute dari depot: route = [depot]	Bertujuan untuk menjamin bahwa setiap rute selalu dimulai dan berakhir di depot (ID 0).	✓
Tambahkan depot di akhir rute (sebagai kembali ke depot) Hapus node yang sudah dikunjungi dari daftar unvisited	Bertujuan untuk memastikan *Node customer* hanya dikunjungi sekali.	✓
Jika rute berisi lebih dari 1 node (artinya ada pelanggan)	Bertujuan untuk memastikan kendaraan hanya disimpan jika minimal ada 1 *customer* yang dikunjungi (selain depot).	✓
Validasi akhir: Ambil semua ID customer dari data lokasi, kecuali depot Bandingkan dengan customer yang telah dikunjungi Jika ada customer yang tidak dikunjungi atau kelebihan kunjungan:	Bertujuan untuk memverifikasi seluruh *customer* atau *node* sudah dilewati agar dilayani hanya 1 kali (tanpa duplikat dan terlewat)	✓

Pseudo Code Verifikasi	Tujuan dan Fungsi	Terverifikasi?
Laporkan error berisi: - daftar customer yang terlewat (missed) - daftar customer yang dikunjungi lebih dari sekali (extra)		

Jika semua fungsi verfikasi sudah terdaftar dalam model algoritma dan algoritma sudah memaksa hasil yang dihasilkan harus sesuai dengan program yang dibangun, maka secara otomatis model sudah terverifikasi secara internal.

B. Validasi model

Validasi model dilakukan untuk memastikan bahwa model yang dibangun baik itu konseptual atau matematis mampu menyelesaikan masalah dunia nyata. Proses validasi dilakukan dengan cara melakukan *benchmarking* atau perbandingan hasil *Genetic Algorithm* (GA) dengan algoritma lain, pada penelitian ini pembandingnya adalah algoritma *Nearest Neighbor.* Tujuan dari *benchmarking* ini adalah untuk memastikan bahwa model yang dikembangkan tidak hanya memberikan solusi yang layak, tetapi juga unggul dalam hal efisiensi operasional. Berikut merupakan *benchmarking* hasil rute sebagai proses validasi:

Tabel 4. 20 Validasi Model

Metode	Jumlah Nodes	Jumlah Kendaraan	Total Demand	Total Jarak	Rata-rata Jarak per Kendaraan	Total Waktu Tempuh	Rata-rata Waktu Tempuh per Kendaraan
NN	101	3	1596 kg	427,37 Km	142,45 km	2654 Menit (44,23 Jam)	884,66 Jam (14,7 Jam)
NI+GA	101	3	1596 kg	366,99 Km	122,33 km	2255,92 Menit	751,9 Jam (12,5 Jam)

Metode	Jumlah Nodes	Jumlah Kendaraan	Total Demand	Total Jarak	Rata-rata Jarak per Kendaraan	Total Waktu Tempuh	Rata-rata Waktu Tempuh per Kendaraan
						(37,5 jam)	

Berdasarkan tabel 4.45 *Benchmarking* ini dianggap membuktikan bahwa model valid karena,

- Jumlah node, demand, dan kendaraan yang digunakan dalam kedua metode adalah identik, yaitu 101 node (100 *customer* + 1 depot), total demand sebesar 1596 kg, dan 3 kendaraan.
- Perbandingan dilakukan secara *head-to-head*, dengan skenario dan data input yang sama persis, sehingga perbedaan hasil mencerminkan performa algoritma.
- Metrik evaluasi yang digunakan mencakup total jarak tempuh dan total waktu tempuh, yang merupakan dua aspek utama dalam optimasi rute pada CVRPTW. Waktu tempuh pada metode pembanding melebihi waktu tersedia yaitu 14,7 jam.

4.3.5 Eksperimen Iterasi dan Implementasi Algoritma

Dalam proses optimasi rute pada permasalahan *Capacitated Vehicle Routing Problem with Time Windows* (CVRPTW), pemilihan parameter pada *Genetic Algorithm* (GA) berperan penting dalam menentukan kualitas solusi akhir. Maka dari itu, dilakukan eksperimen iteratif terhadap tiga parameter utama, yaitu *mutation rate*, jumlah generasi, dan ukuran populasi, untuk menemukan kombinasi yang paling optimal. Eksperimen ini bertujuan untuk mengevaluasi pengaruh masing-masing parameter terhadap performa algoritma dalam menghasilkan solusi terbaik, dengan kriteria evaluasi berupa:

a. Total jarak tempuh (semakin pendek semakin baik),
b. Total waktu tempuh, dan
c. Total penalti keterlambatan (sebisa mungkin nol).

Eksperimen dilakukan dengan mengatur satu parameter sebagai variabel bebas, sementara dua parameter lainnya dijaga tetap (konstan). Proses komputasi dilakukan pada data aktual yang seragam, kemudian hasil parameter yang menghasilkan Solusi terbaik

akan digunakan untuk proses komputasi hasil akhir algoritma VRP. Hasil parameter yang dilihat sebagai output yang baik dilihat dari total jarak, waktu tempuh dan nilai *fitness*. Pendekatan ini dilakukan secara bergantian untuk setiap parameter dengan 1 kali percobaan (1x repetisi) dan penggunaan fungsi *random seed* untuk mencegah keberagaman dan hasil tidak konsisten pada output.

a. Eksperimen terhadap jumlah populasi (P)

Nilai jumlah populasi yang digunakan divariasikan pada angka mulai dari 50, 100, 150, hingga maksimal 300, dengan ukuran *mutation rate* dan jumlah generasi tetap, contoh *mutation rate* 0,2 dan generasi sejumlah 300.

Gambar 4. 33 Hasil Eksperimen Populasi

Jumlah P	**Total Jarak**	**Waktu Tempuh**	**Nilai *Fitness***	**Penalty**
50	427,66	2633,18	512,66	Rp34,000
100	415,64	2656,15	505,64	Rp36,000
150	400,76	2592,5	450,76	Rp20,000
200	374,37	2564,48	429,37	Rp22,000
250	400,4	2619,53	465,4	Rp26,000
300	379,24	2535,5	414,24	Rp14,000

Hasil eksperimen menunjukkan bahwa terdapat kecenderungan menurun pada total jarak tempuh seiring dengan meningkatnya jumlah populasi. Total jarak mengalami penurunan dari 427,66 km pada populasi 50 menjadi 374,37 km pada populasi 200, sebelum mengalami sedikit kenaikan pada populasi 250 dan kembali menurun pada populasi 300. Hal ini menunjukkan bahwa populasi yang lebih besar cenderung mampu mengeksplorasi ruang solusi dengan lebih baik, sehingga menghasilkan rute yang lebih efisien. Penurunan yang konsisten pada total waktu tempuh juga mendukung proses ekperimen, di mana waktu tempuh menurun dari 2.633,18 menit pada populasi 50 menjadi 2.535,50 menit pada populasi 300.

Sementara itu, penalti keterlambatan juga menunjukkan nilai yang relatif menurun dengan bertambahnya jumlah populasi. Penalti tertinggi terjadi pada populasi 100 sebesar Rp36.000, sedangkan penalti terendah ada pada populasi 300 sebesar Rp14.000. Penalti dihitung berdasarkan keterlambatan terhadap batas akhir *time*

window, maka hal ini menunjukkan bahwa peningkatan populasi turut berkontribusi dalam Batasan *time windows*. Maka untuk eksperimen parameter selanjutnya, nilai populasi yang digunakan adalah 300 individu.

b. Eksperimen terhadap *mutation rate* (μ)

Nilai *mutation rate* yang digunakan divariasikan pada angka 0.01, 0.05, 0.1, 0.2, dan 0,3, sementara jumlah generasi dan ukuran populasi ditetapkan konstan dan berasal dari ekperimen sebelumnya (misalnya 300 generasi dan populasi 300 individu). Berikut merupakan hasil yang diperoleh pada eksperimen terhadap jumlah *mutation rate*:

Gambar 4. 34 Hasil Eksperimen *Mutation Rate*

Jumlah μ	Total Jarak	Waktu Tempuh	Nilai *Fitness*	Penalty
0,3	397,35	2514,03	437,35	Rp. 18,000
0,2	394,45	2553,08	424,45	Rp. 12,000
0,1	379,24	2535,5	414,24	Rp. 14,000
0,05	375,23	2541,17	420,23	Rp. 18,000
0,01	430,04	2611,69	480,04	Rp. 20,000

Berdasarkan hasil yang disajikan pada Gambar 4.14, terlihat bahwa *mutation rate* sebesar 0,1 menghasilkan total jarak dan waktu tempuh yang paling rendah. Meskipun demikian, penalti yang dihasilkan pada skenario ini adalah sebesar Rp14.000, yang lebih tinggi dibandingkan dengan skenario *mutation rate* 0,2.

Pada *mutation rate* sebesar 0,2, total jarak tempuh adalah 394,45 km dengan waktu tempuh 2.553,08 menit. Meskipun nilai jaraknya lebih tinggi dibandingkan dengan skenario μ = 0,1, penalti yang tercatat hanya sebesar Rp12.000. Nilai *fitness* pada skenario μ = 0,2 juga merupakan yang terendah dibandingkan seluruh percobaan, yaitu sebesar 424,45, menandakan bahwa solusi yang dihasilkan lebih seimbang antara efisiensi jarak dan keterpenuhan Batasan *time window*. Oleh karena itu nilai *mutation rate* yang akan digunakan untuk eksperimen parameter selanjutnya adalah 0,2.

c. Eksperimen terhadap jumlah generasi (G)

Nilai jumlah generasi yang digunakan divariasikan pada angka mulai dari 100, 300, 500, hingga maksimal 700, sementara ukuran *mutation rate* dan ukuran populasi yang

digunakan berasal dari eksperimen sebelumnya yaitu *mutation rate* 0,2 dan jumlah populasi sebanyak 300.

Gambar 4. 35 Hasil Eksperimen Generasi

Jumlah G	Total Jarak	Waktu Tempuh	Nilai *Fitness*	Penalty
150	378.34	2262,78	436,58	Rp. 18,000
300	398.58	2267.18	402,17	Rp. 10,000
500	366,99	2255,92	94,90	Rp. 0
700	376,12	2256,28	95,90	Rp. 8,000

Eksperimen ini menunjukkan bahwa peningkatan jumlah generasi berdampak positif terhadap kualitas solusi dalam penyelesaian model VRP. Pada generasi 150 hingga 500, terjadi penurunan bertahap terhadap total jarak, waktu tempuh, dan nilai *fitness*.

Performa terbaik dicapai pada generasi ke-500, dengan total jarak 366,99 km, waktu tempuh 2255,92 menit, tanpa penalti keterlambatan, serta nilai *fitness* terendah yaitu 94,90. Hasil ini mengindikasikan bahwa generasi yang lebih panjang memberi peluang eksplorasi solusi yang lebih optimal, selama tetap mempertimbangkan efisiensi waktu komputasi.

4.3.6 Hasil Rute Optimal

Berdasarkan hasil akhir dari model algoritma NI dan GA yang diimplementasikan, diperoleh konfigurasi rute optimal dengan jumlah kendaraan sebanyak tiga kendaraan (*blind van*). Rute ini diperoleh pada eksperimen dengan jumlah generasi 500, populasi 300 dan *mutation rate* 0,2.

Masing-masing kendaraan menghasilkan rute dengan total jarak tempuh dan waktu layanan yang efisien dari segi jarak tempuh dan memenuhi Batasan waktu layanan (*time windows*) yang mana dibuktikan dengan jumlah biaya keterlambatan sebesar Rp. 0. Masing-masing kendaraan melayani *customer* nya masing-masing dengan jumlah titi kantar yang terbagi rata pada ketiga kendaraan. Kapasitas kendaran juga telah dipertimbangkan dengan maksimal kapasitas 1000 kg per kendaraan.

Tabel 4. 21 Hasil Rute Optimal

Kendaraan	Rute Terbentuk	Total Jarak	Total Waktu Tempuh	Total Penalty
1	0, 57, 38, 68, 20, 42, 58, 61, 82, 96, 54, 30, 29, 99, 35, 79, 73, 12, 19, 15, 27, 34, 11, 92, 64, 66, 89, 23, 74, 55, 85, 88, 77, 51, 0.	94.90 km	705.72 Menit	Rp.0
2	0, 72, 80, 40, 63, 37, 91, 78, 90, 39, 5, 9, 50, 65, 13, 59, 36, 21, 17, 84, 81, 52, 100, 87, 45, 3, 24, 70, 46, 53, 44, 83, 93, 75, 0.	131.58 km	757.33 Menit	Rp.0
3	0, 32, 49, 18, 43, 2, 16, 8, 67, 56, 71, 7, 1, 62, 31, 98, 4, 76, 69, 95, 25, 33, 26, 14, 28, 94, 6, 86, 60, 97, 48, 41, 47, 10, 22, 0.	140.52 km	793.14 Menit	Rp.0

BAB V
HASIL DAN PEMBAHASAN

5.1 Analisis Rute Hasil Optimasi

Setelah melalui proses pemodelan, pengajian dan eksperimen terhadap permasalahan *Capacitated Vehicle Routing Problem with Time Windows* (CVRP-TW), diperoleh solusi terbaik atau hasil yang paling optimal berupa tiga rute distribusi optimal yang dilayani oleh tiga kendaraan. Setiap kendaraan dirancang untuk mengantarkan barang dari depot ke sejumlah pelanggan dengan memperhatikan batasan kapasitas maksimum kendaraan yaitu 1000 kg dan *time window* yang telah ditentukan sebelumnya.

A. Kendaraan 1

Hasil optimasi gabungan *Genetic Algorithm* dan *Nearest Insertion* mampu menghasilkan pembentukan rute kendaraan dengan 0 keterlambatan dan total berat angkut barang didalam *blindvan* sebanyak 596 Kg. Dengan waktu kedatangan yang berada didalam jendela layanan, kendaraan 1 berhasil mengirimkan *demand* kepelanggan tepat waktu dan kembali ke titik depot sebelum jam tutup.

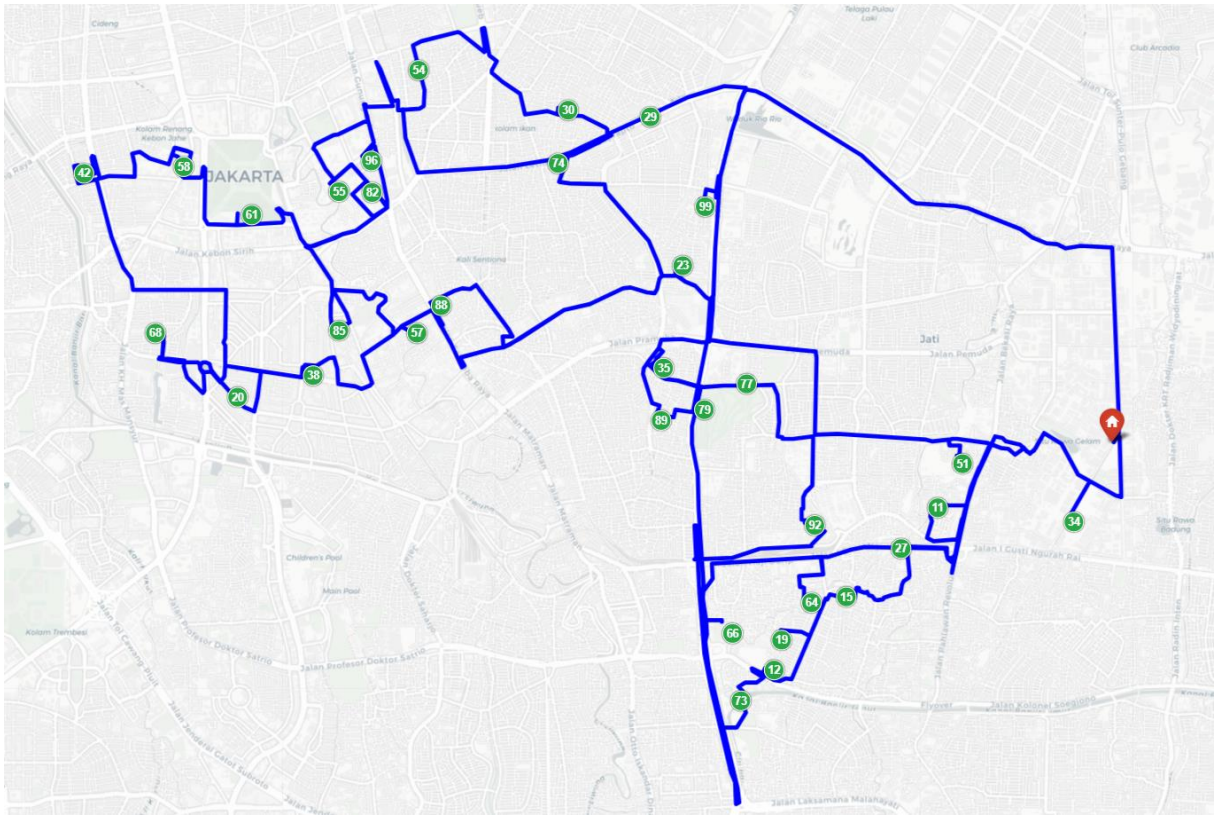
Jarak antar titik relatif bervariasi, namun tetap dalam rentang wajar sehingga mendukung efisiensi perjalanan. Beberapa titik menunjukkan perpindahan jarak yang cukup signifikan, namun berhasil diantisipasi dengan jeda waktu tempuh yang sesuai. Hal ini mengindikasikan bahwa optimasi rute mempertimbangkan *trade-off* antara jarak dan waktu secara seimbang. Kemudian total waktu yang ditempuh pada tiap titik merupakan waktu yang dibutuhkan dalam perjalanan dari titik awal ke titik selanjutnya, dengan penambahan waktu layanan untuk bongkar muat (*unloading*) selama 5 menit. Berikut merupakan rincian rute yang dilalui kendaraan 1:

Tabel 5. 1 Detail Rute Kendaraan 1

Customer	**Waktu Kedatangan (Mnt)**	***Time Window Start***	***Time Window End***	**Keterlambatan**	**Penalti**	**Jarak antar titik (km)**	**Waktu Tempuh (Mnt)**
0	**480**	**480**	**1260**	**-**	**-**	-	-
57	515,37	480	780	*No*	0	12,97	35,37
38	525,96	480	780	*No*	0	2,05	5,59
68	539,08	480	780	*No*	0	2,98	8,12
20	781	781	1020	*No*	0	1,9	5,18
42	799,51	781	1020	*No*	0	4,96	13,51
58	809,48	781	1020	*No*	0	1,82	4,96

Customer	Waktu Kedatangan (Mnt)	*Time Window Start*	*Time Window End*	Keterlambatan	Penalti	Jarak antar titik (km)	Waktu Tempuh (Mnt)
61	818,64	781	1020	*No*	0	1,53	4,17
82	829,71	781	1020	*No*	0	2,22	6,06
96	836,31	781	1020	*No*	0	0,59	1,6
54	846,39	781	1020	*No*	0	1,86	5,08
30	859,11	781	1020	*No*	0	2,83	7,73
29	868,22	781	1020	*No*	0	1,51	4,11
99	878,48	781	1020	*No*	0	1,93	5,26
35	891,77	781	1020	*No*	0	3,04	8,29
79	899,66	781	1020	*No*	0	1,06	2,9
73	919,28	781	1020	*No*	0	5,36	14,62
12	926,54	781	1020	*No*	0	0,83	2,25
19	933,09	781	1020	*No*	0	0,57	1,55
15	942	781	1020	*No*	0	1,43	3,91
27	950,63	781	1020	*No*	0	1,33	3,63
34	964,31	781	1020	*No*	0	3,18	8,68
11	1021	1021	1260	*No*	0	2,49	6,8
64	1033,88	1021	1260	*No*	0	2,89	7,88
66	1043,09	1021	1260	*No*	0	1,54	4,21
92	1054,85	1021	1260	*No*	0	2,48	6,76
77	1067,64	1021	1260	*No*	0	2,86	7,79
89	1077,29	1021	1260	*No*	0	1,71	4,65
88	1094,65	1021	1260	*No*	0	4,53	12,36
85	1104,89	1021	1260	*No*	0	1,92	5,24
55	1116,79	1021	1260	*No*	0	2,53	6,9
74	1132,79	1021	1260	*No*	0	4,03	11
23	1145,85	1021	1260	*No*	0	2,95	8,06
51	1167,93	1021	1260	*No*	0	6,26	17,08
0	**1180,72**	**480**	**1260**	*No*	0	2,86	7,79

Visualiasi rute terbentuk daru urutan pengiriman barang ke *customer 57* hingga *customer* 51. Total jarak tempuh yang dilalui yaitu 94,9 km dan total waktu perjalanan yang ditempuh selama 705,45 menit atau 11 jam 75 menit. Peta rute telah menyesuaikan kondisi aktual di Jakarta Raya sehingga waktu dan jarak kedatangan lebih realistis.



Gambar 5. 1 Visualisasi Rute Optimal Kendaraan 1

B. Kendaraan 2

Pada kendaraan 2, algoritma juga berhasil membentuk rute pengiriman tanpa keterlambatan yang ditandai dengan waktu kedatangan ke setiap titik yang masih didalam jendela layanan (*time window).* Ketepatan waktu pengiriman menjadikan tidak adanya biaya penalti yang perlu dikeluarkan oleh Perusahaan.

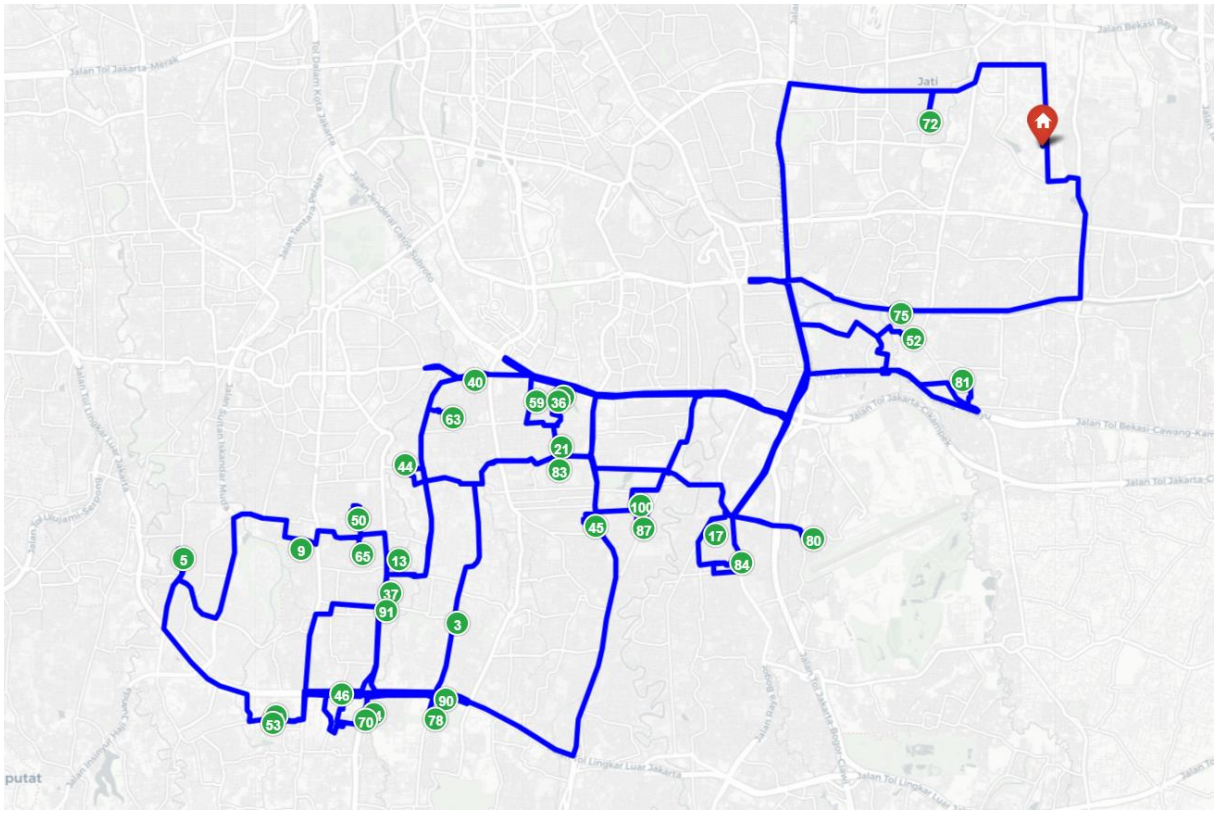
Kendaraan 2 mendistribusikan total barang sebanyak 497 Kg kepada 33 *customer* sehingga masih jauh dibawah batas maksimal muatan yaitu 1000 kg. Rute tiap titik dari segi jarak juga sudah optimal meskpun terdapat variasi. Dengan waktu kedatangan yang berada didalam jendela layanan, kendaraan 1 berhasil mengirimkan *demand* kepelanggan tepat waktu dan kembali ke titik depot sebelum jam tutup yaitu pukul 20:36 (Menit 1221,97).

Tabel 5. 2 Detail Rute Kendaraan 2

Customer	**Waktu Kedatangan (Mnt)**	***Time Window Start***	***Time Window End***	**Keterlambatan**	**Penalti**	**Jarak antar titik (km)**	**Waktu Tempuh (Mnt)**
0	480	480	1260	-	-	-	-

Customer	Waktu Kedatangan (Mnt)	*Time Window Start*	*Time Window End*	Keterlambatan	Penalti	Jarak antar titik (km)	Waktu Tempuh (Mnt)
72	489,67	480	780	No	0	3,55	9,67
80	529,07	480	780	No	0	12,61	34,4
40	563,73	480	780	No	0	10,88	29,66
63	572,23	480	780	No	0	1,28	3,5
78	601,19	480	780	No	0	8,79	23,96
91	615,61	480	780	No	0	3,46	9,42
37	622,09	480	780	No	0	0,54	1,48
13	781	781	1020	No	0	1	2,74
65	788,97	781	1020	No	0	1,09	2,97
50	796,83	781	1020	No	0	1,05	2,85
9	807	781	1020	No	0	1,9	5,17
5	821,34	781	1020	No	0	3,43	9,34
39	840,81	781	1020	No	0	5,31	14,47
90	859,41	781	1020	No	0	4,99	13,6
17	889,48	781	1020	No	0	9,19	25,07
84	897,55	781	1020	No	0	1,12	3,06
81	919,2	781	1020	No	0	6,11	16,65
21	952,95	781	1020	No	0	10,54	28,76
59	962,12	781	1020	No	0	1,53	4,17
36	968,88	781	1020	No	0	0,64	1,75
93	1021	1021	1260	No	0	0,2	0,55
44	1039,71	1021	1260	No	0	5,03	13,71
53	1067,79	1021	1260	No	0	8,46	23,08
46	1078,67	1021	1260	No	0	2,15	5,88
70	1086,48	1021	1260	No	0	1,03	2,82
24	1092,3	1021	1260	No	0	0,3	0,82
3	1107,09	1021	1260	No	0	3,59	9,79
83	1126,76	1021	1260	No	0	5,38	14,66
45	1137,13	1021	1260	No	0	1,97	5,37
87	1145,96	1021	1260	No	0	1,41	3,83
100	1152,8	1021	1260	No	0	0,67	1,84
52	1179,5	1021	1260	No	0	7,96	21,7
75	1199,43	1021	1260	No	0	5,48	14,94
0	**1221,97**	**480**	**1260**	No	0	6,43	17,54

Dari pola rute yang terbentuk pada gambar 5.2, terlihat bahwa algoritma telah mengelompokkan pelanggan ke dalam sub-wilayah distribusi yang relatif berdekatan secara geografis. Hal ini menunjukkan bahwa algoritma mampu melakukan *clustering* pelanggan dengan baik untuk meminimalkan jarak tempuh total. Apabila pada rute terlihat adanya rute yang kembali ke titik dibelakangnya atau diluar jalan nya, maka kendaraan sedang berusaha memenuhi titik dengan *time window* yang perlu dipenuhi dengan jarak yang lebih dekat.



Gambar 5. 2 Visualisasi Rute Optimal Kendaraan 2

C. Kendaraan 3

Kendaraan 3 memulai perjalanannya dari depot (0) pada pukul 480 menit (08:00 WIB) dan mengunjungi sebanyak 34 titik pelanggan tanpa mengalami keterlambatan atau penalti pada seluruh titik kunjungan. Hal ini menunjukkan bahwa rute yang dilalui kendaraan 3 telah sepenuhnya memenuhi batasan *time window* yang telah ditentukan dalam model.

Rangkaian kunjungan dimulai dari depot menuju pelanggan 32, dengan jarak antar titik sebesar 15,55 km yang memakan waktu tempuh 42,41 menit. Kendaraan 3 mendistribusikan total barang sebanyak 527 Kg kepada 34 *customer* sehingga masih jauh dibawah batas maksimal muatan yaitu 1000 kg. Selanjutnya, kendaraan terus melanjutkan

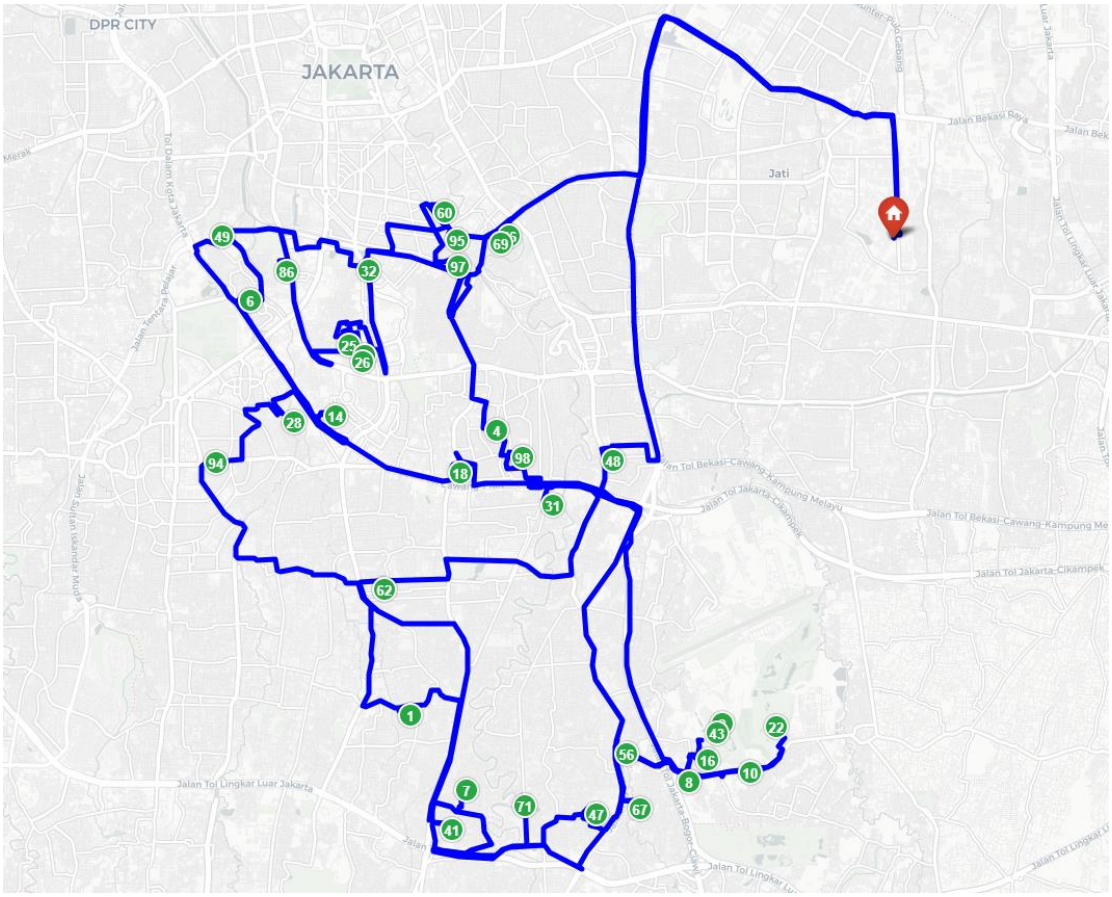
perjalanan ke titik-titik pelanggan lainnya, dengan waktu kedatangan yang selalu berada di dalam rentang waktu layanan (*time window*) yang ditentukan, yakni 480–780 menit untuk kelompok pelanggan pertama, 781–1020 menit untuk kelompok kedua, dan 1021–1260 menit untuk kelompok ketiga. Secara keseluruhan, rute kendaraan 3 dapat dikategorikan sebagai rute yang efisien secara waktu karena seluruh pelanggan berhasil dilayani tepat waktu tanpa penalti, serta cukup efisien dalam hal jarak

Tabel 5. 3 Detail Rute Kendaraan 3

Customer	**Waktu Kedatangan (Mnt)**	***Time Window Start***	***Time Window End***	**Keterlambatan**	**Penalti**	**Jarak antar titik (km)**	**Waktu Tempuh (Mnt)**
0	**480**	**480**	**1260**	**-**	-	-	-
32	522,41	480	780	No	0	15,55	42,41
49	539,36	480	780	No	0	4,38	11,95
18	571,06	480	780	No	0	9,79	26,71
43	605,16	480	780	No	0	10,67	29,1
2	781	781	1020	No	0	0,29	0,79
16	789,03	781	1020	No	0	1,11	3,03
8	796,35	781	1020	No	0	0,85	2,33
67	805,81	781	1020	No	0	1,63	4,45
56	815,28	781	1020	No	0	1,64	4,47
71	829,42	781	1020	No	0	3,35	9,14
7	839,18	781	1020	No	0	1,75	4,76
1	851,63	781	1020	No	0	2,73	7,45
62	866,85	781	1020	No	0	3,75	10,22
31	886,84	781	1020	No	0	5,5	14,99
98	896,39	781	1020	No	0	1,67	4,54
4	904,41	781	1020	No	0	1,11	3,02
95	924,84	781	1020	No	0	5,66	15,42
69	933,26	781	1020	No	0	1,26	3,42
76	939,12	781	1020	No	0	0,31	0,86
48	1021	1021	1260	No	0	7,21	19,66
22	1050,91	1021	1260	No	0	9,13	24,91
10	1060,15	1021	1260	No	0	1,56	4,25
47	1077,84	1021	1260	No	0	4,65	12,68
41	1094,43	1021	1260	No	0	4,25	11,59
94	1134,12	1021	1260	No	0	12,72	34,69
28	1146,11	1021	1260	No	0	2,56	6,98

Customer	**Waktu Kedatangan (Mnt)**	***Time Window Start***	***Time Window End***	**Keterlambatan**	**Penalti**	**Jarak antar titik (km)**	**Waktu Tempuh (Mnt)**
14	1154,44	1021	1260	No	0	1,22	3,33
26	1164,24	1021	1260	No	0	1,76	4,81
33	1169,81	1021	1260	No	0	0,21	0,56
25	1176,25	1021	1260	No	0	0,53	1,44
6	1189,86	1021	1260	No	0	3,16	8,61
86	1198,56	1021	1260	No	0	1,36	3,71
60	1217	1021	1260	No	0	4,93	13,44
97	1226,49	1021	1260	No	0	1,64	4,48
0	1259,49	480	1260	No	0	10,8	33

Berdasarkan visualisasi rute kendaraan 3, terlihat bahwa cakupan rute yang luas ini mengindikasikan bahwa kendaraan 3 melayani pelanggan dengan persebaran yang cukup tersebar namun relatif sistematis. Jalur yang dibentuk terlihat tidak banyak mengalami *looping* atau pergerakan bolak-balik yang tidak efisien. Hal ini menunjukkan bahwa rute telah dioptimalkan untuk meminimalkan waktu tempuh dan menghindari keterlambatan pelayanan.



Gambar 5. 3 Visualisasi Rute Optimal Kendaraan 3

Berdasarkan output detail rute kendaraan tersebut, berikut merupakan rangkuman total jarak tempuh, waktu tempuh dan keterlambatan yang dihasilkan dari hasil optimasi kombinasi *genetic algorithm* dan *nearest insertion* pada kasus CVRPTW ini.

Tabel 5. 4 Ringkasan Hasil Rute Optimal

Kendaraan	Jarak Tempuh (Km)	Waktu Tempuh (Menit)	Keterlambatan (Menit)	Biaya Penalti (Rp)	Kapasitas (Kg)
1	94,9	705.45	0	Rp.0	596
2	131.58	757.33	0	Rp.0	497
3	140.52	793.14	0	Rp.0	527
Grand Total	366.99	2255.92	0	Rp.0	1620

Selain menganalisis urutan kunjungan dan total jarak tempuh, proses optimasi juga mempertimbangkan estimasi waktu kedatangan (*Estimated Time of Arrival*) pada setiap titik layanan. Perhitungan ETA didasarkan pada kecepatan kendaraan sebesar 22 km/jam serta

waktu layanan di setiap titik selama 5 menit. Nilai ETA yang diperoleh kemudian dibandingkan dengan batas awal dan akhir dari jendela waktu layanan (*time window*) masing-masing pelanggan. Melalui pendekatan ini, dapat diidentifikasi titik-titik yang berpotensi mengalami keterlambatan. Analisis ini menunjukkan bahwa efisiensi rute tidak semata ditentukan oleh jarak tempuh, melainkan juga oleh pola urutan titik kunjungan serta alokasi waktu tempuh antar titik secara menyeluruh.

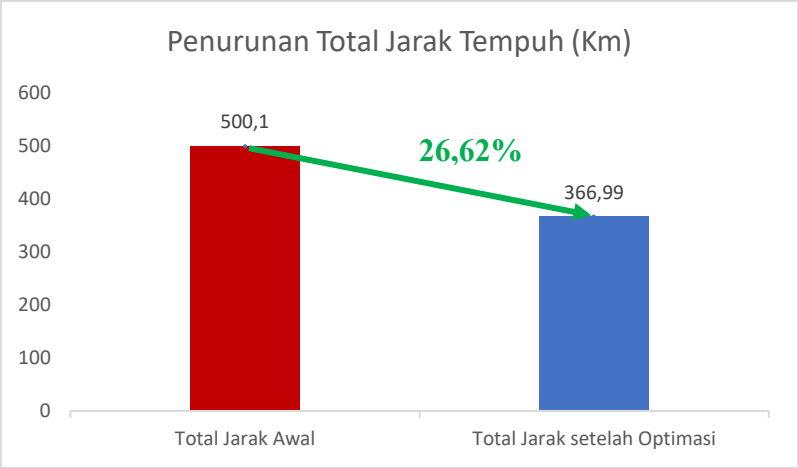
5.2 Analisis Penurunan Jarak dan Waktu Tempuh

Analisis dilakukan untuk membandingkan total jarak dan waktu tempuh antara rute distribusi awal (sebelum optimasi) dan rute hasil optimasi menggunakan algoritma *Nearest Insertion* dan *Genetic Algorithm*. Berikut merupakan tabel ringkasan penurunannya:

Tabel 5. 5 Ringkasan Penurunan Jarak dan Waktu Tempuh

Total Jarak Awal	**Total Jarak setelah Optimasi**	**Penurunan Total Jarak**	**Total Waktu tempuh Awal**	**Total Waktu Tempuh setelah Optimasi**	**Penurunan Total Waktu Tempuh**
131,31	94,9	36,41	743,15	705,45	37,7
154,19	131,58	22,61	799.51	757,33	42,18
214.60	140,52	74,08	902.32	793,14	109,18
∑ 500,1	**366,99**	**133,1**	**2444,98**	**2255,92**	**189,06**

Berdasarkan data pada Tabel 5.5, total jarak tempuh awal sebelum dilakukan optimasi adalah sebesar 500,1 km. Setelah proses optimasi dilakukan, total jarak tempuh berkurang menjadi 366,99km. Artinya, terjadi penurunan total jarak tempuh sebesar 133,1 km, yang secara persentase setara dengan penurunan sebesar 26,62% dari total jarak awal.



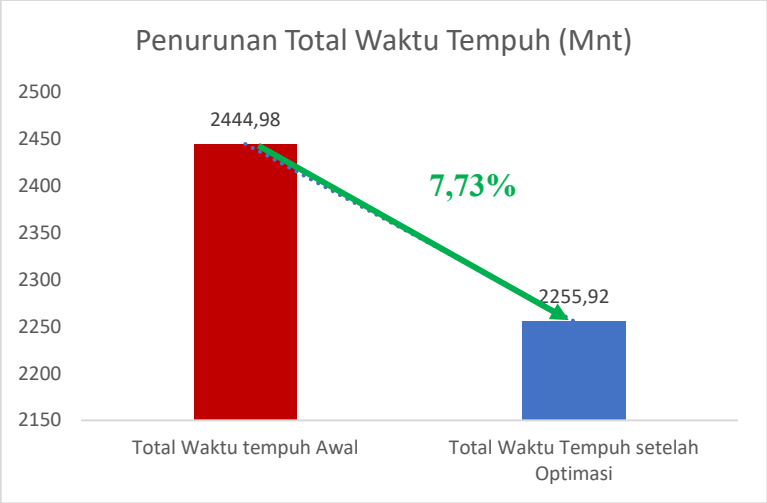
Gambar 5. 4 Grafik Persentase Penurunan Total Jarak Tempuh 3 Rute (Km)

Secara rinci, kendaraan pertama menunjukkan sedikit penurunan jarak tempuh dari 131,31 km menjadi 94,9 km (penurunan 36,41km). Kendaraan kedua mengalami penurnan jarak tempuh sebesar 22,61 km, dari 154,19 km menjadi 131,58 km. Kemudian kendaraan ketiga mengalami penurunan signifikan sebesar 74,08 km, dari 214,6 km menjadi 140,52 km. Penurunan total jarak secara terus menerus menunjukkan bahwa solusi optimasi secara keseluruhan mampu memperbaiki efisiensi rute, meskipun tidak semua kendaraan menunjukkan penurunan individual.

Selain jarak, optimasi juga berdampak pada efisiensi waktu tempuh kendaraan. Total waktu tempuh seluruh kendaraan sebelum optimasi adalah 2444,98 menit, sedangkan setelah optimasi turun menjadi 2255,92 menit. Penurunan total waktu tempuh ini mencapai 189,06 menit atau sebesar 7,7% dari waktu tempuh awal. Semua kendaraan mengalami penurunan waktu tempuh, dengan kendaraan ketiga mencatatkan penurunan paling signifikan yaitu 109,18 menit.

Efisiensi berupa penurunan Jarak Tempuh distribusi barang tersebut dicapai melalui penyusunan urutan kunjungan titik yang lebih optimal, sehingga kendaraan dapat menempuh rute dengan pola yang lebih terarah dan efisien. Selain itu, sistem secara otomatis menghindari jalur bolak-balik atau rute yang tidak searah, yang sebelumnya kerap terjadi dalam perencanaan manual. Pendistribusian beban antar kendaraan juga dilakukan secara lebih seimbang, yang memungkinkan seluruh armada bekerja secara proporsional tanpa ada kendaraan yang bekerja berlebihan atau justru terlalu sedikit. Dampak positif dari penurunan jarak ini sangat signifikan dalam konteks operasional logistik, antara lain efisiensi penggunaan bahan bakar kendaraan

secara langsung, penurunan biaya operasional harian seperti konsumsi solar dan kebutuhan perawatan, serta perpanjangan umur kendaraan akibat berkurangnya beban kerja yang tidak perlu. Maka dari itu, optimasi rute tidak hanya berdampak pada penghematan biaya tetapi juga meningkatkan keberlanjutan operasional logistik perusahaan.



Gambar 5. 5 Grafik Persentase Penurunan Total Waktu Tempuh 3 Rute (Menit)

Penurunan total jarak tempuh dan total waktu tempuh menunjukkan bahwa proses optimasi rute yang dilakukan tidak hanya efektif dalam mengurangi jarak perjalanan, tetapi juga berdampak pada efisiensi waktu operasional kendaraan. Efisiensi ini penting dalam konteks distribusi logistik karena dapat mengurangi konsumsi bahan bakar, meningkatkan ketepatan waktu pengiriman, serta menurunkan potensi penalti akibat keterlambatan.

Salah satu manfaat utama adalah tercapainya efisiensi logistik yang ditunjukkan melalui penurunan total jarak dan waktu tempuh distribusi, sehingga kendaraan dapat bekerja lebih efektif dalam cakupan wilayah yang sama. Selain itu, produktivitas kendaraan turut meningkat karena seluruh armada dimanfaatkan secara optimal tanpa adanya kendaraan yang *idle*, sementara beban kerja didistribusikan secara lebih seimbang antar kendaraan. Hal ini berdampak langsung terhadap peningkatan performa pengiriman secara keseluruhan. Dampak positif lainnya adalah peningkatan kepuasan pelanggan yang dicapai melalui ketepatan waktu pengiriman yang lebih tinggi, karena sistem rute yang dihasilkan mempertimbangkan secara cermat batas waktu pelayanan pada setiap titik. Maka dari itu, pendekatan CVRP-TW tidak hanya efektif secara teknis tetapi juga berdampak signifikan terhadap kualitas layanan dan efisiensi biaya dalam operasional logistik.

5.3 Analisis Keterlambatan dan Penalti

PT Pos Logistik Indonesia menetapkan penalty keterlambatan berupa biaya kompensasi atau pertanggung jawaban atas keterlambatan pengiriman kepada pelanggan yaitu sebesar 20% dari biaya ongkos kirim. Biaya penalti akan semakin besar tergantung kepada titik pengantaran, semakin jauh Lokasi antar maka akan semakin tinggi biaya yang harus dibayarkan.

Pada lingkup area pengiriman Jakarta Raya, PT Pos Logistik menetapkan biaya ongkos kirim flat yaitu Rp.10.000,00 setiap pengantaran. Jika biaya penalti yang diberlakukan adalah 20% dari biaya ongkos kirim, maka biaya yang perlu dibayarkan adalah sebesar Rp.2000,00 per keterlambatan. Meskipun biaya yang relatif rendah, apabila keterlambatan ini terus terjadi akan berdampak kepada kepuasan dan kepercayaan pelanggan. Selain itu efek jangka panjang yang akan terjadi apabila permasalahan keterlambatan terus terjadi selama berbulan-bulan. Terlebih jika PT Pos Logistik Indonesia memperluas area pengiriman dengan jarak dan biaya pengiriman yang tinggi.

A. Rute Eksisting Perusahaan

Pada Rute Eksisting Kebijakan Perusahaan terjadi keterlambatan sebanyak 16 kali. Kendaraan 1 mengalami 2 kali keterlambatan pengantaran, kendaraan 2 sebanyak 6 kali keterlambatan dan kendaraan 3 sebanyak 9 kali keterlambatan. Kendaraan 2 dan 3 juga Kembali ke depot sesudah jam layanan berakhir yaitu pukul 21.00 yang menyebabkan supir kembali terlambat.

Tabel 5. 6 Titik dengan Keterlambatan Pengiriman

Kendaraan	***Customer***	**Waktu Kedatangan (Menit)**	***Time Windows***	**Keterlambatan (Menit)**
1	57	935,4	480–780	155,4
	96	1039.7	781-1020	19,7
2	80	849.4	480-780	114,4
	63	897	480-780	117
	63	894.8	480-780	113,8
	50	1052.8	781-1020	32,8
	39	1081.0	781-1020	61
	9	1153.9	1021-1260	133,9
3	32	797.5	480-780	17,5

Kendaraan	*Customer*	Waktu Kedatangan (Menit)	*Time Windows*	Keterlambatan (Menit)
	18	1130.7	781-1020	40,2
	95	1150.0	781-1020	63
	69	1159.5	781-1020	71,4
	4	1149.5	781-1020	129,4
	98	1157.5	781-1020	137,5
	8	1201	781-1020	181
	47	1268.9	1021-1260	8,9
	94	1315.1	1021-1260	55,1

Dengan biaya penalti yang ditetapkan PT Pos Logistik Indonesia, maka biaya yang harus dikeluarkan dalam distribusi barang diarea Jakarta Raya yaitu Rp. 32.000,00.

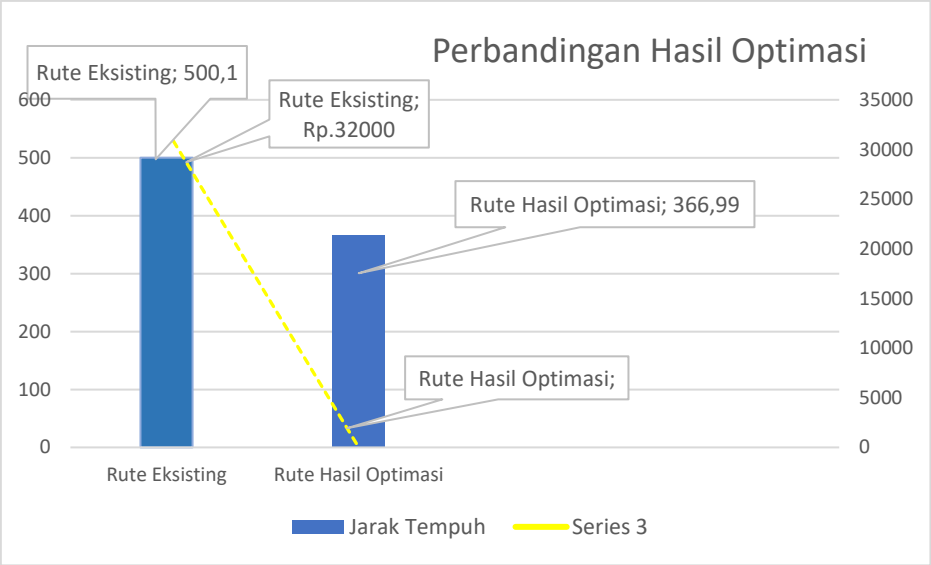
B. Rute Hasil Optimasi

Pada Rute Hasil Optimasi, algoritma berhasil membentuk rute tanpa keterlambatan, baik itu pengiriman ke titik-titik distribusi maupun kepulangan supir ke depot. Seperti pada tabel 5.1 hingga 5.3, waktu kedatangan kendaraan pada setiap titik berada didalam batas jendela layanan yang ditentukan *(time window).* Dengan pengurangan jarak yang tidak begitu signifikan, waktu tempuh dan keterlambatan berhasil dikurangi secara signifikan. Hal ini menandakan algoritma berhasil membuat rute distribusi yang tidak hanya lebih pendek namun mampu secara patuh memenuhi *constraint*.

Berikut ini merupakan perbandingan keseluruhan penurunan jumlah keterlambatan dan penaltt yang berhasil dilakukan:

Tabel 5. 7 Ringkasan Perbandingan

Keterangan	Sebelum Optimasi	Setelah Optimasi	Penurunan	Persentase
Titik Keterlambatan	16 *Customer*	0	16	100%
Biaya Penalti	Rp.32.000,00	Rp.0	Rp.32.000,00	100%



Gambar 5. 6 Grafik Perbandingan Hasil Rute

5.4 Analisis Berbasis *Estimated Time Arrival* (ETA) dan Potensi Pengembangan Model

Penelitian ini telah berhasil meminimalkan total jarak tempuh dan mengurangi jumlah keterlambatan pengiriman berdasarkan batasan waktu layanan (*time window*) yang telah ditetapkan. Parameter keterlambatan dihitung dari selisih antara waktu kedatangan kendaraan dengan batas atas *time window* pada masing-masing pelanggan. Namun, perlu dicermati bahwa pendekatan perhitungan waktu kedatangan dalam model masih bersifat deterministik, dengan mengasumsikan waktu tempuh sebagai hasil dari pembagian jarak antar titik dengan kecepatan kendaraan yang konstan.

Pendekatan tersebut sudah cukup memadai untuk konteks pemodelan awal dan simulasi optimasi rute secara teoritis. Akan tetapi, dalam implementasi nyata, waktu kedatangan kendaraan di lokasi pelanggan tidak hanya dipengaruhi oleh jarak, tetapi juga oleh berbagai faktor dinamis seperti kondisi lalu lintas, kemacetan, dan kondisi yang tidak dapat diprediksi serta karakteristik geografis rute. Dalam hal ini, konsep *Estimated Time of Arrival* (ETA) menjadi sangat relevan untuk memberikan estimasi waktu tiba yang lebih realistis dan mendekati kenyataan operasional. Dalam model yang lebih lanjut, ETA dapat dihitung secara dinamis dengan mempertimbangkan variabel waktu layanan, kecepatan rata-rata yang bervariasi, atau bahkan integrasi data real-time seperti trafik lalu lintas aktual. Dengan pendekatan tersebut, penilaian terhadap keterlambatan menjadi lebih akurat dibanding dengan memperhitungkan jarak sebagai estimasi waktu tempuh.

Selain itu, pendekatan *time window* yang digunakan dalam penelitian ini bersifat tegas, di mana keterlambatan dihitung secara biner (*on-time* atau *late*). Yang mana dalam kondisi nyatanya, seringkali pelanggan masih dapat menerima pengiriman meskipun terjadi keterlambatan dalam batas tertentu. Maka dari itu, pendekatan *fuzzy time window* dapat dipertimbangkan untuk mengakomodasi kondisi lapangan tersebut.

5.5 Analisis Dampak Lingkungan (*Sustainability*)

Penelitian ini tidak secara eksplisit mempertimbangkan emisi karbon dalam fungsi objektif maupun dalam mencapai hasil optimasi. Namun, pendekatan minimasi jarak tempuh dan keterlambatan yang diimplementasikan melalui algoritma *Nearest Insertion* dan *Genetic Algorithm* secara tidak langsung berkontribusi terhadap pengurangan dampak lingkungan, khususnya dalam konteks efisiensi energi dan emisi kendaraan.

Berdasarkan hasil optimasi, diperoleh penurunan jarak tempuh dan keterlambatan pengiriman pada sebagian besar rute. Efisiensi tersebut menunjukkan bahwa kendaraan cenderung menempuh lintasan yang lebih singkat dan waktu operasional yang lebih teratur. Dengan asumsi bahwa emisi kendaraan berbasis bahan bakar fosil sebanding dengan jarak tempuh dan waktu operasional, maka solusi yang lebih efisien secara rute juga berimplikasi pada potensi penurunan emisi gas buang. Hal ini menjadi indikator awal bahwa penerapan metode optimasi rute tidak hanya berdampak terhadap aspek biaya dan ketepatan waktu, tetapi juga memiliki relevansi terhadap tujuan keberlanjutan logistik.

Untuk penelitian selanjutnya, analisis ini dapat dikembangkan lebih lanjut dengan memasukkan estimasi emisi karbon berdasarkan jenis kendaraan, jarak, dan durasi operasional sebagai salah satu bentuk evaluasi. Penambahan aspek keberlanjutan tersebut akan menyempurnakan penelitian dan menghasilkan solusi yang optimal tidak hanya dari sisi operasional, tetapi juga ramah lingkungan.

5.6 Analisis Faktor Kemacetan (*Traffic)*

Faktor kemacetan lalu lintas tidak dipertimbangkan dalam penelitian ini karena adanya keterbatasan data *real-time* yang akurat serta konsisten pada seluruh jalur distribusi. Selain itu, memasukkan variabel *traffic* akan membuat permasalahan semakin kompleks dan mengubah model menjadi *Time Dependent VRP*, yang berada di luar ruang lingkup penelitian ini. Penelitian ini difokuskan pada optimasi jarak tempuh, waktu tempuh, dan keterlambatan pengiriman berdasarkan *time window* pelanggan, sehingga penambahan faktor *traffic*

dikhawatirkan akan mengubah fokus penelitian. Dalam praktik operasional, perusahaan juga umumnya menggunakan asumsi kecepatan konstan sesuai standar operasional yang berlaku yaitu 22 Km/Jam. Dengan demikian, penggunaan kecepatan rata-rata tersebut dianggap cukup representatif untuk penelitian ini, sedangkan pengaruh *traffic* dan pengembangan model *Time Dependent* VRP dapat dijadikan sebagai bahan pengembangan pada penelitian selanjutnya, misalnya melalui pendekatan *time dependent VRPTW* yang terintegrasi dengan data lalu lintas *real-time.*

BAB VI
PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian mengenai penerapan metode *Capacitated Vehicle Routing Problem with Time Windows* (CVRP-TW) untuk optimalisasi rute distribusi di PT Pos Logistik Indonesia, serta analisis terhadap perbandingan antara kebijakan aktual perusahaan dan hasil optimasi algoritmik dengan menggunakan algoritma *Nearest Insertion* dan *Genetic Algorithm*, maka dapat disimpulkan beberapa poin utama sebagai berikut:

1. Penelitian ini menunjukkan bahwa penerapan pendekatan CVRP-TW dengan algoritma optimasi *Nearest Insertion* dan *Genetic Algorithm* secara signifikan mampu menurunkan total jarak tempuh distribusi barang. Berdasarkan data aktual yang diperoleh dari hasil wawancara dengan pihak PT Pos Logistik Indonesia, distribusi dilakukan menggunakan 3 kendaraan untuk melayani 100 titik pengantaran, menghasilkan total jarak tempuh sebesar 500,1 km. Setelah dilakukan proses optimasi rute menggunakan algoritma *Nearest Insertion* dan *Genetic Algorithm*, rute usulan yang juga menggunakan 3 kendaraan berhasil menurunkan total jarak tempuh menjadi 366,99 km. Dengan demikian, penurunan jarak tempuh yang dicapai sebesar 133,1 km, atau secara persentase terjadi efisiensi jarak tempuh sebesar 26,62%. Penurunan ini mengindikasikan bahwa optimasi urutan titik kunjungan, pembagian beban kendaraan yang lebih merata, dan pemenuhan batasan *time windows* secara simultan mampu meningkatkan efisiensi distribusi logistik secara signifikan.
2. Selain dari sisi jarak, pendekatan CVRP-TW juga terbukti efektif dalam menurunkan tingkat keterlambatan pengiriman dan total biaya penalti yang ditanggung perusahaan. Berdasarkan skenario aktual, rata-rata waktu kedatangan ke pelanggan tidak terdefinisi secara sistematis, sehingga potensi keterlambatan sangat terbuka, terutama pada pelanggan dengan batas waktu pelayanan yang ketat. Dalam simulasi rute eksisting perusahaan, diasumsikan terdapat sekitar 16% pelanggan (16 dari 100 titik) yang mengalami keterlambatan pengantaran karena urutan kunjungan yang kurang efisien dan tidak mempertimbangkan batas waktu pengantaran secara eksplisit. Jika penalti keterlambatan yang berlaku adalah Rp2.000 per pelanggan terlambat, maka estimasi total biaya penalti dengan keterlambatan tersebut yakni sebesaqr Rp.32.000,00. Sementara itu,

hasil optimasi rute dengan algoritma CVRP-TW mampu menurunkan jumlah keterlambatan secara drastis. Dari hasil optimasi, ketiga kendaraan tidak mengalami keterlambatan, baik itu pada titik pelanggan maupun titik kepulangan ke depot sehingga total biaya penalti yang harus ditanggung yakni sebesar Rp.0 atau tanpa keterlambatan. Artinya, terjadi penurunan biaya penalti sebesar Rp32.000, atau secara persentase sebesar 100%. Penurunan ini menunjukkan bahwa metode optimasi tidak hanya meningkatkan efisiensi operasional dari sisi jarak dan waktu tempuh, namun juga secara langsung berdampak pada penghematan biaya yang dikeluarkan akibat keterlambatan pengiriman.

6.2 Saran

1. Bagi Perusahaan
   - Berdasarkan hasil penelitian yang dilakukan pada kasus distribusi di PT. Pos Logistik Indonesia, penerapan algoritma optimasi rute seperti *Nearest Insertion* dan Genetic Algorithm dapat memberikan perbaikan signifikan dalam efisiensi jarak tempuh dan keterlambatan pengiriman. Terbukti dengan penurunan jarak tempuh sebesar 26,62% dan penurunan keterlambatan sebesar 100%. Dengan bukti tersebut perusahaan dapat mengaplikasikan rute optimal untuk distribusi area Jakarta Raya sebagai berikut:
     a) Rute Kendaraan 1: 0 - 57 - 38 - 68 - 20 - 42 - 58 - 61 - 82 - 96 - 54 - 30 - 29 - 99 - 35 - 79-73 - 2 - 19 - 15 - 27 - 34 - 11 - 92 - 64 - 66 - 89 - 23 - 74 - 55 - 85 - 88, 77 - 51 – 0.
     b) Rute Kendaraan 2: 0 - 72 - 80 - 40 - 63 - 37 - 91 - 78 - 90 - 39 - 5 - 9 - 50 - 65 - 13 - 59 - 36 - 21 - 17 - 84 - 81 - 52 - 100 - 87 - 45 -3 - 24 - 70 - 46 - 53 - 44 - 83 - 93 - 75 – 0.
     c) Rute Kendaraan 3: 0 - 32 - 49 - 18 - 43 - 2 - 16 - 8 - 67 - 56 - 71- 7 - 1 - 62 - 31 - 98 - 4 - 76 - 69 - 95 - 25 - 33 - 26 - 14 - 28 - 94 - 6 - 86 - 60 - 97 - 48 - 41 - 47 - 10 - 22 – 0.
   - Untuk memperkuat analisis keterlambatan, perusahaan dapat mengklasifikasikan keterlambatan menjadi beberapa tingkat, seperti ringan (<30 menit), sedang (30–60 menit), dan berat (>60 menit). Dengan pendekatan ini, perusahaan dapat menetapkan strategi mitigasi yang berbeda untuk masing-masing jenis keterlambatan. Selain itu, sebaran lokasi pelanggan yang terlambat juga bisa dianalisis untuk mengidentifikasi

area yang konsisten menyebabkan keterlambatan. Analisis ini bertujuan untuk perencanaan ulang zona layanan (*time windows*) atau penyesuaian jadwal pengiriman.

- Perusahaan dapat mempertimbangkan penerapan sistem *Estimated Time Arrival* (ETA) berbasis waktu tempuh aktual. Hal ini dapat dilakukan dengan memperkirakan durasi perjalanan menggunakan data historis, kecepatan rata-rata, atau integrasi API pemetaan. Pendekatam ini mampu mengukur keterlambatan dengan lebih realistis dan meningkatkan pelayanan pelanggan.

2. Bagi Penelitian Selanjutnya

- Penelitian lanjutan dapat menerapkan *time-dependent travel time matrix* sebagai dasar perhitungan ETA dan menerapkan pendekatan *fuzzy time window*, sehingga ketepatan waktu tidak dinilai secara kaku, melainkan dalam rentang toleransi. Ini akan memperkaya analisis dan memberikan solusi rute yang lebih sesuai dengan kondisi nyata. Selain itu, untuk memperoleh perhitungan waktu yang lebih akurat, *matrix* jarak yang dihitung (ETA) dapat dikalikan dengan faktor pengali yaitu 1,3 untuk mengkompensasikan kemacetan.
- Disarankan agar penelitian selanjutnya dapat memasukkan variabel emisi karbon dalam fungsi objektif. Pendekatan ini dapat dilakukan dengan memperhitungkan konsumsi energi berdasarkan demand per titik, jarak tempuh, dan berat muatan, sehingga solusi rute tidak hanya ekonomis namun memperhatikan segi keberlanjutan lingkungan.

DAFTAR PUSTAKA

Affenzeller, M., Wagner, S., Winkler, S. & Beham, A., 2009. *Genetic Algorithm and Genetic Programming: Modern Concept and Practical Application.* Vol 6 ed. London, New York: CRC PRESS: Taylor & Francis Group.

Amin, I. H. A. & W., 2021. IMPLEMENTASI METODE HAVERSINE UNTUK PENCARIAN OPTICAL DISTRIBUTION POINT. *Dinamika Informatika,* Volume 12 No. 1(E-ISSN : 2714-8769 | P-ISSN : 2085-3343), pp. 28-35.

Andani, S. R., 2023. Optimasi Rute Menggunakan Vehicle Routing Problem (VRP) Dengan Algoritma Genetika. *KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen),* Vol. 4, No. 1, Januari (2023), pp. 148-156(Terakreditasi Nomor 204/E/KPT/2022), pp. pp. 148-156.

Andarwulan, N., 2022. STANDAR MUTU CPO UNTUK MINYAK MAKAN DAN PENERAPAN CARA PRODUKSI MINYAK SAWIT YANG BAIK DI INDONESIA: SUATU KENISCAYAAN.. *Policy Brief Pertanian, Kelautan, dan Biosains Tropika, 4(1),* pp. 245-248..

Andoko, A. & Widodoro, 2013. Berkebun Kelapa Sawit " Si Emas Cair". *Jakarta : AgroMedia Pustaka..*

Anggaraeni, W. A. P. & Zainafree, I., 2022. Analisis Manajemen Logistik BMHP pada Masa Pandemi COVID-19. *JOURNAL OF PUBLIC HEALTH RESEARCH AND DEVELOPMENT (HIGEIA),* Vol. 6 No. 3(p ISSN 2541-5581 e ISSN 2541-5603), pp. 436-444.

Anon., 2025. *Pos Logistics.* [Online] Available at: https://poslogistics.co.id/ [Accessed 27 April 2025].

Anshori, D. R., 2019. *OPTIMASI VEHICLE ROUTING PROBLEM WITH TIME WINDOWS (VRPTW) PADA DISTRIBUSI KUE MENGGUNAKAN ALGORITMA GENETIKA,* Surabaya: Departemen Matematika Fakultas Matematika, Komputasi, dan Sains Data Institut Teknologi Sepuluh Nopember.

Area, B. U. M., 2024. *Optimasi: Konsep, Teknik, dan Aplikasinya.* [Online] Available at: https://bpjiid.uma.ac.id/2024/06/10/optimasi-konsep-teknik-dan-aplikasinya/ [Accessed 17 Januari 2025].

Ariansyah, T. S., Fakhriza, M. & Santoso, H., 2024. APLIKASI PELAYANAN DAN PENGELOLAAN DISTRIBUSI LAPTOP DENGAN IMPLEMENTASI ALGORITMA DETERMINISTIK. *Jurnal Sistem Informasi Bisnis (JUNSIBI),* Vol. 5 No. 2(p-ISSN: 2774-3446 e-ISSN: 2774-3454).

Azhar, A. H. & Destari, R. A., 2021. Analisis Konsumen Memilih Minyak Makan Kemasan Menggunakan Analytical Hierarchy Process (AHP). *INFOSYS (INFORMATION SYSTEM) JOURNAL, 6(1),* pp. 63-71.

Chairunisah & Sihotang, D., 2021. OPTIMASI JUMLAH PRODUKSI MAKANAN OLAHAN DENGAN MENGGUNAKAN METODE BRANCH AND BOUND. *JURNAL KARISMATIKA,* VOL. 7 NO. 1 APRIL 2021(p-ISSN : 2443-0366 e-ISSN : 2528-0279), pp. 52-63.

Dewangga, A. & Suseno, S., 2022. Analisa Pengendalian Kualitas Produksi Plywood Menggunakan Metode Seven Tools, Failure Mode And Effect Analysis (FMEA), Dan TRIZ.. *Jurnal Teknologi dan Manajemen Industri Terapan, 1(3),* pp. 243-253..

Dinia, A., Cholissodin, I. & Santoso, E., 2021. Optimasi Pengiriman Barang menggunakan Algoritme Genetika dengan Data Sintesis. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* Vol. 5, No. 10(hlm. 4658-4664), pp. hlm. 4658-4664.

Eche, O. V., Okeyinka, A. E., Abdullah, I. & Abdulrahman, A., 2025. Review of Algorithms to Solve Travelling Salesman. *Journal of Science Innovation & Technology Research (JSITR),* VOL. 05 NO. 9,(E-ISSN 3026-958X), pp. 182-187.

Engelbrecht., A. P., 2006.. *Fundamentals of computational swarm intelligence.* s.l.:John Wiley & Sons.

Erlangga , D. A., 2018. *PERANCANGAN MESIN PENIRIS MINYAK (SPINNER) UNTUK KEBUTUHAN DAPUR RUMAH TANGGA DENGAN MENGGUNAKAN METODE TRIZ,* s.l.: dspaceuii.

Fahreza, S., Walansari, S. P., Putri, H. D. & Fenanlampir, D. R., 2023. Logistik Dan Supplay Chain Management. *Jurnal Ilmiah Wahana Pendidikan,* pp. 622-628 .

Farhan, S. & Hidayat, Y. R., 2021. PENGARUH LOGISTIC MANAGEMENT IMPROVEMENT TERHADAP KINERJA WAREHOUSE PADA PT. FASTINDO PIRANTI KABEL. *Jurnal Manajemen Logistik ,* Vol.1, No1(E-ISSN 2621-6442), pp. pp. 67-71.

Fauzi, M. & Hartati, V., 2021. *Modul Perkuliahan: Manajemen Transportasi dan Distribusi.* s.l.:Biro Akademik dan Pembelajaran Universitas Widya Tama.

Febiani, A. et al., 2024. Implementasi Algoritma “Particle Swarm Optimization” (PSO) Penjadwalan Belajar Mengajar. *IKRAITH-INFORMATIKA ,* Vol 8 No 1 Maret 2024(P-ISSN :2580-4316).

Fitriastutik, E., 2020. *OPTIMASI PENGANGKUTAN SAMPAH OLEH DKRTH SURABAYA: PENYUSUNAN PENGANGKUTAN SAMPAH DAN KAJIAN PENGGANTIAN KENDARAAN,* Surabaya: INSTITUT TEKNOLOGI SEPULUH NOPEMBER.

Galih, S. A. M. & Sukmadewi, R., 2024. OPTIMALISASI ALUR DISTRIBUSI: MEMPERLANCAR EFISIENSI MELALUI DAFTAR HARGA PENGIRIMAN DAN MODA TRANSPORTASI YANG STRATEGIS. *Jurnal Ekonomi, Bisnis dan Manajemen,* Volume. 11 No. 1(E-ISSN: 2620-3391), pp. 77-86.

Gunawan, V., 2020. *Penerapan Dragon Fly Algorithm untuk Menyelesaikan Capacitated Vehicle Routing Problem with Time Windows,* Bandung: Universitas Katolik Parahyangan.

Hajar, G. & Fauzi, M. D., 2022. OPTIMASI PENENTUAN RUTE PENGIRIMAN DENGAN VEHICLE ROUTING PROBLEM SIMULTANEOUS DELIVERY AND PICKUP WITH SPLIT LOAD. *Journal of Industrial Engineering and Operation Management,* Vol. 05, No. 01,(ISSN: 2620-8184), pp. 84-91..

Hajar, G., Rachmaniar, D. N. & Fauzi, M. D., 2024. Penentuan Rute Pembukaan Gerai Baru dengan Vehicle Routing Problem with Time Windows Pendekatan Nearest Neighbor. *Go-Integratif : Jurnal Teknik Sistem dan Industri ,* Volume Vol. 05, No. 01,, pp. 12-21.

Hand, R., 2024. *How to Manage Warehouse Overflow and Optimize Your Ecommerce Storage.* [Online]
Available at: https://www.shipbob.com/blog/overflow-warehouse/
[Accessed 07 Januari 2025].

Haryotejo, B. & Kusumawardhani, A., 2015. ANALISIS PENGARUH KINERJA LOGISTIK PEMASOK TERHADAP KINERJA BISNIS (Studi Pada Bengkel AHASS di Kota Semarang). *DIPONEGORO JOURNAL OF MANAGEMENT ,* Volume 4, Nomor 3(ISSN (Online): 2337-3792), pp. 1-6.

Hasad, A., 2011. *Algoritma Optimasi Probabilistik dan Deterministik.* [Online] Available at: https://andihasad.com/2011/11/14/algoritma-optimasi-probabilistik-dan-deterministik/
[Accessed 17 Januari 2025].

Hasbiyati, I. & Hasriati, 2017. Beberapa Metode pada Masalah Pemrograman Stokastik. *JURNAL MATEMATIKA "MANTIK",* Vol. 03 No. 02(ISSN: 2527-3159 E-ISSN: 2527-3167).

Isai, A. & Nugroho, A., 2024. Pencarian Rute Terpendek menggunakan Algoritma Genetika (Studi Kasus : Pengoptimalan Mobilitas Kota Salatiga Terhadap Kota-Kota Tetangga). *Jurnal Indonesia : Manajemen Informatika dan Komunikasi,* Vol 5 No 1, Januari (E-ISSN: 2723-7079, P-ISSN: 2776-8074), pp. 681-692.

Istiqlaliyah, H., 2015. Perancangan mesin peniris minyak pada keripik nangka dengan kapasitas 2,5 Kg/menit.. *Nusantara of Engineering, 2,* pp. 37-43..

Juanda, D. & Cahyono, B., 2005. Wijen, Teknik Budi Daya dan Analisis Usaha Tani.. *Yogyakarta : Kanisius..*

Karazmoodeh, A., 2023. *Crossovers in genetic algorithms.* [Online] Available at: https://www.linkedin.com/pulse/crossovers-genetic-algorithms-ali-karazmoodeh-tthjf/
[Accessed 08 Maret 2025].

Kasengkang, R. A., Nangoy, S. & Sumarauw, J., 2016. ANALISIS LOGISTIK (STUDI KASUS PADA PT. REMENIA SATORI TEPAS-KOTA MANADO). *Jurnal Berkala Ilmiah Efisiensi,* Volume Volume 16 No. 01 , pp. 750-759.

Ketaren, 2012. . Pengantar Teknologi Minyak dan Lemak Pangan. *Jakarta : UI-Press.*

Kundori, K., 2023. *Implementasi Kebijakan Transportasi Laut dalam Rangka Pengembangan Sistem Logistik Nasional.* 21(1), ed. Yogyakarta: Majalah Ilmiah Bahari Jogja.

Kurniawan, I. S., Susanty, S. & Adianto, H., 2014. Usulan Rute Pendistribusian Air Mineral Dalam Kemasan Menggunakan Metode Nearest Neighbour dan Clarke & Wright Savings (Studi Kasus di PT. X Bandung). *Jurnal Online Institut Teknologi Nasional,* No.04 | Vol. 01(Reka Integra ISSN: 2338-5081), pp. 125-146.

Lydia, P. R., Suyanto & Dayawati, R. N., 2011. *CAPACITATED VEHICLE ROUTING PROBLEM TIME WINDOWS (CVRPTW) MENGGUNAKAN ALGORITMA HARMONY SEARCH,* Bandung: Universitas Telkom.

Marpaung, B., 2013. PEMECAHAN MASALAHOPTIMASI BERSIFAT PROBABILISTIK MENGGUNAKANCHANGE-CONSTRAINED PROGRAMMING. *Jurnal Teknik dan Ilmu Komputer,* Volume Vol. 02No. 05.

Marta, E. S., 2023. Pola Makan Mengandung Kadar Lemak Jenuh dan Kolesterol Tinggi Mengakibatkan Peningkatan Tekanan Darah dan Stroke pada Penderita Hypertensi di Puskesmas Padang Tahun 2023.. *Jurnal Amanah Kesehatan, 5(2),* pp. 18-23..

Mustafa, B., 2017. Pengantar Hukum Administrasi Logistik. In: Bekasi: PT. Citra Aditya Bakti, pp. 5-6.

Nasution, M. N., 2008. *Manajemen Transportasi.* Bogor: Ghalia Indonesia.

Nazarena, Y., Mardiana, M. & Sriwiyanti, S., 2023. Pencegahan Penyakit Jantung Koroner Melalui Pola Konsumsi Yang Sehat.. *Community Development Journal: Jurnal Pengabdian Masyarakat, 4(2),* pp. 2238-2245..

Novianto, A. P., 2016. *Perancangan Wearpack Menggunakan Metode Quality Function Deployment Untuk Oil And Gas Industry (Studi Kasus Di Refinery Unit V Balikpapan),* s.l.: dspaceuii.

Nugraha, R. & Haryono, K., 2022. Metode TRIZ Untuk Meningkatkan Kreativitas Dan Inovasi Pada Bidang Bisnis dan Manajemen Melalui Aplikasi Berbasis Mobile.. *AUTOMATA, 3(2)..*

Nurcahyo, A. D. & Winanti, E. T., 2021. Pengaruh model Problem Based Learning terintegrasi pendekatan induktif terhadap kemampuan berpikir spasial dan pengetahuan siswa pada materi mitigasi bencana.. *Jurnal Pendidikan Geografi, 26(1),* pp. 41-47..

Nurlela, N., 2020. Analisa Bilangan Peroksida Terhadap Kualitas Minyak Goreng Sebelum Dan Sesudah Dipakai Berulang.. *Jurnal Redoks, 5(1),* pp. 65-71..

Nursyanti, Y. & Shalsabila, K., 2020. Optimasi Inventori Dengan Pendekatan Deterministik Dinamis Pada Industri Manufaktur Roti. *Jurnal OPSI - Jurnal Optimasi Sistem Industri,* I Vol 13 No.2(ISSN 1693-2102 (print), 2686-2352 (online)).

Prasetio, P. J. & Ibik, M. K., 2015. Rancang bangun keripik mangga podang kapasitas 10 kg per proses (Bagian: Mesin Peniris).. *Jurnal Teknik Mesin, 4,* pp. 1-25..

Prasetyo, P. E., Susetyo, A. E. & Susanti, D. A., 2021. Perancangan Alat Bantu dan Aktifitas Toilet Portabel Tunadaksa Bagian Bawah dengan Mengintegrasikan Tahapan Front-End Process dan Metode TRIZ. *Science Tech: Jurnal Ilmu Pengetahuan dan Teknologi, 7(2),* pp. 22-38.

Pratiwi, A. I. et al., 2023. Penentuan Rute Terbaik Pendistribusian Produk Wafer dengan Metode Algoritma Genetika (Studi Kasus di Perusahaan Jasa Pergudangan Produk Wafer Karawang). *JISI: JURNAL INTEGRASI SISTEM INDUSTRI,* Volume 10 No. 2(E-ISSN: 2550-083X), pp. 58-164.

PT Pos Logistik Indonesia, 2025. *Pos Logistics.* [Online] Available at: https://poslogistics.co.id/ [Accessed 27 April 2025].

Pujawan, I. N. & Mahendrawati, E., 2010. *Supply Chain Management.* Edisi kedua ed. Surabaya: Guna Widya.

Purnomo, R. E., 2019. *USULAN RUTE DISTRIBUSI DI CV AGRINDO SUPRAFOOD DENGAN MENGGUNAKAN METODE SAVING MATRIX,* Yogyakarta: E-Journal Universitas Atma Jaya Yogyakarta.

Puspitorini, P. S. & Sabdoadi, S. N., 2019. LITERATURE REVIEW: CASCADING FAKTOR DISRUPSI DAN RISIKO KE STRATEGI LOGISTIK PADA THIRD PARTY LOGISTIC. *Seminar Nasional Penelitian dan Pengabdian Masyarakat LP4MP Universitas Islam Majapahit,* pp. 1-9.

Rahimi, A. et al., 2023. Implementasi Algoritma Genetika Dalam Penentuan Rute Terbaik Pendistribusian BBM Pada SPBU Yang Ada di Samarinda. *Prosiding Seminar Nasional Matematika, Statistika, dan Aplikasinya ,* Terbitan III, Agustus 2023(e-ISSN: 2657-232X), pp. 196-207.

Rahmah, A. N. & Ningrat, N. K., 2020. PENENTUAN MODA TRANSPORTASI UNTUK EFISIENSI BIAYA KIRIM DENGAN METODE AHP PADA IKM KERUPUK IDAMAN CIAMIS. *JURNAL INDUSTRIAL GALUH (JIG),* Volume Vol. 2 (2) 2020.

Rahmatullah, M. Z., 2019. Penerapan Kombinatorial pada Vehicle Routing. *Program Studi Teknik Informatika, Makalah IF2120 Matematika Diskrit – Sem. I.*

Rahmiyati, A. L. & Irianto, G., 2021. *Teori dan Praktek Manajemen Logistik Rumah Sakit.* Monograf ed. Bandung: Refika aditama.

Rajeev, S. & Krishnamoorthy, C., 1992. Discrete optimization of structures using genetic algorithms. *Journal of structural engineering.*

Ramadhan, A. R., 2021. Perancangan Mesin Pembuat Pelet Apung Berbahan Maggot Berkapasitas 20 Kg/Jam dengan Metode TRIZ.. *In Prosiding Industrial Research Workshop and National Seminar (Vol. 12, pp. 283-288)..*

Sahputri, J. & Harrika, F., 2023. Pengaruh Kadar Glukosa Darah Puasa Pada Mencit Yang Di Induksi Minyak Jelantah.. *Jurnal Ilmiah Manusia Dan Kesehatan, 6(1),* pp. 132-141..

Septiandre & Siswanto, N., 2016. Penentuan Lokasi Gudang Penyangga Regional PT. "X" Wilayah Jawa Timur. *Jurnal Studi Manajemen Dan Bisnis,* Volume di Manajemen Dan Bisnis, pp. 184-194.

Siraj, M. A., 2024. OPTIMASI JALUR DISTRIBUSI PADA UMKM MITRA TELUR DENGAN MENGGUNAKAN METODE SAVING MATRIX DAN NEAREST NEIGHBOR. *JURNAL ILMIAH SAINS TEKNOLOGI DAN INFORMASI ,* Vol.2, No.2 April(e-ISSN: 2964-3104; p-ISSN: 2964-3090), pp. 29-38.

Soenandi, I. A., Joice & Marpaung, B., 2019. OptimasiCapacitated Vehicle Routing Problem with Time Windows dengan Menggunakan Ant Colony Optimization. *Jurnal Sistem dan Manajemen Industri,* Volume Vol 3 No 1, pp. 59-66.

Sucahyowati, H., 2011. MANAJEMEN RANTAI PASOKAN (SUPPLY CHAIN MANAGEMENT). *"GEMA MARITIM" ,* Volume Vol 13 No. 1 Pebruari 2011, pp. 20-18.

Sugiono, 2013. *Metode Penelitian Kuantitatif Kualitatif, dan R & D.* (19th ed.) ed. s.l.:Alfabeta.

Sugiono, M. C., 2022. Model vehicle routing problem untuk penentuan rute distribusi unit sepeda motor dengan metode saving matrix. *Journal Industrial Servicess,* Volume Vol. 7, no. 2, p. pp. 230–233.

Sugiyono, 2011. Metode Penelitian Kuantitatif, Kualitatif dan R&D. *Bandung. Alfabeta.*

Sukartin, K. & Sitanggang, M., 2005. Gempur Penyakit dengan VCO.. *Jakarta: AgroMedia Pustaka..*

Sunardi, Y., 2020. *Memahami Rantai Pasok.* [Online] Available at: https://www.yosephsunardhi.com/2020/02/memahami-rantai-pasok.html#google_vignette [Accessed 18 Januari 2024].

Sunaryo, Istiyanto, B., Wicaksono, P. & Ermanto, S. A., 2021. PENGEMBANGAN SISTEM INFORMASI LOGISTIK "INVENTORY MANAGEMENT" UNTUK MENDUKUNG PEMBELAJARAN TARUNA D.III MANAJEMEN LOGISTIK POLTRADA BALI. *Jurnal Teknologi Transportasi dan Logistik Pusat Penelitian dan Pengabdian Masyarakat Politeknik Transportasi Darat Bali ,* Volume 2 No 1, Juni 2021(p-ISSN 2716-2990 e-ISSN 2745-8954), pp. Hal 57 - 64 .

Sundarningsih, D., Mahmudy, W. F. & S., 2017. Penerapan Algoritma Genetika untuk Optimasi Vehicle Routing Problem with Time Window (VRPTW) Studi Kasus Air Minum Kemasan. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* Vol. 1, No. 2, Februari (e-ISSN: 2548-964X), pp. 100-107 .

supplychainindonesia.com, n.d. *PERKEMBANGAN DAN PERMASALAHAN LOGISTIK NASIONAL.* [Online] Available at: http://supplychainindonesia.com/new/wp-content/files/Bab_2_Perkembangan_dan_Permasalahan_Logistik_Nasional.pdf [Accessed 2024].

Suriani, N., Risnita & Jailan, M., 2023. Konsep Populasi dan Sampling Serta Pemilihan Partisipan Ditinjau Dari Penelitian Ilmiah Pendidikan. *IHSAN: Jurnal Pendidikan Islam,* Volume 1 Nomor 2(E-ISSN 2987-1298).

Sutini & Wismana, R., 2021. PERANAN LOGISTIK TERHADAP PERUSAHAAN UNTUK MENUNJANG KELANCARAN DUNIA BISNIS. *JURNAL JIMEK (JURNAL ILMU MANAJAMEN, EKONOMI DAN KEWIRAUSAHAAN),* Vol 1 No. 2 (E-ISSN : 2809-9427 P-ISSN : 2809-9893), pp. page 01-07.

Syah, A. N. A., 2005. Virgin Coconut Oil : Minyak Penakluk Aneka Penyakit. *Jakarta : AgroMedia Pustaka..*

Syahputra, R. F., 2024. Menganalisis Konsep Dasar Algoritma Genetika. *Bhinneka: Jurnal Bintang Pendidikan dan Bahasa,* VOLUME 2, NO. 1, JANUARI 2024(e-ISSN: 2963-6167; p-ISSN: 2963-6183), pp. 120-132.

Tanjung, I., Mawengkang, H. & Sawaluddin, 2023. Direct Search Techniques for Mixed Stochastic Nonlinear Programming Model. *Jurnal dan Penelitian Teknik Informatika,* Volume 7, Number 3(e-ISSN : 2541-2019 p-ISSN : 2541-044X).

Tanujaya, W., Dewi, D. R. S. & Endah, D., 2011. PENERAPAN ALGORITMA GENETIK UNTUK PENYELESAIAN MASALAH VEHICLE ROUTING DI PT.MIF. *WIDYA TEKNIK ,* Volume Vol. 10, No. 1, pp. 92-102.

Urrahman, Z., 2009. *MANAJEMEN BUDGETING DAN LOGISTIK KEPERAWATAN.* [Online] Available at: https://rufaidahhomecare.wordpress.com/2009/11/19/88/ [Accessed 18 Januari 2025].

Utomo, P. B. & Setiafindari, W., 2021. OPTIMASI PENJADWALAN PRODUKSI MENGGUNAKAN METODE SIMULATED ANNEALING DI INDUSTRI XYZ. *Borobudur Engineering Review,* Vol. 1 No. 1 (2021) pp. (e-ISSN : 2777-0850), pp. 49-54 .

Widyarto, A., 2012. PERAN SUPPLY CHAIN MANAGEMENT DALAM SISTEM PRODUKSI DAN OPERASI PERUSAHAAN. *BENEFIT Jurnal Manajemen dan Bisnis,* Volume Volume 16, Nomor 2, pp. hlm. 91-98.

Yuan G., 2021. The theory of inventive problem solving (TRIZ)-based strategic mapping of green nuclear energy investments with spherical fuzzy group decision-making approach.. *International Journal of Energy Research, 45(8),* pp. 12284-12300..

Yusuf, A. M. & Soediantono, D., 2022. Supply Chain Management and Recommendations for Implementation in the Defense Industry: A Literature Review. *INTERNATIONAL JOURNAL OF SOCIAL AND MANAGEMENT STUDIES (IJOSMAS),* Vol. 3 No. 3 (e-ISSN : 2775-0809), pp. 63-77.

Zahara, L., Rahmany, M. & Moulana, R., 2024. Penggunaan Phyton dalam Optimasi Bahan Baku pada Perusahaan X. *JURNAL ILMIAH MAHASISWA PERTANIAN ,* Volume 9, Nomor 1, Februari 2024(E-ISSN: 2614-6053 P-ISSN: 2615-2878), pp. 428-433.

LAMPIRAN

```
import math
import random
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

random.seed(42)
np.random.seed(42)

log_population = []
log_generations = []
log_operations = []



# Parameter utama
vehicle_capacity = 1000
vehicle_speed = 22  # km/jam
service_time = 10  # menit
cost_per_customer = 10000
penalty_rate = 0.2
num_vehicles = 3
depot_id = 0

# Contoh data lokasi: (longitude, latitude) -- Ganti dengan data kamu
locations = {
    0: (106.916742, -6.202523),  # Depot
    1: (106.835953, -6.280092),
    2: (106.887305, -6.281572),
    3: (106.819235, -6.279968),
    4: (106.850077, -6.233342),
    5: (106.774135, -6.269443),
    6: (106.809525, -6.212111),
    7: (106.845133, -6.292434),
    8: (106.881958, -6.291090),
    9: (106.793436, -6.267833),
    10: (106.891976, -6.289595),
    11: (106.898122, -6.209064),
    12: (106.881188, -6.225719),
    13: (106.809656, -6.269489),
    14: (106.823482, -6.230990),
    15: (106.888708, -6.218146),
    16: (106.885018, -6.287387),
    17: (106.861826, -6.265529),
    18: (106.844098, -6.240286),
    19: (106.881953, -6.222611),
    20: (106.825788, -6.197719),
    21: (106.836412, -6.251159),
    22: (106.896347, -6.281991),
    23: (106.871795, -6.184190),
    24: (106.805520, -6.294850),
    25: (106.825823, -6.219344),
    26: (106.828024, -6.222171),
    27: (106.894372, -6.213234),
    28: (106.816648, -6.231975),
    29: (106.868407, -6.168974),
    30: (106.859921, -6.168229),
    31: (106.859486, -6.245734),
    32: (106.828957, -6.207175),
    33: (106.828281, -6.221040),
    34: (106.912145, -6.210465),
    35: (106.869765, -6.194733),
    36: (106.835927, -6.243603),
    37: (106.808276, -6.274968),
    38: (106.833572, -6.195469),
    39: (106.789333, -6.295155),
    40: (106.822205, -6.240121),
    41: (106.842677, -6.298913),
    42: (106.809911, -6.174773),
    43: (106.886467, -6.282974),
    44: (106.810636, -6.253956),
    45: (106.842021, -6.264153),
    46: (106.800125, -6.291642),
    47: (106.866565, -6.296323),
    48: (106.869304, -6.238265),
    49: (106.804851, -6.201466),
    50: (106.802915, -6.262841),
    51: (106.900716, -6.204504),
```

```
    52: (106.894460, -6.259981),
    53: (106.788884, -6.296316),
    54: (106.844434, -6.164171),
    55: (106.836247, -6.176636),
    56: (106.871629, -6.286451),
    57: (106.844320, -6.191137),
    58: (106.820175, -6.174079),
    59: (106.832290, -6.243612),
    60: (106.841539, -6.197461),
    61: (106.827254, -6.178995),
    62: (106.831635, -6.259465),
    63: (106.818465, -6.246302),
    64: (106.885036, -6.218770),
    65: (106.803542, -6.268687),
    66: (106.876897, -6.221907),
    67: (106.873810, -6.295405),
    68: (106.817299, -6.191178),
    69: (106.850735, -6.202589),
    70: (106.804147, -6.295848),
    71: (106.854689, -6.294899),
    72: (106.897246, -6.197879),
    73: (106.877758, -6.228875),
    74: (106.858856, -6.173690),
    75: (106.892294, -6.229272),
    76: (106.852149, -6.201513),
    77: (106.878400, -6.196340),
    78: (106.815579, -6.295603),
    79: (106.873986, -6.198975),
    80: (106.877869, -6.266100),
    81: (106.896008, -6.252885),
    82: (106.839621, -6.176756),
    83: (106.836067, -6.254798),
    84: (106.866163, -6.270141),
    85: (106.836196, -6.190860),
    86: (106.815502, -6.207313),
    87: (106.849964, -6.264381),
    88: (106.846757, -6.188341),
    89: (106.869509, -6.200059),
    90: (106.817381, -6.292333),
    91: (106.807501, -6.277920),
    92: (106.885380, -6.210814),
    93: (106.836787, -6.242872),
    94: (106.803855, -6.238716),
    95: (106.843648, -6.202202),
    96: (106.839571, -6.173459),
    97: (106.843722, -6.206440),
    98: (106.854407, -6.237847),
    99: (106.874110, -6.178204),
    100: (106.849477, -6.260628)
}

# Contoh pickup_delivery_demand {customer_id: {'delivery_demand': berat}}
pickup_delivery_demand = {
    # Main Hub (Depot)
    0: {"pickup_demand": 0, "delivery_demand": 0},

    # Customers (1-100)
    1: {"pickup_demand": 0, "delivery_demand": 29},   # Customer 1
    2: {"pickup_demand": 0, "delivery_demand": 22},   # Customer 2
    3: {"pickup_demand": 0, "delivery_demand": 13},   # Customer 3
    4: {"pickup_demand": 0, "delivery_demand": 21},   # Customer 4
    5: {"pickup_demand": 0, "delivery_demand": 7},    # Customer 5
    6: {"pickup_demand": 0, "delivery_demand": 16},   # Customer 6
    7: {"pickup_demand": 0, "delivery_demand": 29},   # Customer 7
    8: {"pickup_demand": 0, "delivery_demand": 28},   # Customer 8
    9: {"pickup_demand": 0, "delivery_demand": 11},  # Customer 9
    10: {"pickup_demand": 0, "delivery_demand": 21},  # Customer 10
    11: {"pickup_demand": 0, "delivery_demand": 24},  # Customer 11
    12: {"pickup_demand": 0, "delivery_demand": 1},   # Customer 12
    13: {"pickup_demand": 0, "delivery_demand": 26},  # Customer 13
    14: {"pickup_demand": 0, "delivery_demand": 12},  # Customer 14
    15: {"pickup_demand": 0, "delivery_demand": 5},   # Customer 15
    16: {"pickup_demand": 0, "delivery_demand": 13},  # Customer 16
    17: {"pickup_demand": 0, "delivery_demand": 30},  # Customer 17
    18: {"pickup_demand": 0, "delivery_demand": 22},  # Customer 18
    19: {"pickup_demand": 0, "delivery_demand": 12},  # Customer 19
    20: {"pickup_demand": 0, "delivery_demand": 27},  # Customer 20
    21: {"pickup_demand": 0, "delivery_demand": 1},   # Customer 21
    22: {"pickup_demand": 0, "delivery_demand": 20},  # Customer 22
    23: {"pickup_demand": 0, "delivery_demand": 30},  # Customer 23
    24: {"pickup_demand": 0, "delivery_demand": 16},  # Customer 24
    25: {"pickup_demand": 0, "delivery_demand": 7},   # Customer 25
    26: {"pickup_demand": 0, "delivery_demand": 25},  # Customer 26
    27: {"pickup_demand": 0, "delivery_demand": 30},  # Customer 27
    28: {"pickup_demand": 0, "delivery_demand": 11},  # Customer 28
```

```
    29: {"pickup_demand": 0, "delivery_demand": 11},  # Customer 29
    30: {"pickup_demand": 0, "delivery_demand": 27},  # Customer 30
    31: {"pickup_demand": 0, "delivery_demand": 3},   # Customer 31
    32: {"pickup_demand": 0, "delivery_demand": 9},   # Customer 32
    33: {"pickup_demand": 0, "delivery_demand": 14},  # Customer 33
    34: {"pickup_demand": 0, "delivery_demand": 17},  # Customer 34
    35: {"pickup_demand": 0, "delivery_demand": 21},  # Customer 35
    36: {"pickup_demand": 0, "delivery_demand": 20},  # Customer 36
    37: {"pickup_demand": 0, "delivery_demand": 6},   # Customer 37
    38: {"pickup_demand": 0, "delivery_demand": 4},   # Customer 38
    39: {"pickup_demand": 0, "delivery_demand": 6},   # Customer 39
    40: {"pickup_demand": 0, "delivery_demand": 27},  # Customer 40
    41: {"pickup_demand": 0, "delivery_demand": 3},   # Customer 41
    42: {"pickup_demand": 0, "delivery_demand": 19},  # Customer 42
    43: {"pickup_demand": 0, "delivery_demand": 1},   # Customer 43
    44: {"pickup_demand": 0, "delivery_demand": 15},  # Customer 44
    45: {"pickup_demand": 0, "delivery_demand": 28},  # Customer 45
    46: {"pickup_demand": 0, "delivery_demand": 4},   # Customer 46
    47: {"pickup_demand": 0, "delivery_demand": 24},  # Customer 47
    48: {"pickup_demand": 0, "delivery_demand": 8},   # Customer 48
    49: {"pickup_demand": 0, "delivery_demand": 15},  # Customer 49
    50: {"pickup_demand": 0, "delivery_demand": 2},   # Customer 50
    51: {"pickup_demand": 0, "delivery_demand": 5},   # Customer 51
    52: {"pickup_demand": 0, "delivery_demand": 8},   # Customer 52
    53: {"pickup_demand": 0, "delivery_demand": 20},  # Customer 53
    54: {"pickup_demand": 0, "delivery_demand": 29},  # Customer 54
    55: {"pickup_demand": 0, "delivery_demand": 17},  # Customer 55
    56: {"pickup_demand": 0, "delivery_demand": 20},  # Customer 56
    57: {"pickup_demand": 0, "delivery_demand": 24},  # Customer 57
    58: {"pickup_demand": 0, "delivery_demand": 11},  # Customer 58
    59: {"pickup_demand": 0, "delivery_demand": 24},  # Customer 59
    60: {"pickup_demand": 0, "delivery_demand": 27},  # Customer 60
    61: {"pickup_demand": 0, "delivery_demand": 20},  # Customer 61
    62: {"pickup_demand": 0, "delivery_demand": 10},  # Customer 62
    63: {"pickup_demand": 0, "delivery_demand": 20},  # Customer 63
    64: {"pickup_demand": 0, "delivery_demand": 28},  # Customer 64
    65: {"pickup_demand": 0, "delivery_demand": 26},  # Customer 65
    66: {"pickup_demand": 0, "delivery_demand": 1},   # Customer 66
    67: {"pickup_demand": 0, "delivery_demand": 2},   # Customer 67
    68: {"pickup_demand": 0, "delivery_demand": 21},  # Customer 68
    69: {"pickup_demand": 0, "delivery_demand": 3},   # Customer 69
    70: {"pickup_demand": 0, "delivery_demand": 6},   # Customer 70
    71: {"pickup_demand": 0, "delivery_demand": 28},  # Customer 71
    72: {"pickup_demand": 0, "delivery_demand": 22},  # Customer 72
    73: {"pickup_demand": 0, "delivery_demand": 17},  # Customer 73
    74: {"pickup_demand": 0, "delivery_demand": 13},  # Customer 74
    75: {"pickup_demand": 0, "delivery_demand": 3},   # Customer 75
    76: {"pickup_demand": 0, "delivery_demand": 10},  # Customer 76
    77: {"pickup_demand": 0, "delivery_demand": 17},  # Customer 77
    78: {"pickup_demand": 0, "delivery_demand": 25},  # Customer 78
    79: {"pickup_demand": 0, "delivery_demand": 22},  # Customer 79
    80: {"pickup_demand": 0, "delivery_demand": 24},  # Customer 80
    81: {"pickup_demand": 0, "delivery_demand": 1},   # Customer 81
    82: {"pickup_demand": 0, "delivery_demand": 8},   # Customer 82
    83: {"pickup_demand": 0, "delivery_demand": 27},  # Customer 83
    84: {"pickup_demand": 0, "delivery_demand": 3},   # Customer 84
    85: {"pickup_demand": 0, "delivery_demand": 30},  # Customer 85
    86: {"pickup_demand": 0, "delivery_demand": 30},  # Customer 86
    87: {"pickup_demand": 0, "delivery_demand": 19},  # Customer 87
    88: {"pickup_demand": 0, "delivery_demand": 15},  # Customer 88
    89: {"pickup_demand": 0, "delivery_demand": 20},  # Customer 89
    90: {"pickup_demand": 0, "delivery_demand": 15}, # Customer 90
    91: {"pickup_demand": 0, "delivery_demand": 9},  # Customer 91
    92: {"pickup_demand": 0, "delivery_demand": 24}, # Customer 92
    93: {"pickup_demand": 0, "delivery_demand": 3},  # Customer 93
    94: {"pickup_demand": 0, "delivery_demand": 11}, # Customer 94
    95: {"pickup_demand": 0, "delivery_demand": 22}, # Customer 95
    96: {"pickup_demand": 0, "delivery_demand": 25}, # Customer 96
    97: {"pickup_demand": 0, "delivery_demand": 4},  # Customer 97
    98: {"pickup_demand": 0, "delivery_demand": 7},  # Customer 98
    99: {"pickup_demand": 0, "delivery_demand": 21}, # Customer 99
    100: {"pickup_demand": 0, "delivery_demand": 30}  # Customer 100
}

# Contoh time windows (start, end) dalam menit dari awal hari, ganti sesuai kebutuhan
time_windows = {
    # Main Hub (Depot) - open from 480 to 1260 minutes (08:00 to 21:00 same day)
    0: {"time_window": (480, 1260)},

    # Customers (1-100) - Time Windows based on TW column
    # TW 1: 480-780 (08:00-13:00 same day)
    # TW 2: 781-1020 (13:01-17:00 same day)
    # TW 3: 1021-1260 (17:01-21:00 same day)
    1: {"time_window": (781, 1020)},   # Customer 1 (TW 2)
    2: {"time_window": (781, 1020)},   # Customer 2 (TW 2)
```

```
    3: {"time_window": (1021, 1260)},  # Customer 3 (TW 3)
    4: {"time_window": (781, 1020)},   # Customer 4 (TW 2)
    5: {"time_window": (781, 1020)},    # Customer 5 (TW 1)
    6: {"time_window": (1021, 1260)},  # Customer 6 (TW 3)
    7: {"time_window": (781, 1020)},   # Customer 7 (TW 2)
    8: {"time_window": (781, 1020)},   # Customer 8 (TW 2)
    9: {"time_window": (781, 1020)},   # Customer 9 (TW 2)
    10: {"time_window": (1021, 1260)},  # Customer 10 (TW 3)
    11: {"time_window": (1021, 1260)},  # Customer 11 (TW 3)
    12: {"time_window": (781, 1020)},   # Customer 12 (TW 2)
    13: {"time_window": (781, 1020)},   # Customer 13 (TW 2)
    14: {"time_window": (1021, 1260)},  # Customer 14 (TW 3)
    15: {"time_window": (781, 1020)},   # Customer 15 (TW 2)
    16: {"time_window": (781, 1020)},   # Customer 16 (TW 2)
    17: {"time_window": (781, 1020)},   # Customer 17 (TW 2)
    18: {"time_window": (480, 780)},   # Customer 18 (TW 2)
    19: {"time_window": (781, 1020)},   # Customer 19 (TW 2)
    20: {"time_window": (781, 1020)},   # Customer 20 (TW 2)
    21: {"time_window": (781, 1020)},  # Customer 21 (TW 3)
    22: {"time_window": (1021, 1260)},  # Customer 22 (TW 3)
    23: {"time_window": (1021, 1260)},  # Customer 23 (TW 3)
    24: {"time_window": (1021, 1260)},  # Customer 24 (TW 3)
    25: {"time_window": (1021, 1260)},  # Customer 25 (TW 3)
    26: {"time_window": (1021, 1260)},  # Customer 26 (TW 3)
    27: {"time_window": (781, 1020)},   # Customer 27 (TW 2)
    28: {"time_window": (1021, 1260)},  # Customer 28 (TW 3)
    29: {"time_window": (781, 1020)},    # Customer 29 (TW 1)
    30: {"time_window": (781, 1020)},   # Customer 30 (TW 2)
    31: {"time_window": (781, 1020)},   # Customer 31 (TW 2)
    32: {"time_window": (480, 780)},    # Customer 32 (TW 1)
    33: {"time_window": (1021, 1260)},  # Customer 33 (TW 3)
    34: {"time_window": (781, 1020)},   # Customer 34 (TW 2)
    35: {"time_window": (781, 1020)},   # Customer 35 (TW 2)
    36: {"time_window": (781, 1020)},   # Customer 36 (TW 2)
    37: {"time_window": (480, 780)},    # Customer 37 (TW 1)
    38: {"time_window": (480, 780)},    # Customer 38 (TW 1)
    39: {"time_window": (781, 1020)},   # Customer 39 (TW 2)
    40: {"time_window": (480, 780)},    # Customer 40 (TW 1)
    41: {"time_window": (1021, 1260)},  # Customer 41 (TW 3)
    42: {"time_window": (781, 1020)},   # Customer 42 (TW 2)
    43: {"time_window": (480, 780)},    # Customer 43 (TW 1)
    44: {"time_window": (1021, 1260)},   # Customer 44 (TW 3)
    45: {"time_window": (1021, 1260)},  # Customer 45 (TW 3)
    46: {"time_window": (1021, 1260)},  # Customer 46 (TW 3)
    47: {"time_window": (1021, 1260)},  # Customer 47 (TW 3)
    48: {"time_window": (1021, 1260)},  # Customer 48 (TW 3)
    49: {"time_window": (480, 780)},   # Customer 49 (TW 2)
    50: {"time_window": (781, 1020)},  # Customer 50 (TW 3)
    51: {"time_window": (1021, 1260)},  # Customer 51 (TW 3)
    52: {"time_window": (1021, 1260)},  # Customer 52 (TW 3)
    53: {"time_window": (1021, 1260)},  # Customer 53 (TW 3)
    54: {"time_window": (781, 1020)},   # Customer 54 (TW 2)
    55: {"time_window": (1021, 1260)},  # Customer 55 (TW 3)
    56: {"time_window": (781, 1020)},   # Customer 56 (TW 2)
    57: {"time_window": (480, 780)},    # Customer 57 (TW 1)
    58: {"time_window": (781, 1020)},   # Customer 58 (TW 2)
    59: {"time_window": (781, 1020)},   # Customer 59 (TW 2)
    60: {"time_window": (1021, 1260)},  # Customer 60 (TW 3)
    61: {"time_window": (781, 1020)},   # Customer 61 (TW 2)
    62: {"time_window": (781, 1020)},   # Customer 62 (TW 2)
    63: {"time_window": (480, 780)},    # Customer 63 (TW 1)
    64: {"time_window": (1021, 1260)},  # Customer 64 (TW 3)
    65: {"time_window": (781, 1020)},   # Customer 65 (TW 2)
    66: {"time_window": (1021, 1260)},  # Customer 66 (TW 3)
    67: {"time_window": (781, 1020)},    # Customer 67 (TW 2)
    68: {"time_window": (480, 780)},    # Customer 68 (TW 1)
    69: {"time_window": (781, 1020)},   # Customer 69 (TW 2)
    70: {"time_window": (1021, 1260)},  # Customer 70 (TW 3)
    71: {"time_window": (781, 1020)},   # Customer 71 (TW 2)
    72: {"time_window": (480, 780)},   # Customer 72 (TW 3)
    73: {"time_window": (781, 1020)},   # Customer 73 (TW 2)
    74: {"time_window": (1021, 1260)},  # Customer 74 (TW 3)
    75: {"time_window": (1021, 1260)},  # Customer 75 (TW 3)
    76: {"time_window": (781, 1020)},   # Customer 76 (TW 2)
    77: {"time_window": (1021, 1260)},   # Customer 77 (TW 2)
    78: {"time_window": (480, 780)},    # Customer 78 (TW 1)
    79: {"time_window": (781, 1020)},   # Customer 79 (TW 2)
    80: {"time_window": (480, 780)},    # Customer 80 (TW 1)
    81: {"time_window": (781, 1020)},   # Customer 81 (TW 2)
    82: {"time_window": (781, 1020)},   # Customer 82 (TW 2)
    83: {"time_window": (1021, 1260)},  # Customer 83 (TW 3)
    84: {"time_window": (781, 1020)},   # Customer 84 (TW 2)
    85: {"time_window": (1021, 1260)},  # Customer 85 (TW 3)
    86: {"time_window": (1021, 1260)},  # Customer 86 (TW 3)
    87: {"time_window": (1021, 1260)},  # Customer 87 (TW 3)
```

```
    88: {"time_window": (1021, 1260)},  # Customer 88 (TW 3)
    89: {"time_window": (1021, 1260)},  # Customer 89 (TW 3)
    90: {"time_window": (781, 1020)},   # Customer 90 (TW 2)
    91: {"time_window": (480, 780)},    # Customer 91 (TW 1)
    92: {"time_window": (1021, 1260)},  # Customer 92 (TW 3)
    93: {"time_window": (1021, 1260)},   # Customer 93 (TW 2)
    94: {"time_window": (1021, 1260)},  # Customer 94 (TW 3)
    95: {"time_window": (781, 1020)},    # Customer 95 (TW 1)
    96: {"time_window": (781, 1020)},   # Customer 96 (TW 2)
    97: {"time_window": (1021, 1260)},  # Customer 97 (TW 3)
    98: {"time_window": (781, 1020)},   # Customer 98 (TW 2)
    99: {"time_window": (781, 1020)},   # Customer 99 (TW 2)
    100: {"time_window": (1021, 1260)}  # Customer 100 (TW 3)
}

# Fungsi hitung jarak Haversine
def distance(a, b, koreksi=True):
    lon1, lat1 = a
    lon2, lat2 = b
    lon1, lat1, lon2, lat2 = map(math.radians, [lon1, lat1, lon2, lat2])
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    A = math.sin(dlat/2)**2 + math.cos(lat1)*math.cos(lat2)*math.sin(dlon/2)**2
    C = 2 * math.atan2(math.sqrt(A), math.sqrt(1 - A))
    R = 6371
    haversine_km = R * C

    if koreksi:
        return haversine_km * 1.600
    else:
        return haversine_km

def travel_time(distance_km):
    return (distance_km / vehicle_speed) * 60

# Buat distance matrix (untuk akses cepat)
num_nodes = len(locations)
distance_matrix = [[0]*num_nodes for _ in range(num_nodes)]
for i in locations:
    for j in locations:
        distance_matrix[i][j] = distance(locations[i], locations[j])

# Fungsi penilaian rute (total distance + penalty)
def route_cost(route):
    total_dist = 0
    total_penalty = 0
    current_time = time_windows[depot_id]["time_window"][0]
    load = 0

    for i in range(len(route)-1):
        from_node = route[i]
        to_node = route[i+1]

        dist = distance_matrix[from_node][to_node]
        total_dist += dist

        travel_t = travel_time(dist)
        arrival = current_time + travel_t

        tw_start, tw_end = time_windows[to_node]["time_window"]
        if arrival < tw_start:
            arrival = tw_start

        # Hitung penalty jika terlambat
        late = max(0, arrival - tw_end)
        penalty = late * penalty_rate * cost_per_customer / 60

        total_penalty += penalty

        # Update load, asumsi hanya delivery demand
        load += pickup_delivery_demand[to_node]["delivery_demand"]
        if load > vehicle_capacity:
            # Beban overload, penalti sangat besar (untuk hindari solusi ini)
            total_penalty += 1e6

        current_time = arrival + service_time

    return total_dist + 5 * total_penalty

# NEAREST INSERTION untuk alokasi customer ke kendaraan (mirip kode kamu)
def solve_vrp_nearest_insertion():
    results = []
    unassigned = set(pickup_delivery_demand.keys()) - {depot_id}

    max_customers_per_vehicle = max(1, len(unassigned) // num_vehicles)
```

```
for vehicle_id in range(1, num_vehicles + 1):
    current_time = time_windows[depot_id]["time_window"][0]
    current_location = depot_id
    load = 0
    route = [depot_id]
    detail = []
    total_distance = 0
    total_penalty = 0
    customers_taken = 0

    while unassigned and customers_taken < max_customers_per_vehicle:
        feasible_customers = []
        for cust in unassigned:
            dist = distance_matrix[current_location][cust]
            travel_t = travel_time(dist)
            arrival = current_time + travel_t

            demand = pickup_delivery_demand[cust]['delivery_demand']
            tw_start, tw_end = time_windows[cust]["time_window"]

            if load + demand > vehicle_capacity:
                continue

            penalty = 0
            if arrival > tw_end:
                late_minutes = arrival - tw_end
                penalty = late_minutes * penalty_rate * cost_per_customer / 60

            feasible_customers.append((cust, arrival, dist, penalty))

        if not feasible_customers:
            break

        feasible_customers.sort(key=lambda x: x[1])
        cust, arrival, dist, penalty = feasible_customers[0]

        route.append(cust)
        load += pickup_delivery_demand[cust]['delivery_demand']
        total_distance += dist
        total_penalty += penalty
        detail.append({
            'customer_id': cust,
            'arrival_time': arrival,
            'time_window': time_windows[cust]["time_window"],
            'late': arrival > time_windows[cust]["time_window"][1],
            'penalty': int(penalty)
        })

        current_time = arrival + service_time
        current_location = cust
        unassigned.remove(cust)
        customers_taken += 1

    # Kembali ke depot
    if route[-1] != depot_id:
        dist_back = distance_matrix[current_location][depot_id]
        total_distance += dist_back
        current_time += travel_time(dist_back)
        route.append(depot_id)

        detail.append({
            "customer_id": depot_id,
            "arrival_time": current_time,
            "time_window": time_windows[depot_id]["time_window"],
            "late": False,
            "penalty": 0
        })

    results.append({
        'vehicle_id': vehicle_id,
        'route': route,
        'distance': total_distance,
        'time': current_time - time_windows[depot_id]["time_window"][0],
        'load': load,
        'penalty': total_penalty,
        'detail': detail
    })

if unassigned:
    print(f"Warning: {len(unassigned)} customer tidak terlayani!")

# ========================
# Alokasikan sisa customer ke rute yang sudah ada (reinsert)
# ========================
```

```
    if unassigned:
        print(f"\n🔁 Mencoba alokasikan {len(unassigned)} customer tersisa ke kendaraan yang ada...")

        remaining = list(unassigned)
        for cust in remaining:
            best_vehicle = None
            best_position = -1
            best_extra_distance = float('inf')

            for v in results:
                route = v['route']
                load = v['load']
                if load + pickup_delivery_demand[cust]['delivery_demand'] > vehicle_capacity:
                    continue

                for i in range(1, len(route)):  # posisi sisip di antara titik
                    new_route = route[:i] + [cust] + route[i:]
                    # Hitung jarak tambahan
                    try:
                        prev_node = new_route[i-1]
                        next_node = new_route[i+1]
                        added = distance_matrix[prev_node][cust] + distance_matrix[cust][next_node]
                        removed = distance_matrix[prev_node][next_node]
                        delta = added - removed
                    except:
                        continue

                    if delta < best_extra_distance:
                        best_extra_distance = delta
                        best_vehicle = v
                        best_position = i

            if best_vehicle and best_position != -1:
                best_vehicle['route'].insert(best_position, cust)
                best_vehicle['load'] += pickup_delivery_demand[cust]['delivery_demand']
                # Kamu bisa tambahkan update waktu/penalty/detail jika ingin
                unassigned.remove(cust)
            else:
                print(f"⚠️ Customer {cust} tetap tidak dapat dimasukkan ke rute manapun.")

    return results, unassigned

# ==================
# Algoritma Genetika untuk optimasi urutan rute kendaraan
# ==================

# (Removed duplicate and incomplete ga_optimize_route definition with indentation error)

# ==================
# Terapkan GA ke setiap kendaraan hasil Nearest Insertion
# ==================

import random
import pandas as pd

def ga_optimize_route(route, pop_size=700, generations=800, mutation_rate=0.2):
    customers = route[1:-1]
    if len(customers) <= 2:
        return route  # terlalu sedikit customer

    def create_individual():
        ind = customers[:]
        random.shuffle(ind)
        return ind

    def fitness(individual):
        r = [depot_id] + individual + [depot_id]
        return route_cost(r)

    def crossover(parent1, parent2):
        size = len(parent1)
        a, b = sorted(random.sample(range(size), 2))
        child = [None]*size
        child[a:b+1] = parent1[a:b+1]
        p2_idx = 0
        for i in range(size):
            if child[i] is None:
                while parent2[p2_idx] in child:
                    p2_idx += 1
                child[i] = parent2[p2_idx]
        return child

    def mutate(individual):
        if random.random() < mutation_rate:
            i, j = random.sample(range(len(individual)), 2)
```

```
            individual[i], individual[j] = individual[j], individual[i]

    # ======== INISIALISASI DAN LOGGING ==========
    population = []
    log_population = []
    log_generations = []
    log_operations = []

    for i in range(pop_size):
        ind = create_individual()
        log_population.append({
            "Generasi": 0,
            "Individu": i + 1,
            "Kromosom": str(ind),
            "Fitness": fitness(ind)
        })
        population.append(ind)

    # ========== EVOLUSI GENERASI ==========
    for gen in range(1, generations + 1):
        population.sort(key=fitness)
        best = fitness(population[0])
        log_generations.append({
            "Generasi": gen,
            "Fitness Terbaik": best
        })

        new_pop = population[:int(pop_size * 0.2)]  # elitism

        while len(new_pop) < pop_size:
            p1, p2 = random.sample(population[:int(pop_size * 0.5)], 2)
            child = crossover(p1, p2)
            mutate(child)

            log_operations.append({
                "Generasi": gen,
                "Parent1": str(p1),
                "Parent2": str(p2),
                "Child": str(child),
                "Fitness Child": fitness(child)
            })

            new_pop.append(child)

        population = new_pop

    # ========== HASIL AKHIR =============
    best_individual = min(population, key=fitness)
    best_route = [depot_id] + best_individual + [depot_id]

    # ========== EKSPOR KE EXCEL =========
    with pd.ExcelWriter("ga_log3.xlsx", engine='openpyxl') as writer:
        pd.DataFrame(log_population).to_excel(writer, sheet_name="Populasi Awal", index=False)
        pd.DataFrame(log_generations).to_excel(writer, sheet_name="Fitness Per Generasi", index=False)
        pd.DataFrame(log_operations).to_excel(writer, sheet_name="Crossover & Mutasi", index=False)

    return best_route

# ==================
# Fungsi untuk optimasi semua rute kendaraan dengan GA
def improve_routes_with_ga(vehicles):
    optimized_vehicles = []
    best_fitness = float('inf')
    for v in vehicles:
        best_route = ga_optimize_route(v['route'])
        # Update detail, distance, time, penalty sesuai best_route baru
        detail = []
        total_distance = 0
        total_penalty = 0
        current_time = time_windows[depot_id]["time_window"][0]
        load = 0
        for i in range(len(best_route)-1):
            from_node = best_route[i]
            to_node = best_route[i+1]
            dist = distance_matrix[from_node][to_node]
            total_distance += dist
            travel_t = travel_time(dist)
            arrival = current_time + travel_t
            tw_start, tw_end = time_windows[to_node]["time_window"]
            if arrival < tw_start:
                arrival = tw_start
            late = arrival > tw_end
            penalty = 0
            if late:
                late_minutes = arrival - tw_end
```

```
                penalty = late_minutes * penalty_rate * cost_per_customer / 60
            total_penalty += penalty
            load += pickup_delivery_demand[to_node]["delivery_demand"]
            detail.append({
                'customer_id': to_node,
                'arrival_time': arrival,
                'time_window': time_windows[to_node]["time_window"],
                'late': late,
                'penalty': int(penalty)
            })
            current_time = arrival + service_time
        optimized_vehicles.append({
            'vehicle_id': v['vehicle_id'],
            'route': best_route,
            'distance': total_distance,
            'time': current_time - time_windows[depot_id]["time_window"][0],
            'load': load,
            'penalty': total_penalty,
            'detail': detail
        })
        # Update best_fitness with the best route cost found
        route_fitness = route_cost(best_route)
        if route_fitness < best_fitness:
            best_fitness = route_fitness
    return optimized_vehicles, best_fitness

# ===================
# Main Run
vehicles_initial, unassigned = solve_vrp_nearest_insertion()

print("Hasil awal dari Nearest Insertion:")
for v in vehicles_initial:
    print(f"Vehicle {v['vehicle_id']} rute: {v['route']}, jarak: {v['distance']:.2f}, penalty:
{v['penalty']:.2f}")

vehicles_optimized, best_fitness = improve_routes_with_ga(vehicles_initial)

print("\nHasil setelah optimasi dengan GA:")
for v in vehicles_optimized:
    print(f"Vehicle {v['vehicle_id']} rute: {v['route']}, jarak: {v['distance']:.2f}, penalty:
{v['penalty']:.2f}")

print(f"\nFitness terbaik dari solusi GA: {best_fitness:.4f}")
# ===================
grand_total_cost = 0
grand_total_penalty = 0
grand_total_shipping_cost = 0

for v in vehicles_optimized:
    # Hitung jumlah customer yang terlambat (exclude depot)
    late_count = sum(1 for stop in v['detail'] if stop['late'] and stop['customer_id'] != depot_id)

    # Biaya penalty flat per customer terlambat
    penalty_cost = late_count * 2000

    # Ongkir dihitung berdasarkan jumlah customer (exclude depot)
    shipping_cost = 10000 * (len(v['route']) - 2)

    total_cost = shipping_cost + penalty_cost

    grand_total_cost += total_cost
    grand_total_penalty += penalty_cost
    grand_total_shipping_cost += shipping_cost

    print(f"\n=== Vehicle {v['vehicle_id']} ===")
    print(f"Total Distance  : {v['distance']:.2f} km")
    print(f"Total Time      : {v['time']:.2f} minutes")
    print(f"Total Load      : {v['load']} kg")
    print(f"Total Penalty   : Rp{penalty_cost:,}")
    print(f"Shipping Cost   : Rp{shipping_cost:,}")
    print(f"Total Cost      : Rp{int(total_cost):,}")
    print(f"Total Customers : {len(v['route']) - 2} (excluding depot)")
    print("Detail per customer:")
    for stop in v['detail']:
        status = "LATE" if stop['late'] else "ON TIME"
        print(f"  - Customer {stop['customer_id']:>3} | Arrival: {stop['arrival_time']:.1f} | "
              f"TW: {stop['time_window'][0]}-{stop['time_window'][1]} | "
              f"Status: {status} | Penalty: Rp{stop['penalty']:,}")

print(f"\n=== GRAND TOTAL COST: Rp{int(grand_total_cost):,} ===")
print(f"Total Penalty Only : Rp{int(grand_total_penalty):,}")
print(f"Total Shipping Cost: Rp{int(grand_total_shipping_cost):,}")

routes, unassigned_customers = solve_vrp_nearest_insertion()
print("Customer yang tidak terlayani (detail):")
```

```
for cust in unassigned:
    demand = pickup_delivery_demand[cust]['delivery_demand']
    tw_start, tw_end = time_windows[cust]['time_window']
    lon = locations[cust]['longitude']
    lat = locations[cust]['latitude']
    print(f"- {cust} | Demand: {demand} kg | TW: {tw_start}-{tw_end} | Lokasi: ({lat}, {lon})")

# Visualisasi rute tiap kendaraan
import matplotlib.pyplot as plt

plt.figure(figsize=(10,8))
colors = ['r', 'g', 'b', 'c', 'm', 'y', 'k']
for i, v in enumerate(vehicles_optimized):
    route_lons = [locations[node][0] for node in v['route']]
    route_lats = [locations[node][1] for node in v['route']]
    color = colors[i % len(colors)]

    plt.plot(route_lons, route_lats, marker='o', color=color, label=f'Vehicle {v["vehicle_id"]}')
    for node in v['route']:
        if node == depot_id:
            plt.scatter(locations[node][0], locations[node][1], c='k', marker='s', s=100)
            plt.text(locations[node][0], locations[node][1]+0.0005, 'Depot', fontsize=9, ha='center')
        else:
            plt.text(locations[node][0], locations[node][1]+0.0005, str(node), fontsize=8, ha='center')

plt.title('Vehicle Routing Problem Solution')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.legend()
plt.grid(True)
plt.show()
```