

## BAB II LANDASAN TEORI

### 2.1. Distribusi

Distribusi merupakan bagian penting dalam rantai pasokan yang mencakup pengiriman barang dari produsen hingga konsumen akhir. Menurut Arianto (2020), distribusi berfungsi untuk mengoptimalkan aliran produk, mengurangi biaya, dan meningkatkan kepuasan pelanggan. Dalam logistik, distribusi yang efisien dapat meningkatkan daya saing perusahaan dengan menekan biaya dan mempercepat pengiriman produk (Tulong, 2018). Namun, perusahaan sering menghadapi tantangan dalam mengelola distribusi, seperti kendala geografis dan keterbatasan infrastruktur transportasi (Sahara & Saputra, 2021). Oleh karena itu, penerapan teknologi dan sistem perencanaan distribusi yang tepat sangat diperlukan untuk mencapai efisiensi dalam pengiriman, seperti penggunaan *Lean Distribution* untuk mengurangi pemborosan dalam rantai pasokan (Arianto, 2020).

Peran teknologi dalam distribusi semakin meningkat, terutama dalam penggunaan aplikasi seperti *Google Maps API* untuk mengoptimalkan rute pengiriman dan meminimalkan biaya transportasi (Scribd, 2020). Selain itu, koordinasi yang baik antar berbagai pihak dalam rantai distribusi sangat penting untuk memastikan kelancaran pengiriman (Tumbel & Palandeng, 2020). Hal ini juga mendukung penelitian yang menunjukkan bahwa sistem distribusi yang terorganisir dengan baik dapat meningkatkan kinerja perusahaan dan kepuasan pelanggan (Tulong, 2018). Oleh karena itu, perbaikan dalam manajemen distribusi adalah langkah krusial untuk mencapai efisiensi dan mempertahankan daya saing perusahaan dalam pasar yang kompetitif.

### 2.2. Vehicle Routing Problem

*Vehicle Routing Problem* (VRP) adalah masalah optimisasi yang bertujuan untuk menentukan rute optimal bagi armada kendaraan dalam melayani sejumlah pelanggan dengan cara yang efisien. VRP pertama kali diperkenalkan oleh Dantzig dan Ramser pada tahun 1959 untuk mendistribusikan bahan bakar dari depot ke berbagai stasiun pengisian (Dantzig & Ramser, 1959). Secara umum, tujuan dari VRP adalah menemukan serangkaian rute kendaraan terbaik, yang dimulai dan

diakhiri di satu depot, untuk memenuhi semua permintaan pelanggan dengan biaya serendah mungkin (Alif et al., 2022). Seiring waktu, VRP berkembang menjadi berbagai varian untuk menangani kompleksitas dunia nyata, seperti *Capacitated VRP* (CVRP) yang mempertimbangkan kapasitas kendaraan, *VRP with Time Windows* (VRPTW) yang membatasi waktu pelayanan, dan *Stochastic VRP* (SVRP) yang memperhitungkan ketidakpastian dalam permintaan atau waktu tempuh (Eksioglu et al., 2009). Dalam konteks distribusi gas industri, VRP menjadi sangat relevan karena melibatkan pengantaran dan pengambilan barang dalam satu perjalanan dengan kendaraan yang sama, yang dikenal sebagai *Vehicle Routing Problem with Pickup and Delivery* (VRP-PD).

Penyelesaian VRP sering kali melibatkan penggunaan algoritma *metaheuristik*, seperti *Genetic Algorithm* (GA), *Simulated Annealing* (SA), dan *Ant Colony Optimization* (ACO), yang dirancang untuk mencari solusi optimal dalam ruang pencarian yang besar dan kompleks (Sidiropoulos et al., 2020). Penelitian oleh Eksioglu et al. (2009) menunjukkan bahwa penerapan algoritma *metaheuristik* dapat menghasilkan solusi yang lebih baik dalam hal penghematan biaya dan waktu tempuh dibandingkan dengan metode eksak, terutama untuk masalah VRP berskala besar. Selain itu, dengan kemajuan teknologi informasi, penggunaan aplikasi seperti *Google Maps API* memungkinkan perencanaan rute yang lebih efisien dengan mempertimbangkan kondisi lalu lintas secara *real-time*, sehingga meningkatkan efisiensi distribusi barang dan kepuasan pelanggan (Sidiropoulos et al., 2020).

### **2.3. Vehicle Routing Problem with Pickup and Delivery**

*Vehicle Routing Problem with Pickup and Delivery* (VRP-PD) adalah varian dari masalah optimisasi rute kendaraan yang mengharuskan kendaraan untuk mengantarkan dan mengambil barang dalam satu perjalanan, dengan mempertimbangkan kapasitas kendaraan dan efisiensi waktu. Dalam VRP-PD, setiap kendaraan harus mengunjungi lokasi pelanggan untuk menyelesaikan kedua tugas tersebut secara simultan, yang menambah kompleksitas dalam perencanaan rute (Jaque Pirabán, 2008). Penelitian oleh Lüer-Villagra et al. (2009) menunjukkan bahwa VRP-PD sering kali diselesaikan menggunakan metode heuristik dan *metaheuristik*, seperti *Genetic Algorithm* (GA), *Simulated Annealing* (SA), dan *Ant Colony Optimization* (ACO), yang dirancang untuk mencari solusi optimal dalam

ruang pencarian yang besar dan kompleks. Selain itu, penelitian oleh Lürer-illagra et al. (2009) juga menyoroti pentingnya mempertimbangkan *faktor-faktor* seperti waktu layanan, kapasitas kendaraan, dan waktu tempuh dalam menyelesaikan masalah VRP-PD, agar dapat mencapai efisiensi dalam distribusi barang.

### **2.3.1. With Backhaul (VRPB)**

*Vehicle Routing Problem With Backhaul* (VRPB) adalah varian dari *Vehicle Routing Problem with Pickup and Delivery* (VRP-PD), di mana kendaraan tidak hanya mengantarkan barang kepada pelanggan, tetapi juga mengumpulkan barang yang kosong atau tidak terpakai dari pelanggan lain dalam satu perjalanan. Tantangan utama dalam VRPB adalah mengoptimalkan rute sehingga kendaraan dapat menyelesaikan kedua tugas, yaitu pengantaran dan pengambilan barang, dengan efisiensi biaya dan waktu, sambil memperhatikan kapasitas kendaraan dan kondisi lainnya (Chen et al., 2014). Penelitian oleh Sidiropoulos et al. (2020) menunjukkan bahwa dalam VRPB, kendaraan sering kali harus merencanakan rute yang lebih kompleks, terutama ketika pengambilan barang dilakukan setelah pengantaran, yang mempengaruhi urutan kunjungan dan pemilihan rute yang optimal. Oleh karena itu, metode heuristik dan *metaheuristik* seperti *Genetic Algorithm* (GA), *Tabu Search*, dan *Ant Colony Optimization* (ACO) sering diterapkan untuk mencari solusi yang efisien dan efektif dalam menangani masalah VRPB ini.

### **2.3.2. Mixed Pickups and Deliveries (VRPMPD)**

*Mixed Pickups and Deliveries* (VRPMPD) adalah varian dari *Vehicle Routing Problem with Pickup and Delivery* (VRP-PD) yang melibatkan pengantaran dan pengambilan barang dalam satu rute kendaraan, tetapi dengan pelanggan yang terpisah antara titik pickup dan delivery. Dalam VRPMPD, beberapa pelanggan hanya membutuhkan pengantaran barang, sementara yang lain hanya membutuhkan pengambilan barang (Moccia et al., 2018). Masalah ini menjadi lebih kompleks karena kendaraan harus merencanakan rute yang efisien dengan memperhatikan jadwal pengambilan dan pengantaran yang berbeda, serta kapasitas kendaraan yang terbatas. Penelitian oleh Salazar-González et al. (2016) menunjukkan bahwa untuk memecahkan masalah VRPMPD, teknik *metaheuristik* seperti *Genetic Algorithm* (GA) dan *Tabu Search* sering digunakan karena

kemampuan mereka untuk mengeksplorasi solusi dalam ruang pencarian yang sangat besar dan kompleks, yang sulit diselesaikan dengan metode eksak.

### 2.3.3. Simultaneous Pickup and Delivery (SPD)

*Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD)* adalah varian *VRP* yang secara spesifik memungkinkan kendaraan untuk melakukan pengiriman (*delivery*) dan pengambilan (*pick-up*) barang secara bersamaan saat mengunjungi satu pelanggan (*Garside & Cahyanti, 2018*). Model ini mengharuskan kendaraan untuk mengantarkan dan mengambil barang dari pelanggan dalam satu rute yang efisien, dengan memperhatikan kapasitas kendaraan, waktu layanan, dan urutan kunjungan yang optimal (*Gendreau et al., 2006*). Penelitian oleh *Solomon & Desrosiers (2005)* menunjukkan bahwa SPD sering kali membutuhkan pendekatan optimisasi yang lebih kompleks karena penggabungan dua tugas dalam satu perjalanan, yang meningkatkan kesulitan dalam merencanakan rute. Dalam penyelesaian masalah ini, algoritma *metaheuristik* seperti *Genetic Algorithm (GA)* dan *Ant Colony Optimization (ACO)* sering digunakan untuk mengeksplorasi ruang solusi yang besar dan menemukan rute yang paling efisien dalam mengurangi biaya dan waktu tempuh (*Gendreau et al., 2006*).

### 2.4. Model Vehicle Routing Problem with Pickup and Delivery Simultaneous Pickup And Delivery

*Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD)* adalah sebuah model optimisasi matematis yang dirancang untuk menentukan serangkaian rute dengan total biaya atau jarak minimum untuk armada kendaraan. Model ini berangkat dari satu depot untuk melayani sejumlah pelanggan yang tersebar, di mana setiap pelanggan membutuhkan pengantaran (*delivery*) sejumlah barang sekaligus pengambilan (*pickup*) sejumlah barang lain pada saat yang bersamaan dalam satu kali kunjungan (*Dethloff, 2001*). Model ini menjadi sangat relevan dalam sistem logistik modern, seperti distribusi minuman atau pengumpulan barang bekas, di mana efisiensi rute secara langsung berdampak pada biaya operasional. Model matematis untuk VRPSPD, seperti yang diformulasikan

oleh Tang Montané dan Galvão (2006), dapat dijelaskan melalui notasi, fungsi tujuan, dan serangkaian batasan sebagai berikut:

$V$  : Set dari pelanggan

$V_0$  : Set dari pelanggan ditambah agen :  $V_0 = V \cup \{0\}$

$\bar{k}$  : Jumlah kendaraan

$Q$  : Kapasitas kendaraan

$n$  : Jumlah pelanggan:  $n = |V|$

$c_{ij}$  : Jarak antara pelanggan  $i$  dan  $j$

$d_j$  : Jumlah pengiriman ke pelanggan  $j$ ;  $j = 1, 2, 3, \dots, n$

$p_j$  : Jumlah pengambilan ke pelanggan  $j$ ;  $j = 1, 2, 3, \dots, n$

$MD$  : Jarak maksimum yang diizinkan untuk setiap rute

$x_{ijk}$   $\begin{cases} 1, & \text{jika kendaraan } k \text{ akan mengunjungi pelanggan } j \text{ setelah pelanggan } i \\ 0, & \text{jika kendaraan } k \text{ tidak akan mengunjungi pelanggan } j \text{ setelah pelanggan } i \end{cases}$

$y_{ij}$  : Jumlah pengambilan sampai pelanggan  $i$  dan akan ditransportasikan melalui jalur  $(i, j)$ .

$Z_{ij}$  : Jumlah pengiriman yang akan dikirimkan setelah pelanggan  $i$  dan akan ditransportasikan melalui jalur  $(i, j)$ .

Berdasarkan prosedur perhitungan jarak tempuh masalah VRPSPD dirumuskan sebagai berikut :

$$\text{Min} \sum_{k=1}^{\bar{k}} \sum_{i=0}^N \sum_{j=0}^N c_{ij} x_{ijk} \quad (2.1)$$

$$\sum_{i=0}^N \sum_{k=1}^{\bar{k}} x_{ijk} = 1, \quad j = 1, \dots, N \quad (2.2)$$

$$\sum_{i=0}^N x_{ijk} - \sum_{i=0}^N x_{jik} = 0, \quad j = 0, 1, \dots, N; k = 1, \dots, \bar{k} \quad (2.3)$$

$$\sum_{j=1}^N x_{0jk} \leq 1, \quad k = 1, \dots, \bar{k} \quad (2.4)$$

$$\sum_{i=0}^N y_{ji} - \sum_{i=0}^N y_{ij} = p_j, \quad \forall j \neq 0 \quad (2.5)$$

$$\sum_{i=0}^N z_{ij} - \sum_{i=0}^N z_{ji} = d_j, \quad \forall j \neq 0 \quad (2.6)$$

$$y_{ij} + z_{ij} \leq Q \sum_{k=1}^{\bar{k}} x_{ijk}, \quad i, j = 0, 1, \dots, N \quad (2.7)$$

$$\sum_{i=0}^N \sum_{j=0}^N c_{ij} x_{ijk} \leq MD, \quad k = 1, \dots, \bar{k} \quad (2.8)$$

$$x_{ijk} \in \{0, 1\} \quad i, j = 0, 1, \dots, N; k = 1, \dots, \bar{k} \quad (2.9)$$

$$y_{ij} \geq 0 \quad i, j = 0, 1, \dots, n \quad (2.10)$$

$$z_{ij} \geq 0, \quad i, j = 0, 1, \dots, n \quad (2.11)$$

Fungsi tujuan pada Persamaan (2.1) yaitu meminimalkan total jarak tempuh. Kendala pada Persamaan (2.2) memastikan setiap pelanggan dikunjungi tepat satu kali. Persamaan (2.3) menjamin alur kontinuitas rute kendaraan. Persamaan (2.4) membatasi jumlah kendaraan yang digunakan. Persamaan (2.5) dan (2.6) adalah rumusan alur untuk permintaan penjemputan (*pickup*) dan pengiriman (*delivery*). Persamaan (2.7) memastikan total muatan tidak melebihi kapasitas kendaraan. Persamaan (2.8) adalah kendala untuk jarak tempuh maksimum setiap rute. Terakhir, Persamaan (2.9), (2.10), dan (2.11) mendefinisikan tipe variabel yang digunakan dalam model.

Walaupun Persamaan (2.7) menetapkan batas kapasitas secara teoretis, implementasinya dalam Algoritma Genetika (GA) memerlukan metode verifikasi praktis. Untuk tujuan ini, digunakan model Muatan Dinamis (*Dynamic Load Flow*) yang memantau muatan kendaraan secara sekuensial pada setiap segmen rute. Proses ini merupakan dasar utama untuk pengecekan kelayakan (*feasibility check*) rute dalam algoritma.

Perhitungan muatan operasional ini didefinisikan dalam dua tahapan:

1. Muatan Awal Kendaraan: Muatan yang dibawa kendaraan saat meninggalkan depot (Node 0) ditetapkan setara dengan total permintaan *delivery* dari seluruh pelanggan yang dilayani dalam rute tersebut. Persamaan untuk menghitung muatan awal adalah sebagai berikut :

$$l_o = \sum_{j=0}^N d_j \quad (2.12)$$

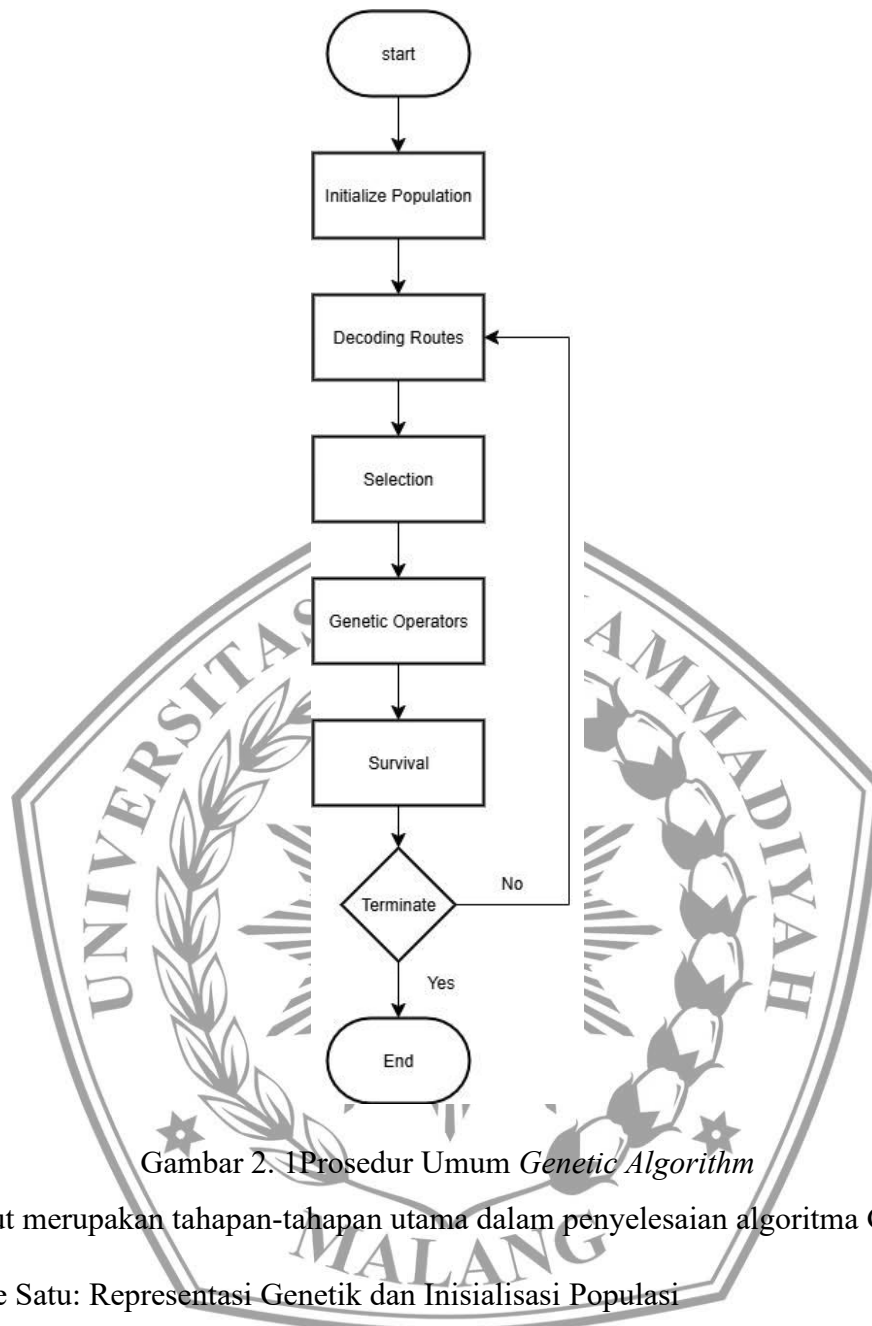
2. Muatan Sekuensial: Muatan kendaraan saat meninggalkan pelanggan ( $l_i$ ) dihitung dari muatan yang tiba di pelanggan tersebut, dikurangi jumlah pengiriman ( $d_j$ ) yang diserahkan, dan ditambah jumlah *pickup* ( $p_j$ ) yang diambil, menghasilkan persamaan :

$$l_j = l_i + p_j - d_j \quad (2.13)$$

Dengan demikian, Muatan Dinamis ini bertindak sebagai mekanisme implementasi langsung dari Kendala Kapasitas (2.7). Logika perhitungan sekuensial ini memastikan kelayakan rute terjamin hanya jika muatan di setiap titik rute tidak pernah melebihi kapasitas kendaraan .

### 2.5. Genetic Algorithm

Metode yang akan diusulkan dalam skripsi ini adalah *Genetic Algorithm* (GA). GA adalah pendekatan *metaheuristik* yang efisien untuk menyelesaikan masalah optimisasi kombinatorial yang kompleks seperti VRPSPD. Algoritma ini terinspirasi dari teori evolusi alam, di mana solusi-solusi terbaik (individu) akan bertahan dan bereproduksi untuk menghasilkan generasi solusi baru yang lebih baik. Seperti yang dijelaskan oleh Tasan dan Gen (2010), pendekatan GA yang efektif harus memiliki representasi genetik yang baik, fungsi *fitness* yang sesuai, serta operator genetik yang tepat. Prosedur umum dari pendekatan GA yang diusulkan ditunjukkan pada Gambar 2.1.



Gambar 2. 1Prosedur Umum *Genetic Algorithm*

Berikut merupakan tahapan-tahapan utama dalam penyelesaian algoritma GA:

a. Fase Satu: Representasi Genetik dan Inisialisasi Populasi

Tahap awal dalam GA adalah merepresentasikan solusi ke dalam bentuk kromosom dan membangkitkan populasi awal.

1. Representasi Genetik: Solusi VRPSPD direpresentasikan menggunakan representasi permutasi. Sebuah kromosom adalah sebuah urutan angka (gen) yang melambangkan pelanggan, di mana panjang kromosom ditentukan oleh jumlah total pelanggan yang harus dilayani,

$$X = [x_1, x_2, \dots, x_n] \quad (2.14)$$

2. Inisialisasi Populasi: Populasi awal, yang terdiri dari sejumlah kromosom, dibangkitkan menggunakan metode permutasi acak (*random permutation*) untuk menciptakan keragaman solusi di awal proses. Ukuran populasi ditentukan sebagai parameter  $N$ .

b. Fase Dua: Evaluasi *Fitness* dan Dekode Rute

Setelah populasi terbentuk, kualitas setiap kromosom dievaluasi.

1. Fungsi *Fitness*: Kualitas setiap solusi (kromosom) diukur menggunakan fungsi *fitness*. Untuk masalah ini, nilai *fitness* dihitung berdasarkan total jarak tempuh. Kromosom dengan total jarak lebih rendah memiliki nilai *fitness* yang lebih baik dan kemungkinan lebih besar untuk bertahan. Fungsi *fitness* diambil dari fungsi tujuan pada model *Vehicle Routing Problem with Simultaneous Pickup and Delivery* seperti pada persamaan 2.1 dengan Kendal dari persamaan 2.2 untuk setiap pelanggan dikunjungi tepat satu kali, persamaan 2.3 untuk keseimbangan aliran, persamaan 2.13 untuk kapasitas kendaraan. Dan jika terdapat pelanggaran kapasitas atau jarak maksimum MD, maka fungsi *fitness* dihitung dengan menambahkan penalty sebagai persamaan baru seperti pada persamaan 2.13 sebagai berikut :

$$Fitness = \frac{1}{Z + \lambda \times Penalty} \quad (2.15)$$

Dengan  $\lambda$  Adalah koefisien Penalti.

2. Dekode Rute dan Penalti: Kromosom yang berisi urutan pelanggan didekode menjadi rute-rute kendaraan. Proses ini memastikan bahwa total permintaan (*pickup* dan *delivery*) dalam satu rute tidak melebihi kapasitas kendaraan (*weak feasibility*). Namun, ada kemungkinan muatan kendaraan melebihi kapasitas di tengah perjalanan (*strong infeasibility*). Untuk menangani hal ini, sebuah biaya penalti ditambahkan ke nilai *fitness* setiap kali terjadi pelanggaran kapasitas untuk mendorong algoritma mencari solusi yang sepenuhnya layak.

c. Fase Tiga: Seleksi

Pada tahap ini, individu-individu terbaik dipilih untuk menjadi "induk" bagi generasi selanjutnya. Metode yang digunakan adalah *Roulette Wheel Selection* (RWS). Dalam metode ini, probabilitas sebuah kromosom untuk terpilih berbanding lurus dengan nilai *fitness*-nya, sehingga individu yang lebih baik memiliki kesempatan lebih besar untuk bereproduksi. Probabilitas sebuah kromosom untuk terpilih berbanding lurus dengan nilai *fitness*-nya, sesuai persamaan baru berikut:

$$P_i = \frac{F_i}{\sum_{k=1}^N F_k} \quad (2.16)$$

d. Fase Empat: Operator Genetik (Crossover dan Mutasi)

Setelah seleksi, operator genetik diterapkan untuk menciptakan generasi baru.

1. Crossover (Pindah Silang): Operator utama ini mensimulasikan reproduksi dengan menggabungkan dua kromosom induk untuk menghasilkan keturunan baru (*offspring*). Metode yang digunakan adalah *Partial-Mapped Crossover* (PMX). PMX menukar sebagian segmen gen antara dua induk dan kemudian memperbaiki kromosom anak agar tetap menjadi permutasi yang valid.
2. Mutasi: Operator ini membuat perubahan kecil dan acak pada satu kromosom untuk menjaga keragaman genetik dalam populasi dan mencegah solusi terjebak pada optimal lokal. Metode yang digunakan adalah Swap Mutation, di mana posisi dua gen (pelanggan) dalam satu kromosom ditukar secara acak:

$$X' = \text{Swap}(x_i, x_j) \quad (2.17)$$

Proses ini (evaluasi, seleksi, dan operator genetik) diulang terus-menerus hingga kriteria pemberhentian (misalnya, jumlah generasi maksimum) tercapai, dan solusi terbaik yang ditemukan selama proses akan menjadi output akhir. Pemilihan GA sebagai metode penyelesaian VRPSPD sangat tepat karena kemampuannya untuk menemukan solusi optimal pada masalah *routing* yang sangat kompleks dan telah terbukti kuat dalam literatur VRP (Khoidir & Garside, 2022).

## 2.6. Penelitian Terdahulu

Sebagian besar penelitian VRPSPD berfokus pada meminimalkan jarak tempuh, kemudian berkembang ke tujuan yang lebih kompleks seperti biaya, waktu, jumlah kendaraan, dan dampak lingkungan. Metodenya juga berevolusi dari heuristik sederhana, *metaheuristik* klasik, hingga algoritma hybrid, bio-inspired, dan terbaru berbasis AI/ML. Arah riset bergerak dari optimasi tunggal menuju multi-objektif dengan pendekatan yang semakin cerdas dan adaptif. Penelitian yang berkaitan dengan *Vehicle Routing Problem with Simultaneous Pickup and Delivery* ditunjukkan pada Tabel 2.1.

Tabel 2. 1 Penelitian Terdahulu

No	Nama Penulis (Tahun)	Varian Penelitian	Fungsi/Tujuan	Pendekatan/Metode
1	Dethloff, J. (2001)	VRPSPD	Meminimalkan total jarak tempuh	<i>Heuristik berbasis penyisipan (Insertion-based Heuristic)</i>
2	Bianchessi, N., & Righini, G. (2007)	CVRPSPDTW	Meminimalkan total jarak tempuh	<i>Constructive Algorithm dan Tabu Search</i>
3	Ai, T. J., & Kachitvichyanukul, V. (2009)	CVRPSPDTW	Meminimalkan total jarak tempuh	<i>Particle Swarm Optimization (PSO)</i>
4	Zachariadis, E. E., et al. (2009)	VRPSPD	Meminimalkan total jarak tempuh	<i>Algoritma Hibrida: Tabu Search dan Guided Local Search</i>
5	Sze, S. N., et al. (2020)	VRPSPD	Meminimalkan total jarak tempuh & memaksimalkan kapasitas kendaraan	<i>Heuristik inisialisasi dan Variable Neighborhood Search (VNS)</i>
6	Goksal, T., et al. (2013)	VRPSPD	Meminimalkan total jarak tempuh	<i>Genetic Algorithm (GA)</i>

7	Tasan, A. S., & Tasan, S. (2012)	1-PDTSP ( <i>One-Commodity Pickup-and-Delivery Travelling Salesman Problem</i> )	Meminimalkan total biaya (tetap & variabel)	<i>GA dengan Savings Algorithm</i>
8	Wassan, N. A., et al. (2008)	1-PDTSP ( <i>One-Commodity Pickup-and-Delivery Travelling Salesman Problem</i> )	Meminimalkan total jarak tempuh	<i>Reactive Tabu Search (RTS)</i>
9	Subaşı, E., & Tasan, A. S. (2016)	PDPTW ( <i>Pickup and Delivery Problem with Time Windows</i> )	Meminimalkan total biaya perjalanan	<i>Ant Colony Optimization (ACO) dengan Savings Algorithm</i>
10	Avcı, M., & Topaloglu, S. (2017)	VRPSPD	Meminimalkan total jarak tempuh	<i>Adaptive Local Search (ALS)</i>
11	Zachariadis, E. E., & Kiranoudis, C. T. (2010)	VRPSPDTW	Meminimalkan total jarak tempuh	<i>Local Search dengan ejection-chain</i>
12	Catay, B. (2009)	VRPSPD	Meminimalkan total jarak tempuh	<i>Hybrid: Ant Colony System (ACS) + Savings Heuristic</i>
13	Dellaert, N., et al. (2018)	MDVRPSPD ( <i>Multi-Depot VRPSPD</i> )	Meminimalkan total biaya (perjalanan & handling)	<i>Branch-and-Price (Eksak)</i>
14	Wang, H., & Chen, Y. (2012)	VRPSPDTW	Meminimalkan total waktu perjalanan	<i>GA dimodifikasi untuk waktu tempuh variabel</i>
15	Olgun, B., & Yilmaz, A. S. (2020)	PVRPSDP ( <i>Period VRPSDP</i> )	Meminimalkan total jarak tempuh	<i>Parallel Simulated Annealing (SA)</i>

16	Liu, R., et al. (2019)	VRPSPDTW	Meminimalkan total biaya (jarak & jumlah kendaraan)	<i>Hybrid: SA + Greedy Heuristic</i>
17	Chen, J. F., & Wu, T. H. (2006)	VRPSPD	Meminimalkan total jarak tempuh	<i>Hybrid: Tabu Search (TS) + Record-to-Record Travel</i>
18	Ropke, S., & Pisinger, D. (2006)	CVRPSPD	Meminimalkan total biaya (jarak, jumlah kendaraan, penalti waktu)	<i>Adaptive Large Neighborhood Search (ALNS)</i>
19	Koç, Ç., & Laporte, G. (2018)	CVRPSPD	Meminimalkan total jarak tempuh	<i>Branch-and-Cut (Eksak)</i>
20	Zhang, T., et al. (2015)	MDVRPSDP ( <i>Multi-Depot VRPSDP</i> )	Meminimalkan jumlah kendaraan & jarak tempuh	<i>Artificial Bee Colony (ABC)</i>
21	Wang, Y., et al. (2015)	VRPSPD	Meminimalkan jumlah kendaraan, total waktu perjalanan, & waktu tunggu	<i>Multi-Objective Particle Swarm Optimization (MOPSO)</i>
22	Soleimani, H., et al. (2017)	VRPSPDTW	Meminimalkan biaya & emisi	<i>Imperialist Competitive Algorithm (ICA)</i>
23	Keskin, M., & Çatay, B. (2018)	VRPB ( <i>Vehicle Routing Problem with Backhauls</i> )	Meminimalkan total jarak tempuh	<i>Matheuristic (VNS + Pemrograman Matematis)</i>
24	Lee, C. (2021)	G-VRPSPD ( <i>Green VRPSDP</i> )	Meminimalkan total biaya perjalanan	<i>Hybrid GA + Local Search (multi-depot)</i>

25	Li, Y., et al. (2022)	VRPSPD	Meminimalkan total jarak tempuh	<i>Whale Optimization Algorithm (WOA)</i>
26	Lu, C., & Zhang, J. (2023)	VRPSPD	Meminimalkan total jarak tempuh	<i>Deep Reinforcement Learning (DRL)</i>
27	Nepomuceno, N., & Viana, A. (2020)	VRPSPD	Mengkaji, menganalisis, & mengklasifikasikan literatur VRPSPD	Tinjauan Literatur Sistematis

Berdasarkan tinjauan pustaka yang disajikan pada Tabel 2.1, dapat disimpulkan bahwa penelitian mengenai *Vehicle Routing Problem with Simultaneous Pickup and Delivery* (VRPSPD) telah berkembang pesat. Sebagian besar penelitian berfokus pada tujuan meminimalkan total jarak tempuh atau biaya, dengan evolusi metode dari heuristik klasik seperti *insertion-based* hingga *metaheuristik* yang lebih canggih seperti *Tabu Search*, *Particle Swarm Optimization*, dan berbagai algoritma hibrida. Meskipun demikian, penerapan *Genetic Algorithm* (GA) yang secara spesifik diadaptasi untuk studi kasus distribusi LPG 3kg di Indonesia, dengan karakteristik permintaan dan kondisi geografis yang unik, masih belum banyak dieksplorasi. Oleh karena itu, penelitian ini akan mengisi celah tersebut dengan mengimplementasikan *Genetic Algorithm* yang disesuaikan, menggunakan *operator Partial-Mapped Crossover* (PMX) dan mekanisme penalti, untuk memberikan solusi rute optimal yang efisien dan aplikatif bagi Pangkalan LPG 3kg Achmad Nasoha. Penelitian ini diharapkan dapat menunjukkan efektivitas GA dalam menangani permasalahan VRPSPD pada konteks nyata di lapangan.