

Pengembangan Optimasi Rute Menggunakan *Deep Reinforcement Learning* Pada Model *Dynamic Vehicle Routing Problem With Time Windows* (DVRPTW)

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Machine Learning (ML) adalah cabang dari kecerdasan artifisial yang memanfaatkan teknik statistik dan algoritma agar dapat belajar dan membuat keputusan atau prediksi berdasarkan data. Dengan menggunakan algoritma ML, komputer dapat meningkatkan efisiensi dalam melakukan berbagai jenis pekerjaan tanpa harus di program secara eksplisit untuk setiap tugasnya (Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, A. M., & Talbi, E. G, 2022). ML memanfaatkan berbagai teknik dari statistik, teori probabilitas, matematika, dan ilmu komputer untuk membangun model dari dataset yang ada. *Machine learning* adalah metode yang secara otomatis menganalisis data untuk mendapatkan aturan, kemudian menggunakan aturan ini untuk memprediksi data yang tidak diketahui. Algoritma dan metode statistik diterapkan untuk memberikan komputer kemampuan untuk "belajar" dari data dan meningkatkan kinerjanya dalam memecahkan masalah tanpa harus memprogram secara eksplisit untuk setiap masalah (Ni, Qiuping & Tang, Yuanxiang, 2023) .

Machine learning dibagi menjadi tiga kategori *supervised learning*, *unsupervised learning* dan *reinforcement learning* (Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, A. M., & Talbi, E. G, 2022). *Reinforcement learning* merupakan bagian dari *machine learning* yang membedakan dari *supervised learning* dan *unsupervised learning* yaitu pada *reinforcement learning* dengan *trial and error* selama interaksi langsung dengan lingkungan sekitar (Panzer & Bender, 2022). *Reinforcement learning* (RL) tidak memerlukan sinyal yang diawasi untuk belajar, RL bergantung pada sinyal umpan balik dari individu (*agent*) di lingkungannya. Umpan balik ini mengoreksi keadaan dan tindakan agen, sehingga agen secara bertahap dapat mempelajari cara memaksimalkan hadiah (*reward*) dengan cara memaksimalkan nilai *cumulative*

reward (hadiah atau imbalan yang dikumpulkan secara kumulatif) dan mencapai kemampuan belajar mandiri yang kuat (Ni & Tang, 2023).

Algoritma RL dapat dibagi menjadi dua kategori yaitu pembelajaran berbasis model dan pembelajaran bebas model (Mousavi, Seyed Sajad, Howley, Enda & Schukat, Michael, 2018). Pembelajaran berbasis model memiliki pengetahuan sebelumnya tentang lingkungan yang dapat dioptimalkan terlebih dahulu. Pembelajaran bebas model lebih rendah daripada yang pertama dalam hal kecepatan pelatihan, tetapi lebih mudah diimplementasikan dan dapat dengan cepat menyesuaikan diri dengan keadaan yang lebih baik dalam skenario nyata. Penerapan *reinforcement learning* telah menunjukkan hasil yang signifikan pada berbagai bidang seperti robotika (Jens Kober, J Andrew Bagnell, Jan Peters, 2013), permainan (Silver, D., Huang, A., Maddison, C. et al. 2016), kesehatan (Liu Siqu, See Kay Choong, Ngiam Kee Yuan, Celi Leo Anthony, Sun Xingzhi, Feng Mengling, 2020) dan distribusi logistik (He Zhenhua, Chen Liang, Liu Bin, 2024).

Perkembangan teknik *machine learning*, khususnya dalam *deep learning*, telah memungkinkan penggunaan arsitektur *neural network* yang lebih kompleks untuk memecahkan berbagai masalah yang sulit diselesaikan dengan pendekatan konvensional. Salah satu teknik yang telah menunjukkan hasil signifikan adalah penggunaan mekanisme perhatian ganda atau *multi-header attention*. *Multi-header attention* adalah mekanisme di mana lapisan perhatian direplikasi beberapa kali untuk memungkinkan model fokus pada bagian berbeda dari urutan masukan secara bersamaan (Cordonnier, Loukas & Jaggi, 2020). Mekanisme ini pertama kali diperkenalkan dalam konteks model Transformer (Vaswani et al., 2017), dan telah digunakan secara luas dalam berbagai aplikasi salah satunya pengoptimalan rute logistik (Xin, Liang, Wen Song, Zhiguang Cao, and Jie Zhang, 2021).

Distribusi merupakan kegiatan proses penyaluran produk dari produsen sampai ke tangan masyarakat atau konsumen secara tepat waktu dan efisien (Tjiptono & Diana, 2020). Pengiriman yang tepat waktu merupakan salah satu tujuan dari proses distribusi yang dapat dilakukan dengan memahami lokasi tujuan distribusi. Terdapat beberapa lokasi tujuan pada proses pendistribusian yang

mengakibatkan biaya transportasi yang cukup tinggi. Biaya transportasi yang melebihi anggaran dikarenakan penentuan rute pendistribusian masih dilakukan secara manual atau acak yaitu penentuan jalur distribusi berdasarkan perkiraan saja.

Vehicle routing problem (VRP) merupakan masalah optimasi kombinatorial klasik yang pertama kali diusulkan oleh George Danzig pada tahun 1959 (Dantzig & Ramser, 1959). *Vehicle routing problem* (VRP) termasuk permasalahan *NP-Hard* yang umum dalam optimasi kombinatorial dan telah dipelajari selama beberapa dekade. Tujuan utama untuk menentukan rute optimal bagi armada kendaraan untuk melayani sekumpulan pelanggan. Penelitian terkait VRP adalah kunci untuk meningkatkan daya saing pada industri logistik. Perkembangan distribusi di dunia nyata dengan bermacam-macam karakteristik membuat banyaknya variasi VRP dari *single objective* hingga *VRP multi objective*. Terdapat berbagai jenis *vehicle routing problem* (VRP) misalnya *capacited vehicle routing problem* (CVRP), *VRP with time windows* (VRPTW), *Multi-Depot Vehicle Routing Problem* (MDVRP) (Abdirad Maryam, Krishnan Krishna, Gupta Deepak, 2022), *dynamic vehicle routing problem with time windows* (DVRPTW) (Ghannam & Gleixner, 2023) dan *vehicle routing problem with pickup and delivery* (VRRPPD) (M. Liu, Q. Song, Q. Zhao, L. Li, Z. Yang, Y. Zhang, 2022).

Dynamic vehicle routing problem with time window (DVRPTW) merupakan permasalahan optimasi rute yang menambahkan batasan jendela waktu ke dalam permasalahan. Hal ini berarti bahwa pelanggan memiliki periode waktu tertentu untuk dilayani. DVRPTW harus menjadwalkan pengiriman supaya barang diterima dalam rentang waktu yang ada sekaligus untuk meminimalkan biaya operasional (Liu et al., 2023). Permasalahan yang terjadi pada DVRPTW yaitu perubahan dinamis seperti pesanan baru, pembatalan, atau keterlambatan serta ketidakpastian lalu lintas, waktu pengiriman dan durasi pelayanan menambah kompleksitas permasalahan. Pengiriman dilakukan dalam jendela waktu yang menjadi kendala penting pada proses distribusi. Sebab jika pengiriman melebihi jendela waktu yang ditetapkan akan mengakibatkan ketidakpuasan pelanggan. Permasalahan ini yang

perlu untuk diselesaikan untuk meningkatkan solusi yang optimal untuk permasalahan yang ada.

Terdapat berbagai solusi untuk menyelesaikan permasalahan ini diantaranya metode eksak, metode heuristik dan metode metaheuristik (M. Liu, Q. Zhao, Q. Song, Y. Zhang, 2023). Penggunaan *machine learning*, khususnya *reinforcement learning* dalam penyelesaian VRP menawarkan pendekatan baru. Ini memungkinkan pengembangan algoritma yang dapat secara otomatis belajar dari lingkungan untuk menghasilkan solusi optimal dalam kondisi yang dinamis, seperti arus lalu lintas dan kedatangan pelanggan baru. Metode *reinforcement learning* yang umum untuk menyelesaikan VRP termasuk *dynamic programming*, algoritma *Q learning*, algoritma *deep Q-network* (DQN), *policy-based reinforce algorithms*, *value and policy combined actor-critic algorithms*, dan *advantage actor-critic algorithms* (Ni & Tang, 2023).

Pada masalah *dynamic vehicle routing problem with time window* (DVRPTW) sudah banyak metode yang digunakan untuk menyelesaikan permasalahan ini seperti *brain storm optimization* (BSO) dan *ant colony optimization* (ACO) (Liu et al. 2022), algoritma *hybrid brain storm optimization* (BSO) (Liu et al. 2023), Algoritma *dynamic hybrid genetic search* (HGS) (Ghannam & Gleixner, 2023). Penggunaan *reinforcement learning* juga banyak digunakan untuk menyelesaikan permasalahan ini seperti yang dilakukan oleh Joe & Lau (2020) menggabungkan *deep reinforcement learning* dan *simulated annealing* (DRLSA).

Penggunaan *deep reinforcement learning* juga dapat digunakan pada berbagai permasalahan optimasi rute seperti yang dilakukan oleh Li et al. (2021) pada model permasalahan *heterogeneous capacited vehicle routing problem* (HCVRP), Jiuxiu Zhao (2020) meneliti pada *vehicle routing problem*. Penggunaan *deep Q-network* seperti yang dilakukan oleh Bdeir et al. (2021) untuk permasalahan route mengusulkan *routing problem deep q-network* (RPDQN) untuk masalah *vehicle routing problem* hasilnya bahwa pendekatan RP-DQN berhasil

meningkatkan kinerja dalam menyelesaikan masalah perutean kendaraan dengan memanfaatkan representasi status dinamis dan efisiensi sampel yang lebih baik.

Penggunaan model berbasis perhatian (*attention*) yang telah diteliti oleh Kool et al. (2018) untuk menyelesaikan masalah optimisasi kombinatorial, khususnya masalah perutean seperti *Travelling Salesman Problem* (TSP) dan *Vehicle Routing Problem* (VRP). Hasilnya menunjukkan bahwa model yang dibuat menunjukkan fleksibilitas yang baik dalam menangani berbagai jenis masalah optimisasi kombinatorial dengan satu set *hyperparameters*.

Berdasarkan penelitian terdahulu penerapan *reinforcement learning* dapat secara otomatis mengidentifikasi pola dan strategi terbaik untuk mengoptimalkan rute distribusi dalam menghadapi berbagai kondisi dinamis yang sering berubah, seperti variabilitas arus lalu lintas yang tidak terduga dan permintaan pelanggan yang muncul tidak dapat diprediksi. Dengan kemampuan adaptasi ini, algoritma berbasis *reinforcement learning* tidak hanya meningkatkan efisiensi logistik dengan menemukan solusi rute yang optimal tetapi juga meningkatkan responsivitas terhadap kebutuhan pelanggan yang berfluktuasi, secara signifikan mengurangi waktu tunggu dan biaya operasional. Pendekatan ini, tidak hanya menjanjikan peningkatan dalam kinerja logistik tetapi juga menawarkan kemampuan untuk merespons secara lebih fleksibel terhadap tantangan operasional yang kompleks, memastikan kepuasan pelanggan dan keberlanjutan operasional dalam lingkungan bisnis yang semakin kompetitif.

Penelitian ini diharapkan mampu meningkatkan efisiensi dalam pemilihan rute pada konteks logistik dan distribusi yang dinamis, khususnya dalam menghadapi *dynamic vehicle routing problem with time windows* (DVRPTW). Masalah ini ditandai oleh kondisi yang terus berubah, seperti fluktuasi dalam arus lalu lintas, kedatangan pelanggan baru dan adanya jendela waktu untuk pengiriman, serta kebutuhan untuk mengirim produk dalam berbagai jenis atau kategori akan mempengaruhi proses pengiriman. Untuk mengatasi tantangan tersebut, penelitian ini mengusulkan penerapan *Deep Q-Network* (DQN) yang diperkaya dengan mekanisme *Multi-Header Attention*. Pendekatan ini dirancang untuk memanfaatkan

kemampuan DQN dalam memahami dan beradaptasi dengan kondisi dinamis, serta mengintegrasikan *Multi-Header Attention* untuk meningkatkan pemrosesan informasi. Penggabungan kedua teknologi ini, diharapkan sistem dapat secara efektif mengidentifikasi rute optimal yang memenuhi semua kriteria dan batasan yang ada, sekaligus menyesuaikan diri dengan perubahan kondisi yang ada.

1.3 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka rumusan masalah yang dapat disusun sebagai berikut.

1. Bagaimana *Deep Q-Network* (DQN) dengan mekanisme *Multi-Header Attention* dapat diterapkan untuk menyelesaikan *Dynamic Vehicle Routing Problem with Time Windows* (DVRPTW)?
2. Apakah penerapan DQN dengan *Multi-Header Attention* dapat meningkatkan efisiensi dan efektivitas dalam menentukan rute optimal pada kondisi dinamis seperti fluktuasi arus lalu lintas, kedatangan pelanggan baru dan adanya jendela waktu?

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dipaparkan, tujuan penelitian dari penelitian ini sebagai berikut.

1. Mengembangkan model berbasis *Deep Q-Network* (DQN) yang diperkaya dengan mekanisme Multi-Header Attention untuk menyelesaikan *Dynamic Vehicle Routing Problem with Time Windows* (DVRPTW).
2. Menilai efektivitas model DQN dengan *Multi-Header Attention* dalam meningkatkan efisiensi pengiriman dan kepuasan pelanggan melalui penentuan rute optimal dalam kondisi yang dinamis.
3. Menunjukkan kemampuan adaptasi model terhadap perubahan kondisi seperti fluktuasi lalu lintas, kedatangan pelanggan baru dan batasan jendela waktu.

Berdasarkan tabel perbandingan di atas dapat diketahui berbagai perbedaan pada masing-masing penelitian. Model permasalahan pada penelitian terdahulu terbagi menjadi beberapa model yaitu *dynamic vehicle routing problem* (DVRP), *Dynamic Vehicle Routing Problem with Time Windows* (DVRPTW), *capacited vehicle routing problem* (CVRP), *heterogeounus capacited vehicle routing problem* (HCVRP), *travel salesman problem* (TSP). Terkait fokus permasalahan yang diambil pada berbagai penelitian mulai dari permintaan pelanggan yang tidak pasti, keadaan lalu lintas, dan terkait kendaraan yang digunakan pada proses pengiriman. Penyelesaian dilakukan dengan menggunakan metaheuristik diantaranya *Hybrid Brain Storm Optimization* (BSO), *ant colony optimization*, dan yang lainnya. Penggabungan algoritma juga dilakukan pada beberapa penelitian terdahulu seperti *hybrid* antara *brain strom optimization* dengan *ant colony optimization*. Penggunaan *machine learning* yaitu *reinforcement learning*, *deep reinforcement learning* dan *deep q-network* digunakan pada berbagai penelitian sebab memiliki kelebihan yaitu lebih optimal pada data yang banyak. *Multi attention* juga digunakan pada penyelesaian permasalahan optimasi rute dan menunjukkan hasil yang optimal.

Pada penelitian selanjutnya fokus penelitian pada model masalah *dynamic vehicle routing problem with time windows* (DVRPTW) dengan fokus terhadap ketidakpastian jalan raya serta ketidakpastian pelanggan yang berubah-ubah, dimana pada proses pengiriman ke pelanggan terdapat jendela waktu atau batasan waktu pengiriman sampai ke pelanggan. Pada penelitian terdahulu hanya fokus pada salah satu saja seperti hanya fokus pada pelanggan yang tidak pasti atau ketidakpastian jalan raya. Penyelesaian dilakukan dengan menggunakan *deep reinforcement learning* pada hal ini menggunakan metode *deep Q-network* (DQN) dengan menggabungkan *multi-header attention* kedalam arsitektur DQN.

3.2 Pengumpulan Data

Langkah awal adalah mengumpulkan dataset yang akurat dan relevan. Dataset didapatkan dari data sekunder, dataset ini merupakan hal yang penting dari simulasi dan eksperimen, mencakup koordinat lokasi yang mungkin meliputi lokasi depot dan titik pengiriman, jendela waktu untuk setiap pengiriman yang menentukan batas awal dan akhir kapan pengiriman harus dilakukan, serta jumlah kendaraan. Data ini harus mencerminkan situasi dunia nyata untuk memastikan bahwa model yang dikembangkan dapat diaplikasikan secara praktis.

3.3 Persiapan Data

Langkah berikutnya adalah persiapan data. Pada persiapan data dilakukan normalisasi data. Normalisasi merupakan proses penting untuk menyamakan skala data, memastikan bahwa model dapat memprosesnya dengan efisien. Normalisasi *min-max* digunakan pada penelitian ini. *Min-max* adalah teknik yang mengubah skala nilai data ke dalam rentang baru seperti 0 hingga 1 atau -1 hingga 1. Teknik ini memastikan bahwa setiap fitur atau kolom data memberikan kontribusi yang seimbang dalam analisis tanpa membiarkan fitur dengan skala besar mendominasi.

Pengecekan matriks korelasi dilakukan untuk memahami hubungan antara variabel-variabel dalam dataset. Korelasi membantu mengidentifikasi fitur-fitur yang saling terkait dan memberikan wawasan tentang bagaimana setiap fitur dapat mempengaruhi model prediksi rute. Koefisien Korelasi *Pearson* digunakan untuk mengukur hubungan linear antara fitur.

3.4 Desain model

Implementasi *Deep Q-Network* (DQN) dengan mekanisme *attention* untuk *Dynamic Vehicle Routing Problem with Time Windows* (DVRPTW) melibatkan beberapa langkah utama, mulai dari pemilihan kerangka kerja hingga pembuatan lingkungan simulasi.

1. Pemilihan Kerangka Kerja

Kerangka kerja yang digunakan yaitu *TensorFlow*, dimana kerangka kerja ini menawarkan lingkungan yang komprehensif dengan *TensorBoard* untuk visualisasi, serta dukungan terhadap TPU untuk akselerasi komputasi. *TensorFlow* mungkin lebih cocok untuk produksi dan skala besar.

2. Desain model DQN dan Multi header-attention

DQN adalah algoritma pembelajaran penguatan yang menggunakan jaringan saraf tiruan untuk memperkirakan fungsi nilai Q, yang merepresentasikan nilai maksimum hadiah kumulatif yang diharapkan, diberikan sebuah state dan semua strategi yang mungkin diambil. Implementasi DQN melibatkan beberapa komponen utama:

Jaringan Q: Jaringan ini memperkirakan nilai Q untuk setiap aksi dari *state* tertentu. Dalam kasus DVRPTW, input bisa berupa representasi dari *state* saat ini (misalnya, lokasi kendaraan, status pengiriman) dan *output* adalah nilai Q untuk setiap kemungkinan aksi (misalnya, memilih lokasi pengiriman berikutnya).

Memory Replay: Untuk meningkatkan stabilitas dan efisiensi pembelajaran, DQN menggunakan teknik *memory replay*, di mana transisi (*state*, aksi, *reward*, *state* baru) disimpan dalam sebuah *buffer*. *Batch* transisi ini kemudian digunakan untuk melatih jaringan Q, memungkinkan pengalaman dari masa lalu digunakan kembali.

Strategi Eksplorasi: Seperti ϵ -greedy, di mana aksi acak dipilih dengan probabilitas ϵ untuk mendorong eksplorasi lingkungan.

Mekanisme *attention* terdiri dari tiga matriks utama: *Query* (Q), *Key* (K), dan *Value* (V) untuk setiap head i . Adapun langkah-langkahnya implementasinya sebagai berikut:

a. Definisikan Ukuran Input dan Parameter

Mentukan jumlah fitur input, dimensi *embedding*, jumlah *heads* untuk mekanisme *attention*, dan jumlah unit dalam lapisan tersembunyi DQN. Serta jumlah tindakan yang mungkin dilakukan oleh agen.

b. Definisikan Multi-Header Attention

Menerapkan mekanisme *Multi-Header Attention* pada representasi vektor dari *embedding layer*. *Multi-Header Attention* menggunakan *Query* (Q), *Key* (K),

dan *Value* (V) untuk menangkap hubungan kontekstual dalam data. Proses ini membantu model untuk fokus pada aspek-aspek penting dari data input. *Attention Score* dihitung dengan mengalikan *Query* dengan *Key*, kemudian membaginya dengan skala (biasanya akar dari dimensi *Key*) dan menerapkan fungsi *softmax* untuk mendapatkan bobot *attention*. *Output Attention* diperoleh dengan mengalikan bobot perhatian dengan *Value*. *Multi-Header Attention* melakukan proses ini beberapa kali secara paralel (dengan beberapa "heads") dan hasilnya digabungkan untuk data input.

c. Definisikan DQN dengan Lapisan Tersembunyi

Membuat beberapa lapisan tersembunyi (*hidden layers*) menggunakan fungsi aktivasi ReLU. Lapisan tersembunyi ini memungkinkan jaringan untuk belajar representasi yang kompleks dari data input. Output dari mekanisme *attention* diberikan sebagai input ke DQN. DQN memperkirakan *Q-values* untuk setiap tindakan yang mungkin dilakukan oleh agen berdasarkan representasi *state* yang telah diperkaya.

d. *Output Layer* untuk *Q-values*

Lapisan output menghasilkan *Q-values* untuk setiap tindakan yang mungkin dilakukan oleh agen. *Q-values* ini menunjukkan seberapa baik setiap tindakan dalam memaksimalkan *reward* di masa depan. Misalnya, jika agen memiliki 5 kemungkinan tindakan, lapisan output akan menghasilkan 5 *Q-values*, satu untuk setiap tindakan.

e. Memilih Tindakan menggunakan Strategi ϵ -greedy

Implementasikan strategi ϵ -greedy untuk memastikan agen mengeksplorasi lingkungan sekaligus mengeksploitasi pengetahuan yang ada. Dengan probabilitas ϵ , agen memilih tindakan secara acak untuk eksplorasi, dan dengan probabilitas $1 - \epsilon$, agen memilih tindakan dengan *Q-value* tertinggi untuk eksploitasi.

f. Memperbarui Model dengan Pengalaman dari *Replay Buffer*

Menggunakan *replay buffer* untuk menyimpan transisi (*state*, *action*, *reward*, *next state*) dan menggunakannya untuk melatih model. *Batch* transisi

diambil secara acak dari *replay buffer* untuk mengurangi korelasi antara sampel pelatihan dan meningkatkan stabilitas pelatihan.

3.5 Pelatihan Model

Selama fase pelatihan, model secara berulang kali dihadapkan pada berbagai skenario dari masalah rute kendaraan. Untuk setiap episode, model mengambil serangkaian aksi berdasarkan *policy* atau kebijakan saat ini yang awalnya adalah kebijakan acak dengan tujuan meminimalkan jarak total dan memenuhi jendela waktu pengiriman. Setelah mengambil aksi, model menerima *feedback* dari lingkungan berupa *reward* yang merupakan ukuran dari performa aksi tersebut dan *state* baru yang mencerminkan kondisi terkini dari lingkungan setelah aksi diambil. Informasi ini digunakan untuk memperbarui kebijakan model dengan cara mengoptimalkan parameter jaringan sehingga meningkatkan estimasi nilai Q, yang merepresentasikan hadiah kumulatif yang diharapkan.

Untuk meningkatkan stabilitas dan efisiensi pelatihan, teknik seperti *experience replay* dan *target networks* digunakan. *Experience replay* memungkinkan model untuk belajar dari pengalaman masa lalu yang disimpan dalam *memory replay*, sedangkan *target networks* membantu mengurangi pergeseran target yang bergerak selama proses pembelajaran. Melalui interaksi yang berulang dan proses optimisasi ini, model secara bertahap belajar untuk memprediksi nilai Q yang lebih akurat untuk setiap kombinasi *state* dan aksi, yang mengarah pada pembentukan kebijakan rute yang lebih optimal.

Selain itu pada tahap pelatihan model ini juga dilakukan penyetelan *hyperparameter* untuk menemukan nilai optimal *hyperparameter* guna meningkatkan kinerja model. Ini merupakan langkah penting dalam *machine learning* karena dapat menghasilkan peningkatan akurasi, efisiensi, dan *generalizability* model. Penyetelan *hyperparameter* menggunakan teknik *random search*, yaitu teknik yang digunakan untuk penyetelan *hyperparameter* yang melibatkan pemilihan acak dari ruang yang ditentukan untuk menemukan kombinasi terbaik yang mengoptimalkan kinerja model. Teknik ini lebih efektif dan

efisien untuk penyetelan *hyperparameter* terutama pada kasus dengan ruang pencarian yang besar, serta dapat menemukan solusi yang baik dalam waktu yang lebih singkat.

3.6 Evaluasi Model

Setelah fase pelatihan model *Deep Q-Network* (DQN) dengan *multi-header attention* untuk *Dynamic Vehicle Routing Problem with Time Windows* (DVRPTW) selesai, langkah evaluasi menjadi penting untuk memahami seberapa efektif model dalam menyelesaikan masalah yang ditargetkan. Evaluasi dilakukan dengan menguji model terhadap kumpulan data pengujian yang tidak terlibat selama proses pelatihan, memberikan masukan penting tentang kemampuan generalisasi model terhadap skenario baru dan belum pernah dilihat. Dalam konteks DVRPTW, metrik yang relevan seperti total jarak tempuh oleh semua kendaraan dan kepatuhan terhadap jendela waktu pengiriman menjadi fokus utama. Total jarak tempuh mencerminkan efisiensi rute yang dihasilkan, sementara kepatuhan terhadap jendela waktu mencerminkan kualitas layanan yang dapat dijamin oleh model.

3.7 Analisis dan Penyempurnaan

Langkah terakhir yaitu analisis secara mendalam kinerja model pada dataset pengujian. Penyempurnaan dilakukan untuk mengatasi kelemahan yang telah dianalisis sebelumnya seperti penyempurnaan pada *tuning hyperparameter* untuk peningkatan kinerja, modifikasi arsitektur dan pelatihan ulang model.

3.8 Jadwal Penelitian

Jadwal penelitian digunakan untuk meningkatkan efektivitas dalam proses penelitian. Adanya jadwal penelitian ini setiap proses penelitian sudah terjadwal dalam tabel 3.1 sehingga penelitian lebih efektif dan optimal.