



Pengembangan Optimasi Rute Menggunakan *Deep Q-Network* dengan Mekanisme *Multi-Header Attention* Pada Model *Dynamic Vehicle Routing Problem With Time Windows (DVRPTW)*

KUALIFIKASI

Alifurrohman

99223116

PROGRAM DOKTOR TEKNOLOGI INFORMASI

UNIVERSITAS GUNADARMA

Juni 2024

DAFTAR ISI

BAB 1	4
PENDAHULUAN.....	4
1.1 Latar Belakang	4
1.2 Batasan Penelitian	8
1.3 Rumusan Masalah.....	9
1.4 Tujuan Penelitian	9
1.5 Kontribusi Dan Manfaat Hasil Penelitian	10
BAB 2	11
TELAAH PUSTAKA	11
2.1 Pengertian <i>Artificial intelligence</i>	11
2.2 <i>Machine Learning</i>	12
2.3 <i>Deep Learning</i>	12
2.4 <i>Reinforcement Learning</i>	13
2.5 <i>Deep Reinforcement Learning</i>	15
2.6 <i>Deep Q-Network (DQN)</i>	15
2.7 Multi-Header Attention	16
2.8 <i>Vehicle Routing Problem (VRP)</i>	17
2.9 <i>Dynamic Vehicle Routing Problem with Time Windows (DVRPTW)</i>	18
2.10 Perbandingan Tinjauan	22
BAB 3	27
METODOLOGI PENELITIAN	27
3.1 Kerangka Umum Penelitian.....	27

3.2	Pengumpulan Data	28
3.3	<i>Preprocessing</i> Data	28
3.4	Desain model	29
3.5	Pelatihan Model	31
3.6	Evaluasi Model	32
3.7	Analisis dan Penyempurnaan	32
3.8	Jadwal Penelitian	32
DAFTAR PUSTAKA		34

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 proses distribusi yaitu ketidakpastian lalu lintas dan waktu pengiriman merupakan permasalahan utama pada distribusi. Pengiriman dilakukan dalam jendela waktu menjadi salah satu kendala penting pada proses distribusi. Sebab jika pengiriman melebihi jendela waktu yang ditetapkan dan kepadatan lalu lintas yang dinamis akan mengakibatkan ketidakpuasan pelanggan dan meningkatkan biaya operasional perusahaan.

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. Dengan kemampuan adaptasi ini, algoritma berbasis *reinforcement learning* dapat meningkatkan efisiensi logistik dengan menemukan solusi rute yang optimal sehingga dapat mengurangi biaya operasional.

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 dan adanya jendela waktu untuk 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 dengan perubahan kondisi yang ada.

1.2 Batasan Penelitian

Batasan pada penelitian ini dapat dijelaskan sebagai berikut.

1. Penelitian ini dibatasi pada penerapan dan evaluasi *Deep Q-Network* (DQN) dengan mekanisme *Multi-Header Attention* dalam konteks DVRPTW, tanpa membandingkan secara langsung dengan semua metode heuristik dan metaheuristik lainnya.
2. Analisis akan fokus pada kondisi dinamis seperti fluktuasi arus lalu lintas, dan adanya jendela waktu (*time window*).
3. Penelitian ini terbatas pada simulasi komputasi dan tidak mencakup implementasi secara langsung pada perusahaan.

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 dan adanya jendela waktu?
3. Bagaimana metode ini dapat beradaptasi dengan perubahan dinamis seperti keadaan lalu lintas dan 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) 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 melalui penentuan rute optimal dalam kondisi yang dinamis dan adanya jendela waktu.

3. Menunjukkan kemampuan adaptasi model terhadap perubahan kondisi seperti keadaan lalu lintas dan batasan jendela waktu.

1.5 Kontribusi Dan Manfaat Hasil Penelitian

Penelitian ini memberikan wawasan tentang penerapan teknologi *machine learning*, khususnya *Deep Q-Network* dan mekanisme *Multi-Header Attention*, dalam menyelesaikan masalah logistik yang kompleks. Menawarkan solusi yang lebih efisien dan efektif untuk penjadwalan dan pengiriman dalam kondisi dinamis, yang dapat meningkatkan kepuasan pelanggan dan mengurangi biaya operasional. Kontribusi pada perusahaan dengan menyediakan kerangka kerja yang dapat diadaptasi oleh perusahaan untuk meningkatkan proses pengambilan keputusan mereka dalam distribusi produk, khususnya dalam menghadapi kondisi yang berubah-ubah dan tidak pasti.

BAB 2

TELAAH PUSTAKA

2.1 Pengertian *Artificial intelligence*

Artificial intelligence (AI) merupakan bidang komputasi yang berfokus pada transmisi kecerdasan dan pemikiran antropomorfik yang dapat membantu manusia dalam berbagai cara. *Artificial intelligence* pertama kali dikemukakan oleh John McCarthy pada tahun 1956 (PK & most of all, 1984). AI secara perlahan-lahan bermunculan dan semakin kuat di berbagai bidang seperti teknik, matematika, fisika, teknologi, yang semuanya telah menyebabkan pergeseran luar biasa saat ini di bidang ini. *Artificial Intelligence* (AI) atau Kecerdasan Artifisial didefinisikan sebagai kapasitas untuk beradaptasi dengan pengetahuan dan sumber daya yang tidak memadai. Definisi ini menekankan pada kemampuan sistem AI untuk melakukan adaptasi dan membuat keputusan dalam situasi di mana informasi tidak lengkap atau sumber daya terbatas, yang mencerminkan aspek penting dari kecerdasan yang dihadapi oleh manusia dalam kehidupan nyata (Wang, 2019).

AI merupakan salah satu bidang terbaru dalam sains dan teknik. Terdapat empat pendekatan terkait pengertian *Artificial intelligence* yaitu *Thinking Humanly*, *Thinking Rationally*, *Acting Humanly*, dan *Action Rationally* (Russell & Norvig, 2016). *Thinking humanly* berfokus pada upaya membuat komputer berpikir dengan cara yang sama seperti manusia. Definisi ini menekankan pada pemahaman dan imitasi proses mental manusia. *Thinking rationally* mempelajari proses mental melalui model komputasional, yang bertujuan untuk menghasilkan pemikiran logis dan rasional. Definisi ini menekankan pada logika dan teori komputasi. *Acting humanly* menekankan pada pembuatan mesin yang dapat melakukan fungsi yang memerlukan kecerdasan saat dilakukan oleh manusia. Fokusnya adalah pada perilaku eksternal mesin yang menyerupai manusia. *Action rationally* melihat kecerdasan komputasional sebagai studi tentang perancangan agen cerdas yang bertindak secara rasional. Fokusnya adalah pada perilaku yang logis dan efisien dalam mencapai tujuan tertentu.

AI dapat diterapkan pada berbagai bidang di kehidupan manusia contohnya AI dapat digunakan pada peralatan kesehatan, bidang manufaktur, keamanan dan pengawasan, dan bidang pendidikan (Karthikeyan, 2021).

2.2 Machine Learning

Machine Learning (ML) merupakan cabang dari ilmu komputer yang memungkinkan sistem komputer memperoleh pengetahuan dari data secara mirip dengan cara kerja manusia. Secara sederhana, ML adalah bentuk dari kecerdasan artifisial yang dapat mengidentifikasi pola dalam data mentah melalui penggunaan algoritma atau teknik tertentu. Tujuan utama ML adalah untuk membuat sistem komputer dapat mengakuisisi pengetahuan dari pengalaman secara otomatis tanpa perlu pemrograman yang spesifik atau intervensi dari manusia.

Machine learning adalah cabang algoritma komputasi yang berkembang dan dirancang untuk meniru kecerdasan manusia dengan belajar dari lingkungan sekitar. *Machine learning* dianggap sebagai hal penting pada era *big data* seperti saat ini dan telah berhasil diterapkan pada berbagai bidang seperti pengenalan pola, *computer vision*, keuangan, dan medis (El Naqa & Murphy, 2015).

Machine learning dapat diklasifikasikan menjadi beberapa diantaranya yaitu *supervised learning*, *unsupervised learning*, *semi-supervised learning*, *transductive inference*, *on-line learning*, *reinforcement learning*, dan *active learning* (Mohri et al., 2018).

2.3 Deep Learning

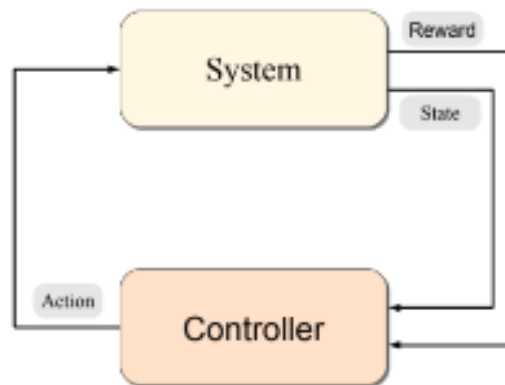
Deep Learning adalah subbidang kecerdasan artifisial yang berfokus pada pembuatan model jaringan saraf besar yang mampu membuat keputusan berdasarkan data yang akurat. *Deep learning* sangat cocok untuk konteks dimana datanya kompleks dan terdapat kumpulan data yang besar (Kelleher, 2019). *Deep learning* merupakan perangkat metodologis untuk membangun saraf multilayer (*deep*) *neural network* untuk memecahkan permasalahan dalam *supervised clasification*, *generative modelling* atau *reinforcement learning* (Saxe, A., Nelli, S., & Summerfield, C. 2021).

Deep learning adalah cabang pembelajaran mesin yang berkembang pesat yang menggunakan banyak lapisan tersembunyi (*hidden layer*) untuk mengekstrak fitur dari kumpulan data besar. pembelajaran mesin yang merupakan bentuk lanjutan dari algoritma jaringan saraf (*neural network*), di mana istilah "*neural*" digunakan untuk menekankan proses pembelajaran yang sebanding dengan otak manusia (Arif, T. M., & Rahim, M. A., 2024).

2.4 Reinforcement Learning

Reinforcement learning (RL) adalah bidang studi yang berfokus pada pengembangan algoritma dan model yang memungkinkan komputer untuk belajar dan membuat prediksi atau keputusan tanpa di program secara eksplisit. RL melibatkan penggunaan teknik statistik dan model komputasi untuk menganalisis dan menafsirkan sejumlah besar data. Hal ini memungkinkan komputer untuk mengidentifikasi pola, membuat prediksi dan meningkatkan kinerja (Szepesvári, 2022).

Reinforcement learning adalah pendekatan komputasional untuk memahami dan mengotomatisasi pembelajaran dan pengambilan keputusan yang berorientasi pada tujuan. Pendekatan ini berbeda dari pendekatan komputasional lainnya karena menekankan pembelajaran oleh agen dari interaksi langsung dengan lingkungannya, tanpa memerlukan pengawasan contoh atau model lengkap dari lingkungan. *Reinforcement learning* menggunakan kerangka formal dari proses keputusan Markov untuk mendefinisikan interaksi antara agen pembelajar dan lingkungannya dalam hal keadaan (*states*), tindakan (*actions*), dan hadiah (*rewards*). Kerangka ini dimaksudkan sebagai cara sederhana untuk merepresentasikan fitur-fitur penting dari masalah kecerdasan artifisial (Sutton & Barto, 2018).



Gambar 2.1 Skenario Dasar *Reinforcement Learning*

Sumber : Szepesvari,2022

Pada gambar 2.1 Pengaturan tipikal dari *reinforcement learning* melibatkan sebuah *controller* (pengendali) yang menerima informasi tentang *state* (keadaan) dari sistem yang dikendalikan serta *reward* (hadiah) yang terkait dengan transisi keadaan terakhir. Berdasarkan informasi tersebut, pengendali menghitung *action* (tindakan) yang harus diambil dan mengirimkannya kembali ke sistem. Sistem kemudian merespons dengan melakukan transisi ke keadaan baru, dan siklus ini berulang. Tujuan utama dari proses ini adalah untuk mempelajari cara mengontrol sistem agar dapat memaksimalkan total hadiah yang diperoleh dari waktu ke waktu (Szepesvari,2022).

Algoritma RL dapat dibagi menjadi dua kategori: pembelajaran berbasis model dan pembelajaran bebas model. Pembelajaran berbasis model memiliki pengetahuan sebelumnya tentang lingkungan, yang dapat dioptimalkan terlebih dahulu. Sedangkan 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. Pembelajaran berbasis model yang umum untuk menyelesaikan VRP termasuk pemrograman dinamis; pembelajaran bebas model algoritma terutama

mencakup *value-based temporal difference algorithm*, *Q-learning*, algoritma *deep Q-Network* (DQN), algoritma *policy-based reinforc.*

2.5 *Deep Reinforcement Learning*

Pembelajaran penguatan mendalam (*Deep Reinforcement Learning* - DRL) menggabungkan kemampuan ekstraksi fitur dari pembelajaran mendalam (*deep learning*) dan kemampuan pengambilan keputusan dari pembelajaran penguatan (*Reinforcement Learning* - RL), yang secara langsung memungkinkan pembuatan keputusan terbaik berdasarkan data input multidimensional. Ini adalah sistem kontrol keputusan ujung ke ujung yang banyak digunakan dalam pengambilan keputusan dinamis, prediksi waktu nyata, simulasi, permainan, dan sebagainya. DRL berinteraksi dengan lingkungan secara *real time*, mengambil informasi lingkungan sebagai input untuk mendapatkan pengalaman kegagalan atau keberhasilan, dan memperbarui parameter jaringan keputusan, sehingga mempelajari keputusan optimal (Ni & Tang, 2023) .

Metode *deep reinforcement learning* diperoleh ketika kita menggunakan *deep neural networks* untuk mendekati salah satu komponen pembelajaran penguatan: fungsi nilai, $v^{\pi}(s; \theta)$ atau $q^{\pi}(s, a; \theta)$ kebijakan $\pi(a|s; \theta)$ dan model (fungsi transisi keadaan dan fungsi hadiah). Perkembangan *deep reinforcement learning* berkembang hingga berkembang menjadi *deep Q-Network* (Li, 2017).

2.6 *Deep Q-Network (DQN)*

Deep Q-Network (DQN) adalah pendekatan pembelajaran penguatan bebas model (*model-free*) yang menggunakan *deep neural networks* (DNN) untuk memperkirakan fungsi Q (*Q-function*) dalam ruang keadaan (*state space*) yang berdimensi tinggi dan kompleks. DQN diparameterisasi oleh bobot jaringan θ yang dapat diperbarui dengan berbagai algoritma *reinforcement learning* (Hong et al., 2017).

Untuk mendekati fungsi Q optimal yang diberikan kebijakan π dan pasangan keadaan-tindakan (s,a) , DQN secara bertahap memperbarui parameternya θ sehingga $Q^*(s,a) \approx Q(s,a;\theta)$.

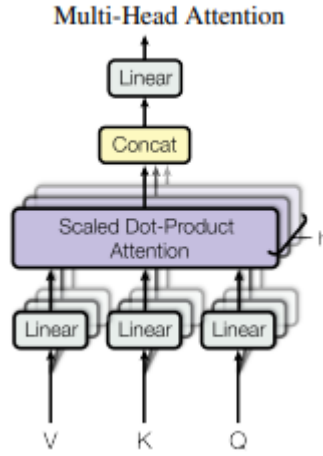
Parameter-parameter ini diperbarui melalui metode penurunan gradien (gradient descent) yang secara iteratif meminimalkan fungsi kerugian $L(\theta)$. Proses ini menggunakan sampel (s, a, r, s') yang diambil dari *memori replay* pengalaman Z . Fungsi kerugian $L(\theta)$ dinyatakan sebagai:

$$L(\theta) = \mathbb{E}_{s,a,r,s' \sim U(Z)} [(y - Q(s,a;\theta))^2]$$

Di mana $y = r + \gamma \max_{a'} Q(s',a';\theta^-)$, $U(Z)$ adalah distribusi acak uniform dari Z , dan θ^- adalah parameter dari jaringan target. Jaringan target adalah versi dari jaringan online, namun parameternya θ^- diperbarui oleh jaringan online pada interval waktu yang telah ditentukan. *Memori replay* dan jaringan target bersama-sama meningkatkan stabilitas proses pembelajaran secara signifikan.

2.7 Multi-Header Attention

Multi-header attention adalah mekanisme dalam model pembelajaran mendalam yang memungkinkan model untuk fokus pada berbagai bagian input secara simultan dengan menggunakan beberapa "head". Setiap "head" memproses input dengan cara yang berbeda, memungkinkan model untuk menangkap berbagai jenis informasi dari data. Hal ini meningkatkan pemahaman dan representasi model terhadap data yang kompleks (Baan et al., 2019).



Mekanisme *multi header-attention* bergantung pada *scaled dot-product attention* yang beroperasi pada *query* (Q), *key* (K), dan *value* (V).

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Dimana d_k adalah dimensi kunci. Pada *self-attention*, *qery*, *key*, dan *value* berasal dari hasil lapisan sebelumnya (Voita, 2019).

Multi-header attention memungkinkan model untuk secara bersamaan memperhatikan informasi dari subruang representasi yang berbeda pada posisi yang berbeda (Vaswani et al., 2017). Mekanisme *multi-header attention* menghasilkan beberapa representasi berbeda dari (Q, K, V) untuk setiap *head*, melakukan perhitungan *scaled dot-product attention* pada setiap representasi, menggabungkan hasilnya, dan memproyeksikan gabungan tersebut melalui lapisan *feed-forward*. Proses ini dapat diungkapkan menggunakan notasi yang sama seperti pada Persamaan.

$$Multihead(Q, K, V) = Concat(Head_1, \dots, Head_h)W^O$$

$$Where Head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

Dimana W_i dan W^O adalah parameter matriks.

2.8 Vehicle Routing Problem (VRP)

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) adalah masalah optimasi yang berfokus pada perencanaan rute terbaik bagi armada kendaraan untuk mengunjungi pelanggan atau lokasi tertentu, dengan tujuan mengoptimalkan biaya seperti jarak atau waktu. VRP merupakan inti dari manajemen distribusi dan dihadapi setiap hari oleh ribuan perusahaan yang bergerak dalam pengiriman dan pengumpulan barang atau orang. Karena kondisi bervariasi antara satu kasus dengan kasus lainnya, tujuan dan kendala dalam praktik juga sangat beragam. Studi VRP telah menghasilkan berbagai teknik solusi eksak dan heuristik yang dapat diterapkan secara umum, dan banyak di antaranya telah diadaptasi untuk menyelesaikan varian lain dari masalah ini (Cordeau et al., 2007).

VRP didefinisikan pada graf tak berarah lengkap $G=(V,E)$. Dimana set $V: \{0, \dots, n\}$ adalah himpunan titik, di mana setiap titik $i \in V \setminus \{0\}$ mewakili pelanggan dengan permintaan tidak negatif q_i , dan titik 0 mewakili depot. Set $E: (i,j)$ adalah himpunan sisi dengan biaya perjalanan c_{ij} untuk setiap sisi. Armada Tetap: Armada terdiri dari m kendaraan identik dengan kapasitas Q , yang tersedia di depot.

VRP didefinisikan pada graf berarah $G=(V,A)$, di mana A adalah himpunan busur. Dalam hal ini, rute kendaraan dihubungkan dengan siklus berarah. Tujuan dari VRP yaitu menentukan serangkaian m rute dengan biaya perjalanan total minimal yang memenuhi (Cordeau, 2007):

1. Setiap pelanggan dikunjungi tepat sekali oleh satu rute.
2. Setiap rute dimulai dan diakhiri di depot.
3. Total permintaan pelanggan dalam satu rute tidak melebihi kapasitas kendaraan Q .
4. Panjang setiap rute tidak melebihi batas yang ditetapkan L .

2.9 *Dynamic Vehicle Routing Problem with Time Windows (DVRPTW)*

Dynamic Vehicle Routing Problem with Time Windows (DVRPTW) merupakan perluasan dari *vehicle routing problem* (VRP) klasik yang mempertimbangkan sifat dinamis dari permasalahan tersebut, memerlukan pembaruan terus-menerus terhadap rute kendaraan saat permintaan pelanggan baru tiba. Permasalahan DVRPTW melibatkan pencarian rute optimal untuk kendaraan melayani pelanggan dengan jendela waktu, memastikan setiap pelanggan dikunjungi tepat sekali oleh satu kendaraan sambil mempertimbangkan batasan waktu dan permintaan yang berkembang (Necula, 2017). *Dynamic vehicle routing problem with time window* (DVRPTW) adalah turunan dari VRPTW tradisional yang mempertimbangkan karakteristik pelanggan yang dinamis. Tujuan DVRPTW adalah menggunakan jumlah kendaraan sedikit mungkin untuk memenuhi permintaan pelanggan dengan total waktu perjalanan yang minimum (Teng, 2024).

DVRPTW dapat didefinisikan pada graf lengkap $G (C,E)$ dimana $C = \{C_0, C_1, \dots, C_n\}$ menyatakan himpunan pelanggan dan C_0 yaitu depot. $E = \{(C_i, C_j) | C_i, C_j \in C, i \neq j\}$ menandakan sisi-sisi yang menghubungkan para pelanggan. Dalam konteks VRPTW, maka depot menampung K kendaraan, masing-masing dengan kapasitas Q dan kecepatan kendaraan ditandai dengan V . Setiap pelanggan memiliki jendela waktu pelayanan yang dinotasikan sebagai $[e_i, l_i]$. Jumlah permintaan pelanggan dinotasikan sebagai q_i . Ketika kendaraan k dijadwalkan untuk melayani pelanggan sangat penting untuk memastikan kapasitas kendaraan Q tidak kurang dari total nilai q_i yang sesuai dengan pelanggan yang dilayani. Kendaraan k harus melayani pelanggan c_i dalam jendela waktu $[e_i, l_i]$. Jika kendaraan tiba sebelum e_i maka kendaraan harus menunggu hingga e_i dimana waktu tunggu adalah w_i . Model matematis didefinisikan sebagai berikut.

K jumlah kendaraan

C jumlah pelanggan

V kecepatan kendaraan

Q kapasitas kendaraan

C_{ij} biaya jarak antara c_i dan c_j

T_{ij} waktu tempuh antara c_i dan c_j

q_i permintaan c_i

e_i waktu paling awal c_i dapat dilayani

l_i waktu terakhir c_i dapat dilayani

s_i waktu yang diperlukan untuk melayani c_i

t_i waktu kendaraan tiba di c_i

w_i waktu tunggu kendaraan di c_i

T'_{ij} waktu tempuh dinamis antara c_i dan c_j pada waktu t , mempertimbangkan kondisi lalu lintas

$D_{ij}(t)$ keterlambatan karena lalu lintas c_i dan c_j pada waktu t

$A_k(t)$ pelanggan baru yang tiba yang bisa dilayani oleh kendaraan k pada waktu t

$R_k(t)$ rute dinamis kendaraan k pada waktu t , sebagai urutan pelanggan yang dilayani. Berikut merupakan fungsi objektif

1. Meminimalkan biaya

$$\min_x \sum_{k \in K} \sum_{(i,j) \in E} \text{cost}_{(i,j)} p x_{(i,j)} p k$$

Fungsi minimal biaya total dari biaya perjalanan, yang disimbolkan sebagai $\text{cost}_{(i,j)}$ dikalikan dengan keputusan rute $p x_{(i,j)}$, dan faktor kendaraan $p k$ dijumlahkan melalui semua kendaraan dalam himpunan K dan semua pasangan titik (i,j) dalam himpunan E .

2. Mewakili variasi biner yang digunakan untuk menunjukkan apakah kendaraan k menggunakan jalur ij

$$x_{ijk} = \begin{cases} 1 & \text{jika kendaraan } k \text{ menuju } (i,j) \\ 0 & \text{sebaliknya} \end{cases}$$

Dimana jika nilai x_{ijk} bernilai 1 maka kendaraan k melakukan perjalanan dari lokasi i ke lokasi j . sebaliknya jika nilai x_{ijk} bernilai 0 maka kendaraan k tidak mengunjungi lokasi i ke lokasi j .

3. Kendaraan harus berangkat dan berakhir di depot

$$\sum_{i \in V} \sum_{p \in E_{(i,0)}} x(i,0)pk = \sum_{j \in V} \sum_{p \in E_{(0,j)}} x(0,j)pk = 1(\forall k \in K)$$

Kendala diatas memastikan bahwa setiap kendaraan k dalam himpunan K mulai dan diakhiri di depot. $x(i,0)$ mengartikan berangkat dari depot dan $x(0,j)$ untuk kembali ke depot. persamaan diatas menyatakan bahwa jumlah dari $x(i,0)$ dan $x(0,j)$ dikalikan dengan pk harus sama dengan 1 untuk setiap kendaraan k .

4. Batasan kapasitas kendaraan

$$\sum_{i \in C} q_i \sum_{j \in C, j \neq i} x(i,j)pk \leq Q(\forall k \in K)$$

Menjelaskan bahwa setiap kendaraan k yang ada dalam himpunan K , jumlah produk permintaan pelanggan q_i dan variabel keputusan $x(i,j)$ yang dijumlahkan untuk semua pelanggan i dalam himpunan C dan untuk semua pelanggan j yang berbeda dari i dalam himpunan C dikalikan dengan parameter pk harus kurang dari atau sama dengan kapasitas total kendaraan Q .

5. Batasan jendela waktu kendaraan

$$t'_1 = t_i + w_i + s_i$$

Menjelaskan waktu layanan dipelanggan berikutnya yaitu penjumlahan antara waktu kedatangan pelanggan saat t_i , waktu tunggu pelanggan pada saat w_i , dan waktu layanan di pelanggan s_i .

$$w_j = \begin{cases} \max\{e_j - t'_i - t_{ij}, 0\} & \text{if } j \in C \\ 0 & \text{if } j \in V \end{cases} \quad (\forall i, j \in V)$$

Waktu tunggu di pelanggan j jika j merupakan pelanggan maka pengurangan antara selisih waktu paling awal pelanggan dapat dilayani dikurangi waktu layanan di pelanggan sebelumnya dan waktu perjalanan dari pelanggan i dan j . jika bukan pelanggan maka waktu tunggu adalah nol.

$$t_i + s_i + t_{ij} + w_i \leq t_j \quad \forall i \in V, j \in C, i \neq j$$

Waktu kedatangan di pelanggan i ditambah waktu layanan di pelanggan i , ditambah waktu perjalanan dari pelanggan i ke j dan ditambah waktu tunggu pada pelanggan i harus kurang dari atau sama dengan waktu kedatangan pada pelanggan j .

$$e_i \leq t_i + w_i \leq l_i \quad (\forall i \in C)$$

Waktu paling awal pelanggan i dapat dilayani harus kurang atau sama dengan waktu kedatangan ditambah waktu tunggu di pelanggan i dan harus kurang dari atau sama dengan waktu paling akhir pelanggan i dapat dilayani.

Parameter yang digunakan pada DVRPTW pada jaringan jalan raya yaitu:

A_t^k menggunakan rute di $(t-1) \sim t$

RN_t pelanggan yang ditolak di $(t-1) \sim t$

T himpunan sepanjang waktu dalam sehari

d_{kt} titik awal kendaraan k pada momen t , berisi titik awal virtual dan depot

6. Jumlah kumulatif biaya pengiriman

$$\min TD = \sum_{t \in T} \sum_{k \in K} \sum_{(i,j) \in A_t^k} \cos t_{(i,j)} p x_{(i,j)} p k$$

7. Jumlah kumulatif permintaan yang tidak terlayani

$$\min RN = \sum_{t \in T} RN_t$$

Nilai yang hendak diminimalkan ini merupakan hasil penjumlahan dari semua RN_t sepanjang himpunan waktu T , yang mana RN_t merepresentasikan jumlah pelanggan yang ditolak pada waktu t .

8. Kendaraan berangkat dari titik awal atau depot dan kembali ke depot

$$\text{Sama dengan: } \sum_{i \in V} \sum_{p \in A_{(i,0)}} x(i,0) p k = \sum_{j \in V} \sum_{p \in A_{(d_{kt},j)}} x(d_{kt},j) p k = 1 \quad (\forall k \in K)$$

2.10 Perbandingan Tinjauan

Perbandingan tinjauan pustaka ini bertujuan untuk memberikan gambaran yang komprehensif mengenai fokus penelitian, permasalahan dan metode dari tiap-tiap pendekatan yang telah diusulkan oleh peneliti sebelumnya. Berikut merupakan tabel perbandingan tinjauan.

Tabel 2.1 Perbandingan Tinjauan

Judul	Penulis	Fokus	Permasalahan	Metode
<i>A Hybrid of Deep Reinforcement learning and Local Search for the Vehicle Routing Problems</i>	Jiuxia Zhao, Minjia Mao, Xi Zhao, dan Jianhua Zou (2021)	Peningkatan masalah waktu dan solusi optimal	<i>Vehicle routing problem</i>	<i>Deep reinforcement learning dan local search</i>
<i>Deep Reinforcement learning Approach to Solve Dynamic Vehicle Routing Problem with Stochastic Customers</i>	Waldy Joe, Hoong Chuin Lau (2020)	Pelanggan yang diketahui dan yang tidak diketahui, serta terdapat jendela waktu	<i>dynamic vehicle routing problem (DVRP)</i>	<i>deep reinforcement learning dan Simulated annealing</i>
<i>Deep Reinforcement learning for Solving the Heterogeneous Capacitated Vehicle Routing Problem</i>	Jingwen Li, Yining Ma, Ruize Gao, Zhiguang Cao, Andrew Lim, Wen Song, Jie Zhang (2021)	Pengambilan Keputusan pada Pemilihan kendaraan	<i>heterogeneous capacitated vehicle routing problem (HCVRP)</i>	<i>Deep reinforcement learning</i>
<i>A novel reinforcement learning based hyper-heuristic for heterogeneous vehicle routing problem</i>	Wei Qin, Zilongzhan, Zizhao Huang, Haozhe Huang (2021)	Rute kendaraan dengan kapasitas kendaraan yang berbeda	<i>heterogeneous vehicle routing problem (HVRP)</i>	<i>mengembangkan reinforcement learning based hyper-heuristic (RLHH)</i>
<i>Efficiently Solving the Practical Vehicle Routing Problem: A Novel Joint Learning Approach</i>	Lu Duan, Yang Zhan, Haoyuan Hu, Yu Gong, Jiangwen Wei, Xiaodong Zhang, Yinghui Xu (2020)	Koordinasi dan permintaan serta jarak antar node atau pelanggan	<i>Vehicle routing problem</i>	<i>Deep reinforcement learning dengan GCN dan fitur edge</i>
<i>Learning Improvement Heuristic for</i>	Yaoxin Wu, Wen Song, Zhiguang Cao,	Peningkatan solusi awal	<i>Travelling salesman problem</i>	<i>Arsitektur baru berdasarkan self attention, dan</i>

Judul	Penulis	Fokus	Permasalahan	Metode
<i>Solving Routing Problems</i>	Jie Zhang, Andrew Lim (2021)	dalam optimasi rute	(TSP) dan <i>capacited vehicle routing problem</i> (CVRP)	<i>kerangka kerja Reinforcement learning</i>
<i>Reinforcement learning with Combinatorial Actions: An Application to Vehicle Routing</i>	Arthur Delarue, Ross Anerson, Christian Tjandraatmadja (2020)	Fokus pada kapasitas yang terbatas saat pengiriman	<i>Capacited vehicle routing problem</i> (CVRP)	<i>Value function based deep reinforcement learning</i>
<i>Hybrid genetic search for dynamic vehicle routing with time windows</i>	Muhammed Ghannam, Ambros Gleixner (2023)	Fokus pada pelanggan datang secara berkelompok dan menentukan pelanggan mana yang akan dikirim	<i>Dynamic Vehicle Routing Problem with Time Windows</i> (DVRPTW)	<i>Dynamic hybrid genetic search (DHGS)</i>
<i>A hybrid brain storm optimization algorithm for dynamic vehicle routing problem with time windows</i>	Mingde Liu, Qi Zhao, Qi Song, Yingbin Zhang (2023)	Kemunculan pelanggan baru yang tidak pasti dan meminimalkan pelanggan yang tidak terlayani	<i>Dynamic Vehicle Routing Problem with Time Windows</i> (DVRPTW)	<i>Hybrid Brain Storm Optimization (BSO)</i>
<i>A Hybrid BSO-ACO for Dynamic Vehicle Routing Problem on Real-World Road Networks</i>	Mingde Liu, Qi Song, Qi Zhao, Ling Li, Zhiming Yang, Yingbin Zhang (2022)	Fokus pada jaringan jalan raya	<i>Dynamic Vehicle Routing Problem with Time Windows</i> (DVRPTW)	<i>algoritma hybrid BSO-ACO</i>
<i>Attention, Learn to Solve Routing Problems</i>	Kool, W., Van Hoof, H., & Welling, M. (2018)	Optimasi permasalahan perutean	<i>Travelling Salesman Problem (TSP) dan Vehicle Routing Problem (VRP)</i>	<i>Attention model untuk urutan rute dan algoritma reinforce dengan baseline rollout greedy</i>

Judul	Penulis	Fokus	Permasalahan	Metode
<i>RP-DQN: An application of Q-Learning to Vehicle Routing Problems</i>	Bdeir, A., Boeder, S., Dernedde, T., Tkachuk, K., Falkner, J. K., & Schmidt-Thieme, L.	Mengatasi permasalahan rute	<i>capacited vehicle routing problem (CVRP), Multi-Depot Vehicle Routing Problem (MDVRP)</i>	<i>routing problem deep q-network (RPDQN)</i>

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 menunjukan hasil yang optimal.

Pada penelitian selanjutnya fokus penelitian pada model masalah *dynamic vehicle routing problem with time windows* (DVRPTW) dengan fokus terhadap ketidakpastian lalu lintas dan pada proses pengiriman ke pelanggan terdapat jendela waktu atau batasan waktu pengiriman. Pada penelitian terdahulu hanya fokus pada salah satu saja seperti hanya fokus pada pelanggan yang tidak pasti atau

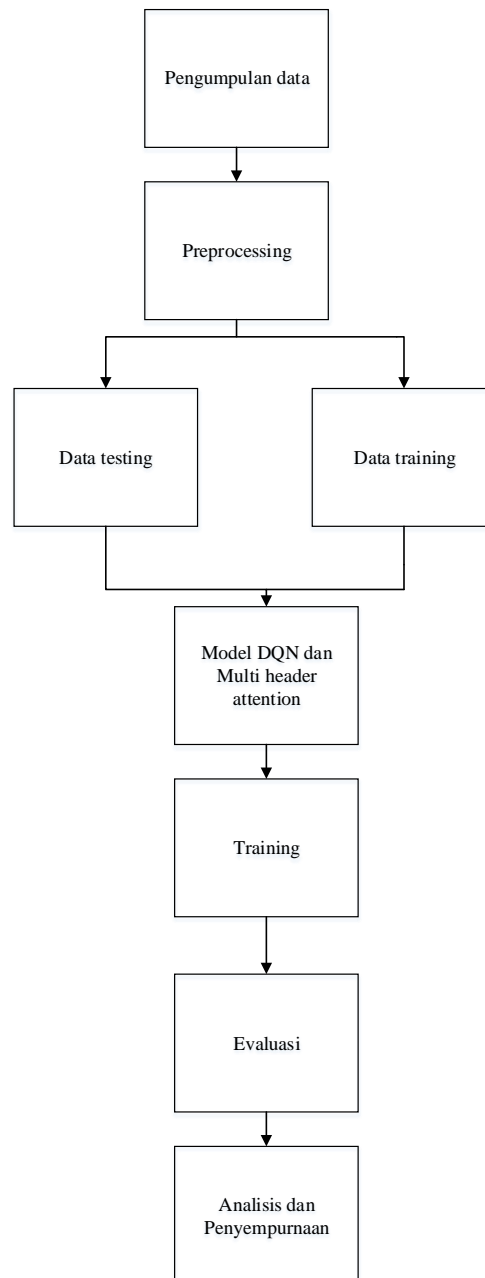
ketidakpastian jalan raya. Penyelesaian dilakukan dengan menggunakan *deep reinforcement learning* pada hal ini menggunakan metode *deep Q-network* (DQN) dengan menggabungkan *multi-header attention* kedalam arsitektur DQN.

BAB 3

METODOLOGI PENELITIAN

3.1 Kerangka Umum Penelitian

Berikut ini merupakan kerangka penelitian yang menjelaskan tahapan yang dilakukan dalam penelitian ini. Berikut gambar 3.1 diagram alir penelitian



*Gambar 3. 1*Diagram Alir Penelitian

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. Data ini harus mencerminkan situasi dunia nyata untuk memastikan bahwa model yang dikembangkan dapat diaplikasikan secara praktis. Data yang didapatkan berjumlah 12 ribu data terkait distribusi barang. Berikut merupakan dataset yang didapatkan.

tglcycle	tname	tadd1	tadd2	tadd3	tadd4	Alamat lengkap	tadd5	tadd6	tadd7	status	nama p	keterangan	tanggal
20211101	FITRI AFRI FITRI AFRI	JL. WARGA/ PEJATEN E PASAR MIJL. WARGA GG U NO 63	PEJATEN BARATPASAR MINGGU	DKI JAKAR	1		1			Sukses	Kotak Suri	Kotak Suri	09/12/2021
20211101	ALFIN RAC ALFIN RAC KOMP. PU 023/006	TANGERANG KOMP. PUSPIPTEK BLOK III K/5023/006	TANGERANG	BANTEN	10000		10000			Sukses	Kotak Suri	Kotak Suri	07/12/2021
20211101	ANTON RI ANTON RI JI. SUBUR DURI PULO	JAKARTA JI. SUBUR RAYA NO. 52 RT. 03/06	DURI PULO, GAMBIRJAKA	DKI	10000		10000			Sukses	Kotak Suri	Kotak Suri	07/12/2021
20211101	SENI	SENI KEPA DUR RW 10 DKI	DKI KEPA DURI BLOK A 3 RT 4RW 10 DKI	JKTDKI	DKI JAKAR	10000	10000			Sukses	Kotak Suri	Kotak Suri	10/12/2021
20211101	MUHAMM MUHAMM JI. ANYER	006/009 JAKARTA JI. ANYER GG XV 1006/009	JAKARTA	DKI JAKAR	10000		10000			Sukses	HAMIM	Keluarga	07/12/2021
20211101	ROHIMAT ROHIMAT JI. HIDUP E GANDARI	JAKARTA JI. HIDUP BARU NO. 18	GANDARIA UTARA, KEBAYORAN BAF	DKI JAKAR	10000		10000			Sukses	ROHIMAT	Keluarga	07/12/2021
20211101	SURYANI SURYANI JALAN KES	JAKARTA JALAN KES 5 RT 008/006	JALAN KES 5 RT 008/006	DKI JAKAR	10000		10000			Sukses	Kotak Suri	Kotak Suri	07/12/2021
20211101	SURYANI SURYANI JALAN KES	JAKARTA JALAN KES 5 RT 008/006	JALAN KES 5 RT 008/006	DKI JAKAR	10000		10000			Sukses	Kotak Suri	Kotak Suri	07/12/2021
20211101	FIKRI ZULFIKRI ZULFIKRI JALAN KRJ	JAKARTA JALAN KRAMAT SENTIONG001/002	JALAN KRAMAT SENTIC	DKI JAKAR	10000		10000			Sukses	FIKRI	Keluarga	07/12/2021
20211101	IR.H.AGUS IR.H.AGUS JI. KECAPI	KEL. JAGAH JAKARTA JI. KECAPI V NO 948 RT5	RWSKEL JAGAKARSAN KEC. JAGAK	DKI JAKAR	10000		10000			Sukses	AGUS	Keluarga	09/12/2021
20211101	ROUDHOT ROUDHOT PEJATEN E PASAR MIJ	JAKARTA PEJATEN BARAT RT 06/08	PASAR MINGGUJAKARTA	DKI JAKAR	10000		10000			Sukses	ROUDHOT	Keluarga	09/12/2021
20211101	SYAMSUL SYAMSUL KS TUBUN PETAMBU	JAKARTA KS TUBUN 24	PETAMBURANJAKARTA	DKI JAKAR	10000		10000			Sukses	Kotak Suri	Kotak Suri	09/12/2021
20211101	DIAN SITA DIAN SITA JI. PULO A	PULOGAD JAKARTA JI. PULO ASEMIN UTARA XI NO 35	PULOGADUNGJAKARTA	DKI JAKAR	10000		10000			Sukses	DIAN	Keluarga	09/12/2021
20211101	MUHAMM MUHAMM TEBET TIM RT/RW 01	JAKARTA TEBET TIMUR IB/8RT/RW 011	005JAKARTA	DKI JAKAR	10000		10000			Sukses	AGUS	Keluarga	10/12/2021
20211101	RISWAND RISWAND JI. JEND.S. KAV.82 SL	JAKARTA JI. JEND.S. PARMANKAV.82	SLIPIJAKARTA BARAT	DKI JAKAR	10000		10000			Sukses	Kotak Suri	Kotak Suri	09/12/2021
20211101	ANTHONIANTHONIJL PETOJO PETOJO U	JAKARTA JI. PETOJO UTARA VII NO 23	RT015/003PETOJO UTARA, GA	DKI JAKAR	10000		10000			Sukses	ANTHONIJ	Keluarga	09/12/2021
20211101	RHOADES RHOADES JALAN KUI KUNINGAI	JAKARTA JALAN KUNINGANKUNINGANJAKARTA SELATAN	DKI JAKAR	10000		10000				Sukses	Kotak Suri	Kotak Suri	09/12/2021
20211101	RHOADES RHOADES JALAN KUI KUNINGAI	JAKARTA JALAN KUNINGANKUNINGANJAKARTA SELATAN	JAKARTA	10000		10000				Sukses	Kotak Suri	Kotak Suri	09/12/2021

Gambar 3. 2 Dataset Yang Didapatkan

3.3 Preprocessing Data

Preprocessing data digunakan untuk mempersiapkan data mentah ke dalam data yang dibutuhkan oleh pengguna. Pada proses ini dilakukan beberapa tahapan yaitu *cleansing*, *validasi data*, *featur extraction* dan *featur selection*.

1. Cleansing

Pada tahap ini dilakukan untuk menghapus atau memperbaiki data yang tidak akurat, hilang, duplikat, atau tidak konsisten.

2. Validasi data

Memastikan bahwa data sudah sesuai dengan kriteria yang digunakan seperti konsistensi, kelengkapan, dan keakuratan data.

3. Featur extraction

Mengubah atau mengekstraksi fitur baru dari dataset yang sudah ada. Pada hal ini menambahkan latitude dan longitude.

4. *Featur selection*

Memilih fitur yang paling relevan untuk diolah seperti pelanggan, kota, alamat penerima, latitude dan longitude.

Berdasarkan tahap diatas berikut hasil dari preprocessing yang sudah dilakukan.

Pelanggan	City	Street Address	Latitude	Longitude
FITRI AFRIANI	Jakarta	Gg. U No.63, RT.1/RW.4, Tegal Parang, Kec. Mampang Prpt., Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12790, Indonesia	-6,2505704	106,8346648
ALFIN RACHMANDA	Jakarta	Jl. Subur No.6 52, RT.12/RW.6, Duri Pulo, Kecamatan Gambir, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta 10140, Indonesia	-6,1654434	106,8031141
ANTON RIYADI	Jakarta	Jl. Anyer, RW.9, Menteng, Kec. Menteng, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta, Indonesia	-6,2032217	106,8444915
SENI	Jakarta	Jl. Hidup Baru No.18, RT.11/RW.3, Gandaria Utara, Kec. Kby. Baru, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12140, Indonesia	-6,2601510	106,7923741
MUHAMMAD HAMIM HARAHAHAP	Jakarta	Jl. Kramat Raya, Kec. Senen, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta, Indonesia	-6,1858778	106,8449292
ROHIMAT	Jakarta	Jl. Kecapi V, Jagakarsa, Kec. Jagakarsa, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta, Indonesia	-6,3181291	106,8204498
SURYANI SIREGAR	Jakarta	Jl. K.S. Tubun No.24, RT.10/RW.6, Petamburan, Kecamatan Tanah Abang, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta 11410, Indonesia	-6,2001986	106,8010076
SURYANI SIREGAR	Jakarta	Jl. Pulo Asem Utara XI No.35, RT.2/RW.2, Jati, Kec. Pulo Gadung, Kota Jakarta Timur, Daerah Khusus Ibukota Jakarta 13220, Indonesia	-6,1888883	106,8957327

Gambar 3. 3 Hasil preprocessing

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.

2. Desain model DQN dan *Multi header-attention*

Implementasi DQN melibatkan beberapa komponen utama seperti:

Jaringan Q: Jaringan ini memperkirakan nilai Q untuk setiap aksi dari *state* tertentu. Dalam kasus DVRPTW, input bisa berupa representasi dari *state* saat ini seperti lokasi kendaraan, status pengiriman. *outputnya* adalah nilai Q untuk setiap kemungkinan aksi seperti 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 (*hidden layers*) 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.

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

Pada proses pelatihan 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. 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, 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.

Tabel 3.1 Jadwal Penelitian

No.	Uraian Kegiatan	2023				2024											
		9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
1.	Penyusunan Proposal																
2.	Uji Kualifikasi																
3.	Evaluasi Progres Pertama																
4.	Paper Pertama																
5.	Evaluasi Progres Kedua																
6.	Paper Kedua																
No.	Uraian Kegiatan	2025												2026			
		1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4
7.	Paper Ketiga																
8.	Evaluasi RKP																
9.	Sidang Tertutup																
10.	Sidang Terbuka																

DAFTAR PUSTAKA

- Abdirad, M., Krishnan, K., & Gupta, D. (2022). Three-stage algorithms for the large-scale dynamic vehicle routing problem with industry 4.0 approach. *Journal of Management Analytics*, 9(3), 313-329. <https://doi.org/10.1080/23270012.2022.2113161>
- Arif, T. M., & Rahim, M. A. (2024). *Deep Learning for Engineers*. CRC Press.
- Baan, J., ter Hoeve, M., van der Wees, M., Schuth, A., & de Rijke, M. J. a. p. a. (2019). Understanding multi-head attention in abstractive summarization.
- Bdeir, A., Boeder, S., Dervede, T., Tkachuk, K., Falkner, J. K., & Schmidt-Thieme, L. (2021). RP-DQN: An application of Q-learning to vehicle routing problems. In *KI 2021: Advances in Artificial Intelligence: 44th German Conference on AI, Virtual Event, September 27–October 1, 2021, Proceedings 44* (pp. 3-16). Springer International Publishing.
- Cordeau, J.-F., Laporte, G., Savelsbergh, M. W., Vigo, D. J. H. i. o. r., & science, m. (2007). Vehicle routing. *14*, 367-428.
- Dantzig, G. B., & Ramser, J. H. J. M. s. (1959). The truck dispatching problem. *6*(1), 80-91.
- El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? In I. El Naqa, R. Li, & M. J. Murphy (Eds.), *Machine Learning in Radiation Oncology: Theory and Applications* (pp. 3-11). Springer International Publishing. https://doi.org/10.1007/978-3-319-18305-3_1
- Ghannam, M., & Gleixner, A. J. a. p. a. (2023). Adapting Hybrid Genetic Search for Dynamic Vehicle Routing.
- He, Z., Chen, L., & Liu, B. J. I. D. T. (2024). Application of integrating reinforcement learning and intelligent scheduling in logistics distribution. (Preprint), 1-18.
- Hong, Z.-W., Su, S.-Y., Shann, T.-Y., Chang, Y.-H., & Lee, C.-Y. J. a. p. a. (2017). A deep policy inference q-network for multi-agent systems.
- Joe, W., & Lau, H. C. (2020). Deep reinforcement learning approach to solve dynamic vehicle routing problem with stochastic customers. *Proceedings of the international conference on automated planning and scheduling*,
- Karthikeyan, J., Su Hie, T., & Yu Jin, N. (Eds.). (2021). *Learning Outcomes of Classroom Research*. L Ordine Nuovo Publication.
- Kelleher, J. D. (2019). *Deep learning*. MIT press.
- Kober, J., Bagnell, J. A., & Peters, J. J. T. I. J. o. R. R. (2013). Reinforcement learning in robotics: A survey. *32*(11), 1238-1274.
- Kool, W., Van Hoof, H., & Welling, M. (2018). Attention, learn to solve routing problems!. *arXiv preprint arXiv:1803.08475*.
- Li, J., Ma, Y., Gao, R., Cao, Z., Lim, A., Song, W., & Zhang, J. J. I. T. o. C. (2021). Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem. *52*(12), 13572-13585.
- Li, Y. J. a. p. a. (2017). *Deep reinforcement learning: An overview*.

- Liu, M., Song, Q., Zhao, Q., Li, L., Yang, Z., & Zhang, Y. (2022). A Hybrid BSO-ACO for Dynamic Vehicle Routing Problem on Real-World Road Networks. *IEEE Access*, 10, 118302-118312. <https://doi.org/10.1109/ACCESS.2022.3221191>
- Liu, M., Zhao, Q., Song, Q., & Zhang, Y. (2023). A Hybrid Brain Storm Optimization Algorithm for Dynamic Vehicle Routing Problem With Time Windows. *IEEE Access*, 11, 121087-121095. <https://doi.org/10.1109/ACCESS.2023.3328404>
- Liu, S., See, K. C., Ngiam, K. Y., Celi, L. A., Sun, X., & Feng, M. J. J. o. m. I. r. (2020). Reinforcement learning for clinical decision support in critical care: comprehensive review. 22(7), e18477.
- Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.
- Mousavi, S. S., Schukat, M., & Howley, E. (2018). Deep reinforcement learning: an overview. Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016: Volume 2,
- Ni, Q., & Tang, Y. J. S. (2023). A Bibliometric Visualized Analysis and Classification of Vehicle Routing Problem Research. 15(9), 7394.
- PK, F. A. J. S. i. n. a. I. i. h. w., perseverance, learning, studying, sacrifice, & most of all, l. o. w. y. a. d. o. l. t. d. (1984). What is artificial intelligence? , 65.
- Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: a modern approach*. Pearson.
- Saxe, A., Nelli, S., & Summerfield, C. (2021). If deep learning is the answer, what is the question?. *Nature Reviews Neuroscience*, 22(1), 55-67.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., . . . Lanctot, M. J. n. (2016). Mastering the game of Go with deep neural networks and tree search. 529(7587), 484-489.
- Su, Y., Liu, J., Xiang, X., & Zhang, X. (2021). A responsive ant colony optimization for large-scale dynamic vehicle routing problems via pheromone diversity enhancement. *Complex & Intelligent Systems*, 7(5), 2543-2558. <https://doi.org/10.1007/s40747-021-00433-7>
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- Szepesvári, C. (2022). *Algorithms for reinforcement learning*. Springer Nature.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. J. A. i. n. i. p. s. (2017). Attention is all you need. 30.
- Xin, L., Song, W., Cao, Z., & Zhang, J. (2021, May). Multi-decoder attention model with embedding glimpse for solving vehicle routing problems. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 13, pp. 12042-12049).
- Wang, P. J. J. o. A. G. I. (2019). On defining artificial intelligence. 10(2), 1-37.