BAB II
KAJIAN PUSTAKA

Bab ini menyajikan landasan konseptual dan teoretis yang menjadi dasar bagi keseluruhan penelitian. Pembahasan dibagi menjadi dua bagian utama. Bagian pertama, Tinjauan Pustaka, menyajikan analisis kritis terhadap evolusi sistem pengendalian sinyal lalu lintas, mulai dari metode konvensional hingga pendekatan canggih berbasis *deep reinforcement learning* (DRL). Tinjauan ini secara sistematis mengidentifikasi pencapaian, tantangan, dan celah penelitian (*research gaps*) dalam literatur yang ada, terutama terkait dengan optimisasi multi-objektif, skalabilitas multi-agen, dan stabilitas algoritma. Bagian kedua, Landasan Teori, menguraikan secara mendalam prinsip-prinsip matematis dan komputasional yang membentuk blok-blok bangunan dari kerangka kerja yang diusulkan. Ini mencakup formalisasi *markov decision process* (MDP), evolusi arsitektur *deep Q-Network* (DQN), serta konsep-konsep kunci seperti *multi-objective deep reinforcement learning* (MODRL) dan *curriculum learning* (CL). Secara keseluruhan, bab ini bertujuan untuk memposisikan kebaruan penelitian ini dalam konteks akademis yang lebih luas dan memberikan justifikasi teoretis yang kuat untuk setiap pilihan metodologis yang akan dirinci pada bab berikutnya.

2.1 Tinjauan Pustaka

Tinjauan pustaka ini secara sistematis memetakan lanskap penelitian dalam sistem pengendalian sinyal lalu lintas adaptif (ATSC), dimulai dari pendekatan konvensional hingga metode berbasis kecerdasan buatan yang paling canggih. Analisis ini bertujuan untuk mengidentifikasi pencapaian utama, tantangan yang belum terpecahkan, dan celah penelitian yang ada, sehingga dapat memposisikan kontribusi disertasi ini secara jelas.

2.1.1 Analisis Bibliometrik Tren Penelitian ATSC

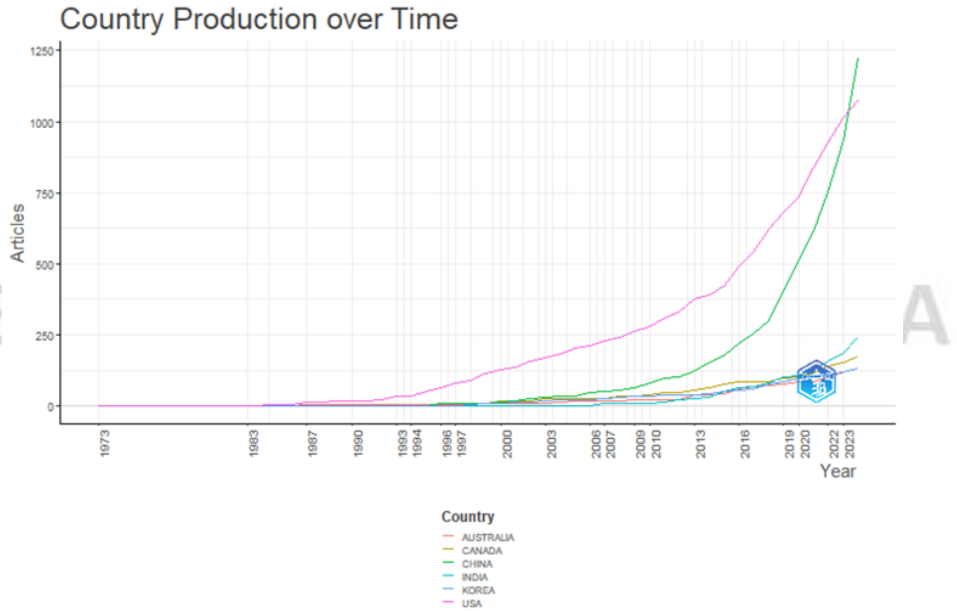
Sebagai langkah awal untuk memetakan lanskap penelitian ATSC secara komprehensif, sebuah analisis bibliometrik dilakukan sebelum tinjauan pustaka kualitatif yang lebih mendalam. Pendekatan kuantitatif ini bertujuan untuk

memberikan gambaran objektif mengenai evolusi, struktur intelektual, dan arah masa depan dari bidang riset ATSC. Urgensi penelitian ATSC didorong oleh dampak signifikan kemacetan lalu lintas perkotaan, yang secara global menyebabkan kerugian ekonomi tahunan melebihi satu triliun dolar AS (Chen dkk., 2023). ATSC menawarkan pendekatan yang lebih adaptif melalui optimalisasi waktu sinyal secara dinamis berdasarkan data lalu lintas *real-time* (Pasquale dkk., 2015), Pendekatan ini memanfaatkan sistem sensor real-time dan algoritma cerdas untuk menyesuaikan fase sinyal terhadap perubahan kondisi lalu lintas, sehingga dapat meminimalkan waktu tunda, panjang antrean, konsumsi bahan bakar, dan emisi kendaraan (Wei dkk., 2019). Meskipun sistem perintis seperti SCATS dan SCOOT telah meletakkan dasar teknis yang penting (Mirchandani & Head, 2001), kemajuan pesat dalam kecerdasan buatan, terutama *reinforcement learning* (RL), serta teknologi *connected vehicles* (CVs), telah secara fundamental mengubah paradigma menuju sistem kontrol yang lebih terdesentralisasi, kooperatif, dan cerdas (Liang dkk., 2019; Ye dkk., 2020). Kendati demikian, tinjauan sistematis yang secara kuantitatif memetakan keseluruhan ekosistem riset ATSC masih terbatas. Oleh karena itu, studi bibliometrik ini dilakukan untuk mengisi celah tersebut dan memberikan fondasi berbasis data bagi analisis selanjutnya.

Metodologi analisis bibliometrik ini dirancang secara sistematis untuk menjawab tiga pertanyaan riset (RQ) fundamental: (RQ1) Siapa saja aktor utama (penulis, institusi, negara) yang mendominasi produksi riset ATSC dan di mana pusat-pusat penelitian terkonsentrasi? (RQ2) Apa saja tema-tema penelitian dominan yang membentuk struktur konseptual bidang riset ini? (RQ3) Apa saja tren penelitian yang sedang berkembang dan arah riset masa depan yang potensial? Untuk menjawab pertanyaan ini, data publikasi ilmiah diekstraksi dari basis data Scopus pada tanggal 7 Oktober 2025. Proses pencarian menggunakan *string* TITLE-ABS-KEY(Adaptive Traffic Signal Control) dengan filter khusus pada jenis dokumen (*article*), bahasa (Inggris), dan rentang waktu publikasi dari tahun 1974 hingga 2023. Filter ini menghasilkan dataset final sebanyak 1.287 artikel yang relevan. Analisis data selanjutnya dilakukan dengan memanfaatkan perangkat lunak *VOSviewer*, sebuah *tool* standar dalam studi bibliometrik, untuk membangun dan

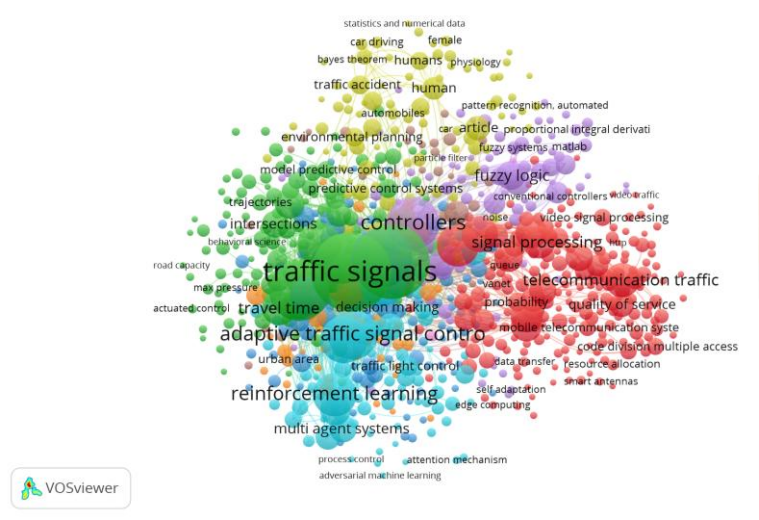
memvisualisasikan jaringan ko-sitasi, ko-penulis, dan ko-okurensi kata kunci. Selain itu, analisis statistik deskriptif juga digunakan untuk mengidentifikasi tren temporal dalam volume produksi riset.

Hasil analisis bibliometrik terkait RQ1 menunjukkan perkembangan produksi publikasi ATSC berdasarkan negara dari waktu ke waktu. Sebagaimana ditampilkan pada Gambar 2.1, jumlah publikasi mengalami peningkatan yang semakin signifikan, terutama sejak dekade terakhir. Amerika Serikat dan Tiongkok menjadi dua negara dengan kontribusi publikasi tertinggi dalam penelitian ATSC. Amerika Serikat menunjukkan pertumbuhan yang relatif konsisten sejak periode awal, sedangkan Tiongkok mengalami peningkatan yang lebih tajam dalam beberapa tahun terakhir dan menjadi negara dengan jumlah publikasi tertinggi pada akhir periode pengamatan. Negara lain, seperti India, Korea Selatan, Kanada, dan Australia, juga menunjukkan peningkatan jumlah publikasi, meskipun jumlahnya masih berada di bawah Amerika Serikat dan Tiongkok. Temuan ini menunjukkan bahwa penelitian ATSC semakin berkembang dan memperoleh perhatian yang semakin besar di berbagai negara.



Gambar 2.1 Produksi Publikasi ATSC Berdasarkan Negara

Terkait RQ2 mengenai struktur konseptual penelitian ATSC, analisis ko-okurensi kata kunci dilakukan menggunakan VOSviewer untuk memetakan keterkaitan antartema berdasarkan frekuensi kemunculan bersama dalam publikasi yang dianalisis. Pada Gambar 2.2, ukuran simpul menunjukkan tingkat kemunculan suatu kata kunci, sedangkan kedekatan posisi dan hubungan antarsimpul menggambarkan keterkaitan tematik antarkata kunci.

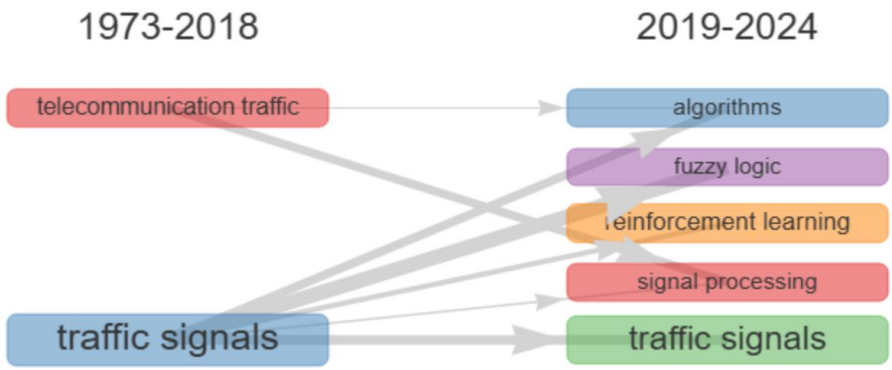


Gambar 2.2 Visualisasi Jaringan Ko-Okurensi Kata Kunci dalam Penelitian ATSC

Hasil visualisasi menunjukkan bahwa kata kunci "traffic signals" merupakan simpul yang paling dominan dan berada pada posisi sentral dalam jaringan. Kata kunci tersebut terhubung dengan sejumlah tema penting lainnya, seperti "controllers", "adaptive traffic signal control", "reinforcement learning", "intersections", "travel time", "predictive control systems", "signal processing", dan "telecommunication traffic". Hubungan tersebut menunjukkan bahwa riset ATSC berkembang melalui integrasi antara kajian pengendalian sinyal lalu lintas, sistem kontrol adaptif, optimisasi kinerja lalu lintas, serta penerapan metode kecerdasan buatan.

Selain itu, kata kunci "reinforcement learning" dan "multi agent systems" terlihat sebagai bagian dari jaringan tema yang berkaitan dengan pengembangan sistem pengendalian adaptif berbasis pembelajaran. Meskipun demikian, visualisasi ini tidak digunakan untuk menyimpulkan bahwa DRL dan MARL merupakan

kluster yang paling dominan secara eksplisit. Kedua pendekatan tersebut lebih tepat diposisikan sebagai arah metodologis yang berkembang dan memiliki keterkaitan dengan tema utama pengendalian sinyal lalu lintas adaptif.



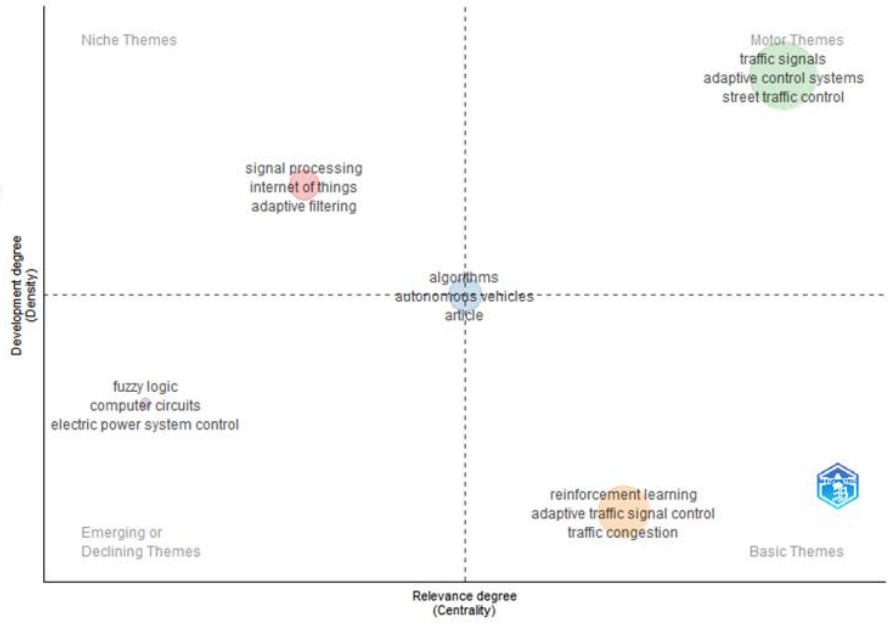
Gambar 2.3 Evolusi Tema Penelitian ATSC pada Periode 1973–2018 dan 2019–2024

Berikutnya mengenai evolusi tema penelitian ATSC, Gambar 2.3 menunjukkan perubahan fokus kajian dari periode 1973–2018 menuju periode 2019–2024. Pada periode awal, tema penelitian masih terkonsentrasi pada dua topik utama, yaitu “traffic signals” dan “telecommunication traffic”. Kedua tema tersebut mencerminkan fokus awal penelitian pada pengendalian sinyal lalu lintas serta dukungan sistem komunikasi dalam pengelolaan arus kendaraan.

Pada periode 2019–2024, struktur tema berkembang menjadi lebih beragam. Tema “traffic signals” tetap bertahan sebagai topik utama, tetapi terhubung dengan sejumlah pendekatan metodologis yang lebih spesifik, yaitu “algorithms”, “fuzzy logic”, “reinforcement learning”, dan “signal processing”. Perubahan tersebut menunjukkan bahwa penelitian ATSC tidak lagi hanya berfokus pada pengaturan sinyal secara konvensional, tetapi mulai mengintegrasikan metode komputasional dan kecerdasan buatan untuk meningkatkan kemampuan adaptasi sistem terhadap kondisi lalu lintas yang dinamis.

Kemunculan tema “reinforcement learning” pada periode 2019–2024 menunjukkan meningkatnya perhatian terhadap pendekatan berbasis pembelajaran dalam pengambilan keputusan sinyal lalu lintas. Sementara itu, tema “algorithms”,

"fuzzy logic", dan "signal processing" memperlihatkan bahwa pengembangan ATSC juga didukung oleh optimisasi algoritmik, pengambilan keputusan berbasis logika, serta pemrosesan informasi lalu lintas. Dengan demikian, evolusi tema pada Gambar 2.3 menggambarkan pergeseran dari kajian pengendalian sinyal yang bersifat umum menuju pendekatan ATSC yang lebih adaptif, terintegrasi, dan berbasis kecerdasan komputasional.



Gambar 2.4 Peta Tematik Penelitian ATSC Berdasarkan Tingkat Sentralitas dan Kepadatan Tema

Gambar 2.4 menunjukkan peta tematik penelitian ATSC berdasarkan dua dimensi, yaitu tingkat relevansi (*centrality*) pada sumbu horizontal dan tingkat perkembangan (*density*) pada sumbu vertikal. Nilai *centrality* menunjukkan tingkat keterhubungan suatu kelompok tema dengan keseluruhan bidang penelitian, sedangkan nilai *density* menunjukkan tingkat perkembangan internal tema tersebut. Berdasarkan kedua dimensi tersebut, tema penelitian dipetakan ke dalam empat kuadran, yaitu *motor themes*, *basic themes*, *niche themes*, serta *emerging or declining themes*.

Pada kuadran kanan atas, yaitu *motor themes*, terdapat tema *traffic signals*, *adaptive control systems*, dan *street traffic control*. Tema-tema tersebut memiliki tingkat relevansi dan perkembangan yang tinggi sehingga dapat diposisikan sebagai

tema penggerak dalam penelitian ATSC. Keberadaannya menunjukkan bahwa pengendalian sinyal lalu lintas adaptif tetap menjadi bagian inti yang memiliki keterkaitan tematik kuat dengan berbagai kajian lain dalam bidang ini.

Pada kuadran kanan bawah, yaitu *basic themes*, terdapat tema *reinforcement learning*, *adaptive traffic signal control*, dan *traffic congestion*. Tema-tema tersebut memiliki relevansi yang tinggi terhadap keseluruhan bidang penelitian, tetapi tingkat perkembangan internalnya masih lebih rendah dibandingkan dengan tema pada kuadran *motor themes*. Posisi tersebut menunjukkan bahwa *reinforcement learning* telah menjadi salah satu fondasi penting dalam pengembangan ATSC, sekaligus masih membuka peluang penelitian lanjutan untuk memperkuat metode, arsitektur model, dan penerapannya pada skenario lalu lintas yang lebih kompleks.

Pada kuadran kiri atas, yaitu *niche themes*, terdapat tema *signal processing*, *internet of things*, dan *adaptive filtering*. Tema-tema tersebut telah berkembang secara relatif kuat dalam kelompok kajiannya, tetapi tingkat keterhubungannya dengan keseluruhan domain ATSC masih lebih terbatas. Posisi ini menunjukkan adanya bidang spesialisasi yang dapat mendukung pengembangan sistem ATSC, terutama melalui pemrosesan data sensor dan integrasi perangkat cerdas.

Pada kuadran kiri bawah, yaitu *emerging or declining themes*, terdapat tema *fuzzy logic*, *computer circuits*, dan *electric power system control*. Tema-tema tersebut memiliki tingkat sentralitas dan kepadatan yang relatif rendah. Namun, peta tematik ini belum dapat digunakan secara langsung untuk menentukan apakah tema tersebut merupakan topik baru yang sedang tumbuh atau topik lama yang mulai mengalami penurunan perhatian. Penentuan tersebut memerlukan analisis temporal tambahan berdasarkan tahun publikasi dan perkembangan frekuensi kata kunci.

Selain keempat kelompok tersebut, tema *algorithms* dan *autonomous vehicles* berada di sekitar garis pembatas antarkuadran. Posisi ini menunjukkan bahwa kedua tema tersebut memiliki karakter lintas bidang dan berpotensi berkembang menuju tema yang lebih sentral apabila semakin banyak penelitian mengintegrasikannya dengan ATSC.

Secara keseluruhan, Gambar 2.4 memperlihatkan bahwa penelitian ATSC tetap bertumpu pada pengendalian sinyal lalu lintas sebagai tema inti, tetapi juga berkembang menuju pendekatan berbasis *reinforcement learning*, integrasi perangkat cerdas, serta pemanfaatan algoritma adaptif untuk menangani kemacetan secara lebih dinamis.

Terkait RQ3 mengenai tren penelitian yang sedang berkembang dan arah riset masa depan yang potensial, hasil analisis bibliometrik menunjukkan adanya pergeseran perhatian menuju pendekatan pengendalian sinyal lalu lintas yang semakin adaptif dan berbasis kecerdasan komputasional. Sebagaimana ditampilkan pada Gambar 2.3, tema *traffic signals* tetap menjadi topik utama pada kedua periode pengamatan. Namun, pada periode 2019–2024 muncul diversifikasi tema yang lebih luas, meliputi *algorithms*, *fuzzy logic*, *reinforcement learning*, dan *signal processing*. Perubahan tersebut menunjukkan bahwa penelitian ATSC berkembang dari kajian pengaturan sinyal secara umum menuju pemanfaatan metode komputasional untuk mendukung pengambilan keputusan yang lebih responsif terhadap dinamika lalu lintas.

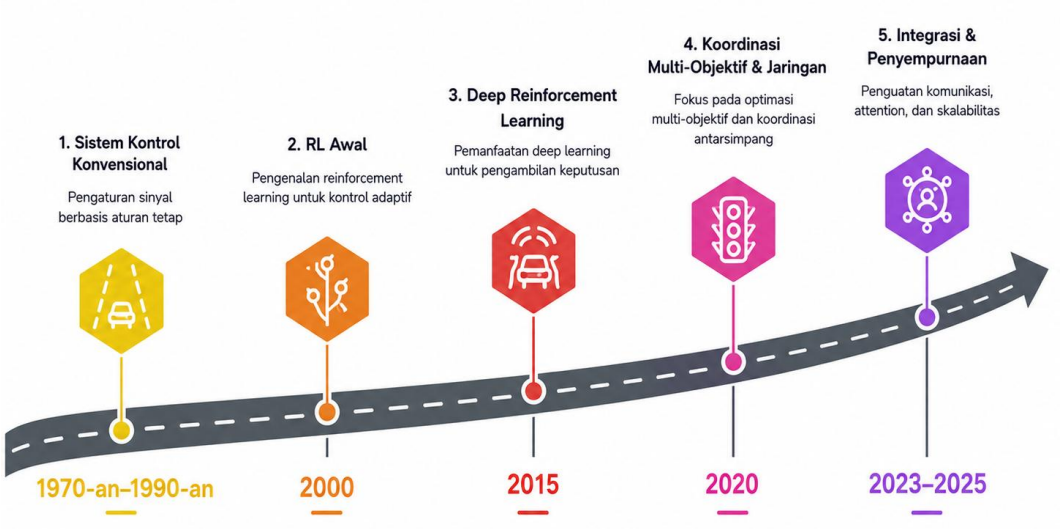
Temuan tersebut diperkuat oleh peta tematik pada Gambar 2.4. Tema *reinforcement learning*, *adaptive traffic signal control*, dan *traffic congestion* berada pada kuadran *basic themes*. Posisi ini menunjukkan bahwa ketiga tema tersebut memiliki tingkat relevansi yang tinggi terhadap keseluruhan bidang penelitian, tetapi masih memiliki peluang pengembangan internal yang luas. Sementara itu, tema *traffic signals*, *adaptive control systems*, dan *street traffic control* berada pada kuadran *motor themes*, yang menunjukkan bahwa pengendalian sinyal lalu lintas adaptif tetap menjadi penggerak utama dalam perkembangan riset ATSC.

Berdasarkan hasil tersebut, arah penelitian ATSC pada masa mendatang berpotensi berkembang melalui peningkatan kemampuan algoritma pembelajaran, penguatan mekanisme adaptasi terhadap perubahan kondisi lalu lintas, serta perluasan model dari pengendalian persimpangan tunggal menuju koordinasi jaringan persimpangan. Pengembangan lebih lanjut juga perlu mempertimbangkan integrasi informasi lalu lintas secara waktu nyata, efisiensi komputasi, skalabilitas

sistem, dan keseimbangan antara beberapa indikator kinerja. Dengan demikian, analisis bibliometrik ini memberikan dasar objektif untuk melanjutkan kajian kualitatif mengenai perkembangan DRL, optimisasi multi-objektif, *curriculum learning*, MARL, dan mekanisme *attention* pada bagian berikutnya.

2.1.2 Evolusi Sistem Pengendalian Sinyal Lalu Lintas

Perkembangan sistem pengendalian sinyal lalu lintas telah mengalami transformasi signifikan, dari sistem berbasis aturan tetap menuju sistem berbasis kecerdasan buatan yang adaptif, seperti yang diperlihatkan pada Gambar 2.5. Tahap awal evolusi ditandai oleh sistem *fixed-time control,* yang menggunakan parameter waktu lampu lalu lintas yang diatur secara periodik berdasarkan data historis arus kendaraan. Meskipun sederhana dan mudah diprediksi, sistem ini bersifat statis dan tidak mampu merespons perubahan volume lalu lintas secara *real-time*, sehingga sering menjadi sumber inefisiensi utama di persimpangan perkotaan (J. Gao dkk., 2017; Kuyer dkk., 2008a; Muresan dkk., 2019).



Gambar 2.5 Evolusi Historis Pengendalian Sinyal Lalu Lintas Adaptif

Sebagai respons awal terhadap keterbatasan ini, dikembangkan sistem *vehicle-actuated*, di mana sensor mendeteksi jumlah kendaraan dan menyesuaikan durasi fase lampu secara otomatis. Sistem ini menawarkan tingkat adaptivitas yang lebih baik dibandingkan *fixed-time*, namun masih terbatas pada optimisasi lokal dan kurang efektif dalam mengelola kondisi lalu lintas yang kompleks atau jenuh

(Rasheed dkk., 2020b; Sabit, 2023). Ketidakmampuan sistem konvensional ini dalam menangani sifat lalu lintas yang dinamis dan stokastik menjadi pendorong utama lahirnya paradigma *adaptive traffic signal control* (ATSC) yang lebih cerdas.

Perkembangan berikutnya adalah sistem *adaptive control* berbasis heuristik seperti SCOOT dan SCATS, yang mampu memperbarui waktu sinyal berdasarkan kondisi lalu lintas secara real-time. Meski adaptif, pendekatan ini masih memerlukan kalibrasi manual dan berbasis model, sehingga kurang optimal untuk jaringan persimpangan besar (Pol & Oliehoek, 2016). Dengan meningkatnya kompleksitas jaringan perkotaan, fokus penelitian beralih pada penerapan *reinforcement learning* (RL), di mana setiap lampu lalu lintas diperlakukan sebagai agen yang belajar dari pengalaman untuk meminimalkan waktu tunggu dan panjang antrean (Kuyer dkk., 2008b; Sutton & Barto, t.t.; Wiering, 2000). Pendekatan berbasis RL ini merepresentasikan awal sistem pengendalian sinyal otonom, di mana agen belajar kebijakan optimal melalui interaksi berulang dengan lingkungan. Studi awal menggunakan algoritma *tabular* Q-*learning* pada representasi keadaan sederhana berhasil menunjukkan peningkatan kinerja dibandingkan metode konvensional (Abdulhai dkk., 2003).

Integrasi *deep reinforcement learning* (DRL) menandai lompatan besar berikutnya. Algoritma seperti *deep* Q-*network* (DQN) (Mnih dkk., 2015) memungkinkan agen belajar langsung dari input sensor berdimensi tinggi tanpa rekayasa fitur manual, meningkatkan kapasitas agen dalam memahami kondisi lalu lintas kompleks. Banyak penelitian awal menggunakan DQN standar untuk optimasi efisiensi *single-objective* (J. Gao dkk., 2017; T. Kim & Lee, 2020; Muresan dkk., 2019), sedangkan DRL lanjutan mengadopsi DDQN, Dueling, dan D3QN untuk meningkatkan stabilitas, eksploitasi/eksplorasi, dan efisiensi (Cai & Wei, 2024; Gu dkk., 2020; Phan dkk., 2025). Seiring berkembangnya DRL, tantangan mulai muncul terkait *single-objective optimization*, di mana fokus pada efisiensi operasional sering mengabaikan keberlanjutan lingkungan (Gong dkk., 2020; Zhang dkk., 2024; Fang dkk., 2023). Pendekatan *multi-objective* mulai diterapkan, termasuk integrasi *reward shaping* dan *curriculum learning* untuk

mempercepat konvergensi pelatihan, meminimalkan masalah *cold-start*, dan mengurangi *reward hacking* (Mannion dkk., 2018; Zheng dkk., 2022).

Selain itu, penerapan DRL tunggal menghadapi kendala skalabilitas dan ketidakstabilan ketika beberapa agen berinteraksi. Konsep *multi-agent reinforcement learning* (MARL) diperkenalkan untuk memungkinkan kolaborasi antar-persimpangan. Setiap agen memiliki observasi lokal tetapi dapat berkoordinasi untuk mencapai tujuan global (Li dkk., 2021; Lowe dkk., 2020; Ma & Wu, 2020). Penelitian terbaru mengembangkan arsitektur MARL berbasis *graph attention networks* (GAT) untuk koordinasi spasial, meningkatkan efisiensi sistem lalu lintas (Benhamza dkk., 2024; Bie dkk., 2024; Fang dkk., 2023). Tren terbaru menekankan integrasi *attention mechanisms* dan *fog computing*. Contohnya, *GreenLight* (Tang & Baskiyar, 2024) menggunakan *attention-based* DRL dalam arsitektur *fog computing* dengan komunikasi V2I, memungkinkan pengolahan data terdistribusi dengan latensi rendah, mengurangi emisi dan konsumsi bahan bakar. Selain itu, *multi-agent soft actor-critic* dengan *attention* (Mao dkk., 2023) memperkuat koordinasi antar-agent di jaringan besar secara efisien.

Perkembangan pengendalian lalu lintas adaptif juga terlihat di Indonesia melalui implementasi *Area Traffic Control System* (ATCS) pada sejumlah kota dan daerah. ATCS pada umumnya digunakan untuk mengendalikan lampu lalu lintas secara terkoordinasi, mengoptimalkan kinerja jaringan jalan, memantau kondisi lalu lintas, serta membantu pengaturan prioritas pada persimpangan tertentu. Sebagai contoh, ATCS di beberapa daerah telah diarahkan untuk mengatur waktu sinyal secara responsif dan terkoordinasi, mengendalikan siklus lampu lalu lintas secara otomatis maupun manual dari pusat kendali, serta memanfaatkan data jumlah kendaraan, antrean, dan waktu tempuh sebagai dasar pengaturan lalu lintas.

Meskipun demikian, evaluasi perkembangan ATCS di Indonesia menunjukkan bahwa sebagian besar penerapannya masih berorientasi pada sistem kendali terpusat, pemantauan CCTV, dan penyesuaian siklus sinyal berdasarkan parameter lalu lintas tertentu. Kondisi tersebut menunjukkan adanya celah pengembangan menuju sistem pengendalian yang lebih cerdas, yaitu sistem yang tidak hanya

merespons kondisi lalu lintas secara real-time, tetapi juga mampu belajar dari pengalaman, mengoptimalkan beberapa tujuan sekaligus, dan melakukan koordinasi antar-persimpangan. Oleh karena itu, penelitian ini menjadi relevan karena mengembangkan kerangka MODRL-ATSC dan perluasannya ke MARL untuk menjawab kebutuhan pengendalian sinyal lalu lintas adaptif yang lebih otonom, efisien, dan sesuai dengan tantangan lalu lintas perkotaan di Indonesia.

Dengan demikian, evolusi metodologis ATSC menunjukkan pergeseran dari *fixed-time* dan *actuated systems* menuju DRL multi-objektif dan MARL dengan *curriculum learning*, yang mampu menangani kompleksitas, skalabilitas, stabilitas, dan koordinasi jaringan. Penelitian ini menempatkan diri pada posisi tersebut, mengembangkan MO-CL-D3QN + MARL untuk menyeimbangkan efisiensi operasional dan keberlanjutan lingkungan, sekaligus memastikan pelatihan stabil dan koordinasi antar-persimpangan optimal. Selanjutnya untuk melihat perbandingan penelitian terdahulu tentang metodologi kunci dan analisis relevansi dengan penelitian yang diusulkan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Perbandingan Penelitian Terdahulu

Penulis	**Fokus Utama**	***Key Methodology*/Kontribusi**	***Relevance to This Study***
Schumacher et al. (2023)	Tantangan desain *reward* untuk tujuan emisi CO_2.	Menunjukkan inefisiensi *reward* berbasis emisi langsung; merekomendasikan *proxy rewards.*	Memberikan justifikasi teoretis kuat untuk penggunaan metrik proksi (konsumsi bahan bakar) dan menyoroti kompleksitas desain *reward* lingkungan.
Mirbakhsh & Azizi (2024)	ATSC multi-objektif (keselamatan, efisiensi, dekarbonisasi).	Mengusulkan DRL-ATSC yang menyeimbangkan tiga tujuan utama secara simultan.	Memvalidasi bahwa pendekatan multi-objektif adalah arah riset yang relevan dan canggih, mendukung tujuan utama penelitian ini.
Cai & Wei (2024)	Peningkatan DRL untuk ATSC dengan arsitektur canggih.	Mengintegrasikan *dueling networks* dan *double* Q-*Learning* (membentuk D3QN) untuk mengatasi *overestimation bias* DQN.	Memberikan justifikasi teknis untuk pemilihan arsitektur D3QN sebagai solusi mengatasi ketidakstabilan algoritma.
Phan et al. (2025)	Studi kasus aplikasi D3QN untuk optimisasi sinyal lalu lintas.	Mendemonstrasikan kinerja D3QN yang stabil dan *robust* dalam simulasi kondisi lalu lintas dunia nyata yang kompleks.	Memberikan bukti empiris mengenai keandalan dan keunggulan arsitektur D3QN yang diadopsi dalam penelitian ini.
Freitag et al. (2025)	*Curriculum learning* untuk fungsi *reward* yang kompleks.	Mengusulkan kurikulum *reward* dua tahap untuk menyeimbangkan komponen *reward* yang bersaing dan menghindari optima lokal.	Memberikan justifikasi teoretis kuat untuk penggunaan strategi *curriculum learning* sebagai metode menangani *reward* multi-objektif.
Zheng et al. (2022)	*Curriculum learning* ("Lokal ke Global") untuk ATSC berbasis RL.	Menerapkan CL berbasis kompleksitas tugas (dari satu persimpangan ke jaringan) untuk mempercepat konvergensi MARL.	Menunjukkan relevansi CL dalam konteks ATSC, meskipun dengan jenis kurikulum yang berbeda (Tugas vs. *Reward*), memperkuat pendekatan CL.

Penulis	Fokus Utama	*Key Methodology*/Kontribusi	*Relevance to This Study*
Fang et al. (2023)	MARL berbasis GNN untuk koordinasi jaringan multi-objektif.	Menggunakan *graph neural networks* untuk komunikasi antar agen dalam jaringan padat dengan reward multi-objektif.	Merepresentasikan *state-of-the-art* dalam koordinasi MARL, menjadi titik perbandingan penting untuk perluasan ke multi-agen penelitian ini.
Mao et al. (2023)	MARL berbasis Attention (MASAC) untuk koordinasi jaringan.	Mengintegrasikan mekanisme *attention* untuk meningkatkan koordinasi dan konvergensi dalam MARL skala besar.	Memberikan justifikasi kuat untuk penggunaan mekanisme *attention* sebagai metode komunikasi dalam fase MARL penelitian ini.
Zhao et al. (2024)	Survei komprehensif DRL untuk ATSC.	Menyajikan taksonomi dan analisis mendalam tentang status riset DRL di ATSC, termasuk algoritma dan skenario aplikasi.	Memposisikan penelitian ini dalam lanskap riset yang lebih luas dan mengonfirmasi tren serta tantangan yang diidentifikasi.

2.1.3 Tantangan dan Perkembangan DRL untuk Adaptive Traffic Signal Control (ATSC)

Meskipun *deep reinforcement learning* (DRL) telah menunjukkan potensi dalam pengendalian sinyal lalu lintas adaptif (*adaptive traffic signal control*-ATSC), berbagai tantangan konseptual dan teknis masih menjadi hambatan untuk penerapannya secara optimal di dunia nyata. Salah satu keterbatasan utama dalam penelitian awal DRL untuk ATSC adalah fokusnya yang masih terbatas pada optimisasi tujuan tunggal. Fokus ini biasanya hanya berorientasi pada efisiensi lalu lintas, seperti pengurangan waktu tunda atau panjang antrean. Pendekatan semacam ini sering kali mengabaikan dampak eksternal, misalnya konsumsi bahan bakar, emisi karbon, serta aspek keselamatan lalu lintas (Tan dkk., 2025). Sebagai contoh, model dengan fungsi *reward* tunggal mungkin berhasil memaksimalkan *throughput*, tetapi di sisi lain dapat meningkatkan konsumsi energi akibat percepatan kendaraan yang mendadak. Oleh karena itu, diperlukan sebuah pendekatan multi-objektif yang dapat menyeimbangkan (*trade-off*) antara efisiensi, keberlanjutan lingkungan, dan keselamatan (Bouktif dkk., 2023; Gu dkk., 2020).

Sejalan dengan itu, terjadi pergeseran paradigma menuju *multi-objective deep reinforcement learning* (MODRL), yang memungkinkan agent mempertimbangkan lebih dari satu fungsi tujuan dalam proses pembelajaran. Pendekatan MODRL berupaya menyeimbangkan berbagai indikator kinerja lalu lintas seperti waktu tunda, panjang antrean, konsumsi bahan bakar, dan tingkat emisi. Studi-studi terkini, seperti yang dilakukan oleh Mirbakhsh & Azizi (2024), memperkenalkan metode pembobotan dinamis pada *reward* untuk menyesuaikan prioritas antar tujuan sesuai kondisi lalu lintas. Sementara itu, Mannion dkk. (2018) menyoroti pentingnya knowledge-based *reward* shaping dalam mempercepat konvergensi pembelajaran multi-objektif dengan memanfaatkan informasi domain, sehingga agent dapat belajar secara efisien dari kondisi kompleks. Model multi-objektif ini juga mulai dikombinasikan dengan curriculum learning untuk memandu agent dari tugas sederhana ke kompleks, sebagaimana diterapkan dalam studi Zheng dkk. (2022) melalui pendekatan *from local to global curriculum learning* yang berhasil meningkatkan stabilitas pelatihan dalam lingkungan berskala besar.

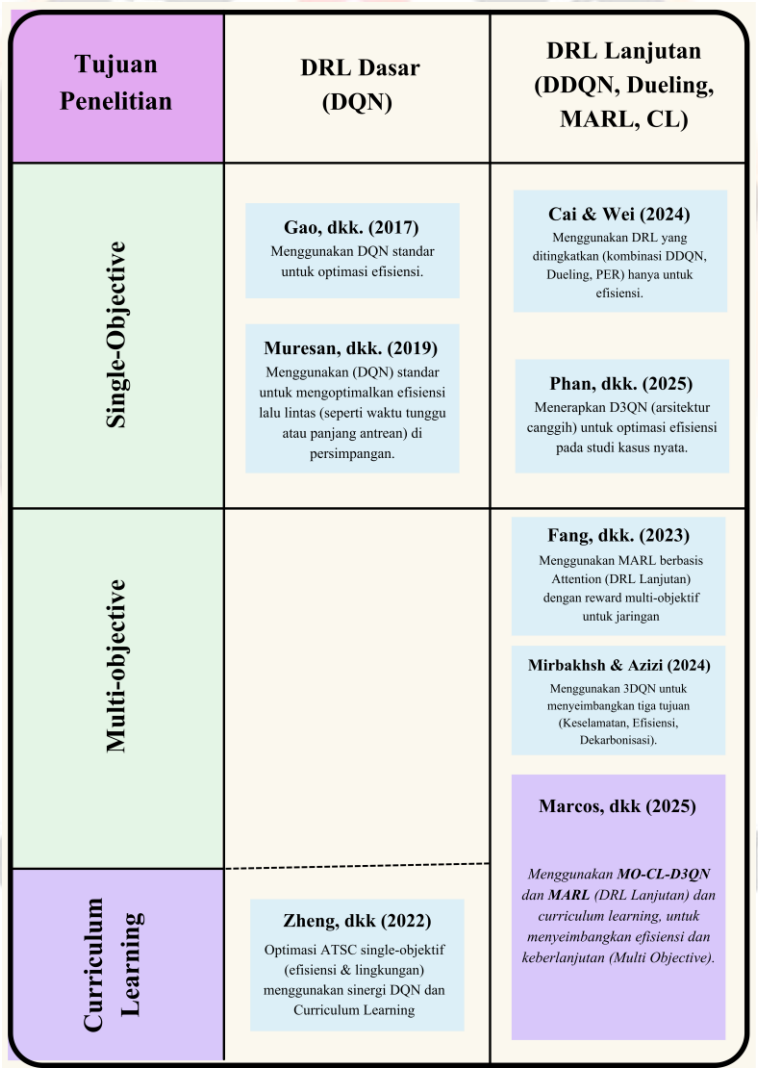
Selain tantangan multi-objektif, masalah skalabilitas menjadi perhatian utama dalam penerapan DRL pada jaringan lalu lintas berskala besar. Pendekatan awal berbasis *single-agent* terbukti tidak efisien karena tidak mampu mempertimbangkan interaksi antar-persimpangan, sehingga solusi yang dihasilkan sering bersifat lokal dan tidak optimal secara global (T. Chu dkk., 2019). Untuk mengatasi keterbatasan ini, penelitian beralih ke *Multi-Agent Reinforcement Learning* (MARL), di mana setiap persimpangan bertindak sebagai agent otonom yang saling berkoordinasi. Pendekatan ini meningkatkan efisiensi global dengan memungkinkan agent untuk berbagi informasi melalui mekanisme komunikasi eksplisit atau implisit (Bie dkk., 2024; Sabit, 2023).

Beberapa studi menunjukkan bahwa komunikasi antar-agent memainkan peran penting dalam meningkatkan kinerja sistem multi-interseksi. Misalnya, Mao dkk. (2023) mengembangkan model *multi-agent attention-based soft actor-critic* (MASAC) yang mengintegrasikan mekanisme attention dan centralized learning untuk meningkatkan koordinasi antar-persimpangan dan mempercepat konvergensi pelatihan. Sementara itu, Fang dkk. (2023) memperkenalkan protokol komunikasi berbasis *graph neural networks* (GNN) untuk mendukung kolaborasi antar-agent dalam lingkungan lalu lintas padat. Pendekatan ini memungkinkan pembagian informasi spasio-temporal yang efisien antar simpang tanpa menambah beban komputasi yang signifikan. Di sisi lain, Zhou dkk. (2025) mengusulkan kerangka *communication-efficient* MARL yang memanfaatkan *selective message passing* untuk menjaga kestabilan pelatihan sekaligus mengurangi kebutuhan *bandwidth* pada jaringan besar.

Evolusi MARL dalam ATSC juga didorong oleh penerapan model koordinasi canggih seperti *value-decomposition* (VDN, QMIX) dan *centralized training with decentralized execution* (CTDE) yang memungkinkan agent belajar secara terpusat namun bertindak secara otonom di lapangan (Mushtaq dkk., 2023). Penelitian-penelitian tersebut menegaskan bahwa strategi komunikasi adaptif dan pembelajaran kolaboratif menjadi kunci keberhasilan DRL berskala besar untuk pengendalian sinyal lalu lintas masa depan, di mana efisiensi sistem harus berjalan seiring dengan keberlanjutan dan keamanan transportasi.

Secara keseluruhan, arah perkembangan DRL untuk ATSC menunjukkan pergeseran dari model efisiensi tunggal menuju sistem pembelajaran cerdas berbasis kolaborasi dan multi-objektif. Tantangan seperti *curse of dimensionality*, ketidakseimbangan *reward* antar tujuan, serta keterbatasan komunikasi antar-agent kini sedang diatasi melalui integrasi *attention mechanisms, graph learning*, dan *multi-objective optimization frameworks*. Dengan kombinasi pendekatan tersebut, masa depan ATSC diproyeksikan akan semakin otonom, adaptif, dan mampu mendukung mobilitas perkotaan yang berkelanjutan.

2.1.4 Posisi Penelitian dan Kebaruan yang Diusulkan



Gambar 2.6 Posisi Penelitian Berdasarkan Orientasi Tujuan Optimisasi dan Perkembangan Metode DRL untuk ATSC

Sebagaimana ditunjukkan pada Gambar 2.6, posisi penelitian dipetakan berdasarkan dua dimensi utama, yaitu orientasi tujuan optimisasi dan tingkat perkembangan metode *deep reinforcement learning* (DRL) untuk *adaptive traffic signal control* (ATSC). Gambar dibaca secara vertikal dari pendekatan *single-objective* menuju *multi-objective* dan *curriculum learning*, sedangkan pembacaan secara horizontal menunjukkan perkembangan metode dari DRL dasar berbasis *deep Q-network* (DQN) menuju DRL lanjutan yang mencakup *double deep Q-network* (DDQN), *dueling architecture*, *dueling double deep Q-network* (D3QN), *multi-agent reinforcement learning* (MARL), dan *curriculum learning* (CL).

Pada kelompok DRL dasar, sebagian besar studi awal menggunakan DQN standar untuk mengoptimalkan efisiensi lalu lintas pada persimpangan tunggal, terutama melalui pengurangan waktu tunggu atau panjang antrean kendaraan (Gao dkk., 2017; Muresan dkk., 2019). Pendekatan tersebut menjadi fondasi awal penerapan DRL dalam pengendalian sinyal lalu lintas adaptif. Namun, DQN standar masih memiliki keterbatasan berupa *overestimation bias* yang dapat memengaruhi stabilitas estimasi nilai aksi dan proses konvergensi model (Anschel dkk., 2017; Ly dkk., 2022).

Perkembangan selanjutnya mengarah pada penerapan arsitektur DRL lanjutan. DDQN digunakan untuk mengurangi *overestimation bias*, sedangkan *dueling architecture* memisahkan estimasi nilai keadaan (*state value*) dan keunggulan aksi (*action advantage*). Integrasi kedua pendekatan tersebut membentuk D3QN. Meskipun mampu meningkatkan stabilitas dan efisiensi pembelajaran, sejumlah studi yang menerapkan DDQN dan D3QN masih berfokus pada satu tujuan utama, yaitu efisiensi operasional lalu lintas (Cai & Wei, 2024; Phan dkk., 2025).

Pada dimensi *multi-objective*, penelitian terdahulu mulai mengembangkan fungsi *reward* yang mempertimbangkan beberapa indikator kinerja secara simultan. Fang dkk. (2023) mengembangkan pendekatan MARL berbasis jaringan untuk menyeimbangkan aspek efisiensi, keselamatan, dan koordinasi lalu lintas, sedangkan Mirbakhsh dan Azizi (2024) menerapkan pendekatan ATSC multi-objektif dengan mempertimbangkan efisiensi, keselamatan, dan dekarbonisasi. Temuan tersebut menunjukkan bahwa pengendalian sinyal lalu lintas tidak cukup

hanya berorientasi pada pengurangan antrean atau waktu tunggu, tetapi juga perlu mempertimbangkan tujuan lain yang dapat saling bertentangan, termasuk keberlanjutan lingkungan.

Selain pengembangan fungsi *reward* multi-objektif, *curriculum learning* mulai digunakan untuk mengatasi kompleksitas proses pelatihan. Pendekatan ini memungkinkan agen mempelajari tugas atau komponen *reward* secara bertahap, mulai dari tingkat kompleksitas yang lebih rendah menuju kondisi yang lebih kompleks. Zheng dkk. (2022) menunjukkan relevansi *curriculum learning* untuk pengembangan ATSC dari skenario lokal menuju jaringan yang lebih luas, sedangkan Freitag dkk. (2025) menunjukkan bahwa strategi kurikulum dapat digunakan untuk menangani komponen *reward* yang saling bersaing.

Pada tingkat jaringan, MARL digunakan untuk mengatasi keterbatasan optimisasi pada persimpangan tunggal. Dalam pendekatan ini, setiap persimpangan direpresentasikan sebagai agen yang dapat berkoordinasi dengan agen lain untuk meningkatkan kinerja lalu lintas secara keseluruhan. Pengembangan MARL berbasis *attention* memungkinkan agen memberikan bobot yang berbeda terhadap informasi dari persimpangan tetangga berdasarkan relevansinya terhadap kondisi lalu lintas yang sedang diamati (Fang dkk., 2023; Mao dkk., 2023).

Berdasarkan pemetaan tersebut, penelitian ini (Marcos dkk., 2025) menempati posisi pada integrasi DRL lanjutan, *multi-objective*, dan *curriculum learning*. Kerangka kerja yang diusulkan menggunakan MO-CL-D3QN (*multi-objective curriculum learning dueling double deep Q-network*) untuk menyeimbangkan efisiensi operasional dan keberlanjutan lingkungan pada persimpangan tunggal. Selanjutnya, kerangka kerja diperluas menggunakan MARL berbasis *attention* untuk mendukung koordinasi antar-persimpangan pada tingkat jaringan. Dengan demikian, penelitian ini mengintegrasikan fungsi *reward* multi-objektif, stabilitas arsitektur D3QN, strategi pembelajaran bertahap, dan koordinasi multi-agen dalam satu kerangka pengembangan ATSC.

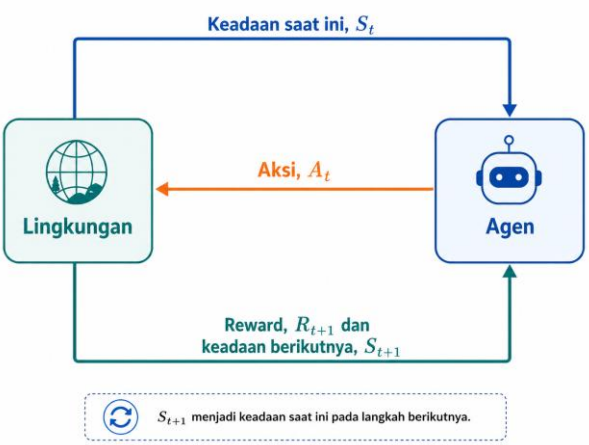
2.2 Landasan Teori

Bagian ini menguraikan fondasi teoretis yang menjadi dasar metodologi penelitian. Pembahasan dimulai dari prinsip fundamental *reinforcement learning* (RL) dan formalisasinya dalam *markov decision process* (MDP), dilanjutkan dengan evolusi arsitektur *deep* Q-*Network* (DQN) yang menjadi inti dari agen yang diusulkan. Terakhir, dibahas pula konsep-konsep kunci yang secara spesifik diimplementasikan dalam kerangka kerja ini, yaitu *multi-objective deep reinforcement learning* (MODRL) dan *curriculum learning* (CL).

2.2.1 *Reinforcement Learning* dan *Markov Decision Process (MDP)*

Reinforcement learning (RL) merupakan salah satu cabang *machine learning* yang mempelajari cara agen menentukan tindakan dalam suatu lingkungan untuk memaksimalkan *reward* kumulatif. Berbeda dengan pembelajaran terawasi, agen RL tidak memperoleh pasangan data masukan dan keluaran yang telah diberi label, tetapi mempelajari kebijakan melalui interaksi berulang dengan lingkungan menggunakan mekanisme *trial-and-error* (Sutton & Barto, 2018).

Sebagaimana ditunjukkan pada Gambar 2.7, siklus interaksi agen–lingkungan berlangsung pada setiap langkah waktu t. Pada awal siklus, lingkungan memberikan keadaan saat ini S_t kepada agen. Berdasarkan keadaan tersebut, agen memilih aksi $A_t$menggunakan kebijakan π. Setelah aksi diterapkan, lingkungan mengalami perubahan dan menghasilkan dua keluaran, yaitu *reward* R_{t+1} dan keadaan berikutnya S_{t+1}. Keadaan S_{t+1} selanjutnya menjadi masukan bagi agen pada langkah waktu berikutnya. Proses tersebut berlangsung secara berulang hingga episode simulasi berakhir atau kriteria penghentian tercapai.



Gambar 2.7 Siklus Interaksi Agen–Lingkungan pada Setiap Langkah Waktu dalam *Reinforcement Learning*

Secara ringkas, satu langkah interaksi agen–lingkungan dapat dituliskan sebagai berikut: $S_t \rightarrow A_t \rightarrow (R_{t+1}, S_{t+1})$

Persamaan (2.1) sampai dengan (2.4) merupakan dasar *reinforcement learning* (Sutton & Barto, 2018), yang memperlihatkan hubungan matematis antara keadaan saat ini, aksi, *reward*, dan keadaan berikutnya dirumuskan melalui probabilitas transisi :

$$p(s', r \mid s, a) = \Pr\{S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a\} \qquad (2.1)$$

dengan:

- S_t menyatakan keadaan lingkungan pada langkah waktu t;
- A_t menyatakan aksi yang dipilih agen pada langkah waktu t;
- R_{t+1} menyatakan *reward* yang diterima setelah aksi $A_t$diterapkan;
- S_{t+1} menyatakan keadaan lingkungan pada langkah waktu berikutnya;
- $s$dan s'masing-masing menyatakan realisasi keadaan saat ini dan keadaan berikutnya;
- a menyatakan realisasi aksi yang dipilih agen; dan
- r menyatakan realisasi *reward* yang diterima agen.

Persamaan (2.1) menunjukkan bahwa probabilitas munculnya keadaan berikutnya S_{t+1} dan *reward* R_{t+1} ditentukan oleh keadaan saat ini $S_t$dan aksi A_t. Dengan demikian, posisi siklus pada Gambar 2.7 terletak pada proses perubahan

keadaan: S_{t+1}yang dihasilkan pada satu langkah waktu digunakan kembali sebagai keadaan saat ini pada langkah waktu berikutnya.

Masalah pengambilan keputusan sekuensial dalam RL secara formal dimodelkan sebagai *Markov decision process* (MDP). Sebuah MDP didefinisikan sebagai tuple:

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma) \tag{2.2}$$

dengan:

- $\mathcal{S}$ adalah himpunan semua kemungkinan keadaan (*states*).
- $\mathcal{A}$ adalah himpunan semua tindakan (*actions*) yang mungkin.
- $\mathcal{P}$ adalah fungsi probabilitas transisi, $\mathcal{P}(s' \mid s, a)$, yang mendefinisikan probabilitas transisi ke keadaan dari keadaan setelah mengambil tindakan .
- $\mathcal{R}$ adalah fungsi *reward*, $\mathcal{R}(s, a, s')$ yang memberikan imbalan langsung setelah transisi dari ke .
- γ adalah faktor diskon ($0 \leq \gamma \leq 1$), yang menentukan seberapa penting *reward* di masa depan dibandingkan dengan *reward* saat ini.

Tujuan dari agen RL adalah untuk mempelajari sebuah kebijakan (π), yaitu sebuah pemetaan dari keadaan ke tindakan, yang memaksimalkan ekspektasi dari *return* kumulatif yang didiskontokan (G_t). *Return* ini didefinisikan sebagai jumlah dari semua *reward* di masa depan dari waktu :

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{2.3}$$

dengan:

- $G_t$menyatakan *return* kumulatif mulai dari langkah waktu t;
- R_{t+k+1}menyatakan *reward* yang diterima pada langkah waktu berikutnya; dan
- γ^kmenyatakan faktor diskonto untuk *reward* pada horizon waktu ke-k.

Untuk mengevaluasi seberapa baik sebuah kebijakan, digunakan fungsi nilai (*value functions*). Fungsi nilai-aksi, $Q^{\pi}(s, a)$, adalah ekspektasi *return* yang akan diterima jika memulai dari keadaan , mengambil tindakan , dan setelah itu mengikuti

kebijakan . Algoritma RL bertujuan untuk menemukan kebijakan optimal, π^*, yang memiliki fungsi nilai-aksi optimal, $Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a)$. Fungsi nilai-aksi optimal ini mematuhi sebuah identitas penting yang dikenal sebagai persamaan optimalitas Bellman (Bellman Optimality Equation):

$$Q^*(s,a) = E\left[R_{t+1} + \gamma \max_{a'} Q^*(S_{t+1}, a') \,\middle|\, S_t = s, A_t = a\right] \quad (2.4)$$

dengan:

- $Q^*(s,a)$ menyatakan nilai optimal dari aksi $a$pada keadaan s;
- R_{t+1} menyatakan *reward* langsung setelah aksi diterapkan;
- S_{t+1} menyatakan keadaan berikutnya;
- a' menyatakan kandidat aksi pada keadaan berikutnya; dan
- γ menyatakan faktor diskon.

Persamaan (2.4) menunjukkan bahwa nilai optimal suatu aksi ditentukan oleh penjumlahan antara *reward* langsung dan nilai maksimum yang dapat diperoleh dari keadaan berikutnya. Persamaan tersebut menjadi dasar bagi berbagai algoritma RL berbasis nilai, termasuk Q-learning dan *deep Q-network* (DQN), yang dibahas pada subbab berikutnya.

2.2.2 *Deep Q-Network* (DQN) dan Varian-variannya

Algoritma *Q-learning* merupakan salah satu metode *reinforcement learning* berbasis nilai (*value-based reinforcement learning*) yang digunakan untuk mempelajari kualitas suatu tindakan pada keadaan tertentu. Pada pendekatan konvensional, nilai tindakan disimpan dalam bentuk tabel yang disebut *Q-table*. Setiap elemen pada tabel tersebut merepresentasikan estimasi nilai jangka panjang dari pasangan keadaan dan tindakan, yaitu $Q(s,a)$. Pendekatan berbasis tabel efektif diterapkan pada permasalahan dengan ruang keadaan dan tindakan yang relatif kecil serta bersifat diskrit. Namun, pendekatan ini menjadi tidak efisien ketika dimensi ruang keadaan meningkat secara signifikan (*curse of dimensionality*) (Watkins & Dayan, 1992).

Pada sistem pengendalian sinyal lalu lintas adaptif (*adaptive traffic signal control*—ATSC), keadaan lingkungan dapat direpresentasikan oleh matriks yang

memuat posisi kendaraan, kecepatan kendaraan, panjang antrean, waktu tunggu, serta fase sinyal aktif. Banyaknya kombinasi keadaan menyebabkan penggunaan *Q-table* menjadi tidak praktis. Untuk mengatasi keterbatasan tersebut, *Deep Q-Network* (DQN) menggantikan *Q-table* dengan jaringan saraf dalam (*deep neural network*) sebagai pendekatan fungsi (*function approximator*) (Mnih dkk., 2015). Jaringan tersebut menerima representasi keadaan $S_t$sebagai masukan dan menghasilkan estimasi *Q-value* untuk setiap tindakan yang tersedia sebagai keluaran.

Dalam konteks ATSC, tindakan A_t dapat berupa pemilihan fase sinyal lalu lintas dari himpunan tindakan diskrit $\mathcal{A}$. Agen memilih tindakan menggunakan strategi ϵ-*greedy*. Pada tahap awal pelatihan, agen lebih sering mengeksplorasi berbagai alternatif fase sinyal. Seiring dengan berjalannya pelatihan, nilai ϵdikurangi secara bertahap agar agen lebih banyak memilih tindakan dengan nilai Qtertinggi.

a. Mekanisme Dasar Deep Q-Network

DQN memiliki dua mekanisme utama untuk meningkatkan stabilitas pembelajaran, yaitu *experience replay* dan *target network*, diperlihatkan pada persamaan (2.5) sampai dengan (2.7) (Mnih dkk., 2015).

1. Experience replay

Setiap interaksi agen dengan lingkungan disimpan dalam *replay buffer* $\mathcal{D}$dalam bentuk transisi:

$$e_t = (S_t, A_t, R_{t+1}, S_{t+1}, d_t) \tag{2.5}$$

dengan S_t sebagai keadaan saat ini, A_t sebagai tindakan yang dipilih, R_{t+1} sebagai *reward* yang diterima, S_{t+1} sebagai keadaan berikutnya, dan d_t sebagai indikator berakhirnya episode. Nilai $d_t = 1$ digunakan apabila transisi menghasilkan *terminal state*, sedangkan $d_t = 0$ digunakan apabila episode masih berlanjut.

Pada proses pelatihan, sejumlah transisi dipilih secara acak dari *replay buffer* untuk membentuk *minibatch*. Pengambilan sampel secara acak mengurangi korelasi

temporal antarpengalaman yang berurutan, meningkatkan efisiensi penggunaan data, dan membantu menjaga kestabilan pembaruan parameter jaringan.

2. Target network

DQN menggunakan dua jaringan saraf dengan struktur yang sama, yaitu:

- *online network* $Q(S, A; \theta)$, yang diperbarui pada setiap proses optimisasi; dan
- *target network* $Q(S, A; \theta^{-})$, yang digunakan untuk menghitung nilai target pembelajaran.

Parameter θ merepresentasikan bobot pada *online network*, sedangkan θ^{-} merepresentasikan bobot pada *target network*. Pemisahan kedua jaringan tersebut mencegah target pembelajaran berubah terlalu cepat selama proses optimisasi.

Nilai target DQN dirumuskan sebagai berikut:

$$y_t^{\mathrm{DQN}} = R_{t+1} + \gamma(1 - d_t)\max_{a' \in \mathcal{A}} Q(S_{t+1}, a'; \theta^{-}) \tag{2.6}$$

dengan:

- y_t^{DQN} sebagai nilai target DQN;
- R_{t+1} sebagai *reward* setelah tindakan A_t dijalankan;
- γ sebagai *discount factor*;
- d_t sebagai indikator *terminal state*;
- S_{t+1} sebagai keadaan berikutnya;
- a' sebagai kandidat tindakan pada keadaan berikutnya; dan
- θ^{-} sebagai parameter *target network*.

Faktor $(1-d_t)$ memastikan bahwa estimasi nilai masa depan tidak dihitung apabila episode telah berakhir.

Parameter *online network* diperbarui dengan meminimalkan selisih kuadrat antara nilai target dan nilai prediksi jaringan. Fungsi *loss* DQN dirumuskan sebagai berikut:

$$\mathcal{L}(\theta) = \frac{1}{B}\sum_{i=1}^{B}\left[y_i^{\mathrm{DQN}} - Q(S_i, A_i; \theta)\right]^2 \quad (2.7)$$

dengan:

- $\mathcal{L}(\theta)$sebagai fungsi *loss*;
- B sebagai ukuran *minibatch*;
- y_i^{DQN} sebagai nilai target untuk sampel ke-i; dan
- $Q(S_i, A_i; \theta)$sebagai nilai prediksi dari *online network*.

b. Double Deep Q-Network

Meskipun DQN dapat menangani ruang keadaan berdimensi besar, algoritma ini masih memiliki keterbatasan berupa *overestimation bias*. Permasalahan tersebut muncul karena operasi maksimum pada DQN menggunakan jaringan yang sama untuk memilih tindakan dan mengevaluasi nilai tindakan. Akibatnya, tindakan tertentu dapat memperoleh estimasi nilai yang terlalu tinggi, sehingga agen berpotensi memilih tindakan suboptimal (van Hasselt dkk., 2016; Ly dkk., 2022).

Double Deep Q-Network (DDQN) mengurangi *overestimation bias* dengan memisahkan proses pemilihan tindakan dari proses evaluasi nilainya. Tindakan terbaik pada keadaan berikutnya dipilih menggunakan *online network*:

$$a_t^* = \arg\max \mathrm{a}' \in \mathrm{A}\, Q(S_{t+1}, a'; \theta) \quad (2.8)$$

Selanjutnya, nilai tindakan tersebut dievaluasi menggunakan *target network*. Nilai target DDQN dirumuskan sebagai berikut:

$$y_t^{\mathrm{DDQN}} = R_{t+1} + \gamma(1 - d_t)Q(S_{t+1}, a_t^*; \theta^-) \quad (2.9)$$

Pada Persamaan (2.9), *online network* berfungsi memilih tindakan a_t^*, sedangkan *target network* berfungsi menghitung nilai dari tindakan tersebut. Pemisahan fungsi ini mengurangi kemungkinan estimasi nilai yang terlalu optimistis dan meningkatkan kestabilan pembelajaran.

c. Dueling Deep Q-Network

Dueling Deep Q-Network memperbaiki kemampuan jaringan dalam mempelajari representasi keadaan dengan memisahkan estimasi *Q-value* menjadi dua aliran (*streams*) (Wang dkk., 2021), yaitu:

1. fungsi nilai keadaan $V(s)$, yang menunjukkan tingkat kepentingan suatu keadaan tanpa mempertimbangkan tindakan tertentu; dan
2. fungsi keunggulan tindakan $A(s, a)$, yang menunjukkan seberapa baik suatu tindakan dibandingkan tindakan lain pada keadaan yang sama.

Kedua aliran tersebut digabungkan untuk memperoleh estimasi *Q-value*:

$$Q(s, a; \theta) = V(s; \theta) + \left[A(s, a; \theta) - \frac{1}{|\mathcal{A}|}\sum_{a' \in \mathcal{A}} A\,(s, a'; \theta)\right] \quad (2.10)$$

dengan:

- $V(s; \theta)$sebagai estimasi nilai keadaan;
- $A(s, a; \theta)$sebagai estimasi keunggulan tindakan;
- $|\ \mathcal{A}\ |$sebagai jumlah tindakan yang tersedia;
- $Q(s, a; \theta)$sebagai estimasi nilai tindakan akhir;
- θ sebagai parameter jaringan saraf yang dipelajari selama proses pelatihan;
- a' sebagai indeks kandidat aksi lain dalam himpunan aksi $\mathcal{A}$ yang digunakan untuk menghitung nilai rata-rata *advantage*

Pengurangan nilai rata-rata *advantage* pada Persamaan (2.10) diperlukan untuk menjaga identifikasi nilai yang konsisten. Arsitektur ini memungkinkan agen mempelajari apakah suatu kondisi lalu lintas secara umum menguntungkan atau merugikan, sekaligus membedakan tindakan fase sinyal yang paling sesuai pada kondisi tersebut (Wang dkk., 2016).

d. Dueling Double Deep Q-Network

Dueling Double Deep Q-Network (D3QN) merupakan integrasi antara DDQN dan *Dueling DQN*. DDQN digunakan untuk mengurangi *overestimation bias* melalui pemisahan proses pemilihan dan evaluasi tindakan. Sementara itu, arsitektur

dueling digunakan untuk memisahkan estimasi nilai keadaan dan keunggulan relatif setiap tindakan.

Dalam penelitian ini, D3QN digunakan sebagai arsitektur dasar agen karena pengendalian sinyal lalu lintas memerlukan keputusan diskrit pada kondisi lalu lintas yang dinamis. Pada keadaan tertentu, beberapa pilihan fase sinyal dapat menghasilkan dampak yang relatif serupa. Pemisahan fungsi $V(s)$dan $A(s,a)$membantu agen mengenali nilai umum suatu keadaan, sedangkan mekanisme DDQN membantu menjaga estimasi nilai tindakan agar tidak terlalu optimistis. Kombinasi tersebut ditujukan untuk menghasilkan proses pembelajaran yang lebih stabil dan kebijakan pengendalian sinyal yang lebih andal.

Parameter *target network* pada penelitian ini diperbarui secara bertahap menggunakan mekanisme *soft update* (Lillicrap dkk., 2016):

$$\theta^- \leftarrow \tau\theta + (1-\tau)\theta^- \tag{2.11}$$

dengan:

- θ sebagai parameter *online network*;
- θ^- sebagai parameter *target network*; dan
- τ sebagai koefisien *soft update*, dengan $0 < \tau \leq 1$.

Nilai τyang relatif kecil menyebabkan perubahan parameter *target network* berlangsung secara bertahap, sehingga target pembelajaran tidak berubah secara drastis. Arsitektur D3QN selanjutnya diintegrasikan dengan fungsi *reward* multi-objektif dan strategi *curriculum learning* untuk membentuk kerangka kerja MO-CL-D3QN yang digunakan dalam penelitian ini.

2.2.3 Multi-Objective Deep Reinforcement Learning (MODRL)

Sebagian besar algoritma *reinforcement learning* (RL) konvensional dirancang untuk mengoptimalkan satu nilai *reward* skalar. Pendekatan tersebut sesuai untuk permasalahan yang memiliki satu tujuan dominan. Namun, pengendalian sinyal lalu lintas adaptif merupakan permasalahan yang secara inheren bersifat multi-objektif karena keputusan fase sinyal perlu mempertimbangkan beberapa indikator kinerja

secara simultan. Pengurangan waktu tunggu dan panjang antrean perlu diseimbangkan dengan peningkatan *throughput* serta pengurangan konsumsi bahan bakar. Optimalisasi salah satu indikator secara berlebihan dapat menurunkan kinerja indikator lainnya, sehingga diperlukan mekanisme untuk mengelola *trade-off* antartujuan.

Multi-objective deep reinforcement learning (MODRL) merupakan pengembangan RL yang memungkinkan agen mengoptimalkan beberapa tujuan secara simultan. Apabila fungsi nilai diaproksimasi menggunakan jaringan saraf dalam, pendekatan tersebut disebut MODRL. Penggunaan jaringan saraf memungkinkan agen mempelajari hubungan nonlinear antara kondisi lalu lintas, keputusan fase sinyal, dan perubahan indikator kinerja pada lingkungan yang kompleks (Mossalam dkk., 2016; Nguyen dkk., 2020).

Berbeda dengan RL tujuan tunggal yang hanya menghasilkan satu nilai *reward*, MODRL menggunakan vektor *reward* yang terdiri atas beberapa komponen objektif. Dalam konteks pengendalian sinyal lalu lintas adaptif, vektor *reward* pada waktu ke-(t) dirumuskan sebagai berikut:

$$\mathbf{r}_t = \begin{bmatrix} r_t^{wait} \\ r_t^{queue} \\ r_t^{tp} \\ r_t^{fuel} \end{bmatrix} \tag{2.12}$$

dengan:

- $\mathbf{r}_t$ adalah vektor *reward* pada waktu ke-*t*;
- r_t^{wait} adalah komponen *reward* yang merepresentasikan perubahan waktu tunggu kendaraan;
- r_t^{queue} adalah komponen *reward* yang merepresentasikan perubahan panjang antrean;
- r_t^{tp} adalah komponen *reward* yang merepresentasikan perubahan *throughput*;
- r_t^{fuel} adalah komponen *reward* yang merepresentasikan perubahan konsumsi bahan bakar.

Setiap komponen perlu dirumuskan dengan arah optimisasi yang konsisten. Penurunan waktu tunggu, panjang antrean, dan konsumsi bahan bakar harus menghasilkan nilai *reward* yang lebih tinggi. Sebaliknya, peningkatan *throughput* juga harus menghasilkan nilai *reward* yang lebih tinggi. Dengan demikian, agen memperoleh umpan balik positif ketika aksi yang dipilih mampu meningkatkan efisiensi operasional sekaligus mengurangi dampak lingkungan.

Arsitektur D3QN tetap memerlukan satu nilai *reward* skalar untuk memperbarui nilai aksi atau *Q-value*. Oleh karena itu, vektor *reward* perlu ditransformasikan menjadi nilai skalar melalui proses skalarisasi (*scalarization*) (Mossalam dkk., 2016; Nguyen dkk., 2020). Salah satu metode yang umum digunakan adalah penjumlahan tertimbang (*weighted sum*), yang dirumuskan sebagai berikut:

$$R_t = \mathbf{w}^T \mathbf{r}_t = \sum_{j=1}^{k} w_j \, r_t^{(j)} \tag{2.13}$$

dengan:

- R_t adalah nilai *reward* skalar pada waktu ke-*t*;
- $\mathbf{w}^T \mathbf{r}_t$ adalah hasil perkalian skalar antara vektor bobot dan vektor *reward*.;
- $r_t^{(j)}$ adalah komponen *reward* untuk objektif ke-*j*;
- k adalah jumlah objektif yang dioptimalkan

Bobot w_j menunjukkan tingkat kepentingan relatif dari setiap objektif. Nilai bobot yang lebih besar memberikan pengaruh yang lebih kuat terhadap proses pembelajaran agen. Penentuan bobot perlu dilakukan secara proporsional agar kebijakan yang dihasilkan tidak hanya unggul pada satu indikator, tetapi juga memberikan kinerja yang seimbang pada keseluruhan sistem.

Dalam penelitian ini, fungsi *reward* skalar dirumuskan berdasarkan empat komponen kinerja lalu lintas sebagai berikut (Fang dkk., 2024; Gong dkk., 2020):

$$R_t = w_{wait} r_t^{wait} + w_{queue} r_t^{queue} + w_{tp} r_t^{tp} + w_{fuel} r_t^{fuel} \tag{2.14}$$

dengan:

- $w_{wait} r_t^{wait}$ adalah bobot untuk komponen waktu tunggu;

- $w_{queue} r_t^{queue}$ adalah bobot untuk komponen panjang antrean;
- $w_{tp} r_t^{tp}$ adalah bobot untuk komponen *throughput*;
- $w_{fuel} r_t^{fuel}$ adalah bobot untuk komponen konsumsi bahan bakar.

Keempat komponen tersebut dapat dikelompokkan menjadi dua tujuan utama, yaitu efisiensi operasional dan keberlanjutan lingkungan. Tujuan efisiensi operasional mencakup waktu tunggu, panjang antrean, dan *throughput*, sedangkan tujuan keberlanjutan direpresentasikan oleh konsumsi bahan bakar. Secara konseptual, pengelompokan tersebut dirumuskan sebagai berikut:

$$r_t^{eff} = \lambda_{wait} r_t^{wait} + \lambda_{queue} r_t^{queue} + \lambda_{tp} r_t^{tp} \quad (2.15a)$$

$$R_t = w_{eff} r_t^{eff} + w_{sus} r_t^{sus} \quad (2.15b)$$

$$r_t^{sus} = r_t^{fuel}$$

dengan:

- r_t^{eff} adalah *reward* efisiensi operasional;
- r_t^{sus} adalah *reward* keberlanjutan lingkungan;
- $\lambda_{wait} r_t^{wait}$, $\lambda_{queue} r_t^{queue}$, dan $\lambda_{tp} r_t^{tp}$ adalah bobot internal pada kelompok efisiensi;
- w_{eff} adalah bobot tujuan efisiensi operasional;
- w_{sus} adalah bobot tujuan keberlanjutan;
- $w_{eff} + w_{sus} = 1$.

Persamaan (2.15a) dan Persamaan (2.15b) memiliki tujuan yang sama. Persamaan (2.15a) menunjukkan bentuk operasional fungsi *reward* berdasarkan empat indikator lalu lintas, sedangkan Persamaan (2.15b) memperjelas pengelompokan indikator tersebut menjadi dua tujuan utama penelitian. Sebelum digabungkan melalui proses skalarisasi, setiap komponen *reward* perlu berada pada rentang nilai yang sebanding agar komponen dengan skala numerik yang lebih besar tidak mendominasi proses pembelajaran.

Penggunaan konsumsi bahan bakar sebagai komponen keberlanjutan didasarkan pada pertimbangan stabilitas sinyal *reward*. Penggunaan emisi karbon secara langsung dapat menghasilkan sinyal yang lebih bising karena dipengaruhi

oleh berbagai faktor eksternal. Konsumsi bahan bakar dapat digunakan sebagai metrik proksi yang lebih stabil dan tetap relevan untuk merepresentasikan dampak lingkungan dari kebijakan pengendalian sinyal lalu lintas (Schumacher dkk., 2023).

Nilai R_t selanjutnya digunakan dalam proses pembaruan *Q-value* pada arsitektur D3QN. Dengan formulasi tersebut, agen tidak hanya diarahkan untuk mengurangi kemacetan, tetapi juga dilatih untuk menghasilkan kebijakan pengendalian sinyal yang lebih efisien dan berkelanjutan. Kerangka kerja MODRL ini menjadi landasan bagi integrasi strategi *curriculum learning* yang dibahas pada subbab berikutnya

2.2.4 Curriculum Learning (CL)

Curriculum learning (CL) adalah strategi pelatihan yang terinspirasi dari cara manusia dan hewan belajar, yaitu dengan memulai dari konsep atau tugas yang lebih mudah dan secara bertahap beralih ke tugas yang lebih kompleks (de Campos dkk., 2021). Dalam konteks *machine learning*, ini berarti menyusun data atau tugas pelatihan dalam urutan yang bermakna untuk mempercepat konvergensi dan meningkatkan kemampuan generalisasi model.

Dalam *reinforcement learning*, CL dapat diimplementasikan dalam berbagai cara, seperti menyederhanakan lingkungan, tugas, atau fungsi *reward* pada tahap awal pelatihan. Pendekatan yang paling relevan untuk penelitian ini adalah kurikulum *reward* (*reward curriculum*). Ketika dihadapkan pada fungsi *reward* multi-objektif yang kompleks, agen RL seringkali mengalami kesulitan pada tahap awal eksplorasi (dikenal sebagai *cold-start problem*) atau terjebak pada optima lokal dengan hanya mengeksploitasi salah satu komponen *reward* (Ibrahim dkk., 2024). Untuk mengatasi ini, kurikulum *reward* menyederhanakan fungsi *reward* pada tahap awal. Sebagai contoh, agen dapat dilatih terlebih dahulu hanya pada tujuan utama yang paling mudah dipelajari (misalnya, efisiensi). Setelah agen mencapai tingkat kompetensi tertentu pada tugas dasar ini, komponen *reward* tambahan yang lebih kompleks (misalnya, keberlanjutan) secara bertahap diperkenalkan (Freitag dkk., 2025). Strategi ini secara efektif mem-bootstrap proses pembelajaran, memungkinkan agen untuk membangun pemahaman dasar tentang

lingkungan sebelum mencoba menavigasi *trade-off* yang lebih rumit, yang pada akhirnya mengarah pada kebijakan akhir yang lebih unggul dan seimbang.

Zheng dkk. (2022) memperluas konsep ini dalam penelitian *from local to global curriculum learning*, di mana proses pelatihan RL dilakukan secara hierarkis—dimulai dari optimisasi pada satu persimpangan (local level) hingga menuju koordinasi antar-persimpangan (global level). Strategi ini tidak hanya meningkatkan stabilitas pelatihan tetapi juga mempercepat transfer pengetahuan antar agen dalam sistem *multi-agent reinforcement learning* (MARL).

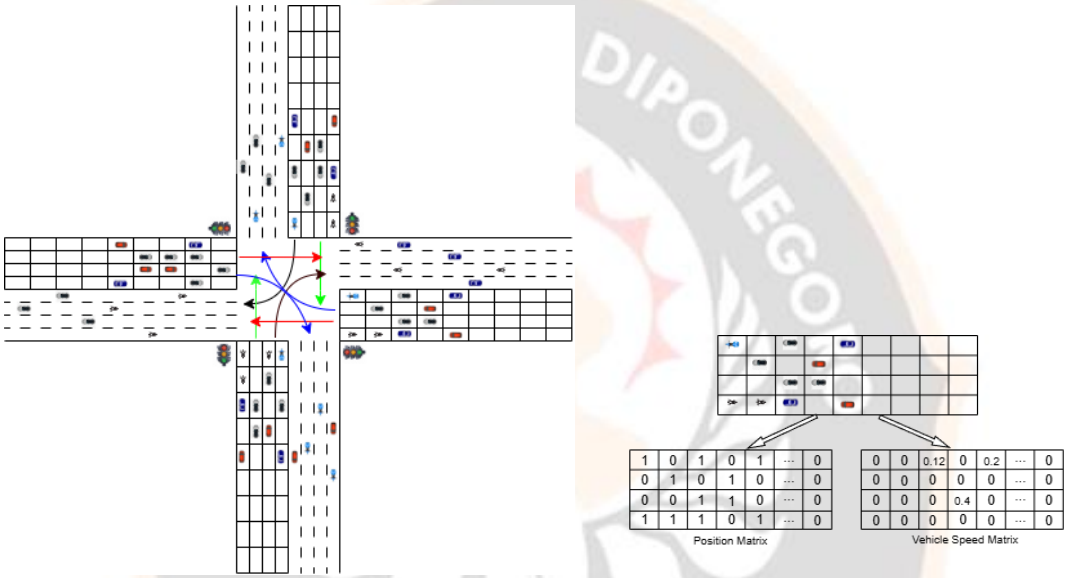
Dalam konteks penelitian ini, prinsip CL diterapkan secara langsung dalam kerangka MO-CL-D3QN (*multi-objective curriculum learning–dueling double deep Q-network*). Pada tahap awal pelatihan, agen fokus pada satu tujuan utama, yaitu meminimalkan *average waiting time* dan *queue length*. Setelah kebijakan dasar yang stabil diperoleh, sistem memperkenalkan reward tambahan yang berkaitan dengan efisiensi bahan bakar dan koordinasi antar-agen. Pendekatan ini memungkinkan agen belajar secara bertahap dari kondisi sederhana ke kompleks, menghasilkan model yang lebih stabil, efisien, dan mampu beradaptasi terhadap dinamika lalu lintas yang bervariasi. Dengan demikian, penerapan *Curriculum learning* dalam penelitian ini tidak hanya berfungsi sebagai strategi pelatihan, tetapi juga sebagai mekanisme stabilisasi pembelajaran multi-objektif, yang memperkuat konvergensi dan mengurangi risiko fluktuasi nilai Q pada arsitektur D3QN dalam sistem pengendalian sinyal lalu lintas adaptif.

2.2.5 Konsep Dasar *Discrete Traffic State Encoding* (DTSE)

Discrete traffic state encoding (DTSE) merupakan metode representasi status (*state representation*) yang digunakan dalam sistem *reinforcement learning* (RL) untuk menggambarkan kondisi lalu lintas secara diskrit berdasarkan segmentasi ruang jalan dan waktu. Teknik ini diperkenalkan untuk mengatasi keterbatasan representasi status konvensional yang hanya menggunakan parameter agregat seperti jumlah kendaraan atau waktu tunggu rata-rata, yang tidak cukup untuk menggambarkan kompleksitas kondisi lalu lintas secara rinci (Bouktif dkk., 2023).

DTSE merupakan metode representasi kondisi lalu lintas yang mempertahankan informasi spasial kendaraan melalui proses diskretisasi ruas jalan.

Setiap jalur masuk menuju persimpangan dibagi menjadi sejumlah sel dengan panjang tetap. Ukuran sel dapat disesuaikan dengan panjang kendaraan agar setiap sel hanya ditempati oleh paling banyak satu kendaraan. Berdasarkan pembagian tersebut, kondisi kendaraan pada setiap jalur dapat direpresentasikan dalam bentuk matriks numerik yang dapat diproses oleh jaringan saraf (Genders & Razavi, 2016; Zhang dkk., 2025).



(a) (b)

Gambar 2.8 Ilustrasi Representasi Status Lalu Lintas Menggunakan DTSE: (a) Segmentasi Jalur Pendekat Persimpangan ke dalam Sel-Sel Diskrit dan (b) Pembentukan Matriks Posisi Kendaraan serta Matriks Kecepatan Kendaraan

Gambar 2.8 menunjukkan proses pembentukan representasi status lalu lintas menggunakan DTSE. Pada Gambar 2.8(a), setiap jalur pendekat persimpangan dibagi menjadi sejumlah sel diskrit untuk merepresentasikan distribusi kendaraan secara spasial. Selanjutnya, sebagaimana ditunjukkan pada Gambar 2.8(b), kondisi kendaraan pada setiap sel dikonversi menjadi matriks posisi dan matriks kecepatan. Matriks posisi merepresentasikan keberadaan kendaraan pada setiap sel, sedangkan matriks kecepatan merepresentasikan kecepatan kendaraan yang telah dinormalisasi. Kedua matriks tersebut menjadi komponen utama dalam penyusunan tensor input bagi agen *deep reinforcement learning*. Berikut adalah penjelasan lengkap mengenai konsep DTSE dimana persamaan ini dirumuskan dalam

penelitian ini dengan mengacu pada (Bouktif dkk., 2023; Genders & Razavi, 2019) dan sirumuskan untuk penelitian ini:

A. Matriks Okupansi Kendaraan

Apabila jumlah jalur masuk dinyatakan sebagai (L) dan jumlah sel pada setiap jalur dinyatakan sebagai (C), matriks okupansi kendaraan pada waktu ke-(t) dirumuskan sebagai:

$$\mathbf{O}_t = \left[o_t^{(l,c)}\right] \in \{0,1\}^{L \times C} \tag{2.16}$$

$$o_t^{(l,c)} = \begin{cases} 1, & \text{jika terdapat kendaraan pada jalur ke-}l \text{ dan sel ke-}c \\ 0, & \text{jika sel tidak ditempati kendaraan} \end{cases}$$

dengan:

- $o_t^{(l,c)}$ adalah nilai okupansi pada jalur ke-l dan sel ke-c pada waktu ke-t;
- nilai 1 menunjukkan bahwa sel ditempati oleh kendaraan;
- nilai 0 menunjukkan bahwa sel tidak ditempati oleh kendaraan;
- l adalah indeks jalur masuk;
- c adalah indeks sel pada jalur;
- t adalah indeks waktu pengamatan.

Nilai 1 menunjukkan bahwa sel ditempati oleh kendaraan, sedangkan nilai 0 menunjukkan bahwa sel dalam kondisi kosong. Matriks okupansi mempertahankan informasi posisi relatif kendaraan terhadap garis henti sehingga agen dapat mengenali distribusi kendaraan pada setiap jalur pendekat.

B. Matriks Kecepatan Kendaraan

$$\mathbf{V}_t = \left[v_t^{(l,c)}\right] \in [0,1]^{L \times C} \tag{2.17}$$

dengan:

- $\mathbf{V}_t$ adalah matriks kecepatan kendaraan yang telah dinormalisasi pada waktu ke-t;
- $v_t^{(l,c)}$ adalah nilai kecepatan kendaraan yang telah dinormalisasi pada jalur ke-l dan sel ke-c;
- L adalah jumlah jalur masuk menuju persimpangan;

- C adalah jumlah sel diskrit pada setiap jalur;
- $[0,1]^{L\times C}$ menunjukkan bahwa matriks berukuran $L \times C$ dan setiap elemennya berada pada rentang 0 hingga 1.

Nilai kecepatan pada setiap sel dinormalisasi menggunakan persamaan:

$$v_t^{(l,c)} = \begin{cases} \frac{v_{t,\text{actual}}^{(l,c)}}{v_{\max}^{(l)}}, & \text{jika terdapat kendaraan pada sel ke-}(l,c) \\ 0, & \text{jika sel tidak ditempati kendaraan} \end{cases} \quad (2.18)$$

dengan:

- $v_t^{(l,c)}$ adalah nilai kecepatan kendaraan yang telah dinormalisasi pada jalur ke-l dan sel ke-c pada waktu ke-t;
- $v_{t,\text{actual}}^{(l,c)}$ adalah kecepatan aktual kendaraan pada jalur ke-l dan sel ke-c;
- $v_{\max}^{(l)}$ adalah batas kecepatan maksimum pada jalur ke-l;
- l adalah indeks jalur masuk;
- c adalah indeks sel pada jalur;
- t adalah indeks waktu pengamatan;
- nilai 0 diberikan apabila sel tidak ditempati kendaraan.

C. Penyusunan Tensor Spasial DTSE

Matriks okupansi dan matriks kecepatan digabungkan melalui proses penumpukan (*stacking*) pada dimensi fitur. Proses tersebut menghasilkan tensor spasial DTSE sebagai berikut:

$$\mathbf{X}_t^{\text{sp}} = \text{stack}(\mathbf{O}_t, \mathbf{V}_t) \in \mathbb{R}^{L\times C\times 2} \quad (2.19)$$

dengan:

- $\mathbf{X}_t^{\text{sp}}$ adalah tensor spasial DTSE pada waktu ke-t;9
- $^{\text{sp}}$ menunjukkan bahwa tensor memuat informasi spasial;
- $\text{stack}(\cdot)$ adalah operasi penumpukan beberapa matriks pada dimensi fitur;

- $\mathbf{O}_t$ adalah matriks okupansi kendaraan;
- $\mathbf{V}_t$ adalah matriks kecepatan kendaraan yang telah dinormalisasi;
- L adalah jumlah jalur masuk menuju persimpangan;
- C adalah jumlah sel diskrit pada setiap jalur;
- angka 2 menunjukkan jumlah kanal fitur spasial, yaitu kanal okupansi dan kanal kecepatan;
- $\mathbb{R}^{L\times C\times 2}$ menunjukkan bahwa tensor memiliki dimensi $L \times C \times 2$ dan elemennya berupa bilangan riil.

Secara umum, apabila DTSE menggunakan F fitur spasial pada setiap sel, bentuk tensor dinyatakan sebagai:

$$\mathbf{X}_t^{\mathrm{sp}} \in \mathbb{R}^{L\times C\times F} \tag{2.20}$$

Pada konfigurasi dasar, nilai (F=2) menunjukkan penggunaan fitur okupansi dan kecepatan kendaraan. Apabila fitur spasial tambahan dimasukkan, seperti waktu tunggu kendaraan pada setiap sel, nilai F bertambah sesuai dengan jumlah fitur yang digunakan.

D. Representasi Fase Sinyal Aktif

Fase sinyal aktif direpresentasikan menggunakan vektor *one-hot encoding*:

$$\mathbf{p}_t = \left[p_{t,1}, p_{t,2}, \dots, p_{t,P}\right]^T \in \{0,1\}^P \tag{2.21}$$

Vektor fase memenuhi ketentuan:

$$\sum_{j=1}^{P} p_{t,j} = 1 \tag{2.22}$$

dengan:

- $\mathbf{p}_t$ adalah vektor fase sinyal aktif pada waktu ke-(t);
- $p_{t,j}$ adalah elemen vektor yang menunjukkan status fase sinyal ke-j pada waktu ke-t;
- j adalah indeks fase sinyal dengan $j = 1,2, \dots, P$;
- P adalah jumlah alternatif fase sinyal yang direpresentasikan;
- ^{T} menunjukkan operasi transpose sehingga vektor disusun dalam bentuk kolom;

- $0,1^P$ menunjukkan bahwa setiap elemen vektor bernilai biner, yaitu 0 atau 1.

E. Pembentukan Vektor Input Jaringan Saraf

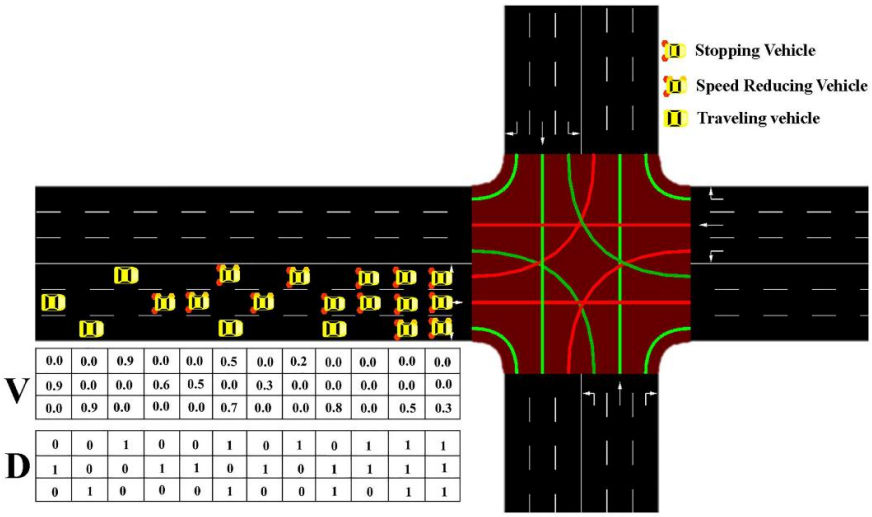
Tensor spasial DTSE selanjutnya diubah menjadi vektor satu dimensi melalui proses *flattening*. Vektor tersebut digabungkan dengan vektor fase sinyal aktif melalui operasi konkatenasi:

$$\mathbf{s}_t = \left[\mathrm{vec}\left(\mathbf{X}_t^{\mathrm{sp}}\right); \mathbf{p}_t\right] \in \mathbb{R}^{2LC+P} \tag{2.23}$$

dengan:

- $\mathbf{s}_t$ adalah vektor *state* akhir yang menjadi masukan jaringan saraf pada waktu ke-*t*;
- vec(.)adalah operasi *flattening* yang mengubah tensor menjadi vektor satu dimensi;
- $\mathbf{X}_t^{\mathrm{sp}}$ adalah tensor spasial DTSE yang memuat fitur okupansi dan kecepatan kendaraan;
- $\mathbf{p}_t$ adalah vektor fase sinyal aktif berbasis *one-hot encoding*;
- tanda titik koma menunjukkan operasi konkatenasi antar *f* fitur;
- *L* adalah jumlah jalur masuk menuju persimpangan;
- *C* adalah jumlah sel diskrit pada setiap jalur;
- *P* adalah jumlah alternatif fase sinyal;
- *2LC* adalah jumlah elemen fitur spasial yang diperoleh dari dua kanal fitur, yaitu okupansi dan kecepatan;
- *2LC+P* adalah jumlah elemen pada vektor input akhir;
- $\mathbb{R}^{2LC+P}$ menunjukkan bahwa vektor input akhir memiliki dimensi *2LC+P* dan elemennya berupa bilangan riil.

Persamaan 2.24 menunjukkan bahwa vektor input akhir terdiri atas *2LC* elemen fitur spasial dan *P* elemen fase sinyal aktif. Sebagai ilustrasi, apabila digunakan 8 jalur masuk, 20 sel pada setiap jalur, dan 4 fase sinyal, tensor spasial memiliki dimensi $8 \times 20 \times 2 = 320$. Setelah proses *flattening*, tensor tersebut menghasilkan 320 elemen. Penambahan empat elemen fase sinyal menghasilkan vektor input akhir dengan dimensi 320+ 4 = 324.



Gambar 2.9 Struktur Penyusunan Tensor DTSE dan Integrasinya sebagai Input Agen D3QN pada Sistem ATSC

F. Penanganan Antrean di Luar Rentang Observasi Utama

Penggunaan DTSE dengan rentang observasi terbatas perlu mempertimbangkan kondisi ketika antrean kendaraan melampaui batas pengamatan. Apabila rentang observasi ditetapkan sebesar 100 meter dan ukuran setiap sel sebesar 5 meter, setiap jalur masuk direpresentasikan oleh 20 sel diskrit. Konfigurasi tersebut menghasilkan informasi spasial yang rinci di sekitar garis henti. Namun, tensor DTSE pada zona utama belum dapat membedakan secara langsung antrean yang hanya sedikit melampaui 100 meter dan antrean yang memanjang jauh ke arah hulu (Faqir dkk., 2025).

Untuk mengatasi keterbatasan tersebut, representasi *state* dibentuk menggunakan dua zona pengamatan. Zona pertama merupakan zona dekat persimpangan (*near-intersection zone*) dengan rentang 0 hingga 100 meter dari garis henti. Zona ini direpresentasikan menggunakan DTSE dengan ukuran sel tetap sebesar 5 meter. Zona kedua merupakan zona hulu (*upstream zone*) yang mencakup bagian jalur setelah 100 meter hingga batas panjang jalur masuk. Zona hulu direpresentasikan menggunakan fitur agregat agar dimensi input tetap terkendali (H. Zhang dkk., 2025).

Pembagian zona pengamatan pada jalur ke-l dirumuskan sebagai berikut:

$$\mathcal{Z}_{\text{near}}^{(l)} = \{x \mid 0 \leq d(x) \leq D_{\text{near}}\} \tag{2.24a}$$

$$\mathcal{Z}_{\text{up}}^{(l)} = \{x \mid D_{\text{near}} < d(x) \leq D_l\} \tag{2.24b}$$

dengan:

- $\mathcal{Z}_{\text{near}}^{(l)}$ adalah zona pengamatan utama pada jalur ke-*l*;
- $\mathcal{Z}_{\text{up}}^{(l)}$ adalah zona pengamatan hulu pada jalur ke-*l*;
- $d(x)$ adalah jarak posisi kendaraan (x) terhadap garis henti;
- D_{near} adalah batas zona pengamatan utama, yaitu 100 meter;
- D_l adalah panjang jalur masuk ke-*l*.

Fitur tambahan pada zona hulu digabungkan menjadi vektor:

$$\mathbf{z}_t^{\text{up}} = \left[\boldsymbol{\rho}_{t,\text{up}}; \overline{\mathbf{v}}_{t,\text{up}}; \hat{\mathbf{d}}_{t,\text{lead}}; \mathbf{b}_{t,\text{ov}}\right] \in \mathbb{R}^{4L} \tag{2.25}$$

dengan:

- $\mathbf{z}_t^{\text{up}}$ adalah vektor fitur agregat pada zona hulu;
- $\boldsymbol{\rho}_{t,\text{up}}$ adalah vektor kepadatan antrean pada seluruh jalur masuk;
- $\overline{\mathbf{v}}_{t,\text{up}}$ adalah vektor kecepatan rata-rata kendaraan pada zona hulu;
- $\hat{\mathbf{d}}_{t,\text{lead}}$ adalah vektor jarak kendaraan terdepan yang telah dinormalisasi;
- $\mathbf{b}_{t,\text{ov}}$ adalah vektor indikator luapan antrean;
- *L* adalah jumlah jalur masuk;
- *4L* menunjukkan bahwa setiap jalur menghasilkan empat fitur agregat.

Vektor input akhir jaringan saraf dibentuk melalui konkatenasi tensor spasial DTSE pada zona utama, vektor fitur zona hulu, dan vektor fase sinyal aktif:

$$\mathbf{s}_t = \left[\text{vec}\left(\mathbf{X}_{t,\text{near}}^{\text{sp}}\right); \mathbf{z}_t^{\text{up}}; \mathbf{p}_t\right] \in \mathbb{R}^{2LC+4L+P} \tag{2.26}$$

dengan:

- $\mathbf{s}_t$adalah vektor *state* akhir pada waktu ke-(t);
- $\mathbf{X}_{t,\text{near}}^{\text{sp}}$ adalah tensor spasial DTSE pada zona utama;
- vec(.) adalah operasi *flattening*;
- $\mathbf{z}_t^{\text{up}}$ adalah vektor fitur agregat pada zona hulu;
- $\mathbf{p}_t$ adalah vektor fase sinyal aktif;

- *L* adalah jumlah jalur masuk;
- *C* adalah jumlah sel pada setiap jalur di zona utama;
- *P* adalah jumlah alternatif fase sinyal;
- *2LC+4L+P* adalah dimensi vektor input akhir.

Representasi dua zona memungkinkan agen membedakan tingkat kepadatan antrean yang melampaui batas 100 meter tanpa memperbesar jumlah sel secara berlebihan. Informasi okupansi dan kecepatan pada zona utama menghasilkan ketelitian spasial di sekitar garis henti. Sementara itu, informasi kepadatan, kecepatan rata-rata, jarak kendaraan terdepan, dan indikator luapan antrean pada zona hulu menghasilkan konteks tambahan mengenai kondisi lalu lintas di luar batas pengamatan utama.

Keputusan fase lampu hijau tidak hanya ditentukan oleh keberadaan kendaraan pada sel-sel terdekat dari garis henti. Agen menghasilkan keputusan berdasarkan kombinasi informasi zona utama dan zona hulu. Apabila zona utama masih kosong, tetapi zona hulu menunjukkan kepadatan kendaraan yang tinggi dengan kecepatan rendah, agen mengenali adanya antrean atau kelompok kendaraan yang mendekati persimpangan. Informasi tersebut memungkinkan agen mengantisipasi kebutuhan fase hijau sebelum kendaraan memasuki zona utama.

Apabila antrean telah melampaui batas 100 meter, indikator luapan antrean bernilai 1. Kondisi tersebut meningkatkan urgensi pelayanan pada jalur terkait karena penundaan tambahan berpotensi memperpanjang antrean dan memicu *queue spillback*. Sebaliknya, apabila kendaraan masih berjarak lebih dari 100 meter tetapi bergerak dengan kecepatan relatif tinggi dan kepadatannya rendah, agen tidak harus segera memberikan fase hijau. Agen dapat mempertahankan fase aktif untuk melayani jalur lain yang memiliki kebutuhan lebih mendesak.

Keputusan akhir tetap dihasilkan melalui nilai $Q(s_t, a)$ untuk setiap alternatif aksi fase sinyal. Agen memilih aksi dengan nilai Q tertinggi berdasarkan pola yang dipelajari selama pelatihan. Penerapan durasi minimum hijau, durasi maksimum hijau, dan fase kuning tetap digunakan untuk mencegah perubahan fase yang terlalu

cepat serta menjaga keselamatan pergerakan kendaraan. Dengan demikian, model tidak hanya merespons jumlah kendaraan, tetapi juga mempertimbangkan tingkat kepadatan, kecepatan pergerakan, jarak kendaraan terhadap garis henti, dan potensi luapan antrean.

G. Integrasi DTSE dalam Pengendalian Sinyal Lalu Lintas Adaptif

Dalam pengendalian sinyal lalu lintas berbasis DRL, vektor *state* $\mathbf{s}_t$ menjadi masukan jaringan saraf untuk menghasilkan nilai $Q(s_t, a)$ bagi seluruh alternatif aksi fase sinyal. Agen memilih aksi berdasarkan nilai Q dan strategi eksplorasi yang digunakan. Aksi terpilih diterapkan pada lampu lalu lintas, kemudian kondisi lingkungan berubah dan menghasilkan *state* berikutnya.

Pada sistem multi-agen, setiap agen memperoleh DTSE lokal dari persimpangan yang dikendalikannya. Informasi lokal tersebut dapat dipertukarkan atau diproses melalui mekanisme komunikasi antaragen untuk mendukung koordinasi pada tingkat jaringan. Dengan demikian, DTSE tidak hanya menyediakan representasi kondisi lalu lintas yang konsisten, tetapi juga menjadi dasar pembentukan persepsi lokal setiap agen dalam sistem MARL.

Selain DTSE, literatur juga menggunakan beberapa pendekatan lain untuk merepresentasikan kondisi lalu lintas, antara lain:

1. *Image-based encoding*, yaitu representasi kondisi lalu lintas sebagai citra dua dimensi;
2. *Graph-based state representation*, yaitu representasi hubungan antarpersimpangan menggunakan struktur graf;
3. *Vectorized statistical representation*, yaitu penggunaan nilai agregat, seperti jumlah kendaraan, waktu tunggu rata-rata, atau panjang antrean.

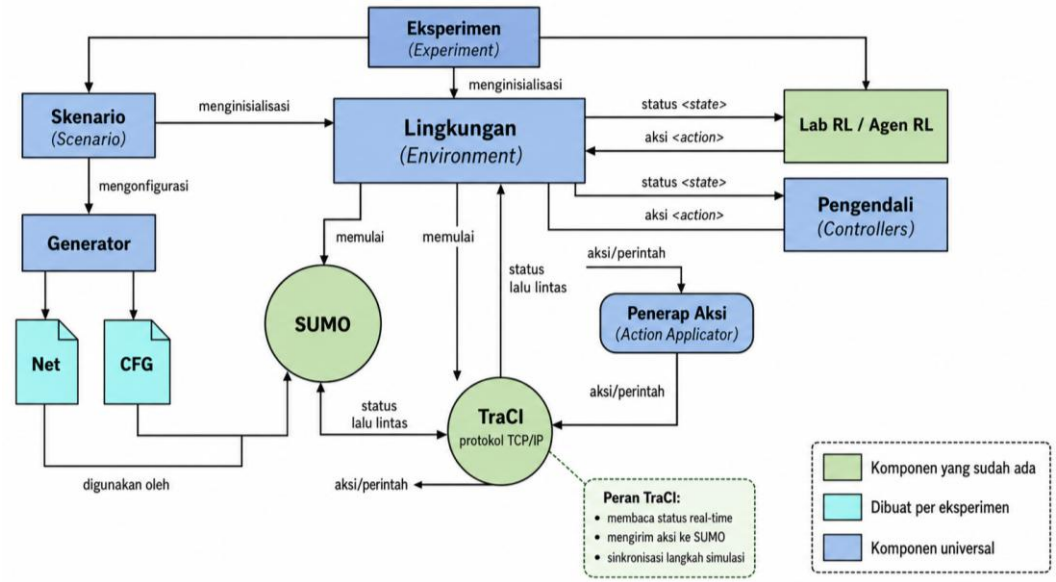
DTSE dipilih karena mampu mempertahankan informasi spasial kendaraan tanpa memerlukan proses pengolahan citra yang kompleks. Representasi tersebut menghasilkan kompromi yang proporsional antara kelengkapan informasi dan efisiensi komputasi. DTSE juga menghasilkan struktur input berdimensi tetap sehingga sesuai digunakan sebagai masukan jaringan saraf pada agen D3QN dan sistem MARL

2.2.6 Simulasi *Simulation of Urban Mobility* (SUMO)

Simulation of Urban Mobility (SUMO) merupakan simulator mikroskopis lalu lintas bersifat *open-source* yang dikembangkan oleh Deutsches Zentrum für Luft- und Raumfahrt (DLR) (Krajzewicz, 2010), Jerman. SUMO dirancang untuk memodelkan pergerakan individual kendaraan secara detail di dalam jaringan jalan, dengan mempertimbangkan interaksi antar kendaraan, perubahan fase sinyal, serta dinamika lalu lintas secara waktu nyata (Lopez dkk., 2018). Karena sifatnya yang modular dan fleksibel, SUMO banyak digunakan sebagai *platform* eksperimental dalam pengembangan dan pengujian sistem pengendalian sinyal lalu lintas berbasis kecerdasan buatan, termasuk algoritma DRL.

Pada dasarnya, SUMO bekerja berdasarkan model mikroskopis, di mana setiap kendaraan dimodelkan sebagai entitas independen dengan perilaku spesifik terhadap percepatan, deselerasi, jarak aman, dan kecepatan maksimal. Selain karakteristik pergerakan individual, SUMO juga dapat membedakan kelas kendaraan melalui definisi tipe kendaraan atau *vehicle type* (<vType>) dan atribut kelas kendaraan (vClass) pada berkas rute .rou.xml. Melalui mekanisme tersebut, kendaraan dalam simulasi dapat diklasifikasikan ke dalam beberapa jenis, seperti mobil penumpang (passenger), sepeda motor (motorcycle), bus (bus), truk (truck), kendaraan pengiriman (delivery), taksi (taxi), dan kendaraan darurat (emergency). Setiap kelas kendaraan dapat diberikan parameter operasional yang berbeda, seperti panjang kendaraan, percepatan, deselerasi, kecepatan maksimum, konsumsi bahan bakar, model emisi, serta hak akses terhadap lajur tertentu. Dengan demikian, SUMO tidak hanya mampu menggambarkan jumlah dan pergerakan kendaraan secara umum, tetapi juga mendukung simulasi lalu lintas heterogen berdasarkan perbedaan karakteristik kendaraan.

Berbeda dari model makroskopis yang hanya meninjau kepadatan lalu lintas secara agregat, pendekatan mikroskopis memungkinkan peneliti menganalisis dampak perubahan sinyal terhadap perilaku kendaraan individual, sehingga hasil simulasi lebih realistis dan dapat digunakan untuk mengevaluasi performa sistem ATSC secara presisi. Gambar 2.10 merupakan struktur arsitektur kerangka kerja simulasi SUMO dan TraCI.



Gambar 2.10 Struktur dasar arsitektur simulasi SUMO dan integrasinya dengan modul *Reinforcement Learning*

Arsitektur kerangka kerja simulasi yang digunakan menempatkan komponen *environment* (lingkungan) sebagai pusat interaksi, sebagaimana ditunjukkan pada Gambar 2.10. Sebuah *experiment* (eksperimen) menginisialisasi *environment* berdasarkan *scenario* (skenario) tertentu. Skenario tersebut, melalui *generator*, menghasilkan berkas konfigurasi jaringan (*Net*) dan simulasi (*CFG*) yang digunakan oleh SUMO. Agen, baik berupa *RL Lab* maupun *controllers*, tidak berinteraksi langsung dengan SUMO. Sebaliknya, agen menerima data <state> (status) dari *environment* dan mengirimkan <action> (aksi) kembali ke *environment*. *Environment* inilah yang bertugas menjalankan simulator SUMO dan antarmuka TraCI, serta menerapkan aksi yang dipilih agen ke dalam simulasi (melalui *action applicator*).

Komunikasi teknis di lapisan bawah antara *environment* dan simulator SUMO dijembatani oleh TraCI, seperti terlihat pada Gambar 2.10. TraCI adalah protokol berbasis TCP/IP yang memungkinkan interaksi dua arah antara SUMO dan program eksternal seperti Python, Java, atau C++. Melalui TraCI, program eksternal (dalam hal ini, *environment*) dapat:

1. Mengambil status lingkungan secara *real-time*, seperti jumlah kendaraan, posisi, kecepatan, panjang antrean, dan status fase lampu lalu lintas.
2. Mengirimkan aksi (perintah) ke SUMO, seperti mengubah fase sinyal, memperpanjang waktu hijau, atau memaksa transisi fase tertentu.
3. Mengatur kecepatan simulasi (*simulation step length*) untuk sinkronisasi dengan waktu pelatihan model RL.

Dalam konteks sistem MO-CL-D3QN, setiap agen (yang mengontrol satu persimpangan) berinteraksi dengan lingkungan SUMO melalui siklus *sense–act–learn* berikut:

1. Agen mengambil status lalu lintas dari TraCI sensor data yang telah dikodekan ke dalam format DTSE.
2. Agen menghitung nilai aksi $Q(s, a)$ menggunakan arsitektur D3QN, memilih aksi optimal dengan kebijakan $\varepsilon - greedy$.
3. Agen mengirimkan aksi ke SUMO melalui TraCI untuk mengubah fase sinyal.
4. SUMO menjalankan langkah simulasi berikutnya dan mengembalikan *reward* (misalnya pengurangan waktu tunggu atau panjang antrean).

Mekanisme ini memastikan pembelajaran agen berlangsung secara sinkron dengan simulasi dinamis lalu lintas (*on-policy training*). Dalam penelitian berbasis SUMO, konfigurasi simulasi didefinisikan melalui berkas *.sumocfg* yang berisi parameter jaringan, rute kendaraan, logika sinyal, dan skenario waktu. Komponen utama dalam konfigurasi ini meliputi:

1. *Network File* (.net.xml) – mendeskripsikan topologi jaringan jalan, panjang jalur, batas kecepatan, dan koneksi antar-persimpangan.
2. *Route File* (.rou.xml) – menentukan jumlah kendaraan, rute perjalanan, waktu kedatangan, serta distribusi arus lalu lintas. Pada berkas .rou.xml, tipe dan kelas kendaraan juga dapat didefinisikan melalui elemen <vType> dan atribut vClass, sehingga skenario simulasi dapat membedakan kendaraan

penumpang, sepeda motor, bus, truk, maupun kendaraan darurat sesuai kebutuhan eksperimen.

3. *Traffic Light Logic File* (.add.xml) – berisi definisi fase sinyal, durasi minimum/ maksimum hijau, serta urutan transisi antar fase.
4. *Configuration File* (.sumocfg) – mengatur parameter simulasi global, termasuk simulation step length, *output* log, dan integrasi dengan TraCI client.

Konfigurasi lingkungan simulasi SUMO mencakup parameter yang berkaitan dengan interval pembaruan simulasi, jangkauan observasi kendaraan, representasi DTSE, durasi fase sinyal, waktu simulasi, dan kepadatan arus lalu lintas. Parameter tersebut digunakan untuk membangun lingkungan pengujian yang konsisten bagi seluruh model pengendalian sinyal lalu lintas. Konfigurasi umum lingkungan simulasi SUMO disajikan pada Tabel 2.2.

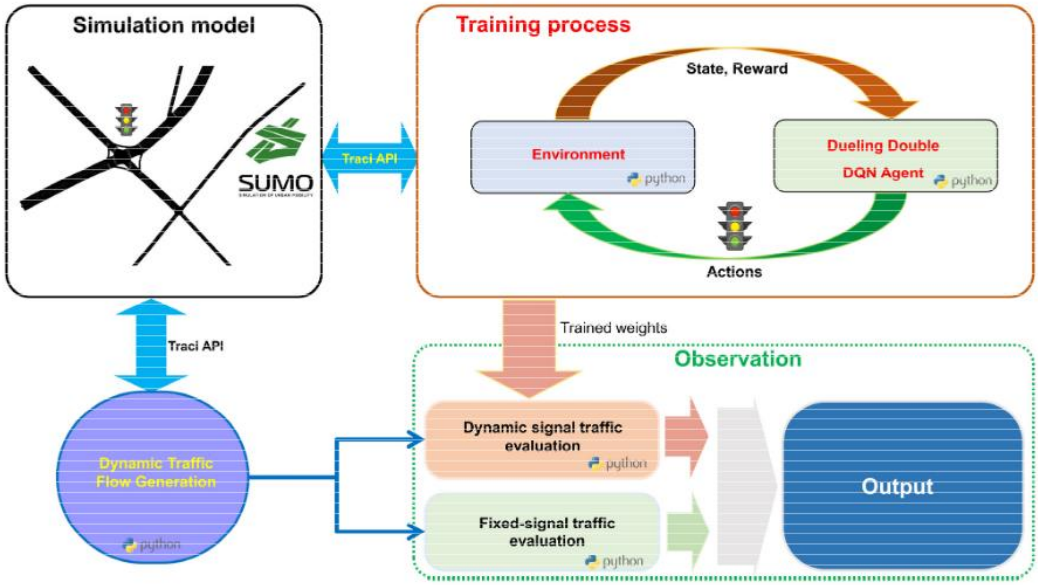
Tabel 2.2 Parameter Umum Konfigurasi SUMO untuk Simulasi Adaptive Traffic Signal Control

No	Parameter	Simbol / Nilai Umum	Deskripsi
1	Simulation step length	1.0 s	Interval pembaruan kondisi lingkungan simulasi.
2	Near-intersection observation range	100 m	Zona utama DTSE dengan sel berukuran tetap
3	Upstream monitoring range	Setelah 100 m hingga batas jalur masuk	Zona tambahan untuk membaca kepadatan, kecepatan rata-rata, jarak kendaraan terdepan, dan luapan antrean
4	Lane cell size	5 m	Panjang setiap sel pada representasi DTSE.
5	Minimum green time	10 s	Durasi minimum fase hijau sebelum pergantian fase diperbolehkan.
6	Maximum green time	60 s	Durasi maksimum fase hijau pada setiap fase sinyal.
7	Yellow phase	3 s	Durasi transisi antar fase
8	Simulation time per episode	3600 s	Durasi simulasi (1 jam)
9	Vehicle spawn rate	400–1200 veh/hr	Kepadatan lalu lintas per arah

Dalam kerangka penelitian ini, SUMO berfungsi sebagai *environment simulator* yang menyediakan data dinamis untuk proses pembelajaran agen DRL.

Sistem ini menggunakan pendekatan *decentralized multi-agent*, di mana setiap persimpangan dijalankan oleh agen independen yang berinteraksi melalui kondisi lalu lintas sekitar (spasial). Langkah-langkah pelatihan pada sistem ini dapat dilihat pada Gambar 2.11 adalah sebagai berikut:

1. Inisialisasi — semua agen dimulai dengan bobot jaringan acak dan parameter D3QN awal.
2. *State encoding* — agen mengonversi data dari SUMO menjadi DTSE tensor berukuran tetap (misalnya 4 jalur × 20 sel).
3. *Action selection* — agen memilih fase sinyal berikut berdasarkan nilai Q tertinggi atau eksplorasi ε.
4. *Environment update* — aksi dikirim ke SUMO via TraCI, simulasi dijalankan satu langkah.
5. *Reward computation* — sistem menghitung reward multi-objektif (delay, queue, throughput, fuel consumption).
6. *Learning update* — agen memperbarui bobot D3QN berdasarkan Bellman target dan *experience replay*.



Gambar 2.11 Alur Proses Pelatihan Agen pada Simulator SUMO

Melalui siklus ini, SUMO berperan sebagai *laboratorium virtual* yang memungkinkan evaluasi sistem ATSC secara sistematis di bawah berbagai skenario lalu lintas, misalnya *peak-hour congestion*, *imbalanced flow*, atau *incident blockage*.

2.2.7 Metrik Evaluasi Kinerja Sistem ATSC

Metrik evaluasi memainkan peran penting dalam menilai efektivitas dan efisiensi sistem ATSC, khususnya dalam konteks *multi-agent deep reinforcement learning* (MO-CL-D3QN). Dalam penelitian ini, beberapa metrik kunci yang digunakan untuk mengevaluasi kinerja sistem ATSC meliputi *average waiting time* (AWT), *queue length* (QL), *throughput* (TP), dan *fuel consumption* (FC). Setiap metrik ini mencerminkan aspek berbeda dari kinerja lalu lintas, baik dari segi efisiensi, kualitas aliran kendaraan, dan dampak lingkungan.

1. Average Waiting Time (AWT)

   Waktu tunggu rata-rata (*Average Waiting Time, AWT*) merupakan metrik yang mengukur rata-rata waktu yang dihabiskan oleh kendaraan di persimpangan. Semakin rendah AWT, semakin efisien sistem pengendalian sinyal tersebut. AWT dihitung sebagai berikut:

   $$AWT = \frac{1}{N}\sum_{i=1}^{N} w_i \tag{2.21}$$

   di mana:

   - w_i adalah waktu tunggu kendaraan ke-i,
   - N adalah total jumlah kendaraan yang melewati persimpangan selama simulasi.

   Contoh Perhitungan AWT:

   Misalkan dalam simulasi terdapat 5 kendaraan dengan waktu tunggu masing-masing sebagai berikut: $w_1 = 10\text{ s}, w_2 = 8\text{ s}, w_3 = 12\text{ s}, w_4 = 9\text{ s}, w_5 = 11\text{ s}$. Maka, AWT dihitung sebagai:

   $$AWT = \frac{1}{5}(10 + 8 + 12 + 9 + 11) = \frac{50}{5} = 10\text{ s}$$

2. Queue Length (QL)

   Panjang antrean (*Queue Length, QL*) mengukur jumlah kendaraan yang menunggu di persimpangan pada setiap waktu tertentu. Metrik ini sangat penting untuk mengetahui kapasitas persimpangan dan apakah kendaraan terlalu lama menunggu. QL dihitung dengan rumus:

$$QL = \frac{1}{T}\sum_{t=1}^{T} n_t \tag{2.22}$$

   di mana:

   - n_t adalah jumlah kendaraan yang berada dalam antrean pada waktu t,
   - T adalah total waktu simulasi.

   Contoh Perhitungan QL:

   Misalkan selama simulasi satu jam (3600 detik), antrean diukur pada 10 titik waktu (dalam detik), dan jumlah kendaraan yang berada di antrean pada setiap titik waktu adalah: $n_1 = 4, n_2 = 5, n_3 = 3, \ldots, n_{10} = 6$. Maka, QL dihitung sebagai:

$$QL = \frac{1}{10}(4 + 5 + 3 + 4 + 6 + 5 + 7 + 3 + 5 + 6) = \frac{48}{10} = 4.8 \text{ kendaraan}$$

3. Throughput (TP)

   Throughput (*TP*) mengukur jumlah kendaraan yang berhasil melewati persimpangan dalam waktu tertentu. Metrik ini menunjukkan efisiensi sistem dalam mengalirkan kendaraan. TP dihitung sebagai berikut:

$$TP = \frac{N_{\text{out}}}{T} \tag{2.23}$$

   di mana:

   - N_{out} adalah jumlah kendaraan yang berhasil melewati persimpangan selama waktu T,
   - T adalah durasi simulasi.

   Contoh Perhitungan TP:

   Misalkan selama simulasi, 300 kendaraan berhasil melewati persimpangan dalam waktu 3600 detik. Maka, TP dihitung sebagai:

$$TP = \frac{300}{3600} = 0.0833 \text{ kendaraan/detik} = 300 \text{ kendaraan/jam}$$

4. Fuel Consumption (FC)

   Konsumsi bahan bakar (*Fuel Consumption, FC*) mengukur jumlah bahan bakar yang digunakan oleh kendaraan selama simulasi, yang penting untuk mengukur dampak lingkungan dari sistem pengendalian lalu lintas. FC dihitung berdasarkan waktu kendaraan berjalan, kecepatan, dan jenis kendaraan. Secara umum, FC dapat dihitung dengan persamaan berikut:

$$FC = \sum_{i=1}^{N} (f_i \cdot d_i) \tag{2.30}$$

   di mana:

   - f_i adalah konsumsi bahan bakar kendaraan ke-i (misalnya dalam liter per km),
   - d_i adalah jarak yang ditempuh kendaraan ke-i,
   - N adalah jumlah kendaraan.

   Contoh Perhitungan FC:

   Misalkan terdapat 3 kendaraan yang masing-masing memiliki konsumsi bahan bakar 0.06 liter/km dan menempuh jarak 5 km, 7 km, dan 6 km. Maka, FC dihitung sebagai:

$$FC = (0.06 \times 5) + (0.06 \times 7) + (0.06 \times 6) = 0.3 + 0.42 + 0.36 = 1.08 \text{ lite}$$