OPTIMASI KONSUMSI ENERGI KENDARAAN LISTRIK MENGGUNAKAN KONTROL MIMO BERBASIS *DEEP REINFORCEMENT LEARNING* PADA KONDISI BERKENDARA DINAMIS

Yusuf Rahman Maulana
Program Studi S-1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia
e-mail : yusuf.22085@mhs.unesa.ac.id

Puput Wanarti Rusimamto
Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia
e-mail : puputwanarti@unesa.ac.id

Abstrak

Meningkatnya adopsi kendaraan listrik menuntut adanya sistem manajemen energi yang lebih cerdas untuk memperpanjang jarak tempuh dan menjaga kesehatan baterai. Penelitian ini bertujuan untuk mengoptimalkan konsumsi energi pada EV melalui implementasi kontroler cerdas berbasis Reinforcement Learning dengan algoritma Soft Actor-Critic (SAC) dalam kerangka kerja Multiple Input Multiple Output (MIMO). Berbeda dengan kontroler konvensional yang umumnya hanya berfokus pada pelacakan kecepatan, arsitektur MIMO yang diusulkan mengintegrasikan variabel State of Charge (SOC) dan suhu baterai sebagai feedback untuk mencapai efisiensi energi yang optimal. Pemodelan sistem dilakukan menggunakan lingkungan MATLAB/Simulink dengan standar siklus berkendara FTP-75. Hasil simulasi menunjukkan bahwa model fisik kendaraan berhasil divalidasi dengan tingkat presisi pelacakan kecepatan yang tinggi. Proses pelatihan agen SAC menunjukkan keberhasilan konvergensi pada episode ke-2060 dengan peningkatan nilai imbalan (reward) yang signifikan, mencapai puncak sebesar 851,5. Hal ini membuktikan bahwa agen cerdas mampu mengadopsi kebijakan berkendara adaptif yang tidak hanya mempertahankan performa traksi, tetapi juga secara efektif memitigasi pengurasan daya berlebih. Penelitian ini menyimpulkan bahwa integrasi RL dalam arsitektur MIMO efektif dalam meningkatkan efisiensi operasional kendaraan listrik melalui manajemen energi yang lebih responsif dan presisi.
Kata Kunci: *Optimasi, konsumsi energi kendaraan Listrik, Kontrol MIMO, Deep Reinforcement Learning*

Abstract

The rising adoption of Electric Vehicles demands smarter energy management systems to extend driving range and maintain battery health. This study aims to optimize energy consumption in EVs through the implementation of an intelligent controller based on Reinforcement Learning using the Soft Actor-Critic (SAC) algorithm within a Multiple Input Multiple Output (MIMO) framework. Unlike conventional controllers that typically focus solely on velocity tracking, the proposed MIMO architecture integrates State of Charge (SOC) and battery temperature variables as feedback to achieve optimal energy efficiency. System modeling was performed using the MATLAB/Simulink environment under the FTP-75 drive cycle standard. Simulation results show that the physical vehicle model was successfully validated with high velocity tracking precision. The SAC agent training process demonstrated successful convergence at the 2060th episode with a significant increase in reward value, peaking at 851.5. This proves that the intelligent agent is capable of adopting an adaptive driving policy that not only maintains traction performance but also effectively mitigates excessive power drainage. This study concludes that RL integration within a MIMO architecture is effective in improving the operational efficiency of electric vehicles through more responsive and precise energy management.
Keywords: *Optimization, Electric Vehicle Energy Consumption, MIMO Control, Deep Reinforcement Learning*

PENDAHULUAN

Transisi global menuju energi berkelanjutan telah mendorong inovasi signifikan dalam industri otomotif, dengan EV menjadi garda terdepan dalam upaya dekarbonisasi sektor transportasi (P. Wang et al., 2019). Kekhawatiran terhadap perubahan iklim akibat emisi gas rumah kaca dari sektor transportasi yang menyumbang 23% dari total emisi CO_2 terkait energi global, dengan 70% berasal dari kendaraan jalan raya, mendorong keinginan untuk mengurangi ketergantungan pada bahan bakar fosil (Tbk., 2020). Di Amerika Serikat sendiri, sektor transportasi menyumbang 28% dari total emisi gas rumah kaca, menjadikannya kontributor terbesar emisi nasio (Tbk., 2024). Kendaraan listrik menawarkan solusi transportasi yang ramah lingkungan dengan potensi mengurangi total emisi hingga 63% dibandingkan dengan kendaraan bermesin pembakaran internal selama siklus hidup kendaraan, serta memberikan manfaat tambahan

berupa pengurangan kebisingan dan peningkatan kualitas udara perkotaan (Padhilah et al., 2025).

Namun, di balik keunggulannya, tantangan utama yang masih menghambat penerimaan massal kendaraan listrik adalah keterbatasan jarak tempuh dan kecemasan pengguna terhadap daya tahan baterai, yang dikenal sebagai "range anxiety" (D. Wang et al., 2023). Konsumsi energi pada kendaraan listrik sangat dipengaruhi oleh berbagai faktor dinamis seperti gaya mengemudi, kondisi lalu lintas, topografi jalan, yang dapat mengurangi jangkauan kendaraan hingga 15-25% (Huang et al., 2024). Manajemen energi yang tidak efisien dapat secara drastis mengurangi jarak tempuh efektif kendaraan, menurunkan kepraktisan dan keandalannya dalam penggunaan sehari-hari (Xu et al., 2022). Strategi manajemen energi berbasis pembelajaran mendalam telah menunjukkan peningkatan efisiensi hingga 20% dibandingkan dengan rute perjalanan konvensional dan 12% dibandingkan dengan eco-driving standar (Xu et al., 2022).

Berdasarkan latar belakang tersebut, beberapa masalah diidentifikasi berikut ini. 1) Konsumsi energi yang belum optimal pada kendaraan listrik menyebabkan jarak tempuh yang terbatas dan menimbulkan range anxiety bagi pengguna. 2) Sifat sistem kendaraan listrik yang kompleks sebagai sistem MIMO (Multi-Input Multi-Output) memerlukan strategi kontrol yang terkoordinasi untuk mencapai efisiensi tinggi. 3) Metode kontrol konvensional kurang adaptif terhadap kondisi berkendara yang dinamis dan bervariasi, sehingga sulit mencapai optimasi energi yang sesungguhnya. 4) Diperlukannya sebuah metode kontrol cerdas yang dapat belajar dan beradaptasi secara real-time untuk memaksimalkan efisiensi energi pada berbagai skenario berkendara.

Ruang lingkup masalah dibatasi pada hal-hal berikut ini. 1) Penelitian ini berfokus pada optimasi energi pada sistem powertrain kendaraan listrik, yang meliputi motor listrik, baterai, dan kontroler. 2) Pengembangan, pelatihan, dan pengujian sistem kontrol dilakukan sepenuhnya dalam lingkungan simulasi perangkat lunak MATLAB/Simulink. 3) Metode machine learning yang digunakan terbatas pada algoritma Deep Reinforcement Learning (DRL). 4) Kontrol MIMO yang digunakan terbatas hanya tiga input dan juga tiga output yang mementingkan tiga faktor utama dalam kendaraan listrik. 5) Model kendaraan listrik yang digunakan dalam simulasi merupakan model matematis yang disederhanakan dan tidak mempertimbangkan semua dinamika kendaraan secara mendetail seperti degradasi baterai atau pengaruh suhu ekstrem. Sehingga fokus penelitian ini adalah merancang dan mengimplementasikan algoritma Deep Reinforcement Learning (DRL) untuk mengelola strategi distribusi energi pada kendaraan listrik secara otomatis, guna meminimalkan konsumsi daya tanpa mengurangi performa kendaraan. Deep Reinforcement Learning (DRL) memiliki beberapa keunggulan dibandingkan metode kontrol konvensional seperti PID, terutama ketika diterapkan pada sistem yang kompleks seperti kendaraan listrik. PID bekerja berdasarkan hubungan antara error sistem dan tiga parameter utama yang harus dituning agar menghasilkan respons kontrol yang diinginkan. Pendekatan ini efektif untuk sistem yang karakteristiknya relatif stabil dan sederhana, tetapi dapat mengalami keterbatasan ketika sistem memiliki banyak variabel yang saling berinteraksi, bersifat nonlinear, atau memiliki kondisi operasi yang berubah-ubah.

Keunggulan lain DRL adalah kemampuannya menangani sistem Multi-Input Multi-Output (MIMO). Pada kendaraan listrik, beberapa parameter seperti kecepatan, torsi motor, arus baterai, tegangan baterai, dan State of Charge (SOC) dapat saling memengaruhi. Pada metode PID, kondisi seperti ini biasanya membutuhkan beberapa loop kontrol yang harus dituning dan dikoordinasikan secara terpisah. Semakin kompleks hubungan antarvariabel, semakin sulit menjaga performa seluruh loop secara bersamaan. DRL dapat mempelajari keterkaitan antarvariabel tersebut sebagai satu kesatuan sehingga lebih sesuai untuk permasalahan kontrol multivariabel.

TINJAUAN PUSTAKA

Penelitian yang dilakukan oleh Mei et al., (2023) berfokus pada manajemen energi kendaraan hybrid listrik (PHEV) multi-mode menggunakan kontrol MIMO berbasis multi-agent deep reinforcement learning (MADRL). Penelitian ini menggunakan algoritma Deep Deterministic Policy Gradient (DDPG) dengan strategi hand-shaking yang memungkinkan dua agen pembelajaran bekerja secara kolaboratif. Hasil penelitian menunjukkan penghematan energi hingga 4% dibandingkan sistem single-agent dan penghematan 23,54% dibandingkan sistem berbasis aturan konvensional (rule-based). Namun, penelitian ini fokus pada kendaraan hybrid dengan dua sumber energi (engine dan battery) dan belum mengeksplorasi penerapan pada kendaraan listrik full-electric (BEV) murni.

Xu et al., (2022) menyajikan tinjauan komprehensif tentang reinforcement learning untuk manajemen energi kendaraan listrik, membandingkan berbagai algoritma DRL seperti Q-learning, DDPG, TD3, dan SAC. Penelitian ini mengidentifikasi bahwa SAC (Soft Actor-Critic) menunjukkan performa superior dalam hal sample efficiency pada tugas-tugas kontrol kompleks, dan Twin Delayed Deep Deterministic Policy Gradient (DDPG). Twin Delayed DDPG lebih efektif dalam mengurangi overestimation bias. Namun, tinjauan ini masih bersifat literatur dan tidak mengintegrasikan kontrol MIMO secara eksplisit.

Kemudian Zhang, (Zhang & Li, 2021) mendesain sistem kontrol MIMO 3×3 untuk sistem tangki terkopel dengan menggunakan metode PID dan decoupling. Hasil penelitian menunjukkan bahwa penambahan strategi decoupling dapat mengurangi Integral Absolute Error (IAE) sebesar 33% dibandingkan tanpa decoupling. Penelitian ini memberikan fondasi penting tentang kompleksitas sistem MIMO coupling, namun aplikasinya masih terbatas pada sistem industrial sederhana dan belum diterapkan pada sistem dinamis nonlinear kendaraan listrik.

Lalu ada juga dari Parvin et al., (2021) yang melakukan studi komparatif tentang transfer learning untuk energy management strategies (EMS) berbasis DRL pada hybrid electric vehicles. Penelitian ini

membandingkan berbagai metode eksplorasi DDPG dalam konteks transfer learning antar driving cycles. Hasil menunjukkan bahwa kombinasi transfer learning dengan DDPG dapat mempercepat konvergensi training namun masih menghadapi keterbatasan dalam generalisasi antar kondisi driving yang sangat berbeda.

A. Kontrol MIMO

Sistem kendaraan listrik modern secara fundamental merupakan sistem multi-input multi-output (MIMO), di mana berbagai sinyal masukan harus diproses secara simultan untuk menghasilkan keluaran terkoordinasi yang optimal. Sistem MIMO didefinisikan sebagai sistem dengan lebih dari satu masukan dan lebih dari satu keluaran, berbeda dengan sistem single-input single-output (SISO) yang hanya menangani satu masukan untuk satu keluaran (Yue et al., 2021). Dalam konteks optimasi energi kendaraan listrik, sistem MIMO mencakup berbagai masukan utama, yaitu 1) Kondisi baterai yang mencakup state of charge (SOC), state of health (SOH). 2) Variabel state kendaraan seperti kecepatan kendaraan dan percepatan longitudinal. 3) Battery Power (Othaganont, 2017; Guzzella & Sciarretta, 2018).

Keluaran dari sistem MIMO EV energy optimization mencakup: 1) Over-actuated EV yang mengalokasikan daya antar motor, 2) Torsi motor yang harus diatur untuk memenuhi perintah pengemudi sambil meminimalkan losses, 3) Sinyal kontrol power split dalam sistem hybrid atau over-actuated EV yang mengalokasikan daya antar motor (Szumska, 2025). Karakteristik utama sistem MIMO pada kendaraan listrik adalah adanya coupling yang kuat antar variabel, di mana perubahan pada satu variabel kontrol akan mempengaruhi multiple output variables secara bersamaan (Peng & He, 2025). Struktur control MIMO ditunjukkan pada Gambar 1 berikut ini.

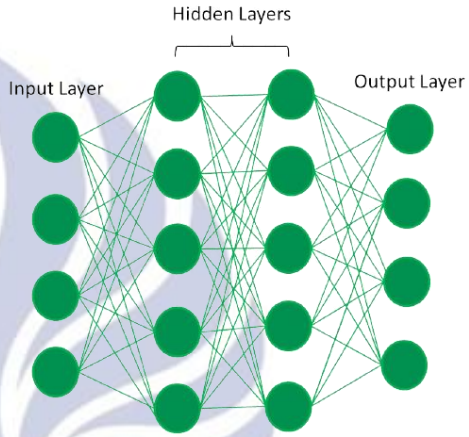
Gambar 1. Struktur Kontrol MIMO

Kontrol MIMO berarti memiliki banyak masukan dan banyak keluaran, dimana input maupun output dari struktur tersebut harus memiliki jumlah lebih dari 1 agar bisa disebut kontrol MIMO. Sama seperti pada Gambar 1 tersebut dimana inputnya berjumlah 2 variabel sementara output dari kontrol tersebut berjumlah 3 variabel.

Struktur MIMO bisa bermacam-macam tergantung bagaimana kita ingin membuatnya, namun poin utama dari Kontrol MIMO adalah jumlah variabel dari input atau output harus memiliki jumlah lebih dari 1 variabel.

B. Deep Learning

Deep learning adalah subdomain dari machine learning dan artificial intelligence yang menggunakan artificial neural networks dengan banyak lapisan untuk belajar dari data dalam jumlah besar yang ditunjukkan pada Gambar 2. Disebut "deep" karena melibatkan banyak lapisan jaringan neural yang membantu sistem memahami dan menginterpretasi data. Teknologi ini memungkinkan komputer memproses informasi dengan cara yang mirip dengan otak manusia, di mana sistem meningkatkan keterampilan dan akurasinya dari waktu ke waktu dengan menganalisis sejumlah besar data, tanpa memerlukan pembaruan manual dari manusia.



Gambar 2. Fully Connected Deep Neural Network (https://www.geeksforgeeks.org/deep-learning/introduction-deep-learning/)

Deep Learning telah merevolusi banyak bidang penelitian dengan kemampuannya untuk mempelajari model yang lebih baik dari volume data yang sangat besar (Sarker, 2021). Teknologi ini bergantung pada generasi baru Artificial Neural Network (ANN) yang disebut Deep Neural Network (DNN). Bagian ini memberikan gambaran singkat tentang teknik Deep Learning sebelum membahas bagian berikutnya. Premisnya adalah bahwa kinerja DNN umumnya lebih unggul daripada ANN klasik, namun dengan biaya waktu pelatihan yang lebih lama. Namun, waktu pelatihan ini dapat dikurangi dengan menggunakan perangkat keras canggih dan teknik khusus (Jmour et al., 2018). Desain DNN sangat krusial untuk kesuksesan. Beberapa arsitektur Deep Learning yang populer, antara lain.

C. Reinforcement Learning

Reinforcement Learning (RL) adalah subbidang dari Machine Learning di mana agen berinteraksi dengan lingkungan untuk mencapai tujuan, dan pembelajaran terjadi secara bertahap melalui interaksi. Bagian ini memperkenalkan beberapa mekanisme dasar dan terminologi. Dalam RL, Agen adalah entitas yang berinteraksi dengan lingkungan tertentu dengan melakukan tindakan, dan menerima umpan balik dari lingkungan setelah setiap tindakan yang dipilih, seperti yang dijelaskan dalam Gambar 3.

Policy adalah strategi yang menunjukkan kepada agen tindakan mana yang harus dipilih dalam setiap keadaan lingkungan. Agen harus belajar kebijakan optimal, yaitu kebijakan yang memaksimalkan imbalan kumulatif dalam jangka panjang. Masalah RL didefinisikan sebagai MDP. MDP adalah tuple (*S, A, Pa, Ra, γ*), di mana *S* adalah himpunan keadaan; *A* adalah himpunan tindakan; $P_a = P_r(s_{t+1} = s' | s_t = s, a_t = a)$ adalah probabilitas transisi (yaitu probabilitas mencapai *s′* pada waktu *t* + *1*, setelah memilih *a* di *s* pada waktu *t*), *Ra*(*s,s′*) adalah imbalan yang diharapkan atau imbalan langsung yang diperoleh saat beralih dari keadaan *s* ke keadaan *s′* setelah mengambil tindakan a, masing-masing; dan *γ* adalah faktor diskon.

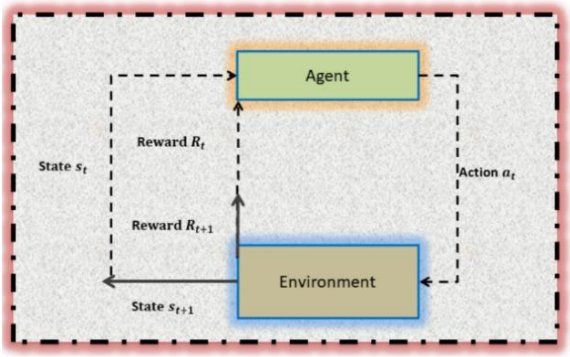
Skema RL umumnya dikategorikan menjadi dua jenis utama: algoritma tanpa model dan algoritma berbasis model. Algoritma RL berbasis model memerlukan deskripsi yang tepat tentang dinamika lingkungan dalam bentuk distribusi probabilitas transisi keadaan. Metode ini menghitung kebijakan optimal dengan memecahkan sistem persamaan. Sementara itu, algoritma tanpa model digunakan ketika tidak ada deskripsi model yang tepat atau solusinya terlalu rumit. Kelas algoritma ini berinteraksi langsung dengan menggunakan skema Trial & Error untuk belajar kebijakan optimal. Dalam RL terbalik (Nair et al., 2015), kita meneliti tujuan, nilai, atau imbalan agen dengan memanfaatkan wawasan dari perilakunya.

Pada RL sendiri terdapat agent-agent yang biasa digunakan untuk training data yang ingin dilatih, seperti SAC, Twin Delayed DDPG, dan lainnya.

Konsep dasar Reinforcement Learning (RL) berpusat pada siklus interaksi berulang antara entitas pembuat keputusan (agent) dan lingkungan eksternal (environment) dalam kerangka kerja Markov Decision Process (MDP). Berbeda dengan supervised learning, agent dalam RL tidak menerima pasangan masukan-keluaran yang benar secara langsung, melainkan belajar mengoptimalkan perilakunya melalui mekanisme trial-and-error berdasarkan umpan balik dinamis dari lingkungan.

Pada setiap langkah waktu (timestep) t, agent mengamati keadaan lingkungan saat ini yang direpresentasikan sebagai state (s_t). Berdasarkan pengamatan tersebut dan strategi atau kebijakan (policy) yang dimiliki, agent mengeksekusi suatu tindakan atau action (a_t). Lingkungan kemudian merespons action a_t dengan mengubah kondisi internalnya menuju state baru (s_{t+1}) serta memberikan umpan balik numerik dalam bentuk reward (R_{t+1}). Reward ini berperan sebagai sinyal evaluatif, di mana nilai positif menandakan tindakan yang menguntungkan dan nilai negatif berfungsi sebagai penalti atas keputusan yang kurang tepat.

Informasi berupa state baru (s_{t+1}) dan reward (R_{t+1}) kemudian diterima kembali oleh agent untuk memperbarui nilai estimasi (value function) atau menyesuaikan parametrik kebijakan yang digunakan. Siklus interaksi yang melibatkan pertukaran komponen $s_t, a_t,$ dan R_t ini berlangsung secara berkelanjutan hingga membentuk trajektori pengalaman. Tujuan akhir dari proses pembelajaran ini adalah untuk menemukan kebijakan optimal (π^*) yang dapat memaksimalkan akumulasi reward jangka panjang (expected cumulative reward) yang diperoleh agent.



Gambar 3. Algoritma Reinforcement Learning
Sumber : https://www.mdpi.com/1424-8220/22/1/309

D. Soft Actor-Critic (SAC)

SAC adalah algoritma deep reinforcement learning yang canggih, off-policy, dan model-free yang telah menetapkan standar baru untuk menyelesaikan tugas kontrol berkelanjutan yang kompleks. SAC menonjol dengan mengintegrasikan maximum entropy reinforcement learning ke dalam framework actor-critic, yang secara fundamental mengubah bagaimana agen mendekati trade-off eksplorasi-eksploitasi.

Berbeda dengan metode RL tradisional yang hanya berfokus pada memaksimalkan expected reward, SAC mendorong agen untuk memaksimalkan baik expected reward maupun entropy dari kebijakannya. Ini berarti agent diinsentif untuk berhasil dalam tugas sambil juga bertindak seacak mungkin. Pendekatan ini memberikan algoritma deep reinforcement learning kapasitas untuk mempertimbangkan dan mempelajari banyak jalur alternatif yang mengarah ke tujuan optimal, dan kapasitas untuk belajar bagaimana bertindak secara optimal meskipun dalam keadaan yang merugikan.

Meskipun kontrol kontinu diimplementasikan oleh algoritma DDPG, beberapa kelemahan bawaan, seperti sensitivitas hiperparameter, pelatihan yang rapuh, dan efisiensi sampling yang rendah, masih perlu diatasi. Oleh karena itu, algoritma SAC yang didasarkan pada entropi maksimum dan kerangka kerja AC diajukan, yang mencapai pelatihan yang lebih stabil, kecepatan konvergensi yang lebih cepat, dan penyesuaian hiperparameter yang lebih mudah dibandingkan dengan algoritma DDPG. Algoritma RL dan DRL sebelumnya biasanya hanya mengejar reward kumulatif maksimum. Algoritma SAC mengejar entropi maksimum tambahan untuk mendorong eksplorasi yang lebih banyak dan mendapatkan kinerja pelatihan yang lebih stabil, dan fungsi kebijakan π^* diformulasikan sebagai berikut:

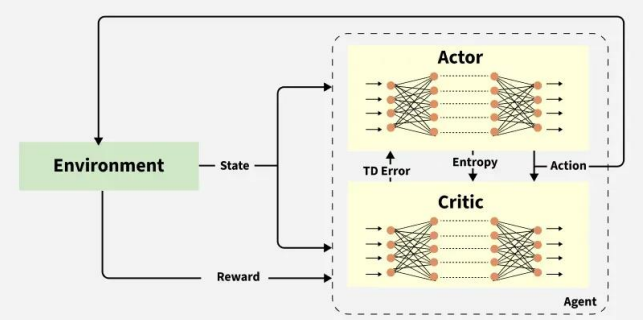
$$\pi^* = argmax \sum_t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[R_t(s_t, a_t) + a.\mathcal{H}(\pi(\cdot\,|s_t))] \quad (1)$$

di mana *Rt(st, at)* mewakili imbalan instan di bawah *st* dan *at*, $\hbar$ *(π(·|st))* mewakili entropi kebijakan saat ini *π*, *α* mewakili koefisien suhu, yang disesuaikan secara otomatis dengan meminimumkan *J(α)* selama proses pelatihan, sebagai berikut:

$$J(\alpha) = \mathbb{E}_{a_t \sim \pi_t} [-\alpha \cdot ln(\pi_t(a_t|s_t) - \alpha \cdot \hbar_0)]$$

$$a \leftarrow a - \lambda \cdot \hat{\nabla} \alpha J(\alpha) \quad (2)$$

Kritik akan mengarahkan aktor untuk menyesuaikan tindakan keluaran melalui residu Bellman lunak ω(t), parameter jaringan aktor ϕ diperbarui dengan persamaan, dan parameter jaringan kritik θi diperbarui dengan persamaan. Dengan mekanisme pengejaran entropi maksimum, penambahan kebisingan acak pada tindakan keluaran yang diterapkan dalam algoritma DDPG tidak diperlukan dalam algoritma SAC. Pengulangan pengalaman juga digunakan untuk meningkatkan efisiensi pembelajaran. SAC Framework ditunjukkan pada Gambar 4.



Gambar 4. SAC Framework
Sumber : https://www.geeksforgeeks.org/deep-learning/soft-actor-critic-reinforcement-learning-algorithm/

Algoritma Soft Actor-Critic (SAC) merupakan salah satu pendekatan off-policy actor-critic dalam Deep Reinforcement Learning yang mengintegrasikan kerangka kerja pemaksimalan entropi (maximum entropy framework). Dalam arsitektur ini, komponen agent terbagi menjadi dua jaringan saraf tiruan (neural network) utama, yaitu Actor dan Critic, yang bekerja secara komplementer saat berinteraksi dengan lingkungan (environment). Actor berfungsi sebagai penentu kebijakan (policy network) yang menghasilkan tindakan (action), sedangkan Critic berperan sebagai penilai (value network) yang mengestimasi kualitas dari tindakan yang diambil tersebut.

Proses interaksi dimulai ketika environment mengirimkan umpan balik berupa keadaan sistem (state) dan imbalan (reward) kepada agent. Informasi state diterima secara simultan oleh Actor dan Critic. Actor memproses state tersebut untuk mengeksekusi action ke environment, sekaligus mentransfer parameter action beserta nilai entropy kebijakan ke jaringan Critic. Komponen entropy ini menjadi faktor diferensiasi utama pada SAC, di mana agent didorong untuk mempertahankan keacakan tindakan dalam rangka mengeksplorasi ruang pencarian secara optimal dan mencegah terjadinya konvergensi dini (premature convergence).

Selanjutnya, jaringan Critic mengevaluasi kombinasi state, action, reward, dan entropy untuk menghitung nilai galat selisih waktu (Temporal Difference Error atau TD Error). Sinyal TD Error ini digunakan untuk memperbarui estimasi nilai pada jaringan Critic, sekaligus dikirimkan kembali ke Actor sebagai petunjuk pembaruan parameter kebijakannya. Melalui siklus tertutup ini, algoritma SAC secara simultan meminimalkan TD Error dan memaksimalkan ekspektasi imbalan kumulatif yang dibobot oleh tingkat policy entropy.

METODE

Pendekatan penelitian ini menggunakan metode eksperimen simulatif dengan penerapan kontrol MIMO berbasis DRL untuk mengoptimalkan konsumsi energi pada kendaraan listrik. Sistem kontrol yang digunakan dirancang agar mampu mengatur beberapa variabel masukan, seperti torsi motor depan dan belakang, serta proporsi pengereman regeneratif terhadap beberapa variabel keluaran, seperti kecepatan kendaraan, stabilitas lateral, dan efisiensi energi, sehingga sistem dapat beroperasi secara optimal dalam berbagai kondisi berkendara.

Tahapan penelitian dimulai dengan pembuatan model kendaraan listrik menggunakan Simulink Electrical yang mencakup dinamika motor listrik, sistem baterai, dan beban kendaraan. Model ini dikembangkan sebagai lingkungan simulasi untuk pelatihan algoritma DRL. Selanjutnya dilakukan perancangan struktur kontrol MIMO, di mana algoritma DRL akan belajar menentukan aksi optimal secara kontinu berdasarkan keadaan sistem. Dalam formulasi DRL, keadaan sistem mencakup kecepatan kendaraan, target kecepatan, SoC baterai, dan torsi motor saat ini. Action berupa pengaturan torsi distribusi antar motor dan tingkat pengereman regeneratif.

E. Teknik Pengumpulan Data

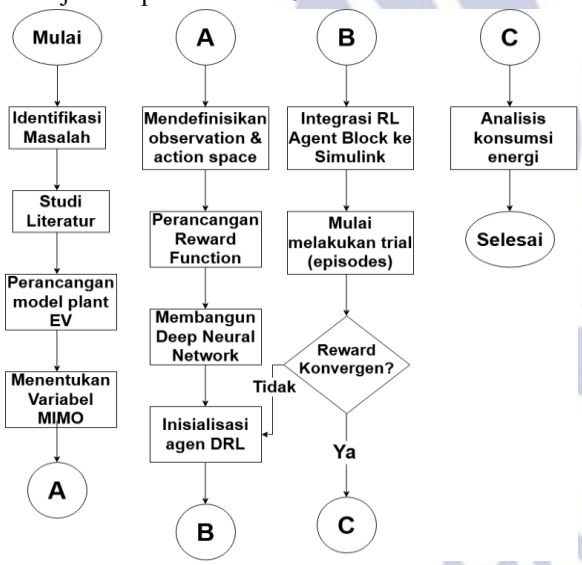
Teknik pengumpulan data pada penelitian ini dilakukan secara simulatif menggunakan model kendaraan listrik berbasis MATLAB/Simulink yang mencakup dinamika motor, baterai, dan sistem penggerak. Data dikumpulkan dari hasil simulasi pengujian kontrol MIMO berbasis DRL pada berbagai skenario berkendara seperti rute datar dan menanjak. Data yang direkam meliputi kecepatan kendaraan, torsi motor, SoC baterai, konsumsi energi, serta aksi kontrol yang dihasilkan oleh algoritma DRL.

Seluruh data dikumpulkan melalui serangkaian episode simulasi dengan variasi parameter sistem menggunakan metode domain randomization guna meningkatkan kemampuan generalisasi model. Data hasil simulasi kemudian melalui tahap pra-pemrosesan berupa normalisasi dan pemeriksaan konsistensi sebelum digunakan untuk pelatihan dan evaluasi algoritma. Dengan

pendekatan ini, penelitian dapat memperoleh data yang terkontrol, akurat, dan representatif terhadap kondisi operasional kendaraan listrik tanpa harus melakukan pengujian langsung di lapangan.

F. Rancangan Penelitian

Rancangan penelitian ini dimulai dengan studi literatur mengenai sistem kontrol MIMO, algoritma DRL, dan optimasi energi kendaraan listrik, kemudian dilanjutkan dengan perancangan model kendaraan listrik di MATLAB/Simulink yang mencakup motor, baterai, dan sistem penggerak. Setelah model terbentuk, dikembangkan struktur kontrol MIMO berbasis DRL dengan penentuan state, action, dan reward yang sesuai tujuan optimasi energi, lalu diintegrasikan ke lingkungan simulasi untuk proses pelatihan hingga mencapai kestabilan kebijakan. Sistem yang telah dilatih diuji pada berbagai skenario berkendara seperti rute datar dan menanjak, kemudian dianalisis untuk menilai efektivitas optimasi energi. Penelitian ditutup dengan penyusunan kesimpulan dan rekomendasi terkait pengembangan sistem kontrol cerdas yang efisien dan adaptif bagi kendaraan listrik di masa depan. Flowchart penelitian ditunjukkan pada Gambar 5 berikut ini.



Gambar 5. Flowchart Penelitian

Gambar 5 dijelaskan berikut ini.

1. Inisialisasi: Langkah Awal (t = 0)

Pada awal proses, Agent berada dalam kondisi "tidak tahu apa-apa".

a. Actor (Neural Network 1): Diinisialisasi dengan bobot acak. Fungsinya adalah sebagai pembuat kebijakan (*policy*), yang menentukan aksi apa yang harus diambil.
b. Critic (Neural Network 2): Juga diinisialisasi secara acak. Fungsinya adalah sebagai pengritik (*value function*), yang memprediksi berapa besar total imbalan yang akan didapat di masa depan.
c. Environment: Menyiapkan kondisi awal atau State (*St*). Dalam konteks EV, ini bisa berupa *State of Charge* (SOC) baterai dan kecepatan kendaraan saat ini.

2. Siklus Interaksi (The Loop)

Proses berjalan dalam sebuah siklus yang terus berulang hingga ribuan kali (*steps*):

a. Observasi State: Agent menerima input State dari Environment.
b. Pemilihan Action (Actor): Berdasarkan *St*, Actor mengusulkan sebuah Action (*At*). Karena ini masih tahap awal, Actor seringkali melakukan eksplorasi (mencoba aksi acak) untuk memetakan kemungkinan.
c. Eksekusi di Environment: Action *At* diterapkan ke Environment.
d. Feedback (Reward & New State): Environment memberikan respons balik berupa:
   1. Reward (*Rt+1*): Nilai numerik yang menunjukkan seberapa "baik" aksi tersebut (misal: positif jika efisiensi naik, negatif jika suhu motor terlalu panas).
   2. New State (*St+1*): Kondisi terbaru setelah aksi dilakukan.

3. Mekanisme Evaluasi (Peran Critic)

Di sinilah letak kecerdasan DRL. Critic akan melihat transisi tersebut dan menghitung Temporal Difference (TD) Error. Critic membandingkan:

a. Prediksi awal tentang seberapa baik state tersebut.
b. Kenyataan yang terjadi (Reward yang diterima + Prediksi nilai di state berikutnya).

Secara matematis, ini didasarkan pada Persamaan Bellman:

$$V(s) \leftarrow V(s) + \alpha[R + \gamma V(s') - V(s)] \quad (3)$$

Di mana $[R + \gamma V(s') - V(s)]$ adalah Advantage (keuntungan). Jika nilainya positif, berarti aksi tersebut lebih baik dari perkiraan.

4. Proses Update: Belajar dari Kesalahan

Data dari interaksi tadi (*S, A, R, S'*) disimpan dalam *Experience Replay*. Kemudian, kedua jaringan saraf diperbarui secara simultan:

a. Update Actor: Bobot jaringan digeser ke arah aksi yang memberikan *Advantage* positif. Tujuannya adalah memperkuat kebijakan yang sukses.
b. Update Critic: Bobot jaringan diperbarui agar prediksinya di masa depan lebih akurat (meminimalkan *Loss Function* antara prediksi dan kenyataan).

5. Menuju Konvergensi

Seiring bertambahnya iterasi (*episodes*), terjadi perubahan perilaku pada Agent:

a. Stabilitas Policy: Actor mulai konsisten memilih aksi yang optimal dan tidak lagi melakukan eksplorasi acak yang drastis.
b. Akurasi Prediksi: Critic menjadi sangat akurat dalam memprediksi imbalan masa depan, sehingga *Loss Function* mendekati nol atau stabil.
c. Titik Konvergen: Konvergensi tercapai ketika grafik *Cumulative Reward* mendatar (mencapai nilai maksimal yang stabil). Pada titik ini, Agent telah menemukan strategi terbaik (Optimal Policy) untuk mengontrol sistem, meskipun kondisi lingkungan sedikit berubah-ubah.

G. Prosedur Penelitian

Berdasarkan rancangan penelitian yang telah dibuat, berikut adalah penjelasan secara rinci langkah-langkah yang dilakukan pada penelitian ini.

1) Identifikasi Masalah

Tahap identifikasi masalah bertujuan untuk menemukan permasalahan utama yang akan diselesaikan melalui penelitian ini. Dalam konteks kendaraan listrik, efisiensi energi menjadi aspek yang sangat penting karena secara langsung memengaruhi jarak tempuh dan umur baterai. Namun, pengaturan energi dan distribusi torsi pada kendaraan listrik yang memiliki beberapa aktuator penggerak (multi-input multi-output) masih menjadi tantangan karena sifat sistemnya yang nonlinier dan dinamis. Ketidakefisienan dalam pengendalian torsi dan penggunaan daya dapat menyebabkan konsumsi energi berlebih serta menurunkan performa kendaraan. Oleh karena itu, diperlukan sistem kontrol yang mampu melakukan optimasi energi secara adaptif dan real-time. Berdasarkan permasalahan tersebut, penelitian ini berfokus pada penerapan kontrol MIMO berbasis Deep Reinforcement Learning untuk mengoptimalkan penggunaan energi kendaraan listrik melalui pembelajaran berbasis interaksi dengan lingkungan simulasi.

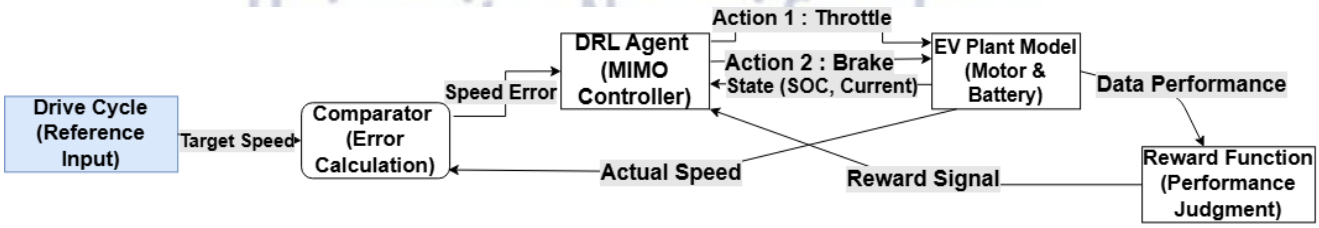
2) Studi Literatur

Setelah permasalahan teridentifikasi, dilakukan studi literatur untuk memperoleh landasan teoritis dan konseptual yang mendukung penelitian. Studi ini mencakup kajian terhadap konsep dasar kendaraan listrik, karakteristik sistem kontrol MIMO, serta prinsip kerja algoritma Deep Reinforcement Learning (DRL) yang digunakan untuk pengendalian sistem dinamis. Literatur yang dikaji berasal dari jurnal ilmiah, prosiding, buku teks, serta publikasi penelitian terkini yang relevan. Melalui studi literatur ini, peneliti dapat memahami parameter-parameter penting yang memengaruhi efisiensi energi kendaraan listrik, menentukan struktur model yang akan digunakan dalam simulasi, serta merancang konfigurasi variabel *state*, *action*, dan *reward function* yang sesuai dengan tujuan penelitian.

Tahapan berikutnya meliputi perancangan model sistem kendaraan listrik, penerapan algoritma kontrol MIMO berbasis DRL, pelatihan dan simulasi model hingga mencapai konvergensi, serta pengujian kinerja untuk menilai efektivitas sistem dalam mengoptimalkan energi kendaraan listrik. Hasil dari setiap tahap akan dianalisis dan disimpulkan untuk memberikan kontribusi terhadap pengembangan sistem kontrol cerdas yang efisien dan adaptif.

3) Perancangan Model Sistem Kendaraan Listrik

Pada tahap ini dilakukan perancangan model sistem kendaraan listrik yang akan digunakan sebagai dasar simulasi dan pengujian kontrol MIMO berbasis DRL. Model kendaraan listrik dikembangkan menggunakan perangkat lunak MATLAB/Simulink karena kemampuannya dalam memodelkan sistem dinamis secara akurat. Model ini mencakup beberapa komponen utama seperti motor listrik, baterai, serta sistem penggerak roda. Setiap komponen dimodelkan menggunakan persamaan fisis yang menggambarkan hubungan antara torsi, arus, tegangan, dan kecepatan kendaraan. Rancangan Sistem ditujukkan pada Gambar 6 berikut ini.



Gambar 6. Rancangan Sistem Kendaraan Listrik

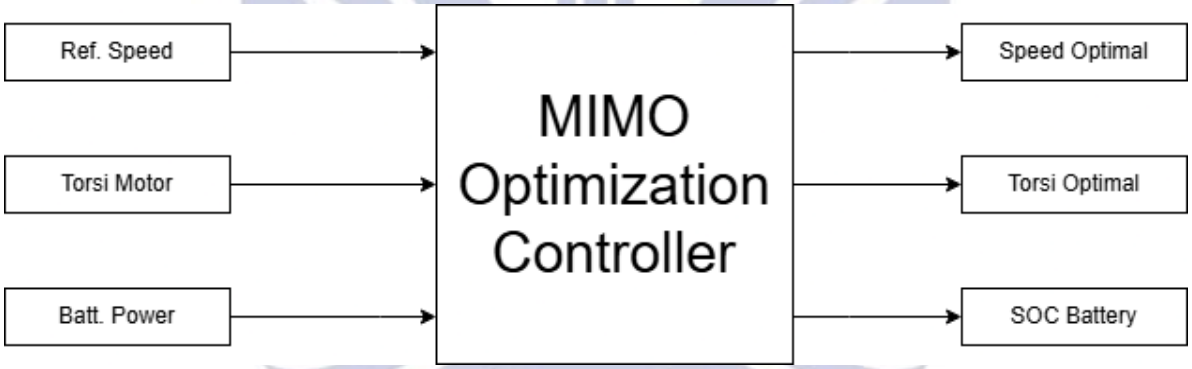
Diagram ini menggambarkan sistem kontrol closed-loop di mana DRL Agent berfungsi sebagai otak pengendali cerdas untuk kendaraan listrik. Proses dimulai dari Drive Cycle yang memberikan target kecepatan referensi. Target ini dibandingkan dengan kecepatan aktual kendaraan oleh Comparator untuk menghasilkan nilai error. Agen DRL menerima input berupa error kecepatan dan status kendaraan, lalu mengolahnya menggunakan DNN berupa dua aksi simultan: sinyal gas dan sinyal rem yang dikirimkan ke model fisik kendaraan untuk dieksekusi.

Setelah kendaraan bereaksi terhadap perintah agen, Reward Function akan mengevaluasi performa tersebut berdasarkan efisiensi energi dan ketepatan kecepatan. Fungsi ini mengirimkan sinyal reward kembali ke agen: nilai tinggi jika hemat energi dan stabil, atau nilai rendah jika boros. Bersamaan dengan itu, kecepatan aktual kendaraan dikirim kembali ke Comparator sebagai umpan balik. Siklus ini berulang terus-menerus selama simulasi, memungkinkan agen DRL untuk "belajar" dari kesalahannya dan secara bertahap menemukan strategi kontrol paling optimal untuk meminimalkan konsumsi energi tanpa mengorbankan performa berkendara.

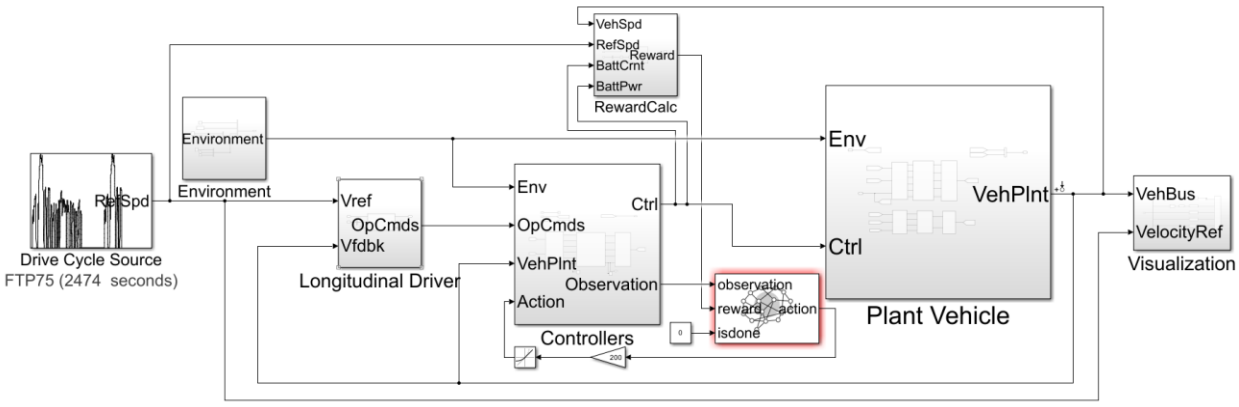
4) Perancangan Struktur Kontrol MIMO

Selanjutnya adalah merancang struktur dari Kontrol MIMO yang ditunjukkan pada Gambar 7. Dalam perancangan arsitektur kontrol MIMO ini, ditetapkan tiga variabel input utama sebagai Observation Space guna memberikan agen DRL pemahaman komprehensif terhadap kondisi internal dan eksternal kendaraan. Variabel Vehicle Speed digunakan sebagai referensi dinamika untuk menjaga performa traksi agar sesuai dengan target lintasan, sementara SOC berfungsi sebagai indikator batasan energi yang tersedia. Signifikansi penambahan variabel Battery Temperature ke dalam vektor input bertujuan untuk mengintegrasikan manajemen termal ke dalam kebijakan kontrol, sehingga agen dapat memitigasi risiko degradasi performa akibat panas berlebih (thermal derating) secara preventif melalui keputusan yang diambil.

Pada sisi aktuasi, strategi kontrol didistribusikan ke dalam tiga sinyal output simultan untuk mencapai optimasi efisiensi energi secara holistik. Selain meregulasi Torsi Motor untuk kebutuhan propulsi standar, sistem dirancang untuk memodulasi Daya Motor Optimal sebagai pembatas dinamis, yang berfungsi memangkas konsumsi arus berlebih saat kondisi baterai kritis atau mengalami penurunan tegangan. Lebih lanjut, integrasi variabel Pendingin Optimal memungkinkan manajemen beban aksesoris secara adaptif, di mana konsumsi energi untuk sistem pendingin hanya diaktifkan berdasarkan kebutuhan termal aktual, meminimalisir kerugian energi parasitik.



Gambar 7. Variabel Kontrol MIMO



Gambar 8. EV Plant System

HASIL DAN PEMBAHASAN

Hasil rancangan Model Sistem EV ditunjukkan pada Gambar 8 Sistem kendaraan listrik dalam penelitian ini dimodelkan dan disimulasikan dengan menggunakan perangkat lunak MATLAB/Simulink. Model yang dikembangkan dengan menggunakan pendekatan sistem kontrol umpan balik untuk mensimulasikan dinamika longitudinal kendaraan. Secara umum, arsitektur top model dari model ini terdiri dari beberapa blok atau sub-

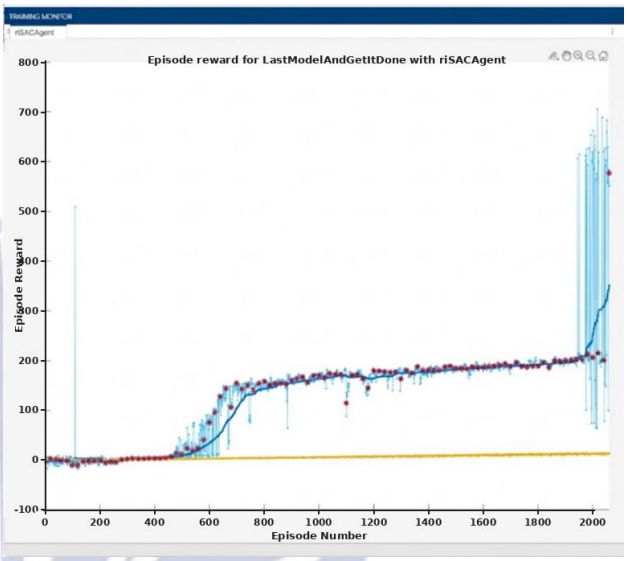
sistem utama yang terintegrasi, yaitu: referensi kecepatan, model pengemudi, unit kontrol, model fisik kendaraan, dan pemantauan hasil.

Berikut adalah rincian fungsi dari masing-masing sub-sistem pada model tersebut:

1. Drive Cycle Source
   Blok ini berperan sebagai sumber utama sistem yang menyajikan profil kecepatan acuan atau sasaran seiring waktu. Dalam simulasi ini, diterapkan standar siklus berkendara untuk mencerminkan kondisi berkendara yang realistis yang harus dipatuhi oleh kendaraan.
2. Environment
   Sub-sistem ini mensimulasikan keadaan lingkungan luar yang akan berdampak pada kinerja dan dinamika kendaraan. Parameter yang umumnya dicantumkan dalam blok ini mencakup kecepatan angin, kemiringan jalan, suhu sekitar, dan tekanan atmosfer.
3. Longitudinal Driver
   Blok ini menggambarkan perilaku pengemudi manusia dalam bentuk virtual. Model ini menerima dua sinyal input: kecepatan referensi dan kecepatan nyata kendaraan. Berdasarkan perbedaan atau kesalahan antara kedua kecepatan itu, pengendali dalam blok ini akan memproduksi perintah untuk akselerasi atau deselerasi guna mencapai kecepatan yang ditargetkan.
4. Controllers and Energy Management
   Sub-sistem ini berfungsi sebagai "otak" dari EV atau Unit Kontrol Kendaraan. Ia menerima instruksi dari model pengemudi dan mengkonversinya menjadi sinyal kontrol tertentu untuk komponen powertrain. 1) Pengendali: mengelola logika distribusi torsi dan menjamin komponen berfungsi dalam batas yang aman. 2) Manajemen Energi: mengelola distribusi daya antara baterai dan motor listrik, termasuk mengontrol sistem pengereman regeneratif saat kendaraan berkurang kecepatan.
5. EV Plant (Model Fisik Kendaraan)
   Ini adalah blok utama yang menggambarkan dinamika fisik EV secara keseluruhan. Blok ini menerima sinyal torsi/energi dari unit kontrol dan menghitung reaksi fisik kendaraan. Pada sub-sistem ini, biasanya terdapat pemodelan komponen yang lebih mendalam, mencakup: 1) Sistem Baterai (ESS): menghitung penurunan State of Charge (SOC), arus, dan tegangan berdasarkan daya yang digunakan atau diisi. 2) Motor Listrik: mengubah energi listrik menjadi energi mekanis (torsi) atau sebaliknya. 3) Dinamika Bodi Kendaraan: Menentukan kecepatan sebenarnya kendaraan dengan memperhitungkan gaya resistansi.
6. Visualization
   Bagian kanan akhir (blok Scope) digunakan untuk memantau, memvisualisasikan, dan merekam data dari variabel-variabel penting saat simulasi berlangsung.

H. Training Agent

Proses optimalisasi unit kontrol pada kendaraan listrik (EV) dalam penelitian ini menggunakan pendekatan *Reinforcement Learning* dengan algoritma Soft Actor-Critic (SAC). Pelatihan dilakukan menggunakan *Reinforcement Learning Episode Manager* pada MATLAB.



Gambar 9. Simulasi Training Agent

Berdasarkan grafik Reinforcement Learning Episode Manager di atas, berikut adalah analisis mendalam mengenai performa model rlSACAgent untuk skenario optimasi energi EV (Electric Vehicle):

1. Analisis Tren Reward (Progress Belajar)
   - Fase Eksplorasi (Episode 0 - 500): Pada tahap awal, reward berada di sekitar angka nol. Ini menunjukkan agen sedang melakukan eksplorasi acak untuk memahami lingkungan (environment) EV, seperti hubungan antara injakan pedal gas dan konsumsi baterai.
   - Fase Pembelajaran (Episode 500 - 1900): Terjadi peningkatan *Average Reward* (garis biru tebal) yang sangat konsisten. Ini adalah indikator positif bahwa agen mulai menemukan kebijakan (policy) yang lebih efisien dalam mengelola energi. Strategi seperti *smooth acceleration* dan optimalisasi *SOC Usage* kemungkinan besar mulai terbentuk di sini.
   - Lonjakan Akhir (Episode 1900 - 2060): Terjadi lonjakan tajam pada reward (mencapai angka 600-800). Hal ini menandakan agen menemukan "jalur optimal" atau strategi berkendara yang sangat hemat energi namun tetap memenuhi kriteria perjalanan.
2. Evaluasi Stabilitas dan Konvergensi

- Variansi (Noise): Garis cyan (Episode Reward) masih menunjukkan fluktuasi yang cukup tinggi bahkan di akhir training. Dalam konteks EV, ini berarti model terkadang sangat efisien, namun di episode lain masih bisa melakukan kesalahan yang boros energi.
- Episode Q0 (Garis Kuning): Nilai estimasi awal (Q0) naik secara perlahan dan stabil. Ini menunjukkan bahwa *critic* dalam algoritma SAC mulai mampu memprediksi total reward masa depan dengan baik, yang sangat penting untuk stabilitas optimasi energi jangka panjang.
- Evaluation Statistic (Bintang Merah): Nilai evaluasi (577.39) berada di atas rata-rata akhir. Ini berarti saat fitur eksplorasi dimatikan, model mampu bekerja dengan sangat baik dan efisien.

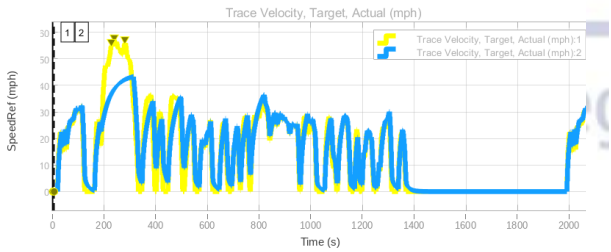
3. Interpretasi untuk Optimasi Energi EV

Dalam optimasi energi EV, grafik ini mencerminkan keberhasilan agen dalam menyeimbangkan beberapa faktor:

1. Minimasi Konsumsi Daya: Agen belajar bahwa akselerasi mendadak mengurangi reward, sehingga ia memilih pola daya yang lebih konstan.
2. Manajemen SoC (State of Charge): Model kemungkinan besar belajar untuk menjaga baterai tetap dalam rentang efisiensi optimal.
3. Constraint Waktu vs Energi: Jika reward mencakup penalti waktu, agen telah menemukan *sweet spot* antara sampai di tujuan tepat waktu dengan penggunaan energi minimal.

I. Evaluasi Nilai Akhir

Output Speed Ref ditunjukkan pada Gambar 9 berikut ini.

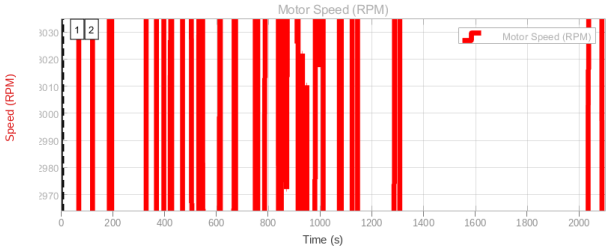


Gambar 10. Output speed Ref

Perhitungan error antar Target Speed dengan Actual Speed ditunjukkan pada Gambar 10 berikut ini.

```
--- ANALISIS STRUCTURE WITH TIME ---
Rentang Waktu : 0.00 ke 2100.00 detik
Max Target    : 56.70 mph
Max Actual    : 43.29 mph
-------------------------------------
MSE    : 22.1882
RMSE   : 4.7104
R2     : 0.9018
```

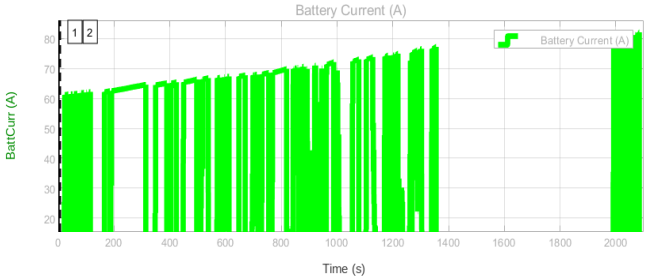
Gambar 11. Perhitungan error antar Target Speed dengan Actual Speed



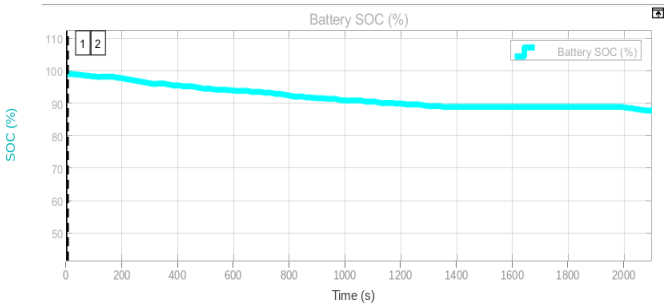
Gambar 12 menujukkan data kecepatan motor dalam satuan RPM.

Gambar 12. Data kecepatan motor dalam satuan RPM

Current Output dan Persentase Baterai ditunjukkan pada Gambar 12 dan Gambar 13 berikut ini.

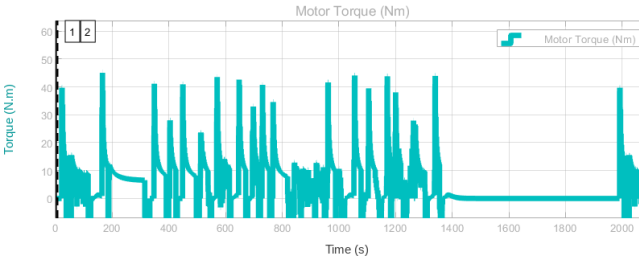


Gambar 13. Battery Current



Gambar 14. Prosentase battery output SOC

Output Torsi ditunjukkan pada Gambar 14 berikut ini.



Gambar 15. Output Torsi

Hasil simulasi model fisik kendaraan listrik diamati melalui beberapa parameter utama yang meliputi dinamika gerak longitudinal (kecepatan dan torsi) serta respons sistem penyimpanan energi (arus dan *State of Charge* baterai). Berdasarkan instrumen pengukuran, kinerja sistem dapat diuraikan berikut ini.

1. Analisis Akurasi Kontrol dan Kecepatan Respon

Berdasarkan hasil simulasi pada Gambar 14, sistem kontrol MIMO berbasis DRL menunjukkan performansi yang sangat tangguh dalam menjaga stabilitas kendaraan.

a. Presisi *Tracking*: Indikator utama keberhasilan sistem ini terlihat pada grafik *Trace Velocity*. Kecepatan aktual mampu menempel ketat pada kecepatan target hampir di seluruh siklus pengujian. Hal ini membuktikan bahwa agen DRL telah mencapai tingkat konvergensi yang optimal, di mana kebijakan (*policy*) yang dipelajari mampu meminimalkan *error* secara instan bahkan pada perubahan gradien kecepatan yang tajam.

b. Akselerasi yang Responsif: Analisis terhadap *Motor Torque* menunjukkan bahwa sistem memberikan respon torsi yang tegas dan tepat sasaran saat terjadi permintaan akselerasi. Kenaikan nilai torsi yang terlihat pada grafik merupakan representasi dari karakteristik motor listrik yang mampu memberikan torsi maksimal sejak putaran awal (0 RPM). Tidak adanya *overshoot* yang berlebihan pada grafik torsi menandakan bahwa kontroler MIMO berhasil mengatur distribusi daya tanpa menyebabkan ketidakstabilan mekanis.

2. Stabilitas Energi dan Efisiensi Baterai

Aspek krusial dari optimasi energi dalam skripsi ini adalah bagaimana sistem mengelola SOC selama operasi.

a. Linearitas Penurunan SOC: Grafik SOC menunjukkan penurunan yang sangat halus dan terukur dari 100% ke 88%. Tidak adanya lonjakan penurunan energi secara mendadak menunjukkan bahwa strategi manajemen energi berbasis DRL berhasil menghindari pembuangan energi yang sia-sia pada sistem *powertrain*.

b. Korelasi Arus dan Beban: Arus baterai bergerak secara proporsional terhadap beban torsi. Kerapatan sinyal pada *Battery Current* menunjukkan bahwa agen DRL bekerja secara *real-time* dengan frekuensi tinggi untuk menyesuaikan input daya terhadap kebutuhan dinamika kendaraan, yang merupakan keunggulan utama dari skema kontrol MIMO cerdas dibandingkan metode konvensional.

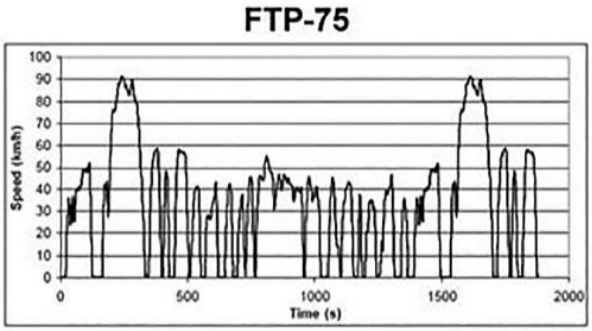
3. Keunggulan Kontrol MIMO Berbasis DRL

Secara keseluruhan, output simulasi ini memalidasi bahwa arsitektur sistem yang dirancang telah memenuhi kriteria sistem kontrol yang ideal untuk kendaraan Listrik, yaitu 1) Robust: Mampu menangani *Drive Cycle* yang kompleks selama 2100 detik tanpa kegagalan sistem. 2) Adaptive: Cepat menyesuaikan diri terhadap transisi dari kondisi *cruising*, akselerasi, hingga *idling*. 3) Efficient: Menjaga profil penggunaan baterai tetap optimal, yang secara langsung berkontribusi pada peningkatan jarak tempuh kendaraan per pengisian daya.

J. Perbandingan PI dengan DRL

Berikut adalah perbandingan antara metode kontrol PI dengan DRL dengan menggunakan Drive Cycle yang sama yakni FTP-75 dengan durasi waktu ~2000 detik.

Dan model kendaraan yang dipakai adalah **BMW i3 60 Ah** yang mana setara dengan **2632 Wh** karena **.**



Gambar 16. Drive Cycle FTP-75

Gambar 17. Kapasitas konsumsi Kontrol PI

Pada gambar 17 terlihat konsumsi daya baterai menggunakan kontrol PI menyentuh angka 114.58 Wh/km dari total daya yang digunakan yakni 2632 Wh. Dan jika dibandingkan dengan Kontrol Cerdas DRL yang sudah tertera pada gambar 14, dimana energi yang terpakai pada kendaraan hanya 72,1 Wh/km.

Yang mana peningkatan efisiensi energi baterai/SOC menggunakan SAC ini bisa dihitung sebagai berikut:

$$\text{Konsumsi PI per menit} = \frac{7{,}91\%}{20} = 0{,}396\%\ \text{SOC/menit}$$

$$\text{Konsumsi SAC per menit} = \frac{11{,}5\%}{40} = 0{,}288\%\ \text{SOC/menit}$$

$$\boxed{\text{Peningkatan efisiensi SAC} = \frac{0{,}396 - 0{,}288}{0{,}396} \times 100\% \approx \mathbf{27{,}3\%}}$$

Gambar 18. Peningkatan Efisiensi Energi

Dimana konsumsi energi dibagi dengan waktu berkendara akan menghasilkan konsumsi energi per waktu tersebut, yang mana dalam kasus perbandingan ini konsumsi PI membutuhkan 0,396% SOC/menit untuk menempuh waktu berkendara 40 menit sedangkan DRL dengan menggunakan agent SAC hanya membutuhkan 0,288% SOC /menit.

Jadi, peningkatan optimasi dan efisiensi menggunakan kontrol DRL dengan agent SAC adalah 27,3% dibandingkan metode kontrol konvensional.

Ucapan Terima Kasih

Penulis menyampaikan apresiasi dan terima kasih yang setinggi-tingginya kepada Prof. Dr. Puput Wanarti Rusimamto S. T., M.T. atas bimbingan, arahan substansial, serta masukan kritis yang berharga sepanjang penyusunan penelitian ini. Ketegasan dan ketelitian beliau dalam mengawal aspek metodologis dan teoritis menjadi kunci utama dalam menjaga integritas ilmiah serta terselesaikannya artikel ini dengan baik.

Kemudian, penulis juga ingin mengucapkan banyak terima kasih kepada pihak yang terlibat dalam pengerjaan penelitian ini terutama pada rekan-rekan Lab. Elektronika dan Lab. Sistem Kendali, atas dukungan semangat dari mereka penulis bisa menyelesaikan penelitian ini hingga tuntas.

PENUTUP

Simpulan

Berdasarkan hasil penelitian dan simulasi yang telah dilakukan mengenai optimasi energi kendaraan listrik menggunakan kontrol MIMO) berbasis DRL, disimpulkan bahwa 1) Perancangan sistem optimasi energi dilakukan dengan mengintegrasikan model dinamika kendaraan listrik pada lingkungan MATLAB/Simulink dengan algoritma DRL, seperti SAC. 2) Kontrol MIMO dirancang dengan memetakan variabel observasi kompleks seperti SOC, temperatur baterai, dan permintaan torsi, menjadi aksi kontrol yang terkoordinasi seperti distribusi torsi motor dan manajemen termal. 3) Keberhasilan rancangan ini sangat bergantung pada definisi reward function yang mampu menyeimbangkan antara efisiensi konsumsi energi dengan performa berkendara, sehingga agen dapat mempelajari kebijakan optimal dalam menghadapi kondisi jalan yang dinamis.

Saran

Meskipun sistem kontrol yang diusulkan telah memenuhi tujuan penelitian, terdapat beberapa keterbatasan yang menjadi ruang pengembangan bagi penelitian selanjutnya, khususnya dari aspek pemodelan dan keandalan sistem. Pertama, penelitian ke depan disarankan untuk menyempurnakan model degradasi sel baterai Lithium-Ion yang masih menggunakan pendekatan makro dan asumsi keseimbangan termal ideal dengan mengintegrasikan model kapasitas *State of Health* (SOH) berbasis karakteristik kimia sel spesifik (seperti LFP atau NMC) serta *thermal runaway model* guna memberikan gambaran penuaan baterai yang lebih empiris terhadap siklus *charge/discharge* jangka panjang. Kedua, untuk meningkatkan ketahanan (*robustness*) model *Reinforcement Learning* yang saat ini terbatas pada siklus tunggal FTP-75, disarankan penerapan metode *randomized training episode* yang menggabungkan variasi siklus berkendara seperti WLTP, HWFET, maupun profil lalu lintas perkotaan acak beserta variasi kemiringan jalan (*road grade elevation*) agar agen tidak mengalami *overfitting*.

Dari aspek validasi eksperimental, mengingat penelitian ini masih sepenuhnya berada pada ranah *Software-in-the-Loop* (SIL), penelitian lanjutan sangat direkomendasikan untuk melangkah ke tahap *Hardware-in-the-Loop* (HIL). Algoritma *Soft Actor-Critic* (SAC) yang telah dilatih perlu diekspor dan ditanamkan ke dalam mikrokontroler aktual atau *Vehicle Control Unit* (VCU) komersial. Pengujian HIL ini menjadi langkah krusial untuk memvalidasi apakah beban komputasi (*computational cost*) dari jaringan saraf tiruan (*neural network*) yang digunakan oleh agen SAC cukup efisien dan responsif untuk diproses secara *real-time* oleh perangkat keras kendaraan listrik yang sesungguhnya.

DAFTAR PUSTAKA

Huang, H., Li, B., Wang, Y., Zhang, Z., & He, H. (2024). Analysis of factors influencing energy consumption of electric vehicles: Statistical, predictive, and causal perspectives. *Applied Energy*, *375*, 124110. https://doi.org/https://doi.org/10.1016/j.apenergy.2024.124110

Jmour, N., Zayen, S., & Abdelkrim, A. (2018). Convolutional neural networks for image classification. *2018 International Conference on Advanced Systems and Electric Technologies*

(IC_ASET), 397–402. https://doi.org/10.1109/ASET.2018.8379889

Mei, P., Karimi, H. R., Xie, H., Chen, F., Huang, C., & Yang, S. (2023). A deep reinforcement learning approach to energy management control with connected information for hybrid electric vehicles. *Engineering Applications of Artificial Intelligence*, *123*, 106239. https://doi.org/10.1016/j.engappai.2023.106239

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Nair, A., Srinivasan, P., Blackwell, S., Alcicek, C., Fearon, R., De Maria, A., Panneershelvam, V., Suleyman, M., Beattie, C., Petersen, S., Legg, S., Mnih, V., Kavukcuoglu, K., & Silver, D. (2015). *Massively Parallel Methods for Deep Reinforcement Learning*. http://arxiv.org/abs/1507.04296

Padhilah, F. A., Surya, I. R. F., Adiatma, J. C., Sari, R. P., Permono, R. H., & Pradityo, R. (2025). *Indonesia Sustainable Mobility Outlook 2025: Driving Transport Decarbonization: Multi-pathways to Sustainable Mobility in Indonesia*. Institute for Essential Services Reform (IESR).

Parvin, K., Lipu, M. S. H., Hannan, M. A., Abdullah, M. A., Jern, K. P., Begum, R. A., Mansur, M., Muttaqi, K. M., Mahlia, T. M. I., & Dong, Z. Y. (2021). Intelligent Controllers and Optimization Algorithms for Building Energy Management Towards Achieving Sustainable Development: Challenges and Prospects. *IEEE Access*, *9*, 41577–41602. https://doi.org/10.1109/ACCESS.2021.3065087

Peng, Z., & He, Z. (2025). Optimization of Regenerative Braking Control Strategy for Dual-Motor Electric Vehicles Based on Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*, *26*(7), 10954–10967. https://doi.org/10.1109/TITS.2025.3553875

Sarker, I. H. (2021). Deep Learning : A Comprehensive Overview on Techniques , Taxonomy , Applications and Research Directions. *SN Computer Science*, *2*(6), 1–20. https://doi.org/10.1007/s42979-021-00815-1

Tbk., P. I. K. T. (2020). *Driving Integrated Car Terminal Ecosystem*.

Tbk., P. I. K. T. (2024). *Digitalization Through Optimizing Vehicle Ecosystem*.

Wang, D., He, H., & Wei, C. (2023). Cellular and potential molecular mechanisms underlying transovarial transmission of the obligate symbiont Sulcia in cicadas. *Environmental Microbiology*, *December*, 1–17. https://doi.org/10.1111/1462-2920.16310

Wang, P., Li, Y., Shekhar, S., & Northrop, W. F. (2019). A Deep Reinforcement Learning Framework for Energy Management of Extended Range Electric Delivery Vehicles. *IEEE Intelligent Vehicles Symposium (IV)*, *Iv*, 1627–1632.

Xu, J., Li, Z., Gao, L., Ma, J., Liu, Q., & Zhao, Y. (2022). A Comparative Study of Deep Reinforcement Learning-based Vehicles. *ArXiv:2202.11514v1*, *February*.

Zhang, H., & Li, S. (2021). Fuzzy Adaptive Control of Uncertain MIMO Chaotic Systems with Unknown Control Direction. *Complex.*, *2021*. https://doi.org/10.1155/2021/9914288