BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 2, February 2026 | Hal 627-632
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i2.984

Analisis Visual Perilaku Agen Q-Learning dan SARSA pada Cliff Walking Problem dengan Explainable Reinforcement Learning

Firas Atqiya[1,*], Muhammad Rizqi Sholahuddin[2]

[1] Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi Teknik Informatika, Universitas Padjadjaran, Sumedang, Indonesia
[2] Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Bandung, Bandung, Indonesia
Email: [1,*]firas.atqiya@unpad.ac.id, [2]muhammad.rizqi@polban.ac.id
Email Penulis Korespondensi: firas.atqiya@unpad.ac.id

Abstrak−*Reinforcement Learning* (RL) telah mencapai berbagai pencapaian luar biasa dalam tugas keputusan berurutan yang kompleks. Namun, model RL modern sering kehilangan sifat *explainability*, menciptakan masalah "kotak hitam" yang serius terutama pada domain berdampak tinggi. Penelitian ini mengusulkan arsitektur visualisasi real-time berbasis Pygame untuk RL, dan mendemonstrasikan manfaatnya dalam studi kasus *Cliff Walking* menggunakan algoritma *Q-Learning* dan SARSA. Kontribusi utama meliputi: (1) arsitektur visualisasi real-time yang memisahkan logika pelatihan dari rendering grafis dengan dukungan hingga ≥ 60 FPS, (2) teknik visualisasi interpretatif termasuk *heatmap diverging*, panah kebijakan dinamis, dan *Ghost Policies*, serta (3) studi empiris komprehensif yang menjelaskan perbedaan karakteristik kedua algoritma. Hasil eksperimen memperlihatkan dengan jelas bagaimana *Q-Learning* memilih jalur efisien namun berisiko sesuai sifat *off-policy* yang optimistis, sementara SARSA konvergen pada jalur lebih aman mencerminkan sifat *on-policy* yang memperhitungkan keamanan eksplorasi. Secara kuantitatif, Q-Learning berhasil mencapai jalur optimal sepanjang 13 langkah dengan akumulasi 10.642 kali jatuh , sedangkan SARSA mengonvergensi jalur aman 23 langkah dengan frekuensi tabrakan hambatan yang jauh lebih tinggi (232.844 kali) guna menghindari penalti ekstrem dari area jurang.

Kata Kunci: Explainable Reinforcement Learning; Q-Learning; SARSA; Visualisasi Real-Time; Cliff Walking

Abstract−Reinforcement Learning (RL) has achieved remarkable success in complex sequential decision tasks. However, modern RL models often lack explainability, creating a serious "black box" problem, especially in high-stakes domains. This study proposes a Pygame-based real-time visualization architecture for RL, and demonstrates its benefits in a Cliff Walking case study using Q-Learning and SARSA algorithms. Key contributions include: (1) a real-time visualization architecture that decouples training logic from graphics rendering with support more than 60 FPS, (2) interpretive visualization techniques including diverging heatmaps, dynamic policy arrows, and Ghost Policies, and (3) a comprehensive empirical study clarifying the distinct characteristics of both algorithms. Experimental results clearly show that Q-Learning selects an efficient but risky path aligned with its optimistic off-policy nature, while SARSA converges on a safer path reflecting its on-policy nature that considers exploration safety. Quantitatively, Q-Learning successfully achieved an optimal 13-step path with an accumulation of 10,642 falls, whereas SARSA converged to a safe 23-step path with a significantly higher collision frequency (232,844 times) to avoid extreme penalties from the cliff zone.

Keywords: Explainable Reinforcement Learning; Q-Learning; SARSA; Visualisasi Real-Time; Cliff Walking

1. PENDAHULUAN

Reinforcement Learning (RL) telah berkembang pesat dari sekadar konsep teoretis dalam kontrol optimal menjadi paradigma kecerdasan buatan yang mampu melampaui kemampuan manusia dalam tugas-tugas keputusan berurutan yang sangat kompleks, mulai dari penguasaan permainan strategi tingkat tinggi hingga kontrol sistem robotika presisi[1]. Kemampuan agen RL untuk belajar secara mandiri melalui interaksi trial-and-error menjadikannya salah satu pilar utama dalam pengembangan kecerdasan buatan modern. Keberhasilan fundamental ini sebagian besar didorong oleh integrasi teknik pembelajaran mendalam (*deep learning*) yang memungkinkan agen untuk mengekstraksi representasi fitur dari input sensorik berdimensi tinggi secara otomatis [2]. Namun, kemajuan luar biasa dalam performa prediktif dan kontrol ini membawa konsekuensi negatif pada aspek transparansi sistem. Seiring dengan meningkatnya kompleksitas model RL, khususnya yang berbasis *deep learning*, muncul tantangan fundamental terkait hilangnya sifat *explainability*, yang berarti bahwa proses penalaran internal agen dan dasar logis di balik pemilihan tindakan tertentu menjadi tidak mungkin untuk dipahami secara intuitif oleh pengamat manusia [3], [4]. Hal ini menciptakan fenomena “kotak hitam” (*black box*) yang serius, di mana ketidakpastian mengenai perilaku agen menciptakan risiko signifikan saat teknologi ini diterapkan pada domain yang berdampak tinggi terhadap keselamatan manusia, terutama pada domain kritis seperti navigasi kendaraan otonom, sistem diagnostik medis, atau manajemen infrastruktur publik, di mana transparansi dan interpretabilitas merupakan prasyarat keamanan yang tidak dapat ditawar [5].

Munculnya sub-bidang *Explainable Reinforcement Learning* (XRL) merupakan respon sistematis terhadap tantangan “kotak hitam” ini, dengan tujuan untuk mengembangkan metodologi yang dapat membantu peneliti, pengembang, dan pengguna akhir memahami, memvalidasi, dan mempercayai perilaku agen RL yang kompleks [3]. XRL mencakup berbagai spektrum pendekatan, mulai dari perancangan model RL yang secara intrinsik dapat diinterpretasikan (*interpretable-by-design*)—seperti penggunaan pohon keputusan, logika simbolik, atau program komputer yang dapat dibaca manusia sebagai representasi kebijakan, hingga pengembangan teknik analisis *post-hoc* yang bertujuan untuk mengekstrak penjelasan dari model yang sudah terlatih [2]. Inovasi dalam XRL kini bergeser dari sekadar metrik evaluasi kinerja statis menuju analisis dinamika temporal, dekomposisi fungsi *reward*, dan identifikasi fitur-fitur lingkungan yang paling berpengaruh terhadap keputusan agen melalui penggunaan peta saliensi atau metode kontrafaktual [5].

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 2, February 2026 | Hal 627-632
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i2.984

Seiring berkembangnya bidang XRL, visualisasi interaktif semakin diakui sebagai jembatan kognitif yang esensial untuk memitigasi kesenjangan pemahaman antara agen kecerdasan buatan dan intuisi manusia. Berbagai metode *visual analytics* canggih telah dikembangkan untuk memberikan wawasan mendalam mengenai bagaimana representasi internal agen berevolusi selama proses pelatihan yang intensif [6], [7]. Melalui pendekatan visual ini, analisis tidak lagi terfragmentasi pada metrik agregat yang bersifat reduksionis, seperti *reward* kumulatif rata-rata, melainkan mampu menyajikan dinamika *return* pada setiap episode, fluktuasi estimasi nilai pada setiap langkah, serta evolusi kebijakan yang terjadi secara bertahap [8], [9]. Namun demikian, alat visualisasi RL yang umum digunakan saat ini masih memiliki keterbatasan fungsional yang nyata. Sebagian besar pustaka visualisasi standar, seperti Matplotlib, didesain untuk menghasilkan output grafis statis yang sangat baik untuk dokumentasi pasca-pelatihan, namun sering kali gagal dalam merepresentasikan dinamika mikro dan perubahan kebijakan agen yang sangat cepat yang terjadi secara real-time selama fase eksplorasi aktif [10]. Keterbatasan temporal ini menghambat kemampuan peneliti untuk melakukan debugging segera terhadap perilaku agen yang anomali atau tidak stabil sebelum proses pelatihan yang memakan waktu lama selesai [11].

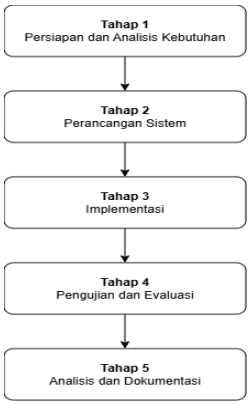
Pentingnya visualisasi dengan *frame rate* (FPS) yang tinggi merupakan aspek teknis yang memiliki implikasi kognitif mendalam dalam *Explainable Reinforcement Learning*. Penelitian dalam psikofisika dan persepsi visual manusia menunjukkan bahwa sistem visual kita memerlukan ambang batas tertentu, biasanya di kisaran 24 hingga 30 FPS, untuk mempersepsikan rangkaian gambar diam sebagai aliran informasi yang halus, kontinu, dan bermakna. Dalam konteks XRL, frame rate yang tinggi hingga 30 FPS atau lebih sangat krusial karena memungkinkan pengamat manusia untuk mendeteksi perubahan gradien nilai-Q yang sangat halus atau pergeseran kebijakan yang terjadi dalam rentang milidetik [12]. Tanpa resolusi temporal yang memadai, fenomena seperti osilasi kebijakan (policy oscillation)—di mana agen terjebak di antara dua tindakan yang memiliki nilai estimasi yang hampir setara—akan tampak sebagai gerakan yang tidak teratur dan membingungkan, padahal hal tersebut merupakan indikator kritis dari ketidakstabilan konvergensi atau masalah pada fungsi reward [13]. Kemampuan visualisasi real-time berkinerja tinggi memungkinkan pemanfaatan ikonik memori manusia untuk mengenali pola, tren, dan pencilan perilaku secara instan, yang pada gilirannya memfasilitasi proses interpretasi yang lebih akurat terhadap logika agen [14].

Kesenjangan teknologi tersebut, penelitian ini mengusulkan sebuah arsitektur visualisasi real-time berbasis Pygame (SDL) untuk Reinforcement Learning yang dirancang untuk mendukung observasi dinamika pembelajaran berkecepatan tinggi. Kontribusi utama dari penelitian ini meliputi: (1) pengembangan arsitektur visualisasi real-time yang memisahkan logika pelatihan agen dari proses rendering grafis melalui mekanisme multi-threading, sehingga mampu mempertahankan visualisasi yang konsisten hingga ≥ 60 FPS; (2) penerapan teknik visualisasi *explainable* untuk RL, termasuk penggunaan *heatmap diverging* yang memungkinkan diferensiasi intuitif antara nilai-Q positif dan negatif [15], panah kebijakan dinamis, serta konsep *Ghost Policies* [13]; dan (3) pelaksanaan studi empiris pada *Cliff Walking Problem* untuk menganalisis perbedaan dinamika pembelajaran visual antara algoritma Q-Learning yang bersifat *off-policy* dan SARSA yang bersifat *on-policy* [16], [17]. Cliff Walking dipilih sebagai skenario evaluasi karena karakteristik lingkungannya yang mampu memperjelas dikotomi antara optimasi jalur terpendek yang berisiko dan pencarian jalur aman yang konservatif, sebuah domain yang sangat relevan untuk dieksplorasi melalui kacamata interpretabilitas visual dan AI Safety [16].

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dilaksanakan melalui serangkaian tahapan sistematis untuk memastikan implementasi yang terstruktur dan hasil yang dapat diverifikasi. Tahapan tersebut dirangkum dalam diagram alur pada Gambar 1.



Gambar 1. Diagram Alur Tahapan Penelitian

2.1.1 Persiapan dan Analisis Kebutuhan

Tahap ini meliputi studi literatur tentang Explainable RL dan identifikasi masalah pada alat visualisasi konvensional. Output utamanya adalah spesifikasi kebutuhan sistem visualisasi real-time dengan target kinerja ≥ 60 FPS.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 2, February 2026 | Hal 627-632
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i2.984

2.1.2 Perancangan Sistem

Pada tahap ini dilakukan perancangan arsitektur hibrida Python-Pygame yang memisahkan logika simulasi dan rendering. Desain mencakup spesifikasi lingkungan Cliff Walking (grid 6x12) serta teknik visualisasi seperti heatmap divergen dan Ghost Policies.

2.1.3 Implementasi

Implementasi dilakukan secara modular terhadap komponen lingkungan, agen RL, dan renderer. Rincian modul yang diimplementasikan disajikan pada Tabel 1.

Tabel 1. Tahapan Implementasi Modul Sistem

No	Modul	Deskripsi Implementasi
1	Environment	Implementasi lingkungan MDP, grid 6x12, dan sistem reward
2	Agent	Implementasi algoritma Q-Learning dan SARSA
3	Visualization	Renderer $\geq$ 60 FPS menggunakan Pygame (SDL)
4	Main	Integrasi sistem simulasi dan visualisasi

2.1.4 Pengujian dan Evaluasi

Tahap ini mencakup pengujian fungsional algoritma, benchmark stabilitas frame rate, dan evaluasi efektivitas visualisasi dalam merepresentasikan nilai Q dan kebijakan agen.

2.1.5 Analisis dan Dokumentasi

Tahap akhir melibatkan analisis komparatif dinamika pembelajaran antara Q-Learning dan SARSA berdasarkan data visual dan metrik yang dikumpulkan, dilanjutkan dengan penyusunan laporan penelitian.

2.2 Lingkungan Cliff Walking

Cliff Walking merupakan permasalahan *grid-world* klasik yang diperkenalkan oleh Sutton & Barto untuk mengilustrasikan perbedaan antara algoritma *on-policy* dan *off-policy* [1]. Pada penelitian ini digunakan varian grid berukuran 6x12 dengan posisi *Start* pada koordinat (5,0) dan *Goal* pada koordinat (5,11), serta dilengkapi dengan sepuluh *Obstacle* yang ditempatkan secara statis pada area *grid*, yaitu pada koorditar berikut (0,4), (0,7), (1,2), (1,5), (1,9), (2,3), (2,6), (2,7), (3,4), dan (3,8).

Tabel 2. Konfigurasi reward lingkungan *Cliff Walking*

Event	*Reward*	Konsekuensi Lingkungan
Memasuki sel *Cliff*	-100	Episode direset ke *Start*
Mengenai *Obstacle*	-20	Pergerakan dibatalkan
Langkah biasa	-1	Tidak ada perubahan status
Mencapai *Goal*	+10	Episode berakhir

Tabel 2 menyajikan konfigurasi *reward* pada lingkungan *Cliff Walking* yang digunakan dalam penelitian ini. Agen menerima penalti sebesar −100 ketika memasuki sel *Cliff* serta konsekuensi episode direset ke posisi awal. Jika agen mencoba memasuki *Obstacle*, diberikan penalti −20 serta konsekuensi pergerakan tersebut tidak diizinkan. Setiap langkah normal dikenai penalti −1, sedangkan pencapaian posisi *Goal* menghasilkan *reward* sebesar +10 dan menandai berakhirnya episode.

Lingkungan ini merepresentasikan *trade-off* antara risiko dan efisiensi, di mana agen dihadapkan pada pilihan antara jalur pendek di dekat jurang dengan risiko tinggi dan jalur alternatif yang lebih aman namun memiliki panjang lintasan yang lebih besar [18].

2.3 Algoritma Q-Learning

Q-Learning merupakan algoritma *off-policy* yang menggunakan pendekatan optimistik [19]. Rumus pembaharuan:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right] \quad (1)$$

dengan keterangan:

s_t : *state* pada waktu t
a_t : aksi yang diambil pada state s_t
r_{t+1} : *reward* yang diterima setelah melakukan aksi
s_{t+1} : *state* berikutnya
α : *learning rate*
γ : *discount factor*
$\max_{a'} Q(s_{t+1}, a')$: estimasi nilai optimal pada state berikutnya

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 2, February 2026 | Hal 627-632
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i2.984

Intuisi dasar dari Q-Learning adalah bahwa agen memperbarui nilai tindakannya dengan mengasumsikan, setelah mencapai state s_{t+1}, agen akan selalu memilih aksi terbaik secara optimal, terlepas dari aksi yang benar-benar diambil dalam proses eksplorasi [20].

2.4 Algoritma SARSA

SARSA (*State-Action-Reward-State-Action*) adalah algoritma *on-policy* [21] dengan rumus pembaharuan sebagai berikut:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (2)$$

dengan keterangan:

s_t : *state* pada waktu t
a_t : aksi yang diambil pada state s_t
r_{t+1} : *reward* yang diterima setelah melakukan aksi a_t
s_{t+1} : *state* berikutnya
a_{t+1} : aksi berikutnya yang dipilih oleh agen sesuai kebijakan saat ini
α : *learning rate*, pengontrol besar pembaruan nilai
γ : *discount factor*, penentu pengaruh reward masa depan
$Q(s_{t+1}, a_{t+1})$: estimasi nilai Q untuk pasangan state–aksi berikutnya

Rumus ini mencerminkan sifat on-policy SARSA, karena pembaruan nilai Q bergantung pada aksi yang benar-benar dipilih oleh agen pada state berikutnya, bukan pada aksi optimal secara teoritis. Berbeda dengan Q-Learning, SARSA belajar nilai dari kebijakan yang tengah diikutinya, termasuk konsekuensi aksi eksplorasi [22]. Konsekuensi dari perbedaan pendekatan tersebut adalah bahwa SARSA cenderung bersifat lebih konservatif, karena proses pembelajarannya turut mempertimbangkan dampak dari aksi eksplorasi yang bersifat sub-optimal. Sebaliknya, Q-Learning memiliki karakteristik yang lebih optimistis, karena pembaruan nilainya didasarkan pada asumsi bahwa agen akan selalu memilih aksi terbaik pada state berikutnya [23].

Dalam konteks *Cliff Walking*, perbedaan karakteristik tersebut menyebabkan SARSA cenderung mempelajari kebijakan yang lebih berhati-hati dengan menjauhi area berisiko di dekat jurang, karena algoritma ini memperhitungkan kemungkinan terjadinya penalti akibat aksi eksplorasi. Sebaliknya, Q-Learning lebih condong memilih jalur terpendek yang berada di sepanjang tepi jurang, karena pembaruan nilainya mengasumsikan bahwa agen akan selalu bertindak secara optimal tanpa mempertimbangkan risiko eksplorasi secara eksplisit [16].

2.5 Arsitektur Visualisasi Real-Time dan Teknik Visualisasi XRL

Arsitektur perangkat lunak yang diusulkan terdiri dari dua komponen utama, yaitu komponen pelatihan RL yang diimplementasikan menggunakan Python dan komponen visualisasi yang dibangun dengan Pygame serta memanfaatkan akselerasi perangkat keras. Kedua komponen ini dijalankan secara terpisah dan berkomunikasi melalui mekanisme *shared memory*, sehingga data pembelajaran seperti posisi agen, nilai Q, dan kebijakan dapat dibagikan secara efisien. Pemisahan ini memungkinkan proses pembaruan grafis dilakukan secara real-time tanpa mengganggu atau memperlambat *loop* pelatihan agen, sehingga kinerja pembelajaran tetap terjaga sekaligus menyediakan visualisasi yang responsif.

Setiap sel pada lingkungan *grid* divisualisasikan menggunakan sejumlah teknik visualisasi *explainable* untuk meningkatkan interpretabilitas model. Nilai Q direpresentasikan dalam bentuk *heatmap* menggunakan skema warna divergen berbasis palet turbo [15], yang memungkinkan perbedaan nilai antaraksi pada setiap state diamati secara jelas. Visualisasi ini membantu dalam mengidentifikasi area dengan potensi reward tinggi maupun risiko penalti yang signifikan.

Kebijakan agen divisualisasikan melalui panah kebijakan dinamis yang menunjukkan aksi dengan nilai Q tertinggi (argmax Q) pada setiap state. Representasi ini memungkinkan pola kebijakan yang dipelajari agen diamati secara langsung, serta mempermudah analisis perubahan kebijakan selama proses pelatihan berlangsung.

Konsep *Ghost Policies*, yaitu visualisasi trajektori kegagalan agen yang ditampilkan sebagai jejak transparan pada lingkungan grid [13], diterapkan untuk mendukung analisis kesalahan pembelajaran. Jejak ini merepresentasikan lintasan episode yang berakhir pada kondisi gagal, sehingga memberikan informasi tambahan mengenai area lingkungan yang berisiko dan perilaku agen pada fase eksplorasi.

Selain visualisasi berbasis *grid*, sistem ini dilengkapi dengan panel metrik real-time yang menyajikan informasi kuantitatif selama pelatihan, meliputi jumlah episode, nilai epsilon, total *reward* per episode, serta jumlah kejadian jatuh ke *Cliff* (*falls*). Penyajian metrik secara langsung ini memungkinkan pemantauan perkembangan dan stabilitas proses pembelajaran agen secara berkelanjutan.

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil eksperimen serta pembahasan terhadap perilaku dan kinerja algoritma Q-Learning dan SARSA pada lingkungan *Cliff Walking*. Analisis dilakukan berdasarkan visualisasi real-time yang dihasilkan selama

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 2, February 2026 | Hal 627-632
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i2.984

proses pelatihan, serta metrik kuantitatif yang diamati. Pembahasan difokuskan pada perbandingan dinamika pembelajaran, karakteristik kebijakan yang dihasilkan, dan implikasinya terhadap aspek *explainability* dalam RL.

3.1 Arsitektur Visualisasi

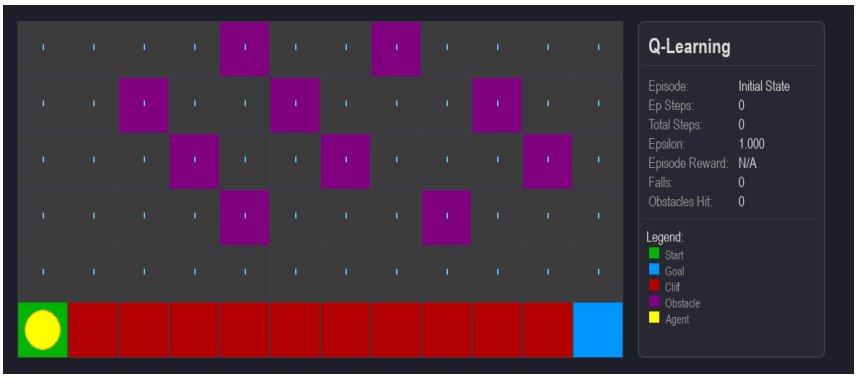
Implementasi arsitektur visualisasi real-time dalam penelitian ini berhasil memvalidasi pemisahan antara logika pelatihan agen dan unit rendering grafis melalui mekanisme shared memory. Integrasi sistem ini memungkinkan data state lingkungan diperbarui secara asinkron tanpa menyebabkan degradasi performa pada kecepatan komputasi algoritma, sehingga stabilitas frame rate tetap terjaga pada kisaran 60 FPS untuk mendukung pengamatan dinamika mikro agen secara lancar. Penggunaan skema warna divergen pada heatmap berbasis palet Turbo secara efektif memberikan kontras visual yang tajam, di mana gradasi dari hijau ke merah memungkinkan peneliti melakukan diagnosis cepat terhadap area yang dianggap menguntungkan atau berisiko tinggi oleh agen. Aspek explainability diperkuat melalui penggunaan panah kebijakan dinamis yang merepresentasikan fungsi $\text{argmax}Q(s,a)$, sehingga perubahan arah panah selama pelatihan dapat digunakan sebagai indikator transisi dari fase eksplorasi menuju konvergensi kebijakan. Selain itu, penerapan konsep *Ghost Policies* memberikan dimensi temporal tambahan dengan menampilkan jejak trajektori kegagalan agen sebagai transparansi visual, yang mempermudah identifikasi pola kesalahan spesifik pada area *Cliff*. Seluruh informasi visual ini disinkronisasikan secara simultan dengan panel metrik real-time di sisi kanan antarmuka, memastikan bahwa observasi kualitatif terhadap perilaku agen selalu selaras dengan parameter numerik seperti nilai ϵ, *total steps*, dan frekuensi jatuh (*falls*) yang terekam sepanjang proses simulasi.

3.2 Visualisasi Kebijakan pada Tahap Kunci

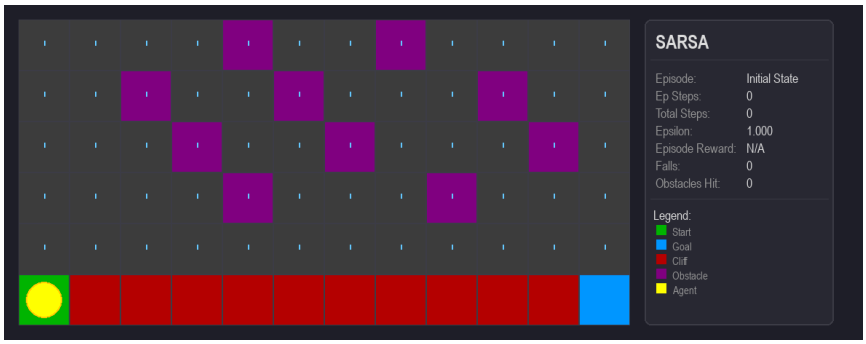
Berdasarkan arsitektur yang telah dijelaskan, bagian ini menyajikan hasil visualisasi kebijakan yang dihasilkan oleh algoritma Q-Learning dan SARSA pada beberapa tahap pembelajaran yang dianggap representatif. Melalui teknik heatmap dan panah kebijakan dinamis, perbedaan karakteristik pembelajaran antara kedua algoritma dapat diamati secara langsung, khususnya dalam merespons risiko lingkungan di sekitar area *Cliff*. Evolusi kebijakan ini direkam dari tahap awal pelatihan hingga tercapainya stabilitas kebijakan final untuk memberikan gambaran komprehensif mengenai dinamika pembelajaran agen.

3.2.1 Episode 0 – Kondisi Awal (Tabula Rasa)

Pada tahap inisiasi pelatihan, kedua algoritma diinisiasi dalam kondisi *tabula rasa* atau belum memiliki pengetahuan mengenai struktur lingkungan *Cliff Walking*..



Gambar 2. Kondisi awal Q-Learning: *Grid* 6×12, Q = 0, ε = 1.0.



Gambar 3. Kondisi awal SARSA: *Grid* 6×12, Q = 0, ε = 1.0.

Sebagaimana terlihat pada Gambar 2 dan Gambar 3, seluruh pasangan *state-action* memiliki nilai fungsi Q yang seragam yaitu nol ($Q(s,a) = 0$), sehingga tidak ada indikator panah kebijakan yang muncul pada setiap sel. Kondisi ini didukung oleh nilai epsilon (ϵ) sebesar 1000 yang memaksa agen (lingkaran kuning) untuk melakukan eksplorasi murni secara acak demi memetakan lingkungan dari titik *Start* (kotak hijau) menuju *Goal* (kotak biru).

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 2, February 2026 | Hal 627-632
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i2.984

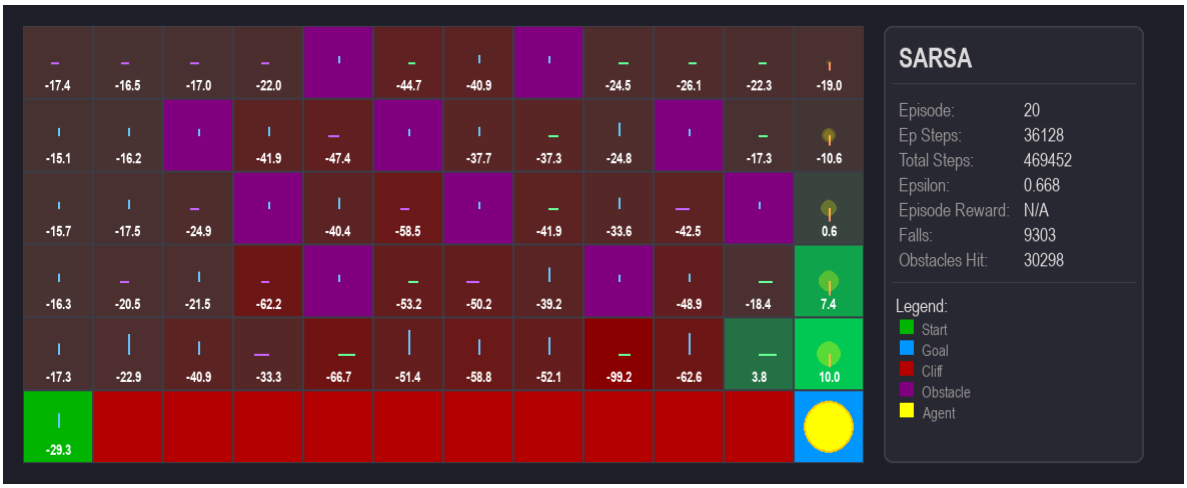
3.2.2 Episode 20 – Awal Pembelajaran ("*The Red Burn*")

Memasuki episode ke-20, kedua agen mulai mengenali pola hukuman dan imbalan dari lingkungan seiring dengan peluruhan nilai epsilon (ϵ) menjadi 0,668. Berdasarkan Gambar 4 dan Gambar 5, transparansi proses belajar ini diperjelas melalui penggunaan *heatmap divergen*, di mana gradasi warna hijau merepresentasikan akumulasi nilai Q positif yang mengarah ke target, sedangkan gradasi merah hingga cokelat gelap mengindikasikan nilai negatif atau zona berisiko tinggi. Nilai numerik yang tertera di tengah setiap sel merepresentasikan $\max Q(s,a)$, yaitu estimasi *return* tertinggi yang diharapkan agen pada posisi tersebut, yang berfungsi sebagai indikator optimisme atau kehati-hatian agen terhadap suatu area.

Indikator garis berwarna di dalam setiap sel memberikan gambaran mikro mengenai preferensi aksi agen pada setiap *state*. Garis hijau merepresentasikan nilai aksi ke arah kanan, garis biru (selain yang di dalam kotak *Obstacle*) ke arah atas, garis ungu ke arah kiri, dan garis oranye ke arah bawah. Panjang dan intensitas warna garis-garis tersebut mencerminkan nilai Q relatif; garis yang lebih dominan menunjukkan aksi yang dianggap paling menguntungkan oleh agen saat itu. Khusus pada sel obstacle berwarna ungu, munculnya garis biru bukan merupakan instruksi pergerakan, melainkan penanda status sel tersebut sebagai hambatan permanen yang tidak dapat dilewati (*non-traversable*).

Pada tahap ini, perbedaan karakteristik antara kedua algoritma mulai terlihat jelas. Q-Learning menunjukkan sifat agresif dengan mulai membangun jalur pendek di sepanjang tepian jurang, meskipun harus membayar harga mahal dengan catatan 10.162 kali terjatuh sebagaimana dikonfirmasi oleh panel metrik pada Gambar 4. Sebaliknya, SARSA pada Gambar 5 menunjukkan perilaku yang lebih konservatif (*risk-averse*). Hal ini dibuktikan dengan penurunan nilai heatmap yang drastis pada sel-sel di atas tebing (mencapai -99.2), yang secara visual mendorong agen untuk lebih banyak berinteraksi dengan hambatan ungu demi menghindari risiko jatuh ke jurang merah. Keberadaan jejak transparan atau *Ghost Policies* pada area *Cliff* di kedua gambar memberikan konteks spasial bagi peneliti mengenai riwayat kegagalan agen, sekaligus memvalidasi bahwa agen sedang dalam proses intensif untuk mempelajari batas-batas keamanan navigasi di dalam lingkungan tersebut.

Gambar 4. Visualisasi Nilai Q dan Kebijakan Agen Q-Learning pada Episode 20



Gambar 5. Visualisasi Nilai Q dan Kebijakan Agen SARSA pada Episode 20

3.2.3 Episode 50 – Polarisasi Strategi ("*The Green Snake*")

Pada episode ke-50, polarisasi strategi antara kedua algoritma menjadi semakin nyata dengan tingkat eksplorasi yang menurun ke $\epsilon = 0.364$.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 2, February 2026 | Hal 627-632
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i2.984

Gambar 6. Visualisasi Nilai Q dan Kebijakan Agen Q-Learning pada Episode 50

Gambar 7. Visualisasi Nilai Q dan Kebijakan Agen SARSA pada Episode 50

Melalui Gambar 6, teramati bahwa Q-Learning secara agresif mempertahankan jalur optimal sepanjang 18 langkah di bibir tebing karena sifat *off-policy* yang hanya mempertimbangkan aksi terbaik di masa depan. Sebaliknya, Gambar 7 memperlihatkan agen SARSA yang memilih rute memutar yang lebih jauh guna meminimalkan risiko terjatuh, meskipun hal tersebut berdampak pada akumulasi total langkah yang jauh lebih masif serta jumlah tabrakan hambatan yang mencapai 99.534 kali.

3.2.4 Episode 200 – Menjelang Konvergensi

Saat mendekati tahap konvergensi pada episode 200, nilai ϵ yang telah meluruh hingga 0.018 membuat kebijakan agen menjadi sangat stabil.

Gambar 8. Visualisasi Nilai Q dan Kebijakan Agen Q-Learning pada Episode 200

Sebagaimana terlihat pada Gambar 8, Q-Learning berhasil mengunci rute terpendek dengan efisiensi 13 langkah per episode dan jumlah jatuh yang tidak lagi bertambah secara signifikan. Di sisi lain, Gambar 9 menunjukkan konvergensi SARSA pada jalur aman yang melambung ke atas, di mana agen membutuhkan 23 langkah untuk mencapai target namun berhasil mengamankan diri dari potensi kesalahan akibat sisa eksplorasi yang masih ada.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 2, February 2026 | Hal 627-632
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i2.984

Gambar 9. Visualisasi Nilai Q dan Kebijakan Agen SARSA pada Episode 200

3.2.5 Kebijakan Final

Pada fase akhir simulasi dengan kondisi $\epsilon = 0.000$, kedua algoritma telah menghasilkan kebijakan deterministik yang permanen. Gambar 10 menampilkan hasil akhir Q-Learning yang berupa rute terpendek melalui tepi jurang, yang membuktikan efisiensi matematis namun memiliki kerentanan tinggi terhadap gangguan. Sebaliknya, Gambar 11 menunjukkan kebijakan final SARSA yang berupa rute aman melingkar sejauh 23 langkah. Perbedaan hasil ini mempertegas karakteristik Q-Learning yang berorientasi pada hasil optimal dan SARSA yang berorientasi pada manajemen risiko dalam lingkungan yang berbahaya.

Gambar 10. Visualisasi Jalur Optimal Agen Q-Learning

Gambar 11. Visualisasi Jalur Optimal Agen Q-Learning

3.3 Pembahasan

Bagian ini menyajikan analisis kuantitatif untuk mengevaluasi efektivitas Q-Learning dan SARSA dalam menyelesaikan permasalahan Cliff Walking. Perbandingan dilakukan dengan meninjau metrik performa utama yang mencerminkan

dinamika pembelajaran, efisiensi jalur, dan kecenderungan perilaku agen terhadap risiko lingkungan. Data metrik yang dikumpulkan dari tahap awal hingga tercapainya kebijakan final dirangkum dalam Tabel 3.

Tabel 3. Perbandingan Metrik Performa Q-Learning vs SARSA

Fase Pembelajaran	Algoritma	Ep Steps	Total Falls	Obstacle Hit	Karakteristik Jalur
Awal (Ep 0)	Q-Learning	0	0	0	Acak (Tabula Rasa)
	SARSA	0	0	0	Acak (Tabula Rasa)
Transisi (Ep 20)	Q-Learning	41	10162	11817	Penelusuran Tepi Jurang
	SARSA	36128	9303	30298	Penghindaran Area Bahaya
Jalur Optimal	Q-Learning	13	10642	11975	Shortest Path
	SARSA	23	22972	232844	Safe Path

Berdasarkan data pada Tabel 3, terlihat perbedaan fundamental dalam cara kedua algoritma memproses pengalaman dan mengelola risiko. Pada fase transisi (Episode 20), Q-Learning menunjukkan agresivitas yang lebih tinggi dengan jumlah langkah per episode (Ep Steps) yang jauh lebih rendah, yakni 41 langkah, berbanding terbalik dengan SARSA yang mencapai 36.128 langkah. Pada tahap transisi, Ep Step SARSA sangat tinggi dikarenakan rute pergerakan yang dipilih menjauhi jurang dan memungkinkan menabrak *obstacle* lebih sering. Tingginya angka Obstacles Hit pada SARSA (mencapai 232.844 pada tahap lanjut) mengindikasikan bahwa algoritma *on-policy* ini lebih memilih menerima penalti kecil akibat menabrak hambatan secara berulang daripada mengambil risiko jatuh ke jurang yang memberikan penalti ekstrem. Hal ini menjelaskan mengapa kebijakan final SARSA memerlukan 23 langkah; agen belajar untuk "melambung" jauh dari tepi jurang demi menjamin keamanan navigasi.

Sebaliknya, Q-Learning berhasil mengonvergensi kebijakan pada jalur terpendek (13 langkah) dengan hanya mencatatkan akumulasi jatuh sebesar 10.642 kali. Sifat *off-policy* pada Q-Learning memungkinkannya untuk mengabaikan potensi kesalahan akibat aksi eksplorasi acak dan tetap berfokus pada nilai aksi maksimal di tepi jurang. Meskipun Q-Learning unggul secara signifikan dalam aspek efisiensi waktu tempuh dan jumlah langkah total, SARSA menawarkan kebijakan yang lebih robust dan stabil dalam lingkungan yang memiliki ketidakpastian tinggi. Secara keseluruhan, temuan ini menegaskan bahwa pemilihan algoritma harus disesuaikan dengan prioritas sistem: Q-Learning untuk efisiensi jalur maksimal, atau SARSA untuk keamanan operasional yang lebih terjamin.

4. KESIMPULAN

Penelitian ini berhasil memvalidasi kerangka kerja visualisasi real-time berkinerja tinggi yang mengatasi tantangan *explainability* pada Reinforcement Learning (RL) melalui pemisahan logika agen dan rendering grafis berbasis Pygame pada kecepatan ≥ 60 FPS. Arsitektur ini menjawab masalah "kotak hitam" dengan menyajikan evolusi kebijakan mikro secara responsif dibandingkan alat statis konvensional. Eksperimen Cliff Walking mengungkap perbedaan fundamental antara Q-Learning yang agresif pada jalur optimal tepi jurang dan SARSA yang konservatif dengan membentuk "zona penyangga" keamanan. Melalui heatmap divergen dan panah kebijakan dinamis, tervisualisasi bahwa Q-Learning mengutamakan efisiensi, sementara SARSA memprioritaskan keamanan operasional dengan menanggung langkah akumulatif lebih tinggi. Salah satu kontribusi utama dalam penelitian ini adalah penerapan teknik Ghost Policies yang tidak hanya memberikan konteks spasial tetapi juga konteks temporal dengan memvisualisasikan sejarah kegagalan masa lalu agen yang biasanya hilang setelah episode berakhir, sehingga memberikan pemahaman yang lebih komprehensif mengenai pola perilaku agen dalam menghadapi risiko. Meskipun efektif pada lingkungan grid-world, pengembangan selanjutnya disarankan untuk menyasar skalabilitas pada Deep RL dengan integrasi teknik reduksi dimensi dan mesin rendering 3D untuk menangkap kompleksitas representasi fitur pada lingkungan simulasi yang lebih rumit.

REFERENCES

[1] R. S. Sutton and A. Barto, *Reinforcement learning: an introduction*, Second edition. in Adaptive computation and machine learning. Cambridge, Massachusetts London, England: The MIT Press, 2020.
[2] Y. Qing *et al.*, "A Survey on Explainable Reinforcement Learning: Concepts, Algorithms, Challenges," 2022, *arXiv*. doi: 10.48550/ARXIV.2211.06665.
[3] S. Milani, N. Topin, M. Veloso, and F. Fang, "Explainable Reinforcement Learning: A Survey and Comparative Review," *ACM Comput. Surv.*, vol. 56, no. 7, pp. 1–36, Jul. 2024, doi: 10.1145/3616864.
[4] L. Saulières, "A Survey of Explainable Reinforcement Learning: Targets, Methods and Needs," Jul. 16, 2025, *arXiv*: arXiv:2507.12599. doi: 10.48550/arXiv.2507.12599.
[5] P. Li, U. Siddique, and Y. Cao, "From Explainability to Interpretability: Interpretable Reinforcement Learning Via Model Explanations," in *Proceedings of the Reinforcement Learning Conference*, 2025. [Online]. Available: https://openreview.net/forum?id=kreQkWaOK5
[6] J. Wang, L. Gou, H.-W. Shen, and H. Yang, "DQNViz: A Visual Analytics Approach to Understand Deep Q-Networks," *IEEE Trans. Visual. Comput. Graphics*, vol. 25, no. 1, pp. 288–298, Jan. 2019, doi: 10.1109/TVCG.2018.2864504.
[7] B. La Rosa *et al.*, "State of the Art of Visual Analytics for eXplainable Deep Learning," *Computer Graphics Forum*, vol. 42, no. 1, pp. 319–355, Feb. 2023, doi: 10.1111/cgf.14733.
[8] Z. Cheng, J. Yu, and X. Xing, "A Survey on Explainable Deep Reinforcement Learning," 2025, *arXiv*. doi: 10.48550/ARXIV.2502.06869.

[9] CodeSignal, "Visualizing Training Statistics in Reinforcement Learning." [Online]. Available: https://codesignal.com/learn/courses/game-on-integrating-rl-agents-with-environments/lessons/visualizing-training-statistics-in-reinforcement-learning
[10] SourceForge, "Matplotlib Alternatives & Competitors." [Online]. Available: https://sourceforge.net/software/product/Matplotlib/alternatives
[11] MathWorks, "What Is Reinforcement Learning?" [Online]. Available: https://uk.mathworks.com/discovery/reinforcement-learning.html
[12] UnitX Labs, "How Frame Rate Impacts Machine Vision Performance." [Online]. Available: https://www.unitxlabs.com/frame-rate-machine-vision-performance/
[13] X. Olaz, "Ghost Policies: A New Paradigm for Understanding and Learning from Failure in Deep Reinforcement Learning," 2025, *arXiv*. doi: 10.48550/ARXIV.2506.12366.
[14] R. Fusco *et al.*, "Visual Perception and Pre-Attentive Attributes in Oncological Data Visualisation," *Bioengineering*, vol. 12, no. 7, p. 782, Jul. 2025, doi: 10.3390/bioengineering12070782.
[15] K. Moreland, "Diverging Color Maps for Scientific Visualization," in *Advances in Visual Computing*, vol. 5876, G. Bebis, R. Boyle, B. Parvin, D. Koracin, Y. Kuno, J. Wang, R. Pajarola, P. Lindstrom, A. Hinkenjann, M. L. Encarnação, C. T. Silva, and D. Coming, Eds., in Lecture Notes in Computer Science, vol. 5876. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 92–103. doi: 10.1007/978-3-642-10520-3_9.
[16] L. Zhong, "Comparison of Q-learning and SARSA Reinforcement Learning Models on Cliff Walking Problem," vol. 180, Dordrecht: Atlantis Press International BV, 2024, pp. 207–213. doi: 10.2991/978-94-6463-370-2_23.
[17] Baeldung, "Q-Learning vs SARSA." [Online]. Available: https://www.baeldung.com/cs/q-learning-vs-sarsa
[18] "Cliff Walking," Gymnasium Documentation. [Online]. Available: https://gymnasium.farama.org/environments/toy_text/cliff_walking/
[19] C. J. C. H. Watkins, "Learning from Delayed Rewards," King's College, 1989.
[20] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach Learn*, vol. 8, no. 3–4, pp. 279–292, May 1992, doi: 10.1007/BF00992698.
[21] G. A. Rummery and M. Niranjan, "On-line Q-learning using connectionist systems," 1994. [Online]. Available: https://api.semanticscholar.org/CorpusID:59872172
[22] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, "Convergence Results for Single-Step On-Policy Reinforcement-Learning Algorithms," *Machine Learning*, vol. 38, no. 3, pp. 287–308, Mar. 2000, doi: 10.1023/A:1007678930559.
[23] ApX Machine Learning, "Comparing SARSA and Q-Learning." [Online]. Available: https://apxml.com/courses/intro-to-reinforcement-learning/chapter-5-temporal-difference-learning/comparing-sarsa-q-learning