

Laporan Praktikum Kontrol Cerdas

Nama : Aureyza Pandu Qinara
NIM : 224308078
Kelas : TKA - 7D
Akun Github (Tautan) : <https://github.com/aureyzapandu>
Student Lab Assistant :-

Judul Percobaan

Week 5: *Deep Reinforcement Learning* untuk Kontrol Kompleks

1. Tujuan Percobaan

Tujuan dari praktikum “*Deep Reinforcement Learning* untuk Kontrol Kompleks”, mahasiswa diharapkan mampu:

- a) Memahami konsep *Deep Reinforcement Learning* (DRL) dalam kontrol sistem kompleks.
- b) Mengimplementasikan *Deep Q-Network* (DQN) untuk kontrol otomatis.
- c) Menganalisis performa DRL dibandingkan dengan metode kontrol konvensional.

2. Landasan Teori

- *Deep Reinforcement Learning* (DRL)

Deep Reinforcement Learning (DRL) merupakan pengembangan dari *Reinforcement Learning* (RL) yang memanfaatkan neural network untuk memecahkan masalah dengan ruang keadaan (*state space*) yang kompleks dan kontinyu. Pada dasarnya, RL berfokus pada pembelajaran berbasis pengalaman, di mana agen belajar mengambil keputusan optimal melalui interaksi dengan lingkungan untuk memaksimalkan *reward* kumulatif. Sementara itu, DRL mengintegrasikan *deep learning* untuk mempelajari representasi fitur dari data mentah sehingga mampu menangani masalah berskala besar dan dinamis. Menurut (Li, 2018), DRL adalah kombinasi dari *deep learning* dan *reinforcement learning* yang memungkinkan agen untuk belajar secara *end-to-end* dari data sensorik mentah menuju tindakan optimal, tanpa memerlukan model eksplisit dari lingkungan. Pendekatan ini

telah berhasil diterapkan pada berbagai bidang seperti pengendalian robot, kendaraan otonom, dan sistem energi pintar (Arulkumaran et al., 2017)

- *Deep Q-Network (DQN)*

DQN merupakan pengembangan dari algoritma *Q-Learning* yang menggabungkan *Deep Neural Network* untuk mengaproksimasi fungsi nilai (*Q-value*). *Q-value* merepresentasikan seberapa baik suatu aksi diambil dalam kondisi tertentu. Dengan adanya *neural network*, DQN dapat menangani *state space* yang besar dan kompleks yang sulit ditangani *Q-Learning* biasa (Mnih et al., 2015).

Modul praktik mencontohkan penerapan DQN untuk mengendalikan robot sederhana pada lingkungan simulasi *CartPole* di *OpenAI Gym*, menggunakan dua *hidden layers* dengan fungsi aktivasi *ReLU*. Pendekatan ini juga menerapkan *epsilon-greedy policy* untuk menyeimbangkan antara eksplorasi dan eksploitasi, serta *experience replay* untuk menstabilkan proses pembelajaran

- Penerapan DRL dalam Sistem Kontrol

Dalam sistem kontrol kompleks seperti robotika, kendaraan otonom, dan sistem tenaga listrik, DRL memungkinkan pembelajaran strategi kontrol adaptif tanpa perlu model sistem yang presisi. Dibandingkan dengan metode kontrol konvensional seperti PID atau *fuzzy logic*, DRL mampu beradaptasi terhadap dinamika lingkungan yang berubah-ubah dan ketidakpastian sistem (Kiumarsi et al., 2014).

Pendekatan ini semakin relevan untuk sistem otomatis modern karena dapat melakukan optimasi berbasis pengalaman secara berkelanjutan, yang menjadikannya unggul dalam konteks adaptive control dan self-learning systems.

3. Analisis dan Diskusi

Metode *Reinforcement Learning* (RL) secara umum berfungsi untuk melatih agen agar dapat belajar dari interaksi dengan lingkungan melalui mekanisme *trial and error*, dengan tujuan memaksimalkan *reward* kumulatif. Namun, pendekatan RL konvensional seperti *Q-Learning* memiliki keterbatasan besar saat diterapkan pada sistem dengan ruang keadaan yang kompleks

atau kontinu, karena metode ini menyimpan nilai Q untuk setiap pasangan keadaan dan aksi dalam bentuk tabel (Q -table). Ketika dimensi keadaan meningkat, ukuran tabel menjadi sangat besar, menyebabkan inefisiensi memori dan waktu komputasi yang tinggi.

Sebagai pengembangan dari RL, muncul konsep *Deep Reinforcement Learning* (DRL) yang mengombinasikan pembelajaran penguatan dengan deep learning. Dalam DRL, jaringan saraf tiruan (neural network) digunakan untuk mengaproksimasi fungsi nilai (*value function*) atau kebijakan (*policy*), sehingga agen dapat melakukan generalisasi terhadap kondisi baru tanpa perlu menyimpan seluruh pasangan keadaan-aksi secara eksplisit. Pendekatan ini menjadikan DRL jauh lebih efisien untuk sistem dinamis kompleks, seperti robotika, kendaraan otonom, maupun sistem kontrol industri.

Salah satu implementasi utama DRL adalah *Deep Q-Network* (DQN), yang merupakan versi lanjutan dari *Q-Learning* dengan fungsi nilai Q yang didekati oleh jaringan saraf. DQN memperkenalkan dua inovasi penting, yaitu *experience replay* dan *target network*, yang berperan dalam menstabilkan proses pembelajaran. *Experience replay* menyimpan pengalaman masa lalu untuk dilatih ulang secara acak, sehingga mengurangi korelasi data antar langkah waktu, sementara *target network* memperbarui bobot jaringan secara terpisah untuk menghindari osilasi nilai Q yang berlebihan.

Dari hasil eksperimen pada modul praktik, penerapan DQN sebagai bagian dari DRL pada *environment CartPole-v1* menunjukkan hasil yang cukup baik. Agen mampu mempelajari strategi keseimbangan dengan rata-rata skor tinggi, menandakan bahwa DQN efektif dalam menangani sistem dengan ruang aksi diskret dan dinamika sederhana. Sebaliknya, pada *Pendulum-v1*, performa DQN relatif rendah karena sistem tersebut memiliki ruang aksi kontinu, yang secara prinsip tidak sesuai dengan rancangan asli DQN. Hal ini menggambarkan keterbatasan DQN dalam *domain* kontrol kontinu, yang umum dijumpai pada aplikasi nyata seperti kendali lengan robot atau drone.

Untuk mengatasi keterbatasan ini, berbagai varian dan pengembangan DRL telah diperkenalkan, seperti:

- *Double DQN* untuk mengurangi bias estimasi nilai Q,
- *Dueling DQN* untuk efisiensi representasi nilai,
- *Rainbow DQN* yang menggabungkan beberapa teknik peningkatan performa, dan
- Algoritma *policy-based* seperti PPO (Proximal Policy Optimization) atau DDPG (*Deep Deterministic Policy Gradient*) yang lebih cocok untuk *continuous action space*.

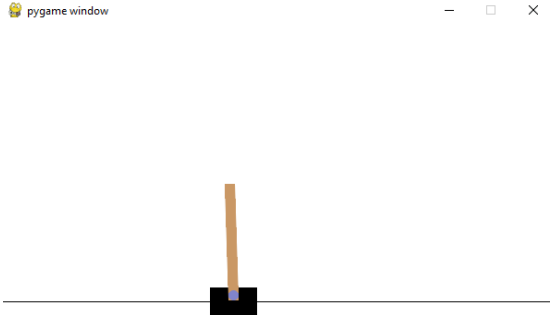
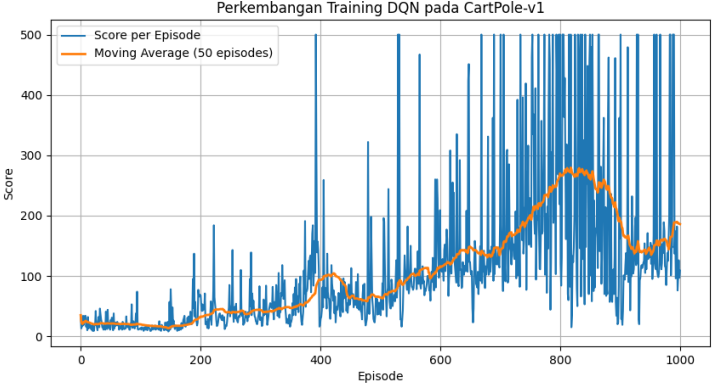

Dalam konteks kendali sistem dinamis nyata, seperti robot industri atau drone, DRL sering diterapkan dengan strategi *Sim-to-Real Transfer*, yaitu melatih agen terlebih dahulu dalam simulasi sebelum diuji di dunia nyata guna mengurangi risiko kerusakan perangkat. Selain itu, digunakan pula pendekatan kontrol **hibrida**, di mana DRL menangani pengambilan keputusan tingkat tinggi (misalnya navigasi atau perencanaan lintasan), sementara kontrol tingkat rendah (seperti kestabilan sudut atau kecepatan motor) tetap dikendalikan oleh metode klasik seperti **PID** atau **LQR**.

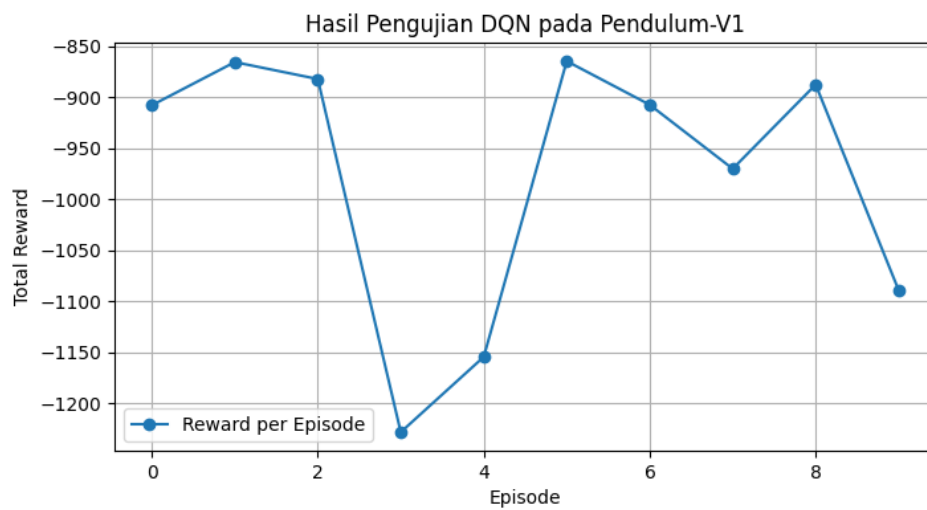
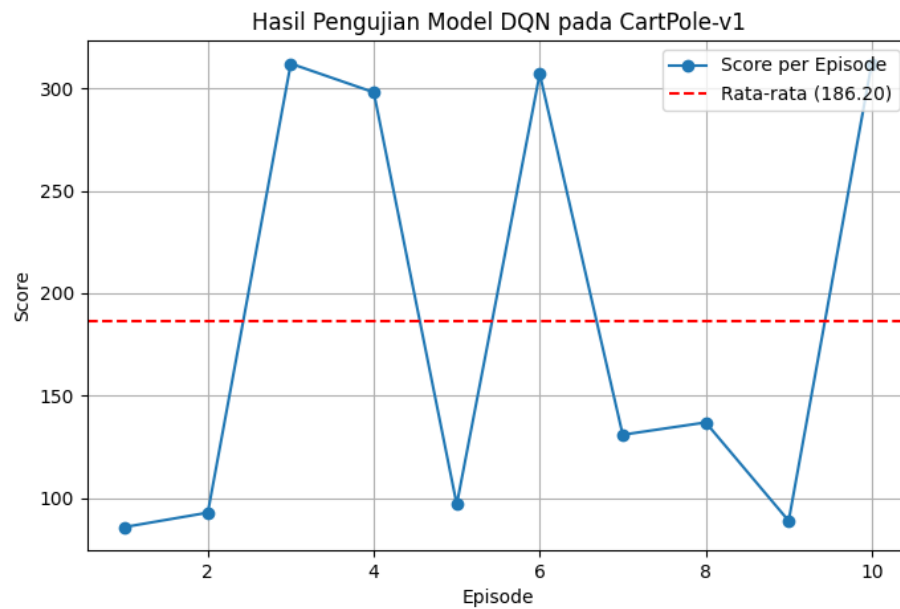
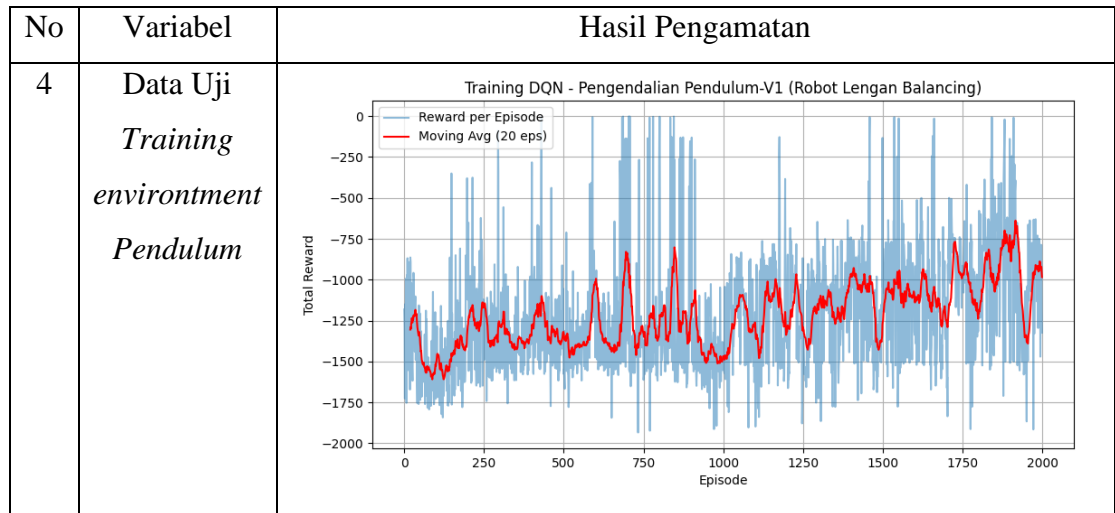
4. Assignment

1. Melakukan studi literatur tentang *Reinforcement Learning* (RL), DQN untuk sistem kontrol.
2. Membuat akun dan *repository* GitHub untuk version control proyek.
3. Menginstal Python, VSCode, Git, serta library: gym, numpy, matplotlib, stable-baselines3.
4. Menyiapkan environment *CartPole-v1* dan Pendulum-v1.
5. Mengimplementasikan DQN untuk CartPole dan Pendulum.
6. Melatih model dengan hyperparameter tertentu (learning rate, gamma, epsilon, dll.).
7. Merekam reward per episode dan membuat grafik *learning curve*.
8. Mengevaluasi model tanpa eksplorasi untuk menguji stabilitas dan performa kendali.
9. Menyimpan hasil eksperimen (model, grafik, video simulasi).

10. Melakukan commit dan push ke GitHub sebagai dokumentasi akhir.

5. Data dan Output Hasil Pengamatan

No	Variabel	Hasil Pengamatan
1	Hasil <i>enviromtment</i> <i>CartPole</i>	
2	Data Uji <i>Training</i> <i>enviromtment</i> <i>CartPole</i>	
3	Hasil <i>enviromtment</i> <i>Pendulum</i>	



Berdasarkan hasil pengujian model *Deep Q-Network* (DQN) pada dua *environment* yang berbeda, yaitu *CartPole-v1* dan *Pendulum-v1*, terlihat perbedaan yang cukup signifikan dalam performa dan stabilitas kontrol yang dihasilkan. Pada pengujian *CartPole-v1*, nilai *score* per episode menunjukkan performa yang cukup fluktuatif namun mampu mencapai nilai tinggi di beberapa episode, dengan rata-rata skor sekitar 186.20. Hal ini menunjukkan bahwa model DQN mampu mempelajari strategi keseimbangan batang dengan cukup baik, meskipun masih terdapat variasi performa akibat proses eksplorasi dan kondisi awal yang berbeda tiap episode. Semakin tinggi skor, semakin lama agen mampu menjaga keseimbangan batang, yang berarti kontrol yang dihasilkan semakin stabil.

Sementara itu, pada *Pendulum-v1*, hasil *reward* per episode berada pada kisaran -800 hingga -1200, menandakan bahwa agen masih mengalami kesulitan dalam mencapai kontrol optimal untuk menstabilkan sudut pendulum. Lingkungan Pendulum memiliki ruang aksi kontinyu yang lebih kompleks dibandingkan CartPole yang bersifat diskret, sehingga proses pembelajaran DQN menjadi lebih sulit dan cenderung tidak konvergen secara cepat.

6. Kesimpulan

- a) Algoritma *Deep Q-Network* (DQN) berhasil diterapkan pada *environment CartPole-v1* dengan hasil rata-rata skor yang tinggi, menunjukkan kemampuan agen dalam menjaga keseimbangan batang secara stabil.
- b) Pada *environment Pendulum-v1*, performa DQN masih rendah dengan nilai *reward* negatif yang besar, menandakan kesulitan agen dalam mengendalikan sistem dengan ruang aksi kontinyu.
- c) Perbedaan hasil menunjukkan bahwa DQN lebih efektif untuk sistem diskret dibanding sistem kontinyu yang membutuhkan kontrol presisi tinggi.
- d) Percobaan ini memberikan pemahaman mendalam tentang konsep *Deep Reinforcement Learning* (DRL), struktur DQN, dan penerapannya dalam sistem kontrol otomatis.

7. Saran

- a) Gunakan algoritma lain seperti PPO (*Proximal Policy Optimization*) atau DDPG (*Deep Deterministic Policy Gradient*) untuk mengatasi keterbatasan DQN pada ruang aksi kontinyu.
- b) Lakukan penyetelan hyperparameter (*learning rate, gamma, epsilon, batch size*) untuk meningkatkan stabilitas dan konvergensi model.
- c) Tambahkan perbandingan hasil antar algoritma guna memperkuat analisis performa kontrol.
- d) Uji model pada *environment* robotik yang lebih kompleks seperti *Reacher-v2* atau *Manipulator-v1* agar mendekati kondisi nyata.

8. Daftar Pustaka

- Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A., 2017. A Brief Survey of Deep Reinforcement Learning. *IEEE Signal Process. Mag.* 34, 26–38. <https://doi.org/10.1109/MSP.2017.2743240>
- Kiumarsi, B., Lewis, F.L., Hamidreza Modares, Karimpour, A., 2014. Reinforcement Q-learning for optimal feedback control. *IEEE Trans. Ind. Electron.*
- Kumar, A., Levine, A., Feizi, S., 2022. Policy Smoothing for Provably Robust Reinforcement Learning. <https://doi.org/10.48550/arXiv.2106.11420>
- Li, Y., 2018. Deep Reinforcement Learning: An Overview. <https://doi.org/10.48550/arXiv.1701.07274>
- 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, 529–533. <https://doi.org/10.1038/nature14236>