Jurnal Teknik Mesin S-1, Vol. 13, No. 3, Tahun 2025
Online: https://ejournal3.undip.ac.id/index.php/jtm

IMPLEMENTASI *REINFORCEMENT LEARNING* UNTUK STABILISASI SUDUT *PITCH* 90° PADA MODEL ROKET 6DOF DI MATLAB SIMULINK

***Cornelius Gian[1], Mochammad Ariyanto[2], Joga Dharma Setiawan[2]**
[1]Mahasiswa Jurusan Teknik Mesin, Fakultas Teknik, Universitas Diponegoro
[2]Dosen Jurusan Teknik Mesin, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Sudharto, SH., Tembalang-Semarang 50275, Telp. +62247460059
*E-mail: corneliusgiann@gmail.com

Abstrak

Perkembangan teknologi roket modern menuntut sistem kontrol yang mampu menjaga kestabilan roket secara presisi. Tantangan muncul akibat dinamika roket yang bersifat nonlinear dan kompleks sehingga metode kontrol konvensional kurang efektif. Penelitian ini bertujuan mengimplementasikan algoritma *Reinforcement Learning* (RL) khususnya *Twin Delayed Deep Deterministic Policy Gradient* (TD3), untuk mengendalikan defleksi *fin* dalam menstabilkan sudut *pitch* 90° pada model roket 6DoF di MATLAB Simulink. Metode penelitian meliputi persiapan model Simulink roket 6DoF, desain fungsi *reward*, pembuatan *environment* RL, pelatihan agen RL, serta pengujian performa agen melalui simulasi dengan gangguan angin. Hasil penelitian menunjukkan bahwa pada sistem tanpa RL, nilai *Mean Absolute Error* (MAE) untuk *gain* 1, 2, 3, 4, dan 5 berturut-turut adalah sebesar 0.6242°, 1.2483°, 1.8719°, 2.4949°, dan 3.1172°. Setelah implementasi RL, nilai MAE menurun menjadi 0.2770°, 0.3738°, 0.4351°, 1.2211°, dan 2.1156°. Sistem dengan RL menunjukkan peningkatan akurasi kontrol *pitch*. Hal ini membuktikan bahwa agen RL TD3 mampu mengatasi dinamika roket yang kompleks secara adaptif.

Kata kunci: defleksi *fin*; kontrol *pitch*; *reinforcement learning*; roket 6dof; td3

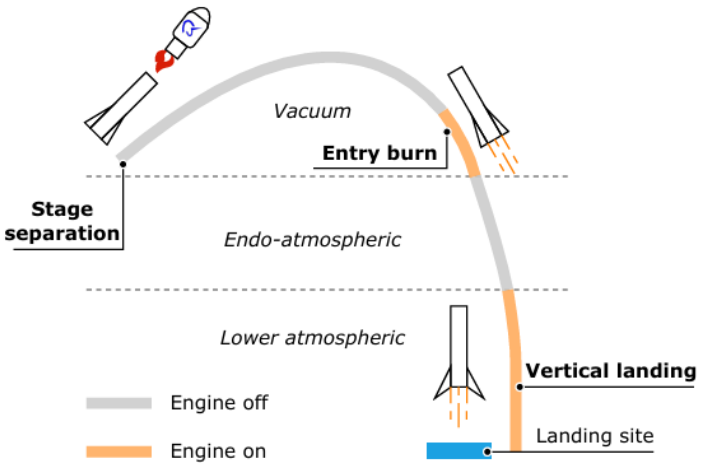
Abstract

The advancement of modern rocket technology demands a control system capable of maintaining rocket stability with high precision. Challenges arise due to the nonlinear and complex dynamics of rockets, making conventional control methods less effective. This study aims to implement a Reinforcement Learning (RL) *algorithm, specifically Twin Delayed Deep Deterministic Policy Gradient* (TD3)*, to control fin deflection for stabilizing the* 90° *pitch angle in a* 6DoF *rocket model using* MATLAB Simulink*. The research methodology includes preparing the 6DoF rocket model in Simulink, designing the reward function, developing the* RL *environment, training the* RL *agent, and testing the agent's performance through simulations under wind disturbances. The results show that in the system without* RL*, the Mean Absolute Error* (MAE) *for gains* 1, 2, 3, 4, *and* 5 *are* 0.6242°, 1.2483°, 1.8719°, 2.4949°, *and* 3.1172°*, respectively. After implementing* RL*, the* MAE *values decrease to* 0.2770°, 0.3738°, 0.4351°, 1.2211°, *and* 2.1156°*. The* RL*-based system demonstrates significant improvement in pitch control accuracy. This confirms that the* TD3 RL *agent is capable of adaptively handling the complex dynamics of rocket systems.*

Keywords: 6dof *rocket*; *fin deflection*; *pitch control*; *reinforcement learning*; td3

1. Pendahuluan

Perkembangan teknologi terus mengalami kemajuan pesat dari tahun ke tahun, termasuk teknologi roket. Roket pada umumnya memiliki dua fungsi utama yaitu untuk kepentingan sipil seperti peluncuran satelit dan eksplorasi ruang angkasa serta kepentingan militer seperti sistem pertahanan dan persenjataan. Banyak negara berinvestasi dalam jumlah besar dalam penelitian dan pengembangan teknologi roket untuk menciptakan roket yang lebih presisi, efisien, dan adaptif [1]. Salah satu inovasi signifikan dalam dunia aerospace adalah pengembangan reusable rocket atau roket yang dapat digunakan kembali sebagaimana ditunjukkan pada Gambar 1. Teknologi ini dikembangkan dengan tujuan utama untuk mengurangi biaya peluncuran secara signifikan, menghemat material, dan memungkinkan frekuensi peluncuran yang lebih sering [2].



Gambar 1. Diagram Fase Pendaratan *Reusable Rocket* [3]

Dalam pengembangan roket modern, sistem kontrol yang mampu mempertahankan stabilitas roket secara presisi sepanjang jalur lintasannya menjadi fokus utama. Pengendalian gerak yang akurat dan stabil menjadi tantangan dalam mengembangkan roket [4]. Hal ini disebabkan oleh dinamika penerbangan roket yang bersifat nonlinear seperti adanya kopling inersia dan karakteristik aerodinamika yang juga nonlinear [5]. Pembuatan model dinamika sistem yang representatif menjadi langkah awal yang krusial dalam perancangan sistem kontrol [6]. Namun, roket modern memiliki enam derajat kebebasan (6DoF) yang membuat model dinamikanya semakin kompleks. Kesalahan dalam kendali dapat berakibat fatal seperti dapat menyebabkan penyimpangan lintasan, kehilangan kendali, dan tabrakan hingga ledakan.

Dalam praktiknya, pendekatan pengendalian roket biasanya menggunakan sistem kontrol tradisional seperti PID *Controller* dan *Model Predictive Control* (MPC). Meskipun teknik ini efektif dalam sistem linear, keduanya sangat bergantung pada model dinamika sistem yang akurat agar berfungsi dengan baik, yang mana terdapat kesulitan untuk memodelkan dinamika sistem roket secara tepat [7]. Selain itu, *settling time* yang lama akibat penggunaan kontrol tradisional menjadi kendala karena roket yang melaju dengan kecepatan supersonik hingga hipersonik membutuhkan respon kontrol yang cepat dan presisi [8].

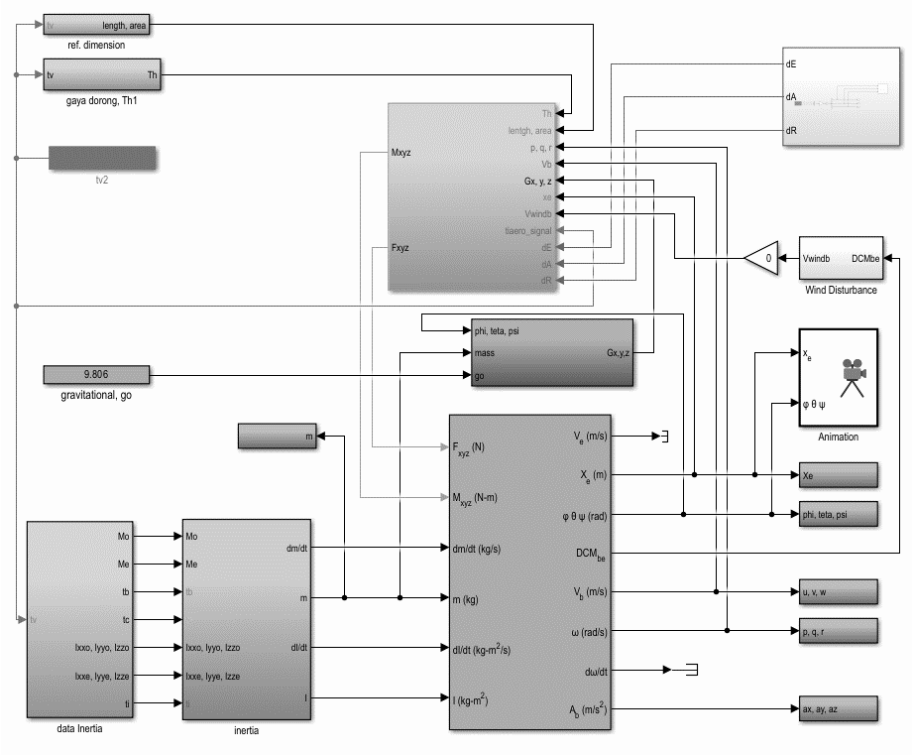
Salah satu pendekatan yang menjanjikan adalah *Reinforcement Learning* (RL). RL merupakan salah satu pengaplikasian *Artificial Intelligence* yang dirancang agar suatu sistem dapat belajar dan berkembang secara mandiri melalui mekanisme *trial* dan *error*. Dalam RL, agen mengambil aksi berdasarkan observasi terhadap lingkungan lalu menerima umpan balik berupa *reward* atau *punishment.* Melalui proses iteratif ini, agen mengumpulkan pengalaman secara bertahap untuk mempelajari kebijakan yang optimal untuk memaksimalkan *cumulative rewards* [3]. Keunggulan RL terletak pada kemampuannya dalam menangani dinamika yang kompleks dan bersifat nonlinear tanpa memerlukan model matematika yang dirumuskan sebelumnya sehingga cocok untuk diaplikasikan pada roket [9]. Selain itu, RL juga lebih fleksibel, responsif, dan adaptif terhadap perubahan lingkungan yang tidak terduga.

2. Bahan dan Metode Penelitian

Penelitian ini menggunakan roket nonlinear yang dimodelkan menggunakan Simulink seperti yang ditunjukkan pada Gambar 2 dengan spesifikasinya pada Tabel 1.

Tabel 1. Spesifikasi *Canard-Based Sounding Rocket*

Parameter	Nilai		Satuan
	Lift off	***Burn Out***	
Diameter	0.203	0.203	m
Jumlah *canard*	4	4	-
Jumlah *fin*	4	4	-
Massa	197.03	123.355	kg
I_{xx}	1.311	0.971	kg m^2
I_{yy}	201.264	153.906	kg m^2
I_{zz}	201.268	153.909	kg m^2



Gambar 2. Model Simulink Roket Nonlinear

Model Matematika

Persamaan gerak enam derajat kebebasan (6DoF) yang lengkap menggambarkan dinamika nonlinear dari penerbangan roket. Persamaan gerak differensial ini atau yang dikenal "*Equation of Motion*" (EOM) berasal dari hukum konservasi momentum linear dan momentum sudut yang dinyatakan melalui hukum Newton ke-2 untuk benda tegar.

Gerak translasi roket dinyatakan dalam bentuk persamaan gaya yang digambarkan pada sumbu x, y, dan z seperti dituliskan pada Persamaan (1) – (3).

$$-mg\sin\theta + F_x = m(\dot{u} + qw - rv) \quad (1)$$

$$mg\cos\theta\sin\varphi + F_y = m(\dot{v} + ru - pw) \quad (2)$$

$$mg\cos\theta\cos\varphi + F_z = m(\dot{w} + pv - qu) \quad (3)$$

Gerak rotasi roket dinyatakan dalam bentuk persamaan momen pada sumbu benda x, y, dan z seperti dituliskan pada Persamaan (4) – (6).

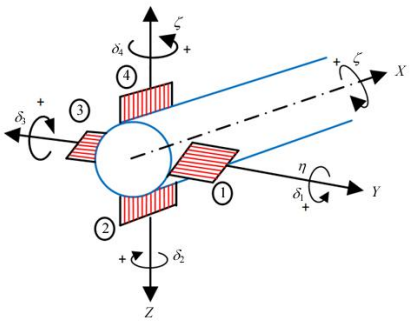
$$L = I_{xx}\dot{p} + (I_{zz} - I_{yy})qr \quad (4)$$

$$M = I_{yy}\dot{q} + (I_{xx} - I_{zz})pr \quad (5)$$

$$N = I_{zz}\dot{r} + (I_{yy} - I_{xx})pq \quad (6)$$

Control Surface

Sirip roket merupakan komponen kendali yang dapat digerakkan untuk menyesuaikan distribusi gaya aerodinamika pada roket sehingga memungkinkan perubahan arah hidung roket (*pitch, yaw, roll*) sesuai kebutuhan. Gambar 3 memperlihatkan mekanisme pergerakan sirip dengan empat sudut defleksi sirip dilambangkan dengan δ_1, δ_2, δ_3, dan δ_4.



Gambar 3. Mekanisme Kontrol Sirip [10]

Sirip-sirip tersebut disusun dalam konfigurasi *cruciform* di mana gerakan *roll, pitch*, dan *yaw* dikendalikan melalui defleksi keempat sirip yang hubungannya dinyatakan pada Persamaan (7) – (9) [11].

$$\delta_{ail} = \frac{1}{4}(\delta_1 + \delta_2 + \delta_3 + \delta_4) \quad (7)$$

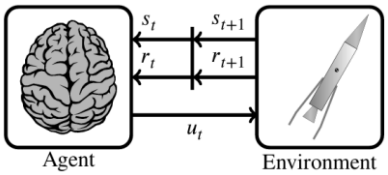
$$\delta_{elev} = \frac{1}{4}(\delta_1 + \delta_2 - \delta_3 - \delta_4) \quad (8)$$

$$\delta_{rud} = \frac{1}{4}(-\delta_1 + \delta_2 + \delta_3 - \delta_4) \quad (9)$$

***Twin Delayed Deep Deterministic Policy Gradient* (TD3)**

Reinforcement Learning (RL) adalah salah satu cabang dari *Machine Learning* yang memetakan situasi menjadi aksi di mana agen belajar mengambil keputusan optimal untuk memaksimalkan total *reward* dalam jangka panjang dengan cara berinteraksi dengan lingkungan [12]. RL memiliki kemampuan untuk mengambil keputusan kompleks, beradaptasi dengan lingkungan yang dinamis, dan belajar dari interaksinya berbasis *trial and error* [13]. Prinsip kerja dari RL berdasarkan Gambar 4 adalah sebagai berikut**:**

1. Agen menerapkan aksi (u_t) ke *environment* berdasarkan *state* di waktu t (s_t).
2. *Environment* menghasilkan *state* baru (s_{t+1}) dan mengirim umpan balik dalam bentuk *reward* yang kemudian digunakan oleh agen untuk melatih dirinya, mengumpulkan pengalaman, dan pengetahuan tentang *environment.*.
3. *State* dievaluasi berdasarkan *policy* untuk menentukan aksi yang akan diambil selanjutnya.



Gambar 4. Skema *Reinforcement Learning* [14]

Twin Delayed Deep Deterministic Policy Gradient (TD3) adalah algoritma *Reinforcement Learning* berbasis *actor-critic* yang dirancang untuk mengatasi kelemahan *overestimation* bias pada algoritma *Deep Deterministic Policy Gradient* (DDPG). TD3 menggunakan dua jaringan kritikus (*twin critics*) untuk menghitung nilai fungsi aksi, kemudian memilih nilai minimum dari kedua kritikus saat melakukan pembaruan *policy*. *Observation space* pada TD3 dapat berupa *continuous* atau *discrete*, sedangkan *action space* bersifat *continuous*.

Pada penelitian ini, *observation space* terdiri atas θ_{error}, θ, u, v, w, α, β, p, q, r, δ_{elev}, δ_{ail}, δ_{rud} dan *action space* mencakup δ_{elev}, δ_{ail}, δ_{rud} dengan rentang defleksi [-10°, 10°]. *Hyperparameter* agen yang digunakan ditunjukkan pada Tabel 2 berikut:

Tabel 2. *Hyperparameter* Agen

Hyperparameter	**Nilai**
Sample Time	0.01
Discount Factor	0.99
Mini Batch Size	256
Experience Buffer Length	1000000
Critic Learning Rate	0.001
Actor Learning Rate	0.0001
Target Smooth Factor	0.005
Target Update Frequency	2
Policy Update Frequency	2

Reward Function

Reward adalah umpan balik yang diberikan oleh *environment* berdasarkan suatu *state* setelah agen mengambil aksi. *Reward function* didesain berdasarkan tujuan utama kontrol, yaitu mempertahankan sudut *pitch* roket pada 90° selama fase terbang. Secara matematis, *reward* tersebut dituliskan pada Persamaan 10 sebagai berikut:

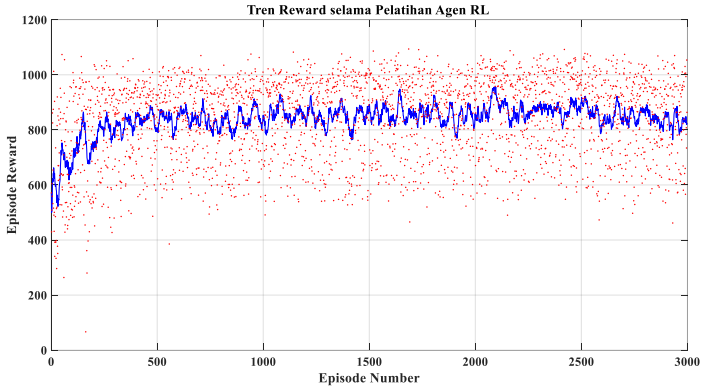
$$\mathbf{r = \exp(-150\theta_e^2) + 0.2_{|\theta_e|<0.00436 \wedge |q|<0.1} - 2p^2 - 250q^2} \quad \mathbf{(10)}$$

Reward ini bertujuan untuk:

- Mendorong agen meminimalkan *error* sudut *pitch* (θ_e).
- Memberi bonus *reward* tambahan sebesar 0.2 ketika sistem berada dalam zona stabil, yaitu saat $|\theta_e| < 0.00436$ rad dan $|q| < 0.1$ rad/s.
- Menekan osilasi dan dinamika tidak diinginkan dengan memberikan penalti terhadap kecepatan sudut *pitch* (q) dan *roll* (p) yang berlebihan yang dapat menyebabkan ketidakstabilan.

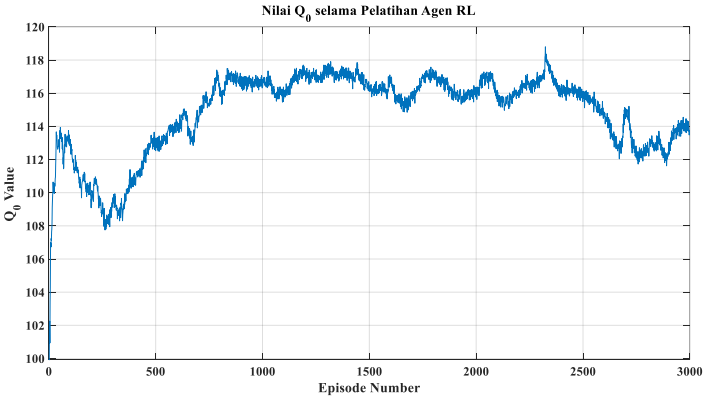
3. Hasil dan Pembahasan

Hasil Pelatihan Agen



Gambar 5. Tren *Reward* selama Pelatihan Agen RL

Gambar 5 menunjukkan tren reward selama proses pelatihan agen *Reinforcement Learning* (RL) selama 3000 episode. Secara umum, dapat diamati bahwa pada fase awal pelatihan (sekitar episode 0–500), terjadi peningkatan reward yang cukup signifikan, menandakan bahwa agen mulai mempelajari hubungan antara aksi yang diambil dan *feedback* yang diberikan oleh lingkungan. Namun, setelah fase awal tersebut, *reward* yang diperoleh agen cenderung berfluktuasi hingga episode terakhir. Meskipun rata-rata *reward* menunjukkan kecenderungan stabil di kisaran tertentu, agen belum menunjukkan tanda-tanda konvergensi penuh terhadap kebijakan optimal. Hal ini terlihat dari sebaran *reward* yang tetap tinggi, menandakan bahwa performa agen masih belum konsisten antar episode.



Gambar 6. Nilai Q_0 selama Pelatihan Agen RL

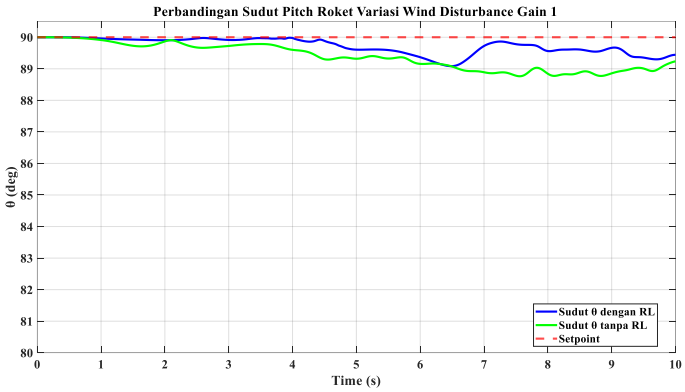
Gambar 6 memperlihatkan perkembangan nilai Q_0 terhadap jumlah episode selama proses pelatihan agen RL. Nilai Q_0 merepresentasikan estimasi fungsi nilai aksi (*action-value function*) pada kondisi awal setiap episode, yang mencerminkan ekspektasi reward kumulatif jika agen mengikuti kebijakan yang sedang dipelajari. Pada awal pelatihan (episode 0 hingga sekitar episode 500), nilai Q_0 menunjukkan fluktuasi cukup besar di kisaran 108–114. Setelah itu, terlihat tren kenaikan yang konsisten hingga episode ke-1000, di mana nilai Q_0 stabil pada rentang sekitar 116–118. Kenaikan ini menunjukkan bahwa agen berhasil mempelajari kebijakan yang lebih baik, ditandai dengan meningkatnya ekspektasi reward kumulatif. Namun, setelah episode ke-1500 hingga mendekati episode ke-3000, grafik memperlihatkan pola osilasi (fluktuasi naik turun) di sekitar nilai 114–118. Bahkan terdapat penurunan nilai Q_0 yang cukup tampak setelah episode ke-2500, di mana nilai Q_0 turun hingga mendekati 112 sebelum sedikit naik kembali mendekati akhir pelatihan. Fluktuasi nilai Q_0 ini dapat disebabkan oleh beberapa faktor, misalnya:

1. Proses eksplorasi agen yang masih aktif, sehingga kebijakan belum sepenuhnya stabil.
2. Kompleksitas dinamika sistem roket yang dilatih, yang menyebabkan agen perlu terus menyesuaikan kebijakan.
3. *Hyperparameter* pelatihan (*learning rate*, *noise*, dsb.) yang mungkin belum optimal.

Secara keseluruhan, grafik ini menunjukkan bahwa agen berhasil meningkatkan performanya dibanding kondisi awal. Namun, pola fluktuasi mengindikasikan pelatihan belum sepenuhnya konvergen atau stabil secara optimal. Hal ini menunjukkan perlunya evaluasi lebih lanjut terhadap desain *reward*, konfigurasi *hyperparameter*, atau strategi eksplorasi agar agen dapat mencapai kebijakan yang lebih stabil.

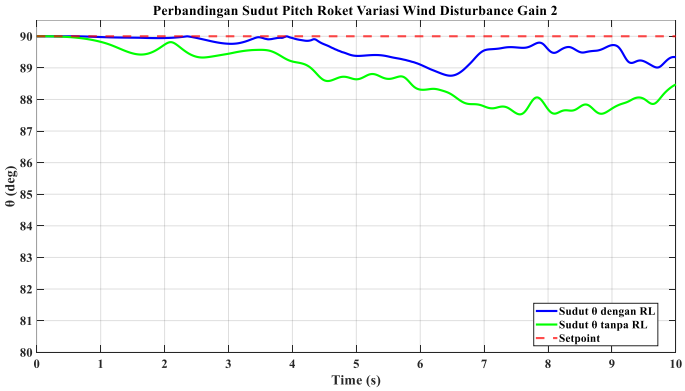
Hasil Pengujian Performa Agen

Simulasi roket dilakukan dengan sudut orientasi *roll*, *pitch*, dan *yaw* awal masing-masing sebesar 0, 90, dan 250 derajat.



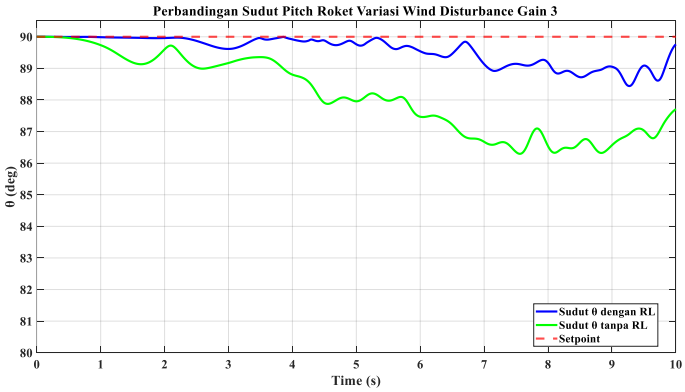
Gambar 7. Perbandingan Sudut *Pitch* Roket Variasi *Wind Disturbance Gain* 1

Gambar 7 menunjukkan perbandingan sudut *pitch* roket variasi *wind disturbance gain* 1. Pada level gangguan angin yang paling ringan, sistem tanpa RL masih mampu menjaga sudut *pitch* mendekati 90°, meskipun terjadi sedikit penurunan seiring waktu. Namun, sistem dengan RL menunjukkan performa yang lebih stabil dan konsisten. Garis biru (dengan RL) tetap berada di kisaran 89–90° sepanjang simulasi, menunjukkan bahwa agen RL sudah mampu memberikan koreksi halus terhadap gangguan ringan. Ini menandakan bahwa RL efektif bahkan dalam kondisi yang tidak terlalu ekstrem. Nilai MAE yang dihasilkan adalah 0.2770° dengan RL, sedangkan tanpa RL sebesar 0.6242°, mengindikasikan perbaikan akurasi kontrol *pitch* oleh sistem berbasis RL.



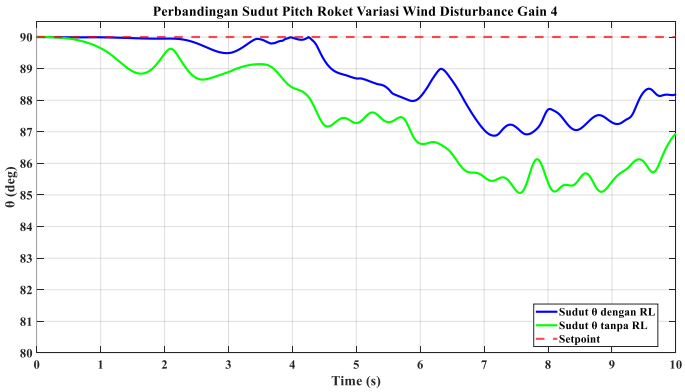
Gambar 8. Perbandingan Sudut *Pitch* Roket Variasi *Wind Disturbance Gain* 2

Gambar 8 menunjukkan perbandingan sudut *pitch* roket variasi *wind disturbance gain* 2. Ketika level gangguan meningkat, sistem tanpa RL mulai menunjukkan penurunan *pitch*, turun ke sekitar 88–89° secara bertahap. Sementara itu, sistem dengan RL masih mempertahankan stabilitas pitch lebih baik, tetap berada dekat 90° dengan sedikit fluktuasi. Ini mengindikasikan bahwa agen RL mampu mempertahankan orientasi vertikal lebih baik dibanding sistem tanpa kontrol. Nilai MAE yang dihasilkan adalah 0.3738° dengan RL, sedangkan tanpa RL sebesar 1.2483°, menunjukkan peningkatan akurasi kontrol *pitch* oleh sistem berbasis RL.



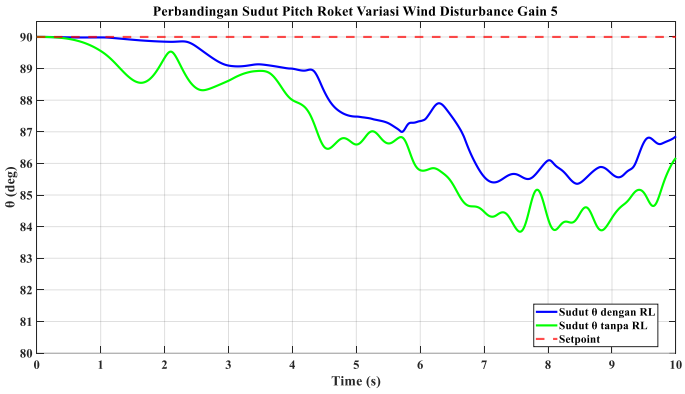
Gambar 9. Perbandingan Sudut *Pitch* Roket Variasi *Wind Disturbance Gain* 3

Gambar 9 menunjukkan perbandingan sudut *pitch* roket variasi *wind disturbance gain* 3. Pada gangguan tingkat sedang, sistem tanpa RL mulai kehilangan kendali arah pitch secara signifikan, dengan sudut menurun hingga di bawah 87° menuju 86°, dan terus menurun seiring waktu. Sebaliknya, sistem dengan RL tetap bertahan dalam kisaran 88–90° meskipun dengan fluktuasi yang sedikit lebih besar dibanding gain sebelumnya. Hal ini menunjukkan bahwa agen RL mulai bekerja lebih keras menghadapi gangguan yang lebih kuat, namun tetap efektif mempertahankan kestabilan. Nilai MAE yang dihasilkan adalah 0.4351° dengan RL, sedangkan tanpa RL sebesar 1.8719°, menunjukkan peningkatan akurasi kontrol *pitch* oleh sistem berbasis RL.



Gambar 10. Perbandingan Sudut *Pitch* Roket Variasi *Wind Disturbance Gain* 4

Gambar 10 menunjukkan perbandingan sudut *pitch* roket variasi *wind disturbance gain* 4. Pada level gangguan angin keempat, performa sistem tanpa kendali (garis merah putus-putus) mulai menunjukkan penurunan sudut *pitch* yang lebih dalam, turun dari sekitar 90° menjadi sekitar 85–86° selama simulasi 10 detik. Penurunan ini menunjukkan adanya pengaruh gangguan yang cukup signifikan terhadap kestabilan orientasi roket. Meskipun tidak sampai menyebabkan kehilangan kendali penuh, respon ini menunjukkan sistem pasif mulai kesulitan menjaga *pitch* mendekati nilai target. Sementara itu, sistem dengan kontrol RL tetap mempertahankan sudut *pitch* secara konsisten di kisaran 89–90° selama 4 detik awal, dengan fluktuasi kecil. Setelah itu, Setelah titik tersebut, kendali RL masih menunjukkan kemampuan adaptif dalam mengoreksi pergerakan pitch yang mulai menurun, sehingga sudut tetap terjaga dalam kisaran 87–89°. Meskipun terjadi sedikit penurunan performa akibat peningkatan gangguan, agen RL tetap berhasil menjaga kestabilan sistem secara signifikan lebih baik dibanding sistem tanpa kendali. Nilai MAE yang dihasilkan adalah 1.2211° dengan RL, sedangkan tanpa RL sebesar 2.4949°, menunjukkan peningkatan akurasi kontrol *pitch* oleh sistem berbasis RL.



Gambar 11. Perbandingan Sudut *Pitch* Roket Variasi *Wind Disturbance Gain* 5

Gambar 11 menunjukkan perbandingan sudut *pitch* roket variasi *wind disturbance gain* 5 yaitu level gangguan tertinggi yang diuji dalam simulasi ini. Pada kondisi tanpa kontrol (garis merah putus-putus), sudut *pitch* menunjukkan penurunan yang lebih signifikan dibanding level sebelumnya, yaitu dari sekitar 90° menjadi sekitar 85° dalam rentang waktu 10 detik. Sementara itu, sistem dengan kontrol RL hanya dapat mempertahankan sudut *pitch* secara konsisten di kisaran 89–90° selama kurang lebih 3 detik awal, dengan fluktuasi kecil. Setelah periode tersebut, respons mulai menunjukkan fluktuasi yang lebih besar, meskipun tetap berada dalam rentang yang lebih tinggi dibandingkan sistem tanpa kontrol. Hal ini menunjukkan bahwa meskipun performa agen RL sedikit menurun di bawah tekanan gangguan maksimum, sistem tetap memberikan kestabilan yang relatif lebih baik dan memperlambat penurunan sudut *pitch* secara signifikan dibanding sistem tanpa kontrol. Nilai MAE yang dihasilkan adalah 2.1156° dengan RL, sedangkan tanpa RL sebesar 3.1172°, menunjukkan peningkatan akurasi kontrol *pitch* oleh sistem berbasis RL.

4. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan untuk mengetahui performa kontrol berbasis *Reinforcement Learning,* yaitu TD3 dalam menstabilkan sudut *pitch* 90° dapat diambil kesimpulan bahwa agen belum optimal untuk menstabilkan sudut *pitch* pada 90°. Namun, hasil evaluasi menunjukkan bahwa RL mampu menstabilkan gerak *pitch* roket secara lebih baik dibandingkan sistem tanpa kendali RL, bahkan di bawah pengaruh gangguan angin dengan intensitas yang bervariasi. Pada sistem tanpa RL, nilai *Mean Absolute Error* (MAE) untuk *gain* 1, 2, 3, 4, dan 5 berturut-turut adalah sebesar 0.6242°, 1.2483°, 1.8719°, 2.4949°, dan 3.1172°. Setelah implementasi RL, nilai MAE menjadi 0.2770°, 0.3738°, 0.4351°, 1.2211°, dan 2.1156°. Nilai MAE yang lebih rendah pada sistem dengan RL menunjukkan peningkatan akurasi kontrol *pitch*.

5. Daftar Pustaka

[1] Ferro C, Cafaro M, Maggiore P. Optimizing Solid Rocket Missile Trajectories: A Hybrid Approach Using an Evolutionary Algorithm and Machine Learning. Aerospace 2024;11. https://doi.org/10.3390/aerospace11110912.
[2] Ferrante Reuben. A Robust Control Approach for Rocket Landing 2017:1–78.
[3] Jiang Y, Yang Y, Lan Z, Zhan G, Li SE, Sun Q, et al. Rocket Landing Control with Random Annealing Jump Start Reinforcement Learning 2024.
[4] Brötje S, Kirchner M, Giovannetti F. Performance and heat transfer analysis of uncovered photovoltaic-thermal collectors with detachable compound. Sol Energy 2018;170:406–18. https://doi.org/10.1016/j.solener.2018.05.030.
[5] Wada D, Araujo-Estrada SA, Windsor S. Unmanned Aerial Vehicle Pitch Control Using Deep Reinforcement Learning with Discrete Actions in Wind Tunnel Test. Aerospace 2021. https://doi.org/https://doi.org/10.3390/aerospace8010018.
[6] Kisabo AB, Adebimpe AF, Samuel SO. Pitch Control of a Rocket with a Novel LQG/LTR Control Algorithm. J Aircr Spacecr Technol 2019;3:24–37. https://doi.org/10.3844/jastsp.2019.24.37.
[7] Xue S, Wang Z, Bai H, Yu C, Li Z. Research on Self-Learning Control Method of Reusable Launch Vehicle Based on Neural Network Architecture Search. Aerospace 2024;11. https://doi.org/10.3390/aerospace11090774.
[8] Putro IE, Subiantoro A, Halim A, Triharjanto RH, Syafiie S. Optimal Control Design of Slow Dominant Transient Response for Longitudinal Missile Dynamics. 2023 IEEE Int Conf Aerosp Electron Remote Sens Technol ICARES 2023 2023:1–8. https://doi.org/10.1109/ICARES60489.2023.10329801.
[9] Iafrate D, Brandonisio A, Hinz R, Lavagna M. Propulsive landing of launchers' first stages with Deep Reinforcement Learning. Acta Astronaut 2025;227:40–56. https://doi.org/10.1016/j.actaastro.2024.11.028.
[10] Kisabo AB, Adebimpe AF, Okwo OC, Samuel SO. State-Space Modelling of a Rocket for Optimal Control System Design. J Aircr Spacecr Technol 2019;3:128–37. https://doi.org/10.3844/jastsp.2019.128.137.
[11] Kim S-H, Lee Y-I, Tahk M-J. New Structure for an Aerodynamic Fin Control System for Tail Fin-Controlled STT Missiles. J Aerosp Eng 2011;24:505–10. https://doi.org/10.1061/(asce)as.1943-5525.0000088.
[12] Chen Y, Ma L. Rocket powered landing guidance using proximal policy optimization. ACM Int Conf Proceeding Ser 2019. https://doi.org/10.1145/3351917.3351935.
[13] Srinivasan A. Reinforcement Learning: Advancements, Limitations, and Real-world Applications. Interantional J Sci Res Eng Manag 2023;07. https://doi.org/10.55041/ijsrem25118.
[14] Tevera-Ruiz A, Garcia-Rodriguez R, Parra-Vega V, Ramos-Velasco LE. Q-Learning with the Variable Box Method: A Case Study to Land a Solid Rocket. Machines 2023;11:1–14. https://doi.org/10.3390/machines11020214.