

Komparasi Metode Optimasi *Adam* dan *SGD* dalam Skema *Direct Inverse Control* untuk Sistem Kendali Data Sikap dan Ketinggian *Quadcopter*

MUHAMMAD SABILA HAQQI, BENYAMIN KUSUMOPUTRO

Departemen Teknik Elektro, Universitas Indonesia, Indonesia
Email: muhammad.sabila@ui.ac.id

Received 24 Februari 2022 | *Revised* 19 Maret 2022 | *Accepted* 22 Maret 2022

ABSTRAK

Terdapat banyak variable nonlinear dalam sistem kendali untuk quadcopter sehingga cukup rumit untuk mengatur dinamika penerbangan wahana ini. Untuk mengatasi masalah tersebut akan dikembangkan suatu skema sistem kendali Direct Inverse Control menggunakan Deep Learning berbasis Artificial Neural Network (ANN). Sistem yang dikembangkan akan mengendalikan data sikap dan ketinggian quadcopter. Pada artikel ini akan dibandingkan kinerja dari dua metode optimasi untuk Mean Squared Error pada simulasi, yaitu Adaptive Moment Estimation dan Stochastic Gradient Descent. Hasil menunjukkan metode Adaptive Moment Estimation mampu memberikan nilai Mean Squared Error yang lebih kecil dibandingkan metode Stochastic Gradient Descent untuk semua data sikap dan ketinggian yang dikendalikan dengan nilai 0.0069 untuk roll rate, 0.0057 untuk pitch rate, 0.0062 untuk yaw rate, dan 0.0042 untuk data ketinggian.

Kata kunci: *Deep Learning, Artificial Neural Network, Adam, SGD, MSE*

ABSTRACT

There are many nonlinear variables in the control system for the quadcopter so it is quite complicated to regulate the flight dynamics of this vehicle. To overcome this problem, a Direct Inverse Control control system scheme using Deep Learning based on Artificial Neural Network (ANN) will be developed. The system developed will control the attitude and altitude data of the quadcopter. In this article, we will compare the performance of two optimization methods for Mean Squared Error in simulation, namely Adaptive Moment Estimation and Stochastic Gradient Descent. The results show that the Adaptive Moment Estimation method is able to provide a smaller Mean Squared Error value than the Stochastic Gradient Descent method for all attitude and altitude data controlled with values of 0.0069 for roll rate, 0.0057 for pitch rate, 0.0062 for yaw rate, and 0.0042 for altitude data.

Keywords: *Deep Learning, Artificial Neural Network, Adam, SGD, MSE*

1. PENDAHULUAN

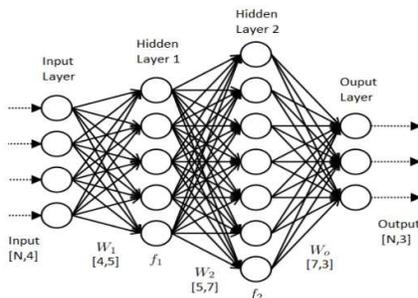
Studi terkait teknik pengendalian *quadcopter* semakin giat dilakukan karena aplikasinya yang sangat penting dalam sipil dan bidang militer. Saat ini, tujuan utama riset untuk pengendalian otonomus *quadcopter* adalah memperbaiki performa pengendalian *quadcopter* itu sendiri (**Kamanditya, dkk, 2020**). Mengendalikan *quadcopter* menjadi suatu hal yang sulit dilakukan karena banyaknya variabel nonlinear, *underactuated* (kurang mampu diperintahkan untuk mengikuti sembarang lintasan pada suatu ruang konfigurasi), dan banyaknya *cross-coupling* dari momen giroskopik motor (**Zhang, dkk, 2014**)(**Patro, dkk, 2015**)(**Navabi, dkk, 2016**).

Salah satu skema yang dapat digunakan dalam mengendalikan *quadcopter* adalah *Direct Inverse Control (DIC)*, dimana *DIC* akan mencari hubungan karakteristik antara sinyal masukan dan keluaran pada *plant* yang dikendalikan (**Kusumoputro, dkk, 2015**). Dalam penelitian ini penulis merepresentasikan skema sistem kendali data sikap *quadcopter* dengan metode *Deep Learning* berbasis *Artificial Neural Network (ANN)* karena saat ini *Deep Learning* merupakan metode pengenalan yang sangat menarik kinerjanya dibandingkan dengan metode *machine learning* lainnya (**Amiruddin, dkk, 2020**). Metode *Deep Learning* juga menjadi metode yang dapat diandalkan karena luasnya ketersediaan data serta kemampuan komputasi komputer yang tersedia (**Nguyen, dkk, 2019**). Nantinya, *Deep Learning* yang dibangun akan disusun sedemikian rupa sehingga terbentuk model dinamis yang direpresentasikan sebagai simulasi dari data penerbangan *quadcopter*.

Artificial Neural Network (ANN) adalah prosesor terdistribusi paralel yang terdiri dari neuron sederhana yang memungkinkan menghafal pengetahuan suatu sistem dengan meniru model nonliniernya setelah proses pelatihan (**Kusumoputro, dkk, 2016**). Metode optimasi pada *ANN* didasarkan pada *Gradient Descent*, dimana update bobot akan dihitung berdasarkan gradien dari loss function. Salah satu metode penurunan gradien yang paling banyak digunakan adalah *Stochastic Gradient Descent (SGD)* (**Ruder, 2016**). *SGD* dianggap sebagai metode optimasi yang efektif dan efisien untuk *Deep Learning* dalam beberapa tahun terakhir (**Deng, dkk, 2013**)(**Graves, dkk, 2013**)(**Krizhevsky, dkk, 2012**). Kemudian pada tahun 2015 diusulkan metode optimasi lain yang disebut *Adaptive Moment Estimation (Adam)* yang dianggap dapat memberikan hasil yang lebih baik dari *SGD* (**Kingma, dkk, 2014**). Dalam makalah ini, kami akan membandingkan pengaruh kedua metode optimasi terhadap kesalahan yang dihasilkan dalam mengendalikan data sikap *quadcopter*.

1.1 Artificial Neural Network

Artificial Neural Network (ANN) adalah arsitektur yang terdiri dari neuron-neuron yang saling berhubungan dengan nilai koneksi tertentu (**Kusumoputro, dkk, 2016**). Arsitektur jaringan saraf terinspirasi oleh model neuron pada otak manusia. *ANN* memiliki arsitektur seperti pada Gambar 1.



Gambar 1. Arsitektur ANN

ANN terdiri dari lapisan masukan, beberapa lapisan tersembunyi, dan lapisan keluaran. Setiap neuron pada suatu lapisan akan terhubung dengan neuron pada lapisan berikutnya dengan nilai bobot tertentu. Nilai suatu neuron merupakan penjumlahan *dot product* antara nilai bobot dengan nilai neuron lapisan sebelumnya yang kemudian akan dilewatkan oleh fungsi aktivasi non-linier. Ada beberapa jenis fungsi aktivasi yang dapat kita gunakan, dan fungsi aktivasi yang digunakan dalam penelitian ini adalah *Hyperbolic Tangent* (Tanh). Proses ini akan diulang sejumlah banyaknya lapisan tersembunyi yang digunakan. Terakhir, pada lapisan keluaran nilai neuron pada lapisan tersembunyi sebelumnya akan di-*dot product* dengan nilai bobot lapisan dan akan digunakan sebagai keluaran tanpa dimasukkan ke dalam fungsi aktivasi.

Terdapat dua tahap yang akan dilakukan dalam proses pelatihan *ANN*, yaitu *Forward Propagation* dan *Backward Propagation*. Dalam menentukan arsitektur optimal akan dilakukan beberapa siklus pelatihan yang disebut *epoch*, dimana dalam satu *epoch* pelatihan akan dilakukan tahap *Forward Propagation* dan *Backward Propagation* secara berurutan.

1.2 Forward Propagation

Forward Propagation merupakan tahap dimana nilai keluaran *ANN* akan ditentukan berdasarkan bobot neuron sementara. Untuk masing-masing model *ANN* yang akan dibuat, misalkan terdapat input yang dinotasikan seperti pada Persamaan (1).

$$I = [X_1 X_2 \dots X_n] \quad (1)$$

dimana I adalah vektor masukan, dan X_1, X_2, \dots, X_n adalah data masukan *ANN*.

Vektor masukan akan di-*dot product* dengan bobot lapisan tersembunyi pertama dan dilewatkan oleh fungsi aktivasi sehingga keluaran lapisan tersembunyi pertama ditunjukkan oleh Persamaan (2).

$$H_1 = \tanh(W_1 I + b_0) \quad (2)$$

dimana H_1, W_1, b_0 masing-masing adalah vektor keluaran lapisan tersembunyi pertama, bobot lapisan tersembunyi pertama, dan bias dari lapisan masukan.

Secara umum, keluaran lapisan tersembunyi ke- n serta lapisan keluaran ditunjukkan oleh Persamaan (3) dan (4), secara berurutan.

$$H_n = \tanh(W_n H_{n-1} + b_{n-1}) \quad (3)$$

$$Y = \tanh(W_{out} H_n + b_n) \quad (4)$$

dimana $H_n, W_n, W_{out}, b_n, Y$ masing-masing adalah vektor keluaran lapisan tersembunyi ke- n , bobot lapisan tersembunyi ke- n , bobot lapisan keluaran, bias dari lapisan tersembunyi ke- n , dan vektor keluaran *ANN*.

1.3 Backward Propagation

Backward Propagation merupakan tahap dimana nilai bobot dan bias pada setiap layer akan diperbaiki menggunakan metode *gradient descent*, dimana metode ini akan memperbaiki nilai bobot dan bias berdasarkan perubahan nilai *loss function* terhadap perubahan bobot lapisan dan bias. Dalam penelitian ini akan dibandingkan dua algoritma *gradient descent* sebagai

metode optimasi *ANN* yaitu *Stochastic Gradient Descent (SGD)* dan *Adaptive Moment Estimation (Adam)*.

1.3.1 *Stochastic Gradient Descent*

Stochastic Gradient Descent (SGD) akan memperbaiki nilai bobot dan bias dari setiap lapisan yang akan ditunjukkan dalam Persamaan (5) dan (6), secara berurutan

$$W_n = W_{n,t-1} - \alpha \frac{\partial J}{\partial W_n} \quad (5)$$

$$b_n = b_{n,t-1} - \alpha \frac{\partial J}{\partial b_n} \quad (6)$$

dimana $W_{n,t-1}$, $b_{n,t-1}$, J , α masing-masing adalah nilai bobot pada lapisan tersembunyi ke- n pada iterasi sebelumnya, nilai bias pada lapisan tersembunyi ke- n pada iterasi sebelumnya, nilai *loss function*, dan laju pembelajaran *ANN*. Nilai α merupakan *hyperparameter* yang dapat ditentukan secara manual dalam interval 0 hingga 1. Secara spesifik, *loss function* yang digunakan dalam penelitian ini adalah *Mean Squared Error* yang ditunjukkan dalam Persamaan (7).

$$MSE = \sum_{i=0}^N \frac{1}{2} (T_i - Y_i)^2 \quad (7)$$

dimana N, T_i masing masing adalah dimensi vektor keluaran *ANN* dan data target dalam proses pelatihan

Untuk mempercepat proses pelatihan, digunakan *hyperparameter* lainnya yaitu momentum. Momentum menggunakan sebagian dari nilai bobot sebelumnya dan perubahan bias untuk mengubah bobot dan nilai bias sehingga diharapkan pelatihan yang dilakukan akan lebih cepat. Perbaikan nilai bobot dan bias dari setiap lapisan tersembunyi menggunakan momentum ditunjukkan dalam Persamaan (8),(9),(10), dan (11) secara berurutan.

$$c_n = m(c_{n,t-1}) - \alpha \frac{\partial J}{\partial W_n} \quad (8)$$

$$d_n = m(d_{n,t-1}) - \alpha \frac{\partial J}{\partial b_n} \quad (9)$$

$$W_n = W_{n,t-1} - c_n \quad (10)$$

$$b_n = b_{n,t-1} - d_n \quad (11)$$

Dimana c_n, d_n masing-masing adalah perbaikan bobot lapisan tersembunyi ke- n dan perbaikan bias lapisan tersembunyi ke- n .

1.3.2 *Adaptive Moment Estimation*

Adaptive Moment Estimation (Adam) akan menghitung beberapa laju pembelajaran adaptif berdasarkan momentum pertama dan kedua dari *gradient* fungsi **(Kingma, dkk, 2014)**. Terdapat beberapa *hyperparameter* yang harus ditentukan terlebih dahulu, yaitu laju pembelajaran (α), *exponential decay rate* untuk estimasi momentum pertama (β_1) and kedua (β_2), sebuah angka yang sangat kecil (mendekati 0) untuk mencegah terjadinya *zero division* dalam implementasi metode (ϵ).

Secara umum, asumsikan parameter yang ingin dioptimalkan adalah θ_t . Selama θ_t belum konvergen, beberapa tahap akan dilakukan dalam proses perbaikan nilai parameter.

Gradient terkait fungsi tujuan parameter dalam *timestep* t akan diperbaiki seperti pada Persamaan (12).

$$g_t = \nabla_{\theta} f_t(\theta_{t-1}) \quad (12)$$

Kemudian, *update bias* untuk estimasi momentum pertama dan kedua dihitung seperti pada Persamaan (13) dan (14), secara berurutan

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (13)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t \quad (14)$$

Kemudian, *bias-corrected* untuk estimasi momentum pertama dan kedua dihitung seperti pada Persamaan (15) dan (16), secara berurutan.

$$\widehat{m}_t = \frac{m_t}{1 - \beta_{1,t}} \quad (15)$$

$$\widehat{v}_t = \frac{v_t}{1 - \beta_{2,t}} \quad (16)$$

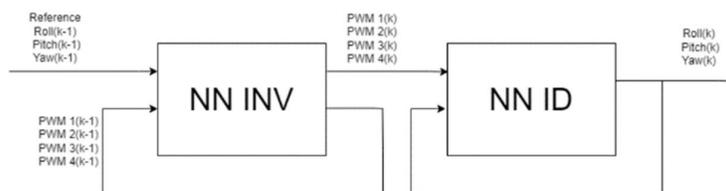
Setelah mendapat *bias-corrected* untuk kedua estimasi momentum, perbaikan parameter dapat dihitung seperti pada Persamaan (17).

$$\theta_t = \theta_{t-1} - \alpha \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \epsilon}} \quad (17)$$

2. METODE PENELITIAN

Pada penelitian ini akan dilakukan simulasi skema *DIC* yang diadaptasi dari penelitian Heryanto, dkk (**Heryanto, dkk, 2017**) untuk mengendalikan data sikap *quadcopter*. Terdapat 1608 data terbang yang digunakan dalam simulasi, dimana 80% data akan digunakan untuk proses pelatihan dan 20% data lainnya akan digunakan untuk proses pelatihan.

Secara garis besar, terdapat 2 blok *ANN* yang akan dilatih, yaitu *ANN INV* dan *ANN ID*. *ANN INV* akan berperan sebagai pengendali dari *plant quadcopter*, sedangkan *ANN ID* berperan sebagai identifikasi *plant quadcopter*. Terdapat 6 set eksperimen untuk setiap arsitektur dengan metode optimasi yang ditentukan, sehingga total 12 set eksperimen yang dilakukan. Di setiap set eksperimen, *ANN INV* dan *ANN ID* diberi waktu pelatihan dan penyetelan *hyperparameter* yang sama. Kemudian, seluruh data akan dinormalisasi menggunakan *Min-Max Normalization* sehingga seluruh data berada dalam interval -1 hingga 1. Blok diagram skema yang dibangun dapat dilihat pada Gambar 2.



Gambar 2. Blok Diagram Sistem Kendali Quadcopter

2.1 Arsitektur Data

Data quadcopter yang digunakan dalam penelitian ini mengacu pada data *quadcopter* yang dibangun oleh kelompok riset *Computational Intelligence and Intelligent Systems (CIIS)* Universitas Indonesia untuk keperluan penelitian mereka. *Quadcopter* ini memiliki berat 1,2 kg. *Quadcopter* ini menggunakan gabungan dari motor *BLDC T-Motor 940 KV* dan *carbon propeller* berukuran 12" x 5" sebagai aktuatornya. Selain itu, digunakan pula *T-Motor 18 A Electronic Speed Control* yang diletakkan di setiap sudut *quadcopter*. *Quadcopter* ditenagai oleh baterai berjenis *Lithium-Polymer 14.8 V*. *Quadcopter* ini menggunakan *Ardupilot 2.5* sebagai *flight control unit (FCU)*, dimana *Ardupilot 2.5* merupakan *open source FCU* yang terdiri dari mikrokontroler dan alat pengukur inersia seperti *gyroscope, accelerometer, dan barometer*. Konfigurasi ini dapat menghasilkan *thrust* ideal sebesar 2.6 kg untuk memungkinkan *quadcopter* melakukan maneuver dengan mudah. Data yang digunakan dalam penelitian ini adalah data sikap *quadcopter* berupa kecepatan sudut (*pitch, roll, dan yaw rate*), data ketinggian *quadcopter*, dan *PWM* masing-masing motor.

Phase Width Modulation (PWM) adalah suatu metode untuk mengendalikan perangkat analog menggunakan sinyal digital (**Muliadi, dkk, 2018**). Karena sistem digital hanya memiliki status *ON* dan *OFF*, *PWM* digunakan untuk menghasilkan sinyal analog yang dapat diterima oleh perangkat yang ingin kita kendalikan.

Besarnya sinyal *PWM* dapat dikontrol melalui besarnya *duty cycle* yang digunakan. *Duty cycle* adalah persentase periode dimana sinyal digital berada pada posisi *ON* selama periode tertentu, dan proses ini akan diulang berkali-kali sehingga sinyal dengan tegangan antara nilai tegangan terkecil dan maksimum dari sinyal digital akan timbul. Besarnya tegangan yang dihasilkan dapat direpresentasikan secara matematis dengan menggunakan Persamaan (18).

$$V = \text{duty cycle} \times V_{max} \quad (18)$$

Dengan mengatur *duty cycle*, akan mampu membangkitkan sinyal yang nantinya digunakan untuk mengendalikan *quadcopter*.

2.2 Pelatihan Sistem Kendali

Karena terdapat 2 blok *ANN* yang dibangun, terdapat dua model *ANN* yang akan dilatih, yaitu *ANN INV* dan *ANN ID*

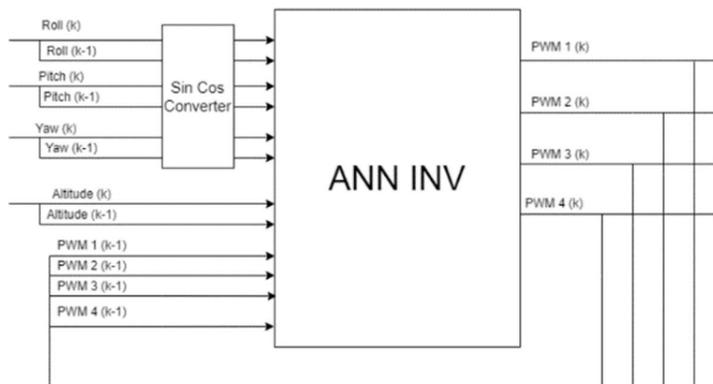
2.2.1 Pelatihan *ANN INV*

ANN INV akan memprediksi sinyal kendali dari *quadcopter* berupa *PWM1, PWM2, PWM3, dan PWM4* berdasarkan data sikap dan ketinggian *quadcopter* serta sinyal kendali pada iterasi sebelumnya. Data sikap *quadcopter* akan masuk terlebih dahulu ke *Sin-Cos Converter* sebelum masuk ke *ANN INV*. Pada prinsipnya, *Sin-Cos Converter* akan merubah data sikap yang berupa data sudut menjadi dua data seperti ditunjukkan pada Gambar 3. *Converter* ini akan mengubah data sikap *quadcopter* menjadi lebih *smooth*.



Gambar 3. Sin-Cos Converter

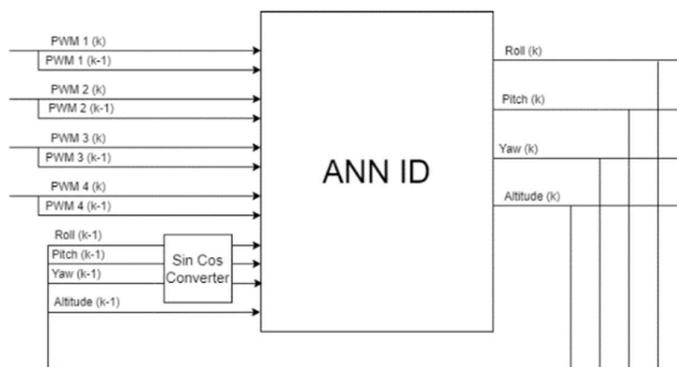
Skema pelatihan *ANN INV* ditunjukkan pada Gambar 4.



Gambar 4. Skema Pelatihan ANN INV

2.2.2 Pelatihan ANN ID

ANN ID akan memprediksi data sikap serta ketinggian *quadcopter* berdasarkan sinyal kendali yang diberikan kepada *quadcopter* serta data sikap dan ketinggian pada iterasi sebelumnya. Data sikap *quadcopter* pada iterasi sebelumnya akan masuk terlebih dahulu ke *Sin-Cos Converter* sebelum masuk ke *ANN ID*. Skema pelatihan *ANN ID* ditunjukkan pada Gambar 5.



Gambar 5. Skema Pelatihan ANN ID

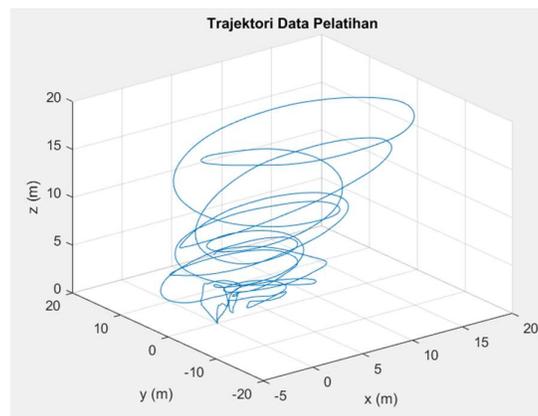
2.2.3 Pengujian DIC

Setelah pelatihan *ANN* selesai dilakukan, diasumsikan bahwa bobot neuron yang sudah dilatih merupakan bobot neuron optimal yang akan digunakan untuk memprediksi data *quadcopter*.

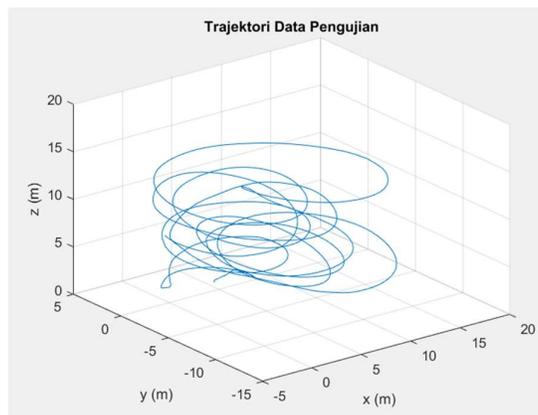
Menggunakan data uji yang sudah disiapkan sebelumnya, akan dilihat bagaimana kemampuan skema *DIC* yang dibangun dalam mengendalikan *quadcopter*.

3. HASIL DAN PEMBAHASAN

Data *quadcopter* yang digunakan dalam penelitian merupakan data terbang *quadcopter* dengan dinamika terbang *real-time* dan diterbangkan berdasarkan beberapa jenis trajektori lintasan. Untuk data pelatihan digunakan lintasan berjenis *clockwise-up helix*, *clockwise-down helix*, dan *counter clockwise-up helix*, sedangkan data pengujian menggunakan lintasan berjenis *counter clockwise-down helix*. Akhirnya didapat trajektori pelatihan dan pengujian yang ditunjukkan pada Gambar 6 dan Gambar 7. Gambar 6 menunjukkan trajektori data terbang yang akan digunakan sebagai data pelatihan, sedangkan Gambar 7 akan digunakan sebagai data pengujian. Semua pelatihan dan pengujian dilakukan menggunakan lingkungan simulasi yang dibangun dengan platform *Anaconda Navigator 3* dengan menggunakan beberapa *package* didalamnya, seperti *tensorflow 2.3.0*, *pandas 1.2.3*, *numpy 1.19.2*, dan *scikit-learn 0.24.1*.



Gambar 6. Trajektori Data Pelatihan



Gambar 7. Trajektori Data Pengujian

Proses pelatihan dilakukan sebanyak 5000 *epoch* untuk setiap set percobaan. Setelah selesai 5000 *epoch* didapat nilai *MSE* untuk masing-masing data pelatihan yang telah dinormalisasi seperti ditunjukkan pada Tabel 1. Nilai *MSE* ini didapat setelah melakukan penyetelan *hyperparameter* terhadap metode optimasi *SGD* dan *Adam* seperti ditunjukkan pada Tabel 2 dan Tabel 3, secara berurutan.

Dari Tabel 1 dapat dilihat bahwa metode optimasi *Adam* mampu memberikan *MSE* yang lebih kecil dibandingkan dengan metode *SGD*, baik dalam pelatihan sinyal kendali yang harus diberikan maupun data sikap serta ketinggian *quadcopter*. Selama penelitian didapat pula bahwa dalam kondisi *epoch* yang lebih rendah ($epoch < 1000$) *MSE* yang dihasilkan untuk setiap jenis data *quadcopter* lebih besar dari 0.01 sehingga sistem masih belum mampu merepresentasikan dinamika *quadcopter* dengan baik.

Tabel 1. Perbandingan *MSE* pada Data Pelatihan (*Normalized*)

Metode Optimasi	Data Quadcopter				
	PWM	Roll	Pitch	Yaw	Z
<i>SGD</i>	0.0078	0.0068	0.0052	0.0015	0.0026
<i>Adam</i>	0.0069	0.0064	0.0047	0.0011	0.0020

Tabel 2. *Hyperparameter* untuk Metode Optimasi *SGD*

<i>Hyperparameter</i>	Nilai
<i>Learning Rate</i>	0.10
Momentum	0.01

Tabel 3. *Hyperparameter* untuk Metode Optimasi *Adam*

<i>Hyperparameter</i>	Nilai
<i>Learning Rate</i>	1e-5
<i>1st Estimate Moment</i>	0.9
<i>2nd Estimate Moment</i>	0.99
Epsilon	1e-7

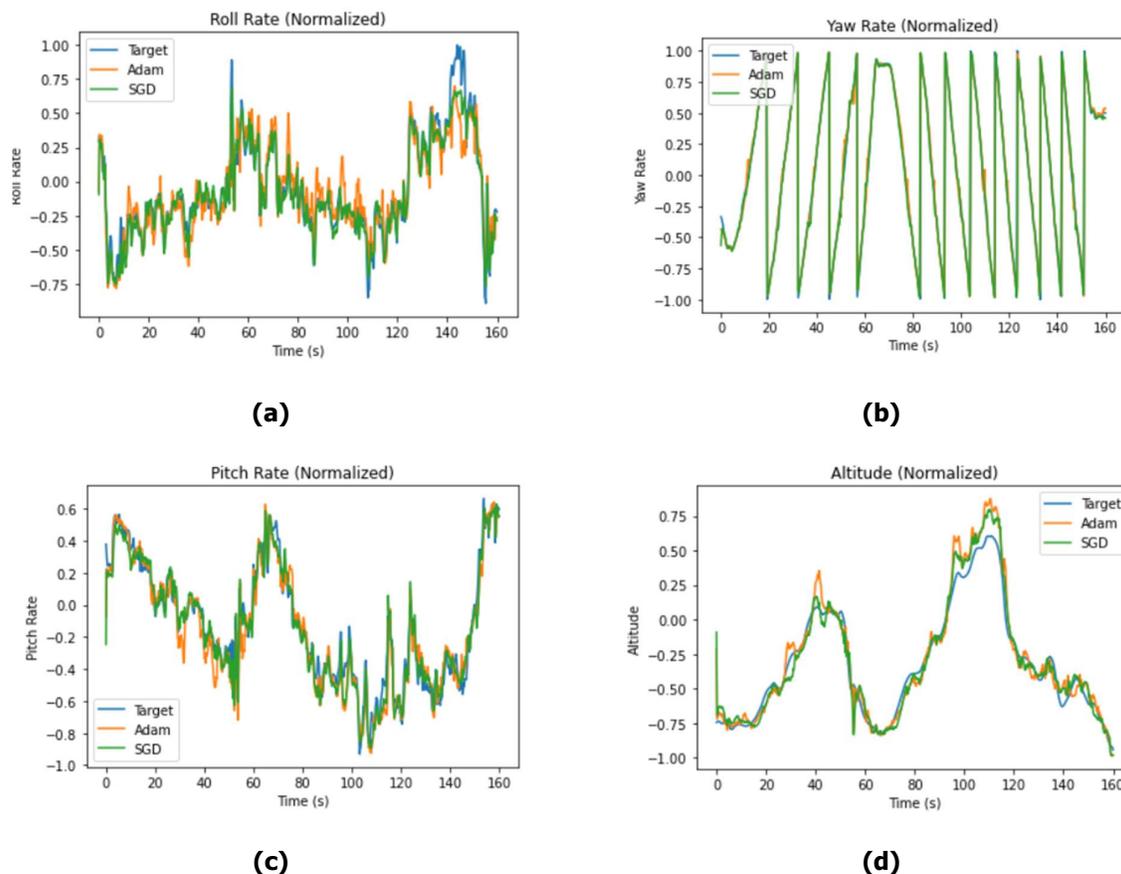
Setelah melakukan tahap pelatihan, sistem yang sudah memiliki bobot neuron yang konvergen akan diuji menggunakan data pengujian. Tabel 4 menunjukkan *MSE* yang didapat pada masing-masing data uji *quadcopter*.

Tabel 4. Perbandingan *MSE* pada Data Pengujian (*Normalized*)

Metode Optimasi	Data Quadcopter			
	Roll	Pitch	Yaw	Z
<i>SGD</i>	0.0070	0.0061	0.0077	0.0045
<i>Adam</i>	0.0069	0.0057	0.0062	0.0042

Terlihat dari Tabel 4 bahwa secara umum skema *DIC* yang dibuat telah mampu mengendalikan data sikap dan ketinggian *quadcopter*. Terlihat pula bahwa seperti pada saat pelatihan, metode optimasi *Adam* mampu memberikan nilai *MSE* yang lebih kecil dibandingkan metode optimasi *SGD*.

Komparasi Metode Optimasi *Adam* dan *SGD* dalam Skema *Direct Inverse Control* untuk Sistem Kendali Data Sikap dan Ketinggian *Quadcopter*



Gambar 8. Komparasi Metode Optimasi pada Tahap Pengujian untuk Data *Roll Rate* (a), *Pitch Rate* (b), *Yaw Rate* (c), dan Ketinggian *quadcopter* (d)

Gambar 8 menunjukkan hubungan antara prediksi metode optimasi *Adam* dan *SGD* terhadap data terbang *quadcopter* yang diujikan. Dapat dilihat dari Gambar 8 bahwa grafik prediksi untuk data sikap serta ketinggian *quadcopter* sudah dapat mengikuti dinamika terbang *quadcopter* secara real time. Hal ini sejalan dengan nilai *MSE* yang relatif kecil (nilai *MSE* lebih kecil dari 0.01) seperti terdapat di Tabel 4. Kemudian, *epoch* yang relatif tinggi dalam proses pelatihan agar memperoleh *MSE* yang kecil menunjukkan bahwa dinamika *quadcopter* relatif sulit dikendalikan. Hal ini terjadi karena dalam data terbang yang dijadikan sebagai patokan, data sikap *quadcopter* cenderung fluktuatif dan berubah relatif cepat.

4. KESIMPULAN

Dari penelitian ini telah ditunjukkan bagaimana membangun sistem kendali *quadcopter* untuk mengendalikan data sikap dan ketinggian *quadcopter* menggunakan skema *Direct Inverse Control*. Simulasi dilakukan menggunakan data *quadcopter* yang diterbangkan berdasarkan beberapa jenis trajektori. Selain itu, berdasarkan *MSE* yang didapat dari tahap pelatihan dan pengujian, ditemukan bahwa metode optimasi *Adam* memiliki performa yang lebih baik dalam memprediksi data sikap dan ketinggian *quadcopter*. Nilai *MSE* yang didapat melalui metode *Adam* adalah 0.0069 untuk *roll rate*, 0.0057 untuk *pitch rate*, 0.0062 untuk *yaw rate*, dan 0.0042 untuk data ketinggian. Penyetelan *hyperparameter* untuk metode *Adam* yang menghasilkan nilai *MSE* diatas adalah $\alpha = 10^{-5}$, $\beta_1 = 0.9$, $\beta_2 = 0.99$, dan $\varepsilon = 10^{-7}$.

DAFTAR RUJUKAN

- Amiruddin, B. P., Iskandar, E., Fatoni, A., & Santoso, A. (2020). Deep Learning based System Identification of Quadcopter Unmanned Aerial Vehicle. *3rd International Conference on Information and Communications Technology (ICOIACT)*, (pp. 165-169). IEEE.
- Deng, L., Li, J., Huang, J. T., Yao, K., Yu, D., Seide, F., ... & Acero, A. (2013). Recent advances in deep learning for speech research at Microsoft. *IEEE International Conference on Acoustics, Speech and Signal Processing*, (pp. 8604-8608). IEEE.
- Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.
- Heryanto, M., Suprijono, H., Suprpto, B. Y., & Kusumoputro, B. (2017). Attitude and altitude control of a quadcopter using neural network based direct inverse control scheme. *Advanced Science Letters*, *23*(5), 4060-4064.
- Kamanditya, B., & Kusumoputro, B. (2020). θ -D Elman Recurrent Neural Networks Based Direct Inverse Control for Quadrotor Attitude and Altitude Control. *International Conference on Intelligent Engineering and Management (ICIEM)*, (pp. 39-43).
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.
- Kusumoputro, B., Priandana, K., & Wahab, W. (2015). System identification and control of pressure process rig system using Backpropagation Neural Networks. *ARPJ Journal of Engineering and Applied Sciences*, *10*(16), 7190-7195.
- Kusumoputro, B., Suprijono, H., Heryanto, M. A., & Suprpto, B. Y. (2016). Development of an attitude control system of a heavy-lift hexacopter using Elman recurrent neural networks. *22nd International Conference on Automation and Computing (ICAC)*, (pp. 27-31). IEEE.
- Muliadi, J., & Kusumoputro, B. (2018). Neural network control system of UAV altitude dynamics and its comparison with the PID control system. *Journal of Advanced Transportation*
- Navabi, M., & Mirzaei, H. (2016). θ -D Based Nonlinear Tracking Control of Quadcopter. *4th International Conference on Robotics and Mechatronics (ICROM)*, (pp. 331-336).
- Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., Lopez Garcia, A., Heredia, I., & Hluchý, L. (2019). Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review*, *52*(1), 77-124.

Patro, S., & Sahu, K. K. (2015). Normalization: A preprocessing stage. arXiv 2015. arXiv preprint arXiv:1503.06462.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Zhang, X., Li, X., Wang, K., & Lu, Y. (2014) A survey of modelling and identification of quadrotor robot. *Abstract and Applied Analysis*. Hindawi.